OpenSSL – проверка POP3, IMAP, SMTP, WEB сервера с TLS/SSL/SASL

Использование OpenSSL для тестирования POP3, IMAP, SMTP, WEB сервера с TLS/SSL/SASL

POP3

Для тестирования работы TLS/SSL на POP3 сервере можно использовать входящий в
состав OpenSSL клиент s_client (для тестирования SSL-клиентов по аналогии можно
использовать s_server):

```
openssl s_client -connect имя_хоста:995
```

после чего можно сэмулировать типичную POP3-сессию:

```
+OK Dovecot ready.
user логин
+OK
pass пароль
+OK Logged in.
```

выводим список сообщений на сервере и их размер:

```
list
+OK 2 messages:
1 1759
2 12422
```

читаем первое сообщение:

```
retr 1
+OK 1759 octets
заголовки и текст
```

IMAP

Тестирование IMAP проводится в соответствии с теми же принципами:

```
openssl s_client -connect imap_хост:993
CONNECTED(00000003)
....
* OK [CAPABILITY IMAP4rev1 LITERAL+ SASL-IR LOGIN-REFERRALS ID ENABLE AUTH=PLAIN AUTH=LOGIN] Dovecot ready.
login логин пароль
a001 OK [CAPABILITY IMAP4rev1 LITERAL+ SASL-IR ... CONTEXT=SEARCH LIST-STATUS QUOTA] Logged in
```

выводим список папок:

```
a002 list "" "*"
* LIST (\HasChildren) "." "INBOX"
* LIST (\HasNoChildren) "." "INBOX.INBOX_Trash"
* LIST (\HasNoChildren) "." "INBOX.Trash"
* LIST (\HasNoChildren) "." "INBOX.read"
* LIST (\HasNoChildren) "." "INBOX.Queue"
* LIST (\HasNoChildren) "." "INBOX.INBOX_Drafts"
```

посмотрим содержимое папки Inbox:

```
a003 examine inbox
* FLAGS (\Answered \Flagged \Deleted \Seen \Draft Junk NonJunk)
* OK [PERMANENTFLAGS ()] Read-only mailbox.
* 10 EXISTS
* 0 RECENT
* OK [UNSEEN 1] First unseen.
* OK [UIDVALIDITY 1291459647] UIDs valid
* OK [UIDNEXT 8026] Predicted next UID
* OK [HIGHESTMODSEQ 2710] Highest
a003 OK [READ-ONLY] Select completed.
```

В папке 10 сообщений, выведем содержимое текста четвертого сообщения, без заголовков:

```
a004 4 rfc822.text
* 4 FETCH (RFC822.TEXT {857}
```

```
 текст
 a005 OK Fetch completed.
```

выходим

```
 a005 logout
 * BYE Logging out
 a005 OK Logout completed.
```

Тестируем сайт по SHTTP:

```
openssl s_client -connect www.test.com:443
GET / HTTP/1.1
Host: test.com
```

SASL

Проверяем SASL-аутентификацию при отправке почты:

```
openssl s_client -connect smtp_хост:25 -starttls smtp
```

```
 220 mail.test.com ESMTP Postfix
 EHLO test.test.com
 250-mail.test.com
 250-PIPELINING
 250-SIZE 10240000
 250-ETRN
 250-AUTH DIGEST-MD5 PLAIN CRAM-MD5
 250 8BITMIME
 AUTH PLAIN пароль_в_base64_нотации
 235 Authentication successful
```

BASE64

Перекодировать файл с паролем в base64-представление можно командой:

```
openssl enc -base64 -in file.txt -out mfile.b64
```

декодировать:

```
openssl enc -d -base64 -in file.b64 -out file.txt
```

Другие полезные команды:

Шифруем файл симметричным шифром *blowfish* (если необходимо сохранение в
base64-представлении добавляем опцию “-a”):

```
openssl enc -e -salt -bf -in file.txt -out file.blowfish
enter bf-cbc encryption password: пароль
```

расшифровываем:

```
openssl enc -d -bf -in file.blowfish -out file.txt
enter bf-cbc decryption password: пароль
```

Рассчитываем *SHA1-хэш* для файла:

```
openssl dgst -sha1 -c file.txt
SHA1(test.txt)= 15:85:f1:af:a7:ca:1c:1c:5a:8b:c3:a7:1e:7f:4b:bd:3c:d4:22:ca
```

Для перехвата и расшифровки SSL/TLS трафика в отладочных целях можно использовать утилиту *ssldump*:

наблюдение за активностью внутри SSL-сессии:

```
ssldump -a -A -H -i eth0
```

со служебными данными для полной отладки SSL-сессии:

```
ssldump -a -A -H -k server.pem -i eth0
```

для расшифровки содержимого сессии:

```
ssldump -d -k server.pem -i eth0
```

Для успешной расшифровки SSL-сессия должна быть перехвачена с самого начала,
включая момент обмена ключами на стадии инициирования сессии.