

راه اندازی سرور با



FreeBSD®

سری مقالات ارائه شده در سایت

www.Mabedini.ir

نویسنده
محمد عابدینی



فهرست مطالب

3مقدمه
4 راه اندازی اولین سرور در قالب FreeBSD سرور ftp:
12 راه اندازی کردن ftp در FreeBSD:
15 راه اندازی کردن سرور DNS در FreeBSD
21 نصب DNS unbound در FreeBSD
29 فایل zone چیست؟
32 راه اندازی NSD در FreeBSD
36 اضافه کردن Zone در NSD
40 راه اندازی DHCP سرور در FreeBSD
43 راه اندازی time سرور در FreeBSD
49 Squid چیست؟
53 نصب Squid
57 راه اندازی Squid و برنامه های sqttop و squidclient
62 سرور NFS در FreeBSD
69 راه اندازی web سرور در FreeBSD
72 نصب و راه اندازی SSH
77 اتصال به سرور SSH از طریق کلید
81 روش اتصال به سرور SSH با putty و کلید عمومی
88 برنامه syslog در FreeBSD
90 برنامه newsyslog در FreeBSD
96 پیکربندی هسته در FreeBSD
101 فایروال IPFW در FreeBSD
104 قواعد رول نویسی در IPFW



108	Nat کردن ترافیک با IPFW در FreeBSD
112	قابلیت port redirection و address redirection در Natd
115	راه اندازی Zebra در FreeBSD
126	پیکربندی FreeBSD Client در NSI
129	RAID0 در FreeBSD
133	راه اندازی RAID1 در FreeBSD
137	راه اندازی RAID03 در FreeBSD
139	راه اندازی کردن PXE در FreeBSD
145	File Flag و kernel secure level در FreeBSD
150	اشتراک گذاشتن دسترسی root با کاربر su و sudo
154	مانیتور کردن وضعیت سیستم با iostat
159	فرمان systat در FreeBSD
170	کار با inetd در FreeBSD
173	فایروال از نوع host-based firewall در FreeBSD
176	مقدمه بر شبکه بی سیم در FreeBSD
178	محدود کردن فضای مصرفی دیسک با Disk quotas



مقدمه

سالها قبل بعد از شروع فعالیت در سایت خودم به آدرس Mabedini.ir تلاش کردم که در ابتدا سری از مقالاتی به عنوان BSD به زبان ساده را در سایت ارایه کنم برای آموزش مقدماتی BSD این دوره بعد از تکمیل شدن در قالب یک فایل در کانال رسمی تلگرام سایت ارایه شد، بعد از اتمام دوره BSD به زبان ساده دوره ای در سایت ایجاد کردم با عنوان FreeBSD برای مهندسين شبکه که به راه اندازی کردن سرويسها و سرورهای شبکه با زبانی ساده و کاربردی پرداختم و تلاش من در ارایه این مطالب دقت در پیاده سازی کامل کدها و دستورات و عملکرد درست سرويسها بود، بعد از اتمام این سری از مقالات به صورت تصویری هم در کانال آپارات رسمی سایت هم دوره تصویری آنرا در بخشهای مختلف ارایه کردم، حال در این زمان و با تغییر ساختار سایت سری مقالات ارایه شده به ترتیب و پشت سر هم ارایه نشده است و نیاز به ارایه کتابی در این زمینه با استفاده از مقالات ارایه شده در سایت را شروع کردم ، باشد که کتابی مفید و کاربردی در دست داشته باشید و با استفاده از آن بتوانید بخشی از نیاز های شبکه ای خود را در قالب سیستم عاملی امن و پر قدرت راه اندازی کنید.



راه اندازی اولین سرور در قالب FreeBSD سرور ftp:

ftp چیست؟

پروتکل ftp یا همان file transfer protocol یک پروتکل انتقال فایل در شبکه است که به نوع سیستم عامل متصل شده به سرور وابسته نبوده سیستم عامل های مبتنی بر ویندوز و لینوکسی و یونیکسی هم می توانند از آن استفاده کنند. راه اندازی آن ساده است و در حالت client/server این سرویس ارایه می شود. اولین بار این پروتکل در سال 1971 توسط Abhay Bhushan در RFC 114 ارایه شد، در سال 1980 یکی از بخش های پروتکل TCP/IP شد و در September 1998 قابلیت استفاده از ورژن 6 آدرس IP به آن اضافه شد.

پروتکل ftp یا همان file transfer protocol یک پروتکل انتقال فایل در شبکه است که به نوع سیستم عامل متصل شده به سرور وابسته نبوده سیستم عامل های مبتنی بر ویندوز و لینوکسی و یونیکسی هم می توانند از آن استفاده کنند. راه اندازی آن ساده است و در حالت client/server این سرویس ارایه می شود. اولین بار این پروتکل در سال 1971 توسط Abhay Bhushan در RFC 114 ارایه شد، در سال 1980 یکی از بخش های پروتکل TCP/IP شد و در September 1998 قابلیت استفاده از ورژن 6 آدرس IP به آن اضافه شد.

برای برقرار ارتباط با سرور کلاینتها هم می توانند از نام کاربری و رمز عبور استفاده کنند و هم می توانند در قابل Anonymous به سرور شما متصل شوند و حالت Anonymous باید در سمت سرور شما فعال شده باشد و یکی از راههای اشتراک گذاری فایل در سطح شبکه های بزرگ است که شما قصد ندارید به هر فردی نام کاربری و رمز عبور دهید.

یکی از نقاط ضعف این سرویس کدگذاری نشدن رمز عبور و نام کاربری در زمان اتصال به سرور است، این بخش به صورت متن ساده منتقل شده و هر فردی که به شبکه شما دسترسی داشته باشد به راحتی با قابلیت اسنایف می تواند به رمز عبور و نام کاربری شما دسترسی پیدا کند. برای افزایش امنیت در این سرور قابلیت ftps به آن اضافه شده است.

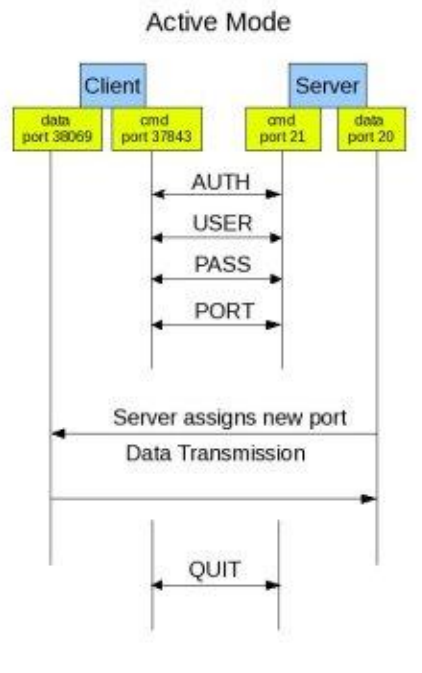
این سرور در دو حالت active و passive به ارایه سرویس می پردازد در ادامه با این دو حالت آشنا می شود.

بررسی حالت Active:

در این روش برقرار ارتباط، سیستم کلاینت با یک پورت تصادفی با عدد بالای 1023 به پورت سرور در شماره 21 متصل می شود، در مرحله بعد کلاینت در پورت تصادفی قبلی بعلاوه یک عدد (n+1) به برقرار ارتباط توسط سرور می نشیند و این پورت را برای سرور ftp ارسال می کند.

نکته:

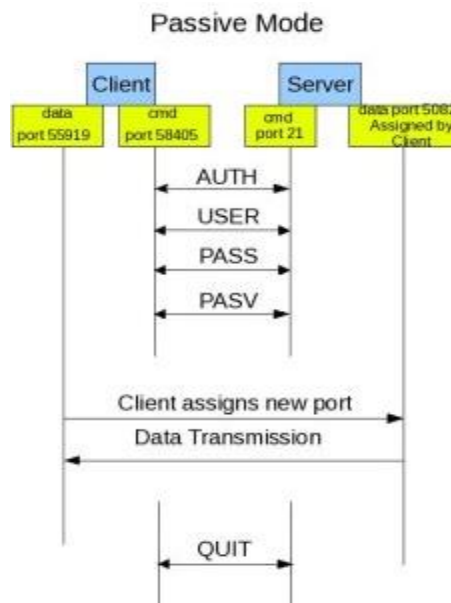
پورتهای ارتباطی در سمت سرور برای اجرای فرمانها همیشه پورت 21 است و سرور برای انتقال فایل همیشه از پورت 20 استفاده می کند.



بررسی حالت Active در ftp

حالت passive

در بعضی از موارد کلاینت در پشت فایروال و nat قرار دارد و سرور ftp نمی تواند به صورت مستقیم به پورت ارسالی از سمت سرور متصل شود برای حل این مطلب حالتی دیگر در برقرار ارتباط به نام passive ایجاد شده است. برای برقرار کردن این نوع ارتباط کلاینت با فرمان pasv به سرور می گوید که قصد برقرار ارتباط در حالت passive را دارد. در این روش قبل از برقرار ارتباط کلاینت دو پورت تصادفی بزرگتر از عدد 1023 به صورت تصادفی با حالت n و n+1 انتخاب می کند و به شماره پورت 21 سرور متصل می شود. بعد از تعیین کردن حالت passive با استفاده از فرمان pasv بین کلاینت و سرور این بار سرور یک پورت تصادفی بالاتر از 1023 باز کرده و به کلاینت اعلام می کند که برای دریافت فایل خود به این شماره پورت با استفاده از پورت تصادفی n+1 خود متصل شود و به دریافت فایل اقدام کند. در این حالت برقرار کنند ارتباط برای دریافت فایل خود کلاینت است، این حالت را در شکل زیر مشاهده می کنید:



بررسی حالت passive در ftp

روشهای برقرار ارتباط با سرور ftp :

شما به دو طریق می توانید با یک سرور ftp ارتباط برقرار کنید، هم از طریق خط فرمان و هم از طریق یک مرورگر وب. در ادامه این دو روش را برای شما توضیح خواهیم داد.

روش استفاده از یک مرورگر وب

اگر شما از DNS سرور برای برقراری ارتباط با سرور های خود استفاده می کنید می توانید از نام بجای آدرس IP برای متصل شدن به سرور استفاده کنید، شما همچنین می توانید از آدرس ip هم به صورت مستقیم برای برقرار کردن ارتباط استفاده کنید. در این روش ما قصد داریم که به سرور ftp مربوطه به سایت ftp.freebsd.org به آدرس ftp.freebsd.org متصل شویم. در این سرور حالت Anonymous فعال بوده و شما نیاز به نام کاربری و رمز عبور ندارید. یک مرورگر وب را باز کنید و آدرس ftp.freebsd.org را در قسمت ادرس وارد کنید به صورت زیر تا به سرور متصل شوید برای اطمینان حاصل کردن از اینکه شما با استفاده از پروتکل ftp به سرور متصل می شوید می توانید خط `ftp://` را به قبل از آدرس خود اضافه کنید تا مرورگر شما از ftp استفاده کند:

← → ↻

Index of /

Name	Size	Date Modified
favicon.ico	5.3 kB	7/18/14, 12:00:00 AM
index.html	682 B	11/2/15, 12:00:00 AM
pub/		7/18/14, 12:00:00 AM

برقراری ارتباط با ftp از طریق وب

در این بخش شما فایلها و پوشه های موجود بر روی سرور را مشاهده می کنید پوشه های به / ختم می شود و با کلیک کردن بر روی آنها می توانید به آنها وارد شوید در شکل زیر وارد پوشه pub شده ایم:

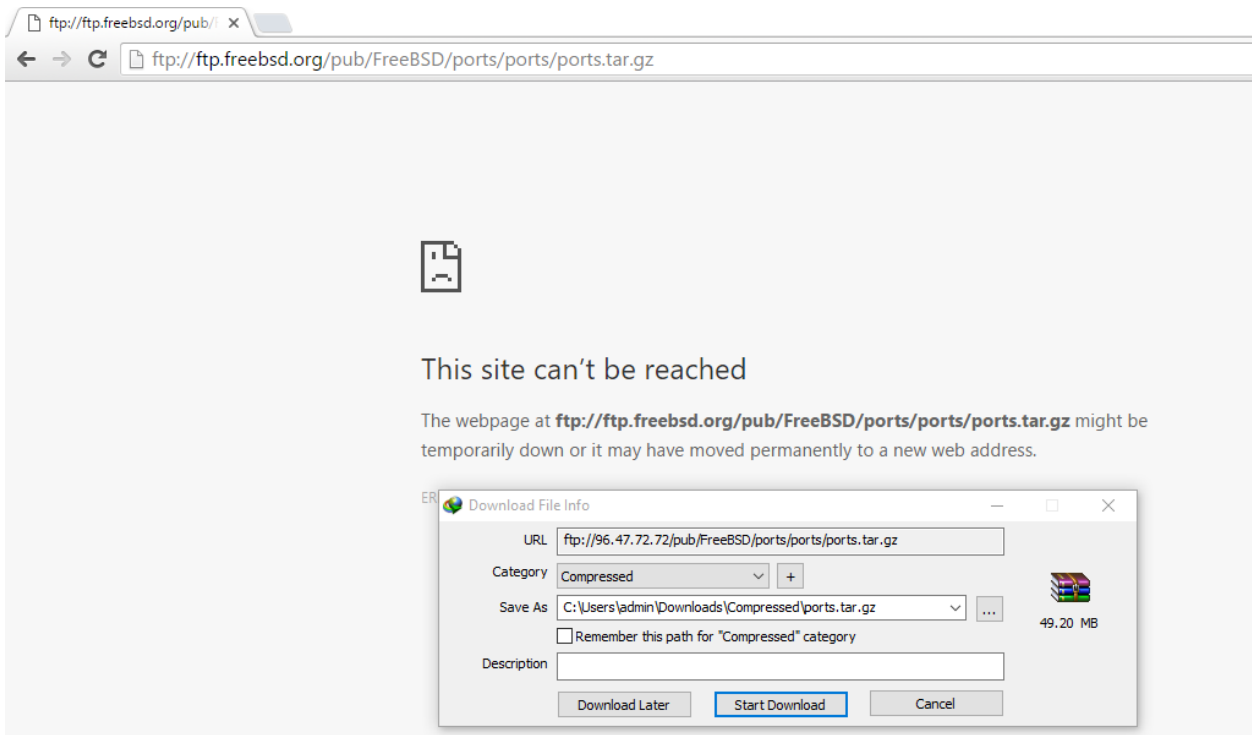


Index of /pub/

Name	Size	Date Modified
[parent directory]		
FreeBSD/		9/2/15, 6:15:00 AM

گشت وگذار در ftp با وب

همانطوری که مشاهده می کنید این بخش خود شامل پوشه دیگری به نام FreeBSD است، در بخش ادرس شما آدرس کامل این بخش یعنی ftp://ftp.freebsd.org/pub را مشاهده می کنید. برای رفتن به شاخه قبل کافیسیت که بر روی parent directory کلیک کنید. برای دانلود کردن یک فایل کافیسیت که بر روی آن کلیک کنید تا نرم افزارهای دانلود باز شده و عمل دانلود را انجام دهند، این عمل در شکل زیر نمایش داده شده است:



دانلود کردن از صفحه وب

ما قصد دریافت فایل ports.tar.gz را از آدرس ftp://ftp.freebsd.org/pub/FreeBSD/ports/ports/ports.tar.gz را داشته ایم شما می توانید از این ادرس هم به صورت مستقیم استفاده کنیم.



روش استفاده از خط فرمان

در این روش برقرار ارتباط شما نیاز به یک خط فرمان دارید باز هم مستقل از سیستم عامل و هر جایی که شما فرمان FTP را داشته باشید می توانید به سرور متصل شوید. کفایت که در خط فرمان ftp را با یک نام و یا یک ادرس ip اجرا کنید، برای برقراری حالت Anonymous کفایت که از سوچ a- قبل از نام سرور استفاده کنید، این فرمان در شکل زیر مشاهده می کنید:

```
Terminal
File Edit View Terminal Tabs Help
root@FreeBSD:/home # ftp -a ftp.freebsd.org
Trying 96.47.72.72:21 ...
Connected to ftp.geo.freebsd.org.
220 This is ftp0.nyi.freebsd.org - hosted at NYI.net.
331 Please specify the password.
230-
230-This is ftp0.nyi.FreeBSD.org, graciously hosted by
230-New York Internet - NYI.net
230-
230-FreeBSD files can be found in the /pub/FreeBSD directory.
230-
230 Login successful.
Remote system type is UNIX.
Using binary mode to transfer files.
ftp>
```

ارتباط با ftp از طریق خط فرمان

همانطوری که مشاهده می کنید شما به سرور متصل شده اید و سیستم عامل سرور UNIX است و برای دریافت فایل باید از حالت binary استفاده کنید. اگر حالت Anonymous فعال نباشد شما باید قبل از ورود به سیستم از نام کاربری و رمز عبور استفاده کنید، در شکل زیر شما روش اتصال به سرور موجود بر روی همان سیستم عامل به آدرس 127.0.0.1 را مشاهده می کنید. در این روش ابتدا فرمان ftp را اجرا کنید تا وارد خط فرمان <ftp> شوید و بعد از اجرا کردن فرمان op شما می توانید آدرس سرور را در مقابل to وارد کنید، بعد از آن شما می توانید نام کاربری و رمز عبور را به صورت زیر وارد کنید، این بخش در شکل زیر نمایش داده شده است :

```
File Edit View Terminal Tabs Help
root@FreeBSD:/home # ftp
ftp> op
(to) 127.0.0.1
Connected to 127.0.0.1.
220 FreeBSD FTP server (Version 6.00LS) ready.
Name (127.0.0.1:root): admin
331 Password required for admin.
Password:
230 User admin logged in.
Remote system type is UNIX.
Using binary mode to transfer files.
ftp> ls
229 Entering Extended Passive Mode (|||63679|)
150 Opening ASCII mode data connection for '/bin/ls'.
total 32
-rw-r--r--  1 admin  admin   1066 Jul 31 20:37 .cshrc
-rw-r--r--  1 admin  admin    252 Jul 31 20:37 .login
-rw-r--r--  1 admin  admin    163 Jul 31 20:37 .login_conf
-rw-----  1 admin  admin    379 Jul 31 20:37 .mail_aliases
-rw-r--r--  1 admin  admin    336 Jul 31 20:37 .mailrc
-rw-r--r--  1 admin  admin    817 Jul 31 20:37 .profile
-rw-----  1 admin  admin    281 Jul 31 20:37 .rhosts
-rw-r--r--  1 admin  admin    978 Jul 31 20:37 .shrc
226 Transfer complete.
```

ارتباط با خط فرمان



این برنامه دارای فرمانهای مختلفی است که در شکل زیر لیست کامل آنرا مشاهده می کنید برای مشاهده این فرمانها کفایت که ؟ را تایپ و بعد Enter کنید تا لیست به صورت نمایش داده شده در شکل زیر برای شما نمایش داده شود:

```

Terminal
File Edit View Terminal Tabs Help
Commands may be abbreviated.  Commands are:
!          epsv6          mget          preserve     sendport
$          exit           mkdir         progress     set
account   features       mls           prompt       site
append    fget           mlsd          proxy        size
ascii     form           mlst          put          sndbuf
bell      ftp            mode          pwd          status
binary    gate           modtime       quit         struct
bye       get            more          quote        sunique
case      glob           mput          rate         system
cd        hash           mreget        rcvbuf       tenex
cdup      help           msend        recv         throttle
chmod     idle           newer         reget        trace
close    image          nlist        remopts      type
cr        lcd            nmap          rename        umask
debug     less           ntrans       reset        unset
delete    lpage         open          restart      usage
dir       lpwd          page          rhelp        user
disconnect ls             passive       rmdir        verbose
edit      macdef        pdir          rstatus      xferbuf
epsv      mdelete       pls           runique      ?
epsv4    mdir          pmlsd        send
ftp>

```

کلیه فرمان های موجود در FTP Client

فرمانهایی که استفاده زیادی در این بخش دارد فرمان های cd و ls است، در مقاله BSD به زبان ساده روش استفاده از این فرمان ها توضیح داده شده است کاربرد این دو فرمان برای نمایش فایلها و شاخه و رفتن به شاخه بعدی است، برای مشاهده شاخه ای که در آن قرار دارید می توانید از فرمان pwd استفاده کنید.

نکته :

فرمان ! یا همان علامت تعجب به شما این امکان را می دهد که در زمانی که به سرور وصل هستید به خط فرمان بازگشته و فرمانهای خود را اجرا کنید بدون اینکه ارتباط شما با سرور قطع شود، این فرمان در زمان دانلود فایلهایی که زمان بر هستند مفید است.

نمایش توضیحات هر فرمان

شما به راحتی می توانید توضیحات کمک کننده هر فرمان را با استفاده از فرمان help مشاهده کنید کفایت بعد از help کفایت که نام فرمان مورد نظر خود را وارد کنید در شکل زیر شما چند help از فرمانهای را مشاهده کنید:



```
Terminal
File Edit View Terminal Tabs Help
ftp> help ls
ls                list contents of remote path
ftp> help pwd
pwd              print working directory on remote machine
ftp> help get
get             receive file
```

نمایش توضیحات هر فرمان

دریافت و ارسال فایل

برای دریافت کردن فایل از سمت سرور باید از فرمان `get` استفاده کنید، برای دریافت یا باید در همان شاخه فایل مورد نظر باشید و یا آدرس آنرا بعد از فرمان `get` وارد کنید، فایل مورد نظر بعد از دانلود شدن در شاخه ای که شما قبل از فرمان `ftp` در قرار داشته اید ذخیره می شود مگر از فرمان `lcd` استفاده کنید و مسیر جدید را تعیین کنید در شکل زیر همان فایل `ports.tar.gz` که در بخش قبلی از طریق مرورگر دانلود کردید را در این بخش هم با استفاده از خط فرمان دانلود می کنید به صورت نمایش در شکل زیر:

```
Terminal
File Edit View Terminal Tabs Help
250 Directory successfully changed.
ftp> ls
229 Entering Extended Passive Mode (|||51720|)
150 Here comes the directory listing.
drwxr-xr-x  2 ftp      ftp          4 Jul 08  2014 distfiles
drwxr-xr-x  2 ftp      ftp          4 Jul 08  2014 local-distfiles
drwxr-xr-x  2 ftp      ftp          4 Oct 28  2012 ports
226 Directory send OK.
ftp> cd ports
250 Directory successfully changed.
ftp> ls
229 Entering Extended Passive Mode (|||64311|)
150 Here comes the directory listing.
-rw-r--r--  1 ftp      ftp          993 Jul 16  2013 README.TXT
lrwxr-xr-x  1 ftp      ftp          47 Oct 28  2012 ports.tar.gz -> ../../de
velopment/tarballs/ports_current.tar.gz
226 Directory send OK.
ftp> pwd
Remote directory: /pub/FreeBSD/ports/ports
ftp> get ports.tar.gz
local: ports.tar.gz remote: ports.tar.gz
229 Entering Extended Passive Mode (|||51965|)
150 Opening BINARY mode data connection for ports.tar.gz (51591521 bytes).
 2% |          | 1193 KiB  91.78 KiB/s  08:55 ETA
```

دانلود کردن فایل

شما در خط آخر از این شکل مقدار زمان مورد نیاز برای دریافت، سرو و حجم فایل برای شما نمایش داده می شود. بعد از پایان دانلود شما می توانید از فرمان `quit` برای خروج از `ftp` و قطع کردن ارتباط با سرور استفاده کنید و سرور پیغامی برای شما در قالب `goodbye` نمایش می دهد.



```
ftp> pwd
Remote directory: /pub/FreeBSD/ports/ports
ftp> get ports.tar.gz
local: ports.tar.gz remote: ports.tar.gz
229 Entering Extended Passive Mode (|||51965|)
150 Opening BINARY mode data connection for ports.tar.gz (51591521 bytes).
100% |*****| 50382 KiB 109.71 KiB/s 00:00 ETA
226 Transfer complete.
51591521 bytes received in 07:39 (109.66 KiB/s)
ftp> quit
221 Goodbye.
root@FreeBSD:/home #
```

نمایش وضعیت دانلود

همانطوری که مشاهده کردید فایل به صورت کامل دریافت شده و با فرمان quit شما ارتباط خود را با سرور قطع کرده اید. برای ارسال فایل در ابتدا شما باید به شاخه ای که قصد آپلود کردن فایل را دارید دسترسی نوشتن داشته باشید و بعد از برقرار ارتباط با سرور و رفتن به شاخه مورد نظر از فرمان put استفاده کنید بعد از آن باید نام فایل با آدرس ذخیره شده بر روی هارد خود را وارد کنید. در شکل زیر به سرور 127.0.0.1 با کاربر admin متصل شده این و فایل ports.tar.gz را با put به سمت سرور ارسال می کنیم:

```
Terminal
File Edit View Terminal Tabs Help
root@FreeBSD:/home # ftp
ftp> op
(to) 127.0.0.1
Connected to 127.0.0.1.
220 FreeBSD FTP server (Version 6.00LS) ready.
Name (127.0.0.1:root): admin
331 Password required for admin.
Password:
230 User admin logged in.
Remote system type is UNIX.
Using binary mode to transfer files.
ftp> put ports.tar.gz
local: ports.tar.gz remote: ports.tar.gz
229 Entering Extended Passive Mode (|||52680|)
150 Opening BINARY mode data connection for 'ports.tar.gz'.
100% |*****| 50382 KiB 45.62 MiB/s 00:00 ETA
226 Transfer complete.
51591521 bytes sent in 00:01 (45.48 MiB/s)
ftp> quit
221 Goodbye.
root@FreeBSD:/home #
```

برقراری ارتباط با سرور FTP



راه اندازی کردن ftp در FreeBSD:

برای راه اندازی کردن سرور ftp شما نیاز به نصب هیچگونه برنامه اضافی ندارید و فقط کافیست که این برنامه را از طریق فرمان های rc و یا فرمان service راه اندازی کنید. در ادامه با روش های پیکربندی به سرویس آشنا می شوید.

فایل `/etc/ftusers`:

یکی از مهمترین بخش های پیکربندی این سیستم تعیین کردن نوه دسترسی کاربران محلی به سرور ftp است، اگر قصد دارید که در حالت Anonymous از این سرویس استفاده کنید نیاز به تنظیمات این بخش ندارید. در هر صورت در سیستم عامل FreeBSD یک سری کاربران سیستم وجود دارند که در زمان نصب به سیستم اضافه می شوند و برای راه اندازی هر سرویس سیستمی ایجاد شده اند. برای جلوگیری از ورود این چنین کاربران در FreeBSD فایلی به نام `ftusers` ایجاد شده است که نام هم کاربری که در آن قرار داشته باشد نمی تواند از طریق ftp به سیستم شما متصل شود. به این نکته توجه کنید که کاربرانی را که شما در مراحل بعد از نصب ایجاد می کنید در این فایل وجود ندارد و برای اضافه کردن کاربران در این فایل کافیست که نام کاربر را در خطی جداگانه در این فایل اضافه کنید.

محدود کردن دسترسی به فایلها با `/etc/ftchroot`:

در بعضی از موارد نیاز می شود که دسترسی کاربر وارد شده از طریق ftp را بر روی یک شاخه خاص محدود کنید، این امر به این خاطر است که زمانی که کاربر محلی از طریق سرویس ftp به سیستم وارد می شود می تواند در همه شاخه ها و پوشه ها گشت و گذار کند و از زمان ls برای مشاهده لیست فایلها و غیر استفاده کند. برای محدود کردن این دسترسی فایل دیگری باید ایجاد کنید که به صورت پیش فرض در سیستم وجود ندارد و باید ایجاد شود. نام این فایل `ftchroot` است که در شاخه `etc/` باید ایجاد شود. این فایل توسط برنامه Ftpd در زمان شروع یک ارتباط ftp خوانده می شود البته بعد از زمانی که کاربر در سیستم با نام کاربری و رمز عبور درست وارد شد.

هر خط در این فایل به یک کاربر و یا گروه خاصی مربوط می شود، اگر نام کاربر و یا گروهی که کاربر به آن تعلق دارد وارد سیستم شما و با خطی در این فایل منطبق شود شاخه `root` کاربر به شاخه ای که در مقابل نام کاربر در این فایل نوشته شده است بر کاربر اعمال می شود.

ترتیب در خواندن این فایل از بالا به پایین است و اولین خطی که با نام کاربری و یا گروه شما منطبق می شود دسترسی آن اعمال می شود. هر بخش هم در این فایل با یک `space` از هم جدا می شود. اولین فیلد در این فایل نام کاربری و یا گروه مورد نظر شماست. اگر قبل از نام گروه علامت `@` قرار گرفته باشد یعنی همه کاربرانی که در آن گروه عضو هستند. اگر هم خطی با فقط `@` شروع شده باشد یعنی همه کاربران. بخش دوم که باید `space` از بخش اول جدا می شود شاخه است که کاربر و یا گروه کاربران به آن محدود خواهند شد. اگر هم هیچ مسیری در این بخش وارد نکرده باشید کاربر فقط به شاخه `log in` خود محدود می شود. در شکل زیر یک مثال از این فایل را مشاهده می کنید:



```

Terminal
File Edit View Terminal Tabs Help
^[ (escape) menu ^y search prompt ^k delete line ^p prev li ^g prev page
^o ascii code ^x search ^l undelete line ^n next li ^v next page
^u end of file ^a begin of line ^w delete word ^b back 1 char
^t top of text ^e end of line ^r restore word ^f forward 1 char
^c command ^d delete char ^j undelete char ^z next word
=====line 6 col 21 lines from top 9 =====
#admin chroot to /var/ftp
admin /var/ftp

#all user in user1 groups chroot to /var/ftp/user
@user1 /var/ftp/user

#every user chroot to public_ftp
@ /var/ftp/public_ftp

```

نمایش فایل ftpchroot

در خط اول کاربر admin به شاخه /var/ftp محدود می شود، در خط دوم همه کاربران عضو در گروه user1 به شاخه /var/ftp/user محدود می شوند و در خط پایانی همه کاربران به شاخه /var/ftp/public_ftp محدود خواهند شد.

فعال کردن حالت Anonymous:

برای راه اندازی کردن سرور ftp در حالت Anonymous شما باید یک کاربر به نام ftp ایجاد کنید و این کاربر بتواند در سیستم وارد شود و نام آن در فایل ftpusers نباشد و در زمان وارد شدن از طریق نام کاربری ftp یا Anonymous اقدام کنید و یا از سویچ a- استفاده کنید (برای مطالعه بیشتر در مورد راه های اتصال به سرور Ftp مقاله ftp چیست را مطالعه کنید) در صورت نیاز به رمز عبور یک آدرس میل وارد کنید و برنامه chroot به صورت خودکار بعد از وارد شدن شما به سیستم مسیر شاخه home کاربر را برای شما chroot می کند.

پیغام های ورود به سیستم:

در زمان وارد شده به سرور Ftp از طریق خط فرمان برای شما دو نوع پیغام خوش آمد گویی نمایش داده می شود، اولین پیغام در برقرار ارتباط با سرور برای شما نمایش داده می شود که این پیغام ها از یک فایل متنی به نام ftpwelcome خوانده شده و نمایش داده می شود که در شاخه etc/ قرار گرفته است. شما می توانید هر متنی را دوست دارید نوشته و این پیغام به صورت عمومی منتشر می شود.

بعد از وارد شدن کاربر به سیستم و به اصطلاح log in کردن صحیح با نام کاربری و رمز عبور معتبر به سیستم پیغام دیگری برای کاربر قابل نمایش است که این پیغام هم از فایل متنی به نام ftpmotd خوانده شده و نمایش داده می شود که در شاخه etc/ قرار دارد شما می توانید هر پیغامی را که مایل هستید کاربر بعد از وارد شدن به سیستم مشاهده کند را در این بخش وارد کنید.



راه اندازی سرور FTP:

برای راه اندازی دائمی این سرور از طریق فایل `etc/rc.conf` اقدام نمایید و خط زیر را در این فایل اضافه کنید:

```
ftpd_enable="YES"
```

بعد از اضافه کردن این بخش از دو طریق می توانید راه اندازی سرور را انجام دهید روش اول با استفاده از فرمان `service` به صورت زیر:

```
#service ftpd start
```

و یا فرمان های `rc` به صورت زیر:

```
#/etc/rc.d/ftpd start
```

چک کردن فعال بودن سرویس

یکی از راه های چک کردن سرویس باز بودن پورت باز سیستم است در FreeBSD فرمانی است به نام `sockstat` که وضعیت پورت های باز را نمایش می دهد در شکل زیر خروجی این فرمان نمایش داده شده است:

```

Terminal
File Edit View Terminal Tabs Help
root@OpenBSD:/etc # sockstat -l4
USER  COMMAND  PID  FD  PROTO  LOCAL ADDRESS  FOREIGN ADDRESS
root  ftpd     95913 6  tcp4   *:21           *: *
root  ss-server 29210 7  tcp4   148.251.210.242:1081  *: *
root  sendmail 650 4  tcp4   127.0.0.1:25    *: *
root  syslogd  513 7  udp4   *:514          *: *
root@OpenBSD:/etc #

```

نمایش وضعیت سرویسهای شبکه

در خروجی فرمان بالا مشاهده می کنید که بر روی پورت 21 برنامه `ftpd` آماده اراه سرویس است.



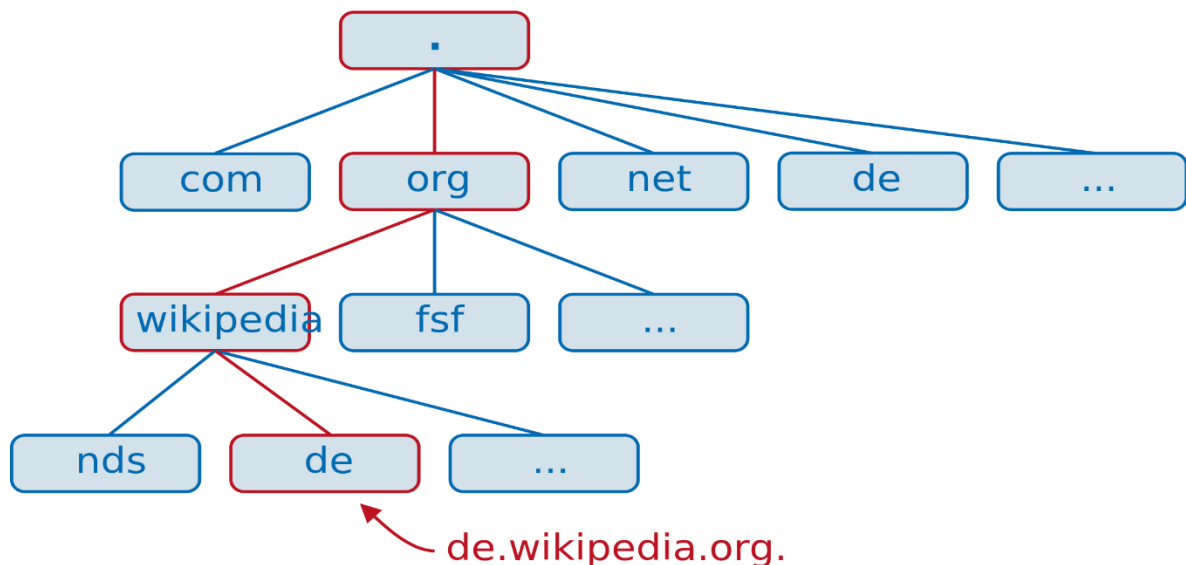
راه اندازی کردن سرور DNS در FreeBSD

DNS چیست؟

ذهن انسان توانایی به خاطر سپاری اسم ها را بهتر از اعداد را دارد، در دنیای شبکه کامپیوترها از آدرس های IP برای برقراری ارتباط استفاده می کنند. در این میان راهی برای تبدیل کردن این نام به آدرس IP وجود دارد آن هم استفاده سرور DNS در شبکه است. این سرور وظیفه تبدیل کردن نام به آدرس IP و برعکس را برعهده دارد و یکی از سرورهای مورد نیاز برای دسترسی به شبکه جهانی اینترنت است. حجم زیادی از ترافیک اینترنت به پرس و جو کردن بین کاربران و سرورهای DNS برای برقرار کردن ارتباط با سیستم های مورد نظر می شود. برای کاهش این ترافیک یک مدلی از DNS ایجاد شده است به نام cache DNS که وظیفه آنها ذخیره کردن اطلاعاتی است که کاربران از آن پرس و جو می کنند. در ادامه با انواع DNS سرورها آشنا می شوید.

روش پرس و جو

پرس و جو کردن از سرورهای DNS به صورت سلسله مراتبی انجام می شود و هر سروری نیازی نیست که اطلاعات کاملی از همه سیستم ها و نام ها داشته باشد فقط باید نام سرورهای دیگری را داشته باشد که اون سرور ها نام سرور بعدی را دارد. در شکل زیر سلسله مراتبی بودن تقسیم بندی نام ها در اینترنت را برای شما نمایش داده شده است :



مدل جستجو کردن دامین ها در اینترنت

همانطوری که مشاهده می کنید یک سرور root به نام . یا همان نقطه در اینترنت وجود دارد که تعداد آنها در حال حاضر 13 عدد است که آدرس IP آنها ثابت است، دلیل ثابت بودن آدرس IP آنها این است که این سرور ها حالت راهنما برای کاربران را دارند، root server ها از سرورهای لایه بعدی یعنی سرورهای DNS مربوطه به حوزه های com net و سرورهای DNS مربوط به هر کشور مطلع هستند و زمانی که درخواستی را دریافت می کنند به تناسب این بخش آدرس سرور بعدی را در اختیار client قرار می دهند و سرورهای level بعدی فقط بخش دوم نام یعنی آدرس ip نام های اصلی هر بخش را در دارند، در

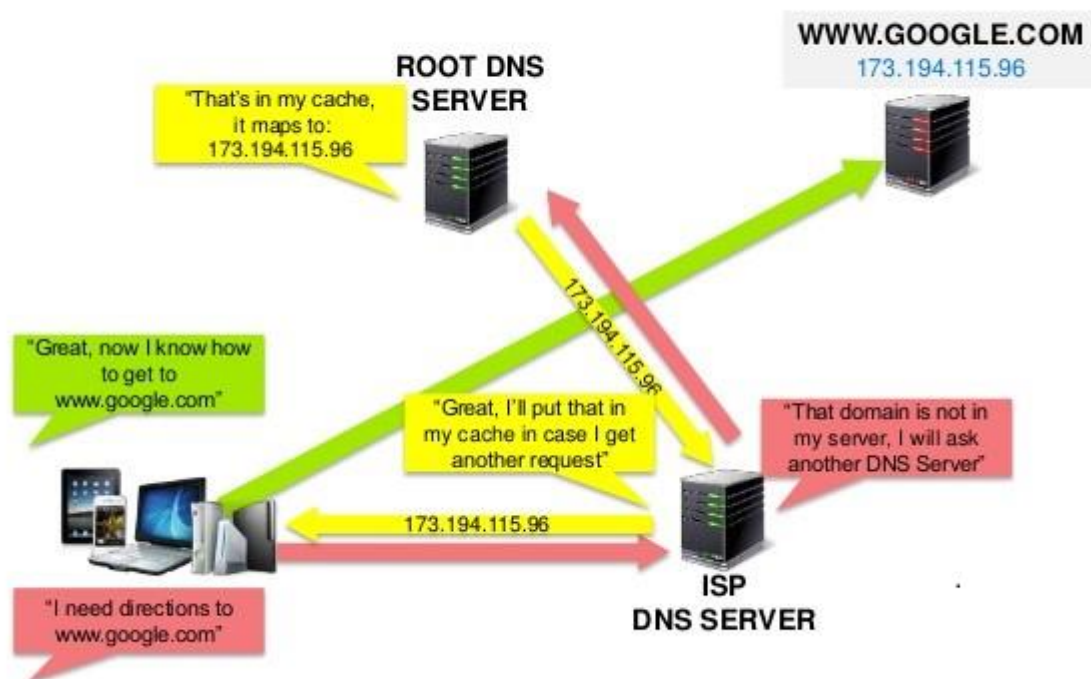


مرحله بعدی این DNS سرور های حوزه های هر نام هستند که ip هر سرور خود را در اختیار client قرار می دهند برای مثال DNS سرور حوزه Wikipedia از ip سرور بخش de این نام حوزه خبر دارد و آنرا می تواند در اختیار client قرار دهد. به این صورت سلسله مراتبی می توانید به هر سرور موجود در اینترنت دسترسی داشته باشید.

سایتی به نام <http://www.root-servers.org> وجود دارد که شما می توانید لیست پخش شده از DNS سرورها را مشاهده کنید.

در شکل زیر روش جستجو برای باز کردن سایت www.google.com برای شما به تصویر کشده شده است، به این نام در اصطلاح FQDN می گویند که .com به این معنا است که این سایت یک سایت تجاریست، google نام حوزه و یا همان اسم سایت است و در بخش آخر هم www قرار دارد که بدین معنی است که شما با سرور www کار دارید. سرور های دیگری هم در این نام حوزه قرار دارند مثل mail.google.com و غیره.

How Does DNS Work?



روش جستجو کردن یک نام

در شکل بالا مشاهده می کنید که ابتدا client از سرور DNS شرکت سرویس دهنده خود با همان ISP در مورد آدرس IP سوال می پرسد و این DNS سرور هست که سایر مراحل را طی می کند تا آدرس را تهیه کرده و به Client می دهد تا بتواند ارتباط خود را برقرار کند.



برنامه های جستجوی نام

دو برنامه برای سوال و جواب کردن با سرور های DNS وجود دارد، برنامه اول به نام nslookup است که برنامه ای قدیمیست و در همه جا وجود دارد. برنامه بعدی dig است که خروجی بیشتری برای شما نمایش می دهد و در سیستم عامل های که از BIND استفاده می کنند وجود دارد و اطلاعات بیشتری برای شما نمایش می دهد و در FreeBSD هم وجود دارد، اگر هم قصد دارید که از این برنامه استفاده کنید و بروی سیستم شما نصب نیست به سایت <http://dig-nlookup.nmonitoring.com> مراجعه کنید.

برنامه nslookup

این برنامه در حال حاضر در FreeBSD ورژن 10 به بعد دیگر وجود ندارد باید برنامه bind-tools را از شاخه برنامه ها به مسیر `usr/ports/dns/bind-tools/` نصب کنید. (برای آموزش روش نصب برنامه در FreeBSD به مقاله آن به نام مقدمه ای بر نصب برنامه در FreeBSD مراجعه کنید). شما با استفاده از این فرمان می توانید به سرور های DNS مختلف متصل شوید و به پرس و جو آدرس IP نام حوزه خود پردازید، اگر سروری که به آن متصل هستید قابلیت تبدیل IP به نام را داشته باشید شما می توانید از این فرمان استفاده کنید. بعد از اجرا این فرمان شما وارد خط فرمان این برنامه می شوید مثل شکل زیر:

```
Terminal
File Edit View Terminal Tabs Help
root@FreeBSD:~ # nslookup
>
```

فرمان nslookup

علامت > بدین معناست که شما وارد برنامه شدید و می توانید فرمان های این بخش را اجرا کنید، برای تغییر دادن سرور پیش فرض از فرمان `lserver` و بعد آدرس IP سرور استفاده کنید، برای جستجو هم کفایت که نامی که مورد نظر است را وارد کنید. این دو امر در شکل زیر نمایش داده شده است:

```
Terminal
File Edit View Terminal Tabs Help
root@FreeBSD:~ # nslookup
> lserver 8.8.8.8
Default server: 8.8.8.8
Address: 8.8.8.8#53
> www.google.com
Server:      8.8.8.8
Address:    8.8.8.8#53

Non-authoritative answer:
Name:   www.google.com
Address: 74.125.206.105
Name:   www.google.com
Address: 74.125.206.104
Name:   www.google.com
Address: 74.125.206.147
Name:   www.google.com
```

جستجوی کردن با فرمان nslookup



برای جستجو بر اساس IP کافیست که آدرس IP مورد نظر را وارد کنید این بخش در شکل زیر نمایش داده شده است:

```
Terminal
File Edit View Terminal Tabs Help
> 8.8.4.4
Server:      8.8.8.8
Address:     8.8.8.8#53

Non-authoritative answer:
4.4.8.8.in-addr.arpa  name = google-public-dns-b.google.com.

Authoritative answers can be found from:
> 4.2.2.4
Server:      8.8.8.8
Address:     8.8.8.8#53

Non-authoritative answer:
4.2.2.4.in-addr.arpa  name = d.resolvers.level3.net.

Authoritative answers can be found from:
>
```

تغییر دادن آدرس سرور DNS پیش فرض

برنامه dig

در FreeBSD و سیستم عامل هایی که از BIND استفاده می کنند فرمانی وجود دارد به نام dig که عمل جستجو را انجام می دهند، این فرمان کل اطلاعات یک zone را برای شما نمایش می دهد. بعد اضافه شدن unbound و حذف شدن dind از FreeBSD این برنامه هم از FreeBSD حذف شده و نیاز به نصب bind-tools دارید. استفاده کردن از این برنامه بسیار ساده است و شما می توانید به راحتی بعد از فرمان dig نام حوزه خود را وارد کنید تا اطلاعات کاملی را مشاهده کنید به صورت زیر:



```

Terminal
File Edit View Terminal Tabs Help
root@FreeBSD:~ # dig www.mabedini.com

;<<>> DiG 9.10.3-P4 <<>> www.mabedini.com
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 19500
;; flags: qr rd ra; QUERY: 1, ANSWER: 2, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; MBZ: 0005 , udp: 8192
;; QUESTION SECTION:
;www.mabedini.com.          IN      A

;; ANSWER SECTION:
www.mabedini.com.         5       IN      CNAME   mabedini.com.
mabedini.com.             5       IN      A       5.135.219.2

;; Query time: 165 msec
;; SERVER: 192.168.233.2#53(192.168.233.2)
;; WHEN: Mon Aug 01 00:41:52 IRDT 2016
;; MSG SIZE rcvd: 75

root@FreeBSD:~ # █

```

خروجی فرمان dig در جستجو کردن

برای مثال شما اطلاعات mabedini.com را در شکل بالا مشاهده می کنید. برای تغییر دادن سرور خود کافیست که بعد از فرمان dig از @ و بعد نام سرور استفاده کنید به صورت زیر :

```

Terminal
File Edit View Terminal Tabs Help
root@FreeBSD:~ # dig @4.2.2.4 mabedini.ir

;<<>> DiG 9.10.3-P4 <<>> @4.2.2.4 mabedini.ir
; (1 server found)
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 26270
;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 8192
;; QUESTION SECTION:
;mabedini.ir.              IN      A

;; ANSWER SECTION:
mabedini.ir.              14391   IN      A       5.135.219.2

;; Query time: 149 msec
;; SERVER: 4.2.2.4#53(4.2.2.4)
;; WHEN: Mon Aug 01 00:45:38 IRDT 2016
;; MSG SIZE rcvd: 56

root@FreeBSD:~ # █

```

تغییر دادن آدرس سرور DNS پیش فرض در dig



برنامه نمایش اطلاعات از همه سرورهای موجود در یک حوزه ای کلمه ANY در بعد از نام حوزه استفاده کنید برای مثال در شکل زیر خروجی این فرمان را مشاهده می کنید:

```

Terminal
File Edit View Terminal Tabs Help
; <<>> DiG 9.10.3-P4 <<>> @4.2.2.4 mabedini.ir ANY
; (1 server found)
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 32787
;; flags: qr rd ra; QUERY: 1, ANSWER: 5, AUTHORITY: 0, ADDITIONAL: 2

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 8192
;; QUESTION SECTION:
;mabedini.ir.                IN          ANY

;; ANSWER SECTION:
mabedini.ir.                14400      IN          SOA         ns231.mihanwebhost.com. mihanspa
.gmail.com. 2016081303 14400 130 1209600 14400
mabedini.ir.                14400      IN          NS          ns232.mihanwebhost.com.
mabedini.ir.                14400      IN          NS          ns231.mihanwebhost.com.
mabedini.ir.                14187     IN          A           5.9.123.24
mabedini.ir.                14400      IN          MX          0 mail.mabedini.ir.

;; ADDITIONAL SECTION:
mail.mabedini.ir.          14400      IN          A           5.9.123.24

```

نمایش همه اطلاعات DNS یک سایت با استفاده از dig

نکته:

در FreeBSD ورژن 10 به بعد از فرمان druil استفاده می شود که روش استفاده از آن مثل فرمان dig است.

معرفی انواع سرورهای موجود DNS:

در بخش بعدی با انواع سرورهای DNS و روشهای نصب و راه اندازی آنها آشنا میشود قبل از آن با انواع آنها به صورت مختصر آشنا می شوید، یک سری از DNS سرورهای فقط در حالت Cache کار می کنند و وظیفه آنها ذخیره کردن اطلاعات پرس و جوی کاربران است و zone از خود ندارند.

نوع دیگری هم از DNS سرورها قابلیت cache نداشته و فقط در مورد حوزه هایی که در zone خود وجود دارند اطلاعات را در اختیار client ها قرار می دهند، به این نوع از Authoritative است.

در بعضی از موارد یک سرور می تواند هر دو عمل بالا را با هم انجام دهند.



نصب unbound DNS در FreeBSD

از ورژن 10 به بعد در FreeBSD DNS سرور BIND از سیستم عامل حذف شد و بجای آن unbound به این سیستم اضافه شد. ان سرور فقط برای استفاده از Chaching استفاده می شود و برای استفاده از حالت aauthoritive باید از سیستم ports برنامه مورد نظر خود را نصب کنید. برنامه unbound به صورت پیش فرض از ورژن 10.1 در FreeBSD نصب شده است و به صورت پیش فرض فقط برای کاربر سیستم محلی یا همان localhost عمل کش کردن را انجام می دهد، برای دسترسی به قابلیت های بیشتر از آن unbound را از سیستم ports به آدرس /usr/ports/unbound/ برنامه را نصب کنید و یا از فرمان زیر از طریق بسته های این کار را انجام دهید:

```
#pkg install unbound
```

در صورتی که شما نیاز به تنظیمات بیشتری داشته باشید بهتر است که این برنامه را از طریق سیستم ports نصب کنید.

راه اندازی Unbound:

بعد از نصب برنامه unbound را با استفاده از این فرمان راه اندازی کنید تا از نصب شدن برنامه بر روی سیستم خود مطمئن شوید، خروجی این فرمان را در شکل زیر مشاهده می کنید:

```
Terminal
File Edit View Terminal Tabs Help
root@FreeBSD:~ # unbound
[1470003579] unbound[55357:0] error: Could not open /var/unbound/unbound.conf: No such file or directory
[1470003579] unbound[55357:0] warning: Continuing with default config settings
[1470003579] unbound[55357:0] error: bind: address already in use
[1470003579] unbound[55357:0] fatal error: could not open ports
root@FreeBSD:~ #
```

خروجی فرمان unbound بعد از نصب کردن

همانطوری که مشاهده می کنید برای راه اندازی شما باید فایل unbound.conf را در شاخه /usr/local/etc/unbound قرار دهید. اگر برنامه unbound را نصب کرده باشید نمونه ای از این فایل پیکربندی در زیر شاخه قرار می گیرد:

```
Terminal
File Edit View Terminal Tabs Help
root@FreeBSD:/usr/local/etc/unbound # ls
unbound.conf.sample
root@FreeBSD:/usr/local/etc/unbound #
```

فایل پیش فرض سرور unbound

نام این فایل unbound.conf.sample است که باید آنرا در شاخه مورد نظر کپی کنید. بعد از کپی کردن این فایل می توانید آنرا با استفاده از فرمان unbound-checkconf آنرا چک کنید و از عدم وجود خطا در این فایل مطمئن شوید، اگر خطایی در این فایل وجود نداشته باشد خروجی به صورت زیر مشاهده می کنید:



```
Terminal
File Edit View Terminal Tabs Help
root@FreeBSD:/usr/local/etc/unbound # unbound-checkconf
unbound-checkconf: no errors in /var/unbound/unbound.conf
root@FreeBSD:/usr/local/etc/unbound # █
```

چک کردن پیکربندی سرور Unbound

فرمان unbound-control

برای مدیریت کردن این برنامه فرمانی وجود دارد به نام unbound-control که شما می توانید از طریق آن به مدیریت کردن برنامه unbound بپردازید در ابتدا برای راه اندازی آن باید فرمان unbound-control-setup برای تولید کردن کلید های مربوط به اتصال را ایجاد کنید، خروجی این فرمان در شکل زیر نمایش داده شده است:

```
Terminal
File Edit View Terminal Tabs Help
root@FreeBSD:/usr/local/etc/unbound # unbound-control-setup
setup in directory /usr/local/etc/unbound
generating unbound_server.key
Generating RSA private key, 3072 bit long modulus
.....++
.++
e is 65537 (0x10001)
generating unbound_control.key
Generating RSA private key, 3072 bit long modulus
.....
.....++
.....++
e is 65537 (0x10001)
create unbound_server.pem (self signed certificate)
create unbound_control.pem (signed client certificate)
Signature ok
subject=/CN=unbound-control
Getting CA Private Key
Setup success. Certificates created. Enable in unbound.conf file to use
root@FreeBSD:/usr/local/etc/unbound # █
```

خروجی فرمان unbound-control-setup

بعد از اتمام این بخش حال شما نیاز دارید که در فایل پیکربندی unbound.conf خط زیر را اضافه کنید:

```
# enable remote-control
remote-control:
    control-enable: yes
```

در مرحله بعد شما باید کلید هایی را که در زمان راه اندازی فرمان unbound-control-setup ایجاد کرده اید را در شاخه /var/unbound/ کپی کنید، شما باید 4 کلید را در شاخه ذکر شده کپی کنید و در اخر هم برای اطمینان از صحت این امر فرمان unbound-checkconf را اجرا کنید، این اعمال در شکل زیر نمایش داده شده است:



```
Terminal
File Edit View Terminal Tabs Help
root@FreeBSD:~ # cd /usr/local/etc/unbound/
root@FreeBSD:/usr/local/etc/unbound # ls
unbound.conf.sample      unbound_control.pem      unbound_server.pem
unbound_control.key      unbound_server.key
root@FreeBSD:/usr/local/etc/unbound # cp unbound_server.key /var/unbound/
root@FreeBSD:/usr/local/etc/unbound # cp unbound_server.pem /var/unbound/
root@FreeBSD:/usr/local/etc/unbound # cp unbound_control.key /var/unbound/
root@FreeBSD:/usr/local/etc/unbound # cp unbound_control.pem /var/unbound/
root@FreeBSD:/usr/local/etc/unbound # unbound-c
unbound-checkconf      unbound-control      unbound-control-setup
root@FreeBSD:/usr/local/etc/unbound # unbound-checkconf
unbound-checkconf: no errors in /var/unbound/unbound.conf
root@FreeBSD:/usr/local/etc/unbound #
```

انجام تنظیمات unbound

بعد از اجرا این بخش شما می توانید از فرمان `unbound-control` استفاده کنید. برای مثال با سوئیچ `status` از این فرمان شما خروجی زیر را مشاهده می کنید:

```
Terminal
File Edit View Terminal Tabs Help
root@FreeBSD:~ # unbound-control status
version: 1.5.7
verbosity: 1
threads: 1
modules: 2 [ validator iterator ]
uptime: 164 seconds
options: control(ssl)
unbound (pid 55723) is running...
root@FreeBSD:~ #
```

نمایش وضعیت سرور با استفاده از فرمان unbound-control

در این فرمان شما وضعیت سرور `unbound` را مشاهده می کنید.

مختصری در مورد فایل `unbound.conf`

این فایل در قالب متغییر و مقدار طراحی شده است و مثل تمام فایل‌های پیکربندی خطوطی که با `#` شروع می شود کامنت بوده و تاثیر در تغییرات سرور نخواهند داشت و برای چک کردن این فایل از فرمان `unbound-checkconf` استفاده کنید. این فایل به صورت کامل دارای توضیحات خوب برای اعمال تغییرات است. در ادامه با چند بخش مهم از تغییرات این فایل آشنا می شوید. این فایل دارای چند بخش اصلی است به نام های `server python remote-control sub-zone` و `forwarder-zone` که بخش به بخش آنها مشخص شده اند و بعد از نام این بخشها به : ختم شده است و شما می توانید تنظیمات خود را اعمال کنید.



بخش اصلی server

سرور unbound برای افزایش امنیت در قالب chroot اجرا می شود و با ناک کاربری خاصی به نام unbound پردازش اصلی خود را ااره اندازی می کند تا در صورت هک شدن برنامه به سیستم شما هکر به کلیه شاخه های سیستم شما دسترسی نداشته باشد و نتواند به سطح دسترسی root برسد، در بخش server از فایل unbound.conf برای تغییر دادن این بخشها متغیرهای در نظر گرفته شده است.

بخش username

در بخش username: شما می توانید نام کاربری که با استفاده از سرور unbound راه اندازی شود را مشخص کنید که به صورت پیش فرض این نام unbound است.

شاخه chroot

در بخش chroot: شما می توانید مسیر نام شاخه chroot را مشخص کنید. این آدرس نباید به / ختم شود و در صورتی که شما قصد نداشته باشید که از این قابلیت استفاده کنید مقدار آنرا "" قرار دهید.

مسیر log file

شما می توانید در بخش logfile: مسیر را که قصد دارید log های برنامه در آن ذخیره شود را تعیین کنید.

مسیر pidfile

هر برنامه ای که در FreeBSD راه اندازی می شود که pid برای خود از سیستم دریافت می کند و این عدد برای هسته سیستم عامل برای برقرای ارتباط با سرور مورد استفاده قرار می گیرد شما می توانید در این بخش مسیر این فایل را تعیین کنید. این فایل شامل عددی است که همان pid تخصیص داده شده به آن می باشد.

بخش directory

شما در این بخش می توانید شاخه کاری برنامه را مشخص کنید که مقدار پیش فرض آن /usr/local/etc/unbound است . شما در این بخش می توانید این مسیر را تغییر دهید.

بخش hide-version

شما می توانید در این بخش ورژن برنامه خود را مخفی کنید که این بخش دارای دو متغیر yes و no است.

بخش تنظیم آدرس Ip ورژن های 4 و 6:

شما می توانید ورژن ip مورد استفاده در unbound را مشخص کنید، برای این کار باید در مقابل دو بخش do-ip4 مقدار yes را قرار دهید تا از ورژن 4 استفاده کند و در صورتی که شما قصد دارید ورژن 6 را هم فعال کنید باید در مقابل do-ip6 مقدار yes را قرار دهید.



بخش do-not-query-localhost

شما می توانید در این بخش تنظیم کنید که به درخواست های local host پاسخ داده شود و یا داده نشود با تعیین کردن دو بخش yes و no ، البته مقدار پیش فرض این بخش yes است و سرور به تمام درخواست های ارسال شده از سمت local host پاسخ می دهد.

بخش interface

شما می توانید در این بخش مشخص کنید که برنامه unbound بر روی کدام کارت های شبکه شما فعال باشد و به کدام sub net شما پاسخ درخواست ها را بدهد.

بخش اصلی Remote Control

شما در این بخش می توانید تنظیمات مربوط به پیکربندی از راه دور را تنظیم کنید و با استفاده از فرمان unbound-control گزارشات و اعمال مدیریت را اعمال کنید در بخش بالا از این مقاله با پیکربندی ساده در این بخش آشنا شدید. در ادامه با زیر شاخه های دیگری از این بخش آشنا می شوید.

بخش control-enable

برای فعال کردن قابلیت کنترل از راه دور شما در ابتدا باید متغیر این بخش را تنظیم کنید که در دو حالت yes و no قرار دارد، البته در همین مقاله روش فعال سازی این بخش به صورت کامل توضیح داده شده است.

بخش control-interface

در این بخش شما می توانید آدرس IP که قصد دارید فقط از طریق آن به درخواست های فرمان unbound-control در شبکه گوش داده شود را مشخص کنید.

بخش control-port

این برنامه به صورت پیش فرض در بر روی پورت 8953 در شبکه به فرمان های ارسال شده گوش می دهد و در این بخش شما می توانید این پورت مدیریت را تغییر دهید، برای اعمال شدن تغییر پورت شما باید برنامه را restart کنید. بخشهایی هم در این قسمت وجود دارد که شما می توانید با استفاده از آنها مسیر فایل های کلید ها را مشخص کنید.

بخش اصلی Forward Zone Options

برای فعال کردن حالت caching در unbound شما باید لیست از سرورهایی را که برنامه unbound باید درخواست ها را به سمت آنها ارسال کند را مشخص کنید. برنامه unbound این قابلیت را دارد که برای هم نام حوزه مورد نظر شما لیستی از DNS سرورها را تعیین کنید و برای کش کردن درخواست های ارسال شده به سمت نام مورد نظر شما از آنها استفاده کند. برنامه unbound به صورت تصادفی از لیست تعیین شده استفاده می کند تا همه درخواست ها به سمت یک سرور ارسال نشود.



نکته :

برای اینکه همه درخواستهای دریافت شده از سمت Client ها به سمت سرور های تعیین شده ارسال شوند شما باید نام "." در بخش name مشخص کنید.

بخش name

در این بخش شما مشخص می کنید که برای یک نام حوزه خاص از که سرورهای برای پرس و جو استفاده شود. شما می توانید برای مثال مشخص کنید که برای نام google از سرورهایی که در بخش forward-addr برای پرس و جو استفاده کند. این دو بخش با هم برای شما حالت کش را فعال می کند و با قرار دادن نام "." و بعد لیست forward-addr ها می توانید حالت کش را فعال کنید.

بخش forward-addr

در این بخش آدرس ip لیست forwarder و مشخص می شود و هر خط شامل یک آدرس IP می باشد. در زیر یک مثال از فایل پیکربندی برای حالت کش کردن برای کاربران شبکه LAN برای شما ارائه شده است:

```
## Simple recursive caching DNS
## unbound.conf
#
server:
    interface: 0.0.0.0
    access-control: 10.0.0.0/16 allow
    access-control: 127.0.0.0/8 allow
    access-control: 192.168.0.0/16 allow
forward-zone:
    name: "."
    forward-addr: 8.8.4.4          # Google
    forward-addr: 8.8.8.8          # Google
    forward-addr: 37.235.1.174     # FreeDNS
    forward-addr: 37.235.1.177     # FreeDNS
    forward-addr: 50.116.23.211    # OpenNIC
    forward-addr: 64.6.64.6        # Verisign
    forward-addr: 64.6.65.6        # Verisign
    forward-addr: 74.82.42.42      # Hurricane Electric
    forward-addr: 84.200.69.80     # DNS Watch
```



```

forward-addr: 84.200.70.40 # DNS Watch
forward-addr: 91.239.100.100 # censurfridns.dk
forward-addr: 109.69.8.51 # puntCAT
forward-addr: 208.67.222.220 # OpenDNS
forward-addr: 208.67.222.222 # OpenDNS
forward-addr: 216.146.35.35 # Dyn Public
forward-addr: 216.146.36.36 # Dyn Public

```

بعد از بخش access-control شما می توانید تعیین کنید که کدام رنج از آدرس های ip می تواند به سمت این سرور درخواست ارسال کنند.

مدیریت کردن استفاده از حافظه سیستم:

یکی از بخش های مهم پیکربندی سرور unbound مدیریت کردن حافظه است، در خطوط زیر شما تنظیمات بخش server را مشاهده می کنید که از طرف تیم ارایه کنند unbound تهیه شده است:

```

# example settings that reduce memory usage
server:
    num-threads: 1
    outgoing-num-tcp: 1 # this limits TCP service, uses less buffers.
    incoming-num-tcp: 1
    outgoing-range: 60 # uses less memory, but less performance.
    msg-buffer-size: 8192 # note this limits service, 'no huge stuff'.
    msg-cache-size: 100k
    msg-cache-slabs: 1
    rrset-cache-size: 100k
    rrset-cache-slabs: 1
    infra-cache-numhosts: 200
    infra-cache-slabs: 1
    key-cache-size: 100k
    key-cache-slabs: 1
    neg-cache-size: 10k
    num-queries-per-thread: 30
    target-fetch-policy: "2 1 0 0 0 0"

```




```
harden-large-queries: "yes"
harden-short-bufsize: "yes"
```

راه اندازی در زمان boot سیستم

برای راه اندازی خودکار در زمان boot شدن سیستم خطوطی را که در شکل زیر نمایش داده شده است را در به فایل rc.conf اضافه کنید:

```
Terminal
File Edit View Terminal Tabs Help
root@FreeBSD:~ # /usr/local/etc/rc.d/unbound rcvar
# unbound
#
unbound_enable="NO"
# (default: "")
root@FreeBSD:~ # █
```

راه اندازی کردن در زمان Boot با استفاده از rc.conf

شما به راحتی می توانید از فرمان های rc موجود در شاخه /usr/local/etc/rc.d/ نسبت به stop start و restart کردن برنامه unbound بپردازید.



فایل zone چیست؟

قبل از راه اندازی DNS سرور در حالت authoritative شما باید روش ایجاد کردن فایل zone را فرا بگیرید. فایل zone یکی از مهمترین بخش های DNS سرور است. این فایل یک فایل متنی است و شما می توانید با ویرایشگر متنی آنرا ایجاد کنید، فقط باید از قواعد آن پیروی کنید و نوع بخشهایی که در آن قابل اضافه شدن و یا اجباری هست را بیاموزید.

انواع رکوردهای موجود در zone:

چند رکورد خاص در zone فایل وجود دارد که در ادامه با آنها آشنا می شوید.

معروفترین و اولین رکورد A است که برای مشخص کردن کلاينت های آدرس IP ورژن 4 مورد استفاده قرار می گیرند.

رکورد AAA برای نمایش کلاينتهای آدرس IP ورژن 6 مورد استفاده قرار میگیرند.

رکورد CNAME برای تعیین کردن یک نام مستعار استفاده می شود.

رکورد MX برای تعیین کردن Mail exchange دامنه استفاده می شود که همان سرور میل است.

از رکورد NS برای مشخص کردن Name Server ها یک دامین استفاده می شود.

رکورد اصلی SOA:

یکی از بخشهای مهم فایل zone رکورد SOA است که باید در همه فایل های zone دامین ها وجود داشته باشد. این رکورد شامل اطلاعات authoritative مثل نام DNS سرور اصلی دامین، آدرس میل مدیری دامین، شماره سریال دامین و آخرین زمان تغییرات در فایل zone را مشخص می کند. این ها در یک خط نوشته می شود به صورت زیر:

```
example.com. IN SOA ns.mabedini.com. admin@example.com. ( 2016120710 1d 2h 4w 1h )
```

شروع یک فایل zone با بخشی شروع می شود به نام TTL@ که در حقیقت time to live است که شما برای مدت زمان Cache شدن اطلاعات دامین خود تعیین می کنید، در بسیاری از موارد سرورهای DNS متغییر های موجود در یک دامنه برای برای کاهش تافیک شبکه cache می کنند که بعد از اتمام مدت زمان تعیین شده در این بخش دوباره درخواست انتقال اطلاعات را به سمت سرور ارسال می کنند، این زمان بسته به نوع zone شما دارد.

نکته:

خطوطی که با علامت ; شروع می شوند توضیحات است و هیچ تاثیری ندارند.

بعد از این خط شما باید رکورد اصلی SOA را تعریف کنید به صورت زیر:

```
abedini.com. IN SOA ns1.mabedini.com. dnsmaster@abedini.com. (
    2016091401 ; Serial
    3H ; refresh after 3 hours
```



```
1H      ; retry after 1 hour
1W      ; expire after 1 week
1D)     ; minimum TTL of 1 day
```

همانطوری که مشاهده می کنید در ابتدا باید نام دامین را مشخص کنید و بعد از SOA باید نام DNS سرور اصلی این دامین را مشخص کنید و بعد از آن هم باید آدرس میل مدیر دامین را بنویسید.

هر دامین باید یک شماره سریال داشته باشد، سرور های دیگری DNS برای بروزرسانی دیتابیس خود ابتدا رکورد SOA را از سرور درخواست می کند و در صورتی که شماره سریال دریافتی با شماره سریالی که در پایگاه داده خود دارند یکسان باشد بدین معناست که نیازی به دریافت تمام فایل zone ندارند و در صورتی که این سریال تغییر کرده باشد از سرور درخواست ارسال کل فایل zone را می کند برای بروزرسانی دیتابیس خودش. بعد از شماره سریال شما باید زمان های cache شدن دامین را مشخص کنید که به نوع سرور شما بستگی دارد .

در بخش بعدی شما حتما باید نام دو یا چند DNS سرور را برای دامین خود مشخص کنید به صورت زیر:

```
; Name Server
IN      NS      ns1.mabedini.com.      ; VeriSign verteilt (anycast)
IN      NS      ns2.mabedini.com.      ; ns.nasa.gov, Mountain View,
```

رکورد برای تعریف کردن این بخش NS است و شما باید دقت کنید که نام ها به . یا همان نقطه ختم می شوند.

در بخش بعدی باید سرور میل خود را مشخص کنید برای این کار باید به صورت زیر عمل کنید :

```
; Mail Exchanger
IN      MX      50 mx1.mail.com. ; Your Mail Server
```

رکورد این بخش MX است.

اگر شما قصد داشته باشید که هاست های WWW و بدون WWW یا همان وب سرور برای دامین خود مشخص کنید باید به صورت زیر این بخش ها را تعریف کنید:

```
abedini.com.      IN A      85.214.123.64
www               IN CNAME  85.214.123.64
```

بعد از نام abedini.com حتما باید نقطه اضافه کنید و نوع آن یک کلاینت با آدرس نوع 4 است و در خط بعدی این client در حقیقت یک نام مستعاری می شود از WWW این دامنه.

برای ایجاد کردن به صورت خودکار فایل zone دامنه خود می توانید به سایت <http://www.zonefile.org> مراجعه کنید و با پر کردن فیلدهای آن یک فایل کامل برای دامین خود ایجاد کنید، صفحه اول این سایت برای شما نمایش داده شده است:



Create new zonefile

www.zonefile.org

HOME - IPv6 - RFCs - CONTACT

Create new zonefile

Base Data

*Domain:

*Adminmail:

\$TTL:

*IP Address or PTR Name:

DNS Server

	Hostname:	IP-Addr:	Comment:
*Primary:	<input type="text" value="a.root-servers.net"/>	<input type="text" value="198.41.0.4"/>	<input type="text" value="VeriSign verteilt (anycast)"/>
Secondary:	<input type="text" value="e.root-servers.net"/>	<input type="text" value="192.203.230.10"/>	<input type="text" value="ns.nasa.gov, Mountain View, Kalif"/>
Nameserver 3:	<input type="text" value="l.root-servers.net"/>	<input type="text" value="199.7.83.42"/>	<input type="text" value="ICANN verteilt (anycast)"/>
Nameserver 4:	<input type="text"/>	<input type="text"/>	<input type="text"/>

Mail Server

	Priority:	Hostname:	Comment:
Primary:	<input type="text" value="50"/>	<input type="text" value="mx1.mail.com"/>	<input type="text" value="Your Mail Server"/>
Secondary:	<input type="text"/>	<input type="text"/>	<input type="text"/>
Mailserver 3:	<input type="text"/>	<input type="text"/>	<input type="text"/>
Mailserver 4:	<input type="text"/>	<input type="text"/>	<input type="text"/>

Time Periods

Refresh: 3H

Waiting for piwik.araxos.net...

سایت ایجاد خودکار فایل zone



راه اندازی NSD در FreeBSD

بعد از نصب کردن unbound که فقط قابلیت Forwarder کردن درخواست های کاربران را دارد و از آن برای cache کردن درخواست ها استفاده می شود نیاز به یک DNS سرور از نوع authoritative است که در این بخش سرور امن NSD را مورد بررسی قرار می دهیم. ساختار فایل پیکربندی و Remote control آن هم شبیه به Unbound است. این سرور فقط به درخواست های Zone خود جواب می دهد. در ادامه با نصب و پیکربندی و راه اندازی آن بیشتر آشنا می شوید.

نصب NSD

برای نصب شما هم می توانید از طریق سیستم ports اقدام کنید هم از طریق سیستم نصب بسته های باینری، در این بخش از روش نصب بسته های باینری استفاده می کنیم:

```

Terminal
File Edit View Terminal Tabs Help
root@FreeBSD:~ # pkg install nsd
Updating FreeBSD repository catalogue...
FreeBSD repository is up-to-date.
All repositories are up-to-date.
The following 1 package(s) will be affected (of 0 checked):

New packages to be INSTALLED:
    nsd: 4.1.12

Number of packages to be installed: 1

The process will require 2 MiB more space.
337 KiB to be downloaded.

Proceed with this action? [y/N]: y
Fetching nsd-4.1.12.txz: 100% 337 KiB 20.3kB/s 00:17
Checking integrity... done (0 conflicting)
[1/1] Installing nsd-4.1.12...
====> Creating groups.
Creating group 'nsd' with gid '216'.
====> Creating users
Creating user 'nsd' with uid '216'.
[1/1] Extracting nsd-4.1.12: 100%
Message from nsd-4.1.12:

```

```

Message from nsd-4.1.12:
*****
*
* To run nsd from startup, add nsd_enable="YES" to your etc/rc.conf
*
* Starting with nsd version 4 the old nsdc control program has been
* replaced by nsd-control. This requires some manual setup with
* nsd-control-setup and editing of the config files.
*
* nsd-control is incompatible with nsdc so when that is used in scripts,
* these should be adapted
*
*****
root@FreeBSD:~ # █

```



مراحل نصب کردن NSD در FreeBSD

بعد از نصب برنامه فایل‌های این برنامه در زیر شاخه `/usr/local/etc/nsd/` ذخیره می‌شود:

```
Terminal
File Edit View Terminal Tabs Help
root@FreeBSD:/usr/local/etc/nsd # ls
nsd.conf          nsd.conf.sample
root@FreeBSD:/usr/local/etc/nsd #
```

فایل پیکربندی NSD

در این شاخه دو فایل وجود دارد که فایل‌های پیکربندی است و شما باید تمام تغییرات مورد نظر خود را در آن اعمال کنید در بخش بعدی شما با بخش‌های مهم فایل `nsd.conf` آشنا می‌شوید.

پیکربندی سرور با فایل `nsd.conf`

برنامه NSD تمام تنظیمات خود را از فایل پیکربندی `nsd.conf` می‌گیرد. این فایل دارای بخش‌های مهمی است که بخش اول آن بخش `Server` است که در آن شما می‌توانید تنظیمات اصلی سرور را تغییر دهید که بخش‌های مهم آن شامل لیست‌های زیر می‌شود:

بخش `ip-address`

در این بخش ما می‌توانید آدرس IP ورژن 4 و 6 که سرور بر روی آن به درخواستها پاسخ دهد را مشخص کنید. با استفاده از `@ports` شما می‌توانید.

بخش `ip-transparent`

این بخش به NSD اجازه می‌دهد که با آدرس‌های IP غیر محلی هم کار کند. این بخش مفید است در زمانی که NSD به یک آدرس IP گوش کند که به کارت شبکه اضافه نشده است و به محض اضافه شدن آن آدرس IP به کارت شبکه فوراً NSD می‌تواند بر روی آن آدرس پاسخگو باشد. پیش فرض این بخش `No` است و این بخش شامل `No` و `Yes` است.

بخش `debug-mode`

این بخش دو حالت `yes` و `no` را دارد و می‌تواند قابلیت دیباگ کردن را فعال و یا غیرفعال کند. این حالت هیچ بار اضافی به پردازش شما نمی‌دهد. به صورت پیش فرض این حالت خاموش است.

بخش `do-ip4`

این بخش شامل دو حالت `yes` و `no` است و اگر `yes` باشد یعنی از ورژن 4 آدرس ip استفاده می‌کند.

بخش `do-ip6`



این بخش شامل دو حالت `yes` و `no` است و اگر `yes` باشد یعنی از ورژن 6 ادرس `ip` استفاده می کند.

بخش database

در مقابل این بخش شما باید مسیر فایل دیتابیس مربوط به برنامه NSD را مشخص کنید که مسیر پیش فرض آن `var/db/nsd/nsd.db/` است. این فایل کامپایل شده اطلاعات `zone` هاست که برای افزایش سرعت سرور استفاده می شود.

بخش zonelistfile

در مقابل این فایل شما باید مسیر فایل `zone.list` را مشخص کنید، این فایل استفاده می شود برای `zone` هایی که به صورت `dynamic` اطلاعات خود را تغییر می دهند. این فایل با استفاده از فرمان `add zone` یا `delet zone` تغییر می کند.

بخش logfile

در این بخش شما باید مسیر فایلی که لاگ برنامه در آن نوشته می شود را مشخص کنید .

بخش server-count

در مقابل این بخش شما باید عددی را مشخص کنید که تعداد سرورهایی را که NSD می تواند راه اندازی کند را مشخص می کند که تعداد پیش فرض آن 1 است.

بخش tcp-count

در مقابل این بخش شما بیشترین تعداد برقراری هم زمان ارتباط TCP با یک سرور را مشخص می کنید.

بخش pidfile

شما در این بخش نام فایلی را که pid برنامه NSD در آن قرار می گیرد را مشخص می کنید.

بخش port

شما در این بخش شماره پورتی را که NSD از طریق آن در شبکه سرویس می دهد را مشخص می کند که پیش فرض آن 53 است.

بخش chroot

در این بخش شما باید مسیر شاخه را مشخص کنید که برنامه NSD را به صورت `chroot` اجرا می کند را مشخص کنید.



بخش username

برای کاهش خطرات حاکی از هک شدن برنامه و محدود کردن سطح دسترسی شما برنامه NSD را باید با نام کاربری راه اندازی کنید که دسترسی ورود به سیستم را نداشته باشد و با ایجاد خطر عملا به سیستم شما آسیبی وارد نشود. هر برنامه در BSD باید از طریق یک نام کاربری اجرا شود تا هسته بتواند آنرا مدیریت کند.

بخش zonesdir

در مقابل این بخش شما باید مسیری که فایل‌های zone در آن ذخیره شده است را مشخص کنید که مسیر پیش فرض آن `/etc/nsd` است.

بخش hide-version

این بخش شامل `yes` و `no` است که پیش فرض آن `no` است.

بخش zonefiles-check

این بخش شامل دو گزینه `yes` و `no` است، این بخش باعث می شود که در صورتی که NSD راه اندازی مجدد شده است در زمان راه اندازی فایل‌های zone را چک کند. پیش فرض این بخش `yes` تعریف شده است.

مدیریت NSD با nsd-control

برای مدیریت کردن سرور NSD برنامه ای وجود دارد به نام `nsd-control` که به صورت پیش فرض غیرفعال است و در زمان فعال شدن هم فقط به همان سیستمی که NSD بر روی آن نصب است پاسخگو است. با این فرمان شما می توانید شما می توانید سرور را `stop` و `start` کنید، فایل پیکربندی را دوباره بارگذاری کنید، فایل `zone` اضافه و پاک کنید و از همه مهم تر گزارش کاملی از سرور خود را مشاهده کنید.

برای استفاده از فرمان شما باید در ابتدا فرمان `nsd-control-setup` را راه اندازی کنید تا کلید و `certification` های مورد نیاز برای برقرار `tls` بر روی `TCP` ایجاد شود، بعد از آن باید در فایل `nsd.conf` بخش `remote control` را فعال کنید و بعد کلید های و `certification` های مورد نیاز را در بخش مشخص در `nsd.conf` کپی کنید در ادامه با مراحل راه اندازی این سرویس آشنا می شوید. در بخش اول فرمان `nsd-control-setup` را راه اندازی کنید

برای مشاهده کردن مراحل نصب و راه اندازی به دلیل استفاده از فایل `gif` به بخش مقالات سایت مراجعه کنید.



اضافه کردن Zone در NSD

بعد از انجام مراحل نصب و راه اندازی nsd dns server حال زمان اضافه کردن zone به این سرور شده است. برای این کار باید به فایل اصلی nsd.conf مراجعه کنید و بخشهایی را در آن فایل اضافه کنید، یکی از بخش های مهم داشتن یک فال zone درست است که در مقاله فایل zone چیست با آن آشنا می شوید.

در فایل nsd.conf بخشی برای اضافه کردن zone در نظر گرفته شده است که با نام zone شروع می شود و شما باید بخش های مورد نظر را در آن اضافه کنید. هر zone باید با این بخش شروع شود و در بخش بعدی باید نام zone را مشخص کنید، نام را باید بعد از name: اضافه کنید و در مرحله بعد هم باید مسیر فایل zone را مشخص کنید برای اینکار باید از بخش zonefile: استفاده کنید و مسیر را در مقابل آن اضافه کنید، در شکل زیر یک مثال ساده از اضافه کردن zone را مشاهده می کنید:

```

Terminal
File Edit View Terminal Tabs Help
^[(escape) menu ^y search prompt ^k delete line ^p prev li ^g prev page
^o ascii code ^x search ^l undelete line ^n next li ^v next page
^u end of file ^a begin of line ^w delete word ^b back 1 char
^t top of text ^e end of line ^r restore word ^f forward 1 char
^c command ^d delete char ^j undelete char ^z next word
=====line 309 col 0 lines from top 309 =====
# RRLconfig
# Response Rate Limiting, whitelist types
# rrl-whitelist: nxdomain
# rrl-whitelist: error
# rrl-whitelist: referral
# rrl-whitelist: any
# rrl-whitelist: rrsig
# rrl-whitelist: wildcard
# rrl-whitelist: nodata
# rrl-whitelist: dnskey
# rrl-whitelist: positive
# rrl-whitelist: all
# RRLend
zone:
name: abedini.com
zonefile: /usr/local/etc/nsd/zone/abedini.com

```

اضافه کردن فایل Zone

حال با استفاده از فرمان nsd-control فایل nsd.conf را دوباره بارگذاری کنید و یا سرور را restart کنید به صورت زیر:

```

Terminal
File Edit View Terminal Tabs Help
root@FreeBSD:/usr/local/etc/nsd/zone # nsd-control reload
ok
root@FreeBSD:/usr/local/etc/nsd/zone #

```



بارگذاری مجدد تنظیمات NSD

در مقاله مدیریت NSD با nsd-control روش ایجاد و استفاده کردن از این فرمان کامل شرح داده شده است.

در مرحله بعدی با استفاده از فرمان nslookup از صحت کارایی سرور مطمئن شوید به شرح زیر:

```

Terminal
File Edit View Terminal Tabs Help
root@FreeBSD:~ # nslookup
> lserver 192.168.233.2
Default server: 192.168.233.2
Address: 192.168.233.2#53
> abedini.com
Server:          192.168.233.2
Address:         192.168.233.2#53

Non-authoritative answer:
Name:   abedini.com
Address: 213.186.33.19
> mail.abedini.com
Server:          192.168.233.2
Address:         192.168.233.2#53

Non-authoritative answer:
mail.abedini.com canonical name = ns0.ovh.net.
Name:   ns0.ovh.net
Address: 213.186.33.20
>

```

چک کردن تنظیمات

با استفاده از فرمان dig هم می توانید به صورت زیر اطلاعات را مشاهده کنید:



```

Terminal
File Edit View Terminal Tabs Help
root@FreeBSD:~ # dig abedini.com any @192.168.233.2 | less

; <<>> DiG 9.10.3-P4 <<>> abedini.com any @192.168.233.2
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 63864
;; flags: qr rd ra; QUERY: 1, ANSWER: 7, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; MBZ: 0005 , udp: 512
;; QUESTION SECTION:
;abedini.com.                IN      ANY

;; ANSWER SECTION:
abedini.com.                5      IN      SOA     dns.ovh.net. tech.ovh.net. 20160
90702 86400 3600 3600000 86400
abedini.com.                5      IN      NS      ns.ovh.net.
abedini.com.                5      IN      NS      dns.ovh.net.
abedini.com.                5      IN      MX      1 mx0.ovh.net.
abedini.com.                5      IN      MX      100 mxb.ovh.net.
abedini.com.                5      IN      A       213.186.33.19
abedini.com.                5      IN      TXT     "v=spf1 include:mx.ovh.com ~all"

;; Query time: 550 msec

```

استفاده کردن از فرمان dig

سرور NSD در دو حالت master و secondary

در بسیاری از روشها استفاده از سرور DNS به دلیل اهمیت زیاد در همیشه در دسترس بودن این سرویس برای شبکه DNS سرورها در حالت master و secondary دارند که در صورتی که سرور اصل نتواند پاسخگوی درخواستها باشد سرور دیگری با همان zone ها بتواند به سرویس دهی بپردازد. در zone های که پویا هستند اطلاعات zone در حال تغییر می باشد این امر بسیار حیاتی است که دو یا چند سرور در شبکه شما وجود داشته باشد. در مورد zone های که پویایی زیادی ندارند شما می توانید دو سرور جداگانه ایجاد کنید، DNS سرور NSD هم می تواند هر دو نقش master و secondary را ایفا کند، برای اینکار باید در زمان تعریف zone این نوع را مشخص کنید در مثال زیر در بخش اول سرور master را تعریف کرده و در بخش دوم هم باید در سرور secondary دامین خود را به صورت زیر تعریف کنید:

سرور Master:

```

zone:

# this server is master, 10.0.2.1 is the secondary.
name: masterzone.com
zonefile: /etc/nsd/masterzone.com.zone
notify: 10.0.2.1 NOKEY
provide-xfr: 10.0.2.1 NOKEY

```



سرور Secondary:

```
zone:  
  
# this server is secondary, 10.0.2.2 is master.  
name: secondzone.com  
zonefile: /etc/nsd/secondzone.com.zone  
allow-notify: 10.0.2.2 NOKEY  
request-xfr: 10.0.2.2 NOKEY
```



راه اندازی DHCP سرور در FreeBSD

سرور DHCP یا همان dynamic Host Configurations protocol به کاربران متصل شده به شبکه شما این اجازه می دهد که به صورت خودکار تنظیمات خاص شبکه شما را دریافت و از منابع موجود در شبکه استفاده کند. این سرور در دو بخش Client و server ارایه می شود، در بخش client در سیستم عامل FreeBSD از برنامه dhcpclient موجود در OpenBSD استفاده می کند و برای دریافت اطلاعات از سرور DHCP از این برنامه استفاده می کند.

در بخش سروری در FreeBSD این برنامه به صورت پیش فرض نصب نشده است و تعداد زیادی برنامه سروری با قابلیت های خاصی در بخش سیستم ports وجود دارد که هر کدام قابل نصب و راه اندازی است. در ادامه یکی از برنامه های سروری را با هم نصب و پیکربندی می کنیم.

نصب و پیکربندی DHCP سرور

در این بخش برای نصب و پیکربندی DHCP از سروری که توسط Internet System Consortium ایجاد و ارایه شده است استفاده می کنید که در بخش سیستم ports از زیر شاخه net/isc-dhcp43-server برای نصب و راه اندازی استفاده می کنیم. این برنامه را هم شما می توانید هم از طریق سیستم pkg هم نصب کنید

بعد از انجام دادن مراحل نصب به صورت خودکار فایل پیکربندی پیش فرضی به نام dhcp.conf.example در زیر شاخه /usr/local/etc ایجاد شده است که با استفاده از فرمان cp به صورت زیر نام این فایل را به dhcp.conf تغییر دهید:

```
#cp /usr/local/etc/dhcp.conf.example /usr/local/etc/dhcp.conf
```

فایل پیکربندی dhcp.conf

در این فایل دو بخش برای اعمال تنظیمات وجود دارد، در بخش اول تنظیمات دامین سیستم، DNS که کاربر اطلاعات نام سایت ها را از آن پرس و جو کند و مدت زمان در اختیار داشتن آدرس و sub net mask کلاینت ها را در آن تنظیم می کنید این بخشها را در زیر مشاهده می کنید:

```
option domain-name "mabedini.ir";
option domain-name-servers 192.168.1.100;
option subnet-mask 255.255.255.0;

default-lease-time 600;
max-lease-time 72400;
```

نکته:

تمام خطوط در فایل dhcp.conf به علامت ; ختم می شود، به استفاده از این علامت توجه کنید.



بخش بعدی شما باید اطلاعات آدرس ip که قصد دارید در اختیار کاربران شبکه خود قرار دهید را مشخص کنید در این بخش شما با استفاده از این فایل به صورت زیر این عمل را انجام دهید:

```
subnet 192.168.1.0 netmask 255.255.255.0 {
    range 192.168.1.50 192.168.1.150;
    option routers 192.168.1.1, 192.168.1.2;
}
```

همانطوری که مشاهده می کنید در ابتدا بخش شبکه ای که قصد دارید از آن استفاده کنید را مشخص کنید و در بخش دوم subnetmask هر کاربر را مشخص کنید. در ادامه باید یک رنج از آدرس که قصد اجاره دادن آنرا دارید را مشخص کنید. در ادامه default gateway کلاینت ها را مشخص کنید. در این بخش ما می توانیم چند default route مشخص کنید و برای جدا کردن آنها از علامت , استفاده کنید.

راه اندازی سرویس dhcp

از آنجایی که سرور DHCP را از طریق نصب بسته به سیستم عامل FreeBSD اضافه کرده اید این برنامه فایل های پیکربندی و راه اندازی را در شاخه /usr/local/ ذخیره می کند و برای راه اندازی آن هم باید از شاخه dc.d واقع در آدرس /usr/local/etc/ استفاده کنید، برای راه اندازی به صورت خودکار هم باید از فایل /etc/rc.conf استفاده کنید. بعد اضافه کردن خط زیر به فایل rc.conf حال شما می توانید از فرمانهای rc.d استفاده کنید به صورت زیر:

```
dhcpcd_enable="YES"
dhcpcd_ifaces="em0"
```

اگر هم قصد راه اندازی یکبار این سیستم را دارید از سیستم onestart استفاده کنید به صورت زیر:

```
Terminal
File Edit View Terminal Tabs Help
root@FreeBSD:~ # /usr/local/etc/rc.d/isc-dhcpd onstart
Starting dhcpd.
Internet Systems Consortium DHCP Server 4.3.4
Copyright 2004-2016 Internet Systems Consortium.
All rights reserved.
For info, please visit https://www.isc.org/software/dhcp/
Config file: /usr/local/etc/dhcpd.conf
Database file: /var/db/dhcpd/dhcpd.leases
PID file: /var/run/dhcpd/dhcpd.pid
Wrote 0 class decls to leases file.
Wrote 0 deleted host decls to leases file.
Wrote 0 new dynamic host decls to leases file.
Wrote 0 leases to leases file.
Listening on BPF/em0/00:0c:29:fc:a8:54/192.168.233.0/24
Sending on BPF/em0/00:0c:29:fc:a8:54/192.168.233.0/24
Sending on Socket/fallback/fallback-net
root@FreeBSD:~ #
```

نمایش حالت راه اندازی OneStart



بعد از اجرا کردن این فرمان و دریافت آدرسهای IP از سمت سرور فایلی در زیر شاخه `/var/db/dhcpd` وجود دارد به نام `dhcp.leases` که اطلاعات آدرس های ارایه شده به سیستم ها را نمایش می دهد محتوای این فایل را در شکل زیر مشاهده می کنید:

```
Terminal
File Edit View Terminal Tabs Help
root@FreeBSD:~/gif/dhcpserver # cat /var/db/dhcpd/dhcpd.leases
# The format of this file is documented in the dhcpd.leases(5) manual page.
# This lease file was written by isc-dhcp-4.3.4

# authoring-byte-order entry is generated, DO NOT DELETE
authoring-byte-order little-endian;

server-uid "\000\001\000\001\037\207Kz\000\014)\374\250T";

lease 192.168.233.200 {
  starts 3 2016/10/05 05:31:05;
  ends 3 2016/10/05 05:41:05;
  cltt 3 2016/10/05 05:31:05;
  binding state active;
  next binding state free;
  rewind binding state free;
  hardware ethernet 00:0c:29:fc:a8:5e;
  uid "\001\000\014)\374\250^";
  client-hostname "FreeBSD";
}
root@FreeBSD:~/gif/dhcpserver #
```

نمایش وضعیت آدرسهای اختصاص داده شده



راه اندازی time سرور در FreeBSD

شاید برای اولین بار باشد که با مفهوم time server آشنا شده باشید اگر از سیستم عامل ویندوزی به سمت FreeBSD مهاجرت کرده باشید چنین سروری را در سیستم خود ندارد ولی در FreeBSD یکی از مهمترین سرورهای شبکه است، اهمیت زمان در شبکه از آنجا مشخص می شود که شما در شبکه خود در بین چندین سرور قصد دارد که واقعه ای که در زمان خاصی اتفاق افتاده باشد را پیگیری کنید، حال اگر زمان سرورهای شما یکی نباشد شما به چه صورتی می توانید آنها چک کنید؟ در این بین سرور را نیاز دارید که زمان را برای شما از سرورهای اصلی در اینترنت پرس و جو کرده و در اختیار سرور ها و سیستم های موجود در شبکه قرار بدهند. در FreeBSD شما از سرویس ntime می توانید برای انجام دادن این عمل استفاده کنید. این پروتکل شبکه هم دو سوپیه است هم قابلیت سروری دارد هم قابلیت کلاینتی، در ادامه اول با بخش کلاینتی آشنا می شود و در بخش دوم هم قابلیت سروری این سرویس را مورد بررسی قرار می دهیم.

بروز رسانی کردن زمان سیستم

به صورت پیش فرض برنامه ای به نام ntpd برای تنظیم کردن ساعت سیستم شما با time سرورهای موجود، برای فعال کردن این سرویس در زمان راه اندازی در فایل /etc/rc.conf باید خط ntpd_enable="YES" را وارد کنید. این برنامه به صورت خودکار فایل /etc/ntp.conf را برای پیدا کردن سرور ها جستجو می کند. در این فایل بخشهای مختلفی وجود دارد که برای جستجو از سرورهای time در این فایل باید بخش server را ویرایش و مشاهده کنید در شکل زیر این بخش برای شما نمایش داده شده است:

```

Terminal
File Edit View Terminal Tabs Help
^[(escape) menu ^y search prompt ^k delete line ^p prev li ^g prev page
^o ascii code ^x search ^l undelete line ^n next li ^v next page
^u end of file ^a begin of line ^w delete word ^b back 1 char
^t top of text ^e end of line ^r restore word ^f forward 1 char
^c command ^d delete char ^j undelete char ^z next word
=====line 32 col 0 lines from top 32 =====
# See http://www.pool.ntp.org/ for details. Note, the pool encourages
# users with a static IP and good upstream NTP servers to add a server
# to the pool. See http://www.pool.ntp.org/join.html if you are interested.
#
# The option `iburst' is used for faster initial synchronization.
#
server 0.freebsd.pool.ntp.org iburst
server 1.freebsd.pool.ntp.org iburst
server 2.freebsd.pool.ntp.org iburst
#server 3.freebsd.pool.ntp.org iburst
#
# If you want to pick yourself which country's public NTP server
# you want sync against, comment out the above servers, uncomment
# the next ones and replace CC with the country's abbreviation.
# Make sure that the hostnames resolve to a proper IP address!
#

```

نمایش فایل ntp.conf



اگر در بخش مقابل نام سرور عبارت **prefer** را اضافه کنید این سرور به عنوان اولین سرور مورد بررسی قرار می گیرد. سعی کنید در این بخش سروری را انتخاب کنید تا قدرت بالایی را داشته باشد. برای دریافت لیستی از سرورهای **time** موجود از سایت <http://www.ntp.org> استفاده کنید و به بخش **Public Time Server Lists** وارد شوید تا صفحه ای به صورت زیر برای شما باز شود:

The screenshot shows the ntp.org website interface. The main content area displays a security notice: "NTP users are strongly urged to take immediate action to ensure that their NTP daemons are not susceptible to being used in distributed denial-of-service (DDoS) attacks. Please also take this opportunity to defeat denial-of-service attacks by [implementing Ingress and Egress filtering through BCP38](#)." Below this, it mentions "ntp-4.2.8p8 Was released on 02 June 2016. It addresses 1 high- and 4 low--severity security issues, 4 bugfixes, and contains other improvements over 4.2.8p7." A link to "NTP Security Notice" is provided. At the bottom, there is a question: "Are you using Autokey in production? If so, please contact Harlan - he's got some questions for you." The left sidebar contains navigation links like "Servers Web", "Rules Of Engagement", "NTP Pool Servers", etc.

سایت ntp.org

حال به تناسب منطقه خود لیست سرور های خود را پیدا کنید برای مثال برای آسیا لیستی به صورت زیر مشاهده می کنید:

Asia — asia.pool.ntp.org

To use this pool zone, add the following to your ntp.conf file:

```
server 0.asia.pool.ntp.org
server 1.asia.pool.ntp.org
server 2.asia.pool.ntp.org
server 3.asia.pool.ntp.org
```

لیست سرورهای زمان بخش آسیا

برنامه ntpdate

برای تنظیم کردن سریع زمان سیستم خود از فرمان **ntpdate** استفاده کنید این فرمان در دو حالت استفاده می شود. در مرحله اول اگر در مقابل این فرمان سروری را مشاهده نکرده باشید از فایل **ntp.conf** برای انتخاب کردن سرور استفاده می کند و در بخش دوم شما می توانید هر سروری را که دوست دارید در مقابل این فرمان وارد کنید تا از طریق آن زمان را جستجو کند به صورت زیر:



```

Terminal
File Edit View Terminal Tabs Help
root@FreeBSD:~ # date
Thu Oct 13 17:21:09 IRST 2016
root@FreeBSD:~ # ntpdate 0.asia.pool.ntp.org
13 Oct 17:36:28 ntpdate[943]: step time server 203.158.118.2 offset 890.546146 s
ec
root@FreeBSD:~ # date
Thu Oct 13 17:36:31 IRST 2016
root@FreeBSD:~ # █

```

تنظیم زمان درست سیستم با استفاده از فرمان `ntpdate`

همانطوری که مشاهده می کنید در اجرای فرمان `date` اول زمان سیستم 17:21 دقیقه است که بعد از راه اندازی `ntpdate` و سرور آسیا به نام `0.asia.pool.ntp.org` زمان سیستم در فرمان دوم `date` به زمان مناسب و درست سیستم تغییر کرده است.

راه اندازی سرور `OpenNTPd`

برای اینکه بخش سروری باید از برنامه ای به نام `OpenNTPd` که در مجموعه پروژه های `OpenBSD` ایجاد شده است نصب و راه اندازی کنید، این برنامه در همه ساختارها به ویژه `FreeBSD` قابل نصب و اجراست برای نصب باید از طریق سیستم `ports` در `FreeBSD` اقدام کنید و در مرحله اول وارد شاخه `/usr/ports/net/openntpd` وارد شوید و با دو فرمان `make` و `make install` این برنامه را نصب کنید در انتها پیغام هایی به صورت زیر مشاهده می کنید:

```

Terminal
File Edit View Terminal Tabs Help
===> Registering installation for openntpd-5.7p4_1,2
Installing openntpd-5.7p4_1,2...
===> Creating users and/or groups.
Creating group '_ntp' with gid '123'.
Creating user '_ntp' with uid '123'.
===> SECURITY REPORT:
This port has installed the following files which may act as network
servers and may therefore pose a remote security risk to the system.
/usr/local/sbin/ntpctl
/usr/local/sbin/ntpd

This port has installed the following startup scripts which may cause
these network services to be started at boot time.
/usr/local/etc/rc.d/openntpd

If there are vulnerabilities in these programs there may be a security
risk to the system. FreeBSD makes no guarantee about the security of
ports included in the Ports Collection. Please type 'make deinstall'
to deinstall the port if this is a concern.

For more information, and contact details about the security
status of this software, see the following webpage:
http://www.openntpd.org/
root@FreeBSD:/usr/ports/net/openntpd # █

```

نمایش بخشهای نصب `OpenNTPD` در `FreeBSD`



این برنامه در زیر شاخه `/usr/local` نصب شده و فایل `ntpd.conf` در زیر شاخه `/usr/local/etc` قرار می گیرد و برای راه اندازی از طریق سیستم `rc.d` از فرمان `/usr/local/etc/rc.d/openntpd` استفاده کنید در فرمان هم به سیستم به نام های `ntpd` و `ntpdctl` و به سیستم اضافه می شود که شما می توانید از طریق آن `openntpd` را مدیریت کنید در ادامه با روش استفاده از `openntpd` آشنا می شوید.

راه اندازی از طریق `rc.conf`

برای راه اندازی `openntpd` از طریق سیستم `rc.d` شما باید خط زیر را در فایل `rc.conf` اضافه کنید تا بتوانید با استفاده از فرمان ای `rc.d` این سرویس را مدیریت کنید، این بخش را در شکل زیر مشاهده می کنید:

```
Terminal
File Edit View Terminal Tabs Help
root@FreeBSD:/usr/local/etc # /usr/local/etc/rc.d/openntpd rcvar
# openntpd
#
openntpd_enable="NO"
# (default: "")
root@FreeBSD:/usr/local/etc # █
```

تنظیمات راه اندازی سرور NTP در فایل `rc.conf`

فایل `ntpd.conf`

فایل پیکربندی این سرور در زیر شاخه `/usr/local/etc` به نام `ntpd.conf` قرار دارد. در این فایل خطوطی که با `#` شروع می شود توضیحات است و این فایل به صورت متغییر و مقدار است. در بخش اول در این فایل باید آدرس `ip` که سرور باید بر روی آن به درخواستها پاسخ دهد را مشخص کنید، این بخش با کلید `listen on` شروع می شود و در خط زیر شما نمونه از این پیکربندی را مشاهده می کنید:

```
Listen on *
Or
Listen on 127.0.0.1
```

در بخش اول یعنی بر روی هر آدرس `ip` به ارایه سرویس پردازش و در بخش دوم فقط بر روی کارت شبکه `localhost` به درخواستها پاسخ می دهد این بدان معناست که فقط به درخواست هایی که از سمت همین سیستم عامل ارسال می شود پاسخ دهد.

در مرحله بعد شما باید سرور هایی که از آنها زمان را سوال کند را مشخص کنید در این بخش شما هم می توانید از نام آن سرور و هم آدرس IP آن استفاده کنید، این بخش با `server` شروع می شود به صورت زیر:

```
Server 10.0.2.2
Or
Server pool.ntp.org
```



حال سرور خود را راه اندازی کنید به صورت زیر:

```
Terminal
File Edit View Terminal Tabs Help
root@FreeBSD:~ # /usr/local/etc/rc.d/openntpd start
Starting openntpd.
root@FreeBSD:~ # sockstat -l4
USER      COMMAND  PID  FD  PROTO  LOCAL ADDRESS  FOREIGN ADDRESS
_ntp      ntpd     4193  7  udp4   192.168.85.128:123  *: *
_ntp      ntpd     4193  10  udp4   127.0.0.1:123    *: *
root      sendmail  799   3  tcp4   127.0.0.1:25     *: *
root      sshd     796   4  tcp4   *:22             *: *
root      syslogd  576   7  udp4   *:514            *: *
```

راه اندازی کردن سرور و نمایش وضعیت پورتهای باز

برای نمایش اطلاعات بیشتر و گزارش بهتر است که از `flags` های `-s` و `-v` در بخش فایل `rc.conf` به صورت زیر اضافه کنید:

```
# openntpd
#
openntpd_enable="YES"
openntpd_flags="-s -v"
# (default: "")
root@FreeBSD:/var/run #
```

تنظیمات بیشتر در فایل `rc.conf`

برای نمایش وضعیت سرور باید از فرمان `ntpctl` استفاده کنید در شکل زیر شما `peers` های که سرور شما از آنها زمان را دریافت می کنند مشاهده می کنید به صورت زیر:

```
Terminal
File Edit View Terminal Tabs Help
root@FreeBSD:/var/run # ntpctl -s peers
peer
  wt tl st next poll      offset      delay      jitter
129.250.35.251 0.asia.pool.ntp.org
  1 10 2 9s 33s 503.772ms 668.605ms 662.320ms
5.160.24.41 from pool pool.ntp.org
  1 2 3 35s 300s ---- peer not valid ----
46.209.15.97 from pool pool.ntp.org
  1 10 3 18s 30s 590.247ms 674.154ms 753.122ms
46.209.14.1 from pool pool.ntp.org
  1 10 3 2s 30s 526.728ms 609.907ms 681.053ms
194.225.50.25 from pool pool.ntp.org
  1 10 2 22s 32s 591.769ms 675.372ms 756.601ms
root@FreeBSD:/var/run #
```

نمایش `peers` در خروجی فرمان `ntpctl`



حال در بخش پایانی برای دریافت اطلاعات از طریق سرور خود با استفاده از فرمان ntpdate به صورت زیر فرمان را با استفاده از آدرس ip سرور خود اجرا کنید:

```

Terminal
File Edit View Terminal Tabs Help
root@FreeBSD:/var/run # ntpdate -dv 192.168.85.128
14 Oct 00:43:19 ntpdate[4425]: ntpdate 4.2.8p8-a (1)
transmit(192.168.85.128)
receive(192.168.85.128)
transmit(192.168.85.128)
receive(192.168.85.128)
transmit(192.168.85.128)
receive(192.168.85.128)
transmit(192.168.85.128)
receive(192.168.85.128)
192.168.85.128: Server dropped: Leap not in sync
server 192.168.85.128, port 123
stratum 3, precision -23, leap 11, trust 000
refid [192.168.85.128], delay 0.02567, dispersion 0.00136
transmitted 4, in filter 4
reference time:      dbaa770a.8fc9cfff  Fri, Oct 14 2016  0:41:38.561
originate timestamp: dbaa7776.58a937ff  Fri, Oct 14 2016  0:43:26.346
transmit timestamp:  dbaa7775.64502f4f  Fri, Oct 14 2016  0:43:25.391
filter delay:  0.02567  0.02579  0.02583  0.02583
                0.00000  0.00000  0.00000  0.00000
filter offset:  0.957379  0.956383  0.955363  0.954366
                0.000000  0.000000  0.000000  0.000000
delay 0.02567, dispersion 0.00136
offset 0.957379

```

نمایش مراحل تنظیم کردن زمان

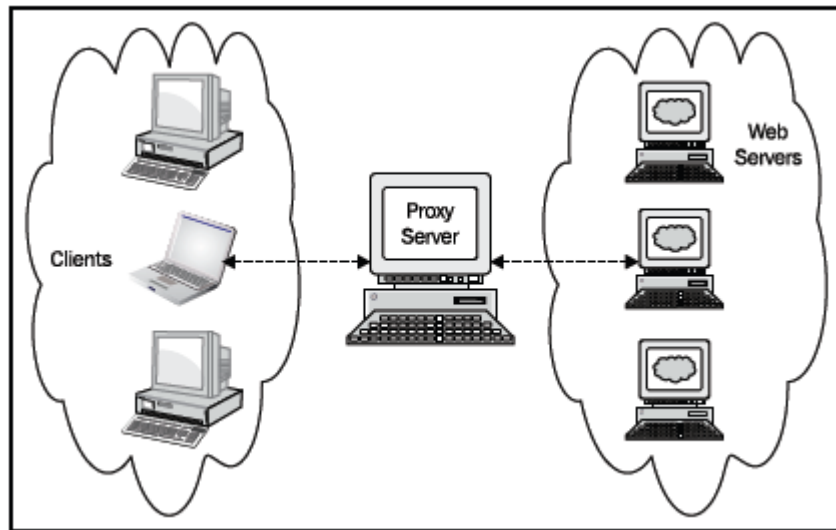


Squid چیست؟

Proxy چیست؟

سرور Proxy سیستمی است که در بین درخواست وب یک کلاینت و سرور مورد نظر وب مورد درخواست قرار می گیرد و به ذخیره کردن مطالب و درخواستها می پردازد. سرور های Proxy بدون تغییر دادن درخواست ها بین کلاینت و سرور عمل می کنند و در حقیقت در میان این ارتباط قرار می گیرند و خود را به جای کلاینت برای سرور و سرور برای کلاینت معرفی می کنند. نوع از proxy سرور به نام transparent وجود دارد که کلاینت از ظهور آن در شبکه اطلاعی ندارد و تصور می کند که خود اقدام به برقرار ارتباط می پردازد.

در حالت پیشرفته از proxy سرور ، این سرور درخواست های ارسال شده را بر اساس یک سری قواعد فیلتر کرده و به درخواستهایی که با قواعد آن مطابقت دارد اجازه خارج شدن از شبکه را می دهد. این قواعد می تواند بر اساس آدرس IP پروتکل های شبکه و نوع مطالب درخواست شده از طرف کلاینت باشد.



نمایش مفهوم پراکسی سرور

همانطوری که در شکل بالا مشاهده می کنید این سرور در بین Client و Web Servers قرار می گیرد و کلاینتها نمی توانند به صورت مستقیم با سرورهای وب ارتباط برقرار کنند. در برخی از انواع proxy ها این سرور در درخواستهای ارسال شده هم دخل و تصرف می کند و گزارشات ارسال شده از سمت کلاینتها را هم ذخیره می کند.

در اصل این سرور ها برای مدیریت پهنای باند، افزایش سرعت بازکردن صفحه های وب درخواستی از سمت کلاینت، مدیریت دسترسی، مشاهده کردن گزارش ترافیک های مصرف شده از سمت کلاینتها و مدیریت ترافیک ارسال شده سمت وب سرورهای مختلف، استفاده از چند ارتباط اینترنت به صورت همزمان است.

یکی از proxy server ها معروف و قدیمی Squid است که در ادامه با آن آشنا می شوید.



دریافت Squid

سرور Squid اولین بار در سال 1996 ارایه شد و تا این لحظه با ورژن ها و قابلیت های مختلفی به صورت رایگان و Open Source در سایت <http://www.squid-cache.org> برای دانلود کردن در دسترس است. این برنامه در قالبهای مختلف قابل دانلود است هم در قالب کد برنامه و هم در قالب برنامه های باینری و از پیش نصب شده برای همه سیستم عاملها مثل Linux و FreeBSD شما به راحتی می توانید در FreeBSD هم از بسته های باینری برای نصب استفاده کنید. ولی خود پروژه به شما پیشنهاد می کند که از کد مبدا Squid را نصب کنید.

قبل از شروع به نصب باید در مورد ورژنهایی که برای Squid وجود دارد توضیحاتی داده شود، برای دریافت آخرین اطلاعات در مورد ورژن های موجود از لینک <http://www.squid-cache.org/Versions/> این بخش در شکل زیر نمایش داده شده است:

Source Code Packages

These allow you to customize your Squid installation when you compile it.

After downloading, refer to [Compiling Squid](#) for assistance with compiling the source code.

Stable Versions:

Current versions suitable for production use.

Version	Latest Release	
3.5	09 Oct 2016	3.5.22
langpack	17 Aug 2016	ROLLING

Beta Versions:

Meant for testing the next release before it is ready for production use.

Suitable for making upgrade plans and similar activities.

Version	Latest Release	
4.0	09 Oct 2016	4.0.15



Development Versions:

Meant for Squid users who are already familiar with Squid. You should expect to find numerous bugs and problems. We do not recommend running a development release on your production cache. If you have any problems with a development release please write to our squid-bugs@squid-cache.org or squid-dev@squid-cache.org lists. DO NOT write to squid-users with code-related problems.

Version	Development Start Date	Development Plans
4	09 Oct 2014	Squid RoadMap

If you are a developer, or want to closely track the source code, repository and rolling release services are outlined in the [Developer Resources](#).

Old Versions:

Provided for archival purposes only. Not intended for general use in new installations.

Version	First STABLE release Date	Latest Release	Latest Release Date
3.4	09 Dec 2013	3.4.14	01 Aug 2015
3.3	09 Feb 2013	3.3.14	01 May 2015
3.2	14 Aug 2012	3.2.14	01 May 2015
3.1	09 Dec 2010	3.1.22	09 Jul 2013

نمایش ورژنهای موجود squid

چهار ورژن وجود دارد Stable ، Beta ، Development Version و Old Versions. برای نصب در محیط های حساس از ورژن Satable استفاده کنید و در بعضی از موارد هم ورژنهای قدیمی توصیه می شود. بعد از انتخاب کردن ورژن مورد نظر و کلیک کردن بروی آن وارد صفحه ای می شوید که می توانید ورژن که نیاز دارید دانلود کنید به صورت زیر:



Squid version 3.4

Release	Date	diff	Download
Latest 3.4 series release			
squid-3.4.14	01 Aug 2015	diff (sig)	tar.gz (sig) / tar.bz2 (sig) / tar.xz (sig)
See langpack for latest Language Package			
Daily auto-generated release. This is the most recent bug-fixed update to the formal release. see Change details for the fixes included in this bundle.			
squid-3.4.14-20160509-r13240	10 May 2016		tar.gz / tar.bz2
squid-3.4.14-20160508-r13239	09 May 2016		tar.gz / tar.bz2
squid-3.4.14-20160506-r13238	07 May 2016		tar.gz / tar.bz2
squid-3.4.14-20160502-r13236	03 May 2016		tar.gz / tar.bz2
squid-3.4.14-20160420-r13235	21 Apr 2016		tar.gz / tar.bz2
squid-3.4.14-20160331-r13232	01 Apr 2016		tar.gz / tar.bz2
squid-3.4.14-20160221-r13231	22 Feb 2016		tar.gz / tar.bz2
squid-3.4.14-20160212-r13230	13 Feb 2016		tar.gz / tar.bz2
squid-3.4.14-20151210-r13229	11 Dec 2015		tar.gz / tar.bz2
squid-3.4.14-20150901-r13228	02 Sep 2015		tar.gz / tar.bz2
squid-3.4.14-20150801-r13227	02 Aug 2015		tar.gz / tar.bz2
Squid 3.4 BTP		Source repo	Launchpad Mirror

آخرین ورژنهای موجود از برنامه Squid

حال ورژن مورد نظر خود را دانلود کنید تا وارد مرحله نصب شوید. در بخش بعدی با نصب squid ادامه می دهیم.



نصب Squid

برای نصب از طریق کد مبدا شما نیاز به کامپایلر دارید که Squid به زبان C/C++ نوشته شده است در نتیجه شما نیاز به این کامپایلر دارید، سیستم عامل FreeBSD شامل کامپایلر C/C++ است در نتیجه شما برای نصب مشکلی نخواهید داشت.

در قدم اول شما باید بعد از دانلود برنامه آنرا در شاخه ای که فضای لازم را دارید کپی کنید و با استفاده از برنامه tar آنرا از حالت فشرده خارج کنید به صورت زیر این فرمان را راه اندازی کنید:

```
#tar -xvzf squid-3.1.10.tar.gz
```

بعد وارد شاخه squid-3.1.10 شده و در این شاخه شما برای نصب باید از سه فرمان make و configure و make install را راه اندازی کنید. قبل از اینکه فرامان Configure را اجرا کنید باید در مورد این فرمان توضیحاتی داده شود، این فرمان تنظیمات مورد نیاز شما را از طریق سوئیچهایی که به آن می دهید اعمال می کند. برای نمایش همه سوئیچ های موجود از فرمان زیر استفاده کنید که در شاخه کد برنامه اجرا شود:

```
# ./configure --help | less
```

چند سوئیچ مهم و پر استفاده برای شما در ادامه شرح داده خواهد شد. به این نکته توجه کنید که هر سوئیچ با -- شروع می شود.

سوئیچ prefix :

با این سوئیچ شما می توانید مسیر نصب را مشخص کنید، در بسیاری از موارد شاید شما قصد داشته باشید که چند ورژن از Squid را هم زمان بر روی سیستم خود نصب و یکی را انتخاب کنید برای این کار بهتر است که مسیر های نصب پیش فرض را تغییر دهید و مسیر های جدید ایجاد و انتخاب کنید، این سوئیچ برای این کار بسیار مفید است. در ادامه یک مثال از فرمان ./configure را مشاهده می کنید:

```
./configure --prefix=/opt/squid/3.1.10/
```

فرمان Configure علاوه بر توانایی در انتخاب مسیر نصب شما می توانید سایر مسیرها مثل فایل های اجرا و غیره را با استفاده از سوئیچ های --bindir, --sbindir تغییر دهید.

نکته:

سایر بخشهایی که شما در ادامه فرمان Configure با آنها سرو کار دارید برای فعال کردن و غیرفعال کردن یک قابلیت در Squid است که کفایت برای اینکار از دو سوئیچ enable-FEATURE برای فعال کردن قابلیت و disable-FEATURE برای غیرفعال کردن یک قابلیت است که در مثال زیر آنرا مشاهده می کنید:



```
./configure --enable-FEATURE # FEATURE will be enabled
./configure --disable-FEATURE # FEATURE will be disabled
```

سویچ **enable-gnuregex**:

با این سویچ قابلیت Regular expression را در Squid فعال می کنید که از آن برای ایجاد کردن لیست کنترل کننده استفاده می کند. اگر از سیستم عاملهای بروز شده یونیکسیسی استفاده می کنید نیازی به فعال کردن این بخش نیست چون در سیستم عامل شما این قابلیت فعال شده است.

سویچ **enable-storeio**:

در زمانی که از قابلیت Cache در Squid استفاده می کنید عملکرد آن به سرعت خواندن و نوشتن بر روی دیسک بسیار وابسته است و شما می توانید در این بخش تکنولوژی های استفاده شده در سیستم عامل خود را معرفی کنید به صورت زیر:

```
./configure --enable-storeio=ufs,aufs,coss,diskd,null
```

سویچ **enable-removal-policies**:

برای حذف کردن و آزاد کردن فضای Cache در سرور squid می توانید از سیاست های ایجاد شده در آن استفاده کنید با استفاده از این بخش سرور Squid در مورد مطالب ذخیره شده برای حذف آنها تصمیم می گیرد در زیر دو سیاست در سرور squid را با استفاده از این فرمان می توانید راه اندازی کنید:

```
./configure --enable-removal-policies=heap,lru
```

سویچ **enable-referer-log**:

با استفاده از این سویچ شما می توانید تمام هدر های درخواست های ارسال شده از سمت کاربران را ثبت کنید.

سویچ **disable-wccp**:

در Squid قابلیت وجود دارد به نام Cisco's Web Cache Communication Protocol که به صورت پیش فرض فعال است و در صورتی که نیاز به آن دارید آنرا فعال کنید و با استفاده از این سویچ می توانید آنرا غیرفعال کنید.

سویچ **disable-snmp**:

در Squid شما می توانید با استفاده از پروتکل SNMP وضعیت سرور خود را مانیتور کنید که به صورت پیش فرض این بخش فعال بود و بعد از اضافه کردن آن در فایل پیکربندی Squid می توانید از آن استفاده کنید اگر هم به این پروتکل نیازی ندارید می توانید با این سویچ آنرا غیرفعال کنید. البته این قابلیت از ورژن 3 به بعد به صورت پیش فرض فعال شده است.

**enable-cachemgr-hostname: سویچ**

در Squid بخش به نام cachemgr.cgi وجود دارد که شما می توانید با استفاده از یک رابط صفحه وب به مدیریت squid بپردازید. شما در این بخش یک نام برای برقرار ارتباط با سرور می توانید به جای آدرس IP و یا localhost استفاده کنید. در زیر روش تنظیم کردن آنرا مشاهده میکنید:

```
./configure --enable-cachemgr-hostname=squidproxy.example.com
```

enable-arp-acl: سویچ

شما با استفاده از این سویچ می توانید قابلیت ACL را مبتنی بر MAC را در سرور خود راه اندازی کنید، این قابلیت به صورت پیش فرض غیرفعال است و شما باید آنرا فعال کنید.

enable-default-err-language: سویچ

در زمانی که squid خطای در برقراری ارتباط پیدا می کند یک صفحه خطا برای کاربر درخواست کنند صفحه ارسال می کند که در زبانهای مختلف وجود دارد شما می توانید زبان این صفحه خطا را با استفاده از این سویچ مشخص کنید به صورت زیر:

```
./configure --enable-default-err-language=Spanish
```

قابلیت transparent در Squid

شما برای استفاده از یک سرور Proxy باید در مرورگر وب خود تنظیماتی را اعمال کنید ولی در بسیار از موارد سرویس دهنده های اینترنت یا همان ISP ها قصد مدیریت پهنای باید مصرفی از سمت کاربران خود را دارد و شاید مدیران شبکه هم دوست دارند گزارشات استفاده هر کاربر را مشاهده کنند پس برای انجام دادن این کار باید تمام ترافیک های وب شبکه خود را به سمت سرور Squid منتقل کنند برای انجام دادن این کار شما نیاز به یک فایروال دارید که در FreeBSD فایروالهای ipfw و pf وجود دارد و شما می توانید از هر کدام از این دو فایروال استفاده کرده و ترافیک را به سمت سرور Squid منتقل کنید و باید سرور Squid هم از این قابلیت پشتیبانی کند برای فعال کردن این قابلیت از سویچ های زیر استفاده کنید:

enable-ipfw-transparent:

این سویچ برای استفاده از فایروال IPFW در نظر گرفته شده است.

enable-pf-transparent:

این سویچ برای استفاده از فایروال IPFW در نظر گرفته شده است.



در بخشی مناسب با روش کار با حالت transparent آشنا خواهید شد.

سوئیچ: with-logdir

شما می توانید برای راحتی در مشاهده Log های برنامه برای آن مسیری با استفاده از این سوئیچ ایجاد کنید در FreeBSD تمام لاگها در زیر شاخه /var/log ذخیره می شود و برای مدیریت جداگانه Log بهتر است که مسیر را خود تعیین کنید. در زیر شما یک مثال از این بخش را مشاهده می کنید:

```
./configure --with-logdir=/var/log/squid/
```

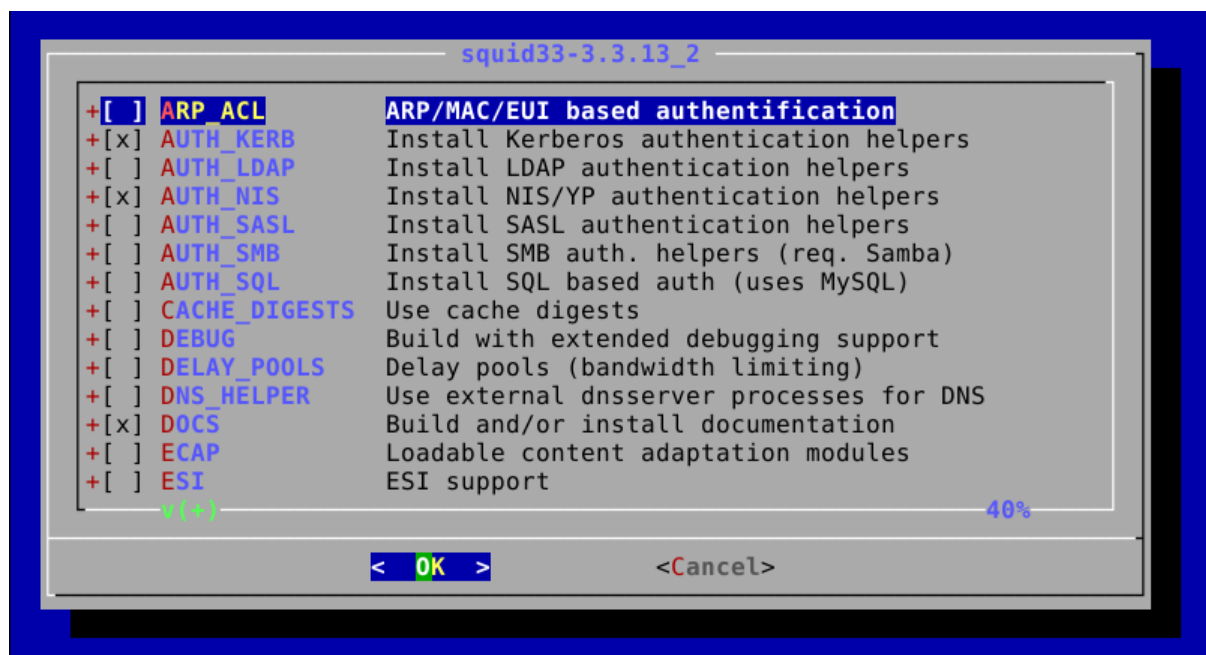
سوئیچ : with-pidfile

در FreeBSD در زیر شاخه /var/run هر برنامه ای که جدید به سیستم اضافه می شود PID خود را در فایل به نام خودش ذخیره می کند که سایر پردازشها بتوانند از طریق این PID با برنامه در ارتباط باشند شما می توانید با استفاده از این سوئیچ مسیر این فایل را تغییر دهید.

نصب Squid با استفاده از کد برنامه:

حال بعد از انتخاب کردن بخشهای مورد نیاز خود در فرمان ./configure این فرمان را اجرا کنید و در صورتی که با خطایی مواجه نشدید با دسترسی عادی فرمان Make را اجرا کنید و در صورتی که این فرمان هم به درستی به پایان رسید با سطح دسترسی کاربر root فرمان make install را اجرا کنید تا Squid نصب شود.

بخشهای فرمان Configure را می توانید با استفاده از سیستم ports در FreeBSD بدون نیاز به تایپ کردن انتخاب کنید و بسته نصبی خاص خود را ایجاد کنید. در شکل زیر آنرا مشاهده می کنید:



نمایش تنظیمات در زمان نصب کردن Squid با استفاده از ساختار ports در FreeBSD



راه اندازی Squid و برنامه های sqtop و squidclient

بعد از اتمام مراحل نصب شما با راه اندازی سریع و دو برنامه کاربردی برای تست کردن و نمایش وضعیت Squid می پردازیم. بعد از نصب در FreeBSD شما برای دسترسی داشتن به فایل اصلی پیکربندی آن باید به شاخه `/usr/local/etc/squid` مراجعه کنید. البته فایل اجرایی squid در شاخه `/usr/local/sbin` قرار گرفته و صفحه های توضیحات man آن هم در زیر شاخه `/usr/local/man` قرار گرفته است. شما با استفاده از فرمان `Whereis` به صورت نمایش داده شده در شکل زیر می توانید محل هر برنامه را مشخص کنید:

```
Terminal
File Edit View Terminal Tabs Help
root@FreeBSD:/usr/local/etc/squid # whereis squid
squid: /usr/local/sbin/squid /usr/local/man/man8/squid.8.gz /usr/ports/www/squid
root@FreeBSD:/usr/local/etc/squid # whereis squidclient
squidclient: /usr/local/sbin/squidclient /usr/local/man/man1/squidclient.1.gz
root@FreeBSD:/usr/local/etc/squid #
```

نمایش محل فایل های اصلی برنامه squid

```
cache_dir ufs /var/cache/ 500 16 256

acl my_pc src 192.168.2.21 # Replace with your IP address

http_access allow my_pc
```

سرور Squid فایل های کش شده خود را بر روی هارد دیسک ذخیره می کند قبل از هر اقدامی باید مسیر این شاخه را مشخص کرده و در مرحله بعدی آنرا ایجاد کنید خط اول این کار را انجام می دهد.

در خط دوم شما با استفاده از `acl` می توانید آدرس IP سیستم خود را مشخص کنید که با استفاده از آن قصد استفاده کردن از Squid را دارید، این بخش به تنظیمات شبکه شما بستگی دارد. در این خط شما لیستی ایجاد کرده اید به نام `my_pc` که فقط یک آدرس در آن اضافه کرده اید

در خط سوم شما با استفاده از تگ `http_access` به لیست بالا اجازه استفاده می دهید، این سه خط را در بالای فایل `squid.conf` قرار داده تا در مرحله بعدی شاخه `Cache` را ایجاد کنید

```
Terminal
File Edit View Terminal Tabs Help
^[(escape) menu ^y search prompt ^k delete line ^p prev li ^g prev page
^o ascii code ^x search ^l undelete line ^n next li ^v next page
^u end of file ^a begin of line ^w delete word ^b back 1 char
^t top of text ^e end of line ^r restore word ^f forward 1 char
^c command ^d delete char ^j undelete char ^z next word
=====line 8 col 0 lines from top 8 =====
cache_dir ufs /var/spool/squid 500 16 256
acl my_pc src 192.168.2.21 # Replace with your IP address
http_access allow my_pc
#
```

ویرایش کردن فایل squid.conf



در مرحله بعدی شما باید شاخه اصلی را که در فایل پیکربندی مشخص کرده اید را ایجاد کنید و صاحب این شاخه را کاربر squid قرار دهید به صورت زیر:

```
# mkdir /var/spool/squid

# chown squid:squid /var/spool/squid
```

حال در مرحله بعد فرمان squid را با سوئیچ Z به صورت زیر اجرا کنید تا خروجی ایجاد فایل را مشاهده کنید به صورت شکل زیر:

```
Terminal
File Edit View Terminal Tabs Help
2016/10/14 02:21:22 kid1| Set Current Directory to /var/squid/cache
2016/10/14 02:21:22 kid1| Creating missing swap directories
2016/10/14 02:21:22 kid1| /var/spool/squid exists
2016/10/14 02:21:22 kid1| Making directories in /var/spool/squid/00
2016/10/14 02:21:23 kid1| Making directories in /var/spool/squid/01
2016/10/14 02:21:23 kid1| Making directories in /var/spool/squid/02
2016/10/14 02:21:23 kid1| Making directories in /var/spool/squid/03
2016/10/14 02:21:23 kid1| Making directories in /var/spool/squid/04
2016/10/14 02:21:23 kid1| Making directories in /var/spool/squid/05
2016/10/14 02:21:23 kid1| Making directories in /var/spool/squid/06
2016/10/14 02:21:23 kid1| Making directories in /var/spool/squid/07
2016/10/14 02:21:23 kid1| Making directories in /var/spool/squid/08
2016/10/14 02:21:23 kid1| Making directories in /var/spool/squid/09
2016/10/14 02:21:23 kid1| Making directories in /var/spool/squid/0A
2016/10/14 02:21:23 kid1| Making directories in /var/spool/squid/0B
2016/10/14 02:21:23 kid1| Making directories in /var/spool/squid/0C
2016/10/14 02:21:23 kid1| Making directories in /var/spool/squid/0D
2016/10/14 02:21:23 kid1| Making directories in /var/spool/squid/0E
2016/10/14 02:21:23 kid1| Making directories in /var/spool/squid/0F

root@FreeBSD:/var # cd /var/spool/squid/
root@FreeBSD:/var/spool/squid # ls
00      02      04      06      08      0A      0C      0E
01      03      05      07      09      0B      0D      0F
root@FreeBSD:/var/spool/squid #
```

ساختار شاخه کش در Squid

راه اندازی squid

بعد از انجام مراحل بالا اگر سرور شما دارای آدرس IP معتبر و تنظیمات DNS درست باشد با استفاده از فرمان زیر راه اندازی می شود:

```
#/usr/local/etc/rc.d/squid onestart
```

در صورتی که تنظیمات درست نباشد شما با پیغامهایی به صورت زیر مواجه می شوید:



```
Terminal
File Edit View Terminal Tabs Help
root@FreeBSD:/var/spool/squid # /usr/local/etc/rc.d/squid onestart
Starting squid.
2016/10/14 02:33:13| WARNING: Could not determine this machines public hostname.
Please configure one or set 'visible_hostname'.
2016/10/14 02:33:14| WARNING: Could not determine this machines public hostname.
Please configure one or set 'visible_hostname'.
2016/10/14 02:33:16| WARNING: Could not determine this machines public hostname.
Please configure one or set 'visible_hostname'.
```

راه اندازی کردن سرور squid با استفاده از ساختار rc.d

برای رفع این مشکل خط زیر را در فایل squid.conf اضافه کنید:

```
visible_hostname squid.example.com
```

حال فرمان را دوباره راه اندازی کنید تا خروجی به صورت زیر مشاهده کنید:

```
Terminal
File Edit View Terminal Tabs Help
root@FreeBSD:/var/spool/squid # /usr/local/etc/rc.d/squid onestart
Starting squid.
root@FreeBSD:/var/spool/squid # sockstat -l4
USER      COMMAND  PID  FD PROTO  LOCAL ADDRESS      FOREIGN ADDRESS
squid     squid    6668  6  udp4   *:58486            *: *
squid     squid    6668  8  udp4   *:58908            *: *
squid     squid    6668  16 tcp4   *:3128             *: *
_ntp      ntpd     4255  7  udp4   192.168.85.128:123 *: *
_ntp      ntpd     4255  10 udp4   127.0.0.1:123     *: *
root      sendmail  799   3  tcp4   127.0.0.1:25      *: *
root      sshd     796   4  tcp4   *:22               *: *
root      syslogd  576   7  udp4   *:514              *: *
```

نمایش وضعیت پورتهای باز سرور Squid

squidtop: برنامه

برنامه ای برای مشاهده وضعیت عملکرد لحظه ای در squid وجود دارد به نام sqtop که شما می توانید از طریق ports و به صورت زیر این بسته را نصب کنید ب صورت زیر:

```
#/usr/ports/net/sqtop
#make
#make install
```

در مرحله بعد فرمان را Sqtop را اجرا کنید تا برنامه برای شما به صورت زیر اجرا شود:



```

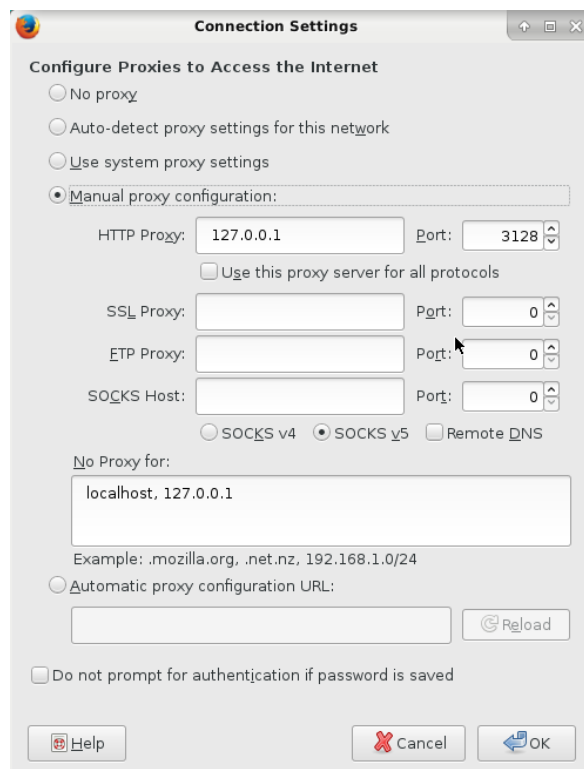
Terminal
File Edit View Terminal Tabs Help
Connected to 127.0.0.1:3128          sqttop-2013-12-17 (C) 2006 Oleg Palij
Host: 127.0.0.1
cache_object://localhost/active_requests

Average speed: 0.0 KB/s      Active hosts: 1      Active connections: 1

```

خروجی فرمان sqttop

اگر سرور squid راه اندازی شده باشد در خط اول از این فرمان درست را وضعیت **connected** را مشاهده می کنید، بعد از تنظیم کردن بخش **proxy** از مرورگر وب خود به صورت زیر صفحه های باز شده را در این برنامه مشاهده می کنید:





حال بعد از باز کردن یک صفحه از طریق مرورگر خود خروجی برنامه Sqttop به صورت زیر تغییر خواهد کرد:

```

Terminal
File Edit View Terminal Tabs Help
Connected to 127.0.0.1:3128          sqttop-2013-12-17 (C) 2006 Oleg Palij
Host: 127.0.0.1
cache_object://localhost/active_requests
http://systat.ir/font/BYekan.woff
http://systat.ir/frame_b.htm
http://systat.ir/frame_c.htm
Average speed: 0.0 KB/s      Active hosts: 1      Active connections: 4

```

نمایش تغییرات sqttop

برنامه squidclient

در زمان نصب برنامه squid برنامه دیگری هم برای چک کردن سرور به صورت خودکار وجود دارد به نام squidclien که کفایست بعد از این فرمان نام سایت مورد نظر را تایپ کنید تا در قالب یک فایل html خروجی را برای شما نمایش دهد در صورتی که هیچ پیغامی برای شما نمایش داده نشود این بدان معنی است که سرور شما به درستی پیکربندی نشده است.



سرور NFS در FreeBSD

سیستم عامل FreeBSD از پروتکل NFS برای اشتراک گذاری فایل در شبکه در بین سیستم عامل های مختلف استفاده می کند، در زمانهای قدیم برای اشتراک فایل بین WINDOWS و FreeBSD سرور دیگری به نام samba ایجاد شد و هنوز هم این سرور وجود دارد، ولی در حال حاضر سیستم عامل windows هم از nfs پشتیبانی می کند، در این مقاله شما با اشتراک گذاری فایل بین windows و FreeBSD با استفاده از NFS آشنا می شوید.

راه اندازی NFS در FreeBSD

سرور NFS به صورت پیش فرض در FreeBSD نصب شده و شما فقط کافیست که آنرا فعال کنید. برای اینکار باید سرویسهای زیر را فعال کنید:

برنامه **nfsd** که برای پاسخ دادن به درخواستهای سرویس گیرنده ها راه اندازی می شود.

برنامه **mountd** این سرویس برای انجام دادن درخواستهای رسیده شده به nfsd استفاده می شود.

برنامه **rpcbind** این برنامه لازم است تا سرویس گیرنده ها بتواند پورت سرور NFS را برای اتصال پیدا کنند.

در قدم اول شما باید از طریق فایل معروف rc.conf این سرویسها را برای راه اندازی تعریف کنید به صورت زیر:

```
rpcbind_enable="YES"
nfs_server_enable="YES"
mountd_enable="YES"
```

حال شما با استفاده از فرمان service یا فرمان های rc.d این سه سرویس را راه اندازی کنید به صورت زیر:

```
Terminal
File Edit View Terminal Tabs Help
root@FreeBSD:~ # /etc/rc.d/rpcbind start
Starting rpcbind.
root@FreeBSD:~ # /etc/rc.d/nfsd start
NFSv4 is disabled
Starting mountd.
Starting nfsd.
root@FreeBSD:~ # █
```

راه اندازی کردن سرویس nfsd با rc.d در FreeBSD

همانطوری که مشاهده می کنید با راه اندازی سرویس nfsd به صورت خودکار سرویس mountd هم راه اندازی می شود. برای اینکه از راه اندازی سرویس ها بر روی شبکه مطمئن باشید فرمان sockstat را به صورت زیر اجرا کنید:



```

Terminal
File Edit View Terminal Tabs Help
root@FreeBSD:~ # sockstat -l4
USER      COMMAND  PID  FD  PROTO  LOCAL ADDRESS  FOREIGN ADDRESS
root      nfsd     1705  5   tcp4   *:2049         *:
root      mountd   1703  8   udp4   *:761          *:
root      mountd   1703  9   tcp4   *:761          *:
root      rpcbind  1675  9   udp4   *:111          *:
root      rpcbind  1675  10  udp4   *:835          *:
root      rpcbind  1675  11  tcp4   *:111          *:
root      nfsuserd 1145  3   udp4   *:930          *:
root      nfsuserd 1144  3   udp4   *:930          *:
root      nfsuserd 1143  3   udp4   *:930          *:
root      nfsuserd 1142  3   udp4   *:930          *:
root      nfsuserd 1141  3   udp4   *:930          *:
root      sendmail 868   3   tcp4   127.0.0.1:25   *:

```

نمایش وضعیت پورت های باز شبکه

به اشتراک گذاشتن شاخه:

برای به اشتراک گذاشتن شاخه شما در ابتدا باید فایلی به نام exports را در زیر شاخه /etc ایجاد کنید به صورت زیر:

```

Terminal
File Edit View Terminal Tabs Help
root@FreeBSD:~ # touch /etc/exports
root@FreeBSD:~ # █

```

ایجاد کردن فایل exports

شاخه های به اشتراک گذاشته شده را در این فایل قرار می دهید در بخش اول شما باید نام شاخه را مشخص کنید و در قسمت دوم نوع دسترسی را مشخص کنید و در بخش آخر هم آدرس ip یا اسم سیستم هایی که قصد استفاده از این شاخه را دارند را مشخص کنید، در صورت مشخص نکردن بخش آخر منظور همه یا همان everyone می باشد در شکل زیر نمونه ای از این فایل را مشاهده می کنید:

```

Terminal
File Edit View Terminal Tabs Help
root@FreeBSD:~ # cat /etc/exports
/usr      -alldirs -network 195.168.85.0/24
/etc      -ro      -network 192.168.85.0/24
root@FreeBSD:~ # █

```

شاخه های به اشتراک گذاشته شده در فایل exports

در این فایل ما دو شاخه به نام های /usr و /etc را share کردیم. شاخه اول را با تمام زیر شاخه هایش را برای شبکه 192.168.85.0 به اشتراک گذاشتیم و شاخه دوم /etc را برای همان شبکه فقط در حالت خوانده شدن به اشتراک گذاشتیم. بعد از اعمال این تغییرات سرویس mountd را به صورت زیر راه اندازی مجدد کنید و از فرمان showmounts به صورت زیر برای نمایش شاخه های به اشتراک گذاشته استفاده کنید:



```

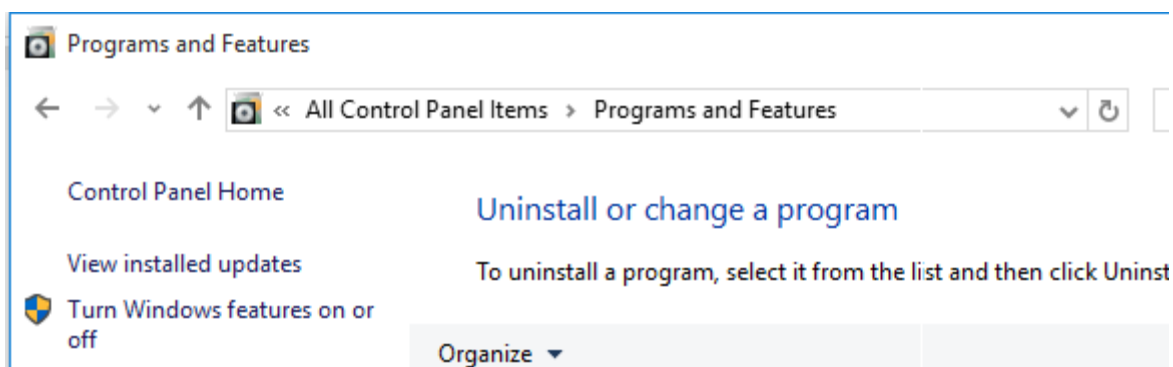
Terminal
File Edit View Terminal Tabs Help
root@FreeBSD:~ # /etc/rc.d/mountd restart
Stopping mountd.
Waiting for PIDS: 1912.
Starting mountd.
root@FreeBSD:~ # showmount -e
Exports list on localhost:
/usr          195.168.85.0
/etc         192.168.85.0
root@FreeBSD:~ # █

```

نمایش وضعیت شاخه های اشتراک گذاشته شده

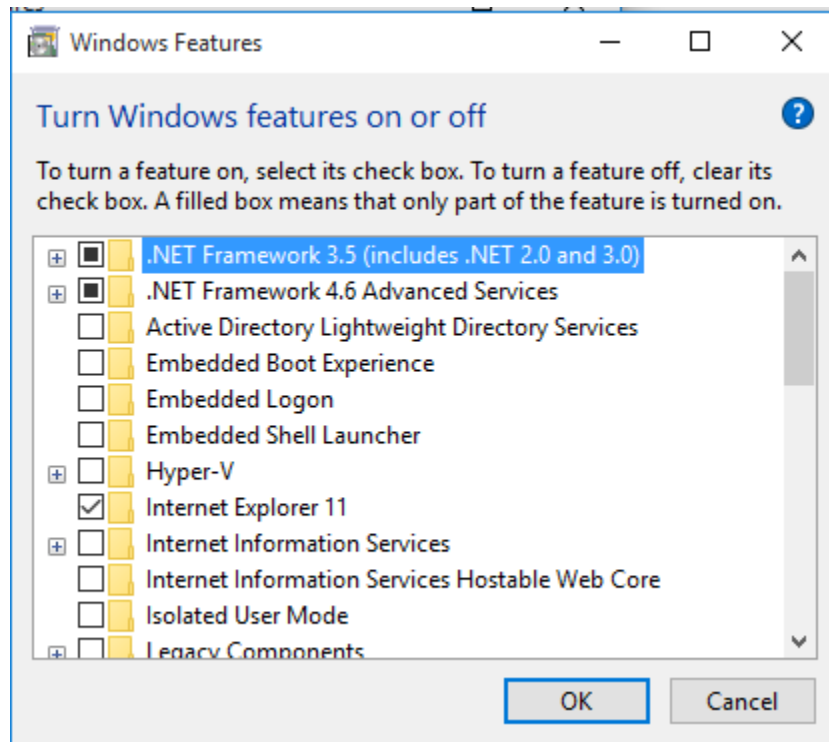
تنظیم کردن ویندوز 10 برای استفاده از شاخه ها:

به صورت پیش فرض NFS client در سیستم عامل ویندوز راه اندازی نشده است و شما باید آنرا فعال کنید برای این امر باید ابتدا به control panle رفته و در بخش Programs and Features وارد شده و در سمت چپ این بخش گزینه turn windows feature on or off را باز کنید:



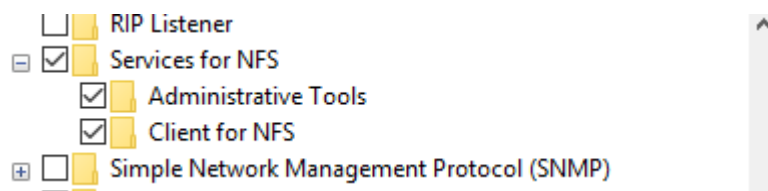
بخش اضافه کردن قابلیت به ویندوز

بعد از باز کردن شما با منوی به صورت زیر مواجه می شوید:



منوی اضافه کردن قابلیت به ویندوز

بخش service for nfs را باز کرده و هر دو تیک این بخش را فعال کنید:



اضافه کردن NSF در بخش کلاینت

بعد از ok کردن این قابلیت به سیستم شما اضافه شده و شما می توانید از فرمان mount در windows استفاده کنید اجرا این فرمان را در cmd مشاهده می کنید:

```

Command Prompt
C:\Users\admin>mount
Local      Remote
-----
C:\Users\admin>

```

اجرا فرمان mount در ویندوز

برای mount کردن شاخه های مورد نظر کفایست که فرمان زیر را در cmd به صورت زیر اجرا کنید:



Command Prompt

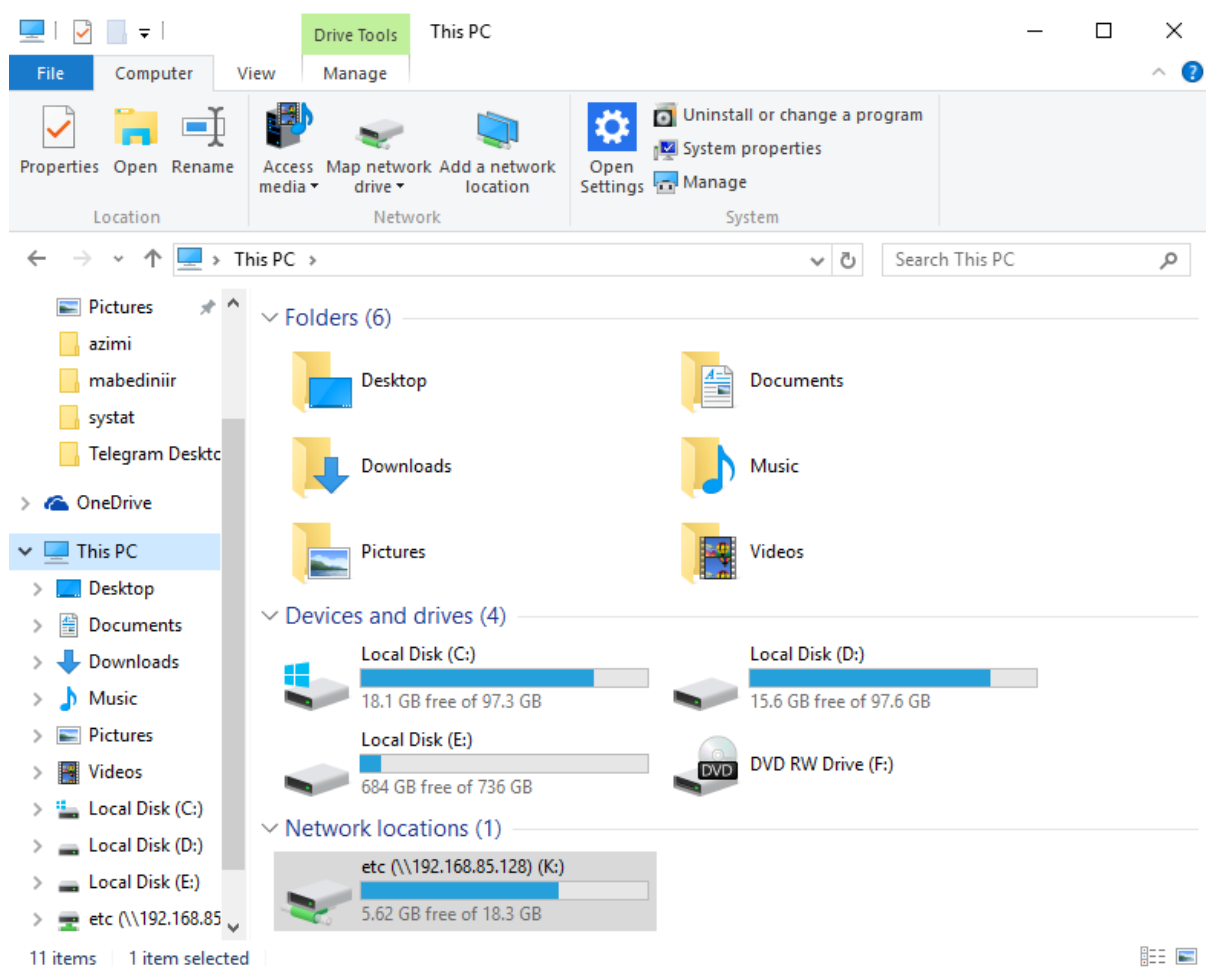
```
C:\Users\admin>mount \\192.168.85.128\etc k:
k: is now successfully connected to \\192.168.85.128\etc

The command completed successfully.

C:\Users\admin>
```

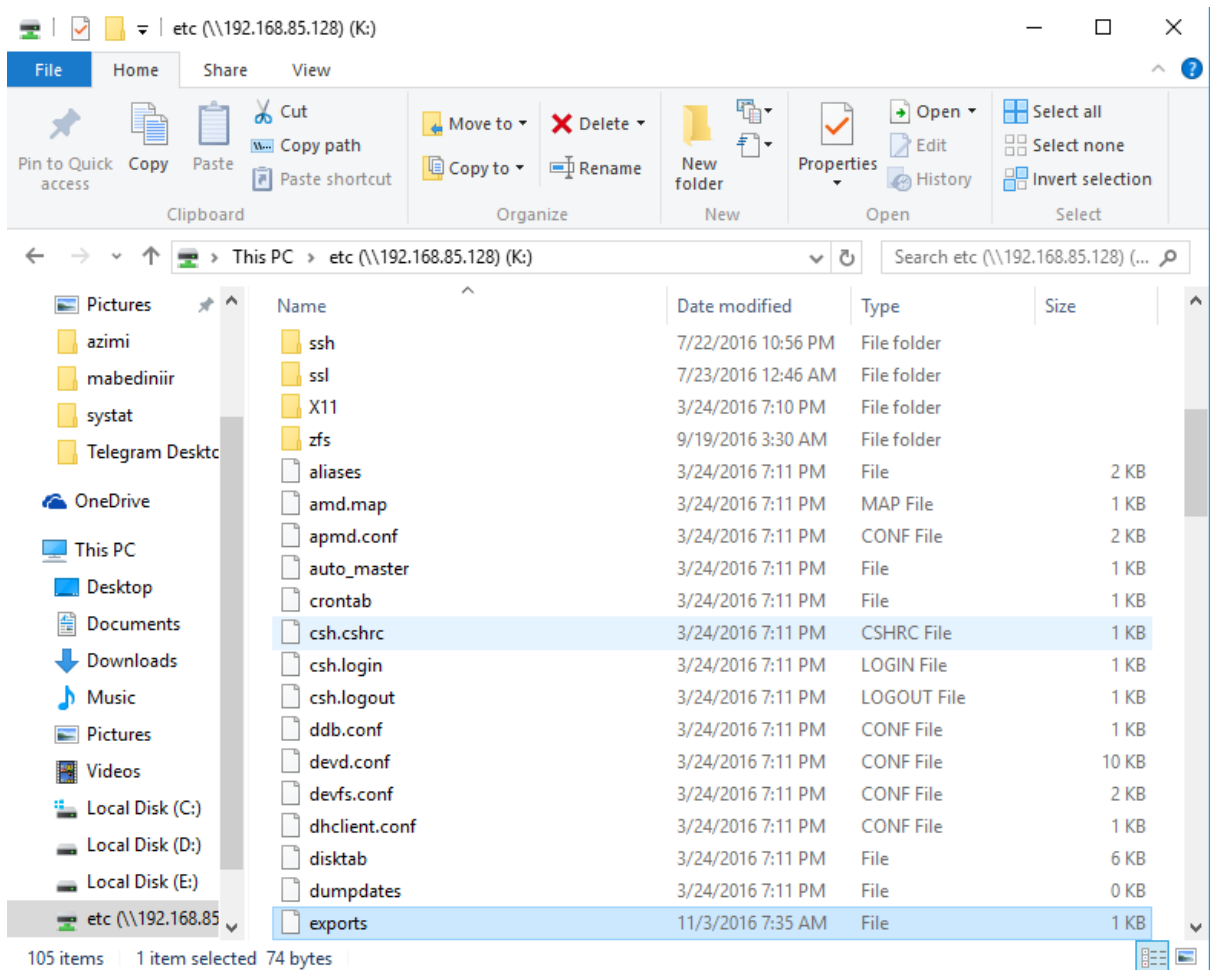
Mount کردن NFS در ویندوز

حال در بخش This PC این شاخه اضافه می شود:



نمایش شاخه mount در شده در My computer

شما می توانید به راحتی وارد این شاخه شوید:



وارد شدن از طریق ویندوز به شاخه های share شده

خروجی فرمان mount را در شکل زیر مشاهده می کنید:

```

Command Prompt

C:\Users\admin>mount

Local      Remote                                Properties
-----
k:         \\192.168.85.128\etc                 UID=-2, GID=-2
                                                rsize=131072, wsize=131072
                                                mount=soft, timeout=0.8
                                                retry=1, locking=no
                                                fileaccess=755, lang=ANSI
                                                casesensitive=no
                                                sec=sys

C:\Users\admin>

```

خروجی فرمان mount در windows



برای `umount` کردن شاخه های اضافه شده کفایت که فرمان `umount` را با سوییچ `-a` به صورت شکل زیر اجرا کنید:

```
Command Prompt
C:\Users\admin>umount -a
You have these active NFS connections:
k:      \\192.168.85.128\etc
Continuing will cancel the connections.
Do you want to continue this operation? (Y/N) [N]:y
Disconnecting      k:      \\192.168.85.128\etc
There are open files and/or incomplete directory searches pending on the connection.
Do you want to continue this operation? (Y/N) [N]:y
The command completed successfully.
C:\Users\admin>
```

اجرا کردن فرمان `umount` در ویندوز

این فرمان در ابتدا شاخه های `mount` شده را نمایش می دهد و شما با پاسخ `Y` به سوالات آن می توانید شاخه ها را `unmounts` کنید.



راه اندازی web سرور در FreeBSD

به صورت پیش فرض در FreeBSD وب سروری نصب نیست و شما می توانید هر وب سروری را که دوست دارید نصب کنید، در سیستم عامل OpenBSD وب سرور apache به صورت پیش فرض نصب است. در FreeBSD هم ما قصد داریم از این وب سرور استفاده کنیم و در ادامه با نصب و راه اندازی آن بیشتر آشنا شوید.

در این مقاله ما روزن 2.x از این وب سرور را مورد بررسی قرار می دهیم.

نصب : apache

شما به دو روش می توانید به نصب این سرور بپردازید در بخش اول استفاده از بسته از پیش کامپایل شده یا همان باینری و بخش دوم اگر قصد دارید که تنظیمات خاص خود را اضافه کنید استفاده از سیستم پورت در FreeBSD البته شما می توانید به صورت دستی هم به نصب بپردازید که در این بخش مورد بحث ما نیست.

برای نصب کردن از طریق بسته های باینری باید از فرمان `pkg install` به صورت زیر استفاده کنید:

برای نصب کردن از طریق سیستم پورت کافیست که وارد شاخه زیر شده:

```
#/usr/local/ports/www/apache24
```

وفرمان های `make` را در ابتدا و فرمان `make install` را با دسترسی کاربر `root` اجرا کنید. در میان مراحل `make` هم شما می توانید تنظیمات دلخواه خود را به بسته اضافه کنید. در انتها هم می توانید با استفاده از `make clean` تمام بسته های اضافه و فایل های سورس را پاک کنید.

فایل پیکربندی:

بعد از نصب کردن `apache` این برنامه فایل های خود را در زیر شاخه `/usr/local/etc` در شاخه ای به نام `apache24` قرار می دهد در این شاخه فایل اصلی پیکربندی این برنامه به نام `httpd.conf` قرار دارد که تمام تنظیمات از این بخش شروع می شود، این فایل هم مثل سایر فایل های پیکربندی دارای بخش ها مختلف است و خطوط که با `#` شروع می شود هم توضیحات بوده و هیچ تاثیری در برنامه ندارند. در این بخش با بخش های مهم این فایل پیکربندی آشنا می شوید.

بخش `ServerRoot "/usr/local"`

این بخش محل قرار گرفتن فایل های برنامه `apache` زیر شاخه های مربوطه است، برای مثال برنامه های اجرایی در زیر شاخه های `bin` و `sbin` از `/usr/local` قرار می گیرند و همه برنامه های دیگر مثل فایل پیکربندی در زیر شاخه `etc/apache24` از زیر شاخه `/usr/local` قرار می گیرند. شما به راحتی در این مسیر می توانید فایل های مورد نظر خود را پیدا کنید. این مسیر هم در فایل پیکربندی استفاده شده و برای آدرس دهی بخش های دیگر هم از این آدرس استفاده می کنند.

بخش `ServerAdmin you@example.com`



اگر برای سرویس شما مشکلی ایجاد شده باشد از طریق این آدرس ایمیل یک میل برای شما ارسال می شود، شما می توانید این بخش را با آدرس مدیر سیستم تغییر دهید.

بخش Listen

در این بخش شما می توانید پورت پیشفرض برنامه که 80 است را تغییر دهید، حتی اگر سرور شما دارای چند آدرس IP باشد شما در این بخش می توانید تنظیم کنید که بر روی هر آدرس بروی کدام پورت به ارایه کردن سرویس پردازد.

بخش "DocumentRoot "/usr/local/www/apache2x/data"

این بخش بسیار بخش مهمی است در سرور apache که در حقیقت محلی است که فایل های وب سایت شما در آن ذخیره می شود. شما بعد از طراحی سایت خود کافیسست که آن را در این محل کپی کنید و فایل های هم به نام index.html داشته باشید که صفحه اول سایت شما باشد بعد از راه اندازی سرور شما برنامه apache این فایل را از این مسیر برای شما بارگذاری می کند.

راه اندازی apache:

شما از چند طریق می توانید برای اینکه بتوانید از طریق سیستم rc.conf سرور apache را راه اندازی کنید و در زمان راه اندازی خودکار سیستم عامل این سرویس هم راه اندازی شود باید ابتدا در فایل معروف rc.conf در زیر شاخه /etc خط زیر را اضافه کنید:

```
apache24_enable="YES"
```

حال شما می توانید به راحتی با استفاده از فرمان های rc.d به صوت زیر این سرور را راه اندازی کنید، غیرفعال کنید و یا راه اندازی مجدد کنید:

```
#/usr/local/etc/rc.d/apache24 start
```

در FreeBSD فرمانی هم به نام service اضافه شده است که شما بتوانید با استفاده از آن سرویس های مورد نیاز خود را مدیریت کنید به صورت زیر می توانید:

```
# service apache24 start
```

خب بعد از اینکه این فرمان بدون خطا راه اندازی شده برای چک کردن این سرویس کافیسست که در مرورگر سیستم خود localhost را وارد کنید تا پیغام It is working را مشاهده می کنید.

برای چک کردن فایل پیکربندی کافیسست که فرمان service را به صورت زیر اجرا کنید:



```
# service apache24 configtest
```

قابلیت : Virtual Hosting

شما با استفاده از این قابلیت می توانید چندین سایت را بر روی یک سرور apache راه اندازی کنید. هاست های مجازی می تواند بر اساس آدرس IP باشد یا بر اساس نام . در روش استفاده از IP برای هر سایت باید یک آدرس IP داشته باشید، در روش استفاده از نام به جای آدرس IP سرور به هدر درخواست ارسال شده از سمت Client نگاه می کند و سایت مورد نظر را در اختیار کاربر قرار می دهد در این روش شما می توانید از یک آدرس IP یکسان برای سایت های مختلف استفاده کنید.

برای پیگیری این بخش شما باید برای هر سرور مجازی از یک بلوک به نام virtual host به صورت زیر ایجاد کنید:

```
<VirtualHost *>
    ServerName www.systat.ir
    DocumentRoot /www/ systat.ir
</VirtualHost>
```

برای هر هاست مجازی باید یک نام در بخش server Name انتخاب کنید و باید یک مسیر برای شاخه DocumentRoot انتخاب کنید.



نصب و راه اندازی SSH

یکی از دلایلی که من به سیستم عامل های خط فرمانی علاقمند هستم قابلیت انجام دادن همه کارها با استفاده از خط فرمان است، شما می توانید از هر مکانی با هر سرعت اینترنتی به سرور خود در هر جای دنیا متصل شوید و کارهای خود را انجام دهید. در زمان های قدیم برای برقرار ارتباط از سرویس telnet استفاده می شود که این سرویس امن نبوده و در مقابل شنود در شبکه بی دفاع است، در مقابل این سرویس، سرویس دیگری به نام ssh ایجاد و طراحی شد ارتباطی امن در بستر نا امن ایجاد می کند و شما به صورت امن می توانید فرمانهای خود را راه اندازی کنید و نگران سرقت شدن اطلاعات خود نباشید. حتی شما در برقرار ارتباط با سرور خود می توانید از کلید های امنیتی به جای رمز عبور استفاده کنید.

در ادامه این مقاله شما با روش فعال کردن سرور ssh در FreeBSD آشنا می شوید و بعد به سراغ روشهای برقرار ارتباط با سرور از طریق فرمان ssh آشنا می شوید. و در مرحله بعدی از کلید به عنوان رمز عبور برای برقرار ارتباط استفاده می کنید و در آخر روش tunnel زدن و bypass کردن firewall را شرح خواهیم داد.

فعال کردن سرور ssh

این سرور به صورت پیش فرض در سیستم عامل FreeBSD نصب شده و شما نیازی به نصب کردن آن ندارید و فقط باید آنرا فعال کنید برای فعال کردن این سرویس مثل همه سرویسهای دیگر شما می توانید هم از فرمان Service استفاده کنید و هم از سیستم rc.d برای این کار استفاده کنید. در ادامه دو روش مورد بررسی قرار می گیرد.

برای فعال کردن از طریق فرمان service کفایت فرمان زیر را اجرا کنید:

```
#service sshd start
```

برای راه اندازی دائمی سرویس ssh و راه اندازی خودکار در زمان راه اندازی سیستم کفایت که خط زیر را در فایل rc.conf در زیر شاخه /etc اضافه کنید.

```
sshd_enable="YES"
```

حال بعد از اضافه کردن می توانید با استفاده از فرمانهای rc.d به صورت زیر این سرویس را راه اندازی کنید:

```
Terminal
File Edit View Terminal Tabs Help
root@FreeBSD:~ # /etc/rc.d/sshd start
Performing sanity check on sshd configuration.
Starting sshd.
root@FreeBSD:~ #
```

راه اندازی کردن ssh

برای اطمینان پیدا کردن از اینکه این سرویس راه اندازی شده و بروی شبکه در حال ارایه دادن سرویس است فرمان sockstat را به صورت زیر اجرا کنید:



```

Terminal
File Edit View Terminal Tabs Help
root@FreeBSD:~ # sockstat -l4
USER      COMMAND  PID  FD  PROTO  LOCAL ADDRESS  FOREIGN ADDRESS
root      sshd     2239 4   tcp4   *:22           *:22
root      ftpd     1448 6   tcp4   *:21           *:21
root      mountd  1228 7   udp4   *:727          *:727
root      mountd  1228 8   tcp4   *:727          *:727
root      sendmail 867 3   tcp4   127.0.0.1:25   *:25
_ntp     ntpd     817 7   udp4   192.168.85.128:123 *:123
_ntp     ntpd     817 10  udp4   127.0.0.1:123  *:123
root      nfsd     749 5   tcp4   *:2049         *:2049
root      rpcbind  617 9   udp4   *:111          *:111
root      rpcbind  617 10  udp4   *:869          *:869
root      rpcbind  617 11  tcp4   *:111          *:111
root      syslogd  613 7   udp4   *:514          *:514
?        ?        ?    ?   udp4   *:2049         *:2049
root@FreeBSD:~ # █

```

نمایش وضعیت پورتهای باز

در زمان راه اندازی اولین بار سرور ssh این سرور به صورت خودکار کلیدهای برقرار ارتباط را به صورت شماتیک برای سیستم شما ایجاد می کند. با استفاده از فرمان rc.d شما می توانید بعد از راه اندازی هم این عمل را به صورت زیر انجام دهید:

```

Terminal
File Edit View Terminal Tabs Help
root@FreeBSD:/etc/ssh # /etc/rc.d/ssh keygen
Generating RSA1 host key.
2048 SHA256:Dl1N06AjYpyXXN7RHG12HfP1hRGsNq0CDnK6g5mn0ck root@FreeBSD (RSA1)
Generating RSA host key.
2048 SHA256:j6Tf03j3MBkDsWsMB9jwj5qEwQaq0/f1+RJI3h3Uz34 root@FreeBSD (RSA)
Generating DSA host key.
1024 SHA256:90YAprQoNGZg95Np6sRrRPtAX/DUCghDqgVDGISTCRA root@FreeBSD (DSA)
Generating ECDSA host key.
256 SHA256:kCYalbGAfcVAKs0TTIUIahRVGGWmAuodDEJlqJ1jX30 root@FreeBSD (ECDSA)
Generating ED25519 host key.
256 SHA256:H5Wwomd9fae7+Lj0H7YFTTm6S03MZMyt9erub+uhqNk root@FreeBSD (ED25519)
root@FreeBSD:/etc/ssh # ls
moduli                ssh_host_ed25519_key.pub
ssh_config            ssh_host_key
ssh_host_dsa_key      ssh_host_key.pub
ssh_host_dsa_key.pub  ssh_host_rsa_key
ssh_host_ecdsa_key    ssh_host_rsa_key.pub
ssh_host_ecdsa_key.pub sshd_config
ssh_host_ed25519_key
root@FreeBSD:/etc/ssh # █

```

ایجاد کردن کلید

پیکربندی sshd:

بعد از راه اندازی سرویس ssh این فرمان sshd است که همه کارها را انجام می دهد. با این فرمان می توانید وضعیت سرور خود را در آن واحد تغییر دهید و یا تنظیمات اصلی سرور خود را از فایل پیکربندی این سرور واقع در مسیر /etc/ssh/sshd_config می خواند. در این بخش با فرمان sshd و فایل پیکربندی sshd_conf آشنا می شوید.



مسیر اصلی این فرمان در زیر شاخه `/usr/sbin` قرار دارد این بدان معناست که فقط کاربر `root` می تواند به آن دسترسی داشته باشد. در ادامه با چند سوئیچ از این فرمان آشنا می شوید:

سوئیچ 4- با استفاده از این سوئیچ سرور `ssh` فقط از `ipv4` استفاده می کند.

سوئیچ 6- با استفاده از این سوئیچ سرور `ssh` فقط از `ipv6` استفاده می کند.

سوئیچ C- در بخش استفاده از کلید برای اتصال به سرور با فایلی آشنا می شوید که محل ذخیره کردن کلید هاست، شما با استفاده از این سوئیچ می توانید محل پیش فرض این فایل را تغییر دهید.

سوئیچ D- این سوئیچ برای مشاهده وضعیت سرور استفاده می شود.

سوئیچ E- بعد از این سوئیچ شما نام فایل گزارش گیری از سرور را انتخاب می کنید، این فایل جدا از گزارشگیری خود سیستم است.

سوئیچ p- بعد از این سوئیچ شما می توانید شماره پورت دلخواه خود را تعیین کنید و سرور را بر روی هر پورت دلخواه راه اندازی کنید.

سوئیچ t- این سوئیچ سرور شما را در حالت تست کردن قرار می دهد برای بروزرسانی کردن سرور قدیمی به سرور جدید استفاده می شود.

فایل پیکربندی `sshd_config`:

فایل پیکربندی دارای بخشهای مختلفی است در ادامه با بخشهایی آشنا می شوید که برای راه اندازی اولیه با آن بیشتر سروکار داریم. این فایل هم مثل همه فایل های پیکربندی در `FreeBSD` است و خطوط که با `#` شروع می شوند و خطوط خالی کامنت است. این بخشها هم به حروف کوچک و بزرگ حساس هستند.

بخش `AllowUsers`

در مقابل این بخش ما باید لیستی از کاربرانی را که قصد دارید به آنها اجازه استفاده از سرویس `ssh` را داشته باشند قرار دهید. هر نام در این لیست باید با نام کاربری که قصد استفاده داشته باشد یکسان بوده و با فاصله از هم جدا شوند. به دلیل اینکه به صورت `Local` این سرویس راه اندازی شده و از `RAS` سرور برای تایید هویت استفاده نمی شود پس باید کاربر مورد نظر بر روی سیستم اضافه شود.

بخش `AllowGroups`

این بخش هم مثل بخش قبلی است و شما می توانید به جای نام کاربران گروهی را تعیین کنید و هر کاربری را که نیاز به دسترسی به این سرویس را دارد را به آن گروه اضافه کنید و در این بخش از نام گروه به جای نام تک تک افراد استفاده کنید.



بخش AllowTcpForwarding

شما با استفاده از این بخش می توانید بعد از اتصال به سرور اگر این قابلیت فعال بود از سرور خود به عنوان یک پراکسی استفاده کنید. در بخش دور زدن فایروال به آن بیشتر می پردازیم. این بخش خود داری متغیرهای **yes** و **all** است که این قابلیت را فعال می کند، **no** که این قابلیت را غیرفعال کرده و اگر هم قصد دارید که کاربر **local** از این قابلیت استفاده کند.

بخش Banner

این بخش مشخص کننده فایلی است که قبل از بخش ورود کاربر به سیستم برای کاربر نمایش داده می شود، این بخش می تواند شامل پیغامهای خوش آمد گویی هم باشد. این بخش به صورت پیش فرض غیرفعال است.

بخش ChrootDirectory

شما با استفاده از این بخش می توانید بعد از وارد شدن کاربر به سیستم مسیری که باید به آن دسترسی داشته باشد را مشخص کنید. شاخه مورد نظر شما باید شامل فایل‌های لازم و شاخه های مورد نظر که برای ورود یک کاربر لازم است را داشته باشد مثل یک Shell به نام **sh** و همچنین شاخه **/dev/** باید با نودهای **stdin stdout stderr** و **null** و **tty** باشد.

بخش DenyGroups

شما در این بخش می توانید نام گروه هایی که قصد دارید به آنها اجازه استفاده از **SSH** را ندهید را وارد کنید و باید نام هر گروه با یک فاصله از هم جدا شود.

بخش DenyUsers

در این بخش ما می توانید نام کاربری را که قصد دارید به آنها اجازه استفاده از **SSH** را ندهید را وارد کنید و باید نام هر گروه با یک فاصله از هم جدا شود.

بخش Port

شما در این بخش می توانید شماره **port** پیش فرض سرور **SSH** که **22** است را تغییر دهید و یا برروی تعداد بیش از یک **port** سرور را راه اندازی کنید .

بخش ListenAddress

شما در این بخش می توانید آدرس **IP** ها و **port** های ماشین محلی خود را که قصد دارید فقط برروی آنها سرویس ارائه شود را مشخص کنید.

بخش MaxAuthTries

در این بخش شما می توانید تعداد تلاش های ناموفق برای ورود به سیستم را که هر ارتباط می تواند داشته باشد را مشخص کنید. بعد از این تعداد ارتباط از سمت سرور قطع خواهد شد.



بخش PermitRootLogin

این بخشی است که در آن می توانید به تو خاص دسترسی کاربر Root را برای ورود به سیستم از طریق ssh را مشخص کنید. این بخش به صورت پیش فرض به کاربر root اجازه ورود نمی دهد. برای اینکه قصد داشته باشید که به کاربر root این اجازه را دهید بهتر است آن را YES قرار دهید. این بخش شامل متغیرهای دیگری مثل `prohibit-password without` هم هست.



اتصال به سرور SSH از طریق کلید

برای بالا بردن سطح امنیتی سرور های SSH شما می توانید از کلید برای برقرار ارتباط با سرور استفاده کنید بجای اینکه از نام کاربری و رمز عبور استفاده کنید، شما می توانید کلید خود را بر روی سرور upload کنید و بعد بدون تایپ نام کاربری و رمز عبور به سرور متصل شوید. در حالت های دیگری شما می توانید سطح دسترسی همه کاربرانی را که با رمز عبور و نام کاربری وارد سیستم می شوند را بسته و فقط استفاده از کلید را اجبار کنید. شما حتی در محیط های ویندوزی هم می توانید این کلید ها را ایجاد و از برنامه putty برای برقرار ارتباط با کلید هم استفاده کنید که در این مطلب شما روش استفاده از این برنامه را آموزش خواهیم دید .

ایجاد کردن کلید در BSD

در مرحله اول هر فردی که قصد برقرار ارتباط را دارد باید بر روی سیستم محلی خود کلید های RSA را ایجاد کند، برنامه ای به نام ssh-keygen به همراه برنامه ssh بر روی سیستم شما نصب می شود که توان ایجاد کردن کلید های عمومی و شخصی به اصطلاح key public و private key را دارد کفایت که با کاربر خود این کلید را ایجاد کنید، به این نکته توجه کنید که کلید های خود را با یک رمزی که به خاطر دارید و سخت است از نظر امنیتی حمایت کنید این بخش passphrase نام دارد که در زمان ایجاد کردن یک کلید از شما پرسیده می شود. در شکل زیر خروجی حاصل از اجرای برنامه ssh – keygen را مشاهده می کنید:

```

Terminal
File Edit View Terminal Tabs Help
admin@FreeBSD:~ % ssh-keygen -t rsa
Generating public/private rsa key pair.
Enter file in which to save the key (/home/admin/.ssh/id_rsa):
Created directory '/home/admin/.ssh'.
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/admin/.ssh/id_rsa.
Your public key has been saved in /home/admin/.ssh/id_rsa.pub.
The key fingerprint is:
SHA256:nd+zvIWi5oh3JHQHItUDiLUandwMbI04KQINq4026w0 admin@FreeBSD
The key's randomart image is:
+----[RSA 2048]-----+
|.o . =0000
|..o oo** . +
|o o..=00. o
|= o o .....
|=o . .S.o.
|.. . . . .
| E o o + .
| + ...o. o +
|.o . ...+o +.
+----[SHA256]-----+
admin@FreeBSD:~ % █

```

ایجاد کردن کلید



برای ایجاد کردن کلید از نوع RSA شما باید بعد از فرمان برنامه یا همان ssh-keygen از سوئیچ -t و مقدار RSA استفاده کنید. در بخش زیر شما باید رمز حمایت کنند کلید خود را در دو بار پشت سر هم وارد کنید:

```
Enter passphrase (empty for no passphrase):
```

```
Enter same passphrase again:
```

اگر از passphrase استفاده کنید هر زمانی که قصد استفاده از کلید را داشته باشید باید این رمز را وارد کنید، اگر از ssh-agent استفاده کنید فقط یکبار کافیست که این کار را انجام دهید.

بعد از اجرا کردن این فرمان در شاخه Home کاربری که فرمان را با آن اجرا کرده اید شاخه ای ایجاد می شود به نام ssh. که یک شاخه مخفی شده است و در داخل آن کلید های public و private ذخیره می شود، نام کلید public در این شاخه id_rsa.pub و نام کلید private در این شاخه id_rsa است.

در شکل زیر فایل های مورد در شاخه sshd را مشاهده می کنید.

```
Terminal
File Edit View Terminal Tabs Help
admin@FreeBSD:~ % cd .ssh/
admin@FreeBSD:~/.ssh % pwd
/usr/home/admin/.ssh
admin@FreeBSD:~/.ssh % ls
id_rsa      id_rsa.pub
admin@FreeBSD:~/.ssh %
```

نمایش کلید های ایجاد شده

شما باید از کلید private خود محافظت کنید و کلید public را برای برقرار ارتباط با شما و یا برقرار کردن ارتباط با سرور ssh بر روی سرور قرار دهید و یا در صورتی که دوست داشته باشید که دیگر مطالب خود را با استفاده از این روش رمزنگاری برای شما ارسال کنند کافیست که کلید عمومی یا همان public را در اختیار آنها قرار دهید و یا حتی روی سایت خود به اشتراک گذارید. کلید private در این بخش نقش مهم و اساسی دارد و هر شخصی که به این کلید دسترسی پیدا کند و تواند به همه داده های رمز شده ای که به سمت شما ارسال می شد به صورت غیر رمز شده دسترسی پیدا کند.

در مرحله بعدی در سمت سرور فایلی وجود دارد به نام authorized_keys در زیر شاخه نام کاربری که قصد دارید با نام آن وارد سیستم شوید که در شاخه sshd قرار دارد. برای اینکار شما می توانید از فرمان ssh به صورت زیر استفاده کنید:

```
#cat ~/.ssh/id_rsa.pub | ssh admin@192.168.85.132 "mkdir -p ~/.ssh && cat >> ~/.ssh/authorized_keys"
```

```
Terminal
File Edit View Terminal Tabs Help
admin@FreeBSD:~/.ssh % cat ~/.ssh/id_rsa.pub | ssh admin@192.168.85.132 "mkdir -p ~/.ssh && cat >> ~/.ssh/authorized_keys"
Password for admin@UnBound:
admin@FreeBSD:~/.ssh %
```



همانطوری که مشاهده کردید به دلیل اینکه من بر روی کلید خود رمزی را قرار داده بودم قبل از استفاده کردن از این کلید از من رمز عبور را درخواست کرد.

حال اگر قصد داشته باشید که همه کاربران از کلید استفاده کنند باید خط زیر را در فایل `sshd.conf` علامت `#` را از ابتدای خط زیر برداشته.

```
PasswordAuthentication no
```

در مرحله بعد هم استفاده کردن از `pam` را با استفاده از خط زیر و تبدیل کردن به `no` این قابلیت را غیر فعال کنید

```
UsePAM no
```

حال شما اگر قصد داشته باشید به این سرور متصل شوید با خطای زیر مواجه می شوید:

```
Terminal
File Edit View Terminal Tabs Help
root@FreeBSD:/home/admin/.ssh # ssh admin@192.168.85.132
Permission denied (publickey,keyboard-interactive).
root@FreeBSD:/home/admin/.ssh #
```

مسدود کردن دسترسی از طریق رمز عبور

حال با استفاده از `su` وارد بخش کاربر `admin` می شوید و دوباره تلاش می کنید:

```
Terminal
File Edit View Terminal Tabs Help
root@FreeBSD:/home/admin/.ssh # ssh admin@192.168.85.132
Permission denied (publickey,keyboard-interactive).
root@FreeBSD:/home/admin/.ssh # su admin
admin@FreeBSD:~/ .ssh % ssh 192.168.85.132
Enter passphrase for key '/home/admin/.ssh/id_rsa':
Last login: Sun Sep 11 14:56:52 2016 from 192.168.85.128

admin@UnBound:~ % w
 3:05PM up 1:34, 2 users, load averages: 0.15, 0.10, 0.08
USER      TTY      FROM          LOGIN@  IDLE WHAT
root      v0      -             1:32PM  -    -csh (csh)
admin     pts/0    192.168.85.128 3:05PM  -    w
admin@UnBound:~ %
```

اتصال از طریق کلید

همانطوری که ملاحظه کردید می توانید با کلید به سرور متصل شوید.



استفاده از برنامه ssh-agent

برنامه ssh-agent در حقیقت یک راه انداز یا همان launch سایر برنامه هاست و شما می توانید با این برنامه محیط های Shell و یا windows ها دیگری را هم باز کنید. برای اینکار کافیست که از نام یک برنامه shell مثل csh بعد از این برنامه استفاده کنید تا برنامه مورد نظر شما باز شود و در مرحله بعدی کلید خود را یکبار با فرمان ssh-add کلید خود را در حافظه ram بارگذاری کنید تا دیگر نیازی به وارد کردن هر بار رمز کلید خود نداشته باشید، این بخش در شکل زیر نمایش داده شده است:

```

Terminal
File Edit View Terminal Tabs Help
admin@FreeBSD:~/ssh % ssh-agent csh
admin@FreeBSD:~/ssh % ssh-add
Enter passphrase for /home/admin/.ssh/id_rsa:
Identity added: /home/admin/.ssh/id_rsa (/home/admin/.ssh/id_rsa)
admin@FreeBSD:~/ssh % ssh 192.168.85.132
Last login: Sun Sep 11 15:05:26 2016 from 192.168.85.128

admin@UnBound:~ % w
 3:09PM up 1:38, 2 users, load averages: 0.14, 0.12, 0.08
USER      TTY      FROM          LOGIN@  IDLE WHAT
root      v0      -             1:32PM   4 -csh (csh)
admin    pts/0    192.168.85.128 3:09PM   - w
admin@UnBound:~ % exit
logout
Connection to 192.168.85.132 closed.
admin@FreeBSD:~/ssh % ssh 192.168.85.132
Last login: Sun Sep 11 15:09:50 2016 from 192.168.85.128

admin@UnBound:~ % exit
logout
Connection to 192.168.85.132 closed.
admin@FreeBSD:~/ssh % █

```

نمایش استفاده از ssh-agent

همانطوری که مشاهده می کنید برای هر بار برقرار ارتباط دیگر نیازی به وارد کردن رمز تنظیم شده بر روی کلید خود ندارید.

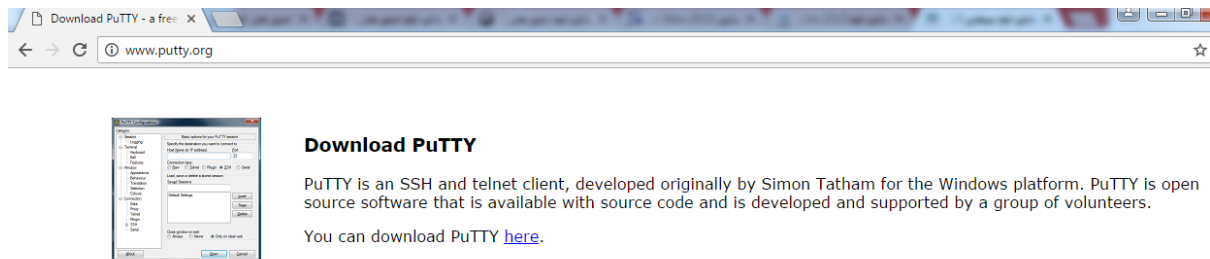


روش اتصال به سرور SSH با putty و کلید عمومی

اگر شما از ویندوز برای انجام کارهای روزمره خود استفاده می کنید فرمان ssh وجود ندارد، نگران نباشید برنامه های برای برقرار اتصال با سرور ssh در ویندوز است که معروفترین آنها برنامه putty است که شما می توانید از آن به صورت رایگان استفاده کنید، این برنامه علاوه بر پروتکل ssh و telnet قابلیت تولید کلید ای public و private را هم برای اتصال با کلید با سرور ssh را دارند. در این مقاله در بخش اول با برقرار ارتباط و در بخش دوم با ساختن کلید و استفاده از کلید بیشتر آشنا می شوید.

دریافت برنامه putty و اتصال به سرور با استفاده از نام کاربری و رمز عبور:

برنامه putty را می توانید از سایت <http://www.putty.org> دانلود کنید صفحه اصلی این سایت به شکل زیر است:



Below suggestions are independent of the authors of PuTTY. They are *not* to be seen as endorsements by the PuTTY project.

سایت putty

برای دانلود بر روی here کلیک کنید تا وارد بخش دیگری از سایت شوید در بخش Binariesa شوید این بخش را در شکل زیر مشاهده می کنید:

Binaries

The latest release version (beta 0.67)

This will generally be a version we think is reasonably likely to work well. If you have a problem with see if we've already fixed the bug, before reporting it.

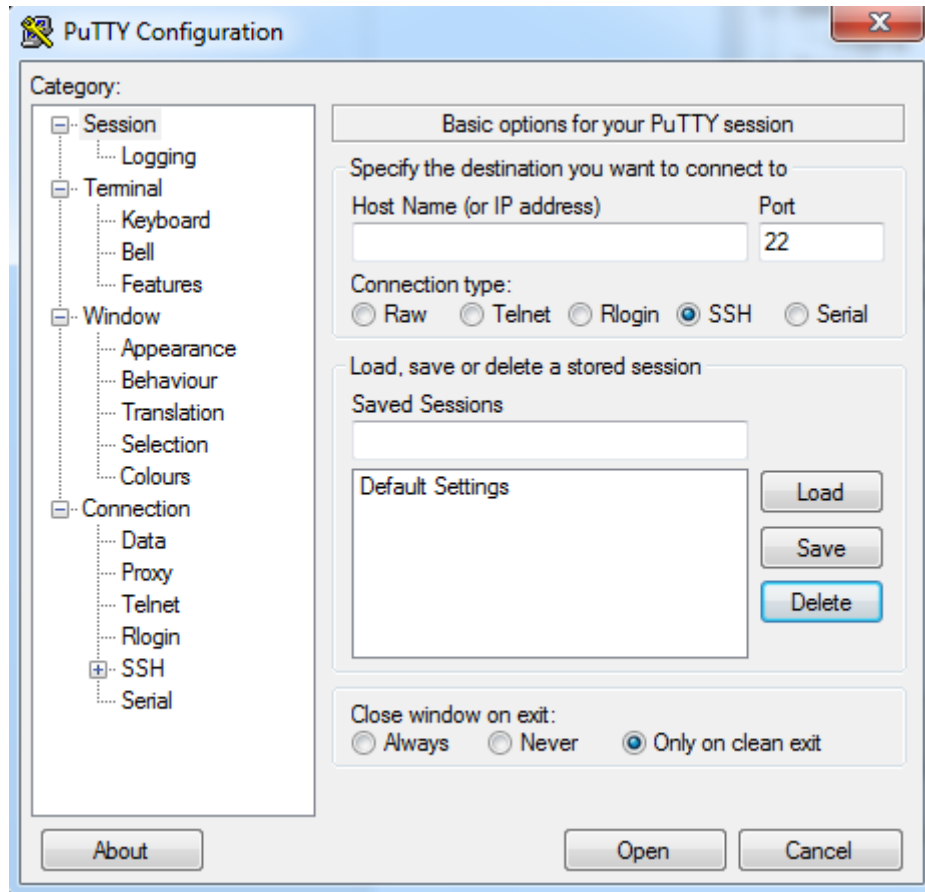
For Windows on Intel x86

PuTTY:	putty.exe	(or by FTP)	(signature)
PuTTYtel:	puttytel.exe	(or by FTP)	(signature)
PSCP:	pscp.exe	(or by FTP)	(signature)
PSFTP:	psftp.exe	(or by FTP)	(signature)
Plink:	plink.exe	(or by FTP)	(signature)
Pageant:	pageant.exe	(or by FTP)	(signature)
PuTTYgen:	puttygen.exe	(or by FTP)	(signature)

لیست فایل های دانلودی

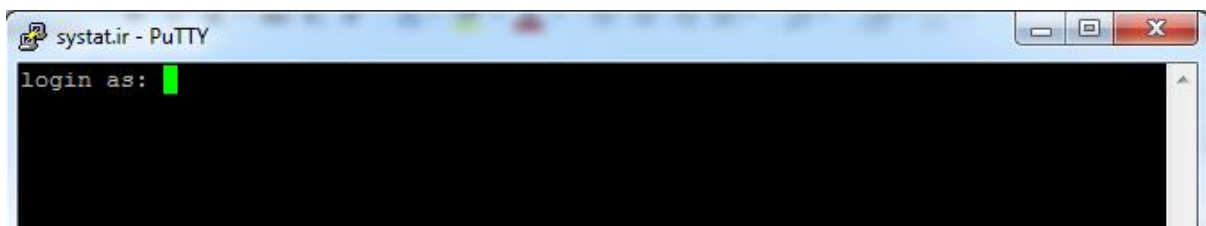


در قدم اول برای یک ارتباط ساده با سرور شما نیاز برنامه خود **putty** به نام **putty.exe** دارید که برای دانلود بروی نام آن کلیک کنید، حجم این برنامه 518 کیلو بایت بوده. بعد از دانلود کردن برنامه را اجرا کنید تا محیطی به صورت زیر برای شما باز شود:



محیط برنامه putty

برای اتصال به سرور مورد نظر خود در بخش **Host Name** یا نام سرور خود را وارد کنید و یا آدرس IP را وارد کنید، در صورتی هم که سرور شما بروی شماره پورت خاصی فعال است می توانید در بخش **Port** شماره مورد نظر سرور خود را وارد کنید، در قسمت زیر هم می توانید نوع ارتباط خود را با سرور مشخص کنید که هم از **telnet** پشتیبانی می کند هم از **ssh** و شما می توانید از این برنامه برای برقرار ارتباط با پورت سریال دستگاه ها استفاده کنید. بعد از وارد کردن مشخصات مورد نظر شما بروی گزینه **Open** کلیک کنید تا به سرور متصل شوید به صورت شکل زیر:



وارد کردن نام کاربری در برنامه putty برای وارد شدن به سیستم



بعد از وارد کردن نام کاربری و Enter کردن شما باید رمز عبور را وارد کنید به صورت شکل زیر:

```
systat.ir - PuTTY
login as: admin
Using keyboard-interactive authentication.
Password for admin@systat.ir: █
```

بخش وارد کردن رمز عبور

بعد از وارد کردن رمز صحیح شما به صورت زیر به سرور متصل می شوید:

```
systat.ir - PuTTY
Password for admin@systat.ir:
Last login: Wed Dec 14 13:46:39 2016 from 5.237.204.101
FreeBSD 10.2-RELEASE (GENERIC) #0 r286666: Wed Aug 12 19:31:38 UTC 2015

Welcome to FreeBSD!

Release Notes, Errata: https://www.FreeBSD.org/releases/
Security Advisories: https://www.FreeBSD.org/security/
FreeBSD Handbook: https://www.FreeBSD.org/handbook/
FreeBSD FAQ: https://www.FreeBSD.org/faq/
Questions List: https://lists.FreeBSD.org/mailman/listinfo/freebsd-questions/
FreeBSD Forums: https://forums.FreeBSD.org/

Documents installed with the system are in the /usr/local/share/doc/freebsd/
directory, or can be installed later with: pkg install en-freebsd-doc
For other languages, replace "en" with a language code like de or fr.

Show the version of FreeBSD installed: freebsd-version ; uname -a
Please include that output and any error messages when posting questions.
Introduction to manual pages: man man
FreeBSD directory layout: man hier

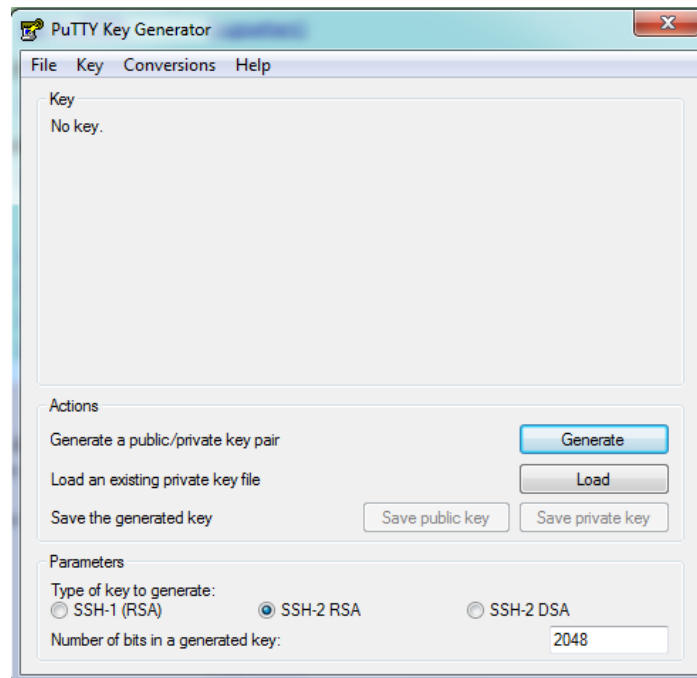
Edit /etc/motd to change this login announcement.
admin@systat:~ % █
```

وارد شدن به سیستم FreeBSD با استفاده از ssh و برنامه putty در ویندوز

حال شما می توانید به راحتی فرمان های مورد نظر خود را وارد کنید. برای خارج شدن از این برنامه کافیست که از exit در شل استفاده کنید.

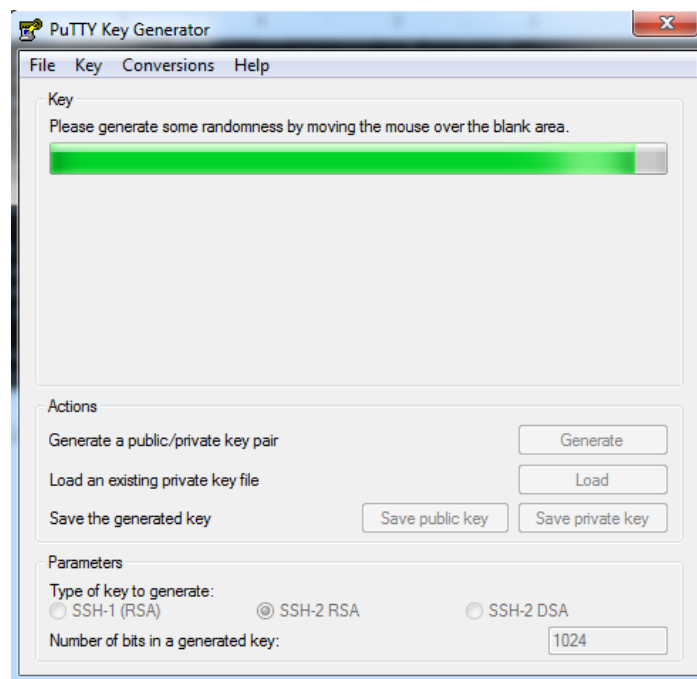
ایجاد کردن کلید RSA باPutty

پروژه putty این امکان را به شما می دهد که کلید های public و private را برای برقرار ارتباط با کلید به سرور SSH را ایجاد کنید. برای این کار شما به برنامه ای به نام puttygen.exe نیاز دارید، این برنامه را از سایت <http://www.putty.org> دانلود کنید و برنامه را اجرا کنید تا وارد محیط برنامه به صورت زیر شوید حجم این برنامه 138 کیلو بایت است:



برنامه putty key gene

شما می توانید با استفاده از این برنامه کلید های RSA برای SSH1 و SSH2 ایجاد کنید و حتی کلید های ایجاد شده را در این برنامه برای نمایش بارگذاری کنید، حتی شما می توانید طول کلید را هم انتخاب کنید، بعد از انتخاب کردن گزینه های مورد نظر خود بروی کلید Generate کلیک کنید تا کلید برای شما ایجاد شود به صورت نمایش داده شده در شکل زیر:



برنامه در حال تولید کردن کلید



بعد از اتمام این مرحله برنامه پیغام هایی به زیر برای شما نمایش می دهد:



اتمام تولید کلید توسط برنامه PuTTY Key Generator

در کادر .. Public key for pasting شما باید این متن را در فایل `authorized_keys` موجود در شاخه `ssh`. کاربر مورد نظر خود کپی کنید. همچنین شما می توانید برای کلید خود رمز هم در بخش `key passphrase` تعریف می کنید.

برای ذخیره کردن کلیدهای `public` و `private` در بخش `Save the generated key` می توانید این دو کلید را `save` کنید. کلید `private` را در محلی مناسب قرار دهید و کلید `public` را می توانید در هر جایی که دوست دارید منتشر کنید، حتی بروی سایت خود.

حال به سرور خود با `putty` وصل شوید و با کپی و `paste` کردن کلید `public` خود را به سرور منتقل کنید، به صورت زیر:

```
systat.ir - PuTTY
admin@systat:~ % mkdir .ssh
admin@systat:~ % cd .ssh/
admin@systat:~/.ssh % touch authorized_keys
admin@systat:~/.ssh %
```

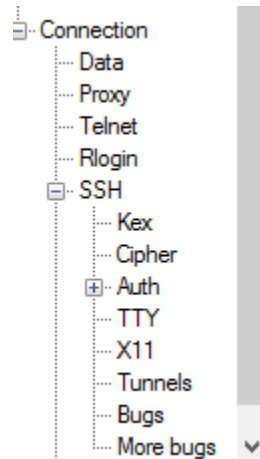
ایجاد کردن فایل مورد نظر برای انتقال کلید عمومی

```
^[(escape) menu ^y search prompt ^k delete line ^p prev li ^g prev page
^o ascii code ^x search ^l undelete line ^n next li ^v next page
^u end of file ^a begin of line ^w delete word ^b back 1 char
^t top of text ^e end of line ^r restore word ^f forward 1 char
^c command ^d delete char ^j undelete char ^z next word
=====line 1 col 217 lines from top 1 =====
jckhsyO12dM3ZSryONJWXEeaWh8jVcUlFrCC/i99MSsCJPGD52YtEK0= rsa-key-20161214
```

مراحل کپی کردن کلید عمومی

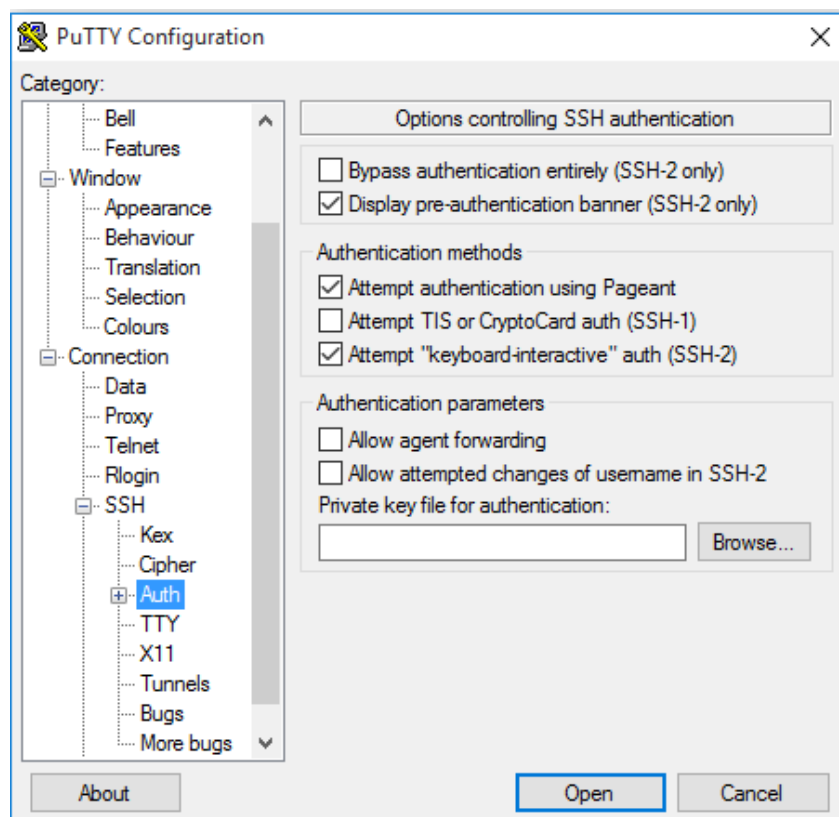


حال در حال باید با استفاده از برنامه putty کلید private خود را بارگذاری کنید، برای این کار وارد برنامه putty شوید و در بخش Connections وارد بخش ssh شوید و بروی بخش AUTH کلیک کنید، این بخش رو در شکل زیر مشاهده می کنید:



منوی دسترسی به AUTH در putty

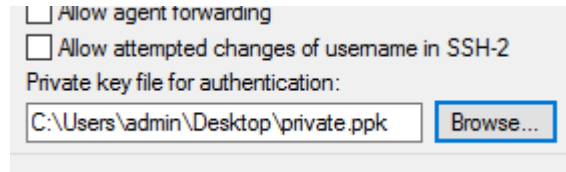
صفحه ای به صورت زیر برای شما نمایش داده می شود:



بخش AUTH در Putty

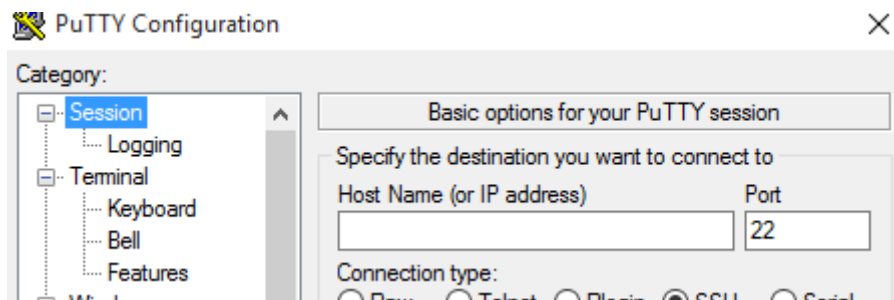


در بخش Private key file for authentications برروی گزینه Browse رفته کلیک کنید و فایل کلید شخصی خود را وارد کنید مثل شکل زیر:



وارد کردن کلید به برنامه putty

حال به بخش sessions رفته تا صفحه اولیه برنامه برای شما باز شود تا مشخصات سرور را وارد کنید، مثل شکل زیر:



بازگشت به بخش sessions

در صفحه login نام کاربر را وارد کنید که فایل کلید عمومی را در شاخه ssh. آن کپی کردین و دیگر نیازی به استفاده از رمز عبور نیست و با استفاده از کلید وارد سیستم شده اید.



برنامه syslog در FreeBSD

در این بخش با مباحث مربوط به فایل‌های Log در FreeBSD آشنا می‌شوید، یکی از بحث‌های مهم در سیستم عاملها توان آن سیستم آن در ثبت واقع است. یکی از وظایف هر مدیر سیستم آشنا با انواع و اقسام فایل‌های Log است. در این مدل از فایل‌ها اطلاعات سخت‌افزاری، نرم‌افزاری و خطاهای برنامه‌ها ذخیره می‌شود، این اطلاعات برای امنیت سیستم شما بسیار مفید است. بیشتر برنامه‌ها و سرویس‌ها می‌توانند فایل‌های Log تولید کرده و در محلی که شما معین می‌کنید آنها را ذخیره کنند، حتی می‌توانید حجم فایلها را مدیریت کنید و به اندازه‌ی مورد مظر شما رسید به اصطلاح rotate شده و به تعدادی که شما معین کرده‌اید ذخیره می‌شوند.

اکثریت این فایلها در زیر شاخه‌ای به نام `/var/log` ذخیره می‌شود در شکل زیر شما فایل‌های این زیر شاخه را مشاهده می‌کنید:

```

Terminal
File Edit View Terminal Tabs Help
root@FreeBSD:/var/log # ls
ConsoleKit          dmesg.yesterday      sendmail.st.1
Xorg.0.log          lpd-errs              sendmail.st.2
Xorg.0.log.old      maillog               sendmail.st.3
aculog              maillog.0.bz2         sendmail.st.4
atop                maillog.1.bz2         sendmail.st.5
auth.log            messages              sendmail.st.6
bsdinstall_log     messages.0.bz2       setuid.today
cron                messages.1.bz2       setuid.yesterday
cups                messages.2.bz2       squid
debug.log           mount.today           userlog
debug.log.0.bz2    mount.yesterday      utx.lastlogin
debug.log.1.bz2    pf.today              utx.log
debug.log.2.bz2    ppp.log               utx.log.0
debug.log.3.bz2    security              xferlog
devd.log            sendmail.st
dmesg.today         sendmail.st.0
root@FreeBSD:/var/log #

```

فایل‌های موجود در شاخه Log

در سیستم عامل FreeBSD از برنامه‌ای به نام `syslogd` برای مدیریت کردن Log ها استفاده می‌کند. به صورت پیش فرض این برنامه در زمان راه‌اندازی سیستم راه‌اندازی می‌شود و شما می‌توانید با استفاده از فایل `rc.conf` نقش‌های مورد نظر خود را با استفاده از `flags` ها با خط `syslog_flags` در فایل `rc.conf` مدیریت کنید.

پیکربندی syslog در FreeBSD

برنامه `syslog` یک فایل پیکربندی در مسیر `/etc` به نام `syslog.conf` دارد. در ادامه با این فایل بیشتر آشنا می‌شوید. در شکل زیر شما نمونه‌ای از این فایل را مشاهده می‌کنید:



```
# Spaces ARE valid field separators in this file. However,
# other *nix-like systems still insist on using tabs as field
# separators. If you are sharing this file between systems, you
# may want to use only tabs as field separators here.
# Consult the syslog.conf(5) manpage.
*.err;kern.warning;auth.notice;mail.crit /dev/console
*.notice;authpriv.none;kern.debug;lpr.info;mail.crit;news.err /var/log/message
security.* /var/log/security
auth.info;authpriv.info /var/log/auth.log
mail.info /var/log/maillog
lpr.info /var/log/lpd-errs
ftp.info /var/log/xferlog
cron.* /var/log/cron
!-devd
*.debug /var/log/debug.log
*.emerg *
```

فایل برنامه syslog

همانطوری که مشاهده می کنید این فایل دارای دو بخش اصلی است، بخش اول نام برنامه ها و یا قابلیت های سیستم است که قصد دارید از آنها گزارش بگیرید که خودش به دو بخش **level** و **facility** تقسیم می شود که در ادامه با آن بیشتر آشنا می شوید و بخش دوم مسیری است که فایل گزارش مربوطه در آن ذخیره می شود.

بخش **facility** نام برنامه ها و دایمون هایی است که سیستم از آنها گزارش می گیرد مثل **uth, authpriv, console, cron, daemon ftp, kern, lpr, mail, mark, news, ntp, security, syslog, user, uucp** که برای برنامه هایی است که شما قصد دارید به این لیست اضافه کنید.

سطح یا همون **Level** میزان اطلاعات و سطح حساس دریافت اطلاعات را از برنامه مشخص می کند، که تربیت در آن مهم است که از سمت چپ به راست در لیست زیر از درجه اهمیت آن کاسته می شود:

emerg, crit, alert, err, warning, notice, info و **debug**.

در برخی از موارد مشاهده کرده اید که در زمان استفاده از سیستم پیغام های خطایی برای شما در صفحه اصلی نمایش داده می شود این پیغام ها به دلیل تنظیمات خط اول شما نمایش داده می شود. این خط شامل خط زیر است:

```
*.err;kern.*;auth.notice;authpriv.none;mail.crit /dev/console
```

بسیار مشاهده شده است که در زمان بارگذاری ماژول در هسته شما پیغام های هسته را مشاهده می کنید این بخاطر بخش **kern.*** است در بخش دوم این خط مسیر ذخیره سازی فایل نمایش داده شده است که مسیر آن **/dev/console** است.

در زیر شاخه **/var/log** فایل جامعی است از گزارشات به نام **messegas** که اطلاعات کاملتری از سیستم شما در آن ذخیره میشود این فایل با استفاده از تنظیم کردن خط زیر ایجاد شده:

```
*.notice;authpriv.none;kern.debug;lpr.info;mail.crit;news.err /var/log/messages
```

اگر شما به دنبال گزارشات امنیتی مثل از سیستم خود هستید فایل **security** را چک کنید.

همانطوری که در خطوط بالا مشاهده کردید برای نوشتن **level** و **facility** می توانید از قواعد جدا کننده با ; استفاده کنید و اگر منظور شما همه چیز باشد از * استفاده کنید.



برنامه newsyslog در FreeBSD

فایل‌های Log در بعضی از موارد به سرعت رشد کرده و بزرگ می‌شوند، این امر می‌تونه از فضای دیسک شما کاسته و کارایی سیستم شما را به خطر بندازد. در FreeBSD برنامه‌ای وجود دارد به نام Newsyslog که هدف اصلی آن مدیریت کردن فایل‌های log است. این برنامه دارای قابلیت‌هایی است که می‌تواند به صورت خودکار فایل‌ها را بر حسب زمان یا حجم به فایل‌های کوچکتر تقسیم کند و یا حتی آنها را فشرده کند تا در میزان استفاده از فضا هم صرفه جویی شود. هر برنامه‌ای مثل syslog می‌تواند فایل Log ایجاد کند، این برنامه Newsyslog است که از طریق برنامه Cron راه اندازی می‌شود و اعمال خواسته شده را انجام می‌دهد. به صورت پیش فرض این برنامه هر یک ساعت یکبار اجرا می‌شود که البته این مقدار در فایل پیکربندی این برنامه قابل تغییر است.

فایل پیکربندی newsyslog.conf

این برنامه شامل یک فایل پیکربندی است به نام newsyslog.conf در مسیر /etc، این فایل شامل خطوطی است که هر خط معرف یک فایل log است که در ابتدای این خط نوشته می‌شود و در ادامه اگر شما قصد داشته باشید که دسترسی خاصی گروه یا کاربری بدهید می‌توانید در بخش بعدی آنرا ذکر کنید. در ادامه نوع سطح دسترسی را به فایل‌هایی که ایجاد شده است را می‌توانید مشخص کنید (برای اطلاعات بیشتر در مورد سطح دسترسی به مقاله آن مراجعه کنید). بخش از این فایل در شکل زیر نمایش داده شده:

```

#
# Note: some sites will want to select more restrictive protections than the
# defaults. In particular, it may be desirable to switch many of the 644
# entries to 640 or 600. For example, some sites will consider the
# contents of maillog, messages, and lpd-errors to be confidential. In the
# future, these defaults may change to more conservative ones.
#
# logfile            [owner:group]  mode count size when  flags [/pid_file]
# [sig_num]
/var/log/all.log     600  7    *    @T00  J
/var/log/amd.log     644  7    100  *    J
/var/log/auth.log    600  7    100  @0101T JC
/var/log/console.log 600  5    100  *    J
/var/log/cron        600  3    100  *    JC
/var/log/daily.log   640  7    *    @T00  JN
/var/log/debug.log   600  7    100  *    JC
/var/log/init.log    644  3    100  *    J
/var/log/kerberos.log 600  7    100  *    J
/var/log/lpd-errors  644  7    100  *    JC
/var/log/maillog     640  7    *    @T00  JC
/var/log/messages    644  5    100  @0101T JC
/var/log/monthly.log 640  12   *    $M1D0 JN
/var/log/pflog       600  3    100  *    JB    /var/run/pfl

```

بخش از فایل newsyslog



بخش count

شما می توانید تعداد فایل‌های ایجاد شده را مدیریت کنید این کار را در بخش count مشخص کنید، برای مثال شما قصد دارید که 7 عدد از این فایلها را ذخیره کنید و بعد از اینکه فایل 88 ایجاد شد فایل اولی به صورت خودکار پاک خواهد شد.

بخش size و when

شما می توانید به دو صورت برای ایجاد کردن فایلها محدودیت اعمال کنید، محدودیت بر اساس میزان حجم فایل و مدت زمانی خاص، برای هر فایل باید یکی از این روشها را انتخاب کنید، اگر هم دو محدودیت را اعمال کنید هر کدام که زودتر اتفاق بیفتد آن محدودیت اعمال می شود، برای مثال شما می خواهید که فایل شما در زمان خاصی ایجاد شود و فایل قبلی ذخیره شود و همچنین گفته اید که به 100 مگابایت برسد هم این عمل را انجام دهد، هر کدام که زودتر اتفاق رخ دهد آن عمل اعمال می شود. شما در بخش size میزان حجم فایل را بر حسب کیلوبایت مشخص می کنید، اگر در این بخش علامت * باشد بدین معناست که این بخش شما محدودیتی نیست.

در بخش زمان شما به دو روش می توانید زمان مورد نظر خود را وارد کنید

روش اول استفاده از استاندارد ISO 8601

این فرمت در بخش when با علامت @ شروع می شود. فرمت کلی این ساختار زمانی به صورت زیر است:

```
[[[[[cc]yy]mm]dd][T[hh[mm[ss]]]]]]
```

بخش اول یا همان CC مربوط به قرن می شود که می تواند عددی بین 00 تا 99 در آن وارد کنید این بخش در بسیاری از موارد نوشته نمی شود.

بخش دوم یا همان yy است که در این بخش باید سال مورد نظر خود را وارد کنید

بخش mm که مربوط همیشه به عدد ماه مورد نظر شما و بخش dd هم مربوطه همیشه به روزی که مورد نظر شماست.

نکته :

اگر شما بخش بالا یا همان بخش Date رو وارد کنید سیستم به صورت خودکار از زمان جاری شما استفاده می کند.

بخش دوم از این فرمت زمانی بعد از حرف T قرار میگیرد که این بخش خودش به 3 بخش دو عددی تقسیم میشه.

بخش hh مربوط میشود به ساعت مورد نظر شما.

بخش mm مربوط می شود به دقیق مورد نظر شما.

بخش ss هم مربوط می شود به ثانیه مورد نظر شما.

فرمت بالا در فایل newsyslog با علامت @ شروع می شود. و تنظیم کردن آن راحت نیست.



روش اول استفاده از حالت: Day Week Month Time

شما در این بخش می توانید به راحتی با استفاده از سه حرف D W M زمان های مورد نظر خود را بنویسید. قالب اصلی این فرمت به صورت زیر است:

[Dhh], [Ww[Dhh]], and [Mdd[Dhh]],

در این فرمت زمان اختیار نیمه شب در نظر گرفته شده است. در زیر رنج هر بخش توضیح داده شده است:

در بخش HH زمان را مشخص می کند و عددی بین 0 تا 23 می تواند در این بخش مورد استفاده قرار گیرد.

در بخش W شامل روزهای هفته است که اعداد آن بین 0 تا 6 است و اولین روز هفته یکشنبه یا Sunday است.

بخش dd روز های ماه است که عددی است بین 1 تا 31 و اگر می خواهید اولین روز هر ماه باش از حرف L و برای آخرین روز ماه از A استفاده کنید.

در زیر شما با چند مثال از این قالب های زمانی آشنا می شوید:

\$D0	فایلها هر شب ساعت 12 rotate می شود.	@T00
\$D23	فایلها هر روز ساعت 23 rotate می شود.	@T23
\$W0D23	فایلها هر هفته در روز یکشنبه در ساعت 23 rotate می شود.	
\$W5D16	فایلها هر هفته در روز جمعه و در ساعت 16 rotate می شود.	
\$M1D0	فایلها در اولین روز هر ماه و در ساعت 23 rotate می شود.	@01T00
\$M5D6	فایلها در روز 5 هر ماه ساعت 6 rotate می شود.	@05T06

بخش: Flags

شما می توانید در این بخش یکی یا چند Flag را در این بخش برای فایل خود اضافه کنید که هر Flag کار خاصی انجام می دهد و شما در ادامه با معروف ترین آنها آشنا می شوید:

این Flag فایل مورد نظر شما را در قالب باینری ذخیره می کند.

C استفاده از این Flag باعث می شود که اگر فایل گزارش شما وجود نداشت برای شما ایجاد کند.

این Flag باعث می شود که برنامه newsyslog با استفاده از BZIP فایلهای شما را فشرده کند و در مصرف فضا صرفه جویی کند.

این Flag باعث می شود که برنامه newsyslog با استفاده از XZ فایلهای شما را فشرده کند و در مصرف فضا صرفه جویی کند.



Z این Flag باعث می شود که برنامه newsyslog با استفاده از Gzip فایل‌های شما را فشرده کند و در مصرف فضا صرفه جویی کند.

سرور Log در FreeBSD

خواندن فایل‌های Log هر سیستم به صورت جداگانه کاری بسیار وقت گیر است، در FreeBSD قابلیت وجود دارد به نام log server که از دو برنامه syslog و newsyslog استفاده می کند و این امکان را برای شما مهیا می کند که از یک سیستم مرکزی تمام گزارشات مورد نظر خودت رو بخوانید و از قابلیت های دو برنامه ذکر شده به صورت کامل استفاده کنید.

قبل از شروع به کار کردن و پیکربندی سرور و کایننت در این سیستم شما باید به دو نکته زیر توجه کنید:

1. برنامه syslog برای دریافت بسته های اطلاعاتی خود در شبکه از شماره پورت 514 در حالت UDP استفاده می کند و شما قبل از راه اندازی و پیاده سازی این سیستم از باز بودن و در دسترس بودن پورت مورد نظر در سمت سرور خود مطمئن شوید و چک کنید که فایروال سیستم شما دسترسی به این پورت را محدود نکرده باشد.
2. برنامه syslog به سرور DNS وابسته است و برای ارسال و دریافت بسته های شبکه از سرویس DNS استفاده می کند اگر در شبکه خود DNS دارید آنرا با نام های مورد نظر خود به درستی پیکربندی کنید یا برروی دو یا هر چند سیستم که در این طراحی قرار میگیرد در فایل /etc/hots تنظیمات خود را انجام دهید.

پیکربندی سرور Log

این بخش شامل دو قسمت است که در بخش اول شما کایننت را مشخص می کنید و در بخش دوم سرور log را برروی شبکه تنظیم می کنید.

بخش اول اضافه کردن کلاینت در سرور:

برای شروع باید فایل /etc/syslog.conf را ویرایش کنید، در این فایل در یک خط خالی شما باید نام کلاینت را که قصد دارید گزارشات Log را از آن دریافت کنید با + شروع کنید و نام را ذکر کنید، در بخش بعدی هم باید سطح log را مشخص کنید و مسیر ذخیره شدن فایل را تعیین کنید:

```
+client.mabedini.com
*.* /var/log/logclient.log
```

راه اندازی سرور:

حال بعد از اضافه کردن کلاینتها نوبت به پیکربندی سرور است، برای انجام دادن این عمل شما باید مقادیری را به فایل /etc/rc.conf به صورت زیر اضافه کنید:

```
syslogd_enable="YES"
```



```
syslogd_flags="-a logclient.example.com -v -v"
```

در خط اول راه اندازی خودکار در زمان boot را فعال می کنید و در خط دوم syslogd را با سویچ a راه اندازی میکند و نام کلاینت را در این بخش مشخص می کنید و -v -v اطلاعات کامل تری را برای شما نمایش می دهد.

نکته :

شما در مقابل سویچ a می توانید مقادیر دیگر را هم ذکر کنید به شرح زیر :

شما می توانید از آدرس IP استفاده کنید و بعد از آدرس IP و Subnetmask بعد از : می توانید شماره پورت UDP سرور خود را هم مشخص کنید.

شما می توانید از کاراکتر * هم استفاده کنید و نام چندین کلاینت را در یک خط در بخشی که مشترک هستند از مشخص کنید به صورت *.mabedini.com این خط به این معناست که بخش * هر نامی که شما قرار دهید می تواند باشد.

در بخش بعدی شما باید فایل log مورد نظر و ذکر شده در فایل syslog.conf را ایجاد کنید برای این کار از فرمان زیر استفاده کنید:

```
# touch /var/log/logclient.log
```

حال زمان راه اندازی مجدد سرویس syslogd با استفاده از فرمان service است و در بخش بعدی هم برای اطمینان از راه اندازی درست سرویس از فرمان pgrep استفاده کنید، در ادامه فرمان های مورد نیاز این بخش را مشاهده می کنید:

```
# service syslogd restart
```

```
# pgrep syslog
```

پیکربندی Client:

سیستم کلاینت وظیفه ارسال کردن فایل های گزارش به سمت سرور را دارد و یک نسخه هم به صورت local ذخیره می کند. در بخش کلاینت هم شما نیاز دارید که خطوط زیر را در فایل /etc/rc.conf اضافه کنید:

```
syslogd_enable="YES"
```

```
syslogd_flags="-s -v -v"
```

خط اول راه اندازی سرویس را در زمان boot سیستم بر عهده دارد و در خط اون هم سویچ S باعث فعال شدن ارسال گزارشات سمت سرور به صورت می شود، در قدم بعدی شما باید در فایل /etc/syslog.conf خطی را اضافه کنید برای ارسال کردن گزارشات به سمت سرور در بخش زیر این خط را مشاهده می کنید که همه سطح های گزارشات را به سمت سرور ارسال کنید که نام سرور با @ شروع می شود.

```
*.* @logserv.mabedini.com
```



حال سرویس syslog را به صورت زیر دوباره راه اندازی کنید:

```
# service syslogd restart
```

برای چک کردن و ایجاد کردن یک گزارش تستی از فرمان logger استفاده کنید این برنامه پیغام شما را در فایل /var/log/messages ذخیره می کند و روش اجرا این فرمان را در شکل زیر مشاهده می کنید:

```
# logger "Test message from client"
```

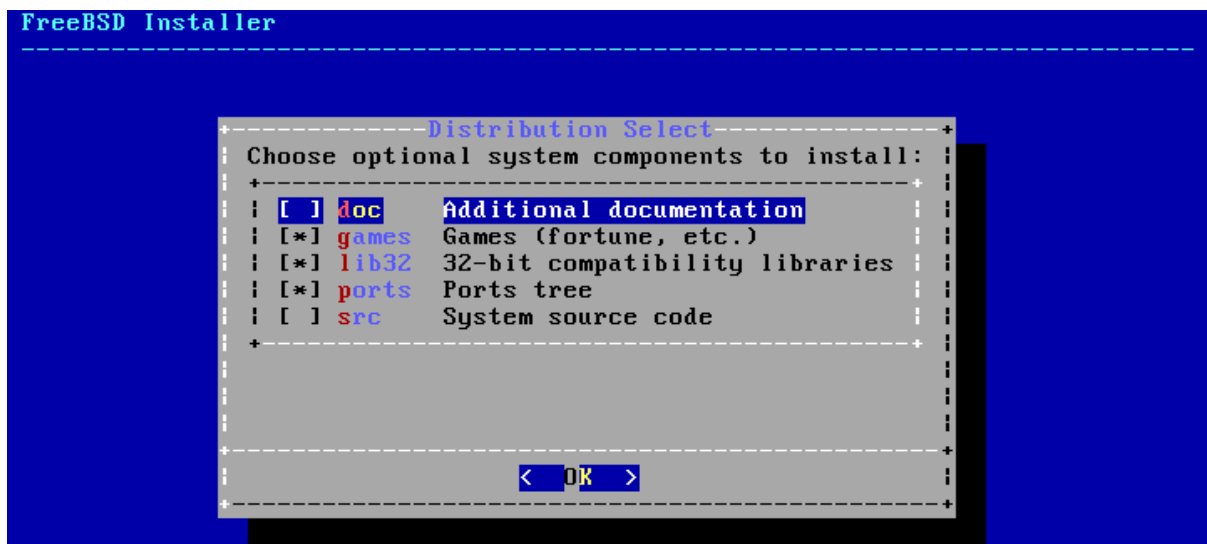


پیکربندی هسته در FreeBSD

با گسترش روز افزون استفاده از ماژول ها و افزایش سرعت سیستم و پشتیبانی از همه سخت افزار های موجود در هسته FreeBSD کمتر کاربران این سیستم عامل نیازی به کامپایل کردن هسته را دارند چونکه این کار بسیار زمانبر هم هست. ولی در موارد بعضی از قابلیت های تخصصی فقط در هسته کامپایل شده و شما نیاز به دانش کامپایل کردن دارید. در این مقاله راه های نصب سورس FreeBSD و بعد کامپایل کردن هسته آموزش داده می شود.

نصب سورس کد FreeBSD

شما به دو روش می توانید این کار را انجام دهید، راه اول در زمان نصب سیستم عامل است و راه دوم به صورت دستی این کار را انجام دهید. در زمان نصب در بخش Distributions Select شما با گزینه های نمایش داده در شکل زیر مواجه می شود:



منوی نصب کردن سورس FreeBSD در زمان نصب

اگر شما بهش SRC را انتخاب کنید سورس سیستم عامل برای شما به صورت خودکار نصب می شود، شاخه این بخش در `/usr/src` است اگر وارد این شاخه شده اید و زی شاخه هایی در آن قرار داشت به این معناست که سورس نصب شده است، در شکل زیر خروجی فرمان `ls` را در این زیر شاخه مشاهده می کنید:

```

File Edit View Terminal Tabs Help
root@FreeBSD:/ # cd /usr/src/
root@FreeBSD:/usr/src # ls
.arccconfig      cddl             rescue
.arclint         contrib         sbin
COPYRIGHT        crypto          secure
LOCKS           etc             share
MAINTAINERS      games          sys
Makefile         gnu            tests
Makefile.incl   include        tools
ObsoleteFiles.inc kerberos5     usr.bin
README          lib            usr.sbin
UPDATING        libexec
bin             release
root@FreeBSD:/usr/src #
  
```



این شاخه دارای زیر شاخه های زیادی است که یکی از معروفترین آنها شاخه SYS است که در زیر شاخه آن فایل‌های مربوط به ساختارهای سخت افزاری به نام های amd64 i386 و ... قرار دارد، به تناسب نوع سیستم عاملی که نصب کرده اید مثل amd64 که به FreeBSD 64 بیتی است وارد آن شاخه شوید ، فایل‌های موجود در این شاخه را مشاهده می کنید:

```

Terminal
File Edit View Terminal Tabs Help
root@FreeBSD:/usr/src/sys # cd amd64/
root@FreeBSD:/usr/src/sys/amd64 # ls
Makefile      compile      include      pci
acpica        conf         linux        vmm
amd64         ia32        linux32
root@FreeBSD:/usr/src/sys/amd64 # cd conf/
root@FreeBSD:/usr/src/sys/amd64/conf # ls
DEFAULTS      GENERIC      GENERIC.hints  Makefile      NOTES
root@FreeBSD:/usr/src/sys/amd64/conf #

```

شاخه اصلی محل هسته

در این زیر شاخه شاخه است به نام conf که که فایل متنی اصلی در آن قرار دارد به نام GENERIC که یک فایل متنی است و شما با ویرایشگر متنی می توانید آنرا باز کنید و بخشهایی که می خواهید به هسته اضافه و یا از هسته کم کرد را در این فایل می توانید آنرا اضافه و کم کنید، چند خط اولیه این فایل را در شکل زیر مشاهده می کنید:

```

Terminal
File Edit View Terminal Tabs Help
#
# $FreeBSD: releng/10.3/sys/amd64/conf/GENERIC 286132 2015-07-31 15:25:07Z gjb $
cpu          HAMMER
ident       GENERIC

makeoptions  DEBUG=-g          # Build kernel with gdb(1) debug symbols
makeoptions  WITH_CTF=1       # Run ctftool(1) for DTrace support

options      SCHED_ULE        # ULE scheduler
options      PREEMPTION       # Enable kernel thread preemption
options      INET             # InterNETworking
options      INET6           # IPv6 communications protocols
options      TCP_OFFLOAD     # TCP offload
options      SCTP          # Stream Control Transmission Protocol
options      FFS           # Berkeley Fast Filesystem
options      SOFTUPDATES   # Enable FFS soft updates support
options      UFS_ACL      # Support for access control lists
options      UFS_DIRHASH  # Improve performance on big directories
options      UFS_GJOURNAL # Enable gjournal-based UFS journaling
options      QUOTA       # Enable disk quotas for UFS
options      MD_ROOT     # MD is a potential root device
options      NFSCL       # New Network Filesystem Client
ESC

```

فایل اصلی هسته

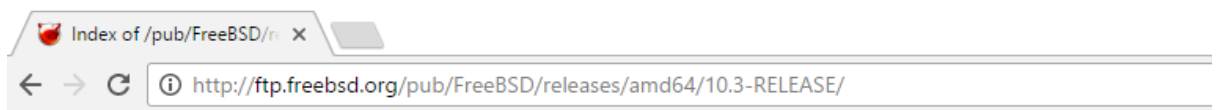
قواعد این فایل بسیار ساده است ، شما می توانید در خط نمایش داده شده در بالا شکل ساختار هسته ای را که این فایل ایجاد می کند را مشاهده کنید، هر خط شامل یک قابلیت است، برای اضافه کردن هر قابلیت از کلمه کلیدی options استفاده کنید



و در روبروی آن نام قابلیت را اضافه کنید، برای مثال برای اضافه کردن NFS Client به هسته باید کلمه NSFCL را بعد از options اضافه کنید.

نصب کردن فایل سورس به صورت دستی:

اگر در زمان نصب SRC را نصب نکردید باید از این راه برای نصب استفاده کنید و ابتدا فایل src.txz را دانلود کنید. ابتدا باید وارد ftp سرور سایت freebsd.org شوید، آدرس <http://ftp.freebsd.org/pub/FreeBSD/releases> یکی از آدرسهای این بخش است، حال نوع ساختار را انتخاب کنید و بخش بعدی ورژن سیستم عاملی را که نصب کرده اید انتخاب کنید صفحه ای بصورت زیر برای شما باز می شود:



Index of /pub/FreeBSD/releases/amd64/10.3-

File Name ↓	File Size ↓	Date
Parent directory/	-	-
MANIFEST	782	201
base.txz	70325324	201
doc.txz	1432464	201
games.txz	886184	201
kernel.txz	97807424	201
lib32.txz	17545052	201
ports.txz	35045976	201
src.txz	126900216	201

دانلود کردن سورس از طریق سایت

حال می توانید فایل src.txz را دانلود کنید و در شاخه FreeBSD / خود قرار دهید،

```

Terminal
File Edit View Terminal Tabs Help
root@FreeBSD:/ # ls
.cshrc      bhyve      home       proc       tmp
.profile    bin        lib        rescue    ttyrecord
.rnd        boot       libexec    root       usr
.snap       dev        mabedini   sbin      var
.sujournal  entropy    media      src.txz
COPYRIGHT  etc        mnt        sys
root@FreeBSD:/ #

```

نمایش فایلهای دانلود شده



حال با استفاده از فرمان tar و سویچ های zxvf این فایل را از حالت فشرده خارج کنید تا در زیر شاخه /usr/src و فایل‌های سورس برای شما ایجاد شود.

حتما فایل سورس را از شاخه / extract / کنید.

خروجی و پایان این فرمان برای شما در شکل زیر نمایش داده شده است:

```

Terminal
File Edit View Terminal Tabs Help
x usr/src/games/fortune/tools/Troff.sed
x usr/src/games/fortune/tools/do_sort
x usr/src/games/fortune/fortune/Makefile
x usr/src/games/fortune/fortune/fortune.6
x usr/src/games/fortune/fortune/pathnames.h
x usr/src/games/fortune/fortune/fortune.c
x usr/src/games/morse/Makefile
x usr/src/games/morse/morse.c
x usr/src/games/morse/morse.6
x usr/src/games/caesar/caesar.c
x usr/src/games/caesar/Makefile
x usr/src/games/caesar/caesar.6
x usr/src/games/caesar/rot13.sh
x usr/src/games/factor/factor.c
x usr/src/games/factor/Makefile
x usr/src/games/factor/factor.6
x usr/src/games/number/number.6
x usr/src/games/number/Makefile
x usr/src/games/number/number.c
x usr/src/games/tests/Makefile
x usr/src/games/grdc/Makefile
x usr/src/games/grdc/grdc.6
x usr/src/games/grdc/grdc.c
root@FreeBSD:/ #

```

Extract کردن فایل‌های هسته

نصب هسته جدید:

این کار با دو فرمان انجام شده و بعد شما برای راه اندازی سیستم خود با هسته جدید نیازی به راه اندازی مجدد سیستم دارید، نصب هسته در FreeBSD تنها عملی است که نیازی به restart کردن کل سیستم دارد. در ادامه مراحل این کار ذکر می شود:

شما هم می توانید بروی فایل اصلی هسته خود تغییرات را اعمال کنید و هم پیشنهاد می شود که از فایل GENERIC یک کپی گرفته و تغییرات را در آن اعمال کنید، در فرمان های زیر ابتدا وارد شاخه مورد نظر شوید و در بخش دوم با فرمان cp یکی کپی از فایل اصلی ایجاد می کنید:

```

# cd /usr/src/sys/amd64/conf
# cp GENERIC MYKERNEL

```




بعد از اعمال تغییرات حال شما به سر شاخه `/usr/src` رفته و اولین فرمان را به صورت زیر اجرا کنید:

```
# make buildkernel KERNCONF=MYKERNEL
```

به این نکته توجه کنید که `KERNCONF=MYKERNEL` در حقیقت نام همان فایلی است که بخش قبلی از آن کپی گرفته اید، اگر هم شما بر روی فایل اصلی تغییرات را ایجاد کرده اید نیازی به این بخش در اجرا فرمان ندارید.

بعد از اتمام این فرمان اگر شما با خطایی مواجه نشوید بخش کامپایل کردن هسته جدید تمام شده و شما حال نیاز به نصب کردن آن دارید، برای انجام این کار کفایت که فرمان زیر را اجرا کنید:

```
# make installkernel KERNCONF=MYKERNEL
```

اگر این بخش هم بدون خطا به اتمام برسد هسته جدید شما نصب شده و فقط شما نیاز دارید که سیستم را `restart` کنید تا FreeBSD با هسته جدید شما راه اندازی شود.



فایروال IPFW در FreeBSD

در سیستم عامل FreeBSD سه نوع فایروال با امکانات خواص برای هر کدام وجود دارد که یکی از بهترین و قدیمیترین آنها فایروال ipfw است که از ورژنهای 4 و 6 آدرس ip پشتیبانی می کند و دارای قابلیت های گزارشگیری، Nat کردن، مدیریت کردن ترافیک با استفاده از dumynet، قابلیت bridge و در نهایت با استفاده از قابلیت ipstealth به شما ان امکان را می دهد که فایروال خود را با تغییر ندادن ttl از دید کاربران شبکه ای که از فرمان tracertr استفاده می کنند مخفی کنید.

علاوه بر این فایروال دو فایروال دیگری هم به نام های pf و ipfilter که هر کدام در بخشهای مورد نظر توضیح داده خواهد شد ولی در این بخش هدف ما آموزش در مورد استفاده از ipfw است.

راه اندازی IPFW

به دو شیوه شما می توانید ipfw را راه اندازی کنید، روش اول با اعمال تغییرات در هسته FreeBSD و اضافه کردن آن به هسته، روش دوم بارگذاری ماژول مربوطه در هسته، روش اول زمان بر بوده و شما نیاز است که مقاله پیکربندی هسته FreeBSD را مطالعه کنید.

برای راه اندازی کردن IPFW در هسته باید خطوط زیر را به فایل GENERIC اضافه کنید و هسته را کامپایل کردن و با هسته جدید سیستم را راه اندازی کنید:

```
options    IPFWIREWALL                # enables IPFW
options    IPFWIREWALL_VERBOSE    # enables logging for rules with log keyword
options    IPFWIREWALL_VERBOSE_LIMIT=5    # limits number of logged packets per-entry
options    IPDIVERT                # enables NAT
options    IPFWIREWALL_DEFAULT_TO_ACCEPT    # sets default policy to pass what is not
                                                explicitly denied
```

خط IPFWIREWALL باعث فعال شدن ipfw می شود، خط IPFWIREWALL_VERBOSE قابلیت log گیری از وضعیت رول های ipfw را در هسته فعال می کند شما می توانید با استفاده از خط IPFWIREWALL_VERBOSE_LIMIT=5 برای گزارشگری محدودیت اعمال کنید تا سیستم شما در اثر فایل های زیاد log دچار از دسترس خارج شدن نشود.

در زمانی که شما به سیستم راه دور متصل هستید و فقط ارتباط ssh به سرور خود دارید و قصد راه اندازی ipfw را دارید بهتر است به این نکته مهم توجه کنید، این فایروال به صورت پیش فرض دارای رولی است که همه ترافیک ها را مسدود می کند مگر در حالتی که شما در زمان کامپایل کردن در هسته خط IPFWIREWALL_DEFAULT_TO_ACCEPT را در فایل هسته خود قرار دهید و بعد عمل کامپایل کردن را انجام دهید.

برای فعال سازی قابلیت NAT و استفاده کردن از این قابلیت باید خط IPDIVERT را در هسته قرار دهید.



روش راه اندازی کردن از طریق ماژول:

شما با استفاده از فرمان `kldload` می توانید ماژول `ipfw` را در هسته سیستم خود بارگذاری کنید و نیازی به راه اندازی سیستم خود ندارید، روش استفاده از این فرمان در شکل زیر نمایش داده شده است:

```

Terminal
File Edit View Terminal Tabs Help
root@FreeBSD:~ # kldload ipfw
root@FreeBSD:~ # kldstat
Id Refs Address          Size   Name
 1   15 0xffffffff80200000 17bc6a8 kernel
 2    1 0xffffffff81a11000 1ee0c8 zfs.ko
 3    1 0xffffffff81c00000 3330   opensolaris.ko
 4    1 0xffffffff81c04000 2ba8   uhid.ko
 5    1 0xffffffff81c07000 114db  ipfw.ko
root@FreeBSD:~ # █

```

نمایش وضعیت `load` شدن ماژولها در هسته

در فرمان اول با استفاده از `kldload ipfw` ماژول مربوطه را بارگذاری کرده اید و با استفاده از فرمان `kldstat` شما وضعیت ماژول های بارگذاری شده در سیستم خود را مشاهده می کنید.

روش راه اندازی از طریق سیستم `rc.conf`

یکی دیگر از روشهای راه اندازی `ipfw` استفاده از فایل `rc.conf` و فرمان های `rc.d` است. برای راه اندازی ابتدایی در زمان `boot` سیستم خط زیر را در فایل `rc.conf` اضافه کنید:

```
firewall_enable="YES"
```

حال باید نوع فایروال خود را مشخص کنید، در فایل `rc.conf` نوع فایروال را با اضافه کردن خط زیر مشخص کنید:

```
firewall_type="open"
```

این بخش دارای حالت های زیر است:

- حالت `Open`: که فایروال هیچ اقدامی انجام نمی دهد و همه ترافیک ها را از خود عبور می دهد، این بخش برای زمانی مناسب است که از راه دور به سیستم متصل می شوید و قصد دارید که در زمان مناسب رولهای خود را اضافه کنید.
- حالت `Client`: در این حالت تمام ترافیک های ارسال شده به سمت سیستمی که `ipfw` بر روی آن راه اندازی شده مسدود می شود.
- حالت `Simple`: در این حالت کل شبکه شما مورد نظارت و حمایت فایروال قرار می گیرد، این حالت برای `NAT` و یا `Gateway` شبکه مفید است.
- حالت `Closed`: تمام ترافیک های ایجاد شده بر روی همه کارتهای شبکه بجز کارت شبکه `loopback` مسدود می شود.



- حالت Workstations سیستمی که ipfw بر روی آن راه اندازی شده است در حالت StateFull مورد حمایت قرار می گیرد.
- نام فایل حاوی رول: شما می توانید رولهای مورد نظر خود را در فایل متنی با قواعد خاص این برنامه بنویسید و آنرا بر روی دیسک خود ذخیره کنید و مسیر آنرا در این بخش ذکر کنید تا در زمان راه اندازی این رولهای شما باید که اعمال شود البته روش دیگری هم وجود دارد.

راه دیگری برای بارگذاری کردن رولهای انتخابی خود استفاده از firewall_script در فایل rc.conf است که شما می توانید در قالب یک فایل اسکریپتی رولهای خود را نوشته و بعد از این بخش نام فایل خود را ذکر کنید به صورت زیر این بخش فعال می شود:

```
firewall_script="/etc/ipfw.rules"
```

در حقیقت فایل etc/ipfw.rules فایل مورد نظر است.

برای فعال کردن حالت گزاشگیری باید خط زیر را در فایل rc.conf اضافه کنید:

```
firewall_logging="YES"
```

برای محدود کردن تعداد لاگ های هر ارتباط باید فرمان زیر را اجرا کنید:

```
#sysctl net.inet.ip.fw.verbose_limit=5
```

یا خط net.inet.ip.fw.verbose_limit=5 را در خط فایل /etc/sysctl.conf اضافه کنید.

برای راه اندازی ipfw از فرمان service به صورت زیر استفاده کنید:

```
# service ipfw start
```

برای کنترل کردن ipfw فرمانی در سیستم عامل FreeBSD به همین نام وجود دارد که در بخش بعدی با آن بیشتر آشنا می شوید، برای تست کردن از صحت راه اندازی شدن ipfw شما می توانید از این فرمان با سوئیچ list استفاده کنید تا تمام رولها برای شما نمایش داده شود، خروجی این فرمان در شکل زیر برای شما نمایش داده شده است :

```
Terminal
File Edit View Terminal Tabs Help
root@FreeBSD:~ # ipfw list
65535 deny ip from any to any
root@FreeBSD:~ #
```

نمایش وضعیت رولها در IPFW



قواعد رول نویسی در IPFW

زمانی که یک بسته به فایروال ipfw بسته با هر رولی که توسط مدیر سیستم نوشته شده باشد از بالا به پایین چک می شود و اولین رولی که با مشخصات بسته دریافتی یکی باشد آن رول بر روی بسته اعمال می شود مثلاً اگر اجازه رد شدن از فایروال را نداشته باشد بسته دریافتی رد می شود به این قابلیت خاصیت First match Wins می گویند. بسته هایی که مشخصات آنها در هیچ یک از رولها نباشد به صورت خودکار با رول شماره 65535 که آخرین رول و رول پیش فرض است تعیین تکلیف می شوند، به صورت پیش فرض این رول به کلی همه بسته ها را deny می کند مگر شما این رول پیش فرض را تغییر داده باشید که در مقاله قبلی روش تغییر دادن آنرا توضیح داده شده.

نوشتن رول در ipfw دارای قواعدی است که در ادامه با آن آشنا میشوید، بعضی از این بخشها اجباری بوده و حتما باید در رول باشند و در بعضی از بخشها هم اجباری وجود ندارد و فقط برای شما کار خاصی را که توضیح داده می شود انجام می دهد.

شما می توانید رول های مورد نظر خود را در قالب یک فایل اسکریپت بنویسید تا هر زمانی که لازم بود آنرا اعمال، ویرایش کنید و در زمان راه اندازی مجدد سیستم تغییرات شما پاک نشود، رولهای شما در ipfw در لحظه اعمال می شود و تا زمان راه اندازی سیستم معتبر خواهد بود پس به این نکته توجه کنید، برای نوشتن این فایل دو قاعده کلی وجود دارد که قاعده اول در همه فایلهای پیکربندی وجود دارد و آن استفاده از # در ابتدای خطوطی است که توضیحات در آن قرار دارد و هیچ تغییری برای شما اعمال نمی کند، قاعده جامع دوم هم این است که خطوطی که خالی هستند هم تغییرات را به همراه نخواهد داشت و بی تاثیر هستند.

در زیر یک نمونه کلی و جامع از یک رول در ipfw را مشاهده می کنید:

```
CMD RULE_NUMBER set SET_NUMBER ACTION log LOG_AMOUNT PROTO from SRC SRC_PORT to DST
DST DST_PORT OPTIONS
```

بخش cmd

در حقیقت یک بخش اجباری است و باید در ابتدای هر رول وجود داشته باشد و با ipfw add شروع می شود.

بخش RULE_NUMBER

هر رول که نوشته می شود باید دارای یک عدد باشد که بین 1 تا 65534 است. این عدد در حقیقت ترتیب عددی رول ها بوده و باید به ترتیب آن دقت کنید به خاطر قاعده First match Wins چند رول هم می توانند شامل یک عدد باشد در این حالت آن رولی که زودتر نوشته شده باشد رول اول است.

بخش set SET_NUMBER

این بخش اجباری نبوده و شما می توانید با استفاده از آن برای یک سری از رول های خود یک ست با عددی بین 0 تا 31 در نظر بگیرید. شما به راحتی می توانید این ست ها را فعال و غیرفعال کنید. این کار سرعت تغییرات شما را افزایش می دهد، اگر شما در بخش SET_NUMBER عددی وارد کنید به صورت خودکار عدد 0 اعمال می شود.

بخش ACTION



در این بخش عملی که بر روی بسته دریافتی اعمال می شود را باید تعیین کنید. این بخش شامل دو کلید واژه اصل است allow که با accept و pass یکی است و به معنی این است که بسته اجازه رد شدن دارد Deny. که با prop مترادف است و هر بسته ای که با آن مچ باشد اجازه عبور نخواهد داشت.

بخش LOG_AMOUNT

در این بخش قسمت log برای هر رول را فعال می کنید که اطلاعات به syslog ارسال شده و با facility name خاص SECURITY ذخیره می شود. گزارشگیری زمانی رخ می دهد که شما بخش تعداد log را مشخص کنید، اگر این بخش را مشخص نکرده باشید به صورت خودکار از net.inet.ip.fw.verbose_limit این مقدار را می گیرد. برای دوباره تنظیم کردن مجدد لاگ از فرمان ipfw reselog استفاده کنید.

بخش PROTO

این بخش جزو بخشهای اجباری بوده و نوع پروتکلی را که قصد اعمال محدودیت یا باز کردن آنرا دارید را مشخص می کنید برای دریافت لیست کاملی از پروتکلها به فایل /etc/protocols مراجعه کنید.

بخش from SRC

کلمه from یک کلمه ای است که باید در این بخش قبل از مشخص کردن مبداء مورد نظر ذکر کنید، در بخش بعدی از این متن شما باید مبداء را مشخص کنید، چند کلمه رایج در این بخش وجود دارد به نام any یعنی هر آدرسی یا me یعنی همین سیستم شما که بر روی آن فایروال را نصب کرده اید برای تعریف کردن آدرس ورژن 6 از me6 استفاده کنید. در این بخش شما می توانید آدرس ip را هم به صورت مستقیم در این بخش وارد کنید، این حالت زمانی اتفاق می افتد که سیستم شما Gateway در شبکه است و می خواهید ارتباطات کامپیوترهای شبکه خود را محدود کنید، روش نوشتن آدرس به دو حالت در زمان نوشتن subnet mask حالت اول CIDR است مثل 24/192.168.1.100 و حالت دوم هم نوشتن عددی آن به صورت 192.168.1.100 255.255.255.0.

بخش SRC_PORT

در این بخش شما به صورت اختیاری می توانید شماره پورت سرویس مبداء بسته را مشخص کنید برای دریافت تمام پورت های مربوطه به هر سرویسی به فایل /etc/services مراجعه کنید.

بخش to DST

این بخش هم برای مصدق بسته تعیین کننده است که توضیحاتی مثل بخش from SRC است و هم می توانید از any و me استفاده کنید و یا آدرس ip را به روش های ذکر شده مشخص کنید.



بخش DST_PORT

در این بخش شما به صورت اختیاری می توانید شماره پورت سرویس مقصد بسته را مشخص کنید برای دریافت تمام پورت های مربوطه به هر سرویسی به فایل `/etc/services` مراجعه کنید.

بخش OPTIONS

در این بخش شما می توانید هر گزینه های `keep-state` و ... در این بخش قرار دهید ولی پرکاربردترین این بخشها دو گزینه `in` و `out` است که مسیر بسته ها را مشخص می کند و شما با استفاده از `via` هم می توانید کارت شبکه را مشخص کنید.

نمونه ای از ولها:

```
Ipfw add 10 allow all from any to any via lo0
```

در رول بالا شما به همه ترافیک هایی که از کارت شبکه `loopback` رد می شود امکان تردد می دهد.

```
Ipfw 110 allow tcp from any to any 21 in
Ipfw 120 allow tcp from any to any 21 out
```

در دو رول بالا شما کلا استفاده از پورت 21 که سرویس `ftp` است را باز می کنید.

```
Ipfw add 00310 deny icmp from any to any in via
```

در رول بالا همه ترافیک های مربوطه به سرویس `ping` یا همان پروتکل `icmp` را مسدود می کنید.

فرمان ipfw

فایروال `ipfw` دارای یک رابط فرمان است به نام `ipfw` که علاوه بر استفاده در اضافه کردن رول برای بخش های مدیریتی هم مورد استفاده قرار می گیرد و سوئیچهای مفیدی دارد که وضعیت رول های سرور شما را نمایش می دهد و یا حتی یک رول خاص و یا تمام رولها را پاک می کند، در ادامه با روش استفاده از این فرمان در قابل دریافت گزارش بیشتر آشنا می شوید.

سوئیچ list

اگر شما بعد از فرمان `ipfw` از این سوئیچ استفاده کنید همه رولهای راه اندازی شده شما را نمایش می دهد.

سوئیچ -t list

این سوئیچ علاوه بر خروجی فرمان بالا `time stamp` هر رول را نیز برای شما نمایش میدهد.



سوئیچ list -a

این سوئیچ عددی را در کنار هر رول نمایش می دهد که شامل شمارش بسته هایی است که با این رول هماهنگ بوده ، به اصطلاح شمارنده ای است از هر رول.

سوئیچ zero

با استفاده از این سوئیچ تمام شمارنده های همه رولها صفر می شود.

پاک کردن یک رول خواص:

برای پاک کردن یک رول خواص کفایست که از فرمان ipfw و سوئیچ delete و شماره رول استفاده کنید(در این زمینه در سایت و مقاله اصلی چند فایل gif تصویری وجود دارد)

IPFW در tables

یکی از امکانات IPFW این است که می توانید با استفاده از جدول بروی هم زمان چند آدرس ip یک عمل خاص را انجام دهید، در قدم اول شما باید یک جدول را ایجاد کنید برای این کار باید از فرمان ipfw و table استفاده کنید، بعد از آن باید نام جدول را ذکر کنید و با کلمه Add و آدرس ip جدول مورد نظر را ایجاد کنید، به عنوان مثال جدول شماره 1 را با اضافه کردن آدرس 192.168.1.1 بصورت زیر اضافه کنید:

```
#ipfw table 1 add 192.168.1.1
```

برای مشاهده کردن لیست همه جداول از فرمان زیر استفاده کنید:

```
#ipfw table all list
```

برای مشاهده کردن لیست های موجود در یک جدول فقط کفایست در فرمان بالا بجای all از نام جدول اضافه کنید.

برای پاک کردن یک جدول خاص فرمان زیر را اجرا کنید:

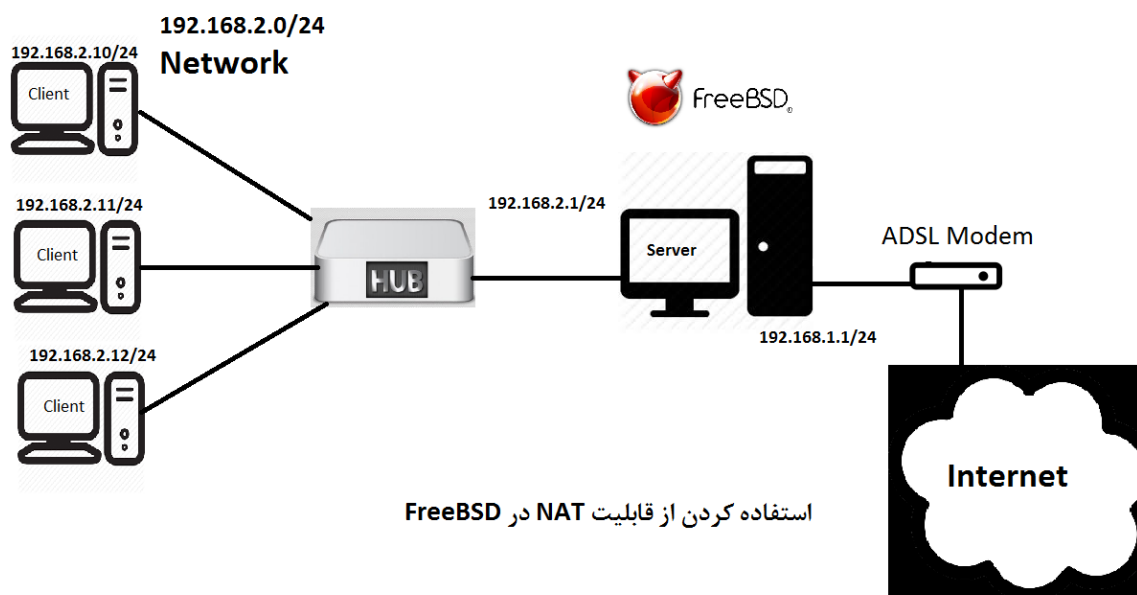
```
#ipfw table 1 flush
```


Nat کردن ترافیک با IPFW در FreeBSD

یکی از راه حل های به اشتراک گذاشتن اینترنت به شبکه های محل استفاده کردن از سرویس NAT است، البته این سرویس کارکردهای دیگری هم دارد که در بخشهای بعدی در مورد آنها بحث می شود. این سرویس برای راه اندازی شدن به فایروال ipfw وابسته است و شما باید ابتدا این فایروال را هم فعال کنید. بر روی سیستم که شما قصد دارید این سرویس را راه اندازی کنید باید دو کارت شبکه داشته باشید، یک کارت شبکه که به شبکه محلی شما که قصد دارید اینترنت را در اختیار آنها قرار دهید متصل باشد و کارت شبکه دوم هم باید به اینترنت متصل باشد. طراحی آدرس IP شبکه ها انتخابی است.

طراحی شبکه برای nat

برای پیاده سازی سناریوی شبکه nat شما باید قبل از شروع آدرس IP شبکه خود را مشخص کنید، شکل زیر را مشاهده کنید:



طراحی شماتیک از شبکه ای با قابلیت Nat

شبکه بالا دارای دو بخش است بخش اول شبکه محلی است که رنج آدرس های آن 192.168.2.0 است و سه عدد PC در آن قرار دارد، تمام این سیستم ها باید آدرس 192.168.2.1 را به عنوان Default Gateway تنظیم کنند. سرور FreeBSD دارای دو کارت شبکه است که یکی از آدرس های آن 192.168.2.1 است و آدرس دوم 192.168.1.1 است که به مودم ADSL وصل است که خود شبکه اینترنت متصل است. شبکه بالا یک طراحی ساده از یک شبکه است. از هاب برای متصل کردن سیستم ها در این شبکه استفاده شده است. در ادامه با راه اندازی کردن این مدل آشنا می شوید.



فعال کردن IPDIVERT و IPFW

شما به دو روش می توانید قابلیت های ذکر شده را در FreeBSD فعال کنید روش اول خطوط زیر را در هسته فعال کنید و هسته دوباره بارگذاری کنید، برای انجام دادن این روش مقاله پیکربندی هسته در FreeBSD را مطالعه کنید:

```
options IPFIREWALL
options IPDIVERT
```

روش بعدی بارگذاری کردن ماژول `ipdivert` در هسته با استفاده کردن از `kldload` به این نکته توجه کنید که ماژول `ipfw` هم به صورت خودکار در هسته بارگذاری می شود و رول پیش فرض آن بستن همه ترافیک هاست. در شکل زیر این عمل نمایش داده شده است:

```
Terminal
File Edit View Terminal Tabs Help
root@FreeBSD:~ # kldstat
Id Refs Address          Size      Name
 1   13 0xffffffff80200000 17bc6a8  kernel
 2    1 0xffffffff81a11000 1ee0c8   zfs.ko
 3    1 0xffffffff81c00000 3330     opensolaris.ko
 4    1 0xffffffff81c04000 2ba8     uhid.ko
root@FreeBSD:~ # kldload ipdivert
root@FreeBSD:~ # kldstat
Id Refs Address          Size      Name
 1   17 0xffffffff80200000 17bc6a8  kernel
 2    1 0xffffffff81a11000 1ee0c8   zfs.ko
 3    1 0xffffffff81c00000 3330     opensolaris.ko
 4    1 0xffffffff81c04000 2ba8     uhid.ko
 6    1 0xffffffff81c07000 4cb6     ipdivert.ko
 7    1 0xffffffff81c0c000 114db    ipfw.ko
root@FreeBSD:~ # ipfw list
65535 deny ip from any to any
root@FreeBSD:~ #
```

خلاصه ای از راه اندازی IPFW در FreeBSD

در مرحله بعدی شما باید قابلیت `Gate way` را هم فعال کنید این کار به دو صورت انجام می شود در روش اول با استفاده از `rc.conf` که باید در این روش خط زیر را در این فایل قرار دهید:

```
# gateway_enable=YES
```

حال در زمان راه اندازی سیستم شما قابلیت `Gate Way` برای شما فعال می شود در حالت دوم هم شما می توانید با استفاده از فرمان `sysctl` این قابلیت را فعال کنید:

```
# sysctl net.inet.ip.forwarding=1
```

در مرحله بعدی قابلیت `natd` را فعال کنید. در این بخش هم می توانید از سیستم `rc.conf` استفاده کنید و یا فرمان `natd` را به صورت مستقیم اجرا کنید، برای فعال سازی `rc.conf` خطوط زیر را در این فایل اضافه کنید:



```

natd_enable="YES"      # enables NAT
natd_interface="em0"  # specify interface name of NIC attached to Internet

```

در خط اول سرویس `natd` را فعال کرده اید و در خط بعدی باید نام کارت شبکه ای را که به شبکه دیگر که اینترنت دارد را مشخص کنید که در این بخش `em0` است و در هر سیستم می تواند متفاوت باشد. شما می توانید با استفاده از `natd_flags` سوئیچ های لازم برای این سرویس را مشخص کنید. حال با استفاده از فرمان زیر سرویس `natd` را فعال کنید:

```
#/etc/rc.d/natd start
```

در روش استفاده از خط فرمان شما از فرمان `natd` در شل می توانید با استفاده از سوئیچ های مختلف این سرویس را راه اندازی کنید، برای مثال برای فعال کردن قابلیت ارایه گزارش از `log` یا `-log` استفاده کنید و بعد از فعال شدن قابلیت `log` فایلی به نام `/var/log/alias.log` ایجاد می شود و `log` ها در آن ذخیره می شود، برای مشخص کردن کارت شبکه ای که با اینترنت متصل است از سوئیچ `-interface` یا `-n` استفاده کنید و در مقابل آن نام کارت شبکه را واردی کنید در زیر این فرمان نمایش داده شده است:

```
#natd -log -interface em0
```

در بخش آخر هم باید در فایروال `ipfw` دو فرمان زیر را وارد کنید:

```

#ipfw add divert natd all from any to any via em0
#ipfw add 500 allow all from any to any

```

تنظیمات بخش Client

در بخش کلاینتی اگر سیستم عامل شما `FreeBSD` باشد بعد از تنظیم کردن آدرس `ip` باید با استفاده از فرمان `route` به صورت زیر آدرس سرور `FreeBSD` خود را به صورت `default gateway` معرفی کنید:

```
#route add default 192.168.2.1
```

سوئیچ های فرمان natd

برنامه `natd` برای `nat` کردن ترافیک های شبکه از سوکت `divert` استفاده می کند و به صورت `daemon` در پس زمینه سیستم راه اندازی می شود و عمل `nat` کردن را انجام می دهد، در این بین این فرمان دارای سوئیچ ها و قابلیت های زیادی است که برخی از آنها را برای شما ذکر می کنم.

بخش log

برای فعال کردن قابلیت `log` در `natd` باید از سوئیچ های `log` یا سوئیچ `-log` استفاده کنید، گزارشات در فایلی به نام `alias.log` در زیر شاخه `/var/log` ذخیره می شود.



بخش مسدود کردن ترافیک های ورودی غیر ضروری:

برای اینکه سرویس natd به ترافیک هایی را که در جدول ترجمه خود وجود ندارند اجازه ورود ندهد از سویچ - deny_incoming و یا -d استفاده کنید. این قابلیت را همیشه فعال کنید

بخش use_sockets

این بخش در حالتی که شما از سرویس هایی که سوکتهای اضافی برای دریافت اطلاعات خود نیاز دارند ایجاد می کند مثل فرمان ftp. برای به درستی کار کردن این نوع از برنامه ها از سویچ use_sockets - یا -s استفاده کنید.

سویچ verbose

این سویچ برنامه natd را راه اندازی نمی کند بلکه برنامه را در حالتی قرار می دهد که بتوانید اطلاعات بیشتری از وضعیت سرویس خود را در کنسول شل مشاهده کنید برای دیباگ کردن برنامه ها بسیار مفید است و با استفاده از سویچ verbose - یا -v فعال می شود.



قابلیت address redirection و port redirection در Natd

یکی از قابلیت‌های که کمتر از در nat مورد بررسی قرار گرفته redirection در port و آدرس است. کاربرانی که در شبکه محلی در پشت سرویس nat قرار دارند به هیچ عنوان از طریق شبکه بیرونی قابل دسترس نبود مگر در سرویس natd قابلیت redirection فعال و پیکربندی شده باشد. در این بخش با روش‌های راه اندازی این قابلیت در natd بیشتر آشنا می‌شوید.

فایل پیکربندی natd قبل از شروع با سوئیچی از برنامه natd آشنا می‌شوید به نام config، این سوئیچ که به اختصار هم می‌توانید آنرا با -f مشخص کنید باید با نام فایلی همراه باشد که فایل پیکربندی است برای natd. این فایل در قابل یک فایل متنی است و هر خط باعث فعال شدن یک قابلیت می‌شود و خطوط خالی تاثیری در رفتار برنامه ندارد و خطی که با # شروع شود حکم توضیحات را دارد، در ادامه با روش‌های ایجاد کردن این فایل بیشتر آشنا می‌شوید، برای مثال برای فعال کردن قابلیت log در natd خطی به صورت زیر در فایل مورد نظر بنویسید log yes: این بخش در حقیقت با سوئیچ log برابر است. یکی از قابلیت‌های این فایل دارا بودن بخش‌های متفاوت است که هر بخش به صورت جداگانه خورد بررسی توسط برنامه natd قرار می‌گیرد و می‌توانید یک یا چند پردازش با قابلیت‌های مختلفی ایجاد کند به این بخشها به اصطلاح instance گفته شده که اولین و پیش فرض آن default نامیده می‌شود، برای بخش بندی کردن فایل پیکربندی با instance‌های مختلف باید از کلمه instance و بعد نام مورد نظر خود استفاده کنید به صورت زیر instance_name instance: شما به راحتی می‌توانید این فایل را ایجاد کنید و صورت زیر فایل را به برنامه natd معرفی کنید #natd -config /etc/natd.conf :
or #natd -f /etc/natd.conf نامنامه فایل natd.conf یک نام انتخابی است.

قابلیت port redirection: شما با استفاده از این قابلیت می‌توانید درخواست‌هایی که از به سمت پورت خاصی از سمت شبکه بیرونی شما می‌آید را به سمت کلاینتی که در شبکه محلی شماست منتقل کند دستور کلی این بخش به صورت زیر است:

```
-redirect_port proto targetIP:targetPOR
```

در زیر مثالی از این قابلیت را مشاهده می‌کنید

```
-redirect_port tcp 192.168.0.3:80 80
```

با استفاده از این فرمان ترافیک‌هایی که به سمت پورت 80 از سمت سیستم‌های خارج از LAN ارسال می‌شوید به کمک برنامه natd ترجمه شده و به سمت کلاینت با آدرس 192.168.0.3 ارسال می‌شود. شما به 3 روش می‌توانید redirection را فعال کنید،

روش اول با استفاده از فرمان natd

```
#natd -redirect_port tcp 192.168.0.3:80 80
```

روش دوم در فایل rc.conf

که در زیر شاخه /etc قرار دارد با استفاده کردن از گزینه natd_flags به صورت زیر این قابلیت را فعال کنید و دوباره سرویس natd را restart کنید :



```
natd_flags="-redirect_port tcp 192.168.0.3:80 80
```

حال برای راه اندازی natd از فرمان زیر استفاده کنید

```
#/etc/rc.d/natd restart
```

و یا خط `redirect_port tcp 192.168.0.3:80 80` در فایل `config` در `instance` پیش فرض قرار دهید و با سوئیچ `f`-فرمان `natd` را اجرا کنید.

قابلیت address redirection

این قابلیت زمانی مورد استفاده قرار می‌گیرد که بیش از یک آدرس ip در بخش nat سرور وجود داشته باشد و برای هر کلاینت در شبکه محلی بتوان یک آدرس ip بیرونی اختصاص داد. با استفاده از این قابلیت سرویس nat می‌تواند ترافیک‌ها را در بین کلاینتها و درخواست‌های بیرونی مدیریت کند و با اصطلاح این نوع از nat به static NAT معروف است. حالت کلی این قابلیت به صورت زیر است

```
-redirect_address localIP publicIP
```

در بخش `localIP` شما آدرس‌های کلاینتهای شبکه محلی خود را نوشته و در بخش `publicIP` آدرس شبکه خارجی نوشته تا این دو آدرس متناظر هم قرار گیرند. در زیر مثالی از این نوع nat را مشاهده می‌کنید:

```
-redirect_address 192.168.0.2 217.218.100.10 -redirect_address 192.168.0.3 217.218.100.11
```

شما مثل بخش قبلی می‌توانید از سه روش فعال سازی این قابلیت استفاده کنید، روش فرمان `natd`، فایل `rc.conf` و فایل پیکربندی.

مخفی کردن NAT از فرمان traceroute

در هسته FreeBSD قابلیت وجود دارده نام IPSTEALTH که با استفاده از آن شما می‌توانید سرور NAT شبکه خود را دیده فرمان `traceroute` مخفی کنید این قابلیت فقط در هسته فعال می‌شود و برای راه اندازی آن نیاز به کامپایل کردن هسته دارید. این قابلیت به این صورت کار می‌کند که TTL بسته‌های دریافتی را بدون تغییر رد کرده، برای فعال کردن خط زیر را در فایل پیکربندی هسته اضافه کنید

```
options IPSTEALTH # Enable stealth forwarding
```



برای فعال کردن این قابلیت از فرمان sysctl به صورت زیر استفاده کنید

```
# sysctl net.inet.ip.stealth=1
```

در بسیاری از موارد شاید شما دوستان با خطای زیر در انجام دادن فرمان های زیر مواجه شوید:

```
natd: instance default: aliasing address not given
```

این خطا در زمانی رخ می دهد که سرویس natd شما در حالت سرویس دهی قرار دارد و برا اضافه کردن قابلیت جدید امدگی ندارد، برای حل این مشکل بهتر است که شما سرویس خود را غیرفعال کنید و تمام تنظیمات خود را در فایل پیکربندی قرار دهید و از طریق سوئیچ f- سرویس natd را دوباره راه اندازی کنید یا در یک فرمان تمام قابلیت های مورد نظر خود را به سرویس اضافه کنید.



راه اندازی Zebra در FreeBSD

در FreeBSD برنامه ای وجود دارد به نام Zebra که با استفاده از آن شما می توانید سیستم خود را به یک router قوی با پروتکل‌های مسیریابی زیر تغییر کاربری دهید، پروتکل هایی که zebra از آن حمایت می کند شامل لیست زیر است:

- پروتکل BGP
- پروتکل OSPF
- پروتکل RIP

علاوه بر این پروتکل های مسیر یابی از آدرس IP ورژن 4 و 6 پشتیبانی کرده و شما می توانید از SNMP برای دریافت گزارشات استفاده کنید. این برنامه از پورت کنسول VTY استفاده می کند، در ادامه با روش نصب و پیکربندی و استفاده از این سرور قوی آشنا می شوید.

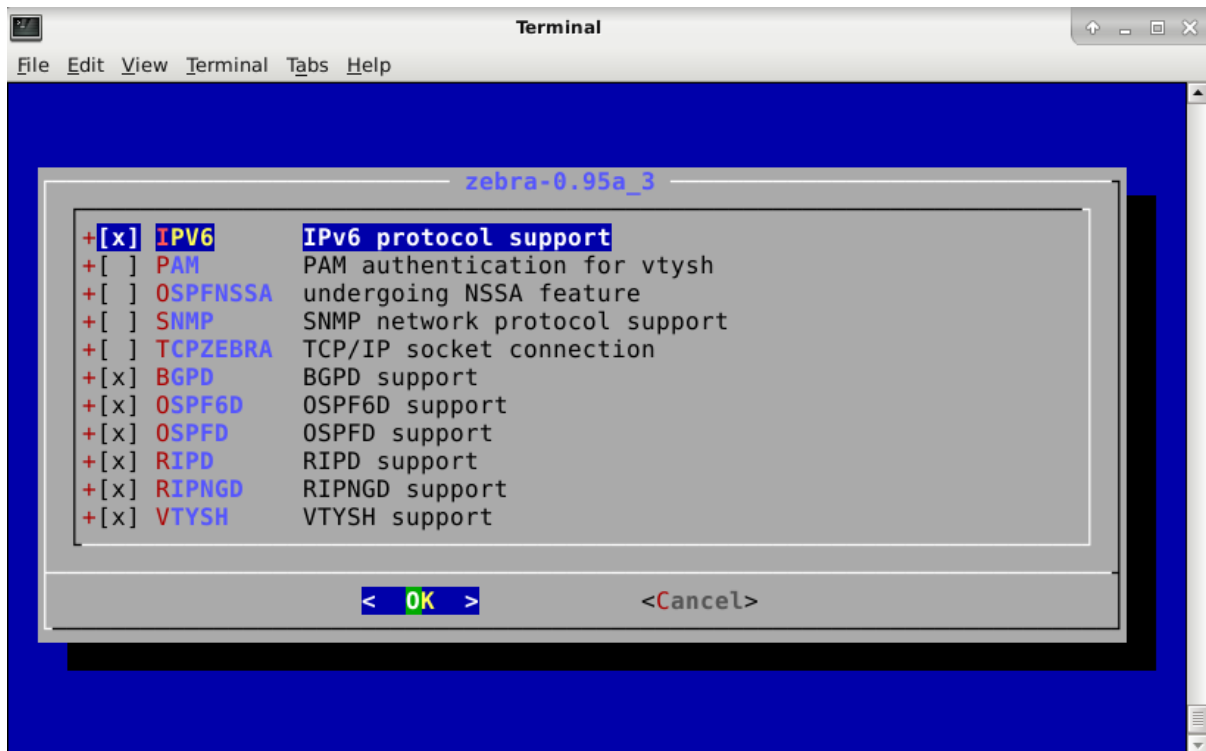
نصب Zebra در FreeBSD:

شما هم می توانید از طریق سیستم بسته های باینری و هم از طریق سیستم پورت برنامه zebra را نصب کنید، بهترین حالت نصب استفاده از سیستم پورت در FreeBSD است به این علت که شما می توانید گزینه های انتخابی و خاصیت های که از این گزینه ها به آن نیاز دارید را در برنامه فعال کنید در قدم اول وارد شاخه /usr/ports/net/zebra شوید و فرمان make را اجرا کنید این فرمان در شکل زیر نمایش داده شده است:

```
Terminal
File Edit View Terminal Tabs Help
root@FreeBSD:~ # cd /usr/ports/net/zebra
root@FreeBSD:/usr/ports/net/zebra # make
```

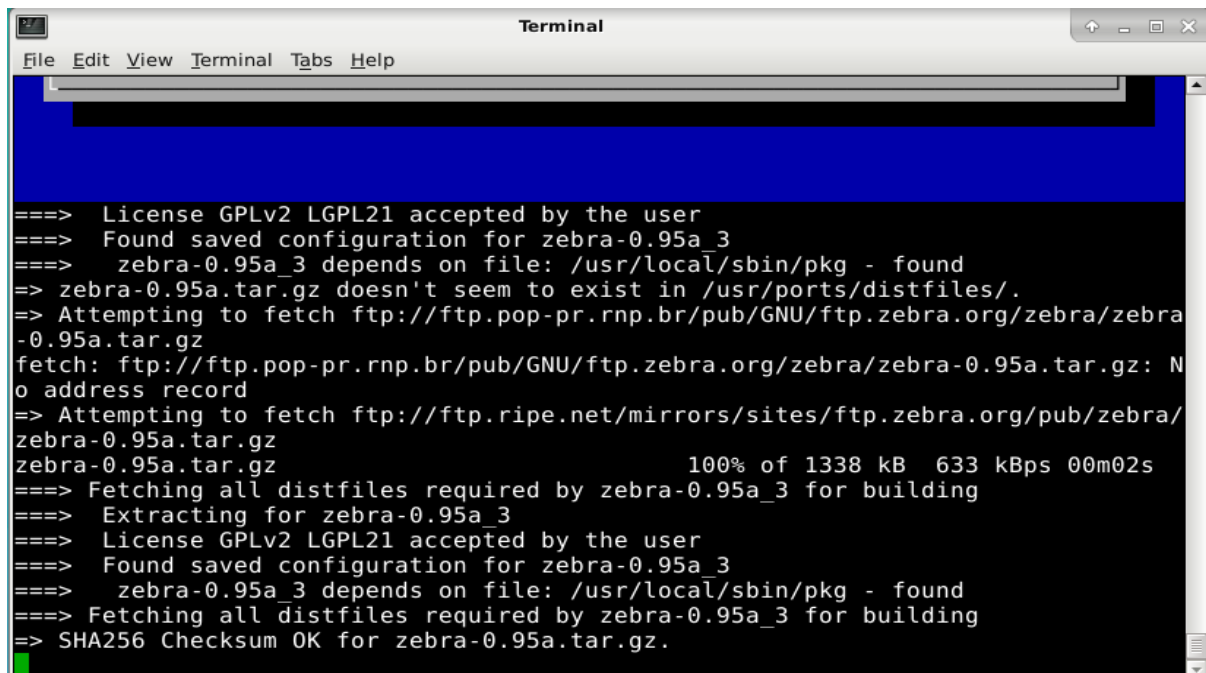
شروع به نصب zebra

در مرحله بعدی شما با صفحه ای مواجه می شوید که در آن گزینه های مورد نیاز را می توانید فعال و یا غیر فعال کنید، برای مثال شما می توانید از SNMP استفاده کنید به شرطی که در برنامه آن را اضافه کرده باشید، برنامه پیش فرض این قابلیت را ندارد:



بخش تنظیمات zebra در زمان نصب

بعد از انتخاب کردن گزینه های مورد نیاز سیستم پورت به صورت خودکار اقدام به دانلود کردن برنامه و پیش نیاز های نصبی zebra کرده و در خروجی مراحل را برای شما نمایش می دهد، بخش ابتدایی از این خروجی برای شما نمایش داده شده است:



نمایش مراحل نصب zebra



بعد از اتمام فرمان make اگر این برنامه با خطایی مواجه نشود و به درستی اجرا شود شما به خط فرمان باز می گردید و خروجی به صورت شکل زیر را مشاهده می کنید:

```
Terminal
File Edit View Terminal Tabs Help
install-info --info-dir=/usr/ports/net/zebra/work/stage/usr/local/info/ /usr/ports/net/zebra/work/stage/usr/local/info//zebra.info
/bin/sh ../mkinstalldirs /usr/ports/net/zebra/work/stage/usr/local/man/man1
install -o root -g wheel -m 0644 ./vttysh.1 /usr/ports/net/zebra/work/stage/usr/local/man/man1/vttysh.1
/bin/sh ../mkinstalldirs /usr/ports/net/zebra/work/stage/usr/local/man/man8
install -o root -g wheel -m 0644 ./bgpd.8 /usr/ports/net/zebra/work/stage/usr/local/man/man8/bgpd.8
install -o root -g wheel -m 0644 ./ospf6d.8 /usr/ports/net/zebra/work/stage/usr/local/man/man8/ospf6d.8
install -o root -g wheel -m 0644 ./ospfd.8 /usr/ports/net/zebra/work/stage/usr/local/man/man8/ospfd.8
install -o root -g wheel -m 0644 ./ripd.8 /usr/ports/net/zebra/work/stage/usr/local/man/man8/ripd.8
install -o root -g wheel -m 0644 ./ripngd.8 /usr/ports/net/zebra/work/stage/usr/local/man/man8/ripngd.8
install -o root -g wheel -m 0644 ./zebra.8 /usr/ports/net/zebra/work/stage/usr/local/man/man8/zebra.8
====> installing zebra startup file...
install -o root -g wheel -m 555 /usr/ports/net/zebra/work/zebractl /usr/ports/net/zebra/work/stage/usr/local/sbin/zebractl
done.
====> Compressing man pages (compress-man)
root@FreeBSD:/usr/ports/net/zebra #
```

نمایش مراحل نصب zebra

حال در مرحله بعدی فرمان make install را اجرا کنید:

```
Terminal
File Edit View Terminal Tabs Help
router_enable="YES"
router="/usr/local/sbin/zebractl"
router_flags="start"
====> SECURITY REPORT:
This port has installed the following files which may act as network
servers and may therefore pose a remote security risk to the system.
/usr/local/sbin/zebra
/usr/local/sbin/bgpd
/usr/local/sbin/ripngd
/usr/local/sbin/ospf6d
/usr/local/sbin/ospfd
/usr/local/bin/vttysh
/usr/local/sbin/ripd

If there are vulnerabilities in these programs there may be a security
risk to the system. FreeBSD makes no guarantee about the security of
ports included in the Ports Collection. Please type 'make deinstall'
to deinstall the port if this is a concern.

For more information, and contact details about the security
status of this software, see the following webpage:
http://www.zebra.org/
root@FreeBSD:/usr/ports/net/zebra #
```

نمایش مراحل نصب zebra



در این بخش اطلاعات جالبی برای شما نمایش داده شده است، برای مثال برنامه هایی که نصب شده است را برای شما نمایش می دهد.

شاخه فایل های سرور Zebra

بعد از اتمام نصب در زیر شاخه /usr/local/etc/zebra فایل هایی قرار دارد که در شکل زیر لیست آنها را مشاهده می کنید:

```
Terminal
File Edit View Terminal Tabs Help
root@FreeBSD:~ # cd /usr/local/etc/zebra/
root@FreeBSD:/usr/local/etc/zebra # ls
bgpd.conf.sample      ospfd.conf.sample      vtysh.conf.sample
bgpd.conf.sample2     ripd.conf.sample       zebra.conf.sample
ospf6d.conf.sample    ripngd.conf.sample
root@FreeBSD:/usr/local/etc/zebra #
```

شاخه فایل های پیکربندی zebra

در قدم اول شما باید فایل اصلی پیکربندی به نام zebra.conf را ایجاد کنید، نام این فایل به صورت پیش فرض zebra.conf.sample است.

```
Terminal
File Edit View Terminal Tabs Help
root@FreeBSD:/usr/local/etc/zebra # mv zebra.conf.sample zebra.conf
root@FreeBSD:/usr/local/etc/zebra #
```

Backup گرفتن از فایل پیکربندی پیش فرض

فایل پیش فرض این برنامه شامل خطوط زیر است:

```
Terminal
File Edit View Terminal Tabs Help
! zebra sample configuration file
!
! $Id: zebra.conf.sample,v 1.14 1999/02/19 17:26:38 developer Exp $
!
hostname Router
password zebra
enable password zebra
!
! Interface's description.
!
!interface lo
! description test of desc.
!
!interface sit0
! multicast
!
! Static default route sample.
!
!ip route 0.0.0.0/0 203.181.89.241
!
!log file zebra.log
root@FreeBSD:/usr/local/etc/zebra #
```

نمایش محتوای فایل پیش فرض برنامه zebra



فایل پیکربندی این برنامه به صورت یک پیکربندی روتر است، برای مثال Hostname برنامه Router در نظر گرفته شده است، دو password مهم این برنامه zebra است که شما می توانید از طریق این فایل آنرا تغییر دهید.

راه اندازی Zebra:

برنامه Zebra از طریق سیستم rc.conf قابل راه اندازیت، برا راه اندازی آن باید در فایل /etc/rc.conf خطوط زیر را وارد کنید:

```
defaultroute="NO"
router_enable="YES"
router="/usr/local/sbin/zebractl"
router_flags="Start"
```

برنامه zebra با دو فرمان راه اندازی می شود، فرمان اول zebra و فرمان دوم zebractl. برای راه اندازی کردن با استفاده از فرمان zebra به سوئیچ های این برنامه دقت کنید، برای راه اندازی کردن در حالت daemon از سوئیچ d و برای اینکه فایل پیکربندی غیر از فایل پیش فرض را قرار دهید از سوئیچ f استفاده کنید.

خروجی فرمان zebractl را هم در شکل زیر مشاهده می کنید:

```
Terminal
File Edit View Terminal Tabs Help
root@FreeBSD:~ # zebractl
/usr/local/sbin/zebractl: error: one argument needed
/usr/local/sbin/zebractl: usage: /usr/local/sbin/zebractl [ start | stop | restart ]
root@FreeBSD:~ #
```

خروجی فرمان zebractl

این فرمان همان کار فرمانهای rc.d را انجام می دهد، در ادامه فرمان zebra را بدون سوئیچ اجرا کنید تا برنامه راه اندازی شود، خروجی این فرمان به صورت زیر است که با چند خطای هشدار برای فعال کردن سیستم log مواجه می شود:

```
Terminal
File Edit View Terminal Tabs Help
root@FreeBSD:~ # zebra
2017/01/24 12:01:35 ZEBRA: ifam_read() doesn't read all socket data
2017/01/24 12:01:35 ZEBRA: ifam_read() doesn't read all socket data
2017/01/24 12:01:35 ZEBRA: ifam_read() doesn't read all socket data
2017/01/24 12:01:35 ZEBRA: ifam_read() doesn't read all socket data
2017/01/24 12:01:35 ZEBRA: ifam_read() doesn't read all socket data
```

خطاهای هشدار در راه اندازی zebra



همانطوری که مشاهده می کنید برنامه راه اندازی شده و خط فرمان برای شما باز نشده است برای اینکه خط فرمان برای شما باز شود از سوئیچ `d` استفاده کنید و در بخش بعدی با استفاده از فرمان `sockstat` وضعیت پورتهای را که برنامه برای انجام اقدامات مدیریتی بر روی آن به گوش کردن از شبکه می پردازد را مشاهده کنید:

```
Terminal
File Edit View Terminal Tabs Help
root@FreeBSD:~ # zebra -d
2017/01/24 12:04:03 ZEBRA: ifam_read() doesn't read all socket data
2017/01/24 12:04:03 ZEBRA: ifam_read() doesn't read all socket data
2017/01/24 12:04:03 ZEBRA: ifam_read() doesn't read all socket data
2017/01/24 12:04:03 ZEBRA: ifam_read() doesn't read all socket data
2017/01/24 12:04:03 ZEBRA: ifam_read() doesn't read all socket data
root@FreeBSD:~ # sockstat -l4
USER      COMMAND  PID  FD  PROTO  LOCAL ADDRESS  FOREIGN ADDRESS
root      zebra    8209  8   tcp4   *:2601         *: *
root      ftpd     1442  6   tcp4   *:21           *: *
root      sendmail 652   3   tcp4   127.0.0.1:25   *: *
root      sshd     649   4   tcp4   *:22           *: *
root      syslogd  513   7   udp4   *:514          *: *
```

تست کردن راه اندازی شدن `zebra`

برای برقرار ارتباط با `Zebra` کافیست که به پورت شماره `2601` `telnet` کنید و بعد از وارد کردن رمز عبوری که در فایل `Zebra.conf` مشخص کرده اید وارد کنسول مدیریت این برنامه بشوید:

```
Terminal
File Edit View Terminal Tabs Help
root@FreeBSD:~ # telnet 127.0.0.1 2601
Trying 127.0.0.1...
Connected to localhost.
Escape character is '^]'.

Hello, this is zebra (version 0.95a).
Copyright 1996-2004 Kunihiro Ishiguro.

User Access Verification

Password:
Router> ena
Router> enable
Password:
Router# █
```

وارد شدن به `zebra` با استفاده از `telnet`

شما به راحتی مثل یک روتر سیسکو می توانید فرمانها را اجرا کنید و خروجی را مشاهده کنید، برای مثال برای مشاهده کردن وضعیت جاری از فرمان `show running-config` به صورت زیر استفاده کنید:



```

Terminal
File Edit View Terminal Tabs Help
Router# show running-config

Current configuration:
!
hostname Router
password zebra
enable password root
!
interface em0
 ip address 192.168.1.1/24
 ipv6 nd suppress-ra
!
interface lo0
!
!
line vty
!
end
Router#

```

فرمان `show running-config` در `zebra`

شما به راحتی مثل یک روتر می توانید وارد بخش پیکربندی شده و بر روی کارت شبکه خود آدرس IP اضافه کنید:

```

Terminal
File Edit View Terminal Tabs Help
Router# configure terminal
Router(config)# interface em0
Router(config-if)# ip address 192.168.2.2/24
Router(config-if)# exit
Router(config)# exit

```

تنظیم کردن آدرس IP در کارت شبکه با استفاده از `zebra`

روش اتصال با `vtys`

در زیر شاخه فایل‌های پیکربندی موجود در شاخه `/usr/local/etc/zebra` فایل وجود دارد به نام `vtys.conf.sample` که شما باید نام آنرا به `vtys.conf` تغییر دهید، این فایل توسط فرمان `vtys` خوانده شده و شما را وارد بخش مدیریت `Zebra` می کند در زیر فایل پیکربندی این فرمان را مشاهده می کنید:

```

Terminal
File Edit View Terminal Tabs Help
^[(escape) menu ^y search prompt ^k delete line ^p prev li ^g prev page
^o ascii code ^x search ^l undelete line ^n next li ^v next page
^u end of file ^a begin of line ^w delete word ^b back 1 char
^t top of text ^e end of line ^r restore word ^f forward 1 char
^c command ^d delete char ^j undelete char ^z next word
=====line 1 col 0 lines from top 1=====
! vtys sample configuratin file
!
!username kunihiro nopassword

```

استفاده از `vtys` در `zebra`



سرویس NIS یا Network Information System در FreeBSD

سرویس NIS برای مدیریت کردن مرکز منابع برای سیستم عاملهای شبه یونیک مثل سولاریس و خانواده BSD طراحی و ایجاد شده است. در زمان قدیمی این سرویس به نام Yellow Pages معروف بود ولی به دلیل مشکلات ایجاد شده برای برند آن نام به NIS تغییر پیدا کرده است ولی باز هم فرمانهای این برنامه با yp شروع می شوند و در بسیاری از موارد به yellow pages معروف است.

NIS از سیستم مشتری/ سروری مبتنی بر پروتکل RPC استفاده می کنید که به همین دلیل می توانید فایل‌های پیکربندی خود را در بین سیستم‌هایی که در یک دامنه NIS قرار دارند به اشتراک بگذارید. همین قابلیت به مدیران سیستم این اجازه را می دهد که به صورت مرکزی بتوانند تغییراتی در سیستم های مورد نظارت خود به صورت مرکزی انجام دهند. در ادامه این مقاله با این سرویس آشنا می شوید.

در سیستم عامل FreeBSD از ورژن 2 NIS استفاده می شود.

اصطلاحات استفاده شده در NIS

بخش NIS domain name

سرور و کلاینت اطلاعات خود را در یک دامنه اسمی به اشتراک گذاشته می شود. این نام انتخابی است.

سرویس rpcbind

این سرویس باعث فعال شدن RPC می شود که باعث برقراری ارتباط بین سرور و کلاینت می شود.

سرویس ypbind

این برنامه هسته اصلی ارتباطات در محیط NIS است. اگر این سرویس بر روی سیستم کلاینت فعال نشود نمی تواند به سرور متصل شود.

سرویس ypserv

این برنامه برای راه اندازی سرور NIS استفاده می شود. در راه اندازی NIS شما می توانید علاوه بر سرور اصلی سروی جایگزین هم داشته باشید تا در صورتی که این سرویس غیرفعال شد درخواستها به سمت سرور جایگزین ارسال شود.

برنامه: rpc.yppasswdd

این برنامه فقط در سرور اصلی NIS نصب و راه اندازی می شود و به کلاینتهای شبکه NIS این اجازه می دهد که بتوانند رمز عبور مربوط به NIS خودشان را تغییر بدهند. اگر این سرویس راه اندازی نشده باشد کلاینت ها توان تغییر دادن رمز عبور خود را نخواهند داشت.



نقش های موجود در NIS

در شبکه های مبتنی بر NIS سه نقش وجود دارد که در ادامه این سه نقش را توضیح می دهیم.

نقش Master

این سرور نقش اصلی را در NIS ایفا می کند و تمام اطلاعات و فایل های پیکربندی شبکه شما بر روی این سیستم ذخیره شده است، فایل های اصلی در این بخش فایل های password فایل groups و سایر فایل های پیکربندی در این سرور ذخیره می شود.

نقش slave

این سرور در حقیقت یک کپی از سرور اصلی است که در صورت از دسترس خارج شدن سرور اصلی از شبکه این سرور پاسخگویی در شبکه داشته باشد. از این سرور هم می توان به عنوان تقسیم کننده درخواستها و ترافیک بین سرور اصلی هم باشد.

نقش Client

سایر سیستم هایی که در شبکه NIS وجود دارد و از سرور اصلی سرویس می گیرد سیستم های Client هستند.

انتخاب کردن نام NIS Domain

از این نام برای برقراری ارتباط درون شبکه ای استفاده می شود. در بعضی از موارد برخی از سازمان ها این نام را با نام DNS خود یکی می کنند که برای رفع کردن ایرادات شبکه ای مشکل ایجاد می کند. پس در انتخاب این نام دقت کنید. در انتخاب کردن سروری که قصد دارید NIS را بر روی آن نصب کنید به این نکته دقت کنید که سروری را انتخاب کنید که همیشه در شبکه در دسترس بوده و خاموش و روشن نشود.

نصب کردن NIS Master Server

قبل از راه اندازی کردن فرمان های مربوط به سرور NIS در ابتدا در فایل r.conf خط زیر را وارد کنید و همه تنظیمات شبکه خود را دوباره راه اندازی کنید، این سرویس به نام دامنه NIS حساس بوده و در صورت موجود نبودن خطا ایجاد می کند:

```
nisdomainname="Domain name"
```

برای راه اندازی کردن مجدد تمام تنظیمات شبکه خود فرمان زیر را وارد کنید:

```
# /etc/netstart
```




تنظیمات DNS خود را هم چک کنید.

تمام فایل های اصلی در این سرور ذخیره می شود، دیتابیس که اطلاعات در آن ذخیره می شود NIS Mape نام دارد که در زیر شاخه /var/yp ذخیره می شود البته در شاخه به نام همان NIS domain این حالت زمانی خوب است که شما بتوانید چندین نام NIS بر روی یک سرور داشته باشید. پردازش اصلی این سرور ypserv است که به همه درخواستها پاسخ می دهد. در حقیقت این پردازش یک مترجم بین درخواستهای کلاینت هاو دیتابیس است.

شما در FreeBSD نیازی نصب NIS ندارید چون به صورت پیش فرض در خود سیستم نصب شده است و شما فقط نیاز دارید که آنرا از طریق فایل rc.conf سرور NIS را فعال کنید برای فعال کردن این سرور خطوط زیر را در فایل /etc/rc.conf اضافه کنید:

```
nisdomainname="abedini"
nis_server_enable="YES"
nis_yppasswdd_enable="YES"
```

در خط اول شما یک نام برای دامنه اسمی خود انتخاب کرده اید به نام abedini ، با استفاده از گزینه خط دوم در زمان راه اندازی سیستم سرویس سرور NIS را فعال می کنید. و در خط سوم سرویسی که مخصوص سرور است را راه اندازی می کند.

بعد از اعمال تنظیمات شما می توانید با فرمان زیر سرور را راه اندازی کنید:

```
# service ypserv start
```

در صورت بروز خطا مبنی بر وجود شاخه در زیر شاخه /var/yp شاخه مورد نظر مطابق با نام NIS Domain را ایجاد کنید.

ایجاد کردن NIS Maps

قبل از راه اندازی این بخش باید ypserv را راه اندازی کنید.

این دیتابیس از فایل های پیکربندی که در زیر شاخه /etc قرار دارد ایجاد می شود، به استثنای فایل /etc/master.passwd. برای جلوگیری از انتشار رمزهای عبور در بین تمام سرور ها به صورت دستی این فایل را زیر شاخه /var/yp کپی کنید و در مرحله بعدی تمام نام کاربری هایی که در سیستم NIS به آنها نیازی ندارید را حذف کنید، از فرمان های زیر برای کپی کردن فایل استفاده کنید:

```
# cp /etc/master.passwd /var/yp/master.passwd
# cd /var/yp
# vi master.passwd
```



سطح دسترسی 600 را بر روی این فایل اعمال کنید. برای ایجاد کردن دیتابیس در FreeBSD فرمانی وجود دارد به نام `ypinit` که شما با استفاده از آن می توانید سرور خود را ایجاد کنید، برای ایجاد کردن نقش `master` از سویچ `-m` استفاده کنید و بعد از آن نام انتخابی خود را وارد کنید، خروجی فرمان زیر به صورت زیر است:

```
FreeBSD# ypinit -m abedini
Server Type: MASTER Domain: abedini
Creating an YP server will require that you answer a few questions.
Questions will all be asked at the beginning of the procedure.
Do you want this procedure to quit on non-fatal errors? [y/n: n] n
Ok, please remember to go back and redo manually whatever fails.
If not, something might not work.
At this point, we have to construct a list of this domains YP servers.
rod.darktech.org is already known as master server.
Please continue to add any slave servers, one per line. When you are
done with the list, type a <control D>.
master server   : pc
next host to add: ^D
The current list of NIS servers looks like this:
ellington
coltrane
Is this correct? [y/n: y] y

[..output from map generation..]

NIS Map update completed.
FreeBSD has been setup as an YP master server without any errors.
```

ر زمان نوشتن نام `slave server` باید برای خارج شدن از آن بخش از کلید های `control` و کلید `D` استفاده کنید، بعد از اتمام این بخش فایلی ایجاد می شود در زیر شاخه `yp` به نام `ypserver` که شما می توانید بعد از اتمام این بخش به آن اضافه کنید.

بعد از اتمام این مراحل خط انتهای خروجی به شما می گویند که سرور شما بدون خطا ایجاد شده است.



پیکربندی FreeBSD Client در NSI

در این بخش با روش های پیکربندی سیستم های client برای برقرار ارتباط با سرور های NIS آموزش داده می شود، در بخش قبلی سرور NIS پیکربندی شده است و در این بخش ما قصد داریم که مدیریت کردن کاربران را با استفاده از NIS مرکزی کنیم، قبل از شروع به کار کردن نکات زیر را رعایت کنید:

قبل از شروع در فایل host واقع شده در زیر شاخه /etc آدرس ip و نام هایی را که از آنها در شبکه خود استفاده می کنید را وارد کنید، فرض کنید که شما دو سیستم در شبکه خود دارید که یکی حال سروری دارد و یکی کلاینت است. به اختصار یکی را server و دیگری را client نام گذاری کنید و در بخش hostname در rc.conf هم این نام ها را تعیین کنید، در شکل زیر یک نمونه از تنظیمات فایل hots را مشاهده می کنید:

```

Terminal
File Edit View Terminal Tabs Help
^[ (escape) menu ^y search prompt ^k delete line ^p prev li ^g prev page
^o ascii code ^x search ^l undelete line ^n next li ^v next page
^u end of file ^a begin of line ^w delete word ^b back 1 char
^t top of text ^e end of line ^r restore word ^f forward 1 char
^c command ^d delete char ^j undelete char ^z next word
=====line 5 col 19 lines from top 5 =====
# $FreeBSD: releng/10.3/etc/hosts 109997 2003-01-28 21:29:23Z dbaker $
#
# Host Database
192.168.3.31 server
192.168.3.30 client
#
# This file should contain the addresses and aliases for local hosts that
# share this file. Replace 'my.domain' below with the domainname of your
# machine

```

تعریف کردن نام ها در فایل host

همانطوری که در شکل بالا مشاهده می کنید برای client آدرس 192.168.3.30 و برای سرور 192.168.3.31 تنظیم شده است. این بخشها اختیاری بوده و به طراحی شبکه شما بستگی دارد، به این نکته توجه کنید که این تنظیمات بر روی سیستم های دیگر هم اعمال شود و یا از سرویس DNS در شبکه استفاده کنید.

نکته مهم:

قبل از راه اندازی کردن فرمان های مربوط به سرور NIS در ابتدا در فایل r.conf خط زیر را وارد کنید و همه تنظیمات شبکه خود را دوباره راه اندازی کنید، این سرویس به نام دامنه NIS حساس بوده و در صورت موجود نبودن خطا ایجاد می کند:

```
nisdomainname="Domain name"
```

برای راه اندازی کردن مجدد تمام تنظیمات شبکه خود فرمان زیر را وارد کنید:

```
# /etc/netstart
```




+.....

نکته:

تا زمانی که کاربر در فایل `master.passwd` در شاخه `/var/yp` اضافه نشده باشد اجازه وارد شدن به سیستم را نخواهد داشت، این بخش را قبل از هر اقدامی تست کرده و حتما برای کاربر رمزعبوری هم با `passwd` در نظر بگیرید؛ کاربر بدون رمز هم نمی تواند وارد شود.



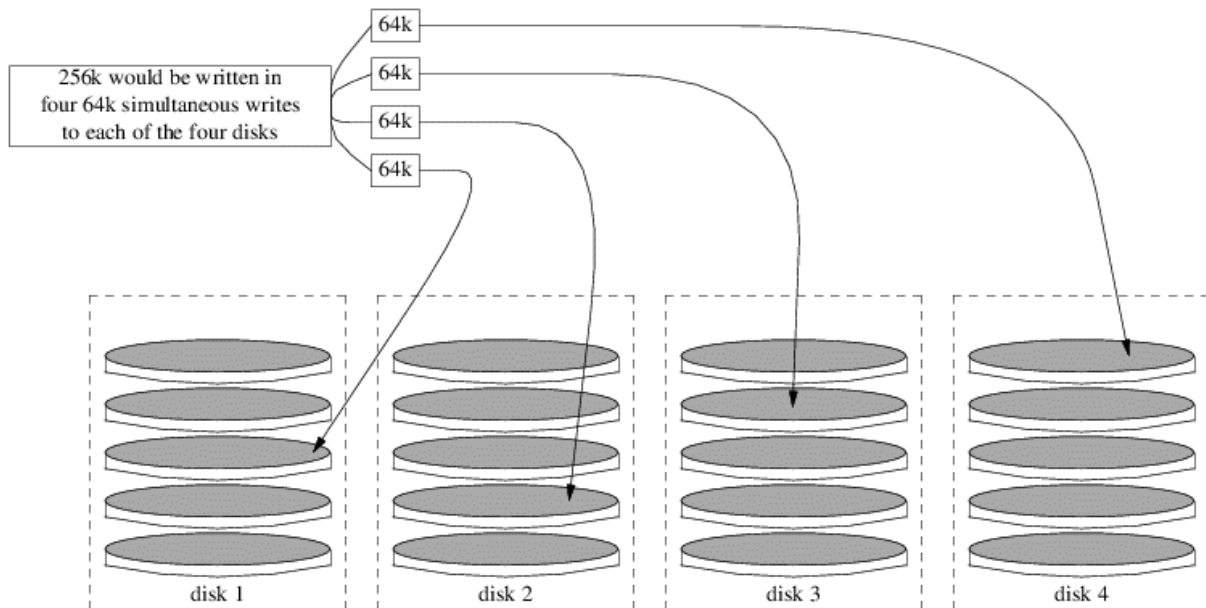
FreeBSD در RAID0

یکی از بخشهای مهم در مدیریت سیستم هایی که اطلاعات مهم شما بر روی آن ذخیره شده است استفاده کردن از مکانیزم ها و ریشهایی است که شما بتوانید در زمانی که اطلاعات شما به هر دلیل از بین رفت، فایل های خود را بازگردانی کنید، برای این منظور دو روش وجود دارد، روش اول استفاده کردن از سیستم ها و روش های backup گیری است که در این روش در ساعات خاصی شما فایل های مهم سیستم خود را در محلی دیگر و بر روی سیستمی دیگر ذخیره می کنید، در این روش هم شما می توانید از فرمانهای ساده برای انجام دادن این کار مثل cp و sftp برای جابجایی و منتقل کردن فایلها بر روی شبکه از آن استفاده کنید و یا از برنامه ای خودکار و حرفه ای Bacula برای انجام دادن این کار استفاده کنید.

این روش دارای یک عیب بزرگ است که اگر هاردی که شما فایل های backup خود را بر روی آن قرار می دهید دچار مشکل شود شما همه اطلاعات خود را از دست داده اید، برای جلوگیری کردن از این مشکل ف سیستمی طراحی شده است به نام RAID که هدف اصلی آن تقسیم کردن فایل های سیستم شما در بین چندین هارد است که در صورتی که یکی از هارد های شما خراب شد به سرعت شما بتوانید از اطلاعات خود استفاده کنید، البته این سیستم هم دارای عیب شکسته شدن RAID است که در صورتی که RIAD از بین برود سیستم بازگردانی فایل های شما دچار مشکل خواهد شد، البته RAID ها برای بالا بردن سرعت خواندن و نوشتن سیستم هم استفاده می شود که انواع مختلفی دارد که برای مثل raid0 قابلیت بازگردانی فایل های سیستم شما را نداشته و فقط برای افزایش سرعت به کار می رود.

در ادامه این بخش و در مقالات پیش رو شما با روش های راه اندازی RIAD هایی نرم افزاری موجود در FreeBSD آشنا می شود، همانطوری که می دانید RAID بر دو نوع کلی سخت افزاری و نرم افزاری تقسیم می شود و در سیستم عامل FreeBSD شما می توانید از RIAD نرم افزاری به استفاده کردن از GEOM framework استفاده کنید و سیستم خود را به صورتی مدیریت کنید که در صورت بروز مشکل بتوانید اطلاعات سیستم خود را بازیابی کنید.

در این مقاله شما با riado آشنا می شوید که در آن شما چندین دیسک را به یک دیسک تبدیل می کنید، با این روش هم شما فضای دیسک خود را افزایش داده و هم سرعت در خواندن و نوشتن را به تناسب سرعت هر دیسک افزایش می دهید، به این نوع از raid به اصطلاح Striping می گویند که در شکل زیر یک طرح کلی و مجازی از آنرا مشاهده می کنید:



معرفی حالت‌های riad

همانطوری که مشاهده می کنید در این نوع از raid نوشتن اطلاعات در بین چهار دیسک موجود سیستم شما تقسیم می شود و سرعت نوشتن شما چهار برابر قبل می شود. اندازه دیسک ها شما در این بخش باید یکسان باشد و I/O سیستم شما به صورت موازی در بین دیسک ها تقسیم می شود. همانطوری که قبلا اشاره شده این نوع از raid قابلیت redundancy ندارد و اگر اطلاعات شما مهم است از سایر روشهای backup گیری استفاده کنید چون زمانی که یک دیسک شما در این سیستم از دسترس خارج شود اطلاعات ذخیره شده روی آن از بین می رود، این سیستم برای سرور های کشی که اطلاعات متغییری را تولید می کنند و این اطلاعات دایمی نیستند بسیار مفید است چون سرعت نوشتن اطلاعات که یکی از باگهای راه اندازی این نوع از سرور ها است را می تواند افزایش دهد.

مراحل راه اندازی RAID0 در FreeBSD

در اولین قدم شما برای راه اندازی کردن این نوع از raid بر روی سیستم خود نیاز به بارگذاری کردن ماژول `geom_stripe.ko` در هسته سیستم عامل خود دارید، برای انجام دادن این عمل از فرمان `kldload` به صورت زیر استفاده کنید:

```
# kldload geom_stripe
```

در بخش بعدی شما باید یک مسیر یا شاخه بر روی سیستم عامل خود ایجاد کنید تا `RADiO` را به اصطلاح در آن شاخه `mount` کنید، البته در `FreeBSD` شاخه ای به نام `/mnt` برای این منظور در نظر گرفته شده است.

در مرحله بعدی شما باید دیسک های را که برای این منظور در نظر گرفته اید را به سیستم خود اضافه کنید و با استفاده از فرمان `gstripe` که در ادامه روش اجرا کردن آنرا مشاهده می کنید دیسک های خود را به یک دیسک واحد تبدیل کنید.

دیسک های اضافه شده در زیر شاخه `/dev` و بسته به نوع آنها در `FreeBSD` ناگذاری می شود،



در مثال زیر فرض بر این است که دو دیسک برای ایجاد کردن RAID0 به سیستم شما اضافه شده که این دیسک ها قبلا هیچ پارتیشنی بر روی خود ندارند و نام آنها `/dev/ad2` و `/dev/ad3` هستند، و به صورت انتخابی و اختیاری نام `st0` را برای RAID خود در نظر بگیرید، فرمان های اجرا شده را در بخش زیر مشاهده می کنید:

```
# gstripe label -v st0 /dev/ad2 /dev/ad3
Metadata value stored on /dev/ad2.
Metadata value stored on /dev/ad3.
Done.
```

همانطوری که مشاهده کرده اید این دو دیسک **Striping** شده. بعد از اجر شدن این فرمان در زیر شاخه `/dev/` شاخه ای به نام **stripe** ایجاد شده که **Device** به نام `st0` که در حقیقت دو دیسک شما در قابل این یک دیسک است در آن شاخه ایجاد شده است.

در قدم بعدی شما باید **label** استاندارد موجود بر روی سیستم **FreeBSD** را که به اصطلاح **partition table** است را بر روی دیسک `st0` ایجاد کنید، برای انجام دادن این بخش از فرمان `bsdlablel` به صورت زیر استفاده کنید:

```
# bsdlablel -wB /dev/stripe/st0
```

بعد از اجرا شدن این فرمان **partition table** مخصوص **FreeBSD** بر روی دیوایس `st0` که خود از دو هارد RAID شده از نوع **Striping** است ایجاد می شود، در مرحله بعدی شما نیاز دارید که فایل سیستمی بر روی دیسک مورد نظر خود ایجاد کنید، برای این منظور باید از فرمان **newfs** به صورت زیر استفاده کنید و این فرمان را اجرا کنید تا فضای دیسک شما برای نوشتن و خواندن آماده شود:

```
# newfs -U /dev/stripe/st0a
```

بعد اجرا این فرمان به درستی و مشاهده کردن خروجی فرمان ، دیسک شما آماده به کار است و شما می توانید به راحتی و با استفاده کردن از فرمان **mount** به صورت زیر از این فضای دیسک بر روی **mount Point** که در نظر گرفته اید استفاده کنید:

```
# mount /dev/stripe/st0a /mnt
```

همانطوری که در فرمان بالا مشاهده می کنید شما می توانید به **RIAD** ایجاد شده را طریق شاخه `/mnt` دسترسی پیدا کنید.

راه اندازی خودکار RAID0 در زمان راه اندازی FreeBSD:

یکی از کارهایی که شما باید بعد از اتمام این کار برای دایمی شدن تنظیمات خود انجام دهید در این بخش توضیح داده می شود، به هر دلیل شاید سیستم شما دوباره راه اندازی شود، پس شما باید تنظیمات خود را در **FreeBSD** ذخیره کنید تا سرویس



هایی که از این شاخه استفاده می کنند در زمان راه اندازی مجدد سیستم دچار مشکل نشوند. این بخش به دو مرحله کلی تقسیم میشود.

در مرحله اول شما باید ماژول `geom_stripe.ko` که مورد نیازی این نوع از RAID است را در زمان راه اندازی سیستم در هسته بارگذاری کنید، شما یا می توانید آن قابلیت را به صورت دائمی در هسته FreeBSD ایجاد کنید و با به صورت زیر با استفاده از فایل `loader.conf` آنرا در زمان راه اندازی به هسته اضافه کنید، برای انجام دادن این کار فرمان زیر را اجرا کنید:

```
# echo 'geom_stripe_load="YES"' >> /boot/loader.conf
```

با اجرا کردن این فرمان خط `geom_stripe_load="YES"` در فایل `loader.conf` قرار میگیرد، این فایل فایلی است که در زمان `boot` شدن سیستم توسط FreeBSD خوانده می شود و ماژول ها را قبل از بارگذاری شدن کامل هسته در هسته بارگذاری می کنید، این عملی است هوشمندانه به این دلیل که یکی از وظیفه های اصلی هسته مدیریت کردن دیسکها هستند.

در FreeBSD فایلی قرار دارد که در زمان راه اندازی شدن سیستم و قبل از راه اندازی کامل خوانده شده و دیسک های سیستم شما را در زیر شاخه ها مورد نظر به اصطلاح `mount` می کند تا سایر فرمان های راه انداز مثل سیستم `rc` بتواند به فایلها و فرمانهای راه اندازی سرویس ها دسترسی پیدا کنند، نام این فایل `fstab` است و در زیر شاخه `/etc` قرار دارد و شما باید فرمان زیر را اجرا کنید تا خط مربوط به راه اندازی RIADO در آن ایجاد شود:

```
# mkdir /stripe
# echo "/dev/stripe/st0a /stripe ufs rw 2 2" \
>> /etc/fstab
```

در خط اول این فرمان شاخه ای انتخابی به نام `stripe` ایجاد شده تا به صورت خودکار هارد های `raid0` بر زیر شاخه آن قرار گیرند.



راه اندازی RAID1 در FreeBSD

در بخش قبلی شما با RAID0 و مراحل راه اندازی، محاسن و معایب آن آشنا شده اید، در این بخش با نوعی دیگر از سیستم RAID به نام RAID1 یا mirroring آشنا می شوید. در این نوع از RAID اطلاعات بر روی بیش از یک دیسک و به صورت کامل نوشته می شود و هر دیسک شامل اطلاعات کاملی از داده های شماست به این دلیل است که این نوع از RAID را حالت full backup هم می نامند و در صورتی که یکی از دیسک ها شما در RAID1 از دسترس خارج شود در سرویس دهی شما اختلالی ایجاد نمی شود و مدیر سیستم می تواند هارد خراب را با هارد جدید تعویض کند و دوباره اطلاعات را در حالت RAID1 ذخیره کند. در این نوع از RAID1 سرعت نوشتن کند بود این علت است که همزمان اطلاعات باید در بین چند دیسک کپی شود و لی سرعت خواندن بالاست چون سیستم عامل شما می تواند اطلاعات را از چند دیسک mirror شده مطالعه کند.

دو حالت در این بخش توضیح داده می شود، یکی ایجاد کردن RAID1 در زمانی که شما قصد دارید دو دیسک جدید را با هم mirror کنید و در حالت شما mirror را از دیسکی می گیرید که اطلاعات بر روی آن ذخیره شده است، این حالت هم زمانی مفید است که شما یکی از دیسک های خود را در RAID1 از دست داده اید و قصد دارید دیسک جدید به سیستم اضافه کنید.

پاک کردن Metadata دیسک:

قبل از شروع شما باید metadata ها موجود بر روی هر دیسک را پاک کنید، این داده ها در پایان فضا هر دیسک ذخیره می شود و قبل از اینکه شما قصد راه اندازی RAID1 را داشته باشید باید پاک شود. در این مرحله شما با دو نوع metadata سره کار دارید، metadata ای که در نوع پارتیشن بندی GPT ایجاد می شود و metadata که از سیستم RAID1 یا همان mirroring بر روی دیسک شما ایجاد شده است.

پاک کردن در حالت GPT:

برای پاک کردن metadata بر روی دو دیسک اصلی و پشتیبان در RAID1 که از GPT استفاده می کنند شما از فرمان gpart می توانید استفاده کنید برای این منظور فرمان زیر را اجرا کنید:

```
# gpart destroy -F ada8
```

خارج کردن دیسک از mirror و پاک کردن اطلاعات:

برای اینکه شما بتوانید یک دیسک را از سیستم RAID1 خارج کنید و اطلاعات metadata آنرا پاک کنید در قدم اول باید فرمان زیر را برای خارج کردن دیسک استفاده کنید:

```
# gmirror remove gm4 ada8
```



حال برای پاک کردن از فرمان زیر استفاده کنید:

```
# gmirror clear ada8
```

ایجاد کرد RAID1 برای دو دیسک جدید:

در این بخش فرض بر این است که در سیستم شما FreeBSD بر روی دیسک ada0 نصب شده و شما دو دیسک جدید به نام های ada1 و ada2 دارید که قصد دارید این دو دیسک را با هم mirror کنید.

در قدم اول شما باید مازول مربوط به mirroring را در هسته بارگذاری کنید، برای اعمال این تغییرات در زمان راه اندازی سیستم از فرمان kldload به صورت زیر استفاده کنید:

```
# gmirror load
```

در مرحله بعدی شما باید با استفاده از فرمان زیر دو دیسک مورد نظر خود را mirror کنید:

```
# gmirror label -v gm0 /dev/ada1 /dev/ada2
```

تا این مرحله حالت mirror ایجاد شده و بعد از ایجاد شده آن در زیر شاخه dev زیر شاخه دیگری به نام mirror ایجاد می شود که دیوایس ها در زیر شاخه آن قرار می گیرد.

هدف بعدی ما در این بخش این است که کل هاردی که FreeBSD بر روی آن نصب است را به حالت mirror ایجاد شده منتقل کنیم. برای انجام دادن این انتقال در مرحله اول باید طرح پارتیشن بندی مورد نظر را با استفاده از gpart بر روی mirror ایجاد کنید برای مثال شما به پارتیشن های / و swap و /var و /tmp نیاز دارید، در طرح کلی شما می توانید فقط دو پارتیشن / و swap را ایجاد می کنیم:

```
# gpart create -s MBR mirror/gm0
# gpart add -t freebsd -a 4k mirror/gm0
# gpart create -s BSD mirror/gm0s1
# gpart add -t freebsd-ufs -a 4k -s 2g mirror/gm0s1
# gpart add -t freebsd-swap -a 4k -s 4g mirror/gm0s1
# gpart add -t freebsd-ufs -a 4k -s 2g mirror/gm0s1
# gpart add -t freebsd-ufs -a 4k -s 1g mirror/gm0s1
# gpart add -t freebsd-ufs -a 4k mirror/gm0s1
```

حال باید هارد mirror شده را bootable کنید، برای این منظور باید از فرمان gpart به صورت زیر استفاده کنید و بعد هم بید پارتیشن active را تعیین کنید:



```
# gpart bootcode -b /boot/mbr mirror/gm0
# gpart set -a active -i 1 mirror/gm0
# gpart bootcode -b /boot/boot mirror/gm0s1
```

در بخش بعدی باید با استفاده از فرمان **newfs** پارتیشن های ایجاد شده را **format** کنید، این عمل را باید برای هر پارتیشن به صورت زیر اجرا کنید:

```
# newfs -U /dev/mirror/gm0s1a
# newfs -U /dev/mirror/gm0s1d
# newfs -U /dev/mirror/gm0s1e
# newfs -U /dev/mirror/gm0s1f
```

منتقل کردن اطلاعات از دیسک قدیمی به **mirror**

برای این انتقال می توانید از دو فرمان **dump** و **restore** استفاده کنید، در هر بخش باید پارتیشن مورد نظر را در زیر شاخه **/mnt** به اصطلاح **mount** کنید و از دو فرمان **dump** و **restore** استفاده کنید، این دو فرمان پایه گرفتن **backup** در **FreeBSD** است، در ادامه شما با فرمان های منتقل کننده آشنا می شوید:

```
# mount /dev/mirror/gm0s1a /mnt
# dump -C16 -b64 -0aL -f - / | (cd /mnt && restore -rf -)
# mount /dev/mirror/gm0s1d /mnt/var
# mount /dev/mirror/gm0s1e /mnt/tmp
# mount /dev/mirror/gm0s1f /mnt/usr
# dump -C16 -b64 -0aL -f - /var | (cd /mnt/var && restore -rf -)
# dump -C16 -b64 -0aL -f - /tmp | (cd /mnt/tmp && restore -rf -)
# dump -C16 -b64 -0aL -f - /usr | (cd /mnt/usr && restore -rf -)
```

انتقال کامل در فایل **fstab**:

برای منتقل کردن کامل ابتدا باید ماژول **mirroring** را در هسته FreeBSD در زمان راه اندازی و با استفاده از فایل **loader.conf** انجام دهید برای انجام دادن این کار باید در فایل **/mnt/boot/loader.conf** خط زیر را اضافه کنید، به این نکته توجه کنید که باید پارتیشن **mirror** شده را در زیر شاخه **mnt** که فعلا موقتی است **mount** کنید:

```
geom_mirror_load="YES"
```

حال باید برای بخش پایانی این قسمت فایل **fstab** را که در مسیر **/mnt/etc/fstab** قرار گرفته است را به صورت زیر ویرایش کنید تا در زمان راه اندازی سیستم به صورت خودکار پارتیشن های شما **mount** شود:

# Device	Mountpoint	FStype	Options	Dump	Pass#
/dev/mirror/gm0s1a	/		ufs	rw	1 1
/dev/mirror/gm0s1b	none		swap	sw	0 0
/dev/mirror/gm0s1d	/var		ufs	rw	2 2
/dev/mirror/gm0s1e	/tmp		ufs	rw	2 2
/dev/mirror/gm0s1f	/usr		ufs	rw	2 2



راه اندازی RIAD03 در FreeBSD

آخرین نوعی از RIAD رو که شما می توانید در FreeBSD راه اندازی کنید RIAD03 است. در این مدل چند دیسک به صورت یک دیسک در می آید و یک دیسک به صورت parity عمل کرده و همه اطلاعات بین تمام دیسکها تقسیم شده تا در صورتی که یک دیسک از دسترس خارج شود اطلاعات موجود بتواند دیسک از دسترس خارج شده را دوباره بازسازی کرده. در این مدل شما به حداقل سه دیسک با سایزها و توان های یکسان نیاز دارید. تعداد دیسک های بیشتر به صورت 5، 9 و 17 عدد است.

در سیستم عامل FreeBSD از `graid3(8)` برای پیاده سازی RIAD03 استفاده می شود و در قدم اول شما باید مازول مورد نظر را در هسته FreeBSD بارگذاری کنید این عمل با استفاده از فرمان زیر قابل اجراست:

```
# graid3 load
```

و یا شما می توانید از فرمان `kldload` به صورت زیر استفاده کنید:

```
# kldload geom_raid3
```

در قدم بعدی شما باید شاخه مورد نظری را که قصد دارید از طریق آن به RIAD03 دسترسی داشته باشید را ایجاد کنید برای انجام این کار از فرمان `mkdir` به صورت زیر استفاده کنید:

```
# mkdir /multimedia
```

در مرحله بعد شما باید با استفاده از فرمان `graid3` به صورت زیر 3 دیسک مورد نظر خود را به یک دیسک تبدیل کنید ، این فرمان در زیر بیان شده است:

```
# graid3 label -v gr0 /dev/ada1 /dev/ada2 /dev/ada3
Metadata value stored on /dev/ada1.
Metadata value stored on /dev/ada2.
Metadata value stored on /dev/ada3.
Done.
```

بعد از اجرا فرمان بالا یک دیسک به نام `gr0` در زیر شاخه `/dev/raid3` ایجاد شده است که شما می توانید با استفاده از فرمان `gpart` بروی آن پارتیشن ایجاد کرده و با استفاده از فرم `newfs` فایل سیستمی بروی آن ایجاد کنید برای این بخش فرمانین مورد استفاده در زیر نمایش داده شده است:

```
# gpart create -s GPT /dev/raid3/gr0
# gpart add -t freebsd-ufs /dev/raid3/gr0
# newfs -j /dev/raid3/gr0p1
```



برای استفاده کردن از این پارتیشن باید آنرا mount کنید، به صورت زیر:

```
# mount /dev/raid3/gr0p1 /multimedia/
```

حال این پارتیشن در زیرشاخه ای که ایجاد کرده اید قابل دسترسی است.

برای mount شدن خودکار این پارتیشن در زمان راه اندازی شدن سیستم باید ماژول مورد نیاز را در زمان راه اندازی در هسته بارگذاری کنید و خط زیر را در فایل `boot/loader.conf/` قرار دهید:

```
geom_raid3_load="YES"
```

در گام بعدی باید فایل `fstab` موجود در زیر شاخه `/etc` قرار دارد را به صورت زیر ویرایش کنید تا در زمان راه اندازی سیستم به صورت خودکار این سیستم mount شود:

```
/dev/raid3/gr0p1 /multimedia    ufs    rw    2    2
```



راه اندازی کردن PXE در FreeBSD

تکنولوژی (PXE) Preboot eXecution Environment این اجازه را به شما می دهد تا بتوانید بدون داشتن هارد دیسک محلی سیستم عامل خود را راه اندازی کنید. برای مثال شما می توانید سیستم عامل FreeBSD را بدون داشتن هارد دیسک local از طریق NFS راه اندازی کنید. سیستم PXE به ساختار سخت افزاری سیستم شما مربوط می شود و از طریق BIOS فعال می شود.

برای راه اندازی کردن سیستم از طریق PXE شما به سرورهای DHCP TFTP و سرور NFS نیاز دارید. در مرحله اول که سیستم شما راه اندازی می شود سیستم شما از طریق DHCP یک آدرس IP دریافت می کند و در قسمت بعدی سیستم عامل از طریق tftp بارگذاری می شود و در بخش پایانی تمام فایل سیستم شما از طریق NFS در اختیار سیستم شما قرار می گیرد. پس برای راه اندازی کردن این سیستم شما به سه سرور ذکر شده در شبکه خود نیاز دارید.

در این مقاله سعی می شود که روش پیاده سازی PXE در سیستم عامل FreeBSD را برای شما بیان کنم. این مقاله خود در دو بخش ارایه می شود.

راه اندازی کردن TFTP و NFS

در اولین قدم با روش پیکربندی و آماده سازی محیط PXE در FreeBSD آشنا می شوید. در اولین قدم شما باید دو سرویسی که در خود سیستم عامل FreeBSD وجود دارد به نام های TFTP و NFS را پیکربندی کنید، در بخش بعدی هم با پیکربندی کردن DHCP آشنا می شوید.

در ابتدای کار شما باید یک شاخه ای که کاربران PXE از فایل های داخل آن استفاده می کنند را ایجاد کنید. برای مثال شاخه ای به آدرس `/b/tftpboot/FreeBSD/install` ایجاد کنید. ایجاد کردن این شاخه بسیار مهم است به این دلیل که در پیکربندی سرور از آن استفاده می کنید. این شاخه هم در فایل `/etc/inted.conf` و هم در فایل `/usr/local/etc/dhcp.conf` مورد استفاده قرار می گیرد.

در اولین بخش باید شاخه ای که ایجاد کردید را به صورت زیر در `nfs` به اصطلاح `mount` کنید به صورت اجرا کردن فرمان زیر:

```
# export NFSROOTDIR=/b/tftpboot/FreeBSD/install
# mkdir -p ${NFSROOTDIR}
```

این شاخه باید دارای فایل های نصبی FreeBSD باشد که در ادامه این بخش روش ایجاد کردن آن را مشاهده می کنید.

در قدم بعدی شما باید سرور NFS را در بخش سرور خود فعال کنید به صورت زیر این کار با ویرایش کردن و اضافه کردن خط زیر در فایل `etc/rc.conf` انجام می شود:

```
nfs_server_enable="YES"
```




در قدم بعدی باید شاخه ایجاد شده را به اشتراک گذارید، برای اشتراک گذاشتن فایل ها و پوشه ها در FreeBSD شما باید از فایل `/etc/exports` استفاده کنید، برای انجام این عمل خط زیر را در داخل فایل مذکور اضافه کنید:

```
/b -ro -alldirs
```

حال با استفاده از فرمان `service nfsd start` سرور `nfs` خود را راه اندازی کنید:

```
# service nfsd start
```

راه اندازی کردن: `tftp`

برای راه اندازی کردن سرور `tftp` شما باید از سیستم قدیمی `inted` استفاده کنید، این سرور در گذشته برای راه اندازی کردن سرویس های مثل `ftp` و حتی `ssh` هم مورد استفاده قرار می گرفته است و در حال حاضر هم در سیستم وجود دارد. فایل اصلی آن `/etc/inted.conf` است و برای فعال سازی آن کفایت که خط زیر را در `rc.conf` به صورت زیر اضافه کنید:

```
inetd_enable="YES"
```

حال در بخش بعدی باید قسمت مربوط به راه اندازی سرور `tftp` را در فایل `/etc/inted.conf` به صورت زیر ویرایش کنید:

```
tftp dgram udp wait root /usr/libexec/tftpd tftpd -l -s /b/tftpboot
```

این فایل شامل خطوطی است که هر خط مربوط به یک سرور است، برای راه اندازی `tftp` شما کفایت که مسیر مورد نظر خود را در بخش آخر اضافه کنید، سرور `tftp` به صورت پیش فرض `udp` است و در برخی از موارد برخی از ورژنهای `PXE` نیاز به برقرار کردن ارتباط از نوع `TCP` با سرور `tftp` دارند که باید شما خطی را فعال کنید که بخش `tcp` ان باشد.

نکته:

ابتدای هر خط علامت `#` یا همان `comment` قرار دارد که شما کفایت که فقط این علامت را از ابتدای خط که سرویس مورد نظر شماست پاک کنید و سرویس `inted` را راه اندازی کنید.

برای راه اندازی کردن سرویس `inted` که خودش باعث راه اندازی `tftp` می شود کفایت که فرمان زیر را در `FreeBSD` خود وارد کنید:

```
# service inetd start
```

برای ایجاد کردن فایل های نصبی در شاخه ایجاد شده شما باید هسته سیستم عامل خود را `rebuild` کنید که این امر با استفاده کردن از اجرای فرمان زیر امکان پذیر است، به این نکته توجه کنید که شما باید سورس سیستم عامل خود را در زمان نصب `FreeBSD` نصب کرده باشید و یا به مقاله پیکربندی هسته در `FreeBSD` برای دریافت اطاعات بیشتر مراجعه کنید:

```
# cd /usr/src
# make buildworld
```



```
# make buildkernel
```

حال شاخه نصبی خود را که با استفاده از nfs بر روی شبکه share کرده اید را با استفاده از فرمان زیر ایجاد کنید:

```
# make installworld DESTDIR=${NFSROOTDIR}
# make installkernel DESTDIR=${NFSROOTDIR}
# make distribution DESTDIR=${NFSROOTDIR}
```

مراحل اجرا شدن فرمان های بالا طولانی بوده و به سرعت سیستم شما بستگی دارد، هر بخش باید بدون خطا به پایان برسد. برای چک کردن از صحت راه اندازی سرور tftp و فایل هایی که شما آنها را ایجاد کرده اید باید فرمان زیر را اجرا کنید و یک فایل را به صورت زیر دانلود کنید:

```
# tftp localhost
tftp> get FreeBSD/install/boot/pxeboot
Received 264951 bytes in 0.1 seconds
```

حال باید فایل fstab مربوط به کلاینتها را ویرایش کنید، به این نکته توجه کنید که در قدم اول وارد شاخه ای شوید که فایل های نصبی را در آن قرار داده اید و در مرحله باید فایل هایی که در زیر شاخه etc/ به نام fstab قرار دارد را ویرایش کنید، فایل اصلی سیستم خود را ویرایش کنید، در این مثال شاخه مورد نظر b/tftpboot/FreeBSD/install/ می باشد در نتیجه فایل fstab در آدرس زیر قرار دارد:

```
/b/tftpboot/FreeBSD/install/etc/fstab
```

این فایل را به صورت زیر ویرایش کنید:

```
# Device                               Mountpoint  FSType  Options  Dump Pass
server:/b/tftpboot/FreeBSD/install    /           nfs     ro       0       0
```

در قسمت server در خط بالا شما آدرس ip سرور nfs خودتون را وارد کنید.

هر کلانیت به رمزعبور کاربر root برای استفاده کردن از PXE نیاز دارد، این کاربر با کاربر root موجود در سیستم شما متفاوت است و باید در قسمت فایل های راه اندازی که در شاخه b/tftpboot/FreeBSD/install/ است رمزعبور کاربر root را تغییر دهید. برای این کار باید از فرمان chroot برای تغییر دادن شاخه ریشه خط فرمان خود به شاخه b/tftpboot/FreeBSD/install/ استفاده کنید و بعد فرمان chpass را اجرا کنید، به صورت زیر می توانید از فرمان chroot استفاده کنید:

```
# chroot /b/tftpboot/FreeBSD/install/
# passwd
```



تا مراحل قبلی در قسمت راه اندازی سروری PXE با راه اندازی NFS و TFTP سرور و ایجاد کردن شاخه مورد نظر برای راه اندازی و بوت شدن سیستم ها در شبکه توضیحاتی ارائه شده است، در قسمت پایانی با راه اندازی کردن DHCP در خدمت شما دوستان هستیم.

راه اندازی DHCP برای PXE سرور:

سرور DHCP مثل دو سرور دیگر این مثال در FreeBSD به صورت پیش فرض نصب نبوده و شما در مرحله اول باید آنرا نصب کنید، شما به دو صورت می توانید این سرور را نصب کنید، برای به منظور به مقاله مقدمه ای بر نصب برنامه در FreeBSD مراجعه کنید، در این بخش از سیستم ports برای نصب استفاده کنید برای این منظور فرمانهای زیر را در اجرا کنید:

```
#cd /usr/ports/net/isc-dhcp43-server
#make
#make install
```

در صورتی که شما با خطایی در مراحل نصب مواجه نشده باشید بسته مورد نظر شما به درستی نصب شده است.

فایل پیکربندی سرور DHCP در مسیر /usr/local/etc قرار دارد البته نام این فایل متفاوت است و باید با استفاده از فرمان زیر نام فایل مورد نظر را به dhcp.conf تغییر دهید:

```
#cp /usr/local/etc/dhcp.conf.example /usr/local/etc/dhcp.conf
```

در این فایل باید بخش netx-server را به صورت زیر ویرایش کنید و بخش های filename و root-path را هم به صورت زیر تغییر دهید:

```
subnet 192.168.0.0 netmask 255.255.255.0 {
    range 192.168.0.2 192.168.0.3 ;
    option subnet-mask 255.255.255.0 ;
    option routers 192.168.0.1 ;
    option broadcast-address 192.168.0.255 ;
    option domain-name-servers 192.168.35.35, 192.168.35.36 ;
    option domain-name "example.com";

    # IP address of TFTP server
    next-server 192.168.0.1 ;

    # path of boot loader obtained via tftp
    filename "FreeBSD/install/boot/pxeboot" ;
```



```
# pxeboot boot loader will try to NFS mount this directory for root FS
option root-path "192.168.0.1:/b/tftpboot/FreeBSD/install/" ;

}
```

در بخش netx-server باید آدرس ip سرور TFPT تنظیم شده در بخش قبل را مشخص کنید.

در بخش filename باید مسیر راه انداز pxeboot را مشخص کنید، به این نکته توجه داشته باشید که /b/tftpboot از این مسیر حذف شده است.

بخش دیگری را هم شما باید پیکربندی کنید به نام root-path که شاخه root سرور NFS را مشخص می کند.

در این بخش سرور dhcp شما آماده راه اندازی است و شما باید در بخش اول در فایل rc.conf خط زیر را وارد کنید:

```
dhcpcd_enable="YES"
```

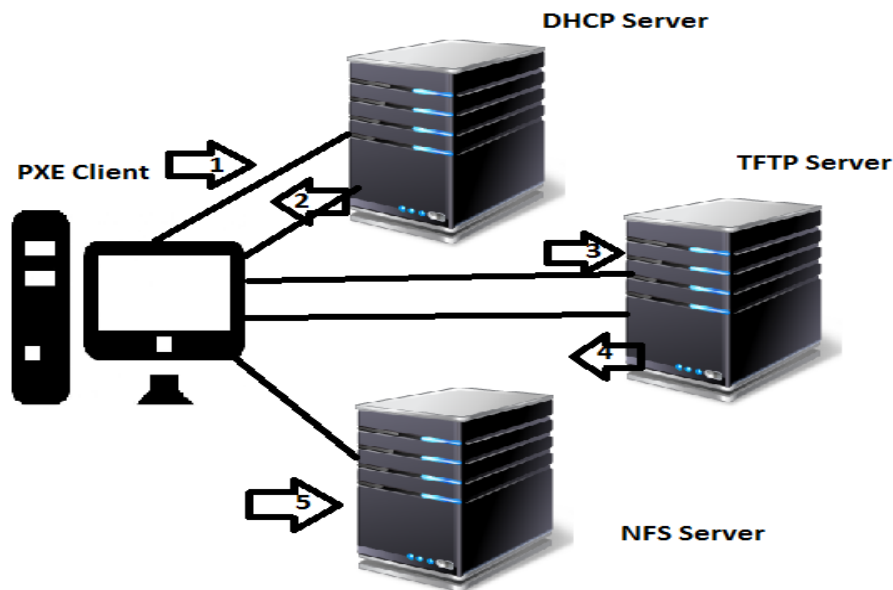
در بخش بعدی باید با استفاده از فرمان Service به صورت زیر سرور را راه اندازی کنید:

```
# service isc-dhcpd start
```

در پایان این بخش قسمت سرور تنظیم شده و آماده کار است حال باید شما در بخش client سیستم های مورد نظر خود را به صورتی تنظیم کنید که از طریق شبکه راه اندازی شود، هم باید کارت شبکه شما از این قابلیت پشتیبانی کند و هم باید در Bios سیستم شما قابلیت راه اندازی از طریق شبکه را داشته باشد.

مراحل راه اندازی سیستم از طریق PXE:

در شکل زیر شما با مراحل راه اندازی شدن به تصویر کشیده شده است:



مراحل راه اندازی

در مرحله اول Client یک بسته با حالت پخش فراگیر به مدل DHCPDISCOVER بروی شبکه ارسال می کند تا سرور DHCP را پیدا کردن تا آدرس TFTP سرور NFS و root-path را از سرور DHCP دریافت می کند.

سرور DHCP در مرحله دوم جواب درخواست سیستم Client را با مشخصات تنظیم شده را ارسال می کند.

در مرحله سوم سیستم Client در خواستی به سرور TFTP ارسال می کند filename را دریافت کند.

در مرحله چهارم درخواست از سمت سرور TFTP ارسال می شود.

در این قسمت سیستم Client به وسیله سیستم pxeboot هسته FreeBSD را راه اندازی می کند و با استفاده از-root path به سرور NFS متصل می شود تا پارتیشن موجود را mount کند تا بتواند به کار خود ادامه دهد.



File Flag و kernel secure level در FreeBSD

یکی دیگر از امکانات پیشرفته سیستم عامل های مبتنی بر BSD استفاده کردن از یک سطح دسترسی است که جدای دو سطح دسترسی معمولی و ACL است. این سطح دسترسی به File Flag معروف است و شما می توانید علاوه بر دو روش ذکر شده سطح دسترسی مثل append-only یا undelete را برای افزایش دادن امنیت سیستم خود به فابلها اضافه کنید. در زیر شما لیست file flag ها موجود در FreeBSD را مشاهده می کنید:

Archived با این flag فایل حتما باید آرشیو شود.

Opaque این flag بر روی شاخه اعمال می شود و در حالتی که شما شاخه ای را در زیر شاخه دیگر mount می کنید باعث مخفی شدن شاخه می شود. به این نوع از mount کردن به اصطلاح unionfs می گویند.

Nodump فایلی که این flag را داشته باشد به هیچ عنوان backup گرفته نمی شود.

Sappend هر فایلی که شامل این flag باشد فقط می توانید به آخر آن اضافه کنید و به هیچ عنوان قابل تغییر محتوای قبلی نیست.

Schange فایلی که شامل این flag باشد به هیچ عنوانی برای هیچ کاربری قابل تغییر دادن نیست.

Sunlink لینک کردن یک فایل یکی از قابلیت های موجود در BSD است شما می توانید بعد از این کار flag مذکور را به فایل اضافه کنید تا کسی نتواند آنرا از حالت link خارج کند.

Uappend فایلی که شامل این flag باشد فقط توسط صاحب آن فایل قابل اضافه کردن است و دیگران به آن دسترسی ندارند.

Uchange فایلی که شامل این flag باشد فقط توسط صاحب آن فایل قابل تغییر دادن است و دیگران به آن دسترسی ندارند.

Uunlink فایلی که شامل این flag باشد فقط توسط صاحب آن فایل قابل تغییر دادن link و دیگران به آن دسترسی ندارند.

در لیست بالا آن دسته از flag هایی که با حرف S شروع می شود فقط توسط کاربر root قابل تغییر است.

پیشنهاد ما در استفاده کردن از flag ها به این صورت است:

- باید سعی کنید که برای فایل های log که بخش مهمی در کشف کردن حملات و تغییرات در سیستم شما هستند از sappend استفاده کنید.
- سعی کنید که فایل های مهم پیکربندی سیستم را در حالتی قرار دهید که قابل تغییر نباشد. حتی در بخش بعدی شما یاد خواهید گرفت که به چه صورت دسترسی کاربر root را هم تغییر دهید تا در صورتی که سیستم شما به خطر افتاد و هکری با کاربر root به سیستم شما وارد شد نتواند فایل های پیکربندی اصلی را که امنیت سیستم شما را حفظ می کند تغییر دهد.



اعمال کردن file flag در FreeBSD

برای اعمال کردن file flag در FreeBSD شما باید از یک فرمان خاص به نام `chflags` استفاده کنید. در شکل زیر یک مثال از این بخش را مشاهده می کنید:

```

Terminal
File Edit View Terminal Tabs Help
root@server:~/flag # touch test
root@server:~/flag # chflags schg test
root@server:~/flag # rm test
override rw-r--r-- root/wheel schg for test? y
rm: test: Operation not permitted
root@server:~/flag # █

```

مثالی از استفاده از فرمان `schg`

در خط اول این بخش با استفاده از فرمان `touch` یک فایل ایجاد شده است.

در خط دوم با استفاده از فرمان `chflags` به فایل ایجاد شده حالت بدون تغییر را اضافه کردیم.

همانطور که مشاهده می کنید دسترسی در shell با کاربر `root` است و در خط بعدی فصد داریم که فایل را با فرمان `rm` با سطح دسترسی کاربر `root` حذف کنیم که این عمل امکانپذیر نیست.

با استفاده از فرمان بالا و `flag` مشاهده کردید که می توانید سطح دسترسی جدید به فایلها اضافه کنید، برای مشاهده کردن `file flag` شما باید از فرمان `ls` به سوئیچ های `-lo` استفاده کنید که در شکل زیر خروجی این فرمان را در شاخه قبلی که شما فایل در این ایجاد کرده اید را مشاهده می کنید:

```

Terminal
File Edit View Terminal Tabs Help
root@server:~/flag # ls -lo
total 0
-rw-r--r--  1 root  wheel  schg 0 Feb 11 18:27 test
root@server:~/flag # █

```

نمایش وضعیت `flag`

همانطوری که مشاهده می کنید در فیلد بعد از مشخص شدن گروه صاحب فایل شما `file flag` اعمال شده به نام `schg` مشاهده می کنید.



FreeBSD در securelevel

هدف اصلی از بیان file flag در FreeBSD بیان کردن قابلیت امنیتی اضافه شده به سیستم عامل های مبتنی بر BSD است به نام security profiles یا همان securelevel ..

هسته سیستم عامل های مبتنی بر BSD به صورتی طراحی شده اند که می توانند حتی دسترسی کاربر root را هم محدود کنند تا سطح امنیتی سیستم شما را افزایش دهند. برای این منظور پنج سطح مختلف از securelevel در FreeBSD در نظر گرفته شده است که هر کدام در ادامه توضیح داده شده است:

سطح -1:

در این سطح هیچ بخش امنیتی بر روی سیستم شما اعمال نمی شود و به صورت پیش فرض هم از این سطح استفاده می شود که هیچ تغییری در سطر هسته بر روی سیستم شما فعال نیست.

سطح 0:

در این سطح اگر کاربری file flag های غیر فایل تغییری به فایلها اضافه کرده باشد می توان در این سطح آنها را خاموش کند. این نوع از flag ها فقط در سطح 0 اعمال می شود و شما اگر سطح 1- را فعال کنید این flag ها غیر فعال می شوند. در این سطح هم همه دستگاه ها قابل خواندن و نوشتن هستند با توجه به سطح دسترسی مشخص شده.

سطح 1:

در این سطح اگر کاربری file flag های غیر فایل تغییری به فایلها اضافه کرده باشد نمی توان در این سطح آنها را خاموش کند.

فایل سیستم های /dev/mem, /dev/kmem و /dev/io اگر در ساختار سخت افزاری شما وجود داشته باشد، در این سطح از امنیت غیر قابل mount کردن است.

یکی از قابلیتهایی که از کاربر در این سطح گرفته می شود این است که دیگر نمی توان در هسته FreeBSD ماژولی با استفاده kldload بارگذاری کرد و یا ماژول های بارگذاری شده را با استفاده از فرمان kldunload از حالت بارگذاری خارج کرد.

سطح 2:

این سطح علاوه بر اینکه قابلیت های سطح 1 را شامل می شود قابلیت های دیگری به صورت زیر در هسته سیستم عامل روش می شود:



در این سطح دیسک ها برای خواندن و نوشتن نمی توانند mount شوند،

فرمان newfs حتی اگر سیستم شما در حالت multi-user محدود شده و دیگر دیسک جدید را شما نمی تواند فرمت کند و دیسک های قبلی را که در سیستم شما هست غیر قابل پاک شدن با استفاده از فرمان newfs است.

این سطح از securelevel بر روی زمان در هسته اعمال می شود و شما نمی توانید زمان هسته را دستکاری کنید.

سطح 3:

این سطح از securelevel به سطح امنیتی شبکه ای معروف است که سطح پایانی در بالا بردن سطح امنیتی سیستم شما که علاوه بر تمام قابلیت های سطح قبلی خود بر روی تنظیمات فایروایل سیستم شما تاثیر می گذارد و فارغ از هر فایروالی که شما استفاده می کنید رولهای آن دیگر قابل تغییر نخواهد بود.

securelevel با استفاده از فرمان init قبل از راه اندازی شدن سیستم در حالت multi-user اجرا و اعمال می شود.

به این نکته توجه کنید که کاربر root یا همان super-user فقط می تواند سطح را افزایش دهد.

کاهش دادن securelevel در زمان راه اندازی شدن سیستم حتی توسط کاربر root هم امکان پذیر نیست.

اگر تغییرات securelevel را بصورت دائمی در سیستم خود اعمال نکرده باشید، که در بخش بعدی روش آن توضیح داده می شود شما با راه اندازی مجدد سیستم می توانید آنرا پایین آورید، ولی اگر به صورت خودکار و در زمان راه اندازی توسط فایل rc.conf تغییر دادن securelevel را اعمال کرده باشید فقط باید در حالت single user mod وارد شوید و خط مربوطه را از فایل rc.conf پاک کنید.

نمایش و تغییر دادن Live سطح securelevel

برای نمایش دادن سطح حاضر سیستم شما می توانید از فرمان sysctl به صورت اجرا شده در شکل زیر استفاده کنید:

```

Terminal
File Edit View Terminal Tabs Help
root@server:~/flag # sysctl kern.securelevel
kern.securelevel: -1
root@server:~/flag #

```

نمایش سطح هسته فعلی سیستم

برای تغییر دادن این سطح کفایت از فرمان زیر استفاده کنید:

```
# sysctl kern.securelevel=2
```



با استفاده از فرمان بالا شما securelevel را به سطح 2 افزایش داده اید، حال در شکل پایین مشاهده می کنید که حتی کاربر root هم نمی تواند سطح securelevel را به سطح پایین تر تغییر دهد:

```

Terminal
File Edit View Terminal Tabs Help
root@server:~/flag # sysctl kern.securelevel=2
kern.securelevel: -1 -> 2
root@server:~/flag # sysctl kern.securelevel=1
kern.securelevel: 2
sysctl: kern.securelevel=1: Operation not permitted
root@server:~/flag # █

```

نمایش عدم تغییر در امنیت هسته توسط کاربر root

تغییر دادن دائمی securelevel

برای انجام دادن این کار کفایت که از فایل rc.conf استفاده کنید و خطوط زیر را در این فایل اضافه کنید:

```
kern_securelevel_enable="YES"
```

```
kern_securelevel="2"
```

با استفاده از این دو خط سطح securelevel به سطح 2 تغییر پیدا می کند.



اشتراک گذاشتن دسترسی root با کاربر su و sudo

در پروژه BSD بسیار تاکید شده است که از کاربر root برای انجام دادن فعالیت های مدیریتی در سیستم استفاده نکنید، در راستای همین هدف و تفکر به شما اجازه ورود کاربر root را از طریق ssh به سیستم را نمی دهد مگر اینکه شما آنرا تنظیم کنید. برای کمک کردن به شما دو راه را در اختیار شما قرار داده است در روش اول استفاده کردن از فرمان su است که یک روش قدیمی است و در روش دوم استفاده کردن از فرمان sudo است. در ادامه با دو روش آشنا می شوید.

فرمان su

در FreeBSD و سایر سیستم عامل های مبتنی بر BSD گروه کاربری به صورت پیش فرض وجود دارد به نام wheel، این گروه یک گروه مهم و کاربردی است. هر کاربری که عضو این گروه باشد می تواند با فرمان su سطح دسترسی خود را به کاربر root ارتقا دهد. برای این کار کاربری که اجازه دارد از این فرمان استفاده کنید باید رمز عبور کاربر root را در اختیار داشته باشد.

برای اضافه کردن کاربر مورد نظر شما به گروه wheel باید از فرمان pw به صورت زیر استفاده کنید:

```
# pw usermod admin -G wheel
```

با استفاده از فرمان بالا گروه admin به گروه wheel اضافه می شود. حال این کاربر می تواند از فرمان su استفاده کند و در صورتی که رمز کاربر root را داشته باشد سطح دسترسی خود را به کاربر root برساند، این عمل در شکل زیر نمایش داده شده است:

```
Terminal
File Edit View Terminal Tabs Help
In order to support national characters for European languages in tools like
less without creating other nationalisation aspects, set the environment
variable LC_ALL to 'en_US.ISO8859-1'.
admin@server:~ % id
uid=1009(admin) gid=1010(admin) groups=1010(admin),0(wheel)
admin@server:~ % su
Password:
root@server:/home/admin # id
uid=0(root) gid=0(wheel) groups=0(wheel),5(operator)
root@server:/home/admin #
```

نمایش وضعیت یک کاربر

فرمان id برای نمایش داده وضعیت کاربر جاری شماست، در زمانی که شما کاربر Admin هستید خروجی فرمان id علاوه بر user ID کاربر شما نمایش می دهد که این کاربر علاوه بر گروه admin عضو گروه wheel هم هست، در نتیجه این کاربر می تواند از فرمان su استفاده کند، بعد از اجرا این فرمان شما وارد خطی می شوید که باید رمز عبور کاربر root را وارد کنید و بعد از وارد کردن صحیح رمز عبور شما وارد سطح دسترسی root می شود و همانطوری که مشاهده می کنید خروجی فرمان id متفاوت با دفعه قبلی می باشد.



نکات فرمان: su

در اجرا کردن فرمان su باید این نکته را بدانید که کاربر root می تواند به سطح دسترسی هر کاربری بدون داشتن رمزعبور آن کاربر برسد کفایت که بعد از فرمان su نام کاربر مورد نظر را وارد کند.

در زمانی که شما از فرمان su استفاده می کنید متغیرهای شما در Shell تغییر نمی کند برای اعمال این تغییر باید از سویچ فرمان استفاده کنید تا به صورت کامل مراحل ورود به سیستم برای شما شبیه سازی شود، این بخش در فرمان زیر نمایش داده شده است.

```
Terminal
File Edit View Terminal Tabs Help
root@server:~ # su admin
admin@server:/root %
```

دسترسی کاربر root برای وارد شدن به هر کاربری با su

در شکل بالا su بدون سویچ اجرا شده است، اما در شکل زیر فرمان su با سویچ اجرا شده است.

```
Terminal
File Edit View Terminal Tabs Help
root@server:~ # su -l admin
Can't remember if you've installed a certain port or not? Try "pkg info
-x port_name".
admin@server:~ %
```

استفاده از سویچ l در su

یکی دیگر از قابلیت های فرمان su این است که به کاربر root این اجازه را می دهد که login class مختلف را تست کند، برای اینکار کفایت که از سویچ c- و نام آن login class استفاده کنید.

فرمان: sudo

روش استفاده از فرمان بالا یک ایراد کلی دارد و آن این است که کاربر به صورت کامل به سطح دسترسی کاربر root ارتقا داده می شود و رمز عبور کاربر root هم در دسترس کاربری قرار می گرد که از فرمان su استفاده می کند، راه دیگری در سیستم عامل های مبتنی بر BSD در نظر گرفته شده است به نام sudo که در بخش بعدی با آن آشنا می شوید.

در بعضی از موارد شما به صورت تیمی کار مدیریت سیستم خود را انجام می دهید، برای این کار دیگر نمی شود از روش su استفاده کنید، افراد گروه شما می تواند دارای دسترسی کاربر معمولی باشند با رمزعبور خود کار مورد نظر را انجام دهد و شما هم به عنوان یک مدیر سیستم می تواند تمرکز بیشتر و بهتری بر روی دسترسی های سیستم خود داشته باشید. در این بخش شما با این فرمان و روش استفاده از آن آشنا می شوید.



برای استفاده کردن از `sudo` ابتدا باید آنرا نصب کنید هم از طریق نصب بسته های باینری این کار را کنید هم از طریق سیستم پورت، اگر از سیستم پورت استفاده می کنید باید وارد شاخه زیر شوید به صورت زیر:

```
# cd /usr/ports/security/sudo/
#make install clean
```

برای نصب کردن از سیستم بسته از فرمان زیر استفاده کنید:

```
# pkg install sudo
```

به دو روش بالا برنامه `sudo` نصب می شود. بعد از نصب برای ویرایش کردن فایل اصلی برنامه `sudo` باید از برنامه `visudo` استفاده کنید، بعد از اجرا کردن این برنامه فایل پیکربندی برنامه به صورت زیر باز می شود:

نمایش فایل پیکربندی `sudo`

مانطوری که مشاهده می کنید این فرمان فایل پیکربندی را با استفاده از ویرایشگر `vi` باز می کند.

نکته

برای راحتی در استفاده از این برنامه به شما دوستان پیشنهاد می کند که فیلم آموزشی روش استفاده از ویرایشگر `vi` را مشاهده کنید



روش استفاده کردن از این فایل پیکربندی بسیار ساده و راحت است، خطوطی که با # شروع می شوند توضیحات هستند و با مطالعه آنها می توانید به راحتی از این فایل استفاده کنید، در ادامه با روش اضافه کردن سطح دسترسی خاص خود را در این فایل آشنا می شوید.

برای شروع ایجاد کردن یک سطح دسترسی کفایت که در ابتدا نام کاربر مورد نظر که قصد دارید به آن اختیار اجرای فرمان را دهید وارد کنید و در سطح بعدی باید دسترسی آنرا بیان کنید به صورت زیر . ALL=(ALL) در بخش بعدی باید فرمانی را که قصد اجرا آنرا دارید بیان کنید خط زیر یک مثالی است که شما به کاربر test اجازه انجام کلیه امور ftpd را به آن می دهید:

```
test ALL=(ALL) /usr/sbin/service ftpd *
```

برای اجرا کردن این فرمان توسط کاربر کفایت که در ابتدا و شروع فرمان از suod استفاده شود توسط کاربر test به صورت زیر:

```
% sudo /usr/sbin/service ftpd start
```



مانیتور کردن وضعیت سیستم با iostat

یکی از بخشهای مهم در مدیریت کردن سرور های خود مانیتور کردن وضعیت I/O سیستم شماست که به شما میزان بار موجود بر روی سیستم شما را معین می کند، در سیستم عامل FreeBSD چندین ابزار برای مشاهده کردن وضعیت سیستم وجود دارد به نام های که به بخش stat ختم می شوند مثل iostat, systat, gstat, vmstat, nfstat و غیره. در این بخش توضیحات کوتاهی در مورد هر یک از فرمان ها برای شما ارایه می شود.

فرمان iostat:

این فرمان وضعیت i/o دیسک های سیستم شما را در لحظه نمایش می دهد و قابلیت هایی دارد که با آن آشنا می شوید، اگر این فرمان را یکبار اجرا کنید برای شما یک خروجی پیشفرض در یک خط نمایش می دهد، این خروجی به صورت زیر است:

```

Terminal
File Edit View Terminal Tabs Help
root@server:~ # iostat
          tty          da0          da1          cd0          cpu
tin tout KB/t tps MB/s KB/t tps MB/s KB/t tps MB/s us ni sy in id
1      35 35.18  1  0.03  6.28  0  0.00  1.69  0  0.00  0  0  0  0 99
root@server:~ #

```

خروجی فرمان iostat

این فرمان برای نمایش وضعیت i/o دستگاهی که به سیستم شما متصل است استفاده می شود و به صورت پیش فرض 5 دستگاه اول سیستم شما را نمایش می دهد مگر اینکه شما دستگاه خاصی را در نظر داشته باشید، طول خروجی این فرمان هم به صورت پیش فرض 80 ستون است. در بخش های بعدی با خروجی این فرمان آشنا می شوید.

هر ستونی که نام دستگاه بر روی آن قرار دارد آخرین ستونی است که اطلاعات آن دستگاه در آن بخش نمایش داده شده است برای مثال بخش tty شامل دو ستون است که tty بر روی آخرین ستون اطلاعات این بخش قرار دارد، در ادامه با خروجی این فرمان بیشتر آشنا می شوید،

بخش اول این فرمان به نمایش وضعیت tty یا همان ترمینال سیستم شما می پردازد که تنها بخشی است که در دو ستون نمایش داده می شود و مقادیری که نمایش می دهد در دو قسمت tin و tout است:

بخش tin: تعداد کاراکترهای خوانده شده از ترمینال را نمایش می دهد.

بخش tout: تعداد کاراکترهای نوشته شده در ترمینال را نمایش می دهد.



در بخش بعدی که نام دستگاه‌هایی که i/o هستند را نمایش می‌دهد اگر از فرمان `iostat` به صورت پیش فرض استفاده کنید این خروجی در سه بخش به صورت زیر تقسیم بندی می‌شود:

KB/t به معنی kilobytes per transfer

tps به معنی transfers per second

MB/s به معنی megabytes per second

این خروجی می‌تواند حالت‌های مختلفی به خود بگیرد مثلاً اگر از سوئیچ `|` حرف `i` بزرگ استفاده کنید خروجی این فرمان اطلاعات زیر را نمایش می‌دهد:

KB/t به معنی kilobytes per transfer

xfrs به معنی total number of transfers

MB به معنی total number of megabytes transferred

وضعیت نمایش CPU سیستم شما هم در بخش آخر از این فرمان نمایش داده می‌شود که با بخش‌های دیگر متفاوت بوده و خروجی آن شامل اطلاعات زیر می‌شود:

us به معنی % of cpu time in user mode

ni به معنی % of cpu time in user mode running niced processes

sy به معنی % of cpu time in system mode

in به معنی % of cpu time in interrupt mode

id به معنی % of cpu time in idle mode

مدیریت کردن خروجی فرمان `iostat`:

شما با استفاده کردن از سوئیچ‌های مختلف می‌توانید حالت‌ها خروجی این برنامه را متفاوت کنید در ادامه با سوئیچ‌های مورد نیاز در این فرمان آشنا می‌شوید:

سوئیچ : -C

به صورت پیش فرض زمانی که شما فرمان `iostat` را اجرا می‌کنید فقط یک خط خروجی برای شما نمایش می‌دهد، برای اینکه به تعداد دلخواه خود بتوانید خروجی را مشاهده کنید بعد از سوئیچ `C` تعداد وارد کنید برای مثال در شکل زیر مشاهده می‌کنید که 5 بار این خروجی برای شما نمایش داده شود:



```

Terminal
File Edit View Terminal Tabs Help
root@server:~ # iostat -C x
          tty          cpu
tin tout us ni sy in id
  1   31  0  0  0  0 99
root@server:~ # █

```

نمایش خروجی -C در فرمان iostat

سوئیچ -d

برای حذف کردن دو بخش tty و cpu و نمایش فقط وضعیت دستگاه های متصل به سیستم شما باید از سوئیچ d استفاده کنید به صورت زیر:

```

Terminal
File Edit View Terminal Tabs Help
root@server:~ # iostat -d
          da0          da1          cd0          pass0
KB/t tps MB/s  KB/t tps MB/s  KB/t tps MB/s  KB/t tps MB/s
12.80  3  0.03  6.11  0  0.00  1.69  0  0.00  0.00  0  0.00
root@server:~ # █

```

نمایش خروجی -d در فرمان iostat

سوئیچ : n

برای محدود کردن تعداد 5 عدد دستگاه نمایش داده شده به تعداد دلخواه شما از سوئیچ n و بعد تعداد مورد نظر خود استفاده کنید.

سوئیچ : t

شما می توانید خروجی سیستم خود را به تناسب نوع دستگاه ها طبقه بندی کنید، برای مثال فقط برای نمایش داده دستگاه هایی که از نوع SCSI هستند بعد از t- باید نوع SCSI را تایپ کنید در زیر لیستی از نوع هایی این فرمان را مشاهده می کنید:

```

da          Direct Access devices

sa          Sequential Access  devices

printer    Printers

proc       Processor  devices

```



worm	Write Once Read Multiple devices
cd	CD devices
scanner	Scanner devices
optical	Optical Memory devices
changer	Medium Changer devices
comm	Communication devices
array	Storage Array devices
floppy	Floppy devices

از نظر نوع رابط می توانید از کلمات زیر استفاده کنید:

IDE	Integrated Drive Electronics devices
SCSI	Small Computer System Interface devices
other	Any other device interface

در شکل زیر روش استفاده از این فرمان را مشاهده می کنید برای حذف کردن `cpu` و `tty` از سوئیچ `-d` استفاده کنید:

```

Terminal
File Edit View Terminal Tabs Help
root@server:~ # iostat -d -t IDE
          cd0
  KB/t tps MB/s
  1.69  0  0.00
root@server:~ # iostat -d -t SCSI
          da0          da1
  KB/t tps MB/s  KB/t tps MB/s
  12.83  3  0.03  6.11  0  0.00
root@server:~ # █

```

خروجی های خاص فرمان `iostat`



همانطوری که در خروجی فرمان بالا مشاهده می کنید با استفاده از سویچ `t` می توانید خروجی `iostat` را بر اساس نوع طبقه بندی کنید.

سویچ `-W`

برای تعیین کردن زمان مکث در بین هر خط از خروجی باید از سویچ `-W` و مقدار مکث به ثانیه تعیین کنید، این سویچ باید با سویچ `C` برای نمایش تعداد خروجی ها استفاده شود.

سویچ `-X`

برای نمایش هرچه بهتر وضعیت دیسک ها در حالت گسترده از سویچ `X` استفاده کنید، در صورتی که تعداد دیسک های سیستم شما زیاد باشد و همه در خروجی فرمان قرار نگیرند بهترین گزینه استفاده کرده از این مدل خروجی است که هر دیسک در یک خط افقی قرار می گیرد، نمایش این خروجی را در شکل زیر مشاهده می کنید:

```

Terminal
File Edit View Terminal Tabs Help
root@server:~ # iostat -x
extended device statistics
device      r/s    w/s    kr/s    kw/s qlen svc_t  %b
da0         2.1    0.4   18.4    13.1    0    3.8    0
da1         0.0    0.0    0.1     0.0    0    1.0    0
cd0         0.0    0.0    0.0     0.0    0    0.4    0
pass0       0.0    0.0    0.0     0.0    0    0.0    0
pass1       0.0    0.0    0.0     0.0    0    0.0    0
pass2       0.0    0.0    0.0     0.0    0    0.0    0
root@server:~ #

```

سویچ `X` در فرمان `iostat`



فرمان systat در FreeBSD

در سیستم عامل FreeBSD چندین فرمان برای نمایش وضعیت سیستم وجود دارد که فرمان پایه ای آن `iostat` بود که در مقاله قبلی با آن آشنا شدید، فرمان دیگری را در این بخش برای شما معرفی می کنم که به صورت کامل تمام وضعیت ها و فرمان های `stat` را پوشش می دهد، اسم این فرمان `systat` است که در این مقاله روش استفاده کردن از این فرمان را آموزش خواهیم دید.

این مقاله شامل بخشهای زیر است:

- بخش `pigs` برای نمایش میزان مصرف پردازش هر کاربر.
- بخش `ifstat` نمایش وضعیت کارت های شبکه سرور شما.
- بخش `icmp` برای نمایش وضعیت `icmp` سرور شما.
- بخش `ip` نمایش وضعیت پروتکل `ip` سرور شما.
- بخش `TCP` برای نمایش وضعیت `TCP` سرور شما.
- بخش `iostat` برای نمایش وضعیت `IO` سرور شما.
- بخش `swap` برای نمایش وضعیت `swap` در سرور شما.
- بخش `netstat` برای نمایش وضعیت پورتهای باز و ارتباطات باز سرور شما.

زمانی که این فرمان را اجرا می کنید شما وارد صفحه اصلی این برنامه می شوید که به دو بخش تقسیم شده است، خروجی این فرمان را در شکل زیر مشاهده می کنید:

```

Terminal
File Edit View Terminal Tabs Help
Load Average /0 /1 /2 /3 /4 /5 /6 /7 /8 /9 /10
              |||||
root          /0% /10 /20 /30 /40 /50 /60 /70 /80 /90 /100
root         idle XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
root         idle XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
root         Xorg X

```

خروجی فرمان `systat` بدون هیچ فرمان اضافه ای

این فرمان از تکنولوژی `ncurses` برای طبقه بندی کردن صفحه نمایش استفاده می کند، در بخش بالایی این خروجی `load` سیستم به صورت `bar` برای شما نمایش داده می شود و در بخش پایینی این فرمان شما می توانید اطلاعات انتخابی مورد نظر خودتون را برای نمایش انتخاب کنید.



برای وارد شدن به بخش اطلاعاتی که این فرمان می تواند برای شما نمایش دهد کفایت که از علامت : استفاده کنید و بعد کلمه **help** را وارد کنید و به محض اجرا **help** با استفاده از کلید **Enter** در قسمت پایینی این فرمان خطی به صورت زیر برای شما نمایش داده می شود:

```
pigs swap iostat vmstat netstat icmp ip icmp6 ip6 sctp tcp ifstat
```

بخشهای قابل فعال در **systat**

12 بخش قابل نمایش برای شما در این فرمان وجود دارد که شما می توانید به تناسب نیاز خود هر کدام را اجرا کنید و از اطلاعات خروجی آن استفاده کنید، در ادامه با بخش های اجرای مختلف این فرمان آشنا می شوید.

قبل از شروع:

قبل از شروع کار با این فرمان باید با گزینه های موجود در این فرمان آشنا شوید. برای وارد شدن به محیط فرمان این برنامه باید از کلید : استفاده کنید، بعد از اجرا : در زیر این صفحه شما علامت : را مشاهده می کنید که فرمان های بعدی را در این بخش باید وارد کنید، بعد از اجرا فرمان شما وارد برنامه می شوید، برای خارج شدن از این برنامه کفایت که ابتدا وارد بخش فرمان شوید و بعد حرف **q** را وارد کنید و بعد **Enter** کنید.

این برنامه به صورت پیش فرض هر 5 ثانیه اطلاعات نمایش را برای شما **refresh** می کند، برای تغییر دادن این مقدار کفایت که در حالت فرمان وارد شوید و بعد عددی را وارد کنید که به ثانیه است و **Refresh** کردن پیش فرض را تغییر می دهد. بعد از وارد کردن عدد مورد نظر در زیر صفحه برنامه خط زیر را مشاهده می کنید:

```
Showing pigs, refresh every 1 seconds.
```

تغییر دادن زمان **Refresh**

این خط به شما اعلام می کند که بخش **pigs** هر 1 ثانیه **refresh** می شود.

برای متوقف کردن **refresh** کفایت که در بخش فرمان **stop** را تایپ کنید و برای راه اندازی مجدد آن کفایت که از فرمان **start** و عددی به ثانیه برای **refresh** کردن وارد کنید.

برای نمایش کردن **load average** سیستم خود هم کفایت که فرمان **load** را وارد کنید تا به صورت شکل زیر شما **load average** سیستم خود را در بخش فرمان مشاهده کنید:

```
0.3 0.6 0.6
```

نمایش **load average**



برای نمایش همه فرمان‌ها در این بخش باید از فرمان `help` استفاده کنید بعد از وارد شدن به بخش فرمان گزینه `help` را وارد کنید تا اطلاعات به صورت زیر برای شما نمایش داده شود:

```
pigs swap iostat vmstat netstat icmp ip icmp6 ip6 sctp tcp ifstat
```

بخشهای قابل فعال در `sysstat`

برای اجرا کردن هر کدام از این بخش‌ها که برای شما سیستم مانیتورینگ جداگانه‌ای باز می‌کند هم می‌توانید به صورت مستقیم آنرا وارد کنید و یا بعد از - آنرا تایپ کنید.

بخش pigs

این بخش به صورت پیش فرض در زمان راه‌اندازی فرمان اجرا می‌شود، هدف اصلی این برنامه نمایش دادن پردازش‌هایی است که در حافظه فعال قرار دارد، این بخش فرمان‌های `idle` را نمایش نمی‌دهد.

بخش ifstat

در بسیاری از موارد شما نیاز دارید که وضعیت `send` و `receive` کارت‌های شبکه سیستم خود را مشاهده کنید، برای اجرا کردن و نمایش خروجی این بخش کافیست که بعد از : گزینه `ifstat` را وارد کنید و بعد `Enter` کرده تا خروجی فرمان در بخش دوم به صورت زیر برای شما نمایش داده شود:

```

Terminal
File Edit View Terminal Tabs Help
Load Average  /0 /1 /2 /3 /4 /5 /6 /7 /8 /9 /10
|
Interface      Traffic      Peak      Total
lo0  in    0.000 KB/s  0.000 KB/s  3.444 MB
      out    0.000 KB/s  0.000 KB/s  3.444 MB
em1  in    0.000 KB/s  0.000 KB/s  1.615 MB
      out    0.000 KB/s  0.000 KB/s 14.788 MB
em0  in    0.000 KB/s  0.000 KB/s 43.907 MB
      out    0.000 KB/s  0.000 KB/s  2.449 MB
Showing ifstat, refresh every 5 seconds.

```

نمایش `ifstat` در `sysstat`



همانطوری که مشاهده می کنید در این سیستم 3 کارت شبکه وجود دارد به نام های lo0 و em0 و em1 ، در بخش های بعد از نام کارت های شبکه میزان ترافیک خارج شده و پیک هر کارت شبکه را مشاهده می کنید و در بخش total شما با مجموع بسته های ارسال شده و دریافت شده را نمایش می دهد.

هر برنامه ای در این بخش شامل زیر فرمان هایی است که رفتار برنامه اجرا شده را تغییر می دهد، برای بخش ifstat زیر فرمانی وجود دارد به نام scale که به شما این امکان را می دهد که بتوانید مقادیر نمایش داده شده را در قالبهای kbit kbyte mbit و gbit .. تغییر دهید. در شکل زیر بعد از اجرا کردن scale mbyte مقدار از کیلو بایت به مگابایت تغییر می کند:

```

Terminal
File Edit View Terminal Tabs Help
Load Average  /0 /1 /2 /3 /4 /5 /6 /7 /8 /9 /10
              |||

Interface      Traffic          Peak            Total
lo0  in         0.000 MB/s      0.252 MB/s     3.460 MB
     out        0.000 MB/s      0.252 MB/s     3.460 MB

em1  in         0.000 MB/s      0.000 MB/s     1.616 MB
     out        0.000 MB/s      0.000 MB/s    14.788 MB

em0  in         0.000 MB/s      0.984 MB/s     44.337 MB
     out        0.000 MB/s      0.779 MB/s     2.760 MB

:scale mbyte

```

تغییر دادن مقیاس در ifstat

بخش: icmp

برای مشاهده کردن اطلاعات پروتکل icmp در این بخش فرمان icmp را در بخش فرمان وارد کنید تا قسمت زیر این فرمان به صورت زیر تغییر می کند:



```

Terminal
File Edit View Terminal Tabs Help
Load Average /0 /1 /2 /3 /4 /5 /6 /7 /8 /9 /10
|||

Interface Traffic Peak Total
lo0 in 0.000 MB/s 0.252 MB/s 3.460 MB
   out 0.000 MB/s 0.252 MB/s 3.460 MB

em1 in 0.000 MB/s 0.000 MB/s 1.616 MB
   out 0.000 MB/s 0.000 MB/s 14.788 MB

em0 in 0.000 MB/s 0.984 MB/s 44.337 MB
   out 0.000 MB/s 0.779 MB/s 2.760 MB

:scale mbyte

```

وارد شدن برای تغییر نمایش وضعیت `icmp`

این بخش همانطوری که مشاهده می کنید به دو بخش در دو ستون تقسیم می شود که در قسمت سمت راست بسته هایی که وارد می شود را نمایش می دهد و در بخش سمت راست بسته هایی که خارج می شوند را در بخش های مختلف نمایش می دهد.

زیر مجموعه این فرمان خود در دو حالت `mode` و `reset` است.

در بخش `mode`

شما می توانید به چهار حالت زیر وضعیت نمایش را تغییر می دهد:

حالت `rate`

این بخش که پیش فرض این فرمان است نرخ تغییر هر بسته را در واحد زمان که به صورت پیش فرض یک ثانیه است نمایش می دهد.

حالت `delta`

این حالت هم مثل حالت بالاست.

حالت `since`

شما در بخش بعدی یاد می گیرید که حالت `reset` را اعمال کنید که شمارنده های این بخش صفر شود، برای نمایش هر بسته که وارد را خارج می شود می توانید از این حالت استفاده کنید.



حالت absolute

این حالت همه اطلاعات را نمایش می دهد که تفاوت این دو حالت rate و absolute را در شکل زیر مشاهده می کند، شکل زیر حالت absolute است:

```

Terminal
File Edit View Terminal Tabs Help
Load Average  /0 /1 /2 /3 /4 /5 /6 /7 /8 /9 /10
              |||

      ICMP Input                                ICMP Output
4711 total messages                            1568 total messages
  0 with bad code                               129 errors generated
  0 with bad length                             0 suppressed - original too short
  0 with bad checksum                           0 suppressed - original was ICMP
  0 with insufficient data                      1439 responses sent
                                              0 suppressed - multicast echo
                                              0 suppressed - multicast tstamp

      Input Histogram                            Output Histogram
3191 echo response                             1439 echo response
1439 echo request                               0 echo request
  81 destination unreachable                   129 destination unreachable
  0 redirect                                   0 redirect
  0 time-to-live exceeded                      0 time-to-live exceeded
  0 parameter problem                          0 parameter problem
  0 router advertisement                       0 router solicitation

:mode absolute

```

حالت absolute

حالت reset هم همه این اعداد را صفر می کند.

برای نمایش دادن وضعیت icmp در ورژن 6 کفایت که از فرمان icmp6 به جای icmp استفاده کنید و همه بخشها و حالت های بالا هم در آن دو مشترک هستند.

حالت ip و ip6

برای نمایش وضعیت بسته های ip بر روی سیستم شما از فرمان های ip و ip6 استفاده کنید که در شکل زیر حالت خروجی این فرمان را مشاهده می کنید:



```

Terminal
File Edit View Terminal Tabs Help

Load Average  /0 /1 /2 /3 /4 /5 /6 /7 /8 /9 /10
|

IP Input
85121 total packets received
0 - with bad checksums
0 - too short for header
0 - too short for data
0 - with invalid hlen
0 - with invalid length
0 - with invalid version
0 - jumbograms
0 total fragments received
0 - fragments dropped
0 - fragments timed out
0 - packets reassembled ok
133 packets forwarded
302 - unreachable dests
0 - redirects generated
0 option errors
0 unwanted multicasts
82923 delivered to upper layer

IP Output
73076 total packets sent
72943 - generated locally
0 - output drops
0 output fragments generated
0 - fragmentation failed
7 destinations unreachable
1752 packets output via raw IP

UDP Statistics
10979 total input packets
0 - too short for header
0 - invalid checksum
0 - no checksum
0 - invalid length
131 - no socket for dest port
4111 - no socket for broadcast
0 - socket buffer full
7369 total output packets

:mode absolute

```

نمایش وضعیت ip

این فرمان هم مثل بخش icmp که توضیح داده شده هم شامل بخش mode است که خود شامل 4 زیر گروه rate delta و since absolute است و هم شما می توانید اطلاعات را به rest به مقادیر 0 ببرید.

بخش tcp:

برای مشاهده کردن بسته های tcp منتقل شده بروی سیستم، باید وارد بخش tcp در بخش فرمان برنامه systat شوید. برای این کار کافیست که از فرمان tcp استفاده کنید تا اطلاعات به صورت شکل زیر برای شما نمایش داده شود:



```

Terminal
File Edit View Terminal Tabs Help
Load Average  /0 /1 /2 /3 /4 /5 /6 /7 /8 /9 /10
|

TCP Connections
2770 connections initiated
1838 connections accepted
4495 connections established
 90 connections dropped
 78 - in embryonic state
 26 - on retransmit timeout
  3 - by keepalive
  0 - from listen queue

TCP Timers
21098 potential rtt updates
24092 - successful
 2777 delayed acks sent
  896 retransmit timeouts
    0 persist timeouts
  292 keepalive probes
  312 - timeouts

TCP Packets
61916 total packets sent
25251 - data
  4 - data (retransmit by dupack)
  0 - data (retransmit by sack)
29471 - ack-only
  0 - window probes
 12 - window updates
  0 - urgent data only
 7225 - control
  0 - resends by PMTU discovery
69319 total packets received
41828 - in sequence
 1372 - completely duplicate
  0 - with some duplicate data
  0 - out-of-order
 6142 - duplicate acks
24741 - acks
  0 - window probes
 486 - window updates

Showing tcp, refresh every 1 seconds.

```

نمایش وضعیت tcp در sysstat

این فرمان هم مثل بخش icmp که توضیح داده شده هم شامل بخش mode است که خود شامل 4 زیر گروه rate delta و since absolute است و هم شما می توانید اطلاعات را به rest به مقادیر 0 ببرید.

بخش iostat

در مقاله قبلی در سایت www.mabedini.ir شما با فرمان isostat آشنا شده اید، در این بخش هم شما می توانید اطلاعات فرمان isostat را در قالبی قابل خواندن به صورت زیر مشاهده کنید:



```

Terminal
File Edit View Terminal Tabs Help
Load Average /0 /1 /2 /3 /4 /5 /6 /7 /8 /9 /10
|
cpu user|
  nice|
  system|
interrupt|
idle|XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
da0 MB/s
  tps|
da1 MB/s
  tps|
cd0 MB/s
  tps|
pass0 MB/s
  tps|
pass1 MB/s
  tps|
Showing iostat, refresh every 1 seconds.

```

نمایش وضعیت iostat در systat

بخش: swap

برای مشاهده کردن وضعیت swap سیستم خود از فرمان swap در بخش فرمان این برنامه استفاده کنید تا خروجی این فرمان را به صورت زیر مشاهده کنید:

```

Terminal
File Edit View Terminal Tabs Help
Load Average /0 /1 /2 /3 /4 /5 /6 /7 /8 /9 /10
|
Disk 1K-blocks Used /0% /10 /20 /30 /40 /50 /60 /70 /80 /90 /100
da0p3 1048412 0 X
Showing swap, refresh every 1 seconds.

```

نمایش وضعیت Swap در systat

این بخش به صورت بار نمایش داده می شود که البته به این دلیلی که این حافظه بر روی سیستم من استفاده نمی شود وضعیت در بخش 0% قرار داد.

بخش **netstat**

شاید برای شما اتفاق افتاده باشد که خروجی فرمان **netstat** را دوست نداشته باشید و نتوانید اطلاعات مفیدی را از آن بدست آورید، در ضمن خروجی این فرمان **netstat** به صورت داینامیک تغییر نمی کند و شما نمی توانید در لحظه وضعیت ارتباط های شبکه ای خود را مشاهده کنید، برای رها شدن از این مشکل در برنامه **systat** شما بخشی دارید که این مشکل را برای شما مرتفع کرده است به نام **netstat** که با اجرا کردن این فرمان رفتار برنامه **systat** به صورت زیر تغییر می کند:

```

Terminal
File Edit View Terminal Tabs Help

Load Average  /0 /1 /2 /3 /4 /5 /6 /7 /8 /9 /10
              ||||

Local Address      Foreign Address    Proto Recv-Q Send-Q (state)
server.ntp         *.*               udp4   0     0
localhost.ntp     *.*               udp6   0     0
fe80:3::1.ntp     *.*               udp6   0     0
localhost.ntp     *.*               udp4   0     0
localhost.45090   localhost.30083   udp6   0     0
192.168.88.129.33146 send.mx.cdnetwor.ntp udp4   0     0
192.168.88.129.22607 ntp.tums.ac.ir.ntp  udp4   0     0
192.168.88.129.10210 ntp.iranet.ir.ntp  udp4   0     0
192.168.88.129.41095 d.resolvers.leve.domai tcp4   0     0 ESTABLISHED

Showing netstat, refresh every 1 seconds.

```

نمایش وضعیت **netstat** در **iostat**

به صورت پیش فرض فقط این برنامه ارتباطات فعال را نمایش می دهد و و حالت **waiting** را نمایش نمی دهد، در نمایش پیش فرض این بخش شما اطلاعات را در قالب **host:prts** مشاهده می کنید. شما می توانید با استفاده کردن از بخشهای زیر مجموعه فرمان **systat** به تغییر رفتار و حالت های برنامه بپردازید.

فرمان **all**

برای نمایش همه وضعیت ارتباطی سیستم خود از این فرمان استفاده کنید تا بخش های **waiting** هم نمایش داده شود.

فرمان **number**

برای اینکه روش نمایش را به صورت عددی تغییر دهید از این فرمان استفاده کنید تا به جای نام سرور ها از آدرس **ip** و شماره های **ports** ها برای نمایش استفاده شود.



کار با inetd در FreeBSD

در FreeBSD برای راه اندازی کردن سرویس ها برنامه ای وجود دارد به نام inetd که در به آن Super server هم گفته می شود، به این دلیل به inetd به اصطلاح super server گفته می شود چون شما با راه اندازی آن می توانید چندین سرور را در قالب یک سرور راه اندازی کنید. زمانی که یک درخواست به این سرور می رسد، این برنامه تعیین می کند که برای درخواست باید کدام سرور را راه اندازی کند. این برنامه مثل یک وکیل بین درخواست و سرور قرار میگیرد و بعد از دریافت درخواست برای برقراری ارتباط بین سرور و کلاینت یک سوکت ایجاد می کند. در حقیقت inetd به تکثیر کردن سایر daemons می پردازد، اما از چند پروتکل ساده در درون خود این برنامه پیاده سازی شده است مثل time echo day time.

بخشهای این مقاله عبارتند از:

- فایل پیکربندی برنامه inetd.
- بخش های مختلف فایل پیکربندی.

فایل پیکربندی برنامه inetd

برای کار کردن با این برنامه شما باید فایل پیکربندی این برنامه را ویرایش کنید مسیر این فایل در زیر شاخه /etc و نام فایل آن inetd.conf است. هر خط در این فایل برای راه اندازی کردن یک سرویس است، خطوطی که با علامت # شروع می شوند اجرا نمی شود و برای اجرا کردن هر برنامه کافیست که علامت # ابتدای خط مورد نظر حذف کنید و سرویس inetd را دوباره راه اندازی کنید. به دو روش می توانید سرور inetd را راه اندازی کنید، برای هر دو روش باید در داخل فایل rc.conf خط زیر را برای راه اندازی خودکار سیستم اضافه کنید:

```
inetd_enable="YES"
```

حال می توانید با استفاده از فرمان Service سرور inetd را راه اندازی کنید برای این کار کافیست که فرمان زیر را اجرا کنید:

```
# service inetd start
```

شما از طریق فرمان های موجود در زیر شاخه /etc/rc.d هم می توانید inetd را به صورت زیر اجرا کنید:

```
# /etc/rc.d/inetd start
```

اگر قصد دارید که برای یکبار فقط این سرویس را راه اندازی کنید می توانید از فرمان زیر به اصطلاح onestart را انجام دهید و در فایل rc.conf خطی اضافه کنید:

```
# /etc/rc.d/inetd onestart
```

در بسیاری از موارد اتفاق می افتد که شما تغییری در فایل پیکربندی inetd اعمال کرده اید برای فقط بارگذاری فایل پیکربندی و قطع نشدن ارتباط قبلی فقط کافیست که از reload به صورت زیر استفاده کنید:



```
# service inetd reload
```

در خط زیر یک مثال از فایل پیکربندی برای راه اندازی کردن سرور ftp را مشاهده می کنید:

```
ftp      stream tcp      nowait  root    /usr/libexec/ftpd      ftpd -l
```

این خط شامل اطلاعاتی است که در ادامه بیشتر با آنها آشنا می شوید، کل اطلاعات این بخش به صورت زیر است:

```
service-name
socket-type
protocol
{wait|nowait}[/max-child[/max-connections-per-ip-per-minute[/max-child-per-ip]]]
user[:group][[/login-class]]
server-program
server-program-arguments
```

بخش service-name

بخش اول از خط مورد نظر شما با نام سرویس آغاز می شود این نام ها باید مطابق با فایل `/etc/services` باشد، که از این فایل شماره پورت مورد نظر جهت ارائه دادن سرویس مشخص می شود. اگر شما از سرویسی استفاده می کنید که در این فایل وجود ندارد در بخش اول باید آنرا در این فایل اضافه کنید.

بخش socket-type

این بخش شامل گزینه های `stream`, `dgram`, `raw`, `seqpacket` که برای ارتباطهای `tcp` باید از `stream` و برای ارتباط از نوع `udp` باید از `dgram` استفاده کنید.

بخش protocol

در این بخش شما برای تنظیم کردن نوع ارتباط `tcp` و `udp` در دو حالت ورژن 4 و 6 باید از کلید های زیر استفاده کنید:

کلید	نما استفاده شده
TCP IPv4	tcp or tcp4
UDP IPv4	udp or udp4
TCP IPv6	tcp6
UDP IPv6	udp6
Both TCP IPv4 and IPv6	tcp46



کلید

نما استفاده شده

Both UDP IPv4 and IPv6

udp46

در یک خط هم شما می توانید دو ورژن 4 و 6 را با هم مشخص کنید.

بخش: **wait | nowait**

این دو گزینه چگونگی مدیریت کردن سوکت توسط برنامه را مشخص می کند، اتباطات از نوع **udp** باید از حالت **wait** استفاده کنند و ارتباط **tcp** که **multi-threaded** هستند باید از حالت **nowait** استفاده کنند، این امر به این دلیل است که حالت **wait** برای یک پردازش چندین سوکت را باز نمی کند و حالت **nowait** می تواند هم زمان از چندین سوکت استفاده کند .

بخش **/max-child**

این بخش دقیقا به از حالت **nowait** استفاده شده و با / از آن جدا می شود و تعداد **child daemons** که **inetd** اجازه دارد برای هر پردازش باز کند را مشخص می کند، برای محدود کردن 10 زیر پردازش باید از عدد 10 استفاده کنید و 0 به معنی بی نهایت است.

بخش **max-connections-per-ip-per-minute/**

این بخش تعداد ارتباطات برقرار شده از یک آدرس **ip** خاص در مدت یک دقیقه را مشخص می کند، اگر تعداد درخواست ها بیشتر از مقدار مورد نظر باشد اتصال برقرار نشده تا مدت زمان محدودیت به اتمام برسد. این امر باعث کاهش مصرف منابع سیستم می شود و در مواردی هم از حمله **ddos** جلوگیری می کند.

بخش **user**

در این بخش می توانید سطح دسترسی کاربری را که می خواهید سرور شما با آن سطح دسترسی کاربر راه اندازی شود را مشخص کنید. این بخش در صورتی که سرور شما هک شود مفید بود و هکر در صورت **shell** گرفتن از سرور با سطح دسترسی کاربر معمولی به سیستم وارد می شود. معروف ترین این بخش ها **daemon** و **nobody** است.

بخش **Server-program**

در این بخش شما مسیر کامل برنامه ای که به عنوان سرور باید توسط **inetd** راه اندازی شود را مشخص کنید.

بخش **Server-program-arguments**

بعضی از سور ها دارای **flags** های خاصی هستند که باعث ایجاد تغییرات در نوع و رفتار سرور مورد نظر شما می شود، با استفاده از این بخش شما می توانید تنظیمات خاص خود را اعمال کنید.



فایروال از نوع host-based firewall در FreeBSD

در FreeBSD علاوه بر فایروال های معروف مثل IPFW و PF نوع دیگری از مدیریت کردن درخواست های ارسالی سمت سرویس ها مثل ftp ssh و غیره وجود دارد به نام host-based firewall که این نوع از فایروال بر روی سیستم محلی راه اندازی می شود و به درخواست هایی که سمت سرور ارسال می شود نظارت می کند، در FreeBSD این نوع از فایروال از تکنولوژی TCP wrappers استفاده می کند که خود این برنامه از طریق inetd راه اندازی می شود. قبل از راه اندازی این سرویس مقاله مربوط به کار با inetd در FreeBSD را مطالعه کنید. در بخش اول از این مقاله با فرمان inetd آشنا شوید که باعث راه اندازی tcp wrapper می شود و در بخش دوم با TCP wrappers آشنا می شوید و در قسمت آخر با فایل hostd_access آشنا می شوید.

بخش ها:

- فرمان inetd.
- آشنایی با TCP wrappers
- فایل hostd_access

فرمان inetd

به صورت پیش فرض برنامه inetd با سویچ های پیش فرض 60 -C -w اجرا می شود که این سویچ ها باعث اجرا شدن TCP wrappers می شود. این سویچ باعث راه اندازی TCP wrappers برای همه سرویس های راه اندازی شده توسط inetd می شود و به صورت پیش فرض از ارتباط بیش از 60 درخواست از سمت یک آدرس IP در یک دقیقه جلوگیری می کند. برای تغییر دادن باید به رفتار inetd خود یک سری flags اضافه کنید که این امر از طریق inetd_flags در فایل etc/rc.conf ممکن می شود.

سویچ -c maximum

به صورت پیش فرض تعداد فراخوانی های همزمان از یک سرویس را تعیین می کند که محدود نشده و بی نهایت است. این بخش ممکن است با استفاده از max-child باطل شود.

سویچ -C rate

این بخش مشخص می کند که یک سرویس در واحد زمان از یک آدرس IP خاص چندبار فراخوانی شود. این بخش با استفاده از سویچ max-connections-per-ip-per-minute می تواند باطل شود.

سویچ -R rate

این بخش بیشترین تعدادی که یک سرویس می تواند در واحد زمانی طلب شود را تعیین می کنید که مقدار پیش فرض آن 256 است و مقدار 0 به معنای بی نهایت است.



سویچ maximum-s-

این بخش تعداد بیشینه که یک سرور در واحد زمانی می تواند از یک آدرس IP خاص در خواست داشته باشد را تعیین می کند و این بخش می تواند با مقدار max-child-per-ip در فایل inetd.conf قابل تغییر است.

کار با TCP Wrapper

برنامه TCP Wrapper یک فایروال تحت سیستم محلی است که با استفاده از برنامه inetd راه اندازی می شود. با استفاده از این برنامه قابلیت گزارش گیری به برنامه اضافه می شود، برنامه های اجرا شده تحت این سیستم می توانند پیام مناسب به سمت کاربر بازگردانی کند و می تواند دسترسی به سرور را محدود کند. برای دریافت اطلاعات بیشتر به tcpd مراجعه کنید. از این برنامه نمی توانید به عنوان فایروال استفاده کنید و فایروال را حذف کنید، بلکه باید از آن در کنای سایر سیستم های مدیریت ارتباطی و فایروال استفاده کنید.

راه اندازی ابتدایی:

همانطوری که در بخش قبلی از این مقاله به آن اشاره شده است، برای راه اندازی کردن TCP Wrapper باید از طریق inetd اقدام کنید. در نتیجه برای راه اندازی باید خط زیر را در فایل /etc/rc.conf اضافه کنید:

```
inetd_enable="YES"
inetd_flags="-Ww"
```

حال در بخش بعدی باید فایل به نام /etc/hosts.allow را ویرایش کنید که در ادامه این فایل است که مدیریت دسترسی ها را انجام می دهد.

نکته:

برخلاف سایر پیاده سازی های TCP Wrapper که از فایل hosts.deny استفاده می کند. این برنامه در FreeBSD از فایل hosts.allow استفاده می کند.

ابتدایی ترین بخش مدیریت در TCP Wrapper این است که دسترسی به یک سرویس را محدود و یا اجازه دهد. این مدل از سطح دسترسی بر مبنای فایل hosts.allow است. به صورت پیش فرض در FreeBSD همه سرویس ها allow هستند.

در سطح مقدماتی از پیکربندی، این فایل شامل خطوطی به صورت زیر است:

daemon : address : action,



بخش daemon نام سروری است که باید برنامه inetd راه اندازی شود که مقادیر آن در فایل inetd قرار دارد. در بخش دوم نام یا آدرس IP سیستمی است که شما قصد دارید دسترسی آنرا مورد بررسی قرار دهید. این بخش هم می تواند IPV4 باشد هم IPV6، باید آدرس IPV6 را داخل براکت قرار دهید. در بخش action دو حال وجود دارد allow و deny.

برنامه TCP Wrapper از قاعده first rule match برای جستجو کردن در فایل پیکربندی خود استفاده می کند و اول خطی که با شرایط مطابقت داشته باشد را می پذیرد و دیگر به جستجو نمی پردازد در نتیجه به قواعدی که در این فایل قرار می دهید دقت کنید.

در خط زیر یک مثالی را مشاهده می کنید که به سرویس pop3 که از طریق inetd راه اندازی می شود در خط زیر اجازه داده می شود که همه کامپیوتر های موجود در شبکه به آن دسترسی داشته باشند، این برنامه qpopper است:

```
# This line is required for POP3 connections:
qpopper : ALL : allow
```

در مرحله بعدی برای اعمال شدن تغییرات باید برنامه inetd را دوباره راه اندازی کنید به صورت زیر:

```
# service inetd restart
```



مقدمه بر شبکه بی سیم در FreeBSD

بیشتر شبکه های بیسیم مبتنی بر استاندارد 802.11 است. این استاندارد در دو حالت infrastructure و ad-hoc network راه اندازی می شود که در حالت infrastructure شبکه مبتنی بر یک Access point راه اندازی می شود و همه سیستم ها بسته ها اطلاعات خود را به سمت AP ارسال کرده و این AP است که به مدیریت کردن شبکه می پردازد. در حالت ad-hoc network هیچ AP برای مدیریت کردن وجود ندارد، این دو حالت از شبکه در سیستم عامل FreeBSD قابل ارایه و پیاده سازی است FreeBSD. از استاندارد های 802.11g, 802.11b, and 802.11a پشتیبانی می کند. مراحل قبل از راه اندازی کردن:

این بخش خود به چند زیر مجموعه تقسیم می شود، در قسمت اول شما باید هسته FreeBSD را پیکربندی کنید، در قدم اول شما باید درایور کارت شبکه خود را در هسته FreeBSD بارگذاری کنید، یکی از کارت های وایرلس متداول در بازار کارت شبکه Atheros است که درایور آن ath نام دارد و به صورت زیر باید آنرا در هسته بارگذاری کنید، برای این کار باید قبل از بارگذاری هسته آنرا در هسته فعال کنید به همین دلیل باید از loader.conf استفاده کنید، این فایل در شاخه boot قرار دارد و با استفاده از اضافه کردن خط زیر درایور در هسته بارگذاری می شود:

```
if_ath_load="YES"
```

بخش اطلاعات بیشتر در مورد انواع کارتهای شبکه در FreeBSD از آنها حمایت می شود و درایور ها آن موجود است به آدرس زیر مراجعه کنید:

<https://www.freebsd.org/releases/10.2R/hardware.html#wlan>

در قدم بعدی باید قابلیت رمزنگاری را هم به هسته FreeBSD اضافه کنید، برای این کار باید ماژولهایی را که این قابلیت را به هسته اضافه می کنند را در هسته بارگذاری کنید، این بخش هم باید در فایل loader.conf اضافه شود. سه ماژول مورد نظر ما به اسم های زیر هستند:

- wlan_ccmp
- wlan_wep
- wlan_tkip

ماژول های wlan_ccmp(4), wlan_tkip(4) زمانی مورد استفاده قرار می گیرند که شبکه شما از WPA یا 802.11i استفاده می شود، به هر حال برای بارگذاری این ماژول ها در هسته از خطوط زیر استفاده کنید، این خطوط را به فایل loader.conf اضافه کنید:

```
wlan_wep_load="YES"
wlan_ccmp_load="YES"
wlan_tkip_load="YES"
```



ماژول wlan_wep(4) برای WEP cryptographic در هسته استفاده می شود.

ماژول wlan_tkip(4) از دو حالت TKIP و Michael cryptographic که برای دستگاه هایی که از استاندارد 802.11 حمایت می کند راه اندازی می کند.

ماژول wlan_ccmp(4) باعث اضافه شدن رمزنگاری AES-CCMP برای دستگاه هایی که از استاندارد 802.11 حمایت می کند راه اندازی می کند.

```
device wlan          # 802.11 support
device wlan_wep      # 802.11 WEP support
device wlan_ccmp     # 802.11 CCMP support
device wlan_tkip     # 802.11 TKIP support
device wlan_amrr     # AMRR transmit rate control algorithm
device ath           # Atheros pci/cardbus NIC's
device ath_hal       # pci/cardbus chip support
options AH_SUPPORT_AR5416 # enable AR5416 tx/rx descriptors
device ath_rate_sample # SampleRate tx rate control for ath
```

بعد از راه اندازی مجدد سیستم شما باید در پیغام های هسته سیستم خود در `dmesg` قرار داده می شود خطوط زیر را مشاهده کنید که به کارت شبکه سیستم شما مربوط می باشد:

```
ath0: <Atheros 5212> mem 0x88000000-0x8800ffff irq 11 at device 0.0 on cardbus1
ath0: [ITHREAD]
ath0: AR2413 mac 7.9 RF2413 phy 4.5
```



محدود کردن فضای دیسک با Disk quota

برای محدود کردن میزان مصرف فضای دیسک از قابلیت Disk quota در FreeBSD استفاده می شود که این محدودیت می تواند به یک کاربر خاص یا گروهی خاص که کاربران در آن قرار دارند اعمال شود.

همانطوری که می دانید در FreeBSD دو فایل سیستم وجود دارد، یکی UFS و دیگری ZFS که در این بخش شما با اعمال محدودیت در فایل سیستم UFS آشنا می شوید.

بخشهای این مقاله:

- فعال سازی disk quota
- تنظیم کردن یک محدودیت با disk quota
- چک کردن محدودیت اعمال شده.

فعال سازی Disk quota

در بخش اول از راه اندازی شما باید چک کنید که آیا هسته سیستم عامل FreeBSD شما آیا قابلیت راه اندازی disk quota را دارد؟ برای این کار از فرمان sysctl به صورت زیر استفاده کنید:

```
% sysctl kern.features.ufs_quota
kern.features.ufs_quota: 1
```

گر مقداری که در خروجی این فرمان برگشت داده می شود عدد 1 باشد بدین معناست که این قابلیت در هسته سیستم عامل شما فعال شده است و شما می توانید آنرا راه اندازی کنید، در غیر این صورت شما باید هسته FreeBSD خود را به خط زیر دوباره کامپایل کنید، این خط را در زیر مشاهده می کنید:

```
options QUOTA
```

برای دریافت اطلاعات بیشتر در زمینه راه اندازی و کامپایل کردن یک هسته جدید به مقاله زیر مراجعه کنید.

پیکربندی هسته در FreeBSD

حال بعد از فعال کردن این بخش در هسته نوبت به فعال سازی آن با استفاده از فایل rc.conf می رسد که شما باید در این بخش خط زیر را در این فایل اضافه کنید:

سیستم quota بعد از راه اندازی شده با استفاده از برنامه quotacheck هر بار به چک کردن وضعیت دیتاسیت فایل سیستم شما می پردازد که این امر زمان گیر است برای غیرفعال کردن این بخش شما باید خط زیر را در فایل Rc.conf قرار دهید:

```
check_quotas="NO"
```



در بخش پایانی از این بخش باید به تناسب نیاز خود در مورد فعال سازی اقدام کنید دو حالت فعال سازی وجود دارد ، حالت اول **per-user** است که با استفاده از آن می توانید برای هر کاربر محدودیت اعمال کنید و حالت دوم **group quotas** که برای اعمال محدودیت برای یک گروه خاص است و کاربرانی که در آن گروه قرار دارند را محدود می کند، شما می توانید به تناسب نیاز خود یکی یا هر دوی این تغییرات را فعال کنید، راه فعال کردن در فایل **fstab** است که در زیر شاخه **/etc** قرار دارد و شما می توانید برای هر فایل سیستمی که نیاز دارید این قابلیت را اعمال کنید.

نکته: اگر شما قصد دارید که برای مثال شاخه **home** کاربران را محدود سازی کنید بهتر است که در زمان نصب شاخه **/home** را یک فایل سیستم جداگانه در نظر بگیرید.

فعال سازی: per-user

برای فعال کردن این بخش کفایت که در پایان هر فایل سیستم یا خطی که در فایل **fstab** قرار دارد کلمه کلیدی **userquota** را اضافه کنید به صورت زیر:

```
/dev/das2g /home ufs rw,userquota 1 2
```

فعال سازی: group quotas

برای فعال کردن این بخش کفایت که در پایان هر فایل سیستم یا خطی که در فایل **fstab** قرار دارد کلمه کلیدی **groupquota** را اضافه کنید به صورت زیر:

```
/dev/das2g /home ufs rw,userquota,groupquota 1 2
```

بعد از راه اندازی کردن این بخش و راه اندازی شدن سیستم دو فایل به نام های **quota.user** و **quota.group** در شاخه اصلی فایل سیستمی که شما در آن **quota** را فعال کرده اید ایجاد می شود. با راه اندازی سیستم و راه اندازی **rc** این عمل اتفاق می افتد.

برای مدیریت کردن بخش راه اندازی در سیستم شما به فرمان های **quotaon(8)**, **quotaoff(8)**, **quotacheck(8)** نیاز دارید که برای دریافت اطلاعات بیشتر به **man** هر کدام از این صفحات مراجعه کنید.

تنظیم کردن یک محدودیت با disk quota

برای چک کردن از وضعیت راه اندازی **quota** از فرمان زیر استفاده کنید:

```
# quota -v
```

اگر خروجی این فرمان شامل خطوطی باشد که در مورد اطلاعات خلاصه وضعیت **quota** باشد بدین معناست که تنظیمات راه اندازی شما به درستی اعمال شده.



بخش های مختلفی برای اعمال محدودیت وجود دارد برای اعمال بر روی کاربران و یا گروه ها که هم می تواند مبتنی بر مقدار فضای مصرفی از هارد باشد به نام **block quota** و هم می تواند بر اساس تعداد فایل های هر کاربر باشد که به آن **inod quota** گویند. اگر شما همزمان هر دوی این محدودیت ها را اعمال کرده باشید هر کدام که زودتر اتفاق افتاد آن محدودیت اعمال می شود. این دو حالت در دو دسته **hard** و **soft** هم طبقه بندی می شود.

Hard limit

اگر در این حالت محدودیت اعمال کرده باشید اجازه پیشروی بیشتر به کاربر داده نمی شود، برای مثال اگر در فایل سیستمی شما فضای یک کاربر را 800 کیلوبایت محدود کرده اید و کاربر 790 کیلوبایت از فضا را مصرف کرده باشد در صورت ایجاد کردن یک فایل با میزان 11 کیلوبایت ایجاد کردن آن **fail** می شود.

Soft limit

این نوع از محدودیت با بازه زمانی خاصی کار می کند که به صورت پیش فرض یک هفته است و اگر کاربر در این مدت زمانی به فضای قبلی خود باز نگشت **soft limit** به حالت **hard limit** تبدیل می شود و کاربر دیگر اجازه ایجاد کردن فایل های بیشتر را ندارد تا زمانی که به مقدار فضای اعمال شده برگردد و دوباره حالت به **soft limit** بازگردانی می شود.

فرمان edquota

برای ویرایش کردن وضعیت محدودیت هر کاربر از این فرمان استفاده می شود. این فرمان شما با استفاده از ویرایشگر متنی کار می کند و مقدار پیش فرض را از متغیر **EDITOR** میگیرد و به صورت پیش فرض این مقدار **vi** است در نتیجه اطلاعات با ویرایشگر **vi** باز می شود. برای اعمال کردن محدودیت به یک کاربر خاص باید نام آن کاربر را بعد از سوئیچ **u** وارد کنید به صورت زیر:

```
# edquota -u abedini
Quotas for user abedini:
/usr: kbytes in use: 65, limits (soft = 50, hard = 75)
      inodes in use: 7, limits (soft = 50, hard = 60)
/usr/var: kbytes in use: 0, limits (soft = 50, hard = 75)
          inodes in use: 0, limits (soft = 50, hard = 60)
```

هر فایل سیستمی که شما در نظر دارید شامل یک خط برای اعمال محدودیت می باشد که می توانید با استفاده از ویرایشگر خود مقدار را تغییر دهید.