# Описание программ

Программа **«Пользователи»** (а конкретно функции программы) - это не просто появляющиеся случайные строки в файле, это реальные обращения к серверу, которые работают с использованием того же модуля симуляций. Использование этого модуля позволяет создать реальную альтернативу посещениям сайта.

Каждая из основных для работы функций «Пользователей» представляет собой имитацию работы посетителя сайта, что включает в себя посещение страниц сайт в определенный промежуток времени и, в некоторых случаях, осуществление атак. В коде нет ограничения на количество активных пользователей, всё зависит лишь от возможностей сервера.

Более того, обращения происходят с разных IP-адресов, что достигается использованием некоторых возможностей сторонних приложений.

Стоит упомянуть также, что помимо программы, к серверу обращаются и реальные пользователи.

Здесь перейдём к описанию программы «Аудитора».

**«Аудитор»** - программа, проверяющая строки в файле логов. Запись в файл логов происходит построчно.

«Аудитор» сравнивает размер файла с последним сохраненным состоянием и открывает его на проверку с последней добавленной строки. Информация по каждой строке фиксируется и сравнивается с шаблоном возможной атаки (например, за какого время пользователь посещал сайт последние несколько раз – таким образом можно определить атаку типа DOS).

Программа «Аудитор» в конечном итоге является полноценной СОПВ, что достигается благодаря:

1. Использованием шаблонов и критериев для проверки действий пользователей
2. Формированием отчёта по проверенным строкам
3. Блокировке нарушителей

# Шаблоны и критерии

Проверка действий пользователей определяется по следующим шаблонам и критериям:

* SQL-инъекция
* DOS-атака
* Парсинг
* Посещение несуществующих ссылок
* Посещение запрещенных ссылок

**SQLi в программе** определяются по списку запрещенных строк, которые могут встретиться в запросе. Например, в инъекциях часто встречаются операторы «SELECT», «AND», «OR» и другие. Наличие подобных и иных аналогичных строк в запросе означает атаку подобного типа. Пользователь, осуществляющий подобную атаку, блокируется из-за высокого уровня уязвимости.

Программное решение **атаки типа DOS** заключается в подсчёте количества посещений за конкретный настраиваемый промежуток времени (например, посещение одной или более ссылок каждую секунду за 60 секунд). При достижение определенного значения счётчиком (который нужен чтобы избежать блокировки на случай, если пользователь, например, хаотично что-то ищет на сайте), подобный нарушитель блокируется известным способом.

**В случае парсинга,** поскольку для нас важно лишь определение того, что какой-то пользователь парсит сайт, программно это определяется так: если пользователь посещает пронумерованные страницы подряд за определенный промежуток времени (так осуществляется парсинг в большинстве случаев) в определенном количестве (например, начиная с трех таких последовательных посещений), то информация об этом заносится в лист отчёта. Пользователь не блокируется, поскольку подобное действие не несёт ощутимой нагрузки на систему из-за малого количества пронумерованных страниц в целом.

**Посещение несуществующих ссылок** – это неоднозначное поведение пользователя, которое не ведёт к использованию уязвимостей системы, но должно быть зафиксировано, поскольку в дальнейшем данный субъект вполне может перейти к непосредственно атакам.

**Посещение запрещенных ссылок** – предполагается, что на сервере не настроены запреты на доступ к каким-либо конфигурационным файлам. Например, тот же файл, содержащий заблокированные IP-адреса, находится в корневой папке сайта. Чтобы избежать получения доступа к этому файлу (например, чтобы разблокировать нарушителей или как минимум узнать их IP-адреса), нужен специальный список, содержащий названия этих файлов. При посещении ссылки со строкой, содержащей название такого файла, пользователь блокируется.

**Формирование отчёта** по проверенным строкам достигается благодаря использованию модуля для работы с Excel, после каждой новой записи в отчёт, происходит его сохранение. Отчёт состоит из нескольких листов, подробно описывающих полученные программой строки, их содержимое, распределение по атакам или иным активностям, а также – статус пользователя, поскольку возможна блокировка.

Собственно, **блокировка нарушителей** осуществляется записью в конкретный конфигурационный файл сервера, где содержится настройка доступа (изначально доступ есть у всех, а блокирование происходит по IP).

# Работа системы

Поговорим о работе веб-сервера.

Поскольку запускать сервер, атаковать и защищать его же с одного рабочего ПК, не совсем целесообразно, то, во-первых, где-то должен быть установлен сам сервер.

Логичным вариантом решения будет установка виртуальной машины, где, собственно, сервер и будет располагаться. Естественно, это предполагает некоторую настройку самой системы и сервера.

Более того, для реалистичности осуществления защиты сервера, должна быть выполнена и его настройка на работу через сеть Интернет.

Для установки веб-сервера и для простоты взаимодействия, была выбрана идентичная работающей операционная система – Windows 10, настроена она была на встроенном компоненте Hyper-V.

Внутри неё была установлено ПО XAMPP, которое содержит необходимые модули для работы веб-сервера. Подключение веб-сервера к сети Интернет, то есть, возможность получить доступ к сайту через неё, осуществляется с помощью модуля pagekite.

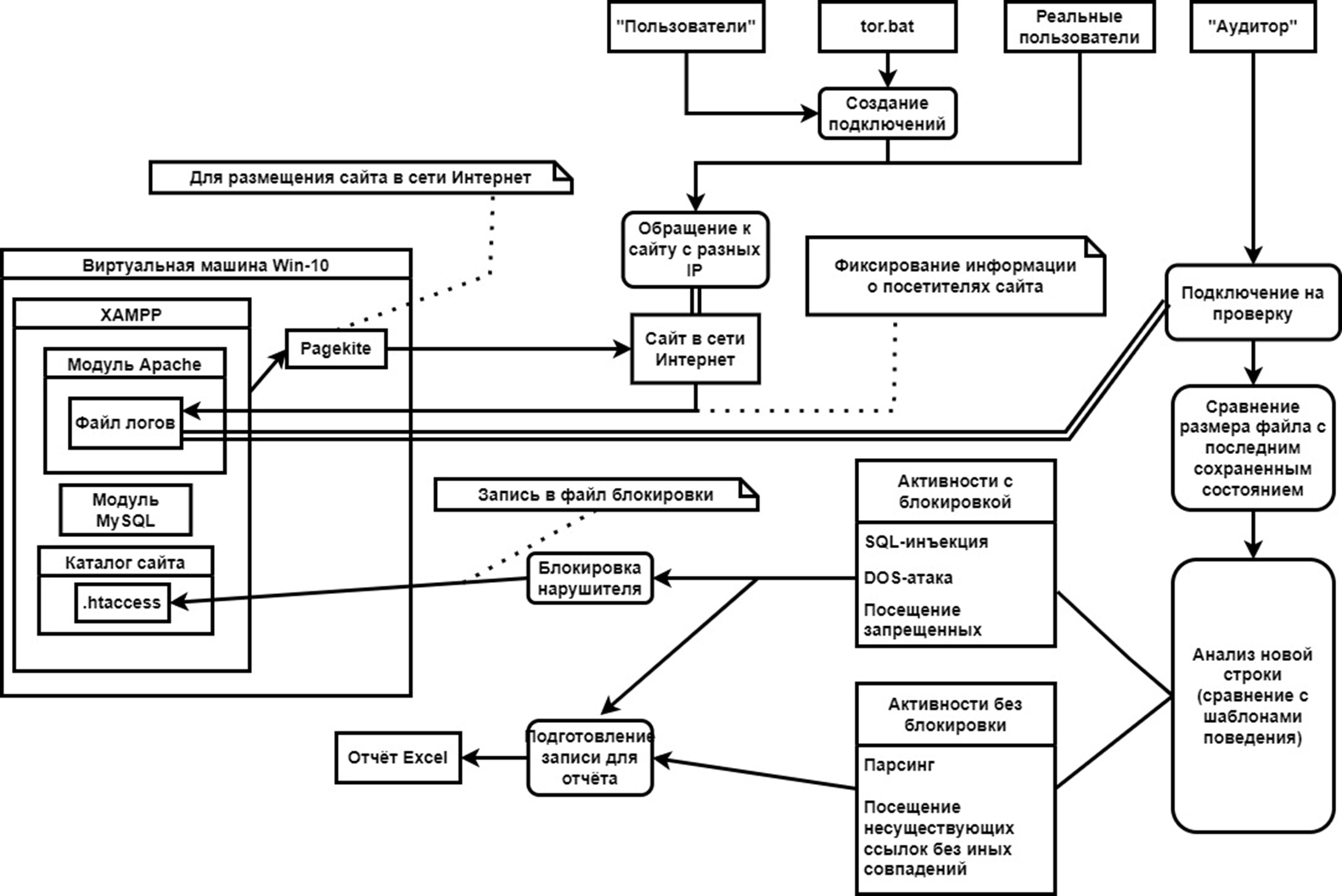
Тестирование выглядело следующим образом:

1. Включаем виртуальную машину на Windows 10, на которой расположен сервер.
2. Запускаем сервер Apache через контрольную панель XAMPP.
3. Запускаем pagekite для доступа к сайту через сеть Интернет.

С рабочего ПК:

1. Запускаем «tor.bat» для доступа к адресам прокси.
2. Запускаем модель «Пользователи», которые будут обращаться к сайту по адресу pagekite.
3. Запускаем программу-аудитора для обработки логов и блокировки нарушителей.

Также параллельно сайт посещали реальные пользователи.



# Тесты

Если говорить о нагрузке, то при одновременной работе модели «Генератора» и программы «Аудитор», можем увидеть следующие результаты.



При определенном количестве пользователей (в данном случае – от пятидесяти тысяч), симуляция начинает работать медленнее, увеличиваясь пропорционально количеству пользователей.

Однако, напомним, что это всё-таки модель, а не реальный сервер.

Проверить такое количество пользователей в связке модели «Пользователи», то есть, работы с сервером, и программы «Аудитор» не представляется возможным из-за технических ограничений pagekite и tor.

Но в целом можно признать систему активного аудита веб-сервера полностью протестированной и работоспособной, тесты показали, что пользователи, нарушающие установленные правила, блокируются, а пользователи, соблюдающие правила, имеют свободный доступ к сайту.