مرکز تخصصی
آپا
دانشگاه صنعتی اصفهان
(آگاهی‌رسانی، پشتیبانی و امداد در حوزه شبکه)

نصب امن کارگزار وب آپاچی

شهریور ۹۲

www.nsec.ir

۱. مقدمه

دستورالعمل ارائه شده در این مقاله برای هر دو شاخه معمول Apache (یعنی 1.x و 2.x) که روی سیستم های Linux نصب می شوند، قابل بکار گیری است. اگر شما یک سیستم یونیکس دیگر را استفاده می کنید، ما این اطمینان را به شما داریم که می دانید که سیستم شما با لینوکس های امروزی چه تفاوت هایی دارد. توصیه های بکار گرفته شده در این فصل برای تنظیمات، درتمام سیستم های غیر یونیکسی (مثل ویندوز) قابل بکار گیری است، البته در مراحل مختلف نصب این تفاوت ها باید مورد توجه قرار گیرد:

> سرور Apache قربانی موفقیت خود شده است. شاخه 1.x آنقدر خوب کار کرد که نیاز به تعویض و بالا رفتن آن حس نشد. برای همین اکثر برنامه نویس ها، به جای اضافه کردن

- ویندوز توانایی
موسوم به chroot را ندارد.
- بعید است شما بتوانید که Apache را بوسیله کد منبع آن ، بر روی ویندوز نصب کنید. به جای آن می توانید که فایل قابل نصب (باینری) آن را از وب سایت Apache دانلود کنید.
- مسیر های دیسک کاملا باهم فرق دارند هرچند که معانی آن ها یکی است.

۲. Source یا Binary

یک تصمیم اولیه این است که آیا سرور را از کد منبع آن Compile کنید، یا از بسته باینری آن استفاده کنید و باید موارد زیر را در نظر بگیرید:

- با کامپایل کردن کد منبع، همه چیز در دست شماست. شما می توانید، اختیارات زمان کامپایل را عوض کنید و ماژول های دلخواه را کم و زیاد کنید. همین طور می توانید کد منبع را عوض کنید. این پروسه زمان زیادی از شما می گیرد.
- نصب و به روز رسانی مثل آب خوردن است، هنگامی که از باینری برای نصب استفاده می کنید(نه کد منبع). از آنجا که بسیاری از فروشنده های سیستم عامل ابزارهای خود را برای به روز رسانی خودکارسیستم عامل در اختیار شما می گذارند. شما مقداری از اختیارات خود را هنگام نصب از دست می دهید، اما در عوض مجبور نیستید که همه چیز را خودتان انجام دهید. البته این بدین معنی است که شما باید منتظر بسته های به روز رسانی بمانید. از طرفی ممکن است که نسخه جدیدتر Apache یا ماژول های مورد علاقه شما تولید نشود و تولید کنندگان فقطبه روز رسانی نسخه قبلی و رفع آسیب های احتمالی آن بپردازند.
- نسخه Apache که شما استفاده می کنید می تواند تصمیمات شما را تحت تاثیر قرار دهد. برای مثال تغییر قابل ملاحظه ای در 1.x نمی افتد، اما در 2.x بهبود های قابل ملاحظه روانه می شود. برخی از سیستم عامل ها به سمت 2.x رفته اند و برخی هنوز ه 1.x وفادار باقی مانده اند.

برای دریافت کد منبع Apach می توانید به آدرس http://httpd.apache.org مراجعه کرده و آخرین نسخه روانه شده به بازار را دریافت کنید.

۳. دانلود کد منبع

در ابتدا باید از تمامیت (Integrity) و صحت کد منبع اطمینان حاصل کنید. به طوری که مشاهده می کنید، اگر چه سایت آپاچی برای دانلود از روی لینک mirror، به سایت های دیگر اشاره می کند، اما تمامی امضاهای آرشیوی (Archive Signature) بر روی خود سایت آپاچی قرار دارند.

یکی از راه های اطمینان از تمامیت یک فایل، استفاده از MD5 sum است. MD5 یک تابع hash است وخروجی آن کاملا شبیه خروجی یک تابع Random است. یعنی امکان آن وجود ندارد که بتوان با داشتن خروجی آن، ورودی تابع را حدس زد. به عبارت دیگر، هر فایل، hash یکتایی دارد که با دانستن آن نمی توان به محتوای فایل پی برد. دو فرمان زیر، اولی مقدار MD5 sum را از فایل دانلود شده، تهیه می نماید و دستور دوم، مقدار checksum را از سایت آپاچی دریافت می کند.

```
$ md5sum httpd-2.0.50.tar.gz
8b251767212aebf41a13128bb70c0b41  httpd-2.0.50.tar.gz

$ wget -O - -q http://www.apache.org/dist/httpd/httpd-2.0.50.tar.gz.md5
8b251767212aebf41a13128bb70c0b41  httpd-2.0.50.tar.gz
```

طبعا اگر حمله گر، بتواند به سایت مبدا نیز دسترسی داشته باشد، میتواند امضاهای آرشیو ها را نیز تغییر دهد و تغییرات را غیر قابل تشخیص بسازد.

راهی مطمئن تر از قبلی این است که از رمز نگاری کلید عمومی استفاده کنیم. در این روش می توان از GnuGP استفاده کرد که بر روی همه سیستم های Unixنصب شده است. ابتدا باید امضای PGP را به روش زیر دانلود کنید:

```
$ wget http://www.apache.org/dist/httpd/httpd-2.0.50.tar.gz.asc
```

تلاش برای verify کردن در این لحظه باعث می شود که GnuPG به شما بگوید که کلید مناسب را برای Verify کردن در دست ندارید:

```
$ gpg httpd-2.0.50.tar.gz.asc
gpg: Signature made Tue 29 Jun 2004 01:14:14 AM BST using DSA key ID DE885DD3
gpg: Can't check signature: public key not found
```

GnuPG یک شناسه یکتا را برمی گرداند(DE885DD3) که می تواند برای گرفتن کلید مناسب از یکی سرورهای کلید استفاده شود.(برای مثال pgpkeys.mit.edu):

```
$ gpg --keyserver pgpkeys.mit.edu --recv-key DE885DD3
gpg: /home/ivanr/.gnupg/trustdb.gpg: trustdb created
gpg: key DE885DD3: public key "Sander Striker <striker@apache.org>" imported
gpg: Total number processed: 1
gpg:               imported: 1
```

تلاش برای چک مردن تمامیت فایل در این مرحله، موفق خواهد بود:

```
$ gpg httpd-2.0.50.tar.gz.asc
gpg: Signature made Tue 29 Jun 2004 01:14:14 AM BST using DSA key ID DE885DD3
gpg: Good signature from "Sander Striker <striker@apache.org>"
gpg:                 aka "Sander Striker <striker@striker.nl>"
gpg:                 aka "Sander Striker <striker@striker.nl>"
gpg:                 aka "Sander Striker <striker@apache.org>"
gpg: checking the trustdb
gpg: no ultimately trusted keys found
Primary key fingerprint: 4C1E ADAD B4EF 5007 579C  919C 6635 B6C0 DE88 5DD3
```

دراین شرایط شما می توانید مطمئن شوید که فایل دانلود شده، درست و واقعی است. در سایت آپاچی و در آدرس http://www.apache.org/dist/httpd/KEYS می توانید تمام کلید های عمومی مورد نیاز را دریافت کنید

۳.۱. دانلود وصله (patch)

معمولا بهترین نسخه آپاچی را نمی توان در آرشیو های موجود پیدا کرد، چرا که آپاچی پیوسته در حال تغییر است. پس بهتر است آپاچی را دانلود کرده و سپس از آدرس http://www.apache.org/dist/httpd/patches/ آن را به روز کنید.

۴. باینری های Static یا ماژول های Dynamic

تصمیم مهم تراین است که آیا یک آپاچی یک پارچه و استاتیک را کامپایل کنید یا این که آپاچی را به صورت ماژول های دینامیک قابل load شدن بسازیم. این دوباره یک tradeoff است برای بدست آوردن امنیت یا از دست دادن وقت.موارد زیر را در نظر بگیرید:

- تحقیقا بدست آمده که نسخه باینری یکپارچه، سریع تر از دیگری است. البته با ارزان تر و سریع تر شدن سرور ها این جنبه در هر دو مورد فرق زیادی نخواهد داشت.
- نسخه باینری یکپارچه فاقد backdoorهای از قبل کامپایل شده است. اضافه کردن یک backdoor به برنامه قابل load در حافظه بسیار راحت است و فقط با تغییر فایل تنظیمات امکان پذیر است. اما اضافه کردن به یم نسخه ایستا نیازمند کامپایل دوباره آن است.
- با وجود یک نیخه ایستا شما مجبور خواهید بود که برای تغییر یک ماژول کوچک، آپاچی را دوباره از نو کامپایل کنید.
- داشتن یک نسخه یکپارچه باعث اتلاف حافظه می شود، درحالی که در نسخه قابل load ، سیستم عامل می تواند بسته به نیاز، هر ماژول را به حافظه بیاورد یا از حافظه خارج کند. همچنین کد مربو به نسخه یکپارچه قابلیت اشتراک گذاشته شدن ندارد. (ولو در سیستم عامل ها یک روند برای کاهش کد هایی که چند کپی از آنها در حافظه وجود دارد بکار می رود که همان forking است.)

۵. محل قرار گیری پوشه ها

در این قسمت ما پوشه های زیر را برای برخی از فایل های خاص فرض خواهیم کرد:

Binaries and supporting files
/usr/local/apache

Public files
/var/www/htdocs *فرض شده است.) Web Server Tree(این مسیر در تمام این کتاب به عنوان*

Private web server or application data
/var/www/data

Publicly accessible CGI scripts
/var/www/cgi-bin

Private binaries executed by the web server
/var/www/bin

Log files
/var/www/logs

محل نصب می تواند با توجه به ذوق شما تغییر کند. اما بایدتوجه داشته باشید که فایل های log همواره در حال بزرگ تر شدن هستند و باید روی پارتیشنی قرار گیرند که جای کافی داشته باشد. مسیر های فرض شده به شرطی بودند که شما یک وب سایت روی کامپیوتر داشته باشید. در صورت داشتن بیش از یک سایت، می توانید مسیر ها را به شکل زیر قرار دهید:

```
/var/www/apachesecurity.net/bin
/var/www/apachesecurity.net/cgi-bin
/var/www/apachesecurity.net/data
/var/www/apachesecurity.net/htdocs
/var/www/apachesecurity.net/logs
```

و مثلا سایت دیگر مانند زیر قرار می گیرد:

```
/var/www/modsecurity.org/bin
/var/www/modsecurity.org/cgi-bin
/var/www/modsecurity.org/data
/var/www/modsecurity.org/htdocs
/var/www/modsecurity.org/logs
```

۶. دستورالعمل نصب

قبل از نصب، آپاچی باید از محیط خود مطمئن شود. این کار بوسیله اسکریپت configure انجام خواهد شد:

```
$ ./configure --prefix=/usr/local/apache
```

این اسکریپت، سیستم عامل شما را بازدید خواهد کرد و یک makefile متناسب با سیستم عامل شما تهیه خواهد کرد. پس از این شما می توانید دستور زیر را برای شروع نصب وارد کنید. البته باید فایل ها را در پوشه ای که در مقابل prefix-- نوشته شده کپی کنید و سپس با دستور apachectl سرور را راه اندازی کنید:

```
$ make
# make install
# /usr/local/apache/bin/apachectl start
```

اگر چه این دستور آپاچی را نصب و به راه می اندازد، اما شما باید سیستم عامل را برای شروع مجدد در هنگام بالا آمدن دوباره سیستم، تنظیم نمایید. این کار در سیستم های مختلف unix با هم کمی فرق دارد، اما عملیات عمده بوجود آوردن یک لینک سمبلیک (symbolic link)به apachectl در Runlevel مربوطه است. (سرورهای معمولا از Runlevel 3 استفاده می کنند.) :

```
# cd /etc/rc3.d
# ln -s /usr/local/apache/bin/apachectl S85httpd
```

۷. تست برای اطمینان از صحت نصب

می توانید از یک مرورگر برای اطمینان از صحت نصب آپاچی استفاده کنید. اگر درست نصب شده باشد، شما می توانید پیغام معروف "Seeing this instead of the website you expected?" را مشاهده کنید. (شکل ۲.۱)

Figure 2-1. Apache post-installation welcome page

در پایین صفحه، لینکی به دفترچه مرجع آپاچی وجود دارد که می توانید آنرا مطالعه کنید و درمورد تنظیمات آن اطلاعات بیشتری پیدا کنید.

با استفاده از ابزار ps شما می توانید مشاهده کنید که چند پروسه مربوط به آپاچی در حال اجرا است:

```
$ ps -Ao user,pid,ppid,cmd | grep httpd
root     31738     1  /usr/local/apache/bin/httpd -k start
httpd    31765 31738  /usr/local/apache/bin/httpd -k start
httpd    31766 31738  /usr/local/apache/bin/httpd -k start
httpd    31767 31738  /usr/local/apache/bin/httpd -k start
httpd    31768 31738  /usr/local/apache/bin/httpd -k start
httpd    31769 31738  /usr/local/apache/bin/httpd -k start
```

بوسیله tail شما می توانید مشاهده کنید که چه چیزی log می شود، هنگامی که یک درخواست وارد سیستم می شود. بوسیله وارد کردن یک فایل غیر حقیقی (که وجود خارجی ندارد) در مرورگر خود می توانید ببینید که در logهای مربوط به دسترسی چه چیزی ثبت می شود. (درمسیر /var/www/logs) در زیر ثبت دو درخواست موجود (۲۰۲) و غیرموجود(۴۰۴) را می بینید:

```
192.168.2.3 - - [21/Jul/2004:17:12:22 +0100] "GET /manual/images/feather.gif
HTTP/1.1" 200 6471
192.168.2.3 - - [21/Jul/2004:17:20:05 +0100] "GET /manual/not-here
HTTP/1.1" 404 311
```

در زیر نیز مقدار error log را برای دو مورد بالا مشاهده می کنید:

```
[Wed Jul 21 17:17:04 2004] [notice] Apache/2.0.50 (Unix) configured
-- resuming normal operations
[Wed Jul 21 17:20:05 2004] [error] [client 192.168.2.3] File does not
exist: /usr/local/apache/manual/not-here
```

۸ انتخاب ماژول های برای نصب

در تئوری وجود دارد که تعداد ماژول های کمتر به معنای تعداد آسیب پذیری های امنیتی کمتر در سیستم است. البته آسیب پذیری بیشتر از قبال پیچیدگی ماژول هاست. احتمال رخنه در ماژول های پیچیده تر مثل mod_ssl و (openSSL در پشت آن)بیشتر است.

استراتژی درست این است که شما ماژول هایی را که واقعا احتیاج دارید را نصب کنید و دیگر هیچ ماژول اضافه ای ا نصب نکنید. شما مطالعه در مورد ماژول می توانید بگویید که کدام به کارتان می آید و دیگران را غیر فعال کنید: برای مطالعه در مورد ماژول ها به مرجع کامل زیر مراجعه کنید:

http://httpd.apache.org/docs-2.0/mod/.

ماژول های زیر از همه خطرناک ترند و شما باید بدانید که آیا واقعا بع آنها احتیاج دارید یا خیر:

- mod_userdir

اجازه می دهد که هر کاربر در پوشه ای شبیه به username~، یک وب سایت داشته باشد(که username همان نام کاربری او است). این ماژول می تواند از طرف نفوذگر ها برای پیدا کردن نام های کاربری استفاده شود. چرا که آپاچی به نام های موجود به گونه ای دیگر جواب می دهد(کدحالت ۴۰۴ را برمی گرداند) و اگر چنین نامی وفضایی ایجاد نشده باشد کد ۴۰۳ بازمی گرداند.

- mod_info

تنظمیات و پیکره بندی آپاچی را به صورت یک صفحه وب بازمی گرداند.

- mod_status

یک سری اطلاعات Real-time در مورد آپاچی بازمیگردانند، البته باز هم به صورت صفحه وب.

- mod_include

یک سری توانایی های ساده اسکریپت نویسی فراهم می آورد که با نام Server-Side Includes (SSI) شناخته می شود. این قبلیت خیلی قدرتمند اما معمولا غیرلازم است.

ازطرفی شما باید ماژول های زیر را نیز در نصب خوداضافه کنید:

- mod_rewrite

این مورد اجازه می دهد که درخواست های ورودی به چیز دیگری باز نویسی شوند. این ماژول را معمولا چاقوی سویسی ماژول های می نامند. (“Swiss Army knife”).

- mod_headers

اجازه می دهد که هدر درخواست و پاسخ دستکاری شود.

- mod_setenvif

اجازه میدهد که متغیر های محیطی (environment variables) تحت شرایطی وابسته به درخواست، تغییر یابند. بسیار از اختیارات(Options) پیکربندی ماژول ها، بر مبنای امتحان کردن متغیرهای محیطی صورت می پذیرد.

در مثال پیکربندی موجود، ما فرض با به پذیرش تمامی ماژولهای پیش فرض قرار داده ایم، اما این مسئله باید در شرایط واقعی این مسئله به ندرت اتفاق می افتد، چرا که شما معمولا می خواهید که لیست ماژولهای خود را به دلخواه تغییر دهید. براید دیدن لیست ماژولهای پیش فرض در آپاچی ۱ ، می توانید از اسکریپت configure سوال کنید. در زیر تنها قسمتی از خروجی آن را در زیر آورده شده است. چرا که خروجی آن بسیار زیاد است:

```
$ ./configure --help
...
[access=yes      actions=yes     alias=yes      ]
[asis=yes        auth_anon=no    auth_dbm=no    ]
[auth_db=no      auth_digest=no  auth=yes        ]
[autoindex=yes   cern_meta=no    cgi=yes        ]
[digest=no       dir=yes         env=yes        ]
[example=no      expires=no      headers=no     ]
[imap=yes        include=yes     info=no        ]
[log_agent=no    log_config=yes  log_forensic=no]
[log_referer=no  mime_magic=no   mime=yes       ]
[mmap_static=no  negotiation=yes proxy=no       ]
[rewrite=no      setenvif=yes    so=no          ]
[speling=no      status=yes      unique_id=no   ]
[userdir=yes     usertrack=no    vhost_alias=no ]
...
```

اگر بخواهیم تفسیر کنیم، userdir=yes بدین معنی است که ماژول mod_userdir به صورت پیش فرض فعال است. استفاده -- enable-module و disable-module-- باعث فعال شدن و غیر فعال شدن ماژول ها در لیست خواهد شد.

```
$ ./configure \
> --prefix=/usr/local/apache \
> --enable-module=rewrite \
> --enable-module=so \
> --disable-module=imap \
> --disable-module=userdir
```

بدست آوردن لیست ماژولها در آپاچی ۲ خیلی سخت تر است. لیست زیر را با کامپایل کردن آپاچی ۲، بدون اعمال تغییر در configure و سپس سوال از httpd بدست آمده:

```
$ ./httpd -l
Compiled in modules:
  core.c
  mod_access.c
  mod_auth.c
  mod_include.c
  mod_log_config.c
  mod_env.c
  mod_setenvif.c
  prefork.c
  http_core.c
  mod_mime.c
  mod_status.c
  mod_autoindex.c
  mod_asis.c
  mod_cgi.c
  mod_negotiation.c
  mod_dir.c
  mod_imap.c
```

```
  mod_actions.c
  mod_userdir.c
  mod_alias.c
  mod_so.c
```

برای تغییر دادن وضعیت پیش فرض در لیست ماژولها آپاچی ۲ باید قواعد متفاوتی را اعمال کرد:

```
$ ./configure \
> --prefix=/usr/local/apache \
> --enable-rewrite \
> --enable-so \
> --disable-imap \
> --disable-userdir
```