

Post Office Protocol (POP) 3 Protocol

Simulate Hundreds of Thousands of POP3 Users Retrieving Electronic Mail Messages from Remote Servers

The Post Office Protocol (POP) has a rich history and is one of the primary reasons that the Internet has become so popular over the past 15 years. POP was originally defined in October of 1984 by J. K. Reynolds on behalf of the Internet Engineering Task Force (IETF). The original "Request for Comment" document (RFC 918) was only five pages long. It described only twelve possible requests and responses between client and server for the POP1 protocol. Shortly after the original RFC was published, Post Office Protocol Version 2 (POP2) appeared as RFC 937. In addition to the original author, Jon Postel and several others were added as authors of the twenty-four page RFC. The next revision of POP came with RFC 1081 in 1988, however this document was rendered obsolete by RFCs 1225, 1460, 1725, and finally 1939. Additionally, POP extensions have been defined by other RFCs for things such as Transport Layer Security (TLS) and additional authorization techniques.

The primary purpose of POP was to allow a workstation to access electronic mail on a remote server. The original POP standard was built on two principles: "if anything goes wrong, close the connection" and "have few options". POP expects that electronic mail transmission will occur via some other transfer protocol, namely the Simple Mail Transfer Protocol (SMTP). POP3 is not designed to provide a mechanism for manipulating the messages on the remote server, but simply to download the electronic mail to a local store. The Internet Message Access Protocol (IMAP) protocol provides for more advanced electronic mail manipulation.

The Internet Assigned Numbers Authority (IANA) has assigned TCP port 110 for POP with an alternate port assigned at 995. While most common POP clients use 110 for communication, some servers such as Google Gmail utilize the alternate port. Upon successful completion of a TCP connection to a POP server, the server responds with a greeting. Once the server has issued the greeting, the client and server exchange commands and responses until the session is completed.

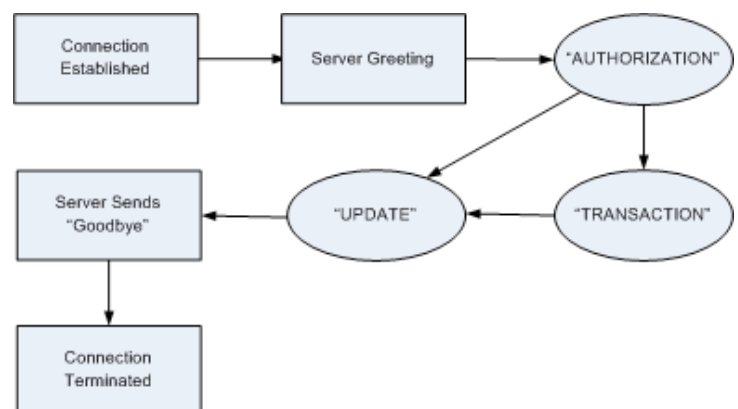
POP sessions traverse a simple state machine that allows the client and server to understand the appropriate commands at a given time. The state machine consists of three states: "AUTHORIZATION", "TRANSACTION", and "UPDATE". The typical path of the state machine

BreakingPoint Testing Tools Simulate the POP3 Protocol:

- BreakingPoint is able to generate hundreds of thousands of POP3 sessions per second using a single, easy-to-use test product.
- Utilizing blended applications and security, users can fully test their network infrastructure to ensure that POP3 transactions are handled properly under full network load.
- Utilizing Application Manager allows users to create both normal and abnormal flows to fully qualify POP3-aware devices performing protocol validation.

begins when the TCP connection has been established and the server has sent its greeting. Once this happens, the session enters the "AUTHORIZATION" state.

Figure 1 -The Three Basic POP States



When in this state, the client must present the appropriate authentication credentials to access the user's mailbox. Once proper credentials have been supplied, the session moves into the "TRANSACTION" state. In this state, the client is able to send commands to the server requesting various behaviors. Once completed, the client issues a "QUIT" command to enter the "UPDATE" state, where the server then issues the GOODBYE and closes the connection.

While the previous scenario describes a basic POP3 transaction flow, many other transitions through the state machine are supported. Figure 1 shows the POP3 state machine with all possible transitions.

Utilizing the BreakingPoint Elite's support of the POP3 protocol, users are able to fully test deep packet inspection (DPI) and content-aware devices as they track individual sessions through the various states. BreakingPoint's unique ability to modify and script protocol flows provides the ability to test devices under unusual state transitions or commands issued at incorrect times in the state machine flow.