



Upgrading POSIX email components to be TLS v1.2 compatible

By Joe Doupnik

jrd@netlab1.net

jdoupnik@microfocus.com

Mindworksuk and Micro Focus

What I wanted to accomplish

Bring current epoch TLS support into use on Apache, Postfix, Dovecot, and SSH. The generic name for this is support **TLS v1.2**

The crypto engine involved is **openssl v1.0.1**

Openssl v0.9.8 libraries can happily co-exist with those from v1.0.1. However, the header files in the -devel versions (needed for building things) do conflict. Thus choose one -devel, build, go back to the original.

I used the SUSE built Postfix and SSH from the **Security Modules** channel. Openssl 1.0.1 libraries were installed on the OES11 SP2 (and SLES 11 SP3).

Only Dovecot was recompiled to take advantage of openssl1 for TLS v1.2

Email handlers, ports, protocols

A short navigational aid to ports of call

SMTP Protocol
Handled by **Postfix**

Port 25 (SMTP)
none, or STARTTLS (no AUTH)
Receive & send

Port 587 (Submission)
TLS, or STARTTLS then AUTH
Receive-only

Port 427 (old SMTPS)
TLS, or STARTTLS then AUTH
Receive-only
Obsolete, avoid

IMAP4 protocol
Handled by **Dovecot**

Port 143 (IMAP)
AUTH, can STARTTLS

Port 993 (IMAPS)
TLS then AUTH

POP3 protocol
Handled by **Dovecot**

Port 110 (POP3)
AUTH

Port 995 (POP3S)
TLS then AUTH

TLS = forced encryption (aka SSL)
STARTTLS = negotiable encryption
AUTH = supply authentication credentials

Postfix uses the SASL mechanism suite for AUTH support

SASL = Simple Authentication and Security Layer

Adds authentication to connection based protocols

<https://www.iana.org/assignments/sasl-mechanisms/sasl-mechanisms.xhtml>

Some of the popular applications using SASL

- ACAP
- AMQP
- BEEP
- IMAP
- IMSP
- IRC (with IRCX, TS6 or the IRCv3 SASL extension^[7])
- LDAP
- libvirt
- ManageSieve (RFC 5804^[8])
- memcached
- POP
- RFB protocol^[7] used by VNC
- SMTP
- Subversion's "svn" protocol
- XMPP

https://en.wikipedia.org/wiki/Simple_Authentication_and_Security_Layer

New and regular Postfix, SLES11 SP3

The screenshot shows the YaST2 package manager interface. The search results table lists several postfix-related packages. The 'postfix-openssl1' and 'postfix-openssl1-doc' packages are highlighted with a red dashed box and labeled 'New'. The 'postfix' and 'postfix-devel' packages are highlighted with a black dashed box and labeled 'Old'.

Package	Summary	Installed (Avail)	Size
<input checked="" type="checkbox"/> postfix-openssl1	A fast, secure, and flexible mailer	2.9.4-0.23.2	2.7 MiB
<input checked="" type="checkbox"/> postfix-openssl1-doc	Documentations for the postfix package	2.9.4-0.23.2	4.7 MiB
<input type="checkbox"/> postfix	A fast, secure, and flexible mailer	(2.9.4-0.23.2)	2.7 MiB
<input type="checkbox"/> postfix-devel	Development headers for the postfix package	(2.9.4-0.23.2)	475.0 KiB
<input type="checkbox"/> postfix-doc	Documentations for the postfix package	(2.9.4-0.23.2)	4.7 MiB
<input type="checkbox"/> postfix-mysql	Postfix plugin to support MySQL maps	(2.9.4-0.23.2)	15.0 KiB
<input type="checkbox"/> postfix-openssl1-devel	Development headers for the postfix package	(2.9.4-0.23.2)	475.0 KiB
<input type="checkbox"/> postfix-openssl1-mysql	Postfix plugin to support MySQL maps	(2.9.4-0.23.2)	15.0 KiB
<input type="checkbox"/> postfix-openssl1-postgresql	Postfix plugin to support PostgreSQL maps	(2.9.4-0.23.2)	19.0 KiB

Postfix-openssl1 is from the **SLE11 Security Modules** repository

Before you install this make a copy of /etc/postfix

Install libopenssl1 (omit -devel)

Package	Summary	Installed (Availa	Size
<input checked="" type="checkbox"/> cyrus-sasl-openssl1	Implementation of Cyrus SASL API	2.1.22-0.27.6	1.1 MiB
<input checked="" type="checkbox"/> cyrus-sasl-openssl1-digestmd5	cyrus-sasl plugin for the DIGESTMD...	2.1.22-0.27.6	52.0 KiB
<input checked="" type="checkbox"/> cyrus-sasl-openssl1-gssapi	cyrus-sasl plugin for the GSSAPI me...	2.1.22-0.27.6	31.0 KiB
<input checked="" type="checkbox"/> cyrus-sasl-openssl1-plain	cyrus-sasl plugin for the PLAIN mec...	2.1.22-0.27.6	19.0 KiB
<input checked="" type="checkbox"/> libldap-openssl1-2_4-2	OpenLDAP Client Libraries for Open...	2.4.26-0.62.3	667.0 KiB
<input checked="" type="checkbox"/> libopenssl-devel	Include Files and Libraries mandator...	0.9.8j-0.80.1	20.3 MiB
<input checked="" type="checkbox"/> libopenssl0_9_8	Secure Sockets and Transport Layer...	0.9.8j-0.80.1	2.1 MiB
<input checked="" type="checkbox"/> libopenssl0_9_8-32bit	Secure Sockets and Transport Layer...	0.9.8j-0.80.1	2.0 MiB
<input checked="" type="checkbox"/> libopenssl1_0_0	Secure Sockets and Transport Layer...	1.0.1g-0.35.1	2.5 MiB
<input checked="" type="checkbox"/> openldap2-client	The OpenLDAP Client	2.4.26-0.62.2	560.0 KiB
<input checked="" type="checkbox"/> openssh-openssl1	Secure Shell Client and Server (Rem...	6.6p1-10.1	5.0 MiB
<input checked="" type="checkbox"/> openssh-openssl1-helpers	OpenSSH AuthorizedKeysCommand...	6.6p1-10.1	66.0 KiB
<input checked="" type="checkbox"/> openssl	Secure Sockets and Transport Layer...	0.9.8j-0.80.1	1.1 MiB
<input checked="" type="checkbox"/> openssl-certs	CA certificates for OpenSSL	1.97-0.3.1	244.0 KiB
<input checked="" type="checkbox"/> openssl1	Secure Sockets and Transport Layer...	1.0.1g-0.35.1	1.3 MiB
<input checked="" type="checkbox"/> perl-Crypt-OpenSSL-RSA	RSA encoding and decoding, using t...	0.25-3.14	83.0 KiB
<input checked="" type="checkbox"/> perl-Net-SSLeay	Perl extension for using OpenSSL	1.64-0.3.1	1.3 MiB
<input checked="" type="checkbox"/> php53-openssl	PHP5 Extension Module	5.3.17-48.1	122.0 KiB
<input checked="" type="checkbox"/> postfix-openssl1	A fast, secure, and flexible mailer	2.9.4-0.23.2	2.7 MiB
<input checked="" type="checkbox"/> postfix-openssl1-doc	Documentations for the postfix pack...	2.9.4-0.23.2	4.7 MiB
<input type="checkbox"/> cyrus-sasl-openssl1-32bit	Implementation of Cyrus SASL API	(2.1.22-0.27.6)	156.0 KiB
<input type="checkbox"/> cyrus-sasl-openssl1-crammd5	cyrus-sasl plugin for the CRAMMD5 ...	(2.1.22-0.27.6)	23.0 KiB

Configuration: the big picture

File `/etc/postfix/main.cf` has settings for the entire MTA

File `/etc/postfix/master.cf` lists communication connectors

Each connector can have `(-o foo)` overrides of setting inherited from `main.cf` and can add other settings

What commands to issue, and where?

I grouped them in `main.cf` and placed `-o` overrides in `master.cf`.

Goals in more detail



SMTP port 25, no-AUTH, **optionally STARTTLS** if other side agrees
SMTP port 587, STARTTLS+AUTH to submit new mail

IMAP4 port 143 as plain text AUTH or STARTTLS+AUTH

IMAP4 port 993 as only TLS+AUTH

Postfix does the mail heavy lifting. Dovecot provides IMAP4/POP3 and authentication by working with Postfix

A major problem was decoding Postfix commands. There are hundreds of them, plus crypto aspects.

Thus I needed to domesticate this complexity

Postfix consolidated controls in main.cf, 1/3

...

smtpd_sender_restrictions = hash:/etc/postfix/access

smtpd_client_restrictions =

smtpd_helo_required = yes

smtpd_helo_restrictions =

strict_rfc821_envelopes = no

smtpd_recipient_restrictions =

 permit_mynetworks,

 permit_sasl_authenticated,

 reject_unauth_destination,

 reject_unauth_pipelining,

 reject_rbl_client sbl.spamhaus.org,

 reject_rbl_client cbl.abuseat.org,

 reject_rbl_client bl.spamcop.net,

permit_tls_clientcerts

permit_mx_backup

Permit local (IP) clients to send anywhere

Permit SASL auth'd clients to send anywhere

Do not be an open relay

Bad guy blockers

Postfix consolidated controls in main.cf, 2/3

Begin **TLS** and **SASL** section

Outgoing, acting as a client (smtp_)

smtp_sasl_tls_security_options = noanonymous

smtp_sasl_password_maps=hash:/etc/postfix/sasl_passwd

**No empty credentials for AUTH
p/w for authenticated relays**

Incoming, acting as a server (smtpd_)

smtpd_sasl_type = dovecot

smtpd_sasl_path = private/auth

smtpd_sasl_security_options = noanonymous

broken_sasl_auth_clients = yes

smtpd_tls_auth_only, yes if AUTH offered only after STARTTLS has been done

smtpd_tls_auth_only = yes

Dovecot provides authentication, uses PAM

No empty credentials for AUTH

Send AUTH= as well as AUTH

important (protect login credentials)

Testing, loglevel and received_header

smtp_tls_loglevel = 0

smtpd_tls_loglevel = 0

smtpd_tls_received_header = no

yes adds email header about use of TLS

End **SASL** section

Postfix consolidated controls in main.cf, 3/3

Start **cipher suite** section

preempt_cipherlist, yes=Postfix chooses here, no=openssl chooses

tls_preempt_cipherlist = yes

important (control crypto here)

for outgoing traffic, use smtp_tls_

smtp_tls_security_level = may

smtp_tls_protocols = !SSLv2, !SSLv3

smtp_tls_mandatory_protocols = !SSLv2, !SSLv3

smtp_tls_mandatory_ciphers = strong

**may = Opportunistic STARTTLS
to work with both plain and TLS
connections.**

for incoming traffic, use smtpd_tls_

smtpd_tls_security_level = may

smtpd_tls_protocols = !SSLv2, !SSLv3

smtpd_tls_mandatory_protocols = !SSLv2, !SSLv3

smtpd_tls_mandatory_ciphers = medium

medium is to be considerate

smtpd_tls_cert_file = /etc/ssl/servercerts/123-certs/triple.pem

smtpd_tls_key_file = /etc/ssl/servercerts/123-certs/private-key.pem

smtpd_tls_CApath = /etc/ssl/certs

“tls_protocols” for level=may, “tls_mandatory_protocols” for level=encrypt

End **cipher suite** and **TLS** sections

Composite certificate stack, in order

Many applications requiring certificate files ask for only the **server** and **key** files, totally ignoring the common set of **intermediary CA** certs

To get round this we become clever & devious:

Construct a pseudo server cert file which is the concatenation of the original server cert, intermediary CAs and optionally the final CA.

Usually the final CA cert can be omitted because the client's trusted cert store has it. Need to include the CA if using self-signed certs.

Example:

```
cat servercert.pem intermediate_doman_ca.pem AlphaSSL-rootCA.pem >  
triple.pem
```

Postfix listener details, partial, 1/2

/etc/postfix/master.cf, top of file
 smtps is commented out

```

#=====
# service type private unpriv chroot wakeup maxproc command + args
#          (yes) (yes) (yes) (never) (100)
#
Port #
=====
25  smtp      inet n    -    n    -    200  smtpd
    -o content_filter=smtp:[127.0.0.1]:10024
427 #smtps     inet n    -    n    -    200  smtpd
    # -o content_filter=smtp:[127.0.0.1]:10024
    # -o smtpd_tls_wrappermode=yes
    # -o smtpd_sasl_auth_enable=yes
    # -o smtpd_etrn_restrictions=reject
587 submission inet n    -    n    -    -    smtpd
    -o content_filter=smtp:[127.0.0.1]:10024
    -o smtpd_tls_security_level=encrypt
    -o smtpd_sasl_auth_enable=yes
    -o smtpd_client_restrictions=permit_sasl_authenticated, reject
    -o smtpd_etrn_restrictions=reject
  
```

Yes forces TLS, no STARTTLS
 “smtps” port 427 is deprecated, avoid.

Use 10026 to add **DKIM** header
 Require STARTTLS to be done
 After that offer AUTH facility
 Allow only TLS auth'd clients

Content_filter is amavisd on localhost ports 1002x

Postfix listener details, partial, 2/2

/etc/postfix/master.cf further down the file
 amavis and dovecot service entries added manually

```
localhost:10025 inet  n    -    n    -    -    smtpd
```

```
    -o content_filter=
```

```
amavis  unix  -    n    n    -    2    smtpd
```

```
## Dovecot LDA support for replying to Return-Path: <MAILER_DAEMON>
```

```
dovecot unix  -    n    n    -    -    pipe flags=DRhu user=vmail:vmail null_sender=
argv=/usr/local/libexec/dovecot/deliver -f ${sender} -d ${user}@${nexthop}
```

 Linkage to amavisd is complicated –

to localhost:10024 (no DKIM additon) or 10026 (DKIM to be added)

See

https://wiki.gentoo.org/wiki/Complete_Virtual_Mail_Server/amvisd_spamassassin_clamav

and <http://shisaa.jp/postset/mailserver-3.html>

for useful postfix/amavisd/clamd configuration details. Fewer “-o”s is often better than many.

DKIM support (/etc/amavisd.conf)

```
$enable_dkim_verification = 1; # enable DKIM signatures verification
$enable_dkim_signing = 1; # load DKIM signing code, keys defined by dkim_key
## ADD DKIM key for example.com, selector named "mail"
dkim_key("example.com", "mail", "/var/db/dkim/example.key.pem");
@dkim_signature_options_bysender_maps = (
  ( '.' => ( ttl => 21*24*3600, c => 'relaxed/simple' ) ));
```

```
@local_domains_maps = ( [ ".$mydomain" ] ); # list of all local domains
@mynetworks = qw( 127.0.0.0/8 [::1] [FE80::]/10 [FEC0::]/10
  10.0.0.0/8 11.22.33.44/24 192.168.0.0/16 );
```

```
##$inet_socket_port = 10024; # listen on this local TCP port(s)
$inet_socket_port = [10024,10026]; # listen on multiple TCP ports
```

The DNS TXT record becomes

mail._domainkey.example.com IN TXT <gibberish from "amavisd showkeys">
where "mail" is the "selector" and ._domainkey. is a required field

See <https://www.ijs.si/software/amavisd/amavisd-new-docs.html#dkim>

DKIM support (/etc/amavisd.conf)

it is up to MTA to re-route mail from authenticated roaming users or
from internal hosts to a dedicated TCP port (such as 10026) for filtering

```
$interface_policy{'10026'} = 'ORIGINATING';
$policy_bank{'ORIGINATING'} = { # mail supposedly originating from our users
    originating => 1, # declare that mail was submitted by our smtp client
    allow_disclaimers => 1, # enables disclaimer insertion if available
    # notify administrator of locally originating malware
    virus_admin_maps => ["virusalert\@$mydomain"],
    spam_admin_maps => ["virusalert\@$mydomain"],
    warnbadhsender => 1,
    # forward to a smtpd service providing DKIM signing service
    ##### forward_method => 'smtp:[127.0.0.1]:10027',
    # force MTA conversion to 7-bit (e.g. before DKIM signing)
    smtpd_discard_ehlo_keywords => ['8BITMIME'],
    bypass_banned_checks_maps => [1], # allow sending any file names and types
    terminate_dsn_on_notify_success => 0, # don't remove NOTIFY=SUCCESS option
};
```

Come here to add DKIM header

Omit, nothing there

Dovecot listener configuration

10-ssl.conf

Dovecot version 2.2.21

```
# SSL/TLS support: yes, no, required.  
ssl = yes
```

```
# SSL ciphers to use
```

```
ssl_cipher_list = ALL:!LOW:!SSLv2:!EXP:!aNULL
```

Allowed crypto algorithms

10-master.cf

```
service imap-login {  
  inet_listener imap {  
    port = 143  
  }  
  inet_listener imaps {  
    port = 993  
    ssl = yes  
  }  
}
```

```
service pop3-login {  
  inet_listener pop3 {  
    port = 110  
  }  
  inet_listener pop3s {  
    port = 995  
    ssl = yes  
  }  
}
```

Dovecot listener, cont'd

10-auth.conf

Disable LOGIN command and all other plaintext authentications unless
SSL/TLS is used (LOGINDISABLED capability). Note that if the remote IP
matches the local IP (ie. you're connecting from the same computer), the
connection is considered secure and plaintext authentication is allowed.

###disable_plaintext_auth = yes

disable_plaintext_auth = no

no allows IMAP4/POP3 with plain text

20-imap.conf

#mail_max_userip_connections = 10

#JRD increase for Macs with many accounts open simultaneously

mail_max_userip_connections = 40

ssl_cert = </etc/ssl/servercerts/123-certs/servercert.pem

ssl_key = </etc/ssl/servercerts/123-certs/private-key.pem

(above, note the required < before the filename)

Encrypt or not, require login or not, and when

Three commands control these Postfix server features:

smtpd_tls_security_level=encrypt, may, none

smtpd_sasl_auth_enable=yes, no

smtpd_tls_auth_only=yes, no

Following are Manual descriptions of each and a table showing what happens when we use the above values. Delivery is not tested here.

smtpd_tls_security_level (default: empty)

The SMTP TLS security level for the Postfix SMTP server; when a non-empty value is specified, this overrides the obsolete parameters [smtpd_use_tls](#) and [smtpd_enforce_tls](#). This parameter is ignored with "[smtpd_tls_wrappermode](#) = yes".

Specify one of the following security levels:

none

TLS will not be used.

may

Opportunistic TLS: announce STARTTLS support to remote SMTP clients, but do not require that clients use TLS encryption.

encrypt

Mandatory TLS encryption: announce STARTTLS support to remote SMTP clients, and require that clients use TLS encryption. According to [RFC 2487](#) this MUST NOT be applied in case of a publicly-referenced SMTP server. Instead, this option should be used only on dedicated servers.

smtpd_sasl_auth_enable (default: no)

Enable SASL authentication in the Postfix SMTP server. By default, the Postfix SMTP server does not use authentication.

If a remote SMTP client is authenticated, the [permit_sasl_authenticated](#) access restriction can be used to permit relay access, like this:

```
# specified only under smtpd_recipient_restrictions.  
smtpd_recipient_restrictions =  
    permit_mynetworks, permit_sasl_authenticated, ...
```

To reject all SMTP connections from unauthenticated clients, specify "smtpd_delay_reject = yes" (which is the default) and use:

```
smtpd_client_restrictions = permit_sasl_authenticated, reject
```

Note: saslauthd can perform various methods to support plain text AUTH logins
In this work Dovecot performs that role instead

smtpd_tls_auth_only (default: no)

When TLS encryption is optional in the Postfix SMTP server, do not announce or accept SASL authentication over unencrypted connections.

“Optional” means when Opportunistic TLS is used.
Authenticated means offer AUTH

Test smtpd_tls_security_level, smtpd_sasl_auth_enable, smtpd_tls_auth_only

Level	Enable	Only	Telnet localhost 587 shows	
-----	-----	-----	-----	
none	no	n/a	no STARTTLS, no AUTH	Crypto = security_level AUTH = auth_enable
none	yes	n/a	no STARTTLS, AUTH	
may	no	n/a	STARTTLS, no AUTH	
may	yes	no	STARTTLS, AUTH, can login plain before or after STARTTLS	
may	yes	yes	STARTTLS, <u>AUTH only after STARTTLS</u>	
encrypt	n/a	n/a	STARTTLS, no AUTH	

(below uses "openssl s_client -connect localhost:587 -starttls smtp")

encrypt	no	n/a	no AUTH
encrypt	yes	n/a	AUTH

n/a = not applicable, no effect

In all cases ehlo works, we can at least enter SMTP commands

Port 587 offer STARTTLS, and insists upon it

```
submission inet n - n - - smtpd
-o content_filter=smtp:[127.0.0.1]:10024
-o smtpd_tls_security_level=encrypt
-o smtpd_sasl_auth_enable=yes
-o smtpd_client_restrictions=permit_sasl_authenticated,reject
-o smtpd_etrn_restrictions=reject
```

To see the AUTH item for authentication the caller must first do the STARTTLS dance (the “encrypt” insistence).

This shields credentials during a login.

At right: STARTTLS, but no AUTH yet

telnet netlab1.net 587

```
Trying 82.70.37.210...
Connected to netlab1.net.
Escape character is '^]'.
220 netlab1.net ESMTP Postfix
ehlo me@here
250-netlab1.net
250-PIPELINING
250-SIZE 102400000
250-VRFY
250-ETRN
250-STARTTLS
250-ENHANCEDSTATUSCODES
250-8BITMIME
250 DSN
quit
221 2.0.0 Bye
Connection closed by foreign host.
```


Crypto test into Submission, port 587

```
# openssl s_client -connect netlab1.net:587 -starttls smtp
```

This test does the STARTTLS part and shows what follows

Much certificate stuff up here, then finally some readable items. The by-hand EHLO shows the AUTH ability after the crypto stuff.

SSL handshake has read 3200 bytes and written 438 bytes

New, TLSv1/SSLv3, Cipher is ECDHE-RSA-AES256-GCM-SHA384

Server public key is 2048 bit

Secure Renegotiation IS supported

Compression: NONE

Expansion: NONE

SSL-Session:

Protocol : TLSv1.2

Cipher : ECDHE-RSA-AES256-GCM-SHA384

...

250 DSN

continues in the column to the right ->

ehlo me@here

250-netlab1.net

250-PIPELINING

250-SIZE 102400000

250-VRIFY

250-ETRN

250-AUTH PLAIN LOGIN

250-AUTH=PLAIN LOGIN

250-ENHANCEDSTATUSCODES

250-8BITMIME

250 DSN

quit

221 2.0.0 Bye

closed

Testing SMTP TLS from the outside with new Postfix

<http://checktls.com/>

TestReceiver

CheckTLS Confidence Factor for "jrd@netlabtest.netlab1.net": 100

MX Server	Pref	Connect	Allowed	Can Use	TLS Adv	Cert OK	TLS Neg	Sndr OK	Rcvr OK
netlabtest.netlab1.net [82.70.37.215]	0	OK (121ms)	OK (148ms)	OK (119ms)	OK (119ms)	OK (387ms)	OK (121ms)	OK (132ms)	OK (203ms)
Average		100%	100%	100%	100%	100%	100%	100%	100%

(double click matrix to select all for copy and paste)

All OK, using TLS v1.2

Prior to change got all OK
but only TLS v1.0

Trying TLS on netlabtest.netlab1.net[82.70.37.215] (o):

```
seconds    test stage and result
[000.121]   Connected to server
[000.268]<--220 netlabtest.netlab1.net ESMTP Postfix
[000.269]   We are allowed to connect
[000.269]-->EHLO checktls.com
[000.388]<--250-netlabtest.netlab1.net
          250-PIPELINING
          250-SIZE 102400000
          250-VRIFY
          250-ETRN
          250-STARTTLS
          250-ENHANCEDSTATUSCODES
          250-8BITMIME
          250 DSN
[000.388]   We can use this server
[000.388]   TLS is an option on this server
[000.388]-->STARTTLS
[000.507]<--220 2.0.0 Ready to start TLS
[000.507]   STARTTLS command works on this server
[000.767]   SSLVersion in use: TLSv1.2
[000.767]   Cipher in use: ECDHE-RSA-AES128-SHA256
[000.767]   Connection converted to SSL
```

Check that the system is working correctly

Dovecot short tests:

<http://wiki.dovecot.org/TestInstallation>

Web based testing tool:

<https://ssl-tools.net/>

Read about Postfix+Dovecot SSL configuration:

<https://blog.tinned-software.net/harden-the-ssl-configuration-of-your-mailserver>

Local interesting general SSL testing script:

<http://testssl.sh> plenty of instructions on that web page, plus the script to download

Telnet and openssl cmd line tests

SMTP

```
telnet host 25  
ehlo myself  
quit
```

IMAP4

```
telnet host 143  
10 login user password  
20 select inbox  
30 logout
```

POP3

```
telnet host 110  
capa  
quit
```

SMTP with STARTTLS

```
openssl1 s_client -connect host:port -starttls smtp  
ehlo myself  
auth plain  
then the base64 SMTP SASL Plain Login string  
quit
```

IMAP4 with STARTTLS

```
openssl1 s_client -connect host:port -starttls imap  
10 login user password  
20 select inbox  
30 logout
```

SMTP SASL Plain Logins require base64 encoding:
echo -en "\0username\0password" | base64

Look for STARTTLS and AUTH in the ehlo response text

See <http://stackoverflow.com/questions/7192130/how-to-connect-imap-using-authenticate-plain-correctly>
(Hidden slide)

Email check, improved Postfix

<https://ssl-tools.net/>

Summary

Report created **Fri, 29 Jan 2016 10:36:36 +0000**

JSON

Refresh

Certificates ?



Trustworthy

Protocol



Secure

DANE ?



Missing

The mailservers of netlab1.net can be reached through a secure connection.

Servers

Incoming Mails

These servers are responsible for incoming mails to **@netlab1.net** addresses.

Hostname / IP address	Priority	STARTTLS	Certificates	Protocol			
netlab1.net 82.70.37.210	0	supported ✓	*.netlab1.net ✓	DANE ?	? missing	TLSv1.2	6 minutes ago
				PFS ?	✓ supported	TLSv1.1	11.0 s
				Heartbleed ?	✓ not vulnerable	TLSv1.0	
				Weak ciphers	✓ not found		

Postfix standard (top) vs improved (below)

<https://ssl-tools.net/>

Two MX records, bottom is a backup machine

Incoming Mails

These servers are responsible for incoming mails to @mindworksuk.com addresses.

Hostname / IP address	Priority	STARTTLS	Certificates	Protocol	TLS
mindworksuk.com 132.199.249.54	10	supported ✓	pa2.mindworksuk.com ⚠	DANE ? PFS ? Heartbleed ? Weak ciphers ? missing ✓ supported ✓ not vulnerable ⚠ supported • RSA_WITH_RC4_128_SHA • RSA_EXPORT_WITH_RC4_40_MD5	TLSv1.0 SSLv3 ⚠
netlab1.net 82.70.37.210	15	supported ✓	*.netlab1.net ✓	DANE ? PFS ? Heartbleed ? Weak ciphers ? missing ✓ supported ✓ not vulnerable ✓ not found	TLSv1.2 TLSv1.1 TLSv1.0

Standard Postfix

Postfix-openssl1

Red stuff

TLS

After improvements (still self-signed cert)

<https://ssl-tools.net/>

Incoming Mails

These servers are responsible for incoming mails to @mindworksuk.com addresses.

Hostname / IP address	Priority	STARTTLS	Certificates	Protocol
mindworksuk.com 132.199.249.54	10	supported ✓	pa2.mindworksuk.com ⚠	DANE ? ? missing TLSv1.2 PFS ? ✓ supported TLSv1.1 Heartbleed ? ✓ not TLSv1.0 vulnerable Weak ciphers ✓ not found
netlab1.net 82.70.37.210	15	supported ✓	*.netlab1.net ✓	DANE ? ? missing TLSv1.2 PFS ? ✓ supported TLSv1.1 Heartbleed ? ✓ not TLSv1.0 vulnerable Weak ciphers ✓ not found

That DANE thing



https://en.wikipedia.org/wiki/DNS-based_Authentication_of_Named_Entities

DNS-based Authentication of Named Entities (DANE) is a protocol to allow [X.509](#) certificates, commonly used for [Transport Layer Security](#) (TLS), to be bound to [DNS](#) names using Domain Name System Security Extensions ([DNSSEC](#)).^[1]

It is proposed in [RFC 6698](#) as a way to authenticate TLS client and server entities without a certificate authority ([CA](#)). It is updated with operational and deployment guidance in [RFC 7671](#). Application specific usage of DANE is defined in [RFC 7672](#) for SMTP and [RFC 7673](#) for using DANE with [Service \(SRV\) records](#).

DANE attempts to control per-hop usage of TLS for SMTP (mail is store & forward) via DNS records. Learn about it but think about the consequences.

DKIM adds hash of the message and a DNS pointer to the check key. This verifies message integrity and the DNS pointer step verifies the sender.



Now we finish the job

We have added TLS 1.2 support, which is good
Next we need to remove the weak ciphers

A useful tool for removals investigation is:

`http://testssl.sh` fetch text file testssl.sh (170KB)

Edit to use openssl1 rather than openssl (0.9.8) and name the result testssl1.sh (so that we have both versions)

That is just adding “1” to name openssl in the script. Easy.

Or use as `testssl.sh --openssl /usr/bin/openssl1 blah blah`

`chmod a+x testssl1.sh`

./testssl1.sh -t smtp myhost:25

Service set: STARTTLS via SMTP

Before cipher suite changes

--> Testing protocols (via openssl, SSLv2 via sockets)

```
SSLv2      not offered (OK)
SSLv3      not offered (OK)
TLS 1      offered
TLS 1.1    offered
TLS 1.2    offered (OK)
SPDY/NPN   (SPDY is a HTTP protocol and thus not tested here)
```

--> Testing ~standard cipher lists

```
Null Ciphers      not offered (OK)
Anonymous NULL Ciphers offered (NOT ok)
Anonymous DH Ciphers offered (NOT ok)
40 Bit encryption offered (NOT ok)
56 Bit encryption Local problem: No 56 Bit encryption configured in /usr/bin/openssl1
Export Ciphers (general) offered (NOT ok)
Low (<=64 Bit)    offered (NOT ok)
DES Ciphers       offered (NOT ok)
Medium grade encryption offered (NOT ok)
Triple DES Ciphers offered (NOT ok)
High grade encryption offered (OK)
```

--> Testing (perfect) forward secrecy, (P)FS -- omitting 3DES, RC4 and Null Encryption here



Not OK: No ciphers supporting Forward Secrecy offered

--> Testing server preferences

```
Has server cipher order?  yes (OK)
Negotiated protocol      TLSv1.2
Negotiated cipher        ECDHE-RSA-AES256-GCM-SHA384
Cipher order
```

```
TLSv1:  ECDHE-RSA-AES256-SHA AECDH-AES256-SHA ADH-AES256-SHA ADH-CAMELLIA256-SHA DHE-RSA-A
```

Postfix after cipher suite changes

Service set: STARTTLS via SMTP

After cipher suite changes

--> Testing protocols (via openssl, SSLv2 via sockets)

```
SSLv2      not offered (OK)
SSLv3      not offered (OK)
TLS 1      offered
TLS 1.1    offered
TLS 1.2    offered (OK)
SPDY/NPN   (SPDY is a HTTP protocol and thus not tested here)
```

--> Testing ~standard cipher lists

```
Null Ciphers      not offered (OK)
Anonymous NULL Ciphers not offered (OK)
Anonymous DH Ciphers not offered (OK)
40 Bit encryption not offered (OK)
56 Bit encryption Local problem: No 56 Bit encryption configured in /usr/bin/openssl
Export Ciphers (general) not offered (OK)
Low (<=64 Bit)    not offered (OK)
DES Ciphers       not offered (OK)
Medium grade encryption offered (NOT ok)
Triple DES Ciphers not offered (OK)
High grade encryption offered (OK)
```

--> Testing (perfect) forward secrecy, (P)FS -- omitting 3DES, RC4 and Null Encryption here

✓ PFS is offered (OK) ECDHE-RSA-AES256-GCM-SHA384 DHE-RSA-AES256-GCM-SHA384 DHE-RSA-AES256-SHA256 DHE-RSA-AES128-GCM-SHA256 ECDHE-RSA-AES128-SHA256 DHE-RSA-AES128-GCM-SHA256 DHE-RSA-AES128-SHA256 DHE-RSA-AES128-SHA

--> Testing server preferences

```
Has server cipher order? yes (OK)
Negotiated protocol      TLSv1.2
Negotiated cipher        ECDHE-RSA-AES256-GCM-SHA384
Cipher order
```

```
TLSv1: ECDHE-RSA-AES256-SHA DHE-RSA-AES256-SHA DHE-RSA-CAMELLIA256-SHA AES256-SHA CAMELLIA256-SHA
SSLv2: ECDHE-RSA-AES128-SHA DHE-RSA-AES128-SHA DHE-RSA-CAMELLIA128-SHA AES128-SHA CAMELLIA128-SHA
```

Postfix cipher suite changes, cont'd

--> Testing vulnerabilities

Before

```
Heartbleed (CVE-2014-0160)      not vulnerable (OK) (timed out)
CCS (CVE-2014-0224)            not vulnerable (OK)
Secure Renegotiation (CVE-2009-3555) not vulnerable (OK)
Secure Client-Initiated Renegotiation VULNERABLE (NOT ok), DoS threat
CRIME, TLS (CVE-2012-4929)      not vulnerable (OK) (not using HTTP anyway)
POODLE, SSL (CVE-2014-3566)     not vulnerable (OK)
TLS_FALLBACK_SCSV (RFC 7507), experim. Downgrade attack prevention supported (OK)
FREAK (CVE-2015-0204)          VULNERABLE (NOT ok), uses EXPORT RSA ciphers
LOGJAM (CVE-2015-4000), experimental VULNERABLE (NOT ok), uses DHE EXPORT ciphers, common primes not checked
BEAST (CVE-2011-3389)          TLS1: ECDHE-RSA-DES-CBC3-SHA EDH-RSA-DES-CBC3-SHA
                                AECDH-DES-CBC3-SHA ADH-DES-CBC3-SHA DES-CBC3-SHA
                                EDH-RSA-DES-CBC-SHA ADH-DES-CBC-SHA DES-CBC-SHA
                                EXP-EDH-RSA-DES-CBC-SHA EXP-ADH-DES-CBC-SHA EXP-DES-CBC-SHA EXP-ADH-DES-CBC-SHA
                                -- but also supports higher protocols (possible mitigation): TLSv1.1
```

Details above tell us which algorithms need to be removed

--> Testing vulnerabilities

After

```
Heartbleed (CVE-2014-0160)      not vulnerable (OK) (timed out)
CCS (CVE-2014-0224)            not vulnerable (OK)
Secure Renegotiation (CVE-2009-3555) not vulnerable (OK)
Secure Client-Initiated Renegotiation VULNERABLE (NOT ok), DoS threat
CRIME, TLS (CVE-2012-4929)      not vulnerable (OK) (not using HTTP anyway)
POODLE, SSL (CVE-2014-3566)     not vulnerable (OK)
TLS_FALLBACK_SCSV (RFC 7507), experim. Downgrade attack prevention supported (OK)
FREAK (CVE-2015-0204)          not vulnerable (OK) (tested with 6/9 ciphers)
LOGJAM (CVE-2015-4000), experimental not vulnerable (OK) (tested w/ 2/4 ciphers only!),
BEAST (CVE-2011-3389)          no CBC ciphers for TLS1 (OK)
RC4 (CVE-2013-2566, CVE-2015-2808) no RC4 ciphers detected (OK)
```

Add smtp_tls_exclude_ciphers

```
## Start cipher suite selection
```

```
# preempt_cipherlist, yes=Postfix chooses here, no=openssl chooses  
tls_preempt_cipherlist = yes
```

```
# for outgoing traffic, use smtp_tls_  
smtp_tls_security_level = may  
smtp_tls_protocols = !SSLv2, !SSLv3  
smtp_tls_mandatory_protocols = !SSLv2, !SSLv3  
smtp_tls_mandatory_ciphers = strong
```

```
smtp_tls_exclude_ciphers= NULL, aNULL, eNULL, RC4, DES, DES+MD5, EXPORT,  
LOW, EXP-EDH-RSA-DES-CBC-SHA,EXP-DES-CBC-SHA, EXP-RC2-CBC-MD5, ECDHE-  
RSA-DES-CBC4-SHA, EDH-RSA-DES-CBC3-SHA, DES-CBC3-SHA, 3DES  
smtp_tls_mandatory_exclude_ciphers= NULL, aNULL, eNULL, RC4, DES, DES+MD5,  
EXPORT, LOW, EXP-EDH-RSA-DES-CBC-SHA,EXP-DES-CBC-SHA, EXP-RC2-CBC-MD5,  
ECDHE-RSA-DES-CBC4-SHA, EDH-RSA-DES-CBC3-SHA, DES-CBC3-SHA, 3DES
```

Add smtpd_tls_exclude_ciphers

```
# for incoming traffic, use smtpd_tls_  
smtpd_tls_security_level = may  
smtpd_tls_protocols = !SSLv2, !SSLv3
```

```
smtpd_tls_exclude_ciphers= NULL, aNULL, eNULL, RC4, DES, DES+MD5, EXPORT,  
LOW, EXP-EDH-RSA-DES-CBC-SHA,EXP-DES-CBC-SHA, EXP-RC2-CBC-MD5, ECDHE-  
RSA-DES-CBC3-SHA, EDH-RSA-DES-CBC3-SHA, DES-CBC3-SHA, 3DES
```

```
smtpd_tls_mandatory_protocols = !SSLv2, !SSLv3  
smtpd_tls_mandatory_ciphers = medium
```

```
smtpd_tls_mandatory_exclude_ciphers= NULL, aNULL, eNULL, RC4, DES,  
DES+MD5, EXPORT, LOW, EXP-EDH-RSA-DES-CBC-SHA,EXP-DES-CBC-SHA, EXP-RC2-  
CBC-MD5,ECDHE-RSA-DES-CBC3-SHA,EDH-RSA-DES-CBC3-SHA,DES-CBC3-SHA,3DES
```

```
smtpd_tls_auth_only, yes if AUTH offered only after STARTTLS has been done  
smtpd_tls_auth_only = yes  
smtpd_tls_cert_file = /etc/ssl/servercerts/123-certs/triple.pem  
smtpd_tls_key_file = /etc/ssl/servercerts/123-certs/private-key.pem  
smtpd_tls_CApath = /etc/ssl/certs
```

```
# End cipher suite selection
```

Observing SMTP TLS on a live system

Portion of `grep TLS /var/log/mail` only some sites support encryption on port 25

See manual about server `smtpd_tls_security_level=may` to offer optional STARTTLS

client `smtp_tls_security_level=may` to accept offered STARTTLS

It says: At the "may" TLS security level, TLS encryption is opportunistic. The SMTP transaction is encrypted if the STARTTLS ESMTP feature is supported by the server. Otherwise, messages are sent in the clear.

```

from mail-lb0-f171.google.com[209.85.217.171]: TLSv1.2 with cipher ECDHE-RSA-AES256-SHA (256/256 bits)
from mail1.mcsignup.com[205.201.139.33]: TLSv1.2 with cipher ECDHE-RSA-AES256-GCM-SHA384 (256/256 bits)
to gmail-smtp-in.l.google.com[74.125.29.27]:25: TLSv1.2 with cipher ECDHE-RSA-AES128-GCM-SHA256 (128/128 bits)
to aspmx.l.google.com[173.194.208.27]:25: TLSv1.2 with cipher ECDHE-RSA-AES128-GCM-SHA256 (128/128 bits)
from 82-70-37-214.dsl.in-addr.zen.co.uk[82.70.37.214]: TLSv1.2 with cipher ECDHE-RSA-AES256-SHA (256/256 bits)
to mn365.mail.protection.outlook.com[23.103.198.42]:25: TLSv1.2 with cipher ECDHE-RSA-AES256-SHA384 (256/256 bits)
to in1-smtp.messagingengine.com[66.111.4.75]:25: TLSv1.2 with cipher ECDHE-RSA-AES256-GCM-SHA384 (256/256 bits)
to mail.finchampstead.com[87.117.246.3]:25: TLSv1.2 with cipher DHE-RSA-AES256-GCM-SHA384 (256/256 bits)
to mail.gernegross-events.de[178.254.50.90]:25: TLSv1 with cipher DHE-RSA-AES256-SHA (256/256 bits)
from mail-lf0-f50.google.com[209.85.215.50]: TLSv1.2 with cipher ECDHE-RSA-AES256-SHA (256/256 bits)
to in1-smtp.messagingengine.com[66.111.4.71]:25: TLSv1.2 with cipher ECDHE-RSA-AES256-GCM-SHA384 (256/256 bits)
to mx.talktalk.net[62.24.202.42]:25: TLSv1.2 with cipher ECDHE-RSA-AES256-GCM-SHA384 (256/256 bits)
to gmail-smtp-in.l.google.com[74.125.29.26]:25: TLSv1.2 with cipher ECDHE-RSA-AES128-GCM-SHA256 (128/128 bits)
from out4-smtp.messagingengine.com[66.111.4.28]: TLSv1.2 with cipher ECDHE-RSA-AES256-GCM-SHA384 (256/256 bits)
from 66-220-144-151.out
to in1-smtp.messagingengine
from out4-smtp.messagingengine
Received: from rhsttp12.uni-regensburg.de (lists.thettp.org [132.199.249.55])
    (using TLSv1.2 with cipher ECDHE-RSA-AES256-GCM-SHA384 (256/256 bits))
    (No client certificate requested)
    by netlab1.net (Postfix) with ESMTPS id 53877103DC6F5
  
```

Dovecot IMAP4, before changes

```

Service set:          STARTTLS via IMAP
Built against openssl 0.9.8
With SSLv3

--> Testing protocols (via openssl, SSLv2 via sockets)

SSLv2      not offered (OK)
SSLv3      offered (NOT ok)
TLS 1      offered
TLS 1.1    not offered
TLS 1.2    not offered (NOT ok)
SPDY/NPN   (SPDY is a HTTP protocol and thus not tested here)

                                     TLS 1.2 needs a better crypto engine

--> Testing ~standard cipher lists

Null Ciphers      not offered (OK)
Anonymous NULL Ciphers not offered (OK)
Anonymous DH Ciphers not offered (OK)
40 Bit encryption not offered (OK)
56 Bit encryption Local problem: No 56 Bit encryption configured
Export Ciphers (general) not offered (OK)
Low (<=64 Bit)    not offered (OK)
DES Ciphers       not offered (OK)
Medium grade encryption offered (NOT ok)
Triple DES Ciphers offered (NOT ok)
High grade encryption offered (OK)

--> Testing (perfect) forward secrecy, (P)FS -- omitting 3DES, RC4 and Null E
✓ PFS is offered (OK) DHE-RSA-AES256-SHA DHE-RSA-CAMELLIA256-SHA DHE-RSA-AES1

```


Dovecot problem area

Heartbleed (CVE-2014-0160)	not vulnerable (OK) (timed out)	Before
CCS (CVE-2014-0224)	not vulnerable (OK)	
Secure Renegotiation (CVE-2009-3555)	not vulnerable (OK)	
Secure Client-Initiated Renegotiation	VULNERABLE (NOT ok), DoS threat	
CRIME, TLS (CVE-2012-4929)	not vulnerable (OK) (not using HTTP anyway)	
POODLE, SSL (CVE-2014-3566)	VULNERABLE (NOT ok), uses SSLv3+CBC (check TLS_FALLBACK_SCSV)	
TLS_FALLBACK_SCSV (RFC 7507), experim.	Downgrade attack prevention NOT supported	
FREAK (CVE-2015-0204)	not vulnerable (OK) (tested with 6/9 ciphers)	
LOGJAM (CVE-2015-4000), experimental	not vulnerable (OK) (tested w/ 2/4 ciphers only!), ciphers: EDH-RSA-DES-CBC3-SHA DES-CBC3-SHA	
BEAST (CVE-2011-3389)	TLS1: EDH-RSA-DES-CBC3-SHA DES-CBC3-SHA	
RC4 (CVE-2013-2566, CVE-2015-2808)	VULNERABLE (NOT ok): RC4-SHA RC4-MD5 RC4-MD5	

Heartbleed (CVE-2014-0160)	not vulnerable (OK) (timed out)	After
CCS (CVE-2014-0224)	not vulnerable (OK)	
Secure Renegotiation (CVE-2009-3555)	not vulnerable (OK)	
Secure Client-Initiated Renegotiation	VULNERABLE (NOT ok), DoS threat	
CRIME, TLS (CVE-2012-4929)	not vulnerable (OK) (not using HTTP anyway)	
POODLE, SSL (CVE-2014-3566)	not vulnerable (OK)	
TLS_FALLBACK_SCSV (RFC 7507), experim.	Downgrade attack prevention NOT supported	
FREAK (CVE-2015-0204)	not vulnerable (OK) (tested with 6/9 ciphers)	
LOGJAM (CVE-2015-4000), experimental	not vulnerable (OK) (tested w/ 2/4 ciphers only!),	
BEAST (CVE-2011-3389)	no CBC ciphers for SSL3 (OK)	
	no CBC ciphers for TLS1 (OK)	
RC4 (CVE-2013-2566, CVE-2015-2808)	no RC4 ciphers detected (OK)	

“After” is after changing Dovecot cipher suite configuration
Still no TLS v1.2

Dovecot cipher suite improvements

File 10-ssl.conf

SSL protocols to use

ssl_protocols = !SSLv2, !SSLv3

Prefer the server's order of ciphers over client's.

ssl_prefer_server_ciphers = yes

SSL extra options. Currently supported options are:

no_compression - Disable compression.

ssl_options = no_compression

SSL ciphers to use

#WAS ssl_cipher_list = ALL:!LOW:!SSLv2:!EXP:!aNULL

ssl_cipher_list = ALL:!LOW:!SSLv2:!EXP:!aNULL:!RC4:!EDH-RSA-DES-CBC3-SHA:!DES-CBC3-SHA

The cipher list must be all on one line

Dovecot recompiled with openssl1-devel

Build against better crypto engine: openssl1 to enable TLS v1.2

Service set: STARTTLS via IMAP

--> Testing protocols (via openssl, SSLv2 via sockets)

```
SSLv2      not offered (OK)
SSLv3      not offered (OK)
TLS 1      offered
TLS 1.1    offered
TLS 1.2    offered (OK)
SPDY/NPN   (SPDY is a HTTP protocol and thus not tested here)
```

--> Testing ~standard cipher lists

```
Null Ciphers          not offered (OK)
Anonymous NULL Ciphers not offered (OK)
Anonymous DH Ciphers  not offered (OK)
40 Bit encryption     not offered (OK)
56 Bit encryption     Local problem: No 56 Bit encryption configured in /usr/bin/op
Export Ciphers (general) not offered (OK)
Low (<=64 Bit)        not offered (OK)
DES Ciphers           not offered (OK)
Medium grade encryption offered (NOT ok)
Triple DES Ciphers    not offered (OK)
High grade encryption  offered (OK)
```

--> Testing (perfect) forward secrecy, (P)FS -- omitting 3DES, RC4 and Null Encryption here

✓ PFS is offered (OK) ECDHE-RSA-AES256-GCM-SHA384 DHE-RSA-AES256-GCM-SHA384 DHE-RSA-AES256-

Dovecot, recompiled with openssl1-devel

--> Testing vulnerabilities

Heartbleed (CVE-2014-0160)	not vulnerable (OK) (timed out)
CCS (CVE-2014-0224)	not vulnerable (OK)
Secure Renegotiation (CVE-2009-3555)	not vulnerable (OK)
Secure Client-Initiated Renegotiation	VULNERABLE (NOT ok), DoS threat
CRIME, TLS (CVE-2012-4929)	not vulnerable (OK) (not using HTTP anyway)
POODLE, SSL (CVE-2014-3566)	not vulnerable (OK)
TLS_FALLBACK_SCSV (RFC 7507), experim.	Downgrade attack prevention supported (OK)
FREAK (CVE-2015-0204)	not vulnerable (OK) (tested with 6/9 ciphers)
LOGJAM (CVE-2015-4000), experimental	not vulnerable (OK) (tested w/ 2/4 ciphers only!)
BEAST (CVE-2011-3389)	no CBC ciphers for TLS1 (OK)
RC4 (CVE-2013-2566, CVE-2015-2808)	no RC4 ciphers detected (OK)

Have TLS v1.0, v1.1 and v1.2

TLS downgrade vulnerability is now mitigated

Dovecot links to openssl for crypto, does not rely upon Postfix for that

GroupWise, looking into ports 25, 587

Testing GroupWise 14.2.0-121330

Service set: STARTTLS via SMTP

--> Testing protocols (via openssl, SSLv2 via sockets)

```
SSLv2      not offered (OK)
SSLv3      not offered (OK)
TLS 1      offered
TLS 1.1    offered
TLS 1.2    offered (OK)
SPDY/NPN   (SPDY is a HTTP protocol and thus not tested here)
```

--> Testing ~standard cipher lists

```
Null Ciphers          not offered (OK)
Anonymous NULL Ciphers not offered (OK)
Anonymous DH Ciphers  not offered (OK)
40 Bit encryption     not offered (OK)
56 Bit encryption     Local problem: No 56 Bit encryption configured in
Export Ciphers (general) not offered (OK)
Low (<=64 Bit)        not offered (OK)
DES Ciphers           not offered (OK)
Medium grade encryption not offered (OK)
Triple DES Ciphers    offered (NOT ok)
High grade encryption  offered (OK)
```

--> Testing (perfect) forward secrecy, (P)FS -- omitting 3DES, RC4 and Null Encry

✓ PFS is offered (OK) ECDHE-RSA-AES256-GCM-SHA384 DHE-RSA-AES256-GCM-SHA384 DHE-R

GroupWise, looking into IMAP4 ports

`./testssl1.sh -U -t imap aGWHost:143 and 993`

Testing GroupWise 14.2.0-121330

```
Heartbleed (CVE-2014-0160)      VULNERABLE (NOT ok)      Port 143
CCS (CVE-2014-0224)            not vulnerable (OK) (timed out)
Secure Renegotiation (CVE-2009-3555) not vulnerable (OK)
Secure Client-Initiated Renegotiation VULNERABLE (NOT ok), DoS threat
CRIME, TLS (CVE-2012-4929)     not vulnerable (OK) (not using HTTP anyway)
POODLE, SSL (CVE-2014-3566)    not vulnerable (OK)
TLS_FALLBACK_SCSV (RFC 7507), experim. Downgrade attack prevention supported (OK)
FREAK (CVE-2015-0204)         not vulnerable (OK) (tested with 6/9 ciphers)
LOGJAM (CVE-2015-4000), experimental not vulnerable (OK) (tested w/ 2/4 ciphers only!), c
BEAST (CVE-2011-3389)         TLS1: ECDHE-RSA-DES-CBC3-SHA EDH-RSA-DES-CBC3-SHA
                                DES-CBC3-SHA
                                -- but also supports higher protocols (possible miti
                                no RC4 ciphers detected (OK)

RC4 (CVE-2013-2566, CVE-2015-2808)
```

```
Heartbleed (CVE-2014-0160)      not vulnerable (OK) (timed out)      Port 993
CCS (CVE-2014-0224)            not vulnerable (OK)
Secure Renegotiation (CVE-2009-3555) not vulnerable (OK)
Secure Client-Initiated Renegotiation not vulnerable (OK)
CRIME, TLS (CVE-2012-4929)     not vulnerable (OK) (not using HTTP anyway)
POODLE, SSL (CVE-2014-3566)    not vulnerable (OK)
TLS_FALLBACK_SCSV (RFC 7507), experim. Downgrade attack prevention supported (OK)
FREAK (CVE-2015-0204)         not vulnerable (OK) (tested with 6/9 ciphers)
LOGJAM (CVE-2015-4000), experimental not vulnerable (OK) (tested w/ 2/4 ciphers only!), c
BEAST (CVE-2011-3389)         TLS1: ECDHE-RSA-DES-CBC3-SHA EDH-RSA-DES-CBC3-SHA
                                DES-CBC3-SHA
                                -- but also supports higher protocols (possible miti
                                no RC4 ciphers detected (OK)

RC4 (CVE-2013-2566, CVE-2015-2808)
```

While we are here, eDir LDAPS?

./testssl1.sh myhost:636

--> Testing protocols (via sockets except TLS 1.2 and SPDY/NPN)

```
SSLv2      supported but couldn't detect a cipher (may need further attention)
SSLv3      offered (NOT ok)
TLS 1      offered
TLS 1.1    not offered
TLS 1.2    not offered (NOT ok)
SPDY/NPN   not offered
```

--> Testing ~standard cipher lists

```
Null Ciphers          not offered (OK)
Anonymous NULL Ciphers not offered (OK)
Anonymous DH Ciphers  not offered (OK)
40 Bit encryption     offered (NOT ok)
56 Bit encryption     Local problem: No 56 Bit encryption configured in /usr/bin/ope
Export Ciphers (general) offered (NOT ok)
Low (<=64 Bit)        offered (NOT ok)
DES Ciphers           offered (NOT ok)
Medium grade encryption offered (NOT ok)
Triple DES Ciphers    offered (NOT ok)
High grade encryption  offered (OK)
```

--> Testing (perfect) forward secrecy, (P)FS -- omitting 3DES, RC4 and Null Encryption here

✗ Not OK: No ciphers supporting Forward Secrecy offered

While we are here, eDir LDAPS?

--> Testing vulnerabilities

```
Heartbleed (CVE-2014-0160)      not vulnerable (OK) (timed out)
CCS (CVE-2014-0224)           not vulnerable (OK)
Secure Renegotiation (CVE-2009-3555) VULNERABLE (NOT ok)
Secure Client-Initiated Renegotiation likely not vulnerable (OK) (timed out)
CRIME, TLS (CVE-2012-4929)     not vulnerable (OK) (not using HTTP anyway)
POODLE, SSL (CVE-2014-3566)    VULNERABLE (NOT ok), uses SSLv3+CBC (check TLS_FALLBACK_SCSV
TLS_FALLBACK_SCSV (RFC 7507), experim. Downgrade attack prevention NOT supported
FREAK (CVE-2015-0204)         VULNERABLE (NOT ok), uses EXPORT RSA ciphers
LOGJAM (CVE-2015-4000), experimental not vulnerable (OK) (tested w/ 2/4 ciphers only!), common pri
BEAST (CVE-2011-3389)         SSL3: DES-CBC3-SHA DES-CBC-SHA
                                EXP-DES-CBC-SHA EXP-RC2-CBC-MD5 EXP-RC2-CBC-MD5
                                TLS1: DES-CBC3-SHA DES-CBC-SHA
                                EXP-DES-CBC-SHA EXP-RC2-CBC-MD5 EXP-RC2-CBC-MD5
RC4 (CVE-2013-2566, CVE-2015-2808) VULNERABLE (NOT ok): RC4-SHA RC4-MD5 RC4-MD5 EXP-RC4-MD5 EXP-
```

This is the pattern which we anticipate when a program is built against the openssl v0.9.8 library and older conventional crypto choices. Times change, alas.



SSH upgrade to use openssl v1.0.1

Before upgrade

<input checked="" type="checkbox"/>	libnsssharedhelper0	Helper functions ...	1.0.10-0.7.33	37.0 KiB
<input checked="" type="checkbox"/>	libnsssharedhelper0-32bit	Helper functions ...	1.0.10-0.7.33	10.0 KiB
<input checked="" type="checkbox"/>	libssh2-1	A library implem...	1.2.9-4.2.4.1	156.0 KiB
<input checked="" type="checkbox"/>	openssh	Secure Shell Cli...	6.2p2-0.24.1	3.0 MiB
<input checked="" type="checkbox"/>	openssh-askpass	A passphrase di...	6.2p2-0.24.1	47.0 KiB
<input checked="" type="checkbox"/>	yast2-sshd	YaST2 - SSH Se...	2.17.2-1.21	114.0 KiB
<input type="checkbox"/>	kdssh	Remote Applicati...	(4.3.5-0.3.3)	80.0 KiB
<input type="checkbox"/>	libssh-devel	SSH library devel...	(0.2-5.16.1)	36.0 KiB
<input type="checkbox"/>	libssh-devel-doc	SSH library api d...	(0.2-5.16.1)	409.0 KiB
<input type="checkbox"/>	libssh2	SSH library	(0.2-5.20.1)	135.0 KiB
<input type="checkbox"/>	libssh2-1-32bit	A library implem...	(1.2.9-4.2.4.1)	155.0 KiB
<input type="checkbox"/>	libssh2-devel	A library implem...	(1.2.9-4.2.4.1)	293.0 KiB
<input type="checkbox"/>	openssh-askpass-gnome	A GNOME-Base...	(6.2p2-0.24.3)	15.0 KiB
<input type="checkbox"/>	openssh-openssl1	Secure Shell Cli...	(6.6p1-10.1)	5.0 MiB

Replace /etc/pam.d/sshd with your older version
which has pam_nam before Linux

Installation of openssh-openssl1

Upgrade choice

The screenshot shows a package manager window with a list of packages. The package **openssh-openssl1** is selected. A conflict resolution dialog is open, showing a conflict between **openssh-openssl1-6.6p1-10.1.x86_64** and **openssh < 6.6p1-10.1**. The dialog offers two options for conflict resolution:

- ☐ Following actions will be done:
 - deinstallation of openssh-6.2p2-0.24.1.x86_64
 - deinstallation of openssh-askpass-6.2p2-0.24.1.x86_64
- ☐ do not install openssh-openssl1-6.6p1-10.1.x86_64

Below the package list, the **openssh-openssl1** package details are shown, including the selected version **6.6p1-10.1-x86_64** from **SLE11-Security-Module** with priority 99 and vendor SUSE LINUX Products.

Replace /etc/pam.d/sshd with your older version
which has pam_nam before Linux items

Openssh touchups

We may be required to remove old contents of `.ssh` in our home directories. Content is often just text file **known_hosts** listing preferred crypto and fingerprint of favourite hosts

Crypto details are left to you as homework reading. Hints:

- `ssh -Q cipher`

- `ssh -Q kex`

- `ssh -Q mac`

For long detailed discussion of SSH crypto details please see:
<https://stribika.github.io/2015/01/04/secure-secure-shell.html>



MindWorks Inc. Ltd
210 Burnley Road
Weir
Bacup
OL13 8QE UK

Telephone: +44 (0) 170 687 1900
Fax: +44 (0) 170 687 8203
Web: www.mindworksuk.com
Email: training@mindworksuk.com