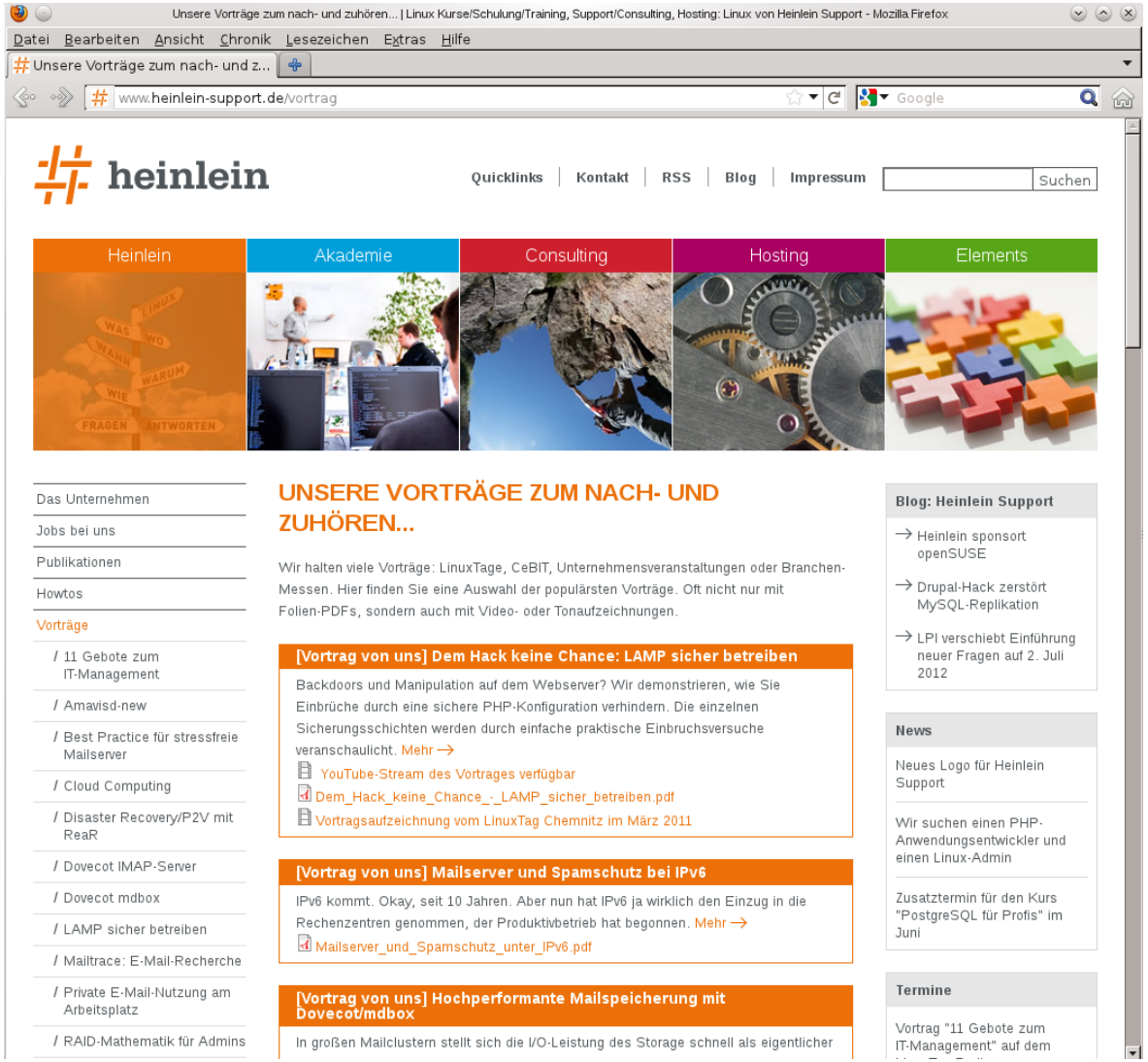
Silke Meyer <s.meyer@heinlein-support.de>

heinlein

Let‘s Encrypt im Alltag

heinlein

Let's Encrypt im Alltag
[SLAC 2016]
Silke Meyer <s.meyer@heinlein-support.de>



Ja, diese Folien stehen auch als PDF im Netz...
https://www.heinlein-support.de/vortrag

Was ist Let‘s Encrypt?

- Neue CA für SSL-/TLS-Zertifikate nach dem x.509-Standard
- Gemeinnützige „Internet Security Research Group“ (ISRG), USA
- Mitglieder u.a.: Mozilla, Electronic Frontier Foundation (EFF), Cisco, Uni Michigan, Akamai
- Finanziert durch Sponsoring
- Beta-Phase im April 2016 beendet
- Mitglied des CA/Browser Forums, aber noch keine eigene in den Browsern vorinstallierte CA.
- Zwischenlösung: Intermediate Zertifikate sind cross-signiert von IdenTrust.

Unterschiede zu anderen CAs

- Ziele
  - Ablösung von http als Standard durch https
  - Mehr Sicherheit soll kein Geld kosten
  - Schnelleres Rollover von kompromittierten Zertifikaten
- Plan zur Erreichung
  - Kostenloses Ausstellen von Zertifikaten
  - Gültigkeitsdauer von nur 90 Tagen
  - Volle Automatisierbarkeit über das extra entwickelte ACME-Protokoll (Automatic Certificate Management Environment)
- Open Source Software
  - Server: CA-Software „boulder“, Mozilla Public License 2.0
  - Client: offiziell empfohlener Client „Certbot“, Apache License 2.0

Funktionsumfang

- Aktuelle Features
  - Nur Domain-validierte Zertifikate nach dem x.509-Standard
  - Bis zu 100 Domainnamen pro Zertifikat (Subject Alternative Name)
  - Certificate Transparency
  - Staging-Umgebung zum Testen der Automatisierung (für uns)
- Geplante Features
  - Validierungsinfrastruktur spricht ipv6
  - IDN-Support
  - ECDSA Intermediate Zertifikate statt RSA (Feature???)

Funktionsumfang

- Fehlende Features
  - Zertifikate mit Organisations- oder erweiterter Validierung
  - Wildcard-Zertifikate (zunächst nicht geplant)
  - Signieren von Code oder E-Mails (anderer Zertifikatstyp)
  - Client-Zertifikate

Funktionsweise von Let‘s Encrypt

- Vertrauen in eine CA: Sie kann verifizieren, dass ein Bewerber der legitime Eigentümer einer Domain ist.
- althergebrachter Prozess dafür ist nicht automatisierbar
- Domain-Validierung mittels Challenge-Response-Verfahren automatisiert
- Keine Prüfung der „real-world identity“ (keine OV/EV)

Das ACME-Protokoll (Automatic Certificate Management Environment)

- Standardisierung der Kommunikation zwischen Client (z.B. Webserver) und Server (Zertifizierungsstelle)
- Domaininhaber-Verifizierung, Ausstellung, Erneuerung und Revocation von Zertifikaten
- Status: Internet Draft bei der Internet Engineering Task Force (IETF)
- JSON over HTTPS
- ACME-Server mit REST-Schnittstelle
- ACME-Client(s)

Das ACME-Protokoll (Automatic Certificate Management Environment)

- Der „User-Account“:
  - Client generiert asymmetrisches Schlüsselpaar für Kommunikation mit Server und signiert damit jeden Request. Er muss den Besitz des privaten Schlüssels *und* die Kontrolle über die Domain beweisen.
  - Server prüft jedesmal gegen den pubkey und autorisiert dieses Schlüsselpaar für bestimmte Domains.
- Replay Protection
  - Client fragt für jeden Request nach Einmalschlüssel („replay nonce“).
  - Server generiert einen, schickt ihn rüber und erwartet ihn in der Antwort.

Das ACME-Protokoll: Die REST-API

- Jeder ACME-Request enthält ein Feld für die angefragte Ressource :

Typen		Werte
New registration	→	new-reg
New authorization	→	new-authz
New certificate	→	new-cert
Revoke certificate	→	revoke-cert
Registration	→	reg
Authorization	→	authz
Challenge	→	challenge
Certificate	→	cert

Das ACME-Protokoll: Die REST-API

- 1. Kontaktaufnahme durch Client: Server schickt seine Ressourcen.

```
Sending GET request to https://acme-
v01.api.letsencrypt.org/directory

Received response […] Content: „new-authz“: „
https://acme-v01.api.letsencrypt.org/acme/new-authz
“, „new-cert“: [etc]
```

- Client kennt dann die URIs für die verschiedenen Anliegen.
- Jedes davon ist im Internet Draft beschrieben.

Das ACME-Protokoll: Domain-Validierung über Challenges

- Client: POST-Request an die new-authz URI
- Server schickt nächste URI („authz“) mit Token, dem Gesamtstatus „pending“ und drei Challenges, alle „pending“, z.B.:

```
"challenges": [
  {
    "type": "http-01",
    "uri": "https://example.com/authz/asdf/0",
    "token": "IlirfxKKXAsHtmzK29Pj8A"
  }
]
```

Das ACME-Protokoll: Domain-Validierung über Challenges

- Was kann man im eigenen Setup am besten automatisieren?
  - „http-01“: HTTP-Ressource zur Verfügung stellen
  - „dns-01“: TXT Record mit berechneter Prüfsumme aus Token und zusätzlichen Key Authorization String im Nameserver hinterlegen.
  - „tls-sni-01“: selbst signiertes Zertifikat mit genau zwei Subject Alternative Names, gehasht aus Token und Key Authorization String
- Client versucht, mind. einen der Challenges zu erfüllen.
  - `certbot.auth_handler:http-01 challenge for le2.mydomain.org`
    `certbot.plugins.webroot:Using the webroot path /var/www/le2.mydomain.org for all unmatched domains.`
    `certbot.plugins.webroot:Creating root challenges validation dir at /var/www/le2.mydomain.org/.well-known/acme-challenge`
    `certbot.plugins.webroot:Attempting to save validation to /var/www/le2.mydomain.org/.well-known/acme-challenge/j7191AU6-K5MnrsYLoyCwsiHCU85b3NUT0QH2yBdU_Y`

Das ACME-Protokoll: Domain-Validierung über Challenges

- Bei Erfolg meldet Server „valid“ für entsprechenden Challenge
- Client erstellt Certificate Signing Request (CSR)
- Server antwortet mit dem Zertifikat und der URI, unter der aktualisierte Versionen des Zertifikats abgeholt werden können.
- Client vermerkt, welche Parameter für welche Domains mitgegeben wurden zwecks Wiederverwendung.

ACME-Clients: Certbot

- Certbot ist eine ACME-Client-Implementierung der Electronic Frontier Foundation (EFF)
- Nachfolge der Referenz-Implementierung „letsencrypt“
- Kompatibel zu allen CAs, die ACME sprechen
- Python
- Pakete: Debian 8 (Backports), Ubuntu 16.04, Arch Linux, Fedora 23, CentOS/RHEL 7 u.a.
- Interaktives Webtool
- Command Line Interface mit man page

ACME-Clients: Certbot-Plugins

- „Authenticators“: keine Autoconfigs, nur Zertifikatserstellung
- Optionale „Installers“ übernehmen die Konfiguration des Webservers
  - Apache (Autoconfig mit optionalem Redirect auf SSL)
  - Nginx, experimentell
  - 3rd-Party-Plugins vorhanden oder in Arbeit (z.B. Plesk, Haproxy, Postfix)
  - Standalone (startet eigenen Webserver für Domain-Validierung)

ACME-Clients: Certbot-Benutzung

- Hier: Nutzung des Webroot-Plugins für den http-01-Challenge.
- A / AAAA Record im DNS ist bereits konfiguriert.
- `certbot certonly --non-interactive --text --agree-tos --webroot --webroot-path /var/www/mydomain.org --e-mail ich@mydomain.org --domain mydomain.org --domain www.mydomain.org`
- mehrere Webroot-Path- und Domainangaben hintereinander reihen
- Erneuerung: `certbot renew`
- erstellt Ordnerstruktur unterhalb von /etc/letsencrypt

ACME-Clients: Certbot-Benutzung

- Erneuerung mit Cronjob: `certbot renew`
- ```
  Processing /etc/letsencrypt/renewal/le1.mydomain.org.conf
  Processing /etc/letsencrypt/renewal/le2.mydomain.org.conf
  The following certs are not due for renewal yet:
    /etc/letsencrypt/live/le1.mydomain.org/fullchain.pem
  (skipped)
    /etc/letsencrypt/live/le2.mydomain.org/fullchain.pem
  (skipped)
  No renewals were attempted.
  ```
- /etc/letsencrypt/renewal enthält für jede Domain die zuletzt verwendeten Parameter.
  ```

ACME-Clients: Alternativen zu Certbot

- Schnell wachsende Anzahl von Client-Implementierungen in divesen Programmiersprachen
- Beispiele:
  - Statische html-Seite https://gethttpsforfree.com
  - Caddy: HTTP/2-Webserver mit eingebautem ACME-Client und standardmäßigen Let's Encrypt-Zertifikaten (Apache 2.0 License)
  - diverse Shell-Skripte für bash oder zsh

Automatisierung von Let‘s Encrypt im Shared Hosting

- Offizieller Integration Guide
- Wir als Hoster sind „Subscriber“ von Let‘s Encrypt (nicht die Endkunden)
- Es ist möglich, einen Account-Key für alle Kunden zu haben oder einen pro Kundin.
- Empfehlung zum Einsatz von Let‘s Encrypt-Clients in Docker Containern: Zertifikate und Keys auf persistentem Speicher vorhalten (Rate-Limits bei Neuausrollen vermeiden)

Silke Meyer <s.meyer@heinlein-support.de>

Automatisierung von Let‘s Encrypt im Shared Hosting

- Let‘s Encrypt-Client auf dem Server, der die Document Roots der Kundendomains hält.
- Client: eins der auf letsencrypt.org verlinkten Bash-Skripte
- zwei Wrapper-Skripte in Bash und Python
- alle Domain-Konfigs in zentralem Key-Value-Store (OpenLDAP)
- neues LDAP-Attribut für Domains „letsencrypt“

Silke Meyer <s.meyer@heinlein-support.de>

Automatisierung von Let‘s Encrypt im Shared Hosting

- Ordnerstruktur im Dateisystem liefert Parameter für Skripte:
  - Kundenaccount
  - Zertifikate, Keys, Certificate-Chains mit Timestamps
  - Domainspezifische Konfig (Pfad zum .well-known-Verzeichnis, DN der Domain, Kontaktadresse)
  - Datei der zu registrierenden Subject Alternative Names
- Bash-Skript parst LDAP nach Attribut „letsencrypt“, schreibt obige Dateien, ruft Let‘s Encrypt-Client auf
- Python-Skript schreibt Zertifikate und SSL-Redirects ins LDAP.
- Cronjob für Deployment der Webserver-Konfigs aus LDAP
- Cronjob für Erstellung / Erneuerung der Zertifikate

Automatisierung von Let‘s Encrypt im Shared Hosting

- Beachtung besonderer Kunden-Konfigurationen
  - well-known-URIs müssen global über http (!) aufrufbar sein, von SSL-Redirects ausnehmen
  - Webserver muss das Aufrufen versteckter Verzeichnisse erlauben
  - Kundendomains mit .htpasswd-Schutz? Globale Ausnahme für well-known-URI nötig

- Danke für das Interesse!
- Natürlich und gerne stehe ich Ihnen jederzeit mit Rat und Tat zur Verfügung und freue mich auf neue Kontakte.
  - Silke Meyer
  - Mail: s.meyer@heinlein-support.de
  - Telefon: 030/40 50 51 - 51
- Wenn's brennt:
  - Heinlein Support 24/7 Notfall-Hotline: 030/40 505 - 110

Wir suchen:

Admins, Consultants, Trainer!

Wir bieten:

Spannende Projekte, Kundenlob, eigenständige Arbeit, keine Überstunden, Teamarbeit

...und natürlich: Linux, Linux, Linux...

http://www.heinlein-support.de/jobs

Silke Meyer <s.meyer@heinlein-support.de>

heinlein

Heinlein Support hilft bei allen Fragen rund um Linux-Server

HEINLEIN AKADEMIE

Von Profis für Profis: Wir vermitteln die oberen 10% Wissen: geballtes Wissen und umfangreiche Praxiserfahrung.

HEINLEIN HOSTING

Individuelles Business-Hosting mit perfekter Maintenance durch unsere Profis. Sicherheit und Verfügbarkeit stehen an erster Stelle.

HEINLEIN CONSULTING

Das Backup für Ihre Linux-Administration: LPIC-2-Profis lösen im CompetenceCall Notfälle, auch in SLAs mit 24/7-Verfügbarkeit.

HEINLEIN ELEMENTS

Hard- und Software-Appliances und speziell für den Serverbetrieb konzipierte Software rund ums Thema eMail.