

image not found or type unknown

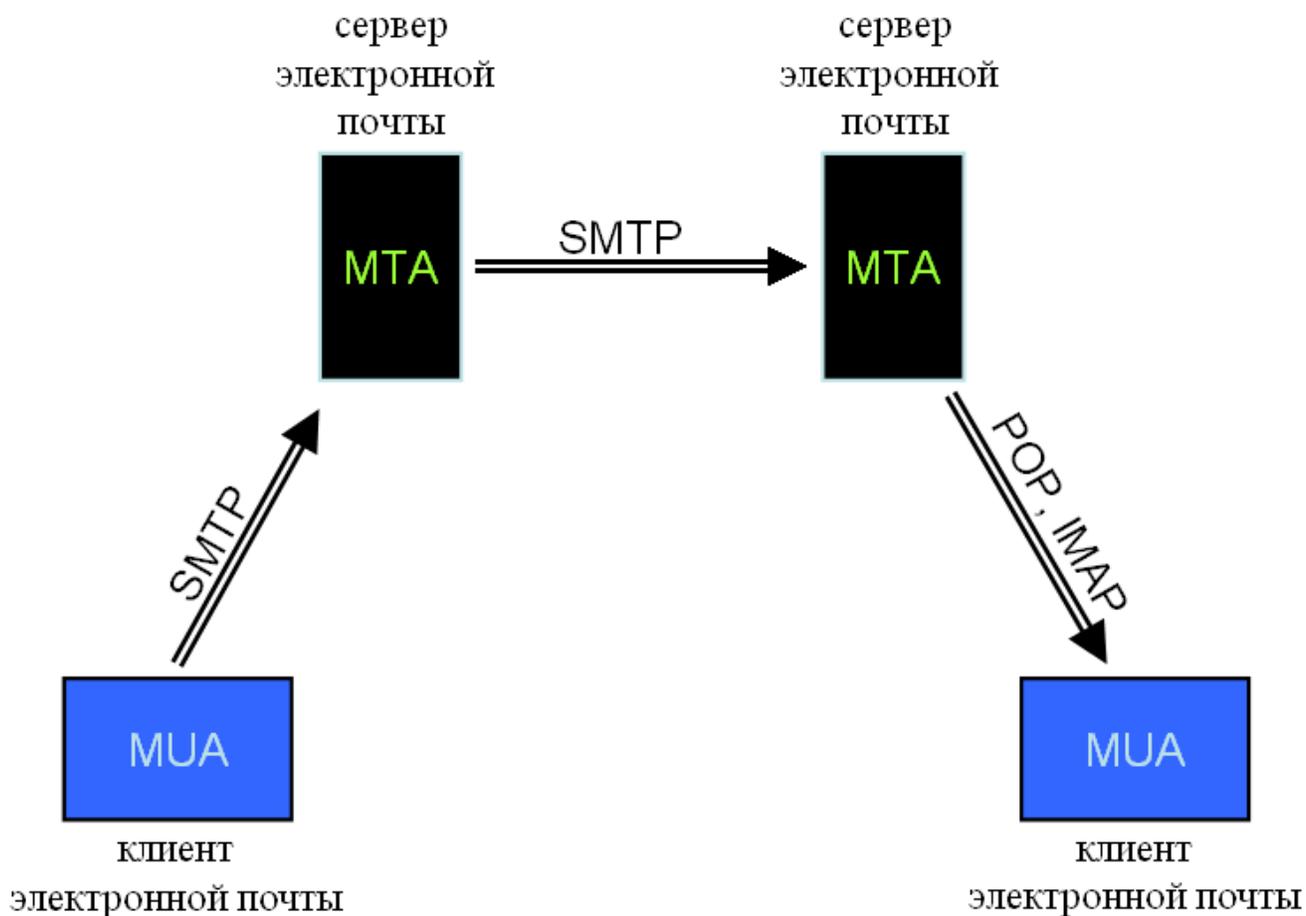


## Введение

Многие системные администраторы испытывают определенные трудности при работе с системами электронной почты. Это неудивительно, почтовый сервер имеет гораздо более сложную структуру, чем файловый сервер, роутер или сервер терминалов. В этой статье мы рассмотрим структуру и принцип работы почтовых серверов, без понимания которых настройка системы электронной почты вполне способна превратиться в шаманские танцы с бубном.

## Содержание

**Почтовый сервер**-в системе пересылки электронной почты так обычно называют агент пересылки сообщений (англ. *mail transfer agent, MTA*). Это компьютерная программа, которая передаёт сообщения от одного компьютера к другому. Обычно почтовый сервер работает «за кулисами», а пользователи имеют дело с другой программой — клиентом электронной почты.



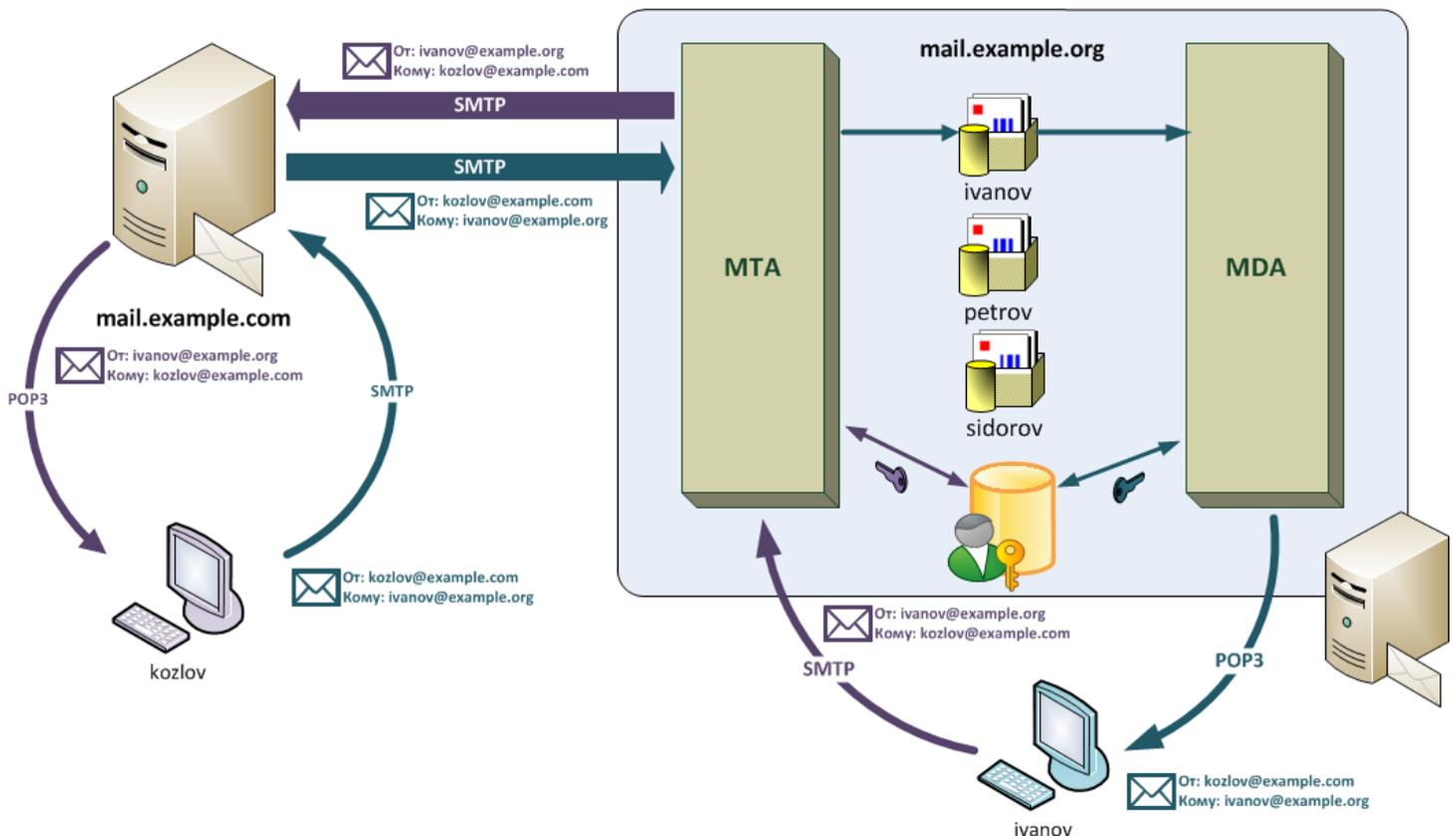
К примеру, в распространённой конфигурации клиентом электронной почты является Outlook Express, однако в последнее время часто используются полноценные версии почтового клиента от Microsoft — Outlook, а также клиента от Mozilla — Thunderbird. Когда пользователь набрал сообщение и посылает его получателю, почтовый клиент взаимодействует с почтовым сервером, используя протокол SMTP.

Для большинства пользователей и начинающих администраторов почтовый сервер представляет собой некий "черный ящик", который получив письмо "неизвестными" путями доставляет его адресату и наоборот. Все взаимодействие с таким сервером заключается в обращении почтового клиента к определенным портам, а то и вообще через веб-интерфейс.

Почтовый серверотправителя взаимодействует с почтовым сервером получателя (напрямую или через промежуточный сервер, который является "черным ящиком", откуда при помощи агента доставки сообщений доставляется клиенту получателя). Часто последние два агента совмещены в одной программе (к примеру, sendmail), хотя есть и отдельные программы (например, postfix и Dovecot) («почтовый ящик»).

Внутри скрыт целый механизм, понимание работы которого имеет ключевое значение для успешной настройки и обслуживания системы электронной почты. Это особенно важно для администрирования серверов на платформе Linux. В отличие от Windows, где почтовый сервер представляет собой законченное программное решение и о внутреннем взаимодействии уже позаботились разработчики, в Linux компоненты почтового сервера представляют собой отдельные программы и настраивать их взаимодействие нужно самостоятельно.

Рассмотрим структуру почтового сервера, а также что происходит, когда пользователь пытается отправить почту.



Важнейшей частью почтового сервера является **MTA** (*Mail Transfer Agent* -- агент пересылки почты) в задачи которого входит прием и передача почты. Очень часто (в Linux / UNIX) MTA называют также почтовым сервером. MTA работает по протоколу SMTP, и его одного, в принципе, уже достаточно для создания системы электронной почты. Когда-то давно именно так и было и для доступа к своему почтовому ящику требовалось обладать определенными техническими знаниями.

Однако прогресс не стоит на месте, MTA, получая письмо, помещает его в почтовый ящик пользователя на сервере, к которому последний должен получить доступ, желательно наиболее простым и понятным способом. Вот здесь на сцену выходит

**MDA** (*Mail Delivery Agent* -- агент доставки почты), его задача по запросу почтового клиента передать ему почту из почтового ящика на сервере. MDA может работать по протоколам POP3 или IMAP, в ряде случаев для "общения" почтового клиента и агента доставки могут применяться собственные протоколы, обладающие расширенной функциональностью, например, MAPI (Exchange Server).

Вопреки распространенному заблуждению, MDA не имеет никакого отношения к процессу передачи почты. Это прерогатива MTA. Если провести аналогию, MTA можно представить, как почтовое отделение, которое занимается приемом и отправкой почты, а MDA с почтальоном, который приносит пришедшую корреспонденцию к вам домой. Если почтальон заболел, то это никак не скажется на работе почты, просто вы не получите письма на дом. Также и MDA, его отказ не приводит к неработоспособности почтового сервера, становится недоступно только получение почты почтовым клиентом, в то же время к ней можно спокойно получить доступ другими путями, например, через веб-интерфейс.

Посмотрим, что происходит при отправке почты. В нашем примере пользователь Иванов, находящийся в домене example.org (ivanov@example.org), пишет письмо Козлову в домен example.com (kozlov@example.com). Для Иванова процесс отправки почты состоит из создания сообщения и нажатия кнопки "Отправить" в почтовом клиенте. Почтовый клиент соединяется с MTA по протоколу SMTP и первым делом сообщает свои учетные данные. Авторизовав пользователя, MTA принимает сообщение и пытается доставить его дальше.

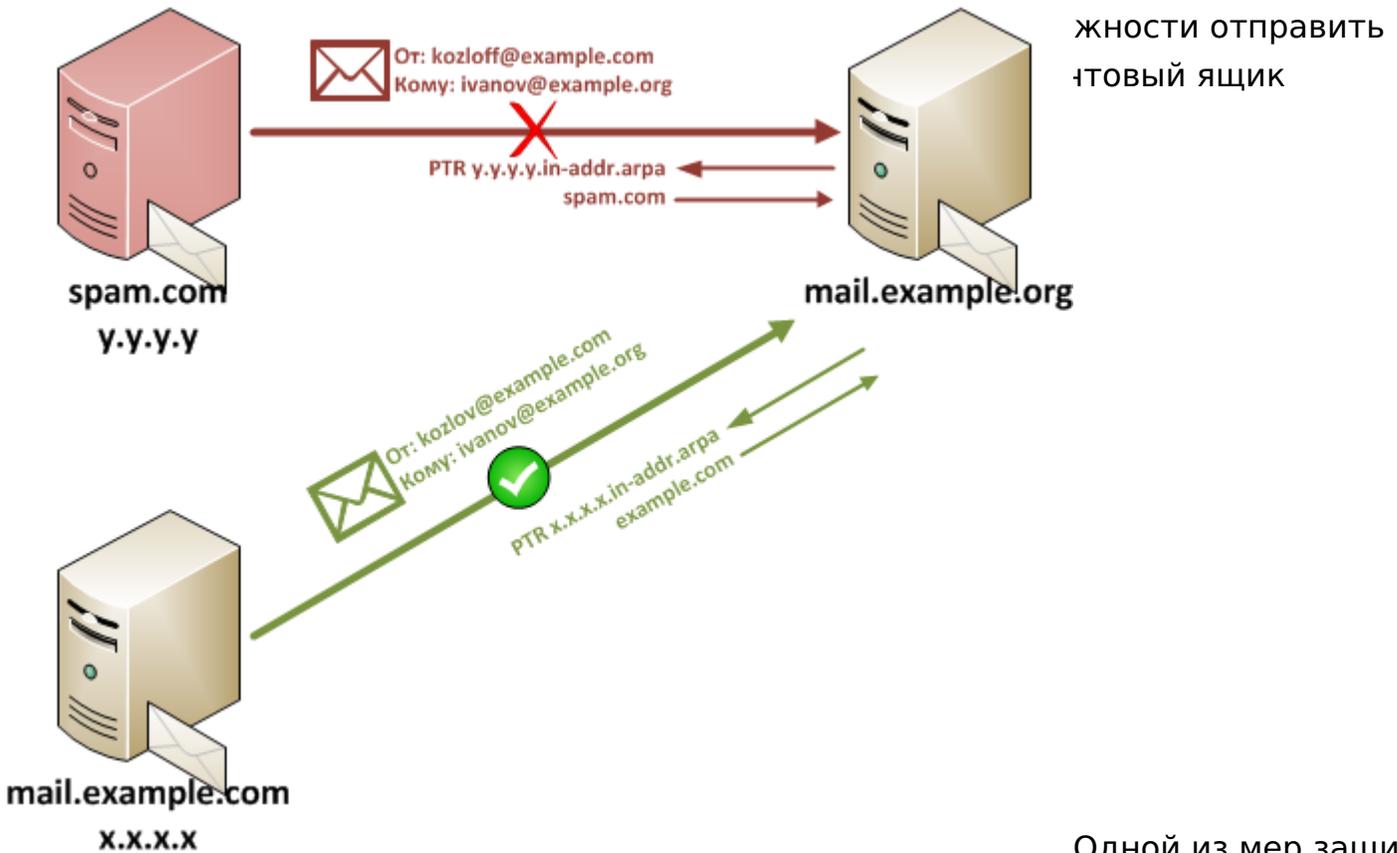
Вообще-то авторизация не является обязательной процедурой для MTA, но без авторизации мы получим открытый релей, т.е. любой может воспользоваться нашим сервером для пересылки почты, а как спамеры обрадуются! В настоящее время открытые релейы возникают в основном из-за ошибок настройки сервера. Однако вполне допустима ситуация, когда MTA без авторизации принимает почту от доверенных пользователей, например, из локальной сети предприятия.

Для авторизации MTA может использовать собственный список пользователей, системный список, списки пользователей LDAP или AD. Также существует способ: авторизация POP прежде SMTP, когда пользователь перед отправкой почты авторизуется на MDA, который, в свою очередь подтверждает аутентификацию пользователя для MTA.

Следующим шагом MTA анализирует служебную информацию письма, определяя домен получателя, если он относится к доменам обслуживаем данным MTA,

производится поиск получателя и письмо помещается в его ящик. Так произошло, если бы Иванов написал письмо Петрову или Сидорову.

Если домен получателя не обслуживается МТА, формируется DNS-запрос, запрашивающий MX-записи для данного домена. MX-запись представляет особый вид DNS-записи, которая содержит имена почтовых серверов, обрабатывающих входящую почту для данного домена. MX-записей может быть несколько, в этом случае МТА пробует последовательно установить соединение, начиная с сервера с наибольшим приоритетом. При отсутствии MX-записи запрашивается А-запись (запись адреса, сопоставляющая доменное имя с IP-адресом) и выполняется



от спама является запрос PTR-записи. PTR-запись (запись указателя) связывает IP-адрес с именем домена. Запрашивая PTR, МТА принимает почту только в том случае если домен отправителя совпадает с доменом отправляющего сервера.

Рассмотрим пример более подробно. Некий спамерский сервер spam.com пытается рассылать письма с поддельным отправителем, якобы от известного нам сервера example.com. В случае фильтрации по белым / черным спискам такое письмо будет

доставлено, так как отправителем числится пользователь из доверенного домена (на что и рассчитывали спамеры). В целях борьбы со спамом МТА формирует запрос PTR записи для IP-адреса отправляющего сервера, который он сообщает в процессе SMTP сессии. Для адреса у.у.у.у PTR-запрос вернет имя домена spam.com, которое не совпадает с доменом отправителя, что будет причиной отказа в приеме данного сообщения. В тоже время сообщения от сервера х.х.х.х будут получены, так как домен из PTR-записи для х.х.х.х (example.com) совпадает с доменом отправителя.

Вывод.

Итак, сообщение получено и находится в почтовом ящике пользователя. Как его прочитать? Почтовое хранилище, где находятся ящики пользователей, может быть организовано самыми различными способами: начиная от банальных папок и фалов, заканчивая базой данных. Не обладая техническими знаниями, прочитать собственную почту вряд ли удастся. Но разве это должно волновать пользователя Иванова? Для него процесс получения почты сводится к нажатию кнопки "Получить" в почтовом клиенте.

Для получения почты клиент устанавливает соединение с MDA по протоколу POP3 или IMAP, обязательно передавая данные для авторизации. MDA проверяет наличие пользователя в списках и, при успешной проверке, передает клиенту все новые сообщения, находящиеся в его почтовом ящике. Пользователь Иванов получает свою корреспонденцию и может работать с ней удобным ему способом.

Список используемой литературы.

1. [https://interface31.ru/tech\\_it/2010/09/pochtovyj-server-struktura-i-princip-raboty.html](https://interface31.ru/tech_it/2010/09/pochtovyj-server-struktura-i-princip-raboty.html)
2. <https://dic.academic.ru/dic.nsf/ruwiki/14365>
3. <https://ru.wikipedia.org/wiki/%D0%9F%D0%BE%D1%87%D1%82%D0%BE%D0%B2%D1%8>