Содержание

[СОДЕРЖАНИЕ 3](#_Toc316859003)

[ВВЕДЕНИЕ 5](#_Toc316859004)

[РАЗДЕЛ 1. СЕРВЕР В ЛОКАЛЬНОЙ СЕТИ. Сетевой трафик 6](#_Toc316859005)

[1.1. Общие понятия о серверах 6](#_Toc316859006)

[1.2. Классификация серверов 7](#_Toc316859007)

[1.2.1. Telnet-серверы 7](#_Toc316859008)

[1.2.2. Почтовые серверы 8](#_Toc316859009)

[1.2.3. Брандмауэры и прокси-серверы 8](#_Toc316859010)

[1.2.4. Web-серверы 9](#_Toc316859011)

[1.2.5. Серверы базы данных 10](#_Toc316859012)

[1.2.6. Файл-серверы 10](#_Toc316859013)

[1.3. Журналирование и журналы сервера 11](#_Toc316859014)

[1.4. Прокси-сервер Squid. Файлы журнала 12](#_Toc316859015)

[1.4.1. Структура файла журнала «store.log» 13](#_Toc316859016)

[1.4.2. Структура файла журнала «access.log» с детальным описанием собственного формата 15](#_Toc316859017)

[1.5. Сервер Samba. Основные файлы журналов 16](#_Toc316859018)

[1.5.1. Структура фала журнала «log.smbd» 17](#_Toc316859019)

[1.6. Анализ сетевого трафика 18](#_Toc316859020)

[1.6.1. Анализ и фильтрация трафика на уровне поставщика услуг 19](#_Toc316859021)

[СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ 21](#_Toc316859022)

ВВЕДЕНИЕ

РАЗДЕЛ 1.  
СЕРВЕР В ЛОКАЛЬНОЙ СЕТИ. Сетевой трафик

# Общие понятия о серверах

Концепция клиент-серверного взаимодействия предполагает наличие условно выделенного компьютера, который предоставляет услуги для множества клиентов. Данную вычислительную систему принято назвать сервером. Основной задачей является, выполнение сервисных функций по запросу клиентов, предоставление доступа к определенным ресурсам.

Для взаимодействия с клиентом сервер выделяет необходимые ресурсы межпроцессного взаимодействия и ожидает запросы на открытие соединения или запросы на предоставляемый сервис. В зависимости от типа такого ресурса, сервер может обслуживать процессы в пределах одной компьютерной системы или процессы на других машинах через каналы передачи данных или сетевые соединения.

Формат запросов клиента и ответов сервера определяется протоколом. Спецификации открытых протоколов описываются открытыми стандартами, например, протоколы Интернета определяются в документах RFC.

В зависимости от выполняемых задач одни серверы, при отсутствии запросов на обслуживание, могут простаивать в ожидании. Другие могут выполнять какую-то работу (например, работу по сбору информации), у таких серверов работа с клиентами может быть второстепенной задачей.

# Классификация серверов

Ниже представлена наиболее распространённая классификация серверов по их функциональным характеристикам.

## Telnet-серверы

Данное определение применимо в отношении к терминальной архитектуре, которая представляет собой организацию сетевой работы информационной системы посредством размещения пользовательских ресурсов на едином центральном сервере. Доступ, к серверам такого вида, осуществляется благодаря рабочим терминалам, которые в выполнены в упрощенной аппаратной реализации. (1)

Telnet-серверы позволяют пользователям подключаться к основному компьютеру и выполнять все поставленные задачи на аппаратном обеспечении удаленной вычислительной системы. Пользователи подключаются к серверу при помощи приложения Telnet, что позволяет получить доступ, не зависимо от местоположения. В большинстве случаев используются следующие протоколы: X11, Telnet, SSH, Virtual Network Computing, ICA, RDP. Протоколы зачастую жестко зависят от используемого оборудования и операционных систем.

## Почтовые серверы

Электронная почта является неотъемлемой частью общения в глобальной сети. Почтовые серверы управляют потоком электронных сообщений, пересылаемых между пользователями компьютерных сетей. В большинстве случаев почтовые серверы сохраняют электронное письмо, что делает их схожими с файловыми серверами.

Разновидностью почтового сервера является сервер почтовой рассылки, который используется для создания, управления и обслуживания адресных списков рассылки. Серверы почтовой рассылки обычно обладают более высокой производительность, чем почтовые серверы. Данный тип почтового сервера используется в основном для распространения электронных журналов, обновлений продуктов, документов технической поддержки, расписаний учебных занятий, рекламных проспектов, а также дискуссионных форумов для клубов и групп.

## Брандмауэры и прокси-серверы

Механизм, который предназначен для защиты от несанкционированного доступа из или в частную компьютерную сеть и, обычно, используется в качестве первой линии обороны при защите частной информации. Брандмауэры могут быть реализованы как аппаратно, так и программно. При правильной организации использования эти средства позволяют предотвратить попытки неавторизованных пользователей получить доступ к соединенной с Интернетом частной сети – в особенности внутренней сети.

Все сообщения, исходящие из или входящие во внутреннюю сеть, проходят через брандмауэр, который проверяет каждое сообщение и блокирует те из них, которые не отвечают установленным критериям безопасности. Существует множество методов межсетевой защиты, включая пакетную фильтрацию, сетевые межпрограммные конверторы, аппаратные шлюзы и прокси-серверы.

Прокси-серверы принято считать наиболее распространенными формами межсетевой защиты. На практике прокси-сервер размещается между клиентской программой и каким-либо внешним сервером. Прокси-сервер эффективно скрывает истинный сетевой адрес, затем отслеживает и перехватывает любые направляемые на внешний сервер или поступающие из Интернета запросы. Это позволяет производить фильтрацию сообщений, повысить производительность и совместно использовать межсетевые соединения.

## Web-серверы

Данные серверы позволяют предоставлять информацию через Интернет посредством языка гипертекстовой разметки (HyperText Markup Language, HTML). При помощи такого программного обеспечения, как, например, Microsoft Internet Information Server, Nginx или Apache, Web-сервер принимает запросы от браузеров и затем отправляет соответствующий документ обратно на тот компьютер, с которого поступил запрос. Для повышения мощности сервера может быть использован ряд серверных технологий, которые позволяют не только доставлять стандартные HTML-страницы, но и применять CGI-сценарии (common gateway interface), протокол безопасности SSL (Secure Sockets Layer).

## Серверы базы данных

Сервер обслуживает базы данных и обеспечивает целостность и сохранность данных при их хранении, а также операциях ввода-вывода при доступе клиента к информации. Сервер базы данных – один из ключевых компонентов в архитектуре вычислительной сети типа "клиент-сервер", в которой пользовательский интерфейс располагаются на менее мощной машине-клиенте, а функции системы управления базами данных размещены на высокопроизводительном сервере. Сервер баз данных работает под управлением серверной операционной системы и, в большинстве случаев, получает клиентские запросы на языке SQL.

Клиентский компьютер посылает свой SQL-запрос на сервер базы данных, который, в свою очередь, обращается в имеющуюся базу данных для обработки этого запроса, а затем возвращает результаты обработки на клиентский компьютер.

Такие системы нуждаются в высокопроизводительной файловой подсистеме и высокой вычислительной мощности, так как поиск в больших массивах данных является сверх ресурсоемкой задачей.

## Файл-серверы

Файл-серверы представляют собой серверы для обеспечения доступа к файлам на диске сервера.

Прежде всего – это серверы передачи файлов по заказу, по протоколам FTP, TFTP, SFTP и HTTP. Протокол HTTP ориентирован на передачу текстовых файлов, но серверы могут отдавать в качестве запрошенных файлов и произвольные данные, например, динамически созданные веб-страницы, картинки, музыку и подобного контента.

Другие серверы позволяют монтировать дисковые разделы сервера в дисковое пространство клиента и полноценно работать с файлами на них. Это позволяют серверы протоколов NFS и SMB.

# Журналирование и журналы сервера

Процесс создания отчета, о событиях, происходящих с объектами сервера и выполняемых задачах, принято называть журналированием. Информация от сервера записывается в файлы, определённым образом структурированные. Такая информация может содержать информацию об ошибках работы сервера, активности пользователей, нагрузку на сервер. В дальнейшем данная информация позволит оценить работу сервера, план по оптимизации конфигурации.

В небольших сетях серверы, наиболее часто, обеспечивают общий доступ к файловым архивам, организацию контроллера домена, общий доступ к принтерам, выступают в роли промежуточного прокси-сервера. Исходя из этого набора заданий, далее будет рассмотрены два наиболее значительных сервера, предоставляющих необходимый функционал.

Прокси-сервер Squid, чаще всего, выступает в качестве фильтра и инструмента контроля трафика между локальной сетью и Интернет.

Samba сервер предоставляет обширные возможности по управлению файлами, реализацию общего доступа к различным ресурсам, поддержку доменной системы.

# Прокси-сервер Squid. Файлы журнала

Squid является высокопроизводительным кеширующим прокси, поддерживающим протоколы передачи данных FTP, GOPHER, HTTP. Основные особенности Squid сервера:

* все запросы выполняет как один, неблокируемый процесс ввода-вывода;
* сохраняет часто запрашиваемые данные в ОЗУ;
* выполняет кеширование DNS запросов, не блокируется при выполнении DNS запросов, и не кеширует неудавшиеся запросы;
* поддерживает SSL, расширенный контроль доступа и полную регистрацию запросов;
* кеши Squid можно расположить иерархически для дополнительного выигрыша в пропускной способности канала.

Журналы Squid представляют источник информации о нагрузке на сервер и его производительности. Журналируются записи не только с информацией о доступе, но также и о системных ошибках и потребляемых ресурсах (память, дисковое пространство).

Все данные о работе сервера записываются в четыре основных файла, которые описывают работу системы.

Файл «squid.out». Данный файл будет создан в случае запуска сервера из скрипта RunCache. Содержимое файла «squid.out» представляет время старта Squid, а также все сообщения о фатальных ошибках, генерируемых неудачным вызовом функции assert().

Файл «cache.log» содержит отладочную информацию и сообщения об ошибках, которые генерирует Squid.

Файл «useragent.log». В файле хранится информация о браузерах и передаваемых ими идентифицирующих заголовках.

Файл «store.log». В данный журнал будут помещены записи об объектах, которые в данный момент сохранены на диск или удалены с диска. Подобное журналирование операций обычно используется в отладочных целях. Точное подтверждение того, находиться ли объект на диске, возможно только после полного анализа файла журнала. Удаление объекта может быть занесено в лог позже, чем сохранение на диск. Файл «store.log» может быть интересен для анализа данных по объектам, хранящимся на пользовательском жестком диске, времени их хранения или информации о том, сколько раз обращались к тому или иному объекту.

Файл «access.log». Наиболее важная информация, о непосредственной работе пользователей и в сети Интернет, пишется в данный файл. Работа большинства программ, анализирующих файл журналов, основана на содержимом «access.log». Стандартная настройка журналирования позволяет получить сокращенную информацию, в то время как создание собственного типа журнала, позволит администратору получить более детальную информацию об определённых особенностях работы сервера, например о работе кеша.

## Структура файла журнала «store.log»

Формат строки, которая заносится в store.log состоит из одиннадцати полей разделенных пробелами в соответствии с функцией storeLog() файла src/store\_log.c:

«%9d.%03d %-7s %08X %4d %9d %9d %9d %s %d/%d %s %s\n»

Строка журнала содержит запись в следующем порядке:

* время, когда запись попала в лог, в UTC с миллисекундами;
* действие, произведенное над объектом (например, CREATE, RELEASE);
* номер файла для размещенного в нем объекта. Номер файла FFFFFFFF указывает на объект, находящийся в памяти. Любой результирующий код для подобного номера файла относиться к объекту, существующему только в памяти, а не на диске;
* код статуса HTTP-ответа;
* значение заголовка HTTP «Date:» ответа;
* значение заголовка HTTP «Last-Modified:» ответа;
* значение заголовка HTTP «Expires:» ответа;
* значение HTTP «Content-Type» либо «unknown», если не может быть определено;
* размер. Состоит из двух значений – заявленная длина содержимого, фактический размер данных;
* метод запроса объекта, к примеру, GET;
* ключ объекта, к примеру, URL.

Пример записанной строки в файл журнала:

1310388332.478 RELEASE -1 FFFFFFFF FCFE604E609B82F4BD3668B10ACD4694 204 1074714690 -1 180221530 image/gif 0/0 GET http://www.google.com.ua/csi?

## Структура файла журнала «access.log» с детальным описанием собственного формата

Рекомендуется использовать собственный формат файла журнала Squid из-за большего количества предоставляемой информации, которая может быть проанализирована впоследствии. Однако для повседневного использования и анализа трафика достаточно стандартной конфигурации. Расширенные параметры логирования следует настраивать только в случае отладки системы или же при изучении определенных особенностей ее работы.

Формат строки вывода, в зависимости от конфигурации, может выглядеть следующей строкой:

«%9d.%03d %6d %s %s/%03d %d %s %s %s %s%s/%s %s»

В данном случае строка состоит, по крайней мере, из 10 полей, разделёнными пробелами.

Ниже описаны основные поля.

Время (time). Временная метка Unix в секунда с начала UTC с миллисекундным разрешением.

Продолжительность запроса (duration). Общее время, которое показывает, сколько миллисекунд заняла у кеша обработка транзакции. Запись заносится в журнал после того, как ответ был полностью отправлен получателю.

IP адрес клиента (client address). Представляет собой IP адрес от которого произошёл запрос. Для того, что бы обеспечить защиту данных может быть настроена опция конфигурации client\_netmask, которая позволяет скрыть информацию о клиенте, что в свою очередь сильно осложнит анализ.

Код результата Squid(result codes). Эта колонка состоит из двух значений, которые разделены слешем. Первым значением является результирующий код, обозначающий статус попадания в кеш. Второе значение, представляет код ответа HTTP о результате выполнения обработки страницы сервером.

Размер переданной информации (bytes). Размер количества данных доставленных клиенту. В этот объем входит вся переданная информация, включая заголовки. Также в результате неудачного запроса возвращается страница с сообщением о ошибке, размер которой записывается в данной колонке.

Метод запроса (request method). Метод запроса на получение объекта, например GET, POST, HEAD.

Адрес запроса (URL). Поле содержит запрошенный URL.

Идентификатор пользователя (rfc931). В данной колонке записан идентификатор пользователя пославшего сапрос.

Код иерархии (hierarchy code). Даное поле может состоять из двух значений, разделённых слешем, кодом, обозначающим тип иерархии, и имя узла, от которого был запрошен объект.

Тип переданных данных (type). Тип содержимого (content type) объекта, извлеченный из заголовка HTTP-ответа. В случае, когда имя типа не указанно, будет установлено значение «-»

Ниже представлена строка журнала «access.log»:

1310388341.533 766 172.16.5.18 TCP\_MISS/200 108233 GET http://www.yandex.ua/ miha\_admin DEFAULT\_PARENT/172.16.11.1 text/html

# Сервер Samba. Основные файлы журналов

Samba – представляет собой свободно распространяемым набором программ, которые реализуют общий доступ к сетевым дискам и принтерам на различных операционных системах. Использует специальные протоколы обмена данными, такие как SMB и CIFS.

Основными функциями Samba можно считать:

* обеспечение разделения файловых систем между UNIX операционными системами и Windows;
* совместное использование принтеров, подключенных в операционных системах Linux и Windows;
* организации системы доменов для клиентов под управлением ОС Windows.

Samba имеет гибкие с настройки аудит. Благодаря дополнительным модулям администратору предоставляется множество всевозможных данных о работе службы. Модуль «full\_audit» позволяет сохранять информацию том кто и когда запросил доступ к конкретному файлу, отмечать события удаления, переименования, изменения файла или каталога.

В зависимости от настроек, информация о доступе к файлам будут записываться в «syslog», либо в лог-файл по умолчанию («log.smbd»)

## Структура фала журнала «log.smbd»

Структура строки значительно меньше, в сравнении с файлом журнала Squid сервера, и состоит из следующих полей:

* дата добавления события, в формате «месяц, день, время»;
* имя сервера;
* тип подключения, например smbd\_audit, su, sshd, ftpd;
* имя пользователя, от которого произошло событие;
* IP-адрес компьютера, от которого был запрошен доступ;
* ключ события
* информация об обработке события;
* путь к файлу или директории, над которой совершались операции

в соответствии вышеописанному списку полей, можно представить следующий пример сохраненных событий:

Mar 5 21:04:01 serv smbd\_audit: vasya|10.44.44.44|pwrite|ok|dir1/somefile.exe

Mar 5 21:04:01 serv smbd\_audit: ann|10.44.44.92|open|ok|r|dir2/database.txt

Mar 5 21:04:01 serv smbd\_audit: editor|10.44.44.34|pread|ok|dir1/somefile.exe

В связи с частым обращением к серверу и активной работой, файл журнала может стремительно увеличиваться в размерах. Чтобы предотвратить использование лишнего пространства, следует периодически очищаться данный файл. Наиболее популярным методом, принято считать, является занесение в список заданий сервера, команды «logrotate».

# Анализ сетевого трафика

На основании множества исследований можно сказать, что при отсутствии фильтрации трафика к глобальной сети, значительная часть сетевых ресурсов затрачивается на предоставление доступа к потенциально опасным и спорным сайтам, и, лишь малую долю, занимают полезные ресурсы, имеющие непосредственное отношение к работе.

Лидерами в списке нежелательных ресурсов являются:

* социальные сети;
* порталы, выкладывающие контент непристойного содержания;
* серверы онлайновых игр;
* ресурсоемкие сайты, генерирующие высокий трафик, отображающие мультимедиа контент;
* сервисы онлайн видео, музыкальные сервисы;
* файловые хранилища и сервисы обмена файлами.

При посещении сотрудниками различных не относящихся к выполняемой ими работе сайтов, помимо нецелевого использования рабочего времени, следует выделить следующие негативные последствия:

* + чрезмерная нагрузка на сеть, вызванная неконтролируемым скачиванием пользователями объемных файлов из Интернет;
  + нерациональное использование ресурсов сети и рабочего времени в результате деятельности использования видео чатов или голосового общения;
  + снижение уровня безопасности внутренней сети. Зачастую посещение сомнительных сайтов и загрузка потенциально опасных файлов, является основной угрозой заражения компьютерной сети предприятия.

Все подобные угрозы безопасности внутренней сети и потенциальное снижение эффективности использования всевозможных ресурсов приводит к необходимости анализа и фильтрации Интернет трафика. Потребность в анализе трафика возросла настолько, что решение, обладающее гибкими механизмами фильтрации, стало жизненно необходимым для множества компаний, использующих Интернет.

## Анализ и фильтрация трафика на уровне поставщика услуг

Основной необходимостью анализа трафика на стороне поставщика услуг Интернет является необходимость учета трафика клиентов, а так же ограничение доступа к тем или иным внешним ресурсам, либо предоставление отчетов об посещенных ресурсах.

Биллинговые системы необходимы для оценки использованного трафика клиентами внутренней сети. Данные программные системы предоставляют детальные виды отчетов используемых сетевых ресурсов по каждому пользователю сети.

Работодатель. Как увидеть, что блокировать

Отдельно вынести Учебное заведение. Нелегальное, ограниченный доступ

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ