پیکربندی امن

Apache HTTP Server 2.2

مرکز مدیریت راهبردی افتا

SCWS-Apache-HTTP-2.2

اسفند ۹۵

فهرست

پیش‌گفتار

مرکز مدیریت راهبردی افتا[1] به منظور ساماندهی امنیت تجهیزات در حوزه فاوا[2]، پروژه «پیکربندی امن محصولات IT در کشور» را آغاز نموده است. یکی از گام‌های اساسی در این پروژه ارائه چک‌لیست و راهنمای پیکربندی امن برای محصولات IT می‌باشد. ارائه چک‌لیست برای محصولات داخلی بر عهده تولیدکننده محصول می‌باشد. تولید کننده ملزم است، چک‌لیست خود را در غالب ارائه شده از سمت مرکز افتا ارائه دهد. چک‌لیست‌های ارائه شده، توسط مرکز افتا مورد ارزیابی قرار گرفته و منتشر می‌گردد. سازمان‌های دولتی ملزم به استفاده از چک لیست‌های مذکور برای محصولات در حال استفاده خود هستند. همچنین سازمان‌های دولتی موظفند قبل از استفاده از محصولات IT، آن‌را مطابق چک‌لیست امنیتی مورد تایید مرکز افتا پیکربندی نمایند.

توجه به این نکته حائز اهمیت می‌باشد که چک‌لیست‌های ارائه شده، یک امنیت سطح پایه برای محصول ایجاد می‌نماید و سازمان‌ها ملزم هستند که برای رسیدن به سطح امنیت مورد نیاز خود، پس از اجرای مدیریت ریسک[3]، الزامات دیگری را نیز به این تنظیمات اضافه و مستند نمایند.

[1] امنیت فضای تولید و تبادل اطلاعات
[2] فناوری اطلاعات و ارتباطات
[3] Risk management

مقدمه

این سند راهنمایی برای پیکربندی امن Apache HTTP Server 2.2 است. در این سند مقادیر و تنظیمات امن برای سیاست‌های پیکربندی محصول مذکور ارائه شده است. مخاطب با استفاده از این سند توانایی پیاده‌سازی تنظیمات ارائه شده را خواهد داشت.

این سند توسط شرکت "فناوران توسعه امن ناجی" و به درخواست و تحت نظارت مرکز مدیریت راهبردی افتا تهیه گردیده است و از تلاش کارشناسان آن شرکت صمیمانه قدردانی می‌گردد. مرکز مدیریت راهبردی افتا ضمن استقبال از نظرات کارشناسان و متخصصان این حوزه برای غنای بیشتر این سند و دیگر اسناد مقاوم سازی، آمادگی دریافت پیشنهادات سازنده از طریق آدرس پست الکترونیکی Hardening@aftasec.ir را اعلام می‌دارد.

در ادامه، تنظیمات مورد نیاز برای پیکربندی امن Apache HTTP Server 2.2 آمده است. در این سند هر تنظیم با یک نام لاتین و شماره مختص آن آورده شده است. برای هر الزام دو بخش شرح اجمالی و نحوه پیاده‌سازی ارائه شده است. در بخش شرح اجمالی، توضیحی مختصر از ماهیت الزام بیان گردیده و در بخش نحوه پیاده‌سازی نیز، راهنمایی برای پیاده‌سازی الزام توسط مدیر سامانه ارائه شده است.

جدول ۱: گروه‌بندی و اختصارسازی نام برای محصولات IT

محصولات IT	
نام گروه	شماره گروه
نرم افزار آنتی ویروس	AV
سرویس‌دهنده نرم‌افزارهای کاربردی[۴]	AS
احراز اصالت[۵]	AU
اتوماسیون	AT
نرم‌افزار مدیریت پیکربندی[۶]	CM
سیستم مدیریت پایگاه داده	DB
نرم‌افزار کاربردی رومیزی[۷]	DA
سرویس گیرنده رومیزی[۸]	DC
سرویس دایرکتوری[۹]	DS
DNS سرور	DN
ایمیل سرور	ES
نرم‌افزار کاربردی سازمانی[۱۰]	EA
دیوار آتش[۱۱]	FI
تجهیزات قابل حمل[۱۲]	HD
مدیریت هویت[۱۳]	IM
سیستم تشخیص نفوذ[۱۴]	ID

[۴] Application Server
[۵] Authentication
[۶] Configuration Management System
[۷] Desktop Application
[۸] Desktop Client
[۹] Directory Service
[۱۰] Enterprise Application
[۱۱] Firewall
[۱۲] Handheld Device
[۱۳] Identity Management
[۱۴] Intrusion Detection System

محصولات IT	
نام گروه	شماره گروه
سرویس‌دهنده ایمیل[۱۵]	MS
راهکارهای موبایلی[۱۶]	MO
مسیریاب شبکه[۱۷]	RO
سوئیچ شبکه	SW
سیستم عامل	OS
تجهیزات جانبی[۱۸]	PD
سرویس دهنده[۱۹]	SR
نرم‌افزار مجازی‌سازی[۲۰]	VI
مرورگر وب	WB
سرویس‌دهنده وب	WS

[۱۵] Mail Server
[۱۶] Mobile Solution
[۱۷] Network Router
[۱۸] Peripheral Device
[۱۹] Server
[۲۰] Virtualization Software

تنظیمات

SCWS-1: برنامه‌ریزی و راه‌اندازی

SCWS-1-1: تهیه چک لیست برنامه‌ریزی پیش از راه‌اندازی

شرح اجمالی:

می‌بایست پیش و یا هنگام نصب و راه‌اندازی موارد و توصیه‌های زیر بررسی گردند:

- بررسی و اجرای سیاست‌های امنیتی داخلی سازمان و مرتبط با امنیت نرم‌افزارهای تحت وب
- پیاده‌سازی و ایجاد زیرساخت امن شبکه با توجه به کنترل دسترسی به/ از وب سرور و با استفاده از تجهیزاتی چون فایروال، روتر و سوییچ
- امن‌سازی سیستم عامل وب سرور بوسیله حداقل‌سازی سرویس‌های مربوط به شبکه، اعمال مناسب بسته‌های ارائه شده و پیکربندی امن آن با توجه به معیار امنیت اینترنتی توصیه شده برای هر پلتفرم
- پیاده‌سازی فرآیندهای نظارتی مرکزی مربوط به لاگ
- اجرای مکانیزم و فرآیندهای مربوط به پایش فضای حافظه و بازبینی لاگ‌ها در سرور
- آموزش و آگاهی رسانی برنامه نویسان به منظور توسعه برنامه‌های کاربردی امن و ادغام امنیت در چرخه عمر توسعه نرم افزار
- اطمینان از محفوظ ماندن اطلاعات ثبت شده مربوط به دامنه وب و عدم وجود اطلاعات حساس و حیاتی به منظور جلوگیری از حملات و خطراتی همچون مهندسی اجتماعی (نام‌های فردی POC)، War Dialing (شماره‌های تلفن) و Brute Force (آدرس‌های ایمیلی که مطابق با نام کاربری‌های سیستم واقعی هستند)
- پیکربندی صحیح و امن سرور DNS به منظور جلوگیری از حملات مخرب با توجه به توصیه‌های CIS BIND DNS Benchmark
- پیاده‌سازی یک سیستم تشخیص نفوذ شبکه (IDS) به منظور نظارت بر حملات علیه وب سرور

نحوه پیاده‌سازی:

مطابق با شرح اجمالی پیاده‌سازی گردد.

SCWS-1-2: عدم نصب سرویس‌های مختلف بر روی یک سرور

شرح اجمالی:

سرور اغلب طیف وسیعی از سرویس‌های پیش‌فرض غیر ضروری را ارائه می‌دهد که برخی از این سرویس‌ها، سرور را از لحاظ امنیتی در معرض خطرات جدی قرار می‌دهند. یک سرور می‌تواند خدمات زیادی انجام دهد این بدان معنا نمی‌باشد که می‌بایست بطور حتم همه این خدمات باید فعال شوند. لذا می‌بایست تعداد خدمات، سرویس‌ها و فرآیندهای پشت صحنه‌ای که بر روی سرور ای که میزبان Apache است، اجرا می‌شوند، به موارد ضروری محدود شده، و وب سرور تنها کارکرد اصلی سرور شود.

نحوه پیاده‌سازی:

به منظور اجرای این بند، می‌بایست سرویس‌های غیرضروری سیستم عامل خود را غیرفعال و یا حذف نمایید. در سیستم عامل لینوکس توزیع Red Hat می‌بایست فرمان زیر اجرا گردد.

```
chkconfig <servicename> off
```

SCSR-1-3: نصب Apache

شرح اجمالی:

معیارCIS Apache حاضر، در اکثر موارد استفاده از آپاچی باینری ارائه شده توسط فروشنده را به منظور کاهش دوباره‌کاری و افزایش اثربخشی تعمیر و نگهداری و امنیت پیشنهاد می‌کند. با این حال، برای حفظ معیار به صورت عمومی و قابل استفاده و انطباق با کلیه پلتفرم‌های یونیکس/ لینوکس، یک منبع پیش‌فرض برای این معیار ایجاد و استفاده شده است.

نحوه پیاده‌سازی:

نصب و راه اندازی بستگی زیادی به پلتفرم سیستم عامل دارد. در خصوص توزیع‌های سورس می‌توان از لینک http://httpd.apache.org/docs/2.2/install.html بهره برده و در خصوص توزیع Red Hat Enterprise Linux 5 می‌توان از دستور زیر استفاده نمود.

```
# yum install httpd
```

SCSR-2: به حداقل رساندن ماژول‌های Apache

SCSR-2-1: فعال نمودن ماژول‌های احراز هویت و اعطای مجوز ضروری

شرح اجمالی:

ماژول‌هایی با عنوان authn_* به منظور احراز هویت و ماژول‌های authz_* جهت ارائه مجوز می‌باشند. همچنین Apache شامل دو نوع احراز هویت basic و digest می‌باشد. لذا تنها ماژول‌هایی را که مورد نیاز می‌باشند، فعال نمایید.

نحوه پیاده‌سازی:

به منظور تصمیم‌گیری درخصوص نصب ماژول‌های لازم و ضروری از مستندات و توضیحاتی که برای هر ماژول موجود است، می‌توان استفاده نمود. البته برخی ماژول‌ها به صورت پیش‌فرض مورد نیاز نیستند و البته غیرفعال هستند. در ضمن یکسری از ماژول‌ها به صورت dynamic در آپاچی وجود دارند و بوسیله کامنت کردن یا حذف دستور LoadModule از فایل پیکربندی (معمولا http.conf) می‌توان آن‌ها را غیرفعال نمود.

SCSR-2-2: فعال نمودن ماژول پیکربندی لاگ

شرح اجمالی:

ماژول log_config قابلیت انعطاف ثبت وقایع درخواست‌های کلاینت را فراهم کرده و پیکربندی به منظور اینکه چه نوع اطلاعاتی درفایل لاگ ذخیره شود، فراهم می‌آورد.

نحوه پیاده‌سازی:

در راستای پیاده سازی وضعیت توصیه شده موارد زیر را اجرا نمایید:

- در خصوص توزیع‌های سورس با ماژول‌های static، اسکریپت configure./ آپاچی را بدون اینکه Script Option، disable-log-config-- فعال شده باشد، اجرا نمایید.

```
$ cd $DOWNLOAD/httpd-2.2.22
$ ./configure
```

- در خصوص ماژول‌هایی که به صورت پویا بارگذاری شده‌اند، دستور LoadModule در صورتی که در حال حاضر در پیکربندی آپاچی به صورت زیر کامنت گذاری نشده است، اضافه و یا اصلاح نمایید.

```
LoadModule log_config_module modules/mod_log_config.so
```

SCSR-2-3: غیرفعال نمودن ماژول WebDAV

شرح اجمالی:

ماژول‌های mod_dav و mod_dav_fs آپاچی از قابلیت WebDAV (احراز هویت و نسخه بندی توزیع شده مبتنی بر وب) برای آپاچی پشتیبانی می‌کنند. WebDAV یک افزونه برای پروتکل HTTP است که به کلاینت اجازه ایجاد، جابجایی، تغییر و یا حذف فایل‌ها و منابع را بر روی وب سرور می‌دهد.

نحوه پیاده‌سازی:

به منظور پیاده‌سازی این تنظیم و غیرفعال کردن WebDAV می‌بایست هر یک از موارد زیر را انجام داد:

۱. در خصوص توزیع‌های سورس با ماژول‌های static اسکریپت configure./. آپاچی را بدون اینکه آیتم‌های mod_dav و mod_dav_fs در Script Option مربوط به --enable-modules=configure فعال شده باشد، اجرا نمایید.

```
$ cd $DOWNLOAD/httpd-2.2.22
$ ./configure
```

۲. در خصوص ماژول‌هایی که به صورت پویا بارگذاری شده‌اند، دستور LoadModule را برای ماژول‌های mod_dav و mod_dav_fs از فایل httpd.conf حذف و یا کامنت گذاری نمایید.

```
##LoadModule dav_module modules/mod_dav.so
##LoadModule dav_fs_module modules/mod_dav_fs.so
```

SCSR-2-4: غیرفعال نمودن ماژول Status

شرح اجمالی:

ماژول mod_status در Apache وضعیت و آمار فعلی سرور را فراهم می‌کند.

نحوه پیاده‌سازی:

به منظور پیاده‌سازی این تنظیم و غیرفعال کردن mod_status می‌بایست هر یک از موارد زیر را انجام داد:

۱. در خصوص توزیع‌های سورس با ماژول‌های static، اسکریپت configure/. آپاچی را بدون اینکه Script Option مربوط به disable-status configure-- فعال شده باشد، اجرا نمایید.

```
$ cd $DOWNLOAD/httpd-2.2.22
$ ./configure --disable-status
```

۲. در خصوص ماژول‌هایی که به صورت پویا بارگذاری شده‌اند، دستور LoadModule را برای ماژول mod_status از فایل httpd.conf حذف و یا کامنت گذاری نمایید.

```
##LoadModule status_module modules/mod_status.so
```

SCSR-2-5: غیرفعال نمودن ماژول Autoindex

شرح اجمالی:

ماژول autoindex به طور خودکار صفحات وبی شامل محتویات دایرکتوری‌ها بر روی سرور را ایجاد می‌کند، که به طور معمول هنگامی استفاده می‌شوند که نیازی به ایجاد یک index.html نیست.

نحوه پیاده‌سازی:

به منظور پیاده‌سازی این تنظیم و غیرفعال کردن mod_autoindex می‌بایست هر یک از موارد زیر را انجام داد:

۱. در خصوص توزیع‌های سورس با ماژول‌های static، اسکریپت configure/. آپاچی را بدون اینکه Script Option مربوط به disable-autoindex configure-- فعال شده باشد، اجرا نمایید.

```
$ cd $DOWNLOAD/httpd-2.2.22
$ ./configure -disable-autoindex
```

۲. در خصوص ماژول‌هایی که به صورت پویا بارگذاری شده‌اند، دستور LoadModule را برای ماژول mod_autoindex از فایل httpd.conf حذف و یا کامنت گذاری نمایید.

```
## LoadModule autoindex_module modules/mod_autoindex.so
```

SCSR-2-6: غیرفعال نمودن ماژول Proxy

شرح اجمالی:

سرور با استفاده از ماژول‌های پروکسی Apache به عنوان یک پروکسی (Forward proxy یا reverse proxy) از HTTP و پروتکل‌های دیگر با ماژول‌های دیگر پروکسی بارگذاری می‌شود. اگر نصب Apache برای درخواست به یا از شبکه دیگری در نظر گرفته نشده باشد، در آن صورت ماژول پروکسی نباید بارگذاری شود.

نحوه پیاده‌سازی:

به منظور پیاده‌سازی این تنظیم و غیرفعال کردن ماژول proxy می‌بایست هر یک از موارد زیر را انجام داد:

۱. در خصوص توزیع‌های سورس با ماژول‌های static اسکریپت configure./. آپاچی را بدون اینکه آیتم mod_proxy در Script Option مربوط به enable-modules=configure-- فعال شده باشد، اجرا نمایید.

```
$ cd $DOWNLOAD/httpd-2.2.22
$ ./configure
```

۲. در خصوص ماژول‌هایی که به صورت پویا بارگذاری شده‌اند، دستور LoadModule را برای ماژول mod_proxy و هر ماژول پروکسی دیگر از فایل httpd.conf حذف و یا کامنت گذاری نمایید.

```
##LoadModule proxy_module modules/mod_proxy.so
##LoadModule proxy_balancer_module modules/mod_proxy_balancer.so
##LoadModule proxy_ftp_module modules/mod_proxy_ftp.so
##LoadModule proxy_http_module modules/mod_proxy_http.so
##LoadModule proxy_connect_module modules/mod_proxy_connect.so
##LoadModule proxy_ajp_module modules/mod_proxy_ajp.so
```

SCSR-2-7: غیرفعال نمودن ماژول‌های دایرکتوری‌های کاربر

شرح اجمالی:

می‌بایست دستور UserDir غیرفعال شود تا بدین وسیله دایرکتوری‌های خانگی کاربر از طریق وب با یک علامت مد (~) قبل از نام کاربری، قابل دسترسی نباشند. این دستور همچنین نام مسیر دایرکتوری قابل دسترس را نشان می‌دهد.. به عنوان مثال:

- http://example.com/~ralph/ ممکن است به یک زیردایرکتوری public_html از دایرکتوری خانگی کاربر ralph دسترسی داشته باشد.
- دستور UserDir./. ممکن است root~/ را به دایرکتوری ریشه (/) نگاشت کند.

نحوه پیاده‌سازی:

به منظور پیاده‌سازی این تنظیم و غیرفعال کردن ماژول user directories می‌بایست هر یک از موارد زیر انجام گیرد:

۱. در خصوص توزیع‌های سورس با ماژول‌های static اسکریپت configure./. آپاچی را با آیتم mod_proxy در Script Option مربوط به configure --disable-userdir اجرا نمایید.

```
$ cd $DOWNLOAD/httpd2.2.22
$ ./configure --disable-userdir
```

۲. در خصوص ماژول‌هایی که به صورت پویا بارگذاری شده‌اند، دستور LoadModule را برای ماژول mod_userdir از فایل httpd.conf حذف و یا کامنت‌گذاری نمایید.

```
##LoadModule userdir_module modules/mod_userdir.so
```

SCSR-2-8: غیرفعال نمودن ماژول info

شرح اجمالی:

ماژول mod_info اطلاعاتی در مورد پیکربندی سرور از طریق دسترسی به server-info/، ارائه می‌کند.

نحوه پیاده‌سازی:

به منظور پیاده‌سازی این تنظیم و غیرفعال کردن ماژول mod_info می‌بایست هر یک از موارد زیر را انجام داد:

۱. در خصوص توزیع‌های سورس با ماژول‌های static اسکریپت configure./. آپاچی را بدون اینکه آیتم mod_info در Script Option مربوط به configure=enable-modules-- فعال شده باشد، اجرا نمایید.

```
$ cd $DOWNLOAD/httpd2.2.22
$ ./configure
```

۲. در خصوص ماژول‌هایی که به صورت پویا بارگذاری شده‌اند، دستور LoadModule را برای ماژول mod_info از فایل httpd.conf حذف و یا کامنت گذاری نمایید.

```
##LoadModule info_module modules/mod_info.so
```

SCSR-3: اصول، مجوزها و مالکیت

SCSR-3-1: اجرای وب سرور Apache با کاربری غیر root

شرح اجمالی:

اگرچه آپاچی معمولاً با دسترسی‌های root به منظور شنود به پورت ۸۰ و ۴۴۳ آغاز می‌شود، اما به منظور انجام سرویس‌دهی می‌تواند و می‌بایست کاربری غیر از کاربر ریشه اجرا شود. از دستورهای User و Group به منظور تعیین کاربر و گروهی که ایجاد کننده فرآیندهای آپاچی فرض می‌شوند، استفاده می‌گردد.

نحوه پیاده‌سازی:

در راستای پیاده سازی وضعیت توصیه شده موارد زیر را اجرا نمایید:

۱. در صورتی که Apache User و Apache Group ایجاد نگردیده است، یک حساب کاربری و گروه منحصر بفرد را ایجاد نمایید:

```
# groupadd -r apache
# useradd apache -r -g apache -d /var/www -s /sbin/nologin
```

۲. پیکربندی User و Group در فایل پیکربندی httpd.conf آپاچی:

```
User apache
Group apache
```

SCSR-3-2: درنظر گرفتن پوسته نامعتبر برای حساب کاربری آپاچی

شرح اجمالی:

نباید از حساب apache به عنوان یک حساب کاربری ورود عادی استفاده شود، و می‌بایست یک پوسته نامعتبر یا nologin به منظور اطمینان از عدم استفاده از این حساب برای ورود در نظر گرفته شود.

نحوه پیاده‌سازی:

تغییر حساب apache به استفاده از یک پوسته nologin یا یک پوسته نامعتبر همانند /dev/null :

```
# chsh -s /sbin/nologin apache
```

SCSR-3-3: قفل نمودن حساب کاربری Apache

شرح اجمالی:

حساب کاربری که آپاچی تحت آن اجرا می‌شود نباید یک رمز عبور معتبر داشته باشد، اما می‌بایست غیرفعال گردد.

نحوه پیاده‌سازی:

از دستور passwd به منظور غیرفعال نمودن حساب کاربری آپاچی استفاده نمایید:

```
# passwd -l apache
```

SCSR-3-4: تنظیم مالکیت بر روی دایرکتوری‌ها و فایل‌های Apache

شرح اجمالی:

دایرکتوری‌ها و فایل‌های آپاچی باید متعلق به root باشند. این امر در مورد تمام دایرکتوری‌های نرم‌افزار آپاچی و فایل‌های نصب شده نیز صدق می‌کند.

نحوه پیاده‌سازی:

پیاده‌سازی موارد زیر:

تنظیم مالکیت در دایرکتوری‌های $APACHE_PREFIX مانند /usr/local/apache2:

```
$ chown -R root $APACHE_PREFIX
```

SCSR-3-5: تنظیم شناسه گروه بر روی دایرکتوری‌ها و فایل‌های Apache

شرح اجمالی:

می‌بایست دایرکتوری‌ها و فایل‌های آپاچی برای داشتن یک شناسه گروه از root، یا یک گروه معادل root تنظیم شوند. این امر در مورد تمام دایرکتوری‌های نرم‌افزار آپاچی و فایل‌های نصب شده نیز به کار برده می‌شود. تنها مورد استثنا این است که ریشه اصلی پوشه apache ($APACHE_PREFIX/htdocs) به یک گروه تعیین شده نیاز دارد تا به محتوای وب اجازه دهد از طریق یک فرآیند مدیریت تغییر (مانند webupdate)، بروزرسانی شود.

نحوه پیاده‌سازی:

پیاده‌سازی موارد زیر:

تنظیم مالکیت در دایرکتوری‌های $APACHE_PREFIX مانند /usr/local/apache2:

```
$ chown -R root $APACHE_PREFIX
```

SCSR-3-6: محدود کردن دسترسی‌های Write بر روی دایرکتوری‌ها و فایل‌های Apache

شرح اجمالی:

به طور کلی مجوزهای دایرکتوری‌های آپاچی و مجوزهای فایل می‌بایست rwxr-xr-x (755) باشد، مگر این که نیاز به دسترسی اجرایی داشته باشد. این امر در مورد دایرکتوری‌های نرم‌افزار آپاچی و فایل‌های نصب شده به استثنا ریشه اصلی پوشه apache $APACHE_PREFIX/htdocs اعمال می‌شود. دایرکتوری‌ها و فایل‌ها در مسیر اصلی پوشه ممکن است به یک گروه با دسترسی نوشتن انتساب شوند تا به محتوای وب اجازه بروزرسانی دهند. علاوه بر این، می‌بایست دایرکتوری /bin و قابلیت اجرایی به طوری تنظیم گردد که توسط سایرین قابل خواندن نباشد.

نحوه پیاده‌سازی:

انجام مورد زیر به منظور حذف دیگر دسترسی روی دایرکتوری‌های $APACHE_PREFIX.

```
# chmod -R o-w $APACHE_PREFIX
```

SCSR-3-7: امن نمودن دایرکتوری Core Dump

شرح اجمالی:

دستور CoreDumpDirectory برای مشخص کردن مسیری که تلاش‌های آپاچی را به منظور ایجاد core dump، نشان می‌دهد، بکار می‌رود. دایرکتوری پیش‌فرض، دایرکتوری ServerRoot می‌باشد، با این حال core dumpها در سیستم‌های لینوکسی به صورت پیش‌فرض غیرفعال است. در رویدادهایی که نیاز به core dumpها می‌باشد، می‌بایست دایرکتوری قابلیت نوشتن را توسط آپاچی داشته باشد و باید نیازمندی‌های امنیتی که در پایین و در قسمت پیاده‌سازی تعریف شده است را تامین کند.

نحوه پیاده‌سازی:

دستور CoreDumpDirectory را از فایل‌های پیکربندی Apache حذف کنید یا از اینکه دایرکتوری پیکربندی‌شده از شرایط زیر برخوردار است اطمینان حاصل کنید.

۱. CoreDumpDirectory، در مسیر اصلی پوشه apache وجود ندارد. ($APACHE_PREFIX/htdocs)
۲. بایستی در ریشه بوده و مالکیت یک گروه Apache را داشته باشد (همانطور که به وسیله‌ی دستور گروه، تعریف شده)

```
# chown root:apache /var/log/httpd
```

۳. نباید هیچ مجوز دسترسی read-write-search برای دیگر کاربران وجود داشته باشد.

```
# chmod o-rwx /var/log/httpd
```

SCSR-3-8: امن نمودن Lock File

شرح اجمالی:

دستور LockFile سبب ایجاد مسیری به فایل Lock در زمانی می‌شود که آپاچی از دستورهای فراخوانی سیستمی مانند fcntl(2) یا flock(2) برای پیاده‌سازی یک mutex (قفل) استفاده می‌کند.

اکثر سیستم‌های لینوکس به طور پیش‌فرض از سمافورها به جای آن استفاده می‌کنند، در صورتی که ممکن است دستوری اعمال و یا به کار برده نشود. با این حال مسئله مهم این است که، در صورتی که یک فایل Lock مورد استفاده قرار گرفت، این فایل در یک دایرکتوری محلی باشد که توسط دیگر کاربران writable نباشد.

نحوه پیاده‌سازی:

۱. مسیر دایرکتوری‌ای را پیدا کنید که LockFile در آن ایجاد می‌شود که به صورت پیش‌فرض، دایرکتوری ServerRoot/logs است.
۲. اگر مسیر یک دایرکتوری در DocumentRoot است، دایرکتوری را اصلاح نمایید.
۳. مالکیت و گروه را به ریشه تغییر دهید: اگر در حال حاضر، ریشه به حساب نمی‌آید.
۴. مجوزها را تغییر دهید، به طوری که دایرکتوری تنها توسط ریشه یا کاربری که Apache ابتدا با او آغاز به کار می‌کند، قابل نوشتن باشد (مقدار پیش‌فرض، ریشه است).
۵. بررسی نمایید که دایرکتوری فایل قفل، به جای یک سیستم فایل نصب‌شده‌ی NFS، بر روی یک هارددیسک نصب‌شده‌ی محلی باشد.

SCSR-3-9: امن نمودن Pid فایل

شرح اجمالی:

دستور PidFile مسیر فایل process ID را تنظیم می‌کند که سرور برای آن Process ID سرور را ثبت می‌کند، که البته برای ارسال یک سیگنال به Process سرور یا بررسی سلامت Process مفید است.

نحوه پیاده‌سازی:

۱. دایرکتوری‌ای را پیدا کنید که PidFile در آن ایجاد می‌شود. مقدار پیش‌فرض، دایرکتوری ServerRoot/logs است.
۲. اگر PidFile در یک دایرکتوری در Apache DocumentRoot است، دایرکتوری را اصلاح کنید.
۳. مالکیت و گروه را به ریشه تغییر دهید: در صورتی که در حال حاضر ریشه نیست.
۴. مجوزها را طوری را تغییر دهید که دایرکتوری، تنها توسط ریشه یا کاربری که Apache ابتدا با او آغاز به کار می‌کند، قابل نوشتن باشد (مقدار پیش‌فرض، ریشه می‌باشد).

SCSR-3-10: امن نمودن فایل ScoreBoard

شرح اجمالی:

دستور ScoreBoardFile مسیر فایلی را تنظیم می‌کند که سرور برای ارتباطات بین Process (IPC) در میان Processهای آپاچی از آن استفاده می‌کند. در اکثر پلتفرم‌های لینوکس، حافظه به اشتراک گذاشته شده، به جای یک فایل در سیستم‌فایل مورد استفاده قرار خواهد گرفت، بنابراین به طور کلی این دستور مورد نیاز نبوده و لازم

نیست تا دایرکتوری مجزایی تعیین شود. با این حال، اگر دستور مشخص باشد در اینصورت آپاچی از فایل پیکربندی شده برای ارتباطات بین Process استفاده خواهد کرد. هرچند که در صورت محرز شدن این موضوع می‌بایست در یک دایرکتوری امن قرار داده شود.

نحوه پیاده‌سازی:

۱. بررسی نمایید که آیا ScoreBoardFile در هیچ یک از فایل‌های پیکربندی Apache، مشخص شده است یا خیر. اگر جواب منفی بود، هیچ تغییری لازم نیست.
۲. اگر دستور، موجود باشد، دایرکتوری‌ای را پیدا کنید که ScoreBoardFile در آن ایجاد می‌شود. مقدار پیش‌فرض، دایرکتوری ServerRoot/logs است.
۳. اگر ScoreBoardFile در یک دایرکتوری در Apache DocumentRoot است، دایرکتوری را اصلاح نمایید.
۴. اگر مالکیت و گروه root نیست آن را به root:root تغییر دهید.
۵. مجوزها را به گونه‌ای تغییر دهید که دایرکتوری، تنها توسط root یا کاربری که Apache ابتدا با او آغاز به کار می‌کند، قابل نوشتن باشد (مقدار پیش‌فرض، ریشه است).
۶. بررسی نمایید که دایرکتوری فایل scoreboard، به جای یک سیستم فایل نصب شده‌ی NFS، بر روی یک هارد دیسک نصب شده‌ی محلی باشد.

SCSR-3-11: محدود کردن دسترسی‌های نوشتن گروه بر روی دایرکتوری‌ها و فایل‌های Apache

شرح اجمالی:

به طور کلی می‌بایست مجوزهای گروهی برای دایرکتوری‌های آپاچی، r-x بوده و مجوزهای فایل نیز می‌بایست همین باشند، مگر اینکه قابل اجرا نباشند و قابل اجرا بودن آن‌ها نیز مناسب نباشد. این امر در مورد تمام دایرکتوری‌ها و فایل‌های نصب شده نرم‌افزار آپاچی به استثناء ریشه اصلی پوشه DOCROOT$ تعریف شده توسط DocumentRoot آپاچی و به طور پیش‌فرض برای APACHE_PREFIX/htdocs$ نیز به کار می‌رود. ممکن است یک گروه توسعه وب برای دسترسی write دایرکتوری‌ها و فایل‌ها در مسیر اصلی پوشه تعیین شوند که به محتوای وب اجازه بروزرسانی داده شود.

نحوه پیاده‌سازی:

به منظور حذف دسترسی نوشتن گروه بر روی فایل‌ها و دایرکتوری‌های $APACHE_PREFIX، دستور زیر را اجرا نمایید.

```
# chmod -R g-w $APACHE_PREFIX
```

SCSR-3-12: محدود نمودن دسترسی Write گروه بر روی ریشه اصلی پوشه و فایل‌ها

شرح اجمالی:

ممکن است لازم باشد مجوزهای گروهی برای دایرکتوری‌های $DOCROOT آپاچی توسط یک گروه مجاز مانند توسعه، پشتیبانی، یا یک ابزار مدیریت محتوای تولید، قابل نوشتن باشند. با این حال، نکته مهم این است که گروه آپاچی مورد استفاده برای اجرای سرور، دسترسی نوشتن بر روی هیچ یک از دایرکتوری‌ها یا فایل‌ها موجود در ریشه سند را نداشته باشد.

نحوه پیاده‌سازی:

به منظور حذف دسترسی نوشتن گروه بر روی فایل‌ها و دایرکتوری‌های $DOCROOT با گروه Apache، دستور زیر را اجرا نمایید.

```
# find -L $DOCROOT -group $GRP -perm /g=w -print | xargs chmod g-w
```

SCSR-4: کنترل دسترسی آپاچی

SCSR-4-1: منع دسترسی به دایرکتوری ریشه OS

شرح اجمالی:

دستور Directory، به منظور پیکربندی خاص دایرکتوری؛ کنترل‌های دسترسی و بسیاری از ویژگی‌ها و گزینه‌های دیگر را میسر می‌سازد. یک استفاده مهم که باعث ایجاد یک سیاست انکار پیش‌فرض می‌گردد این است که اجازه دسترسی به دایرکتوری‌ها و فایل‌های سیستم عامل جز آن‌هایی که به طور خاص مجاز هستند را نمی‌دهد. این کار با مسدود کردن دسترسی به دایرکتوری ریشه سیستم عامل با استفاده از یکی از دو روش زیر می‌باشد که البته تنها یکی از این دو دستور باید انجام شود و مجاز به استفاده از هر دو نیستید:

۱. استفاده از دستور Apache Deny به همراه یک دستور Order
۲. استفاده از دستور Apache Require

هر دو روش موثر هستندو در حال حاضر ترکیبی از Order/Deny/Allow توصیه نمی‌گردد، زیرا آن‌ها سه راه را فراهم می‌سازند که تمامی دستورات طی یک دستور مشخص شده مورد پردازش قرار می‌گیرند. در مقابل، دستور Require بر روی اولین تطابق، مشابه با نحوه عمل Rule در فایروال، عمل می‌کند. دستور Require به صورت پیش‌فرض در آپاچی نسخه 2.4 می‌باشد.

نحوه پیاده‌سازی:

در راستای پیاده سازی وضعیت توصیه شده موارد زیر را اجرا نمایید:

۱. فایل‌های پیکربندی Apache را به منظور یافتن یک عنصر <Directory> ریشه، جستجو نمایید (شامل هر httpd.conf و هر فایل پیکربندی موجود در آن)
۲. یک دستور Require منفرد را اضافه کرده و مقدار آن را all denied تنظیم نمایید.
۳. هر دستور Deny و Allow را از عنصر <Directory> ریشه، حذف نمایید.

```
<Directory>
  . . .
  Require all denied
  . . .
</Directory>
```

SCSR-4-2: اجازه دسترسی مناسب به محتوای وب

شرح اجمالی:

به منظور سرویس‌دهی محتوای وب، می‌بایست از دستور Allow استفاده نموده تا امکان دسترسی مناسب به دایرکتوری‌ها، مکان‌ها و میزبان‌های مجازی که حاوی محتوای وب هستند را میسر سازد.

نحوه پیاده‌سازی:

در راستای پیاده‌سازی وضعیت توصیه شده موارد زیر را اجرا نمایید:

۱. به منظور یافتن همه‌ی عناصر <Directory> و <Location>، فایل‌های پیکربندی Apache را جستجو نمایید (شامل httpd.conf و هر فایل پیکربندی موجود در آن). بایستی یک عنصر برای پوشه اصلی و هر

دایرکتوری یا لوکیشن خاص، وجود داشته باشد. احتمالا دستورهای کنترل دسترسی دیگری نیز در زمینه‌های مختلف، از جمله میزبان‌های مجازی یا عناصر خاص مانند <Proxy> وجود داشته باشند.

۲. شامل دستورهای مناسب Require، با مقادیر متناسب با نوع استفاده هر دایرکتوری می‌باشد.

پیکربندی‌های زیر، تنها چند نمونه ممکن هستند:

```
<Directory "/var/www/html/">
   Require ip 192.169.
</Directory>

<Directory "/var/www/html/">
    Require all granted
</Directory>

<Location /usage>
    Require local
</Location>

<Location /portal>
     Requirevalid-user
</Location>
```

SCSR-4-3: محدودسازی OverRide برای دایرکتوری ریشه OS

شرح اجمالی:

دستور OverRide، به فایل‌های htaccess. اجازه می‌دهند تا بدون در نظر گرفتن بسیاری از پیکربندی‌ها، از جمله احراز هویت، کنترل انواع سند، ایندکس‌های تولید شده به طور خودکار، کنترل دسترسی و Optionها مورد استفاده قرار گیرند. هنگامی که سرور یک فایل htaccess. پیدا می‌کند (که توسط AccessFileName مشخص شده)، می‌بایست از اینکه کدام دستورهای اعلام شده در آن فایل می‌توانند اطلاعات دسترسی قبلی را نادیده بگیرد، مطلع باشد. هنگامی که این دستور بر روی None تنظیم شود، در آن صورت فایل‌های htaccess. به طور کامل نادیده گرفته می‌شوند. در اینصورت، سرور حتی برای خواندن فایل‌های htaccess. در فایل‌سیستم هیچ تلاشی نمی‌کند. لذا چنانچه این دستور بر روی All تنظیم شود، در اینصورت هر دستوری که یک زمینه htaccess. داشته باشد، در فایل‌های htaccess. مجاز می‌باشد.

به منظور کسب جزئیات بیشتر به مستندات مربوط به Apache 2.2 مراجعه نمایید:

*نحوه پیاده‌سازی:*http://httpd.apache.org/docs/2.2/mod/core.html#allowoverride

در راستای پیاده سازی وضعیت توصیه شده موارد زیر را اجرا نمایید:

۱. فایل‌های پیکربندی Apache را به منظور یافتن یک عنصر <Directory> ریشه، جستجو نمایید (شامل httpd.conf و هر فایل پیکربندی موجود در آن)
۲. یک دستور منفرد AllowOverride درصورت عدم وجود، اضافه نمایید.
۳. مقدار AllowOverride را none تنظیم نمایید.

```
<Directory />
  . . .
  AllowOverride None
  . . .
</Directory>
```

SCSR-4-4: محدودسازی OverRide برای کلیه دایرکتوری‌ها

شرح اجمالی:

دستور AllowOverRide، به فایل‌های htaccess. اجازه می‌دهند تا بدون در نظر گرفتن بسیاری از پیکربندی‌ها، از جمله احراز هویت، کنترل انواع سند، ایندکس‌های تولید شده به طور خودکار، کنترل دسترسی و Optionها مورد استفاده قرار گیرند. هنگامی که سرور یک فایل htaccess. پیدا می‌کند (که توسط AccessFileName مشخص شده)، می‌بایست از اینکه کدام دستورهای اعلام شده در آن فایل می‌توانند اطلاعات دسترسی قبلی را نادیده بگیرد، مطلع باشد. هنگامی که این دستور بر روی None تنظیم شود، در آن صورت فایل‌های htaccess. به طور کامل نادیده گرفته می‌شوند. در اینصورت، سرور حتی برای خواندن فایل‌های htaccess. در فایل‌سیستم هیچ تلاشی نمی‌کند. لذا چنانچه این دستور بر روی All تنظیم شود، در اینصورت هر دستوری که یک زمینه htaccess. داشته باشد، در فایل‌های htaccess. مجاز می‌باشد.

به منظور کسب جزئیات بیشتر به مستندات مربوط به Apache 2.2 مراجعه نمایید:

http://httpd.apache.org/docs/2.2/mod/core.html#allowoverride

نحوه پیاده‌سازی:

۱. فایل‌های پیکربندی Apache را به منظور یافتن AllowOverride، جستجو نمایید (شامل httpd.conf و هر فایل پیکربندی موجود در آن)
۲. مقدار همه دستورهای AllowOverride را none تنظیم نمایید.

```
. . .
AllowOverride None
. . .
```

SCSR-5: به حداقل رساندن قابلیت‌ها، محتوا و گزینه‌ها

SCSR-5-1: محدود نمودن Optionها برای دایرکتوری ریشه OS

شرح اجمالی:

دستور Options آپاچی اجازه پیکربندی خاص Optionها، از جمله اجرای CGI، دنبال کردن Symbolic linkها، دستورات سمت سرور و content negotiation را می‌دهد.

به منظور کسب جزئیات بیشتر به مستندات مربوط به Apache 2.2 مراجعه نمایید:

http://httpd.apache.org/docs/2.2/mod/core.html#options

نحوه پیاده‌سازی:

در راستای پیاده سازی وضعیت توصیه شده موارد زیر را اجرا نمایید:

۱. فایل‌های پیکربندی Apache را به منظور یافتن یک عنصر <Directory> ریشه، جستجو نمایید (شامل httpd.conf و هر فایل پیکربندی موجود در آن)
۲. یک دستور منفرد Options درصورت عدم وجود، اضافه نمایید.
۳. مقدار Options را none تنظیم نمایید.

```
<Directory>
    . . .
    Options None
    . . .
</Directory>
```

SCSR-5-2: محدود نمودن Optionها برای دایرکتوری ریشه وب

شرح اجمالی:

دستور Options اجازه پیکربندی خاص Optionها را می‌دهد، از جمله می‌توان به موارد زیر اشاره نمود:

- اجرای CGI
- دنبال کردن Symbolic linkها
- دستورات سمت سرور و
- content negotiation

به منظور کسب جزئیات بیشتر به مستندات مربوط به Apache 2.2 مراجعه نمایید:

http://httpd.apache.org/docs/2.2/mod/core.html#options

نحوه پیاده‌سازی:

در راستای پیاده سازی وضعیت توصیه شده موارد زیر را اجرا نمایید:

۱. فایل‌های پیکربندی Apache را به منظور یافتن یک عنصر <Directory> ریشه، جستجو نمایید (شامل httpd.conf و هر فایل پیکربندی موجود در آن)
۲. اگر به multiviews نیاز می‌باشد، می‌بایست هر دستور Option ایجاد شده، دارای مقدار None یا Multiviews باشد.

```
<Directory "/usr/local/apache2/htdocs">
    . . .
    Options None
    . . .
</Directory>
```

SCSR-5-3: به حداقل رساندن Optionها برای دیگر دایرکتوری‌ها

شرح اجمالی:

دستور Options اجازه پیکربندی خاص Optionها، از جمله اجرای CGI، دنبال کردن Symbolic linkها، دستورات سمت سرور و content negotiation را می‌دهد.

به منظور کسب جزئیات بیشتر به مستندات مربوط به Apache 2.2 مراجعه نمایید:

http://httpd.apache.org/docs/2.2/mod/core.html#options

نحوه پیاده‌سازی:

در راستای پیاده سازی وضعیت توصیه شده موارد زیر را اجرا نمایید:

۱. فایل‌های پیکربندی Apache را به منظور یافتن یک عنصر <Directory> ریشه، جستجو نمایید (شامل httpd.conf و هر فایل پیکربندی موجود در آن)
۲. هر دستور Option موجود که دارای مقدار includes نمی‌باشد را اصلاح یا اضافه نمایید.

دیگر Optionها را در صورتی که ضرورت دارد همانند بالا به صورت مناسب پیاده‌سازی کنید.

SCSR-5-4: حذف نمودن محتوای پیش‌فرض HTML

شرح اجمالی:

محتواهای پیش‌فرضی برای نصب و راه‌اندازی آپاچی موجود است که البته برای استفاده، مناسب و مورد نیاز نمی‌باشد. عملکرد اصلی برای این محتوای نمونه، ارائه یک وب سایت پیش‌فرض، ارائه دفترچه راهنمای کاربر یا نشان دادن ویژگی‌های خاص وب سرور است که می‌بایست تمام محتوایی که مورد نیاز نمی‌باشد حذف گردد.

نحوه پیاده‌سازی:

تمام محتوای از پیش نصب شده را مرور کرده و محتوایی که مورد نیاز نمی‌باشد را حذف نمایید. به خصوص، به دنبال محتوای غیرضروری باشید که ممکن است در یک مسیراصلی یافت شود، یک دایرکتوری پیکربندی می‌تواند دایرکتوری conf/extra یا یک بسته‌ی Unix/Linux باشد.

۱. Index.html یا صفحه welcome به طوری که به صورت پیش‌فرض نصب بوده را حذف و یا به کامنت تبدیل نمایید. در ضمن حذف یک فایل مانند welcome.conf که به صورت زیر نشان داده شده، توصیه نمی‌شود، زیرا با بروزرسانی بسته، ممکن است مجددا جایگزین شود.

```
#
# This configuration file enables the default "Welcome"
# page if there is no default index page present for
# the root URL. To disable the Welcome page, comment
# out all the lines below.
#
##<LocationMatch "^/+$">
## Options –Indexes
```

```
## ErrorDocument 403 /error/noindex.html
##</LocationMatch>
```

۲. محتوای user manual آپاچی را حذف کرده یا پیکربندی‌هایی که به بارگذاری دستی ارجاع می‌دهند را به کامنت تبدیل نمایید.

```
# yum erase httpd-manual
```

۳. هر پیکربندی کنترل‌کننده وضعیت سیستم را حذف یا به کامنت تبدیل نمایید.

```
#
# Allow server status reports generated by mod_status,
# with the URL of http://servername/server-status
# Change the ".example.com" to match your domain to enable.
#
##<Location /server-status>
##          SetHandler server-status
##          Order deny,allow
##          Deny from all
##         Allow from .example.com
##</Location>
```

۴. هر پیکربندی کنترل کننده اطلاعات سیستم را حذف یا به کامنت تبدیل نمایید.

```
#
# Allow remote server configuration reports, with the URL of
# http://servername/server-info (requires that mod_info.c be loaded).
# Change the ".example.com" to match your domain to enable.
#
##<Location /server-info>
##        SetHandler server-info
##        Order deny,allow
##        Deny from all
##        Allow from .example.com
##</Location>
```

۵. هر پیکربندی کنترل کننده دیگری مانند perl-status را حذف یا به کامنت تبدیل نمایید.

```
# This will allow remote server configuration reports, with the URL of
# http://servername/perl-status
# Change the ".example.com" to match your domain to enable.
```

```
#
##<Location /perl-status>
##        SetHandler perl-script
##        PerlResponseHandler Apache2:: Status
##        Order deny,allow
##        Deny from all
##       Allow from .example.com
##</Location>
```

SCSR-5-5: حذف CGI Content Printenv پیش‌فرض

شرح اجمالی:

اکثر وب سروها از جمله بسته نصبی آپاچی به صورت پیش‌فرض دارای محتوای CGI می‌باشند که برای کاربری وب سرور الزامی و مقتضی نمی‌باشد. کاربرد اصلی این برنامه نمونه شبیه‌سازی قابلیت‌های وب سرور می‌باشد. یکی از محتواهای CGI پیش‌فرض برای بسته نصبی آپاچی، اسکریپت prinenv می‌باشد. این اسکریپت تمام مقادیر محیطی CGI را به درخواست کننده بر می‌گرداند که شامل تعداد زیادی توضیحات در مورد پیکربندی سیستم و مسیرهای سیستمی می‌باشد.

نحوه پیاده‌سازی:

در راستای پیاده سازی وضعیت توصیه شده موارد زیر را اجرا نمایید:

۱. محل فایل‌ها و دایرکتوری‌های cgi-bin فعال‌شده در پیکربندی Apache را از طریق دستورهای Script، ScriptAlias، ScriptAliasMatch یا ScriptInterpreterSource را تعیین نمایید.
۲. در صورتی که printenv در دایرکتوری cgi-bin نصب شده بود، آن را حذف نمایید.

```
# rm $APACHE_PREFIX/cgi-bin/printenv
```

SCSR-5-6: حذف CGI Content test-cgi پیش‌فرض

شرح اجمالی:

اکثر وب سروها از جمله بسته نصبی آپاچی به صورت پیش‌فرض دارای محتوای CGI می‌باشند که برای کاربری وب سرور الزامی و مقتضی نمی‌باشد. کاربرد اصلی این برنامه نمونه شبیه‌سازی قابلیت‌های وب سرور می‌باشد. یکی

از محتواهای CGI پیش‌فرض برای بسته نصبی آپاچی اسکریپت test-cgi می‌باشد. این اسکریپت تمام مقادیر محیطی CGI را به درخواست کننده بر می‌گرداند که شامل تعداد زیادی توضیحات در مورد پیکربندی سیستم می‌باشد.

نحوه پیاده‌سازی:

در راستای پیاده سازی وضعیت توصیه شده موارد زیر را اجرا نمایید:

۱. محل فایل‌ها و دایرکتوری‌های cgi-bin فعال‌شده در پیکربندی Apache از طریق دستورهای Script، ScriptAlias، ScriptAliasMatch یا ScriptInterpreterSource را تعیین نمایید.
۲. در صورتی که test-cgi در دایرکتوری cgi-bin نصب شده بود، آن را حذف نمایید.

```
# rm $APACHE_PREFIX/cgi-bin/test-cgi
```

SCSR-5-7: محدود نمودن روش‌های HTTP Request

شرح اجمالی:

با استفاده از دستور <LimitExcept> روش‌های HTTP Request غیر ضروری وب سرور را محدود نموده تا تنها روش‌های درخواست GET، HEAD، POST و OPTIONS HTTP قبول و پردازش شوند.

نحوه پیاده‌سازی:

در راستای پیاده سازی وضعیت توصیه شده موارد زیر را اجرا نمایید:

۱. محل فایل‌های پیکربندی Apache و فایل‌های پیکربندی گنجانده شده در آن را تعیین نمایید.
۲. دستور را در مسیر اصلی پوشه جستجو نمایید، مانند:

```
<Directory "/usr/local/apache2/htdocs">
        . . .
</Directory>
```

۱. دستوری مانند آنچه در زیر نشان داده شده را به گروه دستورهای document root، اضافه نمایید.

```
# Limit HTTP methods to standard methods. Note: Does not limit TRACE
<LimitExcept GET POST OPTIONS>
```

```
        Require all denied
</LimitExcept>
```

۱. در فایل‌های پیکربندی Apache، دستورهای دیگری به جز دایرکتوری ریشه‌ی OS را جستجو کرده و دستورهای مشابهی را به هر کدام اضافه نمایید. درک این مطلب بسیار مهم است که دستورها مبتنی بر سلسله مراتب سیستم‌فایل OS هستند، نه سلسله مراتب مکان‌هایی در URLهای وب سایت که توسط Apache مورد دسترسی قرار می‌گیرند.

```
<Directory "/usr/local/apache2/cgi-bin">
      . . .
      # Limit HTTP methods
      <LimitExcept GET POST OPTIONS>
            Require all denied
      </LimitExcept>
</Directory>
```

SCSR-5-8: غیر فعال کردن روش HTTP TRACE

شرح اجمالی:

از دستور TraceEnable به منظور غیرفعال کردن روش درخواست HTTP TRACE استفاده گردد.

به منظور کسب جزئیات بیشتر به مستندات مربوط به Apache مراجعه نمایید:

http://httpd.apache.org/docs/2.2/mod/core.html#traceenable

نحوه پیاده‌سازی:

در راستای پیاده سازی وضعیت توصیه شده موارد زیر را اجرا نمایید:

۱. محل فایل پیکربندی اصلی Apache را همانند httpd.conf تعیین نمایید.
۲. دستور TraceEnable را به پیکربندی سطح سرور با مقدار off اضافه نمایید.

پیکربندی سطح سرور بالاترین سطح پیکربندی به شمار می‌رود و مانند دایرکتوری‌های دیگری همچون <Directory> یا <Location> مرتبط نمی‌باشد.

```
TraceEnable off
```

SCSR-5-9: محدود نمودن نسخه پروتکل HTTP

شرح اجمالی:

از ماژول‌های mod-rewrite یا mod_security می‌توان برای عدم دسترسی به نسخه‌های پروتکل‌های HTTP قدیمی و نامعتبر استفاده کرد. نسخه ۱,۱ RFC از HTTP مورخ ژوئن ۱۹۹۹ می‌باشد، و از نسخه ۱,۲ توسط آپاچی پشتیبانی شده است. استفاده از این ماژول برای دسترسی به نسخه‌های قدیمی HTTP مانند ۱,۰ و قبل از آن غیر مجاز می‌باشد.

به منظور کسب جزئیات بیشتر به مستندات Apache مربوط به mod_rewrite مراجعه نمایید:

http://httpd.apache.org/docs/2.2/mod/mod_rewrite.html

نحوه پیاده‌سازی:

در راستای پیاده سازی وضعیت توصیه شده موارد زیر را اجرا نمایید:

۱. ماژول mod_rewrite را با انجام هرکدام از موارد زیر، برای Apache بارگذاری نمایید:
   ۱. با اضافه کردن گزینه‌ی --enable-rewrite به اسکریپت ./configure، هنگام نصب Apache، ماژول mod_rewrite نیز نصب می‌گردد.

```
./configure --enable-rewrite
```

   ۲. و یا با استفاده از دستور LoadModule در فایل پیکربندی httpd.conf، ماژول را به صورت dynamic، بارگذاری نمایید.

```
LoadModule rewrite_module modules/mod_rewrite.so
```

۲. به منظور فعال نمودن RewriteEngine، مقدار این دستور را به on در پیکربندی عمومی سرور تغییر دهید.

```
RewriteEngine on
```

۳. محل فایل پیکربندی اصلی Apache مانند httpd.conf را تعیین کرده و شرط Rewrite زیر را برای انطباق با HTTP/1.1 و قانون Rewriteرا برای رد کردن ورژن‌های دیگر پروتکل، به پیکربندی سطح سرور اضافه نمایید.

```
RewriteEngine On
RewriteCond %{THE_REQUEST} !HTTP/1\.1$
RewriteRule .* - [F]
```

۴. به طور پیش‌فرض، تنظیمات پیکربندی mod-rewrite از سرور اصلی توسط میزبان‌های مجازی به ارث برده نمی‌شوند. بنابراین، اضافه کردن دستورهای زیر به هر بخش برای به ارث بردن تنظیمات سرور اصلی نیز ضروری است.

```
RewriteEngine On
RewriteOptions Inherit
```

SCSR-5-10: محدود نمودن دسترسی به فایل‌های *.ht

شرح اجمالی:

دسترسی به هر فایلی که با ht. شروع شده و از دستور FilesMatch استفاده می‌کند را محدود نمایید.

نحوه پیاده‌سازی:

در راستای پیاده سازی وضعیت توصیه شده موارد زیر را اجرا نمایید:

۱. خطوط زیر را در پیکربندی آپاچی در سطح پیکرینی سرور اضافه و یا اصلاح نمایید.

```
<FilesMatch "^\.ht">
    Require all denied
</FilesMatch>
```

SCSR-5-11: محدود نمودن پسوند فایل‌ها

شرح اجمالی:

دسترسی به پسوند فایل‌های نامتناسبی که انتظار نمی‌رود بخش مشروعی از وب سایت باشند، را محدود نمایید به نحوی که نتوان با دستور FilesMatch از آنها استفاده نمود

نحوه پیاده‌سازی:

در راستای پیاده سازی وضعیت توصیه شده موارد زیر را اجرا نمایید:

۱. لیستی از پسوند فایل‌های موجود بر روی وب سرور ایجاد نمایید. دستور find/awk زیر ممکن است مفید باشد، اما احتمالا با توجه به دایرکتوری‌های وب‌روت مناسب برای وب سرور شما، به سفارشی‌سازی نیاز دارد. توجه داشته باشید که دستور find، هر فایل بدون نقطه (.) در نام فایل را نادیده می‌گیرد، زیرا انتظار می‌رود که آن‌ها محتوای مناسبی نباشند.

```
find */htdocs -type f -name '*.*' | awk -F. '{print $NF }' | sort -u
```

۲. لیست پسوندهای فایل موجود برای محتوای مناسب بر روی سرور وب، مرور کرده و آن‌هایی که نامناسب هستند را حذف کرده و هر پسوند فایل اضافه‌ای که انتظار می‌رود در آینده‌ای نزدیک به سرور وب افزوده شود را اضافه نمایید.

۳. به منظور رد نمودن دسترسی به همه فایل‌ها به صورت پیش‌فرض، دستور FilesMatch را اضافه نمایید.

```
# Block all files by default, unless specifically allowed.
<FilesMatch "^.*$">
    Require all denied
</FilesMatch>
```

۴. یک دستور FilesMatch دیگر اضافه کنید که اجازه‌ی دسترسی به پسوند فایل‌هایی که به طور خاص در فرآیند مرور مرحله‌ی ۲، مجوز داشته‌اند را می‌دهد. مثالی از دستور FilesMatch در زیر آمده است. پسوند فایل در عبارت منظم بایستی با لیست تأیید شما مطابقت داشته باشد، نه لزوما با عبارت زیر.

```
# Allow files with specifically approved file extensions
# Such as (css, htm; html; js; pdf; txt; xml; xsl; ...),
# images (gif; ico; jpeg; jpg; png; ...), multimedia
<FilesMatch "^.*\.(css|html?|js|pdf|txt|xml|xsl|gif|ico|jpe?g|png)$">
    Require all granted
</FilesMatch>
```

SCSR-5-12: فیلترینگ درخواست‌ها بر اساس آدرس IP

شرح اجمالی:

از ماژول mod_rewrite می‌توان جهت رد کردن تقاضا برای urlهای خاص استفاده کرد این ماژول این امکان را می‌دهد که به جای IP از Host Name استفاده گردد. عادی‌ترین دسترسی به وب سایت از طریق مرورگرها و نرم‌افزار خودکار با استفاده از یک نام میزبان می‌باشد، و در نتیجه شامل نام میزبان در هدر HTTP HOST خواهد بود.

به منظور کسب جزئیات بیشتر به مستندات مربوط به Apache 2.2 مراجعه نمایید:

http://httpd.apache.org/docs/2.2/mod/mod_rewrite.html

نحوه پیاده‌سازی:

در راستای پیاده‌سازی وضعیت توصیه شده موارد زیر را اجرا نمایید:

۱. ماژول mod_rewrite را با انجام هرکدام از موارد زیر، برای Apache بارگذاری کنید:
   ۱. با اضافه کردن گزینه‌ی enable-rewrite به اسکریپت ./configure، Apache با mod_rewrite بارگذاری شده به صورت static را در طول فرآیند ساخت، بسازید.

```
./configure --enable-rewrite
```

   ۲. یا با استفاده از دستور LoadModule در فایل پیکربندی httpd.conf، ماژول را به صورت dynamic، بارگذاری کنید.

```
LoadModule rewrite_module modules/mod_rewrite.so
```

۲. به منظور فعال نمودن RewriteEngine، مقدار این دستور را به on تغییر دهید.

```
RewriteEngine On
```

۳. محل فایل پیکربندی Apache مانند httpd.conf را تعیین کرده و شرط rewrite زیر را برای مطابقت با نام میزبان مورد انتظار از پیکربندی سطح سرور بالا، اضافه کنید.

```
RewriteCond %{HTTP_HOST} !^www\.example\.com [NC]
RewriteCond %{REQUEST_URI} !^/error [NC]
RewriteRule ^.(.*) - [L,F]
```

SCSR-5-13: محدود کردن دستور Listen

شرح اجمالی:

دستور Listen، شماره پورت‌هایی را که وب سرور آپاچی می‌خواهد برروی آن گوش دهد را مشخص می‌کند. همچنین IP هایی که تنها درخواست آنها پذیرفته خواهد شد، را نیز مشخص می‌کند. به جای نامحدود شدن شنود بر روی تمام آدرس‌های IP در دسترس برای سیستم، آدرس یا آدرس‌های IP خاص در نظر گرفته شده باید به طور صریح مشخص شوند. به طور خاص، یک دستور Listen که هیچ آدرس IP برای آن مشخص نشده، یا با یک آدرس IP از صفرها (0.0.0.0) نباید مورد استفاده قرار گیرد.

نحوه پیاده‌سازی:

در راستای پیاده سازی وضعیت توصیه شده موارد زیر را اجرا نمایید:

۱. هر دستور Listen در فایل پیکربندی Apache بدون هیچ آدرس IP مشخص، یا با همه‌ی آدرس‌های IP برابر صفر را مانند مثال زیر بیابید. به یاد داشته باشید که ممکن است آدرس‌های IPv4 و IPv6 هر دو در سیستم موجود باشند.

```
Listen 80
Listen 0.0.0.0:80
Listen [::ffff:0.0.0.0]:80
```

۲. دستورهای Listen در فایل پیکربندی Apache را برای داشتن آدرس‌های IP مشخص با توجه به استفاده در نظر گرفته شده، اصلاح کنید. ممکن است چندین دستور Listen برای هر آدرس IP و پورت، مشخص شود.

```
Listen 10.1.2.3:80
Listen 192.168.4.5:80
Listen [2001:db8::a00:20ff:fea7:ccea]:80
```

SCSR-5-14: محدود نمودن Optionهای فریم مرورگر

شرح اجمالی:

دستور Header به هدرهای پاسخ HTTP سرور اجازه اضافه، جایگزین یا ادغام را می‌دهد. ما از این دستور برای اضافه کردن یک هدر پاسخ HTTP سرور به منظور توجیه مرورگرها در خصوص اینکه تمام صفحات وب را از قرار گرفتن در چارچوب وب سایت‌های دیگر محدود کنند، استفاده خواهیم نمود.

نحوه پیاده‌سازی:

۱. دستور Header را برای هدر X-Frames-Options در تنظیمات آپاچی به وضعیت always اضافه و یا اصلاح نمایید و همانطور که به صورت زیر نمایش داده شده است یک عمل از append و یک مقدار از SAMEORIGIN یا DENY انتخاب گردد.

```
Header always append X-Frame-Options SAMEORIGIN
```

SCSR-6: عملیات ورود، نظارت و نگهداری

SCSR-6-1: پیکربندی لاگ خطا

شرح اجمالی:

از دستور LogLevel به منظور پیکربندی سطح شدت برای Error Log مورد استفاده قرار می‌گیرد. در حالی که دستور ErrorLog، نام فایل ممیزی خطا را پیکربندی می‌کند. مقادیر Log برای SysLog استاندارد شامل emerge (syslog)، alert، crit، error، warn، notice، info و debug می‌باشد. سطح توصیه شده notice می‌باشد، به طوری که تمام خطاها از سطح emerge تا سطح notice ثبت می‌شوند.

نحوه پیاده‌سازی:

در راستای پیاده سازی وضعیت توصیه شده موارد زیر را اجرا نمایید:

۱. مقدار LogLevel را در پیکربندی آپاچی اضافه نموده ویا تغییر دهید به گونه‌ای که مقدار notice و یا پایین‌تر داشته باشد. توجه نمایید که داشتن یک مقدار از info یا debug نیز می‌توان بکار برد در صورتی که نیاز به لاگ کامل‌تر وجود داشته باشد و فرآیندهای ذخیره‌سازی و نظارت، قادر به کنترل بار اضافی باشند. لذا مقدار پیشنهادی در این خصوص notice می‌باشد.

```
LogLevel notice
```

۲. در صورتی که دستور ErrorLog پیکربندی نشده، آن را اضافه نمایید. مسیر فایل ممکن است نسبی یا قطعی باشد، یا لاگ‌ها برای ارسال به یک سرور syslog، پیکربندی شده باشند.

```
ErrorLog "logs/error_log"
```

۳. در صورتی که افراد مسئول متفاوتی برای میزبان مجازی وب سایت در نظر گرفته شده است برای هر میزبان مجازی پیکربندی شده یک دستور ErrorLog مشابه اضافه نمایید. هر فرد یا سازمان مسئول دسترسی به لاگ‌های وب خود بوده و به مهارت‌ها/آموزش/ابزار برای نظارت بر لاگ‌ها نیاز می‌باشد.

SCSR-6-2: پیکربندی Syslog Facility برای دریافت خطاها

شرح اجمالی:

جهت ارسال لاگ‌ها به یک syslog، دستور ErrorLog پیکربندی شود، به طوری که لاگ‌ها بتوانند همراه با لاگ‌های سیستم پردازش و نظارت شوند.

نحوه پیاده‌سازی:

۱. دستور ErrorLog را در صورت عدم پیکربندی، اضافه نمایید. هر نوع از سطوح SysLog متناسب در Local1 مورد استفاده قرار می‌گیرد.

```
ErrorLog "syslog:local1"
```

۲. در صورت نیاز برای هر میزبان مجازی یک دستور ErrorLog مشابه اضافه نمایید.

SCSR-6-3: تنظیمات لاگ دسترسی

شرح اجمالی:

دستور LogFormat، فرمت و اطلاعات را برای گنجانده شدن در مدخل‌های لاگ دسترسی، تعریف می‌کند. دستور CustomLog فایل لاگ، syslog facility یا ابزار piped logging را مشخص می‌کند.

نحوه پیاده‌سازی:

در راستای پیاده سازی وضعیت توصیه شده موارد زیر را اجرا نمایید:

۱. در پیکربندی Apache برای استفاده از فرمت استاندارد و توصیه شده‌ی combined، دستورهای LogFormat را اضافه یا اصلاح کنید.

```
LogFormat "%h %l %u %t \"%r\" %>s %b \"%{Referer}i\" \"%{User-agent}i\""
combined
```

۲. به منظور استفاده از فرمت combined با یک فایل لاگ مناسب، syslog یا ابزار piped Logging، دستورهای CustomLog در پیکربندی Apache را اضافه یا اصلاح کنید.

```
CustomLog log/access_log combined
```

۳. برای هر میزبان مجازی پیکربندی شده یک دستور CustomLog مشابه اضافه نمایید، اگر افراد مسئول متفاوتی برای میزبان مجازی وب سایت در نظر گرفته شده است. هر فرد یا سازمان مسئول دسترسی به لاگ‌های وب خود بوده و به مهارت‌ها/آموزش/ابزار برای نظارت بر لاگ‌ها نیاز می‌باشد.

فرمت توکن‌های رشته اطلاعات زیر را فراهم می‌کنند:

%h: نام میزبان یا آدرس IP راه دور، اگر HostnameLookups، off باشد که مقدار پیش‌فرض است.
%l: نام لاگ/ شناسه‌ی راه دور.
%u: کاربر راه دور، اگر درخواست، تأیید شده باشد.
%t: زمان دریافت درخواست،
%r: اولین خط درخواست.
%> s: وضعیت نهایی.
%b: اندازه‌ی پاسخ بر اساس بایت.
%{ Referer} i: مقدار متغیر برای هدر Referer.
%{ User-agent} i: مقدار متغیر برای هدر User Agent.

SCSR-6-4: بازبینی و فضای لاگ

شرح اجمالی:

وجود فضای دیسک کافی بر روی پارتیشن به منظور نگهداری تمام فایل‌های لاگ تولید شده امر بسیار مهمی می‌باشد، بازه چرخش لاگ‌ها، اگر از فضای ذخیره‌سازی مرکزی برای ذخیره سازی لاگ‌ها مورد استفاده قرار نمی‌گیرد، برای حفظ حداقل ۳ ماه یا ۱۳ هفته پیکربندی شود.

نحوه پیاده‌سازی:

برای پیاده‌سازی حالت توصیه شده، گزینه a) اگر از Linux logrotate استفاده می‌کنید، گزینه b) اگر از یک ابزار piped Logging استفاده می‌کنید، را به کار بگیرید:

۱. File Logging با Logrotate:

۱. پیکربندی بازبینی لاگ وب را جهت مطابقت با فایل‌های لاگ پیکربندی شده‌ی خود در etc/logrotate.d/httpd/، همانند زیر، اضافه یا اصلاح نمایید.

```
/var/log/httpd/*log {
    missingok
    notifempty
    sharedscripts
    postrotate
    /bin/kill -HUP `cat /var/run/httpd.pid 2>/dev/null` 2> /dev/null ||
true
   endscript
}
```

۲. دوره‌ی بازنگری و تعداد لاگ‌ها را طوری اصلاح نمایید که حداقل ۱۳ هفته یا ۳ ماه لاگ‌ها حفظ شوند. این کار ممکن است به صورت پیش‌فرض برای همه‌ی لاگ‌ها در /etc/logrotate.conf یا در پیکربندی بازبینی لاگ مخصوص وب در etc/logrotate.d/httpd/، مانند زیر انجام شود.

```
# rotate log files weekly
weekly

# keep 1 years of backlogs
rotate 52
```

۳. برای هر میزبان مجازی پیکربندی‌شده با فایل‌های لاگ خود که آن فایل‌های لاگ در یک بازبینی لاگ مشابه نیز جای دارند، اطمینان حاصل نمایید.

۲. Piped Logging:

۱. فواصل بازبینی لاگ و نام‌های فایل لاگ را به یک فاصله‌ی مناسب مانند روزانه، پیکربندی نمایید.

```
CustomLog "|bin/rotatelogs -l /var/logs/logfile.%Y.%m.%d 86400" combined
```

۲. اطمینان از اینکه به منظور نام‌گذاری فایل لاگ و اسکریپت‌های بازبینی، برای باقی ماندن حداقل ۳ ماه یا ۱۳ هفته، فراهم شده‌اند.
۳. برای هر میزبان مجازی پیکربندی شده با فایل‌های لاگ خود که آن فایل‌های لاگ در یک بازبینی مشابه نیز جای دارند، اطمینان حاصل نمایید.

SCSR-6-5: اعمال وصله‌های قابل اجرا

شرح اجمالی:

آخرین وصله‌های آپاچی در دسترس قرار گرفته، در بازه یک ماهه را اعمال نمایید.

نحوه پیاده‌سازی:

آخرین نسخه‌ی در دسترس از Apache را با توجه به یکی از موارد زیر به روزرسانی نمایید:

۱. هنگام ساخت از منبع:
    ۱. نکات منتشر شده و اطلاعات امنیتی مرتبط با آن را بخوانید.
    ۲. آخرین منبع و هر ماژول وابسته مانند mod-security را دانلود نمایید.
    ۳. نرم‌افزار جدید Apache را با توجه به روند ساخت خود با گزینه‌های پیکربندی مشابه ایجاد نمایید.
    ۴. نرم‌افزار جدید را با توجه به فرآیند تست سازمانی خود، نصب و آزمایش نمایید.
    ۵. با توجه به روند توسعه‌ی سازمانی خود، نرم‌افزار را به تولید انتقال بدهید.
۲. هنگام استفاده از بسته‌های پلت‌فرم:
    ۱. نکات منتشر شده و اطلاعات امنیتی مرتبط با آن را بخوانید.
    ۲. آخرین بسته‌ی موجود Apache و هر نرم‌افزار وابسته را دانلود نمایید.
    ۳. نرم‌افزار جدید را با توجه به فرآیند تست سازمانی خود، نصب و آزمایش نمایید.
    ۴. با توجه به روند توسعه‌ی سازمانی خود، نرم‌افزار را به تولید انتقال دهید.

SCSR-6-6: راه‌اندازی و فعال‌سازی ModSecurity

شرح اجمالی:

ModSecurity یک فایروال جهت برنامه‌های تحت وب (WAF) و منبع باز برای نظارت، ثبت، و کنترل دسترسی بلادرنگ برای برنامه تحت وب است. این فایروال مجموعه قوانین قابل تنظیم قدرتمند را که ممکن است برای شناسایی و جلوگیری از حملات متداول برنامه تحت وب مورد استفاده قرار گیرند، فعال می‌کند اما قوانین پیش فرض آن دارای قدرت مناسبی نیستند. نصب و راه اندازی ModSecurity بدون یک مجموعه قوانین، امنیت بیشتری را برای برنامه تحت وب حفاظت شده فراهم نمی‌کند. برای جزئیات بیشتر در مورد یک مجموعه قوانین توصیه شده، به توصیه معیار "نصب و فعال سازی مجموعه قوانین اصلی OWASP ModSecurity" مراجعه کنید.

توجه: همانند سیستم‌های امنیت نرم‌افزار/ فایروال امنیتی، Mod_Security نیز نیازمند تعهد قابل توجه کارکنان برای تنظیم اولیه قوانین و کنترل هشدارهاست. در برخی موارد، این امر ممکن است نیازمند زمان کار اضافه تری از جانب توسعه دهندگان/ نگهداران برنامه، برای اصلاح آن بر اساس تجزیه و تحلیل نتایج لاگ‌ها می‌باشد. بعد از راه اندازی، تعهد دائمی کارکنان، به ویژه بعد از ارتقاء/پچ‌ها، برای تنظیم مداوم مورد نیاز است. بدون این تعهد به تنظیم و نظارت، نصب Mod_Security ممکن است مؤثر نباشد و یک امنیت کاذب ارائه دهد.

نحوه پیاده‌سازی:

۱. در صورت عدم نصب ماژول ModSecurity در modules/mod_security2.so، نصب نمایید. این ماژول، ممکن است از طریق نصب بسته‌ی OS (مانند apt-get یا yum) یا ساخته شده از فایل‌های منبع، نصب شود. به منظور کسب جزئیات بیشتر به آدرس https://www.modsecurity.org/download.html مراجعه نمایید.
۲. در صورت عدم وجود دستور LoadModule در پیکربندی Apache، آن را اضافه یا اصلاح نمایید. دستور LoadModule معمولا در فایلی به نام mod_security.conf که در پیکربندی Apache قرار دارد، جای گرفته است.

```
LoadModule security2_module modules/mod_security2.so
```

SCSR-6-7: راه‌اندازی و فعال‌سازی مجموعه قوانین اصلی OWASP ModSecurity

شرح اجمالی:

مجموعه قوانین اصلی OWASP ModSecurity (CRS)، مجموعه‌ای از قوانین دفاعی برنامه تحت وب منبع باز برای فایروال برنامه تحت وب (WAF) است. OWASP ModSecurity CRS حفاظت‌های اولیه را در دسته‌های حمله/تهدید زیر فراهم می‌کند:

- حفاظت HTTP – تشخیص نقض‌های پروتکل HTTP و یک سیاست استفاده تعریف شده به صورت محلی.
- متغیرهای لیست سیاه بلادرنگ – استفاده از اعتبار IP شخص ثالث
- انکار حفاظت‌های خدمات HTTP – دفاع در برابر جاری شدن HTTP و حملات HTTP DoS آهسته
- حفاظت از حملات متداول وب – تشخیص حملات علیه برنامه‌های وب
- خودکارسازی تشخیص – تشخیص بات‌ها، crawlers، اسکنرها و سایر فعالیت‌های مخرب سطحی
- ادغام با اسکن AV برای آپلود فایل – تشخیص فایل‌های مخرب آپلود شده از طریق برنامه تحت وب
- پیگیری اطلاعات حساس – پیگیری استفاده از کارت اعتباری و مسدود کردن نشتی‌ها
- حفاظت در برابر تروجان – تشخیص دسترسی به تروجان‌ها
- شناسایی ناهنجاری‌های برنامه – هشدار در مورد پیکربندی‌های اشتباه برنامه
- تشخیص خطا و پنهان کردن – پنهان کردن پیام‌های خطای ارسال شده توسط سرور

توجه: همانند سیستم‌های امنیت نرم‌افزار/ فایروال امنیتی، Mod_Security نیز نیازمند تعهد قابل توجه کارکنان برای تنظیم اولیه قوانین و کنترل هشدارهاست. در برخی موارد، این امر ممکن است نیازمند زمان کار اضافه تری از جانب توسعه دهندگان/ نگهداران برنامه، برای اصلاح آن بر اساس تجزیه و تحلیل نتایج لاگ‌ها می‌باشد. بعد از راه اندازی، تعهد دائمی کارکنان، به ویژه بعد از ارتقاء/پچ‌ها، برای تنظیم مداوم مورد نیاز است. بدون این تعهد به تنظیم و نظارت، نصب Mod_Security ممکن است مؤثر نباشد و یک امنیت کاذب ارائه دهد.

نحوه پیاده‌سازی:

OWASP ModSecurity Core Rule Set را نصب، پیکربندی و تست کنید:

۱. OWASP ModSecurity CRS را از صفحه‌ی پروژه

https://www.owasp.org/index.php/Category:OWASP_ModSecurity_Core_Rule_Set_Project دانلود کنید.

۲. دستورالعمل‌های فایل INSTALL را دنبال کنید.
۳. فایل modsecurity_crs_10_setup.conf مورد نیاز است و قوانین دایرکتوری base_rules، پایه‌ی مفیدی برای بیشتر کاربردها هستند.
۴. برنامه‌ی کاربردی را برای عملکرد درست پس از نصب CRS، تست نمایید. لاگ‌های خطای وب سرور و فایل the modsec_audit.log را برای درخواست‌های مسدود شده‌ی ناشی از false positiveها، بررسی نمایید.
۵. همچنین، توصیه می‌شود که پاسخ برنامه‌ی کاربردی به ترافیک مخرب، مانند یک اسکنر خودکار برنامه‌ی تحت وب، برای حصول اطمینان از فعال بودن قوانین، تست شود. لاگ‌های خطای وب سرور و فایل‌های the modsec_audit.log بایستی لاگ حملات و کدهای پاسخ سرورها را نشان دهند.

SCSR-7: استفاده از SSL/TSL

SCSR-7-1: نصب و راه‌اندازی mod_ssl و یا mod_nss

شرح اجمالی:

لایه سوکت‌های امن (SSL) توسط Netscape توسعه داده شده و تبدیل به یک استاندارد فراگیر شده است، و به عنوان بخشی از فرآیند به امنیت لایه انتقال (TLS) تغییر نام داده شد. TLS برای حفاظت از ارتباطات، مهم می‌باشد و احراز هویت سرور و حتی کلاینت را فراهم می‌کند. با این حال برخلاف ادعاهای تولیدکننده، پیاده‌سازی SSL به طور مستقیم وب سرور شما را امن‌تر نمی‌کند! چرا که SSL برای رمزگذاری ترافیک مورد استفاده قرار می‌گیرد و در نتیجه محرمانگی اطلاعات خصوصی و اعتبارات کاربر را تأمین می‌کند. با این حال به خاطر داشته باشید تنها به خاطر این که شما مسیر داده‌ها را رمزگذاری کرده‌اید به این معنا نمی‌باشد که اطلاعات ارائه شده توسط کلاینت امن است، زیرا که این اطلاعات بر روی سرور قرار دارد. همچنین از SSL نمی‌توان انتظار حفاظت از وب سرور را داشت، زیرا مهاجمان به راحتی وب سرورهای دارای SSL را هدف قرار داده و حمله در کانال رمزگذاری شده پنهان خواهد شد. ماژول mos_ssl استاندارد و پر استفاده‌ترین ماژولی است که SSL/TLS را برای آپاچی پیاده‌سازی می‌کند. یک ماژول جدیدتر بر روی سیستم‌های Red Hat می‌تواند یک تعریف یا جایگزینی برای mod_ssl باشد، و قابلیت‌های مشابه به علاوه خدمات امنیتی بیشتری را فراهم نماید. mod_nss یک ماژول

آپاچی جهت پیاده‌سازی نرم افزار خدمات امنیت شبکه (NSS) از Mozilla است و طیف گسترده‌ای از توابع رمزنگاری به علاوه TLS را پیاده‌سازی می‌کند.

نحوه پیاده‌سازی:

در راستای پیاده سازی وضعیت توصیه شده یکی از موارد زیر را اجرا نمایید:

۱. برای نصب Apache با استفاده از توزیع‌های سورس، از گزینه‌ی =with-ssl-- برای مشخص کردن مسیر openssl و از گزینه‌ی پیکربندی enable-ssl-- برای اضافه کردن ماژول‌های SSL به ساخت، استفاده نمایید. در صورت ناسازگاری با نسخه پلت‌فرم پیکربندی گزینه with-included-apr-- ضروری می‌باشد.
به منظور کسب جزئیات بیشتر به مستندات مربوط به Apache
مراجعه نمایید: http://httpd.apache.org/docs/2.2/install.html

```
# ./configure --with-included-apr --with-ssl=$OPENSSL_DIR --enable-ssl
```

۲. برای نصب با استفاده از بسته‌های OS، معمولا تنها نکته مهم حصول اطمینان از نصب بسته‌ی mod_ssl می‌باشد. همچنین بسته‌ی mod_nss نیز ممکن است نیاز به نصب داشته باشد. دستورات yum زیر برای Red Hat Linux مناسب هستند.

```
# yum install mod_ssl
```

SCSR-7-2: نصب یک گواهی مورد اعتماد معتبر

شرح اجمالی:

گواهی SSL پیش‌فرض، self-signe می‌باشد و قابل اعتماد نیست. لذا می‌بایست یک گواهی قابل اعتماد معتبر نصب نمایید. گواهی برای معتبر بودن می‌بایست:

- توسط یک مرجع گواهی مورد اعتماد امضا شده باشد.
- منقضی نشده باشد، و
- یک نام مشترک منطبق بر نام میزبان وب سرور داشته باشد ، مانند www.example.com.

نحوه پیاده‌سازی:

در راستای پیاده سازی وضعیت توصیه شده موارد زیر را اجرا نمایید:

۱. درباره‌ی نام میزبان استفاده شده در گواهی، تصمیم بگیرید. مهم این است که به یاد داشته باشید مرورگر نام میزبان در URL را با نام موجود در گواهی، مقایسه می‌کند. بنابراین مطابقت با نام صحیح میزبان از اهمیت بالایی برخوردار است. به خصوص زمانی که، نام میزبان www.example.com با نام example.com و ssl.example.com یکی نباشد.

۲. یک کلید خصوصی با استفاده از openssl تولید کنید. اگرچه طول کلید اعتبارسنجی رایج در گذشته ۱۰۲۴ بوده، اما برای اعتبارسنجی قدرتمندتر، طول کلید ۲۰۴۸ توصیه می‌شود. کلید بایستی محرمانه باقی بماند و به طور پیش‌فرض با یک عبارت عبور، رمزگذاری شود. در همین راستا مراحل زیر را طی نمایید:

به منظور کسب جزئیات بیشتر به مستندات مربوط به Apache یا OpenSSL مراجعه نمایید:
http://httpd.apache.org/docs/2.2/ssl/ssl_faq.html#realcert
http://www.openssl.org/docs/HOWTO/certificates.txt

```
# cd /etc/pki/tls/certs
# umask 077
# openssl genrsa -aes128 2048 > example.com.key

Generating RSA private key, 2048 bit long modulus
. . . +++
. . . . . +++
e is 65537 (0x10001)
Enter pass phrase:
Verifying - Enter pass phrase:
```

۳. درخواست امضای گواهی (CSR) را به منظور امضا توسط یک مرجع گواهی، تولید نمایید. مهم این است که یک نام مشترک، دقیقا نام میزبان وب را تشکیل دهد.

```
# openssl req -utf8 -new -key www.example.com.key -out www.example.com.csr

Enter pass phrase for example.com.key:
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
```

```
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) [GB]:US
State or Province Name (full name) [Berkshire]:New York
Locality Name (eg, city) [Newbury]:Lima
Organization Name (eg, company) [My Company Ltd]:Durkee Consulting
Organizational Unit Name (eg, section) []:
Common Name (eg, your name or your server's hostname) []:www.example.com
Email Address []:ralph@example.com

Please enter the following 'extra' attributes
to be sent with your certificate request
A challenge password []:
An optional company name []:

# mv www.example.com.key /etc/pki/tls/private/
```

۴. CSR را به یک مرجع امضای گواهی برای امضا بفرستید و دستورالعمل‌های آن‌ها را برای ارائه و اعتبارسنجی، دنبال نمایید. CSR و گواهی امضا شده‌ی نهایی، تنها متن رمزگذاری شده‌ای هستند که یکپارچگی آن‌ها، نه محرمانگی‌شان، بایستی محافظت شود. با این حال این گواهی برای هر اتصال SSL فرستاده خواهد شد.

۵. گواهی امضا شده‌ی نتیجه ممکن است www.example.com.crt نامیده شده و در /etc/pki/tls/certs/ جای بگیرد که برای همه قابل خواندن است (mode 0444) . توجه کنید که مرجع گواهی، نیازی به کلید خصوصی ندارد (example.com.key) و این فایل بایستی به دقت، محافظت شود. با یک کپی رمزگشایی شده از کلید خصوصی، رمزگشایی همه‌ی تعاملات با سرور، امکان‌پذیر است.

۶. عبارت عبور استفاده شده برای رمزگذاری کلید خصوصی را فراموش نکنید. این عبارت، هر زمان که سرور در حالت https آغاز به کار می‌کند لازم است. اگر جلوگیری از نیاز به تایپ عبارت ورود در زمان شروع httpd ضروری است، کلید خصوصی ممکن است در یک متن ساده، ذخیره شود. ذخیره‌ی کلید خصوصی در یک متن ساده، باعث راحتی می‌شود، در حالی که ریسک افشای کلید را افزایش می‌دهد، اما اگر خطرات آن به خوبی مدیریت شود ممکن است برای راه‌اندازی مجدد مناسب باشد. مطمئن شوید که فایل کلید تنها توسط ریشه، قابل خواندن است. برای رمزگشایی کلید خصوصی و ذخیره‌ی آن در فایل متنی، ممکن است از دستور openssl زیر استفاده شود. شما می‌توانید با استفاده از هدرهای کلید خصوصی، رمز بودن یا متن ساده بودن آن را تشخیص دهید.

```
# cd /etc/pki/tls/private/
# umask 077
# openssl rsa -in www.example.com.key -out www.example.com.key.clear
```

۷. محل فایل پیکربندی Apache برای mod-ssl را تعیین نموده و دستورهای SSLCertificateFile و SSLCertificateKeyFile را برای داشتن مسیر صحیح برای کلید خصوصی و فایل‌های گواهی امضا شده، اضافه یا اصلاح نمایید. اگر به یک کلید متن ساده، ارجاع داده شود به عبارت عبور نیازی نیست. درضمن شما از گواهی CA که به جای بسته‌ی نرم‌افزاری CA، گواهی شما را امضا کرده، برای سرعت بخشیدن به اتصال SSL اولیه می‌توانید استفاده نمایید.

```
SSLCertificateFile /etc/pki/tls/certs/example.com.crt
SSLCertificateKeyFile /etc/pki/tls/private/example.com.key

# Default CA file, can be replaced with your CA's certificate.
SSLCACertificateFile /etc/pki/tls/certs/ca-bundle.crt
```

۸. درنهایت سرویس httpd را شروع یا راه‌اندازی مجدد کرده و عملکرد صحیح آن را با مرورگر مورد علاقه‌ی خود، اعتبارسنجی نمایید.

SCSR7-3: حفاظت از کلید خصوصی سرور

شرح اجمالی:

محافظت از کلید خصوصی سرور بسیار مهم می‌باشد. به طور پیش‌فرض کلید خصوصی سرور به عنوان یک سیستم حفاظت از سرور، رمزنگاری شده است. با این حال، رمزنگاری آن به این معنا می‌باشد که هر بار سرور راه اندازی می‌شود به کلمه عبور نیاز است، و در حال حاضر محافظت از کلمه عبور نیز ضروری و مهم می‌باشد. کلمه عبور ممکن است در برخی مواقع به صورت دستی تنظیم شده، یا توسط یک برنامه که به صورت خودکار تولید شود. به منظور کسب جزئیات بیشتر به لینک زیر مراجعه نمایید:

http://httpd.apache.org/docs/2.2/mod/mod_ssl.html#sslpassphrasedialog

به طور خلاصه، Optionهای موجود عبارتند از:

۱. استفاده از SSLPassPhraseDialog builtin، - نیازمند یک کلمه عبور برای ورود به صورت دستی
۲. استفاده از SSLPassPhraseDialog |/path/to/program برای ارائه کلمه عبور

۳. استفاده از SSLPassPhraseDialog exec:/path/to/program برای ارائه کلمه عبور
۴. ذخیره کلید خصوصی به صورت واضح به طوری که یک کلمه عبور مورد نیاز نباشد.

هر یک از گزینه‌های ۱ الی ۴ بالا تا زمانی که کلید و کلمه عبور به شرح زیر محافظت شوند، قابل قبول‌اند. گزینه ۱ مزیت امنیت بیشتری نسبت به عدم ذخیره سازی کلمه عبور دارد، اما این به طور کلی برای اکثر وب سرورهای تولیدی قابل قبول نیست، چون مستلزم این است که وب سرور به صورت دستی آغاز شود. در مورد گزینه‌های ۲ و ۳ در صورتی که برنامه‌هایی که آن‌ها را ارائه می‌دهند امن باشند می‌توانند امنیت بیشتری ارائه دهند. گزینه ۴ از همه ساده‌تر است، و به طور گسترده مورد استفاده قرار گرفته و تا زمانی که کلید خصوصی به طور مناسب محافظت شود، قابل قبول می‌باشد.

نحوه پیاده‌سازی:

در راستای پیاده سازی وضعیت توصیه شده موارد زیر را اجرا نمایید:

۱. می‌بایست کلیه کلیدهای خصوصی جدا از کلیدهای عمومی ذخیره گردند. دستورهای SSLCertificateFile را در فایل‌های پیکربندی آپاچی بیابید. برای هر یک از دستورهای SSLCertificateFile که دارای دستور SSLCertificateKeyFile متناظری نباشد، کلید را به فایلی مجزا از گواهی انتقال دهید، و دستور SSLCertificateKeyFile را برای فایل کلید اضافه نمایید.
۲. برای هر یک از دستورهای SSLCertificateKeyFile، مالکیت و مجوز بر روی سرور کلید خصوصی را به root:root با مجوز 0400 تغییر دهید.

SCSR-7-4: غیرفعال کردن پروتکل‌های SSL ضعیف

شرح اجمالی:

دستور SSLProtocol، پروتکل‌های SSL و TLS مجاز را مشخص می‌کند. می‌بایست هر دو پروتکل SSLv2 و SSLv3 در این دستور غیرفعال شوند، زیرا تاریخ گذشته بوده و نسبت به افشای اطلاعات آسیب پذیر می‌باشند. بدین منظور تنها می‌بایست پروتکل‌های TLS فعال شوند.

نحوه پیاده‌سازی:

در راستای پیاده سازی وضعیت توصیه شده موارد زیر را اجرا نمایید:

دستور SSLProtocol را در فایل‌های پیکربندی Apache جستجو نمایید؛ در صورتی که دستور موجود نبود، آن را اضافه نموده یا مقدار آن را برای مطابقت با یکی از مقادیر زیر، تغییر دهید. زمانی که غیرفعال کردن پروتکل TLSv1.0 قابل قبول باشد تنظیمات "TLS1.2 TLSv1.1" به عنوان اولین تنظیمات ترجیح داده می‌شوند.

```
SSLProtocol TLSv1.1 TLSv1.2
```

```
SSLProtocol TLSv1
```

SCSR-7-5: محدود کردن رمزهای ضعیف SSL

شرح اجمالی:

رمزنگاری SSL ضعیف را با استفاده از دستورهای SSLCipherSuite و SSLHonorCipherOrder غیرفعال نمایید. رمزنگاری‌های مجاز در مذاکره با کلاینت توسط دستور SSLCipherSuite مشخص می‌شوند. در حالی که SSLHonorCipherOrder باعث می‌شود سرور در خصوص انتخاب استفاده از چه نوع رمزنگاری به جای کلاینت تصمیم‌گیری کند.

نحوه پیاده‌سازی:

در راستای پیاده سازی وضعیت توصیه شده موارد زیر را اجرا نمایید:

خط زیر را در پیکربندی سطح سرور Apache و هر میزبان مجازی که SSL آن فعال شده است، اضافه یا اصلاح نمایید.

```
SSLHonorCipherOrder On
SSLCipherSuite ALL:!EXP:!NULL:!ADH:!LOW:!SSLv2:!MD5:!RC4
```

انطباق با استاندارد FIPS: مشخصات رمزنگاری بالا ممکن است برای سروری مورد استفاده قرار بگیرد که تحت شرایط انطباق با FIPS140-2 شکست بخورد، SP800-52 دستورالعمل‌هایی برای رمزنگاری TLS را فراهم می‌کند، زیرا سبب حذف استفاده از رمزنگاری RC4 و Hash نمودن MD5 می‌شود که به نظر نمی‌رسد با FIPS سازگار باشد.

غیرفعال‌سازی رمزهای SSLv3: همانطور که توصیه شده است اگر پروتکل SSLv3 غیرفعال شده باشد، پس رمزنگاری مرتبط با SSLv3 نیز مورد استفاده قرار نخواهد گرفت، و می‌توان آن را از مجموعه مشخصات رمزنگاری حذف نمود.

```
SSLCipherSuite ALL:!EXP:!NULL:!ADH:!LOW:!SSLv2:!SSLv3:!MD5:!RC4
```

SCSR-7-6: محدود نمودن SSL ناامن Renegotiation

شرح اجمالی:

یک نوع حمله man-in-the-middle یا مردی در میان، در SSLv3 و TLSv1 در نوامبر ۲۰۰۹ کشف شد (CVE-2009-3555). http://cve.mitre.org/cgi-bin/cvename.cgi?name=CAN-2009-3555 http://www.phonefactor.com/sslgap/ssl-tls-authentication-patches

ابتدا، یک راه حل و سپس یک اصلاح به عنوان یک استاندارد اینترنت به صورت RFC 574 در فوریه ۲۰۱۰ تصویب شد. راه حلی که Renegotiation را حذف می‌کند، از OpenSSL نسخه 0.9.8l و نسخه‌های جدیدتر قابل دسترس است. برای جزئیات بیشتر به لینک http://www.openssl.org/news/secadv_20091111.txt مراجعه نمایید.

دستور SSLInsecureRenegotiation در Apache 2.2.15 برای وب سرورهای مرتبط با OpenSSL نسخه 0.9.8m و بالاتر اضافه گردیده است. تا اجازه Renegotiation ناامن جهت ایجاد یک ارتباط سازگار با Clientهایی با نسخه SSL قدیمی را دهد. هنگامی که یک ارتباط سازگار ایجاد می‌گردد، فعال نمودن دستور SSLInsecureRenegotiation سرور را همچنین برای حملاتی مانند man-in-the-middle (CVE-2009-3555) آسیب پذیر می‌نماید. بنابراین، دستور SSLInsecureRenegotiation نباید فعال گردد.

نحوه پیاده‌سازی:

در راستای پیاده سازی وضعیت توصیه شده موارد زیر را اجرا نمایید:

۱. دستور SSLInsecureRenegotiation را در فایل‌های پیکربندی Apache جستجو نمایید؛ در صورت وجود دستور مذکور، مقدار آن را به off اصلاح نمایید. اما در صورت عدم وجود نیاز به انجام عمل خاصی نمی‌باشد.

```
SSLInsecureRenegotiation off
```

SCSR-7-7: اطمینان از فعال نبودن فشرده سازی SSL

شرح اجمالی:

دستور SSLCompression فشرده‌سازی SSL هنگام سرویس‌دهی به محتوای HTTPS توسط آپاچی را کنترل می‌کند که آیا مورد استفاده قرار گرفته است یا خیر. توصیه می‌شود که دستور SSLCompression بر روی Off تنظیم شود.

نحوه پیاده‌سازی:

در راستای پیاده سازی وضعیت توصیه شده موارد زیر را اجرا نمایید:

۱. با استفاده از دستور httpd –v از نسخه 2.2.24 و یا بالاتر Apache اطمینان حاصل نمایید.
۲. دستور SSLCompression را در فایل‌های پیکربندی Apache جستجو نمایید.
۳. در صورت وجود دستور، آن را به مقدار off تغییر داده و بروزرسانی نمایید.

SCSR-7-8: غیر فعال کردن پروتکل TSLv1.0

شرح اجمالی:

پروتکل TLSv1.0 در صورت امکان می‌بایست از طریق دستور SSLProtocol غیر فعال شود، همانطور که بیان شده است این پروتکل نسبت به افشای اطلاعات آسیب پذیر می‌باشد.

نحوه پیاده‌سازی:

در راستای پیاده سازی وضعیت توصیه شده موارد زیر را اجرا نمایید:

۱. دستور SSLProtocol را در فایل‌های پیکربندی Apache جستجو نمایید و در صورت عدم وجود آن را اضافه نمایید و یا مقدار آن را به TLSv1.2 TLSv1.1 تغییر دهید.

SCSR-7-9: فعال کردن OCSP Stapling

شرح اجمالی:

OCSP (Online Certificate Status Protocol) وضعیت ابطال کنونی یک گواهی X.509 را ارائه کرده و اجازه می‌دهد یک مرجع گواهی، اعتبار یک گواهی امضا شده را قبل از تاریخ انقضای آن لغو کند. URI برای سرور

OCSP در گواهی گنجانده شده و توسط مرورگر تأیید می‌شود. دستور SSLUseStapling همراه با دستور SSLStaplingCache برای فعال کردن OCSP Stapling توسط وب سرور توصیه می‌شوند. اگر کلاینت OCSP stapling را درخواست نماید، در آن صورت وب سرور شامل پاسخ سرور OCSP همراه با گواهی X.509 وب سرور باشد.

نحوه پیاده‌سازی:

در راستای پیاده سازی وضعیت توصیه شده موارد زیر را اجرا نمایید:

در سطح پیکربندی آپاچی سرور و هر میزبان مجازی که SSL در آن فعال می‌باشد، SSLUseStapling را به on تغییر دهید. همچنین می‌بایست SSLStaplingCache به یکی از Chacheهایی که مشابه به مثال زیر آمده تنظیم گردد.

```
SSLUseStapling On
SSLStaplingCache "shmcb:logs/ssl_staple_cache(512000)"
- or -
SSLStaplingCache "dbm:logs/ssl_staple_cache.db"
- or -
SSLStaplingCache dc:UNIX:logs/ssl_staple_socket
```

SCSR-7-10: فعال کردن HTTP Strict Transport Security

شرح اجمالی:

امنیت انتقال اسکریپت HTTP (HSTS) یک مکانیزم سیاست امنیتی وب سرور انتخابی است که توسط یک هدر سرور HTTP مشخص شده است. هدر HSTS به یک سرور اجازه می‌دهد تا اعلام کند که تنها ارتباطات HTTPS باید به جای ارتباطات HTTP مورد استفاده قرار گیرند.

نحوه پیاده‌سازی:

در راستای پیاده سازی وضعیت توصیه شده موارد زیر را اجرا نمایید:

دستور Header را همانطور که به صورت زیر نشان داده شده است در سطح سرور آپاچی و هر میزبان مجازی که دارای SSL می‌باشد، پیکربندی نمایید. ممکن است Flagهای includedSubDomain و preload در هدر گنجانده شوند در صورتی که لازم و ضروری نمی‌باشد.

```
Header always set Strict-Transport-Security "max-age=600"; includedSubDomain; preload
```

یا

```
Header always set Strict-Transport-Security "max-age=600"
```

SCSR-8: نشت اطلاعات

SCSR-8-1: تنظیم ServerToken به ‘Prod’

شرح اجمالی:

با تنظیم مقدار دستور ServerTokens بر روی Prod یا ProductOnly، آپاچی را برای ارائه حداقل اطلاعات پیکربندی نمایید. در اینصورت، به جای ارائه جزئیات در مورد ماژول‌ها و نسخه‌های نصب شده "Apache"، تنها اطلاعات نسخه، در هدر پاسخ HTTP قابل ارائه خواهد بود.

نحوه پیاده‌سازی:

در راستای پیاده سازی وضعیت توصیه شده موارد زیر را اجرا نمایید:

مقدار دستور ServerTokens را در پیکربندی آپاچی به Prod و یا ProductOnly همانند بند زیر تغییر دهید.

```
ServerTokens Prod
```

SCSR-8-2: تنظیم ServerSignature بر روی Off

شرح اجمالی:

امضاهای سرور، که یک خط امضا به صورت یک پانویس انتهایی در پایین اسناد توسط سرور ایجاد شده، (مانند صفحات خطا)، را غیر فعال نمایید.

نحوه پیاده‌سازی:

در راستای پیاده سازی وضعیت توصیه شده موارد زیر را اجرا نمایید:

مقدار دستور ServerSignature را در پیکربندی آپاچی به off همانند بند زیر تغییر دهید.

```
ServerSignature Off
```

SCSR-8-3: نشت اطلاعات از طریق محتوای پیش‌فرض Apache

شرح اجمالی:

در توصیه‌های قبلی، ما محتوای پیش‌فرض مانند دفترچه‌های راهنمای آپاچی و برنامه‌های CGI پیش‌فرض را حذف کردیم. با این حال، اگر می‌خواهید نشت اطلاعات مربوط به وب سرور را بیشتر محدود کنید، مهم است که محتوای پیش‌فرض مانند آیکون‌ها بر روی وب سایت باقی نماند.

نحوه پیاده‌سازی:

در راستای پیاده سازی وضعیت توصیه شده موارد زیر را اجرا نمایید:

۱. به طور پیش‌فرض در توزیع سورس، محل پیکربندی‌های auto-index و icon را در فایل extra/httpd-autoindex.conf در نظر می‌گیرد، بنابراین می‌تواند با کامنت نمودن خطوط در فایل httpd.conffile ، مانند ذیل اقدام به غیرفعال نمودن آن‌ها نمایید.

```
# Fancy directory listings
#Include conf/extra/httpd-autoindex.conf
```

۲. در روش دیگر، می‌توان دستور alias و پیکربندی کنترل دسترسی دایرکتوری را همانند زیر به کامنت تبدیل نمود:

```
# We include the /icons/ alias for FancyIndexed directory listings. If
# you do not use FancyIndexing, you may comment this out.
#
#Alias /icons/ "/var/www/icons/"

#<Directory "/var/www/icons">
#       Options Indexes MultiViews FollowSymLinks
#       AllowOverride None
#       Order allow,deny
#      Allow from all
#</Directory>
```

SCSR-9: انکار سرویس Mitigation

SCSR-9-1: تنظیم Timeout به ۱۰ یا کمتر

شرح اجمالی:

دستور TimeOut حداکثر زمان را بر اساس ثانیه که سرویس دهنده وب آپاچی باید برای یک فراخوانی ورودی /خروجی برای کامل شدن، منتظر بماند را تعیین می‌کند. لذا توصیه می‌گردد که مقدار TimeOut را بر روی 10 یا کمتر تنظیم نمایید.

نحوه پیاده‌سازی:

در راستای پیاده سازی وضعیت توصیه شده موارد زیر را اجرا نمایید:

مقدار دستور Timeout را در پیکربندی آپاچی به 10 ثانیه و یا کمتر تغییر دهید.

```
Timeout 10
```

SCSR-9-2: تنظیم KeepAlive به On

شرح اجمالی:

دستور KeepAlive، استفاده مجدد آپاچی از اتصال TCP یکسان به ازای هر کلاینت، به منظور آنکه بتواند درخواست‌های HTTP بعدی از آن کلاینت را کنترل کند، بکار می‌رود. توصیه می‌شود که دستور KeepAlive بر روی On تنظیم گردد.

نحوه پیاده‌سازی:

در راستای پیاده سازی وضعیت توصیه شده موارد زیر را اجرا نمایید:

به منظور فعال بودن ارتباطات Keepalive مقدار دستور KeepAlive را در پیکربندی Apache به مقدار On، اصلاح نمایید.

```
KeepAlive On
```

SCSR-9-3: تنظیم MaxKeepAliveRequest به 100 یا بیشتر

شرح اجمالی:

دستور MaxKeepAliveRequests تعداد درخواست‌های مجاز در هر اتصال را هنگامی که KeepAlive فعال باشد، محدود می‌کند. اگر این دستور با مقدار 0 تنظیم شود، درخواست‌های نامحدود، مجاز خواهند شد. لذا توصیه می‌گردد که دستور MaxKeepAliveRequest به 100 و یا بیشتر تنظیم شده باشد.

نحوه پیاده‌سازی:

در راستای پیاده سازی وضعیت توصیه شده موارد زیر را اجرا نمایید:

مقدار دستور MaxKeepAliveRequests را در پیکربندی Apache به مقدار 100 یا بیشتر اصلاح نمایید.

```
MaxKeepAliveRequests 100
```

SCSR-9-4: تنظیم KeepAliveTimeout به 15 یا کمتر

شرح اجمالی:

تعداد ثانیه‌هایی را که آپاچی برای درخواست بعدی منتظر می‌ماند، قبل از این که اتصالی که فعال نگه داشته شده را ببندد، بوسیله دستور keepAliveTimeout مشخص می‌شوند.

نحوه پیاده‌سازی:

در راستای پیاده سازی وضعیت توصیه شده موارد زیر را اجرا نمایید:

دستور KeepAliveTimeout را در پیکربندی Apache به منظور داشتن مقدار 15 یا کمتر، اضافه یا اصلاح نمایید.

```
KeepAliveTimeout 15
```

SCSR-9-5: تنظیم محدودیت Timeout برای درخواست Headers

شرح اجمالی:

دستور RequestReadTimeout اجازه پیکربندی محدودیت‌های زمانی برای درخواست‌های کلاینت را می‌دهد. بخش هدر دستور یک مقدار timeout اولیه، یک حداکثر timeout و یک میزان حداقل timeout ارائه می‌دهد. حداقل میزان timeout مشخص می‌کند که بعد از timeout اولیه، سرور ۱ ثانیه بیشتر برای هر N بایت دریافت

شده منتظر می‌ماند. تنظیمات توصیه شده به منظور داشتن یک حداکثر وقفه ۴۰ ثانیه یا کمتر است. به خاطر داشته باشید که برای میزبان‌های مجازی SSL/TLS زمان برای handshake TLS باید در وقفه جای داده شود.

نحوه پیاده‌سازی:

- با استفاده از پیکربندی زیر ماژول mod_requesttimeout را در پیکربندی آپاچی بارگذاری نمایید.

```
LoadModule reqtimeout_module modules/mod_reqtimeout.so
```

- دستور RequestReadTimeout همانند خط زیر با تنظیم ماکزیمم مقدار وقفه هدر درخواست بر روی 40 ثانیه یا کمتر اضافه نمایید.

```
RequestReadTimeout header=20-40,MinRate=500 body=20,MinRate=500
```

SCSR-9-6: تنظیم محدودیت Timeout برای درخواست Body

شرح اجمالی:

دستور RequestReadTimeout تنظیم مقادیر وقفه برای body یک درخواست را میسر می‌سازد. این دستور یک مقدار timeout اولیه، و یک حداکثر timeout و یک میزان حداقل timeout ارائه می‌کند. میزان حداقل مشخص می‌کند که بعد از timeout اولیه، سرور ۱ ثانیه بیشتر برای هر N بایت دریافت شده منتظر خواهد ماند. تنظیمات توصیه شده به منظور داشتن یک حداکثر وقفه ۲۰ ثانیه یا کمتر است که مقدار پیش فرض body=20، MinRate=500 است.

نحوه پیاده‌سازی:

با استفاده از پیکربندی زیر ماژول mod_requesttimeout را در پیکربندی آپاچی بارگذاری نمایید.

```
LoadModule reqtimeout_module modules/mod_reqtimeout.so
```

دستور RequestReadTimeout همانند خط زیر با تنظیم ماکزیمم مقدار وقفه body بر روی 20 ثانیه یا کمتر اضافه نمایید.

```
RequestReadTimeout header=20-40,MinRate=500 body=20,MinRate=500
```

SCSR-10: محدودیت‌های درخواست

SCSR-10-1: تنظیم دستور LimitRequestLine به 512 یا کمتر

شرح اجمالی:

دستور LimitRequestLine میزان حداکثر تعداد بایت‌هایی را که آپاچی از هر خط از یک درخواست HTTP می‌تواند بخواند را تنظیم می‌نماید. لذا توصیه می‌گردد که مقدار LimitRequestLine بر روی 512 و یا کمتر تنظیم شود.

نحوه پیاده‌سازی:

در راستای پیاده سازی وضعیت توصیه شده موارد زیر را اجرا نمایید:

مقدار دستور LimitRequestline را در پیکربندی آپاچی به 512 یا کمتر تغییر دهید.

```
LimitRequestline 512
```

SCSR-10-2: اطمینان از تنظیم دستور LimitRequestFields به 100 یا کمتر

شرح اجمالی:

دستور LimitRequestFields حداکثر محدودیتی که بر روی تعداد هدرهای درخواست HTTP که به ازای هر درخواست اجازه داده می‌شوند، را مشخص می‌سازد. لذا توصیه می‌گردد که دستور LimitRequestFields به 100 و یا کمتر تنظیم شده باشد.

نحوه پیاده‌سازی:

در راستای پیاده سازی وضعیت توصیه شده موارد زیر را اجرا نمایید:

۱. دستور LimitRequestFields در پیکربندی Apache را برای داشتن مقدار 100 یا کمتر، اضافه یا اصلاح نمایید. اگر دستور موجود نباشد، به طور پیش‌فرض به پیکربندی زمان کامپایل بستگی دارد که مقدار پیش‌فرض 100 می‌باشد.

```
LimitRequestFields 100
```

SCSR-10-3: تنظیم دستور LimitRequestFieldsize به 1024 یا کمتر

شرح اجمالی:

دستور LimitRequestFieldSize ماکزیمم اندازه فیلد هدر درخواست HTTP را تنظیم می‌کند. لذا توصیه می‌شود که دستور LimitRequestFieldSize بر روی 1024 یا کمتر تنظیم گردد.

نحوه پیاده‌سازی:

در راستای پیاده سازی وضعیت توصیه شده موارد زیر را اجرا نمایید:

۱. دستور LimitRequestFieldsize در پیکربندی Apache را برای داشتن مقدار 1024 یا کمتر اصلاح نمایید.

SCSR-10-4: تنظیم دستور LimitRequestBody به 102400 یا کمتر

شرح اجمالی:

دستور LimitRequestBody حداکثر اندازه body درخواست HTTP را تنظیم می‌کند. لذا توصیه می‌شود که دستور LimitRequestBody بر روی 102400 یا کمتر تنظیم گردد.

نحوه پیاده‌سازی:

در راستای پیاده سازی وضعیت توصیه شده موارد زیر را اجرا نمایید:

دستور LimitRequestBody در پیکربندی Apache را برای داشتن مقدار 102400 (100K) یا کمتر، اضافه یا اصلاح کنید. لطفا مستندات Apache را مطالعه نموده سپس درمی‌یابید که این دستور، اندازه‌ی آپلودهای فایل به وب سرور را محدود می‌کند.

```
LimitRequestBody 102400
```

SCSR-11: فعال نمودن SELinux به منظور محدود نمودن فرآیندهای Apache

SCSR-11-1: فعال نمودن SELinux در حالت Enforcing

شرح اجمالی:

SELinux (Security-Enhanced Linux) یک ماژول امنیتی هسته لینوکس می‌باشد و سیستم نظارتی آن بر اساس MAC (Mandatory Access Control) یا کنترل دسترسی اجباری با حالت Enforcing می‌باشد. این ماژول توسط آژانس امنیت ملی آمریکا ایجاد گردید و بر اساس سیاست‌های تعریف شده، قوانین مربوط به فایل‌ها و فرآیندها را در یک سیستم لینوکس اجرا و اقدامات مربوطه را محدود کند.

نحوه پیاده‌سازی:

در راستای پیاده سازی وضعیت توصیه شده موارد زیر را اجرا نمایید:

در صورت عدم فعال بودن SELinux در فایل پیکربندی، فایل /etc/selinux/config را ویرایش کرده و مقدار SELinux را به حالت enforcing تغییر داده و سیستم را برای قبول پیکربندی جدید، مجددا راه‌اندازی نمایید.

```
SELINUX=enforcing
```

اگر حالت جاری، enforcing نیست و راه‌اندازی مجدد به صورت فوری، امکان‌پذیر نبود، می‌توان حالت جاری را با دستور setenacle به enforcing تنظیم نمود.

```
# setenforce 1
```

SCSR-11-2: اجرای فرآیندهای آپاچی در متن محدود شده httpd_t

شرح اجمالی:

SELinux شامل سیاست‌های هدفمند قابل تنظیمی می‌باشد که ممکن است برای محدود کردن سرور http آپاچی برای اجرای حداقل امتیازات مورد استفاده قرار گیرد، به طوری که سرور httpd تنها حداقل دسترسی را به دایرکتوری‌ها، فایل‌ها و پورت‌های شبکه داشته باشد. دسترسی توسط انواع فرآیندهای (دامنه‌ها) تعریف شده برای فرآیندهای httpd کنترل می‌شود. بیش از صد httpd تکی مربوط به انواع تعریف شده در یک سیاست پیش‌فرض SELinux آپاچی وجود دارد که شامل بسیاری از افزونه‌ها و برنامه‌های متداول آپاچی مانند php، nagios، smokeping و نمونه‌های دیگر می‌باشد. سیاست‌های پیش‌فرض SELinux برای نصب یک آپاچی پیش‌فرض به خوبی عمل می‌کنند، اما پیاده‌سازی سیاست‌های مورد نظر SELinux در زمینه یک وب سرور پیچیده یا بسیار سفارشی شده نیازمند یک توسعه نسبتاً قابل توجه و تست کردن مناسب آن است که هم روش کار SELinux و

هم عملیات‌ها و الزامات دقیق برنامه وب را در بر می‌گیرد. تمام دایرکتوری‌ها و فایل‌ها برای این که توسط فرآیند وب سرور قابل دسترس باشند، باید برچسب‌های امنیتی مناسب داشته باشند.

در زیر نمونه‌هایی از پر استفاده‌ترین‌ها ارائه شده است:

۱. http_port_t – پورت‌های شبکه مجاز برای شنود
۲. httpd_sys_content_t – دسترسی خواندن به دایرکتوری‌ها و فایل‌ها با محتوای وب
۳. httpd_log_t – دایرکتوری‌ها و فایل‌هایی که باید برای لاگ‌ها قابل نوشتن مورد استفاده قرار گیرند.
۴. httpd_sys_script_exec_t – دایرکتوری‌ها و فایل‌هایی برای محتوای قابل اجرا.

نحوه پیاده‌سازی:

اگر فرآیندهای httpd در حال اجرا به محتوای httpd-t SELinux محدود نشده باشند، محتوا را برای باینری httpd و باینری apachetl بررسی کرده و باینری httpd را برای داشتن محتوای http-exec-t و apachetl را برای داشتن محتوای initrc-exec-t مانند زیر، تنظیم کنید. همچنین، در نظر داشته باشید که بر روی برخی از پلتفرم‌ها مانند Ubuntu، به جای httpd، فایل اجرایی Apache، apache2 نامیده می‌شود.

```
# ls -alZ /usr/sbin/httpd /usr/sbin/httpd.* /usr/sbin/apachectl
-rwxr-xr-x. root root system_u:object_r:initrc_exec_t:s0 /usr/sbin/apachectl
-rwxr-xr-x. root root system_u:object_r:httpd_exec_t:s0 /usr/sbin/httpd
-rwxr-xr-x. root root system_u:object_r:httpd_exec_t:s0 /usr/sbin/httpd.worker
-rwxr-xr-x. root root system_u:object_r:httpd_exec_t:s0 /usr/sbin/httpd.event
```

اگر فایل‌های اجرایی به درستی برچسب‌گذاری نشده باشند، ممکن است با دستور chcon دوباره برچسب‌گذاری شوند، با این حال، برچسب‌گذاری فایل سیستم، مبتنی بر سیاست‌های محتوایی فایل SELinux است و سیستم‌فایل‌ها در برخی موقعیت‌ها با توجه به این سیاست، مجددا برچسب‌گذاری می‌شوند.

```
# chcon -t initrc_exec_t /usr/sbin/apachectl
# chcon -t httpd_exec_t /usr/sbin/httpd /usr/sbin/httpd.*
```

از آنجایی که ممکن است سیستم فایل بر اساس سیاست SELinux دوباره برچسب‌گذاری شود، بهتر است سیاست SELinux با گزینه‌ی "-l" semanage fcontext بررسی شود. اگر این سیاست موجود نباشد، آنگاه الگو با استفاده

از گزینه‌ی "-a" اضافه می‌شود. دستور restorecon که در زیر نشان داده شده، برچسب محتوای فایل را با توجه به سیاست جاری، بازیابی می‌کند و زمانی مورد نیاز می‌باشد که الگو اضافه شود.

```
# ### Check the Policy
# semanage fcontext -l | fgrep 'apachectl'
/usr/sbin/apachectl regular file system_u:object_r:initrc_exec_t:s0
# semanage fcontext -l | fgrep '/usr/sbin/httpd'
/usr/sbin/httpd regular file system_u:object_r:httpd_exec_t:s0
/usr/sbin/httpd.worker regular file system_u:object_r:httpd_exec_t:s0
/usr/sbin/httpd.event regular file system_u:object_r:httpd_exec_t:s0
# ### Add to the policy, if not present
# semanage fcontext -f -- -a -t httpd_exec_t '/usr/sbin/httpd'
# semanage fcontext -f -- -a -t httpd_exec_t '/usr/sbin/httpd.worker'
# semanage fcontext -f -- -a -t httpd_exec_t '/usr/sbin/httpd.event'
# semanage fcontext -f -- -a -t initrc_exec_t /usr/sbin/apachectl

# ### Restore the file labeling accord to the SELinux policy
# restorecon -v /usr/sbin/httpd /usr/sbin/httpd.* /usr/sbin/apachectl
```

SCSR-11-3: اطمینان از عدم قرارگیری مد http_t بر روی Permissive

شرح اجمالی:

علاوه بر تنظیم کل پیکربندی SELinux در حالت مجاز، می‌توان انواع Processهای مجزا (Domains) مانند httpd_t را در حالت مجاز نیز تنظیم نمود. حالت مجاز هیچ دسترسی و یا عملکردی را محدود نمی‌نماید، در عوض هر اقدامی که منع شده باشد را تنها Log می‌نماید.

نحوه پیاده‌سازی:

در راستای پیاده سازی وضعیت توصیه شده موارد زیر را اجرا نمایید:

اگر http_t از نوع مجاز باشد، دراینصورت می‌بایست حالت مجاز سفارش شده با استفاده از دستور semanage حذف گردد.

```
# semanage permissive -d httpd_t
```

SCSR-11-4: اطمینان از فعال بودن بولین‌های ضروری SELinux

شرح اجمالی:

بولین‌های SELinux اجازه رفتار مختص وب سرور آپاچی را یا می‌دهند یا نمی‌دهند. نمونه‌های متداول در این خصوص عبارتند از این که آیا اجرای CGI مجاز است، یا آیا سرور httpd برای برقراری ارتباط با ترمینال فعلی (tty) مجاز است. برقرای ارتباط با ترمینال، ممکن است برای وارد کردن یک کلمه عبور به منظور رمزگشایی یک کلید خصوصی در طول راه اندازی، ضروری باشد.

نحوه پیاده‌سازی:

به منظور غیرفعال نمودن بولین‌های httpd که غیرضروری شناخته شده‌اند، از دستور setsebool به صورت زیر استفاده گردد که با استفاده از option " p– " اعمال تغییرات به صورت دائمی صورت می‌پذیرد.

```
# setsebool -P httpd enable cgi off
# getsebool httpd enable cgi
httpd_enable_cgi --> off
```

SCSR-12: فعال نمودن AppArmor به منظور محدود نمودن فرآیندهای Apache

SCSR-12-1: فعال نمودن AppArmor Framework

شرح اجمالی:

AppArmor یک ماژول امنیتی هسته لینوکس است که کنترل دسترسی اجباری مبتنی بر نام، به همراه سیاست‌های امنیتی ارائه می‌دهد. AppArmor قوانین مربوط به برنامه‌ها را به منظور دسترسی فایل، اتصالات شبکه و محدود کردن اقدامات مبتنی بر سیاست‌های تعریف شده اجرا می‌کند.

نحوه پیاده‌سازی:

در راستای پیاده سازی وضعیت توصیه شده موارد زیر را اجرا نمایید:

- درصورت عدم وجود دستور aa-status، بسته AppArmor نصب نشده و نیاز به نصب یک بسته مناسب مدیریتی از توزیع لینوکس می‌باشد. به عنوان مثال:

```
# apt-get install apparmor
# apt-get install libapache2-mod-apparmor
```

- مطابق با شرح زیر به منظور فعال‌سازی فریم‌ورک AppArmor می‌بایست اسکریپت init.d اجرا گردد.

```
# /etc/init.d/apparmor start
```

SCSR-12-2: سفارشی نمودن مشخصات AppArmor آپاچی

شرح اجمالی:

AppArmor شامل پروفایل‌های قابل تنظیمی می‌باشد که برای محدود کردن وب سرور آپاچی به اجرای حداقل امتیازات مورد استفاده قرار می‌گیرد به طوری که سرور تنها حداقل دسترسی را به دایرکتوری‌ها، فایل‌ها و پورت‌های شبکه داشته باشد. دسترسی به وسیله پروفایلی که برای process مخصوص apache2 تعریف شده کنترل می‌گردد. پروفایل پیش‌فرض AppArmor معمولاً یک پروفایل مجاز است که اجازه دسترسی خواندن-نوشتن را به تمام سیستم‌فایل‌ها می‌دهد. بنابراین توجه به این نکته مهم است که پروفایل پیش‌فرض به منظور به‌اجرا درآوردن حداقل امتیازات، سفارشی شود. از ابزارهای AppArmor مانند aa-autodep، aacomplain، و aa-logprof را می‌توان جهت ایجاد یک پروفایل اولیه بر اساس استفاده واقعی بهره برد. با این حال، تست کامل، بررسی و سفارشی‌سازی برای اطمینان از این که محدودیت‌های پروفایل آپاچی قابلیت‌های لازم را در حین پیاده‌سازی حداقل امتیازات میسر می‌سازند، ضروری خواهد بود.

نحوه پیاده‌سازی:

در راستای پیاده سازی وضعیت توصیه شده موارد زیر را اجرا نمایید:

- سرور Apache را متوقف کنید.

```
# service apache2 stop
```

- پروفایل خالی apache2 را بر اساس وابستگی‌های برنامه ایجاد نمایید.

```
# aa-autodep apache2
Writing updated profile for /usr/sbin/apache2.
```

- پروفایل apache2 را در حالت complain تنظیم نمایید، به طوری که اقدامات مقابل تخلفات دسترسی اجازه اجرا داده شده و لاگ ثبت گردد.

```
# aa-complainapache2
Setting /usr/sbin/apache2 to complain mode.
```

- سرویس apache2 را آغاز نمایید.

```
# service apache2 start
```

- برنامه‌ی تحت وب را با بررسی همه‌ی قابلیت‌ها به صورت کامل تست نمایید، به طوری که AppArmor، همه‌ی لاگ‌ها ضروری از منابع در دسترس را تولید کند. لاگ‌ها از طریق ابزار syslog فرستاده شده و معمولا در یکی از فایل‌های /var/log/syslog یا /var/log/messages یافت می‌شوند. همچنین لازم بذکر است، متوقف کردن و راه‌اندازی مجدد وب سرور نیز، بخشی از فرآیند تست می‌باشد.
- از aa-logprof برای بروزرسانی پروفایل بر اساس لاگ‌ها تولید شده در طول تست، استفاده نمایید. این ابزار برای اصلاحات پیشنهادی بر روی پروفایل، بر اساس لاگ‌ها به کار گرفته می‌شود. همچنین، ممکن است لاگ‌ها به منظور بروزرسانی پروفایل، به صورت دستی مجددا مورد بازنگری قرار داده شوند.

```
# aa-logprof
```

- با حذف محتوای نامناسب و اضافه کردن قوانین دسترسی مناسب، پروفایل را بررسی و ویرایش کنید. دایرکتوری‌های با فایل‌های در دسترس و با مجوزهای مشابه می‌توانند با استفاده از wild-cardها ساده شوند. همچنین پروفایل بروزرسانی‌شده را با استفاده از دستور apparmorparser مجددا بارگذاری نمایید.

```
# apparmor_parser -r /etc/apparmor.d/usr.sbin.apache2
```

- پروفایل بروز شده جدید را برای بررسی مجدد لاگ‌ها رد شده‌ی apparmor تست نمایید. پروفایل را در صورت لزوم، بروزرسانی کرده و مجددا بارگذاری نمایید. همچنین تست‌های برنامه را تا زمانی که هیچ لاگ رد شده‌ی apparmor ایجاد نشده (به جز دسترسی که بایستی ممنوع باشد) تکرار نمایید.

```
# tail -f /var/log/syslog
```

- پروفایل apache2 را در حالت enforce قرار داده، apparmor را مجددا بارگذاری نموده و سپس عملکرد وب سایت را مجددا تست نمایید.

```
# aa-enforce /usr/sbin/apache2
# /etc/init.d/apparmor reload
```

SCSR-12-3: اطمینان از قرارگیری پروفایل AppArmor در حالت Enforce

شرح اجمالی:

پروفایل‌های AppArmor ممکن است در یکی از این سه حالت باشند: disable، complain یا enforce. در حالت enforce، هر گونه تخطی از کنترل‌های دسترسی ثبت می‌شود، اما محدودیتی اجرا نمی‌شود. همچنین توصیه می‌شود هنگامی که حالت یک پروفایل تغییر پیدا کرد لذا لازم است که سرور آپاچی مجدد راه‌اندازی شود، در غیر اینصورت فرآیند در حال اجرا ممکن است با این سیاست محدود نگردد.

نحوه پیاده‌سازی:

در راستای پیاده سازی وضعیت توصیه شده موارد زیر را اجرا نمایید:

- وضعیت پروفایل را در حالت enforce تنظیم نمایید.

```
# aa-enforce apache2
Setting /usr/sbin/apache2 to enforce mode.
```

- فعالیت سرور آپاچی را متوقف نموده و از عدم اجرای آن اطمینان حاصل نمایید. در برخی موارد ممکن است AppArmor از متوقف شدن صحیح وب سرور جلوگیری و ممانعت نماید که در این حالت شاید نیازمند متوقف نمودن دستی سرویس و یا راه‌اندازی مجدد سرور باشیم.

```
# service apache2 stop
  * Stopping web server apache2
# service apache2 status
  * apache2 is not running
```

- سرویس آپاچی را مجدد راه‌اندازی نمایید.

```
# service apache2 start
  * Starting web server apache2
```

پیوست

در این بخش چک‌لیستی به منظور ممیزی محصول مورد نظر ارائه شده است. چک‌لیست شامل سه جدول است. جدول اول، جدول ممیز می‌باشد. در این جدول، اطلاعات مربوط به شخصی که پیکربندی امن را انجام می‌دهد یا آن را ممیزی می‌کند، وارد می‌شود. همچنین نتایج پیکربندی یا ممیزی به صورت اختصار در این جدول درج می‌گردد. جدول دوم، محل وارد کردن مشخصات سروری است که Apache HTTP Server 2.2 روی آن نصب شده است. جدول سوم، جدول تنظیماتی است که باید بررسی یا اعمال شوند. در صورت صحت اعمال تنظیم در هر ردیف، ستون وضعیت مربوط به آن با علامت √ نمایش داده خواهد شد.

ممیز	
ممیز: نام: ---------- ایمیل: ---------- تلفن: ----------	**تاریخ:**

تنظیمات	**تعداد**	**توضیحات**
تطابق		
عدم تطابق		
تنظیمات حذف شده		
تنظیمات اضافه شده		
مجموع تنظیمات اعمال شده		

مشخصات سرور	
آدرس MAC	
آدرس IP	
نام ماشین	
شماره اموال	
مدیر سیستم	نام: ---------- ایمیل: ---------- تلفن: ----------
تاریخ	

جدول ممیزی

جدول ممیزی خلاصه‌ای از تمامی الزامات بیان شده در متن سند می‌باشد. قابل ذکر است که ستون‌های "وضعیت" و "قابلیت پیاده‌سازی" باید توسط ممیز و برای هر سیستم حاوی این برنامه تکمیل گردد. در ستون وضعیت، ممیز باید از عبارت‌های "قبول" و "رد" متناسب با وضعیت الزام در محصول مورد ارزیابی استفاده نماید. در ستون قابلیت پیاده‌سازی، ممیز باید قابلیت پیاده‌سازی الزام برای محصول مورد ارزیابی را با عبارات "دارد" و "ندارد" بیان نماید. در صورتی که الزامی برای محصول مذکور قابلیت پیاده‌سازی نداشته باشد، علت عدم قابلیت پیاده‌سازی آن باید در ذیل جدول توضیح داده شود.

شناسه	وضعیت	تنظیمات	قابلیت پیاده‌سازی تنظیمات	مقدار پیش‌فرض	مقدار مطلوب
SCWS-1		برنامه‌ریزی و راه‌اندازی			
SCWS-1-1		تهیه چک لیست برنامه‌ریزی پیش از راه‌اندازی	دارد	ندارد	تهیه چک لیست و اعمال آن
SCWS-1-2		عدم نصب سرویس‌های مختلف بر روی یک سرور	دارد	فعال	غیرفعال
SCSR-1-3		نصب Apache	دارد	ندارد	استفاده از معیار منتشر شده جهت نصب
SCSR-2		به حداقل رساندن ماژول‌های Apache			
SCSR-2-1		فعال نمودن ماژول‌های احراز هویت و اعطای مجوز ضروری	دارد	ندارد	فعال
SCSR-2-2		فعال نمودن ماژول پیکربندی Log	دارد	ندارد	فعال
SCSR-2-3		غیرفعال نمودن ماژول WebDAV	دارد	ندارد	غیرفعال
SCSR-2-4		غیرفعال نمودن ماژول Status	دارد	ندارد	غیرفعال
SCSR-2-5		غیرفعال نمودن ماژول Autoindex	دارد	ندارد	غیرفعال
SCSR-2-6		غیرفعال نمودن ماژول Proxy	دارد	ندارد	غیرفعال
SCSR-2-7		غیرفعال نمودن ماژول‌های دایرکتوری‌های کاربر	دارد	ندارد	غیرفعال

SCSR-2-8		غیرفعال نمودن ماژول info	دارد	ندارد	غیرفعال
SCSR-3		**اصول، مجوزها و مالکیت**			
SCSR-3-1		اجرای وب سرور Apache با کاربری غیر root	دارد	ندارد	اجرا با کاربری غیر root و پیکربندی User و Group
SCSR-3-2		درنظر گرفتن پوسته نامعتبر برای حساب کاربری آپاچی	دارد	ندارد	در نظر گرفتن پوسته نامعتبر یانologin به منظور ورود
SCSR-3-3		قفل نمودن حساب کاربری Apache	دارد	ندارد	استفاده از دستور passwd به منظور غیرفعال نمودن حساب کاربری
SCSR-3-4		تنظیم مالکیت بر روی دایرکتوری‌ها و فایل‌های Apache	دارد	به صورت پیش‌فرض مالکیت از آن کاربری می‌باشد که نرم‌افزار را ایجاد کرده و کاربر root	مطابق نحوه پیاده‌سازی اجرا گردد.
SCSR-3-5		تنظیم شناسه گروه بر روی دایرکتوری‌ها و فایل‌های Apache	دارد	به صورت پیش‌فرض مالکیت از آن کاربری می‌باشد که نرم‌افزار را ایجاد کرده و کاربر root	مطابق نحوه پیاده‌سازی اجرا گردد.
SCSR-3-6		محدود کردن دیگر دسترسی‌های Write بر روی دایرکتوری‌ها و فایل‌های Apache	دارد	ندارد	مطابق نحوه پیاده‌سازی اجرا گردد.
SCSR-3-7		امن نمودن دایرکتوری Core Dump	دارد	ندارد	تنظیم دستور CoreDumpDirectory برای دایرکتوری‌های متعلق به کاربر ریشه
SCSR-3-8		امن نمودن Lock File	دارد	ندارد	مطابق نحوه پیاده‌سازی اجرا گردد.
SCSR-3-9		امن نمودن Pid فایل	دارد	ندارد	مطابق نحوه پیاده‌سازی اجرا گردد.
SCSR-3-10		امن نمودن فایل ScoreBoard	دارد	ندارد	مطابق نحوه پیاده‌سازی اجرا گردد.
SCSR-3-11		محدود کردن دسترسی‌های Write گروه بر روی دایرکتوری‌ها و فایل‌های Apache	دارد	ندارد	مطابق نحوه پیاده‌سازی اجرا گردد.
SCSR-3-12		محدود نمودن دسترسی Write گروه بر روی ریشه اصلی پوشه و فایل‌ها	دارد	ندارد	مطابق نحوه پیاده‌سازی اجرا گردد.

SCSR-4		کنترل دسترسی آپاچی			
SCSR-4-1		منع دسترسی به دایرکتوری ریشه OS	دارد	ندارد	مسدود کردن دسترسی به دایرکتوری ریشه سیستم عامل
SCSR-4-2		اجازه دسترسی مناسب به محتوای وب	دارد	مطابق بند ۱ از توضیحات پیش‌فرض جدول	استفاده از از دستورات Allow یا Require
SCSR-4-3		محدودسازی نادیده گرفتن برای دایرکتوری ریشه OS	دارد	<Directory /> . . . AllowOverride None . . . </Directory>	تنظیم مقدار AllowOverride بر روی none
SCSR-4-4		محدودسازی نادیده گرفتنOverRide برای کلیه دایرکتوری‌ها	دارد	ندارد	تنظیم مقدار همه دستورهایAllowOverride بر روی none
SCSR-5		به حداقل رساندن قابلیت‌ها، محتوا و گزینه‌ها			
SCSR-5-1		محدود نمودن Optionها برای دایرکتوری ریشه OS	دارد	ندارد	تنظیم مقدار Option بر روی none
SCSR-5-2		محدود نمودن Optionها برای دایرکتوری ریشه وب	دارد	ندارد	تنظیم مقدار همه دستورهای Option و بر روی none یا Multiviews
SCSR-5-3		به حداقل رساندن Optionها برای دیگر دایرکتوری‌ها	دارد	ندارد	تنظیمات همانند بند SCR-1-5-2
SCSR-5-4		حذف نمودن محتوای پیش‌فرض HTML	دارد	ندارد	مطابق نحوه پیاده‌سازی اجرا گردد.
SCSR-5-5		حذف CGI Content Printenv پیش‌فرض	دارد	ندارد	حذف printenv از دایرکتوری cgi-bin
SCSR-5-6		حذف CGI Content test-cgi پیش‌فرض	دارد	ندارد	حذف test-cgi از دایرکتوری cgi-bin
SCSR-5-7		محدود نمودن متدهای HTTP Request	دارد	بدون محدودیت	مطابق نحوه پیاده‌سازی اجرا گردد.
SCSR-5-8		غیر فعال کردن متد HTTP TRACE	دارد	ندارد	تنظیم TraceEnable با مقدار off در پیکربندی سطح سرور
SCSR-5-9		محدود نمودن نسخه پروتکل HTTP	دارد	ندارد	مطابق نحوه پیاده‌سازی اجرا گردد.
SCSR-5-10		محدود نمودن دسترسی به فایل‌های .ht*	دارد	غیرقابل دسترس	مطابق نحوه پیاده‌سازی اجرا گردد.

SCSR-5-11		محدود نمودن پسوند فایل‌ها	دارد	بدون محدودیت	مطابق نحوه پیاده‌سازی اجرا گردد.
SCSR-5-12		فیلترینگ درخواست‌ها بر اساس آدرس IP	دارد	ندارد	مطابق نحوه پیاده‌سازی اجرا گردد.
SCSR-5-13		محدود نمودن دستور Listen	دارد	ندارد	مطابق نحوه پیاده‌سازی اجرا گردد.
SCSR-5-14		محدود نمودن Optionهای فریم مرورگر	دارد	ندارد	مطابق نحوه پیاده‌سازی اجرا گردد.
SCSR-6		**عملیات ورود، نظارت و نگهداری**			
SCSR-6-1		پیکربندی لاگ خطا	دارد	ندارد	مطابق نحوه پیاده‌سازی اجرا گردد.
SCSR-6-2		یکربندی Syslog Facility برای دریافت خطاها	دارد	ErrorLog"logs/error_log"	ErrorLog"syslog:local1"
SCSR-6-3		تنظیمات لاگ دسترسی	دارد	ندارد	مطابق نحوه پیاده‌سازی اجرا گردد.
SCSR-6-4		بازبینی و فضای لاگ	دارد	ندارد	مطابق نحوه پیاده‌سازی اجرا گردد.
SCSR-6-5		اعمال وصله‌های قابل اجرا	دارد	ندارد	مطابق نحوه پیاده‌سازی اجرا گردد.
SCSR-6-6		راه‌اندازی و فعال‌سازی ModSecurity	دارد	غیرفعال	راه‌اندازی و فعال‌سازی ModSecurity
SCSR-6-7		راه‌اندازی و فعال‌سازی مجموعه قوانین اصلی OWASP ModSecurity	دارد	غیرفعال	نصب و پیکربندی OWASP ModSecurity Core Rule Set
SCSR-7		**پیکربندیSSL/TSL**			
SCSR-7-1		نصب و راه‌اندازی mod_ssl و یا mod_nss	دارد	ندارد	مطابق نحوه پیاده‌سازی اجرا گردد.
SCSR-7-2		نصب یک گواهی مورد اعتماد و معتبر	دارد	ندارد	مطابق نحوه پیاده‌سازی اجرا گردد.
SCSR-7-3		حفاظت از کلید خصوصی سرور	دارد	ندارد	مطابق نحوه پیاده‌سازی اجرا گردد.
SCSR-7-4		غیرفعال کردن پروتکل SSL v3.0	دارد	SSLProtocol all –SSLv2	مطابق نحوه پیاده‌سازی اجرا گردد.

SCSR-7-5		محدود کردن رمزهای ضعیف SSL	دارد	مطابق بند ۲ از توضیحات پیش‌فرض جدول	مطابق نحوه پیاده‌سازی اجرا گردد.
SCSR-7-6		غیرفعال نمودن SSL ناامن Renegotiation	دارد	SSLInsecureRenegotiation off	تنظیم SSLInsecurRenegotiation به off
SCSR-7-7		اطمینان از فعال نبودن فشرده سازی SSL	دارد	مقدار SSLCompression در نسخه 2.2.24 تا 2.2.25، on و برای نسخه 2.2.26، off	تنظیم SSLCompression به off
SCSR-7-8		غیر فعال کردن پروتکل TSLv1.0	دارد	SSLProtocol all -SSLv2	تنظیم SSLProtocol به TLSv1.2 یا TSLv1.1
SCSR-7-9		فعال کردن OCSP Stapling	دارد	SSLUseStapling Off SSLStaplingCache no default value	تنظیم SSLUseStapling به on
SCSR-7-10		فعال کردن HTTP Strict Transport Security	دارد	غیرفعال	پیکربندی دستور Header همانند نمونه ذکر شده در قسمت نحوه پیاده‌سازی
SCSR-8		**نشت اطلاعات**			
SCSR-8-1		تنظیم ServerToken به 'Prod'	دارد	ندارد	Full
SCSR-8-2		تنظیم امضاء سرور بر روی Off	دارد	ندارد	off
SCSR-8-3		نشت اطلاعات از طریق محتوای پیش‌فرض Apache	دارد	ندارد	عدم دسترسی به آیکون‌های آپاچی
SCSR-9		**انکار سرویس Mitigation**			
SCSR-9-1		تنظیم Timeout به ۱۰ یا کمتر	دارد	ندارد	۱۰ یا کمتر
SCSR-9-2		تنظیم دستور KeepAlive به On	دارد	ندارد	تنظیم دستور KeepAliveبه On
SCSR-9-3		تنظیم MaxKeepAliveRequest به ۱۰۰ یا بیشتر	دارد	ندارد	۱۰۰ یا بیشتر
SCSR-9-4		تنظیم KeepAliveTimeout به 15 یا کمتر	دارد	ندارد	۱۵ و یا کمتر
SCSR-9-5		تنظیم محدودیت Timeout برای درخواست Headers	دارد	header=20-40 MinRate=500	۴۰ ثانیه

SCSR-9-6		تنظیم محدودیت Timeout برای درخواست Body	دارد	body=20 MinRate=500	۲۰ ثانیه
SCSR-10		**محدودیت‌های درخواست**			
SCSR-10-1		تنظیم دستور LimitRequestLine به ۵۱۲ یا کمتر	دارد	ندارد	۵۱۲ و کمتر
SCSR-10-2		تنظیم دستور LimitRequestFields به ۱۰۰ یا کمتر	دارد	ندارد	۱۰۰ و کمتر
SCSR-10-3		تنظیم دستور LimitRequestFieldsize به ۱۰۲۴ یا کمتر	دارد	ندارد	۱۰۲۴ و کمتر
SCSR-10-4		تنظیم دستور LimitRequestBody به ۱۰۲۴۰۰ یا کمتر	دارد	ندارد	۱۰۲۴۰۰ و کمتر
SCSR-11		**فعال نمودن SELinux به منظور محدود نمودن فرآیندهای Apache**			
SCSR-11-1		فعال نمودن SELinux در حالت Enforcing	دارد	SELinux غیرفعال	Enforcing
SCSR-11-2		اجرای فرآیندهای آپاچی در متن محدود شده httpd_t	دارد	SELinux غیرفعال	مطابق نحوه پیاده‌سازی اجرا گردد.
SCSR-11-3		اطمینان از عدم قرارگیری مد http_t بر روی Permissive	دارد	عدم وجود در حالت غیرمجاز	غیر فعال نمودن از حالت Permissive
SCSR-11-4		اطمینان از فعال بودن بولین‌های ضروری SELinux	دارد	SELinux غیرفعال	فعال بودن بولین‌های ضروری SELinux
SCSR-12		**فعال نمودن AppArmor به منظور محدود نمودن فرآیندهای Apache**			
SCSR-12-1		فعال نمودن AppArmor Framework	دارد	AppArmor فعال	فعال
SCSR-12-2		سفارشی نمودن مشخصات AppArmor آپاچی	دارد	پروفایل آپاچی به صورت پیش‌فرض غیرسخت گیرانه می‌باشد.	تغییر پروفایل پیش‌فرض به حداقل امتیازات

SCSR-12-3		اطمینان از قرارگیری پروفایل AppArmor در حالت Enforce	دارد	enforce	enforce

توضیحات مقادیر پیش‌فرض جدول:

۱. SCSR-1-4-2: تنظیمات مسیر ریشه وب به صورت زیر نمایش داده شده است:

```
<Directory "/usr/local/apache2/htdocs">
    . . .
    Require all granted
    . . .
</Directory>
```

۲. SCSR-1-7-5: مقادیر پیش‌فرض به صورت زیر نمایش داده شده است:

```
SSLCipherSuite default depends on OpenSSL version.
SSLHonorCipherOrder Off
```