Playing with Openssl, part 2.

by Philippe Bogaerts, alias xxradar.
http://www.radarhack.com
mailto:xxradar@radarhack.com.

Version 1.0 20-09-2003

1. Introduction

Openssl is a very versatile tool. The next couple of chapters show how to use openssl to test various SSL and other security related things. I actually started using openssl in trying to get a more in depth knowledge about ssl en encryption.

2. Interrogating https servers.

Everybody knows how you can try to figure out some details about an http server by telnetting on port 80. It becomes more difficult if that possible vulnerable webserver talks https. The following example shows how you can use openssl to replace telnet.

Create a file command.txt, containing "GET / HTTP/1.0" followed by two CR/LF, if you want to save you the burden of typing the commands blindly.

```
C:\openssl>type command.txt | openssl s_client -connect www.radarhack.com:443
Loading 'screen' into random state - done
CONNECTED(000002C8)
depth=0 /C=US/ST=VA/L=Chantilly/O=Plesk Inc./CN=plesk.com/Email=support@plesk.com
verify error:num=18:self signed certificate
verify return:1 depth=0 /C=US/ST=VA/L=Chantilly/O=Plesk
Inc./CN=plesk.com/Email=support@plesk.com
verify error:num=10:certificate has expired
notAfter=Nov 10 23:24:41 2001 GMT
verify return:1
depth=0 /C=US/ST=VA/L=Chantilly/O=Plesk Inc./CN=plesk.com/Email=support@plesk.com
notAfter=Nov 10 23:24:41 2001 GMT
verify return:1
---
Certificate chain
 0 s:/C=US/ST=VA/L=Chantilly/O=Plesk Inc./CN=plesk.com/Email=support@plesk.com
   i:/C=US/ST=VA/L=Chantilly/O=Plesk Inc./CN=plesk.com/Email=support@plesk.com
---
Server certificate
-----BEGIN CERTIFICATE-----
MIIDPzCCAqigAwIBAgIBADANBgkqhkiG9w0BAQQFADB5MQswCQYDVQQGEwJVUzEL
MAkGA1UECBMCVkExEjAQBgNVBAcTCUNoYW50aWxseTETMBEGA1UEChMKUGxlc2sg
SW5jLjESMBAGA1UEAxMJcGxlc2suY29tMSAwHgYJKoZIhvcNAQkBFhFzdXBwb3J0
QHBsZXNrLmNvbTAeFw0wMDExMTAyMzI0NDFaFw0wMTExMTAyMzI0NDFaMHkxCzAJ
BgNVBAYTAlVTMQswCQYDVQQIEwJWQTESMBAGA1UEBxMJQ2hhbnRpbGx5MRMwEQYD
VQQKEwpQbGVzayBJbmMuMRIwEAYDVQQDEwlwbGVzay5jb20xIDAeBgkqhkiG9w0B
CQEWEXN1cHBvcnRAcGxlc2suY29tMIGfMA0GCSqGSIb3DQEBAQUAA4GNADCBiQKB
gQCdu0cjQwLuPs7CgCZB/0mvhJuflLN3wV+jM5By4W/ykZ2hkH22BnS2TWucvoqm
Zp3qGaebXwT76xG5AaXFLKUgAB8tV3q5jTGVujAa1YE/AnP5c+jJCVLVkYf3jFnz
jBftB9QWX7kq4RAwz7I3Iuyt6vdL+qbyiMb1mYf1Iy+oUwIDAQABo4HWMIHTMB0G
A1UdDgQWBBT2zaAoFZ5VmekJeHsaUbp9TVhYHzCBowYDVR0jBIGbMIGYgBT2zaAo
FZ5VmekJeHsaUbp9TVhYH6F9pHsweTELMAkGA1UEBhMCVVMxCzAJBgNVBAgTAlZB
MRIwEAYDVQQHEwlDaGFudGlsbHkxEzARBgNVBAoTClBsZXNrIEluYy4xEjAQBgNV
BAMTCXBsZXNrLmNvbTEgMB4GCSqGSIb3DQEJARYRc3VwcG9ydEBwbGVzay5jb22C
AQAwDAYDVR0TBAUwAwEB/zANBgkqhkiG9w0BAQQFAAOBgQAmHfypVB/5muH/sF0B
gAOH9MBe35xvu+JaYBucuCPZz2VTeVpg/6pdLavlXVE7LHPjDm8gM31vzeFvY5/J
Z54BcyQ0HIluPG2MMF2+BBYPEJ0ubl9BK/XHaNk2ff1FhbJPBbWjnwqR1sUyuVij
3Z2oznbWvsE7cpdYJJBYOw1c7Q==
-----END CERTIFICATE-----
subject=/C=US/ST=VA/L=Chantilly/O=Plesk Inc./CN=plesk.com/Email=support@plesk.com
issuer=/C=US/ST=VA/L=Chantilly/O=Plesk Inc./CN=plesk.com/Email=support@plesk.com

---
No client certificate CA names sent
---
SSL handshake has read 1391 bytes and written 314 bytes
---
New, TLSv1/SSLv3, Cipher is EDH-RSA-DES-CBC3-SHA
Server public key is 1024 bit
SSL-Session:
    Protocol  : TLSv1
```

```
    Cipher    : EDH-RSA-DES-CBC3-SHA
    Session-ID: 1835735F3757605E08D7648A87D5E567DDCC6F51D872D1BB0675B46903D2D19B

    Session-ID-ctx:
    Master-Key: D944BD06FF590185A103793CD9913B06818AFF5702314DDF7214A062D6B5BFAB
99C047068C82869BB9824EB1160BCFD0
    Key-Arg   : None
    Start Time: 1063823232
    Timeout   : 300 (sec)
    Verify return code: 10 (certificate has expired)
---
HTTP/1.1 200 OK
Date: Wed, 17 Sep 2003 18:26:23 GMT
Server: Apache/1.3.27 (Unix) mod_jk/1.2.0 Chili!Soft-ASP/3.6.2 mod_perl/1.26 mod
_throttle/3.1.2 PHP/4.3.1 FrontPage/4.0.4.3 mod_ssl/2.8.11 OpenSSL/0.9.6h
Last-Modified: Thu, 05 Dec 2002 21:11:56 GMT
ETag: "31c32-62f-3defc11c"
Accept-Ranges: bytes
Content-Length: 1583
Connection: close
Content-Type: text/html

<html>
<head>
<title>..::.. under construction ..::..</title>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1">
</head>

<body bgcolor="#666699" text="#666699">
<br>
…….
    </td>
  </tr>
</table>
</body>
</html>

closed

C:\openssl>
```

If you take the time to read the output, you can learn a lot about this https server. This may be very useful in troubleshooting ssl connections or assist in auditing the server.

2. Simulating an https server, to test browsers.

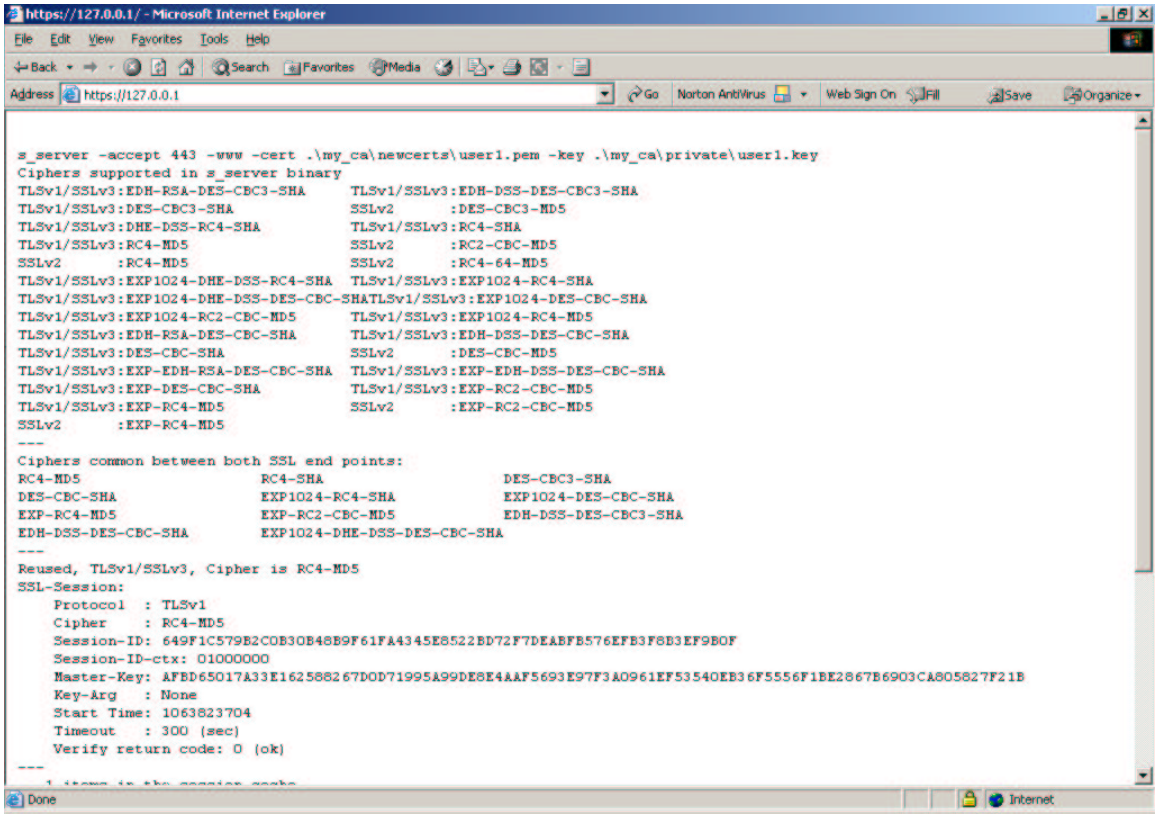
You can put openssl in server mode, specifying certificates to use, as well as encryption algorithms, etc… Nice to know is that if you run openssl on a MS machine running IIS in SSL mode, openssl has precedence on that IIS server.

```
C:\openssl>openssl s_server -accept 443 -www -cert .\my_ca\newcerts\user1.pem -key
.\my_ca\private\user1.key
Loading 'screen' into random state - done
Using default temp DH parameters
Enter PEM pass phrase:
ACCEPT
ACCEPT
ACCEPT
accept error 10004
   1 items in the session cache
   0 client connects (SSL_connect())
   0 client renegotiates (SSL_connect())
   0 client connects that finished
   2 server accepts (SSL_accept())
   0 server renegotiates (SSL_accept())
   2 server accepts that finished
   1 session cache hits
   0 session cache misses
```

```
   0 session cache timeouts
   0 callback cache hits
   0 cache full overflows (128 allowed)

C:\openssl>
```

That is what you see in your browser screen. This can be useful in analyzing what protocols and algorithms are actually supported by the browser.



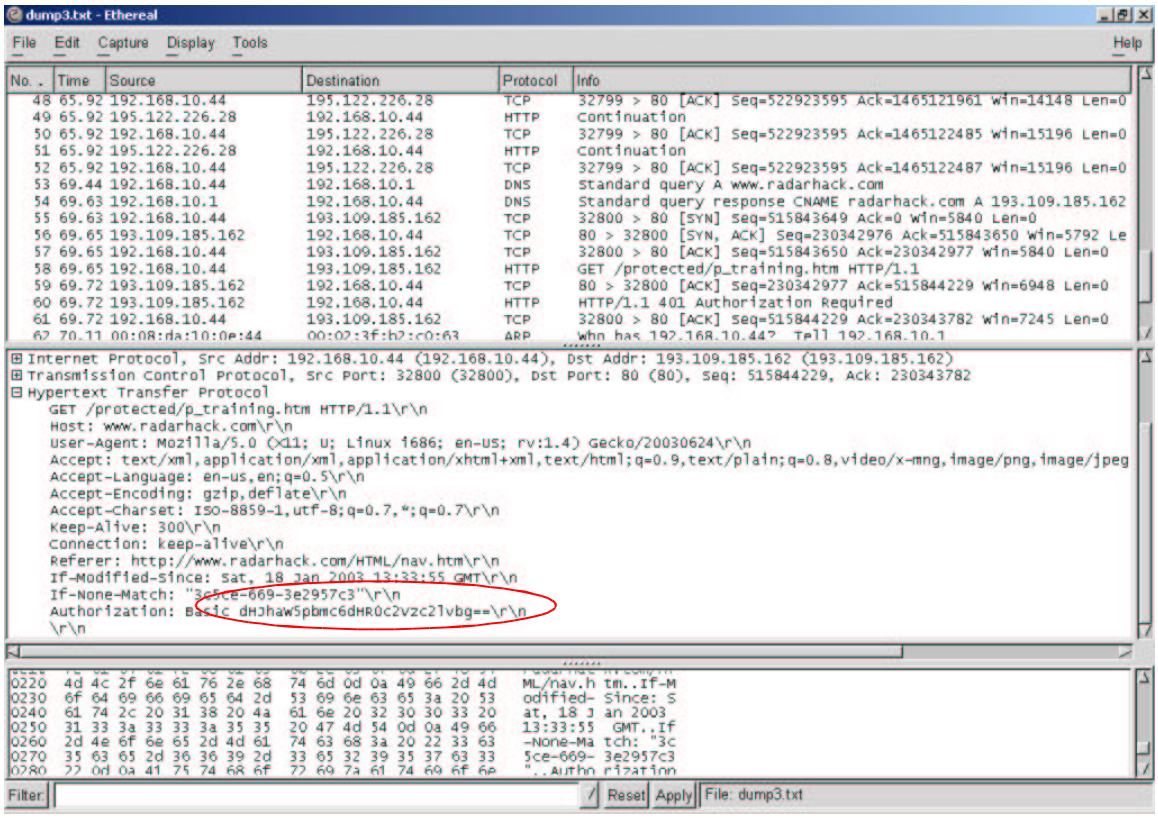
3. Calculating file digest with openssl.

```
C:\openssl>openssl dgst -md5 openssl.cnf   (or any other filename)
MD5(openssl.cnf)= 09e06d4951985b3ce56fc849d4a7d541

C:\openssl>openssl dgst -sha1 openssl.cnf
SHA1(openssl.cnf)= cdc68f70815fce03bf3d279ebf8299fafe6023c3
```

4. Decoding http BASIC AUTHENTICATION with openssl.

Basic authentication in http, actually sends the username and password in an insecure way. At first sight this seems encrypted, but in fact the user name and password are base64 encoded. This is NOT encryption, its simply data encoding.




```
C:\openssl>echo dHJhaW5pbmc6dHR0c2Vzc2lvbg== >base64.in

C:\openssl>openssl base64 -d  -in base64.in -out decoded.out

C:\openssl>type decoded.out
training:tttsession
C:\openssl>
```

5. Conclusion

Although the quite difficult command line, Openssl can actually be of great benefit in learning and auditing SSL servers and other ssl related stuff. The tool also permits to get familiar with cryptographic operations as hashing, a-symmetric and symmetric encryption.