SSL/PHP Web Server

Steven Lu

Functions

- Serve static content (Web Server)
- Send content securely (SSL)
- Execute scripts (PHP execution)
- Multi-threading

Web Server

1. Receive header
   `“GET /index.html HTTP/1.1”`
2. Parse header in parts, specifically for the filename.
   `“/index.html”`
3. Sandbox file, access only within defined directory
   `“content/index.html”`
4. Access file, generate response headers.

Web Server

5. If file not found, send proper header
   `“HTTP/1.1 404 Not found”`
6. If found, determine MIME type and file size.
   `“Content-Type: text/html”`
   `“Content-Length: 638”`
7. Send headers.
8. Send data through chunks.

SSL Encryption

- Uses GNU TLS Library, most functionality already taken care of by this library.
- A matter of setting up with proper certificates and binding TLS to sockets.
- Handshake, send data through TLS session.

SSL Encryption

1. Set up certificates, initialize GNU TLS lib.
   `“gnutls_certificate_set_dh_params()”`
2. Set up generic server.
   `“bind(); listen();”`
3. Setup TLS session to sockets.
   `“initialize_tls_session()”`
4. Handshake on requests.
   `“gnutls_transport_set_ptr(); gnutls_handshake();”`

SSL Encryption

5. Receive content through session.
   `"gnutls_record_recv()"`
6. Send content through session.
   `"gnutls_record_send()"`

PHP Execution

1. During MIME detection, detect PHP.
2. Close current file pointer, create a new one based on the output of script execution.

   `“popen(/usr/bin/php ./content/index.php, ‘r’)”`

Multi-threading

- fork() after each accept.
- Handle children and parent processes appropriately.