(19) Евразийское патентное ведомство

(11) 040394 (13) B1

(12) ОПИСАНИЕ ИЗОБРЕТЕНИЯ К ЕВРАЗИЙСКОМУ ПАТЕНТУ

(45) Дата публикации и выдачи патента
2022.05.26

(21) Номер заявки
201992850

(22) Дата подачи заявки
2019.12.26

(51) Int. Cl. ***G06F 17/00*** (2019.01)
G06F 21/00 (2013.01)

(54) АВТОМАТИЗИРОВАННАЯ СИСТЕМА ТЕСТИРОВАНИЯ СОБЫТИЙ КИБЕРБЕЗОПАСНОСТИ

(31) **2019142435**
(32) **2019.12.19**
(33) **RU**
(43) **2021.06.30**

(71)(73) Заявитель и патентовладелец:
ПУБЛИЧНОЕ АКЦИОНЕРНОЕ ОБЩЕСТВО "СБЕРБАНК РОССИИ" (ПАО СБЕРБАНК) (RU)

(72) Изобретатель:
Тамойкин Андрей Юрьевич (RU)

(74) Представитель:
Герасин Б.В. (RU)

(56) WO-A1-2019040613
US-A1-20190253448
US-B1-10346612
US-A1-20170032695
US-A1-20140245422

(57) Настоящее техническое решение в общем относится к области вычислительной обработки данных, а в частности к автоматизированным системам тестирований событий кибербезопасности. Автоматизированная система тестирования событий кибербезопасности, содержащая модуль оркестрации запросов, выполненный с возможностью перенаправления запросов пользователей к модулям системы, причем пользовательский запрос содержит параметры формирования алгоритмов тестирования; модуль автоконфигурации инфраструктуры в виртуальной среде, выполненный с возможностью автоматического развертывания инфраструктуры в виртуальной среде из конфигурационного файла пользователя и взаимодействия с внешними инструментами; модуль проведения тестирования, выполненный с возможностью автоматического тестирования событий кибербезопасности на основе информации алгоритмов тестирования, содержащейся в конфигурационном файле пользователя; базу данных, содержащую конфигурационные файлы пользователей с заданными параметрами алгоритмов для тестирования событий кибербезопасности.

040394 B1

040394 B1

Область техники

Настоящее техническое решение в общем относится к области вычислительной обработки данных, а в частности к автоматизированным системам тестирования событий кибербезопасности.

Уровень техники

Основной задачей кибербезопасности является поддержание и повышение уровня защищенности объектов защиты соразмерно возникающим рискам, связанным с киберугрозами данным объектам защиты. При этом выявление действительного направления изменения (повышение или понижение) уровня защищенности при применении каких-либо действий на это направленных (например, изменение настроек групповой политики домена, внедрение нового СЗИ или добавление второго фактора аутентификации) по сути также является гипотезой. На текущий момент данная гипотеза зачастую подтверждается или опровергается либо экспертным мнением, либо тестированием с низкой точностью (на промышленных инфраструктурах, где отсутствует возможность использования действительно вредоносной активности). Ни то, ни другое не является достаточным доказательством для принятия решения о применении каких-либо действий, особенно в промышленной среде.

Таким образом, что в рамках исследовательской деятельности, что в рамках операционной деятельности в области кибербезопасности постоянно возникает потребность в проведении экспериментов (тестирований). При этом основной особенностью проведения таких тестов в "точном" исполнении является использование реальных вредоносных действий, программ, нагрузок, которые, в свою очередь, способны разрушить всю инфраструктуру проведения тестирования, что приведет к дополнительным трудозатратам или невозможности выполнения тестирования.

Помимо этого, гипотезы, сформулированные в исследовательских лабораториях и внутри организации, зачастую имеют различный характер и требуют различных тестовых сред для проведения тестирования (в лаборатории это может быть стенд с общими инфраструктурными настройками, тогда как внутри организации это должен быть стенд, отражающий реальную инфраструктуру с определенной точностью).

Более того, статические тестовые среды также не всегда подходят для тестирования в области кибербезопасности, так как ряд существующих на данный момент атак использует динамические данные, генерируемые пользователями и приложениями (например, пользовательский сетевой трафик или область памяти, в которой работает программа).

Ввиду этого наличие системы, позволяющей автоматизировать развертывание среды для тестирования, проводить заданный сценарий тестирования с подготовленной динамической активностью внутри среды теста и изолированием этой среды от внешних взаимодействий является приоритетной задачей для проведения исследовательской деятельности.

Такие системы широко известны из уровня техники, например CybExer Range Platform (см. https://cybexer.com), Cyber Exercise & Research Platform (см. https://www.kypo.cz/en). Общим недостатком существующих решений в данной области является низкая скорость проведения тестирования и отсутствие автоматизации процесса тестирования.

Сущность технического решения

Заявленное техническое решение предлагает новый подход в области тестирования событий кибербезопасности.

Решаемой технической проблемой или технической задачей является создание новой автоматизированной системы для тестирования событий в области кибербезопасности, обладающей высокой скоростью и точностью проведения тестирования.

Основным техническим результатом, достигающимся при решении вышеуказанной технической проблемы, является автоматизация процесса проведения тестирования в области событий кибербезопасности.

Дополнительным техническим результатом, достигающимся при решении вышеуказанной технической проблемы, является повышение скорости проведения тестирования в области событий кибербезопасности.

Заявленные результаты достигаются за счет автоматизированной системы тестирования событий кибербезопасности, содержащей

модуль оркестрации запросов, выполненный с возможностью перенаправления запросов пользователей к модулям системы, причем пользовательский запрос содержит параметры формирования алгоритмов тестирования;

модуль автоконфигурации инфраструктуры в виртуальной среде, выполненный с возможностью автоматического развертывания инфраструктуры в виртуальной среде из конфигурационного файла пользователя и взаимодействия с внешними инструментами;

модуль проведения тестирования, выполненный с возможностью автоматического тестирования событий кибербезопасности на основе информации алгоритмов тестирования, содержащейся в конфигурационном файле пользователя;

модуль мониторинга инфраструктуры, выполненный с возможностью автоматического добавления механизмов мониторинга внутри созданной инфраструктуры и обработки событий, возникающих в результате мониторинга активности объектов инфраструктуры;

по меньшей мере одну базу данных, содержащую конфигурационные файлы пользователей с заданными параметрами алгоритмов для тестирования событий кибербезопасности.

В одном из частных вариантов осуществления система содержит прокси сервер, выполняющий алгоритмы безопасного соединения.

В другом частном варианте осуществления модуль проведения тестирования выполняет алгоритм обработки сценария тестирования.

В другом частном варианте осуществления модуль проведения тестирования выполняет алгоритм подготовки очереди задач, которые должны быть реализованы до начала тестирования.

В другом частном варианте осуществления модуль проведения тестирования выполняет алгоритм формирования обратной связи с инфраструктурой во время тестирования.

В другом частном варианте осуществления модуль проведения тестирования выполняет алгоритм оптимизации инфраструктуры в виртуальной среде.

Описание чертежей

Признаки и преимущества настоящего изобретения станут очевидными из приводимого ниже подробного описания изобретения и прилагаемых чертежей.

Фиг. 1 иллюстрирует общую схему заявленной автоматизированной системы тестирования событий кибербезопасности.

Фиг. 2 иллюстрирует общую схему мониторинга в реальном времени.

Фиг. 3 иллюстрирует общую схему мониторинга по результатам активности в инфраструктуре.

Фиг. 4 иллюстрирует общую схему мониторинга с помощью гипервизора платформы виртуализации.

Фиг. 5 иллюстрирует общий вид вычислительного устройства для реализации системы.

Осуществление изобретения

Автоматизированная система тестирования событий кибербезопасности предназначена для автоматического развертывания инфраструктуры в виртуальной среде из конфигурационного файла, созданного пользователем, автоматической подготовки и добавления различной динамической активности в созданную инфраструктуру, разработки сценария тестирования в области кибербезопасности, оптимизации инфраструктуры, участвующей в тестировании во время его проведения, проведения тестирования согласно заранее созданному сценарию, хранения и предоставления ранее созданных (другими пользователями) конфигурационных файлов пользователям системы. В рамках автоматического развертывания инфраструктуры в виртуальной среде система послойно (например, сначала виртуальные машины, затем сеть между ними, затем настройка домена и т.д.) воспроизводит инфраструктуру, описанную в конфигурационной файле, созданном пользователем. Для этого система с помощью собственных алгоритмов автоконфигурации анализирует конфигурационный файл и транслирует его в конкретные директивы, которые располагаются в порядке, который не приведет к возникновению ошибок при автоматическом развертывании. Данные директивы представляют собой вызовы к внешним подсистемам. Система оркестрирует директивы, перенаправляя их в правильном порядке к внешним подсистемам, обрабатывая результат выполнения директивы и корректируя следующую директиву в соответствии с полученным результатом. Таким образом, постепенное выполнение всех директив позволяет в конечном итоге получить законченную инфраструктуру, соответствующую заданной в конфигурационном файле пользователя.

Система использует внешние подсистемы как инструмент для выполнения директив и позволяет, но не ограничивается, реализовывать следующие директивы:

создать виртуальные машины (из заранее подготовленных шаблонов или с помощью создания новой виртуальной машины с заданными характеристиками),

создать сетевую топологию для созданных виртуальных машин (с эмуляцией реального сетевого оборудования),

создать базовые интеграции между виртуальными машинами (развертывание и конфигурация домена, почтового сервиса, системы динамического распределения IP-адресов и т.д.),

развернуть программное обеспечение на созданных виртуальных машинах,

развернуть систему мониторинга и сбора данных, полученных в ходе тестирования, для созданного киберпространства,

изолировать созданное киберпространство от внешних воздействий виртуальной среды.

При этом система позволяет добавлять внешние подсистемы для реализации новых директив.

В рамках автоматической подготовки и добавления различной динамической активности в созданную инфраструктуру система позволяет подключить различные инструменты для симуляции той или иной активности в созданной виртуальной инфраструктуре. Система использует сторонние инструменты (например, интеллектуальные агенты) для реализации динамической активности и позволяет, но не ограничивается, реализовывать следующие активности:

симуляция зловредной активности (как действия злоумышленников, так и доставка, и запуск вредоносного программного обеспечения),

симуляция действий пользователей (на рабочий станциях),

симуляция сетевого трафика,

симуляция публичного сегмента сети Интернет,

симуляция сетевых сервисов.

При этом система позволяет добавлять внешние подсистемы для реализации новых активностей.

В рамках разработки сценария тестирования в области кибербезопасности пользователь указывает системе инфраструктуру и активность, которую необходимо провести в выбранной инфраструктуре. Помимо этого, пользователь может задать начальные условия активности, условия окончания активности и триггеры для осуществления тех или иных событий, которые пользователь задает отдельно (например, изменение таблицы межсетевого экрана в случае обнаружения определенного события), количество повторений, динамические переменные (например, несколько значений одной и той же настройки, чтобы в рамках одинаковых тестирований она изменялась и можно было увидеть разницу) и т.д.

В рамках оптимизации инфраструктуры участвующей в тестировании во время его проведения система предполагает различную степень "точности" (под "точностью" понимается приближенность к оригиналу, например: реальный компьютер можно виртуализировать - виртуальная машина с реальной ОС; эмулировать - запуск ОС внутри реальной ОС, установленной, например, на виртуальной машине; симулировать - запуск программы, которая будет способна принимать вызовы и возвращать ответы, симулировать поведение ОС) воспроизведения того или иного объекта (хост, сервис, сетевая топология, сетевое оборудование, пользователь в ОС, траффик в сети и т.д.) в виртуальной среде, путем выбора его представления: симуляция объекта, эмуляция, виртуализация, контейнеризация или использование полноценного оригинала (в аппаратном исполнении). Причем система может как дополнить основную инфраструктуру конфигурационного файла объектами с меньшей "точностью" (например, вокруг реальных виртуальных машин, реализующих один сетевой сегмент будет развернута симуляция других сетевых сегментов или еще один сетевой сегмент, реализующий рабочие станции с помощью контейнеризации), что позволяет расширить область тестирования, не увеличивая нагрузку на производительность системы, так и воспроизвести часть инфраструктуры из конфигурационного файла в менее "точном" варианте с последующим увеличением "точности" в режиме реального времени - используется для уменьшения необходимой мощности системы в рамках проведения тестирования в пошаговом режиме.

В рамках проведения тестирования согласно заранее созданному сценарию система, используя информацию о тестировании, оптимизации инфраструктуры тестирования (если необходимо), динамических переменных тестирования, количестве повторений (если заданы на этапе создания сценария тестирования) формирует план проведения тестирования, в рамках которого

определяется необходимая мощностью поддержания инфраструктуры тестирования, рассчитывается максимально возможное количество параллельных потоков тестирования с учетом динамических переменных и количества повторений,

формируется очередь задач по проведению тестирования.

После формирования плана проведения тестирования система начинает его автоматическое проведение. При этом формируется заранее рассчитанное количество параллельных потоков, каждый из которых получает задачу из очереди, которая (задача) состоит из следующих этапов:

развертывание инфраструктуры в виртуальной среде,

добавление динамической активности,

начало тестирования,

обработка событий согласно триггерам тестирования,

обнаружение события окончания тестирования,

сбор полученных данных,

сохранение данных,

уничтожение развернутой инфраструктуры.

Помимо этого, присутствует режим пошагового проведения тестирования, в котором пользователь заранее формирует шаги активностей, а система автоматически подготавливает необходимую часть инфраструктуры в заданной "точности. В рамках хранения и предоставления ранее созданных конфигурационных файлов пользователям система обеспечивает единую базу данных, в которой сохраняются ранее созданные сценарии тестирований и конфигурационные файлы инфраструктур. Также, если сценарий и/или инфраструктура уже использовались, то система может сохранять техническую информацию, которая позволяет в случае повторного использования пропускать ряд шагов, выполняемых системой для запуска тестирования или развертывания инфраструктуры. Также система осуществляет контроль версий указанных файлов и предоставляет интерфейс для поиска, просмотра, редактирования данных файлов. Помимо этого, система предоставляет программный интерфейс (API) для обмена и синхронизации содержимого (с учетом прав доступа) с базами данных таких же систем. Основные входные данные для системы предоставляются в конфигурационном файле, который может представлять один из следующих форматов: YAML, XML, JSON, но не ограничиваться ими.

Задачей системы является обработка конфигурационного файла и реализация указаний, которые в нем содержаться с целью проведения тестирования в области кибербезопасности. Целями данного тестирования являются, но не ограничиваются:

получение набора данных соответствующих той или иной активности в условиях конкретной инфраструктуры,

оценка мер защищенности (или изменения мер защищенности) той или иной инфраструктуры от различного вида компьютерных атак,

поиск мер защиты от различного вида компьютерных атак,

анализ и выявление техник и тактик зловредного программного обеспечения,

обучение алгоритмов искусственного интеллекта проведению компьютерных атак на те или иные инфраструктуры,

обучение алгоритмов искусственного интеллекта детектированию компьютерных атак, обучение алгоритмов искусственного интеллекта применению мер защиты при детектировании тех или иных компьютерных атак.

Помимо проведения тестирований в области кибербезопасности система способна выполнять следующие задачи:

тестирование и оценка эффективности средств защиты от различных видов компьютерных атак,

тренировка персонала организации, ответственного за обеспечение кибербезопасности, прототипирование решений, как для определения уровня их защищенности, так и для определения их влияния на защищенность инфраструктуры в целом,

демонстрация новых технологий.

Для начала работы с системой пользователь должен пройти аутентификацию, которая может представлять собой любой общеизвестный способ аутентификации: с помощью отдельной учетной записи и секрета, с помощью единой учетной записи и секрета (Single-Sign-ON), с помощью личного сертификата пользователя, биометрическая аутентификация, двухфакторная аутентификация и т.д.

В случае отсутствия у пользователя учетной записи он может предварительно осуществить регистрацию нового пользователя - в этом случае будут назначены минимальные привилегии в системе, которые пользователь сможет повысить путем предоставления идентифицирующей информации (например, личный сертификат или биометрические данные и т.д.).

После прохождения аутентификации пользователю назначаются права в соответствии с его ролью в системе, однако прокси-сервер может оценивать каждый запрос пользователя отдельно от назначенных ему прав путем динамического формирования скоринга пользователя в соответствии с его действиями и методами аутентификации и доступа. И в случае значительного отклонения данного скоринга блокировать выполнение запросов, которые предусмотрены ролью пользователя (принцип нулевого доверия). Прокси-сервер реализует классические алгоритмы обеспечения безопасности: регистрацию пользователей, аутентификацию пользователей, авторизацию запросов, маршрутизацию запросов с целью балансировки нагрузки, разграничение прав доступа пользователей. Помимо этого, наличие прокси-сервера позволяет реализовать принцип нулевого доверия в части входящих сетевых соединений и/или внешних запросов к серверной части системы.

Аутентифицированный пользователь (с поправкой на собственную роль) имеет возможность выполнять стандартные операции типа

поиск в базе данных;

просмотр готовых конфигураций инфраструктур, настроек и шаблонов (например, посмотреть наполнение конкретного шаблона виртуальной машины или атомарные действия злоумышленника в рамках осуществления той или иной атаки), корректировать настройки доступа к собственным ресурсам и т.д.

На фиг. 1 представлена общая схема заявленной системы (100).

Система (100) принимает потоки информации, которые генерируются пользователями.

Информация, поступающая в систему (100) распределяется по соответствующим модулям, ответственным за ее обработку в автоматизированном режиме. Система (100) содержит набор из взаимосвязанных модулей обработки данных.

Модуль оркестрации запросов (101) предназначен для перенаправления запросов пользователей к другим модулям системы, а также для перенаправления запросов между модулями системы. В качестве входных данных модуль (101) ожидает сетевое сообщение, которое состоит из заголовка и тела (<header> <body>). В заголовке сообщения должна содержаться вся техническая информация: категория запроса; команда запроса; аргументы и флаги команды запроса; данные пользователя или модуля, совершившего запрос; дополнительная информация, если необходимо - (<category> <command> <flags> <user_data> <payload>).

Модуль оркестрации запросов (101) обрабатывает, по меньшей мере, следующие запросы со стороны пользователя:

отображение конфигурационных файлов/файла, содержащихся в базе данных,

отображение данных конкретных сущностей, содержащихся в базах данных или хранилищах внешних систем (например, данные о сохраненной атаке в базе данных атак или данные о шаблонах виртуальных машин из платформы виртуализации),

поиск по конфигурационным файлам и их содержимому,

получение (скачивание) конфигурационного файла,

запуск авторазвертывания инфраструктуры,

уничтожение развернутой инфраструктуры,

запуск тестирования.

Также модуль оркестрации запросов (101) обрабатывает, по меньшей мере, следующие запросы со стороны модуля проведения тестирования (103):

запуск авторазвертывания инфраструктуры,

уничтожение развернутой инфраструктуры,

получение элемента из базы данных (например, конфигурация сетевой топологии).

Также модуль оркестрации запросов (101) обрабатывает, по меньшей мере, следующий запрос со стороны модуля автоконфигурации (102), направленный на получение элемента из базы данных (104) (например, конфигурация сетевой топологии). Модуль оркестрации запросов (101) реализует следующие алгоритмы. Алгоритм обработки запроса пользователя и других модулей системы. При получении входных данных (сетевого сообщения содержащего, по меньшей мере следующую информацию: <category> <command> <flags> <user_data> <payload> - категория запроса; команда запроса; аргументы и флаги команды запроса; данные пользователя или модуля, совершившего запрос; дополнительная информация, если необходимо) модуль оркестрации запросов (101) анализирует полученную техническую информацию, содержащуюся в заголовке сообщения. По информации в поле <category> определяется направление передачи сообщения. Для найденного направления определяется корректность команды, которая содержится в поле <command>. В случае если команда некорректна, формируется ответ, который содержит ошибку выполнения запроса. В случае если команда корректна, то анализируется сама команда. Если команда относится к взаимодействию с базами данных, то запускается алгоритм взаимодействия с базами данных. Если команда относится к запуску тестирования или запуску авторазвертывания инфраструктуры, то запускается алгоритм первичной проверки конфигурационного файла пользователя. После успешной проверки конфигурационного файла и в случае, если команда не относится к перечисленным выше, сообщение передается по направлению в нужный модуль системы (100).

Алгоритм взаимодействия с базами данных.

В случае если команда относится к взаимодействию с базами данных, модуль оркестрации запросов (101) самостоятельно обращается к нужной базе данных. Для этого существует механизм подключения баз данных, которые может быть реализован с помощью дополнительного программного скрипта, который реализует взаимодействие с API базы данных (104) (например, при интеграции с MongoDB); дополнительного программного скрипта, который реализует SQL-подобные запросы к базе данных (104) (например, при интеграции с MySQL); а также любых других известных способов интеграции с базами данных. При получении сообщения категории взаимодействия с базой данных (104), модуль оркестрации запросов (101) анализирует сообщение и определяет базу данных (104), к которой необходимо выполнить запрос. Получив данную информацию модуль оркестрации запросов (101) обращается к указанному методу коммуницирования с базой данных (104), куда передает информацию по подключению и запрос, который содержится в сообщении.

Далее формируется подключение к базе данных (104), а запрос из сообщения преобразовывается в вызов API подключенной базы данных (104) или в SQL-подобный запрос, который передается в базу данных (104). По результатам выполнения команды в базе данных (104) модуль оркестрации запросов (101) формирует ответ и возвращает его пользователю или модулю, от которого пришло входящее сообщение.

Алгоритм первичной проверки конфигурационного файла пользователя.

При получении входящего сообщения с задачей запуска тестирования или запуска авторазвертывания инфраструктуры, модуль оркестрации запросов (101) проводит первичный анализ корректности конфигурационного файла. В рамках данного анализа осуществляются следующие проверки: наличие или отсутствие конфигурационного файла во входящем сообщении, полнота конфигурационного файла (корректно и полностью считывается из сетевого сообщения), проверка формата конфигурационного файла (соответствует ли формату, который обрабатывает модуль автоконфигурации (102)), наличие такого же конфигурационного файла в базе данных (104) с конфигурационными файлами.

В случае успешной проверки или в случае обнаружения конфигурационного файла в базе данных (104) модуль оркестрации запросов (101) передает сообщение далее в модули системы (а также техническую информацию по директивам реализации, если она есть в базе данных). В случае обнаружения ошибки модуль оркестрации запросов (101) формирует ответ с указанием ошибки.

Модуль автоконфигурации инфраструктуры в виртуальной среде (102) предназначен для автоматического развертывания инфраструктуры в виртуальной среде из конфигурационного файла пользователя, а также для взаимодействия с внешними инструментами по запросу пользователя или других модулей системы вне задачи автоматического создания инфраструктуры в виртуальной среде.

В качестве входных данных модуль (102) ожидает конфигурационный файл пользователя, в котором описываются необходимые параметры инфраструктуры. Конфигурационный файл представляет из себя структуру данных (yaml, json, xml), заранее сформированную для понимания системой (100), в которой определенным ключам (соответствующим модели данных системы и являющимся неизменяемыми) соответствуют какие-либо значения (указанные пользователем и являющиеся основными параметрами конфигурирования инфраструктуры). При этом файл состоит из разделов, каждому из которых со-

ответствует основной ключ, который соответствует названию стороннего инструмента, которым должен обрабатываться.

Модуль автоконфигурации запросов (102) может обрабатывать, по меньшей мере, следующие запросы:

запросы на создание инфраструктуры из конфигурационного файла,

запросы на предоставление дополнительной информации в рамках создания инфраструктуры,

запросы на предоставление дополнительной информации из баз данных сторонних инструментов,

запросы на предоставление информации по созданной инфраструктуре,

запросы на действия для ряда сторонних инструментов.

Модуль автоконфигурации инфраструктуры в виртуальной среде (102) реализует следующие алгоритмы.

Алгоритм подключения сторонних инструментов.

Подключение сторонних инструментов к системе (100) подразумевает интеграцию системы (100) с API подключаемого инструмента. Интеграция заключается в разработке скрипта, который реализует взаимодействие системы (100) с API подключаемого инструмента. Для подключения нового стороннего инструмента необходимо в конфигурационный добавить информацию об этом инструменте вида: <tool_name>:{<tool_call_directory><order_id>} - а также интеграционный скрипт, который содержит алгоритм проверки достаточности конфигурационного файла в части подключаемой системы который реализует проверку части конфигурационного файла, за реализацию которого ответственен данный внешний инструмент на предмет возможности реализации указанных действий и достаточности информации для реализации данных действий, путем сравнения структуры конфигурационного файла (в своей зоне ответственности) с шаблонной структурой, ожидаемой данным интеграционным скриптом, а также проверкой ключевых слов с ожидаемыми, алгоритм формирования запросов с требованиями дополнительной информации о развернутой инфраструктуре, директивы предоставления дополнительной информации о проведенных работах, директивы трансляции информации в конфигурационном файле в конкретные вызовы API внешнего инструмента. В случае отсутствия информации для реализации тех или иных действий, указанных в конфигурационном файле алгоритм формирования запросов с требованиями дополнительной информации о развернутой инфраструктуре, формирует запрос к модулю автоконфигурации (102) о предоставлении дополнительных данных от одного из отработавших ранее внешних инструментов (или для передачи запроса в модуль оркестрации запросов с целью получения данных из одной из баз данных).

Алгоритм упорядочивания процесса развертывания инфраструктуры и добавления динамической активности. При получении файла конфигурации модуль автоконфигурации (102) анализирует основные разделы файла конфигурации для сбора информации по необходимым слоям развертывания. Полученная информация сравнивается с информацией о доступных интеграционных скриптах. Информации об основных разделах присваивается значение <order_id> соответствующее значению из настроек системы (100), сформированных при добавлении интеграционных скриптов. После чего основные разделы с присвоенным <order_id> ранжируются по порядку от низшего к высшему. Полученный массив данных является упорядоченным перечнем запуска интеграционных скриптов развертывания инфраструктуры в виртуальной среде.

Алгоритм формирования директив из файла конфигурации пользователя. После алгоритма упорядочивания процесса развертывания инфраструктуры и добавления динамической активности модуль автоконфигурации (102) передает конфигурационный файл в интеграционный скрипт. Интеграционный скрипт осуществляет проверку переданного интеграционного файла и в случае успеха вызывает директивы, соответствующие вызовам API внешнего инструмента. Соответствие между конфигурационным файлом, директивой и вызовом API определяется на этапе разработки интеграционного скрипта и выражается в формате описания создаваемой части инфраструктуры в конфигурационном файле. При этом при запуске директивы и вызова API они записываются в отдельный файл, где указан порядок запуска, наименование и аргументы, передаваемые внешней системе (100), что позволяет в результате сформировать файл, содержащий всю техническую информацию о развертывании инфраструктуры (в том числе информацию, которая запрашивается интеграционными скриптами дополнительно в рамках своей работы).

Алгоритм оркестрации директив с целью развертывания инфраструктуры и добавления динамической активности.

После алгоритма упорядочивания процесса развертывания инфраструктуры и добавления динамической активности модуль автоконфигурации (102) передает конфигурационный файл в интеграционный скрипт, начиная с минимального порядкового номера. Интеграционный скрипт реализует алгоритм проверки достаточности конфигурационного файла в части подключаемой системы (100). В случае достаточности данных интеграционный скрипт преобразует конфигурационный файл в директивы и вызовы API внешнего инструмента. В случае недостаточности данных формирует с помощью алгоритма формирования запросов с требованиями дополнительной информации о развернутой инфраструктуре запрос о предоставлении дополнительных сведений и направляет его в модуль автоконфигурации (102).

Модуль автоконфигурации (102) после получения запроса о предоставлении дополнительных све-

дений перенаправляет его в нужный интеграционный скрипт. Интеграционный скрипт, получив запрос о предоставлении дополнительных сведений, реализует директивы предоставления дополнительной информации о проведенных работах и возвращает результат модулю автоконфигурации (102). Модуль автоконфигурации (102) возвращает результат запросившему сведения интеграционному скрипту. Интеграционный скрипт анализирует полученные сведения и либо формирует новый запрос на предоставление сведений (например, к другому модулю), либо формирует ошибку, либо преобразует конфигурационный файл в директивы и вызовы API внешнего инструмента, после чего возвращает ответ в модуль автоконфигурации (102), который формирует ответ по отработавшей части интеграционных скриптов и направляет его в модуль оркестрации запросов (101). После чего модуль автоконфигурации (102) переходит к следующему порядковому номеру и передает конфигурационный файл в следующий интеграционный скрипт.

Алгоритм изолирования созданной инфраструктуры.

Данный алгоритм запускается после развертывания инфраструктуры в виртуальной среде и механизмов мониторинга инфраструктуры в виртуальной среде, если в конфигурационном файле пользователя указана необходимость изоляции инфраструктуры. С целью развертывания инфраструктуры и добавления динамической активности между системой с внешними инструментами и виртуальной средой, где разворачивается инфраструктура, создается сетевое соединение, с помощью которого все инструменты реализуют свои задачи на виртуальных машинах в создаваемой инфраструктуре. Данное сетевое соединение создается с помощью директив, указанных в интеграционных скриптах. При этом скрипты также могут не использовать прямое сетевое соединение, а работать через доступные вызовы гипервизора. После окончания всех задач созданное соединение блокируется для любого использования (входящий/исходящий трафик) и создаваемая среда становится изолированной от любого сетевого доступа извне. В этом случае доступ в среду осуществляется только через гипервизор. В случае если есть необходимость осуществлять изменения уже после создания изолированной среды, модуль автоконфигурации (102) может развернуть внутри создаваемой среды заранее подготовленную виртуальную машину со всеми данными для внесения изменений в автоматическом режиме. Модуль проведения тестирования (103) предназначен для автоматического проведения тестирования событий кибербезопасности на основе информации алгоритмов тестирования, содержащейся в конфигурационном файле пользователя. Модуль (103) взаимодействует с модулем оркестрации запросов (101) и с созданной инфраструктурой в виртуальной среде.

В качестве входных данных модуль (103) ожидает конфигурационный файл пользователя, который состоит из трех разделов: алгоритм тестирования, описание инфраструктуры, в которой тестирование будет проводится и описание механизма мониторинга созданной инфраструктуры. Конфигурационный файл представляет из себя структуру данных (yaml, json, xml), заранее сформированную для понимания системой (100), в которой определенным ключам соответствуют какие-либо значения.

Описание инфраструктуры, в которой тестирование будет проводится, аналогично конфигурационному файлу для задачи автоматизированного развертывания инфраструктуры в виртуальной среде и может в том числе быть только указателем на сохраненный (готовый) конфигурационный файл в базе данных системы (100).

Описание механизма мониторинга инфраструктуры, в которой тестирование будет проводится, аналогично конфигурационному файлу для задачи организации мониторинга внутри созданной инфраструктуры в виртуальной среде и может в том числе быть только указателем на сохраненный (готовый) конфигурационный файл в базе данных системы (100)

Алгоритм тестирования может включать в себя следующие данные (но не ограничиваться):

данные о воздействии на инфраструктуру - представляет собой информацию о действии, которое необходимо выполнить в инфраструктуре для проведения тестирования (запуск вредоносного файла, запуск атаки, которая состоит из нескольких шагов, запуск того или иного трафика, выполнение того или иного действия пользователя рабочей станции, установка программы и т.д.); может являться указателем на сохраненное (готовое) воздействие в базе данных системы или набором ключ-значение для создания нового (зависит от внешнего инструмента, который используется для реализации воздействия),

данные о количестве повторений тестирования - количество запусков с нуля одного и того же тестирования (начиная с развертывания инфраструктуры в виртуальной среде и заканчивая ее уничтожением),

динамические переменные тестирования - значения в файле описания инфраструктуры, в которой тестирование будет проводится, которые подлежат изменению в рамках проведения тестирования (может быть в формате массива значений для ключа, если их несколько); вместе с количеством повторений подразумевается проведение тестирования с изменением определенных параметров инфраструктуры в том количестве, которое указано,

данные о старте тестирования - может быть одной из стадий работы системы (100) (например, окончание работы модуля автоконфигурации (102)),

данные о окончание тестирования - может быть одной из стадий работы системы (например, окончание работы модуля автоконфигурации (102)) или событием в системе мониторинга от развернутой инфраструктуры,

триггеры для дополнительных событий тестирования - может быть одной из стадий работы системы (например, окончание работы модуля автоконфигурации (102)),

дополнительные данные о событии тестирования - является аналогом "воздействия на инфраструктуру", но запускаемым по триггеру,

данные о типе проведения тестирования - автоматизированно или пошагово,

и любые другие данные для проведения тестирования в области кибербезопасности не ограничиваясь.

Модуль проведения тестирования (103) реализует следующие алгоритмы.

Алгоритм обработки сценария тестирования.

При получения конфигурационного файла проведения тестирования модуль проведения тестирования (103) формирует запросы к модулю оркестрации запросов (101) для проверки полученного конфигурационного файла и получения дополнительных данных. Если в конфигурационном файле проведения тестирования указаны ссылки на сохраненные конфигурационные файлы, то запрос формируется с целью поиска и загрузки этих файлов из баз данных. Если в конфигурационном файле указаны значения для запуска с помощью внешних инструментов, то запрос формируется с целью определения корректности и достаточности данных для выполнения (запрос отправляется на модуль оркестрации запросов (101), который перенаправляет его в базу данных или в модуль автоконфигурации инфраструктуры в виртуальной среде (102), откуда он попадает в интеграционный модуль внешней системы и проходит проверку с помощью алгоритма проверки достаточности конфигурационного файла в части подключаемой системы).

В случае успешного прохождения указанных проверок модуль (103) анализирует содержимое сценария тестирования, файла описания инфраструктуры и файла описания механизмов мониторинга инфраструктуры. В случае необходимости модуль проведения тестирования (103) обогащает описание инфраструктуры дополнительными данными (например, необходимость развернуть конкретный механизм мониторинга). В случае необходимости (наличия свойства изолированности инфраструктуры) модуль проведения тестирования (103) добавляет в описание инфраструктуры заранее подготовленную виртуальную машину, которая будет реализовывать функционал модуля (103) в изолированной среде. После успешной проверки модуль проведения тестирования (103) запускает алгоритм подготовки очереди задач.

Алгоритм подготовки очереди задач.

В рамках подготовки плана проведения тестирования модуль (103) готовит очередь задач, которые должны быть реализованы до начала проведения тестирования. Сформированный пользователем и системой (100) (на предыдущем этапе) конфигурационный файл разделяется на сценарий тестирования и описание инфраструктуры. В соответствии с триггерами событий (начало, окончание, завершение) формируются правила мониторинга данных событий, и конфигурация для их реализации добавляется в описание инфраструктуры.

Полученный файл описания инфраструктуры обрабатывается в соответствии с алгоритмом оптимизации инфраструктуры (если указано в сценарии тестирования). В соответствии с динамическими переменными создается несколько копий описания инфраструктуры (каждая из которых содержит свое значение динамической переменной). После чего для каждой пары формируется задача, которая состоит из следующих подзадач: развернуть инфраструктуру, ожидание триггеров запуска воздействия, запуска дополнительных воздействий, окончания тестирования, реакция на триггеры, уничтожение инфраструктуры, передача данных (результатов) в централизованное хранилище. Каждая задача тиражируется в количестве, указанном в сценарии тестирования и добавляется в очередь задач. После чего запускается алгоритм проведения тестирования.

Алгоритм проведения тестирования.

Модуль проведения тестирования (103) передает описание инфраструктуры и сценарий тестирования на обработку алгоритму оптимизации инфраструктуры в части определения возможности параллелизма запуска тестирования. В случае определения такой возможности модуль (103) создает необходимое количество потоков и за один раз получает равное количество задач (в порядке очереди) из пула задач. В случае невозможности параллелизма модуль (103) берет по одной задаче из пула задач. После получения задачи модуль (103) формирует запрос на развертывание инфраструктуры в виртуальной среде для модуля автоконфигурации инфраструктуры (102).

Модуль автоконфигурации (102) разворачивает инфраструктуру и сообщает об этом модулю проведения тестирования (103). Модуль проведения тестирования (103) регистрирует события окончания развертывания инфраструктуры и, в случае присутствия изолированности, формирует запрос модулю конфигурации (102) на развертывание дополнительного сегмента управления с экземпляром модуля проведения тестирования (103).

Модуль автоконфигурации (102) разворачивает сегмент управления и сообщает об этом модулю проведения тестирования (103). Модуль проведения тестирования (103) регистрирует события окончания развертывания сегмента управления и формирует запрос на добавление механизмов мониторинга развернутой инфраструктуры для модуля мониторинга инфраструктуры (105).

Модуль мониторинга инфраструктуры (105) организует механизмы мониторинга инфраструктуры (с учетом изолированности или не изолированности инфраструктуры) и сообщает об этом модулю про-

ведения тестирования (103).

Модуль проведения тестирования (103) регистрирует события окончания развертывания механизмов мониторинга инфраструктуры и формирует запрос модулю автоконфигурации (102) на изолирование созданной инфраструктуры, если это предусмотрено сценарием тестирования.

Модуль проведения тестирования (103), в случае необходимости изолирования инфраструктуры, ожидает завершения данной задачи от модуля автоконфигурации (102) и переходит в режим ожидания триггеров событий. При срабатывании триггера модуль (103) реализует алгоритм обработки событий в рамках проведения тестирования. При обнаружении события завершения тестирования модуль (103) формирует запрос на сбор полученных данных (если данные не выводились в централизованную базу данных ранее).

После сбора данных модуль (103) формирует запрос модулю автоконфигурации (102) на уничтожение созданной инфраструктуры.

После уничтожения созданной инфраструктуры модуль проведения тестирования (103) переходит к выполнению следующей задачи из очереди задач тестирования. Далее после выполнения последней задачи модуль (103) формирует ответ о выполнении тестирования (в том числе с помощью сбора и анализа полученных данных в формате отчета) и отправляет ответ пользователю.

Алгоритм формирования обратной связи с инфраструктурой во время тестирования.

Модуль проведения тестирования (103) на этапе обработки сценария тестирования анализирует достаточность описания механизма мониторинга инфраструктуры. В случае необходимости модуль (103) конвертирует триггеры, описанные в сценарии тестирования в правила фильтрации событий мониторинга и действия по результату фильтрации событий мониторинга, которые добавляются в конфигурационный файл для модуля мониторинга инфраструктуры (105), который реализует заданные параметры во время выполнения своей задачи. В случае отсутствия изолированности созданной инфраструктуры модуль проведения тестирования (103) генерирует конфигурационный файл для модуля мониторинга инфраструктуры (105) таким образом, что действием по результату фильтрации событий мониторинга будет являться отправка сообщения в модуль проведения тестирования (103) по стандартным каналам связи с использованием таких атрибутов сетевого взаимодействия как IP-адрес, порт и т.д. В случае присутствия изолированности созданной инфраструктуры модуль проведения тестирования (103) формирует конфигурационный файл и запрос для модуля автоконфигурации (102), в рамках которого происходит (с помощью алгоритмов, описанных в модуле автоконфигурации (102)) создание среды управления, интегрированной с изолированной инфраструктурой и со средой мониторинга, созданной с помощью модуля мониторинга инфраструктуры (105). Конфигурационный файл для модуля мониторинга инфраструктуры (105) генерируется таким образом, что действием по результату фильтрации событий мониторинга будет являться отправка сообщения в модуль проведения тестирования (103) по каналам связи, созданным внутри изолированной инфраструктуры с использованием таких атрибутов сетевого взаимодействия, как IP-адрес, порт и т.д.

Сообщение от механизма мониторинга, полученное модулем проведения тестирования (103) внутри изолированной инфраструктуры будет соответствовать конкретному триггеру в описании сценария тестирования.

В случае обнаружения триггера модуль проведения тестирования (103) определяет, к какому событию (описанному в сценарии тестирования: старт, окончание, запуск вредоносной активности, запуск определенного действия пользователя и т.д.) относится данный триггер, и реализует данное событие в развернутой инфраструктуре. В случае обнаружения события завершения тестирования модуль проведения тестирования (103) отправляет запрос платформе виртуализации на уничтожение всех виртуальных машин в рамках данной инфраструктуры.

Алгоритм оптимизации инфраструктуры.

Данный алгоритм осуществляет две активности: оптимизацию параллельного запуска задач из очереди и оптимизацию построенной инфраструктуры.

Оптимизация параллельного запуска задач из очереди.

Алгоритм получает описание инфраструктуры и сценарий тестирования. Из сценария тестирования алгоритм путем умножения количества значений динамических переменных на количество повторения тестирования получает общее количество запусков тестирования (длину очереди задач). Из описания инфраструктуры формируются требования к мощности создаваемой инфраструктуры (максимальные значения RAM и CPU). Если эти значения не указаны в явном виде, то формируется запрос к базе данных с целью определения настроек шаблонов виртуальных машин, из которых будет создана инфраструктура. После этого свободные ресурсы платформы виртуализации делятся на 2 и затем на требования к мощности одной инфраструктуры. Результатом, округленным в меньшую сторону, является количество одновременных потоков по созданию инфраструктуры.

Оптимизация построения инфраструктуры.

Алгоритм получает описание инфраструктуры и сценарий тестирования. Алгоритм определяет наличие свободных ресурсов после развертывания инфраструктуры. В случае наличия свободных ресурсов в конфигурационный файл добавляется действие по развертыванию симуляции дополнительных сетевых

сегментов (количество определяется в зависимости от наличия свободных ресурсов и указания в разделе оптимизации в сценарии тестирования). Маршрутизация и месторасположение сегментов выбирается случайно, но с оптимальным распределением относительно всех остальных сетевых сегментов развертываемой инфраструктуры.

База данных (104) или базы данных используются, по меньшей мере, для хранения конфигурационных файлов пользователей, которые описывают инфраструктуру в виртуальной среде и/или тестирование, контроля версий конфигурационных файлов пользователей, хранения информации, относящейся к учетным данным пользователей, ролевым моделям, правам доступа пользователей, а также информации, на основании которой может проводится динамическая авторизация запросов пользователей (инвентаризационные данные об устройствах пользователей, секреты для взаимодействия модулей системы, сертификаты, история поведения пользователей и модулей системы и т.д.), хранения информации, необходимой для работы модуля автоконфигурации инфраструктуры (102), хранения файлов, предназначенных для реализации активности в рамках развертывания инфраструктуры или воздействия в рамках проведения тестирования (например, интеллектуальные агенты, реализующие симуляцию поведения пользователей на виртуальных машинах в развернутой инфраструктуре), хранение информации, связанной с областью кибербезопасности: данные по уязвимостям, патчам безопасности, зловредным файлам, атакам и т.д., хранение информации конфигурационного характера для функционирования системы.

Модуль мониторинга инфраструктуры (105) предназначен для организации полного цикла обработки событий, возникающих в объектах инфраструктуры, созданной с помощью модуля автоконфигурации (102). Модуль (105) взаимодействует с модулем оркестрации запросов (101), с созданной инфраструктурой в виртуальной среде и модулем проведения тестирования (103).

В качестве входных данных модуль (105) ожидает конфигурационный файл пользователя с описание требуемого механизма мониторинга инфраструктуры.

Конфигурационный файл представляет из себя структуру данных (yaml, json, xml), заранее сформированную для понимания системой (100), в которой определенным ключам соответствуют какие-либо значения.

Описание механизма мониторинга инфраструктуры аналогично конфигурационному файлу для задачи автоматизированного развертывания инфраструктуры в виртуальной среде и может в том числе быть только указателем на сохраненный (готовый) конфигурационный файл в базе данных системы (100).

Алгоритм организации мониторинга внутри созданной инфраструктуры может включать в себя следующие данные (но не ограничиваться):

данные о типе механизма мониторинга - представляет собой информацию о механизме мониторинга, который требуется использовать: мониторинг в реальном времени - с помощью стандартных средств (Splunk, Qradar, ELK, Zabbix и т.д.), который осуществляется непосредственно в развернутой инфраструктуре, мониторинг по результатам активности в инфраструктуре - сбор данных для их последующего анализа во внешней системе (PCAP, logs и т.д.), мониторинг с помощью гипервизора платформы виртуализации - мониторинг на уровне гипервизора в реальном времени с целью усиления уровня изолированности и повышению уровня осведомленности,

данные об объектах инфраструктуры, которые необходимо мониторить - представляет собой информацию об объектах развернутой инфраструктуры (под объектом подразумевается любая созданная сущность: виртуальная машина, сеть передачи данных, операционная система, приложение и т.д.), а так же конкретные журналы событий этих объектов (например, для операционной системы Windows это могут быть журналы Application, Security, System; для сети передачи данных - мета-данные о сетевых потоках, изменение конфигурации сетевого оборудования, "полезная нагрузка" сетевого запроса, например в процессе передачи данных по протоколу HTTP и т.д.),

Алгоритм обработки событий мониторинга может включать в себя следующие данные (но не ограничиваться):

данные о правилах фильтрации событий мониторинга - представляет собой информацию о выборе отдельных событий, которые поступили от объекта мониторинга, например событие успешного входа пользователя в ОС Windows, которое поступило от объекта ОС из журнала Security. Правила фильтрации могут использовать любой контекст, доступный из описания инфраструктуры, например помимо указания объектов конкретного типа можно указать и конкретный объект: "виртуальная машина с идентификационным номером X", "пользователь Y" и т.д.,

данные о действиях по результату фильтрации событий мониторинга - представляет собой информацию о действии, которое необходимо выполнить в результате обнаружения события, попадающего под тот или иной фильтр. Такими действиями могут быть: генерация нового события, отправка сигнала по заданному направлению, сохранение в базу данных, отброс события (уничтожение) и т.д.

Модуль мониторинга инфраструктуры (105) реализует следующие алгоритмы.

Алгоритм организации мониторинга внутри созданной инфраструктуры. Данный алгоритм запускается после развертывания инфраструктуры в виртуальной среде, если в конфигурационном файле пользователя указана необходимость мониторинга или по запросу от модуля проведения тестирования (103),

если для его корректной работы требуется механизм мониторинга инфраструктуры. Пользователь может указать один из трех типов механизмов мониторинга внутри созданной инфраструктуры. В случае если указан мониторинг в реальном времени (фиг. 2), модуль мониторинга инфраструктуры (105) формирует запрос к модулю автоконфигурации (102) для добавления внутрь развернутой инфраструктуры новой виртуальной машины и перестроения сетевой топологии. Добавленная виртуальная машина размещается в развернутую сетевую топологию в виде отдельного сегмента сети, маршрутизация в который жестко ограничена по всем участникам сетевого обмена: входящее соединение разрешено по узкому перечню портов, исходящее соединение запрещено. Аналогичные действия производятся с сетевым оборудованием для сбора сетевого трафика. Данные собираются и хранятся в той же инфраструктуре, где проходит тестирование или любая другая активность, созданная пользователем. При данном типе мониторинга пользователь может самостоятельно выбрать или добавить систему мониторинга.

В случае если указан мониторинг по результатам активности в инфраструктуре (фиг. 3), модуль мониторинга инфраструктуры (105) формирует запрос к модулю автоконфигурации (102) для создания отдельной среды для мониторинга, в которой разворачиваются виртуальные машины с системой мониторинга и базой данных (данная отдельная среда может не создаваться каждый раз с нуля, а использоваться готовая среда с развернутыми системами, что обеспечивает хранение различных наборов данных централизованно в единой базе данных). Между развернутой инфраструктурой согласно конфигурационному файлу и инфраструктурой мониторинга создается заранее подготовленная виртуальная машина, которая является шлюзом между двумя средами -"перекладчик", на данной виртуальной машине разворачивается специализированное программное обеспечение, которое является фильтром проходящих данных и реализует функционал антивирусного программного обеспечения, средства предотвращения вторжений и т.д. - любые данные, передающиеся между двумя средами проходят сквозь данный фильтр и в случае обнаружения угрозы передача блокируется. Добавленная виртуальная машина размещается в виде отдельного сегмента сети для двух сред, маршрутизация в которых жестко ограничена по всем участникам сетевого обмена: входящее соединение разрешено по узкому перечню портов, исходящее соединение запрещено для развернутой из конфигурационного файла пользователя среды; исходящее соединение разрешено по узкому перечню портов, входящее соединение запрещено для среды мониторинга. В среду, развернутую из конфигурационного файла пользователя, также добавляется заранее подготовленная виртуальная машина с программным обеспечением базы данных, которая выполняет роль временной базы данных для снижения влияния на производительность.

Добавленная виртуальная машина размещается в развернутую из конфигурационного файла пользователя, сетевую топологию в виде отдельного сегмента сети, маршрутизация в который жестко ограничена по всем участникам сетевого обмена: входящее соединение разрешено по узкому перечню портов, исходящее соединение запрещено. Все виртуальные машины в инфраструктуре дополняются специализированным программным обеспечением и/или настройками для сбора и отправки журналов событий в систему мониторинга.

Аналогичные действия производятся с сетевым оборудованием для сбора сетевого трафика. Данные собираются и хранятся в той же инфраструктуре во временной базе данных, откуда "наборами" передаются в "перекладчик", который анализирует безопасность набора и передает его в среду мониторинга в централизованную базу данных, откуда система мониторинга их получает для дальнейшей обработки и анализа. При данном типе мониторинга пользователь может самостоятельно выбрать или добавить систему мониторинга (путем добавления стороннего инструмента с помощью алгоритма добавления стороннего инструмента), а также системы сбора и отправки журналов событий.

В случае если указан мониторинг с помощью гипервизора платформы виртуализации (фиг. 4), модуль мониторинга инфраструктуры (105) формирует запрос к модулю автоконфигурации (102) для создания отдельной среды для мониторинга, в которой разворачиваются виртуальные машины с системой мониторинга и базой данных (данная отдельная среда может не создаваться каждый раз с нуля, а использоваться готовая среда с развернутыми системами, что обеспечивает хранение различных наборов данных централизованно в единой базе данных). Сбор данных осуществляется на уровне гипервизора. Так как все сетевые взаимодействия, а также операции с файловой системой виртуальной машины осуществляются через гипервизор, то эти данные могут быть собраны при прохождении через него и переданы в другую виртуальную машину, которая размещается в развернутой среде мониторинга. Также на уровне гипервизора производится фильтрация данных на предмет зловредности и, в случае обнаружения подозрительных данных, они блокируются. После передачи данных в среду мониторинга они сохраняются в централизованной базе данных, откуда попадают в систему мониторинга для дальнейшей обработки и анализа.

Алгоритм обработки событий мониторинга.

Данный алгоритм запускается после организации механизма мониторинга инфраструктуры и предназначен для задания фильтрации полученных в результате мониторинга событий и действий по результатам фильтрации. Все действия осуществляются согласно конфигурационному файлу, созданному пользователем или модулем проведения тестирования (103). Модуль (105) анализирует конфигурационный файл и транслирует его конфигурацию (в части фильтрации) в конкретные директивы в рамках заданно-

го ранее механизма мониторинга. Директивами в данном случае могут быть: YARA-описание фильтрации событий мониторинга; регулярные выражения; булева логика и поиск соответствий в парах ключ-значение; булева логика и поиск значений в тэгах таких форматов структуризации данных как YAML, JSON, XML и др.

Модуль (105) анализирует конфигурационный файл и транслирует его конфигурацию (в части действий по результатам фильтрации) в конкретные директивы в рамках заданного ранее механизма мониторинга. Директивами в данном случае могут быть: отправка syslog-сообщения определенного формата на определенный адрес; запуск внешнего заранее подготовленного скрипта; отправка сообщения по сети передачи данных в формате YAML, JSON, XML и др.

Основной процесс работы системы (100) можно разбить на два независимых направления: автоматизированное развертывание инфраструктуры в виртуальной среде и проведение тестирования.

Задача автоматизированного развертывания инфраструктуры в виртуальной среде подразумевает создание инфраструктуры с нуля согласно конфигурационному файлу и предоставление пользователю возможности работать в созданной инфраструктуре, как если бы он сделал ее самостоятельно в ручном режиме. Пользователь создает конфигурационный файл в формате YAML, XML или JSON и отправляет команду на создание инфраструктуры из этого файла в виртуальной среде. Загрузка файла осуществляется через веб-интерфейс или через интерфейс командной строки. Клиентская сторона формирует запрос, в который добавляет конфигурационный файл и отправляет его на прокси-сервер. Прокси-сервер осуществляет авторизацию запроса в соответствии с собственными внутренними настройками. В случае успеха перенаправляет запрос в модуль оркестрации запросов (101) серверной части, в случае неудачной авторизации - запрос отклоняется.

Модуль оркестрации запросов (101) осуществляет первичную проверку конфигурационного файла с помощью алгоритма первичной проверки конфигурационного файла пользователя. В случае успешной проверки запрос передается в модуль автоконфигурации инфраструктуры (102), в случае обнаружения ошибки в конфигурационном файле запрос возвращается клиенту с указанием ошибки. Модуль автоконфигурации (102), получив конфигурационный файл, с помощью алгоритма упорядочивания процесса развертывания инфраструктуры и добавления динамической активности формирует порядок использования тех или иных сторонних инструментов для безошибочного построения слоев инфраструктуры, в порядке сформированной очереди, раздел конфигурационного файла преобразовывается в конкретные директивы для внешнего инструмента.

Сформированные на предыдущем этапе директивы отправляются на выполнение во внешний инструмент, ответственный за реализацию данных директив. Внешний инструмент реализует полученные директивы непосредственно в виртуальной среде или на развернутых виртуальных машинах и подготавливает результат выполнения операций, в который включает некоторые (любые) дополнительные данные, сформированные в результате выполнения директив.

Результат выполнений возвращается в модуль автоконфигурации (102), где цикл обработки повторяется вновь со следующим элементом сформированной очереди, при этом с помощью алгоритма оркестрации директив с целью развертывания инфраструктуры и добавления динамической активности каждый внешний инструмент может сформировать запрос дополнительных данных от инструментов, которые уже завершили свою работу (необходимо для случаев, когда данные для работы инструмента могут быть получены только в результате работы другого инструмента - например, установка программного обеспечения по IP-адресу при динамическом назначении IP-адресов во время создания виртуальных машин).

Модуль автоконфигурации (102) после отработки всех внешних инструментов формирует и отправляет результат работы в модуль оркестрации запросов (101). Модуль оркестрации запросов (101) с помощью алгоритма взаимодействия с базами данных сохраняет конфигурационный файл и все технические данные, сформированные по результату выполнения задачи, в базу данных.

Модуль оркестрации запросов (101) анализирует исполнение конфигурационного файла пользователя, в случае обнаружения необходимости организации механизмов мониторинга инфраструктуры модуль (101) передает запрос в модуль мониторинга инфраструктуры (105).

Модуль мониторинга инфраструктуры (105) анализирует конфигурационный файл и с помощью формирования запросов к модулю автоконфигурации (102) добавляет в созданную инфраструктуру требуемый механизм мониторинга.

Модуль мониторинга инфраструктуры (105) после завершения всех своих задач формирует и отправляет результат работы в модуль оркестрации запросов (101). Модуль оркестрации запросов (101) с помощью алгоритма взаимодействия с базами данных сохраняет конфигурационный файл и все технические данные, сформированные по результату выполнения задачи, в базу данных.

Задача проведения тестирования включает в себя задачу автоматизированного развертывания инфраструктуры в виртуальной среде, а также ряд других задач, связанных с тестированием: воздействие на инфраструктуру согласно сценарию тестирования (например, автоматизированный запуск атаки), дополнительные события, возникающие в ходе воздействия на инфраструктуру (например, изменение таблицы блокировки сетевых соединений как реакция на выявления подозрительного трафика), повторение тестирования согласно плану (например, проведение одного и того же тестирования 10 раз и сравнение

результатов или сбор наборов данных), повторение тестирования с незначительным изменением виртуальной инфраструктуры (например, запуск одной и той же атаки в инфраструктуре 10 раз, но каждый раз настройки одной из политик домена в инфраструктуре будут изменяться).

Процесс работы системы (100) при проведении тестирования происходит поэтапно. Пользователь создает конфигурационный файл в формате YAML, XML или JSON и отправляет команду на проведение тестирования из этого файла в виртуальной среде. Загрузка файла осуществляется через веб-интерфейс или через интерфейс командной строки. Клиентская сторона формирует запрос, в который добавляет конфигурационный файл и отправляет его на прокси-сервер. Прокси-сервер осуществляет авторизацию запроса в соответствии с собственными внутренними настройками. В случае успеха перенаправляет запрос в модуль оркестрации запросов (101) серверной части, в случае неудачной авторизации - запрос отклоняется.

Далее модуль оркестрации запросов (101) передает конфигурационный файл в модуль проведения тестирования (103). Модуль проведения тестирования (103) подготавливает необходимые данные и проводит тестирование, для этого он может обращаться к модулю оркестрации запросов (101), который в свою очередь перенаправляет запрос в модуль автоконфигурации инфраструктуры (102), модуль мониторинга инфраструктуры (105) или в базу данных (104), а также напрямую в созданную виртуальную инфраструктуру.

Конфигурационный файл проведения тестирования оценивается на выполнимость с помощью алгоритма обработки сценария тестирования. После чего с помощью алгоритма подготовки плана проведения тестирования формируется план проведения тестирования: какая будет использоваться инфраструктура, какие варианты изменения инфраструктуры, если есть динамические переменные, как будет запускаться воздействие (удаленно без изоляции, или необходимо развернуть копию, запускающую воздействие внутри инфраструктуры), события реакции на триггеры (аналогично -удаленно или внутри), количество повторений, событие окончания тестирования и т.д. После формирования плана модуль проведения тестирования (103) рассчитывает с помощью алгоритма оптимизации инфраструктуры наилучший вариант проведения тестирования и одновременный запуск нескольких вариантов (например, когда указано количество запусков или динамические переменные) тестирования. После этого модуль (103) начинает осуществлять план проведения тестирования, для этого в несколько потоков (если возможно) осуществляются следующие действия: создание инфраструктуры в виртуальной среде, запуск воздействия с помощью модуля автоконфигурации (102), создание механизмов мониторинга инфраструктуры с помощью модуля мониторинга инфраструктуры (103), изолирование инфраструктуры, если необходимо, с помощью модуля автоконфигурации (102), мониторинг триггеров событий, запуск событий с помощью модуля автоконфигурации (102) или напрямую, ожидание событий окончания тестирования, сбор данных с помощью механизма мониторинга, уничтожение виртуальной инфраструктуры, повтор с созданием новой виртуальной инфраструктуры (если предполагается несколько запусков или динамические переменные).

По окончанию проведения тестирования данные предоставляются пользователю аналогично процессу работы в рамках автоматизированного развертывания инфраструктуры в виртуальной среде.

На фиг. 5 представлен пример общего вида вычислительного устройства (300), для реализации системы (100).

В общем случае вычислительное устройство (300) содержит объединенные общей шиной информационного обмена один или несколько процессоров (301), средства памяти, такие как ОЗУ (302) и ПЗУ (303), интерфейсы ввода/вывода (304), устройства ввода/вывода (1105), и устройство для сетевого взаимодействия (306). Процессор (301) (или несколько процессоров, многоядерный процессор и т.п.) может выбираться из ассортимента устройств, широко применяемых в настоящее время, например, таких производителей как Intel™, AMD™, Apple™, Samsung Exynos™, MediaTEK™, Qualcomm Snapdragon™ и т.п. Под процессором или одним из используемых процессоров в устройстве (300) также необходимо учитывать графический процессор, например GPU NVIDIA или Graphcore, тип которых также является пригодным для полного или частичного выполнения исполнения системы (100), а также может применяться для обучения и применения моделей машинного обучения в различных информационных системах.

ОЗУ (302) представляет собой оперативную память и предназначено для хранения исполняемых процессором (301) машиночитаемых инструкций для выполнение необходимых операций по логической обработке данных. ОЗУ (302), как правило, содержит исполняемые инструкции операционной системы и соответствующих программных компонент (приложения, программные модули и т.п.). При этом в качестве ОЗУ (302) может выступать доступный объем памяти графической карты или графического процессора.

ПЗУ (303) представляет собой одно или более устройств постоянного хранения данных, например жесткий диск (HDD), твердотельный накопитель данных (SSD), флэш-память (EEPROM, NAND и т.п.), оптические носители информации (CD-R/RW, DVD-R/RW, BlueRay Disc, MD) и др.

Для организации работы компонентов вычислительного устройства (300) и организации работы внешних подключаемых устройств применяются различные виды интерфейсов В/В (304). Выбор соответствующих интерфейсов зависит от конкретного исполнения вычислительного устройства, которые могут представлять собой, не ограничиваясь: PCI, AGP, PS/2, IrDa, FireWire, LPT, COM, SATA, IDE, Lightning, USB (2.0, 3.0, 3.1, micro, mini, type C), TRS/Audio jack (2.5, 3.5, 6.35), HDMI, DVI, VGA, Display

Port, RJ45, RS232 и т.п.

Для обеспечения взаимодействия пользователя с вычислительным устройством (300) применяются различные средства (305) В/В информации, например клавиатура, дисплей (монитор), сенсорный дисплей, тач-пад, джойстик, манипулятор мышь, световое перо, стилус, сенсорная панель, трекбол, динамики, микрофон, средства дополненной реальности, оптические сенсоры, планшет, световые индикаторы, проектор, камера, средства биометрической идентификации (сканер сетчатки глаза, сканер отпечатков пальцев, модуль распознавания голоса) и т.п.

Средство сетевого взаимодействия (306) обеспечивает передачу данных посредством внутренней или внешней вычислительной сети, например Интранет, Интернет, ЛВС и т.п. В качестве одного или более средств (306) может использоваться, но не ограничиваться: Ethernet карта, GSM модем, GPRS модем, LTE модем, 5G модем, модуль спутниковой связи, NFC модуль, Bluetooth и/или BLE модуль, Wi-Fi модуль и др. Представленные материалы заявки раскрывают предпочтительные примеры реализации технического решения и не должны трактоваться как ограничивающие иные, частные примеры его воплощения, не выходящие за пределы испрашиваемой правовой охраны, которые являются очевидными для специалистов соответствующей области техники.

ФОРМУЛА ИЗОБРЕТЕНИЯ

1. Автоматизированная система тестирования событий кибербезопасности, содержащая

модуль оркестрации запросов, выполненный с возможностью перенаправления запросов пользователей к модулям системы, причем пользовательский запрос содержит параметры формирования алгоритмов тестирования;

модуль автоконфигурации инфраструктуры в виртуальной среде, выполненный с возможностью автоматического развертывания инфраструктуры в виртуальной среде из конфигурационного файла пользователя и взаимодействия с внешними инструментами;

модуль проведения тестирования, выполненный с возможностью автоматического тестирования событий кибербезопасности на основе информации алгоритмов тестирования, содержащейся в конфигурационном файле пользователя;

модуль мониторинга инфраструктуры, выполненный с возможностью автоматического добавления механизмов мониторинга внутри созданной инфраструктуры и обработки событий, возникающих в результате мониторинга активности объектов инфраструктуры;

базу данных, содержащую конфигурационные файлы пользователей с заданными параметрами алгоритмов для тестирования событий кибербезопасности.

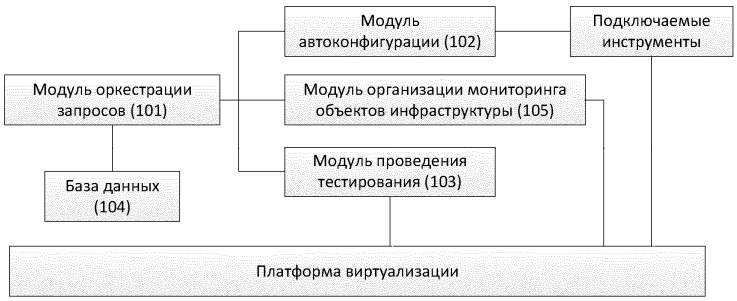
2. Система по п.1, характеризующаяся тем, что содержит прокси сервер, выполняющий алгоритмы безопасного соединения.

3. Система по п.1, характеризующаяся тем, что модуль проведения тестирования выполняет алгоритм обработки сценария тестирования.

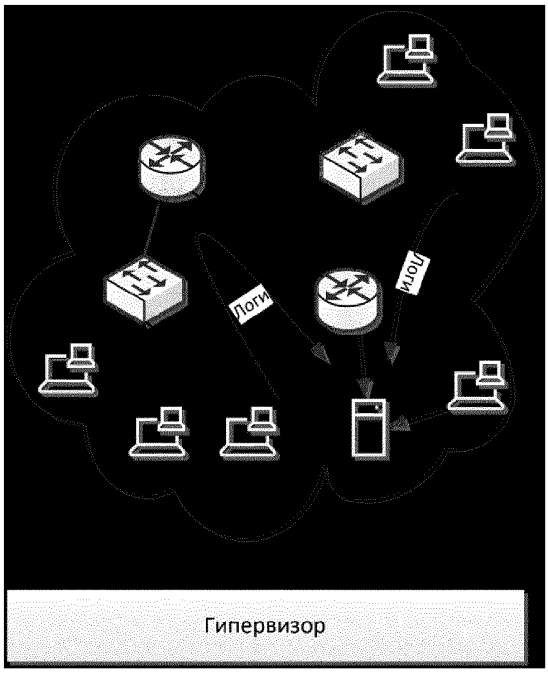
4. Система по п.1, характеризующаяся тем, что модуль проведения тестирования выполняет алгоритм подготовки очереди задач, которые должны быть реализованы до начала тестирования.

5. Система по п.1, характеризующаяся тем, что модуль проведения тестирования выполняет алгоритм формирования обратной связи с инфраструктурой во время тестирования.

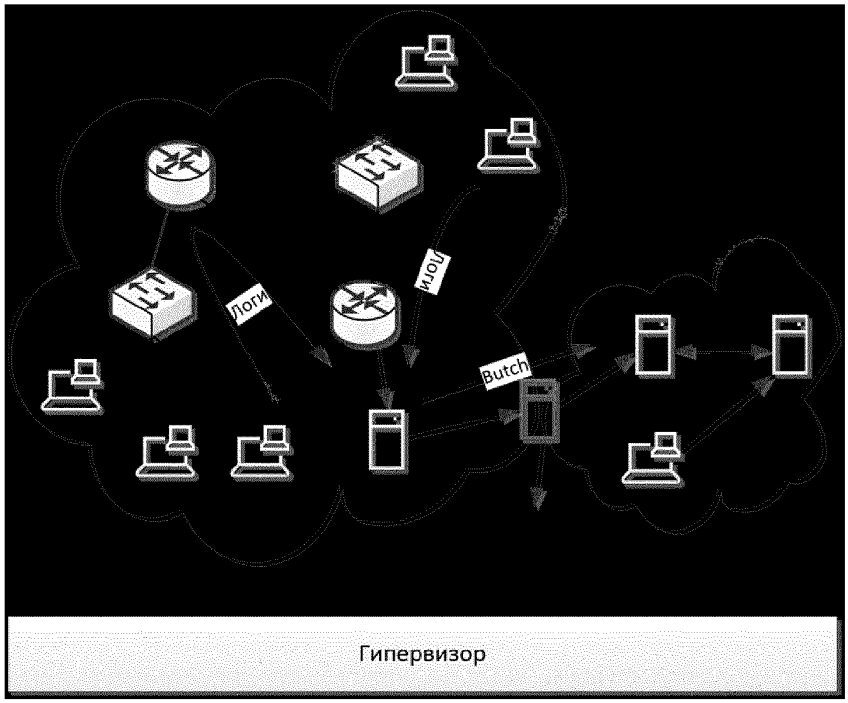
6. Система по п.1, характеризующаяся тем, что модуль проведения тестирования выполняет алгоритм оптимизации инфраструктуры в виртуальной среде.



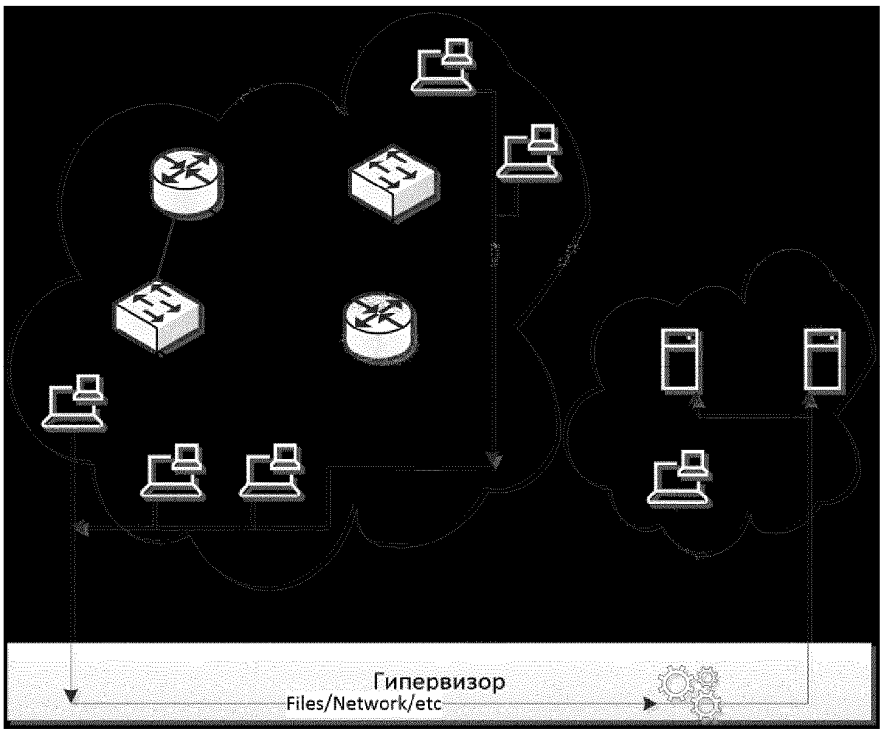
Фиг. 1



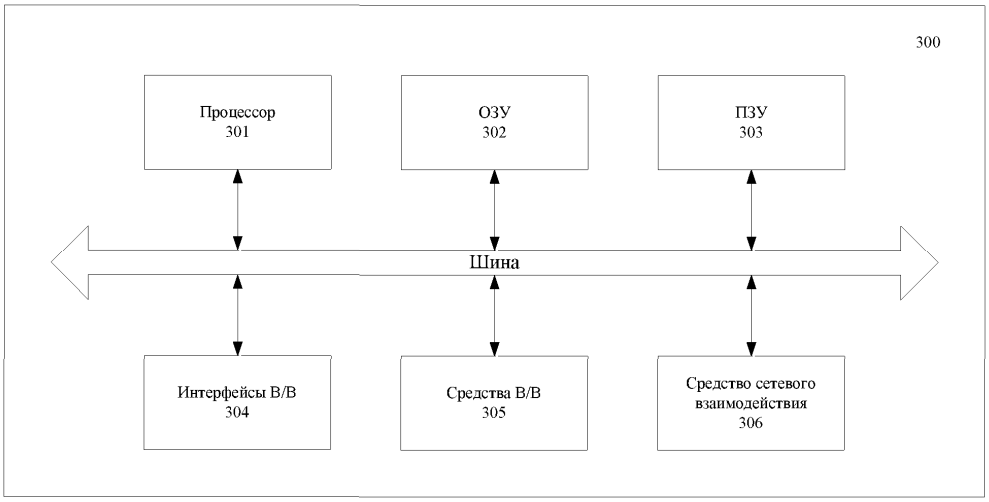
Фиг. 2



Фиг. 3


Гипервизор
Files/Network/etc

Фиг. 4


300
Процессор
301
ОЗУ
302
ПЗУ
303
Шина
Интерфейсы В/В
304
Средства В/В
305
Средство сетевого взаимодействия
306

Фиг. 5