



وزارت ارتباطات و فناوری اطلاعات
سازمان فناوری اطلاعات ایران

معاونت امنیت فضای تولید و تبادل اطلاعات

نحوه‌ی پیکربندی امن SSL/TLS بر روی وب سرور آپاچی



فهرست مطالب

۱	مقدمه	۱
۱	موارد پیشنهادی برای ارتقای امنیت	۲
۲	تنظیم الگوریتم‌های قدرتمند و Forward secrecy	۲-۱
۳	به‌روزرسانی نرم‌افزارها و نسخه‌ها	۲-۲
۳	فعال کردن OCSP Stapling	۲-۳
۴	فعال کردن HSTS	۲-۴
۴	فعال کردن HPKP	۲-۵
۵	مراجع	۳

۱ مقدمه

پروتکل‌های SSL و TLS جهت امن کردن ارتباط میان کاربر و سرور از طریق تصدیق هویت، رمزنگاری و صحت طراحی و پیاده‌سازی شده است. جهت امن کردن داده‌ها این پروتکل‌ها از cipher suite هایی استفاده می‌کنند. هر cipher suite ترکیبی از الگوریتم‌های اصالت‌سنجی، رمزنگاری و کد تصدیق هویت پیغام (MAC) است. در زمان پیکربندی TLS/SSL باید تنظیمات به‌درستی انجام شده و cipher suite های امن مورد استفاده قرار گیرد. برخی از مهم‌ترین این تنظیمات شامل غیرفعال کردن SSL 2.0 و SSL 3.0، غیرفعال کردن TLS 1.0 Compression و cipher suite های ناامن و استفاده از آخرین نسخه‌ی نرم‌افزارها است. پیکربندی ارائه شده بر روی سروری با مشخصات زیر انجام شده است.

نام نرم‌افزار	نسخه‌ی مورد استفاده
سیستم عامل	SMP Debian 4.0.4-1+kali2 (2015-06-03)
OpenSSL	OpenSSL 1.0.1k 8 Jan 2015 built on: Thu Dec 3 18:28:10 2015
Apache	Server version: Apache/2.4.10 (Debian) Server built: Aug 1 2015 21:26:38

جهت امن‌سازی می‌توان از تغییر قابل پیکربندی قرار گرفته در مسیرهای زیر استفاده کرد:

/etc/apache2/mods-enabled/ssl.conf

/etc/apache2/sites-enabled/default-ssl.conf

/etc/apache2/sites-enabled/default.conf

۲ ارزیابی وضعیت فعلی سرویس دهنده

برای ارزیابی وضعیت امنیتی SSL/TLS در سرویس دهنده خود از سرویس زیر استفاده نمایید:

<https://sslcheck.certcc.ir/>

پس از انجام موارد امنیتی زیر مجدداً با استفاده از آدرس‌های فوق سرویس خود را پویش کنید تا از برطرف شدن مشکلات موجود مطمئن شوید.

۳ موارد پیشنهادی برای ارتقای امنیت

۱-۳ تنظیم الگوریتم‌های قدرتمند و Forward secrecy

در صورتی که بخواهید از الگوریتم‌های قدرتمند استفاده کنید باید خطوط زیر در فایل پیکربندی ssl.conf اضافه و یا ویرایش گردد. با پیکربندی SSLCipherSuite به صورت زیر می‌توان آن را بهینه و امن کرد:

```
SSLProtocol all -SSLv2 -SSLv3
```

```
SSLHonorCipherOrder on
```

با پیکربندی SSLCipherSuite به صورت زیر می‌توان آن را امن کرد:

```
SSLCipherSuite  
HIGH:EECDH+ECDSA+AESGCM:EECDH+aRSA+AESGCM:EECDH+ECDSA+SHA  
384:EECDH+ECDSA+SHA256:EECDH+aRSA+SHA384:EECDH+aRSA+SHA256:EE  
CDH+aRSA+RC4:EECDH:EDH+aRSA:RC4:!aNULL:!eNULL:!LOW:!3DES:!MD5:!E  
XP:!PSK:!SRP:!DSS:!RC4:!MEDIUM
```

برای محدودتر کردن یک url خاص (مانند /strong/area/) به نحوی که سرور به ازای آن url از cipherهای امن استفاده کند. باید به صورت زیر عمل کرد:

```
<Location "/strong/area">  
# but https://hostname/strong/area/ and below  
# requires strong ciphers  
SSLCipherSuite  
HIGH:EECDH+ECDSA+AESGCM:EECDH+aRSA+AESGCM:EECDH+ECDSA+SHA  
384:EECDH+ECDSA+SHA256:EECDH+aRSA+SHA384:EECDH+aRSA+SHA256:EE  
CDH+aRSA+RC4:EECDH:EDH+aRSA:RC4:!aNULL:!eNULL:!LOW:!3DES:!E  
XP:!PSK:!SRP:!DSS:!RC4:!MEDIUM</Location>
```

حداقل نیازمندی‌های لازم برای پیکربندی Forward secrecy به صورت زیر است:

- OpenSSL 1.0.1c+
- Apache 2.4x

پیکربندی این قابلیت، نیاز به دو مرحله دارد:

۱. پیکربندی وب سرور به نحوی که به صورت فعال suiteها را انتخاب کند

۲. پیکربندی رشته Suiteها

با انجام مراحل قبل برای تنظیم الگوریتم‌های قدرتمند، Forward secrecy نیز فعال می‌شود.

۲-۳ به‌روزرسانی نرم‌افزارها و نسخه‌ها

برخی از آسیب‌پذیری‌ها موجود بر روی SSL/TLS مربوط به نسخه‌های نرم‌افزارهای مورد استفاده است. به‌عنوان نمونه آسیب‌پذیری heartbleed بر روی نسخه‌های قدیمی openssl وجود دارد. این نرم‌افزار باید به آخرین نسخه ارتقا پیدا کند. دستور زیر منجر به به‌روزرسانی نسخه‌های نرم‌افزار شما می‌گردد:

```
apt-get upgrade openssl
```

۳-۳ فعال کردن OCSP Stapling

روشی برای بالا بردن سرعت در چک کردن لیست ابطال کلید برای گواهی است. با استفاده از OCSP Stapling نیاز نیست که سرویس گیرنده درخواستی را به سرور OCSP بدهد و با استفاده از اطلاعات مهیا شده همراه گواهی، می‌تواند از باطل نبودن گواهی اطمینان حاصل کند. جهت فعال‌سازی این قابلیت باید خطوط زیر به فایل پیکربندی اضافه گردد. باید دقت گردد که این قابلیت بر روی نسخه‌های بالاتر از 2.3.3 قابل فعال‌سازی خواهد بود. این خط باید فایل پیکربندی قرار گرفته در مسیر زیر را ویرایش کرد:

```
SSLUseStapling On
SSLStaplingCache shmcb:/tmp/ssl_stapling(32768)
SSLStaplingResponderTimeout 5
SSLStaplingReturnResponderErrors off
```

در صورتی که از فایل‌های پیکربندی مانند default-ssl.conf قرار گرفته در مسیرهای دیگر استفاده می‌شود باید توجه گردد که SSLStaplingCache حتماً بیرون از تگ VirtualHost باشد. بقیه‌ی پارامترها می‌توانند در درون این تگ قرار بگیرند. البته قبل از آن باید مطمئن بود که Intermediate Certificate به‌درستی نصب شده است. باید مسیر این certificate مشابه زیر در فایل پیکربندی قرار داده شده باشد.

```
SSLCACertificateFile /etc/ssl/ca-certs.pem
```

برای نصب Intermediate Certificate می‌توان از دستورات زیر استفاده کرد، سپس مسیر را در فایل پیکربندی مشابه بالا قرار داد.

```
cd /etc/ssl
```

```
wget -O - https://www.startssl.com/certs/ca.pem  
https://www.startssl.com/certs/sub.class1.server.ca.pem | tee -a ca-certs.pem > /dev/null
```

۳-۴ فعال کردن HSTS

HTTP Strict Transport Security یک بهبود امنیتی برای برنامه‌های تحت وبی است که از پروتکل HTTPS استفاده می‌کنند. وجود این مکانیسم باعث جلوگیری از Downgrade Attack و Cookie Hijacking می‌شود. این قابلیت همچنین مرورگر را ملزم می‌کند که حتماً از پروتکل HTTPS برای ارتباط با سرور استفاده کند. برای فعال‌سازی این قابلیت می‌توان به صورت زیر عمل کرد.
در ابتدا باید با استفاده از دستور زیر مازول headers را فعال کرد:

```
a2enmod headers
```

همچنین باید قبل از آن در فایل پیکربندی default-ssl.conf خط زیر بیرون از تگ VirtualHost قرار داده شود:

```
LoadModule headers_module /usr/lib/apache2/modules/mod_headers.so
```

سپس باید خط زیر درون تگ <VirtualHost *:443> اضافه گردد.

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

همچنین جهت اجبار الزام کاربر به استفاده از HTTPS باید خط زیر را درون تگ <VirtualHost *:80> قرار گرفته در فایل /etc/apache2/sites-enabled/default.conf اضافه کرد:

```
ServerName example.com
```

```
Redirect permanent / https://example.com/
```

باید به جای example.com نام سرور موردنظر قرار گرفته شود.

۳-۵ فعال کردن HPKP

HTTP Public Key Pinning یک قابلیت است که به وبسایت‌هایی که از HTTPS استفاده می‌کنند اجازه می‌دهد تا نسبت به جعل هویت حمله‌کننده مقاوم باشند. بدین معنی که تنها CAهای معتبر، مجاز به امضای گواهی وبسایت می‌باشند. در غیر این صورت هر CA قرار گرفته در لیست مرورگر قادر به امضای گواهی خواهد بود. بنابراین امکان جعل هویت را از حمله‌کننده می‌گیرد.

جهت فعال‌سازی این قابلیت باید خط زیر درون تگ `<VirtualHost *:443>` قرار گرفته درون فایل `default-ssl.conf` اضافه گردد:

```
Header set Public-Key-Pins "pin-  
sha256=\"klO23nT2ehFDXCfx3eHTDRESMz3asj1muO+4aIdjiuY=\"; pin-  
sha256=\"633lt352PKRXbOwf4xSEa1M517scpD3l5f79xMD9r9Q=\"; max-age=2592000;  
includeSubDomains"
```

۴ مراجع

- [1]. https://httpd.apache.org/docs/2.4/ssl/ssl_howto.html
- [2]. http://httpd.apache.org/docs/2.0/ssl/ssl_howto.html
- [3]. <https://hynek.me/articles/hardening-your-web-servers-ssl-ciphers/>