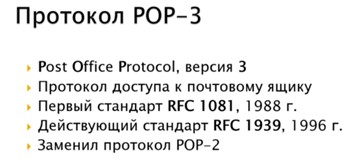
Что касается получения почты. Для получения почты существуют несколько протоколов и на текущий момент используются два.

Первый, самый простой из них, это протокол **POP-3**. Расшифровывается это как Post Office Protocol, версия 3.

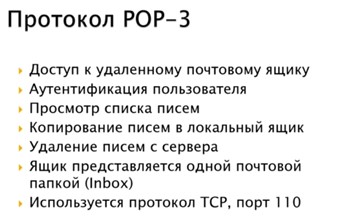
Это протокол доступа к почтовым ящикам. Вы не сможете с помощью него послать письмо. Вы сможете обратиться к почтовому серверу и просмотреть свой почтовый ящик. Протокол довольно таки старый, появился в конце 80-х, та версия, которой мы пользуемся сейчас - это середина 90-х. Он заменил устаревший протокол POP-2. Наверное, когда-то был протокол POP-1.



Сам по себе протокол POP-3 – это, как уже было сказано, протокол доступа к почтовому ящику, он поддерживает функцию аутентификации пользователя ( потому что только сам пользователь, знающий логин и пароль, имеет право доступа к своему почтовому ящику), а также поддерживает несколько ключевых команд:

* просмотр списка писем, которые в настоящий момент есть на сервере
* копирование письма в локальный почтовый ящик, который находится на вашем локальном клиенте
* удаление писем с сервера

Это, собственно, основные функции, которые есть, то есть POP-3 предполагает, что **основная работа** с электронной почтой **происходит на локальном компьютере**, а сервер используют только для того, чтобы хранить вновь поступившие письма, ну или в крайнем случае архив писем. При этом, на сервере ящик представляется **одной большой почтовой папкой** (условно называющейся Inbox) и POP-3 не предполагает возможности хранения на сервере разных почтовых папок. Если вы хотите иметь много разных почтовых папок, то это всё делается на клиенте, уже средствами какогото клиента протокола POP-3. POP-3 использует **TCP**, для него зарезервирован 110-ый порт.



Сам протокол POP-3 очень простой. Он использует текстовый интерфейс, но, при этом, он использует RFC формат команд, но формат ответов там не RFC, а значительно более простой.

Формат ответов здесь – это не код ошибки и сообщение об ошибке, а два ключевых слова: ключевое слово +OK означает, что ошибки не было, ключевое слово –ERR означает, что ошибка была.

Итак, начинается все с аутентификации. Когда клиент подключается к серверу, мы проводим аутентификацию (передача логина и пароля).

Передача логина происходит командой **USER** с параметром “имя почтового ящика”.

Как вы думаете, какой будет ответ, если такого ящика, пользователя не существует, ok или error?

OK

Почему?

Чтобы нельзя было узнать, что этого пользователя не существует.

Конечно, иначе бы это была мечта спамеров.

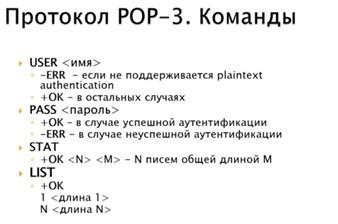
Поэтому, если у нас ошибка с именем пользователя, всё-равно выдаётся OK, если правильный пользователь, выдаётся OK. ERROR выдаётся только в том случае, если мы не поддерживаем передачу пароля в открытом виде *(может, имя пользователя? Тут то мы пароль пока не передаём никакой)*. Как вы видите, в команде USER, это же просто поток TCP, имя пользователя передаётся просто в открытом виде.

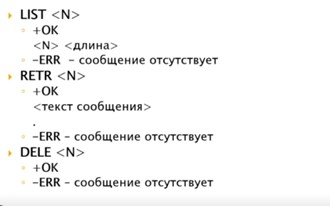
Пароль передается командой **PASS**.

И уже здесь выдаётся OK в случае, если пара “логин пароль” правильная и ERROR, если пара неправильная. Обращаю ваше внимание, пароль передается тоже в открытом виде. Но это всё верно только для plaintext authentication.

После того, как мы успешно прошли аутентификацию, у нас есть несколько команд:

* команда **STAT** показывает статус почтового ящика, в котором два параметра: N и M. **N** - сколько всего писем в настоящий момент имеется в почтовом ящик. **M** - их общая длина в байтах.
* команда **LIST** - пролистать почтовый ящик. Она выдает сначала сообщение +OK, а потом показывает список всех писем (их номера и длины). То есть вы можете посмотреть, сколько есть писем, и какой длины каждое письмо. В основном, длина писем в байтах и количество писем - это информация для почтового клиента чтобы он мог показать процедуру скачивания писем в виде красивого прогресс бара.
* команда **LIST <N>** - это посмотреть информацию об одном конкретном сообщении. Если оно есть с таким параметром, то выдаётся его длина, если нет, то ошибка.
* команда **RETR <N>** - это команда получения письма. По сути, команда скачивания письма. Указывается номер того письма, которое необходимо скачать и дальше, если все хорошо, то вам говорят +OK и прямо в канал TCP выкидывается всё сообщение, и, когда оно закончится, будет точка на отдельной строке. Дальше, почтовый клиент, посмотрев это сообщение, копируют его себе в локальную папку, ну и -ERR приходит, если сообщения нет.
* команда **DELE** **<N>** - помечает сообщение на удаление. А удалять его или нет - зависит от того, как мы закончим транзакцию.
* команда **RSET** - это сброс транзакции и сброс всех флагов сообщений.
* команда **QUIT** - завершение транзакции и удаление всех сообщений, помеченных для удаления.



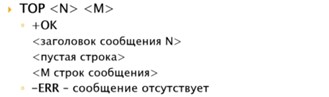


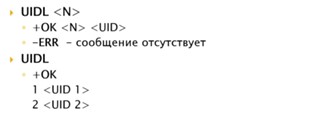


Итого, подключились, прошли аутентификацию, посмотрели состояние ящика, посмотрели список писем, скачали нужные письма, удалили ненужные письма, закрыли соединение.

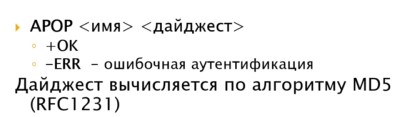
Это тот минимум, который нужен для общения с почтовым ящиком. Дополнительно к этому, в RFC 96-го года появилось несколько команд, которые помогают более осмысленно работать с почтовым ящиком:

* команда **TOP <N> <M>** - позволяет сделать маленький предпросмотр сообщения. Она показывает для **N**-го сообщения его заголовок, пустую строку и первые **M** строк сообщения (в почтовых сообщениях используется формат, что заголовок отделяется от самого сообщения пустой строкой). Таким образом, мы можем посмотреть все заголовки сообщения, не скачивая само тело сообщения. Особенно это полезное, если сообщение очень большого объема и мы даже не знаем нужно нам его скачивать или нет.
* команда **UIDL <N>** (с параметром или без параметра) - эта команда присылает нам уникальный идентификатор сообщения с номером N, или всех сообщений, если команда без параметра. Уникальный идентификатор присваивается почтовым сервером всем письмам, которые сервер получает и он уникальный в пределах данного сервера. То есть, если у вас сервер функционирует, то каждое следующее сообщение будет отличаться идентификатором от предыдущего. Для чего это делается? Если, например, вы хотите сохранить свои сообщения не только на клиентской папке, а еще и на сервере. Зачем, кстати это нужно будет? Почему нам недостаточно хранить сообщения только на своём клиенте? Чтобы можно было с нескольких клиентов посмотреть сообщения. Действительно, если мы хотим иметь доступ с рабочего компьютера, с домашнего, и, например, с планшета, то если первый же будет забирать сообщения и удалять их, то у вас будут разные копии на всех локальных клиентах. Но, с другой стороны, если вы все сообщения храните на сервере, то они у вас там будут накапливаться, и каждый раз при соединении у вас будет показано что на сервере есть сначала 10 писем, потом 20 30 писем, 1000 писем. Но вы же хотите увидеть только новое письмо, а не все которые пришли. А POP-3 этого не позволяет делать, потому что все сообщения представлены одной папкой Inbox. Но вот, собственно, UIDL - уникальный идентификатор сообщения, как раз позволяет отличить новые от старых. Когда почтовый клиент загружает себе все сообщения, которые есть на сервере, он в том числе загружает эти идентификаторы, и в следующий раз, когда он подсоединяется к этому серверу, тот показывает ему список сообщений, мы получаем их уникальные идентификаторы и сравниваем, те, которые были, это старые, те которых не было - это новые. Ну и таким образом мы можем сказать сколько реально новых сообщений находится на сервере, ну и, например, позволить пользователю скачать только их.





* В последних версиях протокола POP-3 есть **не plaintext** аутентификация. Аутентификация командой **APOP <имя> <дайджест>** (Authenticated POP) заменяет пару команд USER и PASS. Эта команда передает имя пользователя в открытом виде, а вместо пароля передает криптографический дайджест от этого пароля. В итоге, сам пароль почти не передается.



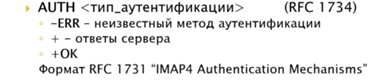
Теперь вопрос на засыпку. А нужно ли хакеру знать сам пароль, если он может прослушивать команду APOP, увидеть имя, увидеть hash пароля и в следующий раз подсоединиться от имени пользователя с этими данными, не зная пароля? А там сервер отправляет какуюнибудь соль при этом? Да, все именно так, конечно, hash считается не только от пароля, но и от строки, которую высылает сервер. Строка примерно такого вида:



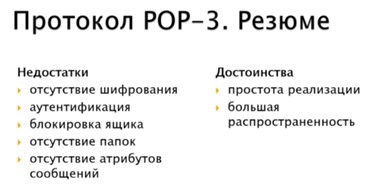
В ней есть pid процесса, системный таймер и, собственно, hostname сервера. Эта строка уникальна, она каждый раз меняется и поэтому дайджест, вычисленный по паролю и по этой строке, фактически каждый раз разный. Поэтому даже если злоумышленник прослушал ваш hash, в следующий раз hash будет другой и это никак ему не поможет.

* Ну и существует еще второй способ аутентификации. Он похож на аутентификацию такую же, как происходит в протоколе SMTP и IMAP4. В SMTP была команда **AUTH**. Здесь такая же команда **AUTH <тип аутентификации>** существует в альтернативных реализациях, как

в SMTP. Но, насколько я знаю, используется она реже, чем команда APOP. APOP в этом смысле конечно значительно проще и по реализации, и по пониманию, и именно с точки зрения протокола все проще. А форматы для этой команды, какие могут быть типы методов аутентификации, указано в RFC 1731.



Ну вот собственно и всё про POP-3, крайне простой протокол, его очень легко разработать. У него есть достоинства и недостатки:



Из достоинств, понятно, это **простота реализации** и **большая распространенность**, все почтовые сервисы поддерживают этот протокол.

Из недостатков, **трафик не шифруется**. Все ваши письма могут прочитать те, кто читает трафик. То есть нужно уже средствами почтовых программ шифровать сообщения.

Кроме того, существует два варианта аутентификации и **один из них опасный** (через команды USER и PASS). То, что я вам не рассказал, но что является фактом, **POP-3 блокирует почтовый ящик**. То есть на время работы с почтовым ящиком клиент блокирует ящик на сервере и никакой другой клиент к нему в этот момент не может присоединиться. Это сделано из-за особенностей POP3, так как там есть команды, которые выдают конкретный список папок и так далее и сообщений и у вас конечно может быть проблема, если параллельно кто-то присоединился и какие то сообщения удалил, например. Кроме того, к минусам можно отнести **отсутствие серверных папок**, то что сообщение-это единый блок это, конечно, очень плохо. Ну и **нет возможности вешать на сообщения метки**, что было бы очень полезно сейчас (важное там сообщение, новое и тд, POP-3 ничего этого не поддерживает).