МИНОБРНАУКИ РОССИИ
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«САРАТОВСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ ИМЕНИ Н.Г. ЧЕРНЫШЕВСКОГО»

Кафедра теоретических основ компьютерной безопасности и криптографии

Программные средства контроля и отслеживания состояния систем

АВТОРЕФЕРАТ
дипломной работы

студента 6 курса 631 группы
специальности 10.05.01 Компьютерная безопасность
факультета компьютерных наук и информационных технологий
Титова Антона Сергеевича

Научный руководитель
ассистент _____ А. А. Лобов
21.01.2023 г.

Заведующий кафедрой
д. ф.-м. н., доцент _____ М. Б. Абросимов
21.01.2023 г.

Саратов 2023

ВВЕДЕНИЕ

С развитием технологий у человека появилась возможность получать огромное количество информации. Вопрос сбора, хранения и обработки информации решается путем усовершенствования и увеличения количества вычислительных устройств. Помимо этого, тенденция, вызванная влиянием пандемии, к дистанционному взаимодействию привела к всплеску спроса на облачные сервисы, что в свою очередь привело к необходимости в дополнительном оборудовании и расширению сетей дата-центров.

Другим аспектом является развитие информационных технологий в ключевых отраслях экономики. Например, контроль трафика на дорогах. Необходимо обеспечивать бесперебойное функционирование этих систем. Для решения данного вопроса создаются дополнительные системы контроля, позволяющие отслеживать состояние вычислительной техники и ее программного обеспечения и организованно управлять этой техникой.

Кроме того, некоторые решения также позволяют реагировать на недопустимые события, например, несанкционированный доступ к критически важным объектам.

Целью данной работы является изучение принципов работы систем мониторинга, а также реализация подобной системы.

Таким образом в рамках дипломной работы решаются следующие задачи:

1. Обзор методов сбора метрик;
2. Обзор существующих программных решений;
3. Реализация собственного программного продукта.

КРАТКОЕ СОДЕРЖАНИЕ

В дипломной работе в разделе 1 приводятся необходимые определения и сокращения, которые используются в данной работе.

В разделе 2 рассматриваются принципы работы систем мониторинга, а также требования к этим системам в общем виде.

Мониторинг ИТ-инфраструктуры представляет собой полный цикл решений по мониторингу серверного и коммутационного оборудования, систем хранения данных, системного и прикладного программного обеспечения. Система мониторинга устанавливает взаимосвязь между различными объектами, объединяя их в сервисы, позволяя строить различные топологии. Это дает возможность контролировать сервисы в реальном времени, оперативно оценивать и решать критически важные проблемы, способные повлиять на бизнес.

Мониторинг ИТ-инфраструктуры относится к прикладным задачам. По этой причине не удается установить научно-обоснованные методы и требования к системам мониторинга. Однако при исследовании данного вопроса, были проанализированы различные источники и составлен собственный список требований:

- Принцип достаточности – при построении необходимо использовать минимальное необходимое число функционирующих компонентов;
- Принцип информационной полноты – требуется осуществлять сбор всех необходимых для анализа характеристик;
- Принцип дружественности интерфейса – система должна обеспечивать быстрое и легкое восприятие оператором информации, должна быть проста в настройке;
- Принцип структурной гибкости и программируемости – система должна быть легковесной, масштабируемой, высокопроизводительной.

В разделе 3 «Методы сбора метрик» рассматриваются подходы к сбору метрик. Их можно разделить следующим образом:

1. Метод сбора и анализа проблем с производительностью инфраструктуры (железо, сеть);
2. Метод сбора высокоуровневых данных и анализа (веб-сервисы, базы данных, очереди и так далее);
3. Метод сбора и анализа бизнес-метрик.

В данной работе не рассматриваются методы анализа бизнес-метрик.

Метод USE — это акроним от терминов Utilization, Saturation и Errors (Утилизация, Насыщение и Ошибки). Использовать USE нужно для выбора и анализа низкоуровневых метрик, например утилизацию процессора, количество свободного места, памяти, превышение допустимой температуры оборудования или нагрузка самой сети.

RED — это акроним от терминов Request (Rate), Errors и Duration (Запросы или Скорость, Ошибки и Продолжительность). Он рассчитан на сбор метрик с самих приложений. Использовать RED нужно для более высокоуровневых сервисов, которые обслуживают запросы. Например, различные веб-сервисы, базы данных, очереди и так далее.

LTES – акроним от терминов Latency, Traffic, Errors, Saturation (Задержка, Трафик, Ошибки, Насыщенность). Этот метод рассчитан на отслеживание всей системы целиком. Метрики, собираемые с использованием данного метода, могут указать как на проблемы с приложениями, так и на низкоуровневые проблемы.

В разделе 4 обозреваются существующие системы мониторинга.

Zabbix — свободная система мониторинга статусов разнообразных сервисов компьютерной сети, серверов и сетевого оборудования. Для хранения данных используется MySQL, PostgreSQL, SQLite или Oracle Database, веб-интерфейс написан на PHP.

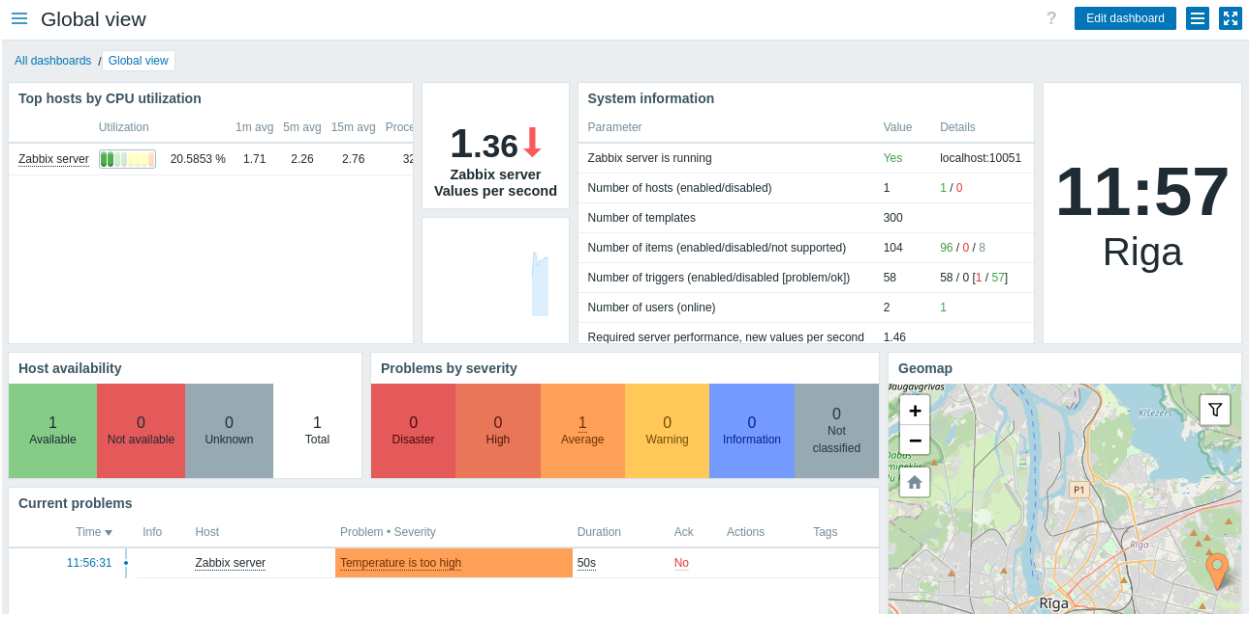

Рисунок 1 – Панель Zabbix в дефолтной раскладке

Prometheus – система мониторинга серверов и программ с открытым исходным кодом, написанная на компилируемом языке Golang и частично на Ruby. Эта система мониторинга следует Pull-модели и может собирать информацию о состоянии серверов и систем, а также получать предупреждения о проблемах. В состав Prometheus входят такие компоненты, как Prometheus Server и Exporters.

Nagios – программа с открытым кодом, предназначенная для мониторинга компьютерных систем и сетей: наблюдения, контроля состояния вычислительных узлов и служб, оповещения администратора в том случае, если какие-то из служб прекращают или возобновляют свою работу.

В разделе 5 «Реализация системы мониторинга» описана разработанная программа, основанная на методе USE, и имеющая сходство с системой Zabbix. Программа состоит из трех значимых компонентов: Агент, Сервер, Веб-интерфейс.

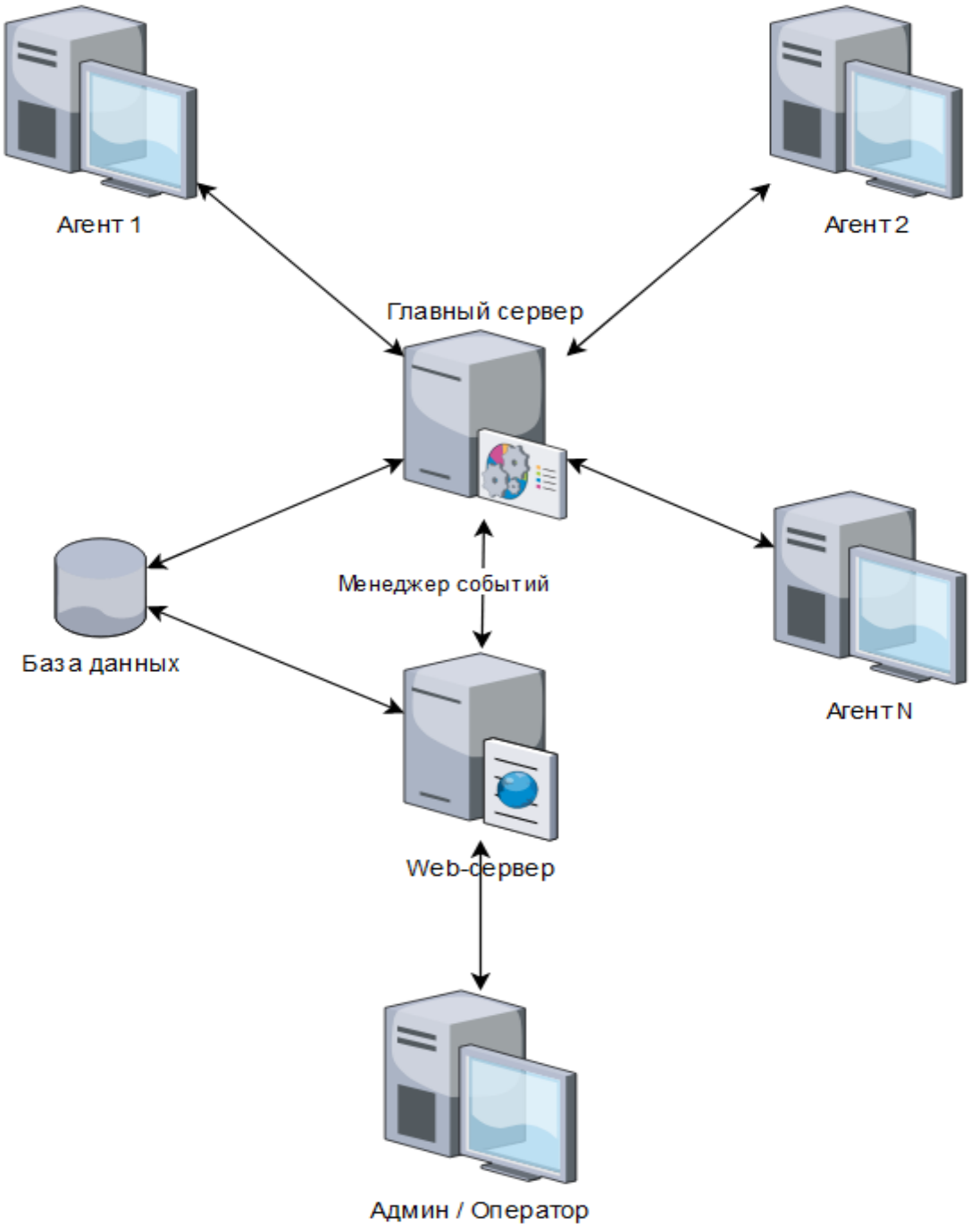

Рисунок 2 – Схема взаимодействия компонентов системы

Программа агента может собирать данные при помощи описанной в данном разделе библиотеки gopsutil, а также получать команду от Главного Сервера на закрытие какого-либо TCP-соединения.

Сохранение данных осуществляется с использованием СУБД PostgreSQL. На рисунке 3 представлена схема нормализованной базы данных.

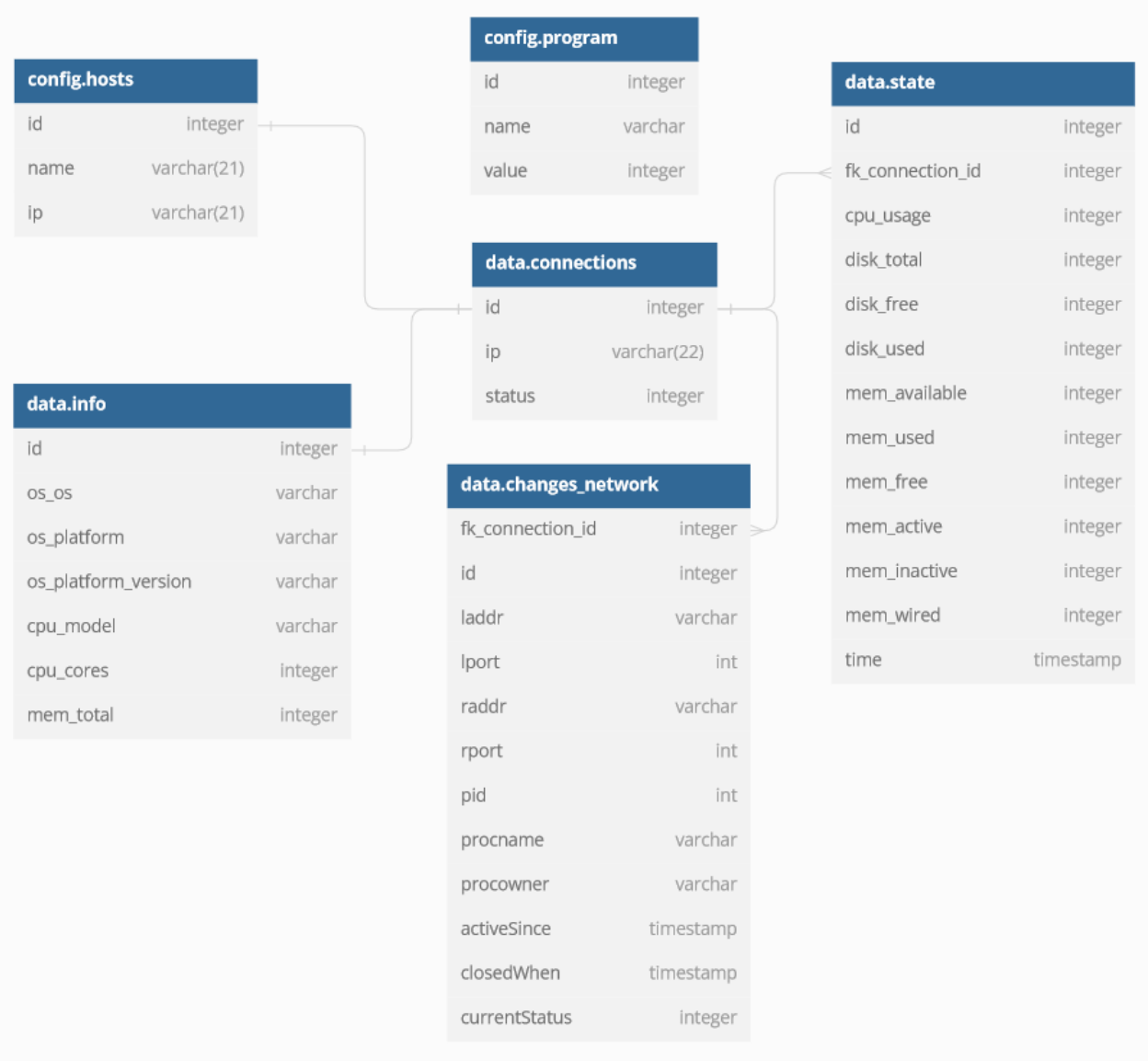

Рисунок 3 – Схема базы данных

Отображение данных осуществляется посредством веб-интерфейса. Предполагается, что оператор системы (администратор) будет наблюдать за состоянием устройств при помощи веб-браузера. Данные пересылаются через протокол websocket в формате JSON. На рисунке 4 представлена главная панель веб-интерфейса, на которой отображен список отслеживаемых устройств, а также список TCP-соединений, выбранного устройства.

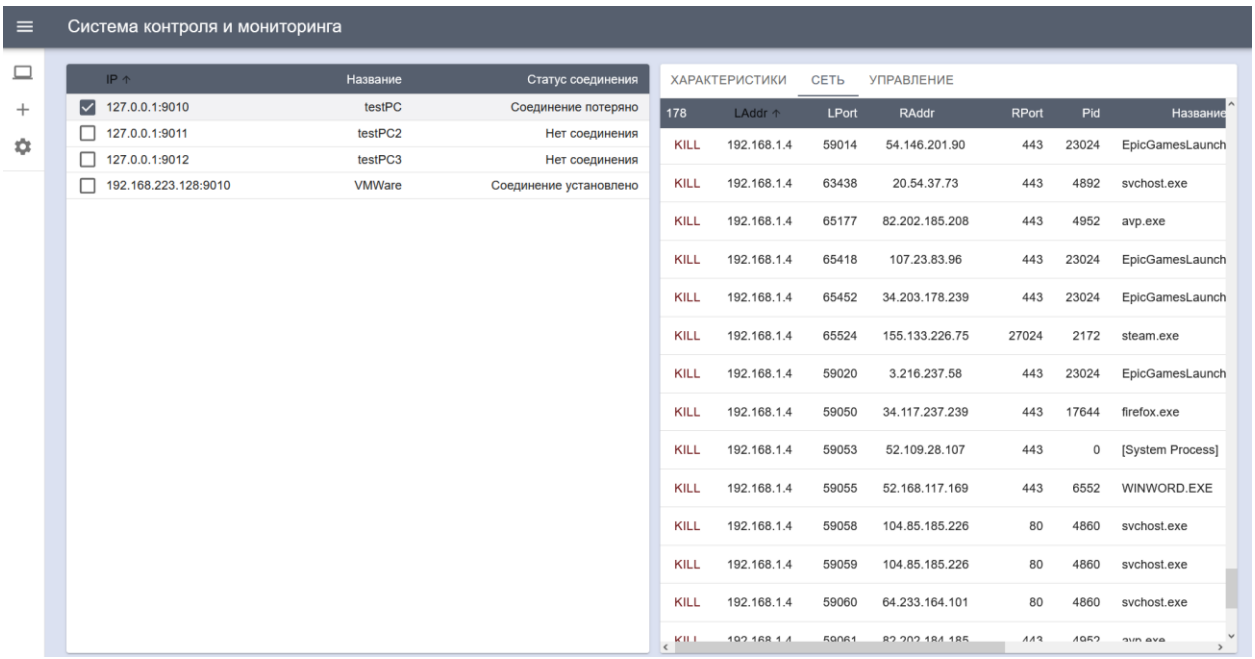

Рисунок 4 – Главная панель системы мониторинга

Программа была протестирована на все возможные ситуации: добавление нового устройства для отслеживания, удаление устройства, изменение в записи устройства (при некорректном добавлении), а также закрытие TCP-соединений по команде. Все функции выполняются корректно.

ЗАКЛЮЧЕНИЕ

В ходе выполнения работы были рассмотрены методы сбора технической информации о работе и состоянии отслеживаемых устройств.

Также в работе обозревались некоторые из представленных на рынке решений по контролю и мониторингу целевых систем. Были рассмотрены программы такие как Zabbix, Prometheus, Nagios, Tripwire. Часть из них были протестированы с целью ознакомления с их интерфейсом и возможностями.

Практической частью работы являлась разработка программного продукта, который реализует исследованные методы сбора метрик, и в целом, является аналогом представленных на рынке решений. Разработанный продукт является прототипом, так как конкурировать с популярными программами в силу их богатой функциональности он не может.

Разработанная система была проверена на работе виртуальных машин. Было установлено, что система работает корректно, выполняя все заложенные в нее функции.