Packet Clearing House

# All TLS, All The Time

## Using *Apache* and *Let's Encrypt* to set up a secure web server

**Ashley Jones**

**Packet Clearing House**

**SANOG 27 - 26th January 2016**

**https://pch.net/sanog27**

# Requirements

- root
- Supported Let's Encrypt environment (e.g. Ubuntu)
- DNS entry for your server

PCH
Packet Clearing House

# Tools

- Ubuntu 14.04
- Apache v.2.4.x
- Let’s Encrypt, public beta

# Why Encrypt?

- Public servers are exposed to hackers
- Private servers are exposed to internal hackers ;) *
- Firesheep can hijack sessions
- Man in the middle (MiTM) protection
- Snowden says, “encryption works”

* Let’s Encrypt requires public IP

Packet Clearing House

# Netherlands Vodaphone MiTM

**Before:**

```
<html><body>Hello World</body></html>
```

**After:**

```
<html>
<script src="http://1.2.3.4/bmi-int-js/bmi.js"
language="javascript"></script>
<body>Hello World</body>
</html>
<script language="javascript"><!-
bmi_SafeAddOnload(bmi_load,"bmi_orig_img",
1);//--></script>
```

PCH
Packet Clearing House

# Why encrypt well?

- IPv4 and IPv6 with the same effort
- POODLE, BEAST, Heartbleed, Logjam and more for sure
- Weaponized enables script kiddies
- HTTP Strict Transport Security (HSTS)
- Easy Tools:
  - Cipher List https://cipherli.st
  - Mozilla TLS Config Generator https://mozilla.github.io/server-side-tls/ssl-config-generator/
  - Qualys SSL Labs https://ssllabs.com/ssltest

Packet Clearing House

# Install Apache and Git

```
apt-get update&&apt-get upgrade

apt-get install apache2 git

a2enmod ssl headers rewrite

a2enconf security
```

# Apache Config Edit: ssl.conf

```
<IfModule mod_ssl.c>
 SSLRandomSeed connect builtin
 SSLRandomSeed connect file:/dev/urandom 512
 AddType application/x-x509-ca-cert .crt
 AddType application/x-pkcs7-crl .crl
 SSLPassPhraseDialog exec:/usr/share/apache2/ask-for-passphrase
 SSLSessionCache shmcb:${APACHE_RUN_DIR}/ssl_scache(512000)
 SSLSessionCacheTimeout  300
 SSLCipherSuite EECDH+AESGCM:EDH+AESGCM:AES256+EECDH:AES256+EDH
 SSLHonorCipherOrder on
 SSLProtocol All -SSLv2 -SSLv3
 SSLCompression off
 SSLUseStapling on
SSLStaplingCache "shmcb:logs/stapling-cache(150000)"
</IfModule>
```

Packet Clearing House

# Apache Config Edit: default-ssl.conf

```
<IfModule mod_ssl.c><VirtualHost _default_:443>
ServerAdmin webmaster@localhost
DocumentRoot /var/www/html
ErrorLog ${APACHE_LOG_DIR}/error.log
CustomLog ${APACHE_LOG_DIR}/access.log combined
SSLEngine on
SSLCertificateFile /etc/ssl/certs/ssl-cert-snakeoil.pem
SSLCertificateKeyFile /etc/ssl/private/ssl-cert-snakeoil.key
<FilesMatch "\.(cgi|shtml|phtml|php)$">
    SSLOptions +StdEnvVars
</FilesMatch>
<Directory /usr/lib/cgi-bin>
    SSLOptions +StdEnvVars
</Directory>
BrowserMatch "MSIE [2-6]" \
    nokeepalive ssl-unclean-shutdown \
    downgrade-1.0 force-response-1.0
BrowserMatch "MSIE [17-9]" ssl-unclean-shutdown
</VirtualHost></IfModule>
```

Packet Clearing House

# Apache Config Edit: security.conf

```
ServerTokens Prod
ServerSignature Off
TraceEnable Off
Header set X-Content-Type-Options: "nosniff"
Header always set X-Frame-Options DENY
```

Packet Clearing House

# Apache Config Create: 100-tls-test.conf

```
<VirtualHost *:80>
  ServerName 1.tls-test.plip.com
  ServerAdmin mrjones@pch.net
  DocumentRoot /var/www/html/
  ErrorLog ${APACHE_LOG_DIR}/error_log
  TransferLog ${APACHE_LOG_DIR}/access_log
  RewriteEngine On
  RewriteRule (.*) https://%{HTTP_HOST}%{REQUEST_URI} [R=301,L]
</VirtualHost>
<VirtualHost *:443>
  ServerName 1.tls-test.plip.com
  DocumentRoot /var/www/html/
  ServerAdmin mrjones@pch.net
  ErrorLog ${APACHE_LOG_DIR}/ssl_error_log
  TransferLog ${APACHE_LOG_DIR}/ssl_access_log
  SSLEngine On
  Header always set Strict-Transport-Security "max-age=15768000"
  SSLCertificateFile /etc/ssl/certs/ssl-cert-snakeoil.pem
  SSLCertificateKeyFile /etc/ssl/private/ssl-cert-snakeoil.key
</VirtualHost>
```

Packet Clearing House

# Apache Enable and Restart

```
a2ensite 100-tls-test.conf

service apache2 reload
```

Packet Clearing House

# Careful!

- HSTS **`includeSubdomains`**
- SSLUseStapling, SSLCompression, SSLStaplingCache (version 2.4 or higher)
- SSLSessionTickets Requires Apache (version 2.4.10 or higher)
- Reference Mozilla’s TLS Config Generator

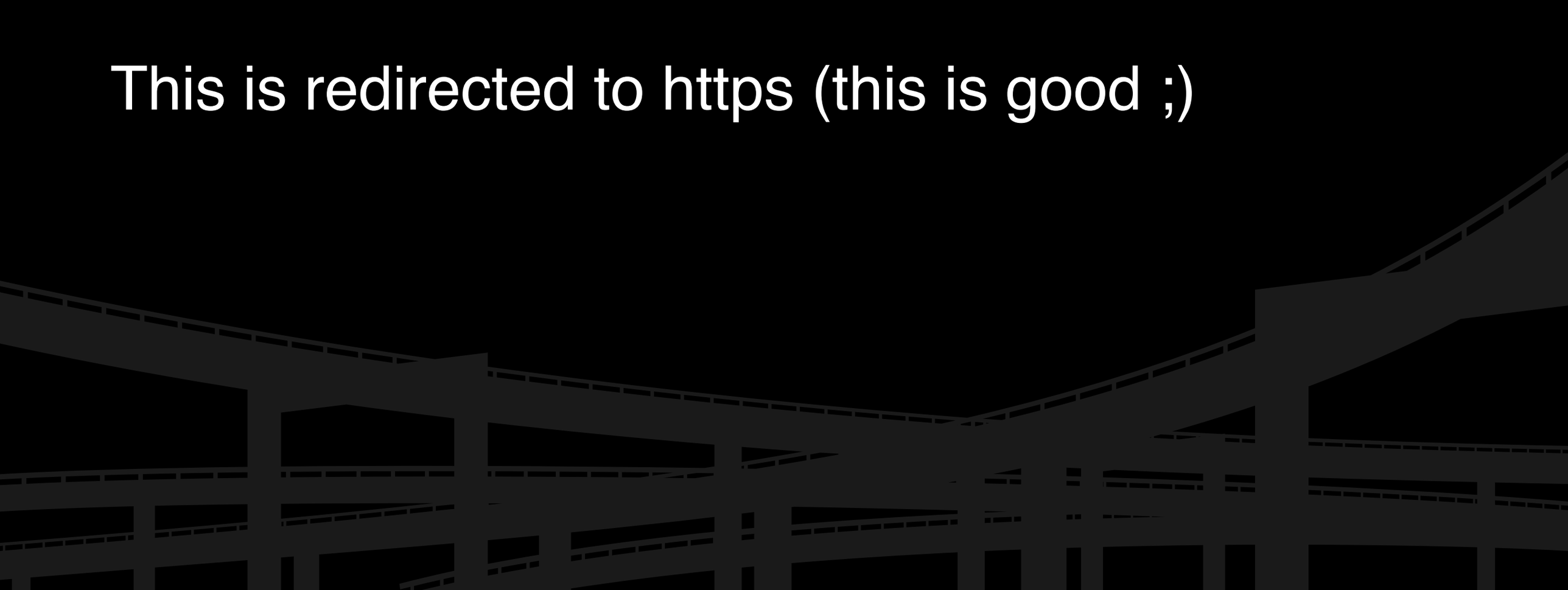
Packet Clearing House

# Test HTTP

Curl is your friend!

```
curl -I http://1.tls-test.plip.com
```

This is redirected to https (this is good ;)



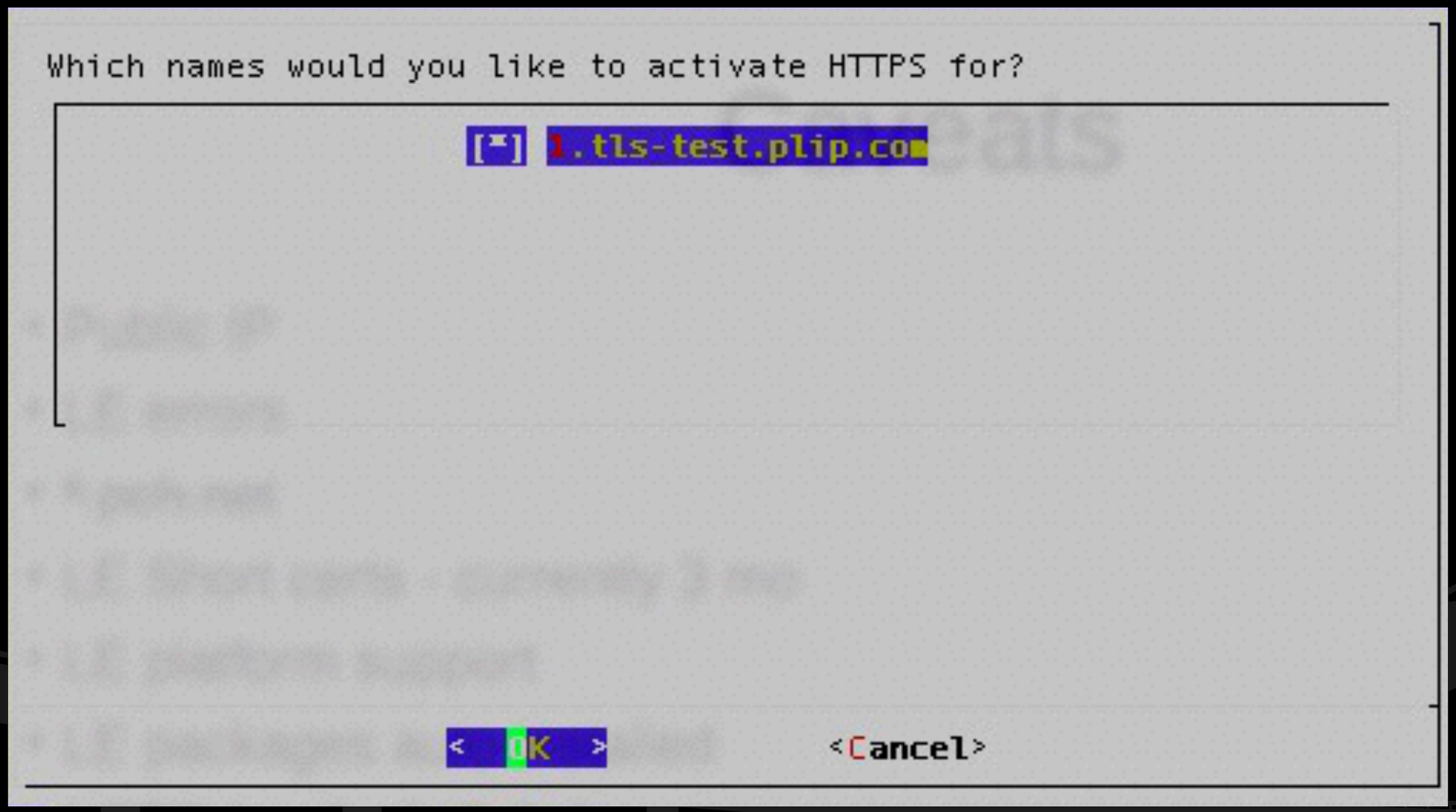
Packet Clearing House

# Let’s Encrypt: Install & Create Cert

```
git clone https://github.com/letsencrypt/letsencrypt

./letsencrypt/letsencrypt-auto --apache
```

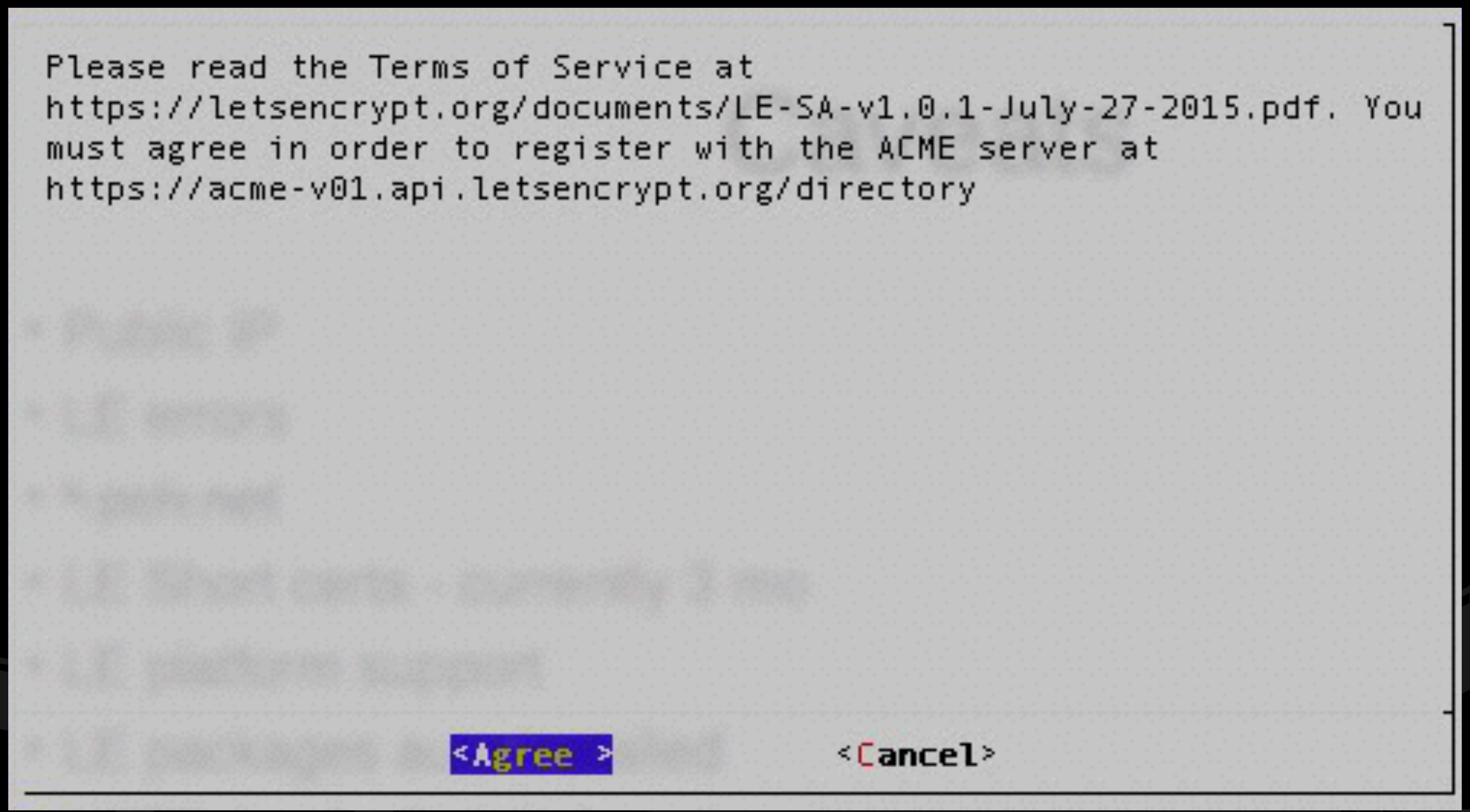
# Let’s Encrypt: Install & Create Cert



# Let’s Encrypt: Install & Create Cert

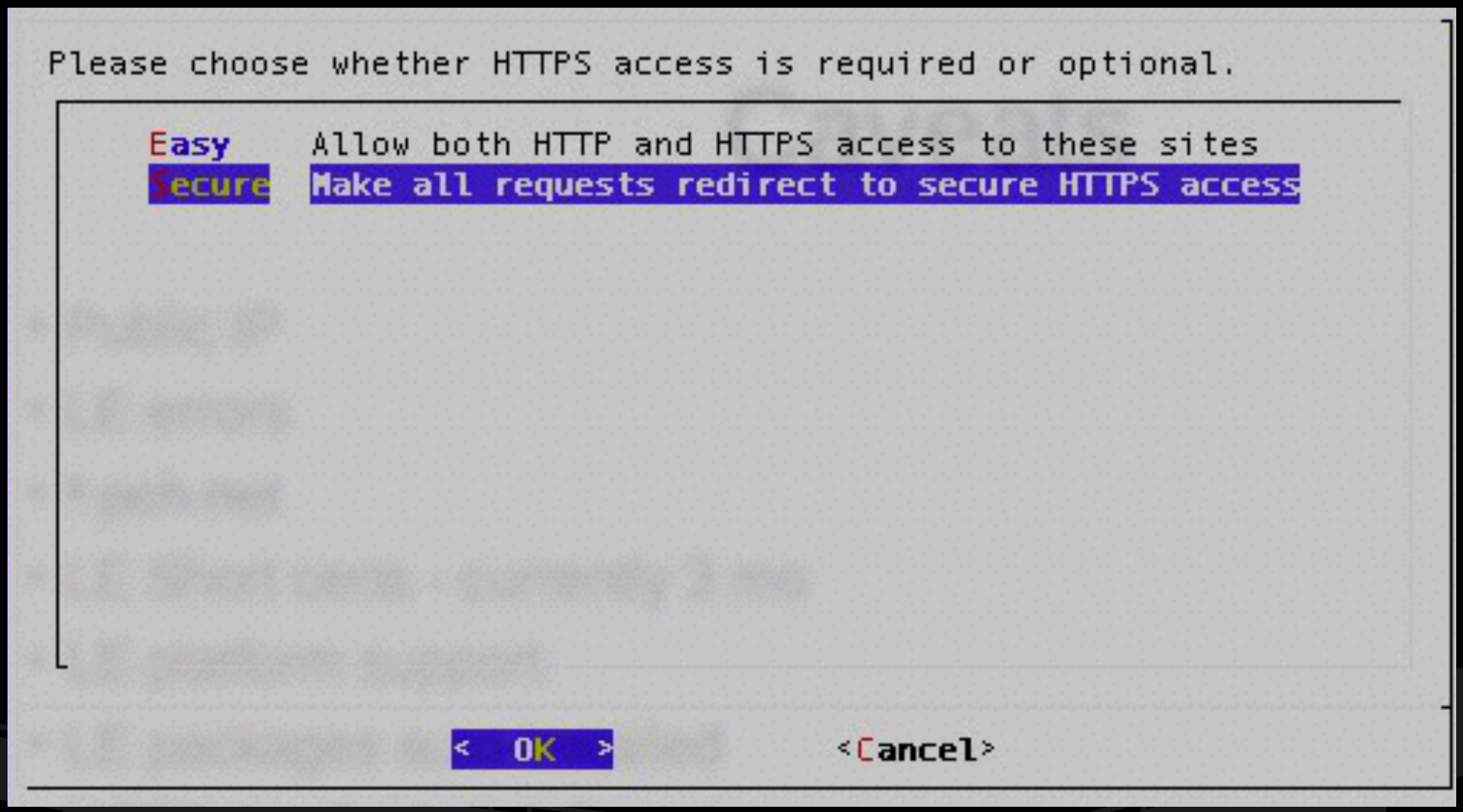
Packet Clearing House

# Let’s Encrypt: Install & Create Cert



Packet Clearing House

# Let’s Encrypt: Install & Create Cert



Packet Clearing House

# Let’s Encrypt: Install & Create Cert

```
Congratulations! You have successfully enabled
https://1.tls-test.plip.com

You should test your configuration at:
https://www.ssllabs.com/ssltest/analyze.html?d=1.tls-test.plip.com


                          < OK >
```

# Resulting 100-tls-test.conf

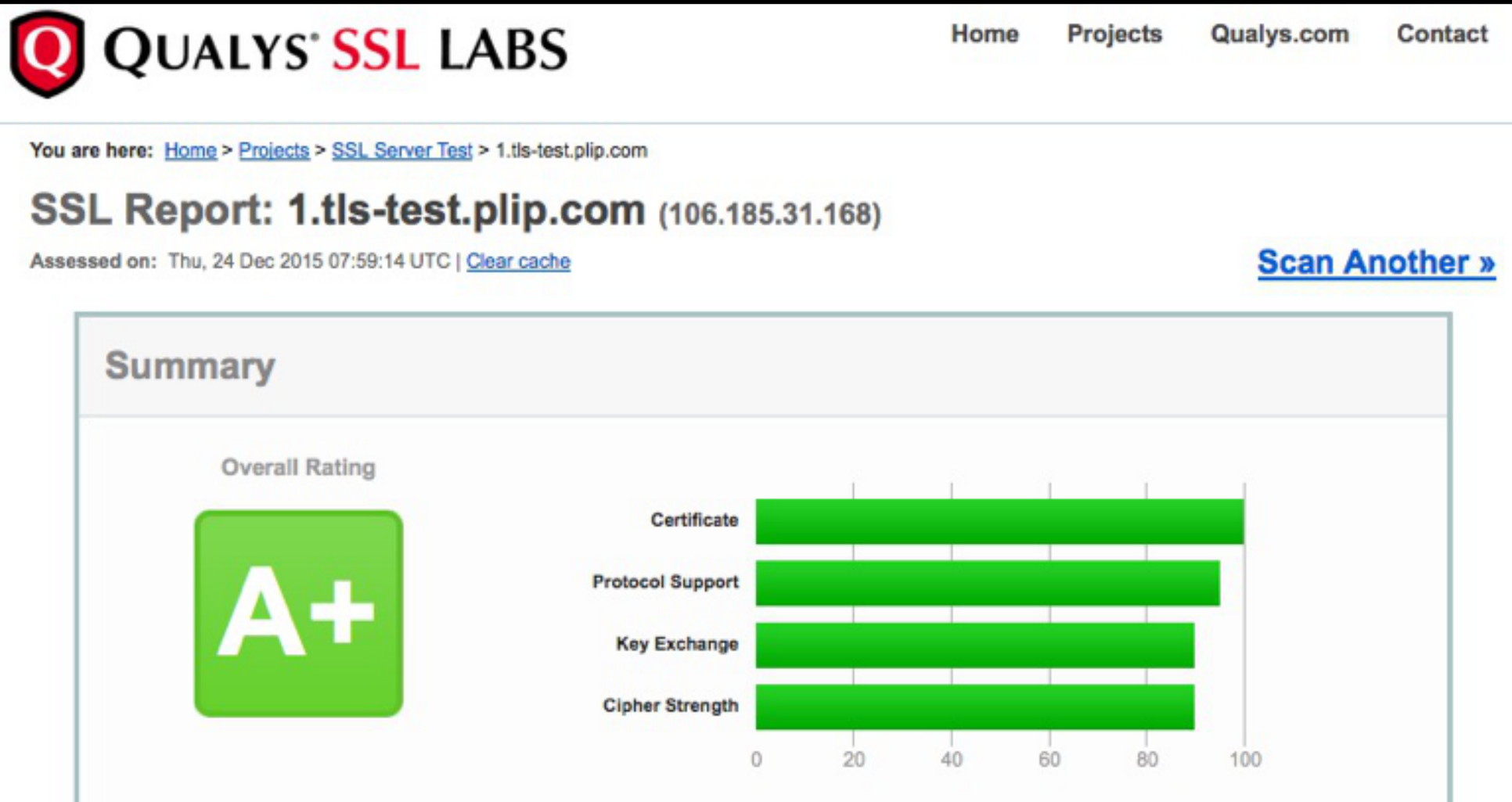
```
<VirtualHost *:443>
[SNIP]
SSLCertificateFile
/etc/letsencrypt/live/1.tls-test.plip.com/cert.pem
SSLCertificateKeyFile
/etc/letsencrypt/live/1.tls-test.plip.com/privkey.pem
SSLCertificateChainFile
/etc/letsencrypt/live/1.tls-test.plip.com/chain.pem
</VirtualHost>
```

Packet Clearing House

# Test HTTPS

Curl is still your friend! (so is SSL Labs)

```
curl -I https://1.tls-test.plip.com
```



Packet Clearing House

# Myth: TLS is slow

“On our production machines, TLS accounts for less than 1% CPU load, less than 10 KB RAM per connection and less than 2% of network overhead. People believe that TLS takes a lot of CPU...we hope our numbers will dispel that.”

-Adam Langley, Google “Overclocking SSL” **2010**

See https://istlsfastyet.com/

Packet Clearing House

# Myth: Let's Encrypt requires root

- Non-root solution is available!
- Let's Encrypt has  APIs and SDKs which don't all require root

# Myth: Managing certs is hard

- 100% automated
- Command line or
- GUI:

```
You have an existing certificate that contains exactly the same
domains you requested and isn't close to expiry.
(ref: /etc/letsencrypt/renewal/1.tls-test.plip.com.conf)

What would you like to do?

  1  Attempt to reinstall this existing certificate
  2  Renew & replace the cert (limit ~5 per 7 days)
  3  Cancel this operation and do nothing

          < OK >                <Cancel>
```

Packet Clearing House

# Myth: TLS requires 1 IP per certificate

- Server Name Indication (SNI) solves this
- Widely supported
- Compatible with IPv4 and IPv6

Packet Clearing House

# Caveats

- Requires Public IP
- Let’s Encrypt is in beta - errors?!#@
- No Wildcards (~~*.tls-test.plip.com~~)
- Short certs - currently 3 months
- Only specific platform supported
- Related packages auto-installed
- Non HTTPS assets (eg JS) cause warnings
- Orphaned browsers

Packet Clearing House

# Thanks! Questions?
# https://pch.net/sanog27

- Ashley Jones - mrjones@pch.net - personal blog https://blog.plip.com
- PCH https://pch.net
- Vodaphone in Netherlands: http://www.sphaero.org/blog:2012:0418_am_i_hacked_oh_it_s_just_vodafone
- Let's Encrypt https://letsencrypt.org
- Apache https://httpd.apache.org
- Ubuntu https://ubuntu.com
- Cipherli.st https://cipherli.st/
- iptables http://www.netfilter.org/projects/iptables/
- POODLE https://en.wikipedia.org/wiki/POODLE
- BEAST https://en.wikipedia.org/wiki/Transport_Layer_Security#BEAST_attack
- Heartbleed http://heartbleed.com/
- Logjam https://weakdh.org/
- RFC2246 https://tools.ietf.org/html/rfc2246
- List of all Let's Encrypt client https://community.letsencrypt.org/t/list-of-client-implementations/
- No Root Let's Encrypt How-To  https://github.com/diafygi/letsencrypt-nosudo
- Mozilla's TLS Config Generator https://mozilla.github.io/server-side-tls/ssl-config-generator/
- Is TLS Fast Yet? https://istlsfastyet.com/