Wildcardzertifikate mit Let's Encrypt

PC-Treff-BB
Roland Egeler

Über diesen Vortrag

- Anschlussvortrag an „Let‘s Encrypt“ [Vorgänger]
- Es wird vorgestellt, wie man ein TLS-Zertifikat erhalten kann, das für alle Subdomains einer Domain gilt
- Zertifikate gratis bei „Let‘s Encrypt“ [Let's Encrypt] bei Nachweis der Kontrolle über Domain
- Erhalt des Zertifikats, ohne höhere Rechte zu brauchen
- Vor- und Nachteile von Wildcardzertifikaten

Warum Zertifikate?

- Zertifikate ermöglichen verschlüsselten Datenverkehr im Internet
- Stichwort TLS: Transport Layer Security
- Benutzt hauptsächlich für Webserver (HTTPS) und E-Mail (IMAPS und SMTPS)
- Browser bemängeln unverschlüsselte Webseiten immer häufiger
- Auch wichtig für Suchmaschinenoptimierung (SEO)
- Extrem wichtig, wenn Passwörter im Spiel sind, da diese bei HTTP unverschlüsselt durch's Netz gehen

Warum Wildcardzertifikate?

- „Normale“ Zertifikate gelten nur für spezifische Domains (Bsp.: www.example.org oder example.org)
- Wildcardzertifikate gelten für alle Subdomains einer Domain (Bsp.: *.example.org)
- Gelten nicht für Hauptdomain (hier example.org)
- Wildcardzertifikate können benutzt werden, um Dienste hinter der Hauptdomain gezielt anzusprechen
- Beispiele:
  - mail.example.org
  - nextcloud.example.org
  - minecraft.example.org

Wie erhält man ein Zertifikat bei „Let‘s Encrypt“?

- „Challenge“-Verfahren: Man muss nachweisen können, dass man die Kontrolle über die Domain hat
- Fünf Schritte:
  - Anfrage nach Zertifikat bei „Let‘s Encrypt“
  - „Let‘s Encrypt“ stellt Aufgabe
  - Informationen aus Aufgabe werden in Domain integriert
  - „Let‘s Encrypt“ findet die Daten innerhalb der Domain
  - Zertifikat wird erstellt und unterschrieben
- Zertifikat wird weltweit akzeptiert, da die großen Browser bzw. Betriebssysteme der ausstellenden CA (Certificate Authority) vertrauen

Automatisierung der Zertifikatsanfrage

- „Let‘s Encrypt“ stellt momentan nur Zertifikate mit einer Laufzeit von drei Monaten aus
- Zertifikat muss also alle drei Monate erneuert werden
- Fehleranfällig und lästig bei händischem Verfahren
- Anforderung nach Automatisierung
- Verlangt hohe Rechte (mindestens Webserver, evtl. root)
- Lädt automatisch neue Versionen nach
- Verlangt Vertrauen in Implementierer des „ACME“-Protokolls
- Will man das?

Methoden zu Erhalt von Zertifikaten

- HTTP-01
  - Es muss an einem bekannten Pfad eine Datei mit einem zufälligen Namen und Inhalt erstellt werden
- TLS-SNI-01 (-02)
  - Es wird per „Server Name Indication“ ein Zertifikat angefordert, das einen zufälligen Text enthält
- DNS-01
  - Der „TXT Record“ einer bestimmten Subdomain muss einen zufälligen Text enthalten
- TLS-ALPN-01
  - Erweiterung des „TLS“-Protokolls, noch nicht implementiert

HTTP-01

- Serverbetreiber beweist Kontrolle über Seiteninhalte
- Für diese Zwecke festgelegter Pfad:
  - „$WEB_ROOT/.well-known/acme-challenge“
- „Let‘s Encrypt“ Challenge (Herausforderung):
  - Eine Datei mit einem langen zufälligen Namen
  - Darin ein zufälliger Inhalt
- Ist diese Datei mit dem korrekten Inhalt vorhanden, ist die Kontrolle erwiesen
- Das Zertifikat wird dann ausgestellt

HTTP-01

- Vorteile:
  - Funktioniert ohne vorhandenes Zertifikat (Initialfall)
  - Einfaches Verfahren
  - Lässt sich automatisieren
- Nachteile:
  - Port 80 (HTTP) muss geöffnet werden (unverschlüsselt)
  - Es muss daran gedacht werden, Port 80 wieder zu schließen
  - Wahlweise: Eigener Abschnitt in Serverkonfiguration, der „.well-known/acme-challenge“ von Standardkonfiguration trennt
  - Port 80 wird dann ausschließlich für Zertifikate benutzt
  - Kein Sicherheitsproblem, da nur einmal benutzte Zufallsdaten

TLS-SNI-01 (-02)

- Serverbetreiber beweist Kontrolle durch Ausstellung eines Zertifikats mit zugewiesenem Inhalt
- Hat das Zertifikat den korrekten Inhalt, ist die Kontrolle erwiesen
- Warum nicht „HTTPS-01“?
- Mehrere Server hinter einer IP-Adresse (bei Providern)
- Beim Verbindungsaufbau mit TLS wird das Zertifikat bereits benötigt, man weiß aber noch nicht welches
- TLS-Erweiterung „SNI“ enthält angeforderte URL
- In allen wichtigen Webservern implementiert (apache, nginx, ...)

TLS-SNI-01 (02)

- Vorteile:
  - Benutzt verschlüsselte Kommunikation
  - Lässt sich automatisieren
- Nachteile:
  - Ermöglicht bei manchen Servern jedem Domainbesitzer die Ausstellung von Zertifikaten für andere Domains
  - Problem tritt auf, wenn viele Domains unter einer IP-Adresse erreichbar sind und gleichzeitig beliebige Zertifikate ohne Prüfung hochgeladen werden können
  - „TLS-SNI-02“ war Anpassung auf „ACME v2“
  - „Let's Encrypt“ hat nichts falsch gemacht
  - Wird nicht mehr von „Let‘s Encrypt“ angeboten [TLS-SNI]

DNS-01

- Serverbetreiber beweist Kontrolle über Domain
- Anlage von Subdomains muss möglich sein
  - Entweder eigenen DNS betreiben
  - Oder der Provider (bzw. Vertrag) muss Anlage von Subdomains zulassen
- In der Subdomain „_acme-challenge“ muss im „TXT Record“ die zugewiesene Information stehen
- Eigenkontrolle z.B. über „DNSWatch“ [DNSWatch]
- Kommandozeile (z.B.):
  - nslookup -type=txt _acme-challenge.example.org
- Zertifikat wird ausgestellt, wenn die Informationen übereinstimmen

DNS-01

- Vorteile:
  - Seiteninformationen müssen nicht verändert werden
  - Verschlüsseltes Protokoll
  - Lässt Wildcardzertifikate zu
- Nachteile:
  - Provider muss Subdomains zulassen
  - Normalerweise keine Automatisierung möglich
    - Provider müsste API haben
    - Oder man betreibt DNS-Server selbst
  - Man muss Vertrauen haben, dass der Provider seine Zugangsdaten sicher verwahrt

TLS-ALPN-01

- Serverbetreiber beweist Kontrolle über benutzerdefiniertes Protokoll
- ALPN: Application Level Protocol Negotiation
- Erweiterung des TLS-Protokolls
- Eingeführt als Nachfolger von „TLS-SNI-0X“
- Soll die dort auftretenden Probleme vermeiden
- Bisher implementiert kein Webserver dieses Protokoll
- Daher noch nicht nutzbar

TLS-ALPN-01

- Vorteile:
  - ?
- Nachteile:
  - ?

Zwischenbilanz

- Automatisierung nicht wünschenswert, da fremder Code mit hohen Rechten ausgeführt wird
- HTTP-01 erfordert offenen Port 80 (nicht empfohlen)
- TLS-SNI-0X wird nicht mehr angeboten
- TLS-ALPN-01 ist noch nicht implementiert
- Bleibt DNS-01
- Bonus: Kann Wildcardzertifikate

ACME-Client

- Wird zur Zertifikatsanforderung benutzt
- ACME: Automatic Certificate Management Environment
- Sehr viele verschiedene Implementierungen [Clients]
- Benutzen unterschiedliche Skriptsprachen
- Nicht alle können DNS-01 (braucht „ACME v2“)
- Offizieller Client: „certbot“ (von EFF, braucht „python“)
- Ausgewählt: „acme.sh“ (nur shell-Skript) [acme.sh]

Keine hohen Rechte

- Es wird ein eigener Benutzer dafür angelegt
- Man braucht das shell-Skript nur aus gitHub auschecken
- Kommandozeilenbefehle:
  - wget 'https://github.com/Neilpang/acme.sh/archive/master.zip'
  - unzip master.zip
  - cd acme.sh-master/
  - ./acme.sh --issue -d "*.example.org" --dns --yes-I-know-dns-manual-mode-enough-go-ahead-please
- Erneuerung mit „--renew“
- Daten liegen dann unter „~/.acme.sh/*.example.org“
- Probleme in der Kommandozeile wegen „*“ im Namen

Beispiel am Provider „Strato“

- Siehe [Strato]
- Strato hat eine Weboberfläche zur Konfiguration
- Navigation zur Domain
  - Domains verwalten
  - verwalten
  - DNS Einstellungen
  - TXT Records inklusive SPF und DKIM Einstellungen
- Dort einen neuen TXT-Record anlegen
- Präfix: „_acme-challenge“
- Wert: Informationen aus „DNS-01“-Challenge

Erfolgserlebnis

- Nach erneutem Ausführen von „acme.sh“ wird der Zufallswert geprüft
- Bei Übereinstimmung wird das neue Zertifikat ausgestellt
- Nacharbeiten:
  - Herausfinden, welche Zertifikatsdateien der eingesetzte Server benutzt
  - Evtl. Sicherungskopie der alten Dateien
  - Neue Dateien an die geeignete Stelle kopieren
  - Neustart des Servers
  - Aufrufen der Webseite (evtl. neu laden)
  - Überprüfen der Zertifikatsdaten

Ausblick

- Auswahl der lokalen Server nach Subdomain
- Konfiguration nicht trivial
- Zu verwendende Server:
  - apache
  - nginx
- Wird nachgeliefert...

Vielen Dank!

Quellen

- Let‘s Encrypt: https://letsencrypt.org
- Vorgänger: http://www.pc-treff-bb.de/Vortraege/Lets_Encrypt.pdf
- DNSWatch: https://www.dnswatch.info
- acme.sh: https://github.com/Neilpang/acme.sh
- Clients: https://letsencrypt.org/docs/client-options/#acme-v2-compatible-clients
- Strato: https://www.strato.de
- TLS-SNI: https://www.heise.de/newsticker/meldung/Letsencrypt-sperrt-TLS-SNI-Domainvalidierung-3938738.html

Weitere Quellen

- https://community.letsencrypt.org/t/2018-01-09-issue-with-tls-sni-01-and-shared-hosting-infrastructure/49996
- https://letsencrypt.org/docs/challenge-types/
- https://tools.ietf.org/html/draft-ietf-acme-tls-alpn-01
- https://tools.ietf.org/html/rfc7301
- https://certbot.eff.org/docs/challenges.html?highlight=http
- https://letsencrypt.org/how-it-works/
- https://labs.detectify.com/2018/01/12/how-i-exploited-acme-tls-sni-01-issuing-lets-encrypt-ssl-certs-for-any-domain-using-shared-hosting/

Weitere Quellen

- https://community.letsencrypt.org/t/comparison-of-acme-challenges/38864
- https://letsencrypt.org/docs/faq/
- https://community.letsencrypt.org/t/acme-v2-production-environment-wildcards/55578
- https://de.wikipedia.org/wiki/Let%E2%80%99s_Encrypt
- https://community.letsencrypt.org/t/2018-01-11-update-regarding-acme-tls-sni-and-shared-hosting-infrastructure/50188
- ...