Информатика, вычислительная техника и управление

УДК 004.89

ПРОГРАММНАЯ РЕАЛИЗАЦИЯ И ИССЛЕДОВАНИЕ СИСТЕМЫ ИНТЕЛЛЕКТУАЛЬНО-АДАПТИВНОГО УПРАВЛЕНИЯ ИНФОРМАЦИОННОЙ ИНФРАСТРУКТУРОЙ ПРЕДПРИЯТИЯ

Е.А. Басыня

Новосибирский государственный технический университет,
Научно-исследовательский институт информационно-коммуникационных технологий
Россия, 630073, г. Новосибирск, пр-т К. Маркса, 20

Аннотация. *Описывается проблематика информационно-коммуникационного взаимодействия технических объектов и систем. Анализируются уязвимости стека протоколов TCP/IP, несовершенство операционных систем и прикладного программного обеспечения, сложность полноценной взаимной интеграции различных систем. Излагается следствие экономии вендоров сетевых решений на вычислительных мощностях в виде архитектурных ограничений и уязвимостей. Рассматривается проектирование, программная реализация и экспериментальное исследование авторской системы интеллектуально-адаптивного управления информационной инфраструктурой предприятия, функционирующей на основе ранее представленного одноименного метода. Применяется современный стек технологий в сочетании с гибкой методологией разработки. Научная новизна работы заключается в предлагаемой архитектуре комплексного программного продукта по управлению информационной инфраструктурой предприятия. Повышение эффективности, надежности, безопасности и отказоустойчивости функционирования технических объектов и систем достигается применением платформы автоматизации развертывания и управления приложениями в среде многослойной виртуализации. Осуществляется изоляция рабочих сервисов с возможностью автоматического перезапуска любого компонента без нарушения штатного режима работы инфраструктуры. Снимается ряд ограничений, присущих существующим решениям: число одновременно поддерживаемых VPN-подключений, задаваемых политик, правил. Повышение уровня конфиденциальности сетевого взаимодействия, как и противодействия анонимным несанкционированным возмущениям, достигается полноценной интеграцией с оверлейными технологиями и сетями, поддержкой широкого спектра современных криптоустойчивых алгоритмов шифрования с возможностью многослойной инкапсуляции. Для оценки эффективности разработанной системы приводится экспериментальный сравнительный анализ с различными существующими решениями: комплексными межсетевыми экранами, маршрутизаторами, системами обнаружения и предотвращения вторжений, а также другими инструментами управления трафиком.*

Басыня Евгений Александрович (к.т.н.), доцент Новосибирского государственного технического университета, директор Научно-исследовательского института информационно-коммуникационных технологий.

Ключевые слова: *интеллектуально-адаптивное управление, системный анализ, обработка, сетевой трафик, локальные информационные процессы, TCP/IP, IDS/IPS, SIEM, UTM, NGFW.*

Введение

Автоматизацию бизнес-процессов любого предприятия не стоит рассматривать без учета аспектов информационной безопасности, актуальность которых непрерывно возрастает. Только за первые восемь месяцев 2019 года количество зарегистрированных киберпреступлений в России показало годовой рост на 66,8 % по данным Генпрокуратуры.

Одной из ключевых проблем в этой области является отсутствие комплексного подхода к безопасному системному анализу, обработке и управлению данными в рамках информационной инфраструктуры предприятия. Мировые вендоры инфокоммуникационных решений реализуют стек управляемого сетевого оборудования от коммутаторов до межсетевых экранов. Другие компании предоставляют продукты в области системного и сетевого администрирования. Третьи – в области информационной безопасности. Четвертые – в области автоматизации бизнес-процессов. Данный перечень включает десятки наименований. Анализ проблематики предметной области выводит на первый план ряд существенных недостатков индустрии:

– уязвимости стека протоколов TCP/IP [1, 2] (жесткая логика функционирования, отсутствие проверки подлинности субъектов взаимодействия в базовых протоколах и мн. др. В качестве примера стоит привести возможность подмены DHCP сервера <англ. Dynamic Host Configuration Protocol – протокол динамической настройки узла> через фальсификацию первого ответа);

– несовершенство программного обеспечения [3, 4] (неотлаженные технологические процессы разработки программных продуктов в ряде компаний: отсутствие корректного применения гибкой методологии с инспекцией / рецензированием кода, проверкой всех технических параметров автотестами и осуществлением непрерывной интеграции <англ. Continuous Integration & Continuous Delivery, CI/CD>);

– несовершенство операционных систем (помимо проблем предшествующего пункта стоит выделить возможность наличия бэкдоров <скрытых инструментов несанкционированного управления и сбора данных об объекте> в проприетарных продуктах с закрытым исходным кодом, а также сложность распределенной разработки решений с открытым исходным кодом на энтузиазме. В качестве примера стоит привести уязвимость в командной оболочке bash CVE-2014-6271, эксплуатирующую некорректное определение функций и обработку переменных окружения. Она просуществовала 22 года, позволяя удаленно выполнять код);

– сложность полноценной взаимной интеграции продуктов [5, 6] (даже внутри одного модельного ряда крупной компании встречаются разные версии решений с различными незадокументированными программными интерфейсами приложений <англ. Application programming interface>. Это связано с отсутствием системы контроля версий при использовании аутсорсинга в разных компаниях);

– экономия на вычислительных мощностях (использование слабой аппаратной базы MIPS/ARM <англ. Microprocessor without Interlocked Pipeline Stages и Advanced RISC Machine> приводит к вынужденному применению устаревших компонентов программной реализации, а также алгоритмов шифрования с низкой криптографической стойкостью [7–10]. Например, DES или 3DES <англ. Triple Data Encryption Standard> вместо AES <Advanced Encryption Standard>);

– архитектурные уязвимости [11, 12]: неправильный выбор стека технологий и компонентов, отсутствие изоляции модулей, отсутствие корректной интеграции компонентов (следствие: ограничение на количество задаваемых политик, размер адресного пула правил, скорость обработки данных, порог надежности и отказоустойчивости систем);

– ограничения законодательства различных стран (навязывание устаревших алгоритмов, низкой битности шифрования, интеграции сторонних обфусцированных модулей при сертификации).

Большинство проблем обусловлены погоней корпораций за рентабельностью, которая часто включает плагиат решений с открытым исходным кодом в нарушение открытого лицензионного соглашения (англ. GNU General Public License) – полученные результаты продаются, а не передаются в общественную собственность.

Описанные ключевые недостатки имеют общий существенный признак – жесткую логику функционирования, не адаптированную под нештатные незадекларированные внутренние и внешние возмущения. Важно отметить, что с технической точки зрения требуется учитывать проблемы «нулевого дня», приводящие к нарушениям штатной работы объектов, но ранее не опубликованные в общем доступе [13, 14]. Сбой одного компонента системы может повлечь за собой крах всей критической инфраструктуры.

Цель работы

Целью данной работы являлось проектирование, программная реализация и исследование системы интеллектуально-адаптивного управления информационной инфраструктурой предприятия (далее – Система или СИАУ ИИП), функционирующей на основе ранее представленного одноименного метода [15].

Научная новизна работы заключается в создании комплексного универсального решения по управлению информационной инфраструктурой предприятия, повышающего эффективность, надежность, безопасность и отказоустойчивость функционирования технических объектов и систем за счет созданной программной архитектуры с применением платформы автоматизации развертывания и управления приложениями в среде многослойной виртуализации. Данная архитектура позволила снять ряд ограничений, присущих межсетевым хостам: число одновременно поддерживаемых VPN подключений, задаваемых политик, правил. Повышение уровня конфиденциальности сетевого взаимодействия, как и противодействия анонимным несанкционированным возмущениям, достигается полноценной интеграцией с оверлейными технологиями и сетями, поддержкой широкого спектра современных криптоустойчивых алгоритмов шифрования с возможностью многослойной инкапсуляции. Прогнозирование реакции систем и сервисов на различные внешние воздействия обеспечивается виртуализацией и многослойной изоляцией модельных объектов. Это, в свою очередь, позволяет осуществлять изоляцию рабочих сервисов с возможностью автоматического перезапуска любого компонента без нарушения штатного режима работы инфраструктуры.

1. Программная реализация предлагаемого решения

В рамках программной инженерии архитектура авторской системы интеллектуально-адаптивного управления информационной инфраструктурой предприятия была выполнена с использованием платформы автоматизации развертывания

и управления приложениями в среде виртуализации, что обеспечивает дополнительную надежность и отказоустойчивость решения с изоляцией его компонентов (рис. 1 и 2).

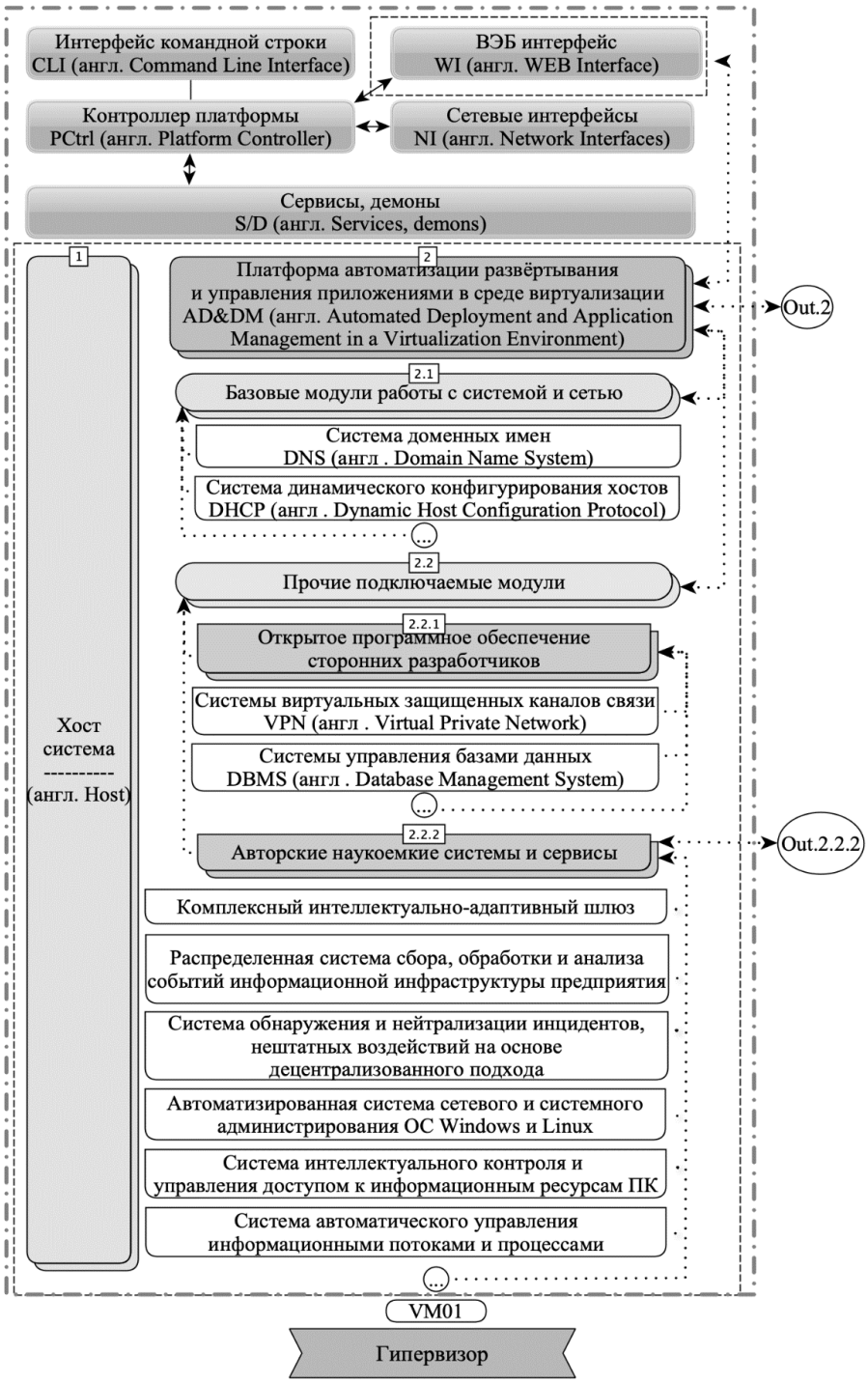

Рис. 1. Архитектура СИАУ ИИП (часть 1)

Контроллер платформы (англ. Platform controller) – это средство достижения инкапсуляции, изоляции и управления информационными потоками между модулями системы. Взаимодействие системы с пользовательскими ЭВМ, сетевым оборудованием, серверными решениями на базе гипервизора и физических серверов проиллюстрировано на рис. 2.

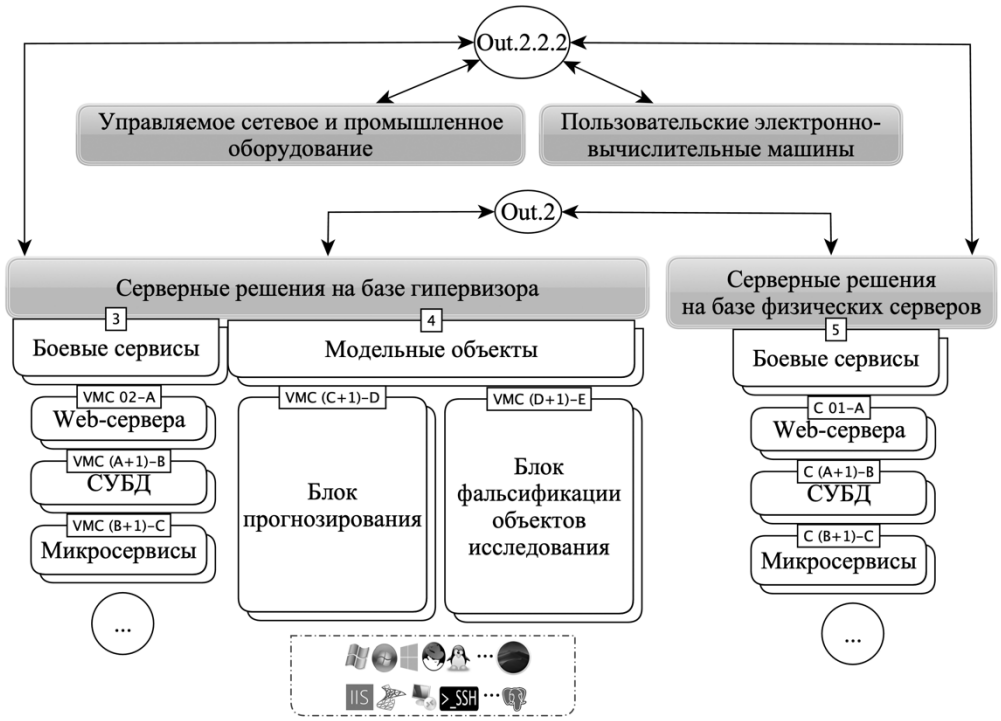

Рис. 2. Архитектура СИАУ ИИП (часть 2)

Рассмотрим реализацию отдельных компонентов системы с аргументацией выбора стека технологий:

1) веб-приложения Системы были написаны на языке Python с использованием веб-фреймворка Django, что является унифицированным решением для типовых задач информационных систем с учетом масштаба Системы, низкой нагрузки на веб-часть, использования стандартных элементов, низкого порога вхождения в долгосрочную поддержку и сопровождения СИАУ ИИП;

2) автоматическая установка и конфигурирование серверных решений производились на основе написанных оригинальных скриптов системы Ansible. Такой выбор решения задачи автоматического развертывания обусловлен ее гибкостью и управляемостью, а также возможностью реализации удаленного развертывания и настройки программного обеспечения без установки клиентского компонента программы на конечный хост, что является ключевым преимуществом данной системы перед аналогами, в том числе – Puppet и Chef. Системой Ansible производится проверка управляемого узла на соответствие описанному в сценарии состоянию, при несоответствии осуществляется выполнение задач согласно их порядку. Также реализован механизм добавки в сценарий обработчиков событий, для задания которых используется параметр notify. Ключевой особенностью Ansible является формирование информативного отчета о результатах выполнения сценариев, содержащего причины возникших ошибок. Помимо этого, данная система предусматривает большое количество дополнительных модулей, позволяющих решить

множество задач управления и развертывания, разделяемых на следующие группы: работа с облачными сервисами, управление базами данных, управление информационной инфраструктурой, работа с системными ресурсами, работа с файлами, в том числе выполнение их шаблонизации, реализация оповещений о процессе применения плейбуков и т. д.;

3) система изолированной сборки и развертывания приложений была реализована на платформе Docker, выполняющей размещение программного обеспечения в специализированных контейнерах с целью облегчения развертывания, отладки и переносимости программной реализации компонентов системы. Docker имеет клиент-серверную архитектуру, согласно которой демон Docker запускается на целевом узле и предназначается для обеспечения изоляции контейнеров, а клиентская часть служит пользовательским интерфейсом для обращения к демону;

4) дополнительный слой виртуализации организован на базе виртуальных машин гипервизора ESXI, где развертываются модельные объекты блоков прогнозирования и фальсификации объектов исследования, а также рабочие корпоративные серверные решения;

5) автоматизация развертывания, масштабирования и управления контейнеризированными высоконагруженными рабочими корпоративными приложениями осуществляется через Kubernetes;

6) управление репозиториями кода и разработкой Системы выполнялось с использованием систем GitLab (показывающей корректную работу в связке с Docker) и Redmine;

7) модуль высокопроизводительного асинхронного обмена сообщениями был написан с использованием библиотеки ZeroMQ;

8) обработка очередей сообщений реализовывалась на базе RabbitMQ;

9) распределенная асинхронная очередь заданий включала Celery;

10) в качестве операционной системы для реализации решения по соотношению надежности, безопасности, отказоустойчивости, актуальности версий ПО, а также вследствие наименьшего потребления ресурсов процессора и памяти была выбрана LinuxCentOS. Помимо этого, поддерживается совместимость с AlpineLinux;

11) сборка дистрибутива алгоритма производится конвейером gitlab из различных компонентов, тогда как с целью достижения изоляции, а также обеспечения стабильности и воспроизводимости компонентов выполнено их размещение в контейнеры Docker.

Основными компонентами программной реализации системы, взаимодействие которых обеспечено с помощью zeromq message broker, являются:

1) графический интерфейс UI, реализованный на Python с применением веб-фреймворка Django, СУБД PostgreSQL, очереди заданий Celery и хранилища данных Redis, предназначенный для выполнения настройки системы. Взаимодействие с прочими компонентами осуществляется с помощью API с использованием синхронного веб-сервера. Для реализации веб-интерфейса использованы фреймворки bootstrap и jQuery;

2) интерфейс командной строки CLI, представляющий собой программный комплекс, написанный на языке программирования Python и необходимый для настройки алгоритма из командной строки. Реализована поддержка шаблонов настройки на языке yaml. Для обеспечения вывода данных аналитики был использован rddtool, а для реализации графического интерфейса настройки применена библиотека curses. В целях автоматизации развертывания и настройки алгоритма

реализован неинтерактивный режим работы утилит управления и настройки из шаблонов;

3) модуль удаленного доступа remote, реализованный на Python и представляющий собой систему самоорганизующегося виртуального защищенного канала связи на основе стохастического многослойного шифрования и оверлейных технологий. Модуль состоит из клиентской части, интегрированной с клиентскими компонентами протоколов SSH и RDP, и серверной части, ответственной в том числе за управление модулем netfilter ядра ОС;

4) модуль управления конфигурацией операционной системы configurator, реализованный на Python и предназначенный для управления конфигурацией запущенных в docker-swarm контейнеров, а также конфигурацией ОС с помощью ролей Ansible. Передача команд конфигурации модулю configurator от UI и CLI выполняется через информационную шину, для хранения параметров конфигурации использован UI, для их настройки – UI и CLI;

5) модуль сбора, анализа и управления сетевым трафиком network, реализованный на C++ с использованием библиотеки libpcap. Для непосредственно управления трафиком использован модуль ядра, netfilter и настройки таблицы маршрутизации. Для подготовки данных для анализа задействован компонент, реализованный на C++, анализ выполняется частично на C++ и Python, для интеграции кода на данных языках использован Python;

6) модуль сбора, систематизации и анализа журналов ОС и установленных компонентов log, реализованный на C++. Хранение собранных из сообщений системных журналов и компонентов ОС данных реализовано в СУБД. В случае, когда работа выполняется в тестовом окружении, производится отправка данных в стек ELK;

7) модуль сбора мониторинговой информации monitoring, реализованный на C++ и выполняющий предварительную обработку данных об использовании системных ресурсов и метрики работающих компонентов. Для сбора данных использован демон collectd, настраиваемый модулем configurator. Хранение собранных данных реализовано в СУБД. В случае, когда работа выполняется в тестовом окружении, аналогично производится отправка данных в стек ELK;

8) модуль отправки аналитики и информации о сбоях telemetry, реализованный на языке программирования Python и предназначенный для сбора данных журналов о своей работе, а также данных журналов из контейнеров с прочими модулями. Для выполнения сбора и отправки данных на сервер технической поддержки задействуются средства Python async через протокол HTTPS в случае, если данные настройки указаны в UI;

9) модуль для ядра ОС, интегрированный с network и monitoring – mod, реализованный на C и служащий для сбора информации о внутренних структурах данных ядра с последующей передачей ее модулям network и monitoring. Данный модуль обеспечивает возможность модификации логики работы ядра, непосредственно относящейся к сетевому стеку;

10) библиотеки с общим исходным кодом на c++ и python – commonlibs, содержащие повторяющийся в различных модулях код (вынесены в отдельные модули, подключение которых может производиться как gitsubmodules);

11) сборщик дистрибутива installer. Манифесты для установщика omnibus, на базе которых выполняется сборка установочного пакета с алгоритмом, включающие в себя пошаговое описание развертывания и начальной настройки его модулей;

12) тестовые стенды для unit, интеграционного, системного, нагрузочного тестирования, тестирования безопасности testworkbenches. Манифесты для Vargant для развертывания в виртуальных машинах VirtualBox независимых тестовых окружений для интеграционных, системных, нагрузочных автоматизированных тестов и для автоматизированных тестов безопасности. Для тестирования безопасности выполняется установка фреймворка metasploit и авторской программы Researcher в VirtualBox. Для всех запущенных в окружении компонентов выполняется конфигурирование для отправки данных мониторинга и журналов в стек ELK. Нагрузочное тестирование осуществляется с помощью фреймворка locust;

13) конфигурация для конвейера GitLab – gitlabcimanifests. Для каждого модуля составлены соответствующие файлы gitlab-ci.yml. Каждый модуль последовательно проходит следующие этапы: сборка docker-образа на основе Centos/Alpine, добавление в образ исходного кода, установка требуемых apk пакетов и python библиотек, установка в образ фреймворка pytest. Запуск unit тестов для кода модуля. При сборке образа используется генератор манифестов для docker;

14) кроссплатформенное мобильное приложение для управления системой – написано с использованием фреймворка Xamarin.

Была осуществлена интеграция с авторскими наукоемкими решениями (от автоматизированной системы сетевого и системного администрирования операционных систем семейства Windows и Linux до системы интеллектуального контроля и управления доступом к информационным ресурсам персонального компьютера).

Все компоненты были проименованы в соответствии с выполняемым функционалом, написаны на языках программирования Python, C++, C, C#, серверные модули контейнеризованы, а в некоторых случаях и расположены на отдельных виртуальных машинах гипервизора.

Их сетевое взаимодействие осуществляется с использованием API внутри защищенных виртуальных каналов связи. Основные операционные системы серверной части (в том числе управляемого сетевого оборудования) – Centos и AlpineLinux, клиентской части – пользовательские версии семейства Linux, Windows и MacOS.

2. Тестирование предлагаемого решения

Для проверки работоспособности было выполнено ручное и автоматизированное тестирование программной реализации авторской системы интеллектуально-адаптивного управления информационной инфраструктурой предприятия (пример представлен на рис. 3).

Было произведено тестирование всех разработанных ролей Ansible. В ходе данного процесса были выполнены следующие действия:

1) для каждой роли, предназначенной для настройки сетевого компонента, был разработан юнит тест (англ. unittesting – модульное тестирование), написанный с использованием фреймворка Testinfra;

2) аспекты компонента, не покрываемые Testinfra, были покрыты с помощью docker-py и других модулей языка Python, подходящих для тестирования инфраструктуры;

3) непосредственно перед тестом была выполнена сборка образа Docker, запущенного в виртуальной машине Virtualbox, поднимаемой Vagrant. Внутри вирту-

альной машины из образа Docker был запущен контейнер с докеризованным компонентом системы;

```
============================= test session starts ==============================
platform linux -- Python 3.6.8, pytest-5.0.0, py-1.8.0, pluggy-0.12.0
rootdir: /home/         /workspace/ark_gateway/netstack_controller
collected 1115 items

tests/firewall/test_chains.py ..........................................................  [  7%]
tests/firewall/test_controller.py ......................................................  [ 13%]
tests/firewall/test_counters.py ........................................................  [ 22%]
tests/firewall/test_mock.py ............................................................  [ 30%]
tests/firewall/test_policies.py ........................................................  [ 35%]
tests/firewall/test_rollback.py ........................................................  [ 42%]
tests/firewall/test_rules.py ...........................................................  [ 48%]
tests/firewall/signatures/test_binmask.py ..............................................  [ 55%]
tests/firewall/signatures/test_heuristic_basic.py ......................................  [ 62%]
tests/firewall/signatures/test_neural_chunked.py .......................................  [ 69%]
tests/firewall/signatures/test_neural_pt_base.py .......................................  [ 74%]
tests/firewall/signatures/test_regex.py ................................................  [ 81%]
tests/routing/test_ctl.py ..............................................................  [ 88%]
tests/routing/test_hdr_parser_wrapper.py ...............................................  [ 95%]
tests/routing/test_rp_wrapper.py .......................................................  [100%]

========================== 1115 passed in 186.34 seconds ===========================
```

Рис. 3. Результат примера выполнения части юнит-тестов Системы с использованием фреймворка pytest

4) далее был выполнен запуск инфраструктурных тестов, которые были пройдены также продуктом целиком. Запуск тестов автоматизирован: для юнит-тестов он выполнялся при каждом внесении изменений в репозиторий Gitlab, для интеграционных тестов – при сборке релиза.

В целях тестирования веб-приложений информационной системы использовался инструмент Selenium для автоматизации действий веб-браузера.

3. Исследование предлагаемого решения

Важно отметить, что не существует полноценного аналога предлагаемого решения, на рынке представлены лишь аппаратно-программные объекты, выполняющие атомарный функционал представленного комплекса. Для исследования эффективности работы СИАУ ИИП был произведен экспериментальный сравнительный анализ с различными существующими решениями: комплексными межсетевыми экранами и маршрутизаторами D-LINK DFL-870, ZYXEL USG210, CISCO ASA5550-DC-K8, Juniper SRX345, Advantech FWA-660, FP-Stonesoft 1100-c1, HN SG-6000-E1600, FG-51E, Sangfor M4500, M5100, MikroTik CCR1036-8G-2S+EM, Huawei USG2260, Kerio Control, часто именуемыми маркетологами NGFW (англ. Next generation firewall) и UTM (англ. Unified threat management), системами обнаружения и предотвращения вторжений Suricata, Bro с общедоступными базами знаний, различными проприетарными инструментами NGIPS (англ. Next-Generation IPS), а также другими инструментами управления трафиком.

Рассматриваемая СИАУ ИИП продемонстрировала отсутствие ограничений на число одновременно поддерживаемых VPN подключений, за исключением лимита вычислительных мощностей сервера, подтвердила отсутствие ограничений на число задаваемых политик и правил. Важно отметить, что практически все альтернативные решения имеют ограничения на количество VPN-туннелей и политик

фильтрации порядка 200 и 2000 соответственно. В авторской Системе эти ограничения снимаются посредством спроектированной архитектуры, выдерживающей высокие нагрузки, а также применением технологии нулевой маршрутизации. Существенными недостатками ряда конкурентов выступали: некриптоустойчивое шифрование (по типу устаревшего алгоритма DES), отсутствие интеграции с оверлейными технологиями и сетями, составление сигнатур и черных списков на стороне серверов правообладателя с последующей синхронизацией перед фильтрацией трафика и многое другое. Это не выдерживает сравнения с авторским гибким многослойным инкапсулированным шифрованием на базе криптоустойчивых алгоритмов, полноценной интеграцией с оверлейными технологиями и автоматическим составлением черных списков с самообучением системы.

В рамках проведения следующего эксперимента были задействованы инструменты пассивного и активного анализа информационных систем: сканеры, зондеры и инструменты пентеста. Настройки СИАУ ИИП проиллюстрированы на рис. 4.

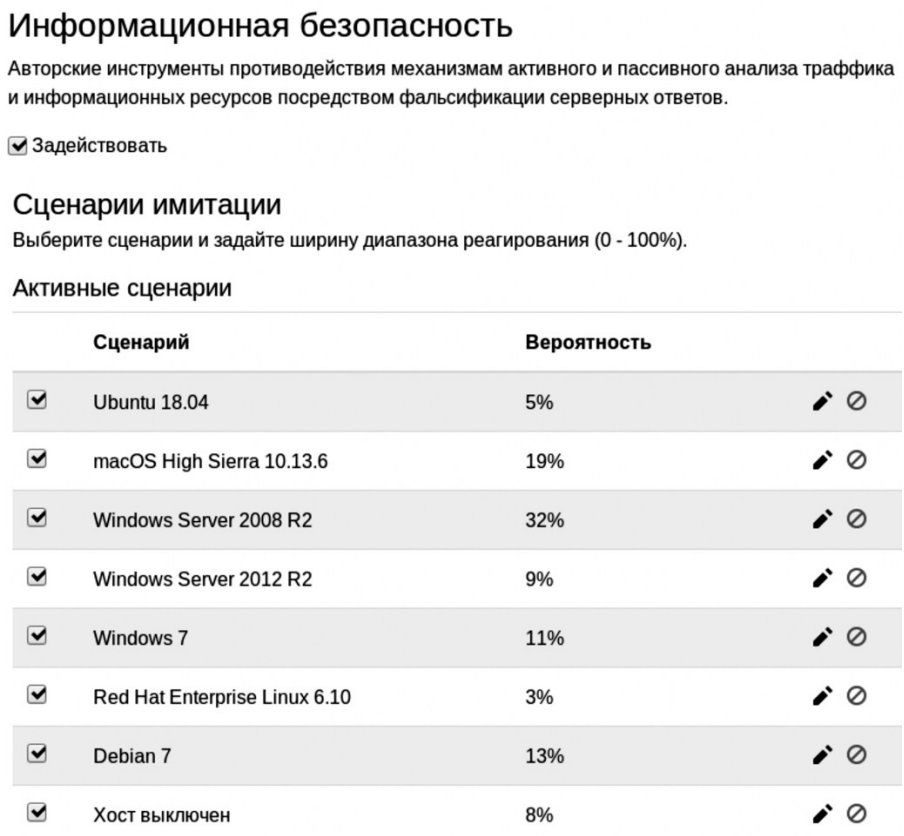

Рис. 4. Настройка противодействия СИАУ ИИП несанкционированным исследованиям

Использовались следующие программные продукты: Nmap, Nessus, Rapid 7 NeXpose, OpenVAS, X-Scan, XSpider 7, Microsoft BSA, GFI LANguard, RetinaNSA, SAINT и другие средства. Жесткая логика поведения существующих решений поз-

волила идентифицировать их вплоть до версии прошивки и актуальных уязвимостей. Например, обхода аутентификации (CVE-2019-1912). Соответственно, не составляло труда осуществить нарушение штатного режима работы исследуемого узла. СИАУ ИИП была настроена на противодействие подобным нештатным возмущениям.

Для множественных итераций исследования со стороны одного хоста осуществлялась эмуляция однотипного сценария фальсификации – Windows Server 2008 R2. В целях усложнения эксперимента сканирование и зондирование осуществлялось одним источником с использованием смены личностей (выходных IP-адресов) оверлейной сети TOR (англ. The Onion Router).

При этом СИАУ ИИП однозначно идентифицировала воздействия посредством корреляционного анализа пула адресных пространств, типов исследований, процентов их исчерпания, временных задержек и другой метаинформации. Пример протокола результатов анализа разработанной системы представлен на рис. 5.

CVSS	Plugin	Name
10.0	97833	MS17-010: Security Update for Microsoft Windows SMB Server (4013389) (E (ETERNALROMANCE) (ETERNALSYNERGY) (WannaCry) (EternalRocks) (Petya)
10.0	108797	Unsupported Windows OS
6.8	90510	MS16-047: Security Update for SAM and LSAD Remote Protocols (3148527)
5.0	57608	SMB Signing not required
N/A	10114	ICMP Timestamp Request Remote Date Disclosure
N/A	10150	Windows NetBIOS / SMB Remote Host Information Disclosure
N/A	10287	Traceroute Information
N/A	10394	Microsoft Windows SMB Log In Possible
N/A	10736	DCE Services Enumeration
N/A	10785	Microsoft Windows SMB NativeLanManager Remote System Information Disclosure
N/A	11011	Microsoft Windows SMB Service Detection
N/A	11219	Nessus SYN scanner
N/A	11936	OS Identification
N/A	12053	Host Fully Qualified Domain Name (FQDN) Resolution

Рис. 5. Пример протокола результатов анализа разработанной системы

Таким образом, источник несанкционированных исследований Системы идентифицировал операционную систему Windows Server 2008 R2 с двумя критическими уязвимостями, подозрений о дезинформации у него и не могло возникнуть. При последующих враждебных действиях он был бы перенаправлен на соответствующий изолированный объект блока фальсификации с отслеживанием действий, автоматическим изучением потенциально новой злоумышленной активности и выработкой стратегии по ее упреждению.

Очередным интересным экспериментом выступила генерация более 5000 распределенных сетевых атак различных типов, совмещенных с полезными тестовыми сигналами, на каждый рассматриваемый продукт в отдельности. Пропускная способность канала связи составляла 1 Гбит/с. При этом применялись оверлейные технологии, сервисы анонимизации (прокси и VPN) и другие инструменты для усложнения анализа и обработки возмущений системами защиты. Таким образом, задействовалось более 20 000 «белых» IP-адресов глобальной сети Интернет. Диаграмма пропускной способности канала связи в интервале проведения атак с 10-й по 50-ю секунды представлена на рис. 6. В рамках соблюдения принципов и норм научной этики существующие альтернативные решения в области управления трафиком, с которыми производилось сравнение, проименованы на диаграмме латинскими буквами: A, B, C, D, E, F, G (было выбрано 7 решений в ценовом диапазоне от 40 000 до 600 000 рублей с учетом аппаратной части).

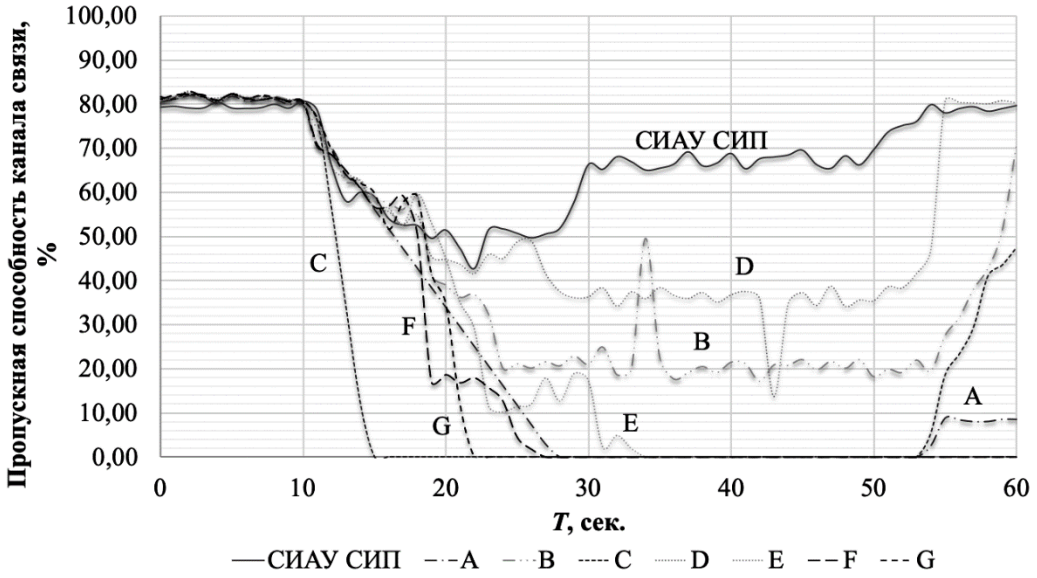

Рис. 6. Сводная диаграмма загрузки канала связи

Разработанная система обеспечила бесперебойную обработку полезных тестовых сигналов в штатном режиме, оптимизировав загрузку каналов связи посредством выработки оптимальных стратегий реагирования на несанкционированные внешние возмущения различных типов и уровней рисков. Даже в штатном режиме работы СИАУ ИИП имитирует работу различных сервисов и служб, генерируя фальшивый трафик в качестве ловушек. По умолчанию процентное отношение такого рода информационных потоков к общему объему трафика не может превышать 2 % (значение в данном интервале выбирается стохастически). Данный

параметр корректируется через панель администрирования. Таким образом, избыточность объема трафика в нормальном режиме работы выступает одним из компонентов стоимости эффективности работы Системы при обработке непредвиденных ситуаций.

Обсуждение результатов и заключение

В ходе работы была спроектирована, программно реализована и исследована система интеллектуально-адаптивного управления информационной инфраструктурой предприятия [16]. Функционирование данной системы осуществляется на основе ранее представленного одноименного метода. СИАУ ИИП обеспечивает надежную, отказоустойчивую и качественную работу ее технических объектов и систем в едином стеке с оптимизацией загрузки каналов связи при различных типах воздействий и инцидентах, самообучается и предоставляет интеллектуальную поддержку при принятии управленческих решений в технических системах в условиях неопределенности.

В рамках ноу-хау программной инженерии ее архитектура была выполнена с использованием платформы автоматизации развертывания и управления приложениями в многослойных инкапсулированных средах виртуализации и изоляции с аналогичным подходом к шифрованию объектов, что обеспечивает дополнительную надежность и отказоустойчивость решений с моментальным восстановлением работоспособности любого компонента даже в случае технического сбоя без последствий для информационной инфраструктуры.

В целях проверки работоспособности было выполнено ручное и автоматизированное тестирование всего программного комплекса. Для исследования эффективности работы Системы был произведен экспериментальный сравнительный анализ с существующими решениями в области управления трафиком и обеспечения сетевой информационной безопасности, которые фактически выполняют лишь атомарный функционал системы (в связи с отсутствием полноценных аналогов).

В сравнении с альтернативными решениями загрузка канала связи в штатном режиме повышается на 0,0001–2 % (издержки противодействия несанкционированным идентификациям и исследованиям технических систем и объектов, параметр настраивается опционально в панели администрирования с выбором сценариев фальсификации доступных шаблонов), при массовых нештатных возмущениях среднего уровня риска снижается в среднем на 27,4 %, а при нештатных воздействиях высокого уровня риска не превышает 70 %, в то время как множество конкурентов переходят в состояние недоступности и не обрабатывают стандартные запросы. Таким образом, повышается пропускная способность для штатных информационных потоков.

Дополнительным преимуществом перед существующими решениями является объединение всей информационной инфраструктуры в единый стек, единую экосистему. Снимаются ограничения на количество задаваемых политик, размеры адресных пулов правил и многое другое.

Цена данных преимуществ – повышенные требования к вычислительным мощностям: DDR4 ECC от 16 ГБ вместо DDR3 2ГБ, Intel Core от i5 вместо i3, SSD 1 Тб вместо HDD 100 ГБ. Другим недостатком является избыточность информации: децентрализованный реестр событий дублируется на всех хостах.

Данные аспекты являются незначительными в сравнении с рентабельностью внедрения Системы, которая выражается в повышения эффективности, надежности, отказоустойчивости и безопасности функционирования технических объектов даже в нештатных ситуациях.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. *Yanyan L., Keyu J.* Prospect for the Future Internet: A Study Based on TCP/IP Vulnerabilities. Proceedings of the International Conference on Computing, Measurement, Control and Sensor Network, Taiyuan, China. 2012. P. 52–55.
2. *Басыня Е.А*. Проблематика управления трафиком вычислительной сети с коммутацией пакетов на основе стека протоколов TCP/IP // Техника и технология: новые перспективы развития. 2014. № 15. С. 48–57.
3. *Christoph B., Horst S., Olaf S*. Soft-Error Detection and Correction for Concurrent Data Structures // Transactions on Dependable and Secure Computing. 2017. Vol. 14. No. 1. P. 22–36.
4. *Перевощиков В.А*. Обзор уязвимостей программного обеспечения комплексных электронных систем безопасности // Актуальные научные исследования в современном мире. 2017. № 3–2 (23). С. 76–77.
5. *Motta R.C., Marcal K.O., Travassos G.H.* Rethinking Interoperability in Contemporary Software Systems. Proceedings of the 5th International Workshop on Software Engineering for Systems-of-Systems and 11th Workshop on Distributed Software Development, Software Ecosystems and Systems-of-Systems (JSOS), Buenos Aires, Argentina. 2017. P. 9–15.
6. *James D.A., James B.D.* Applying standard independent verification and validation (IV&V) techniques within an Agile framework: Is there a compatibility issue? Proceedings of the IEEE International Systems Conference (SysCon), Montreal, QC, Canada. 2017. P. 1–5.
7. *Монарев В.А., Пестунов А.И.* Эффективное обнаружение стеганографически скрытой информации посредством интегрального классификатора на основе сжатия данных // Прикладная дискретная математика. 2018. № 40. С. 59–71.
8. *Монарев В.А., Пестунов А.И.* Повышение эффективности методов стегоанализа при помощи предварительной фильтрации контейнеров // Прикладная дискретная математика. 2016. № 2. С. 87–99.
9. *Пестунов А.И.* О формализации и систематизации основных понятий дифференциального криптоанализа итеративных блочных шифров // Проблемы информационной безопасности. Компьютерные системы. 2014. № 3. С. 109–114.
10. *Пестунов А.И., Ковалев В.А.* Определение критической цены продажи технологической машины на рынке оборудования с позиции покупателя // Вестник НГУЭУ. 2019. № 1. С. 240–246.
11. *Мутигуллин А.С., Прасолова Е.А.* Обзор методологий разработки корпоративных информационных систем // Научное обозрение. Технические науки. 2018. № 6. С. 41–45.
12. *Тайлаков В.А., Израилов К.Е*. Способы автоматизации поиска уязвимостей в программном обеспечении на соответствующих уровнях его разработки // Научный аспект. 2018. Т. 6. № 4. С. 719–726.
13. *Gorbacheva A., Smirnov S*. Converging technologies and a modern man: emergence of a new type of thinking. AI & Society. 2017. Т. 32. № 3. С. 465–473.
14. *Крохин Г.Д., Аракелян Э.К., Мухин В.С., Пестунов А.И.* Применение методологии искусственного интеллекта для формализации результатов обработки нечеткой информации // Вестник Московского энергетического института. 2017. № 5. С. 130–138.
15. *Басыня Е.А., Сафронов А.В.* Децентрализованный подход к сбору и обработке данных информационной инфраструктуры предприятия // Вестник УрФО. Безопасность в информационной сфере. 2019. № 3 (33). С. 43–54.
16. *Басыня Е.А*. Система интеллектуально-адаптивного управления сетевой инфраструктурой предприятия // Свидетельство о государственной регистрации программы для ЭВМ № 2019660561 РФ, опубл. 07.08.2019. РОСПАТЕНТ.

Статья поступила в редакцию 27 января 2020 года

SOFTWARE IMPLEMENTATION AND RESEARCH OF THE SYSTEM FOR INTELLECTUALLY ADAPTIVE MANAGEMENT OF THE ENTERPRISE INFORMATION INFRASTRUCTURE

E.A. Basinya

Novosibirsk State Technical University, Research Institute of Information and Communication Technologies
20, Prospekt K. Marksa, Novosibirsk, 630073, Russian Federation

***Abstract**. The paper describes the problems of information and communication interaction of technical objects and systems. The vulnerabilities of the TCP / IP protocol stack, the imperfection of operating systems and application software, the complexity of full mutual integration of various systems are analyzed. The consequence of saving vendors of network solutions on computing power in the form of architectural constraints and vulnerabilities is described. The design, software implementation and experimental study of the author's system of intellectually adaptive management of the enterprise's information infrastructure, which operates on the basis of the previously presented method of the same name, is considered. A modern technology stack is used in combination with a flexible development methodology. The scientific novelty of the work lies in the proposed architecture of a comprehensive software product for managing the enterprise information infrastructure. Improving the efficiency, reliability, security and fault tolerance of the operation of technical objects and systems is achieved by using the automation platform for deploying and managing applications in a multi-layer virtualization environment. Isolation of work services is carried out with the ability to automatically restart any component without violating the normal operating mode of the infrastructure. A number of restrictions inherent in existing solutions, such as the number of simultaneously supported VPN connections, as well as defined policies and rules are removed. Increasing the level of confidentiality of network interaction, as well as countering anonymous unauthorized disturbances, is achieved by full integration with overlay technologies and networks, and also by supporting a wide range of modern cryptographic encryption algorithms with the possibility of multi-layer encapsulation. In order to evaluate the effectiveness of the developed system, an experimental comparative analysis is presented with various existing solutions: comprehensive firewalls, routers, intrusion detection and prevention systems, as well as other traffic management tools.*

***Keywords:** intellectually adaptive management, system analysis, processing, network traffic, local information processes, TCP / IP, IDS / IPS, SIEM, UTM, NGFW.*

REFERENCES

1. *Yanyan L., Keyu J.* Prospect for the Future Internet: A Study Based on TCP/IP Vulnerabilities // *Proceedings of the International Conference on Computing, Measurement, Control and Sensor Network*, Taiyuan, China, 2012, pp. 52–55.
2. *Basinya E.A.* Packet-switched computing network traffic management issues based on the TCP/IP protocol stak. *Texnika i texnologiya: novy`e perspektivy` razvitiya*, 2014, no. 15, pp. 48–57 (in Russian).
3. *Christoph B., Horst S., Olaf S.* Soft-Error Detection and Correction for Concurrent Data Structures. *Transactions on Dependable and Secure Computing*. 2017, vol. 14, no. 1, pp. 22–36.
4. *Perevoshhikov V.A.* Integrated Electronic Security Software Vulnerability Overview. *Aktual`ny`e nauchny`e issledovaniya v sovremennom mire*, 2017, no. 3–2 (23), pp. 76–77 (in Russian).
5. *Motta R.C., Marcal K.O., Travassos G.H.* Rethinking Interoperability in Contemporary Software Systems. *Proceedings of the 5th International Workshop on Software Engineering for Systems-of-Systems and 11th Workshop on Distributed Software Development, Software Ecosystems and Systems-of-Systems (JSOS)*, Buenos Aires, Argentina, 2017, pp. 9–15.

Evgeny A. Basinya (Ph.D. (Techn.)), Associate Professor.

6. *James D.A., James B.D.* Applying standard independent verification and validation (IV&V) techniques within an Agile framework: Is there a compatibility issue? *Proceedings of the IEEE International Systems Conference (SysCon)*, Montreal, QC, Canada. 2017, pp. 1–5.
7. *Monarev V.A., Pestunov A.I.* Effective detection of steganographically hidden information by means of an integrated classifier based on data compression. *Prikladnaya diskretnaya matematika*. 2018, no. 40, pp. 59–71 (in Russian).
8. *Monarev V.A., Pestunov A.I.* Improving the effectiveness of steganalysis methods by pre-filtering containers. *Prikladnaya diskretnaya matematika*. 2016, no. 2, pp. 87–99 (in Russian).
9. *Pestunov A.I.* On the formalization and systematization of the basic concepts of differential cryptanalysis of iterative block ciphers. *Problemy` informacionnoj bezopasnosti. Komp`yuterny`e sistemy`*. 2014, no. 3, pp. 109–114 (in Russian).
10. *Pestunov A.I., Kovalev V.A.* Determination of the critical selling price of a technological machine in the equipment market from a buyer's perspective. *Vestnik NGUE`U*. 2019, no. 1, pp. 240–246 (in Russian).
11. *Mutigullin A.S., Prasolova E.A.* Overview of corporate information systems development methodologies. *Nauchnoe obozrenie. Texnicheskie nauki*. 2018, no. 6, pp. 41–45 (in Russian).
12. *Tajlakov V.A., Izrailov K.E.* Ways to automate the search for vulnerabilities in software at appropriate levels of its development. *Nauchny`j aspect*, 2018, vol. 6, no. 4, pp. 719–726 (in Russian).
13. *Gorbacheva A., Smirnov S*. Converging technologies and a modern man: emergence of a new type of thinking. *AI & Society*. 2017, vol. 32, no. 3, pp. 465–473.
14. *Kroxin G.D., Arakelyan E`.K., Muxin V.S., Pestunov A.I.* The use of artificial intelligence methodology for the formation of fuzzy information processing results. *Vestnik Moskovskogo e`nergeticheskogo instituta. Vestnik ME`I*. 2017, no. 5, pp. 130–138 (in Russian).
15. *Basinya E.A., Safronov A.V*. Decentralized approach for collecting and processing data of the enterprise information infrastructure. *Vestnik UrFO. Bezopasnost` v informacionnoj sfere*. 2019, No. 3 (33), pp. 43–54 (in Russian).
16. *Basinya E.A.* System for intellectually adaptive management of the enterprise network infrastructure. *State registration of an intellectual property object (computer programs) № 2019660561 RF*, publicati.