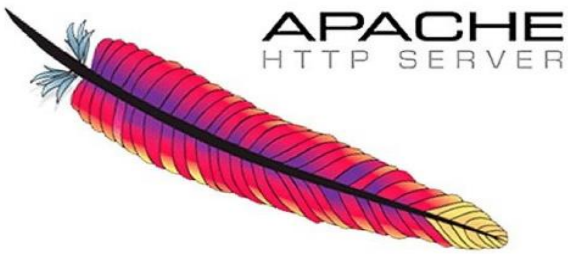
بسمه تعالی

مباحث امنیتی و پیکربندی امن
Apache HTTP Server 2.2, 2.4



Secure Config Web Server (SCWS)

SCWS-Apache-HTTP-2.2, 2.4

تهیه کننده

محمد مقاله

9619185686

فهرست مطالب

مقدمه

نکات امنیتی ذکر شده، در این سند پیرامون سرور آپاچی نسخه 2.2.x و 2.4.x بوده هر چند که در مواردی با یکدیگر مشابه ولی در برخی موارد نیز با یکدیگر متفاوت می باشد. این موارد در هر بخش زیر بیان شده است.
در سند زیر نحوه پیاده سازی هر نسخه ذکر شده است.
ضمناً در بخش 13، به طور اختصاصی به نکات امنیتی Apache Http Server نسخه 2.4.x پرداخته شده است.
قبل از ایجاد تغییرات در وب سرور آپاچی شما بایستی نکات پایه ای درباره ی آپاچی سرور را بدانید.

1. دایرکتوری ریشه اسناد در مسیر مقابل قرار دارد : /var/www/html یا /var/www
2. فایل پیکربندی (configuration) اصلی : /etc/hhtpd/conf/httpd.conf در لینوکس های RHEL/CentOs/Fedora و /etc/apache2/apache2.conf در لینوکس های Debian/Ubuntu.
3. پورت پیش فرض HTTP: 80 TCP
4. پورت پیش فرض HTTPS: 443 TCP
5. برای تست کردن کردن فایل تنظیمات پیکربندی و syntax آن از دستور مقابل استفاده می شود: httpd –t
6. دسترسی به فایل های log وب سرور در مسیر مقابل قرار دارد : /var/log/httpd/access_log
7. فایل های Error وب سرور در مسیر مقابل قرار دارد : /var/log/httpd/error_log

تنظیمات

SCWS-1: برنامه ریزی و راه اندازی

SCWS-1-1: تهیه چک لیست برنامه ریزی پیش از راه اندازی

شرح اجمالی:

می بایست پیش و یا هنگام نصب و راه اندازی موارد و توصیه های زیر بررسی گردند:

- بررسی و اجرای سیاست های امنیتی داخلی سازمان و موارد مرتبط با امنیت نرم افزارهای تحت وب
- پیاده سازی و ایجاد زیرساخت امن شبکه با توجه به کنترل دسترسی به/ از وب سرور و با استفاده از تجهیزاتی چون فایروال، روتر و سوییچ

- امن سازی سیستم عامل وب سرور بوسیله حداقل سازی سرویس های مربوط به شبکه، اعمال مناسب بسته های ارائه شده و پیکربندی امن آن با توجه به معیار امنیت اینترنتی توصیه شده برای هر پلتفرم
- پیاده سازی فرآیندهای نظارتی مرکزی مربوط به لاگ
- اجرای مکانیزم و فرآیندهای مربوط به پایش فضای حافظه و بازبینی لاگ ها در سرور
- آموزش و آگاهی رسانی برنامه نویسان به منظور توسعه برنامه های 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 (نام های فردی ، POCDialing) (شماره های تلفن) و Brute Force (آدرس های ایمیلی که مطابق با نام کاربری های سیستم واقعی هستند)
- پیکربندی صحیح و امن سرور DNS به منظور جلوگیری از حملات مخرب با توجه به توصیه های CIS BIND DNS Benchmark
- پیاده سازی یک سیستم تشخیص نفوذ شبکه (IDS) به منظور نظارت بر حملات علیه وب سرور

نحوه پیاده سازی:

مطابق با شرح اجمالی پیاده سازی گردد.

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

شرح اجمالی:

سرور اغلب طیف وسیعی از سرویس های پیش فرض غیر ضروری را ارائه می دهد که برخی از این سرویس ها، سرور را از لحاظ امنیتی در معرض خطرات جدی قرار می دهند. یک سرور می تواند خدمات زیادی انجام دهد این بدان معنا نمی باشد که می بایست بطور حتم همه این خدمات باید فعال شوند. لذا می بایست تعداد خدمات، سرویس ها و فرآیندهای پشت صحنه ای که بر روی سروری که میزبان Apache است ، اجرا ، به موارد ضروری محدود شده، و وب سرور تنها کارکرد اصلی سرور شود.

نحوه پیاده سازی:

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

```
chkconfig <servicename> off
```

SCWS-1-3 : نصب Apache

شرح اجمالی:

معیار CIS Apache (Center for Internet Security) حاضر، در اکثر موارد استفاده از آپاچی باینری ارائه شده توسط فروشنده به منظور کاهش دوباره کاری و افزایش اثربخشی تعمیر و نگهداری و امنیت پیشنهاد می کند. با این حال، برای حفظ معیار به صورت عمومی و قابل استفاده و انطباق با کلیه پلت فرم های یونیکس/ لینوکس، یک منبع پیش فرض برای این معیار ایجاد و استفاده شده است.

نحوه پیاده سازی:

نصب و راه اندازی بستگی زیادی به پلتفرم سیستم عامل دارد. در خصوص توزیع های سورس می توان از لینک http://httpd. apache. org/docs/2. 2/install. Html بهره برده و در خصوص توزیع Red Hat Enterprise Linux 5 می توان از دستور زیر استفاده نمود.

```
# yum install httpd
```

SCWS-2: به حداقل رساندن ماژول های Apache

SCWS-2-1: فعال نمودن ماژول های احراز هویت و اعطای مجوز ضروری

شرح اجمالی:

ماژول هایی با عنوان authn_* به منظور احراز هویت و ماژول های authz_* جهت ارائه مجوز می باشند. همچنین Apache شامل دو نوع احراز هویت basic و digest می باشد. لذا تنها ماژول هایی را که مورد نیاز می باشند، فعال نمایید.

نحوه پیاده سازی:

به منظور تصمیم گیری درخصوص نصب ماژول های لازم و ضروری از مستندات و توضیحاتی که برای هر ماژول موجود است، می توان استفاده نمود. البته برخی ماژول ها به صورت پیش فرض مورد نیاز نیستند و البته غیرفعال

هستند. در ضمن یکسری از ماژول ها به صورت dynamic در آپاچی وجود دارند و بوسیله کامنت کردن یا حذف دستور LoadModule از فایل پیکربندی (معمولا http. conf) می توان آن ها را غیرفعال نمود.

SCWS-2-2: فعال نمودن ماژول پیکربندی لاگ

شرح اجمالی:

ماژول log_config قابلیت انعطاف ثبت وقایع درخواست های کلاینت را فراهم کرده و پیکربندی به منظور اینکه چه نوع اطلاعاتی درفایل لاگ ذخیره می شود را فراهم می آورد.

نحوه پیاده سازی:

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

- در خصوص توزیع های سورس با ماژول های 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 ماژول هایی که به صورت پویا بارگذاری شده اند ، 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
```

SCWS-2-3: غیرفعال نمودن ماژول WebDAV

شرح اجمالی:

ماژول های mod_dav و mod_dav_fs آپاچی از قابلیت WebDAV (احراز هویت و نسخه بندی توزیع شده مبتنی بر وب) برای آپاچی پشتیبانی می کنند. WebDAV یک افزونه برای پروتکل HTTP است که به کلاینت اجازه ایجاد، جابجایی، تغییر و یا حذف فایل ها و منابع را بر روی وب سرور می دهد.

نحوه پیاده سازی:

به منظور پیاده سازی این تنظیم و غیرفعال کردن WebDAV می بایست هر یک از موارد زیر را انجام داد:

1. در خصوص توزیع های سورس با ماژول های static اسکریپت configure/. آپاچی را بدون اینکه آیتم های 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
```

2. در خصوص ماژول هایی که به صورت پویا بارگذاری شده اند ، دستور LoadModule را برای ماژول های 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
```

SCWS-2-4: غیرفعال نمودن ماژول Status

شرح اجمالی:

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

نحوه پیاده سازی:

به منظور پیاده سازی این تنظیم و غیرفعال کردن mod_status می بایست هر یک از موارد زیر را انجام داد:

1. در خصوص توزیع های سورس با ماژول های 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
```

2. در خصوص ماژول هایی که به صورت پویا بارگذاری شده اند، دستور LoadModule را برای ماژول mod_status از فایل httpd.conf حذف و یا کامنت گذاری نمایید.

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

SCWS-2-5: غیرفعال نمودن ماژول Autoindex

شرح اجمالی:

ماژول autoindex به طور خودکار صفحات وبی شامل محتویات دایرکتوری ها بر روی سرور را ایجاد می کند، که به طور معمول هنگامی استفاده می شوند که نیازی به ایجاد یک index.html نیست.

نحوه پیاده سازی:

به منظور پیاده سازی این تنظیم و غیرفعال کردن mod_autoindex می بایست هر یک از موارد زیر را انجام داد:

1. در خصوص توزیع های سورس با ماژول های static، اسکریپت configure/. آپاچی را بدون اینکه Script Option مربوط به configure disable-autoindex-- فعال شده باشد، اجرا نمایید .

```
$ cd $DOWNLOAD/httpd-2. 2. 22
$. /configure -disable-autoindex
```

2. در خصوص ماژول هایی که به صورت پویا بارگذاری شده اند، دستور LoadModule را برای ماژول mod_autoindex از فایل httpd.conf حذف و یا کامنت گذاری نمایید.

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

2-6 –SCWS: غیرفعال نمودن ماژول Proxy

شرح اجمالی:

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

نحوه پیاده سازی:

به منظور پیاده سازی این تنظیم و غیرفعال کردن ماژول proxy می بایست هر یک از موارد زیر را انجام داد:

1. در خصوص توزیع های سورس با ماژول های static اسکریپت configure/. آپاچی را بدون اینکه آیتم

--enable-modules=configure مربوط به Script Option در mod_proxy فعال شده باشد، اجرا نمایید.

```
$ cd $DOWNLOAD/httpd-2. 2. 22
$. /configure
```

2. در خصوص ماژول هایی که به صورت پویا بارگذاری شده اند ، 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
```

SCWS-2-7: غیرفعال نمودن ماژول های دایرکتوری های کاربر

شرح اجمالی:

می بایست دستور UserDir غیرفعال شود تا بدین وسیله دایرکتوری های 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 می دهد.. 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 پیاده سازی:

به منظور پیاده سازی این تنظیم و غیرفعال کردن ماژول user directories می بایست هر یک از موارد زیر انجام گیرد:

1. در خصوص توزیع های سورس با ماژول های 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
```

2. در خصوص ماژول هایی که به صورت پویا بارگذاری شده اند، دستور LoadModule را برای ماژول mod_userdir از فایل httpd.conf حذف و یا کامنت گذاری نمایید.

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

SCWS-2-8: غیرفعال نمودن ماژول Info

شرح اجمالی:

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

نحوه پیاده سازی:

به منظور پیاده سازی این تنظیم و غیرفعال کردن ماژول mod_info می بایست هر یک از موارد زیر را انجام داد:

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

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

2. در خصوص ماژول هایی که به صورت پویا بارگذاری شده اند، دستور LoadModule را برای ماژول mod_info از فایل httpd.conf حذف و یا کامنت گذاری نمایید.

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

SCWS-3: اصول، مجوزها و مالکیت

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

شرح اجمالی:

اگرچه آپاچی معمولاً با دسترسی های root به منظور شنود به پورت 80 و 443 آغاز می شود، اما به منظور انجام سرویس دهی می تواند و می بایست کاربری غیر از کاربر ریشه اجرا شود. از دستورهای User و Group به منظور تعیین کاربر و گروهی که ایجاد کننده فرآیندهای آپاچی فرض می شوند، استفاده می گردد.

نحوه پیاده سازی:

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

1. 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
```

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

```
User apache
Group apache
```

تنظیمات مربوطه در آپاچی نسخه 2.4.x بدین صورت می باشد.

```
# groupadd apache

# useradd –G apache apache
```

مالکیت دایرکتوری نصب آپاچی را به کاربر غیر ممتاز بدهید.

```
# chown –R apache:apache /opt/apache
```

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

```
User apache
Group apache
```

SCWS-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 می بایست یک پوسته نامعتبر یا nologin به منظور اطمینان از عدم استفاده از این حساب برای ورود در نظر گرفته شود.

نحوه پیاده سازی:

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

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

SCWS-3-3: قفل نمودن حساب کاربری Apache

شرح اجمالی:

حساب کاربری که آپاچی تحت آن اجرا می شود نباید یک رمز عبور معتبر داشته باشد، اما می بایست غیرفعال گردد.

نحوه پیاده سازی:

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

```
# passwd -l apache
```

SCWS-3-4: تنظیم مالکیت بر روی دایرکتوری ها و فایل های Apache

شرح اجمالی:

دایرکتوری ها و فایل های آپاچی بایستی متعلق به root باشند. این امر در مورد تمام دایرکتوری های نرم افزار آپاچی و فایل های نصب شده نیز صدق می کند.

نحوه پیاده سازی:

پیاده سازی موارد زیر:

تنظیم مالکیت در دایرکتوری های $APACHE_PREFIX مانند /usr/local/apache2:

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

SCWS-3-5: تنظیم شناسه گروه بر روی دایرکتوری ها و فایل های Apache

شرح اجمالی:

می بایست دایرکتوری ها و فایل های آپاچی برای داشتن یک شناسه گروه از root، یا یک گروه معادل root تنظیم شوند. این امر در مورد تمام دایرکتوری های نرم افزار آپاچی و فایل های نصب شده نیز به کار برده می شود. 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 پیاده سازی:

تنظیم مالکیت در دایرکتوری های $APACHE_PREFIX مانند /usr/local/apache2:

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

SCWS-3-6: محدود کردن دسترسی های Write بر روی دایرکتوری ها و فایل های Apache

شرح اجمالی:

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

نحوه پیاده سازی:

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

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

تنظیمات مربوطه در آپاچی نسخه 2.4.x بدین صورت می باشد.

در آپاچی نسخه 2.4.x به دایرکتوری $web_server بروید:

مجوز فولدر های bin و conf را به صورت زیر تغییر دهید:

```
# chmod –R 750 bin conf
```

SCWS-3-7: امن نمودن دایرکتوری Core Dump

شرح اجمالی:

دستور CoreDumpDirectory برای مشخص کردن مسیری که تلاش های آپاچی را به منظور ایجاد core dump، نشان می دهد، بکار می رود. دایرکتوری پیش فرض، دایرکتوری ServerRoot می باشد، با این حال core dump ها در سیستم های لینوکسی به صورت پیش فرض غیرفعال است. در رویدادهایی که نیاز به core dump ها می باشد، می بایست دایرکتوری قابلیت نوشتن را توسط آپاچی داشته باشد و باید نیازمندی های امنیتی که در پایین و در قسمت پیاده سازی تعریف شده است را تأمین کند.

نحوه پیاده سازی:

دستور CoreDumpDirectory را از فایل های پیکربندی Apache حذف کنید یا از اینکه دایرکتوری پیکربندی شده از شرایط زیر برخوردار است اطمینان حاصل کنید.

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

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

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

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

SCWS-3-8: امن نمودن Lock File

شرح اجمالی:

دستور LockFile سبب ایجاد مسیری به فایل Lock در زمانی می شود که آپاچی از دستورهای فراخوانی سیستمی مانند fcntl(2) یا flock(2) برای پیاده سازی یک mutex (قفل) استفاده می کند.

اکثر سیستم های لینوکس به طور پیش فرض از سمافورها به جای آن استفاده می کنند، 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 پیاده سازی:

1. مسیر دایرکتوری ای را پیدا کنید که LockFile در آن ایجاد می شود که به صورت پیش فرض، دایرکتوری ServerRoot/logs است.
2. اگر مسیر یک دایرکتوری در DocumentRoot است، دایرکتوری را اصلاح نمایید.
3. مالکیت و گروه را به ریشه تغییر دهید: اگر در حال حاضر، ریشه به حساب نمی آید.
4. مجوزها را تغییر دهید، به طوری که دایرکتوری تنها توسط ریشه یا کاربری که Apache ابتدا با او آغاز به کار می کند، قابل نوشتن باشد (مقدار پیش فرض، ریشه است.)
5. بررسی نمایید که دایرکتوری فایل قفل، به جای یک سیستم فایل نصب شده ی NFS، بر روی یک هارد دیسک نصب شده ی محلی باشد.

SCWS-3-9: امن نمودن Pid فایل

شرح اجمالی:

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

نحوه پیاده سازی:

1. دایرکتوری ای را پیدا کنید که PidFile در آن ایجاد می شود. مقدار پیش فرض، دایرکتوری ServerRoot/logs است .
2. اگر PidFile در یک دایرکتوری در Apache DocumentRoot است، دایرکتوری را اصلاح کنید.
3. مالکیت و گروه را به ریشه تغییر دهید: در صورتی که در حال حاضر ریشه نباشد.
4. مجوزها را طوری را تغییر دهید که دایرکتوری، تنها توسط ریشه یا کاربری که Apache ابتدا با او آغاز به کار می کند، قابل نوشتن باشد (مقدار پیش فرض، ریشه می باشد).

SCWS-3-10: امن نمودن فایلScoreBoard

شرح اجمالی:

دستور ScoreBoardFile مسیر فایلی را تنظیم می کند که سرور برای ارتباطات بین Process (IPC) در میان Process های آپاچی از آن استفاده می کند. در اکثر پلت فرم های لینوکس، حافظه به اشتراک گذاشته شده، به جای یک فایل در سیستم فایل 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 می بایست در یک دایرکتوری امن قرار داده شود.

نحوه پیاده سازی:

1. بررسی نمایید که آیا ScoreBoardFile در هیچ یک از فایل های پیکربندی Apache، مشخص شده است یا خیر. اگر جواب منفی بود، هیچ تغییری لازم نیست.
2. اگر دستور، موجود باشد، دایرکتوری ای را پیدا کنید که ScoreBoardFile در آن ایجاد می شود. مقدار پیش فرض، دایرکتوری ServerRoot/logs است.
3. اگر ScoreBoardFile در یک دایرکتوری در Apache DocumentRoot است، دایرکتوری را اصلاح نمایید.
4. اگر مالکیت و گروه root نیست آن را به root:root تغییر دهید.
5. مجوزها را به گونه ای تغییر دهید که دایرکتوری، تنها توسط root یا کاربری که Apache ابتدا با او آغاز به کار می کند، قابل نوشتن باشد (مقدار پیش فرض، ریشه است.)
6. بررسی نمایید که دایرکتوری فایل scoreboard، به جای یک سیستم فایل نصب شده ی NFS، بر روی یک هارد دیسک نصب شده ی محلی باشد.

SCWS-3-11: محدود کردن دسترسی های نوشتن گروه بر روی دایرکتوری ها و فایل های Apache

شرح اجمالی:

به طور کلی می بایست مجوزهای گروهی برای دایرکتوری های آپاچی، r-x بوده و مجوزهای فایل نیز می بایست همین باشند، مگر اینکه قابل اجرا نباشند و قابل اجرا بودن آن ها نیز مناسب نباشد. این امر در مورد تمام دایرکتوری

ها و فایل های نصب شده نرم افزار آپاچی به استثناء ریشه اصلی پوشه $DOCROOT تعریف شده توسط DocumentRoot آپاچی و به طور پیش فرض برای $APACHE_PREFIX/htdocs نیز به کار می رود. ممکن است یک گروه توسعه وب برای دسترسی write دایرکتوری ها و فایل ها در مسیر اصلی پوشه تعیین شوند که به محتوای وب اجازه بروزرسانی داده شود.

نحوه پیاده سازی:

به منظور حذف دسترسی نوشتن گروه بر روی فایل ها و دایرکتوری های $APACHE_PREFIX، دستور زیر را اجرا نمایید.

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

SCWS-3-12: محدود نمودن دسترسی Write گروه بر روی ریشه اصلی دایرکتوری و فایل ها

شرح اجمالی:

ممکن است لازم باشد مجوزهای گروهی برای دایرکتوری های $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 دایرکتوری ها یا فایل ها موجود در ریشه سند را نداشته باشد.

نحوه پیاده سازی:

به منظور حذف دسترسی نوشتن گروه بر روی فایل ها و دایرکتوری های $DOCROOT با گروه Apache، دستور زیر را اجرا نمایید.

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

SCWS-4: کنترل دسترسی آپاچی

SCWS-4-1: منع دسترسی به دایرکتوری ریشه ی OS

شرح اجمالی:

دستور Directory، به منظور پیکربندی خاص دایرکتوری؛ کنترل های دسترسی و بسیاری از ویژگی ها و گزینه های دیگر را میسر می سازد. یک استفاده مهم که باعث ایجاد یک سیاست انکار پیش فرض می گردد این است که اجازه دسترسی به دایرکتوری ها و فایل های سیستم عامل جز آن هایی که به طور خاص مجاز هستند را نمی دهد. این کار با مسدود کردن دسترسی به دایرکتوری ریشه سیستم عامل با استفاده از یکی از دو روش زیر می باشد که البته تنها یکی از این دو دستور باید انجام شود و مجاز به استفاده از هر دو نیستید:

1. استفاده از دستور Apache Deny به همراه یک دستور Order
2. Apache Require استفاده از دستور

هر دو روش موثر هستند و در حال حاضر ترکیبی از Order/Deny/Allow توصیه نمی گردد، زیرا آن ها سه راه را فراهم می سازند که تمامی دستورات طی یک دستور مشخص شده مورد پردازش قرار می گیرند. در مقابل، دستور Require بر روی اولین تطابق، مشابه با نحوه عمل Rule در فایروال، عمل می کند. دستور Require به صورت پیش فرض در آپاچی نسخه 2 .4 می باشد.

نحوه پیاده سازی:

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

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

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

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

شرح اجمالی:

به منظور سرویس دهی محتوای وب، می بایست از دستور Allow استفاده نموده تا امکان دسترسی مناسب به دایرکتوری ها، مکان ها و میزبان های مجازی که حاوی محتوای وب هستند را میسر سازد.

نحوه پیاده سازی:

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

1. به منظور یافتن همه ی عناصر <Directory > و <Location>، فایل های 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 زمینه های مختلف، از جمله میزبان های مجازی یا عناصر خاص مانند <Proxy> وجود داشته باشند.
2. شامل دستورهای مناسب Require، با مقادیر متناسب با نوع استفاده هر دایرکتوری می باشد.

پیکربندی های 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>
```

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

شرح اجمالی:

دستور OverRide، به فایل هایhtaccess. اجازه می دهند تا بدون در نظر گرفتن بسیاری از پیکربندی ها، از جمله احراز هویت، کنترل انواع سند، ایندکس های تولید شده به طور خودکار، کنترل دسترسی و Optionها مورد استفاده قرار گیرند. هنگامی که سرور یک فایل htaccess. دسترسی پیدا می کند (که توسط AccessFileName مشخص شده) ، می بایست از اینکه کدام دستورهای اعلام شده در آن فایل می توانند اطلاعات دسترسی قبلی را نادیده بگیرد، مطلع باشد. هنگامی که این دستور بر روی None تنظیم شود، در آن صورت فایل های htaccess. به طور کامل نادیده گرفته می شوند. در اینصورت، سرور حتی برای خواندن فایل های htaccess. در فایل سیستم هیچ تلاشی نمی کند. لذا چنانچه این دستور بر روی All تنظیم شود، در این صورت هر دستوری که یک زمینه htaccess. داشته باشد، در فایل های htaccess. مجاز می باشد.

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 پیاده سازی:

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

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

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

تنظیمات مربوطه در آپاچی نسخه 2.4.x بدین صورت می باشد.

به دایرکتوری $web_server/conf بروید و httpd.conf را باز کرده و در سطح ریشه Directory را جستجو کنید:

```
<Directory />
```

```
Options –Indexes
 AllowOverride None
...
</Directory>
```

SCWS-4-4: محدودسازی OverRide برای کلیه دایرکتوری ها

شرح اجمالی:

دستور AllowOverRide، به فایل های htaccess. اجازه می دهند تا بدون در نظر گرفتن بسیاری از پیکربندی ها، از جمله احراز هویت، کنترل انواع سند، ایندکس های تولید شده به طور خودکار، کنترل دسترسی و Option ها مورد استفاده قرار گیرند. هنگامی که سرور یک فایل htaccess. پیدا می کند (که توسط AccessFileName مشخص شده)، می بایست از اینکه کدام دستورهای اعلام شده در آن فایل می توانند اطلاعات دسترسی قبلی را نادیده بگیرد، مطلع باشد. هنگامی که این دستور بر روی None تنظیم شود، در آن صورت فایل های htaccess. به طور کامل نادیده گرفته می شوند. در اینصورت، سرور حتی برای خواندن فایل های htaccess. در فایل سیستم هیچ تلاشی نمی کند. لذا چنانچه این دستور بر روی All تنظیم شود، در اینصورت هر دستوری که یک زمینه htaccess. داشته باشد، در فایل های htaccess. مجاز می باشد.

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 ی پیاده سازی:

1. فایل های پیکربندی Apache را به منظور یافتن AllowOverride، جستجو نمایید (شامل httpd.conf وهر فایل پیکربندی موجود در آن)
2. مقدار همه دستورهای AllowOverride را none تنظیم نمایید.

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

SCWS-5: به حداقل رساندن قابلیت ها، محتوا و گزینه ها

SCWS-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 می دهد.

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 پیاده سازی:

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

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

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

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

شرح اجمالی:

دستور Options اجازه پیکربندی خاص Optionها را می دهد، از جمله می توان به موارد زیر اشاره نمود:

- اجرای CGI دنبال کردن Symbloic link ها 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#option

نحوه پیاده سازی:

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

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

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

SCWS-5-3: به حداقل رساندن Optionها برای دیگر دایرکتوری ها

شرح اجمالی:

دستور Options اجازه پیکربندی خاص Optionها، از جمله اجرای CGI، دنبال کردن Symbolic link ها، دستورات سمت سرور و content negotiation را می دهد.

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 پیاده سازی:

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

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

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

SCWS-5-4: حذف نمودن محتوای پیش فرض HTML

شرح اجمالی:

محتواهای پیش فرضی برای نصب و راه اندازی آپاچی موجود است که البته برای استفاده، مناسب و مورد نیاز نمی باشد. عملکرد اصلی برای این محتوای نمونه، ارائه یک وب سایت پیش فرض، ارائه دفترچه راهنمای کاربر یا نشان دادن ویژگی های خاص وب سرور است که می بایست تمام محتوایی که مورد نیاز نمی باشد حذف گردد.

نحوه پیاده سازی:

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

1. Index. html یا صفحه welcome به طوری که به صورت پیش فرض 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 نمی شود، زیرا با بروزرسانی بسته ، ممکن است مجدداً 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>
```

2. محتوای user manual آپاچی را حذف کرده یا پیکربندی هایی که به بارگذاری دستی ارجاع می دهند را به کامنت تبدیل نمایید.

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

3. هر پیکربندی کنترل کننده 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>
```

4. 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>
```

5. 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>
```

SCWS-5-5 : حذف **CGI Content Printenv**پیش **فرض**

شرح اجمالی:

اکثر وب سرور ها از جمله بسته نصبی آپاچی به صورت پیش فرض دارای محتوای CGI می باشند که برای کاربری وب سرور الزامی و مقتضی نمی باشد. کاربرد اصلی این برنامه نمونه شبیه سازی قابلیت های وب سرور می باشد. یکی از محتواهای CGI پیش فرض برای بسته نصبی آپاچی، اسکریپت prinenv می باشد. این اسکریپت تمام مقادیر محیطی CGI را به درخواست کننده بر می گرداند که شامل تعداد زیادی توضیحات در مورد پیکربندی سیستم و مسیرهای سیستمی می باشد.

نحوه پیاده سازی:

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

1. محل فایل ها و دایرکتوری های cgi-bin فعال شده در پیکربندی Apache را از طریق دستورهای Script،

ScriptInterpreterSource یا ScriptAliasMatch ، ScriptAlias را تعیین نمایید.

2. در صورتی که printenv در دایرکتوری cgi-bin نصب شده بود، آن را حذف نمایید.

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

SCWS-5-6: حذف **CGI Content test-cgi**پیش **فرض**

شرح اجمالی:

اکثر وب سرورها از جمله بسته نصبی آپاچی به صورت پیش فرض دارای محتوای CGI می باشند که برای کاربری وب سرور الزامی و مقتضی نمی باشد. کاربرد اصلی این برنامه نمونه شبیه سازی قابلیت های وب سرور می باشد. یکی از محتواهای CGI پیش فرض برای بسته نصبی آپاچی اسکریپت test-cgi می باشد. این اسکریپت تمام مقادیر محیطی CGI را به درخواست کننده بر میگرداند که شامل تعداد زیادی توضیحات در مورد پیکربندی سیستم می باشد.

نحوه پیاده سازی:

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

1. محل فایل ها و دایرکتوری های cgi-bin فعال شده در پیکربندی Apache از طریق دستورهای Script، ScriptInterpreterSourceیا ScriptAliasMatch ، ScriptAlias را تعیین نمایید
2. در صورتی که test-cgi در دایرکتوری cgi-bin نصب شده بود ، آن را حذف نمایید.

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

SCWS-5-7 محدود نمودن روش های **HTTP Request**:

شرح اجمالی:

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

نحوه پیاده سازی:

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

1. محل فایل های پیکربندی Apache و فایل های پیکربندی گنجانده شده در آن را تعیین نمایید.
2. دستور را در مسیر اصلی پوشه جستجو نمایید، مانند:

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

1. 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>
```

1. در فایل های پیکربندی Apache، دستورهای دیگری به جز دایرکتوری ریشه ی 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 سیستم فایل OS هستند ، نه سلسله مراتب مکان هایی در URLهای وب سایت که توسط Apache مورد دسترسی قرار می گیرند.

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

تنظیمات مربوطه در آپاچی نسخه 2.4.x بدین صورت می باشد.

به مسیر دایرکتوری $Web_Server/conf بروید و درفایل httpd.conf موارد زیر را اضافه کنید.

```
<LimitExcept GET POST HEAD>

Deny from all

</LimitExcept>
```

SCWS-5-8: غیرفعال 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 پیاده سازی:

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

1. محل فایل پیکربندی اصلی Apache را همانند httpd.conf تعیین نمایید.

2. دستور TraceEnable را به پیکربندی سطح سرور با مقدار off اضافه نمایید.
پیکربندی سطح سرور بالاترین سطح پیکربندی به شمار می رود و مانند دایرکتوری های دیگری همچون <Directory> یا <Location> مرتبط نمی باشد .

```
TraceEnable off
```

SCWS-5-9: محدود نمودن نسخه پروتکل HTTP

شرح اجمالی:

از ماژول های mod-rewrite یا mod_security می توان برای عدم دسترسی به نسخه های پروتکل های HTTP قدیمی و نامعتبر استفاده کرد. نسخه 1,1 RFC از HTTP مورخ ژوئن 1999 می باشد، و از نسخه 1,2 توسط آپاچی پشتیبانی شده است. استفاده از این ماژول برای دسترسی به نسخه های قدیمی HTTP مانند 1,0 و قبل از آن غیر مجاز می باشد.

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 پیاده سازی:

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

1. ماژول mod_rewrite را با انجام هرکدام از موارد زیر، برای Apache بارگذاری نمایید:
   1. با اضافه کردن گزینه ی enable-rewrite-- به اسکریپت ./configure، هنگام نصب Apache، ماژول mod_rewrite نیز نصب می گردد.

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

   2. 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
```

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

```
RewriteEngine on
```

3. 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 ورژن های دیگر پروتکل، به پیکربندی سطح سرور اضافه نمایید.

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

4. به طور پیش فرض، تنظیمات پیکربندی mod-rewrite از سرور اصلی توسط میزبان های مجازی به ارث برده نمی شوند. بنابراین، اضافه کردن دستورهای زیر به هر بخش برای به ارث بردن تنظیمات سرور اصلی نیز ضروری است.

```
RewriteEngine On
RewriteOptions Inherit
```

SCWS-5 10: محدود نمودن دسترسی به فایل های *ht .

شرح اجمالی:

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

نحوه پیاده سازی:

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

1.خطوط زیر را در پیکربندی آپاچی در سطح پیکریندی سرور اضافه و یا اصلاح نمایید.

```
<FileMatch “^.\ ht”>

Require all denied

</FileMatch>
```

SCWS-5 -11: محدود نمودن پسوند فایل ها

شرح اجمالی:

دسترسی به پسوند فایل های نامتناسبی که انتظار نمی رود بخش مشروعی از وب سایت باشند، را محدود نمایید به نحوی که نتوان با دستور FilesMatch از آن ها استفاده نمود

نحوه پیاده سازی:

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

1. لیستی از پسوند فایل های موجود بر روی وب سرور ایجاد نمایید. دستور find/awk زیر ممکن است مفید باشد، اما احتمالا با توجه به دایرکتوری های وب روت مناسب برای وب سرور شما، به سفارشی سازی نیاز دارد. توجه داشته باشید که دستور find، هر فایل بدون نقطه (.) در نام فایل را نادیده می گیرد، زیرا انتظار می رود که آن ها محتوای مناسبی نباشند.

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

2. لیست پسوند های فایل موجود برای محتوای مناسب بر روی سرور وب، مرور کرده و آن هایی که نامناسب هستند را حذف کرده و هر پسوند فایل اضافه ای که انتظار می رود در آیندهای نزدیک به سرور وب افزوده شود را اضافه نمایید.

3. به منظور رد نمودن دسترسی به همه فایل ها به صورت پیش فرض ، دستور FilesMatch را اضافه نمایید.

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

4. یک دستور FilesMatch دیگر اضافه کنید که اجازه ی دسترسی به پسوند فایل هایی که به طور خاص در فرآیند مرور مرحله ی 2، مجوز داشته اند را می دهد. 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>
```

SCWS-5-12: فیلترینگ درخواست ها بر اساس آدرس IP

شرح اجمالی:

از ماژول mod_rewrite می توان جهت رد کردن تقاضا برای url های خاص استفاده کرد این ماژول این امکان را می دهد که به جای IP از Host Name استفاده گردد. عادی ترین دسترسی به وب سایت از طریق مرورگرها و نرم افزار خودکار با استفاده از یک نام میزبان می باشد، 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 پیاده سازی:

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

1.ماژول mod_rewrite را با انجام هرکدام از موارد زیر، برای Apache بارگذاری کنید:

1. با اضافه کردن گزینه ی enable-rewrite به اسکریپت. /Apache, configure با mod_rewrite بارگذاری شده به صورت static را در طول فرآیند ساخت، بسازید.

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

2. 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
```

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

```
RewriteEngine On
```

3. 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]
```

SCWS-5 -13: محدود کردن دستور Listen

شرح اجمالی:

دستور Listen، شماره پورت هایی را که وب سرور آپاچی می خواهد بر روی آن گوش دهد را مشخص می کند. همچنین IP هایی که تنها درخواست آن ها پذیرفته خواهد شد، را نیز مشخص می کند. به جای نامحدود شدن شنود بر روی تمام آدرس های IP در دسترس برای سیستم، آدرس یا آدرس های 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 پیاده سازی:

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

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

```
Listen 80

Listen 0.0.0.0:80

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

2.دستورهای Listen در فایل پیکربندی Apache را برای داشتن آدرس های 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
```

SCWS-5 -14: محدود نمودن Optionهای فریم مرورگر

شرح اجمالی:

دستور Header به هدرهای پاسخ HTTP سرور اجازه اضافه، جایگزین یا ادغام را می دهد. 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 سایت های دیگر محدود کنند، استفاده خواهیم نمود.

نحوه پیاده سازی:

1.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
```

SCWS-6: عملیات ورود، نظارت و نگهداری

SCWS-6-1: پیکربندی لاگ خطا

شرح اجمالی:

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

نحوه پیاده سازی:

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

1. مقدار LogLevel را در پیکربندی آپاچی اضافه نموده و یا تغییر دهید به گونه ای که مقدار notice و یا پایین تر داشته باشد. توجه نمایید که داشتن یک مقدار از info یا debug نیز می توان بکار برد در صورتی که نیاز به لاگ کامل تر وجود داشته باشد و فرآیندهای ذخیره سازی و نظارت، قادر به کنترل بار اضافی باشند. لذا مقدار پیشنهادی در این خصوص notice می باشد.

```
LogLevel notice
```

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

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

3. 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 لاگ های وب خود بوده و به مهارت ها/آموزش/ابزار برای نظارت بر لاگ ها نیاز می باشد.

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

شرح اجمالی:

جهت ارسال لاگ ها به یک syslog، دستور ErrorLog پیکربندی می شود، به طوری که لاگ ها بتوانند همراه با لاگ های سیستم پردازش و نظارت شوند.

نحوه پیاده سازی:

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

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

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

SCWS-6-3: تنظیمات لاگ دسترسی

شرح اجمالی:

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

نحوه پیاده سازی:

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

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

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

2. 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
```

3. 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 لاگ های وب خود بوده و به مهارت ها/آموزش/ابزار برای نظارت بر لاگ ها نیاز می باشد.

فرمت توکن های رشته اطلاعات زیر را فراهم می کنند:

h%: نام میزبان یا آدرس IP راه دور، اگر HostnameLookups، off باشد که مقدار پیش فرض است.

l%: نام لاگ/ شناسه ی راه دور.

u%: کاربر راه دور، اگر درخواست، تأیید شده باشد.

t%: زمان دریافت درخواست،

r%: اولین خط درخواست.

s >%: وضعیت نهایی.

b%: اندازه ی پاسخ بر اساس بایت.

i {Referer }%: مقدار متغیر برای هدر Referer .

i {User-agent }%: مقدار متغیر برای هدر User Agent .

SCWS-6-4: بازبینی و فضای لاگ

شرح اجمالی:

وجود فضای دیسک کافی بر روی پارتیشن به منظور نگهداری تمام فایل های لاگ تولید شده امر بسیار مهمی می باشد، بازه چرخش لاگ ها، اگر از فضای ذخیره سازی مرکزی برای ذخیره سازی لاگ ها مورد استفاده قرار نمی گیرد، برای حفظ حداقل 3 ماه یا 13 هفته پیکربندی شود.

نحوه پیاده سازی:

برای پیاده سازی حالت توصیه شده، گزینه a) اگر از Linux logrotate استفاده می کنید، گزینه b) اگر از یک ابزار piped Logging استفاده می کنید، را به کار بگیرید:

1. File Logging با Logrotate:

1. پیکربندی بازبینی لاگ وب را جهت مطابقت با فایل های لاگ پیکربندی شده ی خود در d/httpd، etc/logrotate، 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
}
```

2. دوره ی بازنگری و تعداد لاگ ها را طوری اصلاح نمایید که حداقل 13 هفته یا 3 ماه لاگ ها حفظ شوند. این کار ممکن است به صورت پیش فرض برای همه ی لاگ ها 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
```

3. برای هر میزبان مجازی پیکربندی شده با فایل های لاگ خود که آن فایل های لاگ در یک بازبینی لاگ مشابه نیز جای دارند، اطمینان حاصل نمایید.

2. Piped Logging :

1. فواصل بازبینی لاگ و نام های فایل لاگ را به یک فاصله ی مناسب مانند روزانه، پیکربندی نمایید.

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

2. اطمینان از اینکه به منظور نام گذاری فایل لاگ و اسکریپت های بازبینی، برای باقی ماندن حداقل ماه یا 13 هفته، فراهم شده اند.

3. برای هر میزبان مجازی پیکربندی شده با فایل های لاگ خود که آن فایل های لاگ در یک بازبینی مشابه نیز جای دارند، اطمینان حاصل نمایید.

SCWS-6-5: اعمال وصله های قابل اجرا

شرح اجمالی:

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

نحوه پیاده سازی:

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

1. هنگام ساخت از منبع:

1. نکات منتشر شده و اطلاعات امنیتی مرتبط با آن را بخوانید.
2. آخرین منبع و هر ماژول وابسته مانند mod-security را دانلود نمایید.
3. نرم افزار جدید Apache را با توجه به روند ساخت خود با گزینه های پیکربندی مشابه ایجاد نمایید.
4. نرم افزار جدید را با توجه به فرآیند تست سازمانی خود، نصب و آزمایش نمایید.
5. با توجه به روند توسعه ی سازمانی خود، نرم افزار را به تولید انتقال بدهید.

2. هنگام استفاده از بسته های پلتفرم:

1. نکات منتشر شده و اطلاعات امنیتی مرتبط با آن را بخوانید.
2. آخرین بسته ی موجود Apache و هر نرم افزار وابسته را دانلود نمایید.
3. نرم افزار جدید را با توجه به فرآیند تست سازمانی خود، نصب و آزمایش نمایید.
4. با توجه به روند توسعه ی سازمانی خود، نرم افزار را به تولید انتقال دهید.

SCWS-6-6: راه اندازی و فعال سازی ModSecurity

شرح اجمالی:

ModSecurity یک فایروال جهت برنامه های تحت وب (WAF) و منبع باز برای نظارت، ثبت، و کنترل دسترسی بلا درنگ 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 می کند 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 نمی کند. 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 سیستم های امنیت نرم افزار/ فایروال امنیتی، Mod_Security نیز نیازمند تعهد قابل توجه کارکنان برای تنظیم اولیه قوانین و کنترل هشدار هاست. در برخی موارد، این امر ممکن است نیازمند زمان کار اضافه تری از جانب توسعه دهندگان/ نگه داران برنامه، برای اصلاح آن بر اساس تجزیه و تحلیل نتایج لاگ ها می

باشد. بعد از راه اندازی، تعهد دائمی کارکنان، به ویژه بعد از ارتقاء/پچ ها، 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 پیاده سازی:

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

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

SCWS-6-7: راه اندازی و فعال سازی مجموعه قوانین اصلی OWASP ModSecurity

شرح اجمالی:

مجموعه قوانین اصلی OWASP ModSecurity (CRS)، مجموع های از قوانین دفاعی برنامه تحت وب منبع باز برای فایروال برنامه تحت وب (WAF) است. OWASP ModSecurity CRS حفاظت های اولیه را در دسته های حمله/تهدید زیر فراهم می کند:

- حفاظت HTTP – تشخیص نقض های پروتکل HTTP و یک سیاست استفاده تعریف شده به صورت محلی.
- متغیرهای لیست سیاه بلادرنگ – استفاده از اعتبار IP شخص ثالث
- انکار حفاظت های خدمات HTTP – دفاع در برابر جاری شدن HTTP و حملات HTTP DoS آهسته
- حفاظت از حملات متداول وب – تشخیص حملات علیه برنامه های وب
- خودکارسازی تشخیص – تشخیص باتها ، crawlers، اسکنرها و سایر فعالیت های مخرب سطحی

- ادغام با اسکن AV برای آپلود فایل – تشخیص فایل های مخرب آپلود شده از طریق برنامه تحت وب
- پیگیری اطلاعات حساس – پیگیری استفاده از کارت اعتباری و مسدود کردن نشتی ها
- حفاظت در برابر تروجان – تشخیص دسترسی به تروجان ها
- شناسایی ناهنجاری های برنامه – هشدار در مورد پیکربندی های اشتباه برنامه
- تشخیص خطا و پنهان کردن – پنهان کردن پیامهای خطای ارسال شده توسط سرور

توجه: همانند سیستم های امنیت نرم افزار/ فایروال امنیتی، Mod_Security نیز نیازمند تعهد قابل توجه کارکنان برای تنظیم اولیه قوانین و کنترل هشدار هاست. در برخی موارد، این امر ممکن است نیازمند زمان کار ا ضافه تری از جانب تو سعه دهندگان/ نگه داران برنامه، برای ا صلاح آن بر ا ساس تجزیه و تحلیل نتایج لاگ ها می باشد. بعد از راه اندازی، تعهد دائمی کارکنان، به ویژه بعد از ارتقاء/پچ ها، 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 پیاده سازی:

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

1. OWASP ModSecurity CRS را از صفحه ی پروژه

https://www.owasp.org/index.php/Category:OWASP_ModSecurity_Core_Rule_Set_Proj. etc

دانلود کنید.

2. دستورالعمل های فایل INSTALL را دنبال کنید.
3. فایل modsecurity_crs_10_setup.conf مورد نیاز است و قوانین دایرکتوری base_rules، پایه ی مفیدی برای بیشتر کاربردها هستند.
4. برنامه ی کاربردی را برای عملکرد درست پس از نصب CRS، تست نمایید. لاگ های خطای وب سرور و فایل the modsec_audit.log را برای درخواست های مسدود شده ی ناشی از false positiveها، بررسی نمایید.

5. همچنین، توصیه می شود که پاسخ برنامه ی کاربردی به ترافیک مخرب، مانند یک اسکنر خودکار برنامه ی تحت وب، برای حصول اطمینان از فعال بودن قوانین، تست شود. لاگ های خطای وب سرور و فایل های the modsec_audit. log بایستی لاگ حملات و کدهای پاسخ سرورها را نشان دهند.

SCWS-7 :استفاده از SSL/TSL

SCWS-7-1: نصب و راه اندازی **mod_ssl** و یا **mod_nss**

شرح اجمالی:

لایه سوکت های 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 می باشد و احراز هویت سرور و حتی کلاینت را فراهم می کند. با این حال برخلاف ادعاهای تولیدکننده، پیاده سازی SSL به طور مستقیم وب سرور شما را امن تر نمی کند! چرا که SSL برای رمزگذاری ترافیک مورد استفاده قرار می گیرد و در نتیجه محرمانگی اطلاعات خصوصی و اعتبارات کاربر را تأمین می کند. با این حال به خاطر داشته باشید تنها به خاطر این که شما مسیر داده ها را رمزگذاری کرده اید به این معنا نمی باشد که اطلاعات ارائه شده توسط کلاینت امن است ، زیرا که این اطلاعات بر روی سرور قرار دارد. همچنین از SSL نمی توان 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 استفاده ترین ماژولی است که SSL/TLS را برای آپاچی پیاده سازی می کند. یک ماژول جدیدتر بر روی سیستم های Red Hat می تواند یک تعریف یا جایگزینی برای mod_ssl باشد، و قابلیت های مشابه به علاوه خدمات امنیتی بیشتری را فراهم نماید. mod_nss یک ماژول آپاچی جهت پیاده سازی نرم افزار خدمات امنیت شبکه (NSS) از Mozilla است و طیف گسترده ای از توابع رمزنگاری به علاوه TLS را پیاده سازی می کند.

نحوه پیاده سازی:

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

1. برای نصب Apache با استفاده از توزیع های سورس، از گزینه ی =with-ssl-- برای مشخص کردن مسیر openssl و از گزینه ی پیکربندی enable-ssl-- برای اضافه کردن ماژول های SSL به ساخت،

استفاده نمایید. در صورت ناسازگاری با نسخه پلتفرم پیکربندی گزینه with-included-apr-- ضروری می باشد.

به منظور کسب جزئیات بیشتر به مستندات مربوط به Apache

http://httpd. apache. org/docs/2. 2/install. html

مراجعه نمایید:

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

2. برای نصب با استفاده از بسته های OS، معمولاً تنها نکته مهم حصول اطمینان از نصب بسته ی mod_ssl می باشد. همچنین بسته ی mod_nss نیز ممکن است نیاز به نصب داشته باشد. دستورات yum زیر برای Red Hat Linux مناسب هستند.

```
# yum install mod_ssl
```

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

شرح اجمالی:

گواهی SSL پیش فرض، self-signe می باشد و قابل اعتماد نیست. لذا می بایست یک گواهی قابل اعتماد معتبر نصب نمایید. گواهی برای معتبر بودن می بایست:

- 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 پیاده سازی:

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

1. درباره ی 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 می کند. 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.

2. یک کلید خصوصی با استفاده از openssl تولید کنید. اگرچه طول کلید اعتبارسنجی رایج در گذشته 1024 بوده، اما برای اعتبارسنجی قدرتمندتر، طول کلید 2048 توصیه می شود. کلید بایستی محرمانه باقی بماند و به طور پیش فرض 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:
```

3. 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/
```

4. CSR را به یک مرجع امضای گواهی برای امضا بفرستید و دستورالعمل های آن ها را برای ارائه و اعتبارسنجی، دنبال نمایید. CSR و گواهی امضا شده ی نهایی، تنها متن رمزگذاری شده ای هستند که یکپارچگی آن ها، نه محرمانگی شان، بایستی محافظت شود. با این حال این گواهی برای هر اتصال SSL فرستاده خواهد شد.
5. گواهی امضا شده ی 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 همه ی تعاملات با سرور، امکان پذیر است.
6. عبارت عبور استفاده شده برای رمزگذاری کلید خصوصی را فراموش نکنید. این عبارت، هر زمان که سرور در حالت https آغاز به کار می کند 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. ذخیره ی کلید خصوصی در یک متن ساده، باعث راحتی می شود، در حالی که ریسک افشای کلید را افزایش می دهد، اما اگر

خطرات آن به خوبی مدیریت شود ممکن است برای راه اندازی مجدد مناسب باشد. مطمئن شوید که فایل کلید تنها توسط ریشه، قابل خواندن است. برای رمزگشایی کلید خصوصی و ذخیره ی آن در فایل متنی، ممکن است از دستور openssl زیر استفاده شود. شما می توانید با استفاده از هدر های 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
```

7. محل فایل پیکربندی Apache برای mod-ssl را تعیین نموده و دستورهای SSLCertificateFile و SSLCertificateKeyFile را برای داشتن مسیر صحیح برای کلید خصوصی و فایل های 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 بسته ی نرم افزاری CA، گواهی شما را امضا کرده، برای سرعت بخشیدن به اتصال SSL اولیه می توانید 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
```

8. در نهایت سرویس httpd را شروع یا راه اندازی مجدد کرده و عملکرد صحیح آن را با مرورگر مورد علاقه ی خود، اعتبارسنجی نمایید.

SCWS7-3: حفاظت از کلید خصوصی سرور

شرح اجمالی:

محافظت از کلید خصوصی سرور بسیار مهم می باشد. به طور پیش فرض کلید خصوصی سرور به عنوان یک سیستم حفاظت از سرور، رمزنگاری شده است. با این حال، رمزنگاری آن به این معنا می باشد که هر بار سرور راه اندازی می شود به کلمه عبور نیاز دارد، و در حال حاضر محافظت از کلمه عبور نیز ضروری و مهم می باشد. 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، Option های موجود عبارتند از:

1. استفاده از SSLPassPhraseDialog builtin، - نیازمند یک کلمه عبور برای ورود به صورت دستی
2. استفاده از SSLPassPhraseDialog |/path/to/program برای ارائه کلمه عبور
3. استفاده از SSLPassPhraseDialog exec:/path/to/program برای ارائه کلمه عبور
4. ذخیره کلید خصوصی به صورت واضح به طوری که یک کلمه عبور مورد نیاز نباشد.

هر یک از گزینه های 1 الی 4 بالا تا زمانی که کلید و کلمه عبور به شرح زیر محافظت شوند، قابل قبول اند. گزینه 1 مزیت امنیت بیشتری نسبت به عدم ذخیره سازی کلمه عبور دارد، اما این مورد 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 گزینه های 2 و 3 در صورتی که برنامه هایی که آن ها را ارائه می دهند امن باشند می توانند امنیت بیشتری ارائه دهند. گزینه 4 از همه ساده تر است، و به طور گسترده مورد استفاده قرار گرفته و تا زمانی که کلید خصوصی به طور مناسب محافظت شود، قابل قبول می باشد.

نحوه پیاده سازی:

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

1. می بایست کلیه کلیدهای خصوصی جدا از کلیدهای عمومی ذخیره گردند. دستورهای SSLCertificateFile را در فایل های 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.
2. برای هر یک از دستورهای SSLCertificateKeyFile، مالکیت و مجوز بر روی سرور کلید خصوصی را به root:root با مجوز 0400 تغییر دهید.

SCWS-7-4: غیرفعال کردن پروتکل های SSL ضعیف

شرح اجمالی:

دستور SSLProtocol، پروتکل های SSL و TLS مجاز را مشخص می کند. می بایست هر دو پروتکل SSL v2 و SSLv3 در این دستور غیرفعال شوند، زیرا تاریخ گذشته بوده و نسبت به افشای اطلاعات آسیب پذیر می باشند.
بدین منظور تنها می بایست پروتکل های TLS فعال شوند.

نحوه پیاده سازی:

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

دستور SSLProtocol را در فایل های 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 می شوند.

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

```
SSLProtocol TLSv1
```

SCWS-7-5: محدود کردن رمزهای ضعیف SSL

شرح اجمالی:

رمزنگاری SSL ضعیف را با استفاده از دستورهای SSLCipherSuite و SSLHonorCipherOrder غیرفعال نمایید. رمزنگاری های مجاز در مذاکره با کلاینت توسط دستور SSLCipherSuite مشخص می شوند. در حالی که SSLHonorCipherOrder باعث می شود سرور در خصوص انتخاب استفاده از چه نوع رمزنگاری به جای کلاینت تصمیم گیری کند.

نحوه پیاده سازی:

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 دستورالعمل هایی برای رمزنگاری TLS را فراهم می کند، زیرا سبب حذف استفاده از رمزنگاری RC4 و Hash نمودن MD5 می شود که به نظر نمی رسد 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 می توان آن را از مجموعه مشخصات رمزنگاری حذف نمود.

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

SCWS-7-6 : محدود نمودن SSL ناامنRenegotiation

شرح اجمالی:

یک نوع حمله مرد در میان در SSLv3 و TLSv1 در نوامبر 2009 کشف شد.

ابتدا، یک راه حل و سپس یک اصلاح به عنوان یک استاندارد اینترنت به صورت RFC 574 در فوریه 2010 تصویب شد. راه حلی که Renegotiation را حذف می کند، از OpenSSL نسخه 0.9.8l و نسخه های 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

Client هایی با نسخه SSL قدیمی را دهد. هنگامی که یک ارتباط سازگار ایجاد می گردد، فعال نمودن دستور SSLInsecureRenegotiation سرور را همچنین برای حملاتی مانند man-in- (CVE-2009-3555 the-middle) آسیب پذیر می نماید. بنابراین، دستور SSLInsecureRenegotiation نباید فعال گردد.

نحوه پیاده سازی:

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

1. دستور SSLInsecureRenegotiation را در فایل های 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 نمی باشد.

```
SSLInsecureRenegotiation off
```

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

شرح اجمالی:

دستور SSLCompression فشرده سازی SSL هنگام سرویس دهی به محتوای HTTPS توسط آپاچی را کنترل می کند که آیا مورد استفاده قرار گرفته است یا خیر. توصیه می شود که دستور SSLCompression بر روی Off تنظیم شود.

نحوه پیاده سازی:

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

1. با استفاده از دستور httpd –v از نسخه 2.2.24 و یا بالاتر Apache اطمینان حاصل نمایید.
2. دستور SSLCompression را در فایل های پیکربندی Apache جستجو نمایید.
3. در صورت وجود دستور، آن را به مقدار off تغییر داده و بروزرسانی نمایید.

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

شرح اجمالی:

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

نحوه پیاده سازی:

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

1. دستور SSLProtocol را در فایل های 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.

SCWS-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 می دهد 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 می شود. دستور SSLUseStapling همراه با دستور SSLStaplingCache برای فعال کردن OCSP Stapling توسط وب سرور توصیه می شوند. 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 پیاده سازی:

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 می باشد ، SSLUseStapling را به on تغییر دهید. همچنین می بایست SSLStaplingCache به یکی از Chache هایی 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
```

SCWS-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 می دهد تا اعلام کند که تنها ارتباطات HTTPS باید به جای ارتباطات HTTP مورد استفاده قرار گیرند.

نحوه پیاده سازی:

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 می باشد، پیکربندی نمایید. ممکن است Flag های includedSubDomain و preload در هدر گنجانده شوند در صورتی که لازم و ضروری نمی باشد.

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

یا

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

SCWS-8: نشت اطلاعات

SCWS-8-1 : تنظیم Server Token به 'Prod'

شرح اجمالی:

اگر می خواهید نسخه وب سرور شما هنگام ارتباطات فاش نشود بایستی این موارد را به دقت اجرا کنید. افشای اطلاعات نسخه وب سرور به مهاجمان برای فعالیت های شناسایی شان سرعت می دهد. تنظیمات پیش فرض آپاچی، ورژن و سیستم عامل آن را فاش می کند.

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

نحوه پیاده سازی:

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
```

SCWS-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 پیاده سازی:

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
```

SCWS-8-3: نشت اطلاعات از طریق محتوای پیش فرض Apache

شرح اجمالی:

در توصیه های قبلی، ما محتوای پیش فرض مانند دفترچه های راهنمای آپاچی و برنامه های CGI پیش فرض را حذف کردیم. با این حال، اگر می خواهید نشت اطلاعات مربوط به وب سرور را بیشتر محدود کنید، مهم است که محتوای پیش فرض مانند آیکون ها بر روی وب سایت باقی نماند.

نحوه پیاده سازی:

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

1. به طور پیش فرض در توزیع سورس، محل پیکربندی های auto-index و icon را در فایل-extra/httpdautoindex. conf در نظر می گیرد، بنابراین می تواند با کامنت نمودن خطوط در فایل httpd.conffile ، مانند ذیل اقدام به غیرفعال نمودن آن ها نمایید.

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

2. در روش دیگر ، می توان 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>
```

SCWS 8-4: آسیب پذیری **Etag**

در آپاچی نسخه 2.4.x مقدار پیش فرض این سرآیند به مهاجم راه دور اجازه می دهد تا اطلاعات حساس مانند inode number, multipart MIME boundry و فرآیند های فرزند را از طریق سرآیند Etag به دست آورید. برای جلوگیری از این آسیب پذیری نیاز به تنظیم PCI compliance است.

در تنظیمات httpd.conf مورد زیر را اجرا کنید:

```
FileETag None
```

SCWS-9: انکار سرویس Mitigation

SCWS-9-1: تنظیم Timeout به 10 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 می کند. لذا توصیه می گردد که مقدار TimeOut را بر روی 10 یا کمتر تنظیم نمایید.

نحوه پیاده سازی:

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
```

SCWS-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 درخواست های HTTP بعدی از آن کلاینت را کنترل کند، بکار می رود. توصیه می شود که دستور KeepAlive بر روی On تنظیم گردد.

نحوه پیاده سازی:

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
```

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

شرح اجمالی:

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

نحوه پیاده سازی:

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
```

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

شرح اجمالی:

تعداد ثانیه هایی را که آپاچی برای درخواست بعدی منتظر می ماند، قبل از این که اتصالی که فعال نگه داشته شده را ببندد، بوسیله دستور keepAliveTimeout مشخص می شوند.

نحوه پیاده سازی:

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
```

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

شرح اجمالی:

دستور RequestReadTimeout اجازه پیکربندی محدودیت های زمانی برای درخواست های کلاینت را می دهد. بخش هدر دستور یک مقدار timeout اولیه، یک حداکثر timeout و یک میزان حداقل timeout ارائه

می دهد. حداقل میزان timeout مشخص می کند که بعد از timeout اولیه، سرور 1 ثانیه بیشتر برای هر N بایت دریافت شده منتظر می ماند. تنظیمات توصیه شده به منظور داشتن یک حداکثر وقفه 40 ثانیه یا کمتر است. به خاطر داشته باشید که برای میزبان های مجازی SSL/TLS زمان برای handshake TLS باید در وقفه جای داده شود.

نحوه پیاده سازی:

- 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
```

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

شرح اجمالی:

دستور RequestReadTimeout تنظیم مقادیر وقفه برای body یک درخواست را میسر می سازد. این دستور یک مقدار timeout اولیه، و یک حداکثر timeout و یک میزان حداقل timeout ارائه می کند. میزان حداقل مشخص می کند که بعد از timeout اولیه، سرور 1 ثانیه بیشتر برای هر N بایت دریافت شده منتظر خواهد ماند. تنظیمات توصیه شده به منظور داشتن یک حداکثر وقفه 20 ثانیه یا کمتر است که مقدار پیش فرض body=20، MinRate=500 است.

نحوه پیاده سازی:

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
```

SCWS-10: محدودیت های درخواست

SCWS-10-1: تنظیم دستور LimitRequest به 512 یا کمتر

شرح اجمالی:

دستور LimitRequestLine میزان حداکثر تعداد بایت هایی را که آپاچی از هر خط از یک درخواست HTTP می تواند بخواند را تنظیم می نماید. لذا توصیه می گردد که مقدار LimitRequestLine بر روی 512 و یا کمتر تنظیم شود.

نحوه پیاده سازی:

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
```

SCWS-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 می شوند، را مشخص می سازد. لذا توصیه می گردد که دستور LimitRequestFields به 100 و یا کمتر تنظیم شده باشد.

نحوه پیاده سازی:

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

1. دستور LimitRequestFields در پیکربندی Apache را برای داشتن مقدار 100 یا کمتر، اضافه یا اصلاح نمایید. اگر دستور موجود نباشد، به طور پیش فرض به پیکربندی زمان کامپایل بستگی دارد که مقدار پیش فرض 100 می باشد.

```
LimitRequestFields 100
```

SCWS-10-3 : تنظیم دستور **LimitRequestFieldsize** به 1024 یا کمتر

شرح اجمالی:

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

نحوه پیاده سازی:

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

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

SCWS-10-4 : تنظیم دستور **LimitRequestBody به 102400 یا کمتر**

شرح اجمالی:

دستور LimitRequestBody حداکثر اندازه body درخواست HTTP را تنظیم می کند. لذا توصیه می شود که دستور LimitRequestBody بر روی 102400 یا کمتر تنظیم گردد.

نحوه پیاده سازی:

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 در می یابید که این دستور، اندازه ی آپلودهای فایل به وب سرور را محدود می کند.

```
LimitRequestBody 102400
```

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

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

شرح اجمالی:

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

نحوه پیاده سازی:

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، مجدداً راه اندازی نمایید.

```
SELINUX=enforcing
```

اگر حالت جاری ، enforcing نیست و راه اندازی مجدد به صورت فوری، امکان پذیر نبود، می توان حالت جاری را با دستور setenacle به enforcing تنظیم نمود.

```
# setenforce 1
```

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

شرح اجمالی:

SELinux شامل سیاست های هدفمند قابل تنظیمی می باشد 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 دایرکتوری ها، فایل ها و پورت های شبکه داشته باشد. دسترسی توسط انواع فرآیندهای (دامنه ها) تعریف شده برای فرآیندهای httpd کنترل می شود. بیش از صد httpd تکی مربوط به انواع تعریف شده در یک سیاست

پیش فرض SELinux آپاچی وجود دارد که شامل بسیاری از افزونه ها و برنامه های متداول آپاچی مانند nagios، php، smokeping و نمونه های دیگر می باشد. سیاست های پیش فرض SELinux برای نصب یک آپاچی پیش فرض به خوبی عمل می کنند، اما پیاده سازی سیاست های 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 عملیات ها و الزامات دقیق برنامه وب را در بر می گیرد. تمام دایرکتوری ها و فایل ها برای این که توسط فرآیند وب سرور قابل دسترس باشند، باید برچسب های امنیتی مناسب داشته باشند.

در زیر نمونه هایی از پر استفاده ترین ها ارائه شده است:

1. http_port_t – پورت های شبکه مجاز برای شنود
2. httpd_sys_content_t – دسترسی خواندن به دایرکتوری ها و فایل ها با محتوای وب
3. httpd_log_t – دایرکتوری ها و فایل هایی که باید برای لاگ ها قابل نوشتن مورد استفاده قرار گیرند.
4. httpd_sys_script_exec_t – دایرکتوری ها و فایل هایی برای محتوای قابل اجرا.

نحوه پیاده سازی:

اگر فرآیندهای httpd در حال اجرا به محتوای SELinux httpd-t 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 پلتفرم ها مانند Ubuntu ، به جای httpd، فایل اجرایی apache2، Apache نامیده می شود.

```
# 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 فایل های اجرایی به درستی برچسب گذاری نشده باشند، ممکن است با دستور chcon دوباره برچسب گذاری شوند، با این حال، برچسب گذاری فایل سیستم، مبتنی بر سیاست های محتوایی فایل SELinux است و سیستم فایل ها در برخی موقعیت ها با توجه به این سیاست، مجددا برچسب گذاری می شوند.

```
# 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 برچسب گذاری شود، بهتر است سیاست SELinux با گزینه ی "-l" semanage fcontext بررسی شود. اگر این سیاست موجود نباشد، آنگاه الگو با استفاده از گزینه ی "-a" اضافه می شود. دستور restorecon که در زیر نشان داده شده، برچسب محتوای فایل را با توجه به سیاست جاری، بازیابی می کند و زمانی مورد نیاز می باشد 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
```

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

شرح اجمالی:

علاوه بر تنظیم کل پیکربندی SELinux در حالت مجاز، می توان انواع Process های مجزا (Domains) مانند httpd_t را در حالت مجاز نیز تنظیم نمود. حالت مجاز هیچ دسترسی و یا عملکردی را محدود نمی نماید، در عوض هر اقدامی که منع شده باشد را تنها Log می نماید.

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

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

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

SCWS-11-4: اطمینان از فعال بودن بولین های ضروری SELinux

شرح اجمالی:

بولین های SELinux اجازه رفتار مختص وب سرور آپاچی را یا می دهند یا نمی دهند. نمونه های 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 پیاده سازی:

به منظور غیرفعال نمودن بولین های httpd که غیرضروری شناخته شده اند، از دستور setsebool به صورت زیر استفاده گردد که با استفاده از option " p– " اعمال تغییرات به صورت دائمی صورت می پذیرد.

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

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

SCWS-12 -1 : فعال نمودن AppArmor Framework

شرح اجمالی:

AppArmor یک ماژول امنیتی هسته لینوکس است که کنترل دسترسی اجباری مبتنی بر نام، به همراه سیاست های امنیتی ارائه می دهد. AppArmor قوانین مربوط به برنامه ها را به منظور دسترسی فایل، اتصالات شبکه و محدود کردن اقدامات مبتنی بر سیاست های تعریف شده اجرا می کند.

نحوه پیاده سازی:

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 می باشد. به عنوان مثال:

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

- مطابق با شرح زیر به منظور فعالسازی فریم ورک AppArmor می بایست اسکریپت init. d اجرا گردد.

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

SCWS-12-2: سفارشی نمودن مشخصات AppArmor آپاچی

شرح اجمالی:

AppArmor شامل پروفایل های قابل تنظیمی می باشد که برای محدود کردن وب سرور آپاچی به اجرای حداقل امتیازات مورد استفاده قرار می گیرد به طوری که سرور تنها حداقل دسترسی را به دایرکتوری ها، فایل ها و پورت های شبکه داشته باشد. دسترسی به وسیله پروفایلی که برای process مخصوص apache2 تعریف شده کنترل می گردد. پروفایل پیش فرض AppArmor معمولاً یک پروفایل مجاز است که اجازه دسترسی خواندن-نوشتن را به تمام سیستم فایل ها می دهد. بنابراین توجه به این نکته مهم است که پروفایل پیش فرض به منظور به اجرا درآوردن حداقل امتیازات، سفارشی شود. از ابزارهای AppArmor مانند aacomplain، aa-autodep، و aa-logprof را می توان جهت ایجاد یک پروفایل اولیه براساس استفاده واقعی بهره برد. با این حال، تست کامل، بررسی و سفارشی سازی برای اطمینان از این که محدودیت های پروفایل آپاچی قابلیت های لازم را در حین پیاده سازی حداقل امتیازات میسر می سازند، ضروری خواهد بود.

نحوه پیاده سازی:

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 وابستگی های 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
```

- برنامه ی تحت وب را با بررسی همه ی قابلیت ها به صورت کامل تست نمایید، به طوری که AppArmor، همه ی لاگ ها ضروری از منابع در دسترس را تولید کند. لاگ ها از طریق ابزار syslog فرستاده شده و معمولاً در یکی از فایل های /var/log/syslog یا /var/log/messages یافت می شوند. همچنین لازم بذکر است، متوقف کردن و راه اندازی مجدد وب سرور نیز، بخشی از فرآیند تست می باشد.
- از aa-logprof برای بروزرسانی پروفایل بر اساس لاگ ها تولید شده در طول تست، استفاده نمایید. این ابزار برای اصلاحات پیشنهادی بر روی پروفایل، بر اساس لاگ ها به کار گرفته می شود. همچنین، ممکن است لاگ ها به منظور بروزرسانی پروفایل، به صورت دستی مجدداً مورد بازنگری قرار داده شوند.

```
# aa-logprof
```

- با حذف محتوای نامناسب و اضافه کردن قوانین دسترسی مناسب، پروفایل را بررسی و ویرایش کنید.

دایرکتوری های با فایل های در دسترس و با مجوزهای مشابه می توانند با استفاده از wild-cardها ساده شوند. همچنین پروفایل بروزرسانی شده 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 لاگ ها رد شده ی apparmor تست نمایید. پروفایل را در صورت لزوم، بروزرسانی کرده و مجدداً بارگذاری نمایید. همچنین تست های برنامه را تا زمانی که هیچ لاگ رد شده ی 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
```

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

شرح اجمالی:

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

نحوه پیاده سازی:

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 راه اندازی مجدد سرور باشیم.

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

- سرویس آپاچی را مجدد راه اندازی نمایید.

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

13. نکات امنیتی Apache Http Server 2.4.x

برخی از اشـارات و نکات مربوط به مسـائل امنیتی در راه اندازی یک سـرور وب، اشـاره شـده اسـت. برخی از پیشنهادات کلی هستند، برخی دیگر مختص سرور آپاچی هستند.

13.1 بروز نگه داشتن آپاچی (Keep up to Date)

HTTP سرور آپاچی رکورد های خوبی برای مسائل امنیتی دارد و دارای یک جامعه توسعه دهنده که در مورد مسـائل امنیتی توجه ویژه ای دارند. اما برخی از مشـکلات اجتناب ناپذیر اسـت، کوچک یا بزرگ بودن آن بعد از انتشـار نرم افزار کشـف خواهند شـد. به همین دلیل، حفظ آگاهی ها از به روز رسـانی ها (updates) های داده شده به نرم افزار ها بسیار مهم است. اگر نسخه خود را از HTTP سرور آپاچی به صورت مستقیم به دست آورده اید، به شـدت توصـیه می کنیم که در لیسـت اعلان های (Announcements List) HTTP سـرور آپاچی اطلاعات خود را با دیگران به اشتراک بگذارید تا بتوانید از اطلاعات جدید و به روزسانی های جدید امنیتی مطلع باشید. خدمات مشابه از سوی عامل های سوم، توزیع کنندگان نرم افزار آپاچی در دسترس است.

البته، اغلب اوقات زمانی که یک سـرور وب به خطر می افتد، علت آن به دلیل مشـکل موجود در کد HTTP سرور آن نیست. در عوض، این مسئله ناشی از مشکلات مربوط به اضافه کردن کد، اسکریپت های CGI یا سیستم عامل می باشد. بنابراین مهم است که شما باید از مشکلات و بروز رسانی ها از سیستم نرم افزاری خود آگاه باشید.

13.2 حملات منع سرویس (DOS)

تمام سرور های شبکه می توانند در معرض حملات منع سرویس قرار گیرند که تلاش می کنند با گره زدن منابع سـرور از پاسـخ به کلاینت ها جلوگیری کنند. جلوگیری از چنین حملاتی ممکن نیسـت، اما شـما می توانید کارهایی را انجام دهید تا مشکلاتی که ایجاد شده را کاهش دهید.

اغلب موثرترین ابزار ضد حملات DOS، یک فایروال یا تنظیمات سیستم های عامل هستند. برای مثال، بیشتر فایروال ها می توانند برای محدود کردن تعداد اتصالات همزمان از هر آدرس IP مجزا یا شبکه ای شکل بگیرند

در نتیجه از گستره ای از حملات ساده جلوگیری می کنند. البته این مورد هیچ کمکی علیه حملات منع سرویس توزیع شده (DDOS) نمی کند.

همچنین تنظیمات پیکربندی Apache Server HTTP نیز وجود دارد که می تواند به کاهش مشکلات کمک کند:

- دستورالعمل های RequestReadTimeout اجازه می دهد زمان ارسال درخواست کلاینت محدود شود.
- TimeOut و دستورات مربوط به آن بایستی در سایت هایی که مربوط به حملات DoS هستند، کاهش یابند. تنظیم این مسئله تا زمانی که در حد چندثانیه کوتاه باشد، مناسب است. همانطور که TimeOut در حال حاضر برای چندین عملیات مختلف استفاده می شود، تنظیم آن به یک مقدار پایین، مشکلاتی را با استفاده از اسکریپت های CGI در طول اجرا بلند مدت ایجاد می کند.
- دستورالعمل KeepAliveTimeout در سایت هایی که مستعد به حملات DoS هستند، خطرات را کاهش می دهد. برخی از سایت ها به طور کامل keepalives را با viaKeepAlive خاموش می کنند که البته اشکالاتی را در عملکردشان ایجاد می کنند.
- مقادیر دستورالعمل های مربوط به مدت زمان مختلف ارائه شده توسط ماژول های دیگر بایستی بررسی شوند.
- دستورالعمل های LimitRequestBody, LimitRequestFields, LimitRequestFieldSize, LimitRequestLine و LimitXMLRequestBody بایستی به دقت بررسی شوند تا مصرف منابع ایجاد شده توسط درخواست کلاینت را محدود کنند.
- سیستم عامل هایی که از آن ها پشتیبانی می کنند مطمئن شوید که از دستورالعمل های AcceptFilter برای انتقال بخشی از پردازش درخواست به سیستم عامل استفاده می کنید. این مورد به طور پیش فرض در Apache httpd فعال است، اما ممکن است نیازمند پیکربندی(تنظیمات) مجدد kernel (هسته) شما باشد.
- تنظیم دستورالعمل MaxRequestWorkers برای اجازه دهی به سرور برای مدیریت به حداکثر تعداد اتصالات همزمان بدون استفاده از منابع. همچنین برای تنظیم کارآیی آنها به اسناد مربوطه شان مراجعه کنید.
- استفاده از mpm threaded ممکن است به شما این امکان را دهد که ارتباطات همزمان بیشتری را کنترل کنید در نتیجه حملات DoS را کاهش دهید. علاوه بر این، رویداد mpm از پردازش غیرهمزمان برای اجتناب از اختصاص یک رشته (thread) به هر اتصال استفاده می کند. به دلیل ماهیت کتابخانه

OpenSSL، رویداد mpm درحال حاضـــر با mod_ssl و دیگر فیلترهای ورودی ناســـازگار اســـت. در این موارد در دسته رفتار worker mpm قرار می گیرید.

- تعدادی از ماژول های عامل سوم وجود دارد که از طریق http://modules.apache.org/ موجود ا ست و می تواند رفتارهای خاص کلاینت را محدود کند و در نتیجه مشکلات و مسائل DoS را کاهش دهد.

13.3 مجوز های مربوط به دایرکتوری های ServerRoot (Permissions on ServerRoot Directories)

در یک عملیات معمول، Apache توسط کاربر root راه اندازی(start) می شود، و به کاربر تعریف شده توسط User directive برای serve hits (دادن خدمت) تغییر داده می شـــود. همانطور که در مورد هر فرمان که توسط root اجرا می شود، بایستی مراقبت باشید که از تغییرات توسط کاربران غیر root محافظت شود. نه تنها فایل ها بایســتی با دســترســی root نوشــته شــوند بلکه بایســتی دایرکتوری ها و همه دایرکتوری های والد نیز توسط دستور root نوشته شوند.

برای مثال، اگر می خواهید ServerRoot را در مکان usr/local/apache انتخاب کنید پیشنهاد می شود که دایرکتوری را به عنوان root ایجاد کنید با عبارت هایی مانند موارد زیر این کار انجام می شود:

```
mkdir /usr/local/apache
cd /usr/local/apache
mkdir bin conf logs
chown 0 . bin conf logs
chgrp 0 . bin conf logs
chmod 755 . bin conf logs
```

فرض بر این است که usr/ ,/ و usr/local/ محل هایی هستند که تنها با دسترسی root قابل تغییر هستند. زمانی که فایل قابل اجرای httpd را نصب می کنید بایستی اطمینان حاصل کنید که به طور مشابه محافظت می شوند:

```
cp httpd /usr/local/apache/bin
chown 0 /usr/local/apache/bin/httpd
chgrp 0 /usr/local/apache/bin/httpd
chmod 511 /usr/local/apache/bin/httpd
```

شـــما می توانید یک زیر دایرکتوری htdocs قابل تغییر به کاربران دیگر را ایجاد کنید – چون کاربر root هیچ فایلی خارج از آن را نمی تواند اجرا کند، و نباید در آنجا پرونده هایی ایجاد کند.

اگر به کاربران غیر root اجازه دهید هر فایلی را اصلاح کنند در صورتی که فقط کاربر root می تواند آن را اجرا یا بنویسد بنابراین شما سیستم خود را نسبت به خطرات مجوزهای دسترسی root ، آسیب پذیر می کنید. برای مثال، یک نفر می تواند باینری httpd را جایگزین آن کند به طوری که در دفعات بعدی زمانی که آن را شـــروع کردید، یک کد مخرب دلخواه فرد اجرا شـود. اگر دایرکتوری لاگ فایل قابل نوشـتن باشـد (توسـط کاربری غیر root)، ممکن ا ست ک سی فایل لاگی را با یک فایل symlink به یک فایل سی ستمی دیگری جایگزین کند و سپس ممکن ا ست root آن فایل ایجاد شده را با داده های خود بازنوی سی کند. اگر فایل های لاگ خود شان قابل نوشتن باشند (توسط کاربر غیر root)، پس ممکن است یک نفر بتوانند فایل لاگ را با داده های ساختگی (bogus) پر کند.

13.3.1 Include های سمت سروری (Server Side Includes)

Include های سمت سرور (SSI) مدیر سرور را با چندین خطر بالقوه امنیتی روبرو می کند.

اولین خطر load شدن افزایشی روی سرور است. تمام فایل ها با قابلیت فعال شدن SSI بایستی توسط آپاچی تجزیه (parsed) شوند اگرچه دستورالعمل های SSI وجود داشته باشند که در فایل ها، Include شده اند. در حالی این افزایش بار(load) جزئی باشـد که این موارد در محیط اشـتراک گذاشـته شـده می توانند قابل توجه باشند.

فایل های SSI نیز همان خطراتی را نشان می دهند که با اسکریپت های CGI در حالت کلی مرتبط هستند. با اسـتفاده از عنصـر cmd exec، فایل های قابل فعال سـازی SSI می توانند هر گونه اسـکریپت CGI یا برنامه کامپیوتری ت حت کنترل کاربر و گروه Apache را اجرا ک ند، ه مانطور که در httpd.conf پیکرب ندی (configured) شده است.

راه هایی برای افزایش امنیت پرونده های SSI وجود دارد در حالی که هنوز از مزایایی که ارائه می دهند بهره می بریم.

برای جدا کردن (ایزوله کردن) یک آ سیب، فایل SSI می تواند عامل آن باشد، مدیر سرور می تواند suexec را به گونه ای که در بخش عمومی CGI توضیح داده شده است، فعال کند.

فعال کردن SSI برای فایل ها با پسوند html. یا htm. می تواند خطرناک باشد. این امر به خصوص در یک محیط اشتراکی و یا ترافیک بالای محیط سرور صدق می کند. فایل هایی با قابلیت فعال سازی SSI هم بایستی یک پسوند جداگانه داشته باشند، مانند پسوند های مرسوم shtml. این کار به نگه داشتن load مناسب سرور در حداقل و امکان مدیریت آسان ریسک کمک می کند.

راه حل دیگر، غیرفعال کردن اجرای اسکریپت ها و برنامه ها از صفحات SSI است. برای انجام این کار، Includes را با IncludesNOEXEC در گزینه های دستورالعمل جایگزین کنید. توجه داشته باشید که کاربران هنوز هم ممکن است از <-- "..."=include virtual #--> برای اجرای اسکریپت های CGI استفاده کنند در صورتی که این متن در دایرکتوری های تعیین شده توسط دستور العمل ScriptAlias طراحی شده است.

تنظیمات مربوطه در آپاچی نسخه 2.4.x بدین صورت می باشد.

به دایرکتوری $Web_Server/conf بروید. و httpd.conf را باز کنید. دایرکتوری را جستجو می کنید و دستور العمل زیر را اضافه کنید.

```
<Directory /opt/apache/htdocs>
Options-Indexes_Includes
Order allow, denyAllow from all
</Directory>
```

CGI 13.4 به صورت کلی (CGI in General)

در ابتدا، همیشه بایستی به خاطر داشته باشید که به نویسندگان اسکریپت ها/ برنامه های CGI و توانایی آنها در تشخیص حفره های امنیتی بالقوه در کامپیوتر اعتماد داشته باشید، چه آنها موارد را به صورت آگاهانه و یا به صورت تصادفی انجام دهند. اسکریپت های CGI می توانند اساساً دستور دلخواه خود را بر روی سیستم شما با سطح مجوز دسترسی کاربر وب سرور اجرا کنند بنابراین اگر به صورت دقیق بررسی نشده باشند می توانند بسیار خطرناک باشند.

همه ی اسکریپت های CGI در همان کاربر موجود اجرا خواهند شد، بنابراین آنها پتانسیل تصادم (عمدی یا صحفی) را با ا سکریپت های دیگر دارند به عنوان مثال کاربر A از کاربر B تنفر دارد، بنابراین یک ا سکریپت به "پایگاه داده CGI" کاربر B می نویسد و آن را حذف می کند. یک برنامه که می تواند برای اجازه دادن به اسکریپت ها برای اجرا به عنوان کاربران مختلف استفاده شود suEXEC است که با Apache به نام 1.2

استفاده می شود و تله ی خاص (special hooks) در کد سرور آپاچی نامیده می شود. یک روش محبوب دیگر برای انجام این کار استفاده از CGIWrap است.

13.4.1 نام جایگزین (مستعار) غیر اسکریپتی CGI (Non Script Aliased CGI)

اجازه دادن به کاربران برای اجرای اسکریپت های CGI در هر دایرکتوری بایستی تنها در صورتی در نظرگرفته شود که:

- شما به کاربران خود اطمینان دارید که عمدی یا صحفی اسکریپتی ننویسند و سیستم شما را در معرض حمله قرار دهند.
- شما بایستی امنیت سایت خود را در مناطق دیگر هم در نظر بگیرید تا منطقه ای بسیار ضعیف نبا شد، در نتیجه یک حفره بالقوه دیگر را بی اهمیت نشان ندهد.
- شما هیچ کاربری و هیچ کسی را ندارید که از سرور شما بازدید کند.

13.4.2 اسکریپت های جایگزین CGI (Script Aliased CGI)

محدود کردن CGI به دایرکتوری های خاصی، کنترل Admin (مدیر) را در مورد آنچه که در دایرکتوری ها قرار دارد باعث می شود. این امر به طور اجتناب ناپذیر امن تر از اسکریپت های non script aliased CGI است، اما تنها در صورتی کاربرانی که به دایرکتوری ها دسترسی نوشتن داشته باشند، مورد اعتماد خواهند بود و یا مدیر مایل باشد هر برنامه/ اسکریپت جدید CGI را برای حفره های امنیتی بالقوه امتحان کند.

اغلب سایت ها این گزینه ها را بر روی رویکرد non script aliased CGI انتخاب می کنند.

13.5 منابع دیگر محتوای پویا (Other sources of dynamic content)

گزینه های اسکریپت های توکار(جاسازی شده) که به عنوان بخشی از خود سرور اجرا می شود، مانند mod_php, mod_perl, mod_tcl وmod_python تحت هویت خود سرور اجرا می شوند (دستورالعمل های کاربر را ببینید)، و بنابراین اسکریپت های اجرا شده توسط این موتور ها به طور بالقوه می توانند به هر چیزی که کاربر دسترسی دارد دسترسی پیدا کنند. برخی از موتور های اسکریپت ممکن است محدودیت هایی را ایجاد کنند، اما بهتر است ایمن باشند و موردی را از قبل فرض نکنند.

13.5.1 امنیت محتوای پویا (Dynamic content security)

هنگامی که محتوای پویا را تنظیم می کنید، مانند mod_php, mod_perl یا mod_python بسیاری از ملاحظات امنیتی از محدوده خود httpd خارج می شوند و شما بایستی از اسناد خود ماژول ها استفاده کنید. برای مثال، PHP به شما اجازه می دهد تا حالت ایمن (Safe Mode) را راه اندازی کنید، که اغلب به صورت پیش فرض غیرفعال می شود. برای اطلاعات بیشتر در مورد آن، هر سند پروژه را مطالعه کنید.

در سطح آپاچی، یک ماژول به نام mod_security را می توان به عنوان یک فایروال HTTP مشاهده کرد و اگر به خوبی آن را پیکربندی (configure) کنید، می تواند به شما کمک کند تا امنیت محتوای پویای خود را افزایش دهید.

13.6 محافظت از تنظیمات سیستم (Protecting System Settings)

برای هدایت یک کشتی در محدوده باریک، شما بایستی کاربران را از تنظیمات فایل **.htaccess** باز دارید که با انجام این کار می توانند مشخصه های امنیتی که از قبل پیکربندی نموده اید، را باز نویسی کنند. فقط یک راه حل برای انجام این کار وجود دارد.

در فایل پیکربندی سرور قرار دهید:

```
<Directory "/">
   AllowOverride None
</Directory>
```

این کار مانع از استفاده از فایل **.htaccess** در تمام دایرکتوری ها جدا از آنهایی که فعال سازی شده اند، می شود.

توجه داشته باشید این تنظیمات به صورت پیش فرض در آپاچی 2.39 موجود هستند.

13.7 محافظت فایل های پیش فرض سرور (Protect Server Files by Default)

یک جنبه از آپاچی که گهگاهی به اشتباه فهمیده می شود، ویژگی دسترسی پیش فرض است. یعنی اگر شما برای تغییر آن اقدام لازمی را انجام ندهید، سرور بتواند راه خود را به پرونده از طریق نگاشت قوانین URL معمولی پیدا کند، بنابراین می تواند به تقاضای کلاینتان خدمت دهد.

مثال زیر را در نظر بگیرید:

```
# cd /; ln -s / public_html
Accessing http://localhost/~root/
```

این مورد به کلاینت ها اجازه می دهد تا کل فایل سیستم (filesystem) را طی کنند. برای کار در این مورد، بلوک زیر را به پیکربندی سرور اضافه کنید:

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

این مورد باعث جلوگیری از د ستر سی پیش فرض به موقعیت های فایل سی ستم (filesystem) خواهد شد. بلوک های راهنمای مناسبی را اضافه کنید تا تنها به مناطقی که می خواهید دسترسی داشته باشید. برای مثال:

```
<Directory "/usr/users/*/public_html">
    Require all granted
</Directory>
<Directory "/usr/local/httpd">
    Require all granted
</Directory>
```

توجه ویژه ای به تعاملات موقعیت و د ستورالعمل های دایرکتوری توجه دا شته با شید، به عنوان مثال، حتی اگر <Directory "/"> دسترسی را منع می کند، یک دستور العمل <Location "/"> ممکن است آن را تغییر دهد.

همچنین نسبت به بازی کردن با دستورالعمل UserDir محتاط باشید، و آن را به چیزی به این مورد /. تنظیم کنید که همان تأثیر قبلی را خواهد داشت. برای root، به عنوان مثال در بالا ما قویاً توصیه می کنیم که خط زیر را در فایل های پیکره بندی سرور خود بگنجانید:

```
UserDir disabled root
```

13.8 به لیست log ها هم توجه کنید (Watching Your Logs)

برای به روز نگه داشتن آنچه که در سرور علیه شما در حال انجام است بایستی فایل های لاگ را بررسی کنید. حتی با وجود اینکه فایل های لاگ فقط گزارش می دهند که چه اتفاقاتی واقعاً رخ داده است، آنها به شما درکی از آنچه که حملات سرور انجام شده است را خواهند داد و اجازه بررسی می دهند که آیا سطح لازم از امنیت وجود دارد یا نه.

یک جفت مثال:

```
grep -c "/jsp/source.jsp?/jsp/ /jsp/source.jsp??" access_log
grep "client denied" error_log | tail -n 10
```

اولین مثال، تعداد حملاتی را که برای بهره برداری از آپاچی Tomcat برای تقاضـا های مخرب Source.JSP از آسـیب پذیری افشـای اطلاعات را لیسـت می کند. مثال دوم، ده کلاینت آخر منع شـده را می دهد به عنوان مثال:

```
[Thu Jul 11 17:18:39 2002] [error] [client foo.example.com] client denied by server configuration: /usr/local/apache/htdocs/.htpasswd
```

همانطور که می توانید ببینید، فایل های لاگ تنها گزارشی می دهند که چه اتفاقاتی رخ داده است، بنابراین اگر کلاینت بتواند به فایل htpasswd. دسترسی داشته باشد. ممکن است چیزی شبیه به این مورد اتفاق بیفتد:

```
foo.example.com - - [12/Jul/2002:01:59:13 +0200] "GET /.htpasswd HTTP/1.1"
```

در log دسترسی شما. این مورد بدان معنی است که شما احتمالاً موارد زیر را در پرونده پیکربندی کارگزار خود بیان کرده اید:

```
<Files ".ht*">
    Require all denied
</Files>
```

13.9 ادغام بخش های پیکربندی (Merging of configuration sections)

ادغام بخش های پیکربندی پیچیده و گاهی اوقات دستور خاصی لازم دارد. همیشه هنگام ایجاد وابستگی های مربوط به نحوه ادغام دستورها، تغییرات خود را تست کنید.

برای ماژول هایی که هیچ منطق ادغام مانند mod_access_compat را اجرا نمی کند رفتار در بخش های بعدی بستگی به این دارد که بخش بعدی هر دستورالعمل از ماژول را داشته باشد. این پیکربندی تا زمانی به ارث می رسد که یک تغییر ایجاد شود، که در آن نقطه پیکربندی جایگزین و ادغام نشده باشد.

14. امنیت نرم افزار تحت وب

وب سرور آپاچی و تنظیمات غلط آن یا به درستی محکم نشده اند و می توانند از نرم افزار تحت وب بهره کشی کنند. محکم کردن پیکربندی سرور دشواری خاصی داشته باشد.

14.1 کوکی ها

14.1.1 غیرفعال کردن درخواست Trace HTTP :

به طور پیش فرض متد Trace در وب سرور های آپاچی فعال است. فعال بودن این مورد می تواند حمله ی Cross Site Tracing را به دنبال داشته باشد و به مهاجم فرصت دزیده شدن اطلاعات کوکی شما را بدهد. در هنگام telnet به درخواست Trace به سروری روی پورت مشخص ، آن سرور پاسخ درخواست را می دهد برای غیرفعال کردن آن بایستی اقدامات زیر را انجام دهیم:

به دایرکتوری $Web_Server/conf بروید و دستورالعمل زیر را وارد کرده و در httpd.conf ذخیره کنید.

```
Restart apache
```

14.1.2 کوکی را با مقدار HttpOnly و پرچم ایمن مقدار دهی کنید

شما می توانید اغلب حملات cross site scripting را با استفاده از HttpOnly و مسدود کردن پرچم ایمن در کوکی، کاهش دهید. بدون داشتن HttpOnly و پرچم ایمن، امکان سرقت یا دستکاری نشست ها در نرم افزار کاربردی وبی و کوکی آنها وجود دارد و این مورد خطرناک است.

پیاده سازی:

اطمینان حاصل کنید mod_headers.so در تنظیمات httpd.conf فعال است. به مسیر دایرکتوری $Web_Server/conf بروید و در httpd.conf دستورات زیر را ذخیره کنید.

```
Header edit Set-Cookie ^(.*)$$1; HttpOnly; Secure
```

14.2 حمله clickjacking

Clickjacking آسیب پذیری شناخته شده ای در نرم افزار وبی است.

پیاده سازی:

اطمینان حاصل کنید که mod_headers.son در httpd.conf فعال است. به دایرکتوری $web_server / conf بروید و در httpd.conf دستورات زیر را ذخیره کنید.

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

14.3 حفاظت XSS

حفاظت حمله ی xss می تواند در بسیاری از مرورگرها دور زده شوند. شما می توانید این حفاظ را برای نرم افزاری وبی خود به کار ببرید اگر این مورد توسط توسط کاربر غیرفعال شده باشد.

پیاده سازی:

اطمینان حاصل کنید که mod_headers.so در httpd.conf فعال است. به دایرکتوری $web_server conf/ بروید و در httpd.conf دستورات زیر را ذخیره کنید.

```
Header set X-SS-Protection “1; mode=block”
```

14.4 پروتکل HTTP 1.0 را غیر فعال کنید.

زمانی در مورد امنیت صحبت می کنیم، بایستی بیش از بیش حفاظت را رعایت کنیم. چرا بایستی ما از نسخه قدیمی HTTP استفاده کنیم، بنابراین بایستی آن را غیرفعال کنیم.

HTTP 1.0 ضعف امنیتی نسبت به ربایش نشست ها دارد. با استفاده از ماژول mod_rewrite می توانیم آن را غیرفعال کنیم.

پیاده سازی:

اطمینان حاصل کنیم که ماژول mod_rewrite در فایل httpd.conf لود شده است.

دستورالعمل RewriteEngine را فعال کنید در مثال زیر و شرط Rewrite را برای اجازه به only HTTP 1.1 اضافه کنید.

```
RewriteEngine On
RewriteCond        %{THE_REQUEST}
!HTTP/1.1$
RewriteRule .*-[F]
```

پیوست

در این بخش چک لیستی به منظور ممیزی محصول مورد نظر ارائه شده است. این چک لیست شامل سه جدول است. جدول اول، جدول ممیز می باشد. در این جدول، اطلاعات مربوط به شخصی که پیکربندی امن را انجام می دهد یا آن را ممیزی می کند، وارد می شود. همچنین نتایج پیکربندی یا ممیزی به صورت اختصار در این جدول

درج می گردد. جدول دوم، محل وارد کردن مشخصات سروری است که Apache HTTP Server 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 خلاصه ای از تمامی الزامات بیان شده در متن سند می باشد. قابل ذکر است که ستون های "وضعیت" و "قابلیت پیاده سازی" باید توسط ممیز و برای هر سیستم حاوی این برنامه تکمیل گردد. در ستون وضعیت، ممیز باید از عبارت های "قبول" و "رد" متناسب با وضعیت الزام در محصول مورد ارزیابی استفاده نماید. در ستون قابلیت پیاده سازی، ممیز باید قابلیت پیاده سازی الزام برای محصول مورد ارزیابی را با عبارات "دارد" و "ندارد" بیان نماید. در صورتی که الزامی برای محصول مذکور قابلیت پیاده سازی نداشته باشد، علت عدم قابلیت پیاده سازی آن باید در ذیل جدول توضیح داده شود.

شناسه	وضعیت	تنظیمات	قابلیت پیاده سازی تنظیمات	مقدار پیش فرض	مقدار مطلوب
SCWS 1		**برنامه ریزی و راه اندازی**			
SCWS-1-1		تهیه چک لیست برنامه ریزی پیش از راه اندازی	دارد	ندارد	تهیه چک لیست و اعمال آن

SCWS-1-2		عدم نصب سرویس های مختلف بر روی یک سرور	دارد	فعال	غیرفعال
SCWS 1-3		نصب Apache	دارد	ندارد	استفاده از معیار منتشر شده جهت نصب
SCWS 2		**به حداقل رساندن ماژول های Apache**			
SCWS 2-1		فعال نمودن ماژول های احراز هویت و اعطای مجوز ضروری	دارد	ندارد	فعال
SCWS 2-2		فعال نمودن ماژول پیکربندی Log	دارد	ندارد	فعال
SCWS 2-3		غیرفعال نمودن ماژول WebDAV	دارد	ندارد	غیرفعال
SCWS 2-4		غیرفعال نمودن ماژول Status	دارد	ندارد	غیرفعال
SCWS 2-5		غیرفعال نمودن ماژول Autoindex	دارد	ندارد	غیرفعال
SCWS 2-6		غیرفعال نمودن ماژول Proxy	دارد	ندارد	غیرفعال
SCWS 2-7		غیرفعال نمودن ماژول های دایرکتوری های کاربر	دارد	ندارد	غیرفعال
SCWS 2-8		غیرفعال نمودن ماژول info	دارد	ندارد	غیرفعال
SCWS 3		**اصول، مجوزها و مالکیت**			

SCWS 3-1		اجرای وب سرور Apache با کاربری غیر root	دارد	ندارد	اجرا با کاربری غیر root و Groupو User پیکربندی
SCWS 3-2		درنظر گرفتن پوسته نامعتبر برای حساب کاربری آپاچی	دارد	ندارد	در نظر گرفتن پوسته نامعتبر یاnologin به منظور ورود
SCWS 3-3		قفل نمودن حساب کاربری Apache	دارد	ندارد	استفاده از دستور passwd به منظور غیرفعال نمودن حساب کاربری
SCWS 3-4		تنظیم مالکیت بر روی دایرکتوری ها و فایل های Apache	دارد	به صورت پیش فرض مالکیت از آن کاربری می باشد که نرم افزار را ایجاد کرده و کاربر root	مطابق نحوه پیاده سازی اجرا گردد.
SCWS 3-5		تنظیم شناسه گروه بر روی دایرکتوری ها و فایل های Apache	دارد	به صورت پیش فرض مالکیت از آن کاربری می باشد که نرم افزار را ایجاد کرده و کاربر root	مطابق نحوه پیاده سازی اجرا گردد.
SCWS 3-6		محدود کردن دیگر دسترسی های Write بر روی	دارد	ندارد	مطابق نحوه پیاده سازی اجرا گردد.

		دایرکتوری ها و فایل های Apache			
SCWS 3-7		امن نمودن دایرکتوری Core Dump	دارد	ندارد	تنظیم دستور CoreDumpDirectory برای دایرکتوری های متعلق به کاربر ریشه
SCWS 3-8		امن Lock File نمودن	دارد	ندارد	مطابق نحوه پیاده سازی اجرا گردد.
SCWS 3-9		امن نمودن Pid فایل	دارد	ندارد	مطابق نحوه پیاده سازی اجرا گردد.
SCWS 3-10		امن نمودن فایل ScoreBoard	دارد	ندارد	مطابق نحوه پیاده سازی اجرا گردد.
SCWS 3-11		محدود کردن دسترسی های Write گروه بر روی دایرکتوری ها و فایل های Apache	دارد	ندارد	مطابق نحوه پیاده سازی اجرا گردد.
SCWS 3-12		محدود نمودن دسترسی Write گروه بر روی ریشه اصلی پوشه و فایل ها	دارد	ندارد	مطابق نحوه پیاده سازی اجرا گردد.

SCWS 4		**کنترل دسترسی آپاچی**			

SCWS 4-1		منع دسترسی به دایرکتوری ریشه OS	دارد	ندارد	مسدود کردن دسترسی به دایرکتوری ریشه سیستم عامل
SCWS 4-2		اجازه دسترسی مناسب به محتوای وب	دارد	مطابق بند 1 از توضیحات پیش فرض جدول	استفاده از از دستورات Allow یا Require
SCWS 4-3		محدودسازی نادیده گرفتن برای دایرکتوری ریشه OS	دارد	<Directory /> ... AllowOverride None ... </Directory>	تنظیم مقدار AllowOverride بر روی none
SCWS 4-4		محدودسازی نادیده گرفتن OverRide برای کلیه دایرکتوری ها	دارد	ندارد	تنظیم مقدار همه دستورهای AllowOverride بر روی none
SCWS 5		**به حداقل رساندن قابلیت ها، محتوا و گزینه ها**			
SCWS 5-1		محدود نمودن Optionها برای دایرکتوری ریشه OS	دارد	ندارد	تنظیم مقدار Option بر روی none
SCWS 5-2		محدود نمودن Optionها	دارد	ندارد	تنظیم مقدار همه دستورهای Option و بر روی none یا Multiviews

		برای دایرکتوری ریشه وب			
SCWS 5-3		به حداقل رساندن Optionها برای دیگر دایرکتوری ها	دارد	ندارد	تنظیمات همانند بند -1- SCR 5-2
SCWS 5-4		حذف نمودن محتوای پیش فرض HTML	دارد	ندارد	مطابق نحوه پیاده سازی اجرا گردد.
SCWS 5-5		CGI Content حذف پیش فرض Printenv	دارد	ندارد	از دایرکتوری printenv حذفcgi-bin
SCWS 5-6		CGI Content حذف test-پیش فرضcgi	دارد	ندارد	حذف test-cgi از دایرکتوری cgi-bin
SCWS 5-7		محدود نمودن متدهای HTTP Request	دارد	بدون محدودیت	مطابق نحوه پیاده سازی اجرا گردد.
SCWS 5-8		غیر فعال کردن متد HTTP TRACE	دارد	ندارد	تنظیم TraceEnable با مقدار off در پیکربندی سطح سرور
SCWS 5-9		محدود نمودن نسخه پروتکل HTTP	دارد	ندارد	مطابق نحوه پیاده سازی اجرا گردد.
SCWS 5-10		محدود نمودن دسترسی به فایل های. ht*	دارد	غیرقابل دسترس	مطابق نحوه پیاده سازی اجرا گردد.

SCWS 5-11		محدود نمودن پسوند فایل ها	دارد	بدون محدودیت	مطابق نحوه پیاده سازی اجرا گردد.
SCWS 5-12		فیلترینگ درخواست ها بر اساس آدرس IP	دارد	ندارد	مطابق نحوه پیاده سازی اجرا گردد.
SCWS 5-13		محدود نمودن دستور Listen	دارد	ندارد	مطابق نحوه پیاده سازی اجرا گردد.
SCWS 5-14		محدود نمودن Optionهای فریم مرورگر	دارد	ندارد	مطابق نحوه پیاده سازی اجرا گردد.
SCWS 6		**عملیات ورود، نظارت و نگهداری**			
SCWS 6-1		پیکربندی لاگ خطا	دارد	ندارد	مطابق نحوه پیاده سازی اجرا گردد.
SCWS 6-2		یکربندی Syslog Facility برای دریافت خطاها	دارد	ErrorLog"logs/error_log"	ErrorLog"syslog:local1"
SCWS 6-3		تنظیمات لاگ دسترسی	دارد	ندارد	مطابق نحوه پیاده سازی اجرا گردد.
SCWS 6-4		بازبینی و فضای لاگ	دارد	ندارد	مطابق نحوه پیاده سازی اجرا گردد.
SCWS 6-5		اعمال وصله های قابل اجرا	دارد	ندارد	مطابق نحوه پیاده سازی اجرا گردد.
SCWS 6-6		راه اندازی و فعالسازی ModSecurity	دارد	غیرفعال	راه اندازی و فعالسازی ModSecurity

SCWS 6-7		راه اندازی و فعالسازی مجموعه قوانین اصلی OWASP ModSecurity	دارد	غیرفعال	نصب و پیکربندی OWASP ModSecurity Core Rule Set
SCWS 7		**پیکربندیSSL/TSL**			
SCWS 7-1		نصب و راه اندازی mod_ssl و یا mod_nss	دارد	ندارد	مطابق نحوه پیاده سازی اجرا گردد.
SCWS 7-2		نصب یک گواهی مورد اعتماد و معتبر	دارد	ندارد	مطابق نحوه پیاده سازی اجرا گردد.
SCWS 7-3		حفاظت از کلید خصوصی سرور	دارد	ندارد	مطابق نحوه پیاده سازی اجرا گردد.
SCWS 7-4		غیرفعال کردن پروتکل SSL V3.0	دارد	SSLProtocol all –SSLv2	مطابق نحوه پیاده سازی اجرا گردد.

SCWS 7-5		محدود کردن رمزهای ضعیف SSL	دارد	مطابق بند 2 از توضیحات پیش فرض جدول	مطابق نحوه پیاده سازی اجرا گردد.
SCWS 7-6		غیرفعال نمودن SSL ناامن Renegotiation	دارد	SSLInsecureRenegotiation off	تنظیم SSLInsecurRenegotiation off به on

SCWS 7-7		اطمینان از فعال نبودن فشرده سازی SSL	دارد	در SSLCompression مقدار نسخه 2. 2. 24 تا 2. 2. 25، on و برای نسخه 2. 2. 26، off	به SSLCompression تنظیم off
SCWS 7-8		غیر فعال کردن پروتکل TSLv1. 0	دارد	SSLProtocol all -SSLv2	به SSLProtocol تنظیم TSLv1. 1 یا TLSv1. 2
SCWS 7-9		فعال کردن OCSP Stapling	دارد	SSLUseStapling Off SSLStaplingCache no default value	به SSLUseStapling تنظیم on
SCWS 7-10		فعال کردنHTTP Strict Transport Security	دارد	غیرفعال	پیکربندی دستور Header همانند نمونه ذکر شده در قسمت نحوه پیاده سازی
SCWS 8		**نشت اطلاعات**			
SCWS 8-1		به ServerToken تنظیم ‘Prod’	دارد	ندارد	Full
SCWS 8-2		تنظیم امضاء سرور بر روی Off	دارد	ندارد	off
SCWS 8-3		نشت اطلاعات از طریق محتوای پیش فرض Apache	دارد	ندارد	عدم دسترسی به آیکون های آپاچی

SCWS 9		انکار سرویس Mitigation			
SCWS 9-1		تنظیم Timeout به 10 یا کمتر	دارد	ندارد	10 یا کمتر
SCWS 9-2		تنظیم دستور KeepAlive به on	دارد	ندارد	به KeepAliveتنظیم دستور on
SCWS 9-3		تنظیم MaxKeepAliveRequest به 100 یا بیشتر	دارد	ندارد	100 یا بیشتر
SCWS 9-4		تنظیم به KeepAliveTimeout 15 یا کمتر	دارد	ندارد	15 و یا کمتر
SCWS 9-5		تنظیم محدودیت Timeout برای درخواست Headers	دارد	header =20-40 MinRate=500	40 ثانیه
SCWS 9-6		تنظیم محدودیت Timeout برای درخواست Body	دارد	body=20 MinRate=500	20 ثانیه
SCWS 10		محدودیت های درخواست			
SCWS 10-1		تنظیم دستور به LimitRequestLine 512 یا کمتر	دارد	ندارد	512 و کمتر
SCWS 10-2		تنظیم دستور بهLimitRequestFields 100 یا کمتر	دارد	ندارد	100 و کمتر
SCWS 10-3		تنظیم دستور	دارد	ندارد	1024 و کمتر

		LimitRequestFieldsize به 1024 یا کمتر			
SCWS 10-4		تنظیم دستور به LimitRequestBody 102400 یا کمتر	دارد	ندارد	102400 و کمتر
SCWS 11		**فعال نمودن SELinux به منظور محدود نمودن فرآیندهای Apache**			
SCWS 11-1		فعال نمودن SELinux در حالت Enforcing	دارد	غیرفعال SELinux	Enforcing
SCWS 11-2		اجرای فرآیندهای آپاچی در متن محدود شده httpd_t	دارد	غیرفعال SELinux	مطابق نحوه پیاده سازی اجرا گردد.
SCWS 11-3		اطمینان از عدم قرارگیری مد Permissive بر روی http_t	دارد	عدم وجود در حالت غیرمجاز	غیر فعال نمودن از حالت Permissive
SCWS 11-4		اطمینان از فعال بودن بولین های ضروری SELinux	دارد	غیرفعال SELinux	فعال بودن بولین های ضروری SELinux
SCWS 12		**فعال نمودن AppArmor به منظور محدود نمودن فرآیندهای Apache**			
SCWS 12-1		فعال نمودن AppArmor Framework	دارد	فعال AppArmor	فعال
SCWS 12-2		سفارشی نمودن مشخصات AppArmor آپاچی	دارد	پروفایل آپاچی به صورت پیش فرض غیرسخت گیرانه می باشد.	تغییر پروفایل پیش فرض به حداقل امتیازات
SCWS 12-3		اطمینان از قرارگیری پروفایل در حالت AppArmor Enforce	دارد	enforce	enforce

توضیحات مقادیر پیش فرض جدول:

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

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

2. SCWS-1-7-5: مقادیر پیش فرض به صورت زیر نمایش داده شده است:

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

منابع و مراجع

[1] https://geekflare.com/apache-web-server-hardening-security/

[2] https://www.tecmint.com/apache-security-tips/

[3] https://geekflare.com/10-best-practices-to-secure-and-harden-your-apache-web-server/

[4] https://httpd.apache.org/docs/2.4/misc/security_tips.html

[5] https://www.slashroot.in/how-to-secure-apache-web-server

[6] Apache Http Server مرکز مدیریت راهبردی آفتا