УДК 004.056

Н. Е. Изъюров[1]*

Разработка инструмента пассивного анализа трафика вычислительной сети

[1] Национальный исследовательский ядерный университет «МИФИ», г. Москва, Российская федерация
* email: nicolayizyurov@gmail.com

Аннотация. В статье описана разработка программного инструмента для идентификации информационных потоков и декапсуляции трафика в сетях, функционирующих на основе стека протоколов *TCP/IP* версии 4 и использующих технологию канального уровня *Ethernet*. Минимальное жизнеспособное решение, реализованное в данной работе, применяет виртуальную машину ядра *eBPF*, библиотеку скомпилированных выражений *Lpcap*, программный интерфейс типа raw socket. Дополнительная функциональность инструмента позволяет расширить возможности фильтрации как входящего, так и исходящего трафика. Особенностью разработанного средства является возможность загрузки в основную программу сетевого анализатора внешних фильтрующих выражений, исходники которых написаны на языке программирования *C* или *eBPF Assembler*. Область применения инструмента — обеспечение сетевой информационной безопасности корпоративных вычислительных сетей, в частности, конечных узлов сети под управлением операционных систем *GNU/Linux*.

Ключевые слова: анализатор трафика, пакетный фильтр, сетевая безопасность, пассивный анализ трафика*, eBPF*, *tcpdump*.

N. E. Izyurov[1]*

Development of a Tool for Passive Analysis of Computer Network Traffic

[1] National Nuclear Research University "MEPhI", Moscow, Russian Federation
* email: nicolayizyurov@gmail.com

Abstract. The paper describes the development of a software tool for information flow identification and traffic decapsulation in networks based on *TCP/IP* protocol stack (version 4) and using *Ethernet* link layer technology. The minimum viable solution implemented in this paper uses the *eBPF* kernel virtual machine, the *Lpcap* compiled expressions library, and a raw socket type program interface. Additional functionality of the tool allows to extend the filtering capabilities of both incoming and outgoing traffic. The specific feature of the developed tool is the ability to load external filtering expressions, source code of which is written in *C* or *eBPF Assembler* programming language, into the main program of the network analyzer. The field of application of the tool is to provide network security of corporate networks, particularly of the network endpoints under *GNU/Linux* operating systems.

Keywords: traffic analyzer, packet filter, network security, passive traffic analysis, e*BPF, tcpdump*

Введение

Атаки на сетевую инфраструктуру частных компаний и государственных учреждений с каждым годом становятся всё изобретательнее и продолжают

наносить организациям финансовый и репутационный ущерб. В качестве примера приводится отчет вендора информационной безопасности (ИБ) *Acronis* от 2022 года, в котором общемировой ущерб от действий шифровальщиков оценивается в 20 миллиардов долларов США [1]. Другим показательным фактором выступает статистика, содержащаяся в бюллетне «Кибератаки на российские компании в 2022 году» от провайдера сервисов и услуг ИБ «РТК-Солар»: за первое и второе полугодие 2022 года было выявлено 416 и 495 тысяч событий информационной безопасности соответственно, причем доля подтвержденных инцидентов возросла на 71% [2]. Приведенные данные означают недостаточную защищенность сетевой инфраструктуры у региональных компаний и крупных корпораций как в Российской Федерации, так и в зарубежных государствах.

Регулярно проводятся атаки на сетевую инфраструктуру государственных учреждений: так, в 2021 году в России им массировано подверглись информационные ресурсы Министерства Обороны, Министерства Цифрового Развития, портал Госуслуг. Стоит отметить, что Национальный координационный центр по компьютерным инцидентам (НКЦКИ) в феврале 2022 года оценил уровень угрозы кибератак на российские информационные ресурсы как критический, и по состоянию на конец апреля 2023 года введенный режим сохраняется. Другим иллюстративным примером является статистика из доклада Министерства Внутренних Дел Российской Федерации «Состояние преступности в России» за январь-февраль 2023 года, в котором было зарегистрировано на 17% больше преступлений с использованием информационно-коммуникационных технологий и в сфере компьютерной информации, чем за аналогичный период 2022 года [3].

Сетевое направление информационной безопасности основывается на комплексном анализе информационных потоков с декапсуляцией сетевых пакетов. Возрастание объема зашифрованных данных и количественное увеличение трафика по сей день создают проблемы наблюдаемости и управляемости в процессе мониторинга сетевой активности. Средства комплексной сетевой безопасности вследствие вышеописанных тенденций становятся малоэффективными и усложняются в развертывании и обслуживании. Из инструментов подобного класса выступают агенты первичного сбора данных, основная задача которых заключается в предоставлении исчерпывающей информации о сетевой инфраструктуре предприятия [4]. В частности, такие сведения могут использоваться для обнаружения угроз безопасности в корпоративной сети предприятия. В данной статье предлагается спроектировать и разработать пассивный анализатор трафика вычислительной сети, функционирующий на канальном уровне стека протоколов *TCP/IP* версии 4, способный идентифицировать информационные потоки и декапсулировать сетевой трафик в операционных системах *GNU/Linux*. Разработанное решение предполагается использовать как в корпоративных сетях предприятий, так и в качестве средства профилактики безопасности для персональных компьютеров.

Постановка задачи

Целью настоящей работы является идентификация уязвимостей информационной инфраструктуры предприятия посредством пассивного анализа трафика вычислительной сети. В ходе достижения данной цели выделены следующие задачи:

– исследование предметной области научной работы;
– проектирование сетевого анализатора;
– разработка программного продукта;
– тестирование разработанного инструмента.

Исследование предметной области

На сегодняшний день подавляющее большинство вычислительных сетей основываются на стеке протоколов *TCP/IP*. Данная сетевая модель является фундаментом глобальной сети Интернет. Важно отметить, что исходной мотивацией для создания данного стека протоколов являлась необходимость практического обеспечения обмена информацией между разными узлами в компьютерной сети. Соответственно, безопасность как важный аспект взаимодействия в процессе разработки технологии в приоритет не ставилась, что привело к недостаточной защищенности от злонамеренных действий. Ниже приводятся некоторые существенные уязвимости стека протоколов *TCP/IP*:

– существует угроза перехвата потока трафика и его последующий анализ третьими лицами. В терминах специалистов по информационной безопасности подобные атаки именуются подслушиванием — listening (англ. — слушать). Одним из возможных вариантов обеспечения защиты от атак подслушивания выступают методы множественного (инкапсулированного) шифрования информационных потоков [5];

– в базовых протоколах стека *TCP/IP* как правило не предусмотрено шифрование данных и проверка подлинности субъекта взаимодействия в полном объеме и надлежащем качестве;

– общая уязвимость физических устройств, функционирующих на основе данной технологии перед атаками, направленными на перегрузку сетевой аппаратуры: серверов, маршрутизаторов, домашних компьютеров. Они именуются атаками наполнением — *flood*-атаками (англ. *flood* — наполнение).

Защиту от сетевых атак на аппаратно-программном и программном уровне обеспечивают комплексные системы с широким кругом задач обеспечения безопасности, которые предполагают охват всех уровней модели взаимодействия открытых систем *OSI* (англ. *Open Systems Interconnection*). Подобные продукты могут состоять из разных составных частей, однако особый интерес в контексте научной работы вызывают подсистемы (системы) анализа сетевого трафика *NTA* (англ. *Network Traffic Analysis*), предназначенные для идентификации потоков данных и обнаружения признаков комплексных целенаправленных атак *APT* (англ. *Advanced Persistent Threat*). Устройства подобного класса предоставляют возможность проведения ретроспективных исследований сетевых инцидентов,

способны детектировать и расследовать действия злоумышленников в корпоративной информационной инфраструктуре, позволяя оперативно реагировать на исследуемые инциденты. *NTA* позволяют добиться большей эффективности при условии эксплуатации совместно с системами обнаружения и реагирования на конечные точки *EDR (*англ. *Endpoint Detection and Response)*. Дополнительно системы анализа трафика используются в качестве доверенного и всестороннего источника информации для *SIEM*-систем (англ. *Security Information and Event Management*) или центров мониторинга и быстрого реагирования на инциденты информационной безопасности *SOC* (англ. *Security Operation Center*) [6]. Как правило, типовая архитектура *NTA*-систем состоит из сетевого сенсора, собирающего трафик с информационных потоков сети, серверов централизованного управления и консоли мониторинга [7].

Инструмент пассивного анализа трафика, выступающий в качестве сетевого сенсора в подобных системах, при реализации собственной функциональности позволяет:

– предоставлять актуальную информацию о проходящих сетевых пакетах для своевременного обнаружения атак;

– сохранять проходящий трафик в файлы дампа (*.pcap, .pcapng* и иные) для обеспечения возможности его анализа в дальнейшем в контексте применения в компьютерной криминалистике.

Стоит отметить, что мониторинг сети, основанный на идентификации информационных потоков и декапсуляции трафика, является одним из наиболее актуальных и эффективных методов, применяемых системными администраторами и аналитиками безопасности для обнаружения угроз в локальной сети. Однако в условиях высокой нагруженности исследуемых сетей он занимает существенные системные ресурсы. Так, для повышения скорости обработки сетевых пакетов в работе ученых *Vazquez, Felix & Ferreira* предлагается совместное использования raw-сокетов и программной библиотеки *scapy* для манипулирования сетевыми пакетами на языке программирования *Python* [8].

В контексте применения в компьютерной криминалистике анализаторы трафика подробно описываются в статье *Leslie F. Sikos* [9], в которой дополнительно приводятся варианты для повышения их эффективности, начиная от введения элементов машинного обучения в процесс идентификации сетевых потоков и заканчивая разработкой новых алгоритмов для сетей с высокой долей зашифрованного трафика. Дополнительно ставится проблема о проведении пассивного анализа в программно-определяемых сетях (англ. *SDN – Software-Defined Networks*).

Разработкой подобных инструментов для компьютерных сетей занимаются коллективы исследователей как из Российской Федерации, так и иностранных государств. В области отечественной науки данной проблематикой занимается аналитическая группа под управлением д.т.н. М. В. Филиппова из Московского Государственного Технического Университета имени Н. Э. Баумана, в своей работе предложившая в качестве основы для сетевого анализатора структуру *struct_net_device* совместно с *struct_net_ops* из пакета *NetFilter* [10]. Следует отметить, что представленный инструмент реализует собранную архитектуру, со-

ответственно в работе предполагается установка данного средства на конечных узлах локальной сети. Решение, предложенное специалистами, позволяет обеспечить универсальность фильтрации и возможность адаптации предложенных функций к потребностям пользователя, однако у него присутствуют определенные недостатки. В первую очередь — избыточность загружаемого в ядро кода. Данный недостаток связан с особенностями пакета *NetFilter*, вследствие которых для анализа входящих и исходящих соединений требуется два загружаемых фильтрующих выражения, дополнительно для каждого из них — отдельный программный код для удаления и загрузки, что приводит к усложнению логики работы программы. Далее необходимо отметить отсутствие верификации при загрузке фильтрующих выражений в пространство ядра: при некорректном конфигурировании загружаемого кода возможен критический сбой операционной системы, что недопустимо в контексте обеспечения безопасности.

Следует отметить, что в рамках научных изысканий направлением разработки и интеграции интеллектуально-адаптивных систем информационной безопасности занимается научная школа под руководством к.т.н. Е. А. Басыни [11].

В рамках исследования предметной области был спроектирован и реализован тестовый стенд с элементами автоматизации, более подробно описанный в разделе «Экспериментальное исследование». В контексте данной инфраструктуры устанавливаются и конфигурируются незащищенные сервисы *vsftpd* и *telnetd* с целью воспроизведения уязвимости перехвата потока трафика и его последующего анализа третьими лицами. Процесс воспроизведения состоял из трех шагов: автоматизированная реализация сервисов (*ansible-playbook*), запуск клиентского решения и ввод конфиденциальных пользовательских данных, перехват и последующий анализ информационных потоков инструментом *Wireshark* с использованием опции отслеживания *TCP*-соединений. Результаты воспроизведения уязвимости проиллюстрированы на рис. 1.

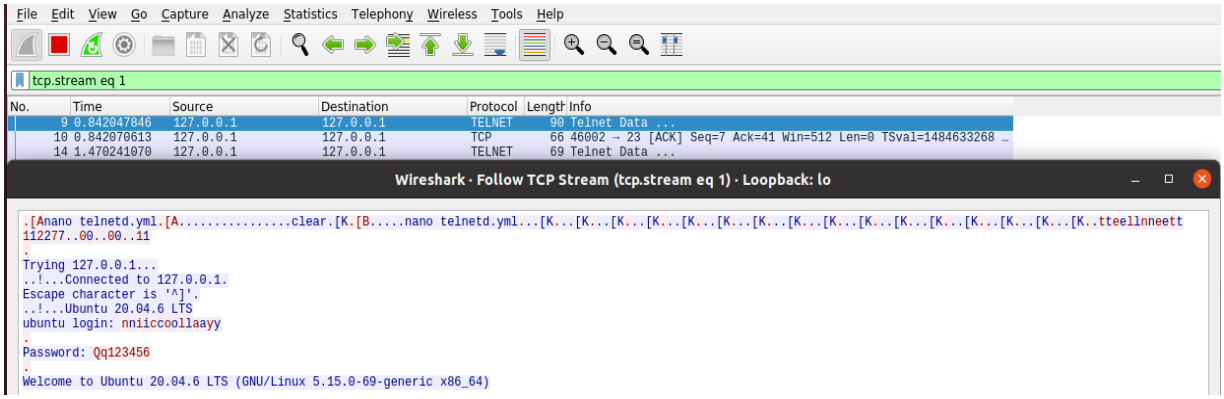

Рис. 1. Результат отслеживания TCP-соединений инструментом Wireshark

Проектирование и разработка сетевого анализатора трафика

В рамках архитектурного решения для собственного инструмента пассивного анализа трафика вычислительной сети рассматривался собранный и распре-

деленный вариант построения. В результате проведения сравнительного анализа был выбран первый вариант, поскольку согласно задумке программный продукт располагается на одном из конечных узлов локальной сети. Дополнительную обоснованность выбора собранной архитектуры подтверждают следующие факторы: во-первых, подобное решение не предполагает наличия промежуточного программного обеспечения, соответственно вся функциональность продукта достигается на единственном узле; во-вторых, отсутствует необходимость в различных системных буферах между приложениями и сетью, как следует в случае распределенной архитектуры. Таким образом, благодаря данным соображениям достигается больший уровень надежности и отказоустойчивости, чем в распределённых системах. Далее следует выбрать настройки окружения, в котором будет разрабатываться инструмент.

В качестве среды разработки был выбран дистрибутив *Linux Ubuntu 22.04.01 LTS* с версией ядра не ниже 4.0.6. В качестве языка программирования взят Си, соответствующий последнему опубликованному стандарту ISO/IEC 9899:2018. Дополнительно на операционную систему необходима установка программного компонента *linux-gnu11* для корректного взаимодействия со средой сетевых вызовов Linux. В целях поддержки расширенной функциональности инструмента требуется компилятор языка Си *Clang* актуальной версии, пакет с низкоуровневой виртуальной машиной *LLVM* (англ. *Low Level Virtual Machine*) и утилита командной строки *bpf_asm.* Фильтрующие выражения *pcap* в операционной системе Linux по умолчанию скомпилированы в программы *eBPF* (англ. *enhanced Berkeley Packet Filter*) и хранятся в директории средства *tcpdump*.

Основой для разработки инструмента пассивного анализа трафика вычислительных сетей в результате проведенных исследований предметной области выбрана утилита командной строки *tcpdump*. В поддержку данного программного продукта дополнительно выступают следующие обстоятельства:

– активная поддержка и сопровождение утилиты;

– модульное строение программы, обеспечивающее простоту включения дополнительной функциональности;

– открытый исходный код;

– совместимость с *ОС* семейства *GNU/Linux* и механизмами ядра данной системы.

Утилита *tcpdump* является интерфейсом для библиотеки захвата пакетного трафика *Lpcap* (англ. *Library for Packet Capture Process*), позволяющей пользователю определять высокоуровневые выражения фильтрации. Для своей работы инструмент применяет данные ограничивающие условия к обнаруженным информационным потокам, затем выполняет остальную программу с применением выбранных выражений. Соответственно, пользователь при эксплуатации утилиты командной строки фактически компилирует и загружает в ядро операционной системы программу *eBPF* (англ. *enhanced Berkeley Packet Filter*) [12].

Применяемая в проектируемом сетевом анализаторе технология *eBPF* является встроенной в пространство ядра виртуальной машиной, которая обладает возможностью запуска программ, внедренных из пользовательского простран-

ства. В контексте разрабатываемого инструмента пассивного анализа трафика *BPF*-машина используется для осуществления фильтрации пакетов на определенном сетевом интерфейсе — сокете. Данный механизм ядра принимает скомпилированное в байт-код выражение и проверяет приходящие пакеты на соответствие данному выражению [13].

Важно отметить, что глобально существует два подхода к математической абстракции фильтра: дерево булевых выражений и направленный ациклический граф потока управления *CFG*, применяемый совместно с технологией e*BPF*. Важно обратить внимание, что в использовании модели отличаются: модель дерева отображается в код для обыкновенной стековой машины, а *CFG* отображается в код для регистровой машины e*BPF*. На рис. 2 приведена модель *CFG* на примере фильтра, который распознает пакеты типа *IPv4 (*англ. *Internet Protocol version 4)* или *ARP (*англ. *Address Resolution Protocol)* в сети *Ethernet*.

Следует отметить, что в математической модели *CFG* каждый узел представляет предикат поля пакета, а ребра представляют передачи управления. Соответственно правая ветвь обходится, если предикат истинен, левая — ложна. Допускаются только два конечных листа, которые представляют результат предиката для всего фильтрующего выражения.

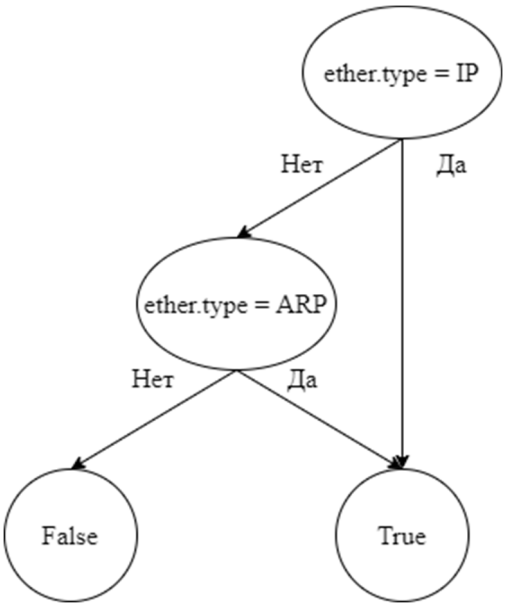

Рис. 2. Граф потока управления *CFG* для фильтрации *IP* пакетов и *ARP* кадры

Важно отметить, что в отличие от утилиты *tcpdump*, на базе которой разрабатывается инструмент пассивного анализа трафика, собственное приложение обладает дополнительной функциональностью, относящейся к применяемым фильтрам. Она выражается в возможности загрузки извне нестандартных фильтров: как написанных на языке программирования Си, так и ассемблере *Motorola 6502*, применяющемся в виртуальной машине *eBPF*. Для реализации подобной функциональности требуется подготовка загружаемого выражения с предобра-

боткой: использованием компилятора *Clang* и низкоуровневой виртуальной машины *LLVM*, поскольку выражения передаются в основную программу в виде полностью скомпилированных функций. На рис. 3 показан процесс компиляции итогового выражения.

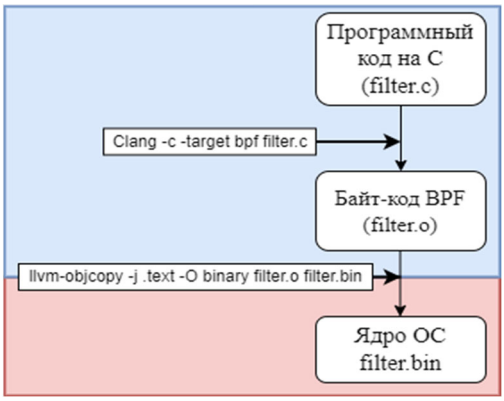

Рис. 3. Компиляция фильтра для подачи в программу

В соответствии с рис. 4 вторым вариантом разработки ограничивающего выражения является написание кода на ассемблере *BPF*. В таком случае для формирования листинга, загружаемого в ядро, следует применять полноценный ассемблер *BPF*, который проведет необходимые процедуры компиляции самостоятельно. Ниже представлен порядок команд, необходимый для достижения корректной работы загружаемого выражения.

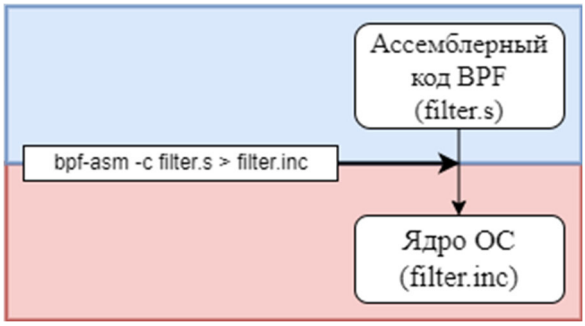

Рис. 4. Компиляция фильтра в случае исходного файла на ассемблере

Следует отметить, что разработанная программа-демонстратор в одном из режимов работы поддерживает использование библиотеки *Lpcap*. Средствами данной библиотеки возможно дополнительно сохранять трафик в файлы дампа, что позволяет привнести свойство неотказуемости информации.

Предлагаемое решение

Предлагаемое решение основано на стеке технологий утилиты *tcpdump* и задействует различные механизмы обеспечения сетевого взаимодействия операци-

онной системы *Linux*. Для обеспечения фильтрации трафика выбраны следующие настройки работы разработанного инструмента:

– программный интерфейс типа raw-socket для обеспечения низкого уровня сетевой абстракции при зеркалировании пакетов с физического интерфейса, что предоставляет доступ к заголовкам канального уровня стека протоколов *TCP/IP* (заголовки, обеспечивающие межсетевое взаимодействие);

– встроенная в ядро операционной системы виртуальная машина *eBPF*, способствующая быстродействию процесса идентификации пакетов и гибкости загрузки фильтрующего выражения;

– направленный ациклический граф потока управления *CFG*, позволяющий эффективно взаимодействовать с машиной *eBPF* в контексте обработки фильтра в ассемблерном модуле;

– библиотека *Lpcap*, позволяющая бесшовно интерпретировать пользовательские фильтрующие выражения в байт-код и привносить в работу инструмента свойство неотказуемости информации путем сохранения отфильтрованного трафика в файлы дампа;

– аналитический модуль внутри программы, позволяющий обнаруживать подозрительную активность в *ICMP*-трафике для возможной идентификации атак типа *ICMP-flood*.

На рис. 5 представлена диаграмма последовательностей разработанного решения, в которой приводится порядок выполнения отдельных операций с привязкой к течению времени и основным взаимодействующим акторам: пользователю, инструменту и операционной системе.

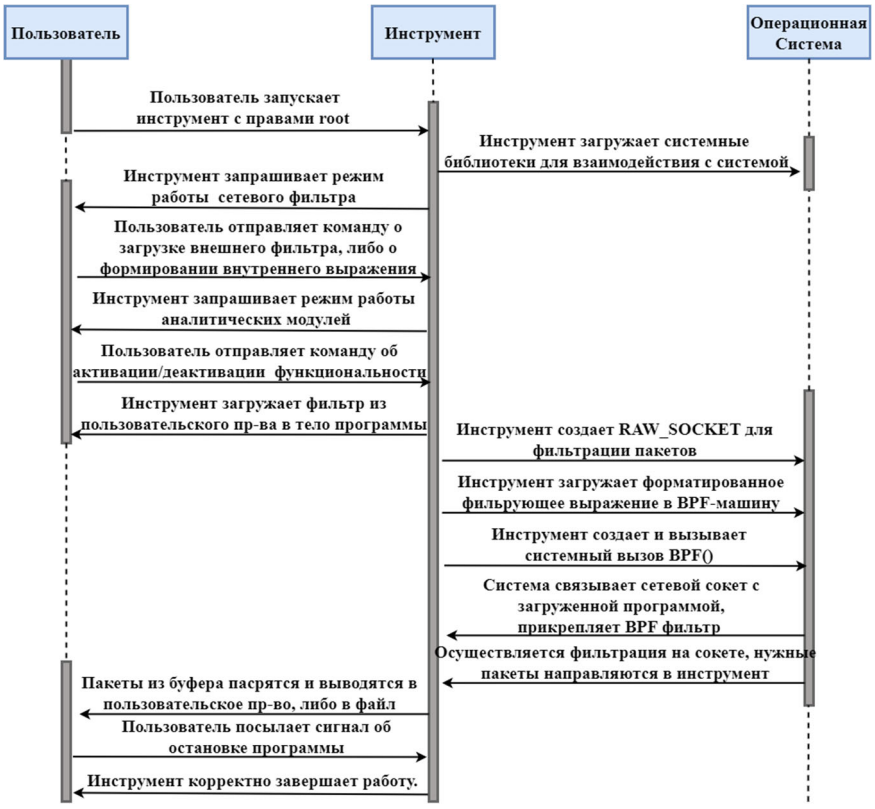

Рис. 5. Диаграмма последовательностей инструмента-демонстратора

Важно отметить, что в качестве дополнительной функциональности в инструмент добавлен аналитический модуль, позволяющий обнаруживать возможные атаки типа *ICMP-flood* путем сравнения высчитанной метрики количества пакетов в единицу времени с эталонным значением, вводимым пользователем на этапе ввода в программу аргументов.

В дальнейшем планируется распространить действие модуля на любые протоколы, связанные с атаками типа «отказ в обслуживании» (*DoS*) и усложнить логику его работы с помощью введения элементов машинного обучения и нейросетей.

Экспериментальное исследование

В целях проведения тестирования с привнесением элементов автоматизации на виртуальной машине с установленным разработанным решением проектировалась и реализовывалась инфраструктура, в соответствии с рис. 6 содержащая в себе три составные части:

1) намеренно некорректно сконфигурированные сервисы, разворачиваемые при помощи технологии *Ansible* (*ansible-playbook*);

2) различные *Bash*-скрипты, в которых запускались системные утилиты, генерирующие реальный служебный трафик с протоколами *ICMP, DNS* и *ARP*;

3) инструмент с открытым исходным кодом для проведения нагрузочного тестирования *Cisco TRex Traffic Generator* в целях проверки поведения разработанного решения на больших объемах и типах трафика.

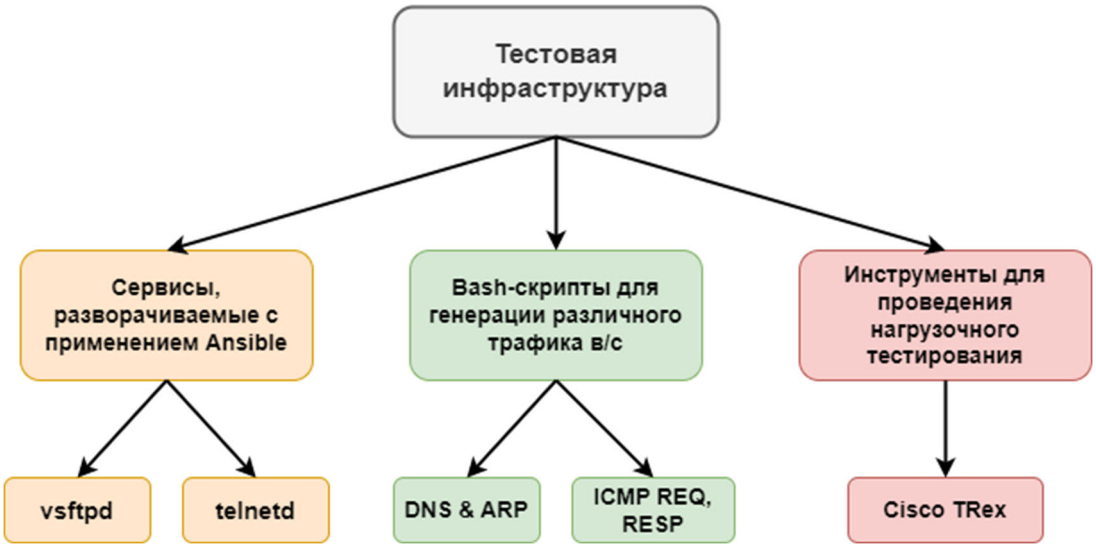

Рис. 6. Блок-схема тестовой инфраструктуры

Важно отметить, что разворачивание сервисов потребовалось для определения возможности инструмента предупреждать пользователя о неверно сконфигурированных или условно небезопасных сервисах. Так, в клиент-серверном взаимодействии *vsftpd* (протокол *FTP* (англ. *File Transfer Protocol*)) и *telnetd* (протокол *Telnet*) по умолчанию отсутствует шифрование передаваемых данных. Это

может привести к компрометации информации, поскольку предполагаемый злоумышленник, реализуя особенности специального программного обеспечения, имеет возможность перехватывать *TCP/IP*-пакеты от пользователя и получать доступ к его аутентификационным данным.

Экспериментальное тестирование проводилось в два этапа: запуск собственного *Bash*-сценария совместно с разработанным инструментом с фильтрацией *ICMP*-пакетов и включенным аналитическим модулем, затем — нагрузочное тестирование с установкой и запуском клиент-серверного решения *Cisco TRex* на *loopback*-интерфейсе для генерации трафика с различными типами протоколов. Результаты тестирования приводятся на рис. 7–9.

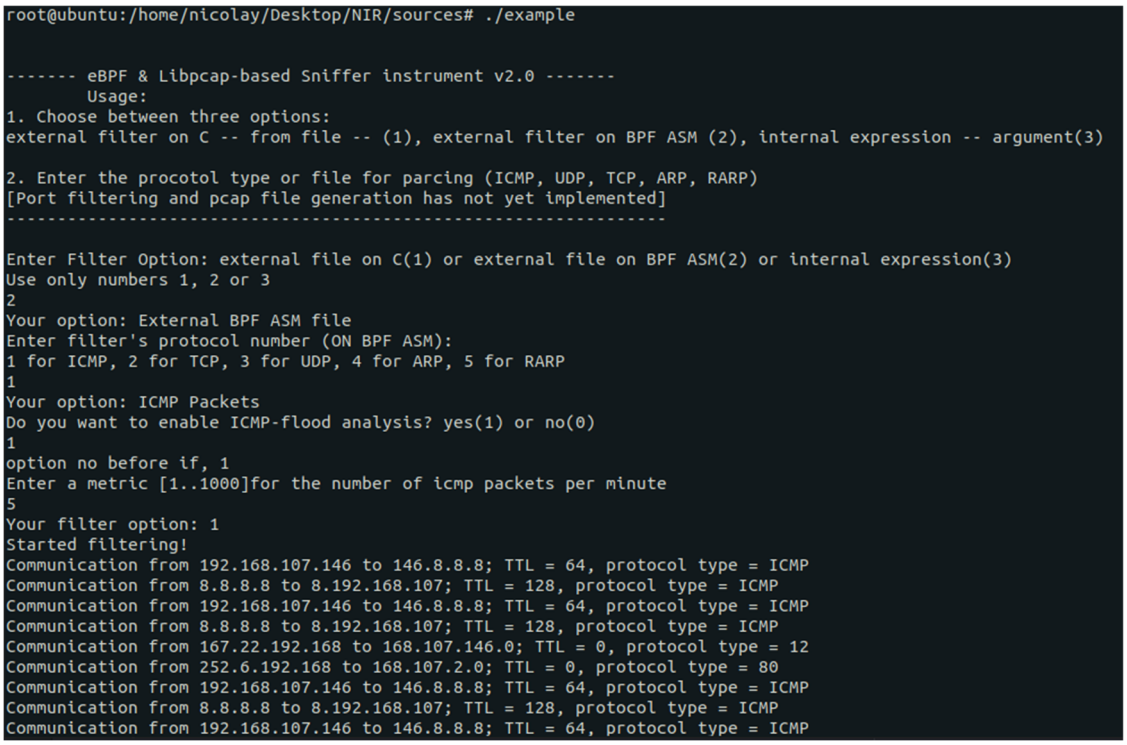

Рис. 7. Запуск инструмента с *ICMP*-фильтром и подключенным аналитическим модулем

Следует отметить, что в данном случае инструмент запускался с опциями: фильтрующее выражение на языке *BPF ASM* (скомпилированное), тип *ICMP*, включение аналитического модуля, введение нужной метрики максимального количества пакетов заданного протокола в единицу времени.

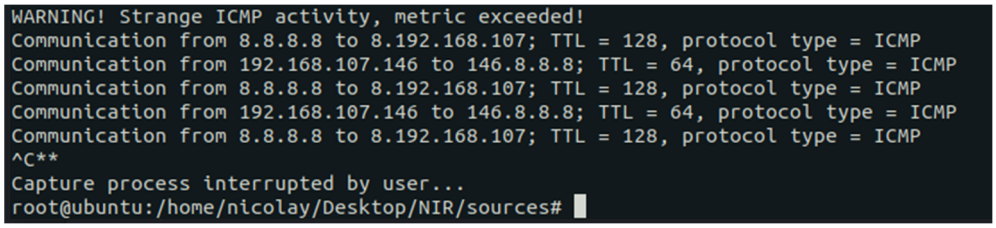

Рис. 8. Уведомление пользователя о превышении заданной метрики

В данном случае среди потока информационных сообщений о пришедших на интерфейс и отфильтрованных пакетах высвечивается особое служебное уведомление для пользователя.

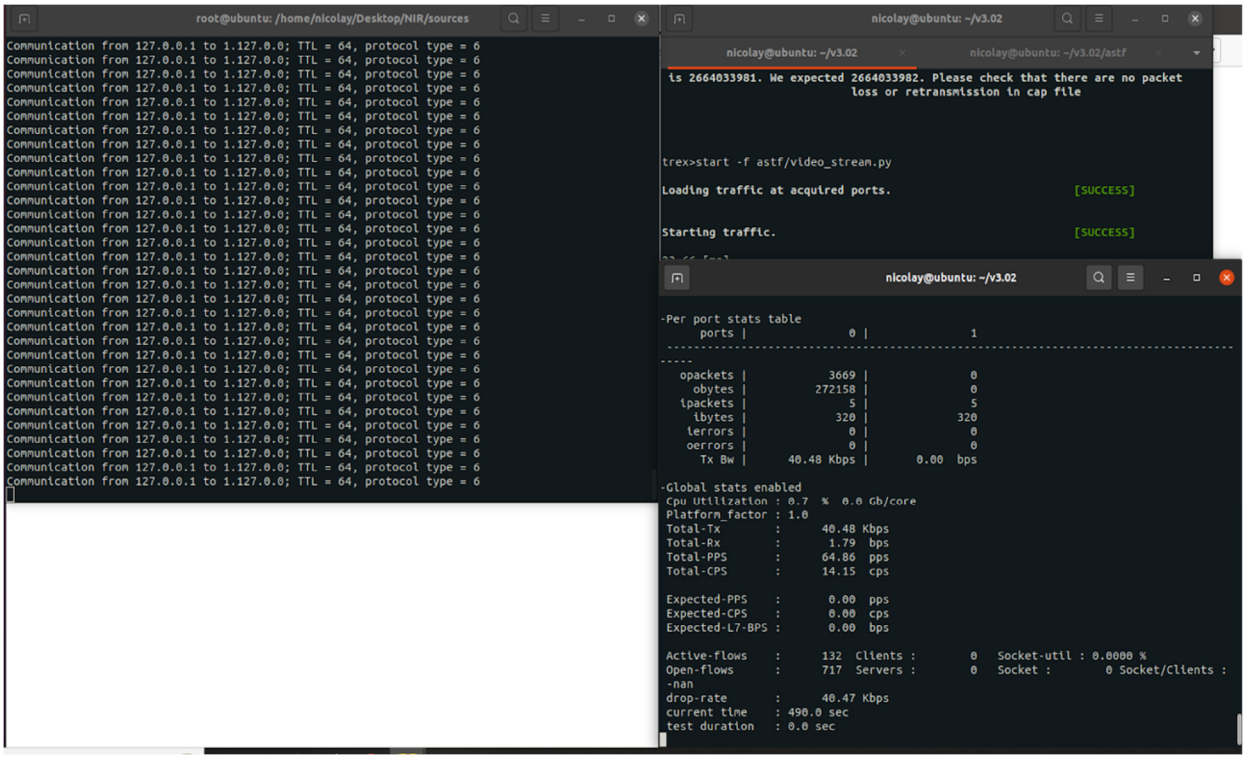

Рис. 9. Запуск клиент-серверного решения совместно с разрабатываемым инструментом с генерацией потока *TCP*-сегментов

В нижнем окне терминала запущен процесс сервера, в правом верхнем – клиента (для обеспечения взаимодействия с сервисом), слева – процесс с запущенным сетевым анализатором. Наиболее высокой скорости передачи данных соответствовал шаблон *videostream.py*, который высылал поток *TCP*-сегментов.

Следует отметить, что разработанное решение показало себя успешно, корректно обнаружило различный генерируемый трафик, стабильно отработало в условиях нагруженного тестирования и идентифицировало подозрительный *ICMP*-трафик в контексте использования аналитического модуля.

Заключение

В результате работы цель, заключающаяся в идентификация уязвимостей информационной инфраструктуры предприятия посредством пассивного анализа трафика вычислительной сети, достигнута. Все заявленные задачи реализованы. Практическим результатом работы выступает разработанная демонстрационная версия сетевого анализатора. Проведено тестирование продукта с привнесением процессов автоматизации. Практическая значимость работы заключается в возможности идентификации несанкционированных внутренних и внешних воздействий на основе анализа информационных потоков, что позволяет повы-

сить эффективность фильтрации трафика системами защиты: как отдельными *IDS/IPS*, так и в комбинации с *SIEM* системами.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Исследование компании Acronis [Электронный ресурс]. — 2022. — URL: https://www.acronis.com/en-eu/blog/posts/acronis-cyberthreats-report-year-end-2022-data-under-attack/ (дата обращения 02.12.2022).

2. Исследование Всемирного Союза Электросвязи [Электронный ресурс]. — 2019. — URL: https://www.itu.int/en/ITU-D/Statistics/Pages/stat/default.aspx (дата обращения 02.12.2022).

3. Исследование Всемирного Союза Электросвязи [Электронный ресурс]. — 2019. — URL: https://rt-solar.ru/upload/iblock/4a4/ghus61x9rd8cv5vczms5ig1svts4tlep/Otchet-o-kiberatakakh-na-rossiyskie-kompanii-v-2022-godu.pdf (дата обращения 02.12.2022).

4. Басыня Е. А. Комплексный мониторинг информационной инфраструктуры предприятия = Comprehensive monitoring of the company's information infrastructure / Е. А. Басыня, Д. С. Худяков // Защита информации. Инсайд = Zasita informacii. Inside. - 2020. - № 6 (96). - С. 28-33.

5. Basinya E. A. Automatic traffic control system for SOHO computer networks / E. A. Basinya, A. A. Rudkovskiy // Studies in Systems, Decision and Control. - 2019. - Vol. 119 : Recent Research in Control Engineering and Decision Making. ICIT 2019. - P. 743-754. - DOI: 10.1007/978-3-030-12072-6_60.

6. Исследование компании Gartner [Электронный ресурс]. — 2019. — URL: www.gartner.com/en/documents/3902353 (дата обращения 02.04.2023).

7. Iglesias Vázquez NTARC: A Data Model for the Systematic Review of Network Traffic Analysis Research /Félix & Ferreira, Daniel & Vormayr, Gernot & Bachl, Maximilian & Zseby, Tanja // Applied Sciences — 2020. — Vol. 10. — P.2-24. — DOI: 10.3390/app10124307.

8. Robles, David & Nuño, Pelayo & Bulnes, Francisco & Candas, Juan. (2021). Performance Analysis of Packet Sniffing Techniques Applied to Network Monitoring. IEEE Latin America Transactions. 19. 490-499. 10.1109/TLA.2021.9447699.

9. Lesley F. Sikos Packet analysis for network forensics: A comprehensive survey, // Forensic Science International: Digital investigation. – 2020. – Vol. 32C. – P.1-13. – DOI:10.1016/j.fsidi.2019.200892.

10. Филиппов, М. В. Метод перехвата исходящих и входящих сетевых пакетов /Филиппов М.В., Рязанцев Б.И., Рязанова Н.Ю.— Текст : непосредственный // Машиностроение и компьютерные технологии. — 2017. — № 12. — С. 45-56.

11. Басыня Е. А. Метод идентификации киберпреступников, использующих инструменты сетевого анализа информационных систем с применением технологий анонимизации = Method to identify cybercriminals using network analysis of information systems with anonymization / Е. А. Басыня, В. Е. Хиценко, А. А. Рудковский // Доклады Томского государственного университета систем управления и радиоэлектроники. - 2019. – Т. 22, № 2. – С. 45–51. - DOI: 10.21293/1818-0442-2019-22-2-45-51

12. Luca Deri Combining System Visibility and Security Using eBPF, /Luca Deri, Samuele Sabella, Simone Mainardi // ITASEC2019 : [proc.], Pisa,Italia, 12-15 Feb. 2019. — Pisa: ITASEC, 2019. - Mode of access: https://luca.ntop.org/ITASEC2019.pdf.

13. S. Schmid Efficient Network Monitoring Applications in the Kernel with eBPF and XDP [Electronic resource] /S. Schmid, M. Abranches, O. Michel, E. Keller // IEEE Conference on Network Function Virtualization and Software Defined Networks (NFV-SDN) : [proc.], Heraklion, Greece, 9-11 Nov. 2021. — Heraklion: IEEE, 2021. - Mode of access: https://ieeexplore.ieee.org/document/9665095/. - Title from screen -DOI:10.1109/NFV SDN53031.2021.9665095