Check email encryption process

To facilitate this analysis, it is good to 'talk' directly to your SMTP or IMAP server.

TipsMake.com - Analyze POP3, IMAP and SMTP protocols via SSL security mechanism

To facilitate this analysis, it is good to "talk" directly to your SMTP or IMAP server. But things get complicated when conducting end-to-end data encryption, but with the right tools, this won't be too difficult.

Typically, almost all mail server systems require a connection encryption mechanism. The following two methods are used - either all addresses sent via SSL or another mechanism called StartTLS will be used to activate the encryption process after receiving the connection request.

First, take a look at SSL services, which are often used with special, special requirements via TCP. The following is a reference to other important ports:

Service Abbreviation TCP portHTTP over SSLhttps443IMAP over SSLimaps993IRC over SSL994POP3 over SSLpop3s995SMTP over SSLssmtp465

The service will listen to requests from TCP ports, especially those directly over SSL, for example, which email client systems that do not support SSL will not be able to communicate with the IMAPS server via port 993. Once these Encrypted data and parameters have been implemented, they will be 'licensed' and create a tunnel - a separate tunnel through which data transfer is performed in practice. Based on the combinations and related components in the SSL connection, when any problem occurs, support tools such as telnet and netcat tend to shorten this process.

Next is a little test step with **OpenSSL** , including a small SSL client example that can be used to connect to SSL services like *https://www.heise.de* :

$ openssl s_client -host www.heise.de -port 443
CONNECTED (00000003)
[.]

Certificate chain
0 s: / C = DE / ST = Niedersachsen / L = Hannover / O = Heise Zeitschriften Verlag GmbH Co KG / OU = Netzwerkadministration / OU = Terms of use at www.verisign.com/rpa (c) 05 / CN = www .heise.de
i: / O = VeriSign Trust Network / OU = VeriSign, Inc./OU=VeriSign International Server CA - Class 3 / OU = www.verisign.com/CPS Incorp.by Ref. LIABILITY LTD. (C) 97 VeriSign
1 s: / O = VeriSign Trust Network / OU = VeriSign, Inc./OU=VeriSign International Server CA - Class 3 / OU = www.verisign.com/CPS Incorp.by Ref. LIABILITY LTD. (C) 97 VeriSign
i: / C = US / O = VeriSign, Inc./OU=Class 3 Public Primary Certification Authority

---
[.]

The above information is provided and authenticated by openssl, allowing us to check which other certificates are used. If you don't do that, it's like managers at the door and waiting for man-in-the-middle attacks. Technically, those who can use ettercap technology can simply get the admin password in a simple way.

The parameter encrypts and decrypts the SSL signal completely "invisible", so users can contact the server directly:

GET / HTTP / 1.1
Host: www.heise.de

HTTP / 1.1 302 Found
Date: Wed, 16 Sep 2009 10:24:44 GMT
Server: Apache / 1.3.34
Location: http://www.heise.de/
[.]

Log in to IMAPS

This process is only slightly more complicated:

$ openssl s_client -host imap.irgendwo.de -port 993
[.]
** OK IMAP4 Ready 0.0.0.0 0001f994*
1 Login user-ju secret
1 OK You are compared in
2 LIST "" ""*
** LIST (HasChildren) "." "INBOX"*
** LIST (HasNoChildren) "." "INBOX.AV"*
[.]
2 OK Completed (0.130 secs 5171 calls)
3 logout
** BYE LOGOUT received*
3 OK Completed

When you're done with this step, don't forget to rearrange the sequence numbers corresponding to the previous IMAP statement. For the same POP3 protocol, we must authenticate within the SSL 'tunnel' with the USER statement and PASS POP3:

$ openssl s_client -host pop.irgendwo.de -port 995
[.]
+ OK POP server ready H mimap3
USER user-ju
+ OK password required for user "user-ju"

PASS secret
+ OK mailbox "user-ju" has 0 messages (0 octets) H mimap3
thoát
+ OK POP server signing off

This can be considered as an appropriate alternative for telnet-ssl tool.

StartTLS

Internet service providers especially like to use SSL model, Transport Layer Security through StartTLS. This model has an advantage with many options while still allowing clients not to communicate with the server without encryption. The downside of this is that email clients need to interact directly with the server if they want to deny any TLS connection.

The default email client option is " *TLS, if available* " that comes with risk, man-in-the-middle attacks can 'gently' change the StartTLS statement - with the trigger feature too Encoder, into XtartTLS. Then, the server will respond that it does not execute the XtartTLS command, and cause the email client to send data in unencrypted form to an unknown form back to the user. Therefore, it is recommended to double-check that the server can handle the StartTLS command, and then enable this feature. If any error message is received, it is obvious that there is a problem somewhere in the system.

The ports that TLS service operates on depend on the vendor. In principle, these types of encryption can embed an 'invisible' way - transparent, into the system without requiring any action. To find out if the mail server system supports this feature:

$ nc smtp.irgendwo.de smtp
220 Mailserver ESMTP Exim 4.69 Wed, Sep 16, 2009 13:05:15 +0200
ehlo test
250-Mailserver Hello loki [10.1.2.73]
250-SIZE 78643200
250-PIPELINING
250-STARTTLS
250 HELP
thoát
221 Mailserver closing connection

This list should be accompanied by the StartTLS command, the main function is to activate the Transport Layer Security encryption process:

STARTTLS
220 TLS go ahead

At this point, Netcat will cause some confusing problems, but OpenSSL can fix this easily. Developers have created the SSL client system smart enough to require TLS encryption for SMTP, POP3, IMAP and FTP protocols, although they do not work with all servers:

$ openssl s_client -host mail.irgendwo.de -port 25 -starttls smtp
CONNECTED (00000003)
[.]
250 HELP
ehlo test
250-Mailserver Hello loki [10.1.2.73]
250-SIZE 52428800
250-PIPELINING
250-AUTH PLAIN LOGIN
250 HELP

SMTP authentication mechanism

Authentication in SMTP is a bit more complicated. For most servers, as in this example, supports the AUTH PLAIN method, where the data must be Base64 compliant. This process is handled by the following Pearl statement:

$ perl -MMIME :: Base64 -e 'print encode_base64 ("00user-ju00secret")'
AHVzZXItanUAc2VjcmV0

The results will have to match the request from the SMTP server:

AUTH PLAIN AHVzZXItanUAc2VjcmV0
235 Authentication succeeded

The received signals are ready for the next SMTP commands, for addresses and servers that do not support OpenSSL, users can use gnutls-cli available in the gnutls-bin package. First, it creates a cleartext connection to any TLS proprietary service such as:

$ gnutls-cli -s -p submission smtp.heise.de
Resolving 'smtp.heise.de' .
Connecting to '10 .1.2.41: 587 '.

- Simple Client Mode:

220 taxis03.heise.de ESMTP Exim 4.69 Wed, Sep 16, 2009 18:03:01 +0200
ehlo test
250-taxis03.heise.de Hello loki.ct.heise.de [10.10.22.75]
250-SIZE 78643200
250-PIPELINING
250-STARTTLS
250 HELP
starttls
220 TLS go ahead

Next, switch to the second statement to process the ID of the tools and send the SIGALARM signal directly there:

$ ps aux | grep gnutls
ju 6103 pts / 3 S + 18:03 0:00 gnutls-cli [.]
$ kill -s SIGALRM 6103

This will cause gnutls-cli to settle with TLS standard and automatically reconnect ***stdin*** and ***stdout*** parameters to create a new 'tunnel'. Also, there are some interesting information about the newly created TLS connection:

**** Starting TLS handshake*
- Certificate type: X.509
- Got a certificate list of 1 certificates.

- Certificate [0] info:
 # The hostname in the certificate 'smtp.heise.de'.
 # valid since: Thu Dec 14 14:08:41 CET 2006
 # expires at: Sun Dec 11 14:08:41 CET 2016
 # fingerprint: 28: 8C: E0: 29: B9: 31: 9B: 96: F6: 3D: B4: 49: 10: CD: 06: 80
 # Subject's DN: C = DE, ST = Niederachsen, L = Hannover, O = Heise Zeitschriften Verlag GmbH Co KG, OU = Netzwerkadministration, CN = smtp.heise.de, EMAIL = admin @ heise.de
 # Issuers DN: C = DE, ST = Nacherachsen, L = Hannover, O = Verlag Heinz Heise GmbH & Co. KG, OU = Netzwerkadministration, CN = admin @ heise.de, EMAIL = admin @ heise.de

- Peer's certificate issuer is unknown
- Peer's certificate is NOT trusted
- Version: TLS 1.0
- Key Exchange: DHE RSA
- Cipher: AES 256 CBC
- MAC: SHA
- Compression: NULL
thoát
221 taxis03.heise.de closing connection
- Peer has closed the GNUTLS connection

This allows users to connect directly to the service store library to enable TLS. If the user wants to test further, make sure OpenSSL supports ***s_server*** to be able to execute commands and send to the ***www*** server. The ***gnutls-serv*** feature also provides the same functionality in the ***gnutls-bin*** package.