|  |  |
| --- | --- |
| |  | | --- | | МИНОБРНАУКИ РОССИИ  Федеральное государственное бюджетное образовательное учреждение  высшего образования  **«МИРЭА – Российский технологический университет»**  **РТУ МИРЭА** | |

Институт Информационных Технологий

Кафедра корпоративных информационных систем (КИС)

**ОТЧЕТ**

по Практической работе №2

на тему

«Анализ и мониторинг элементов информационной системы»

по дисциплине

«Безопасность функционирования информационных систем»

Выполнил студент группы ИКБО-08-18 Валяев Д.А.

Принял к.т.н., доцент Башлыкова А.А.

Выполнено «\_\_\_» \_\_\_\_\_\_\_\_\_2021 г.

Зачтено «\_\_\_» \_\_\_\_\_\_\_\_\_2021 г.

**Практическая работа № 2.**

**Тема:** «Составление плана мероприятий для организации безопасного функционирования ИС на базе ОС Debian 7.11»

**Цель работы:** Анализ выбранной ОС, анализ его ядра, определение категории требований к безопасности РИС.

**Раздаточный материал:** дидактический материал

**Перечень и характеристики оборудования, приборов и материалов:**

Тип компьютера: PC

Операционная система: Windows 10 Pro

Пакет обновления: 10.0.18362 Сборка 18362

Internet Explorer: 11.476.18362.0

DirectX: DirectX 12

Тип ЦП: 8х, 4.4 GHz AMD Ryzen 7 3700x

Системная память: 16 Гб = 16384 MB

Видеоадаптер: NVIDIA GeForce RTX 2060

Монитор: Samsung Odyssey G3

Дисковый накопитель: Samsung EVO 870 (500 ГБ)

## Исходные данные

1) Zabbix — свободная система мониторинга и отслеживания статусов разнообразных сервисов компьютерной сети, серверов и сетевого оборудования от компании Zabbix LLC.

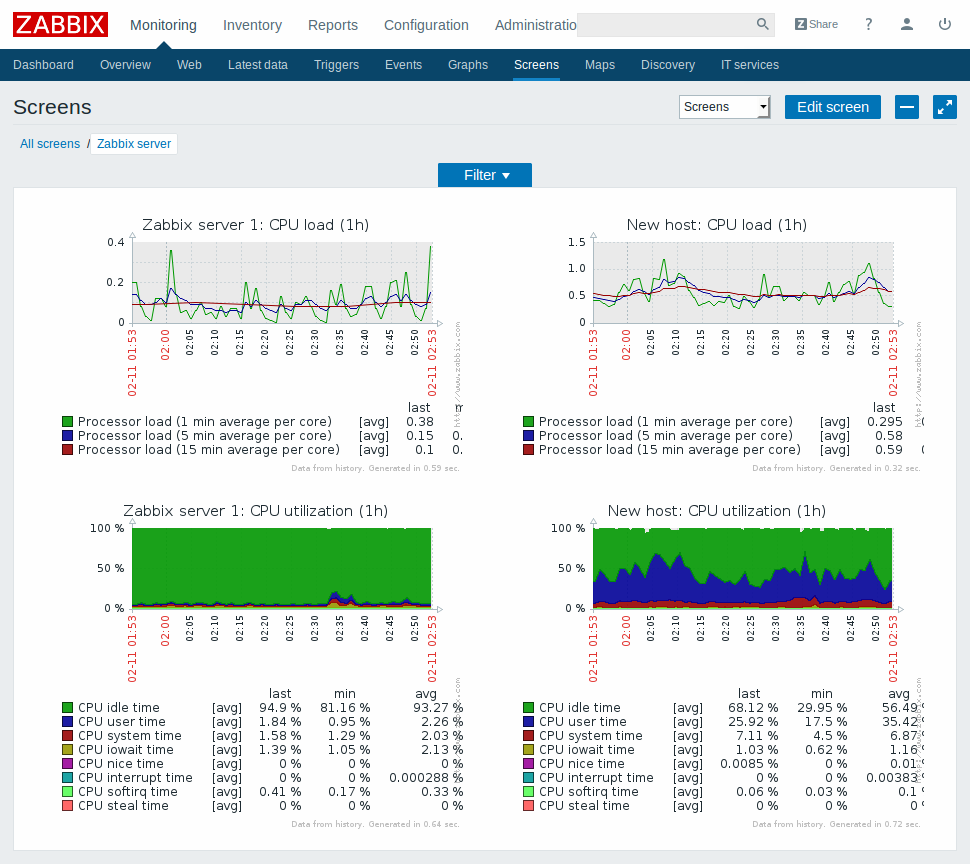
Поддерживает функцию проверки доступности и реакции стандартных сервисов, поддерживает мониторинг загрузки индивидуальных хостов и проверять выполнение внешних программ.

Имеет возможность распределенного мониторинга, создания сценариев на основе мониторинга, автоматического обнаружения новых устройств, создания отчетов, поддержки SLA-мониторинга, поддержки SNMP, создания карты сетей, расширения за счет выполнения внешних скриптов.

Также имеет функцию автоматического оповещения, для которой не надо устанавливать каких-либо дополнительных модулей.

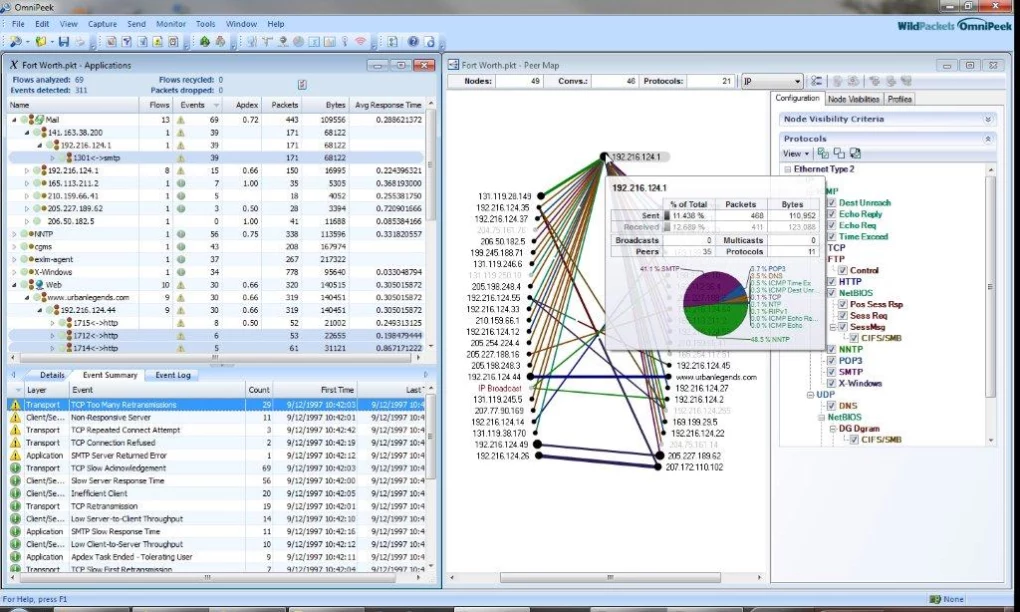
Данная система имеет свой API, который можно использовать для интеграции с другими системами.

С интерфейсом Zabbix можно познакомиться на рисунке 1

Рисунок 1 – Интерфейс системы мониторинга Zabbix

2) Onmipeek — ПО от компании Savvius, предназначенное для полного анализа работы корпоративных сетей. Обеспечивает централизованный экспертный анализ для всех управляемых подсетей. Так же дает возможность анализа и диагностики не только сетей, но и трафика. Имеет возможность расширения через плагины благодаря поддержке API.

С интерфейсом Omipeek можно ознакомиться на рисунке 2

Рисунок 2 – Интерфейс анализатора работы сети Omnipeek

3) SevOne — масштабируемое и высокопроизводительное мониторинговое решение, которое обеспечивает сбор данных, их анализ и отчетность, необходимые для управления работой локальной сети.

Выполняет следующие функции: собирает и хранит статистическую информацию о функционировании распределенной сети, диагностирует каналы связи и контролирует их качественные характеристики посредством использования тестов IP SLA, собирает данные о сетевом трафике с сетевого оборудования, обеспечение детального визуального контроля состояния сетевого оборудования, обнаруживает на ранней стадии изменения в параметрах трафика, которые могут повлиять на функционирование сетевых служб, локализует важные сетевые неполадки, вызванные неверным конфигурированием оборудования или непредвиденным трафиком, решает задачу эффективного планирования наращивания мощностей сетевой инфраструктуры.

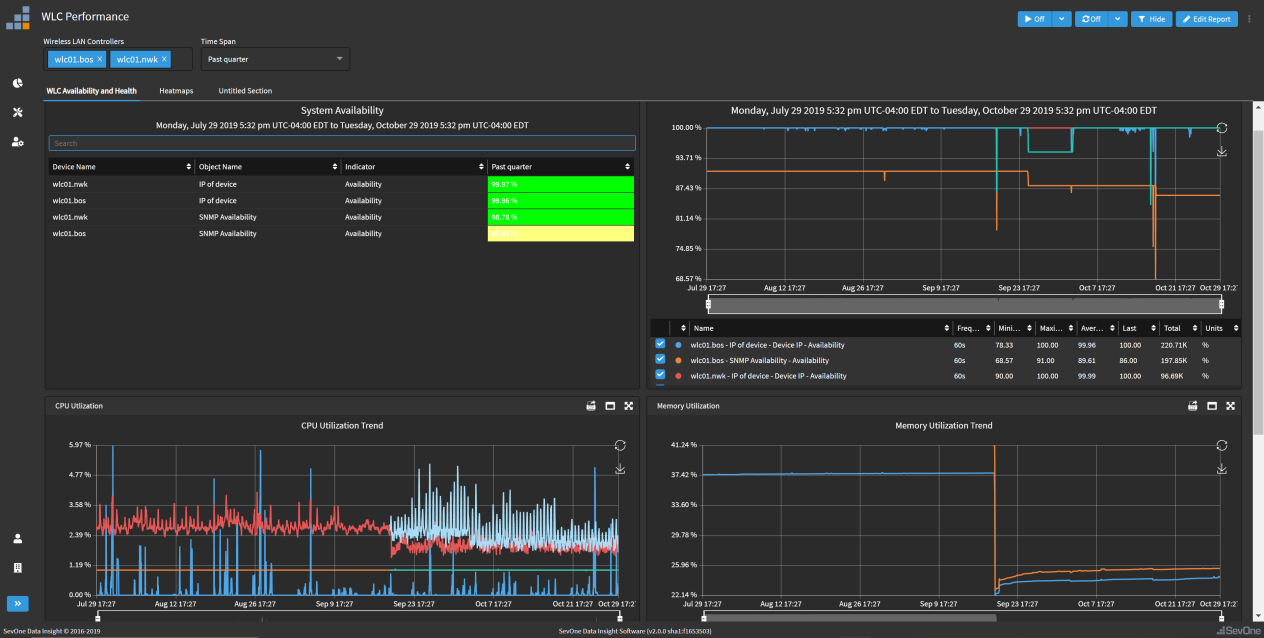
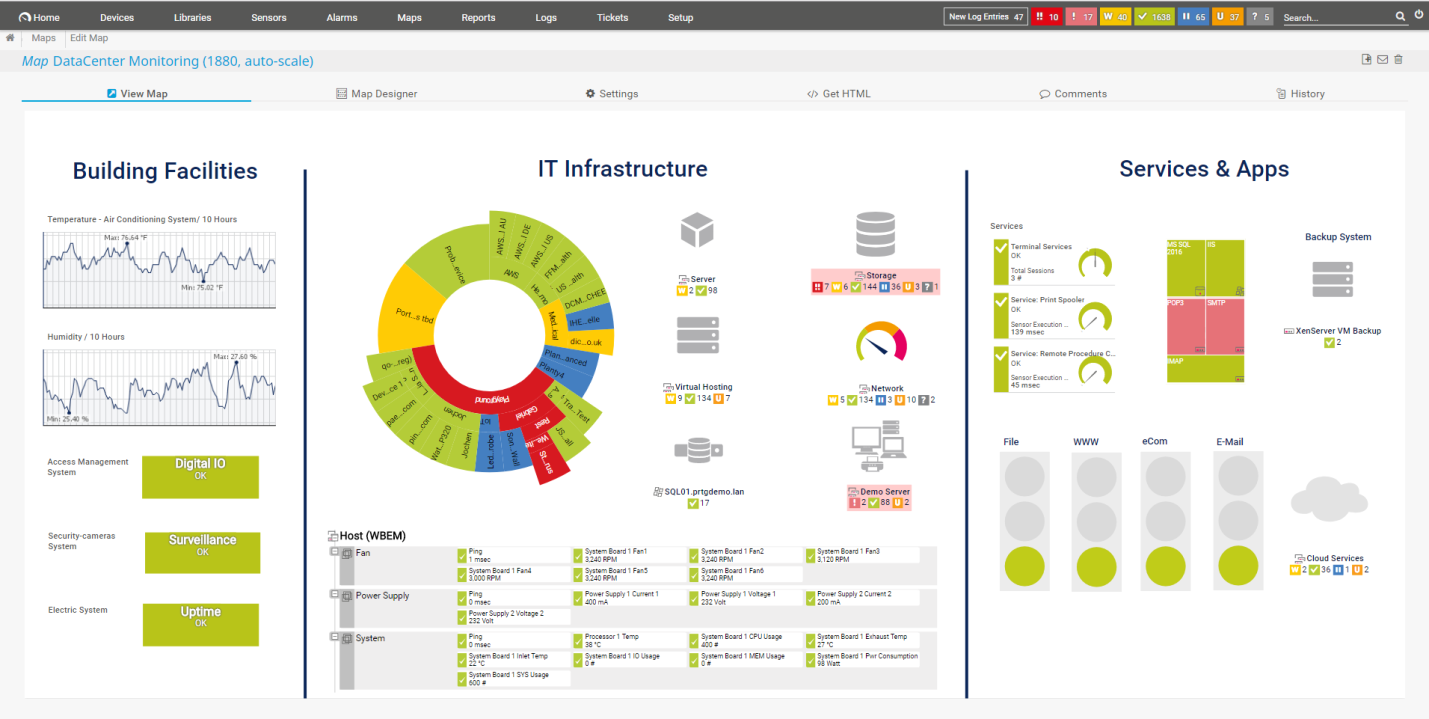
На рисунке 3 представлен пример интерфейса SevOne.

Рисунок 3 – Интерфейс системы мониторинга SevOne

4) PRTG Network Monitor — ПО от компании Paessler AG, предназначенное для безагентного мониторинга использования сети. Имеет интегрированную базу данных, оптимизированную для мониторинга. Благодаря интегрированным технологиям обеспечивает поддержку SNMP, WMI и счетчиков производительности, SSH, анализа трафика через протоколы Flow, REST API, возвращающих XML и JSON. Для большего удобства позволяет создавать карту состояния сети.

Реализует следующие функции: сбор информации о потоках данных, просмотр статистики в базе данных в виде графиков и таблиц, предоставляет средства для анализа трафика сети, автоматическое обнаружение новых устройств, мониторинг исходных исторических данных по различным периодам, поддержка всех распространенных протоколов, мониторинга аппаратного и программного обеспечения, виртуальных сред и приложений.

На рисунке 4 представлен пример интерфейса PRTG Network Monitor.

 Рисунок 4 – Интерфейс системы мониторинга PRTG Network Monitor

5) Network Bandwidth Analyzer — программное решение от компании SolarWinds, предоставляющее возможность мониторинга и анализа сетевого трафика.

Позволяет мониторить сеть, состоящую из оборудования разных поставщиков, анализировать сетевой трафик, мониторить пропускную способность в сети, обеспечивает возможность беспроводного мониторинга и управления сетью, отслеживать WLC трафик, отслеживать, диагностировать и решать проблемы в работе сети.

На рисунке 5 можно ознакомиться с примеров интерфейса Network Bandwidth Analyzer.

## Рисунок 5 – Интерфейс системы мониторинга Network Bandwidth Analyzer

**Анализ и сравнительная таблица характеристик программных продуктов, по функциональным возможностям**

Таблица 1.1 – Сравнительная таблица характеристик ПП, схожих по функциональным возможностям

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Название продукта** | **Мониторинг сети** | **Сбор результатов мониторинга** | **Анализ результатов мониторинга** | **Планирование модернизации** |
| Разрабатываемый продукт | - | + | + | + |
| Omnipeek | + | + | + | - |
| SevOne | + | + | + | +- |
| PRTG NM | + | + | + | - |
| Network Bandwidth Analyzer | + | + | + | - |
| Zabbix | + | + | - | - |

Таблица 1.2 – Сравнительная таблица характеристик ПП, схожих по решаемым задачам

|  |  |  |  |
| --- | --- | --- | --- |
| **Название продукта** | **Сбор результатов мониторинга** | **Анализ результатов мониторинга** | **Планирование модернизации** |
| Разрабатываемый продукт | + | + | + |
| Omnipeek | + | + | - |
| SevOne | + | + | +- |
| PRTG NM | + | + | - |
| Network Bandwidth Analyzer | + | + | - |
| Zabbix | + | - | - |

**Время реакции**

В общем случае время реакции определяют, как интервал времени между возникновением пользовательского запроса и получением ответа на этот запрос.

Вариантами этого критерия могут служить времена реакции, полученные при полностью ненагруженной сети, когда к серверу обращается только один клиент, и нагруженной сети.

При оценке эффективности работы сети не по отношению к отдельным парам узлов, а ко всем узлам в целом используются следующие критерии: средневзвешенные и пороговые.

Средневзвешенный критерий представляет собой сумму времен реакции всех или некоторых узлов при сообщении со всеми или некоторыми серверами сети, получая следующую формулу (1):

(∑i \* ∑j \* Tij)/(n \* m) (1)

где Tij — время реакции i-го клиента при обращении к j-му серверу, n — количество клиентов, m — количество серверов.

Оптимизация функционирования сети по данному критерию заключается в нахождении таких значений параметров, при которых критерий имеет минимальное значение или не превышает заданное число.

Пороговый критерий отражает худшее время реакции по всем возможным сочетаниям клиентов и серверов и имеет следующую формулу (2):

maxij \* Tij (2)

где i и j имеют тот же смысл. Оптимизация может выполняться с целью минимизации критерия или его достижением некоторой заданной разумной величины.

Чаще всего используются пороговые критерии, так как они гарантируют всем клиентам некоторый удовлетворительный уровень времени реакции на их запросы, когда средне-взвешенные критерии могут игнорировать некоторых клиентов с большим временем реакции и все-равно получить приемлемый результат.

**Пропускная способность**

Есть несколько вариантов определения критериев пропускной способности. Эти варианты могут отличаться друг от друга выбранной единицей измерения количества передаваемой информации, характером учитываемых данных (только клиентские или клиентские и служебные), количеством точек измерения передаваемого трафика, способом усреднения результатов.

В качестве единиц измерения обычно используют либо пакеты данных или биты. Так, пропускная способность измеряется либо в пакетах в секунду, либо битах в секунду.

Чаще всего используется измерение в пакетах в секунду, но это может вносить некоторые неточности так как у всех протоколов, кроме протокола ATM, размер пакетов не фиксирован. В качестве эталонных выбраны пакеты минимальной длины, так как они создают для коммутационного оборудования наиболее тяжелый режим работы — вычислительные операции, производимые с каждым принятым пакетом, слабо зависят от его размера, поэтому на единицу переносимой информации обработка пакета минимальной длины требует выполнения большего количества операций, чем для пакета максимальной длины.

Измерение пропускной способности в битах в секунду дает большую точность, чем при использовании пакетов.

Так как в сети данные проходят несколько транзитных промежуточных этапов обработки перед прибытием до узла назначения, то в качестве критерия эффективности работы сети может рассматриваться пропускная способность отдельного промежуточного элемента сети — канала, устройства коммутации или сегмента сети.

Наличие знаний об общей пропускной способности между двумя узлами не может дать полной информации о способах ее повышения, так как невозможно понять какой из промежуточных этапов обработки сильнее тормозит систему. Поэтому данные о пропускной способности отдельных элементов тоже важны для принятия решений по оптимизации.

Общая пропускная способность сети будет равна минимальной пропускной способности элемента сети, а задержка передачи одного пакета будет равна сумме задержек от каждого элемента. Поэтому для повышения пропускной способности составного маршрута сначала необходимо разобраться с самыми медленными элементами.

**Потери пакетов**

Потери пакетов могут происходить по нескольким причинам, таким как:

– переполнение сети, так как сеть достигает максимальной пропускной способности и может испытывать потерю пакетов из-за возросшего трафика;

– ошибки программного обеспечения;

– проблемы сетевого оборудования, как, например, устаревшее оборудование не может поддерживать необходимую скорость передачи данных

**Берстность**

Берстность, или нормированная максимальная битовая скорость — это метрика, которая позволяет оценить качество оборудования, составляющего сеть или систему передачи данных, судить о размере буфера оборудования и вычислять условия надежности.

**Zabbix API**

Zabbix API — это API на основе web и поставляется как часть web-интерфейса. Использует JSON и JSON-RPC для интегрирования со сторонними утилитами и сервисами, т.е. запросы и ответы между клиентом и API закодированы с использованием формата JSON.

Zabbix API позволяет программно извлекать и изменять конфигурацию Zabbix, а также дает доступ к данным истории. Широко используется для следующих задач:

– создание новых приложений для работы с Zabbix;

– интеграция Zabbix со сторонни ПО;

– автоматизация рутинных задач.

Zabbix API состоит из методов, которые условно сгруппированы в отдельные API. Каждый метод выполняет одну отдельную задачу. Большинство API имеют как минимум четыре стандартных метода: get, create, update и delete для получения, создания, обновления и удаления данных соответственно, но некоторые API могут иметь совершенно другой набор методов.

JSON-RPC 2.0 — это протокол без сохранения состояния для создания API в стиле RPC (Remote Procedure Call), использующий JSON для кодирования сообщений.

JSON-RPS работает, отсылая запросы серверу, реализующему протокол. В качестве клиента обычно выступает программа, которой нужно вызвать метод на удаленной системе. Удаленный метод вызывается отправлением запроса на удаленный сервер с помощью HTTP или TCP/IP-сокета.

Все передаваемые данные — простые записи, сериализованные в JSON. Запрос — это вызов определенного удаленного метода на удаленной системе. Он должен содержать три обязательных свойства:

– method — строка с именем вызываемого метода;

– params — массив данных, которые должны быть переданы методу как параметры;

– id — значение любого типа, которое используется для установки соответствия между запросом и ответом.

Сервер должен отослать корректный ответ на каждый полученный запрос. Ответ должен содержать три обязательных свойства:

– result- данные, возвращаемые методом. При ошибке свойство будет установлено в null;

– error — код ошибки, если произошла ошибка, иначе null;

– id — то же значение, что и в запросе, к которому относится данный ответ.

Для примера можно рассмотреть JSON запрос для получения ID, имени и интерфейсов всех настроенных узлов сети с помощью метода host.get и последующий ответ на него:

{

"jsonrpc": "2.0",

"method": "host.get",

"params": {

"output": [

"hostid",

"host"

],

"selectInterfaces": [

"interfaceid",

"ip"

]

},

"id": 2,

"auth": "0424bd59b807674191e7d77572075f33"

}

Ответ сервера:

{

"jsonrpc": "2.0",

"result": [

{

"hostid": "10084",

"host": "Zabbix server",

"interfaces": [

{

"interfaceid": "1",

"ip": "127.0.0.1"

}

]

}

],

"id": 2

}

В большинстве случаев все это делается из скриптов, с помощью инструментов скриптового языка, но всегда остается возможность отправить запросы «вручную», как это было показано выше.

**2. Возможность разработки программного средства**

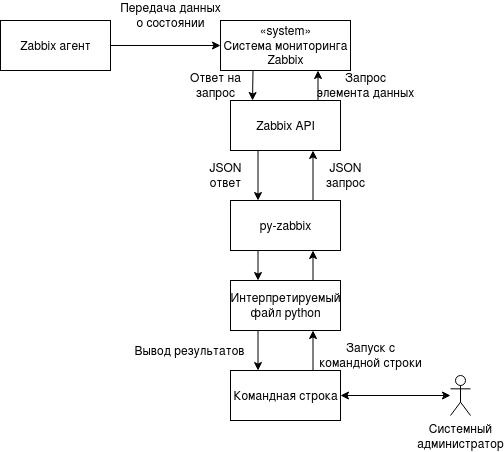
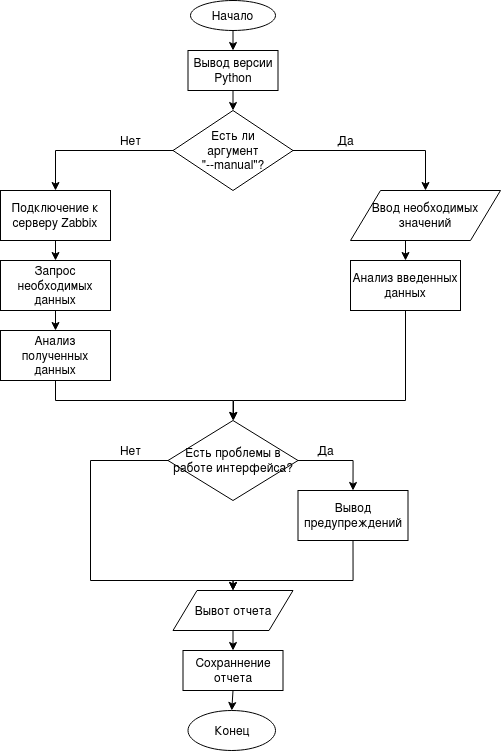
Разрабатываемое программное средство поддерживает коммуникацию с системой Zabbix через Zabbix API с помощью модуля py-zabbix, разработанного для языка программирования Python для безопасности функционирования ИС

Рисунок 2.1 – Архитектура подсистемы мониторинга для учёта в оценке безопасности функционирования

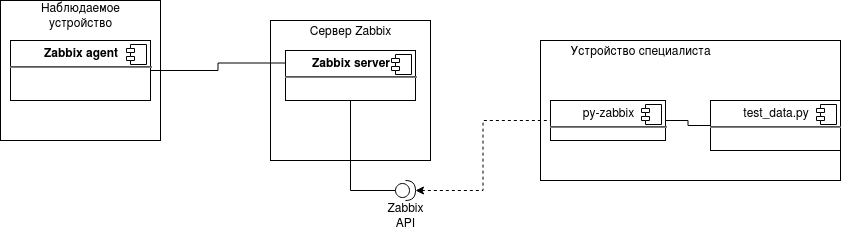
Как видно на схеме, архитектура имеет в себе: устройство, у установленным на нем Zabbix агентом, который передает информацию о работе устройства на Zabbix сервер, интерпретируемый файл Python, который запускается системным администратором с командной строки и получает данные от системы мониторинга через Zabbix API, запросы к которому формируются с помощью библиотеки py-zabbix.

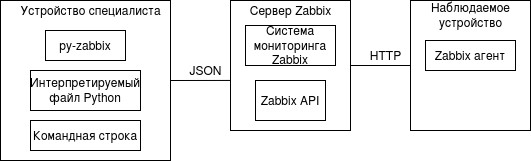
Разрабатываемое программное средство будет оценивать эффективность работы на основе работы интерфейсов наблюдаемого устройства, а, точнее, будет анализировать входящий и исходящий с интерфейсов трафик на количество ошибок и отбрасываемых пакетов данных, так как эта информация может рассказать о корректности настройки сетевого интерфейса и выбранной пропускной способности канала.

Для большей наглядности алгоритм работы программного средства был представлен в виде блок-схемы на рисунке 2.2:

 Рисунок 2.2 – Блок-схема алгоритма работы программного средства

Также на рисунке 2.3 и 2.4 можно увидеть диаграммы компонентов и диаграмму развертывания соответственно.

 Рисунок 2.3 – Диаграмма компонентов

 Рисунок 2.4 – Диаграмма развертывания

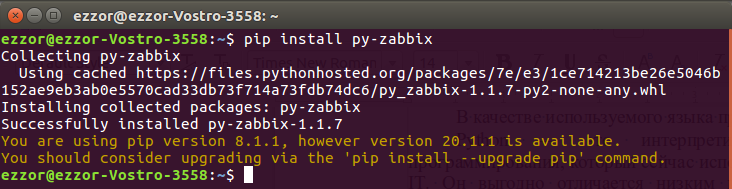
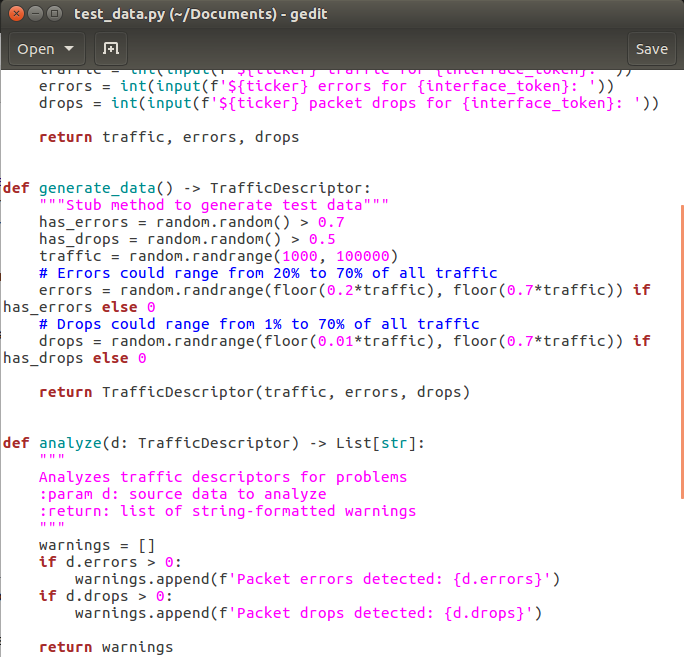
Для работы с Zabbix в Python была установлена библиотека py-zabbix, состоящая из модулей pyzabbix.api для работы с Zabbix API и pyzabbix.sender для передачи метрик в Zabbix. Команда для установки данной библиотеки показана на рисунке 2.5

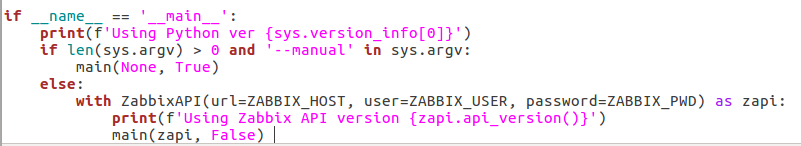
Рисунок 2.5– Установка библиотеки py-zabbix

Так как Python — это интерпретируемый язык, т.е. программы запускаются из командной строки, для написания программ на нем не требуется использование высоко функциональных IDE. Поэтому в качестве среды разработки был использован gedit, стандартный текстовый редактор рабочей среды GNOME.

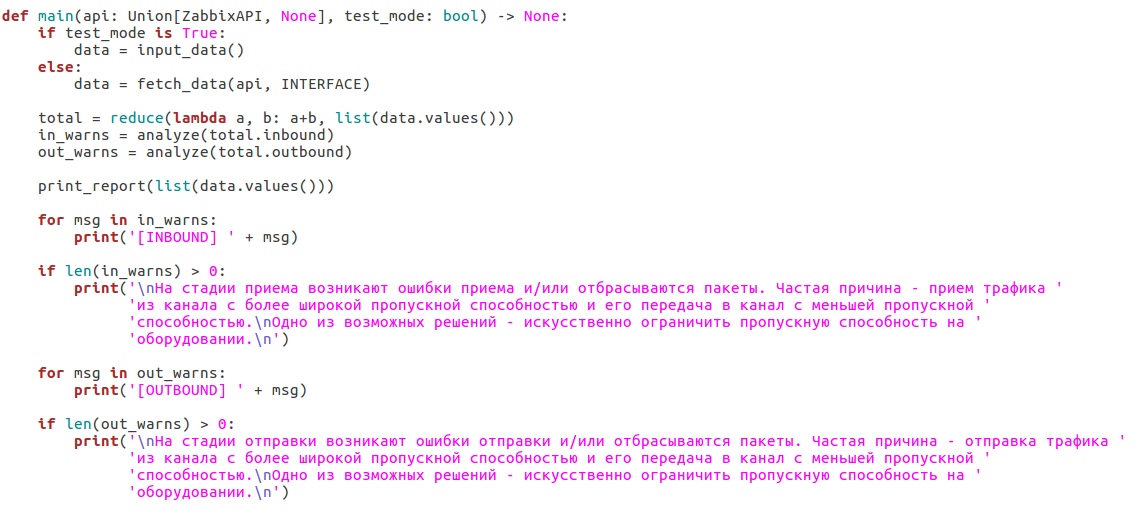
Пример интерфейса gedit с подсветкой синтаксиса

**Описание разработанного программного средства**

При запуске скрипта идет проверка на наличие аргумента «--manual» и переход в режим ввода данных о трафике вручную. Иначе идет подключение к серверу Zabbix. На рисункепоказана часть кода, отвечающая за это.

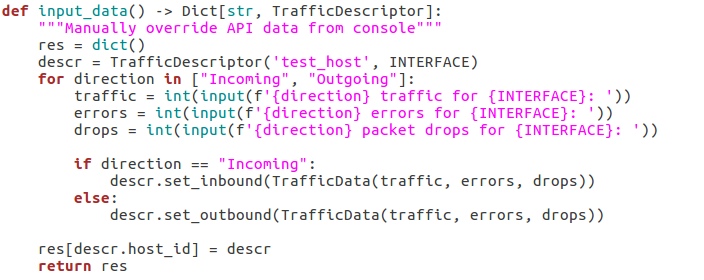
Подключение к серверу Zabbix

Далее вызывается основная функция, которая либо вызывает функцию, которая отвечает за ручной ввод данных о трафике, либо передает необходимые параметры для функции, собирающей данные о работе интерфейсов, и для функции, анализирующей получаемые данные о работе интерфейса. Далее она выводит общее количество полученных и отправленных пакетов и предупреждения, которые появились в процессе анализа. Если количество предупреждений больше нуля, то эта функция выводит общее предупреждение о работе интерфейса и возможную причину некорректной работы.



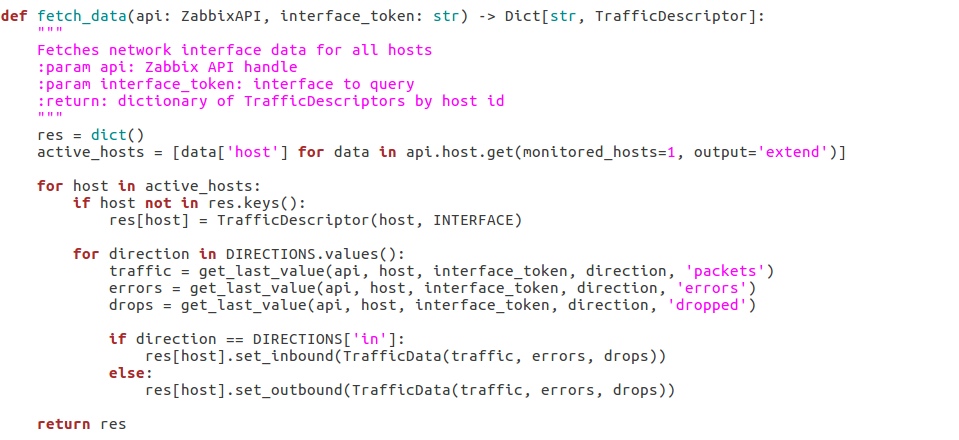
Основная функция

Если был выбран режим ручного ввода, то вызывается функция, которая просит ввести необходимые данные о трафике.



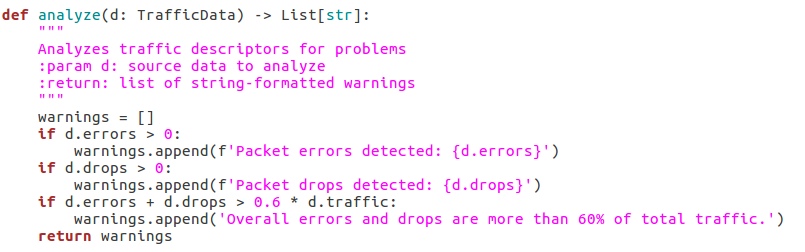
Функция для ввода данных вручную

Далее для сбора необходимых данных вызывается функция, реализующая методы net.if.in и net.if.out, которые собирают информацию о работе порта в соответствии с заданными им параметрами. В параметр «if» входит название интерфейса, а в параметр «mode» входят тип собираемых данных: packets — сбор общего трафика в пакетах, errors — получение количества ошибок на интерфейсе, dropped — получение количества отброшенных пакетов на интерфейсе. Данная функция представлена на рисунке 4.6.



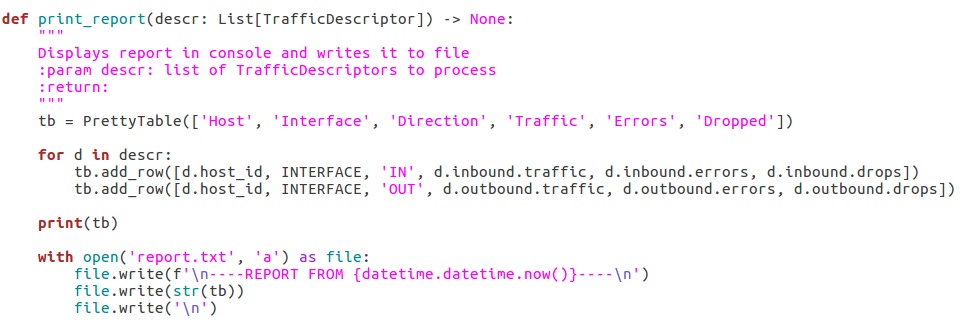
Функция сбора данных

Следующим этапом вызывается функция для анализа полученных данных. Она анализирует данные на наличие ошибок и отброшенных пакетов и добавляет в список предупреждения, если находит таковые. Также выносит предупреждение, если общее количество ошибок и отброшенных пакетов превышает 60% от общего трафика.



Функция анализа получаемых данных

После этого вызывается функция для формирования отчета по результатам анализа полученных данных.



**Выполнение практической работы**

**Задание 1.** Определите архитектуру ИС, доставшейся Вам по варианту.

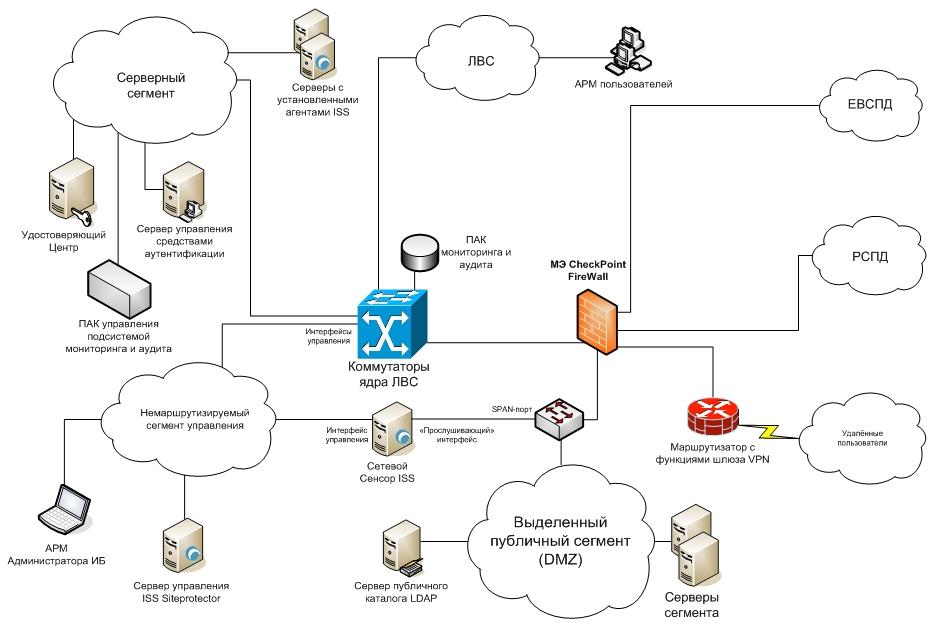
****

Рис 1.1 – архитектура ИС вариант 3

|  |  |  |
| --- | --- | --- |
| Название компонента  ИС | Выполняемая функция | Критический компонент ИС?  Да/нет |
| Удостоверяющий центр | Обеспечение единой системы управления ключевой информацией и цифровыми сертификатами | Да, т.к. используется проверки подлинности и актуальности сертификата со стороны клиента |
| ПАК управления подсистемой мониторинга и аудита | Оперативное оповещение об инцидентах безопасности, генерация отчетов с рекомендациями по управлению ИБ, обработка данных о событиях безопасности, собранных с ПАК мониторинга и аудита | Да |
| ПАК мониторинга и аудита | Сбор событий безопасности с сетевых устройств и агентов | Да |
| Сервер управления средствами аутентификации | Создание, изменение, удаление учетными записями пользователей, обновление/отзыв сертификатов, управление правами (полномочиями) пользователей | Да, т.к. обеспечивает функции аутентификации пользователей |
| АРМ пользователей | Доступ к ресурсам информационной системы | Нет |
| АРМ администратора ИБ | Администрирование сети | Нет |
| Сетевой сенсор ISS | Обнаружение атак на узлы сети | Да |
| Сервер управления ISS Siteprotector | Управление сенсорами обнаружения атак, обновление баз данных сигнатур, получение данных с сенсоров обнаружения атак | Да |
| Сервер публичного каталога LDAP | Хранение открытых ключей сертификатов, списков отозванных сертификатов, а также поддержка публичного доступа к ним | Да |
| Маршрутизатор с функциями шлюза VPN | Поддержка взаимодействия с удаленными подразделениями, защита трафика маршрутизатора, пакетная фильтрация трафика | Да, т.к. при сбоях будет нарушен доступ к данным для удаленных пользователей |
| РСПД (распределенная сеть передачи данных) | Связь администрации компании с филиалами | Да |
| ЕВСПД (единая ведомственная сеть передачи данных) | Связь администрации компании с информационными ресурсами корпоративного провайдера связи и ресурсами Интернет | Да |
| Коммутаторы ядра ЛВС | Высокоскоростная коммутация трафика | Да, .т.к пользователя могут потерять доступ к данным |
| Межсетевой экран (CheckPoint FireWall) | Обеспечение детальной защиты контента, контроль доступа, аутентификацию пользователей, клиентов, сессий, регистрация событий | Да |

**Задание 3.** Для данной конкретной ИС (конкретной архитектуры) сформулируйте понятие «предсказуемое поведение» элемента (системы).

Предсказуемое поведение информационной системы:

* Сбор событий безопасности (ПАК мониторинга и аудита);
* Анализ событий безопасности (ПАК управления подсистемой мониторинга и аудита);
* Управление учетными записями пользователей (Сервер управления средствами аутентификации);
* Обеспечение высокоскоростной коммутации трафика (Коммутаторы ядра ЛВС);
* Обеспечение безопасного доступа к данным (Межсетевой экран, маршрутизатор с функциями шлюза VPN);
* Обнаружение атак на узлы сети, оперативное обновление баз данных сигнатур (Сервер управления ISS Siteprotector);

**Задание 4.** Определите возможные дефекты, причины и возможные последствия работы компонентов ИС.

* АРМ пользователей или АРМ администратора ИБ – причинами сбоев могут быть проблемы с электропитанием (скачки напряжения, короткое замыкание), выход комплектующих из строя, программные сбои. Последствиями может быть полный выход автоматизированного рабочего места из строя, замена на новый.
* Коммутаторы ядра ЛВС или маршрутизатор с функциями шлюза VPN – причинами сбоя могут быть неправильная конфигурация устройства, нарушение целостности кабелей. Последствиями может быть нарушение или полное прекращение отправки/получения сетевых пакетов.
* Удостоверяющий центр, сервер управления средствами аутентификации или сервер публичного каталога LDAP – причинами сбоев могут выступать проблемы с сетью, ошибки в настройках устройств, перегрузка системы (большое количество запросов от пользователей, которое не может быть обработано). Последствиями может быть недоступность данных.
* Межсетевой экран – причинами сбоев может быть некорректная конфигурация устройства. Последствиями может стать нарушение доступа к данным.

**Задание 5.** Проведите оценку влияния отказов компонентов ИС на производительность.

Большая часть компонентов системы являются критичной, так как не продублирована. При отказе таких компонентов как: 1) Сервер управления средствами аутентификации – пользователи потеряют доступ к своим учетным записям и, следовательно, доступ к данным; 2) Маршрутизатор с функциями шлюза VPN – удаленные пользователи потеряют доступ к данным; 3) Сервер публичного каталога LDAP – будет утеряна возможность получать списки отозванных сертификатов, публичные ключи; 4) Межсетевой экран (CheckPoint FireWall) – вызовет нарушение работы удаленных пользователей и связи между администрацией и филиалами.

**Задание 6.** К выше представленному разработанному программному средству дайте свой комментарий. Если возможность оптимизировать функции? Или дополнить функционал?

Программное средство соответствует стандартам разработки, соблюдает функциональную безопасность ИС, поэтому оно не нуждается в дополнительной оптимизации или добавлении нового функционала.

Если возможностей ИС будет недостаточно для удовлетворений потребностей копании, ее можно будет расширить путем добавления нового функционала.

**Вывод:**

Для обеспечения безопасности функционирования информационной системы необходимо проанализировать архитектуру ИС, найти слабые места, и понять, как можно их устранить, сделать надежнее и эффективнее. Критические места необходимо устранить, следовательно, требуется учесть множество факторов, как внешних, так и внутренних, которые могут повлиять на функционирование системы. Если в информационной системе используются сети, необходимо максимально оптимизировать использование сети, постараться избавиться от излишнего трафика и обеспечить бесперебойную работу и соединение.

В контроле за функциональной безопасностью программных средств и безопасностью функционирования информационных систем средства мониторинга и аудита могут сыграть очень большую роль, поскольку данные средства помогают контролировать состояние различных компонентов системы и быстро среагировать на возникающие неполадки.

Список используемой литературы и источников

1. Межсетевой экран Check Point Firewall [Электронный ресурс] – <https://www.azone-it.ru/mezhsetevye-ekrany/checkpoint-firewall>
2. Локально-вычислительная сеть [Электронный ресурс] – <https://ksegroup.ru/services/lokalno-vychislitelnaya-set/>
3. Особенности построения комплексной системы информационной безопасности предприятия [Электронный ресурс] – <https://lib.itsec.ru/articles2/Inf_security/mostransgaz-poleschuk>