lilo.linux.or.jp の話 (2017年1月)

Kazuhiro NISHIYAMA

LILO&東海道らぐオフラインミーティング

2017-01-07

lilo.linux.or.jp とは?

主な用途:

- LILO の Web サーバー (apache)
- ML サーバー (mailman)

環境

- さくらの VPS
- Debian GNU/Linux

今回の話

前回以降の話

- unattended-upgrades
- HSTS
- mailman の fix_url
- certbot
- OCSP Stapling などの SSL/TLS 設定変更

unattended-upgrades (1/5)

- セキュリティアップデートの自動化
- 似た機能の cron-apt はすでに入れていたが purge した
  - sudo aptitude purge cron-apt

unattended-upgrades (2/5)

- 設定ファイルは /etc/apt/apt.conf.d/50unattended-upgrades
- /etc/cron.daily/apt で自動更新
  - apt 公式に近いように見えたので cron-apt から乗り換えた

unattended-upgrades (3/5)

- メール送信設定
  - `Unattended-Upgrade::Mail "root";`
  - mailx が必要
- 自動再起動設定
  - `Unattended-Upgrade::Automatic-Reboot "true";`
  - sudo dpkg-reconfigure -plow unattended-upgrades で Yes

unattended-upgrades (4/5)

- /boot が溢れないように autoremove 設定
  - `Unattended-Upgrade::Remove-Unused-Dependencies "true";`

unattended-upgrades (5/5)

- `Unattended-Upgrade::Origins-Pattern` で自動更新対象を制限
  - デフォルトはセキュリティアップデートのみ自動更新
- ポイントリリースや backports (や milter-manager) は手動更新

unattended-upgrades 例1

メールの例1 (単純な更新)

```
Subject: unattended-upgrades result for 'chiyoko': 'True'

自動アップグレードは以下を返しました: True

Packages that were upgraded:
 libidn11

Unattended-upgrades log:
初期状態でブラックリストにあるパッケージ:
Initial whitelisted packages:
自動アップグレードスクリプトを開始
許可されているパッケージ導入元: ['origin=Debian,codename=jessie,label=Debian-Security']
Packages that will be upgraded: libidn11
dpkg のログを '/var/log/unattended-upgrades/unattended-upgrades-dpkg.log' に書き込み中
全てのアップグレードがインストールされました
```

unattended-upgrades 例2

メールの例2 (再起動が必要な時)

```
Subject: [reboot required] unattended-upgrades result for 'chiyoko': True

自動アップグレードは以下を返しました: True

Warning: A reboot is required to complete this upgrade.

Packages that were upgraded:
 linux-image-3.16.0-4-amd64 linux-libc-dev

Unattended-upgrades log:
初期状態でブラックリストにあるパッケージ:
Initial whitelisted packages:
自動アップグレードスクリプトを開始
許可されているパッケージ導入元: ['origin=Debian,codename=jessie,label=Debian-Security']
Packages that will be upgraded: linux-image-3.16.0-4-amd64 linux-libc-dev
dpkg のログを '/var/log/unattended-upgrades/unattended-upgrades-dpkg.log' に書き込み中
全てのアップグレードがインストールされました
```

HSTS (1/5)

- HTTP Strict Transport Security
- 次回以降のアクセス時に http から https にブラウザー側で置き換える
- ブラウザーが HSTS preload list を持っていて、登録されているサイトは最初から https になる (mail.google.com など)

HSTS (2/5)

- Strict-Transport-Security: max-age=31536000
  - 365日 (31,536,000秒) に設定
- includeSubDomains はつけていない
- max-age の間、http に戻せないようなものなので慎重に
  - DNS の TTL と同じようなもの

HSTS (3/5)

apache2 での設定:

- sudo a2enmod headers
- VirtualHost の設定に追加

```
Header set Strict-Transport-Security "max-age=31536000"
```

- sudo service apache2 restart

HSTS (4/5)

http から https へリダイレクト

- http の VirtualHost に RedirectPermanent を設定

```
RedirectPermanent / https://lilo.linux.or.jp/
```

HSTS (5/5)

- Google Chrome だと HSTS のキャッシュは `chrome://net-internals/#hsts` で消せる

(動作確認に必要だった)

mailman の fix_url (1/3)

- /etc/mailman/mm_cfg.py で設定
  - `DEFAULT_URL_PATTERN = 'https://%s/cgi-bin/mailman/'`
- この設定をしても admindb の URL が http のままだった

mailman の fix_url (2/3)

- Mailman admindb using http instead of https in formaction
  - https://mail.python.org/pipermail/mailman-users/2011-October/072312.html
  - http://wiki.list.org/x/7oA9 and http://wiki.list.org/x/mIA9.
- 「`withlist -l -a -r fix_url`」で解決

mailman の fix_url (3/3)

実行結果 (一部の ML の出力を抜粋)

```
# # withlist -l -a -r fix_url
# fix_url を import 中...
# fix_url.fix_url() を実行中...
# lilo のリストを読み込中 (ロック完了)
# リストを保存中
# mailman のリストを読み込中 (ロック完了)
# リストを保存中
# 最終処理中
```

certbot (1/10)

/etc/letsencrypt/cli.ini を設定

```
rsa-key-size = 4096
post-hook = /etc/letsencrypt/post-hook
```

certbot (2/10)

- 公開鍵のビット数
- `rsa-key-size = 4096` に設定
- デフォルトは 2048

certbot (3/10)

```
post-hook = apachectl graceful; service postfix reload >/dev/null
```

と書いたら

```
certbot: error: Unexpected line 1
in /etc/letsencrypt/cli.ini:
post-hook = apachectl graceful;
service postfix reload >/dev/null
(実際は1行)
```

というエラー

certbot (4/10)

post-hook ファイルに分離して解決

```
% cat /etc/letsencrypt/post-hook
apachectl graceful
service postfix reload >/dev/null
```

(post-hook より renew-hook の方が望ましいがまだ変更していない)
(http://blog.n-z.jp/blog/2017-01-03-certbot-renew-hook.html 参照)

certbot (5/10)

- 2016-11-08 に `/etc/cron.daily/local-letsencrypt` は消した
- `/etc/cron.d/certbot` より `/lib/systemd/system/certbot.timer` が優先されるようになったため
  - ローカル時刻の 00:00:00 と 12:00:00 に自動実行

certbot (6/10)

- RandomizedDelaySec がきいていないが lilo.linux.or.jp では対処せず (https://bugs.debian.org/843607 参照)

certbot (7/10)

```
% cat /etc/systemd/system/certbot.service.d/diffmail.conf
[Service]
ExecStopPost=/bin/bash -c "diff -u
<(cut -d: -f4- /var/log/letsencrypt/letsencrypt.log.1
| egrep -v '^DEBUG') <(cut -d: -f4-
/var/log/letsencrypt/letsencrypt.log | egrep -v
'^DEBUG') | ifne mail -s 'Change certbot log' root"
```

(ExecStopPost は実際は1行)

certbot (8/10)

- service に直接 ExecStopPost を設定するのが良さそうにみえた
- 毎回変わる日時などが混ざる部分を除外して diff
- moreutils の ifne で diff があるときだけメール送信

certbot (9/10)

- 詳細は「Mini Debian Conference Japan 2016」での発表資料の「Certbotで無料TLSサーバー」 http://slide.rabbit-shocker.org/authors/znz/debian-certbot/ を参照

certbot (10/10)

- 前回の発表以降3回の更新あり
  - 2016-08-26 06:33
  - 2016-10-25 06:49
  - 2016-12-24 12:00
    - ここから rsa-key-size = 4096
- 有効期限が30日未満になったら自動更新で90日になるので約2ヶ月ごと

OCSP Stapling (1/4)

- OCSP (Online Certificate Status Protocol)
- 証明書の失効を確認するプロトコル
- その結果をステープル (ホッチキス) で止めるようにくっつけて、サーバー証明書と一緒に返す

OCSP Stapling (2/4)

apache 2.4 だと数行

```
SSLUseStapling          on
SSLStaplingResponderTimeout 5
SSLStaplingReturnResponderErrors off
SSLStaplingCache        shmcb:/var/run/ocsp(128000)
```

- それぞれ、有効にする、タイムアウト、成功の時だけ、キャッシュ設定

OCSP Stapling (3/4)

- キャッシュ設定は VirtualHost の外にする必要あり
- 他は VirtualHost の中でも可能
- 後述のサイトに合わせて全部外にした

OCSP Stapling (4/4)

- `openssl s_client -connect lilo.linux.or.jp:443 -status </dev/null` で確認可能
  - `OCSP response: no response sent` なら未設定
  - `OCSP Response Data:` が返ってきたら成功
  - https://www.cybertrust.ne.jp/sureserver/support/files/apache_ocsp.pdf 参照

SSL 設定全般 (1/5)

- https://mozilla.github.io/server-side-tls/ssl-config-generator/
- https://mozilla.github.io/server-side-tls/ssl-config-generator/?server=apache-2.4.10&openssl=1.0.1t&hsts=yes&profile=intermediate の設定を採用

SSL 設定全般 (2/5)

```
% cat /etc/apache2/conf-available/local-ssl.conf
# intermediate configuration, tweak to your needs
SSLProtocol             all -SSLv3

SSLCipherSuite          ECDHE-ECDSA-CHACHA20-POLY1305:
ECDHE-RSA-CHACHA20-POLY1305:ECDHE-ECDSA-AES128-GCM-SHA256:
ECDHE-RSA-AES128-GCM-SHA256:ECDHE-ECDSA-AES256-GCM-SHA384:
ECDHE-RSA-AES256-GCM-SHA384:DHE-RSA-AES128-GCM-SHA256:
DHE-RSA-AES256-GCM-SHA384:ECDHE-ECDSA-AES128-SHA256:
ECDHE-RSA-AES128-SHA256:ECDHE-ECDSA-AES128-SHA:ECDHE-RSA-AES256-SHA384:
ECDHE-RSA-AES128-SHA:ECDHE-ECDSA-AES256-SHA384:ECDHE-ECDSA-AES256-SHA:
ECDHE-RSA-AES256-SHA:DHE-RSA-AES128-SHA256:DHE-RSA-AES128-SHA:
DHE-RSA-AES256-SHA256:DHE-RSA-AES256-SHA:ECDHE-ECDSA-DES-CBC3-SHA:
ECDHE-RSA-DES-CBC3-SHA:EDH-RSA-DES-CBC3-SHA:AES128-GCM-SHA256:
AES256-GCM-SHA384:AES128-SHA256:AES256-SHA256:AES128-SHA:AES256-SHA:
DES-CBC3-SHA:!DSS
# (SSLCipherSuite は実際は1行)

SSLHonorCipherOrder     on
SSLCompression          off

# OCSP Stapling, only in httpd 2.3.3 and later
SSLUseStapling          on
SSLStaplingResponderTimeout 5
SSLStaplingReturnResponderErrors off
SSLStaplingCache        shmcb:/var/run/ocsp(128000)

# vim: syntax=apache ts=4 sw=4 sts=4 sr noet
```

SSL 設定全般 (3/5)

- a2enconf local-ssl で有効化
- VirtualHost の中

```
SSLEngine on
SSLCertificateFile      /etc/letsencrypt/live/lilo.linux.or.jp/fullchain.pem
SSLCertificateKeyFile   /etc/letsencrypt/live/lilo.linux.or.jp/privkey.pem
BrowserMatch "MSIE [2-6]" \
                nokeepalive ssl-unclean-shutdown \
                downgrade-1.0 force-response-1.0
# MSIE 7 and newer should be able to use keepalive
BrowserMatch "MSIE [17-9]" ssl-unclean-shutdown
Header always set Strict-Transport-Security "max-age=15768000"
```

  - (HSTS は Configuration Generator に合わせて1年から半年に変更)

SSL 設定全般 (4/5)

- https://www.ssllabs.com/ssltest/ での評価が「A-」(The server does not support Forward Secrecy with the reference browsers. Grade reduced to A-.) から「A+」に
- HSTS なしだと「A」

SSL 設定全般 (5/5)

- Mozilla SSL Configuration Generator で Modern にしても評価は変わらず、接続できるクライアントが大幅に減るので Intermediate にした
  - 自分の持っている Android 端末 (4.2.2) も Modern の対象外だったのも理由のひとつ

まとめ

- unattended-upgrades で自動更新
- HSTS で常時 https 化
- mailman の fix_url で http が残っていたのを修正
- certbot で自動更新の現状
- その他 SSL/TLS 設定見直し