Wie verschlüssele ich die Kommunikation mit meiner Website?

Intention

Besuchern der eigenen Website sollte man ein sicheres Umfeld bereitstellen. Daher ist das Anbieten einer verschlüsselten Kommunikation ein Muss. Dieses Pattern geht darauf ein, wie man ein *SSL / TLS Zertifikat* erhält und dieses in die eigene Seite einbindet.

Problemstellung

Eine sichere Verbindung zwischen dem *Webbrowser* der Seitenbesuchers und der eigenen Website anzubieten ist gerade beim Austausch personenbezogener Daten (z.B. Namen, Adressen oder Bankdaten) zwingend erforderlich. Die verschlüsselte Kommunikation garantiert Datenintegrität, gewährt Sicherheit und ist für einige Web-Applikationen sogar Voraussetzung. Obwohl viele Seiten mit personenbezogenen Daten arbeiten, nutzen heutzutage nur 40% der Websites eine verschlüsselte Kommunikation.

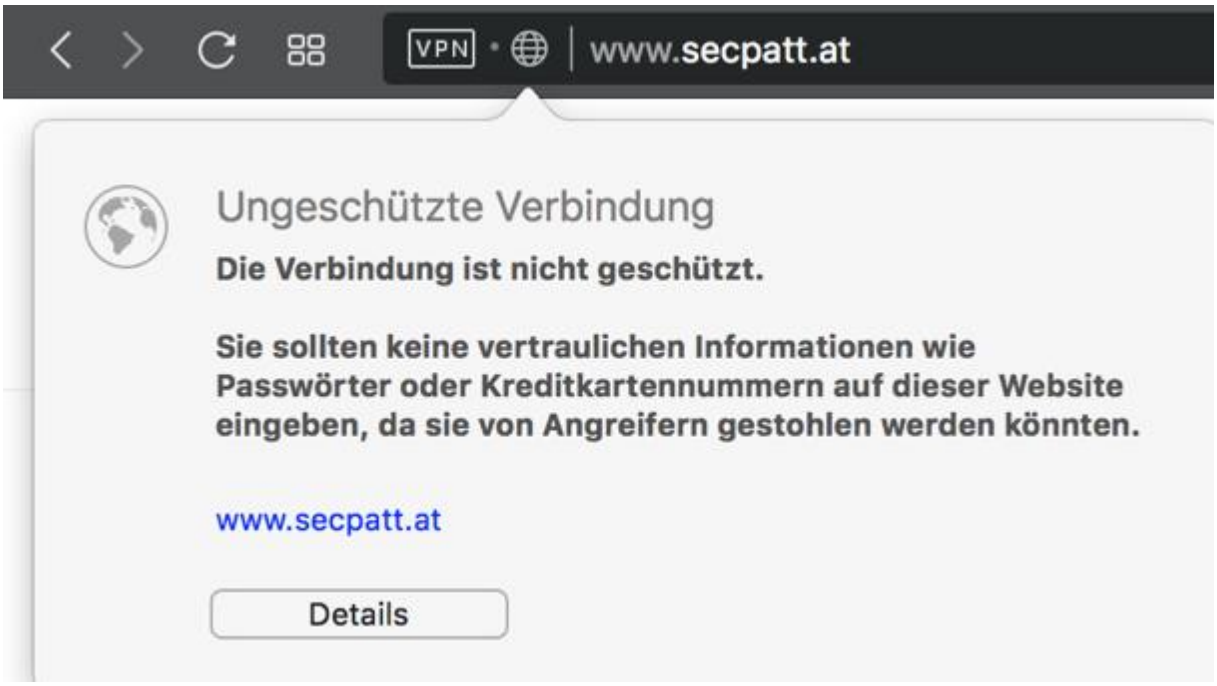
Szenario

Die eigene Website stellt keine verschlüsselte Verbindung bereit, dies soll geändert werden.

Lösung

Um HTTPS auf einer Website zu aktivieren ist es zu aller erst notwendig ein SSL / TLS Zertifikat von einer Zertifizierungsstelle (*Certificate authority*) zu erhalten. Nach Erhalt des Zertifikats muss dieses in die Seite eingebunden werden. Dies geschieht in der Regel im Admin-Panel des *Hosters*. Nachdem das Zertifikat eingebunden ist, empfiehlt es sich Einstellungen und Anpassungen zu unternehmen um nur mehr verschlüsselte Verbindungen zur Website zu erlauben. Daraufhin sollte mit verschiedenen Browsern getestet werden, ob nur mehr eine verschlüsselte Verbindung mit der Website möglich ist. Um herauszufinden ob eine Website eine sichere Verbindung nutzt, sollte man in gängigen Browser nach einem grünen Schloss neben der URL Ausschau halten.
Wie man im folgenden Bild sieht, wird bei einer verschlüsselten Verbindung ein grünes Schlosssymbol angezeigt:

Bei ungeschützten Verbindungen erscheint kein grünes Schloss Symbol:



Es gibt eine Handvoll von Zertifizierungsstellen die SSL / TLS Zertifikate ausstellen. Dieses Pattern nimmt die Zertifizierungsstelle https://letsencrypt.org, welche ihre Zertifikate gratis, automatisiert und offen anbietet als Beispiel.

Beispiele

Let's Encrypt SSL-Zertifikat erhalten

- Man besucht die Website https://letsencrypt.org und klickt auf den Button „Get Started".

- Auf der Nächsten Seite steht erklärt wie man in welchem Fall vorgehen sollte.

Dieses Pattern erläutert zwei Methoden, die genutzt werden können um das Zertifikat zu erhalten und einzubinden. Die erste Methode benötigt Shell-Zugriff (u.a. auch Terminal- bzw. Kommandozeilen-Zugriff genannt) und die benötigten Rechte. Methode Nummer zwei benötigt keinen Shell-Zugriff, ist dadurch jedoch umständlicher und gestaltet sich komplizierter. Bei https://letsencrypt.org ist es notwendig zu beweisen, dass man der Betreiber der Domain ist. Dies geschieht folgendermaßen:

- Man hat **Zugriff auf die Shell**
  - Man geht auf https://certbot.eff.org.
  - Wählt die Software und das System die der Server benutzt.
  - Der weitere Verlauf wird auf der Seite (englisch) erklärt.
  - Sofern Certbot nicht den eigenen Vorstellungen entsprechen sollte, besteht die Möglichkeit sich aus folgender Liste weitere ACME v2 kompatible Programme (Clients) auszuwählen:
    - https://letsencrypt.org/docs/client-options/

- Man hat **keinen Zugriff auf die Shell** aber …
  - der Hoster unterstützt Let's Encrypt
    - Der einfachste Weg in diesem Fall ist das Kontaktieren des Hosters. Dieser sollte sich normalerweise darum kümmern, das Zertifikat von Let's Encrypt zu erhalten.
  - der Hoster unterstützt Let's Encrypt nicht
    - Sofern der Hoster Let's Encrypt nicht unterstützen sollte, gibt es die Möglichkeit das Zertifikat manuell zu erhalten. In diesem Fall wird Certbot auf dem eigenen Computer installiert und ausgeführt. Daraufhin erhält man eine Datei, welche auf den Webserver hochgeladen werden muss. Da dieser Prozess aufwändig ist und jedes Mal wiederholt werden muss wenn das Zertifikat abläuft, ist dies nicht zu empfehlen.
      - https://certbot.eff.org/docs/using.html#manual

- Man bittet den Hoster darum Let's Encrypt zu unterstützen.
- Zu einem Hoster wechseln, der Let's Encrypt unterstützt.

Eine detailreiche Erklärung für Let's Encrypt auf Englisch ist unter folgender URL zu finden: https://letsencrypt.org/getting-started/. Da das Einbinden von Let's Encrypt meist über das Admin-Panel des Hoster geschieht, empfiehlt es sich bei Unklarheiten oder Anleitungen den Kontakt mit dem Hosting-Anbieter aufzunehmen.

Herausfinden ob ein Hoster Let's Encrypt unterstützt
Um herauszufinden ob ein Hoster „Let's Encrypt" unterstützt oder eine andere Möglichkeit anbietet um ein SSL Zertifikat zu generieren, bietet es sich an https://www.webhostingvergleich.eu/at/oesterreich/ einen Vergleich anzustellen. In diesem wird aufgezeigt, ob und in welcher Form eine SSL-Verschlüsselung bei einem Hoster möglich ist. Als Alternative ist es sinnvoll sich die Angebote von diversen Hosting-Anbieter direkt auf deren Website anzusehen.

TELEMATICA Tarif: **Web Basic**	**Österreich**	50 GB	**SSL** kostenlos Let's encrypt	**PHP** 7.2	**MySQL** 10 Datenbanken (MariaDB)
webgo Tarif: **Mach 2**[1]	**Deutschland**	200 GB	**SSL** ab 5,90 € / Monat	**PHP** 7.1	**MySQL** 25 Datenbanken (SSD)

Wie man einen sicheren Hoster findet, wird im Pattern „Was ist ein sicherer Hoster und wie wähle ich diesen aus?" beschrieben.

HTTPS auf der eigenen Seite nutzen
Es gibt mehrere Möglichkeiten eine verschlüsselte Kommunikation mit der Seite zu ermöglichen. Einerseits kann man direkt in die .php Dateien folgende Codezeilen einfügen um beim Aufrufen eben jener Seite explizit HTTPS zu verwenden:

```
// Require https


if ($_SERVER['HTTPS'] != "on") {


    $url = "https://". $_SERVER['SERVER_NAME'] . $_SERVER['REQUEST_URI'];
```

```
    header("Location: $url");


    exit;


}
```

Wenn eine sichere Verbindung für jede Unterseite der Website gelten soll, ist es möglich die .htaccess Datei anzupassen. Diese ist üblicherweise auf der obersten Ebene des Servers (Root, wenn ein Apache Server verwendet wird) zu finden. Um eine sichere Verbindung zu erzwingen und alle HTTP-Aufrufe direkt auf HTTPS umzuleiten, kann folgender Code in die .htaccess Datei eingefügt werden:

```
# HTTP to HTTPS redirecting



RewriteEngine On

RewriteCond %{HTTP_HOST} ^example\.com [NC]
RewriteCond %{SERVER_PORT} 80
RewriteRule ^(.*)$ https://www.example.com /$1 [R=301,L]
```

Statt example.com sollte natürlich die eigene Webadresse eingetragen werden.

HSTS Header mit Apache Server
Wenn eine sichere Verbindung erzwungen werden soll, ist es möglich die .htaccess Datei auch folgendermaßen anzupassen:

```


# Use HTTP Strict Transport Security to force client to use secure
connections only

Header set Strict-Transport-Security "max-age=3600" env=HTTP
```

HTTPS Server mit NGINX einrichten
http://nginx.org/en/docs/http/configuring_https_servers.html#

Referenzen

Mehr **Informationen zu SSL und TLS** findet man auch hier:
https://www.ssllabs.com
https://developers.google.com/web/fundamentals/security/encrypt-in-transit/why-https
https://developers.google.com/web/progressive-web-apps/
https://www.w3.org/2001/tag/doc/web-https
https://mod-rewrite-cheatsheet.com/#basics-enable-htaccess
https://www.ssllabs.com

Keywords

HTTPS, Kryptografie, Verschlüsselung, SSL, TLS