مرکز صدور گواهی الکترونیکی پارس‌ساین

راهنمای استفاده از گواهی الکترونیکی در نرم‌افزار

Apache 2.x

تدوین‌کننده: شرکت امن‌افزار گستر شریف

شمارهٔ سند.....SSW_UG_PKI_91110_1

تاریخ..... ۱۸/آبان/۱۳۹۲

نگارش..... ۱.۵

آدرس: تهران، خیابان آزادی، خیابان حبیب‌الله، خیابان قاسمی غربی، شمارهٔ ۳۷، طبقه پنجم

تلفن: ۲۰-۶۱۹۷۵۵۰۰ (۰۲۱) **فاکس:** ۶۶۰۹۰۲۹۹ (۰۲۱) **سایت اینترنتی:** www.parssignca.ir

حق طبع و نشر

فهرست مطالب

۱ مقدمه ۱

۲ فرضیات این سند ۲

۳ ایجاد درخواست امضای گواهی (CSR) و کلید خصوصی ۲

۴ پیکربندی Apache به منظور استفاده از گواهی الکترونیکی ۳

۴-۱ نصب ماژول mod_ssl در وب‌سرور ۳

۴-۲ تنظیم وب‌سرور برای استفاده از پروتکل HTTPS ۴

۵ پیکربندی دیواره آتش ۷

۶ بررسی صحت نصب گواهی SSL ۷

۷ مشکلات احتمالی پس از نصب گواهی ۸

۷-۱ عدم نمایش قفل کنار عبارت https و عدم نمایش صحیح وب‌سایت ۸

۷-۲ نمایش صفحه هشدار SSL در مرورگر ۱۰

۷-۳ نمایش صفحه دیگری به جای صفحه سایت ۱۳

۷-۴ نمایش صحیح سایت HTTPS در یک مرورگر و عدم نمایش آن در مرورگری دیگر ۱۳

۸ پیوست ۱۴

۸-۱ بررسی PEM بودن فرمت گواهی و تبدیل این فرمت ۱۴

۸-۲ حذف گذوراژه کلید خصوصی ۲۰

۱ مقدمه

تأمین امنیت ارتباطات و تبادلات الکترونیکی در شبکه‌ها خصوصاً محیط اینترنت از جمله مسائل ویژه‌ای است که امروزه سازمان‌ها با آن مواجه هستند. به دلیل حساسیت امنیتی حجم قابل توجهی از این تبادلات در محیط وب، باید تدابیر امنیتی لازم در پروتکل‌ها و سرویس‌های مورد استفاده در این نوع ارتباطات اتخاذ گردد. پروتکل HTTP (که عموماً به عنوان پروتکل وب شناخته می‌شود) یکی از این پروتکل‌ها است که استفاده گسترده‌ای دارد. داده‌های HTTP به صورت ناامن روی شبکه منتقل می‌شوند؛ از این‌رو، داده‌هایی که بین سرویس‌دهنده[1] (سمت وب‌سایت) و سرویس‌گیرنده[2] (سمت کاربر وب‌سایت) مبادله می‌شود، توسط مهاجمین قابل مشاهده و حتی قابل تغییر هستند.

وب‌سایت‌هایی که اطلاعات مهم و محرمانه (مانند گذرواژه، شماره کارت اعتباری، اطلاعات بانکی، و دیگر اطلاعات خصوصی) با کاربران مبادله می‌کنند، نباید از پروتکل ناامن HTTP استفاده نمایند. نسخه امن‌شده این پروتکل به نام HTTPS، پرکاربردترین پروتکل امنیتی مبتنی بر رمزنگاری کلید عمومی است که در پیاده‌سازی این گونه وب‌سایت‌ها به کار می‌رود. این پروتکل مبتنی بر پروتکل SSL می‌باشد.

لازم به ذکر است، تأمین محرمانگی و عدم تغییر (جامعیت) اطلاعات تبادل‌شده بین کاربر و وب‌سایت تنها کاربرد HTTPS نیست. HTTPS برای جلوگیری از حملاتی مانند حمله فیشینگ مبتنی بر جعل سایت نیز استفاده می‌شود. در حمله مذکور، حمله‌کننده (مثلاً یک رقیب تجاری) با ایجاد یک وب‌سایت با ظاهری کاملاً مشابه سایت اصلی، کاربران آن سایت را به سایت جعلی هدایت کرده و امنیت آن‌ها را به مخاطره می‌اندازد؛ مثلاً اطلاعات کاربران را به سرقت برده یا در مورد آن‌ها اطلاعات جمع‌آوری می‌کند. در صورتی که از گواهی HTTPS استفاده شود، از این حمله جلوگیری می‌شود.

در پیکربندی وب‌سرور برای استفاده از HTTPS، از یک گواهی الکترونیکی SSL استفاده می‌شود. این گواهی باید توسط یک مرکز صدور گواهی الکترونیکی معتبر (مانند مرکز صدور گواهی الکترونیکی پارس‌ساین) صادر شده باشد تا کاربران بتوانند به آن اعتماد کرده و اطلاعات محرمانه خود را بر اساس این اعتماد، برای وب‌سایت ارسال نمایند.

---

[1] Server
[2] Client

در این سند، کلیه مراحل لازم برای پیکربندی وب‌سرور Apache 2.x به منظور استفاده از گواهی الکترونیکی توضیح داده شده است.

۲ فرضیات این سند

در این سند، نحوه استفاده از گواهی الکترونیکی، بر اساس سناریوی فرضی زیر در نظر گرفته شده است:

«می‌خواهیم برای یک وب‌سایت با دامنه www.mydomain.com، وب سرور Apache نسخه 2.x را طوری تنظیم نماییم که امکان برقراری ارتباط HTTPS با وب‌سایت فراهم شود.»

از این‌رو، به منظور استفاده از این سند در سناریوی واقعی، باید به جای دامنه www.mydomain.com، نام دقیق دامنه وب‌سایت خود را وارد نمایید (دامنه‌ای که برای آن گواهی SSL دریافت نموده‌اید).

۳ ایجاد درخواست امضای گواهی (CSR) و کلید خصوصی

برای دریافت گواهی الکترونیکی از مرکز صدور گواهی الکترونیکی پارس‌ساین، ابتدا باید یک درخواست امضای گواهی[1] (CSR) ایجاد نماییم. لازم به ذکر است که فیلدهای CSR باید طبق سیاست‌های مرکز صدور گواهی الکترونیکی پارس‌ساین تکمیل گردد. در غیر این صورت، مرکز پارس‌ساین برای آن CSR، گواهی الکترونیکی صادر نخواهد نمود.

به منظور پیروی از فرایند ایجاد CSR از سیاست‌های مرکز پارس‌ساین و زیرساخت کلید عمومی کشور توصیه می‌شود برای ایجاد CSR تنها از نرم‌افزار ParsKey Utility استفاده نمایید. برای دریافت این نرم‌افزار به بخش دانلود سایت مرکز میانی صدور گواهی الکترونیکی پارس‌ساین به آدرس www.parssignca.ir مراجعه نمایید. سند "راهنمای ایجاد CSR با استفاده از نرم‌افزار ParsKey Utility" را نیز می‌توانید از بخش راهنماها در سایت دریافت نمایید. نحوه ایجاد CSR برای گواهی SSL، در بخش "پروفایل گواهی SSL" از سند مذکور تشریح شده است.

پس از ایجاد CSR و کلید خصوصی، فایل CSR باید به مرکز پارس‌ساین تحویل داده شود تا گواهی SSL متناظر آن را صادر نماید. دقت نمایید برای صدور گواهی توسط مرکز پارس‌ساین، تنها به فایل CSR نیاز می‌باشد و احتیاجی به ارائه فایل کلید خصوصی به این مرکز نیست.

[1] Certificate Signing Request (CSR)

نکته مهم: سرور از فایل کلید خصوصی برای رمزگذاری و رمزگشایی داده‌ها استفاده می‌نماید. از این‌رو، در محافظت از این فایل دقت نمایید. در صورتی که این کلید در دسترس افراد غیرمجاز قرار گیرد، کلیه داده‌های رمزشده بین وب‌سایت و کاربران می‌تواند توسط این افراد رمزگشایی شود.

نکته مهم: بین CSR، کلید خصوصی (این کلید هنگام ایجاد CSR تولید می‌شود)، و گواهی الکترونیکی یک تناظر وجود دارد. از این‌رو، هر گواهی فقط با کلید خصوصی متناظر آن قابل استفاده است. چنانچه هنگام نصب گواهی روی سرور، فایل کلید خصوصی متناظر آن گواهی وجود نداشته باشد، گواهی روی سرور قابل استفاده نخواهد بود.

۴ پیکربندی Apache به منظور استفاده از گواهی الکترونیکی

پیکربندی HTTPS در نرم‌افزار Apache 2.x به ترتیب شامل مراحل زیر می‌باشد:

۱. **نصب ماژول mod_ssl در وب‌سرور؛**
۲. **تنظیم وب‌سرور برای استفاده از پروتکل HTTPS.**

در بخش‌های بعد، هر یک از مراحل فوق با جزییات تشریح می‌گردد.

۴-۱ نصب ماژول mod_ssl در وب‌سرور

برای استفاده از قابلیت SSL در نرم‌افزار Apache ابتدا باید ماژول mod_ssl را روی سرور نصب نماییم. برای این منظور مراحل زیر را انجام می‌دهیم:

۱. به سرور Login می‌نماییم.
۲. با استفاده از دستور زیر اطمینان حاصل می‌نماییم که mod_ssl روی سیستم نصب گردیده است یا خیر.

```
#rpm -qa |grep mod_ssl
```

در صورت عدم نصب آن، با استفاده از دستور زیر آن را نصب می‌نماییم.

```
#yum install mod_ssl
```

۲-۴ تنظیم وب‌سرور برای استفاده از پروتکل HTTPS

پس از دریافت گواهی الکترونیکی از مرکز صدور گواهی، باید وب‌سرور را برای استفاده از HTTPS به صورت زیر تنظیم نماییم:

۱. ابتدا باید فایل‌های زیر را در مسیری امن از سیستم کپی می‌نماییم:

- فایل گواهی الکترونیکی صادرشده برای دامنه موردنظر با فرمت PEM (در مثال ما، گواهی www.mydomain.com.crt)؛
- فایل کلید خصوصی متناظر با گواهی (در مثال ما، فایل key-www.mydomain.com.pem).

ما برای مثال فرضی خود، دایرکتوری /etc/httpd/ssl را با دستور زیر ایجاد می‌نماییم.

```
#mkdir /etc/httpd/ssl
```

سپس فایل‌های بالا را در آن کپی می‌نماییم. برای کپی هر یک از فایل‌ها، به دایرکتوری حاوی آن فایل رفته و دستور کپی متناظر با آن فایل را به صورت زیر وارد می‌نماییم.

```
#cp www.mydomain.com.crt /etc/httpd/ssl
#cp key-www.mydomain.com.pem /etc/httpd/ssl
```

نکته: در صورتی که از SELinux استفاده می‌کنید، فایل‌ها را Move (دستور mv) نکنید بلکه کپی کنید. در غیر این‌صورت، Apache قادر به شناسایی فایل‌های گواهی نخواهد بود.

۲. به منظور محافظت از فایل گواهی صادرشده و به ویژه کلید خصوصی، با استفاده از دستورات زیر مالکیت این فایل‌ها را به کاربر root و مجوزهای آن‌ها را به 400 تغییر می‌دهیم:

```
#chown root:root /etc/httpd/ssl/key-www.mydomain.com.pem
#chown root:root /etc/httpd/ssl/www.mydomain.com.crt
#chmod 400 /etc/httpd/ssl/key-www.mydomain.com.pem
#chmod 400 /etc/httpd/ssl/www.mydomain.com.crt
```

۳. آپاچی به منظور پیکربندی SSL از فایل پیکربندی مجزایی به نام ssl.conf استفاده می‌کند که در مسیر /etc/httpd/conf.d/ قرار دارد (در برخی نسخه‌های جدید مثل ۲.۴، این فایل با عنوان http-ssl.conf در مسیر نصب آپاچی و در دایرکتوری conf/extra/ قرار دارد). این فایل را به منظور ویرایش باز می‌نماییم:

```
#vi /etc/httpd/conf.d/ssl.conf
```

در این فایل، توضیح برخی از گزینه‌های مهم به صورت زیر می‌باشد (لازم به ذکر است که برای فعال نمودن گزینه‌ها، باید با حذف کاراکتر #، آن‌ها را از حالت Comment خارج نماییم):

- **Listen**: این گزینه، شماره درگاه[1] پیش‌فرض HTTPS را تعیین می‌کند. دقت کنید که شماره درگاه 443 به طور پیش‌فرض برای HTTPS می‌باشد؛ در صورتی که شماره درگاه دیگری را انتخاب نماییم، سرویس‌گیرندگان[2] برای برقراری ارتباط HTTPS باید از این شماره درگاه مطلع شوند. بنابراین توصیه می‌شود از همان شماره درگاه پیش‌فرض 443 استفاده نمایید. لازم به ذکر است، این بخش به طور پیش‌فرض مشابه عبارت زیر فعال می‌باشد:

```
Listen 443
```

- **SSLEngine**: این گزینه برای تنظیم استفاده یا عدم استفاده از SSL می‌باشد. به منظور استفاده از HTTPS، باید این گزینه مقدار on داشته باشد. این گزینه مشابه عبارت زیر، به صورت پیش‌فرض فعال می‌باشد.

```
SSLEngine on
```

- **SSLCertificateFile**: در این گزینه، باید محل فایل گواهی سرور (که توسط مرکز صدور گواهی صادر شده است) را مشخص نماییم. برای مثال، مقدار این گزینه به صورت زیر می‌باشد. این گزینه نیز به صورت پیش‌فرض فعال می‌باشد.

```
SSLCertificateFile /etc/httpd/ssl/www.mydomain.com.crt
```

نکته: فرمت گواهی الکترونیکی باید PEM باشد. نحوه بررسی PEM بودن فایل گواهی و ایجاد یک گواهی با فرمت PEM در پیوست سند (بخش ۸) تشریح شده است.

- **SSLCertificateKeyFile**: در این گزینه، باید محل فایل کلید خصوصی متناظر با گواهی سرور را مشخص نماییم. برای مثال، مقدار این گزینه که به صورت پیش‌فرض فعال است، به صورت زیر می‌باشد:

```
SSLCertificateKeyFile /etc/httpd/ssl/key-www.mydomain.com.pem
```

۴. تغییرات اعمال‌شده در فایل را ذخیره می‌نماییم.

[1] Port
[2] Client

۵. قبل از راه‌اندازی مجدد Apache، باید با استفاده از دستور زیر صحت پیکربندی آن را بررسی نماییم تا خطایی در آن وجود نداشته باشد.

```
#apachectl configtest
```

۶. Apache را با استفاده از دستور زیر راه‌اندازی مجدد می‌نماییم.

```
#/etc/rc.d/init.d/httpd restart
```

در نسخه‌های جدید آپاچی ممکن است این دستور متفاوت باشد، مثلاً در نسخه ۲.۴ به صورت زیر است:

```
#apachectl -k restart
```

نکته: با راه‌اندازی مجدد Apache، برای فعال نمودن قابلیت HTTPS باید گذرواژه کلید خصوصی را وارد نماییم (این گذرواژه با عنوان Private Key Password هنگام ایجاد CSR تنظیم شده است). در صورتی که نخواهید هر بار با راه‌اندازی مجدد Apache، گذرواژه کلید خصوصی را وارد نمایید، باید گذرواژه کلید خود را با استفاده از نرم‌افزار OpenSSL حذف نمایید. البته به دلایل امنیتی این کار توصیه نمی‌شود. نحوه برداشتن پسورد کلید خصوصی در پیوست (بخش) آمده است.

توجه: لازم به ذکر است در بالا، برخی تنظیمات پایه‌ای بیان شد که با انجام آن‌ها می‌توان از سرویس HTTPS استفاده نمود. با این وجود، برای برخی کاربردها ممکن است نیاز به انجام تنظیمات خاص آن کاربرد باشد. برای نمونه، ممکن است در یک کاربرد لازم باشد که دسترسی به برخی صفحات تنها از طریق HTTPS (و نه پروتکل ناامن HTTP) امکان‌پذیر باشد؛ برای این مورد باید از گزینه SSLRequireSSL استفاده نمایید. برای آگاهی از چنین تنظیماتی به مستندات راهنمای Apache مراجعه نمایید.

توجه: پس از تنظیم سرور، در صورتی که دیواره آتش روی سیستم فعال باشد، دیواره آتش نیز باید تنظیم گردد. بدین‌منظور، به بخش ۵ مراجعه نمایید.

۵ پیکربندی دیواره آتش

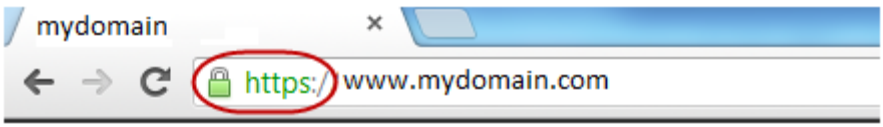
به منظور دسترسی به سایت از طریق HTTPS، باید درگاه HTTPS که در تنظیمات ssl.conf مشخص نمودیم را در دیواره آتش باز نماییم (برای مثال ما، شماره این درگاه 443 می‌باشد). بدین منظور دستورات زیر را وارد می‌نماییم:

```
#/sbin/iptables -I INPUT -p tcp --dport 443 -m state --state
NEW,ESTABLISHED -j ACCEPT
#/sbin/iptables -I OUTPUT -p tcp --sport 443 -m state --state
ESTABLISHED -j ACCEPT
#/sbin/service iptables save
```

۶ بررسی صحت نصب گواهی SSL

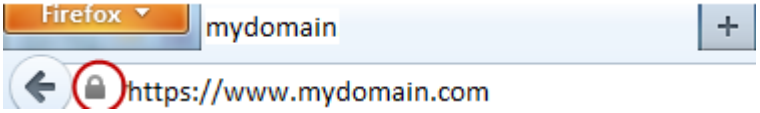
برای برقراری ارتباط صحیح SSL میان مرورگر کاربر و وب‌سایت، کاربر وب‌سایت باید زنجیره گواهی را روی سیستم یا مرورگر خود نصب نماید. رویه این کار در سند «راهنمای نصب زنجیره گواهی» تشریح شده است. این سند را می‌توانید از بخش "راهنماها" در وب‌سایت مرکز پارس‌ساین به آدرس www.parssignca.ir دانلود و مطالعه نمایید. پس از نصب صحیح زنجیره گواهی، در نوار آدرس مرورگر، آدرس https://www.mydomain.com را وارد می‌نماییم (به جای www.mydomain.com باید آدرس دقیق سایت خود را وارد نمایید). نمایش علامت قفل به همراه عبارت https در نوار آدرس مرورگرها یکی از نشانه‌های صحیح بودن نصب گواهی SSL در وب‌سایت می‌باشد. نمایش علامت قفل در سه مرورگر Google Chrome، Internet Explorer، و Firefox به صورت زیر می‌باشد:

- **مرورگر Google Chrome**: در نوار آدرس (مطابق شکل زیر) قفل سبز رنگ در کنار عبارت سبز رنگ https ظاهر می‌شود.



- **مرورگر Internet Explorer**: علامت قفل در سمت راست نوار آدرس (شکل زیر) ظاهر می‌شود.

- **مرورگر فایرفاکس**: در نوار آدرس (مطابق شکل زیر) آیکن قفل کنار عبارت https ظاهر می‌شود.



برای اطمینان کامل از صحت نصب گواهی، در مرورگر خود روی علامت قفل کلیک نموده و روی لینک Certificate Information در Google Chrome، View Certificates در Internet Explorer، و More Information در Firefox کلیک نمایید. سپس اطمینان حاصل کنید که گواهی نشان‌داده شده، همان گواهی است که مرکز پارس‌ساین برای وب‌سایت شما صادر نموده است.

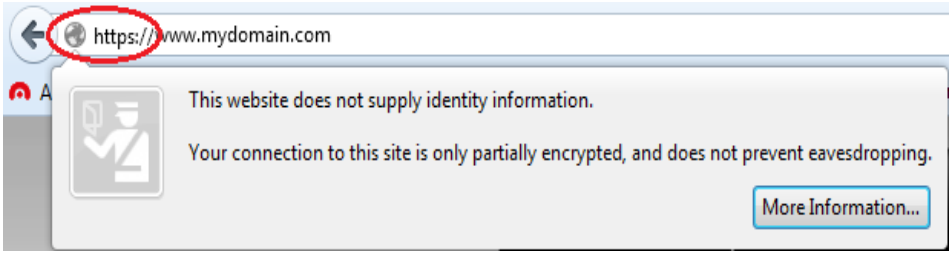
۷ مشکلات احتمالی پس از نصب گواهی

۷-۱ عدم نمایش قفل کنار عبارت https و عدم نمایش صحیح وب‌سایت

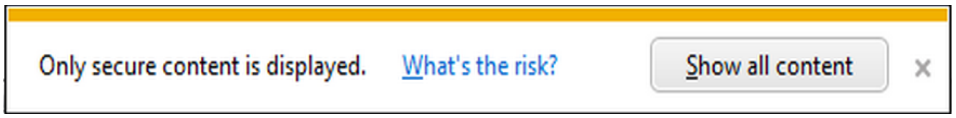
در صورتی که نصب گواهی روی سرور، همچنین نصب زنجیره روی مرورگر، هر دو به درستی انجام شده باشند اما در مرورگرها شکل‌های زیر نمایش داده شود:

- **مرورگر Google Chrome**: در نوار آدرس مرورگر، علامت مثلث روی قفل کنار https نشان داده شود (مانند شکل زیر).

- **مرورگر Firefox**: در نوار آدرس مرورگر، علامت قفل کنار https ظاهر نشود (مانند شکل زیر).



- **مرورگر Internet Explorer**: در پایین صفحه، پیامی مانند شکل زیر ظاهر گردد.



- **مرورگر Opera**: در نوار آدرس مرورگر، علامت Secure کنار https ظاهر نشود (مانند شکل زیر).

- **مرورگر Safari**: در انتهای نوار آدرس مرورگر، علامت قفل ظاهر نشود (مانند شکل زیر).

این مشکل معمولاً Mixed content warning نامیده می‌شود که ممکن است یک حمله‌کننده با استفاده از آن امنیت کاربران سایت شما را به مخاطره اندازد. این مشکل مربوط به گواهی SSL سایت نیست، بلکه به کد وب‌سایت یا تنظیمات سرور شما مربوط می‌باشد. این مشکل بدین دلیل است که در وب‌سایت شما از محتوای[1] ناامن استفاده شده است. مثال‌هایی از محتوای ناامن، عکس‌ها، اسکریپت‌ها، یا CSSهایی هستند که به طور ناامن (از طریق HTTP) در سایت شما بارگذاری می‌شوند. مرورگر از بارگذاری یا اجرای این محتواها جلوگیری می‌کند. به همین دلیل ممکن است CSS وب‌سایت شما بارگذاری نشده و ساختار وب‌سایت شما با HTTPS به هم بریزد، عکسی بارگذاری نشود، یا اسکریپتی اجرا نشود.

برای کشف محتوای ناامن، می‌توانید از قابلیت Web Developer Toolbar در اغلب مرورگرها استفاده کنید. کشف محتوای ناامن با استفاده از مرورگرهای رایج معمولاً به صورت زیر می‌باشد:

- **مرورگر Google Chrome و Internet Explorer**: فشردن کلید F12 و مشاهده خطاها در بخش Console.
- **مرورگر Firefox**: فشردن کلیدهای Ctrl + Shift + K و مشاهده خطاهای Mixed Content.

پس از کشف محتوای ناامن، آن‌ها را از طریق HTTPS بارگذاری نموده و در صورت عدم امکان بارگذاری با استفاده از HTTPS، آن‌ها را از وب‌سایت خود حذف نمایید. در زیر چند سناریو از محتوای ناامن توصیف شده است.

- برای مثال، فرض کنید CSS سایت به صورت http://www.mydomain.com/styles/mycss.css در صفحه سایت فراخوانی شده باشد. در این صورت، به دلیل استفاده از http (به جای https) این محتوا توسط مرورگر ناامن تشخیص داده شده و بارگذاری نمی‌شود (در نتیجه ساختار سایت بهم می‌ریزد). بنابراین توصیه می‌شود به جای استفاده از آدرس‌های قطعی (hardcoded) برای عکس‌ها،

[1] Content

CSS، و اسکریپت‌های سایت، از روش‌هایی مانند آدرس‌های نسبی، توابع (مثلاً include())، یا هر روش دیگری استفاده نمایید که با استفاده از آن بتوان محتوا را با استفاده از https بارگذاری نمود. در صورتی که از روشی مانند آدرس‌دهی نسبی استفاده کنید اما مشکل همچنان باقی باشد، مشکل مربوط به تنظیمات SSL در سرور شما می‌باشد.

- در برخی موارد، ممکن است آدرس عکس، CSS، یا اسکریپت با https باشد اما مرورگر آن را بارگذاری نکند. این مشکل معمولاً به این دلیل است که مثلاً CSS با آدرس https://mydomain.com/styles/mycss.css (یعنی بدون www) فراخوانی شده است اما آدرس درج شده در گواهی سایت، با www است. در این مثال، راه حل این است آدرس https به طور دقیق و طبق آنچه در گواهی SSL سایت درج‌شده وارد شود.
- مثال دیگر از محتوای ناامن، استفاده از اسکریپت‌هایی است که با استفاده از http اطلاعاتی را از سایت ما برای سایت دیگری (مثلاً برای یک سایت ثبت آمار کاربران) ارسال می‌کنند. همچنین، اسکریپت‌ها، عکس‌ها، و غیره که از سایت دیگر به صورت http در سایت ما بارگذاری می‌شوند. مرورگر از بارگذاری و اجرای این محتواها نیز جلوگیری می‌کند. در صورتی که امکان بارگذاری این محتواها از طریق https وجود ندارد، آن‌ها از وب‌سایت خود حذف نمایید. روش دیگر این است که دو صفحه مجزا، یکی برای http و یکی برای https طراحی کنید؛ در صفحه مربوط به http، همه محتوای مورد نظر را قرار دهید اما در صفحه https، محتوای ناامن را حذف کنید. سپس در تنظیمات SSL روی سرور، درخواست‌های SSL را به آدرس صفحه https ارجاع دهید. راه‌حل دیگر این است که به جای دریافت محتوا (مثلاً عکس) از سایت دیگر، محتوا را از سایت مذکور دریافت و در منابع سایت خود قرار دهید. سپس به طور محلی آن‌ها را فراخوانی/بارگذاری کنید.
- در برخی موارد، ممکن است محتوا از سایت دیگر و با https فراخوانی شود، اما توسط مرورگر بارگذاری نشود. این مشکل معمولاً دلایل مختلفی می‌تواند داشته باشد که همگی بستگی به سرویس HTTPS سایت ارسال‌کننده محتوا دارد. مثلاً ممکن است گواهی SSL آن سایت، برای مرورگر مورد اعتماد نباشد. برای حل این مشکل، وضعیت سرویس HTTPS سایت ارسال‌کننده را بررسی نمایید.

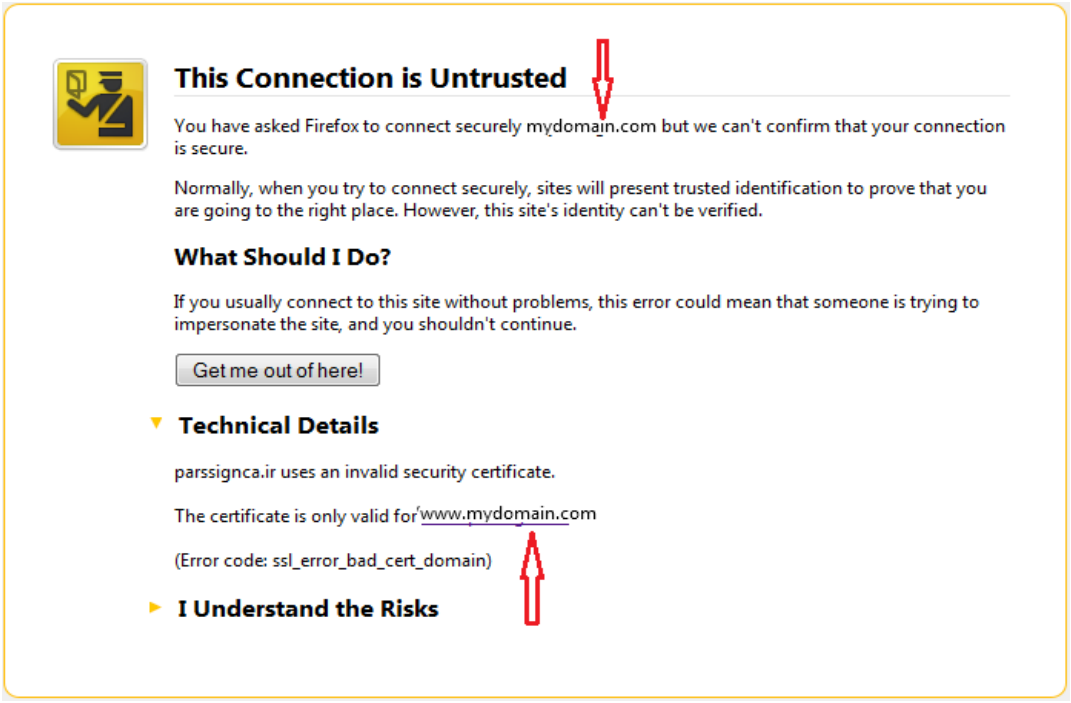
۲-۷ نمایش صفحه هشدار SSL در مرورگر

در حالت کلی این گونه هشدارها را به دقت مطالعه نموده و دلیل آن را بررسی نمایید.

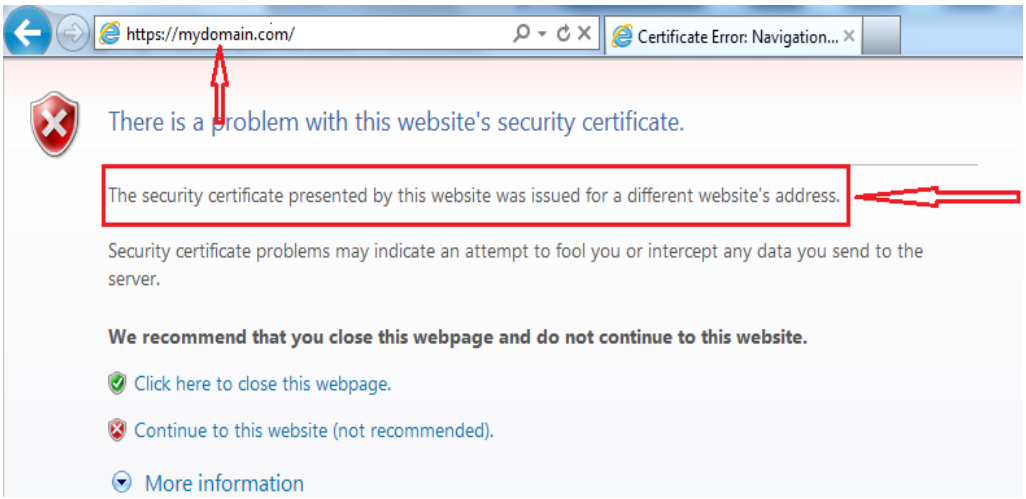
در صورتی که نصب گواهی روی سرور، همچنین نصب زنجیره روی مرورگر، هر دو به درستی انجام شده باشند اما در مرورگرها شکل‌های زیر نمایش داده شود:

- **مرورگر Google Chrome**: پیام زیر نمایش داده شود.

- **مرورگر Firefox**: پیام زیر نمایش داده شود.



- **مرورگر Internet Explorer**: پیام زیر نمایش داده شود.



دلیل مشکل فوق این است که آدرس سایت را بطور دقیق وارد ننموده‌اید. در زیر، چند سناریو که منجر به چنین خطایی می‌شوند آورده شده است:

- اگر گواهی فقط برای www.mydomain.com صادر شده باشد اما آدرس https://mydomain.com (یعنی بدون www) را در مرورگر وارد نمایید، با صفحه خطای فوق مواجه می‌شوید. برای رفع این مشکل، یا باید همیشه آدرس دقیق وارد شود یا از مرکز صدور گواهی، گواهی‌ای درخواست کنید که در آن هر دو دامنه بالا ذکر شده باشد.
- ممکن است یک سایت، به چند دامنه (مثلاً mydomain.com و mydomaint.ir) تخصیص داده شده باشد. مثلاً اگر گواهی سایت برای www.mydomain.com صادر شده باشد اما آدرس https://www.mydomain.ir را در مرورگر وارد نمایید، با صفحه خطا مواجه می‌شوید. در صورتی که برای یک سایت چند دامنه وجود داشته باشد اما شما فقط برای یک دامنه درخواست گواهی داده باشید، باید از مرکز صدور گواهی درخواست گواهی SAN نمایید. کاربرد گواهی SAN برای استفاده از یک گواهی برای چندین دامنه است.
- ممکن است برای یک سایت فقط یک دامنه وجود داشته باشد اما در آن دامنه، زیردامنه[1] نیز وجود داشته باشد (مثلاً mail.mydomain.com). حال اگر آدرس https://mail.mydomain.ir را در مرورگر وارد نمایید، با صفحه خطا مواجه می‌شوید. در صورتی که بخواهید برای زیردامنه خود نیز از HTTPS استفاده نمایید، باید یک گواهی مجزا برای این زیر دامنه درخواست نمایید. در صورتی

[1] Subdomain

که بخواهید فقط از یک گواهی برای یک دامنه و تمام زیردامنه‌هایش استفاده کنید، می‌توانید درخواست گواهی Wildcard نمایید.

توجه: ممکن است صفحه‌های هشدار مشابه دیگری با دلایل خاص خود وجود داشته باشد که معمولاً مربوط به پیکربندی سرور شما می‌باشد. در صورت مواجهه با چنین خطاهایی، آن‌ها را به دقت مطالعه نموده و با انجام تنظیمات لازم روی سرور، آن‌ها رفع نمایید.

۳-۷ نمایش صفحه دیگری به جای صفحه سایت

در صورتی که آدرس IP سایت با سایت‌های دیگر به اشتراک گذاشته شده باشد (Shared باشد)، هنگام درخواست HTTPS، سرور معمولاً صفحه پیش‌فرضی را نشان می‌دهد (زیرا نمی‌تواند تشخیص دهد صفحه مربوط به چه دامنه‌ای را برگرداند). برای استفاده از HTTPS برای یک دامنه (وب‌سایت)، باید یک آدرس IP اختصاصی (Dedicated IP Address) به آن دامنه تخصیص داده شود.

۴-۷ نمایش صحیح سایت HTTPS در یک مرورگر و عدم نمایش آن در مرورگری دیگر

این مشکل مربوط به گواهی SSL سایت نیست، زیرا گواهی صادرشده توسط مرکز میانی پارس‌ساین از استانداردی (استاندارد X.509) تبعیت می‌کند که وابسته به مرورگر یا سیستم‌عامل خاصی نیست. این مشکل معمولاً به دلیل استفاده از مرورگر یا سروری است که بروزرسانی نشده است. در صورتی که مرورگر و سرور هر دو به روز باشد، مشکل در تنظیمات SSL سرور است.

۸ پیوست

۸-۱ بررسی PEM بودن فرمت گواهی و تبدیل این فرمت

از روی پسوند گواهی، نمی‌توان به PEM بودن فرمت فایل گواهی پی برد. برای بررسی PEM بودن گواهی، آن را با یک ویرایشگر متن (نرم‌افزار WordPad در ویندوز و cat در لینوکس) مشاهده می‌نماییم. در صورتی که فایل با عبارت -----BEGIN CERTIFICATE----- شروع و به -----END CERTIFICATE----- ختم شده باشد، فرمت فایل از نوع PEM می‌باشد.

در صورتی که فرمت گواهی PEM نباشد، به یکی از دو روش زیر می‌توانیم، آن را تبدیل به یک گواهی با فرمت PEM نماییم:

- **روش اول:** با استفاده از نرم‌افزار OpenSSL در لینوکس (که معمولاً بطور پیش‌فرض روی سیستم‌های لینوکس وجود دارد)، می‌توانیم عملیات تبدیل را انجام دهیم. بدین‌منظور، دستور زیر را در ترمینال لینوکس وارد می‌نماییم:

```
#openssl x509 -inform der -in www.mydomain.com.cer -out www.mydomain.com.crt
```

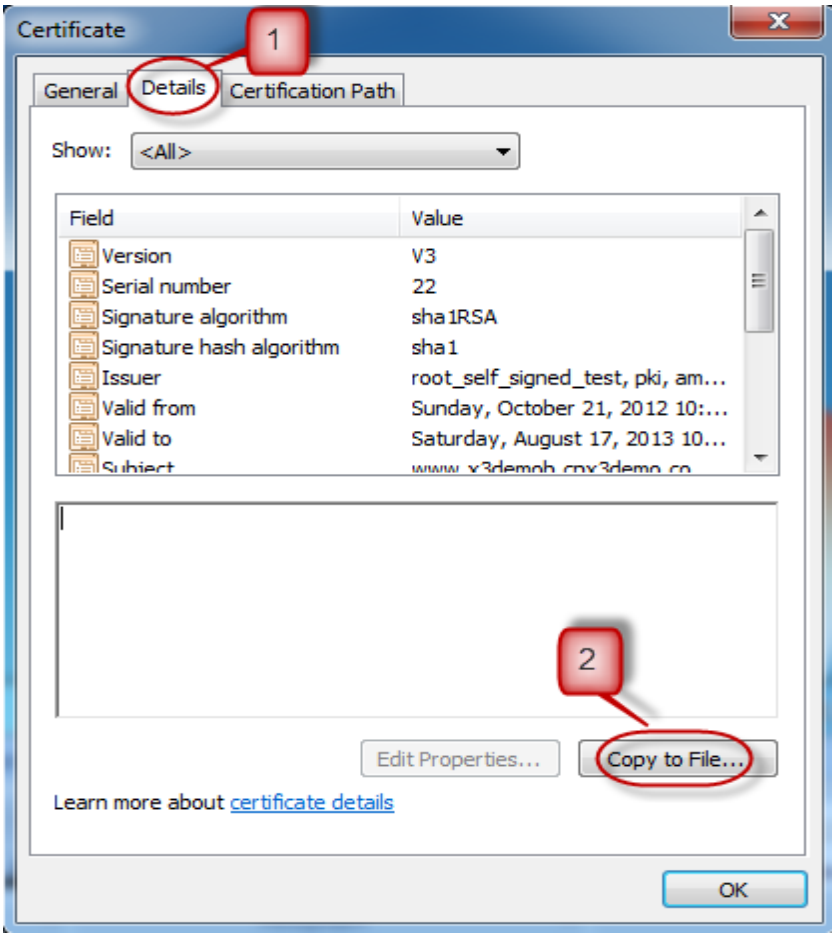
در دستور بالا، www.mydomain.com.cer فایل گواهی ورودی در فرمت DER است و فایل www.mydomain.com.crt فایل گواهی خروجی در فرمت PEM است. لازم به ذکر است در این دستور، باید آدرس دقیق فایل‌های مذکور را وارد نماییم. مثلاً اگر گواهی‌ها را در دایرکتوری /etc/httpd/ssl ذخیره می‌کنیم، دستور به صورت زیر خواهد بود:

```
#openssl x509 -inform der -in /etc/httpd/ssl/www.mydomain.com.cer
 -out /etc/httpd/ssl/www.mydomain.com.crt
```

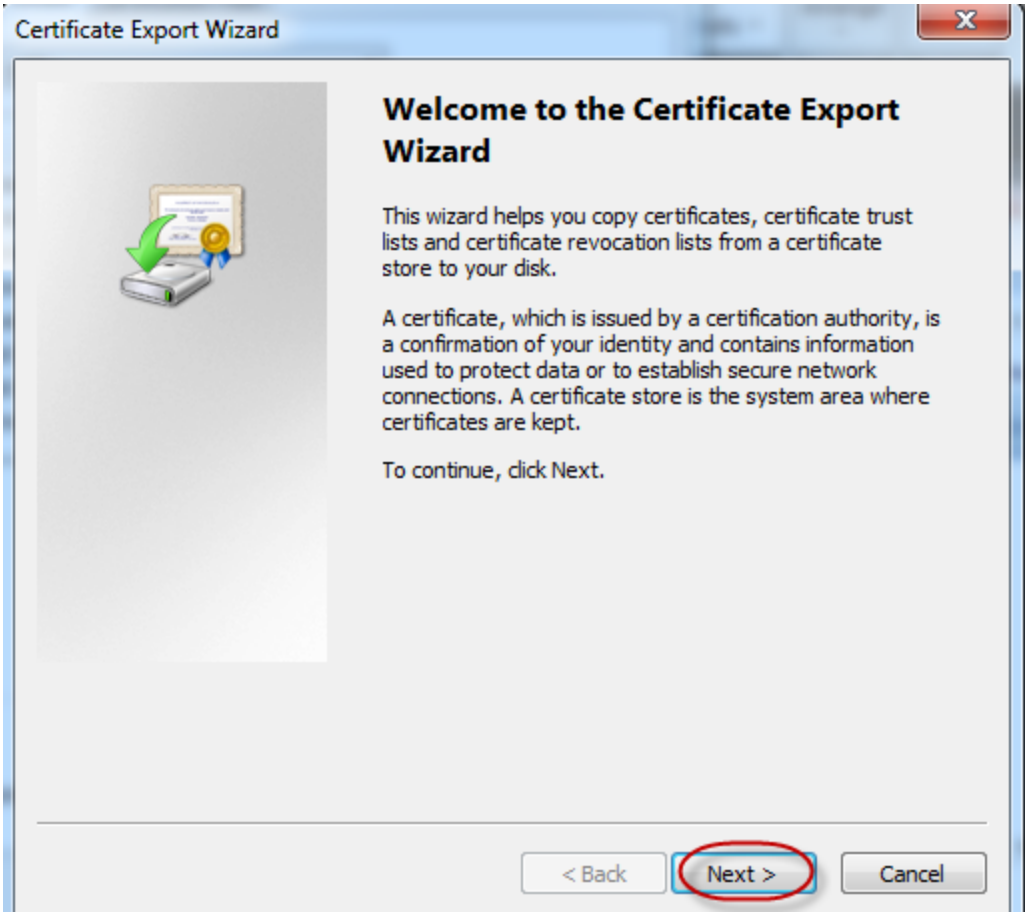
- **روش دوم:** در ویندوز می‌توانیم عمل تبدیل فرمت را به صورت زیر انجام دهیم:

۱. ابتدا فایل مورد نظر را باز می‌نماییم. برای این کار، روی فایل گواهی دابل‌کلیک نموده، سپس در پنجره ظاهرشده روی دکمه Open کلیک می‌نماییم.

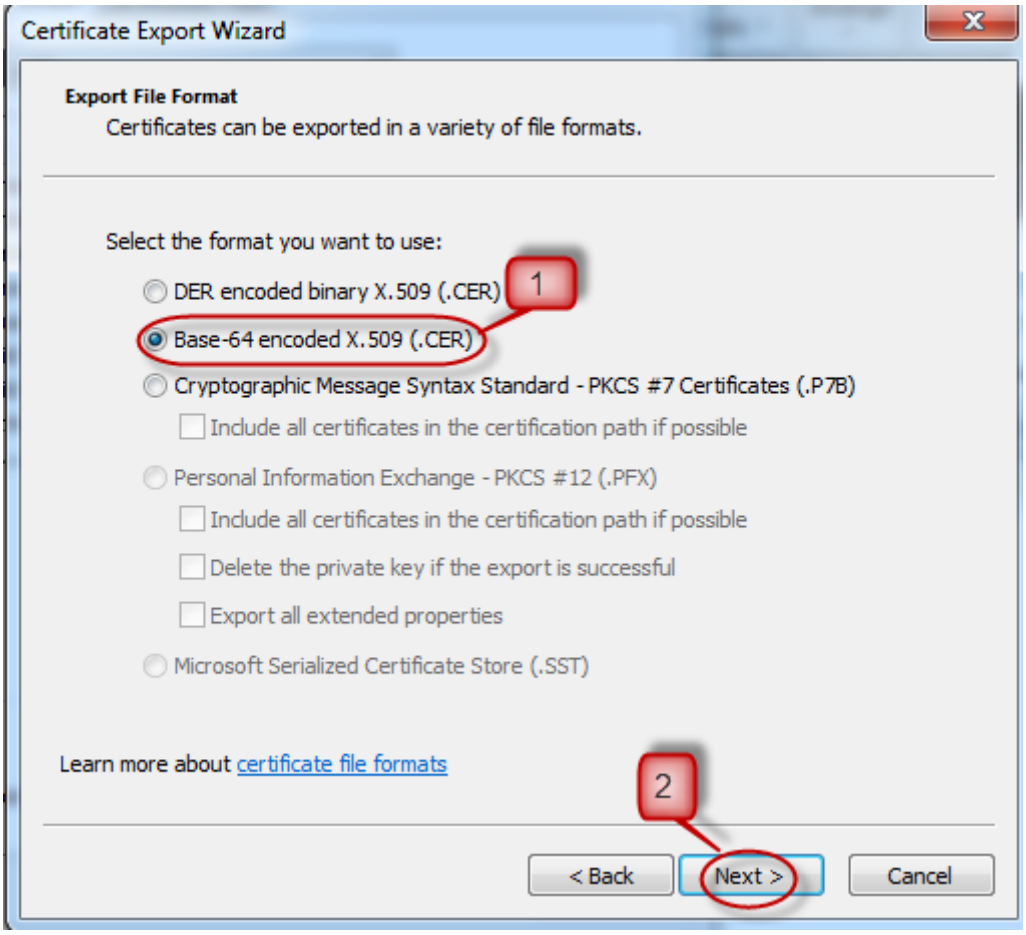
۲. در پنجره بازشده، در سربرگ Details روی دکمه Copy to Files کلیک می‌نماییم (مانند شکل زیر).



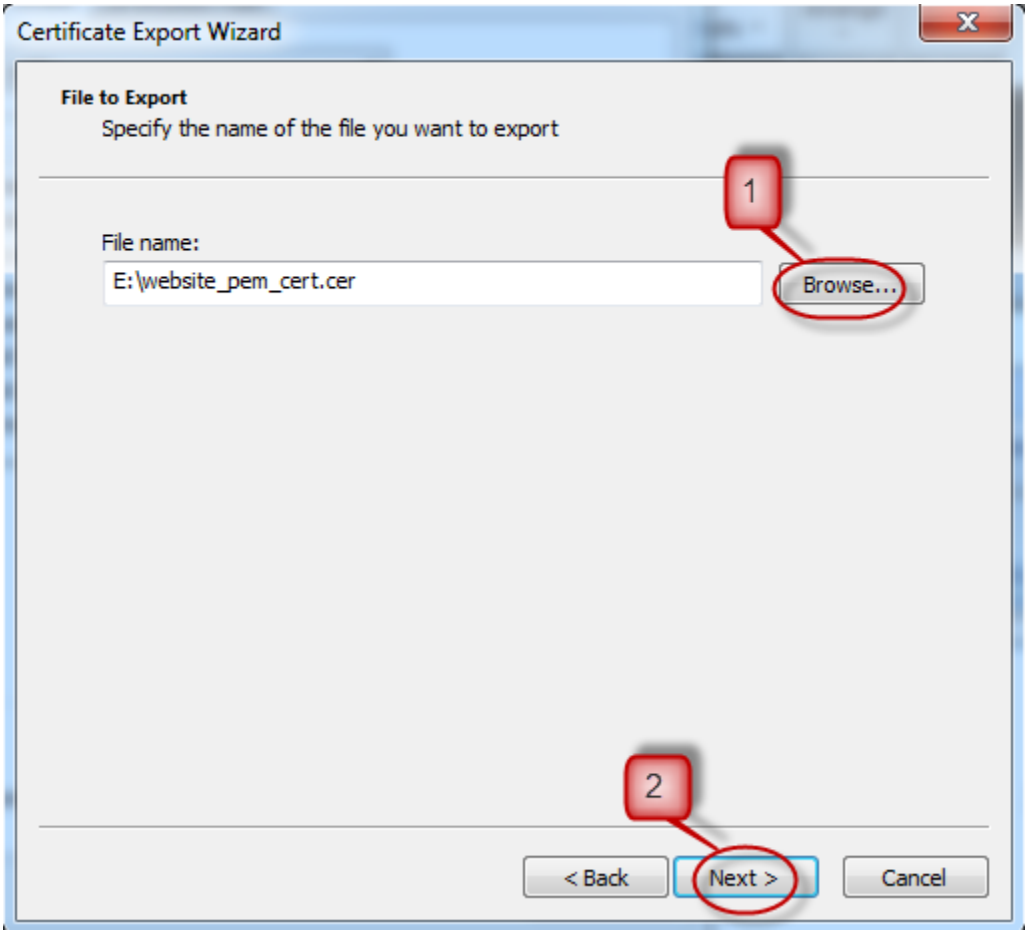
۳. در پنجره Certificate Export Wizard روی Next کلیک می‌نماییم (مانند شکل زیر).



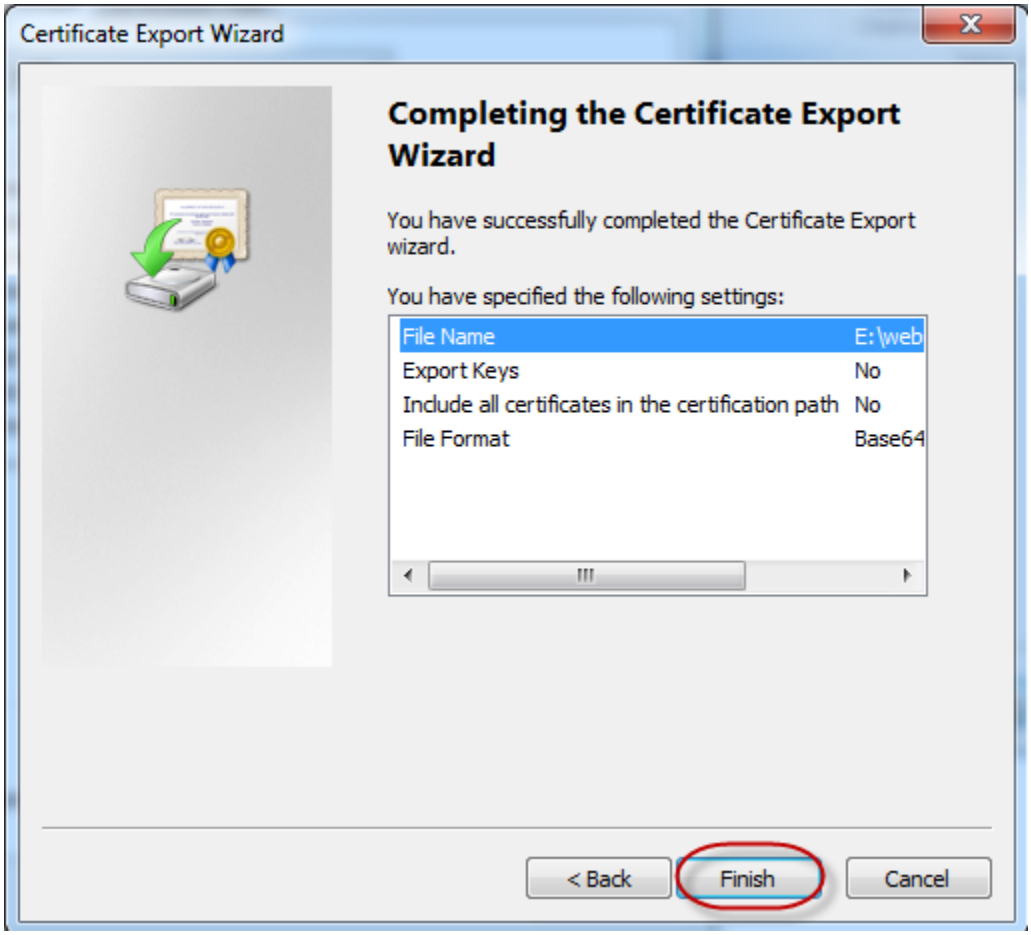
۴. در پنجره Export File Format در بخش Select the format you want to use گزینه Base-64 encoded X.509 (.CER) را انتخاب نموده، سپس روی Next کلیک می‌کنیم (مانند شکل زیر).



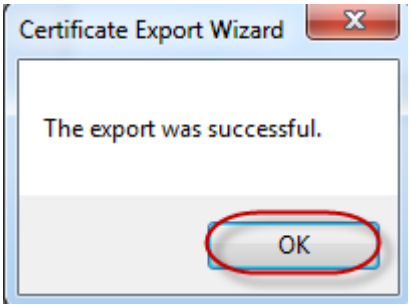
۵. در پنجره File to Export برای انتخاب مسیر ذخیره فایل با فرمت PEM، روی Browse کلیک می‌نماییم. پس از انتخاب نام و مسیر ذخیره فایل، روی Next کلیک می‌نماییم (مانند شکل زیر).



۶. در پنجره Completing the Certificate Export Wizard روی Finish کلیک می‌نماییم (مانند شکل زیر).



۷. پنجره زیر ظاهر می‌گردد که نشان‌دهنده ایجاد موفقیت‌آمیز یک گواهی جدید با فرمت PEM می‌باشد. در این پنجره روی OK کلیک می‌نماییم.



۸-۲ حذف گذوراژه کلید خصوصی

جهت حذف گذرواژه کلید خصوصی می‌توانید از نرم‌افزار OpenSSL استفاده نمایید. بدین‌منظور از دستور زیر استفاده می‌کنیم:

```
#openssl rsa -in mydomain.key -out mydomain_nopass.key
```

با اجرای دستور بالا، از شما خواسته می‌شود که گذرواژه کلید خصوصی را وارد نمایید که باید گذرواژه انتخاب‌شده برای کلید هنگام ایجاد CSR را وارد کنید. در دستور بالا، فایل mydomain.key، فایل کلید خصوصی است که می‌خواهیم گذرواژه آن را حذف کنیم. فایل mydomain_nopass.key، فایل کلید خصوصی جدید است که با اجرای دستور ایجاد می‌گردد و در آن گذرواژه حذف شده است. لازم به ذکر است که برای هر یک از فایل‌های فوق، باید مسیر دقیق آن‌ها را در دستور وارد نماییم. برای مثال، در صورتی که فایل mydomain.key را در /etc/httpd/ssl/ قرار داده باشیم و می‌خواهیم فایل mydomain_nopass.key نیز در همین مسیر ذخیره شود، دستور به صورت زیر خواهد بود:

```
#Openssl rsa -in /etc/httpd/ssl/mydomain.key -out
/etc/httpd/ssl/mydomain_nopass.key
```