ویرایش
1

کوکی لینوکس به زبان فارسی

Linux Cookie in Persian

نویسنده : حسام الدین توحید

SKYWAN13@YAHOO.COM


بسم الله الرحمن الرحیم

شناسنامه کتاب

سرشناسه: کوکی لینوکس به زبان فارسی

عنوان و نام پدید آور: کوکی لینوکس به زبان فارسی ; حسام الدین توحید

مشخصات نشر: انتشار به صورت اینترنتی طبق لیسانس GPL v3 – ایران 1393

مشخصات ظاهری: 195 ص.:جدول،نمودار،متن.

وضعیت فهرست نویسی: فیپا

یادداشت: عنوان اصلی **Linux Cookie in Persian**

موضوع: سیستم عامل لینوکس ------ سیستم های عامل (کامپیوتر)

شناسه افزوده: توحید، حسام الدین، 1359 – ، مترجم و نویسنده

ناشر: اینترنتی

قیمت: رایگان

نسخه ویرایش: یک

مقدمه مولف :

آنچه پیش رو دارید به صورت رایگان و تحت لیسانس GPL v3 به علاقه مندان لینوکس هدیه می گردد . در تهیه این کتاب از سر فصل های درسی گفته شده در دوره های LPIc2 و RHCE استفاده شده ولازم می دانم از **مهندس مهدوی فر** به خاطر راهنمایی های مفیدشان و **مرکز آموزشهای پیشرفته دانشگاه شریف** تشکر کافی را داشته باشم. کتابی که پیش روی شماست سعی دارد با بکارگیری قابلیت های مختلف رابط خط فرمان، بدور از پیچیدگی های غیر ضروری و با نگاهی کاربردی، مروری داشته باشد بر راه اندازی و استفاده از چند سرویس مهم و پرکاربرد لینوکس که امید است مطالب ارائه شده بتواند باعث ارتقاء دانش فنی کاربران لینوکس و متخصصین IT شود. هر گونه استفاده نامناسب از محتوای ارائه شده بر عهده کاربر بوده و تمام حقوق این اثر به نویسنده آن تعلق دارد لذا با حفظ حقوق مولف در نشر آن بکوشید. خواهشمند است هرگونه نقص در محتوا را به ایمیل نگارنده ارسال فرمائید.

موفق باشید

حسام الدین توحید

مرداد 1393

فهرست مطالب

فصل پنجم – زمان بندی فرایندها توسط Cron

فصل ششم – بررسی LVM در لینوکس

فصل هفتم – راه اندازی سرویس DHCP در لینوکس

فصل هشتم – بررسی Xinetd

ضمیمه یک

فصل اول

راه اندازی FTP در لینوکس

آشنائی با پروتکل FTP

مقدمه:

امروزه از پروتکل های متعددی در شبکه های کامپیوتری استفاده می گردد که صرفا" تعداد اندکی از آنان به منظور انتقال داده طراحی و پیاده سازی شده اند . اینترنت نیز به عنوان یک شبکه گسترده از این قاعده مستثنی نبوده و در این رابطه از پروتکل های متعددی استفاده می شود. برای بسیاری از کاربران اینترنت همه چیز محدود به وب و پروتکل مرتبط با آن یعنی HTTP است ، در صورتی که در این عرصه از پروتکل های متعدد دیگری نیز استفاده می گردد. FTP نمونه ای در این زمینه است.

پروتکل FTP چیست؟

تصویر اولیه اینترنت در ذهن بسیاری از کاربران، استفاده از منابع اطلاعاتی و حرکت از سایتی به سایت دیگر است و شاید به همین دلیل باشد که اینترنت در طی سالیان اخیر به سرعت رشد کرده و متداول شده است . بسیاری از کارشناسان این عرصه اعتقاد دارند که اینترنت گسترش و عمومیت خود را مدیون سرویس وب می باشد .

فرض کنید که سرویس وب را از اینترنت حذف نمائیم. برای بسیاری از ما این سوال مطرح خواهد شد که چه نوع استفاده ای را می توانیم از اینترنت داشته باشیم؟ در صورت تحقق چنین شرایطی ، یکی از عملیاتی که کاربران قادر به انجام آن خواهند بود ، دریافت داده ، فایل های صوتی ، تصویری و سایر نمونه فایل های دیگر با استفاده از پروتکل FTP (برگرفته از Transfer Protocol File) است.

ویژگی های پروتکل FTP

پروتکل FTP ، اولین تلاش انجام شده برای ایجاد یک استاندارد به منظور مبادله فایل بر روی شبکه های مبتنی بر پروتکل TCP/IP است که از اوایل سال 1970 مطرح و مشخصات استاندارد آن طی RFC 959 در اکتبر سال 1985 ارائه گردید .

پروتکل FTP دارای حداکثر انعطاف لازم و در عین حال امکان پذیر به منظور استفاده در شبکه های مختلف با توجه به نوع پروتکل شبکه است .

پروتکل FTP از مدل سرویس گیرنده – سرویس دهنده تبعیت می نماید . برخلاف HTTP که یک حاکم مطلق در عرصه مرورگرهای وب و سرویس دهندگان وب است ، نمی توان ادعای مشابهی را در رابطه با پروتکل FTP داشت و هم اینک مجموعه ای گسترده از سرویس گیرندگان و سرویس دهندگان FTP وجود دارد .

برای ارسال فایل با استفاده از پروتکل FTP به یک سرویس گیرنده FTP نیاز می باشد . ویندوز دارای یک برنامه سرویس گیرنده FTP از قبل تعبیه شده می باشد ولی دارای محدودیت های مختص به خود می باشد . در این رابطه نرم افزارهای متعددی تاکنون طراحی و پیاده سازی شده است :WSFTP Professional، FTP Explorerو Smart FTP نمونه هائی در این زمینه می باشند .

پروتکل FTP را می توان به عنوان یک سیستم پرس وجو نیز تلقی نمود چراکه سرویس گیرندگان و سرویس دهندگان گفتگوی لازم به منظور تائید یکدیگر و ارسال فایل را انجام می دهند. علاوه بر این، پروتکل فوق مشخص می نماید که سرویس گیرنده و سرویس دهنده، داده را بر روی کانال گفتگو ارسال نمی نمایند . در مقابل ، سرویس گیرنده و سرویس دهنده در خصوص نحوه ارسال فایل ها بر روی اتصالات مجزا و جداگانه (یک اتصال برای هر ارسال داده) با یکدیگر گفتگو خواهند کرد (نمایش لیست فایل های موجود در یک دایرکتوری نیز به عنوان یک ارسال فایل تلقی می گردد) .

پروتکل FTP امکان استفاده از سیستم فایل را مشابه پوسته یونیکس و یا خط دستور ویندوز در اختیار کاربران قرار می دهد . سرویس گیرنده در ابتدا یک پیام را برای سرویس دهنده ارسال و سرویس دهنده نیز به آن پاسخ خواهد داد و در ادامه ارتباط غیرفعال می گردد . وضعیت فوق با سایر پروتکل هائی که به صورت تراکنشی کار می کنند ، متفاوت می باشد (نظیر پروتکل HTTP) . برنامه های سرویس گیرنده زمانی قادر به شبیه سازی یک محیط تراکنشی می باشند که از مسائلی که قرار است در آینده محقق شوند ، آگاهی داشته باشند . در واقع ، پروتکل FTP یک دنباله statefull از یک و یا چندین تراکنش است.

سرویس گیرندگان ، مسئولیت ایجاد و مقداردهی اولیه درخواست ها را برعهده دارند که با استفاده از دستورات اولیه FTP انجام می گردد. دستورات فوق ، عموما" سه و یا چهار حرفی می باشند (مثلا" برای تغییر دایرکتوری از دستور CWD استفاده می شود). سرویس دهنده نیز بر اساس یک فرمت استاندارد به سرویس گیرندگان پاسخ خواهد داد (سه رقم که به دنبال آن از space استفاده شده است به همراه یک متن تشریحی) . سرویس گیرندگان

می بایست صرفا" به کد عددی نتیجه استناد نمایند چراکه متن تشریحی تغییر پذیر بوده و در عمل برای اشکال زدائی مفید است (برای کاربران حرفه ای) .

پروتکل FTP دارای امکانات حمایتی لازم برای ارسال داده با نوع های مختلف می باشد . دو فرمت متداول، اسکی برای متن (سرویس گیرنده با ارسال دستور TYPE A ، موضوع را به اطلاع سرویس دهنده می رساند) و image برای داده های باینری است (توسط TYPE I مشخص می گردد) . ارسال داده با فرمت اسکی در مواردی که ماشین سرویس دهنده و ماشین سرویس گیرنده از استانداردهای متفاوتی برای متن استفاده می نمایند ، مفید بوده و یک سرویس گیرنده می تواند پس از دریافت داده آن را به فرمت مورد نظر خود ترجمه و استفاده نماید . مثلا" در نسخه های ویندوز از یک دنباله carriage return و linefeed برای نشان دادن انتهای خط استفاده می گردد در صورتی که در سیستم های مبتنی بر یونیکس صرفا" از یک linefeed استفاده می شود . برای ارسال هرنوع داده که به ترجمه نیاز نداشته باشد،می توان از ارسال باینری استفاده نمود.

اتخاذ تصمیم در رابطه با نوع ارسال فایل ها در اختیار سرویس گیرنده است (برخلاف HTTP که می تواند به سرویس گیرنده نوع داده ارسالی را اطلاع دهد) . معمولا" سرویس گیرندگان ارسال باینری را انتخاب می نمایند و پس از دریافت فایل ، ترجمه لازم را انجام خواهند داد . ارسال باینری ذاتا" دارای کارآئی بیشتری است چراکه سرویس دهنده و سرویس گیرنده نیازی به انجام تراکنش های on the fly نخواهند داشت . ارسال اسکی گزینه پیش فرض انتخابی توسط پروتکل FTP است و در صورت نیاز به ارسال باینری ، سرویس گیرنده می بایست این موضوع را از سرویس دهنده درخواست نماید .

یک اتصال پروتکل TCP/IP (نسخه شماره چهار) شامل دو نقطه مجزا می باشد که هر نقطه از یک آدرس IP و یک شماره پورت استفاده می نماید . برقراری ارتباط بین یک سرویس گیرنده و یک سرویس دهنده منوط به وجود چهار عنصر اطلاعاتی است : آدرس سرویس دهنده ،پورت سرویس دهنده ، آدرس سرویس گیرنده و پورت سرویس گیرنده . در زمان برقراری یک ارتباط ، سرویس گیرنده از یک شماره پورت استفاده می نماید . این شماره پورت می تواند متناسب با نوع عملکرد برنامه سرویس گیرنده به صورت اختیاری و یا اجباری باشد . مثلا" برخی برنامه های سرویس گیرنده به منظور ارتباط با سرویس دهنده ، نیازمند استفاده از یک شماره پورت خاص می باشند (نظیر برنامه های سرویس گیرنده وب و یا مرورگرهای وب که از پورت شماره 80 به منظور ارتباط با سرویس دهنده وب استفاده می نماید) . در مواردی که الزامی در خصوص شماره پورت وجود ندارد از یک شماره پورت موقتی و یا ephemeral استفاده می گردد . این نوع پورت ها موقتی بوده و توسط IP stack ماشین مربوطه به متقاضیان نسبت داده شده و پس از خاتمه ارتباط ، پورت آزاد می گردد . با توجه به این که اکثر IP Stacks

بلافاصله از پورت موقت آزاد شده استفاده نخواهند کرد (تا زمانی که تمام pool تکمیل نشده باشد) ، در صورتی که سرویس گیرنده مجددا" درخواست برقراری یک ارتباط را نماید ، یک شماره پورت موقتی دیگر به وی تخصیص داده می شود .

پروتکل FTP منحصرا" از پروتکل TCP استفاده می نماید(هرگز از پروتکل UDP استفاده نمی شود) . معمولا" پروتکل های لایه Application (با توجه به مدل مرجع OSI) از یکی از پروتکل های TCP و یا UDP استفاده می نمایند (به جزء پروتکل DNS) . پروتکل FTP نیز از برخی جهات شرایط خاص خود را دارد و برای انجام وظایف محوله از دو پورت استفاده می نماید . این پروتکل معمولا" از پورت شماره 20 برای ارسال داده و از پورت 21 برای گوش دادن به فرامین استفاده می نماید . توجه داشته باشید که برای ارسال داده همواره از پورت 20 استفاده نمی گردد و ممکن است در برخی موارد از پورت های دیگر استفاده شود .

اکثر سرویس دهندگان FTP از روش خاصی برای رمزنگاری اطلاعات استفاده نمی نمایند و در زمان login سرویس گیرنده به سرویس دهنده ، اطلاعات مربوط به نام و رمز عبور کاربر به صورت متن معمولی در شبکه ارسال می گردد . افرادی که دارای یک Packet sniffer بین سرویس گیرنده و سرویس دهنده می باشند ، می توانند به سادگی اقدام به سرقت نام و رمز عبور نمایند . علاوه بر سرقت رمزهای عبور ، مهاجمان می توانند تمامی مکالمات بر روی اتصالات FTP را شنود و محتویات داده های ارسالی را مشاهده نمایند . پیشنهادات متعددی به منظور ایمن سازی سرویس دهنده FTP مطرح می گردد ولی تا زمانی که رمزنگاری و امکانات حفاظتی در سطح لایه پروتکل IP اعمال نگردد (مثلا" رمزنگاری توسط IPsecs) ، نمی بایست از FTP استفاده گردد خصوصا" اگر بر روی شبکه اطلاعات مهم و حیاتی ارسال و یا دریافت می گردد .

همانند بسیاری از پروتکل های لایه Application ، پروتکل FTP دارای کدهای وضعیت خطاء مختص به خود می باشد (همانند HTTP) که اطلاعات لازم در خصوص وضعیت ارتباط ایجاد شده و یا درخواستی را ارائه می نماید . زمانی که یک درخواست (GET , PUT) برای یک سرویس دهنده FTP ارسال می گردد ، سرویس دهنده پاسخ خود را به صورت یک رشته اعلام می نماید . اولین خط این رشته معمولا" شامل نام سرویس دهنده و نسخه نرم افزار FTP است .در ادامه می توان دستورات GET و یا PUT را برای سرویس دهنده ارسال نمود . سرویس دهنده با ارائه یک پیام وضعیت به درخواست سرویس گیرندگان پاسخ می دهد .

FTP ، یک پروتکل ارسال فایل است که با استفاده از آن سرویس گیرندگان می توانند به سرویس دهندگان متصل و صرفنظر از نوع سرویس دهنده اقدام به دریافت و یا ارسال فایل نمایند. پروتکل FTP به منظور ارائه خدمات خود

از دو حالت متفاوت استفاده می نماید : Active Mode و Passive Mode . مهمترین تفاوت بین روش های فوق جایگاه سرویس دهنده و یا سرویس گیرنده در ایجاد و خاتمه یک ارتباط است.

همانگونه که اشاره گردید ، یک اتصال پروتکل TCP/IP (نسخه شماره چهار) شامل دو نقطه مجزا می باشد که هر نقطه از یک آدرس IP و یک شماره پورت استفاده می نماید . برقراری ارتباط بین یک سرویس گیرنده و یک سرویس دهنده منوط به وجود چهار عنصر اطلاعاتی است : آدرس سرویس دهنده ،پورت سرویس دهنده ، آدرس سرویس گیرنده و پورت سرویس گیرنده . در زمان برقراری یک ارتباط ، سرویس گیرنده از یک شماره پورت استفاده می نماید . این شماره پورت می تواند متناسب با نوع عملکرد برنامه سرویس گیرنده به صورت اختیاری و یا اجباری باشد . مثلا" برخی برنامه های سرویس گیرنده به منظور ارتباط با سرویس دهنده ، نیازمند استفاده از یک شماره پورت خاص می باشند (نظیر برنامه های سرویس گیرنده وب و یا مرورگرهای وب که از پورت شماره 80 به منظور ارتباط با سرویس دهنده وب استفاده می نماید) . در مواردی که الزامی در خصوص شماره پورت وجود ندارد از یک شماره پورت موقتی و یا ephemeral استفاده می گردد . این نوع پورت ها موقتی بوده و توسط IP stack ماشین مربوطه به متقاضیان نسبت داده شده و پس از خاتمه ارتباط ، پورت آزاد می گردد . با توجه به این که اکثر IP Stacks بلافاصله از پورت موقت آزاد شده استفاده نخواهند کرد (تا زمانی که تمام pool تکمیل نشده باشد) ،در صورتی که سرویس گیرنده مجددا" درخواست برقراری یک ارتباط را نماید ، یک شماره پورت موقتی دیگری به وی تخصیص داده می شود.

معروف ترین این سرویس دهنده هاVSFTP و PROFTP می باشد که البته بهترین آنها از نظر ردهت VSFTP می باشد . در لینوکس بهترین نرم افزار کلاینتی آن FileZila و Lftp می باشد .

مقایسه ای بین VSFTP و PROFTP

VSFTP از امنیت بالایی برخوردار بوده و به شدت Stable می باشد و از مهمترین قابلیت های آن می توان به Multi Homing بودن آن اشاره کرد. این قابلیت اجازه می دهد چندین Ftp Daemon روی یک سرور اجرا شود که هر کدام از اینها دارای تنظیمات مختص به خود می باشد. مبنای احراض هویت VSFTP فایل /etc/passwd است. یعنی به صورت پیش فرض بانک جداگانه ای برای یوزرها ندارد بلکه از یـــوزرهای Localپشتیبانی به عمل می آورد. تنها عیب VSFTP راه اندازی مشکل و سرعت پائین آن به نسبت PROFTP است. VSFTP به صورت پیش فرض به صورت Standalone بالا می آید ولی قابلیت این را دارد که تحت نظر Xinetd ارائه سرویس کند.

اکثر Hostingها از PROFTP استفاده می کنند چون سرعت احراض هویت بالاتری دارد و علت آن هم نداشتن ماژول امنیتی بر روی آن است که باعث می شود به راحتی مورد حمله هکرها قرار گیرد. مهمترین مزیت آن فقط سرعت بالای احرض هویت و اتصال آن می باشد و البته بسیاری از فیچرهای VSFTP را هم ندارد. PROFTP به صورت پیش فرض از فایل /etc/passwd استفاده نمی کند بلکه برای احرض هویت یوزرها از فایل جداگانه ای بهره می برد. پس از این مقدمه ، در ادامه به بررسی هر یک از روش های Active و Passive در پروتکل FTP خواهیم پرداخت .

مدهای کاری سرویس دهنده FTP

ActiveMode

Active Mode ، روش سنتی ارتباط بین یک سرویس گیرنده FTP و یک سرویس دهنده می باشد که عملکرد آن بر اساس فرآیند زیر است :

تمام FTP های دنیا در دو mode کار می کنند. یا Activeهستند یا Passive . و البته به طور پیش فرض FTPها در حالت اکتیو کار می کنند.

کلاینتیک پورت رندوم باز کرده و از طریق آن یک ارتباط با پورت 21 سرویس دهنده FTP برقرار می نماید و روی پورت 21 احراض هویت می شود. . پورت 21 ، پورتی است که سرور به آن گوش فرا می دهد تا از صدور فرامین آگاه و آنان را به ترتیب پاسخ دهد . کلاینت برای برقراری ارتباط با سرور از یک پورت تصادفی و موقتی (بزرگتر از 1024) استفاده می نماید(پورت x).

کلاینت شماره پورت لازم برای ارتباط سرویس دهنده با خود را از طریق صدور دستور PORT N+1 به وی اطلاع می دهد (پورت x+1)

سرور یک ارتباط را از طریق پورت 20 خود با پورت مشخص شده کلاینت (پورت x+1) برقرار می نماید .

در فرآیند فوق ، ارتباط توسط کلاینت آغاز و پاسخ به آن توسط سرور و از طریق پورت x+1 که توسط کلاینت مشخص شده است ، انجام می شود . در صورتی که کلاینت از سیستم ها و دستگاه های امنیتی خاصی نظیر فایروال استفاده کرده باشد ، می بایست تهمیدات لازم به منظور ارتباط کامپیوترهای میزبان راه دور به کلاینت پیش بینی تا آنان بتوانند به هر پورت بالاتر از 1024 کلاینت دستیابی داشته باشند . بدین منظور لازم است که پورت های اشاره شده بر روی ماشین کلاینتopen باشند . این موضوع می تواند تهدیدات و چالش های امنیتی متعددی را برای سرویس گیرندگان به دنبال داشته باشد .

Passive Mode

در Passive Mode ، که به آن " مدیریت و یا اداره سرویس گیرندگان FTP" نیز گفته می شود از فرآیند زیر استفاده می گردد :

کلاینت دو پورت را فعال می نماید (پورت x و x+1)

ارتباط اولیه از طریق پورت xکلاینت با پورت 21سرورآغاز می گردد کلاینت از این پورت احراض هویت انجام میدهد .

سرور یک پورت را فعال (Y) و به کلاینت شماره پورت را اعلام می نماید .

در ادامه کلاینت یک اتصال از طریق پورت x+1 با پورت yسروربرقرار می نماید. در فرآیند فوق ، کلاینت دارای نقش محوری است و فایروال موجود بر روی کلاینت می تواند درخواست های دریافتی غیرمجاز به پورت های بالاتر از 1024 را به منظور افزایش امنیت بلاک نمایند . در صورتی که بر روی کامپیوترهای سرور نیز فایروال نصب شده باشد ، می بایست پیکربندی لازم به منظور استفاده از پورت های بالاتر از 1024 بر روی آن آنجام و آنان open گردند . باز نمودن پورت های فوق بر روی سرور می تواند چالش های امنیتی خاصی را برای سرور به دنبال داشته باشد و متاسفانه تمامی کلاینتهایFTP از Passive Mode حمایت نمی نمایند . اگر یک کلاینت بتواند به یک سرورlogin نماید ولی قادر به ارسال داده بر روی آن نباشد ، نشاندهنده این موضوع است که فایروال و یا Gateway برای استفاده از PassiveMode به درستی پیکربندی نشده است .

مقایسه بین اکتیو و پسیو به زبان ساده :

در حالت Active باید فایروال کلاینتها پیکربندی شود ولی در حالت Passive فایروال سمت سرور کانفیگ میشود. مشکل Active در همین است که باید فایروال سمت کلاینت توسط خود کاربر پیکربندی شودکه این می تواند خود مشکل آفرین باشد. اما در حالت Passive تنظیمات فایروال توسط مدیر شبکه انجام می شود بنابراین لازم است به سرور اجازه داده شود که به اتصالات هر پورت بالاتر از 1024 پاسخ دهد . ترافیک فوق ، معمولا" توسط فایروال سرور بلاک می گردد . در چنین شرایطی امکان استفاده از Passive Mode وجود نخواهد داشت .

حالت پسیو دارای سرعت بالاتر و overhead کمتری است و نداشتن مشکلات فایروالی هم جزو محسنات این مد به حساب می آید . با توجه به مستندات درج شده در RFC 1579 ، استفاده از Passive Mode به دلایل متعددی به Active Mode ترجیح داده می شود :

تعداد سرویس دهندگان موجود بر روی اینترنت به مراتب کمتر از سرویس گیرندگان می باشد . با استفاده از امکانات موجود می توان سرویس دهندگان را پیکربندی تا بتوانند از مجموعه پورت های محدود و تعریف شده ای با در نظر گرفتن مسائل امنیتی ، استفاده نمایند.

نصب راه اندازی VSFTP

جهت نصب این سرویس دهنده میتوان از Yum و یا اگر پکیج آن از قبل موجود باشد می توان از rpm برای نصب اسفاده کرد. ولی توصیه می شود در صورت امکان از Yum برای نصب نرم افزارها استفاده کنید زیرا نیازمندی های لازم را دانلود و نصب می کند. این قابلیت در rpm وجود ندارد لذا جهت نصب از دستور زیر استفاده می کنیم .

ابتدا باید از نصب بودن پکیج vsftp اطمینان حاصل کنیم لذا با دستور زیر از سیستم query می گیریم :

#rpm -qa | grep vsftp

در صورت نصب نبودن ، در سیستم های ردهت جهت نصب vsftpd از yum استفاده می کنیم :

#yum -y install vsftp

بعد از نصب ، باید اطمینان حاصل کنیم گه آیا پکیج vsftp بر روی سیستم نصب شده است یا خیر لذا با دستور زیر از سیستم query می گیریم :

#rpm -qa | grep vsftp

سپس با دستور زیر شاخه ها و مسیرهایی که فایل های این سرویس در آن ایجاد شده است را چک می کنیم :

#rpm -ql vsftpd

و با این دستور هم اطلاعات لازم را در مورد پکیج این سرویس به دست می آوریم :

#rpm -qi vsftpd

سپس با دستور chkconfig مشخص می کنیم در چه runlevel هایی فعال باشد :

#chkconfig vsftpd on

و در انتها سرویس را reset می کنیم :

#service vsftpd restart

مهمترین مسیرهای ایجاد شده توسط VSFTP

با نصب این سرویس چندین شاخه و مسیر جدید به سیستم اضافه می شود که شش عدد از انها در زیر مختصرا توضیح داده شده اند :

/etc/vsftpd/
/etc/logrotate.d/vsftpd.log
/etc/pam.d/vsftpd
/etc/rc.d/init.d/vsftpd
/var/ftp/pub/
/usr/sbin/vsftpd/

/etc/vsftpd/:

در زیر این دایرکتوری چهار فایل مهم پیکربندی این سرویس قرار دارد که به ترتیب شرح **داده می شود :**

Vsftpd.conf
User_list
Ftpusers
Vsftpd_conf_migrate.sh

Vsftpd.conf: این فایل اصلی ترین فایل پیکربندی سرویس vsftpd می باشد.
User_list: این فایل شامل لیست یوزرهایی است که به آنها دسترسی یا عدم دسترسی به ftp داده می شود به شرط آنکه گزینه **userlist_deny** را مقدار دهی کنیم.
Ftpusers: هر یوزری که نام آن در این فایل قرار بگیرد به آن اجازه login به ftp داده نمی شود. در اصل این فیل یک blacklist می باشد.
Vsftpd_conf_migrate.sh: برای migrate کردن و جابه جایی بین دو ftp از این اسکریپت استفاده میشود.

/etc/logrotate.d/vsftpd.log:

فایل کانفیگ rotate لاگ این سرویس در این آدرس قرار دارد.

:/etc/pam.d/vsftpd

Pam یک مکانیزم امنیتی است که برای کنترل سرویس ها به کار می رود. اگر بخواهیم مکانیزم احراض هویت vsftpd در اختیار pam باشد باید در این فایل تنظیمات لازم را اعمال کنیم.

:/etc/rc.d.init.d/vsftpd

اسکریپت اجرای سرویس در این مکان قرار دارد . vsftpd به صورت standalone اجرا میشود و زیر مجموعه init قرار دارد.

:/var/ftp/pub/

این مسیر برای قرار دادن فایل ، و دایرکتوری جاری یوزرهایی که لاگین می کنند به کار می رود. به این مسیر ftproot گفته می شود. تمام یوزرهای anonymous به طور پیش فرض وارد این دایرکتوری می شوند و کاربران local هم بعد از ورود به ftp به دایرکتوری home مربوطه هدایت خواهند شد مگر اینگه ما این مسیر را تغییر دهیم .

:/usr/sbin/vsftpd

فایل دستور vsftpd در این مسیر قرار دارد .

مروری بر تنظیمات فایل کانفیگ اصلی VSFTP

Syntax این فایل بدین گونه است که هر چیزی که درون آن نوشته می شود باید بدون فاصله باشد. کلا برای اتصال به vsftpd دو نوع یوزر داریم این یوزرها یا یوزر local سیستم هستند یا یوزرهای anonymous . کانفیگ کلی vsftpd بدون آپشن خاصی 13 خط می باشد که در زیر مهمترین آنها توضیح داده شده است . برای کسب اطلاعات تکمیلی به man vsftpd.conf مراجعه شود.

#vi /etc/vsftpd/vsftpd.com

anonymous_enable=YES

Yes بودن این گزینه به کاربران anonymous اجازه می دهد که از طریق محیط گرافیکی بدون پسورد وارد دایرکتوری pub سرور FTPشوند. یوزرهای anon اجازه chroot را ندارند .

anon_root=/opt/dir_anon

اگر بخواهیم کاربران anon به محض ورود به FTPبه دایرکتوری مشخصی هدایت شوند در جلوی این گزینه مسیر مورد نظر را وارد می کنیم .

anon_upload_enable=YES

Yes بودن این آپشن اجازه می دهد یوزرهای anon بتوانند در FTPفایل آپلود کنند.

anon_mkdir_enable=YES

با yes قرار دادن مقدار این خط یوزرهای anon می توانند در FTPدایرکتوری ایجاد کنند.

anon_max_rate=4000

مقدار این خط نرخ حداکثر سرعت دانلود و آپلود یوزرهای anon را مشخص می کند. این نرخ بر اساس بایت می باشد.

no_anon_passwd=YES

با فعال کردن این خط ، FTPدر محیط cli از یوزرهای anon پسورد نمی خواهد.

anon_umask=???

مجوز های پیش فرض ایجاد فایل و دایرکتوری را برای کاربرanonymous تعیین می کند.

anon_other_write_enable = no

اگر این خط برابر YES باشد کاربران anon به غیر از آپلود و ایجاد دایرکتوری مجاز به انجام عملیات نوشتن حذف و تغییر نام خواهند شد. به طور کلی این کار توصیه نمی شود اما برای تکمیل گنجانده شده است.

local_enable=YES

با فعال کردن این گزینه یوزرهای local ای که داخل فایل passwd هستند این اجازه را پیدا می کنند با یوزر و پسورد خود به FTPلاگین کرده و وارد دایرکتوری خانگی خودشان بشوند. یوزرهای local اجازه changroot را دارند که این برای سیستم یک خطر امنیتی محسوب شده و باید این قابلیت را غیرفعال کرد.

write_enable=YES

این گزینه به یوزرهای local اجازه آپلود فایل در FTP را می دهد. این آپشن زمانی کاربرد دارد که فایل را تحت FTPبسازیم نه تحت bash .

local_umask=022

این umask پرمیژن پیش فرضی است که برای فایلهای آپلود شده توسط کاربران local ستفاده می شودکه پیش فرض 775 می باشد.

local_max_rate=10000

مقدار این خط نرخ حداکثر سرعت دانلود و آپلود یوزرهای local را مشخص می کند. این نرخ بر اساس بایت می باشد .

local_root=/var/tmp

اگر بخواهیم کاربران local به محض ورود به ftp به دایرکتوری مشخصی هدایت شوند در جلوی این گزینه مسیر مورد نظر را وارد می کنیم.

dirmessage_enable=YES

این گزینه فقط مختص یوزرهایی است که تحت cli به FTP لاگین می کنند. ما می توانیم برای هر دایرکتوری یک message ایجاد کنیم تا به محض ورود کاربر به آن دایرکتوری پیام مورد نظر برای کاربر به نمایش داده شود. برای ایجاد پیام به داخل دایرکتوری مورد نظر رفته و یک فایل به نام message. ایجاد می کنیم و پیام مودر نظرمان را درون ان ذخیره می کنیم. با yes بودن این خط ، vsftpd به محض ورود کاربر به دایرکتوری از قبل مشخص شده ، ابتدا انجا را چک می کند تا ببیند آیا چنین فایلی وجود دارد یا خیر . در موجود بودن پیام داخل آن را برای کاربر به نمایش می گذارد.

xferlog_enable=YES

لاگ های vsftpd به دو صورت ذخیره می شوند. یا با فایل vsftpd.log درون خود vsftpd ذخیره می شود که باید این گزینه NO باشد. یا با YES قرار دادن این خط کاری می کنیم log آن توسط log سرور در دایرکتوری مربوطه ذخیره شود.

listen=yes

اگر این گزینه yes باشد FTPدر مد standalone و تحت نظر init کار می کند. اگر بخواهیم این سرویس تحت نظر xinetd اداره شود باید این گزینه را برابر no قرار داده و در زیر xinetdیک فایل کانفیگ به نام vsftpdبسازیم . در انتهای این مقاله نمونه ای از این فایل آورده شده است.

max_per_ip=20

این خط مشخص می کند چه تعداد کانکشن می تواند از هر IP به سرور متصل شود.

max_cllient=100

این خط بیان می کند در یک زمان حداکثر 100 یوزر می توانند هم زمان به FTP متصل شوند.

chown_uploads=YES

نام یوزر =chown_username

اگر بخواهیم مالک (owner) فایل های که یوزرهای anon آپلود می کنند یوزر دیگری باشد تا خاصیت اجرا را از یوزر anon بگیریم خط اول را برابر yes قرار داده و در خط دوم نام یوزری که می خواهیم owner فایل ها باشد را وارد می کنیم.

ftp_banner=welcome to FTP
banner_file=/opt/ftp/ftp.txt

هر متنی که در جلوی عبارت خط اول نوشته شود برای یوزرهایی که با محیط cli لاگین کرده اند نمایش درمی آید. باید دقت داشت در جلو این عبارت نمی توان بیش از یک خط نوشت. اگر متن ما بیش از چند خط بود باید آن را در یک فایل جداگانه نوشته و آدرس آن را در جلوی خط دوم وارد کرد.

pam_service_name=VSFTPD
tcp_wrappers=YES

این دو خط مکانیزم های امنیتی کنترل سرویس FTP را مشخص می کند. اگر بخواهیم رنجی از IP را Block کنیم تا به FTP دسترسی نداشته باشندباید از مکانیزم امنیتی tcp_wrappers استفاده شود. البته باید از قبل IP های مجاز و غیر مجاز را درون فایل های /etc/hosts.allow و /etc/hosts.deny وارد کنیم جهت اطلاع از چگونگی تنظیم این فایل ها_accessman host را مطالعه کنید.

userlist_enable=YES
userlist_file=/etc/vsftpd/user_list

اگر مقدار این خط yes باشد محتویات فایلی که در خط دوم مسیر دهی شده خوانده می شود و فقط به یوزرهای که در این فایل ثبت شده اند اجازه دسترسی به FTP داده می شود.

userlist_deny=YES
userlist_file=/etc/vsftpd/user_list

اگر مقدار این خط yes باشد محتویات فایلی که در خط دوم مسیر دهی شده خوانده می شود و اسامی وزرهایی که در این فایل قرار دارند نمی توانند به FTP دسترسی داشته باشند.

nopriv_user=ali

اگر اجازه آپلود را به یوزرهای anon بدهیم باید قابلیت اجرا را از آنها بگیریم. برای این کار یک یوزر ساخته و اجازه اجرای فایل را از آن می گیریم. با وارد کردن نام یوزر مربوطه در اینجا از این به بعد سطح دسترسی این یوزر به یوزرهای anon اعمال می شود. این کار برای امن سازی FTP لازم است.

chroot_local_user=YES

اگر این خط برابر YES باشد از تمام یوزرهای local قابلیت change root گرفته می شود.

chroot_list_enable=YES
chroot_list_file=/etc/vsftpd/chroot_list

این خطوط برای jail کردن یوزرها به کار می رود. اگر قابلیت change root را از یوزری بگیریم اصطلاحا میگوییم یوزر را jail کرده ایم. اگر بخواهیم بعضی از یوزرها local قابلیت change root نداشته باشند خط اول را برابر YES قرار داده و در خط دوم آدرس لیست یوزرهای انتخابی را وارد می کنیم. این فیچر مخصوص یوزرهای local می باشد .

deny_email_enable=YES
banned_email_file=/etc/vsftpd/banned_emails

وقتی کاربر با مرورگر خود به یک FTP وصل می شود در صورتی که یوزر local نباشد به صورت یوزر anon لاگین کرده و به صورت پیش فرض وارد مسیر /var/ftp/ می شود. اگر دقت کرده باشید بسیاری از سایتهای ftp موقع ورود از شما یوزر و پسورد نمی خواهد در صورتی که یوزر anon هم برای ورود نیاز است یک پسورد دلخواه حتی یک کارکتر وارد کند. توضیحی که برای این اتصال بدون پسورد وجود دارد این است که همه مرورگرها یک یوزر و پسورد پیش فرض داخلی برای احراض هویت دارند که در چنین مواقعی استفاده میکنند. به طور مثال پسورد داخلی فایرفاکس mozila@example.com است . حال اگر این پسورد را در فایل banned_email وارد کنیم هیچ کاربری نمی تواند با مرورگر فایرفاکس به FTP متصل شود. این کار را برای محدود سازی اتصال با مرورگرهای خاص است. به این کار banned کردن ایمیل گفته می شود.

port_enable=YES
pasv_enable=YES
خط اول مشخص می کند FTP ما در Active mode کار کند و خط دوم حالت Passive

idle_session_timeout=300
این خط بیان میکند در صورتی که کاربر غیر فعال بود بعد از چند ثانیه ارتباط او توسط سرور قطع شود.

delete_failed_uploads=YES
اگر این گزینه فعال (YES) باشد تمامی آپلود هایfailed شده پاک خواهند شد .

download_enable=YES
اگر فعال (YES) باشد تمامی درخواست های دانلود رَد خواهند شد.

listen_port=21
به طور پیش فرض پورت این سرویس ۲۱ است که برای امنیت بیشتر می توان این پورت را تغییر داد. البته همزمان باید در فایل کانفیگ این سرویس در Xinetd و فایل etc/services/ تغییراتی را اعمال نمود.

listen_address=192.168.1.1
اگر چندین کارت شبکه روی سرور داشته باشیم می توانیم یکی از آنها را به سرویس ftp اختصاص دهیم. حتی اگر درخواست ها زیاد باشد می توان دو یا چندین کارت شبکه را به این امر اختصاص داد. اگر این چارامتر مقداردهی نشود تمام کارت های شبکه برای این کار استفاده می شوند.

allow_anon_ssl = YES
اگر YES باشد کاربران anon مجاز به استفاده از ارتباطات امن SSL می باشند.

ascii_download_enable=YES
اگر YESباشد انتقال داده به صورت اسکی خواهد بود.

force_anon_logins_ssl =YES

در صورت فعال بودن ssl_enable و این گزینه کاربران anon مجبور به یک اتصال امن SSL برای ارسال رمز عبور خواهند بود.

force_dot_file=YES

در صورتی که این خط مقدار YES داشته باشد حتی اگر دستور فهرست کردن دایرکتوری ها بدون سوییچ a باشد باز هم فایل های که ابتدایشان نقطه دارند نشان داده نخواهد شد(فایل های مخفی)

ls_recurse_enable=YES

اگر این آپشن فعال باشد اجازه اجرای دستپر ls –R را دارید. فقط یک مشکلی که دارد این است که بکارگیری این آپشن در سایت های که حجم بسیاری فایل بر روی انها وجود دارد باعث هدر رفتن منابع سیستم می گردد.

dirlist_enable =NO

اگر NO باشد به هیچ کدام از دستورات directory list اجازه اجرا نخواهد داد.

hide_ids=YES

اگر این گزینه فعال باشد همه اطلاعات کاربر و گروه در لیست دایرکتوری را نمایش می دهد.

no_anon_password=YES

هنگامی که فعال باشد از کاربر anon درخواست یوزر و پسورد نمی کند.

dual_log_enable=YES

اگر این گزینه را فعال کنیم دو نوع لاگ برای ما تهیه می کند . یکی xferlog که لاگ پیش فرض است و یکی هم لاگvsftpd را ثبت می کند.

force_dot_files=YES

اگر بخواهیم فایل های مخفی برای کاربران به نمایش در بیاید این گزینه را برابر YES قرار می دهیم.

جهت اطلاعات بیشتر لطفا man vsftpd.conf را مطالعه بفرمائید.

تغییر مسیر یوزرهای Local بعد از login به FTP

یوزرهای anon به صورت پیش فرض بعد از login به مسیر /var/ftp/ هدایت می شوند ولی یوزرهای Localسیستم بعد از Login به دایرکتوری Home خود وارد می شوند.

ما می توانیم کارکنیم که یوزرهای Local به جای اینکه به دایرکتوری Home خودشان وارد شوند به /var/ftp/ هدایت شوند و یا قابلیت ساخت دایرکتوری، فایل، آپلود و دانلود را هم به آنها داده ، یا آنها را بنا بر صلاح دید سازمان محدود کنیم. برای چنین کاری خطوط زیر را از فایل اصلی پیکربندی اصلاح می کنیم :

#vi /etc/vsftpd/vsftpd.conf

Anonymous_enable=NO
Local_enable=YES
Write_enable=YES
Local_umask=002
Dirmessage_enable=YES
Local_root=/var/ftp/pub

نکته : معمولا سیستم های گرافیکی کانکشن FTP را cash میکنند.

نکته مهم : اگر گزینه ای را بخواهیم غیر فعال کنید بهتر است به جای کامنت کردن ، در جلوی آن NO را بنویسیم .

محدود کردن دسترسی یوزرها به FTP

اگر بخواهیم تعداد خاصی از یوزرهای Local به FTP دسترسی داشته باشند باید اسامی آنها را داخل یکی از فایل های ftpusers و user_list وارد کرده و در فایل کانفیگ تغییراتی را اعمال کنیم.اگر بخواهیم از بین تعداد زیادی ازیوزرها فقط بعضی اجازه دسترسی به FTP داشته باشند باید گزینه ???=userlist_deny را در فایل کانفیگ برابر NO قرار دهیم. در صورت NO بودن این گزینه فقط یوزرهای موجود در فایل user_list می توانند به FTP دسترسشی داشته باشند.

YES بودن این گزینه چندان منطقی نیست. اگر این گزینه برابر YES باشد و اسامی تعریف شده در این فایل در ftpusers هم موجود باشند آموقت YES بودن این گزینه بی معنی می شود چون از طریق بررسی فایل ftpusers به یوزرها اجازه دسترسی داده می شود. در این حالت هر دو فایل جهت اعطا حق دسترسی مورد بررسی قرار می گیرند. اما اگر این مقدار برابر NO باشد دیگر فایل ftpusers مورد بررسی قرار نمی گیرد.

Jail کردن یوزرها در FTP

وقتی کاربری به FTP لاگین می کند نباید بتواند به دایرکتوری به غیر از ریشه ای که به آن وارد شده برود. به طور مثال اگر تعریف کرده باشیم یوزر به محض ورود به دایرکتوری /var/ftp/pub هدایت شود ، نباید بتواند به سمت دایرکتوری بالایی تغییر دایرکتوری بدهد . به جلوگیری از چنین کاری Jail کردن یوزر گفته می شود.

با این کار قابلیت chroot را از یوزرها سلب کردیم. برای Jail کردن یوزرها اپشن زیر باید مقدار **YES داشته باشد قابلیت chroot غیر فعال می شود.**

chroot_local_user=YES

اگر هم بخواهیم قابلیت chroot برای بعضی از یوزرها فعال شود کافی است اسمی آنها را در یک فایل قرار داده و در خط زیر آدرس دهی کنیم .

chroot _list_enable=YES
chroot_list_file=/etc/vsftpd/chroot_list

بر طرف کردن Error 500

در centos سری 6 ممکن است در هنگام login یوزرهای local به آنها error 500 نشان داده شود . ممکن است نتوانند به دایرکتوری Home خود بروند یا اجازه write پیدا نکنند. یا فرضا ما یکسری از قابلیت ها را فعال می کنیم ولی در عمل کار نکنند. دلیل آن هم به خاطر عدم پیکربندی متغیرهای بولین Selinux است. این متغیرها را با دستور زیر می توان مشاهده کرد :

#getsebool -a

یکی از مهمترین این متغیرها allow_ftpd_anon_write=off می باشد. مثلا اگر در فایل کانفیگ اجازه رایت به یوزرهای anon داده شده باشیم تا این متغیر on نشود اجازه write به یوزرهای anon داده نمی شود .یکی دیگر از مهمترین متغیرها ftp_home_dir = off می باشد . این متغیر به یوزرهای local اجازه می دهد از طریق FTP وارد دایرکتوری home خود بشوند. فعال بودن این متغیر است که باعث می شود که500 error برای یوزرها نمایش داده شده و از رفتن به دایرکتوری خانگی انها ممانعت به عمل آید. با دستور setsebool میتوان مقدار این متغیر ها را تغییر داد:

#setsebool -P ftp_home_dir=1

در Selinux برای هر سرویس مقدار زیادی متغیر وجود دارد که باید بعد از راه اندازی هر سرویس متغیرهای آن را پیکربندی کنیم . بزرگترین اشتباه آن است که Selinux را خاموش کنیم . Selinux در سه مد کار می کند:

1. enforcing : این مد بالاترین درجه امنیت در Selinux را دارا میباشد. اگر این مد را فعال کنیم تمام ماژول های امنیتی سیستم enable می شود.
2. permissive : در این مد Selinux فعال نیست و چیزی را denay نمی کند اما از همه چیز log برداری میکند .
3. disable : با فعال کردن این مد Selinux کاملا غیر فعال می شود.

فایل کانفیگ Selinux در مسیر /etc/selinux/config قرار دارد. برای تغییر در مدهای آن این فایل را باز کرده و در مقابل کلمه SELINUX مد مربوطه را وارد می کنیم و برای اعمال شدن آن حتما باید سیستم را یکبار ریست کنیم . از دستورات زیر هم جهت تغییر مد آن می توان استفاده کرد :

#setenforce 0
#echo 0 > /selinux/enforce

با دستور زیر هم می توان از وضعیت Selinux و مدهای کاری آن کسب اطلاع کرد :

#setstatus

ایجاد Mullti Homing در FTP

به طور معمول بر روی هر سیستم فقط یک سرویس دهنده ftpراه اندازی می شود در حالی که با Vsftpd می توان چندین سرویس دهنده مستقل FTP را روی یک سرور راه اندازی کرد. فرض کنید یک سرور داریم که هم زمان به اینترنت و شبکه داخلی سرویس می دهد و می خواهیم یک ftp به کاربران اینترنتی و یک ftp دیگر به کاربران داخلی سرویس بدهد. در چنین شرایطی از خاصیت mullti homing استفاده می کنیم . برای هر سرور باید یک فایل کانفیگ جداگانه با نام منحصر به فرد، در زیر دایرکتوری /etc/vsftpd ایجاد کنیم. برای هر کارت شبکه یک آدرس اختصاصی تنظیم کرده و در هر فایل کانفیگ یکی از آنها را وارد می کنیممهمترین گزینه ای که در فایل کانفیگ باید آورده شود listen_address است.

طبق توصیه اکید ردهت مکان ذخیره لاگ هر کدام از این سرورها باید با دیگری فرق داشته باشد. قبلا گفته شد که vsftpd به دو صورت log برداری میکند که xferlog نحوه پیش فرض لاگ گرفتن این سرویس می باشد. به طور مثال می توانیم به یک کارت شبکه سرور ۲ IP اختصاص داده و هر IP را مختص یک سرویس دهنده فایل قرار دهیم به این کار Virtual Host گفته می شود. این موضوع سوال المپیاد لینوکس می باشد. برای این کار باید از فایل کانفیگ اصلی یک کپی با یک نام دلخواه ایجاد کرده و تنظیمات مربوطه را درون آن ایجاد می کنیم. دلخواه ایجاد کرده و آدرس IP مورد نظر و پورت دلخواه را درون آن وارد کنیم.

#vi /etc/vsftpd/vsftpd2.conf
listen=YES
local_enable=NO
anonymous_enable=YES
write_enable=YES
anon_max_rat=YES
an0n_root=/opt
listen_address=192.168.1.1
listen_port=2020

#vsftpd /etc/vsftpd/vsftpd2.conf

با این دستور فقط فایل vsftpd2.conf ریست می شود و بقیه ftpها به کارشان ادامه می دهند. چون نیازی نیست همه آنها با هم ریست شوند پس بهتر است فقط فایل کانفیگ مربوطه را ریست کنیم.

نمونه ای از فایل ایجاد شده در Xinetd برای سرویس Vsftpd

همانطور که گفته شد سرویس vsftpd به صورت standalone کار می کند حال اگر بخواهیم این سرویس زیر نظر xinetd اداره شود باید گزینه ???=listen را برابر YES قرار داده و در زیر دایرکتوری xinetd یک فایل کانفیگ بسازیم.

```
servicevsftp
   {

socket_type       =stream
user              =root
server            =/usr/sbin/vsftpd
server_args       =/etc/vsftod/vsftpd.conf
nice              =10
disable           =no
flags             =ipv4

}
```

استفاده از دستور ftp به عنوان نرم افزار کلاینتی

نرم افزار پیش فرض کلاینتی اکثر توزیع های لینوکس دستور ftp می باشد که برای کپی، انتقال، rename، حذف یک فایل یا فولدر و یا ساختن یک فولدر جدید و همچنین تغییر سطح دسترسی فایل ها و فولدر ها می تواند از آن استفاده کرد. برای جلوگیری از سرقت اطلاعات بسیار بهتر است که همواره از sftp یا همان secure ftp استفاده کنید که انتقال امن را فراهم می آورد. اگر FTP به صورت امن راه اندازی نشود اطلاعات را به صورت clear Text رد و بدل می کند.دستور ftp یک دستور تعاملی است یعنی یک چیزی به آن می دهیم و یک چیزی به ما بر می گرداند و برای هر کاری باید یک دستور به آن بدهیم. در ftp به طور پیش فرض نمی توان به صورت anon به سرور متصل شد بلکه باید حتما نام یوزر local را وارد کرد این دقیقا بر عکس دستور lftp می باشد.

نکته : اگر اول کامندی از علامت ! استفاده کنیم یعنی این دستور را روی سرور اجرا نکن بلکه باید ان را روی سیستم local اجرا کند.
نکته : زمانی که به سرور لاگین می کنیم یکسری کد به همراه پیامهایی به نمایش در می اید. این کدها از قبل تعریف شده هستند و برای ثبتlog استفاده می شوند.

برای اتصال به یک سرویس دهنده فایل با استفاده از دستورftpبه شیوه زیر عمل کنیم :

```
ftp ftp.example.com
username
password
```

به جای ftp.example.com می بایست hostname سرور مربوطه و یا نام یکی از دامنه های مستقر بر روی آن را بنویسید و برای ورود اطلاعات اکانت کاربری ftp متعلق به سرور مقصد را وارد نمائید. با دستور ftp نمی توان همزمان هم احراض هویت و هم اتصال برقرار کرد اما در lftp می توان با یک دستور هم لاگین کرده و احراض هویت کنیم.

به طور مثال ، مراحل زیر را مشاهده می فرمائید:

```
Trying 87.51.34.132...
Connected to ftp.freebsd.org.
220 ftp.beastie.tdk.net FTP server (Version 6.00LS) ready.
Name (ftp.freebsd.org:vivek): ftp
331 Guest login ok, send your email address as password.
Password:
230 Guest login ok, access restrictions apply.
Remote system type is UNIX.
Using binary mode to transfer files.
ftp>
```

از این پس، به سبب اینکه از پروتکل ftp بهره می گیرید، اعلان پرامپت شما مطابق زیر خواهد بود:

```
ftp>
```

برای نمایش فایل ها و فولدر ها می توانید از دستور زیر استفاده نمائید:

```
ftp>ls
```

برای مثال احتمالا، اطلاعاتی شبیه به اطلاعات زیر را دریافت می کنید:

```
229 Entering Extended Passive Mode (|||60692|)
150 Opening ASCII mode data connection for '/bin/ls'.
total 10
drwxrwxr-x  2 0    5     512 Jul 19  2007 .snap
drwx------  2 0    0    2048 Jul 19  2007 lost+found
drwxr-xr-x  3 1006  1006   512 Sep 21  2009 pub
drwxr-xr-x  3 1006  1006   512 Jun  5  2007 sup
drwxr-xr-x  4 1006  0     512 Sep 18  2009 www
226 Transfer complete.
ftp>
```

دقت نمائید که ستون آخر نمایش دهنده نام فایل ها و فولدر ها می باشد.

برای ورود به یک فولدر دیگر می توانید از دستور زیر استفاده کنید:

```
ftp> cd folder-name
```

برای دریافت یک فایل می توانید از دستور get مطابق مثال زیر استفاده کنید. اگر بخواهیم فایل های دانلودی در یک مسیر مشخص ذخیره شوند باید در محیط دستور ftp ابتدا به آن مسیری که روی سیستم

local قرار دارد lcdکرده و سپس اقدام به دانلود فایل ها نمائیم و اگر از سوئیچ c– استفاده کنیم در هر مرحله از دانلود یا اپلود ارتباط ما قطع شود در ارتباط بعدی از باقی مانده کار شروع به دانلود یا آپلود می کند :

```
ftp> get–c resume.pdf
```

و مطابق ذیل مشاهده خواهید کرد که فایل مربوطه دانلود می گردد:

```
local: resume.pdf remote: resume.pdf
229 Entering Extended Passive Mode (|||55093|)
150 Opening BINARY mode data connection for 'resume.pdf' (53077 bytes).
100%
|***********************************************************************
*******| 53077      12.58 KiB/s    00:00 ETA
226 Transfer complete.
53077 bytes received in 00:04 (12.57 KiB/s)
```

اگر در همین زمان می خواهید، محل دایرکتوری خود در سیستم Local و مبدا را تغییر دهید، دستور زیر مفید خواهد بود:

```
ftp>lcd /path/to/new/dir
```

مثل:

```
ftp>lcd /tmp
```

حتی می توانید با دستور زیر محل دایرکتوری خود در سرور اصلی مشخص نمائید:

```
ftp>lpwd
```

برای دریافت چندین فایل می توانید از دستور زیر استفاده نمائید:

```
ftp>mget *
```

و یا :

```
ftp>mget *.jpg
```

برای حذف یک فایل:

```
ftp>deletefileName
ftp> delete output.jpg
```

و اما دستور زیر که شاید برای خیلی ها تازگی داشته باشد؛ اگر می خواهید فایلی را در سرور از طریق shell آپلود نمائید، یعنی به سروری که متصل شده اید منتقل کنید، کافی است دستور زیر را استفاده کنید:

```
ftp> put FileName
```

مثلا می خواهید فایل logo.jpg را از کامپیوتر محلی خود به سرور از طریق shell انتقال دهید:

```
ftp> put logo.jpg
```

و برای آپلود چندین فایل:

```
ftp>mput *
ftp>mput *.pl
```

اضافه کردن یک دایرکتوری:

```
ftp>mkdirdirName
```

حذف کردن یک دایرکتوری:

```
ftp>rmdirdirName
```

و در نهایت، برای خروج از ftp می توانید دستورات زیر را بکار ببرید:

```
ftp> quit
```

نکته : NOT دستور cd کامند lcd است.

فرق بین مد باینری و اسکی

کلا در دنیا دو نوع کلی فایل وجود دارد 1- اسکی 2 – باینری

فایل های اسکی فایل هی text base می باشند مثل فایل های php,asp,html,pdf و کلا هر فایلی که بتوان محتوای آن را خواند فایل اسکی است ، به غیر از این ، تمام فایل ها باینری هستند مثل عکس ، فیلم ، آهنگ و ...

دستور ftp می تواند در دو مد اسکی و باینری فایل ها را منتقل کند. اگر مد انتقال فایل با فایل دریافتی هم خوانی نداشته باشد فایل ها در مقصد برای باز شدن دچار مشکل خواهند شد. پس اگر فایل باینری باشد باید در مد باینری و اگر اسکی باشد باید در مد اسکی نقل و انتقال صورت پذیرد. برای تغییر مد کافی است کلمه ascii را تایپ کنیم . به همین سادگی مد ترانسفر تغییر می کند.

```
ftp>ascii
    200 switching to Ascii mode
```

نکته : دستور ftp مانند سرویس دهنده آن در دو مد active و passive ارتباط برقرار می کند.

SCP یک جایگزین امن برای FTP

از دیدگاه شبکه، سرویس FTP سرویس امنی نیست، زیرا نام کاربری، کلمه عبور و داده ها همگی بدون هیچ گونه رمزنگاری برروی شبکه مبادله می شوند. شکل امن این سرویس SFTP وSCP هستند، که به عنوان جزئی از بسته Openssh در دسترس بوده و به شکل پیش فرض در سیستم های Redhat و CentOS نصب می باشد. به خاطر داشته باشید که SCP برخلاف FTP قابلیت پشتیبانی از بارگیری بی نشان (Anonymous Download) را دارا نیست. فرمان SCP در لینوکس، قالبی همانند فرمان cp را داراست. اولین پارامتر فایل مبدا و دومین پارامتر فایل مقصد را مشخص می کند. در هنگام کپی کردن یا گذاشتن فایل ها در سرویس دهنده SSH ، کاربر باید توسط scp وارد سرویس دهنده شود که برای این کار باید نام سرویس دهنده، نام کاربری و کلمه عبور را با موفقیت به- عنوان آرگومان های ورودی به آن ارسال کند.

پس از این فایل موردنظر با پیشوندی از نام کاربری و سرویس دهنده که با یک @ از یکدیگر جدا شده اند، در سمت سرویس دهنده پردازش می شود. قالب مربوط به این موضوع بدین شکل است:

username@servername:filename
username@servername:directoryname

به طور مثال فرض کنید نیاز به کپی کردن فایل etc/syslog.conf/ بر روی سرویس دهنده ای با آدرس 192.168.1.100 و نام کاربری Peter داریم. بدین منظور از قالب

etc/syslog.conf/: peter@192.168.1.100

استفاده می کنیم . در صورت تمایل به کپی برداری از کل شاخه etc/ قالب فوق بدین شکل تغییر می یابد.

/etc/:1.100 . Peter@192.168

نکته: جهت تهیه و نصب نسخه ویندوزی فرمان scp در سمت کاربر، می توانید نرم افزار WinScp را از آدرس زیر تهیه نمایید:

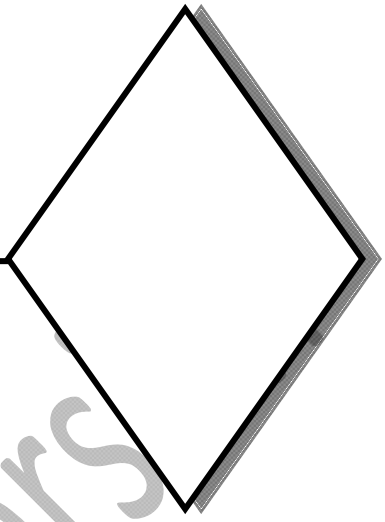
http://winscp.vse.cz/eng

متداولترین کدهای وضعیت FTP

متداولترین کدهای وضعیت FTP به همراه مفهوم هریک در جدول زیر نشان داده شده است .

کدهای وضعیت سری 100	
110	Restart reply
120	Service ready in x minutes
125	Connection currently open, transfer starting
150	File status okay, about to open data
کدهای وضعیت سری 200	
200	Command okay
202	Command not implemented, superfluous at this site
211	System status/help reply
212	Directory status
213	File status
214	System Help message
215	NAME system type
220	Service ready for next user.
221	Service closing control connection. Logged off where appropriate
225	Data connection open; no transfer in progress.
226	Closing data connection. Requested action successful
227	Entering Passive Mode
230	User logged in, continue
250	Requested file action okay, completed
257	"PATHNAME" created.
کدهای وضعیت سری 300	
331	User name okay, need password.
332	Need account for login
350	Requested file action pending further information.
کدهای وضعیت سری 400	
421	Service not available, closing control connection.

425	Can't open data connection
426	Connection closed; transfer aborted.
450	Requested file action not taken. File not available - busy etc..
451	Request aborted: error on server in processing.
452	Requested action not taken. Insufficient resources on system
کدهای وضعیت سری 500	
500	Syntax error, command unrecognized
501	Syntax error in parameters or arguments.
502	Command not implemented.
503	Bad sequence of commands
504	Command not implemented for that parameter.
530	Not logged in.
532	Need account for storing files
550	Requested action not taken. File unavailable
552	Requested file action aborted. Exceeded storage allocation
553	Requested action not taken. File name not allowed
مفهوم برخی از کدهای متداول	
226	دستور بدون هیچگونه خطائی اجراء گردید .
230	زمانی این کد نمایش داده می شود که یک سرویس گیرنده رمز عبور خود را به درستی درج و عملیات login با موفقیت انجام شده باشد .
231	کد فوق نشاندهنده دریافت username ارسالی سرویس گیرنده توسط سرویس دهنده می باشد و تائیدی است بر اعلام وصول Username (نه صحت آن) .
501	دستور تایپ شده دارای خطاء گرامری است و می بایست مجددا" دستور تایپ گردد .
530	عملیات login با موفقیت انجام نشده است . ممکن است Username و یا رمز عبور اشتباه باشد .
550	فایل مشخص شده در دستور تایپ شده نامعتبر است .



فصل دوم

راه انداز یSSH در لینوکس

آشنایی با پروتکل SSH

مقدمه :

در سال 1995 یک دانشجوی دانشگاه هلسینکی به نام Tatu Ylönen پس از آنکه اطلاعات مهمی مثل رمز و نام های کاربری در شبکه دانشگاه مورد Sniff قرار گرفت به فکر ایجاد یک شبکه امن افتاد که این فکر در نهایت منجر به ایجاد یک Shell امن شد که جایگزینی برای ftp , rsh , rcp , rlogin , telnet شد. البته این دستورات هنوز در لینوکس وجود دارند اما با وجود SSH دیگر کسی از آنها استفاده نمی کند و تمامی اینها با دستورات scp و ssh جایگزین شدند.

SSH مخفف Secure Shell است. SSH یک پروتکل ارتباطی امن بر پایه TCP/IP بین سرویس دهنده و سرویس گیرنده است که با رمز گذاری داده ها از افشای اطلاعات در طول مسیر جلوگیری کرده و یک کانال امن در سیستم عامل سرور برای دستیابی به خط فرمان برای کلایتنت ایجاد می کند.

کلمه Shell ممکن است این تصور را ایجاد کند که SSH یک مفسر فرمان است اما این کاملا اشتباه بوده بلکه یک پروتکل اتباط امن می باشد. الگوی اولیه رمزنگاری در سرویس ssh که در سال 1995 ارائه گردید در زمان خودش ابزاری مناسب محسوب می شد ولی با گذشت زمان محدودیت هایی در استفاده از آن پدیدار گشت که به جهت رفع این محدودیت ها نسخه دوم این سرویس ارائه شد. همواره سعی کنید تا با تنظیم عبارت Protocol در فایل های پیکربندی سیستم، خود را ملزم به استفاده از نسخه دوم این سرویس کنید. استفاده از SSHمحدودیتهایی نظیر لایسنس و پرداخت هزینه را در بردارد ، لذا برای رفع این محدودیت گروه OpenBSD شروع به ارائه موازی نسخه جدیدی به صورت رایگان نمود که نام این محصول را OpenSSH گذاشتند. با خرید لایسنس شرکت Tectia که ارائه کننده SSH تجاری می باشد علاوه بر پشتیبانی ازکاربر ، از صفحه مدیریت تحت وب SSH بهره مند می شوید ولی چنین مزایایی در OpenSSH وجود ندارد. راز محبوبیت پروتکل SSH کدگذاری شبکه ، ایجاد تونل امن و پشتیبانی از انواع متد های دیگر ارتباط امن می باشد.

مزایای استفاده از SSH

(نسبت به شیوه های قدیمی ارتباط از راه دور)

1. **رمز گذاری دادها : (Encryption Data)**

همانطور که گفته شد نیاز به یک اتصال امن بین سرور و کلاینتها و جلوگیری از Sniff ، مهمترین دلیل استفاده از SSH می باشد.

2. **بررسی یکپارچگی دادها (Data Integrity)**

این خاصیت برای جلوگیری از حمله های Insertion and Replay Attacks بسیار مفید می باشد. لازم به ذکر است رمزگذاری داده ها بدون استفاده از Session ID نمی تواند از حملات replay attacks جلوگیری کند. از نسخه دو این پروتکل این قابلیت اضافه شد تا پکتها در مسیر ارسال جایگزین و یا شبیه سازی نشوند. در این نوع حمله هکر دیتای تبادلی در نشست را مانیتور نمی کند بلکه بلکه مثل نرم افزارهای Keylogger خروجی صفحه کلید را مانیتور نموده و با مقایسع پکتهای تایپ شده با ترافیک جاری SSH متوجه کارکترهای خاص تایپ شده می شود.

3. **قابلیت فشرده سازی : (Compression)**

این پروتکل علاوه بر رمز نگاری ، اطلاعات ارسالی را فشرده می کند که این کار در ارتباطات کم سرعت بسیار مفید خواهد بود.

4. **عدم اتصال به سرور جعلی : (Prevent Impersonation of host)**

در یک انصال SSH هنگام اتصال به سرور، هویت سنجی صورت می گیرد و اگر یک ماشین با مشخصات سرور در مسیر کلاینت قرار گرفته باشد امکان میزبانی کلاینت و یا بالعکس را ندارد. در حالی که در پروتکل های قدیمی تر مثل Telnet این اتفاق اجتناب ناپذیر است. این نوع حمله به حمله مرد میانی موسوم است. (Man-In-The-Middle-Attack or MITM Attack).

5. **لاگ فایل : (Log Access)**

SSH امکان فعال و یا غیر فعال شدن فرایند تهیه لاگ فایل ها را دارد با فعال شدن این امکان در مواقع بروز مشکل مدیر سیستم بعد از بروز خطا اولین موردی که برای رفع مشکل بررسی میکند لاگ فایل ها می باشد.

6. امکان استفاده از X11 Applications

SSH این قابلیت را دارد که برنامه های دیگر مثل نرم افزارهای گرلفیکی را کد گذاری کند. به قابلیت Port Forwarding هم می گویند. از این قابلیت برای Tunneling هم استفاده می شود.

7. موجود بودن کامندهای موسوم به r-Command

تمام امکانات کامندهای موسوم به r-Command در SSH وجود دارد. به عنوان مثال از سرور 1 به سرور 2 دستور date را اجرا می کنیم :

```
skywan13@localhost ~]$ ssh userx@x.x.x.x date
: ssh userx@x.x.x.x 's password
Tue Sep 21 18:11:28  IRDT  2014
```

علاوه بر کاربردهای رایج این پروتکل ، انعطاف پذیری بر حسب نیاز موجب محبوبیت این پروتکل در بین کاربران و متخصصان کامپیوتر شده است.

نصب و راه اندازی سرویس OpenSSH

عموما به شکل پیش فرض سرویس Openssh در زمان نصب سیستم عامل نصب می شود. همچنین از آنجا که ssh و scp جزیی ازیک برنامه هستند، هر دو از یک فایل پیکربندی استفاده کرده و توسط سرویس SSH مدیریت می شوند .بسته های rpm سرویس SSH را به راحتی می توان ازمنابع این نوع بسته ها در Internet تهیه کرد . معتبرترین مرجع جهت تهیه بسته های مربوط به این سرویس سایت ssh.com است که در آن شما قادر خواهید بود نسخه های تجاری و غیرتجاری سرویس SSH را به راحتی تهیه کنید. این سرویس از طریق کامپایل کد منبع آن و ابزار apt-get در سیستم های مبتنی بر debian نیز به راحتی قابل تهیه و استفاده است. همچنین کاربران فدورا با استفاده از yum می توانند این بسته را نصب کنند هرچند به صورت پیش فرض این بسته روی اکثر توزیعات لینوکسی نصب هست. SSH با حروف بزرگ به طور کلی به پروتکل SSH اطلاق میشود و ssh با حروف کوچک به نرم افزر سمت کلاینت گفته می شود که برای اتصال به سرور به کار می رود. پیش نیاز نصب SSH پکیج های zlib و OpenSSL است که در صورت استفاده از Yum این نیازمندیها به صورت اتوماتیک نصب خواهند شد. SSH تحت نظرinit اداره می شود.

ابتدا باید از نصب بودن پکیج Openssh اطمینان حاصل کنیم لذا با دستور زیر از سیستم query می گیریم :

#rpm -qa | grep openssh

در صورت نصب نبودن ، در سیستم های ردهت جهت نصب openssh از yum استفاده می کنیم :

#yum –y install openssh

بعد از نصب ، باید اطمینان حاصل کنیم که آیا پکیج Openssh بر روی سیستم نصب شده است یا خیر لذا با دستور زیر از سیستم query می گیریم :

#rpm -qa | grep openssh

سپس با دستور زیر شاخه ها و مسیرهایی که فایل های این سرویس در آن ایجاد شده است را چک می کنیم :

#rpm -ql openssh

و با این دستور هم اطلاعات لازم را در مورد پکیج این سرویس به دست می آوریم :

#rpm -qi openssh

سپس با دستور chkconfig مشخص می کنیم در چه runlevel هایی فعال باشد :

#chkconfig openssh on

و در انتها سرویس را reset می کنیم :

#service sshd restart

با دستور netstat از غیرقابل استفاده بودن آن توسط سایر برنامه های سیستم اطمینان حاصل کنیم:

 # netstat -an| grep 435

نکته : با نصب openssh چهار پکیج روی سیستم نصب می شوند که این دو پکیج مهمترین آنها هستند :

openssh-server openssh-client

همانطور که گفته شد SSH یک پروتکل کلاینت سروری است و در هر دو طرف یعنی سرور و کلاینت لینوکسی باید پیکر بندیهای لازم صورت پذیرد.

در سمت سرور در مسیر /etc/ssh/ و در سمت کلاینت لینوکسی در مسیر /home/user/.ssh/ باید تغییراتی اعمال شود . از فایلهای مسیر /etc/ssh/ برای پیکربندی سرور SSH استفاده می شود که این تنظیمات global بوده و به همه کلاینتها و یوزرها اعمال می شود. اما تغییراتی که در /home/user/.ssh/ انجام می دهیم برای تحت تاثیر قرار دادن عملکرد کامندهای ssh و scp است و فقط در سیستم کلاینت اعمال شده و local می باشد.

در این مقاله ابتدا تنظیمات سمت سرور را مرور کرده سپس به تنظیمات سمت کلاینت خواهیم پرداخت.

بررسی فایل های موجود در /etc/ssh

در زیر این دایرکتوری فایل های پیکربندی و الگوریتم های این پروتکل قرار دارد که مهمترین آنها در زیر توضیح داده شده است :

sshd _config
ssh _config
ssh _host_key
ssh _host_dsa_key
ssh _host_rsa_key
moduli

sshd_config : این فایل مهمترین فایل این دایرکتوری است . با این فایل سرویس SSH را پیکربندی می کنیم. تنظیمات این فایل global بوده و همه یوزرها اعمال می شود.

ssh_config : با این فایل کلاینتهای ssh مورد پیکربندی قرار می گیرند. اگر در این فایل تغییراتی اعمال کنیم به همه یوزرها اعمال می شود.

ssh _host_key : این فایل کلید SSH ورژن یک می باشدکه از الگوریتم خاصی برای رمز نگاری استفاده نمی کند.

ssh _host_dsa_key
ssh _host_rsa_key
این فایل ها کلید SSH ورژن 2 می باشند که از الگوریتمهای rsa و dsa برای رمز نگاری استفاده می کند.

moduli: اطلاعات dh که در معاوضه کلیدها بین طرفین اسفاده می شود در این فایل قرار می گیرد.

پیکربندی سرویس SSH

در این قسمت به تشریح بعضی از قسمتهای فایل پیکربندی سرور SSH می پردازیم. برای اطلاع دقیق از تمامی آپشن های این سرویس به man sshd_conf رجوع کنید :

Port 445

پورت پیش فرض سرویس SSH پورت tcp 22 می باشد. SSH با این پورت روی تمام کارت های شبکه به حالت listen میرود. هر گاه زمانی احساس کردید افرادی قصد نفوذ به سیستم شما، از طریق پورت شناخته شده ای مثل 22 را دارند، می توانید با تغییر آن ، به پورتی که تداخلی با برنامه های کاربردی موجو د در سیستم ندارد، از این امر پیشگیری کنید. این کار را می توان تنها یک پیشگیری اولیه محسوب کرد، زیرا برنامه هایی در شبکه جهت تشخیص پورت هایی که هم اکنون در حال اجرای سرویس ssh هستند، نیز وجود دارند.

AddressFamily any

این خط می تواند سه مقدار داشته باشد. مقدار این خط مشخص کننده این است که از چه ورژن IP پشتیبانی کند.

مقادیر این خط :

inet : مشخص کننده IP ورژن 4 است.

int6 : مشخص کننده IP ورژن 6 است.

any : یعنی از هر دو ورژن IP پشتیابنی کند.

LitenAddress 192.168.1.1

Ssh به طور پیشفرض با تمام IPها ارتابط برقرار می کند و اجازه ارتباط می دهد در اینجا میتوانیم کارت شبکه خاصی را به عنوان کارت اختصاصی SSH آدرس دهی کنیم. این خط مربوط به IP ورژن 4 است.

ListenAddress : :

در IP ورژن 6 مقدار این خط معنی همه را می دهد.

HostKey /etc/ssh/ssh_host_rsa_key
HostKey /etc/ssh/ssh_host_dsa_key
با این آپشن ها مسیر کلید های rsa و dsa را معین می کنیم که البته نیازی به تعویض این مسیرها نیست.

KeyRegenerationInterval 1h
سرور SSH به طور پیش فرض هر یک ساعت کلیدهای تولید شده روی سرور را تولید مجدد می کند تا از حملات Capture data و یافتن الگوریتم جلوگیری کند.

ServerkeyBits 2048
طول کلیدهای رمزنگاری تولید شده به طور پیش فرض 1024 بیت است که برای امنیت بیشتر بهتر است بر روی 2048 تنظیم شود.

SyslogFacility AUTHPRIV
این خط مشخص میکند log هر کاربری که با ssh به سرور لاگین کرده و یوزر پسورد وارد کند ثبت شود.

LogLevel INFO
این خط می تواند دو مقدار INFO و DEBUG داشته باشد که سطوح log برداری را مشخص می کنند.

PermitRootLogin no
به صورت پش فرض یوزر root اجازه Remote access از طریق ssh را دارد که باید این اجازه از آن گرفته شود. با قراردادن no این اجازه را از آن می گیریم.

LoginGreaceTime 2m
این زمان مشخص می کند کلاینت که session برقرار کرده و صفحه لاگین را در اختیار دارد 120 ثانیه وقت دارد یوزر پسورد را تایپ و به سرور لاگین نماید اگر در این مدت اقدامی صورت ندهد کانکشن آن قطع شده و صفحه لاگین بسته می شود.

MAXAuthTrise 6

اگر به صورت پیش فرض یوزری 6 مرتبه پسورد را اشتباه وارد کند ، ssh صفحه لاگین را از او گرفته کانکشن را قطع می نماید.

Allow and Deny
Allow Users
Deny Users
Allow Groups
Deny Groups

در حالت پیش فرض تمامی یوزرها می توانند با استفاده از یوزرشان در سرور به آن ssh زده ولاگین کنند. اگر بخواهیم به یوزر یا گروهی اجازه دسترسی به ssh را داده یا دسترسی آنها را محدود کنیم باید در جلوی این آپشن ها یا نام یوزرها و گروه ها را جدا از هم وارد کرده و یا در لیستی جداگانه وارد و در اینجا آدرس فایل مربوطه را وارد کنیم .

AuthorizedkeysFiles .ssh/authorized_keys

این فایل جهت احراض هوی keybase مورد استفاده قرار می گیرد که بر روی دایرکتوری Home هر یوزر لینوکس قرار دارد. این خط مسیر این فایل را مشخص می کند.

PasswordAuthentication yes
PubkeyAuthentication yes

در SSH دو نوع احراض هویت وجود دارد ، یمی بر اساس Password و دیگر ی بر اساس Keybase می باشد. احراض هویت پیش فرض از نوع پسورد است. خط اول احراض هویت را بر اساس پسورد و خط دوم احراض هویت را بر اساس کلید فعال می کند.

PermitEmptyPassword no

بعضی از اکانتها دارای پسورد نیستند برای جلوگیری از لاگین شئن چنین اکانت هایی مقدار این خط را برابر no قرار می دهیم.

UsePam yes or no

سرویس SSH می تواند توسط دو مکانیزم امنیتی Pam و tcp_wrapper کنترل می شود. فعال کردن این گزینه باعث می شود کنترل امنیتی این سرویس تحت اختیار Pam قرار می گیرد که در این صورت باید فایل زیر را مورد پیکربندی قرار دهیم. /etc/security/access.conf و اگر بخواهیم با مکانیزم tcp_wrapper مورد کنترل قرار بگیرد باید فایل /etc/host.allow را کانفیگ کنیم.

جهت کار کردن با این دو فایل حتما باید راهنمای آنها را مورد مطالعه قرار دهیم. اگر بخواهیم بفهمیم کدام سرویس توسط tcp_wrapper کنترل می شود از این دستورات استفاده می کنیم.

#whereis sshd
#ldd /usr/sbin/sshd

دستور ldd تمامی ماژول هائی که یک سرویس از آن استفاده می کند را نشان می دهد.خروجی این دستور لیست تمام ماژل های sshd را نشان می دهد. سومین ماژول نشان داده شده در خروجی ، ماژول libwarrp.so است. هر سرویسی که این ماژول را داشته باشد یعنی توسط tcp_wrapper کنترل میشود.

Accept ENV

این عبارت متغیرهایی را نشان می دهد که سرور ssh اجازه Export کردن آنها بر روی کلاینت را دارد.

X11Forwarding yes

این امکان را به ما می دهد تا بعد از اتصال به سرور SSH به صورت گرافیکی بتوانیم سرور را پیکربندی کنیم. بعد از اتصال ، ما با یک صفحه مشکی رنگ مواجه می شویم ، اما می توانیم از دستوراتی استفاده کنیم که دارای یک محیط گرافیکی هستند مثل دستورات system-config-network , rconf , setup . این دستورات کنسول گرافیکی خاصی را بر روی سرور اجرا می کنند. بهتر است این گزینه بر روی no تنظیم شود.

Banner /etc/ssh/banner

می توانیم پیامی طراحی کرده و در مسیر گفته شده قرار دهیم تا در موقع اتصال هر یوزر به SSH این بنر به نمایش در بیاید.

Printmotd yes

اگر این خط برابر yes باشد می توان پیامی طراحی و در مسیر /etc/motd قرار داد تا در موقع اتصال آن را نمایش دهد.

Protocol 2

Ssh دارای دو ورژن 1 و 2 می باشد ورژن 1 دارای آسیب پذیری های mitmاست و نباید به هیچ عنوان از آن استفاده شود.

ClientAliveInterval 600
ClientAliveCountMax 0

می توان یک حالت Idle TimeOut برای یوزرها ایجاد کرد تا اگر بین کار آنها فاصله ای بیافتد و عملیاتی صورت نگیرد باعث Logout در یوزر شودبا اضافه کردن این دو خط و مقادیر مورد نظر این امکان فعال میشود.

HostBasedAuthentication no

تا وقتی که این ویژگی فعال باشد یک یوزر از هاست خودش می اوند به هاست دیگری در شبکه هم لاگین کند .

IgnoreRhosts yes

فعال بودن این خط باعث می شود یوزرها نتوانند به فایل های shosts. و rhosts. دسترسی داشته باشند.

نکته : در حالت پیش فرض یوزرها امکان دسترسی به دایرکتوری هایی به جز دایرکتوری اصلی خود را دارندو می توانند به دایرکتوری هایی مانند bin,etc,..... دسترسی پیدا کنند. با استفاده از سیستم عامل هایی که بر پایه chroot هستند و یا ابزاری مانند rssh می توان ssh را در برابر دیگر یوزرها ایمن کرد.

استفاده از SSH جهت اتصال به یک کامپیوتر در شبکه

استفاده از ssh بسیار شبیه telnet است. جهت اتصال به یک ماشین دیگر در شبکه یا یک سرور SSH تحت یک کاربر دلخواه از آن ماشین از سوئیچ l- استفاده می کنیم. در اینجا چند مثال جهت اتصال به یک سرویس دهنده SSH ارائه شده که به بررسی آنها می پردازیم :

با استفاده از این فرمان تحت کاربر root به کامپیوتر 192.168.1.1 در شبکه متصل می شویم.

 # ssh 192.168.1.1

حال اگر بخواهیم تحت یک کاربر دیگر این اتصال را انجام دهیم، از یکی از قالب های زیر می توانیم استفاده کنیم :

 # ssh -l ali 192.168.1.1
 # ssh ali@192.168.1.1

و اگر بخواهیم برروی پورتی غیر از پورت 22 عملیات Login را انجام دهیم، دستور فوق بدین شکل تغییر پیدا می کند:

 # ssh -P 435 ali@192.168.1.1

و در صورتی که مایل باشیم ، دستوراتی مانند system-config-network که محیط گرافیکی دارند را اجرا کنیم از سوئیچ X– استفاده می کنیم :

 # ssh -X ali@192.168.1.1

در اولین مرتبه ای که از طریق کلاینت لینوکسی با سروری ارتباط ssh برقرار می کنیم یک هشدار دریافت خواهیم کرد، دلیل آن هم این است که سرور مقابل می خواهد بر روی سیستم ما یک کلید آپلود کند.
یکی از امکانات جالب ssh قابلیت ورود و اجرای فرامین منفرد در یکی از سیستم های شبکه است. برای این کار کافی است فرمان مورد نظر را در یک جفت کوتیشن، در انتهای فرمان ssh قرار دهیم. در مثال زیر یک

کاربر قصد دارد به نسخه کرنل موجود برروی سرویس دهنده 192.168.1.1 پی ببرد که برای این کار فرمان uname -a را بر روی سرویس دهنده اجرا می کند و بلافاصله خروجی آن به نمایش در می آید :

```
# ssh   ali@192.168.1.1 "uname -a"
Linux yadi 2.6.8-1.521 #1 Mon Aug 10:10:17 EDT 2004 i686 i686 i386
```

مبحث SSH Client

در این قسمت از آموزش ، دیگر با سرور SSH کاری نداریم بلکه می خواهیم تنظیمات ssh client را بررسی کنیم. زمانی که برای اولین بار از طریق ssh به یک سرور یا ماشینی متصل می شویم، پیامی مبنی بر اینکه ماشین ما توسط سیستم مقصد شناخته شده نیست را دریافت خواهیم کرد. در همین زمان درخواستی جهت ذخیره سازی یک نسخه از کلیدهای شناسایی، ssh سرور مقصد برروی کامپیوتر خودمان دریافت می کنیم که با تائیدآن یک RSA key fingerprint که همان کلید Public سرور مقصد است بر روی سیستم Local آپلود می شود. این رویه را در زیر می توانید مشاهده کنید :

روال ذخیره سا زی کلید:

 # ssh 192.168.1.1
The authenticaty of host 192.168.1.1 (192.168.1.1)' can't be established
RSA key fingerprint is 5d:d2:f5:21:fa:07:64:0d:63:1b:3b:ee:a6:58:58:bb
Are you sure you want to continue connecting (yes/no)? Yes
Warning : Permanently added '192.168.1.1' (RSA) to the list of known hosts.
root@192.168.1.1' password
Last login: The Nov 13 11:17:36 2014 from 192.168.1.1
No mail

این کلید در دایرکتوری home کاربر در پوشه ای به نام ssh. که یک پوشه مخفی است ذخیره می شود. در این پوشه فایلی به نام known_host وجود دارد که حاوی کلید های Public سرورهایی است که ما به آنها کانکشن زده ایم.

ssh سیستم local با public key دریافتی از سرور و private Key خودش اطلاعات را رمزنگاری کرده و به سمت سرور ارسال می کند. pubkey سرور مثل قفل میباشد که کلید آن privkey سرور است و با آن قفل دیتا رمز شده را باز میکند.

اگر سیستم عامل سرور یا سرویس ssh را بر روی سرور (server ssh) مجددا نصب کنیم و یا ip سرور را تغییر دهیم، کلیدهای pub تولید شده در سمت سرور با کلیدهای ذخیره شده در known_hosts کامپیوترهای سرویس گیرنده تطابق نخواهند داشت و از این رو ارتباط ssh بین کلاینتها و سرور برقرار نشده

و کاربر پیغام خطایی مانند زیر دریافت می کند که در آن احتمال بروز حمله از طریق هکرها هشدار داده می شود. علت اصلی این پیام ، عدم همخوانی کلید pub سرور با کلید pub موجود بر روی کلاینت است.

پیغام های خطای سرویس دهندهssh

#ssh ali@192.168.1.1

@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
@ WARNING: REMOTE HOST IDENTIFICATION HAS CHANGED @
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
IT IS POSSIBLE THAT SOMEONE IS DOING SOMETHING NASTY
Someone could be eavesdropping on you right now (man-in-the-middle-attack)
It is also possible that the RSA host key has just been changed
The fingerprint for the RSA key sent by the remote host is
d:d2:f5:21:fa:07:64:0d:63:1b:3b:ee:a6:58:bb5
Please contact your system administrator
Add correct host key in /root/.ssh/known_hosts:2
RSA host key for 192.168.1.10 has changed and you have requested strict checking
Host key verification failed

اگر اطمینان دارید که بروز پیام به دلیل نصب مجدد سرویس یا سیستم عامل سرور ویا تغییر ip سرور است، کافی است :known_host را ویرایش کرده و خطوط مربوط به کلید قبلی سرویس دهنده ssh را از آن حذف کنیم. پس از این کار با اتصال مجدد به سرویس دهنده ssh مجددا پیغامی مبنی بر ذخیره سازی کلید جدید در فایل ssh/known_hosts/~ دریافت می کنید و از این پس جلسات کاری مربوط به سرویس ssh بدون مشکل انجام خواهد شد چون عدم همخوانی کلیدها با دانلود کلید جدید و ذخیره آن در فایل known_host بر طرف گردیده است. به ازای ارتباط با هر سرور یک محتوای کلید pub در فایل known_host ذخیره می شود.

اگر بخواهیم دو کلید pub را با هم مقایسه کنیم یکی از آنها را از درو ن فایل known_host به یک فایل جدید منتقل کرده سپس توسط دستور diff آن دو را با هم مقایسه می کنیم.

 # diff file1 file2

نصب و راه اندازی SSH Client

همانطور که گفته شد ssh یک نرم افزار انحصاری است که بابت قابلیت های آن باید لایسنس خریداری کرد. اما بسته openssh رایگان بوده و محدودیت های ssh را ندارد. لذا جهت استفاده از ssh در کلاینتهای لینوکسی هم از openssh استفاده می شود.

ابتدا باید از نصب بودن پکیج openssh-clients اطمینان حاصل کنیم لذا با دستور زیر از سیستم query می گیریم :

#rpm -qa | grep openssh-clients

در صورت نصب نبودن ، در سیستم های ردهت جهت نصب openssh-clients از yum استفاده می کنیم :

#yum –y install openssh-clients

بعد از نصب ، باید اطمینان حاصل کنیم گه آیا پکیج Openssh بر روی سیستم نصب شده است یا خیر لذا با دستور زیر از سیستم query می گیریم :

#rpm -qa | grep openssh

سپس با دستور زیر شاخه ها و مسیرهایی که فایل های این سرویس در آن ایجاد شده است را چک می کنیم :

#rpm -ql openssh-clients

و با این دستور هم اطلاعات لازم را در مورد پکیج این سرویس به دست می آوریم :

#rpm -qi openssh-clients

سپس با دستور chkconfig مشخص می کنیم در چه runlevel هایی فعال باشد :

#chkconfig sshd on

کانفیگ سرور SSH زمانی سودمند است که کلاینتها بخواهند به آن کانکشن بزنند اما کلاینت را برای این تنظیم میکنیم تا بتواند از سروری که تنظیمات آن پیش فرض نیست استفاده کند. فایل پیکربندی ssh Client در مسیر /etc/ssh/ssh_config قرار دارد. و در انتها با دستور زیر چک میکنیم ssh روی پورت 22 و یا هر پورت دیگری به حالت listen رفته باشد.

#netstat -ntpl | grep 22

استفاده از ارتباط بدون پسورد در SSH Client

در مواردی برای اجرای مجموعه ای از دستورالعمل های متوالی در قالب یک اسکریپت لازم است که ، امکان کپی کردن فایل ها از طریق SCP را بدون وارد کردن کلمه عبور داشته باشیم. یا جهت مصون ماندن از خطر لو رفتن پسورد توسط نرم افزارهای مخرب نخواهیم پسوردی جهت احراض هویت وارد کنیم. به این نوع از احراض هویت Key Base Authentication گفته می شود.

در زمان استفاده از این امکان در سرویس ssh لازم نیست هیچ گونه نگرانی برای فاش شدن کلمه عبور و یا از کار افتادن اسکریپت مورد نظرمان با تغییر کلمه عبور داشته باشیم. به راحتی جهت انجام این کار می توان سرویس SSH را پیکربندی کرد تنها باید کلید این نوع از ارتباط را به سرور معرفی کنیم.

بدین وسیله سرویس دهنده ها قادر خواهند بود به کمک این کلیدهای از پیش نصب شده یکدیگر را تایید کرده، به تبادل دیتا بپردازند. ریسک امنیتی که در این روش وجود دارد این است که امکان دسترسی به یک حساب کاربری برروی سرورتنها از طریق وارد کردن نام کاربری صورت می گیرد که برای کم کردن خطر احتمالی این کار باید از حساب های کاربری غیر مدیریتی در سرور استفاده کنیم، تا در صورت فاش شدن حساب کاربری مربوط به SSH ، امکان اعمال نفوذ در کارهای مدیریتی سیستم میسر نباشد.

در روش اول کلاینت ، pubkey سرور را دریافت و اطلاعات را با pubkey سرور و privkey خودش رمز کرده و برای سرور ارسال می کند.

اما در این روش عکس این عمل را انجام می دهد. یعنی کلاینت یک pubkey و یک privkey تولید کرده و pubkey را بر روی سرور آپلود میکند تا تمام کارها به عهده خودش باشد. در این شیوه دیگر نیازی نیست برای احراض هویت پسورد وارد کنیم بلکه پسورد ما کلیدی است که بر روی سرور قرار داده دایم. در زیر شیوه این کار به طور کامل توضیح داده شده است .

پیکربندی SSH Client و تولید کلید

در اینجا به بررسی مراحلی که جهت تبدیل کردن یک کامپیوتر به سرویس گیرنده SSH، بدون درخواست کلمه عبور انجام شود می پردازیم دستورات و مراحلی که در ادامه شرح داده می شود بر روی سیستم کلاینت اعمال می گردد. در ssh برای تولید کلید از دو الگوریتم rsa و dsa استفاده می شود. از الگوریتم dsa برای امضاء دیجیتال استفاده می شود اما rsa هم برای امضاء دیجیتال و هم برای رمزنگاری کاربرد دارد. dsa سریع تر بوده ، ولی امنیت کم تری دارد ولی rsa کندتر بوده و نسبت به dsa از امنیت بیشتری برخوردار می باشد. الگوریتم پیش فرض برای تولید کلید ، الگوریتم rsa است .

1. ابتدا در کلاینت لینوکس یک جفت کلید رمزنگاری SSH که همان کلیدهایpub و priv هستند را برای حساب کاری که قرار است از آن جهت کپی کردن فایل ها استفاده شود، ایجاد می کنیم. این کار توسط فرمان ssh-keygen صورت می گیرد که نحوه انجام آن در زیر نشان داده شده است. دقت کنید زمانی که درخواست وارد کردن یک کلمه عبور از شما می شود تنها کلید Enter را فشار دهید، و هیچ کلمه ای را وارد نکنید ، البته با سوئیچ p– می توان پسورد آن را بعد از تولید کلید عوض کرد. دقت کنید در هر مسیری که باشید کلید تولید شده در همانجا ذخیره می شود، توصیه می شود کلیدها را در پوشه ssh. ذخیره کنید :

#cd /home/skywan13/.ssh
#ssh-keygen –b 2048
 # ssh-keygen
Generating public/private dsa key pair
Enter file in which to save the key:(root/.ssh/id_dsa/)
Enter passphrase (emty for no passphrase):
Enter same passphrase again:
Your identification has been savd in.root/ .ssh.id_rsa
Your public key has been saved in.root/ .ssh/id_rsa.pub/
The key fingerprint is: e:73:59:83:96:93:4a:50:33:aa1

2. بعد از اتمام مراحل کار فایل های مربوط به کلیدهای رمزنگاری ساخته شده در پوشه ssh. از شاخه خانگی کاربر جاری ذخیره می شوند. فایل id_dsa.pub به عنوان کلید عمومی است که با سرویس دهنده مقصد به اشتراک گذاشته می شوند. و فایل id_rsa به عنوان کلید priv مورد استفاده قرار می گیرد. بعد از اتمام مراحل تولید کلید باید به سرور مورد نظر ، کلید pub ساخته شده را معرفی کنیم. این انتقال باید به صورت امن صورت پذیرد لذا با دستور scp آن را به سرور مورد نظر منتقل می کنیم :

#cd /home/skywan13/.ssh
#scp id_rsa.pub h.tohid@192.168.1.1:/home/h.tohid/.ssh

3. حال باید به سرور مقصد بفهمانیم چه طور از این کلید کپی شده استفاده کند. چون سرور هنوز متوجه نمی شود که فایل کپی شده ، pubkey ما می باشد. لذا برای مطلع کردن سرور باید محتوای فایل id_rsa.pub را داخل فایل جدیدی به نام authorized_keys قرار دهیم. سرور کلیدهای pub یوزرها را فقط از این فایل می خواند.

#cat id_rsa.pub > authorized_keys

سپس باید در سرور ، وارد فایل پیکربندی SSH شده و دوخط زیر را تغییر دهیم. اگر هر دو خطوط yes باشند یعنی یوزر هم از طریق کلید و هم از طریق پسورد اجازه احراض هویت دارد. روش ارتباط با کلید امن تر می باشد و بهتر است خط پسورد مقدار no داشته باشد.

#vi /etc/ssh/sshd_config
PubkeyAuthentication yes
Password Authentication no
RSAAuthentication yes
#service sshd restart

نکته مهم : می توانیم برای کپی به جای استفاده از scp از دستور زیر برای این کار استفاده کنیم که بسیار بهتر است ، زیرا زمانی که عمل کپی را انجام می دهد، در سرور فایل کپی شده را با نام authorized_keys ذخیره کرده و فایل را در دایرکتوری یوزر مورد نظر و در پوشه /.ssh قرار می دهد ، لذا دیگر نیازی نیست که با ssh به سرور مورد نظر وصل شده و محتویات فایل id_rsa.pub را درون فایل authorized_keys قرار دهیم.

#ssh-copy-id -i ~/.ssh/id_rsa.pub h.tohid@192.168.1.1

می توانیم برای ایجاد امنیت بیشتر در زمانی که کلید را می سازیم یک پسورد اختصاصی به آن بدهیم تا احراض هویت کاربر با دو لایه امنیت صورت پذیرد.. اگر هر دو آپشن گفته شده در فایل کانفیگ yes باشند ، 3 مرتبه اول پسورد یوزر را می پرسد و اگر اشتباه تایپ شود برای احراض هویت پسورد کلید را می پرسد.

با اتمام این کار، سرور از بابت هر کانکشنی که بر مبنای ssh کار کند از یوزر مربوطه پسورد نمی خواهد زیرا احراض هویت بر اساس کلید انجام میشود.

در زیر مروری دوباره ای خواهیم داشت بر دستورات تایپ شده در ssh Client :

Local System:

#cd /home/skywan13/.ssh
#ssh-keygen -t rsa -b 2048
#scp id_rsa.pub h.tohid@192.168.1.1:/home/h.tohid/.ssh
#ssh h.tohid@192.168.1.1

or

#ssh-copy-id -i ~/.ssh/id_rsa.pub h.tohid@192.168.1.1

Remote System :

#cd /home/h.tohid/.ssh
#cat id_rsa.pub > authorized_keys
#vi /etc/ssh/sshd_config
PubkeyAuthentication yes
Password Authentication no
RSAAuthentication yes

#service sshd restart

نکات کاربردی :

1. با دستور زیر می توانید پسورد کلید را بر روی سیستم کلاینت Cash کرد :

#ssh-add

2. این دستور نشان می دهد بر روی سیستم کلاینت چه کلیدهایی موجود است :

#ssh-add -l

3. و این دستور Cash را پاک میکند :

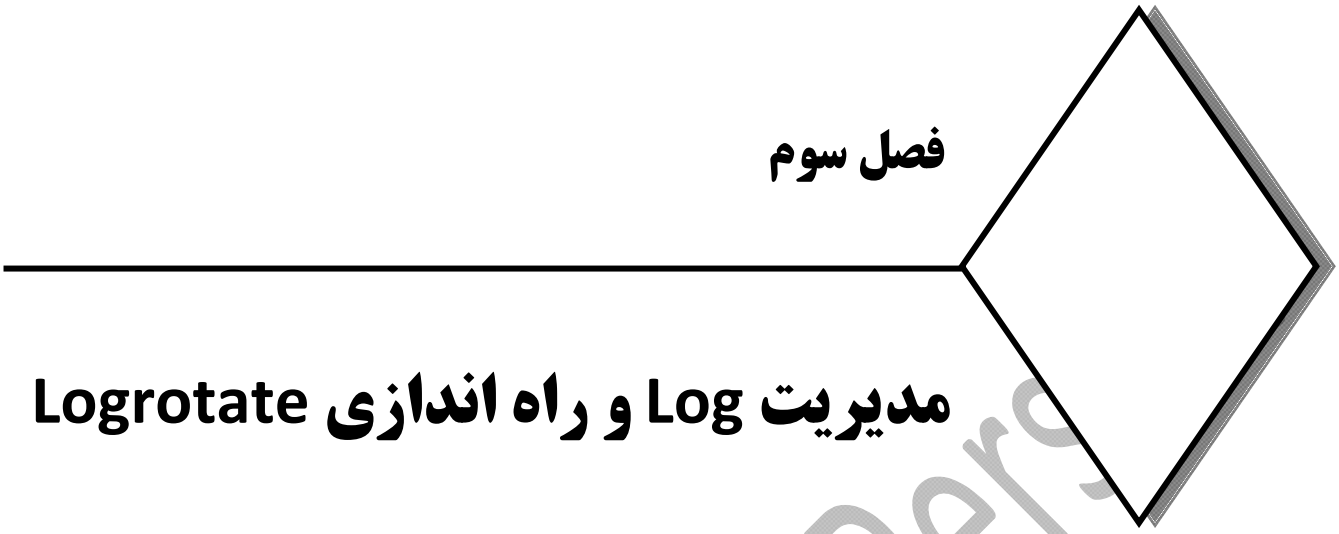
#ssh-add -d

4. در صورت برقرار نشدن ارتباط ssh با سرور از دستور زیر برای رفع مشکل استفاده می کنیم :

#ssh -vv h.tohid@192.168.1.1

5. کلاینتها می توانند بدون OpenSSH server هم کار کنند. اگر نیازی ندارند به اینکه کلاینتی به آنها remote login بزند و فقط می خواهید یک طرف به منبعی وصل شوید می توانید openssh server را از روی سیستم پاک کنید :

#chkconfig sshd off
#yum erase openssh-server
#netstat -ntlp | grep 22



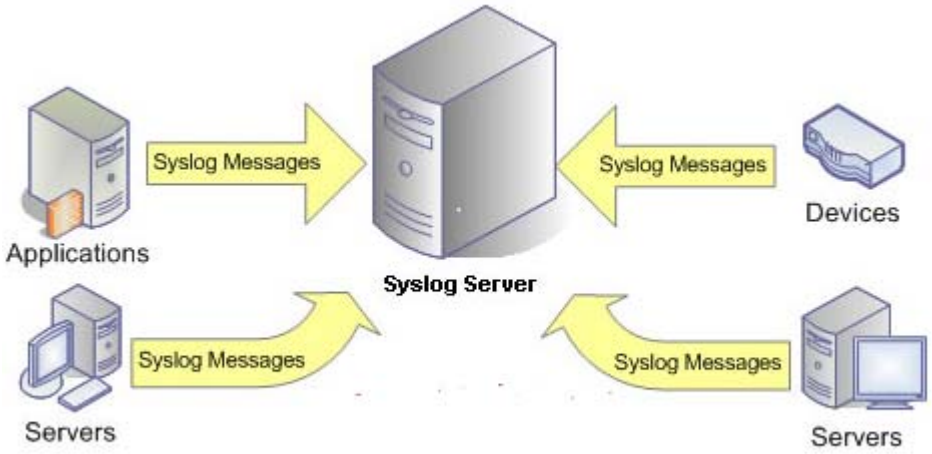
فصل سوم

مدیریت Log و راه اندازی Logrotate

پیکربندی و مدیریت لاگها با Syslog

مقدمه:

در سیستم عامل لینوکس سرویس ها ، نرم افزار ها و خود هسته در هر لحظه رویداد هایی مانند خطاها و تغییر در روند سرویس یا هر چیزی را ، در غالب فایل هایی متنی ثبت می کند که به این کار Logingیا ثبت رویداد گفته می شود. مدیران این فایل ها را مرتب و در هنگام بروز مشکل و حوادث امنیتی آنها را بررسی می کنند. مستند کردن این فایل ها یکی از وظایفی است که مدیران شبکه در شرکت ها انجام می دهند. هنگامی که یک سرویس Start یا Stop می شود و یا هر تغییر و خطایی رخ می دهد حتی هنگامی که یک عمل با موفقیت انجام می شود یک پیام در فایل Log مرتبط با آن سرویس ثبت خواهد شد. syslog سرور بر روی هر دو پلتفرم ویندوز و لینوکس قابل راه اندازی است.



فایل های Log در مسیر دایرکتوری /var/log/ قرار دارند و مرتبط با هر سرویس مانند sshd یا dhcpd یک فایل Log وجود دارد. لینوکس های ردهت تا ورژن5 از ابزاری به نام syslogd که مخفف System Log Daemonاست برای ثبت رویداد ها استفاده می کند و از ورژن 6 به بعد از rsyslog برای این کار استفاده می کند. این برنامه با سرویس ها و نرم افزار های در ارتباط بوده و آنها رویداد های خود را به این ابزار می دهند. syslogd رویداد ها را جمع آوری کرده و در فایل های Log خود آنها ثبت می کند.

فایل های Log فایل های متنی هستند و می توان با دستور های cat , less , vim, vi و دستور های head, tail آنها را مشاهده کرد اما پیشنهاد می کنم از دستور های head , tail وless استفاده کنید. از ابزار

های دیگر مرتبط با Log ها در لینوکس نرم افزار logwatch است که در بیشتر توزیع های لینوکسی وجود دارد. در دایرکتوری /etc/log.d و تحت فایل logwatch.conf قابل پیکربندی می باشد.

از اعمال مرتبط با Log ها Log Rotate یا گردش Log است. وقتی که اندازه فایل های Log زیاد می شود بایستی از آنها یک پشتیبان تهیه کرد و یا اینکه دنباله Log کردن را دریک فایل جدید ادامه داد و فایل قدیمی را آرشیو کرد. این اعمال بصورت خودکار و در غالبRotate کردن انجام می شود پیکربندی عملیات Rotate به کنترل حجم و بازخوانی ساده تر فایل ها کمک می کند.

بهتر است که در هنگام پارتیشن بندی یک پارتیشن مجزا برای دایرکتوری /var/log در نظر بگیریم چونکه رشد اندازه فایل های Log بسیار بالاست و در نظر گرفتن پارتیشن مجزا خارج از دایرکتوری / از بروز مشکل جلوگیری می کند.

یکی دیگر از موضوعاتی که قابل بحث است ذخیره رویداد ها بصورت محلی و راه دور می باشد. محلی بودن ثبت رویداد کاملا واضح است و رویداد در خود آن ماشین ذخیره می شوند اما راه دور به معنی است که یک سیستم را بعنوان Log Server انتخاب کرده و تمام ماشین ها رویداد هایشان را به این سرور ارسال کنند. توصیه می شود برای حفظ محرمانگی ، داده ها تحت ssh مبادله شوند و بهتر است که ثبت روبداد را هم بصورت محلی(یعنی در خود همان ماشین) و هم بصورت راه دور(یعنی در یک سرور مجزا) انجام دهید.

loggingراه دور یک قابلیت امنیتی فوق العاده است. با قراردادن-log هایتان در سیستم راه دور، می توانید از رخنه ها و نفودهای امنیتی که به راحتی می توانند فایل را تغییر دهند، جلوگیری کنید.

دو سرویس یا دایمون (daemon) وجود دارد که گزارش گیری را کنترل می کند klogd وsyslogd.

klogd فقط با پیغامهای کرنل و syslogd با دیگر پیغامهای سیستم مانند برنامه های کاربردی سر و کار دارد. شما می توانید رفتار این دو ابزار را با ویرایش فایل /etc/syslog.conf و فایل تغییر فیچرهای سرویس یعنی /etc/sysconfig/syslog پیکربندی کنید.

همچنین می توانید اطلاعات بیشتر را در صفحه راهنمای /etc/syslog.conf کسب نمایید.

هر پیغامی که توسط نرم افزاری تولید می شود اطلاعاتی در مورد محتوای پیغام و مبدا و تولید کننده آن می دهد. فایل /etc/syslog.conf به شما امکان می دهد که هر گونه پردازشی را بر روی پیام ها تعیین کنید.

به طور موقت می توانید این اطلاعات را در فایل **message** انبار کنید. همچنین می توانید آنها را در یک فایل سفارشی ذخیره سازید. می توانید آنها را به یک میزبان (host) راه دور، جایی که میزبان آنها را مطابق با پیکربندی syslogd خودش پردازش خواهد کرد، ارسال نمایید .

نصب و راه اندازی سرویس Syslog

در بیشتر زمان ها syslog در موقع نصب سیستم نصب می شود و شما نیازی به نصب مجدد ندارید.

ابتدا باید از نصب بودن پکیج syslog اطمینان حاصل کنیم لذا با دستور زیر از سیستم query میگیریم :

#rpm -qa | grep syslog

در صورت نصب نبودن ، در سیستم های ردهت جهت نصب syslog از yum استفاده می کنیم :

#yum –y install syslog

بعد از نصب ، باید اطمینان حاصل کنیم که آیا پکیج syslog بر روی سیستم نصب شده است یا خیر لذا با دستور زیر از سیستم query می گیریم :

#rpm -qa | grep syslog

سپس با دستور زیر شاخه ها و مسیرهایی که فایل های این سرویس در آن ایجاد شده است را چک می کنیم :

#rpm -ql syslog

و با این دستور هم اطلاعات لازم را در مورد پکیج این سرویس به دست می آوریم :

#rpm -qi syslog

سپس با دستور chkconfig مشخص می کنیم در چه runlevel هایی فعال باشد :

 # chkconfig --level 35 syslog on

و در انتها سرویس را reset می کنیم :

#service syslog restart

نکته مهم : در لینوکس های redhat base سری 6 به بعد به جای syslog از rsyslog استفاده می شود.

مبناهای کاری Syslog

syslog بر دو مبنا کار می کند

1. Unix domain socket
2. Internet socket

اگر syslog ، لاگهای سیستم را در سیستم local ذخیره کند بر مبنای Unix domain socket کار می کند و اگر لاگها را از طریق پورت 514 udp به درون سیستم مشخص شده ای در شبکه منتقل کند در مد Internet socket فعالیت می کند.

فایل کانفیگ این سرویس در مسیر /etc/syslog.conf قرار دارد. در این فایل یکسری rule یا همان قوانین وجود دارد که از سه قسمت عمده تشکیل شده است :

Facility severity where

1)**Facility** : مشخص میکند از چه چیزهایی log برداری شود.

auth - authentication (login) messages
cron - messages from the memory-resident scheduler
daemon - messages from resident daemons
kern - kernel messages
lpr - printer messages (used by JetDirect cards)
mail - messages from Sendmail
user - messages from user-initiated processes/apps
local0-local7 - user-defined (for cisco,servers,...)
syslog - messages from the syslog process itself

اگر بخواهیم log پروسه ای ، به صورت جداگانه در جایی به غیر از facility ثبت شود از local استفاده می کنیم.علاوه بر آنکه می توانید مشخص کنیم لاگ ها به کجا بروند , می توانید نوع پیغام هایی که برای سرور لاگ فرستاده می شود را توسط سطوح مختلف severity مشخص کنیم. این سطوح که به آنها level هم گفته می شود استاندارد بوده و براساس شماره و یا حروف اختصاری بکار برده می شوند.

2) **Severity**: درجه اهمیت ، یا level لاگ را مشخص می کند.

7 - Emergency (emerg)
6 - Alerts (alert) اخطار
5 - Critical (crit) شرایط بحرانی
4 - Errors (err) خطا
3 - Warnings (warn) هشدار
2 - Notification (notice) اطلاعیه
1 - Information (info) اطلاعات بیستر
0 - Debug (debug)

در سیستمهای Unix Base درجه اهمیت از صفر الی 7 متغییر است. بالاترین درجه اهمیت کمترین اطلاعات را به ما می دهد و بیشترین اطلاعات راdebug اعلام میکند.
اگر تعریف کنیم لاگ برداری از Debug شروع شود ، سرور از Debug به بالا، همه لاگ ها را ثبت می کند. یعنی از هر کجا تعریف کنیم از آنجا به بالا را لاگ برداری می کند.

3) **where** : این قسمت مکان ذخیره سازی فایل های لاگ مشخص می کند. در اینجا سه متغیر می توانند قرار گیرد : tty file address /dev/console
اگر مشخص کنیم که لاگ بر روی /dev/console قرار بگیرد بر روی مانیتور تمام یوزرها قابل رویت می باشد.
نکته : در اینجا ستاره به معنی همه می باشد.

سطوح Log ها

رتبه بندی	واژه	شرح
0	emergencies	سیستم عملا غیرقابل استفاده است
1	alerts	باید سریعا عکس العمل نشان دهیم
2	critical	شرایط بحرانی می باشد
3	errors	خطائی در سیستم وجوددارد
4	warnings	اخطار...
5	notifications	شرایط عادی ولی مشکلاتی وجوددارد
6	informational	جهت اطلاع....
7	debugging	پیام های مربوط به **Debugging** سیستم

توضیح : جدول سطوح Log

تنظیم لاگ بر اساس Unix domain socket

اگر بخواهیم لاگها در سیستم local ذخیره شود از شیوه زیر برای نوشتن تنظیمات در فایل کانفیگ log بهره می بریم.

facility.severity **where+ log-file-name**

در ادامه برای درک بهتر مفهوم Unix domain socket چند مثال آورده شده است.

*.info;mail.none;authpriv.none;cron.none /var/log/message

*: در اینجا به معنی همه facility ها می باشد.

info: آوردن این کلمه یعنی از info به بالا لاگ بگیرد.

None: اگر بخواهیم از سرویس و یا پروسه ای لاگ بردار نشود از کلمه none استفاده می کنیم.none به معنای منفی شدن است.

Authpriv.* /var/log/secure
Mail.* /var/log/maillog
Cron.* /var/log/cron

ستاره در سه مثال قبل یعنی اینکه لاگ مربوطه تمام severity ها را شامل شود.

*.emerg *

.* اول این خط یعنی تمام facility ها را شامل می شود و ستاره آخر مشخص می کند خروجی لاگ در tty , file , /dev/console نشان داده شود.

مثال) به طور پیش فرض لاگ های dhcp در مسیر /var/log/message ذخیره می شود. حال می خواهیم کاری کنیم لاگ های dhcp در مسیر /var/log/dhcp ذخیره گردد.

ابتدا وارد فایل کانفیگ dhcp شده و عبارت زیر را به آن اضافه می کنیم :

```
log-facility    local2;
```

سپس فایل /etc/syslog.conf را باز کرده و مشخص می کنیم لاگهای مربوط به local2 درون چه فایل ذخیره شوند.

```
# vi   /etc/syslog.conf

*.info;mail.none;authpriv.none;cron.none;local2.none  /var/log/message

local2.*                                              /var/log/dhcp
```

همانطور که از مثال ها مشخص است , می توان به چندین شکل syslog را جهت نگه داری پیغام ها تنظیم کنیم .

می توانید با * تمامی severityها را مشخص کنید تا در یک فایل ذخیره شوند یا اینکه با مشخص کردن نام آن فقط آن severity را ذخیره کنید. همچنین می توانیدseverity های مختلف را در فایل های مختلف ذخیره کنید. توصیه می شود که لاگ ها را بر اساس نیازتان در فایل های مجزا تقسیم بندی کنید تا در آینده آنالیز آنها راحت تر باشد.

تنظیم لاگ بر اساس Internet socket

برای استفاده از syslog به جهت دریافت لاگ از دستگاه ها وسرورهای دیگر , می بایست ویژگی Udp logging را در سیستمهای مورد نظر فعال کنیم تا ارسال لاگ بر روی شبکه از طریق پورت 514 پروتکل udp و آدرس تنظیم شده انجام پذیرد.

ابتدا در سرور syslog ، وارد مسیر /etc/sysconfig شده و فایل syslog را edit می کنیم.

 # vi /etc/sysconfig/syslog

SYSLOG_OPTIONS= " m 0 -r "

r: برای فعال کردنremote UDP logging

m 0: برای حذف پیام هاMARK

X: برای غیر فعال کردنDNS lookup

برای آنکه مطمئن شویم که UDP logging فعال شده و سرور بر روی پورت 514 به حالت Listen رفته از دستور زیر استفاده می کنیم.

 # service syslog restart

 # netstat -nulp | grep 514

udp 0 0.0.0.0:514 0.0.0.0:*

8621/syslogd

حال برای اینکه سیستم های دیگر لاگ خود را به سرور ارسال کنند در سیستمهای ارسال کننده لاگ ، وارد فایل /etc/syslog.conf شده و طبق مثال زیر آدرس سروری که می خواهیم لاگها به آن ارسال شوند را وارد می کنیم.

 # vi /etc/syslog.conf

***.info;mail.none;authpriv.none;cron.none @10.10.10.1**

 # service syslog restart

توضیح : در قسمت آدرس به جای وارد کردن یک مسیر local آدرس سرور syslog را وارد کنید.

Log فایل های مهم

در توزیع CentOS و دیگر توزیع ها در زیر دایرکتوری /var/log چندین فایل وجود دارد که به مهمترین آنها اشاره می کنیم:

message : گزارشهای پیغام (message logs) هسته فایل log سیستم هستند. این فایل، شامل پیغامهای بوت و پیغامهای وضعیت و اجراهای سیستم می باشد. خطاهای IO ، شبکه و دیگر خطاهای عمومی سیستم در این فایل گزارش می شوند. سایر اطلاعات از قبیل مواقعی که یک فرد، root میشود نیز در اینجا فهرست می شوند. اگر سرویسهایی مانند سرور DHCP اجرا شوند، فعالیتها را در فایلهای پیغام می توانید مشاهده کنید. فایل /var/log/messages معمولا اولین مکانی است که در مواقع به وجود آمدن دردسر می توانید به آن مراجعه نمائید.

XFree86.0.log : این log نتایج آخرین اجرای کارساز Xfree86 Xwindow را نشان می دهد. اگر در بالا آمدن مود گرافیکی دچار مشکل شدید، این فایل معمولا جوابهایی برای عوامل سوال برانگیز مشکل فراهم می آورد.

auth.log: لاگهتی مربوط به احراض هویت در این فایل ذخیره می شود.

kern.log : این فایل حاوی اطلاعات و رویداد های کرنل سیستم عامل می باشد.

cron.log : این فایل حاوی اطلاعات مربوط به این سرویس cron است.

mail.log : اطلاعات و رویداد های Mail Server ها و MTA هایی مانندsendmail در این فایل ثبت می شود.

qmail : در صورتی که qmail را نصب کرده باشید. رویدادهای این سرویس دهنده میل در این فایل قرار می گیرند.

httpd : این فایل مرتبط با وب سرور آپاچی (در صورتی که httpd را نصب کرده باشید وجود دارد)

boot.log : این فایل مرتبط با اطلاعات و رویداد های فرایند بوت شدن سیستممی باشد.

mysqld.log : این فایل مرتبط با پایگاه داده MySQL می باشد البته در صورتی که MySQL را نصب کرده باشید .

secure : حوادث امنیتی سیستم در این فایل ثبت می شوند.

yum.log : مختص سیستم های مبنتی بر RedHat که در ارتباط با دستور yum است.
خواندن و مشاهده این فایل ها و حتی استفاده از دستور های خاصی مانند last نیاز به دسترسی کاربر ریشه دارد. یعنی یک کاربر عادی نمی تواند این فایل ها را تغییر دهد یا حتی خود مدیر هم شاید نتواند این فایل هامانند wtmp را تغییر دهند چون اطلاعات ضروری در آنها ثبت شده اند.

ابزارهای گزارش گیری(logging)

هر گونه ابزار متنی را می توان برای کار با فایلهای log به کار برد. در ادامه برخی از این ابزارهای مفید را معرفی نموده ایم:

- **dmesg**

برای مرور اجمالی log بوت در آخرین بار بوت شدن سیستم، می توانید از دستور dmesg استفاده کتید. خروجی این دستور، عموما متن طویلی است. بنابراین آن را برای مشاهده صفحه به صفحه پایپ کنید.

- **tail**

برخی اوقات می خواهید فقط یک مرور اجمالی و کوتاه در فایل log فعالیتهای در حال وقوع بیندازید tail برای نمایش آخرین خطوط یک فایل متنی طراحی شده است. با افزودن سوییچ -f ، دستور tail به نمایش خروجیهای جدیدی که ناشی از رخ دادن آخرین وقایع است، ادامه می دهد.

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

دستور فوق، آخرین ۱۰ خط فایل var/log/messages/ را نشان می دهد، سپس به نظارت در فایل و خروجی هر فعالیت جدید ادامه می دهد. جهت متوقف ساختن دستور فوق، از Ctrl + C برای کنسل کردن این فرایند استفاده کنید.

- **more**

دستور more همان کاری را انجام می دهد که در نگارش DOS انجام می داد. شما می توانید آن را به همراه اسم فایل و نیز برای پایپ کردن اطلاعات در صفحه نمایش استفاده کنید. به عنوان مثال، برای نمایش صفحه به صفحه محتویات فایل logآغازگر (startup) از دستور زیر استفاده کنید:

#more /var/log/XFree86.0.log

• less

دستور less نیز یک مشاهده گر متنی دیگر است که به امکان scroll در یک فایل و نیز جستجوی اطلاعات در آن را می دهد.

#less /var/log/messages

دستور فوق محتویات فایل /var/log/messages را نشان خواهد داد. با استفاده از "q" می توان از مود مشاهده فایل خارج شد و با استفاده از "h" اطلاعات بیشتری در مورد نحوه کارکرد دستور فوق دریافت می کنید.

•logger

ممکن است بخواهید پیغامهای خودتان را در یک فایل log قرار دهید. کافی است پیغام log را به انتهای فایل متنی درستی، ضمیمه (append) کنید. اما مجبور خواهید شد که اطلاعات گزارش را تکرار کنید.

همچنین باید کد خود را در صورت سفارشی بودن سیستم logging تغییر دهید .دستور logger امکان ارسال پیغامهای شما را به ابزار موجود برای logging می دهد. از این دستور در اسکریپتهایی برای تهیه پیغامهایی در مورد نحوه اجرا و خطاها استفاده می شود.

چرخش لاگها با logrotate

زمانی که سرور تراکنش دیتا بالا و یوزر استفاده کننده زیادی داشته باشد حجم فایلهای log به مرور می تواند خیلی بزرگ و حجیم شود که این حجیم شدن فایلهای لاگ هم فضای سیستم را اشغال می کند و هم واکشی و خواندن آنها را همراه با تاخیر میکند. لینوکس ابزاری برای چرخش این-log ها دارد که به صورت دوره ای لاگهای قدیمی را جابه جا و می چرخاند. بنابراین اطلاعات log جاری شما با اطلاعات نامربوط قدیمی، ترکیب نمی شوند. با این کار حجم لاگها کمتر و مدیریت آنها بهتر می شود.

معمولا logrotateبه طور خودکار بر اساس یک برنامه زمان بندی اجرا می شود. اما به طور دستی نیز قابل تنظیم و اجراست. شما فایلهایی در شاخه /var/log مشاهده می کنید که با یک عدد تمام می شوند .اینها آرشیوهای دوار (چرخشی) هستند. هنگامی که این سرویس اجرا می شود، logrotate، نگارش جاری فایلهای log را گرفته و یک "۱،" به انتهای نام فایل می افزاید.

از آن به بعد، ترتیب دیگر فایلهای چرخش یافته به صورت "۲،" ، "۳،" و غیره خواهد بود. عدد بزرگتر بعد از نام فایل، نشان دهنده گزارشهای جدیدتر میباشد. رفتار خودکار logrotate را با ویرایش فایل /etc/logrotate.confمی توانید پیکربندی کنید.

نصب و راه اندازی سرویس logrotate

در بیشتر زمان ها logrotate در موقع نصب سیستم نصب می شود و شما نیازی به نصب مجدد ندارید.

ابتدا باید از نصب بودن پکیج logrotate اطمینان حاصل کنیم لذا با دستور زیر از سیستم query میگیریم

#rpm -qa | grep logrotate

در صورت نصب نبودن ، در سیستم های ردهت جهت نصب syslog از yum استفاده می کنیم :

#yum –y install logrotate

بعد از نصب ، باید اطمینان حاصل کنیم که آیا پکیج logrotate بر روی سیستم نصب شده است یا خیر لذا با دستور زیر از سیستم query می گیریم :

#rpm -qa | grep logrotate

سپس با دستور زیر شاخه ها و مسیرهایی که فایل های این سرویس در آن ایجاد شده است را چک می کنیم :

#rpm -ql logrotate

و با این دستور هم اطلاعات لازم را در مورد پکیج این سرویس به دست می آوریم :

#rpm -qi logrotate

سپس با دستور chkconfig مشخص می کنیم در چه runlevel هایی فعال باشد :

 # chkconfig --level 35 logrotate on

نکته : logrotate یک سرویس است ولی اسکریپت اجرایی ندارد و خودش فایل ها را چک نمی کند بلکه می رود کار را با cron کامل می کند.

مهمترین فایل های logrotate

بعد از نصب این سرویس تعدادی مسیر و فایل به سیستم اضافه می شود که سه عدد از مهتمرین آنها که کار تنظیم و پیکربندی این سرویس را انجام می دهند به شرح زیر می باشد:

/etc/cron.daily/logrotate
/etc/logrotate.conf
/etc/logrotate.d

توضیح /etc/cron.daily/logrotate

این فایل ارتباط بین logrotate و سرویس cron را برقرار ساخته و به logrotate می گوید از چه مسیری فایل کانفیگش را بخواند.

توضیح /etc/logrotate.conf

فایل logrotate.conf فایل پیکربندی گلوبال این سرویس است. تنظیمات این فایل به همه اعمال می شود ولی کانفیگ لاگ هر سرویس به تنهایی بر کانفیگ گلوبال ارجحیت دارد. اگر در خود فایل گلوبال و در انتهای آن تنظیماتی برای یک سرویس نوشته شود (داخل کروشه) این بر تنظیمات اصلی ارجحیت اجرایی دارد. در ادامه نمونه ای از یک فایل پیکر بندی آورده شده که بعضی از جزئیات آن را شرح می دهیم:

 # vi /etc/logrotate.conf

```
# see "man logrotate" for details
# rotate log files weekly
weekly
# keep 4 weeks worth of backlogs
rotate 4
# create new (empty) log files after rotating old ones
create
# use date as a suffix of the rotated file
dateext
```

```
# uncomment this if you want your log files compressed
compress
# RPM packages drop log rotation information into this directory
include /etc/logrotate.d
# no packages own wtmp and btmp -- we'll rotate them here
/var/log/wtmp {
   monthly
   create 0664 root utmp
      minsize 1M
   rotate 1
}
/var/log/btmp {
   missingok
:
monthly
   create 0600 root utmp
rotate 1
```

 # rotate log files weekly
weekly

این گزینه زمان rotate شدن log فایل ها را مشخص می کند که سه مقدار daily,weekly,monthly می تواند داشته باشد.

 # keep 4 weeks worth of backlogs
rotate 4

مقدار این خط مشخص می کند تعداد دفعات rotate چند مرتبه باشد.

 # RPM packages drop log rotation information into this directory
include /etc/logrotate.d

این خط مشخص می کند اطلاعات log های rotate شده به چع مسیری اضافه شود.

 # uncomment this if you want your log files compressed
 # compress

این خط مشخص می کند که آیا لاگهای rotate شده در هنگام ذخیره شدن فشرده شوند . این آپشن به بقیه فایل های کانفیگ لاگ اعمال نمی شود مگر اینکه کانفیگ لاگ سرویسی را درون خود فایل اصلی پیکربندی logrotate بیاوریم.

size 100k

اگر بخواهیم به جای هفتگی یا تایمی به صورت حجمی عمل rotate انجام شود از این گزینه استفاده می کنیم.

بررسی /etc/logrotate.d

در زیر این دایرکتوری فایل کانفیگ logrotate سرویس های مختلفی قرار دارد. تمام سرویس هایی که باید از عملکرد آنها لاگ جداگانه تهیه شود در این مسیر یک فایل پیکربندی دارند تا توسط logrotate عمل چرخش لاگ انها انجام شود. اگر بخواهیم logهای سرویس های مورد نظر در این دایرکتوری فشرده شوند باید فیلد compress را درون هر کدام می خواهیم اضافه کنیم. یکی از مهمترین فایلهای کانفیگی که در این مسیر وجود دارد فایل لاگ سرویس httpd می باشد که جهت آشنایی با گزینه های دیگر گانفیگ logrotate ، این فایل را مورد بررسی قرار می دهیم .

 # vi /etc/logrotate.d/httpd.log

```
/var/log/httpd/*log {
size 100k
compress
rotate 5
missingok
notifempty
sharedscripts
postrotate
    Sbin/service httpd reload > /dev/null 2> /dev/null || true
endscript
}
```

نکته مهم : چون rotate این فایل درون خودش نیامده ، آن را از فایل global سرویس logrotate خوانده و اجرا می کند. اگر زمان بندی rotate را در این فایل بیاوریم ارجحیت پیدا می کند به زمان بندیی که در فایل کانفیگ سرویس درج شده است.

در ادامه به تشریح بعضی از قسمتهای این فایل می پردازیم:

/var/log/httpd/*log : این خط مسیر لاگ فایل ها را مشخص می کند.

size : سایز لاگ فایل را مشخص می کند.

rotate : تعداد دفعاتی که لاگ فایل قبل از پاک شدن rotate می شود.

missingok : یعنی اگر فایل لاگی موجود نبود ایراد نگیرد.

notifempty : مشخص می کند اگر فایل لاگ خالی بود آن را rotate نکند.

postrotate : این گزینه یک فایل لاگ جدید ساخته و سرویس را reload می کند.

postscripts : مشخص می کند بعد از اینکه rotate را انجام داد اسکریپت و یا دستور مورد نظر را انجام دهد.

prescripts : مشخص می کند قبل از اینکه rotate را انجام بدهد اسکریپت و یا دستور مورد نظر را اجرا کند .

dateext : این گزینه خیلی مهم و کاربردی است چون در انتهای فایل لاگ تاریخ rotate شدن را درج میکند.

Mail : نتیجه را به آدرس مشخص شده میل می کند.

مثال : فایل کانفیگ لاگی بنویسید که اگر حجم فایل مشخصی به 300 بایت رسید آن را rotate کرده و نتیجه را میل و در انتها به جای عدد در نام فایل ها تاریخ را درج کند.

ابتدا با دستور dd چند فایل با حجم های متفاوت ایجاد می کنیم.

```
# mkdir /root/logs
# dd  if=/dev/zero  of=/root/logs/test.log  bs=300  count=1
```

سپس یک فایل کانفیگ می نویسیم تا لاگ این فایل را rotate کند.

 # vi /etc/logrotate.d/test

```
/root/logs/*.log  {
size 100k
compress
rotate 5
mail  root@localhost
dateext
}
```

سپس با دستور زیر آن را rotate می کنیم :

 # logrotate   /etc/logrotate.conf

نکته مهم :

در مسیر های /var/log/ و /var/run/ دو فایل به نام های wtmp و utmp وجود دارد .

/var/log/wtmp
/var/run/utmp

این فایل ها باینری هستند و به سادگی خوانده نمی شوند.

wtmp فایل لاگ History Call سیستم است و درون آن اتفاقاتی مثل crash کردن سیستم ، یا اینکه چه یوزری از چه pts ای لاگین کرده است ، ثبت می شود.

utmp هم برای لاگ کردن لاگین های موفق و نا موفق به کار می رود ولی History Call نیست.

دستور last آخرین اطلاعات سیستم را به صورت History Call نمایش می دهد و با دستور lastb می توان Bad login های سیستم را مشاهده کرد.

فصل چهارم

اشتراک سیستم فایل با NFS

مقدمه‌ای بر NFS

NFSکه مخفف Network File System است بطور خلاصه امکانیست که اجازه می دهد تا یک سیستم فایل محلی قابل mount شدن توسط دیگر سیستم ها در شبکه مورد استفاده قرار بگیرد. NFS یک روش سنتی برای share کردن دایرکتوری بین سیستم های Unix Base می باشد که به وسیله‌ی Sun Microsystemsدر سال ۱۹۸۰ توسعه و به طور پیش فرض بر روی سیستم های Redhat Base نصب است و در توزیع CentOS از ورژن 3 آن استفاده می شود. ورژن 4 از سری 6 لینوکس های Redhat Base به آنها اضافه شد. با کمک NFS ، ما می‌توانیم اشتراک فایل بین سیستم یونیکس به لینوکس و لینوکس به یونیکس را راه اندازی کنیم. تا پیش از NFSv4 تمامی نسخه های NFS بصورت Stateless بودند، به این معنی که در پروتکل هایStateless ، هر درخواست هیچ اطلاعاتی از درخواست های پیشین ندارد و هر درخواست مستقل از درخواست های قبلی می باشد. انواع متفاوتی از سیستم فایل ها قابل دسترسی هستند که این کار با استفاده از یک سری API انجام می گیرد ، به این api ها Virtual File Systemگفته می شود.

هدف اصلی ازVFS ، اجازه دادن به برنامه های کاربردی کلاینت برای دسترسی به انواع سیستم فایل ها در یک شکل واحد بدون آگاهی از این تفاوت هامی باشد، که در تمامی سیستم های عامل مانند ویندوز،لینوکس،مکینتاش و تمامی شبه یونیکس ها استفاده می شود. یک سیستم فایل NFS که بر روی ماشینی Mount شده بسیار شبیه به سیستم فایل محلی همان ماشین است .

هدف از VFS دسترسی به سیستم فایل های مختلف بر روی ماشین است. تمام اعمال روی سیستم فایل ماشین محلی که یک سیستم فایل راه دور بر روی آن Mount شده است، از طریق VFS انجام می گیرد.

در ارائه سرویس NFS هیچ محدودیتی در نوع کاربرد سیستم نیست یعنی در یک شرکت با 100 سیستم نباید حتما یکی از آنها بعنوان سروردهنده NFS باشد و مابقی از آن سرویس بگیرند بلکه هرکدام از این سیستم ها می توانند نقش سرویس دهنده NFS را داشته باشند و دایرکتوری ها و فایل های خود را به اشتراک بگذارند.

اما می توان از دیدگاهی دیگر راه اندازی آنرا به دو بخش تنظیم سرویس دهنده و تنظیم سرویس گیرنده تقسیم کرد چون بالاخره یکی دایرکتوری را به اشتراک می گذارد و بقیه استفاده می کنند پس شاید بنا به سیاست یک سیستم هم دایرکتوری را برای دیگران به اشتراک بگذارد و هم از اشتراکات دیگران استفاده کند. تنها مطلبی که باید دقت شود اطمینان از فعال بودن سرویس های لازم در هردو سمت سسرویس دهنده و سرویس گیرنده بود که برای کارکرد صحیح سرویس NFS بسیار لازم و ضروری هستند.

تمامی نسخه های NFS از پروتکل TCP استفاده می کنند. NFSv2 امروزه کمتر استفاده می شود اما NFSv3 و NFSv4 بسیار کاربرد دارند. NFSv4 بواسطه ثابت بودن پورت و تنظیمات فایروال می تواند برروی اینترنت کار کند و همچنین از ویژگی ACLکه نیز پشتیبانی می کند. NFSv2 و NFSv3 از پروتکل UDPنیز می توانند استفاده کنند.مشکل اصلی ورژن های 3و2و1 رندوم بودن پورت های ارتباطی آنها می باشد چون نمی توانیم پورت های مربوطه را بر روی فایروال باز کنیم، که این مشکل در ورژن 4 برطرف گردیده است. فایل /etc/exports بخش اصلی تنظیم NFS است که فهرستی از دایرکتوری های اشتراکی را درون خود نگه می دارد. در این فایل تعیین می شود کدام دایرکتوری ها با چه مجوز هایی باید برای چه کسانی و با چه مجوز هایی به اشتراک گذاشته شوند. به صورت پیش فرض این فایل بدون محتوا بوده و باید توسط کاربر مقدار دهی شود.

در اوایل سرویس های NFS بصورت Stateless بودن یعنی توانایی ثبت و نگهداری تاریخچه ای از فعالیت ها و اینکه کدام فایل تو سط کدام ماشین ها و کاربران راه دور استفاده شده اند را نداشتند. این اطلاعات و تاریخچه ها برای قفل کردن فایل ها لازم می باشند. یک راه حل برای این مشکل بکار بردن ابزار های مجزا از NFS مانند statd و lockd برای قفل کردن فایل هاست که در NFSv2 و NFSv3 استفاده می شوند. اما NFSv4 ویژگی statfull را که نشان دهنده مبتنی بودن بر حالت و نگه داشتن تاریخچه ای از فعالیت ها را دارا می باشد.

NFSv4 مبتنی بر state است و هم برروی کلاینت و هم برروی سرویس دهنده اطلاعاتی را نگه می دارد مانند اینکه کدام فایل ها باز هستند و یا کدام ها قفل شده اند. این اطلاعات در صورتی که سرویس دهنده دچار مشکل شوند برای رفع اشکال سرویس دهنده میان آنها رد و بدل خواهد شد.

مزایای استفاده از NFS

1. NFS اجازه‌ی دسترسی محلی به فایل‌های از راه دور را می‌دهد.
2. NFS از معماری سرویس گیرنده / سرویس دهنده استاندارد برای به اشتراک گذاری فایل بین همه‌ی ماشین‌های مبتنی بر nix*،استفاده می کند.
3. با NFS ،هیچ نیازی نیست که روی هر دو ماشین ،سیستم‌عامل مشابه اجرا شده باشد.
4. با کمک NFS ما می‌توانیم راه حل‌های ذخیره‌سازی مرکزی را پیکربندی کنیم.
5. کاربران می‌توانند اطلاعات خود را بدون توجه به موقعیت فیزیکی ، دریافت کنند.
6. هیچ refresh دستی برای فایل‌های جدید نیاز نمی‌شود.
7. نسخه جدیدتر NFS همچنین ACL و مانت ریشه‌ی کاذب را پشتیبانی می کند.
8. می‌توان با Firewallها و Kerberos آن را امن‌تر کرد.

ویژگی های NFS.v4

1. سازگاری با فایروال ها و ابزار هایNAT
2. امنیت بالا
3. بر طرف کردن مشکلات Authentication
4. پشتیبانی از کلاینت های یونیکسی (Linux,BSD,Mac OSX)و ویندوزی
5. پشتیبانی از نوشتن ACL
6. پشتیبانی از نام فایل های یونیکد
7. کارایی بالا حتی بر روی شبکه ها با پهنای باند پایین
8. برمبنای tcp کار کرده و پورت ثابت آن 2049 می باشد و البته نیاز به نرم افزارهای کمکی مثل Portmapper را ندارد.
9. کلاینت های ویندوزی می توانند از share های nfs استفاده کنند.
10. NFSv4 بر روی شبکه های wan هم کار میکند.

11. مبحث RootDirectory به آن اضافه شده است.

تا ورژن 3 ، سرویس NFS از پورت استاتیک استفاده نمی کرد. نرم افزارهای دیگری مثل Portmapper پیشنهاد دهنده پورت به کلاینت بودند. Portmapper چک می کند NFS با چه پورتی کار میکند ، به محض اینکه درخواستی از یک کلاینت برسد می آید پورت مربوطه را به کلاینتها پیشنهاد می دهد.

RHEL و دیگر توزیع های لینوکسی ترکیبی از سرویس ها را برای انجام NFS استفاده می کنند. تمامی نسخه های NFS متکی بر RPC یا Remote Procedure Call میان کلاینت ها و سرور ها هستند. سرویس های RPC در لینوکس تحت سرویس portmap کار می کنند. در زیر فهرستی از سرویس ها که بصورت ترکیبی با هم در حال ارتباط و کار کردن برای اجرای NFS هستند آمده است:

nfs : اصلی ترین سرویس که دیگر سرویس ها با فعال شدن آن نیز فعال می شوند.

nfslockd : در سمت کلاینت اجازه می دهد تا فایل ها را بروی سرور قفل یا lock کند.

portmap : همانطور که گفته شد سرویس های RPCتحت لینوکس توسط این سرویسس کنترل می شوند و مسئول تنظیم کردن اتصال ها برای سرویس های RPC درخواست شده است.

RPC های مورد استفاده در NFS

وقتی دو کلاینت لینوکسی قصد دارند به منابع اشتراکی هم متصل شوند از پروتکل NFS استفاده می کنند و خود NFS در پس زمینه از RPCها برای اتصال استفاده میکند. NFS برای ارتباط نیاز به شش نوع RPC دارد. تمام فایل سیستم هایی که تحت شبکه کار می کنند برای عملکرد صحیح خود از سرویس به نام RPC استفاده می کنند. در واقع RPC پروسه ای است که Computing دستورات بین مبدا و مقصد را انجام می دهد تا کلاینت درگیر پروسه ارتباط نشود. فهرست این rpc ها به همراه توضیح در زیر آمده است:

rpc.mountd : این سرویس در خواست های mount را از سمت کلاینت در یافت کرده و بررسی می کند که دایرکتوری درخواست شده در حال حاظر به اشتراک گذاشته شده یا خیر. این سرویس عمل mount شدن منبع اشتراکی را انجام می دهد. فهرست دایرکتوری ها صادر شده در فایل /etc/exportsقرار می گیرند. exports شدن به این معنی است که یک دایرکتوری را در شبکه به اشتراک بگذاریم. این سرویس در هنگام فعال کردن سرویس nfs بصورت خودکار فعال می شود.

rpc.nfsd : این سرویس با هسته لینوکس برای مواجه شدن با در خواست های پویا از سمت کلاینت کار می کند.

rpc.nfslock : به کلاینت اجازه می دهد فایل های خود را بر روی سرور قفل کند اگر فایل مورد استفاده lock نشود ممکن است توسط شخص دیگری مورد تغییر قرار بگیرد. این سرویس در NFSv4 استفاده نمی شود.

rpc.statd : کلاینت ها را در صورتی که سرور restart شود باخبر می کند. کار این RPC مانیتور کردن مبداء و مقصد می باشد یعنی اگر هر کدام از طرفین ریبوت شوند به طرف مقابل خبر داده می شود تا lock فایل ها برداشته شئد. این سرویس توسط nfslock فعال می شود و در NFSv4 استفاده نمی شود.

rpc.rquotad : این سرویس اطلاعات سهمیه بندی را برای کاربران راه دور (remote users)فراهم می کند.سهمیه بندی یعنی اینکه هر کاربر راه دور چه مقدار فضا می تواند برای ایجاد فایل و ... داشته باشد.این سرویس به همراه فعال کردن سرویس nfs خودکار فعال می شود.

rpc.idmapd : بعنوان نگاشت کننده نام ها و ID ها در NFS است.این سرویس UIDها و GID ها را به نام ها ترجمه می کند. فایل مرتبط با آن /etc/idmapd.conf می باشد.

rpc.gssd : یک پروتکل امنیتی است که در ایجاد ارتباط Security Conetxt های لازم برای کرنل را load می کند . نکته ضروری دیگر اینکه برای استفاده از NFS برروی سیستم تان مطمئن شوید که بسته های portmap, nfs-utils و nfs-utils-lib نصب شده باشد که در بخش نصب سرویس انها را نصب خواهیم کرد.

بسته سرور NFS شامل سه سرویس گنجانده شده در بسته‌های portmapو nfs-utils می‌باشد.

نصب و راه اندازی سرویس NFS

ابتدا باید از نصب بودن پکیج NFS اطمینان حاصل کنیم لذا با دستور زیر از سیستم query می گیریم :

#rpm -qa | grep nfs

در صورت نصب نبودن ، در سیستم های ردهت جهت نصب nfs از yum استفاده می کنیم :

#yum –y install nfs-utils

بعد از نصب ، باید اطمینان حاصل کنیم که آیا پکیج NFS بر روی سیستم نصب شده است یا خیر لذا با دستور زیر از سیستم query می گیریم :

#rpm -qa | grep nfs

سپس با دستور زیر شاخه ها و مسیرهایی که فایل های این سرویس در آن ایجاد شده است را چک می کنیم :

#rpm -ql nfs

و با این دستور هم اطلاعات لازم را در مورد پکیج این سرویس به دست می آوریم :

#rpm -qi nfs

سپس با دستور chkconfig مشخص می کنیم در چه runlevel هایی فعال باشد :

 # chkconfig --level 35 nfs on

و در انتها سرویس را reset می کنیم :

#service nfs restart

#exportfs -rva

توضیح دستور exportfs :

با این دستور می توانیم لیست export های سیستم را مشاهده کنیم.

r : این سوئیچ export های حذف شده را پاک و یک لیست به روز را نشان میدهد.

v : جزئیات را کاملا نشان میدهد.

a : export های جدید را نشان می دهد

برای اینکه بفهمیم چه پورتهایی به این سرویس اختصاص داده شده از این دستور بهر می بریم:

#rpcinfo -p

#showmount -e خروجی این دستور منابع اشتراکی شبکه را نشان می دهد

192.168.10.1

نکته : سرویسی که باعث می شود در هر بار ریست سرویس پورت جدیدی به آن تعلق بگیرد portmapper می باشد که در ورژن 4 هیچ کاربردی ندارد.

Exportدایرکتوری در NFS

توضیحات سمت سرور :

Exportکردن به این معنی است که تعیین کنیم کدام یک از دایرکتوری ها بر روی ماشین محلی برای کدام یک از ماشین ها راه دور و با چه مجوز هایی قابل mount شدن باشد. در NFSv2 و NFSv3 هر دارکتوری صادر شده بعنوان یک ورودی مستقل بود اما در NFSv4 این طور نمی باشد.

NFS سرویسی است که امکان اشتراک گذاری سیستم فایل مابین سیستم عامل های یونیکسی مانند لینوکس ها، بی اس دی ها و مکینتاش را فراهم می کند، حتی امکان استفاده از این سرویس بین سیستم عامل های یونیکسی و ویندوز سرور 2008R2 نیز فراهم شده است.

روند کار برای پیاده سازی سرویس NFS بدین گونه است که ابتدا باید سرویس ها و برنامه های لازم را هم برروی سرویس دهنده (NFS Server) و هم برروی (NFS Client) نصب کنیم. سپس تعیین کنیم چه دایرکتوری هایی که باید اشتراکی شوند و در نهایت دایرکتوری های اشتراکی شده در سیستم های کلاینتmount می کنیم مطلب دیگر اینکه لازم است Firewall نیز بصورت مناسب برای اجازه به ترافیک NFS تنظیم شود.

همانطور که گفته شد NFS دارای فایلی به نام exports در زیر دایرکتوری /etc است که باید آنرا ویرایش کنیم.

شکل کلی هر خط این فایل بصورت زیر است:

shared_directory IP or machin_name(OPTIONS)

که در آن shared_directory نام دایرکتوری از NFS Server است که برای دیگر ماشین های شبکه Shareشده است و IP or machin_name آدرس یا نام ماشین(های) کلاینتی است که مجاز به mountکردن دایرکتوری در سیستم محلیشان می باشند و OPTIONS هم گزینه هایی هستندکه بر شیوه استفاده کلاینت از دایرکتوری اشتراکی تاثیر می گذارند. در زیر تعدادی از مهمترین این Option توضیح داده شده است :

ro : مخفف Read Only است که کلاینت های تنظیم شده با این گزینه، تنها دسترسی فقط خواندنی روی دایرکتوری Mount شده دارند.

rw : مخفف Read Write که کلاینت های تنظیم شده با این گزینه، دسترسی خواندن و نوشتن بر روی دایرکتوری اشتراکی را دارند.

sync : این گزینه باعث می شود که سرور تنها پس از اینکه نوشتن داده ها (اعمال تغییرات) به طور کامل انجام شد، به کلاینت پاسخ دهد. این گزینه، بعنوان پیشفرض است و اگر ننویسید، همین گزینه در نظر گرفته می شود.

async : نقطه مقابل sync است و یعنی اینکه سرور قبل از تکمیل تغییرات بر رویStorage ، به در خواست های دیگر نیز پاسخ خواهد داد.و یوزرهای مقابل می تواند از هر مقدار فایل که کپی شده به صورت Real Time استفاده کند.

نکته : sync قابلیت اطمینان و امنیت بالاتری دارد اما async دارای کارائی و سرعت بالاتری می باشد.

no_root_squash : گزینه ای بسیار مهم در تنظیم یک دایرکتوری برای اشتراک گذاشتن است، چون که باعث کاهش امنیتی در سرور NFS خواهد. اگر از این گزینه استفاده می کنید باید بدانید که کاربر rootروی ماشین کلاینت، روی دایرکتوری اشتراک شده، دسترسی root مطابق با root ماشین سرور را خواهد داشت و توصیه می شود که از این گزینه استفاده نکنید.

root_squash:نقطه مقابل no_root_squash است. با استفاده از این گزینه، درخواست های آمده از uid=0و gid=0 به ک کاربر anonymous که به nobody user یا nfsnobody user شناخته می شود، نگاشت خواهد شد. یعنی دسترسی کاربر root روی ماشین کلاینت بر روی دایرکتوری Share شده، معادل با دسترسی کاربر root روی ماشین سرور **نخواهد بود** که این کار باعث افزایش امنیت خواهد شد. از این به بعد سطح دسترسی فایل مربوطه nfsnobody می باشد. به طور پیش فرض این آپشن فعال است.

no_subtree_check : این گزینه باعث عدم پیمایش در دایرکتوری بالا دستی دایرکتوری اشتراکی می شود .یعنی وقتی یک دایرکتوری اشتراکی می شود، با تنظیم این گزینه نمی گذاریم که کلاینت ها به

دایرکتوری های بالایی دایرکتوری اشتراکی شده دسترسی پیدا کنند و تنها به زیر دایرکتوری های، دایرکتوری اشتراکی دسترسی دارند.

نکته : هر یوزری که با nfs به منبعی وصل می شود ، یوزر پسورد پرسیده نمی شود به جای آن وقتی وارد یک سیستم می شود با یوزر nfsnobody به آن وصل می شود به این تکنیک squash گفته می شود.

در زیر چندین مثال از فرمت های گوناگون exports شدن یک منبع آورده شده است:

در مثال زیر دایرکتوری /nfs/nfs-share/ برای یک کلاینت با آدرس 192.168.10.1 به اشتراک گذاشته شده و در خط دوم برای دو ماشین به آدرس های 192.168.10.2و 192.168.10.1 به اشتراک گذاشته شده است.

/nfs-share/ 192.168.10.1(rw,sync)
/nfs-share/ 192.168.10.1(rw,sync) 192.168.10.2 (rw,sync)

در مثال زیر یک محدوده IP ها (IP Range)در نظر گرفته می شود.

/nfs-share/ 192.168.10.0/24(rw,sync)

و در این مثال یک ماشین عضو دامنه skywan13.local را شامل می شود.

/nfs-share/ pc2.skywan13.local(rw,sync)

در مثال زیر تمامی ماشین های عضو دامنه skywan13.local را شامل میشود.

/nfs-share/ *.skywan13.local(rw,sync)

برای دیدن مثال های بیشتر به exports 5 man رجوع کنید.

Example :

فرض کنید می خواهیم دایرکتوری/nfs-share/ را export کنیم تا کاربران شبکه 192.168.10.0/24 با گزینه های ro و root_squash از آن استفاده کنند.

```
#mkdir  /nfs-share
#groupadd  nfs-users
#chgrp  nfs-users  /nfs-share
#chmod  g+s  /nfs-share
#chmod  -R  777  /nfs-share
#vi  /etc/exports
   /mnt/nfs-share  192.168.10.10(rw,sync)
 #service  nfs  restart
```

Export دایرکتوری به سبک NFS v4

در ورژن های قبل از NFSv4 باید مسیر کامل دایرکتوری را هه در فایل eports و هم در خط mount وارد می کردیم . یعنی حتما باید از ریشه مسیردهی شود. به کار rootDirectory گفته می شود. این یک نقص امنیتی است که در ورژن 4 اصلاح گردیده است. جهت بررسی از ورژن 3 و 4 مثالهای آورده شده است .

NFS.v3

```
#mkdir -p /mnt/nfs-tes
#vi /etc/exports
   /mnt/nfs-test     *(ro)
#service nfs restart
#exportfs -rva
#mount -t nfs -o vers=4 192.168.10.1:/mnt/nfs-test
```

NFS.v4

```
#mkdir -p /mnt/nfs-test
#vi /etc/exports
   /mnt    *(ro,fsid=0)
  /mnt/nfs-test    *(rw,nohide)
#service nfs restart

#mount -t nfs -o vers=4 192.168.10.1:/nfs-test
```

یا اینکه به این صورت می نویسیم :

```
#mount -t nfs4 192.168.10.1:/nfs-test
```

توضیح :

Fsid=0 : این گزینه نشان می دهد /mnt با NFS ورژن 4 به اشتراک گذاشته شده است.

Nohide : یعنی محتوا را نشان دهد .

Export دایرکتوری Home

هیچ یوزری دایرکتوری home/ واقعی خودش را نباید در اختیار یوزر دیگری قرار بدهد. این کار بهترین روش امنیتی برای share کردن بین دو لینوکس می باشد ، مزیت این کار جلوگیری از rootdirectory ست .

```
#mkdir    /mnt/home
#mount  --bind  /home  /mnt/home    ۱
#vi  /etc/exports
   /mnt  *(ro,fsid=0)
   /mnt/home    *(rw,nohide)
#service  nfs  restart
```

--bind : این آپشن باعث می شود محتوای home اصلی با /mnt/home یکسان شود .

اختصاص پورت های ثابت به NFS

اگر از NFS زیر ورژن 4 استفاده می کنید می توانید پورتهای آن را ثابت کنید تا دیگر درگیر مشکلات فایروال نشوید. برای این کار به مسیر زیر رفته و فایل nfs را باز میکنیم. فایل nfs فایل اصلی پیکربندی سرویس NFS میباشد. هر زمان که NFS اجرا می شود محتوای این فایل را چک می کند تا اگر تغییری دید آنها را اعمال کند.

```
#vi  /etc/sysconfig/nfs
```

پس از اجرای دستور بالا باید خطوط زیر را Uncoment کنید یعنی علامت # ابتدای خطوط زیر را بردارید

```
LOCKD_TCPPORT=32803
LOCKD_UDPPORT=32769
MOUNTD_PORT=892
RQUOTAD_PORT=875
STATD_PORT=662
STATD_OUTGOING_PORT=2020
```

بعد از ذخیره تغییرات ، سرویس را ریست کرده و دوباره از پورت های NFS یک لیست می گیریم :

#service nfs restart
#rpcinfo -p

این کار برای کسانی مناسب است که می خواهند فایروال سیستم روشن باشد و این پورتها را درون آن باز کنند و در انتها به منظور پیکربندی iptables خطوط زیر را به فایل /etc/sysconfig/iptables اضافه می کنیم .

#/etc/sysconfig/iptables]

```
Firewall configuration written by system-config-firewall #
.Manual customization of this file is not recommended #
filter*
A INPUT -m state --state NEW -m udp -p udp --dport 2049 -j ACCEPT-
A INPUT -m state --state NEW -m tcp -p tcp --dport 2049 -j ACCEPT-
A INPUT -m state --state NEW -m udp -p udp --dport 111 -j ACCEPT-
A INPUT -m state --state NEW -m tcp -p tcp --dport 111 -j ACCEPT-
A INPUT -m state --state NEW -m udp -p udp --dport 32769 -j ACCEPT-
A INPUT -m state --state NEW -m tcp -p tcp --dport 32803 -j ACCEPT-
A INPUT -m state --state NEW -m udp -p udp --dport 892 -j ACCEPT-
A INPUT -m state --state NEW -m tcp -p tcp --dport 892 -j ACCEPT-
A INPUT -m state --state NEW -m udp -p udp --dport 875 -j ACCEPT-
A INPUT -m state --state NEW -m tcp -p tcp --dport 875 -j ACCEPT-
A INPUT -m state --state NEW -m udp -p udp --dport 662 -j ACCEPT-
A INPUT -m state --state NEW -m tcp -p tcp --dport 662 -j ACCEPT-
[INPUT ACCEPT [0:0:
[FORWARD ACCEPT [0:0:
[OUTPUT ACCEPT [0:0:
A INPUT -m state --state ESTABLISHED,RELATED -j ACCEPT-
A INPUT -p icmp -j ACCEPT-
A INPUT -i lo -j ACCEPT-
A INPUT -m state --state NEW -m tcp -p tcp --dport 22 -j ACCEPT-
A INPUT -j REJECT --reject-with icmp-host-prohibited-
A FORWARD -j REJECT --reject-with icmp-host-prohibited-
COMMIT
```

در آخر هم سرویس iptables را ریست می کنیم .

#service iptables restart]

دستورات مهم برای NFS

برخی از دستورات مهم برایNFS

showmount -e :
نمایش share های در دسترس روی ماشین محلی

showmount -e server-ip or hostname :
لیست shareهای در دسترس در سرور از راه دور(remote)

showmount -d :
لیست تمام زیر دایرکتوری‌ها(sub directorie)

exportfs -v :
نمایش یک لیست از فایل های shareشده و آپشن‌های روی یک سرور

exportfs -a :
Exportهمه‌یshare های لیست شده در /etc/exports، با توجه به نام

exportfs -u :
Unexportهمه‌یshare های لیست شده در /etc/exports،با توجه به نام

exportfs -r :
تازه کردن (Refresh) لیست سرور پس از تغییر /etc/exports

استفاده از دایرکتوری Export شده در کلاینت ها

سمت کلاینت :

پس از تمامی کارهای بالا، نوبت به پیکربندی کلاینت ها می رسد. پیش از هر کاری باید در هر کلاینت در مسیر مناسب یک دایرکتوری بسازیم تا دایرکتوری اشتراکی را به آن متصل کنیم. بهترین جا برای اتصال سیستم فایل های اشتراکی دایرکتوری mnt/ است.

#cd /mnt
#mkdir nfs-share

سپس به ازای هر خط فایل exports باید یک دستورmount بصورت زیر اجرا کنیم.

#mount -t nfs -o vers=4 192.16.10.1:0/mnt/nfs-share
#mount -t nfs4 192.168.10.1:/mnt/nfs-share یا به این صورت:

مشکل دستور mount موقتی بودن آن است یعنی پس از خاموش شدن سیستم، نقطه اتصال دایرکتوری اشتراکی mount شده از بین می رود اگر می خواهید که دایرکتوری مورد نظر بصورت دائمی باشد با یک خط به فایل /etc/fstab اضافه کنید .

#vi /etc/fstab
192.168.10.1:/mnt/nfs-share /mnt nfs defaults 0 0

با اضافه کردن این خط به فایل fstab نقطه اتصال ما به دایرکتوری اشتراکی مورد نظر دائمی می باشد. در صورتی که با خطای زیر در هنگام mount کردن یک دایرکتوری در کلاینت موجه شدید دلیل آن بسته بودن پورت های 111 و 2049 است. بطور کلی برای تست می توانید در سرور و کلاینت فایروال را خاموش یا غیر فعال کنید.

mount.nfs: mount to NFS server '192.168.10.1’ failed: System Error: No route to host

سرویس nfs برروی پورت 2049 و portmap برروی پورت 111 کار می کنند.

مبحث Autofs

وقتی یک کلاینت دایرکتوری share شده توسط سرور را mount می کند یعنی یک session و کانال ارتباطی شکل گرفته است ، حال اگر ارتباط سرویس دهنده قطع شود کلاینت متوجه این موضوع نشده و سیستم درگیر چک کردن مداوم آن می شود . این حالت در کار سیستم کلاینت اخلال ایجاد کرده و کلا آن را از کار می اندازد و به هیچ وجه اجازه umount شدن را نمی دهد مگر اینکه سیستم را ریست کنیم . این که ما همیشه یک دایرکتوری را به صورت ثابت mount شده داشته باشیم یک ضعف به حساب می آید چون اگر سرور از کار بیافتد کلاینتها در کارکرد با مشکل مواجه خواهند شد. برای رفع این ضعف سرویس autofs معرفی شد. لذا هر وقت به دایرکتوری خاصی احتیاج داشتیم autofs آن را برایمان mount کرده و بعد از طی شدن زمان خاصی آن دایرکتوری را umount می کند.

لازم است این سرویس بر روی سیستم کلاینت نصب شود چون کلاینت استفاده کننده از دایرکتوری اشتراکی میباشد.

با نصب این پکیج در زیر دایرکتوری etc/ تعدادی فایل ایجاد می شود که با auto شروع می شوند :

auto.ftp
auto.master
auto.misc

auto.master کانفیگ اصلی و گلوبال سرویس auto.master در این فایل قرار دارد. در این فایل آدرس دایرکتوری سیستم local را که قرار است چیزی در آن mount شود را مشخص می کنیم.

```
#vi  /etc/auto.master

/media          /etc/auto.media          ==timeout=20
```

توضیح خطوط تعریف شده :

/media : آدرس دایرکتوریی از سیستم local که قرار است چیزی درون آن mount شود.

/etc/auto.media : فایل کانفیگ دایرکتوری که قرار است چیزی درون آن mount شود.

==timeout=20 : این خط مشخص میکند بعد چه مدتی دایرکتوری mount شده umount گردد.(بر حسب ثانیه)

در مرحله دوم باید فایل auto.media را ایجاد کنیم. برای این کار از فایل auto.misc یک کپی با نام auto.media می سازیم.

```
#cp /etc/auto.misc /etc/auto.media
```

سپس فایل auto.media را باز کرده تغییرات لازم را اعمال می کنیم.

```
#vi /etc/auto.media

nfs-share -rw,sync 192.168.10.1:/mnt/nfs-share
```

توضیح خطوط تعریف شده :

nfs-share : نام فولدری که قرار است در سیستم کلاینت mount شود. این نام به دلخواه انتخاب می گردد.

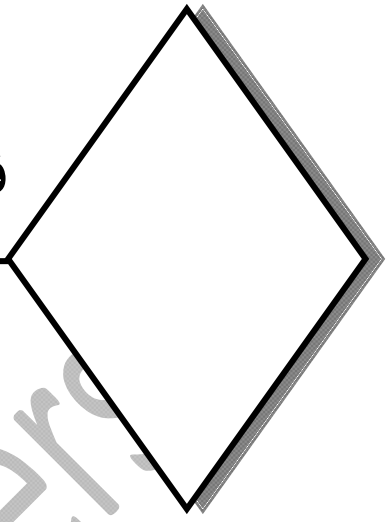
-rw,sync : دایرکتوری مورد نظر به چه صورتی mount شود.

192.168.10.1:/mnt/nfs-share : در اینجا آدرس سروری که قرار است از دایرکتوری export شده آن استفاده کنیم را وارد می کنیم.

بعد از انجام تغییرات ، autofs را ریستارت می کنیم.

```
#service autofs restart
```

ما مشخص کردیم دایرکتوری /mnt/nfs-share از سرور به دایرکتوری /media کلاینت و با نام nfs-share مونت شود. وقتی در کلاینت وارد /media می شویم و ls میگیریم محتویات پیش فرض در آن قرار دارد. ولی به محض اینکه ئستور cd nfs-share را تایپ کنیم دایرکتوری /mne/nfs-share برای ما mount می شود. دقت کنید حتما باید در مسیر مشخص شده باشیم و اگر طبق زمان تعریف شده استفاده ای از دایرکتوری mount شده نداشته باشیم توسط این سرویس umount می گردد.



فصل پنجم

زمان بندی فرایندها توسط Cron

زمان بندی اجرای برنامه ها توسط Cron

مقدمه :

Cron بر گرفته از chromos یک کلمه یونانی به معنای زمان است که به خدای زمان یونان باستان گفته می شد و در یونیکس و سولاریس برای خودکارسازی انجام دستور ها و پروسس ها از آن استفاده می شود.
برنامه ریزی و زمانبندی برای انجام فعالیت های مختلف در لینوکس امر بسیار مهمی هست که شاید خیلی از ما روزانه با آن سر و کار داشته باشیم. برخی اوقات ما دستور یا دستورات مورد نظر خود را به طور مستقیم از شل میخواهیم و انجام میشود اما برخی اوقات نیاز هست تا در زمان (یا زمان هایی) مقرر سیستم عامل به طور خودکار برای ما کاری را انجام دهد. برای مثال نیاز داریم تا سیستم برای ما هرروز در ساعتی که مشخص میکنیم یکاپ بگیرد. از مهمترین راه های زمانبندی برنامه ها در سیستم عامل های شبه یونیکس استفاده از نرم افزار cron و دستور at می باشد. زمانی که به صورت یه دوره متناوب بخواهیم فعالتی انجام شود میتوانید از Crond و Cron Table ها کمک بگیریم.

مدیریت زمانبندی اجرای دستورات توسط این برنامه در قالب یک فایل به نام crontab که بر گرفته از crontable است و معمولا در مسیر /etc/crontab/ قرار دارد انجام می شود. وقتی ما خطی را به فایل crontab اضافه می کنیم ، برای اعمال شدن آن حتما باید سرویس cron ریست شود تا cron مجبور به خواندن فایل پیکربندی شده و اسکریپت جدید را در صف اجرا قرار دهد.

همچنین هرکاربر دارای یک فایل شخصی cron است که در مسیر var/spool/cron قرار دارد. پروسه cron هر یک دقیقه یکبار به فایل crontab رجوع کرده و از آن stat می گیرد ، اگر تغییری در این فایل اعمال شده باشد یکبار از اول این فایل را می خواند.

نصب و راه اندازی سرویس Cron

ابتدا باید از نصب بودن پکیج cron اطمینان حاصل کنیم لذا با دستور زیر از سیستم query می گیریم. در سری 6 به بعد CentOS نام این سرویس به cronie تغییر یافته است:

#rpm -qa | grep cronie

در صورت نصب نبودن ، در سیستم های ردهت جهت نصب cron از yum استفاده می کنیم :

#yum –y install cronie

بعد از نصب ، باید اطمینان حاصل کنیم که آیا پکیج cron بر روی سیستم نصب شده است یا خیر لذا با دستور زیر از سیستم query می گیریم :

#rpm -qa | grep cronie

سپس با دستور زیر شاخه ها و مسیرهایی که فایل های این سرویس در آن ایجاد شده است را چک می کنیم :

#rpm -ql cronie

و با این دستور هم اطلاعات لازم را در مورد پکیج این سرویس به دست می آوریم :

#rpm -qi cronie

سپس با دستور chkconfig مشخص می کنیم در چه runlevel هایی فعال باشد :

 # chkconfig --level 35 crond on

و در انتها سرویس را reset می کنیم :

#service crond restart

مسیرها و فایل های اضافه شده به سیستم

با نصب این سرویس چندین مسیر مهم در سیستم اضافه می شود که در زیر مختصرا توضیح داده می شود :

/etc/cron.d
/etc/pam.d/crond
/etc/rc.d/init.d/crond
/etc/sysconfig/crond
/etc/crontab
/usr/bin/crontab

/etc/cron.d: در این دایرکتوری محتوایی وجود ندارد. فایل های cron ای که می سازیم در این دایرکتوری قرار می گیرد.

/etc/pam.d/crond: مکانیزم احراض هویت cron توسط pam اجرا می شود. در این دایرکتوری فایل کانفیگ آن قرار دارد.

/etc/rc.d/init.d/crond: سرویس cron زیر مجموعه init اداره میشود. در این مسیر اسکریپت startup سرویس cron قرار دارد.

/etc/sysconfig/crond: اگر بخواهیم سرویس cron با یکسری فیچر خاص استارت شود باید فیچرهای مورد نظر را در این فایل قرار دهیم. مثل debugging mod و یا اجرای سرویس روی یک کارت شبکه خاص .

/etc/crontab: در این مسیر فایل اصلی پیکربندی cron قرار دارد.

/usr/bin/crontab: و در این جا هم فایل دستور cron قرار گرفته است.

چندین فایل وجود دارد که برای تمامی سیستم بوده و مالک این فایل ها کاربر root است. همه کاربران دارای یک فایل crontab مخصوص خود هستند ولی فایلهای زیر متعلق به کل سیستم می باشند:

- etc/cron.d/ : دایرکتوری شامل چندین فایل
- etc/cron.daily/ : زمانبندی برای انجام روزانه
- etc/cron.hourly/ : زمانبندی بصورت ساعتی

- etc/cron.monthly/ : زمانبندی ماهانه
- etc/cron.weekly/ : زمانبندی هفتگی

به طور مثال فایل هایی که در شاخه /etc/cron.daily/ قرار دارند- بطور روزانه اجرا خواهند شد. در زیر نمونه ای از محتویات این دایرکتوری را مشاهده میکنید:

```
# ls -l /etc/cron.daily/
total 56
-rwxr-xr-x 1 root root  ۳۱۱ Jun 20  ۲۰۱۰ ۰anacron
-rwxr-xr-x 1 root root 15399 Apr 20  ۲۰۱۲ apt
-rwxr-xr-x 1 root root  ۳۱۴ Mar 30  ۲۰۱۲ aptitude
-rwxr-xr-x 1 root root  ۵۰۲ Mar 31  ۲۰۱۲ bsdmainutils
-rwxr-xr-x 1 root root  ۲۵۶ Apr 13  ۲۰۱۲ dpkg
-rwxr-xr-x 1 root root  ۳۷۲ Oct  ۵  ۲۰۱۱ logrotate
-rwxr-xr-x 1 root root ۱۳۶۵ Mar 31  ۲۰۱۲ man-db
-rwxr-xr-x 1 root root  ۶۰۶ Aug 17  ۲۰۱۱ mlocate
-rwxr-xr-x 1 root root  ۲۴۹ Apr  ۹  ۲۰۱۲ passwd
-rwxr-xr-x 1 root root  ۳۸۳ Apr 25  ۲۰۱۲ samba
-rwxr-xr-x 1 root root ۲۹۴۷ Apr  ۲  ۲۰۱۲ standard
```

توضیح فایل پیکربندی سرویس Cron

سرویس cron در دو scope کار می کند : 1) Global 2) User Base و به طور کلی دو نوع فایل برای cron مورد استفاده قرار میگیرد User cron table & system cron table .

1) **Global (system cron table):** این نوع از کانفیگ مربوط به مدیریت پروسه های سرور است و ربطی به یوزرها نداشته و فقط در اختیار یوزر root می باشد. فایل پیکربندی گلوبال cron در مسیر /etc/crontab و فایل های سیستم نیز برای زمانبندی فعالیت ها در/etc/cron.d/ قرار دارد. نوشتن این نوع از تنظیمات فقط در اختیار یوزر root می باشد و یوزرهای عادی فقط اجازه دیدن فایل ها را دارند.
2) **User Base (user cron table):** ولی این نوع از کانفیگ را یوزرهای عادی سیستم برای انجام کارهای شخصی خودشان انجام می دهند. فایل های یوزر در آدرس /var/spool/cron میباشد و همانطور که از نام آن پیداست این فایل توسط کاربران ایجاد میشود .

توضیحات نوشتن cron در بخش بعدی آمده است. فایل crontab ، فایل اصلی پیکربندی این سرویس می باشد تنظیمات global از طریق این فایل به سیستم اعمال می گردد. در ادامه به تشریح خطوط مهم این فایل می پردازیم :

#vi /etc/crotab

```
SHELL = /bin/bash
PATH = sbin/bin: /usr/sbin: /usr/bin
MAILTO = root
HOME = /
 *  *  *  *  *    root    run-parts    /etc/cron.daly
```

SHELL = /bin/bash: این مشخص می کند اسکریپتهای که می خواهیم با cron اجرا کنیم با چه شلی اجرا شوند.

PATH = sbin/bin: /usr/sbin: /usr/bin: یک اسکریپت مجموعه ای از دستورات است که با هم ترکیب شده اند. این خط مشخص می کند دستورات داخل اسکریپت از چه مسیرهایی اجرا شوند. اگر مسیر دستورات داخل اسکریپت در اینجا درج نشود آن بخش از اسکریپت و یا همه آن کار نمی کند.

MAILTO = root: به طور پیش فرض سرویس cron خروجی کار خود را بر روی صفحه نمایش نشان نمی دهد بلکه در حالت پیش فرض cron خروجی را برای email کاربر مالک job میفرستد. بهتر است به جای ایمیل شدن خروجی cron ، آن را به یک فایل redirect کنید تا دسترسی به آن راحت تر باشد. این دستور در صورتی اجرا خواهد شد که از /dev/null/ برای redirect استفاده نکرده باشید. اگر به آخر فایل مربوطه چیزی اضافه نشد یعنی اینکه خط نوشته شده ایراد دارد.

HOME = /: مسیر خانگی cron را نشان می دهد.

*** * * * * root run-parts /etc/cron.daly/**

این خط مهمترین قسمت این فایل است که دارای 8 فیلد می باشد:

Field 1 (Minuets) 0 – 59:

فیلد اول نشان دهنده دقیقه بوده و از 0 تا 59 متغیر است. محدودیت cron در دقیقه است. یعنی حداقل زمان بین دو کار یک دقیقه می باشد و نمی تواند به ثانیه کار کند.

Field 2 (Hour) 0 – 23:

فیلد دوم نشان دهنده ساعت بوده و از صفر تا 23 متغیر است.

Field 3 (Day of Month) 1 – 31:

این فیلد نشان دهنده روز از ماه می باشد.

:Field 4 (Month) 1 – 12

فیلد چهارم به ماه اشاره دارد.

:Field 5 (Day of Week) 0 – 7

فیلد پنجم مربوط به هفته می باشد. در cron یکشنبه برابر با صفر است و صفر با 7 فرقی ندارد لذا به جای 7 از 6 استفاده می شود.

نکته : ستاره (*) در cron معنی هر می دهد مثل هر ساعت یا هر دقیقه.

:Field 6 (root)

این فیلد مشخص می کند اسکریپتی که مسیر آن را در اینجا مشخص کرده ایم توسط چه یوزری اجرا می شود.

:Field 7 (run-parts)

اگر ما تعداد زیادی اسکریپت را درون یک دایرکتوری قرار دهیم و بخواهیم بوسیله cron اجرا شود تنها راه آن استفاده از دستور run-parts می باشد. برای اجرای یک مجموعه اسکریپت در زمان مشخص از run-parts استفاده میکنیم. این دستور اسکریپتهای یک دایرکتوری را به ترتیب حروف الفباء ، به صورت تک تک پشت سر هم اجرا می کند. این دستور با پکیج crontabs بر روی سیستم نصب می شود.

دستور `witch run-parts` #rpm -qf مشخص میکند run-parts مربوط به چه پکجی است.

:Field 8 (/etc/cron.daily)

فیلد آخر هم اسکریپت اجرایی و مسیر آن را مشخص می کند.

نکته :

1) در خطوط cron نباید از دبل کوتیشن استفاده کنید.
2) با دستور زیر می توانیم از زمان آخرین تغییرات یک فایل مطلع شویم.

 # stat /etc/crontab

3) و با این دستور می توانیم log تغییرات crontab را مشاهده کنیم.

 # tail -f /var/log/cron

برای ویرایش crontab یک کاربر، از طریق کاربر root بدین شکل عمل میکنیم:

#crontab -e -u username

استفاده از سرویس Cron

همانطور که گفته شد اسکوپ Global مخصوص یوزر root است و فقط root حق edit آن را دارد. یوزرهای عادی فقط اجازه دیدن فایل Global را دارند حال اگر یوزرهای عادی سیستم بخواهتد یک cron تعریف کنند باید از اسکوپ UserBase بهره ببرند. مکان ذخیره سازی فایل های cron یوزرها در آدرس /var/spool/cron/ میباشد و همانطور که از نام آن پیداست این فایل توسط کاربران ایجاد میشود. این دایرکتوری به طور پیش فرض خالی است و cron تعریف شده توسط یوزرها در اینجا ذخیره می شود. یوزر عادی برای تعریف cron باید از دستور crontab -e استفاده کند که editor پیش فرض آن ، vi است.

نکته مهمی که وجود دارد این است که در cron جدید دیگر نیازی نیست که تمام محتویات یک فایل cron را واردآن کنیم بلکه فقط خط مربوط به اجرای اسکریپت و زمانبندی را وارد می کنیم.

```
# crontab   -e
2  17  *  *  *  ls /
```

هر فایل cron ای که در /var/spool/cron/ ذخیره شود با نام یوزر سازنده آن ذخیره می شود و فقط همان یوزر و یوزر root حق خواندن و edit آن را دارند. یوزر root با دستور زیر می تواند cron متعلق به یک یوزر را edit کند.

```
# crontab   -u   skywan13   -e
```

اگر یوزری بخواهد چند cron داشته باشد باید تمام آنها را در یک فایل نوشته و ذخیره کند. اگر هم فیلدی را به اشتباه در فایل cron وارد کند در هنگام ذخیره به کاربر هشدار داده می شود. چک کردن فایل cron توسط crontab انجام می شود.

نکته مهم : یک سری علایم در نوشتن ورودی cron استفاده میشود که سعی شده با مثال توضیح داده شود:

* (ستاره) برای نادیده گرفتن یک فیلد در نظر گرفته شده، یعنی اگر مثلا در فیلد ساعت بود بدون توجه به این فیلد سر هر ساعت دستور اجرا میشود و یا اگر در فیلد دقیقه بود سر هر دقیقه و....

, (کاما) در هر فیلد که احتیاج داریم چندین بار دستور در ساعات مختلف اجرا بشود کاربرد دارد مثلا در فیلد دقیقه 15,30 به معنای اجرا در دقیقه های 15 و 30 یا در فیلد ساعت 2,9 به معنای اجرا در ساعتهای 9 و 2 میباشد و....

– (خط تیره) برای تعیین یک بازه زمانی است مثلا در فیلد روزهای ماه اگر بخواهیم بین روزهای 8 تا 15 دستور اجرا بشود بدین صورت مینویسیم 15-8 و....

همانطور که پیش تر در بالای همین مطلب ذکر شد، شما می توانید با دستور "crontab -e"یک فایل crontab بسازید. به هر حال ممکن است شما از قبل یک فایل crontab داشته باشید. برای مشخص کردن فایل خود، دستور زیر را وارد می کنیم :

crontab -u <username> <crontab file>
crontab -u skywan13 sky.log

سپس دستور زیر را وارد کنید تا فایل crontabکاربر skywan13 با نام crontab در پوشه آن خانگی آن ذخیره شود.

crontab –u skywan13 ~/crontab

و برای حذف فایل crontab دستور زیر را در cli وارد می کنیم :

 # crontab -r

برای لیست کردن jobهایی که با crontab -e اضافه کردیم میتوانیم با استفاده از دستور زیر cron jobها را لیست شده ببینیم:

 # crontab -l

تا اینجا دیدید که برخلاف ظاهر پیچیده ، crontab به آسانی تنظیم میشود و ابزاری کاربردی و مهم در فرآیند مدیریت سیستم میباشد.

نکته مهم : برای استفاده از cron باید از crontab جهت load کردن jobها استفاده کرد که برای این منظور 2 راه پیش رو داریم :

۱-استفاده از crontab -e ,یعنی مستقیما دستور مربوط رو در crontab مینویسیم.

۲-از طریق فایل یعنی ابتدا یک فایل متنی میسازیم و طبق قوانین توضیح داده شده در بالا ورودی مناسب با شرایط خودتان رو در فایل مورد نظر قرار داده و سپس فایل را Load میکنید.

قبل از Load با استفاده از crontab -l لیست jobهای جاری را تهیه کرده و هر کدام را که لازم داریم در فایل جدید مینویسید چون با load این فایل تمامی jobهای گذشته پاک خواهند شد. جهت درک بهتر موضوع به مثال زیر توجه کنید:

```
#touch /skywan13/mycrontab
#echo "30  4  *  *  *    ls –s   / skywan13/web >> / skywan13/webdirlist.log
2>&1" > / skywan13/mycrontab
#crontab  -u  skywan13/  skywan13/mycrontab
#crontab -l
```

با اجرای crontab -l از load شدن فایل اطمینان حاصل میکنیم. توجه داشته باشید که برای افزودن job جدید یا ویرایشjob های موجود کافی است با ویرایش فایل mycrontab و افزودن دستور مورد نظر در خط جدید و پاک کردن crontab با استفاده از crontab -r فایل را دوباره Load کنیمcron jobهای تحت کاربری که فایل را با آن load کردیم اجرا خواهد شد.

محدود کردن دسترسی یوزرها برای استفاده از Cron

اگر بخواهیم برای استفاده از cron دسترسی یوزر و یا گروهی از یوزرها را محدود کنیم و یا فقط به بعضی از یوزرها دسترسی بدهیم باید در فایل های cron.deny و cron.allow تغییراتی اعمال کنیم. این دو فایل در زیر شاخه /etc قرار دارند. اگر هم وجود نداشتند مهم نیست چون می توانیم آنها را بسازیم.
اگر فایل cron.allow موجود باشد **حتما** باید اسامی یوزرهای مجاز در آن وارد شود، در غیر این صورت به هیچ یوزری استفاده از cron داده نمی شود.

اگر هم فایل cron.allow وجود نداشته باشد چک می شود آیا cron.deny وجود دارد یا خیر. اگر موجود باشد و نام یوزری در آن آمده باشد از دسترسی آن به cron جلوگیری می شود. اگر هیچ کدام از این دو فایل وجود نداشته باشد اجازه کار با cron به همه یوزرها داده می شود.

نکته مهم : چون ارجحیت اجرا با cron.allow است ابتدا این فایل مورد بررسی قرار می گیرد در صورت وجود نداشتن و یا خالی بودن آن ، فایل cron.deny مورد بررسی قرار می گیرد.

مثالهای برای Cron

برای درک بهتر ، مثال هایی به همراه توضیحات آورده شده است. ابتدا مثالهایی ساده بیان می شود و سپس مثال هایی پیشرفته مورد بررسی قرار می گیرد.

نوشتن فایل crontab ممکن است برای اولین بار کمی گیج کننده به نظر برسد. بنابراین در زیر تعدادی مثال ساده به همراه توضیح آورده شده است تا ابتدا با شیوه نوشتن cron آشنا شوید:

هر دقیقه اجرا می شوند<command> * * * * *

هر ساعت رأس دقیقه ۳۰ام اجرا می شوند<command> * * * * 30

هر روز ساعت ۶:۴۵ صبح اجرا می شوند<command> * * * 6 45

هر روز صبح ساعت ۶:۴۵ بعد از ظهر اجرا می شوند<command> * * * 18 45

هر یکشنبه ساعت ۱ صبح(بامداد؟) اجرا می شوند<command> 0 * * 1 00

هر یکشنبه ساعت ۱ بامداد اجرا می شوند<command> 7 * * 1 00

هر یکشنبه ساعت ۱ بامداد اجرا می شوند<command> Sun * * 1 00

اولین روز هر ماه ساعت ۸:۳۰<command> * * 1 8 30

ماه پنجم هرروز در ساعت 5 هر دقیقه یکبار <command> 5 * 5 *

روز اول ماه اول سال <command> * 1 1 0 0

هر پنج دقیقه راس هر ساعت <command>* * * * */5

از دقیقه 5 هر 15 دقیقه به 15 دقیقه <command>* * * * 5/15

از روز سوم ، 3 روز به 3 روز <command>* * 3/3 * *

هر روز، هر ساعت دقیقه 20 و 50 <command>* * * 20,50

دقیقه 5، هر 3 ساعت یکبار روزهای اداری 1-5 * * */3 5

هر روز بین 5 الی 10 صبح <command>* * * 5,6,7,8,9,10 *

برای اجرای برنامه ها در startup از طریق crontab ، در زمان بوت سیستم کافیست برنامه مورد نظر را بدین طریق در crontab قرار دهیم:

@reboot /path/to/my/program @reboot updatedb

این برنامه در هر بار بوت مجدد سیستم اجرا خواهد شد. در ادامه مثالهایی برای درک بهتر موضوع آورده شده است :

@reboot <command> هنگام بوت سیستم اجرا می شود

@yearly <command> هر سال اجرا می شود[* 1 1 0 0]

@annually <command> هر سال اجرا می شود[* 1 1 0 0]

@monthly <command> هر ماه اجرا می شود[* * 1 0 0]

@weekly <command>هر هفته اجرا می شود[0 * * 0 0]

@daily <command> هر روز اجرا می شود[* * * 0 0]

@midnight <command> هر روز اجرا می شود[* * * 0 0

@hourly <command> هر ساعت اجرا می شود[* * * * 0]

برای اجرای چندین دستور پی درپی، آنها را با استفاده از "&&" به صورت پی در پی بنویسید. مثال زیر ابتدا دستور command_01 و سپس دستور command_02را در هر روز اجرا می کند :

@daily <command_01> && <command_02>

مثال های پیشرفته :

1) در مثال زیر در ساعت 3:12 دقیقه صبح هر روز از هر ماه، cron با اجرای این خط شروع به تهیه پشتیبان از /etc/ میکند. **dev/null 2>&1** به معنی ارسال هرگونه خروجی استاندارد به dev/null که سطل آشغال لینوکس است و هدایت خطاهای استاندارد 2 (standard error) به همان جایی که خروجی استاندارد رفته است بدون هر گونه خروجی در ترمینال یا هر نقطه دیگری از سیستم. اگر بجای >> از > استفاده کنیم در هر دفعه باز اجرا شدن دستور و فرستادن خروجی به فایل مربوطه , محتویات فایل پاک و خروجی جدید جایگزین میشود ولی با استفاده از >> خروجی به انتهای فایل افزوده خواهد شد.

12 3 * * * root tar cfz /tohid/backup.tar.gz /etc >> /dev/null 2>&1

2) در این مثال در تاریخ یکشنبه 7 oct ساعت 15:30 از ماه دهم روز 7 پیغام تبریکی برای کاربر ارسال میشود.

30 15 7 10 0 * root echo "happy birthday,tohid!!"

3) مثال بالا را به شیوه زیر هم میتوان نوشت، دقت کنید حروف اول روز و یا ماه باید بزرگ نوشته شود:

30 15 7 Oct Sun * root echo "happy birthday,tohid!!"

4) اگر می خواهید که کاربر tohid یک دستور را دقیقه 15 ، بعد از هر ساعت بدون توجه به تاریخ اجرا کند بدین طریق عمل میکنیم:

```
15 * * * * tohid  echo  "I'm skywan13 Remember"
```

5) یا اگر تمایل دارید هر 15 دقیقه اجرا بشود از این خط استفاده می کنیم :

```
*/15 * * * * tohid  echo "I'm skywan13 Remember"
```

6) برای اجرا شدن یک دستور هر 2 ساعت یعنی در 2 صبح, 4 صبح, 6 صبح و....12 ظهر, 2 بعد از ظهر, 4 بعدازظهر و.... از این خط بهره می بریم .

```
0 */2 * * * tohid  echo  "I'm skywan13 Remember"
```

7) با اقزودن ' , ' در یک فیلد امکان چندین مرتبه اجرا شدن دستور مورد نظر بدست میاد. مثلا برای اجرای دستور در ۱۵ و ۳۰ دقیقه گذشته از هر ساعت بدین طریق عمل میکنیم:

```
15,30 * * * * tohid echo "I'm skywan13 Remember"
```

8) برای اجرای دستور مورد نظر در یک زمان مشخص، برای اولین هفته ماهی که شما تمایل دارید در فیلد روز از ۱-۷ استفاده میکنیم (خط تیره به معنی تا و یا الی می باشد). در این خط مشخص کرده ایم در ۱۵ و ۳۰ دقیقه گذشته هر دو ساعت از روز 1 تا 7 این خط اجرا شود :

```
15,30 */2 1-7 * * tohid echo "I'm skywan13 Remember"
```

9) خط زیر هر 2 دقیقه به 2 دقیقه به صفحه ایندکس یک وب سرور وصل شده و آن را دانلود کرده و سپس در دایرکتوری کاربر ذخیره می کند:

```
*/2 * * * wget http://192.168.1.10/index.php >> /home/skywan13/cron
```

10) در ساعت 12:30 هر روز فایل های خالی دایرکتوری /tmp را پاک می کند. دستور find برای اجرا شدن توسط crond از مجوز های کاربر root استفاده می کند. باید ابتدا دستور crontab -e را اجرا کنید تا فایل crontab شما برای ویرایش باز شود.

```
30 0 * * * root find/tmp -type f -empty -delete
```

11) دستور زیر در 10 ژوئن ساعت 8:30 صبح یک backup توسط اسکریپتی گرفته می شود. توجه کنید برای ساعت 8:30 شب باید از 20:30 استفاده کنید.

```
30 8 10 06 * root /home/skywan13/full-backup
```

برای گرفتن backup در ساعت 11 ظهر و 4 بعد از ظهر(16) هر روز از دستور زیر استفاده کنید:

```
00 11,16 * * * /home/skywan13/bin/incremental-backup
```

12) برای انجام در روز های خاص از هفته مثل روز دوم هفته (یک شنبه روز اول هفته میلادی و عددش 0 است و دوشنبه روز دوم و عددش 1 است) تا روز ششم یعنی جمعه هر هفته بین ساعت های 9 صبح تا 6 عصر انجام می شود.

```
00 09-18 * * 1-5 /home/skywan13/bin/check-db-status
```

13) برای اجرای وظایف در یک محدوده زمانی خاص مثل بین ساعت 9 صبح تا 6 عصر (18) از دستور زیر استفاده می کنیم.

```
00 09-18 * * * /home/skywan13/bin/check-db-status
```

14) می خواهیم اسکریپتی به نام clean.cache که cache سیستم را پاک می کند، هر 10 روز یکبار اجرا شود. لذا فایل اسکریپت مربوطه را در مسیر /etc/cron.daily/ قرار می دهیم. محتوای اسکریپت به شرح زیر می باشد :

```
#!/bin/bash
# A sample shell script to clean cached file from lighttpd web server
CROOT=”/tmp/cachelighttpd/”
DAYS=10
```

```
LUSER=”lighttpd”
LGROUP=”lighttpd”
# start cleaning
/usr/bin/find ${CROOT} -type f -mtime +${DAYS} | xargs -r /bin/rm
# if directory deleted by some other script just get it back
if [ ! -d $CROOT ]
then
/bin/mkdir -p $CROOT
/bin/chown ${LUSER}:${LGROUP} ${CROOT}
fi
```

سپس برای اجرایی کردن اسکریپت از خطوط زیر استفاده می کنیم:

```
# crontab  -l  >  /backup/cron/cronjobs.bakup
# crontab  -u  username  -l  >  /backup/cron/cronjobs_username.bakup
```

اجرا برنامه ها با واسط گرافیکی کاربر

برنامه های که کاربر می خواهد اجرا کند به دو صورت می باشند :

برنامه هایی که دارای محیط گرافیکی کاربر هستند (GUI) و نیاز به تعامل با سرویس دهنده پنجره X هستند مانند مرورگرFirefox و برنامه هایی که بدون GUI هستند که این برنامه ها خروجی و ورودی آن ها در پوسته خط فرمان است و برای اجرا شدن نیازی به تعامل با سرویس دهنده پنجره X ندارند.

برنامه هایی که در دسته دوم (CLI) قرار می گیرند بدون هیچ مشکلی به وسیله Cron اجرا می گردند. اما برنامه های دسته اول که دارای GUI هستند فقط با نوشتن دستور مورد نظر اجرا نخواهند شد. قبل از دستور باید به سرویس دهنده X بگویید که برنامه در کدام صفحه نمایش برای شما اجرا شود.

برای این منظور قبل از دستور مورد نظر از env DISPLAY=:0 استفاده می کنیم. به عنوان مثال برای اجرای برنامه gedit در ساعت 10:25 هر روز صبح در فایل crontab خط زیر را وارد می کنیم :

```
25  10  *  *  *   env DISPLAY=:0   /usr/bin/gedit
```

DISPLAY=:0 به Cron می گوید که برنامه در صفحه جاری (desktop) اجرا شود.

و اگر دارای چندین صفحه نمایش هستید از دستور زیر استفاده می کنیم.

DISPLAY=:0.0 به Cron می گوید که برنامه در صفحه نمایش اول و در صفحه جاری اجرا شود. این دستور حتما باید قبل از اجرا شدن یک برنامه GUI به وسیله Cron اجرا شود. شما می توانید این دستور را در قسمت Startup Applications قرار دهید تا هنگام بوت شدن سیستم اجرا شود

```
25  10  *  *  *   env DISPLAY=:0.0   /usr/bin/gedit
```

زمانبندی اجرای فرامین توسط Anacron

anacron را میتوان برای اجرای دستورات بصورت دوره ای استفاده کرد که این فواصل به روز تعیین می شوند. برخلاف cron , تصور نمی کند که دستگاه 24 ساعت بطور مستمر روشن است. از این رو آن را میتوان در دستگاههایی که 24 ساعت روشن نیستند برای کنترل jobهای روزانه هفتگی و ماهانه که بطور معمول با cron صورت میگیرد استفاده کرد. بنابراین درanacron موضوع اصلی اجرا شدن job هاست نه اجرا شدن اونها سر ساعت و دقیقه تعیین شده همانند cron.

cron-daily در لینوکس CentOS در ساعت چهار و دو دقیقه اجرا می شود حال فرض کنید سیستم راس ساعت 4 خاموش شود و در ساعت 5 مجددا UP شود و این مدت 23 ساعتی که باید به چهار و دو دقیقه بعدی برسد ممکن است در سیستم کلی اتفاق رخ دهد که ما در این صورت یک روز کامل را از دست داده ایم. یا فرض کنید سیستمی داریم که باید 5 روز به 5 روز از آن بکاپ بگیریم. اگر 5 روز اول را از دست بدهیم و بکاپ دوم را با cron بگیریم در اصل ده روز را از دست داده ایم. این خلاء یک نقص برای سیستم و سرویس cron محسوب می شود. این ضعف با سرویسی به نام anacron پوشش داده می شود.

کار anacron اجرای cron ای است که از دست رفته و زمان اجرای آن گذشته و اجرا نشده است. anacron برای سیستم های است که UP و DOWN زیادی دارند مثل سیستمهای خانگی و یا کلاینتهای تحت شبکه. محدودیت cron به دقیقه بود اما anacron محدود به روز است. یعنی اگر cron ساعتی یک بار کاری را انجام ندهد anacron یک روز بعد شروع به انجام آن می کند. این سرویس مناسب سرور نیست چون محدودیت زمانی آن به روز می باشد.

برای استفاده از anacron می بایست بسته ی مربوط نصب و سپس سرویس anacron را اجرا کنیم.

```
# yum –y install cronie-anacron
# rpm  -qa  |  grep   cronie-anacron
# rpm  -qi cronie-anacron
```

تنظیم کردن وظایف Anacron

(tasks)وظایف anacron در فایل etc/anacrontab/ لیست شده است. علاوه بر این ، در دایرکتوری var/spool/anacron/ هم برای خودش مانند cron یک دایرکتوری مستقل دارد.

محتویات فایل etc/anacron/ تقریبا شبیه فایل کانفیگ cron است اما تفاوتهایی هم دارد. خطوط اصلی و متفاوت این فایل در انتهای آن قرار دارد که حاوی 4 فیلد اصلی می باشد. هر خط در این فایل تنظیمات مربوط به یک وظیفه است و بدین ترتیب نوشته شده اند :

Period	Delay	Job-identifier	Command
1	**65**	**cron.daly**	**ran-parts /etc/……..**

: period

عدد روزهایی که باید بین اجرای دستورات طی بشود مثلا 9 ، یعنی دستور هر 9 روز یکبار اجرا میشود یا 7 برای اجرای هفتگی است. این فیلد بر مبنای روز می باشد.

: delay

برای هر job , anacron بررسی میکند که آیا این دستور در n(n=period) روز گذشته اجرا شده است یا نه. اگر نشده باشد anacron آن را اجرا میکند. در اصل این فیلد زمانی است که سیستم بعد از UP شدن شروع به انجام کار عقب افتاده می کند. مبنای زمانی این فیلد دقیقه می باشد.

: Job-identifer

آخرین زمانی که anacron یک کاری را انجام داده است در این فایل ثبت می شود.

: command

دستوراتی که خواهان اجرای آن هستیم.

خطوط زیر در فایل اصلی کانفیگ anacron آمده که به ترتیب توضیح داده می شود.

```
1   65   cron.daily
7   70   cron.weekly
30   75   cron.monthly
```

cron.daily 65 1: این خط می گوید یک روز به یک روز که سیستم up می شود 65 دقیقه صبر کند سپس این فایل را بررسی کند، اگر روز قبل این کار انجام شده باشد که هیچ، در غیر این صورت باید job مورد نظر انجام شود.

cron.weekly 70 7: این خط بیان می کند هر 7 روز به 7 روز که سیستم up شد 70 دقیقه صبر کندو job مورد نظر را در صورت انجام نشدن انجام دهد.

cron.monthly 75 35: این خط هم می گوید هر 30 روز یکبار که سیستم بالاآمد 75 دقیقه صبرکند بعد job مورد نظر را انجام می دهد حال فرض کنید 15 روز از اول ماه گذشته و سیستم را up می شود anacron توسط این خط بررسی می کند چون 15 روز به سر ماه بعدی مانده ، job مربوطه را اجرا نمی کند.

متغیرهای محیطی همچون SHELL و PATH را در بالای فایل /etc/anacrontab میتوانید تنظیم کنید چنانکه در cron هم تنظیم میکردیم.

برای period میتوانید هم از اعداد برای نشان دادن دوره ی اجرای دستور استفاده کنید هم از نشانه هایی مانند زیر :

@daily
@weekly
@monthly

در این مثال هر روز با تاخیر یک دقیقه ای جمله ی hi,how are u today به انتهای فایل /skywan13/hi افزوده میشود.

@daily 1 skywan13 echo "hi,how are u today?" >> /skywan13/hi

نکته : این تایمی که بر اساس ساعت آمده به این علت است که سیستم بعد از up شدن به یک پایداری برسد.

نکته مهم : این سرویس به صورت پیش فرض stop است و اگر فعال شود هر روز علاوه بر کارهای خودش وظایف cron را هم انجام می دهد، آنوقت است که بین cron و anacron تضاد کاری پیش می آید. اگر anacron زودتر از cron اجرا شود، cron متوجه نمی شود و هر دو job مورد نظر را انجام می دهند.

وقتی anacron یک job را برای دفعه اول اجرا میکند فایلی همنام با job-identifer را در /var/spool/anacron/ میسازد که محتوای فایل تاریخ اجرای job است (نه ساعت).اصطلاحا به این فایل timestamp گفته میشود. بعد از اجرای مجدد این job دوباره با تاریخ بازنویسی می شود.
کاربر عادی در حالت پیش فرض قادر به استفاده از anacron نیست به یک دلیل ساده ، چونکه اجازه ساخت فایل timestamp را در دایرکتوری var/spool/anacron/ ندارد.
برای حل این مشکل بدون اینکه مشکل جدیدتری بوجود آید بدین ترتیب عمل میکنیم :

1- یک گروه میسازیم و کاربرها را به آن اضافه میکنیم:
برای انجام اینکار از groupadd یا addgroup میتوانید استفاده کنید :

 # groupadd anacron
or
 # addgroup anacron

حالا شما یک گروه بدون کاربر دارین که باید کاربران مورد نظرتون را به این گروه اضافه نمایید:

#adduser skywan13 anacron

2- مجوز مالکیت /var/spool/anacron را تغییر میدهیم برای تغییر مالکیت حتما باید با کاربر root وارد شده باشید:

 # chown root.anacron /var/spool/anacron
 # chmod g+w /var/spool/anacron

خوب حالا شما عضو گروه anacron هستید و مجوز نوشتن را در دایرکتوری مربوطه دارید.

3- برای ادامه کار باید فایل anacron مربوط به خودتان را ایجادکنید:
فایل anacrontab را به یک جایی در دایرکتوری خانه خودتان مثلا /home/skywan13/anacron کپی کنید و طبق آموزش های بالا فایل را تنظیم نمائید.

4- anacron یک daemon نیست و فقط هنگام بالا آمدن سیستم برای کاربر root اجرا میشود پس باید برای خودتان هم اجرا کنید. بدین ترتیب عمل میکنیم:

```
# echo anacron -t $HOME/etc/anacrontab >> .bashrc

# echo anacron -t $HOME/etc/anacrontab >> .bash_profile
```

زمان‌بندی دستورات با at

خیلی وقت ها شده که بخواهیم یک دستور را برای اجرا در زمانی خاص زمان‌بندی کنیم. مثلا ممکن است در ساعاتی از شب دریافت فایل از اینترنت رایگان باشد ولی در آن ساعات خواب باشیم و بیدار ماندن سخت! برای حل این مشکل در ویندوز IDM داشتیم، در لینوکس چه کنیم ؟

در این جا نرم‌افزاری رو به شما معرفی می کنم به نام at که برای برنامه‌ریزی کردن دستورات است و کار با آن نیز بسیار ساده است. at تقریبا در همه توزیع‌های لینوکس نصب است. at فایل یا اسکریپت را اجرا نمی کند بلکه فقط توانایی اجرای یک دستور، در یک زمان خاص را دارد و البته دوره زمانی هم ندارد. خروجی دستور at به یوزر استفاده کننده میل می شود و با ریست سیستم هم خط نوشته شده at از بین میرود.

نحوه نوشتن خط at :

```
  at    time    date
# at    now     +1min
at> ls  /etc
```

یک مثال ساده

دستورات زیر را در ترمینال وارد کنید:

```
# at  2:00
at> echo  \"Hello World!\"  >>  /home/$USER/log
```

و سپس Ctrl + D بزنید.

مثال بالا از at می‌خواهد که دستور خط دوم را که خود عبارت Hello World! را در لاگ کاربر کپی کند و در ساعت ۲ صبح اجرا کند. زدن Ctrl + D بعد از وارد کردن خط دوم، به at پایان وارد کردن لیست کارها را اعلام می کند.

برای این که at بتواند دستورات را اجرا کند باید Daemon آن در حال اجرا باشد:

```
# service atd start
```

همان‌جور که مشخص است با at می‌توان تمامی کارها را زمان‌بندی کرد. فرض کنید لیستی از فایل‌ها برای دانلود دارید و می‌خواهید دانلود ساعت ۲ صبح شروع شده و در ساعت ۸ صبح پایان یابد. ابتدا لینک‌های مورد نظر را در فایلی متنی به طوری که هر لینک در یک خط باشد کپی کنید.

در این مثلا ما فایل را dl-list.txt نامیدیم و از نرم‌افزار دانلود Aria2 برای دانلود کمک گرفتیم:

```
# at 2:00 + 1000 days
at> aria2c -i ~/dl-list.txt -j 1 -x 5
```

و سپس Ctrl + D مثال بالا فرمان دانلود را با کمکat ، به مدت ۱۰۰۰ روز پیاپی در ساعت ۲ صبح اجرا می کند.

حالا برای بسته شدن دانلود در ۸ صبح:

```
# at 8:00 + 1000 days
at> pkill aria2c
```

بعضی از وب‌سایت‌ها ممکن است فایل را تنها در اختیار کاربرانی که در آن وب‌سایت حساب دارند بگذارند مثل Rapidshare که در آن صورت کافیست نام کاربری و رمزعبور خود را در قالب اطلاعات درخواست دانلود با aria2 بفرستید:

```
at> aria2c -i ~/dl-list.txt -j 1 -x 5 --http-user=ali --http-passwd=123456
```

دستور بالا فایل‌های لیست شده در dl-list.txt را با نام کاربری ali و رمزعبور 123456 دانلود می کند.

جزئیات at

atq لیست دستورات برنامه‌ریزی شده را به همراه شماره آن‌ها نشان میدهد.

atrm دستورات برنامه‌ریزی شده را پاک می کند:

 # atrm que_id

برای یافتن que_id دستور مورد نظر، از atq کمک بگیرید.

قالب زمانی به صورت HH:MM وارد می‌شود، استفاده از am و pm هم معتبر است. مثلا ۸ صبح در مثال بالا را am 8 هم می‌توان نوشت.

تاریخ به صورت [CC]YY-MM-DD باید وارد شود، از مخفف ماه و روزها نیز می‌توان استفاده کرد.

عبارت‌هایی مثل فردا، امروز، عصر و نیمه شب هم معتبر هستند.

sun mon tue wed thu fri sat
jan feb mar apr may jun jul aug sep oct nov dec
tomorrow today noon midnight

برای تکرار یک کار در چند روز:

+ N days

چند زمان‌بندی پیچیده‌تر با at :

at 3:00pm tomorrow
at 2:00am jul 5 + 4 days
at 2:00 2012-7-5
at 2:00 wed

نکته:

واحد های زمانی کوچک تر در اول قرار دارند. یعنی مثلا ساعت و دقیقه قبل از ماه.

aria2فقط یک نمونه برنامه برای دانلود است؛ Axel, wget و lftp از دیگر مثال‌ها هستند.

atd همان طور که در بالا گفته شد یکی از فرمان ها at است . برای استفاده از آن میشود از سیلابس هایی جهت نشان دادن دقیق زمان استفاده کرد که به آن ها میپردازیم. همانطور که در قبل اشاره شد از دستور at برای کارهایی که یکبار انجام میشوند استفاده میشود:

at now
at 04:11 am
at now +5 min
at now +5 hours
at now +4 days
at now +4 weeks
at 13:13 pm October 18

برای مثال با این فرمان به این صورت میتوان در ۵ دقیقه آینده سیستم را خاموش کرد:

 # at now +5 min
at> date > /file1

پس از اعلان سیستم از شما اطلاعات مربوط را میگیرد و در زمان تعیین شده کار را انجام میدهد . برای مشاهده لیست کار هایی که توسط این فرمان صورت خواهد کرفت از فرمان at -l ویا atq استفاده میکنیم .

 # at -l
2 Sat Aug 24 10:35:00 2013 a Ali
 # atq
2 Sat Aug 24 10:35:00 2013 a Ali

فصل ششم

بررسی LVM در لینوکس

LVM چیست؟

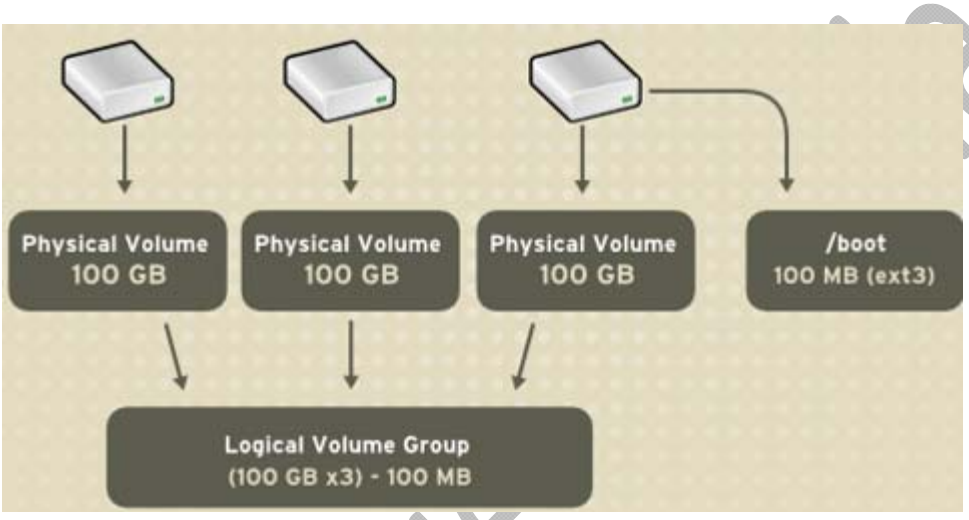
LVM بر گرفته از Logical Volume Management ابزاری است که امکان مدیریت Disk Drive ها و پارتیشن ها را می دهد و به وسیله آن به راحتی می توان پارتیشن را resize کنید. امروزه در تمامی توزیع های لینوکسی LVM وجود داشته و می توانید در زمان نصب توزیع لینوکسی یا پس از نصب توزیع، از LVM برای مدیریت دیسک ها و پارتیشین ها استفاده کنید LVM .امکان resize کردن پارتیشن، ایجاد Snapshot Backupیا ترکیب چندین دیسک برای اجرا شدن تحت یک پارتیشن واحد و بسیاری دیگر از امکانات را فراهم می کند. به طور کلی LVM انعطاف پیذیری بهتری در مدیریت دیسک ها و پارتیشن ها ارائه میکند. بدون وجود LVM، تغییر اندازه یا resize کردن یک پارتیشن کاری دشوار است و ممکن است اطلاعات پارتیشن از بین برود و یا باعث Downtime و از دسترس خارج شدن سیستم شود ولی با استفاده از LVM به راحتی می توان این کار را انجام داد.

در حالت قدیمی مدیریت دیسک ها، سیستم عامل در زیر دایرکتوری dev/ به دنبال دیسک ها تحت نام هایی مانند /dev/sda/ و /dev/sdb/ می گردد.

با LVM دیسک ها و پارتیشن ها می توانند شامل دیسک ها و پارتیشن های متعددی در غالب یک دستگاه واحد باشند .LVM به سیستم عامل، در غالب Volume Group (دیسک ها) وLogical Volume (پارتیشن ها) نشان داده می شود. چونکه Volume Group ها و Logical Volume ها وابسته به هارد درایوها نیستند می توان به راحتی دیسک ها و پارتیشن ها را تغییر اندازه داد یا دیسک و پارتیشن جدید ایجاد کرد. فرض کنید یک Logical Volume به نام /dev/mapper/VolGroup00-LogVol00/ بر روی دایرکتوری ریشه mount شده است و حجم آن به اندازه 50 گیگابایت است و می خواهیم حجم آنرا توسط دیسک دیگری که اندازه 200 گیگابایت دارد افزایش دهیم تا حجم Logical Volume به 250 گیگابیات افزایش یابد. این کار بدون downtime انجام می گیرد. ویژگی دیگر LVM که در سیستم فایلی مانند EXT3 وجود ندارد این است LVM قادر به ایجاد Snapshot Backup از Logical Volume بدون Unmountکردن سیستم فایل است.

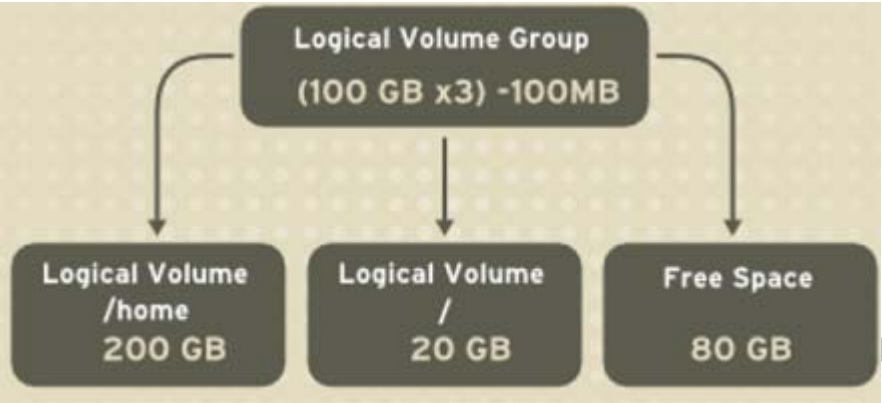
هر Volume Group مجموعه ای از Physical or Logical Volume ها است که در عموم سیستم ها تنها به یک Volume Group نیاز است که شامل تمامی Physical or Logical Volume های موجود

است. Physical Volumes برابر با دیسک هاست که Block Deviceهایی هستند مانند دیسک sda که فضای ذخیره سازی Logical Volume ها را فراهم می کنند Logical Volumeها برابر با پارتیشن ها هستند که سیستم فایل بر روی آنها سوار می شود. (کل یک دیسک می تواند به صورت یک پارتیشن باشد یا اینکه به چندین پارتیشن تقسیم شود)



از شکل بالا مشخص است که دایرکتوری boot نمی تواند LVM باشد زیرا Bootloader نمی تواند آنرا بخواند پس اگر دایرکتوری / یا دایرکتوری root به صورت LVM بود، می بایست boot را در پارتیشنی جدا از دایرکتوری / و غیر LVM قرار داد.

همانطور که گفته شد Volume Group می تواند به چندین Logical Volume تقسیم شود که به نقاط اتصالی مانند دایرکتوری / و یا دایرکتوری /home/ و غیره اختصاص داده شده باشند. وقتی که فضای یک پارتیشن پر می شود، فضای اضافی را می توان از یک Volume Group به پارتیشن اختصاص داد تا فضای پارتیشن افزایش یابد. زمانی که یک دیسک جدید (یک هارد جدید) به سیستم اضافه می شود، می تواند به Volume Groupاضافه شود و پارتیشن هایی که Logical Volume هستند می توانند افزایش یابند.



مزایای LVM

برای سیستم های کوچک : زمانی که شما در سیستم خانگی خودتون با مشکل کم بودن فضا مواجه می شوبد و برای مثال شاخه home شما پر می شود می توانید به راحتی یک هارد دیسک جدید تهیه کنید و فضای جدید را به راحتی به به پارتیشن home خود اضافه کنید.بدون آنکه نیاز به نصب مجدد سیستم عامل داشته باشید.

برای سیستم های بزرگ: برای سیستم های بزرگ مدیریت دیسکها می تواند کار بسیار زمان بری باشد. با کمک LVM مدیر سیستم می تواند ،تنها زمانی که به فضای بیشتری نیاز داشت ،یک دیسک جدید به سیستم اضافه کرده و آن را به فضای قبلی اضافه کند.

ساختمان LVM

```
hda1  hdc1      (PV:s on partitions or whole disks)
                    \   /
                     \ /
                 diskvg          (VG)
                    /  |  \
                   /   |   \
              usrlv rootlv varlv (LV:s)
                |       |      |
           ext2  reiserfs  xfs (filesystems)
```

برای کار با LVM باید با بخشهای مختلف ساختمان LVM آشنا شوید که در اینجا آنها را تعریف خواهیم کرد.

Physical Volume (PV):

PV ها معمولا یک هاردیسک یا چیزی شبیه با آن (مثلا یک Raid Device) می باشد.

Volume Group (VG) :

VG بالاترین سطح ظاهری است که به وسیله LVM استفاده می شود. VG مجموعه ای از LV ها و PV ها را در یک واحد مدیریتی جمع می کند.

Logical Volume (LV) :

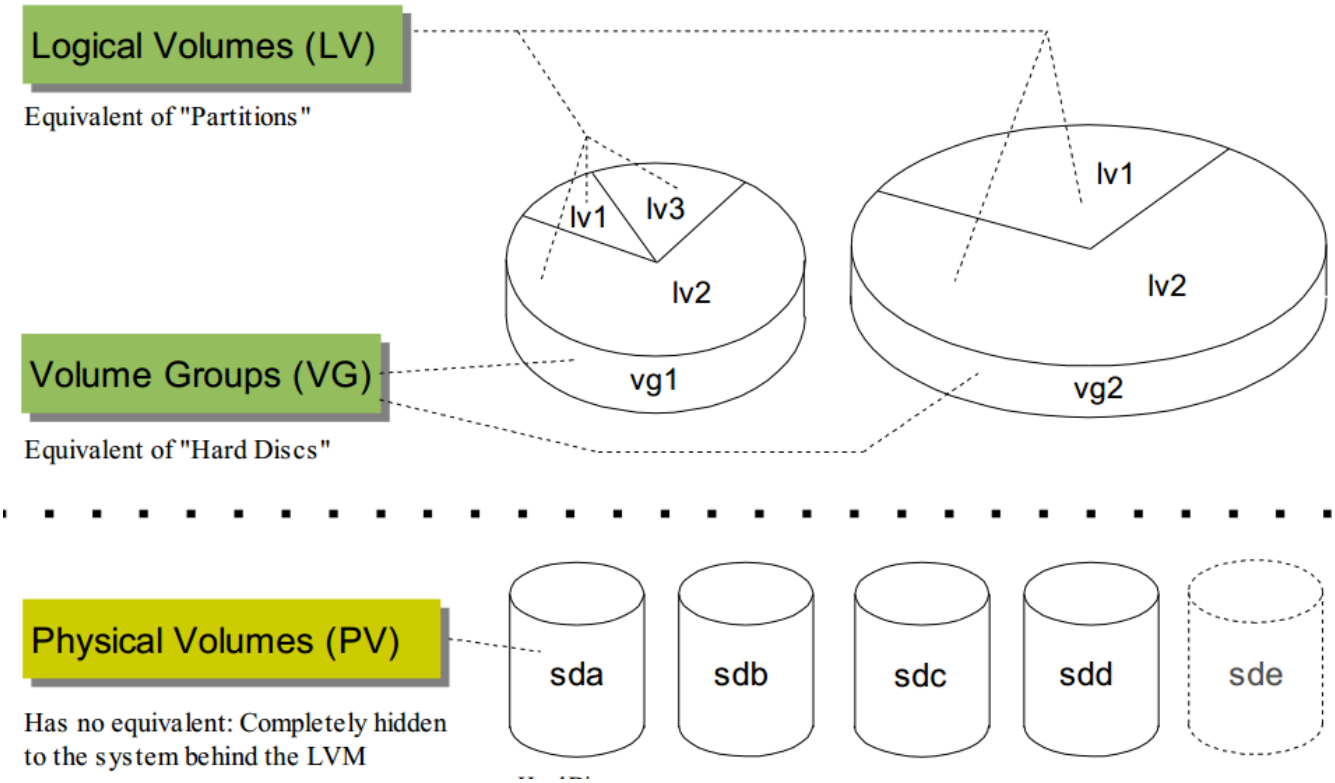
مساوی پارتیشن ها در سیستم های غیر LVM است.

Physical Extent (PE) :

هر PV به تکه های بزرگی از داده تقسیم می شود به نام PE ،این تکه ها (extent) دارای اندازه ای برابر LE در LV ها هستند.

Logical extent (LE) :

هر LV به تکه های بزرگی از داده تقسیم می شود به نام LE ،این اندازه برای تمام LV ها در VG یکسان است.



شروع کار با LVM

اولین کاری که باید انجام دهیم Initialize کردن پارتیشن هاست. قبل از این کارها باید توسط دستورات fdisk هارد یا پارتیشن مورد نظر را پیکربندی کرده و آنها را جزء پارتیشن های LVM ای قرار داده باشیم. ساخت PV توسط دستور pvcreate انجام می‌گیرد. این دستور یک توصیفگر VG در اول دیسک ایجاد می‌کند.

ساخت PV

 # pvcreate /dev/sda5

ساخت VG

خوب حالا می توانیم یک VG بسازیم.

 # vgcreate my_volume_group /dev/sda5

my_volume_group : یک نام دلخواه است که باید به VG داده شود.

اضافه کردن یک PV به VG

در صورتی که بخواهید یک PV دیگر را به VG اضافه کنید، می توانید به شکل زیر عمل نمایید:

 # vgextend my_volume_group /dev/sdb6

ساخت LV

برای ساختن یک LV به ظرفیت G۱۰ به صورت زیر عمل کنید

 # lvcraete -L 10G my_volume_group –name my_logical_volume

اگر بخواهید یک LV بسازید که تمام VG را در بر داشته باشد از vgdisplay استفاده کنید تا مجموع PE های موجود را ببینید سپس دستور lvcreate را اجرا کنید:

 # vgdisplay | grep “Total PE”
Total PE 3576

در اینجا ۳۵۷۶ عدد PE در این VG وجود دارد. برای ساخت LV که تمام این فضا را شامل شود از lvcreate به شکل زیر می توانیم استفاده کنیم:

 # lvcreate -l 3576 my_volume_group –name my_logical_volume

دقت کنید در اینجا از حرف کوچک **l** برای مقدار دهی استفاده کردیم.

ساخت سیستم فایل

اکنون LV آماده است و شما می توانید با آن به صورت یک پارتیشن معمولی رفتار کنید .آن را فرمت کنید:

 # mkfs.ext3 /dev/my_volume_group/my_logical_volume

سپس آن را mount کنید و از آن استفاده نمایید. در صورتی که می خواهید در هنگام راه اندازی سیستم به صورت خودکار mount شود آن را به fstab اضافه کنید.

توسعه یک LV

در صورتی که یکPV به VG اضافه کردید، یا در VG فعلی فضای خالی در اختیار دارید می توانید LV را توسعه دهید.برای توسعه LV به دو صورت می توانید عمل کنید:

#lvextend -L12G /dev/my_volume_group/my_logical_volume

دستور بالا حجم LV را به ۱۲G افزایش می دهد.

#lvextend -L+1G /dev/my_volume_group/my_logical_volume

دستور بالا یک گیگابایت به my_logical_volume اضافه می کند.

بعد از آنکه LV را توسعه دادید ،شما باید سیستم فایل را به اندازه ایی که با آن مطابقت داشته باشد افزایش دهید. با کمک دشتور resize2fs می توانید این کار را انجام دهید .نیاز نیست نگران مشخص کردن اندازه باشید.قبل از اجرای resize2fs ،سیستم فایل را چک کنید:

#e2fsck -f /dev/my_volume_group/my_logical_volume
#resize2fs /dev/my_volume_group/my_logical_volume

پاک کردن LVM

برای پاک کردن LVM بر عکس مسیر ساخت عمل می کنیم. ابتدا باید LV را پاک شود ،قبل از همه باید آن را umount کنید .سپس با کمک دستور زیر آن را remove کنید:

#lvremove /dev/my_volume_group/my_logical_volume

بعد از اون نوبت به پاک کردن VG می‌رسد:

#vgremove my_volume_group

و در آخر پاک کردن PV :

#pvremove /dev/sdb1

جهت درک بهتر مطالب گفته شده ، در ادامه به دو روش ، ساخت LVM آموزش داده می شود. ابتدا به صورت کامندی و بر اساس چند سناریو کوچک و در ادامه به صورت گرافیکی در زمان نصب توزیع CentOS بیان می شود.

راه اندازی LVM به صورت کامندی

در این سناریو ما سیستمی داریم که علاوه بر اینکه یک هارد دومی دارد مقداری از هارد اول آن استفاده شده و مقداری از آن باقی مانده است ما می خواهیم ابتدا از باقی مانده آن یک پارتیشن لینوکسی بسازیم لذا با دستور fdisk –l ابتدا یک آمار از موجودی ظرفیت هارد می گیریم:

```
[root@localhost ~]# fdisk -l

Disk /dev/sda: 10.7 GB, 10737418240 bytes
255 heads, 63 sectors/track, 1305 cylinders
Units = cylinders of 16065 * 512 = 8225280 bytes

   Device Boot      Start         End      Blocks   Id  System
/dev/sda1   *           1          38      305203+  83  Linux
/dev/sda2              39         430     3148740   83  Linux
/dev/sda3             431         688     2072385   83  Linux
/dev/sda4             689        1305     4956052+   5  Extended
/dev/sda5             689         819     1052226   82  Linux swap / Solaris

Disk /dev/sdb: 16.1 GB, 16106127360 bytes
255 heads, 63 sectors/track, 1958 cylinders
Units = cylinders of 16065 * 512 = 8225280 bytes

   Device Boot      Start         End      Blocks   Id  System
[root@localhost ~]# _
```

سپس با دستور fdisk وارد فضای پارتیشن بندی شده و یک پرینت از پارتیشن های هارد می گیریم

```
[root@localhost ~]# fdisk /dev/sda

The number of cylinders for this disk is set to 1305.
There is nothing wrong with that, but this is larger than 1024,
and could in certain setups cause problems with:
1) software that runs at boot time (e.g., old versions of LILO)
2) booting and partitioning software from other OSs
   (e.g., DOS FDISK, OS/2 FDISK)

Command (m for help): p

Disk /dev/sda: 10.7 GB, 10737418240 bytes
255 heads, 63 sectors/track, 1305 cylinders
Units = cylinders of 16065 * 512 = 8225280 bytes

   Device Boot      Start         End      Blocks   Id  System
/dev/sda1   *           1          38      305203+  83  Linux
/dev/sda2              39         430     3148740   83  Linux
/dev/sda3             431         688     2072385   83  Linux
/dev/sda4             689        1305     4956052+   5  Extended
/dev/sda5             689         819     1052226   82  Linux swap / Solaris

Command (m for help): _
```

در اینجا به ما اعلام میکند از سیلندر 820 تا 1305 هارد خالی است لذا ما اولین سیلندر را 821 قرار داده و در خط بعد ظرفیتی که می خواهیم پارتیشن مذکور داشته باشد را تایپ می کنیم.

```
Command (m for help): n
First cylinder (820-1305, default 820): 821
Last cylinder or +size or +sizeM or +sizeK (821-1305, default 1305): +2G

Command (m for help): _
```

همانطور که در شکل زیر می بینید پارتیشن جدید با نام sda6 ساخته شد.

```
Command (m for help): p

Disk /dev/sda: 10.7 GB, 10737418240 bytes
255 heads, 63 sectors/track, 1305 cylinders
Units = cylinders of 16065 * 512 = 8225280 bytes

   Device Boot      Start         End      Blocks   Id  System
/dev/sda1   *           1          38      305203+  83  Linux
/dev/sda2              39         430     3148740   83  Linux
/dev/sda3             431         688     2072385   83  Linux
/dev/sda4             689        1305     4956052+   5  Extended
/dev/sda5             689         819     1052226   82  Linux swap / Solaris
/dev/sda6             821        1064     1959930   83  Linux

Command (m for help): _
```

پارتیشن جدید لینوکسی می باشد که باید آن را به یک پارتیشن LVM ای تبدیل کنیم. برای این کار از دستور زیر استفاده می کنیم

```
Command (m for help): t
Partition number (1-6): 6
Hex code (type L to list codes): _
```

و سپس با تایپ دستور l لیست پارتیشنهایی که می تواند برای ما بسازد را مشاهده کرده و نوع دلخواه را انتخاب می کنیم.

```
 0  Empty           1e  Hidden W95 FAT1 80  Old Minix       bf  Solaris
 1  FAT12           24  NEC DOS         81  Minix / old Lin c1  DRDOS/sec (FAT-
 2  XENIX root      39  Plan 9          82  Linux swap / So c4  DRDOS/sec (FAT-
 3  XENIX usr       3c  PartitionMagic  83  Linux           c6  DRDOS/sec (FAT-
 4  FAT16 <32M      40  Venix 80286     84  OS/2 hidden C:  c7  Syrinx
 5  Extended        41  PPC PReP Boot   85  Linux extended  da  Non-FS data
 6  FAT16           42  SFS             86  NTFS volume set db  CP/M / CTOS / .
 7  HPFS/NTFS       4d  QNX4.x          87  NTFS volume set de  Dell Utility
 8  AIX             4e  QNX4.x 2nd part 88  Linux plaintext df  BootIt
 9  AIX bootable    4f  QNX4.x 3rd part 8e  Linux LVM       e1  DOS access
 a  OS/2 Boot Manag 50  OnTrack DM      93  Amoeba          e3  DOS R/O
 b  W95 FAT32       51  OnTrack DM6 Aux 94  Amoeba BBT      e4  SpeedStor
 c  W95 FAT32 (LBA) 52  CP/M            9f  BSD/OS          eb  BeOS fs
 e  W95 FAT16 (LBA) 53  OnTrack DM6 Aux a0  IBM Thinkpad hi ee  EFI GPT
 f  W95 Ext'd (LBA) 54  OnTrackDM6      a5  FreeBSD         ef  EFI (FAT-12/16/
10  OPUS            55  EZ-Drive        a6  OpenBSD         f0  Linux/PA-RISC b
11  Hidden FAT12    56  Golden Bow      a7  NeXTSTEP        f1  SpeedStor
12  Compaq diagnost 5c  Priam Edisk     a8  Darwin UFS      f4  SpeedStor
14  Hidden FAT16 <3 61  SpeedStor       a9  NetBSD          f2  DOS secondary
16  Hidden FAT16    63  GNU HURD or Sys ab  Darwin boot     fb  VMware VMFS
17  Hidden HPFS/NTF 64  Novell Netware  b7  BSDI fs         fc  VMware VMKCORE
18  AST SmartSleep  65  Novell Netware  b8  BSDI swap       fd  Linux raid auto
1b  Hidden W95 FAT3 70  DiskSecure Mult bb  Boot Wizard hid fe  LANstep
1c  Hidden W95 FAT3 75  PC/IX           be  Solaris boot    ff  BBT
Hex code (type L to list codes): _
```

با وارد کردن گزینه 8e نوع پارتیشن مزکور به LVM تغییر پیدا می کند.

```
Hex code (type L to list codes): 8e
Changed system type of partition 6 to 8e (Linux LVM)

Command (m for help): p

Disk /dev/sda: 10.7 GB, 10737418240 bytes
255 heads, 63 sectors/track, 1305 cylinders
Units = cylinders of 16065 * 512 = 8225280 bytes

   Device Boot      Start         End      Blocks   Id  System
/dev/sda1   *           1          38      305203+  83  Linux
/dev/sda2              39         430     3148740   83  Linux
/dev/sda3             431         688     2072385   83  Linux
/dev/sda4             689        1305     4956052+   5  Extended
/dev/sda5             689         819     1052226   82  Linux swap / Solaris
/dev/sda6             821        1064     1959930   8e  Linux LVM

Command (m for help): _
```

سپس جهت اعمال تغییرات از گزینه **W** استفاده می کنیم اما همانطور که میبینید پیام سیستم این مفهوم را می رساند که باید سیستم ریست شود تا جدول پارتیشن سیستم بروز گردد. این کار در سرور امکان پذیر نیست و ممکن است باعث صدماتی شود .

```
Command (m for help): w
The partition table has been altered!

Calling ioctl() to re-read partition table.

WARNING: Re-reading the partition table failed with error 16: Device or resource
 busy.
The kernel still uses the old table.
The new table will be used at the next reboot.
Syncing disks.
[root@localhost ~]# _
```

لذا برای حل این گونه مشکلات شرکت ردهت دستور partpeobe را ارائه کرد. اگر بخواهیم کل جدول پارتیشن سیستم به روز شود از دستور اول استفاده و اگر بخواهیم فقط تغییرات صورت گرفته در جدول پارتیشن سیستم درج شود از دستور دوم استفاده می کنیم.

```
[root@localhost ~]# man partprobe
[root@localhost ~]# partprobe /dev/sda
[root@localhost ~]# partprobe /dev/sda6
[root@localhost ~]#
```

همانطور که می بینید یک sdb هم وجود دارد که جهت اجرای سناریو آن را هم پارتیشن بندی می کنیم.

```
[root@localhost ~]# fdisk -l

Disk /dev/sda: 10.7 GB, 10737418240 bytes
255 heads, 63 sectors/track, 1305 cylinders
Units = cylinders of 16065 * 512 = 8225280 bytes

   Device Boot      Start         End      Blocks   Id  System
/dev/sda1   *           1          38      305203+  83  Linux
/dev/sda2              39         430     3148740   83  Linux
/dev/sda3             431         688     2072385   83  Linux
/dev/sda4             689        1305     4956052+   5  Extended
/dev/sda5             689         819     1052226   82  Linux swap / Solaris
/dev/sda6             821        1064     1959930   8e  Linux LVM

Disk /dev/sdb: 16.1 GB, 16106127360 bytes
255 heads, 63 sectors/track, 1958 cylinders
Units = cylinders of 16065 * 512 = 8225280 bytes

   Device Boot      Start         End      Blocks   Id  System
[root@localhost ~]# fdisk /dev/sdb

The number of cylinders for this disk is set to 1958.
There is nothing wrong with that, but this is larger than 1024,
and could in certain setups cause problems with:
1) software that runs at boot time (e.g., old versions of LILO)
2) booting and partitioning software from other OSs
   (e.g., DOS FDISK, OS/2 FDISK)

Command (m for help): n
Command action
   e   extended
   p   primary partition (1-4)
p
Partition number (1-4): 1
First cylinder (1-1958, default 1):
Using default value 1
Last cylinder or +size or +sizeM or +sizeK (1-1958, default 1958): +2G

Command (m for help): p

Disk /dev/sdb: 16.1 GB, 16106127360 bytes
255 heads, 63 sectors/track, 1958 cylinders
Units = cylinders of 16065 * 512 = 8225280 bytes

   Device Boot      Start         End      Blocks   Id  System
/dev/sdb1               1         244     1959898+  83  Linux

Command (m for help):
```

همانطور که میبینید پارتیشن ساخته شده لینوکسی است و باید به LVM تغییر type پیدا کند:

```
Command (m for help): t
Selected partition 1
Hex code (type L to list codes):
```

در اینجا کد تغییر type را وارد کرده تغییرات را write می کنیم :

```
Hex code (type L to list codes): 8e
Changed system type of partition 1 to 8e (Linux LVM)

Command (m for help): p

Disk /dev/sdb: 16.1 GB, 16106127360 bytes
255 heads, 63 sectors/track, 1958 cylinders
Units = cylinders of 16065 * 512 = 8225280 bytes

   Device Boot      Start         End      Blocks   Id  System
/dev/sdb1               1         244     1959898+  8e  Linux LVM

Command (m for help): w
The partition table has been altered!

Calling ioctl() to re-read partition table.
Syncing disks.
[root@localhost ~]#
```

در انتها جدول پارتیشن را بروز کرده و دوباره از پارتیشن های موجود جهت اطمینان از تغییر لیست می گیریم

```
[root@localhost ~]# partprobe /dev/sdb1
[root@localhost ~]# fdisk -l

Disk /dev/sda: 10.7 GB, 10737418240 bytes
255 heads, 63 sectors/track, 1305 cylinders
Units = cylinders of 16065 * 512 = 8225280 bytes

   Device Boot      Start         End      Blocks   Id  System
/dev/sda1   *           1          38      305203+  83  Linux
/dev/sda2              39         430     3148740   83  Linux
/dev/sda3             431         688     2072385   83  Linux
/dev/sda4             689        1305     4956052+   5  Extended
/dev/sda5             689         819     1052226   82  Linux swap / Solaris
/dev/sda6             821        1064     1959930   8e  Linux LVM

Disk /dev/sdb: 16.1 GB, 16106127360 bytes
255 heads, 63 sectors/track, 1958 cylinders
Units = cylinders of 16065 * 512 = 8225280 bytes

   Device Boot      Start         End      Blocks   Id  System
/dev/sdb1               1         244     1959898+  8e  Linux LVM
```

تا الان ما فقط دو پارتیشن با نوع lvm ساخته ایم. برای ادامه کار باید با آنها یک PV تشکیل دهیم. لازم به ذکر است این پارتیشن ها بر روی هارد های جداگانه قرار دارد.

```
[root@localhost ~]# pvdisplay
  /dev/hdc: open failed: No medium found
[root@localhost ~]# pvcreate /dev/sda6 /dev/sdb1
  Physical volume "/dev/sda6" successfully created
  Physical volume "/dev/sdb1" successfully created
[root@localhost ~]#
```

بعد از ساخت PV، جهت اطمینان از صحت ساخت با دستور pvdisplay یک آمار از سیستم می گیریم :

```
[root@localhost ~]# pvdisplay
  /dev/hdc: open failed: No medium found
  "/dev/sda6" is a new physical volume of "1.87 GB"
  --- NEW Physical volume ---
  PV Name               /dev/sda6
  VG Name
  PV Size               1.87 GB
  Allocatable           NO
  PE Size (KByte)       0
  Total PE              0
  Free PE               0
  Allocated PE          0
  PV UUID               kBFkGt-seno-jfVU-5hF6-yXXb-a3IN-2aTT3c

  "/dev/sdb1" is a new physical volume of "1.87 GB"
  --- NEW Physical volume ---
  PV Name               /dev/sdb1
  VG Name
  PV Size               1.87 GB
  Allocatable           NO
  PE Size (KByte)       0
  Total PE              0
  Free PE               0
  Allocated PE          0
  PV UUID               1ES0eL-h3LS-YWZo-xlYU-PHct-W2DU-c7ogPC
```

حال باید PV های درست شده را در یک گروه قرار دهیم که به این گروه VG گفته می شود. یک vg مجموع pvها می باشد.

```
[root@localhost ~]# vgdisplay
  /dev/hdc: open failed: No medium found
[root@localhost ~]# vgcreate testlvm /dev/sda6 /dev/sdb1
  /dev/hdc: open failed: No medium found
  Volume group "testlvm" successfully created
[root@localhost ~]# echo $?
0
[root@localhost ~]#
```

سپس از محتویات VG یک لیست میگیریم:

```
[root@localhost ~]# vgdisplay
  /dev/hdc: open failed: No medium found
  --- Volume group ---
  VG Name               testlvm
  System ID
  Format                lvm2
  Metadata Areas        2
  Metadata Sequence No  1
  VG Access             read/write
  VG Status             resizable
  MAX LV                0
  Cur LV                0
  Open LV               0
  Max PV                0
  Cur PV                2
  Act PV                2
  VG Size               3.73 GB
  PE Size               4.00 MB
  Total PE              956
  Alloc PE / Size       0 / 0
  Free  PE / Size       956 / 3.73 GB
  VG UUID               ysD7D2-lgaK-T51R-bPH6-geC3-D6ax-nCzLoY
```

بعد از این مرحله باید از VG یک LV بسازیم که یا به یک دارکتوری ما اضافه شود و یا خودش یک دایرکتوری مجزا باشد.

L-: با این آپشن حجم Lv را مشخص می کنیم.

testlvm : نام VG را وارد می کنیم.

-n datalvm : با این آپشن نام LV را مشخص می کنیم.

```
[root@localhost ~]# lvcreate -L 3G testlvm -n datalvm
  /dev/hdc: open failed: No medium found
  Logical volume "datalvm" created
[root@localhost ~]# lvdisplay
  /dev/hdc: open failed: No medium found
  --- Logical volume ---
  LV Name                /dev/testlvm/datalvm
  VG Name                testlvm
  LV UUID                4FpK8m-43TG-x48k-wi2d-d2M5-yWlm-hXRRLZ
  LV Write Access        read/write
  LV Status              available
  # open                 0
  LV Size                3.00 GB
  Current LE             768
  Segments               2
  Allocation             inherit
  Read ahead sectors     auto
  - currently set to     256
  Block device           253:0
```

و با این دستور هم لیست LV ها را مشاهده می کنیم:

```
[root@localhost ~]# lvdisplay
  /dev/hdc: open failed: No medium found
  --- Logical volume ---
  LV Name                /dev/testlvm/datalvm
  VG Name                testlvm
  LV UUID                4FpK8m-43TG-x48k-wi2d-d2M5-yWlm-hXRRLZ
  LV Write Access        read/write
  LV Status              available
  # open                 0
  LV Size                3.00 GB
  Current LE             768
  Segments               2
  Allocation             inherit
  Read ahead sectors     auto
  - currently set to     256
  Block device           253:0
```

Lg ای که ساخته ایم هنوز خام است و باید برای آن فایل سیستم انتخاب کرده و آن را فورمت کنیم. برای این کار از دستور زیر بهره می بریم.

```
[root@localhost ~]# mkfs.ext3 /dev/testlvm/datalvm
mke2fs 1.39 (29-May-2006)
Filesystem label=
OS type: Linux
Block size=4096 (log=2)
Fragment size=4096 (log=2)
393216 inodes, 786432 blocks
39321 blocks (5.00%) reserved for the super user
First data block=0
Maximum filesystem blocks=805306368
24 block groups
32768 blocks per group, 32768 fragments per group
16384 inodes per group
Superblock backups stored on blocks:
        32768, 98304, 163840, 229376, 294912

Writing inode tables: done
Creating journal (16384 blocks): done
Writing superblocks and filesystem accounting information: done
```

در انتها هم باید این فضای ساخته شده را mount کنیم:

```
[root@localhost ~]# mount /dev/testlvm/datalvm /DATA/
[root@localhost ~]# df -ah
Filesystem            Size  Used Avail Use% Mounted on
/dev/sda3             2.0G  874M  986M  47% /
proc                     0     0     0   -  /proc
sysfs                    0     0     0   -  /sys
devpts                   0     0     0   -  /dev/pts
/dev/sda2             3.0G   69M  2.7G   3% /home
/dev/sda1             289M   16M  258M   6% /boot
tmpfs                 252M     0  252M   0% /dev/shm
none                     0     0     0   -  /proc/sys/fs/binfmt_misc
sunrpc                   0     0     0   -  /var/lib/nfs/rpc_pipefs
/dev/mapper/testlvm-datalvm
                      3.0G   69M  2.8G   3% /DATA
```

حال فرض کنید که هارد سیستم پر شده و می خواهیم یک PV را به VG موجود اضافه کنیم تا مشکل کمبود ظرفیت بر طرف شود. لذا از هاردی که قبلا به سیستم اضافه شده یک مقداری را طبق روش زیر جدا می کنیم:

```
[root@localhost DATA]# fdisk /dev/sdb

The number of cylinders for this disk is set to 1958.
There is nothing wrong with that, but this is larger than 1024,
and could in certain setups cause problems with:
1) software that runs at boot time (e.g., old versions of LILO)
2) booting and partitioning software from other OSs
   (e.g., DOS FDISK, OS/2 FDISK)

Command (m for help): n
Command action
   e   extended
   p   primary partition (1-4)
p
Partition number (1-4): 2
First cylinder (245-1958, default 245):
Using default value 245
Last cylinder or +size or +sizeM or +sizeK (245-1958, default 1958): +5
G
Command (m for help): t
Partition number (1-4): 8e
Value out of range.
Partition number (1-4): 2
Hex code (type L to list codes): 8e
Changed system type of partition 2 to 8e (Linux LVM)

Command (m for help): p

Disk /dev/sdb: 16.1 GB, 16106127360 bytes
255 heads, 63 sectors/track, 1958 cylinders
Units = cylinders of 16065 * 512 = 8225280 bytes

   Device Boot      Start         End      Blocks   Id  System
/dev/sdb1               1         244     1959898+  8e  Linux LVM
/dev/sdb2             245         853     4891792+  8e  Linux LVM

Command (m for help): w
The partition table has been altered!

Calling ioctl() to re-read partition table.
```

سپس برای اعمال تغییرات سیستم عامل را مجبور می کنیم تا جدول پارتیشن های سیستم را بخواند:

```
[root@localhost DATA]# partprobe /dev/sdb
[root@localhost DATA]# partprobe /dev/sdb2
```

حالا باید با دستور pvcreate از پارتیشن ساخته شده یک PV بسازیم

```
[root@localhost DATA]# pvcreate /dev/sdb2
  Physical volume "/dev/sdb2" successfully created
```

در این قسمت باید PV ساخته شده را به VG موجود اضافه کنیم

```
[root@localhost DATA]# vgextend testlvm /dev/sdb2
  /dev/hdc: open failed: No medium found
  Volume group "testlvm" successfully extended
[root@localhost DATA]#
[root@localhost DATA]# vgdisplay
  /dev/hdc: open failed: No medium found
  --- Volume group ---
  VG Name               testlvm
  System ID
  Format                lvm2
  Metadata Areas        3
  Metadata Sequence No  3
  VG Access             read/write
  VG Status             resizable
  MAX LV                0
  Cur LV                1
  Open LV               1
  Max PV                0
  Cur PV                3
  Act PV                3
  VG Size               8.40 GB
  PE Size               4.00 MB
  Total PE              2150
  Alloc PE / Size       768 / 3.00 GB
  Free  PE / Size       1382 / 5.40 GB
```

برای جدا کردن یک PV از VG طبق روش زیر عمل می کنیم:

```
[root@localhost DATA]# vgreduce testlvm /dev/sdb2
  /dev/hdc: open failed: No medium found
  Removed "/dev/sdb2" from volume group "testlvm"
[root@localhost DATA]#
```

```
[root@localhost DATA]# vgdisplay
  /dev/hdc: open failed: No medium found
  --- Volume group ---
  VG Name               testlvm
  System ID
  Format                lvm2
  Metadata Areas        2
  Metadata Sequence No  4
  VG Access             read/write
  VG Status             resizable
  MAX LV                0
  Cur LV                1
  Open LV               1
  Max PV                0
  Cur PV                2
  Act PV                2
  VG Size               3.73 GB
  PE Size               4.00 MB
  Total PE              956
  Alloc PE / Size       768 / 3.00 GB
  Free  PE / Size       188 / 752.00 MB
  VG UUID               ysD7D2-lgaK-T51R-
```

حال فرض کنید حجم یک پارتیشن کم است و ما می خواهیم یک PV به VG اضافه کرده تا بتوانیم حجم LV را افزایش دهیم

```
[root@localhost DATA]# vgextend testlvm /dev/sdb2
  /dev/hdc: open failed: No medium found
  Volume group "testlvm" successfully extended
[root@localhost DATA]# vgdisplay
  /dev/hdc: open failed: No medium found
  --- Volume group ---
  VG Name               testlvm
  System ID
  Format                lvm2
  Metadata Areas        3
  Metadata Sequence No  5
  VG Access             read/write
  VG Status             resizable
  MAX LV                0
  Cur LV                1
  Open LV               1
  Max PV                0
  Cur PV                3
  Act PV                3
  VG Size               8.40 GB
  PE Size               4.00 MB
  Total PE              2150
  Alloc PE / Size       768 / 3.00 GB
  Free  PE / Size       1382 / 5.40 GB
```

ما می خواهیم 5گیگ به حجم LV اضافه شود

```
[root@localhost DATA]# lvdisplay
  /dev/hdc: open failed: No medium found
  --- Logical volume ---
  LV Name                /dev/testlvm/datalvm
  VG Name                testlvm
  LV UUID                4FpK8m-43TG-x48k-wi2d-d2M5-yWlm-hXRRLZ
  LV Write Access        read/write
  LV Status              available
  # open                 1
  LV Size                3.00 GB
  Current LE             768
  Segments               2
  Allocation             inherit
  Read ahead sectors     auto
  - currently set to     256
  Block device           253:0
```

دستور **lvresize** هم می تواند حجم را کم کند و هم می تواند آن را افزایش دهد. در اینجا عدد 8 بدین معنا است که ظرفیت lv به 8 گیگ برسد. اگر +5G هم قرار می دادیم همین کار را انجام می داد. آدرسی که در اینجا وارد می کنیم آدرس LV است که قبلا ایجاد کرده ایم. در اینجا بدون مشکل خاصی عمل resize با موفقیت انجام شد ولی ممکن است در بعضی مواقع پیغام Busy با ما نشان داده شود. علت آن هم این است که دایرکتوری مورد نظر در حال ارائه سرویس به کاربران و یا سیستم می باشد که در چنین مواقعی باید ابتدا دایرکتوری مورد نظر را umount کرده و بعد آن را resize کنیم.

```
[root@localhost DATA]# lvresize -L 8G /dev/mapper/testlvm-datalvm
  /dev/hdc: open failed: No medium found
  Extending logical volume datalvm to 8.00 GB
  Logical volume datalvm successfully resized
[root@localhost DATA]#
```

ما عمل resize را انجام دادیم ولی همانطور که می بینید افزایش حجمی به ما نشان نمی دهد

```
[root@localhost DATA]# df -ah
Filesystem            Size  Used Avail Use% Mounted on
/dev/sda3             2.0G  874M  986M  47% /
proc                     0     0     0   -  /proc
sysfs                    0     0     0   -  /sys
devpts                   0     0     0   -  /dev/pts
/dev/sda2             3.0G   69M  2.7G   3% /home
/dev/sda1             289M   16M  258M   6% /boot
tmpfs                 252M     0  252M   0% /dev/shm
none                     0     0     0   -  /proc/sys/fs/binfmt_misc
sunrpc                   0     0     0   -  /var/lib/nfs/rpc_pipefs
/dev/mapper/testlvm-datalvm
                      3.0G   69M  2.8G   3% /DATA
[root@localhost DATA]#
```

چون ما قبلا روی lv مورد نظر یک فایل سیستم ایجاد کرده ایم که خود آن بلاک وآینود درست می کند. حجم با موفقیت تغییر می کند ولی مقدار صحیح را به ما نشان نمی دهد ، ما باید به طریقی به آن بفهمانیم که سایز تغییر کرده است. لذا از دستور زیر برای قابل استفاده شدن فضای مورد نظر استفاده می کنیم :

p- : این گزینه درصد resize را به ما نشان می دهد.

f- : این گزینه هم به صورت force کار درخواستی را انجام می دهد.

```
[root@localhost DATA]# resize2fs -p -f /dev/mapper/testlvm-datalvm
resize2fs 1.39 (29-May-2006)
Filesystem at /dev/mapper/testlvm-datalvm is mounted on /DATA; on-line
resizing required
Performing an on-line resize of /dev/mapper/testlvm-datalvm to 2097152
(4k) blocks.
The filesystem on /dev/mapper/testlvm-datalvm is now 2097152 blocks lon
g.
```

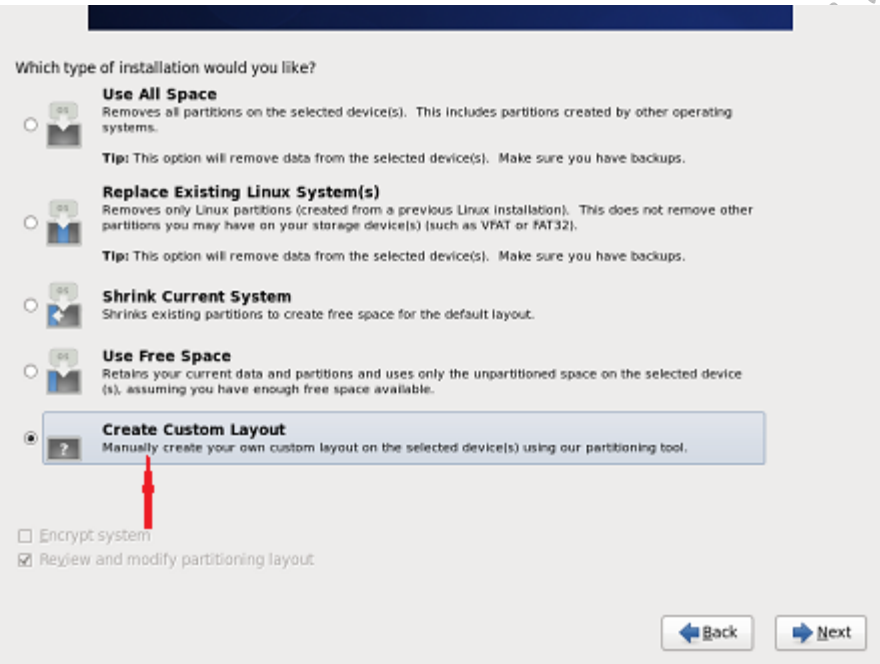
با وارد کردن مجدد دستور df -ah می توانید تغییر حجم را مشاهده کنید:

```
[root@localhost DATA]# df -ah
Filesystem            Size  Used Avail Use% Mounted on
/dev/sda3             2.0G  874M  986M  47% /
proc                     0     0     0   -  /proc
sysfs                    0     0     0   -  /sys
devpts                   0     0     0   -  /dev/pts
/dev/sda2             3.0G   69M  2.7G   3% /home
/dev/sda1             289M   16M  258M   6% /boot
tmpfs                 252M     0  252M   0% /dev/shm
none                     0     0     0   -  /proc/sys/fs/binfmt_misc
sunrpc                   0     0     0   -  /var/lib/nfs/rpc_pipefs
/dev/mapper/testlvm-datalvm
                      7.9G   71M  7.5G   1% /DATA
[root@localhost DATA]#
```

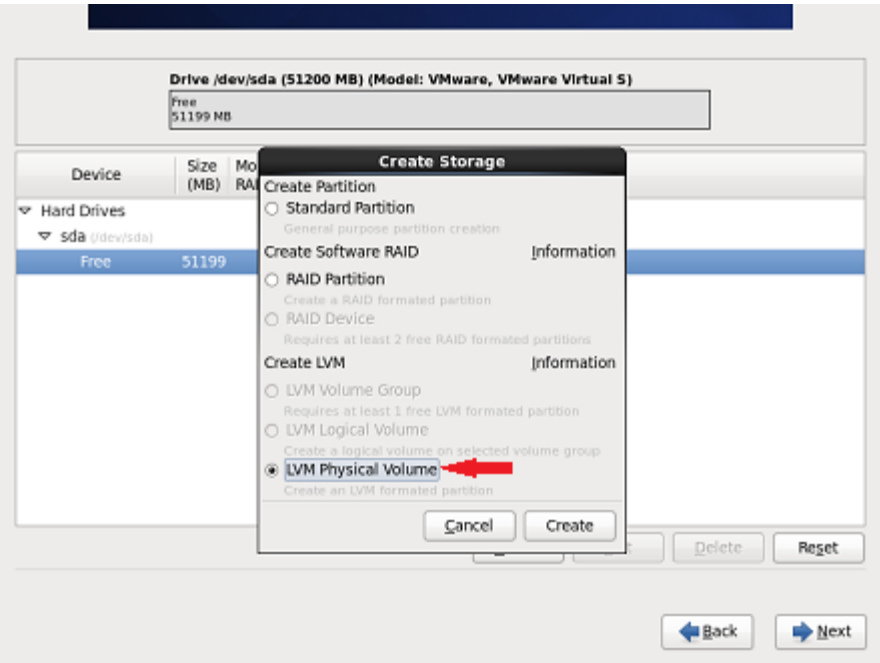
ایجاد LVM Partition در زمان نصب لینوکس

در بخش قبلی راه اندازی و مدیریت LVM را به صورت کامندی توضیح دادیم ، در این بخش راه اندازی LVM در زمان نصب سیستم عامل را بررسی می کنیم .

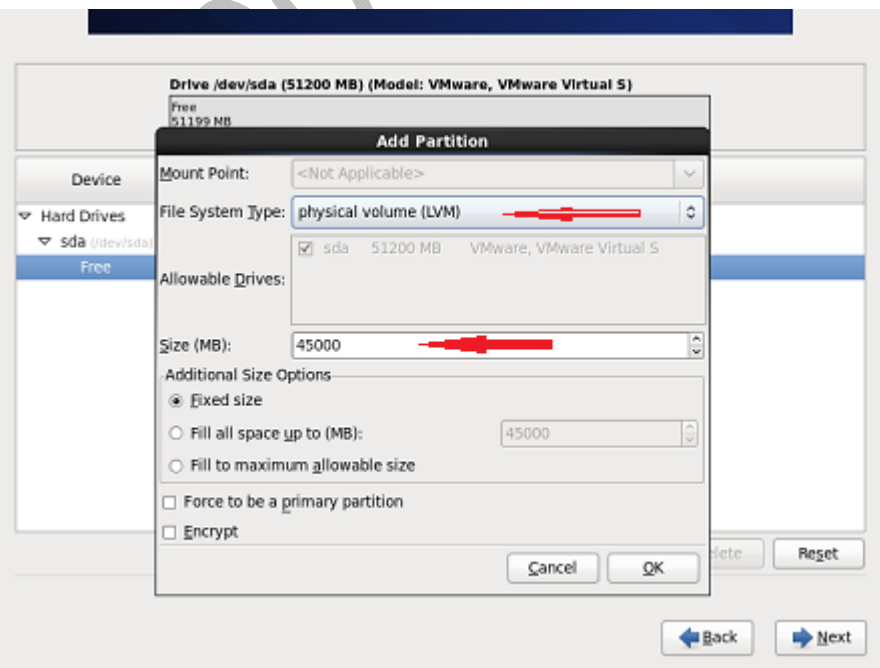
پس از بوت شدن CentOS 6 در پنجره شکل زیر گزینه Create Custom Layout را انتخاب کنید.



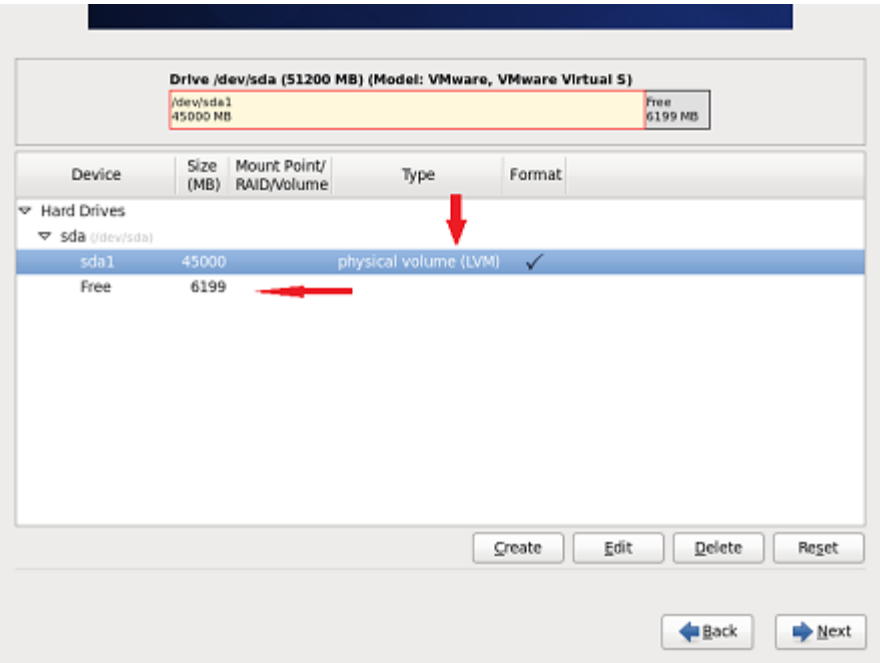
سپس از پنجره شکل زیر Create را انتخاب و از پنجره باز شده LVM Physical Volume را انتخاب و Createرا کلیک کنید .نخستین گام ایجاد یک LVM Physical Volume است. سپس Group Volumeها را ایجاد و در نهایت Logical Volume ها را ایجاد می کنیم که نقش پارتیشن را داشته و به دایرکتوری ها متصل یا mount می شوند.



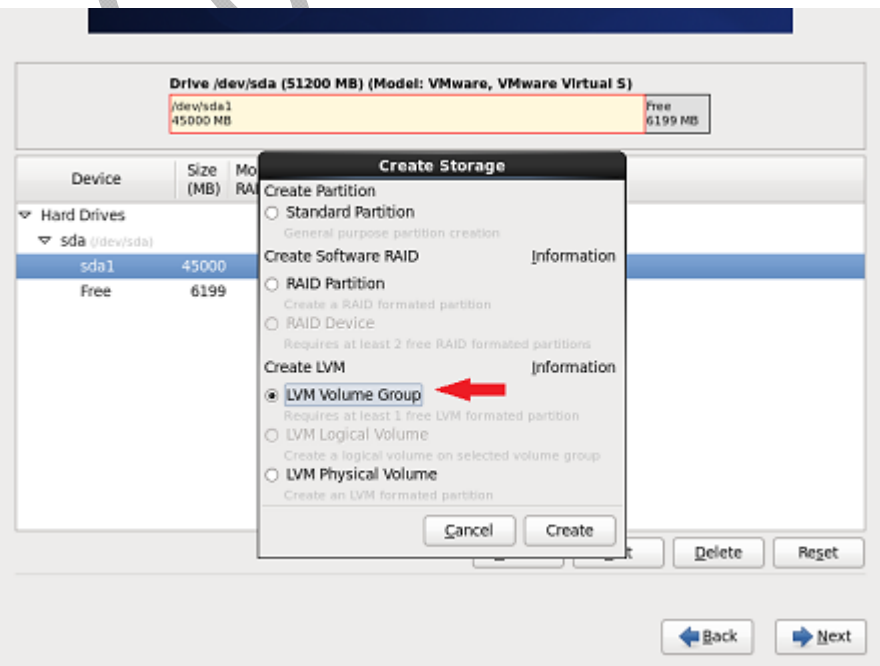
سپس مطابق با شکل زیر می توانید اندازه آنرا تعیین کنید. من از 50 گیابایت دیسکی که توسط VMware در اختیار داشتم 45000 مگابایت یا 45 گیگابایت آنرا برای LVM Physical Volume انتخاب کرده ام. توجه کنید که پارتیشن boot نباید LVM باشد پس فضایی البته فضایی کم برای آن در این سناریو لازم است.



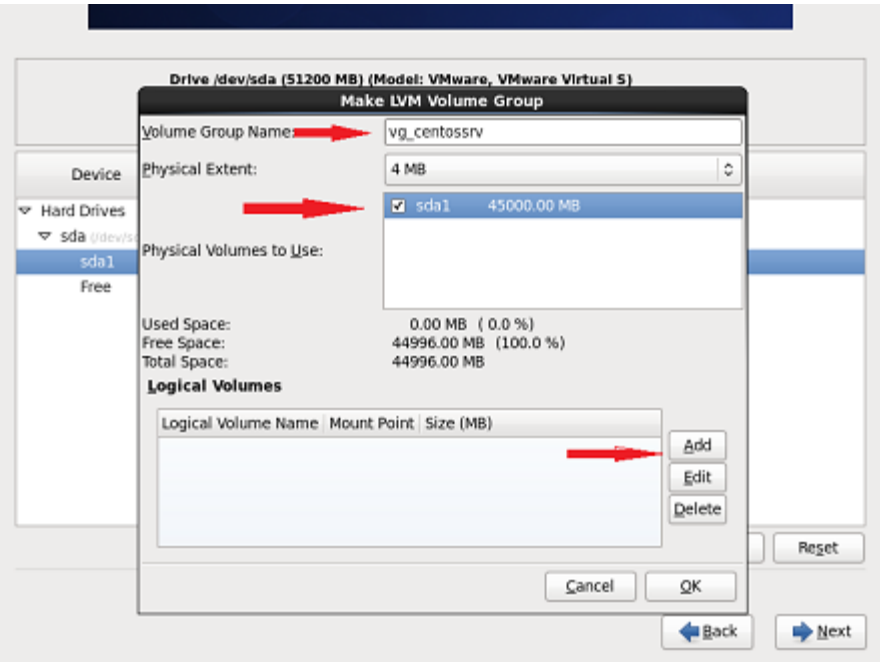
مطابق با شکل زیر خواهید دید که LVM Physical Volume به حجم 45 گیابایت ایجاد شد و 6 گیگابایت نیز بلااستفاده ماند. در گام بعدی باید یک Group Volume بر روی LVM Physical Volumeایجاد کنیم.



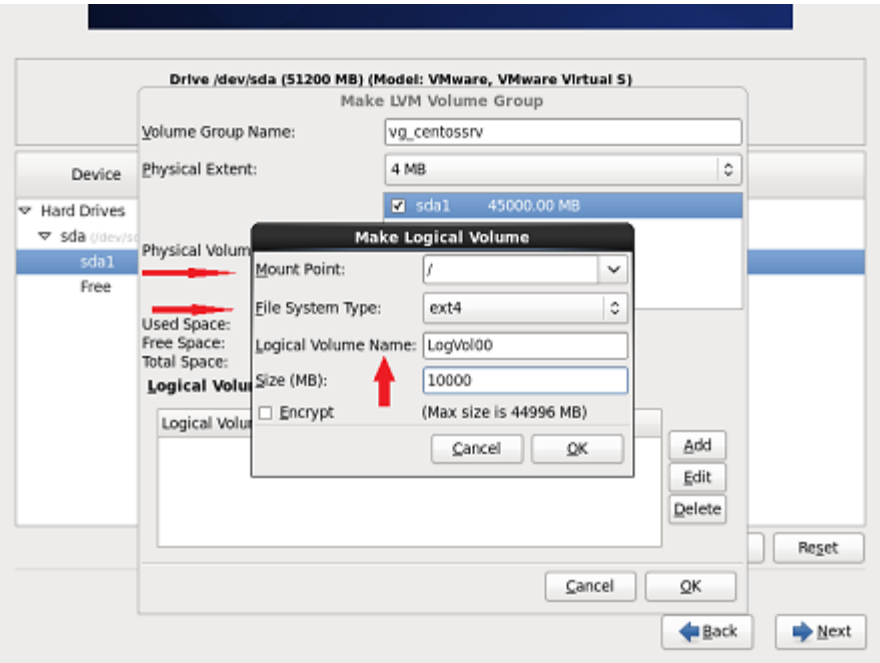
برای ایجاد یک LVM Group Volume بر روی Create پنجره شکل بالا کلیک کنید تا پنجره شکل زیر باز شود و سپس LVM Group Volume را انتخاب کنید.



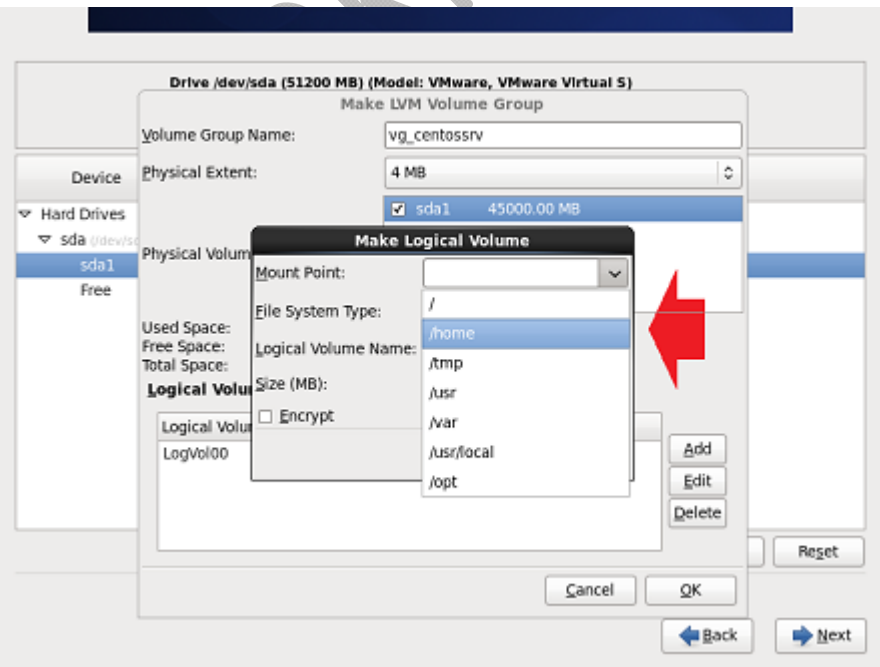
سپس در پنجره باز شده شکل زیر نام LVM Group Volume را انتخاب کنید. اگر چندین LVM Physical Volumeداشته باشید می توانید آنها را برای عضویت در Group Volume انتخاب کنید. همانطور که گفته شد هر Group Volume می تواند چندین Physical Volume را داشته باشد.



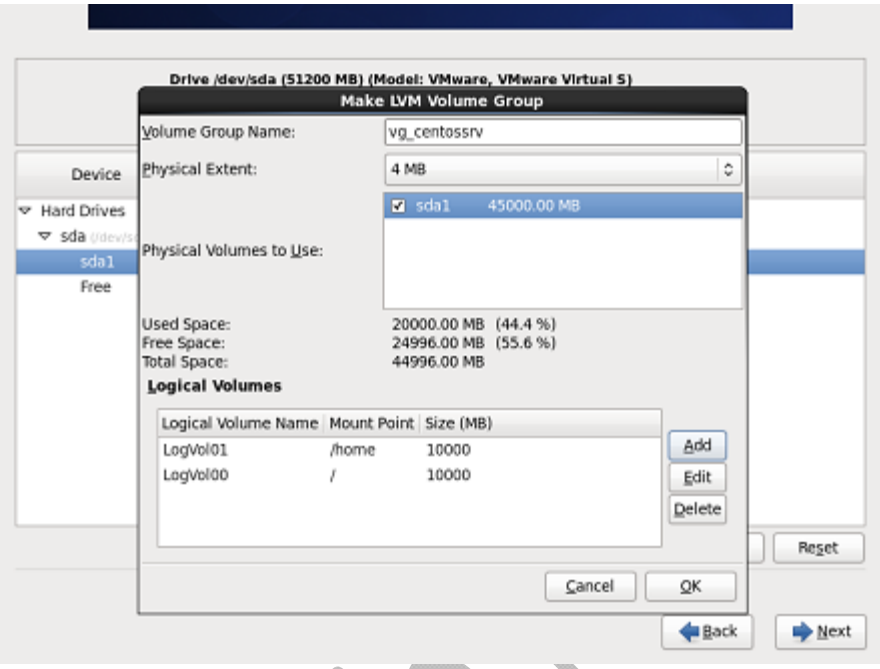
پس از تعیین نام Group Volume و انتخاب Physical Volumeهای آن و پیش از کلیک روی OK از شکل بالا، می توانید روی Addاز شکل بالا کلیک کنید و همین جا Logical Volume ها که نقش پارتیشن را دارند ایجاد کنید و آنها را به دایرکتوری های پیشفرض موجود در نصاب CentOS متصل کنید. به هر حال نخست می بایست Physical Volume ها و سپس Group Volume ها و در نهایت Logical Volumeها را ایجاد کنید. (یعنی می توانید از شکل بالا OK کند و سپس یک Logical Volumeبسازید. شکل زیر پس کلیک روی Add چگونگی ایجاد یک Logical Volume به اندازه 10000 مگابایت یا 10 گیگابایت را که روی دایرکتوری / یا دایرکتوری root با سیستم فایل ext4 نشان می دهد.



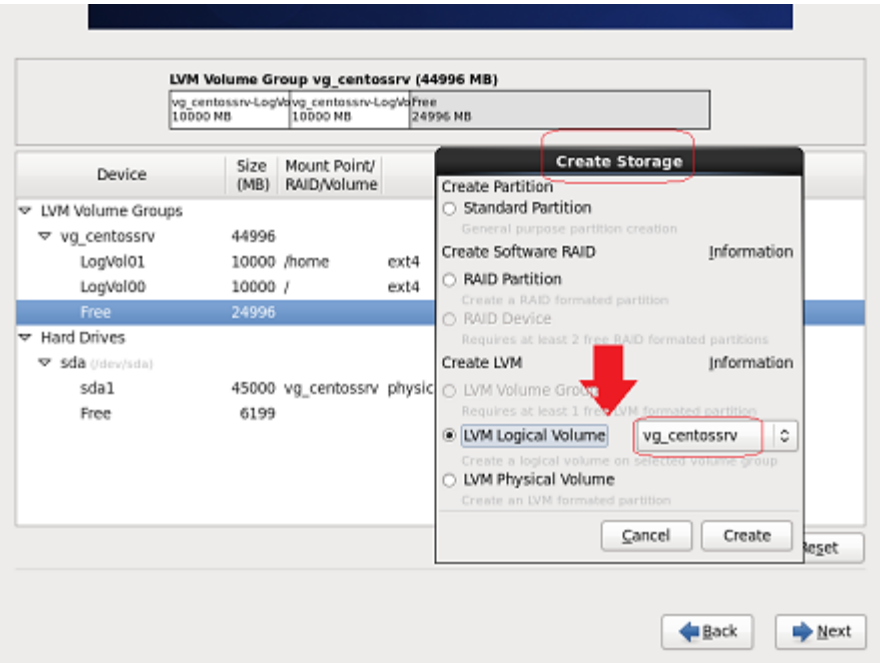
یا در شکل زیر یک از همان پنجره و پیش از OKکردن و کلیک بر روی Add یک Logical Volume به اندازه 10 گیابایت ایجاد می کنیم. همانطور که در شکل پایین مشاهده می کنید، دایرکتوری /boot در لیست نقاط اتصال نیست.



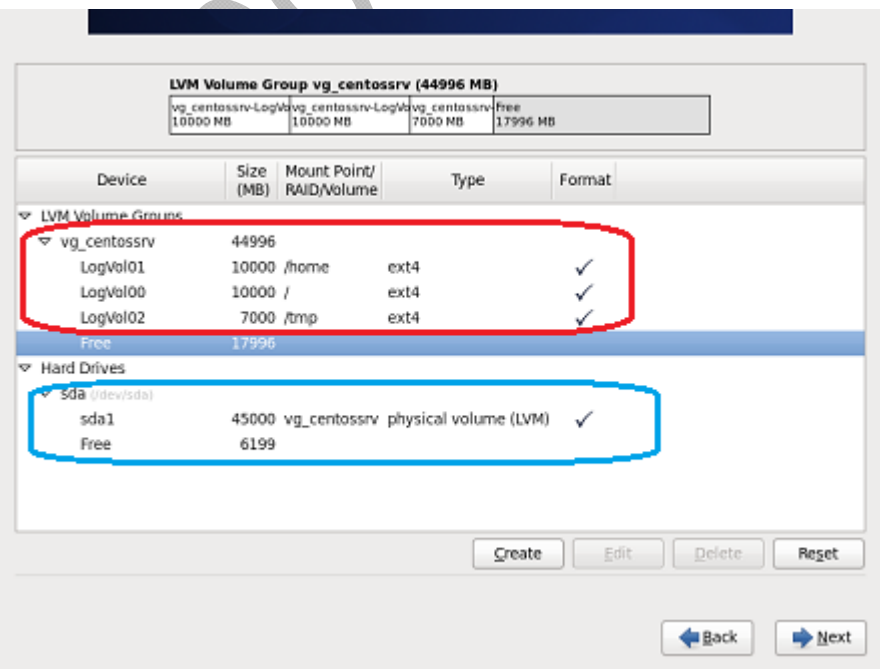
شکل زیر دو Logical Volume ایجاد شده را نشان می دهد. روی OK کلیک کنید.



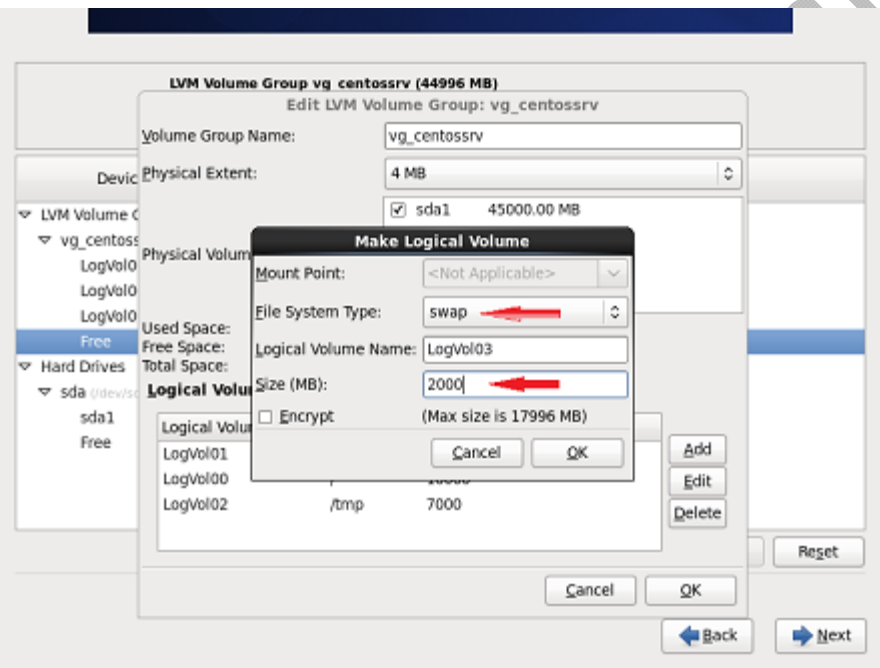
در پنجره شکل زیر نخست روی Create کلیک کنید تا پنجره Create Storage باز شود و از آن گزینه LVM Logical Volumeرا انتخاب کنید. در جلوی آن فهرستی از Group Volume ها خواهد آمد که باید گروه مورد نظر را انتخاب و در نهایت روی Create کلیک کنید تا یک Logical Volume دیگر برای اتصال بر روی دایرکتوری دیگری مانند /tmp ایجاد کنیم. (پنجره ای که پس از کلیک روی Createباز خواهد شد، مانند پنجره بالا است)



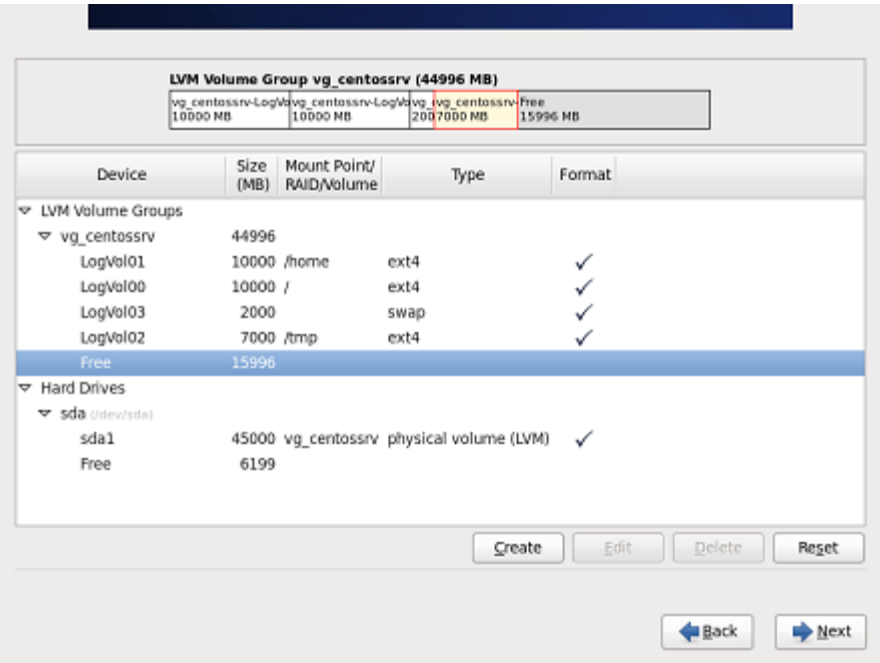
در شکل زیر خواهید دید که سه Logical Volume تحت گروه vg_centossrv ایجاد کردیم. همانطور مه می بینید هارد دیسک به دو قسمت 45 گیابایتی و 6 گیابایت فضای خالی تقسیم شده است. از این 6 گیگ فضای خالی بعد استفاده می کنیم تا دایرکتوری /boot را بر روی آن mount کنیم.



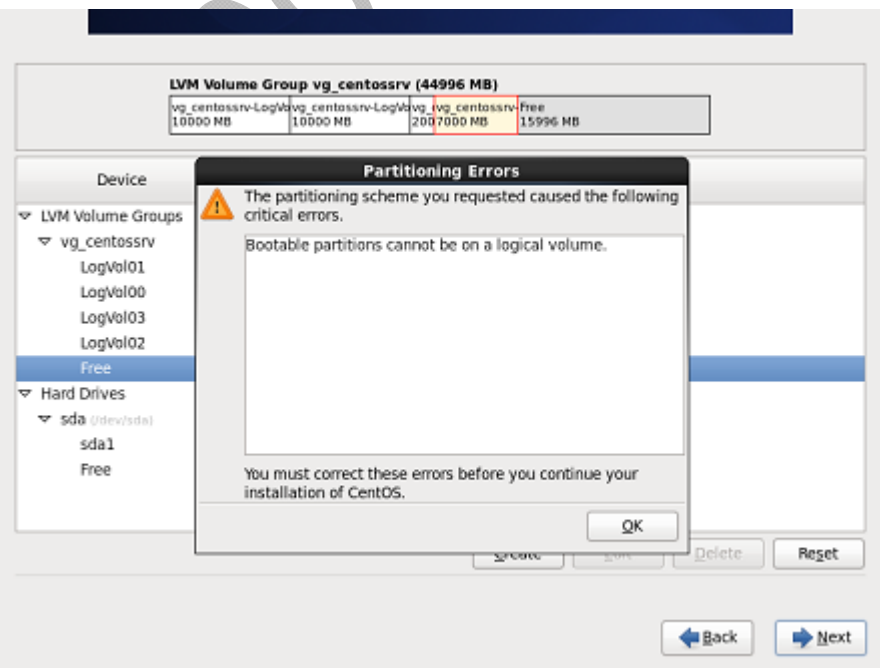
شکل زیر چگونگی ایجاد یک Logical Volume برای mount کردن پارتیشن swap را نشان می دهد swap .و دایرکتوری / دو دایرکتوری هستند که باید حتما ایجاد شوند. در دو مطلب 1و 2چگونگی ایجاد یک swap file و یک swap partition اضافی را گفته ام اما یکی دیگر از راه های ایجاد swap چه در زمان نصب و چه پس از نصب ایجاد آن تحت LVMاست و این باعث می شود که بعد با انعطاف بیشتری فضای آنرا تغییر دهید.



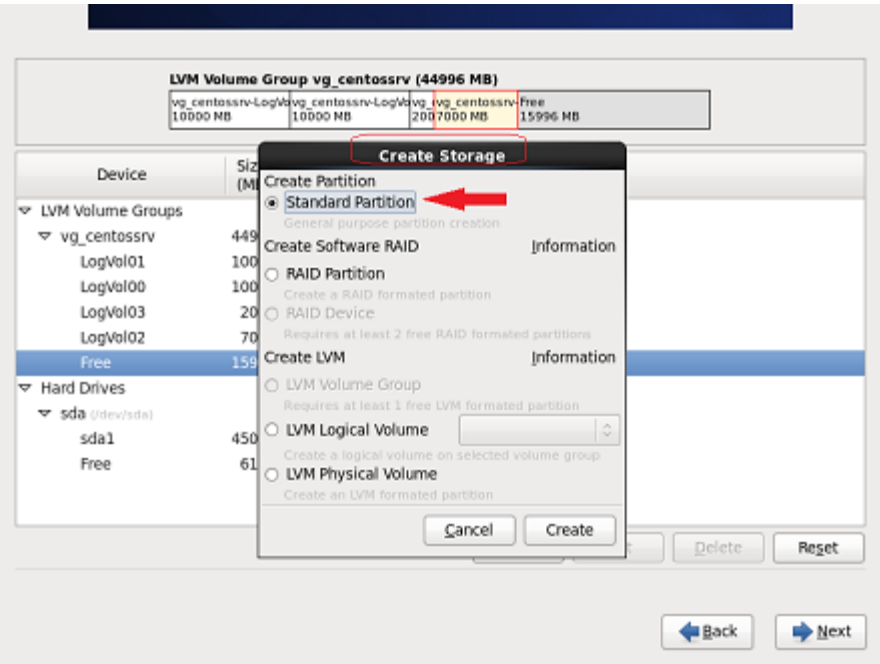
مطابق شکل زیر دو دایرکتوری لازم و اجباری بعلاوه دایرکتوری های دیگر ایجاد شده اند. روی Next کلیک کنید تا وارد مرحله بعدی نصب CentOS شویم.



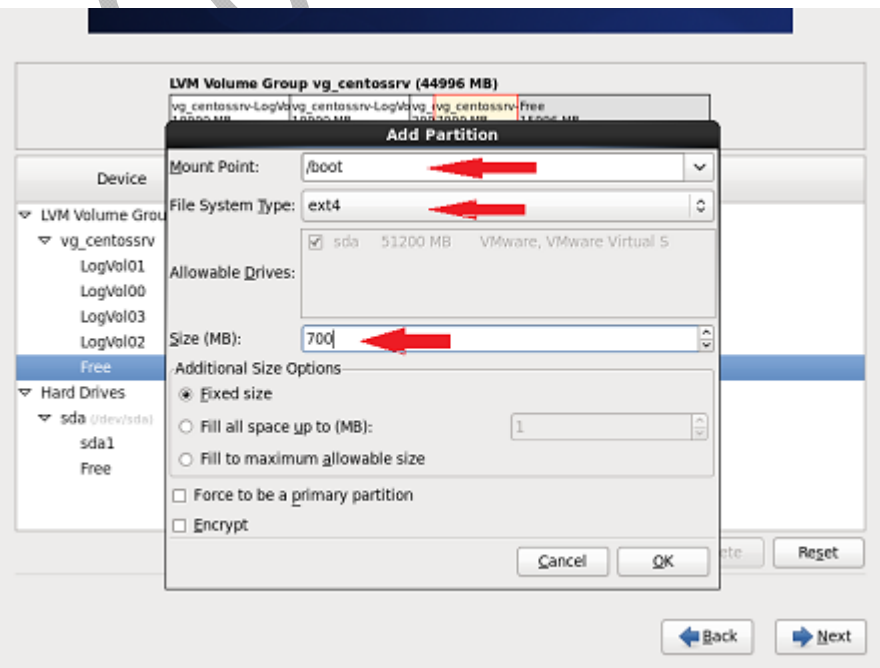
پس از کلیک روی Next شکل بالا باید خطای زیر نشان داده شود و به این خاطر است که دایرکتوری /boot تحت دایرکتوری / که به صورت LVM است و همانطور که در بالا گفته شد دایرکتوری /boot نمی تواند به صورت LVM باشد.



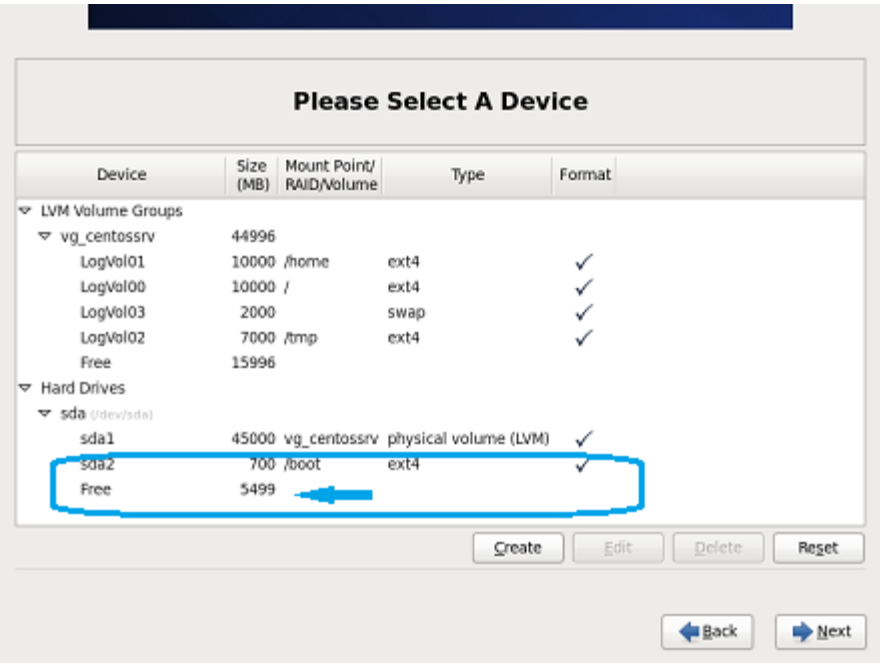
برای ایجاد دایرکتوری /boot از فضاهای Free f به اندازه 6 گیگابایت استفاده کنید (آنرا انتخاب کنید) و سپس روی Create کلیک تا پنجره Create Storage مانند شکل زیر باز شود و از شکل زیر Standard Partitionرا انتخاب کنید.



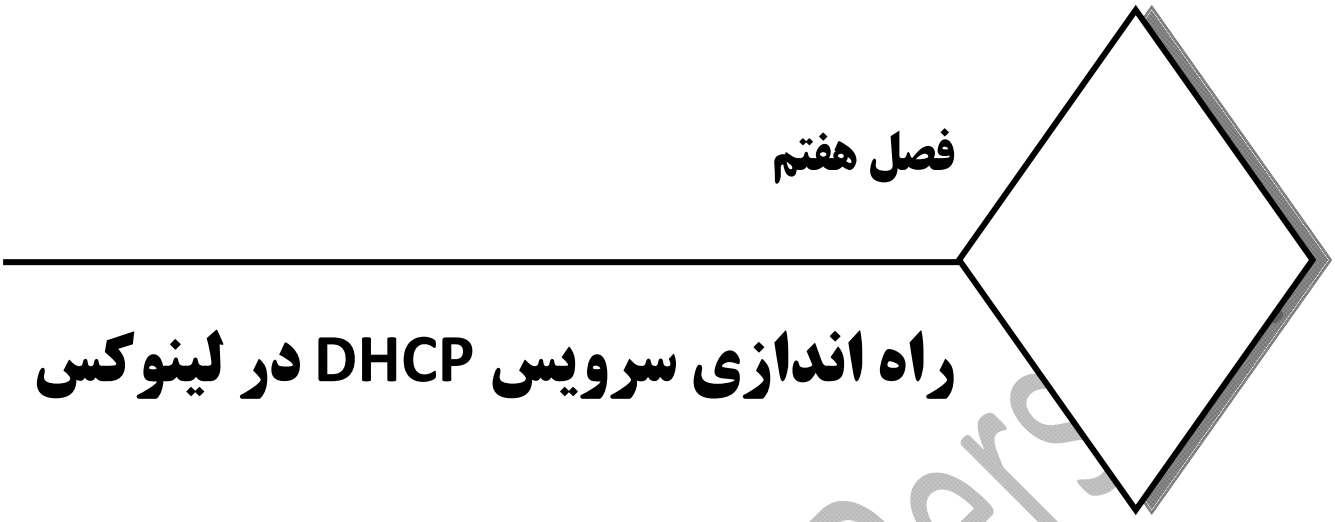
مطابق با شکل زیر من 700 مگابایت را به دایرکتوری /boot اختصاص داده ام.



مطابق با شکل زیر تمامی موارد لازم انجام شده و می توانید برای رفتن به گام بعدی نصب CentOS روی Nextکلیک کنید .همانطور که می بینید از فضای خالی 6 گیابایتی کم برای ایجاد /boot کم شده و نه از فضای آزاد در بخش.LVM Physical Volume



ایجاد LVMدر تمامی توزیع های لینوکس شامل همین سه مرحله Create LVM Physical Volume و سپس Create Group Volume و در نهایت Create Logical Volume و اتصال آنها به دایرکتوری های مورد نظر است. (توجه کنید LVM Logical Volume نقش پارتیشن را بازی می کنند). اگر بخواهید پس از نصب توزیعی، این کار را انجام دهید دستور هایی وجود دارد که آنها در تمامی توزیع های لینوکسی یکسان هستند اما در زمان نصب تنها در ظاهر واسط گرافیکی تفاوت هایی وجود دارد.



فصل هفتم

راه اندازی سرویس DHCP در لینوکس

سرویس DHCP در لینوکس

مقدمه

در شبکه‌هائی با ساختار کوچک پیکربندی شبکه کامپیوترهای رومیزی به صورت دستی امکان پذیر است اما پیکربندی شبکه سیستم‌ها در ساختار های بزرگتر امری است مشکل و بعضا غیر ممکن. حتی در شبکه‌های کوچک، پیکربندی سیستم‌ها با هر تغییراتی، فعالیت کاربران را مختل و نارضایتی هایی را در نحوه پشتیبانی بوجودخواهد آورد. در نتیجه راه حل این است که یک سرور فعالیت آدرس دهی شبکه و پیکربندی را بطور اتوماتیک انجام دهد.

پروتکل Bootstrap

این پروتکل (BootP) اولین پروتکل پیکربندی بوده که می‌تواند کلیه اطلاعات برای پیکربندیTCP/IP یک کلاینت که تقاضای IP نموده را فراهم کند تا براحتی با سرور شبکه و سایر کلاینت‌ها در ارتباط باشد. مکانیزم عملکرد این پروتکل بدین صورت است که یک بسته به نام BootRequest به پورت های UDP شماره ۶۷ همه دستگاه‌های شبکه ارسال می‌کند. کلاینت همه اطلاعاتی که راجع به خودش دارد را در بسته BootRequestقرار می‌دهد که می‌تواند فقط آدرس لایه فیزیکی باشد. فقط هنگامی که یک سرور BootPیک بسته روی پورت ۶۷ دریافت می‌کند، یک بسته BootReply می‌سازد و داخل آنرا تا آنجایی که بتواند با اطلاعات پیکربندی مورد نیاز کلاینت پر می‌کند. سپس سرور با استفاده از پورت شماره ۶۸ UDP، بسته را روی شبکه انتشارمی دهد. کلاینت روی پورت ۶۸ منتظر دریافت بسته است. هنگام که یک بسته روی پورت دریافت کرد که حاوی آدرس فیزیکی‌اش است از اطلاعات داخل بسته برای پیکربندی Tcp/IP خود استفاده می کند.

پروتکل DHCP (Dynamic Host Configuration Protocol)

dhcp مخفف Dynamic Host Configuration است و وظیفه آن واگذاری اتوماتیک تنظیمات tcp/ip است..

در لینوکس سرویس dhcpd برای همین منظور استفاده می شود و پیکر بندی آن بسیار ساده و راحت است. یک سرویس dhcpd از یک Scopeکه در لینوکس به آن range گفته می شود به همراه یک سری از optionها تشکیل شده است. تعیین range و تعیین آدرس شبکه برای اختصاص دادن به کلاینت ها لازم و ضروری است اما استفاده از option ها اختیاری است.

DHCPروی پورت‌های مشابه BootP یعنی ۶۷ و ۶۸ کار می‌کند و کلیه سرویس‌های BootP بعلاوه تعدادی کاربرد مهم دیگر را ارائه می‌دهد. یک خاصیت مهم اضافه در DHCP این است که آدرس‌ها می‌توانند بصورت پویا به یک سیستم داده شوند.آدرس IP که با استفاده ازBootP ، به یک سیستم تخصیص می یابد، به صورت دائم برای آن در نظر گرفته می شود و سیستم دیگری در شبکه نمی تواند از آن استفاده کند. اما DHCP آدرس را برای یک مدت زمانی مشخص به کلاینت اجاره می‌دهد و بعد از اتمام دوره اجاره، می‌تواند آن آدرس را به کلاینتهای دیگر واگذار کند. مزیت استفاده از آدرس‌دهی پویا، استفاده بهینه از تعداد کم آدرس است. آدرس‌های بلا استفاده برای استفاده سایر کلاینت‌ها آزاد می‌شود. آدرس دهی پویا بوسیلهDHCP ، مانند همه چیزهای دیگر بدون نقص نیست. از آنجاییکه DNS از آدرس‌هایی که توسط DHCPتخصیص می‌یابد اطلاعی ندارد در نتیجه کامپیوترهای خارج از شبکه داخلی نمی‌توانند سیستمی را که از طریقDHCP ، آدرس دریافت کرده را پیدا کنند در نتیجه این سیستم نمی‌تواند به سایر سیستمهای راه دور سرویس ارائه دهد. این یک نقص است ولی باعث اختلال در شبکه نمی شود.

از آنجاییکه اولا فقط سرورها باید به سایر سیستم‌ها سرویس ارائه کنند، می‌توان پیکر بندی شبکه آنها را بصورت دستی تنظیم نمود. دوم اینکه تعداد سرورها نسبت به مجموع تعداد سیستم‌ها کمتر است. در نتیجه هزینه و حجم پیکربندی سرورها به نسبت کمتر می باشد. پس نتیجه می‌گیریم که سیستم‌های رومیزی نسبت به سرورهاگزینه های بهتری برای پیکربندی توسط DHCP هستند. اما تکنیکهایی برای انطباق آدرس‌های DHCPو DNS با استفاده از Dynamic DNS (DDNS) وجود دارد که در ادمه این مبحث توضیح داده خواهد شد.

شروط دریافت IP توسط کلاینت لینوکسی

کلاینت لینوکسی به دو شرط می تواند از dhcp آدرس IP و دیگر تنظیمات را بگیرد که این بر عکس ویندوز می باشد.

1. شرط اول بین مایکروسافت و بین لینوکس مشترک است، آنکه گزینه obtain فعال شود. البته در لینوکس چون ممکن است محیط گرافیکی نداشته باشیم باید از طریق خط فرمان وارد مسیر و فایل /etc/sysconfig/network-script/ifcfg-eth0 شده گزینه BOOTPROTO را برابر static قرار دهیم.
2. شرط دوم فقط مخصوص لینوکس است . یک کلاینت لینوکسی به شرطی می تواند از dhcp آدرس IP بگیرد که در فایل /ets/sysconfig/network گزینه NETWORKING مقداری برابر yes داشته باشد. اگر این گزینه برابر yes نباشد برقرار بودن شرط اول هم نمی تواند در گرفتن IP هیچ کمکی به ما بکند.

مبحث DHCP Failover

مفهوم failover مانند سرورهای بکاپ می باشد. یعنی اگر سرور اصلی ما down شد سرور دیگری باشد که به درخواستها پاسخ بدهد. در سرورهای چیزی به نام failover وجود ندارد بلکه می آیند ناحیه ای که می خواهند به آن IP بدهند را به دو scop تقسیم می کنند یعنی یک رنج ip را به دو قسمت تقسیم کرده که اگر یکی از scop ها از کار افتاد scop دیگر بتواند به شبکه مشخصات tcp/ip را واگذار کند.

در لینوکس برای failover دو سرور slave و master راه اندازی می شود. به روز رسانی سرور slave از طریق سرور master صورت می گیرد. فاصله زمانی این آپدیت ها هر 10 دقیقه یکبار می باشد. حال اگر برای 10 دقیقه ، سرور master آپدیتی ارسال نکند در دو مرحله توسط سرور slave درخواست آپدیت ارسال می شود اگر جوابی دریافت نکند سرور slave نقش slave/master را با هم اجرا کرده و شروع به سرویس دادن به شبکه می کند. حال اگر سرور master فعال شود ، تا زمانی که دیتابیس آن به روز نشود

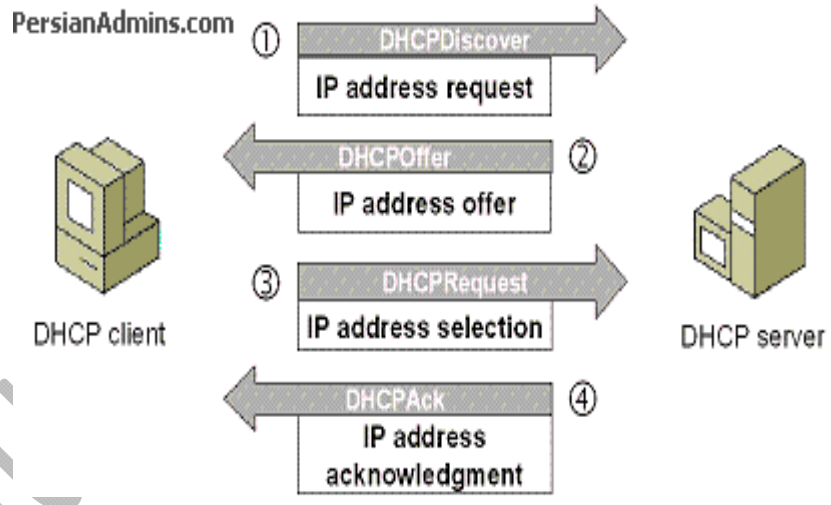
نقش slave به خود می گیرد و در این نقش باقی می ماند اما به محض آپدیت به نقش قبلی خود بر می گردد.

نکته مهم : تحت هیچ شرایطی نباید سرورهای شبکه ، سوئیچهای قابل مدیریت ، پرینترهای تحت شبکه و کلا دیوایس های مهم شبکه را با dhcp آدرس دهی کنیم بلکه باید آدرس آنها را در سرور dhcp رزرو کرده و به صورت static آدرس را بر روی آنها تنظیم کنیم.

مراحل Dora

یک کلاینت طی 4 مرحله که به آن Dora گفته می شود از سرور IP اجاره می کند.

1. Dhcp Discovery
2. Dhcp Offer
3. Dhcp Request
4. Dhcp Acknolegment



Dhcp discovery

در مرحله اول کلاینت بعد از up شدن متوجه می شود با وجود فعال بودن خاصیت obtain هنوز آدرس IP ندارد لذا یک درخواست تولید کرده و آدرس مبدا آن را 0/0/0/0 و آدرس مقصد آن را 255/255/255/255 قرار می دهد. وقتی که آدرس مقصد چهار عدد ، 255 باشد یعنی این بسته از نوع Broadcast بوده و از روی پورت 68 بر روی شبکه ارسال می شود. این پکت در شبکه فریاد می زند که

این سیستم IP ندارد آیا کسی هست به آن IP واگذار کند. چون کارت شبکه IP ندارد از آدرس 0/0/0/0 استفاده می کند.

Dhcp Offer

در دومین مرحله ، اولین dhcp سروری که پکت offer را دریافت کند از scop خودش یک IP به صورت random انتخاب کرده و آن را به صورت uncast از روی پورت 67 به کلاینت پیشنهاد می دهد. علت اینکه پیام سرور unicast میباشد این است که سرور از بسته دریافتی آدرس mac فرستنده را استخراج کرده است به چنین پکتی در دنیای شبکه Dhcp offer گفته می شود.

Dhcp Request

در مرحله سوم ، اگر کلاینت IP ارسالی از سرور را قبول کند یک پکت به نام dhcp request تولید کرده و به صورت broadcast در شبکه اعلام می کند که این IP متعلق به خودش بوده و آن را قبول کرده است. دلیل این کار این است که اگر dhcp سرور دیگری در شبکه وجود داشته باشد متوجه شود که این کلاینت از چه IP ای استفاده می کند.

Dhcp Acknowledgment

در مرحله چهارم سرور dhcp با پکت dhcpAck تائید می کند که IP مورد نظر به چه کلاینت تعلق دارد . این پکت هم از نوع uncast می باشد.

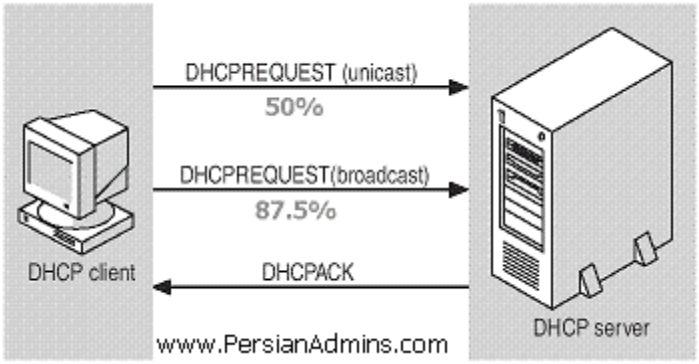
خلاصه مراحل Dora

پس از اینکه گزینه obtain ip addressautomatically را در client انتخاب کردیم مراحل زیر اتفاق می افتد :

1. DHCP Client یک بسته **DHCP Discover** را برای جیستجوی DHCP Server ، به صورت همگانی منتشر می کند.

2. DHCP server بوسیله بسته **DHCP Offer** یک IP address به clinet تقدیم میکند.
3. DHCP Clinet یک بسته را که **DHCP Requset** مینامیم به DHCP Server به منظور تحقیق اینکه آیا **Server DHCP** معتبراست یا خیر خواهد فرستاد.
4. DHCP server بوسیله بسته **DHCP acknowledgement** جواب client را خواهد داد.

مکانیزم Lease Duration



IP هایی که سرور dhcp به کلاینتها می دهد دائمی نبوده و اجاره ای می باشد و برای خودشان زمان اجاره دارند که به آن Lease Duration گفته می شود.

در سرورهای مایکروسافتی زمان پیش فرض اجاره IP ، هشت روز ، در سرورهای لینوکسی چهار روز و در سیستم های سیسکو یک روز می باشد.

زمان T1 :

کلاینت وقتی IP را گرفت باید تا 50 درصد زمان پیش فرض که معادل 2 روز می باشد از آن استفاده کند. به این زمان 50 درصدی ، T1 گفته می شود.

کلاینت در روز دوم که مساوی است با همان 50 درصد استفاده شده از IP ، باید نسبت به تمدید زمان اجاره اقدام کند.

زمان T2 :

حال اگر سرور dhcp فعال نباشد که این آدرس را تمدید کند ، کلاینت از این IP استفاده می کند تا به 87.5 درصد زمان اجاره برسد. به این زمان T2 گفته می شود. در این تایم باید زمان اجاره IP تمدید شود، در صورتی که سرور فعال نباشد کلاینت آدرس IP را آزاد کرده و آدرس 0/0/0/0 را می گیرد و مراحل Dora را دوباره انجام داده تا یک سرور dhcp جدید پیدا کند. در ویندوز دستور ipconfig /renew کار تمدید اجاره ip و دستور ipconfig /release آدرس قبلی را آزاد کرده و آدرس 0/0/0/0 را به کارت شبکه می دهد تا دوباره دنبال یک سرور dhcp بگردد .

در اینجا مراحل Dora هر 32 و 16 و 8 و 4 و 0 ثانیه یکبار تکرار می شود، اگر سرور dhcp را پیدا نکرد که از آن IP بگیرد کارت شبکه می رود سراغ آدرس های Apipa که از رنج 169.254.0.0 یک IP بگیرد. بعد از اینکه کلاینت از Apipa آدرس IP گرفت هر 5 دقیقه یکبار در شبکه دنبال سرور dhcp می گردد که این کار ترافیک شبکه را بالا برده و کارائی آن را پائین می آورد.

با رنج Apipa نمی توان بر روی اینترنت رفت ، زیرا Apipa فقط IP واگذار می کند و آدرس default gateway اعلام نمی کند و دلیل دیگر آن این است که رنج Apipa در اینترنت مسیر یابی نمی شود.

مکانیزم Duplicate Address Detection

کلاینت قبل از اینکه به صورت نهایی IP را روی کارت شبکه اش set کند یک ARP به IP خودش ارسال می کتد. با این کار می فهمد آیا IP پیشنهادی از سوی سرور به کلاینت دیگری تعلق دارد یا فقط توسط خودش استفاده می شود . اگر سیستمی از این IP استفاده کند ، کلاینت می فهمد که duplicate اتفاق افتاده و IP تکراری است لذا دوباره مراحل چهار گانه DORA را انجام می دهد. به این عمل DAD گفته می شود.

در این شرایط کلاینت یک پکت به نام dhcp client تولید کرده و به سمت سرور ارسال می کند تا به سرور بفهماند این آدرس Duplicate است و سرور باید یک IP دیگر را پیشنهاد بدهد. DAD یک مکانیزم برای کشف IP های تکراری است.

نصب و راه اندازی سرویس Dhcp

اکثر توزیع‌های لینوکس، DHCPرا در مراحل اولیه نصب در صورت درخواست کاربر، برروی سیستم نصب می‌کنند. اگر لینوکس مورد نظر شما حاوی سرویس DHCP نبود یا یک نسخه جدیدتر از نسخه‌ای که به همراه سیستم‌تان است را می‌خواهید، می توانید به راحتی آن را توسط دستور Yum آخرین ورژن آن را نصب کنید.

dhcpd با هسته لینوکس ۲.۴ به بالا به خوبی کار می‌کند اما اگر لینوکس مورد استفاده از هسته قدیمی استفاده می کند امکان دارد مشکلاتی بوجود آید که یکی از آنها مشکل سرویس‌دهی به کلاینتهای ویندوزی است. پس ترجیحاً از نسخه‌های بالاتر DHCP و کرنل‌های جدیدتر استفاده کنید.

در بیشتر زمان ها dhcpd در موقع نصب سیستم نصب می شود و شما نیازی به نصب مجدد آن ندارید اما اگر به هر نحوی در سیستم موجود نبود طبق روش زیر عمل می کنیم :

ابتدا باید از نصب بودن پکیج dhcpd اطمینان حاصل کنیم لذا با دستور زیر از سیستم query میگیریم :

#rpm -qa | grep dhcp

در صورت نصب نبودن ، در سیستم های ردهت جهت نصب dhcpdاز yum استفاده می کنیم :

#yum –y install dhcp

بعد از نصب ، باید اطمینان حاصل کنیم که آیا پکیج dhcpdبر روی سیستم نصب شده است یا خیر لذا با دستور زیر از سیستم query می گیریم :

#rpm -qa | grep dhcp

سپس با دستور زیر شاخه ها و مسیرهایی که فایل های این سرویس در آن ایجاد شده است را چک می کنیم :

#rpm -ql dhcp

و با این دستور هم اطلاعات لازم را در مورد پکیج این سرویس به دست می آوریم :

#rpm -qi dhcp

سپس با دستور chkconfig مشخص می کنیم در چه run level هایی فعال باشد :

 # chkconfig --level 35 dhcpd on
 # chkconfig --list dhcpd

و در انتها سرویس را reset می کنیم :

#service dhcpd restart

در اینجا سرویس استارت نمی شود و به ما پیام Failed نشان می دهد. دلیل آن هم این است که فایل پیکربندی سرویس dhcp خالی است . برای اینکه جلوی Failed دادن آن را بگیریم باید فایل کانفیگ آن را مقدار دهی کنیم . این فایل در مسیر /etc/dhcp/dhcpd.conf قرار دارد که در ادامه آپشن های موجود در آن و شیوه تنظیمات این سرویس را توضیح خواهیم داد.

پیکربندی و تنظیمات سرویس Dhcp

همانطور که گفته شد فایل کانفیگ dhcp در مسیر /etc/dhcp/dhcpd.conf قرار دارد و درون آن به صورت پیش فرض خالی بوده و کانفیگ خاصی ندارد. علت آن هم این است که سرویس dhcp به محض فعال شدن بدون کانفیگ خاصی شروع به کار کردن می کند، لذا اگریک dhcp دیگر در شبکه وجود داشته باشد شبکه دچار Collision و چند دستگی می شود.

در لینوکس هزاران فایل راهنما و نمونه برای راه اندازی یک سرویس وجود دارد، dhcp از این قائده مستثنی نیست و برای راه اندازی و کانفیگ این سرویس باید به آنها رجوع شود.

فایل نمونه پیکربندی dhcp در مسیر زیر قرار گرفته است که باید آن را به دایرکتوری اصلی سرویس کپی کنیم :

 # cp/usr/share/doc/dhcp-4.1.1/dhcpd.conf.sample /etc/dhcp/dhcpd.conf

قبل از کپی ، سیستم عامل پیغام می دهدآیا مایل به owerwrite فایل موجود هستید یا خیر که ما با تایپ کلمه yes این کار را تائید می کنیم. بعد از کپی وارد مسیر اصلی فایل کانفیگ شده و آن را با یکی از ادیتورهای موجود باز می کنیم. این فایل از بخش های اصلی و option هایی تشکیل شده است که در پایان هر بخش و یا option علامت سمی کولون (;) قرار می گیرد. optionها میان {} قرار گرفته وتمامی دستورات با حروف کوچک شروع می شوند.

زبان پیکربندی شامل امکانات زیادی است که عملکرد سرور و پروتکل DHCP را کنترل می‌کند. اول لغات Allow و Deny که کنترل می‌کنند dhcpd درخواستهای کلاینت‌ها را چگونه مورد کنترل قرار دهد. هر

عبارت با دستور allow یا deny به دنبال کلمه کلیدی که بیان می‌کند چه درخواستی مورد قبول یا رد قرار گیرد، آغاز می‌شود. سه کلمه کلیدی ممکن، عبارتند از:

1. unknown_clients
2. bootp
3. booting

unknown_clients: درخواست‌های پیکربندی، از کلاینت‌هایی که برای سرور نا آشنا هستند و تعریف نشده اند، می‌توانند قبول یا رد شوند. به طور پیش‌فرض این کلاینتها مورد قبول واقع می‌شوند در غیر اینصورت برخی از قابلیت‌های DHCP از بین می‌رود.

bootp: سرور می‌تواند پیکربندی درخواست‌هایی که از کلاینت‌های bootp می‌آیند را قبول یا رد کند. به طور پیش‌فرض کلاینتهای bootp مجازند که توسط این سرور پیکربندی شوند در نتیجه تمام کلاینتهای bootpو DHCP توسط یک سرور پیکربندی خواهند شد.

booting: عبارت deny booting در قسمت host استفاده میشود و به سرور می‌گوید که سرور نباید درخواست یک کلاینت مشخص را بررسی کند. پیش فرض این است که درخواست‌های کلاینت جواب داده شود و احتیاجی به عبارت allow booting نیست. علاوه بر سه عبارت که در بالا توضیح داده شد، پارامترهای پیکربندی متفاوتی وجود دارند که عملیات پروتکل را کنترل می‌کنند که در اینجا به برخی از آنها اشاره می گردد.

 # vi /etc/dhcp/dhcpd.conf

با استفاده از فایل dhcpd.conf ، میتوان هر اطلاعات پیکربندی که مورد نیاز هر میزبان یا زیرشبکه‌ای که از سیستم شما سرویس می گیرد را فراهم نمود. در ادامه بعضی از قسمتهای این فایل را توضیح می دهیم:

ddns-update-style interim;
ignore client-update;

این خط معرف ddns یا همان داینامیک dns است . این فیچر به dhcp می فهماند به هر کسی IP داد باید نام آن را در داخل dns ثبت کند. این خط می تواند سه مقدار زیر را داشته باشد:

- interim : این گزینه در dhcp های ورژن بالا مورد استفاده قرار می گیرد که به ما فیچرهایی مثل dhcp failower را می دهد و معمولا هم از همین گزینه استفاده می شود.
- adhoc : این گزینه یک فیچر قدیمی است و در dhcp سرورهای قدیمی مورد استفاده قرار می گرفت.
- none : این عبارت ویژگی ddns را غیر فعال می کند.

authoritative;

از این گزینه برای جلوگیری از حملات dhcp Rouge استفاده می شود. در این نوع از حملات یک dhcp تقلبی توسط یک نفوذگر وارد شبکه شده و باعث چند قسمتی شدن آن می گردد. برای جلوگیری از این نوع حملات باید به صورتی به کلاینت ها بفهمانیم که به غیر از سرور اصلی از سرور دیگری IP نگیرد. این گزینه دقیقا همین کار را انجام می دهد. به طور مثال می توانیم این خط را به این صورت authoritative 192.168.10.1; مقدار دهی کنیم . در این خط مشخص کرده ایم که اگر کلاینتها نتوانستند از سرور اصلی ip بگیرند فقط از این آدرس درخواست IP کنند . می توانیم چندین آدرس را پشت سر هم و یا در چندین خط وارد کنیم.

default-lease-time 84600;

زمان اجاره IP بر اساس ثانیه را مشخص می کند. (زمان پیش فرض) اگر کلاینت مدت زمان اجاره را مشخص نکند. سیستم این زمان را به طور پیش فرض برای مدت زمان اجاره یک آدرس در نظر می گیرد.

max-lease-time 604800;

بیشترین مدت زمان اجاره IP را بدون در نظر گرفتن مدت زمان درخواستی کاربر بیان می کند. این مدت زمان به ثانیه بیان می شود.

get_lease_hostnames flag false;

اگر flag برابر true باشد، dhcpdیک جستجوی معکوس برای هر آدرسی که اجاره می‌دهد انجام می‌دهد. داده و نام میزبان را از DNS دریافت نموده و به کلاینت ارسال می‌کند. این فرایند می‌تواند باعث بالا رفتن حجم فعالیت سرور روی شبکه‌های بزرگ شود. این flag به صورت پیش‌فرض false است و هیچ جستجویی در DNS صورت نمی‌گیرد.

آپشن ها (Options)

آپشن ها تنظیمات اضافه تری هستند که dhcp علاوه بر IP به کلاینتها واگذار می کند. تنظیماتی مثل dns,gw,ntp و .... خطوطی که جزو آپشن محصوب می شوند حتما باید با کلمه options شروع گردند خطوطی مثل :

option domain-name tohid.com;

این خط نام domain شبکه را مشخص می کند.

option domain-name-server 4.2.2.4;

آدرس dns را مشخص و اعلام می کند.

option router 10.6.4.1;

آدرس default gateway را مشخص و اعلام می کند.

option time-offset +12000;

این آپشن محدوده زمانی location ما را اعلام می کند.

option nntp-server 192.168.1.100;

آدرس تایم سرور را اعلام می کند. اکتیو دایرکتوری در شبکه نقش تایم سرور را بازی می کند.

option subnet-mask 255.255.255.0

این آپشن آدرس زیر شبکه را اعلام می کند.

option broadcast-address 10.10.10.255

آدرس اعلام همگانی شبکه را به کلاینتها اعلام می کند.

Option Levels

آپشن ها در محل های مختلفی می توانند اعمال شوند . پنج سطح آپشن وجود دارد که در زیر به ترتیب اولویت آمده است. ترتیب اعمال آپشن ها از بالا به پائین می باشد.

- host
- group
- subnet
- shared-network
- global

Option at Host level: در dhcp اگر بخواهیم چیزی را برای کسی رزرو کنیم آن را با آپشن host معرفی می کنیم مثل رزرو IP. آپشن host ، آپشنی است که به یک کلاینت خاص اعمال می شود. مثلا در این سطح از آپشن می توانیم مشخص کنیم چه IP ای به چه mac آدرسی تعلق بگیرد(Reservation IP) و یا چه مشخصاتی به آن داده شود. این آپشن جزو Policy های امنیتی شبکه می باشد. آدرس اترنت که در عبارت host قرار دارد به عنوان کلیدی برای شناسایی اینکه چه کلاینتی چه اطلاعات پیکربندی دریافت می‌کند استفاده می‌شود. در مثال، نام میزبان و آدرس IPکلاینت مشخص شده است. به کلاینت‌ها آدرس‌های ایستا داده می‌شود بنابر این احتیاج به تمدید مدت اجاره آدرس‌ها ندارند. به غیر از اطلاعاتی که در عبارت host برای کلاینت مشخص شده، dhcpd برای آنها الگو زیرشبکه، نام دامنه، آدرس‌های سرور DNS و آدرس سرور چاپگر که در قسمت عمومی فایل پیکربندی تعریف شده است را نیز می فرستد.

مثال :

```
host myserver {
  hardware Ethernet xxxxxxxxxxxx;
  fixed-address 192.168.6.200;
```

```
  option  router  192.168.6.1;
}
```

Option at group Level: این نوع از آپشن برای چند کامپیوتر مشخص شده قابل اعمال است. پیکربندی یک گروه با عبارات host خاتمه می‌یابد. عبارت گروه از مجموعه‌ای از عبارات host تشکیل شده است. می توان در عبارات host از پارامتر use_host_decl_name استفاده کرد که برای همه میزبان‌ها اعمال می‌شود.

نمونه:

```
group {
    option    [use-host-decl-names  true;]
       host  PC1  {
.
.
.
.
.
     }
      host  PC2  {
.
.
.
.
.
     }
}
```

Option at Subnet level: این نوع از آپشن برای یک رنج IP خاص قابل اعمال می باشد. شبکه‌ای که dhcpd به آن سرویس ارائه می‌دهد توسط یک آدرس و یک الگو آدرس در عبارت subnetمشخص می‌شود. dhcpd تنها به کلاینتهایی که در این شبکه قرار دارند و یا مستقیماً توسط

عبارات host مشخص شده‌اند، سرویس ارائه می‌کند. گزینه‌ها و پارامترها در عبارت subnet فقط برای زیر شبکه و کلاینتهای آن اعمال می شود عبارت subnet در مثال زیر دارای دو محدده برای ساخت چند گروه مجزا برای آدرس های پویا است. این نشان می‌دهد می توان یک فضای آدرس گسسته با استفاده از چند عبارت range داشت.

نمونه:

```
subnet 192.168.6.0 netmask 255.255.255.0 {
    range 192.168.6.1    192.168.6.200;
}
subnet 192.168.7.0 netmask 255.255.255.0 {
    range dynamic-bootp 192.168.7.1    192.168.7.200;
}
subnet 192.168.9.0 netmask 255.255.255.0 {
    range 192.168.91    192.168.9.19;
    range 192.168.9.31    192.168.9.200;
    option domain-name-server 4.2.2.4;
    option router 192.168.9.250;
}
```

در قسمت آخر ما از IP ، 20 تا 30 را جدا کردیم. آپشن هایی که در زیر هر subnet آورده شوند ، مختص به همان ساب نت بوده و فقط به آن اعمال می شود. آپشن های هر ساب نت باید بین دو اکولاد اصلی آن قرار بگیرند تا به آن اعمال شوند.

نکته: اگر پارامتر range در یک عبارت بیان شده باشد و هر کلاینت DHCP در subnet ای درخواست آدرس نماید، تا زمانیکه آدرس موجود باشد، به آن کلاینت آدرس مورد نظر اختصاص می‌یابد و اگر پارامتر range بیان نشده باشد آدرس‌دهی پویا فعال نمی‌شود.

Option at Shared-network level: این آپشن به چند رنج IP اعمال می شود. در لینوکس به تعداد رنج IP می توان بر روی کارت شبکه IP تعریف کرد لذا با یک سرور می توان چندین رنج آدرس واگذار کرد. بهترین مثال برای این قسمت تقسیم IP در یک سوئیچ با vlan های مختلف است.

نمونه :

```
shared-network   tohid {
    option
        subnet  192.168.5.0    netmask  255.255.255.0 {
.
.
.
.
.
      }
      subnet  192.168.6.0    netmask  255.255.255.0 {
.
.
.
.
.
      }
}
```

Option at Global level: محتویات این آپشن به تمام حوزه کاری dhcp اعمال می شود. در کل option هایی که بین آکولاد باز/بسته {} باشد تنها در همان حوزه اثر دارد اما می توان option هایی را نیز تعریف کرد که بصورت سراسری یا globalباشند. فرض کنید یک option مانند routers که default getwayرا نشان می دهد هم بصورت سراسری و هم بصورت محلی (داخل آکولاد باز/بسته) باشد اما مقادیرشان با هم متفاوت است. در این صورت اولویت با option محلی است یعنی آن option ی که در حوزه محلی می باشد الویت بالاتری نسبت به option سراسری دارد.

شروع به کار سرویس Dhcp

بعد از پایان تعاریف و تنظیمات ابتدا باید از صحت درستی تغییرات اطمینان حاصل کرده و سپس سرویس را استارت کنیم. برای این کار از دستور dhcpd بهره می بریم. این دستور صحیح بودن تنظیمات فایل Config را بررسی کرده و در صورتی که مشکلی وجود داشته باشد آن را نشان خواهد داد. این دستور به طور معمول دو error را گوش زد می کند که در زیر به توضیح آنها پرداخته ایم:

1. No Configured to listen on any interfaces
2. No Subnet declaration for ethxx

No Configured to listen on any interfaces: معنی این پیغام این است که ما هیچ کارت شبکه ای را روی سرور مشخص نکرده ایم تا به پکتهای Dora گوش کند. با edit فایل زیر برای آن اینترفیس لازم را مشخص می کنیم:

```
# vi   /etc/sysconfig/dhcpd
DHCPDARGS=ethxxx
```

در جلوی این عبارت ، نام کارت شبکه را بدون فاصله وارد می کنیم.ممکن است سرور 10 عدد کارت شبکه داشته باشد ولی آن کارت شبکه ای که اینجا مشخص می کنیم می تواند به کلاینتهاIP واگذار کند. در ویندوز به این کار Binding گفته می شود. در لینوکس حتما باید کارت شبکه ای که IP پخش می کند را مشخص کنیم که این بر خلاف ویندوز است.

No Subnet Declaration for ethxx: این پیام هم می گوید کارت شبکه سرور روی یک رنج آدرس قرار دارد اما هنوز subnet ای برای آن تعریف نشده است. بعد از برطرف کردن اشکالات نشان داده شده دوباره دستور dhcpd را اجرا می کنیم. این دستور علاوه بر اینکه فایل Config سرویس را خوانده و صحت آن را چک و سرویس را start می کند. لذا دیگر نیازی به استفاده از دستور start نیست.

نکته: سرویس dhcpd به صورت Standalone کار کرده و زیر مجموعه init محسوب می شود.

آمار IP های واگذار شده

dhcp آمار آدرس های واگذار شده را در فایل dhcpd.leases نگهداری می کند که در مسیر /var/lib/dhcpd/dhcpd.leases قرار دارد. اگر این فایل وجود نداشته باشد سرویس دچار مشکل جدی خواهد شد. هنگامی که سرور برای اولین بار نصب و استارت می شود یک فایل خالی dhcpd.lease ایجاد میکند تا سرویس از ابتدا به درستی شروع به کار کند.

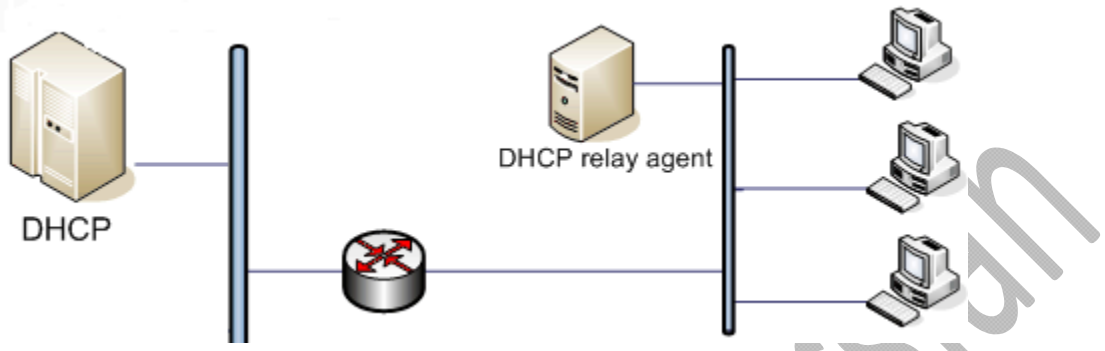
لازم به ذکر است که فایل lease فقط توسط dhcp استفاده شده و تغییر می یابد و احتیاجی به تغییرات کاربر ندارد. کاربر فقط باید فایل dhcpd.conf را مورد پیکربندی قرار دهد.

در مسیر /var/lib/dhcpd سه فایل به نام های زیر وجود دارد:

dhcpd.leases
dhcpd.leases~
dhcpd.leases.rpmsave

آمار آدرس های واگذار شده به کلاینتها در فایل dhcpd.leases قرار دارد. حال اگر سیستم به هر نحوی crash کند و این فایل از می رود و در up مجدد سیستم ، dhcp نمی تواند آمار IP های واگذار شده را بدست بیاورد چون دیتابیس آن پاک شده است. لذا خود سرویس در یک زمان بندی منظم می آید از این فایل یک کپی گرفته و در فایل dhcpd.leases~ ذخیره می کند تا در صورت بروز مشکل و از بین رفتن فایل اصلی بتواند به دیتابیس خود دسترسی داشته باشد.

مفهومDHCP Relay Agent



relay agent درخواست هایی که از زیر شبکه های بدون dhcp ارسال می گردد را برای dhcp سرورهای دیگر که مشخص شده اند بازپخش میکند.

دلیل استفاده از DHCP Relay Agent اینست که مسیریاب، Broadcast را از خود عبور نمیدهد وکلاینت ها برای گرفتن IP باید مراحل Dora را طی کنند و چون پکتهای Dora از نوع Broadcast است روترها آنها را از خود رد نمی کنند، پس برای حل این مشکل ما از یک DHCP Relay Agent استفاده میکنیم .

DHCP Relay Agent بدین ترتیب عمل میکند، ابتدا پیغام هایی که به صورت Broadcast ارسال میشود و تقاضای IP میکنند را جمع آوری کرده و سپس خود بصورت unicast با DHCP سرور درخواست مورد نظر را مطرح می کند سپس جواب گرفته شده را برای client ارسال میکند. در واقع نقش میانجی را ایفا می کند.

برای edit فایل پیکربندی به مسیر زیر رفته و فایل dhcrelay را باز می کنیم:

 # vi /etc/sysconfig/dhcrelay

مهمترین قسمتهای این فایل گزینه های زیر می باشند:

INTERFACE:"eth1"

در اینجا مشخص می کنیم این کارت شبکه ترافیک Broadcast را دریافت کند.

DHCPSERVER:"192.168.7.1"

و در این خط آدرس سرور dhcp را وارد می کنیم. با تغییر این دو خط ، این سرویس فعال می شود. حال باید یک استاتیک routeبنویسم و مشخص کنیم Relay Agent پکتها را به چه IP ای بدهد و در انتها سرویس را ریست کنیم:

```
# route add -net 10.0.0.0 netmask 255.0.0.0 gw 192.168.1.254
# service dhcrelay start
```

در این خط route مشخص کردیم پکتهای Broadcast درخواست IP را به ورودی روتر بفرستد. اگر سرویس net را ریست کنیم ، خطی که به عنوان route دستی نوشتیم پاک می شود و route پیش فرض سیستم load می شود.

تنظیمات سرور و کلاینت dhcp

در **سرور** باید یک کارت شبکه را بعنوان سرویس دهنده DHCP انتخاب کنیم. حتی اگر چند کارت شبکه داشته باشیم تنها یکی از آنها باید برای سرویس DHCP استفاده شود و روی آن یک IP بصورت استاتیک(ثابت) تنظیم گردد. هر کارت شبکه در مسیر /etc/sysconfig/network-scripts/ یک فایل به نام ifcfg-ethx دارد که x شماره کارت شبکه می باشد و از عدد صفر شروع می شود. برای کانفیگ اولین کارت شبکه با نام eth0 از دستور زیر استفاده می کنیم:

#vi /etc/sysconfig/network-scripts/ifcfg-eth0

برای دادن یک آدرس IP ثابت به کارت شبکه باید متغیر های زیر را با مقادیر مناسب تکمیل کنیم. این عبارات در زیر توضح داده شده است:

IPADDR: همان آدرس IP ثابتی است که می خواهیم به کارت شبکه اختصاص دهیم.

NETMASK: همان subnet mask است.

NETWORK: معرف آدرس شروع شبکه است و با توجه به range مشخص می گرددمثل آدرس10.10.10.0 .

سه عبارت بالا باید با مقادیر صحیح مقدار دهی شوند اما تغیراتی هم باید در فایل اعمال شود:

- ابتدا باید مقدار عبارت BOOTPROTO به static تغییر پیدا کند.
- سپس عبارت ONBOOT باید به yes تغییر پیدا کند.

تنظمات بخش **کلاینت** بسیار ساده است و حتی می توان گفت تنظیم خاص هم ندارد و تنها باید در فایل ifcfg-eth0در هر کلاینت مقدار BOOTPROTO برابر با dhcp باشد.

BOOTPROTO=dhcp

اگر بخواهیم به صورت موقت تیک گزینه Obtain کارت شبکه را بزنیم از دستور زیر استفاده می کنیم:

 # dhclient eth0

و اگر از دستور زیر استفاده کنیم به محض اینتر از dhcp آدرس IP میگیرد:

 # ifconfig eth0 dynamic

به محض تایپ این دستور IP ای که به صورت استاتیک تنظیم شده از بین می رود و با ریبوت سیستم دوباره به شرایط قبل باز می گردیم.

اگر بخواهیم در ویندوز با کامند ، کارت شبکه را وادار کنیم که از dhcp سرور IP بگیرد از این دستور استفاده می کنیم :

netsh int ip set "local area connection" dhcp

حال اگر بخواهیم زمان 50% درصد را شبیه سازی (تمدید) کنیم دوباره از دستور dhclient استفاده کرده و اگر بخواهیم زمان 87/5% شبیه سازی کنیم (درخواست IP جدید) کافی است به انتهای آن r– را اضافه کنیم. دستور dhclient همان درخواست IP را ارسال می کند:

 # dhclient -r eth0

معادل ویندوزی آن دستورات ipconfig /renew و ipconfig /release می باشد.

اگر هم بخواهیم به صورت دائمی تیک Obtain کارت شبکه را فعال کنیم تا همیشه از سرور dhcp درخواست آدرس کند باید فایل مربوط به اینترفیس شبکه را باز کرده و خطوط زیر را اصلاح کنیم :

 # vi /etc/sysconfig/network-script/ifcfg-eth0

BOOTOROTO=dhcp
PEERDNS=yes

اگر عبارت PEERDNS برابر yes نباشد تنظیمات dns ای که از dhcp دریافت می شود در این فایل ذخیره نشده و به سیستم عامل اعمال نمی شود.

و می توانیم با تغییر در فایل زیر مشخص کنیم کدام اینترفیس بتواند از سرور IP بگیرد:

 # vi /etc/sysconfig/dhcpd

DHCPDARGS=eth0;
DHCPD_INTERFACE="eth0";

ساختن یک فایل نمونه

در انتها یک نمونه فایل پیکربندی سرور dhcp آمده است :

```
# Define global values that apply to all systems.
max−lease−time 604800;
default−lease−time 86400;
option subnet−mask 255.255.255.0;
option domain "foobirds.org";
option domain−name−servers 172.16.55.1, 172.16.5.1;
option pop−server 172.16.18.1;
# Define the dynamic address range for the subnet.
subnet 172.16.55.0 netmask 255.255.255.0 {
option routers 172.16.55.1;
option broadcast−address 172.16.55.255;
range 172.16.55.64 172.16.55.192;
range 172.16.55.200 172.16.55.250;
}
# Use host statements for clients that get static addresses
group {
use−host−decl−names true;
host kestrel {

hardware ethernet 00      c7:aa:a8:04;
fixed−address 172.16.55.4;
}
host ibis {
hardware ethernet 00:00:c0:a1:5e:10;
fixed−address 172.16.55.16;
}
```

فصل هشتم

بررسی Xinetd

در لینوکسxinetd

اگر init نباشد یا درست کار نکند می توان گفت که عملا سیستم غیرعملیاتی است.چون هر نرم افزاری بخواهد کار کند init هم به طریقی درگیر می شود.اما وظیفه init نیست که ping کند یا telnet راه اندازی کند بلکه وظیفه اش این است که یک سری نرم افزار را وارد چارت سازمانی سیستم عامل کند:

اسم مدیر شبکه در لینوکس xinetd یا super server است. نرم افزارهای تحت سیستم عامل Linux به دو صورت زیر به کاربران تحت شبکه سرویس می دهند:

- به صورت مستقل (Standalone)
- تحت نظارت و کنترل پروسس xinetd

بسیاری از سرویس های شبکه ای از جمله telnet تحت نظارت و سرپرستی پروسسی به نام xinetd که اصطلاحا" Super Server خوانده می شود قرار دارند.

سوپر سرور یک سرویس قدرتمند است که سرویس های دیگر می توانند در قالب آن کار کنند. شما باید در سازمانتان تصمیم بگیرید که کدام یک از نرم افزارهایتان زیر نظر init باشد و کدام یک زیر نظر xinetd کار کند.

Xinetd فقط مدیریت درخواست عای یک سرویس را به عهده می گیرئ ، مدیریت کارهای یک سرویس یه عهده خود آن است .اگر نرم افزاری به تنهایی در حافظه قرار بگیرد در مد Standalone قرار دارد.

xinetd نرم افزاری است که می تواند نرم افزارهای سرویس دهنده شبکه را مدیریت کند. مثلا وقتی در لینوکس آپاچی سرور را راه اندازی کردیم در قسمت از پارامترهای config می توانیم مشخص کنیم که آیا به صورت standalone اجرا شود یا تحت نظر xinetd اجرا شود.

هرکس از هر جای دنیا به هر سرویسی وصل شود سرویس دهنده باید به پورت گوش کند. نرم افزار stand alone خودش گوش می کند اما نرم افزار هایی که فرزند xinetd هستند xinetd به جایشان به پورت گوش می دهد ودر مواقع ضروری آنها را آگاه می کند.همه نرم افزار های جدی در همه سیستم عامل ها فایل پیکربندی دارند. سرویس ها زمانی که بالا می آیند فایلconf مربوط به خود را می خوانند تا بدانند چه کاری باید انجام دهند.

سرویس xinetd که در بعضی از گونه های Linux و یا Unix با نام inetd شناخته می شود در زمان فعال شدن، به فایل ها و دایرکتوری زیر مراجعه نموده و با آنالیز نمودن اطلاعاتی که به دست می آورد آماده سرویس دهی می شود:

etc/xinetd.conf/

etc/xinetd.d/

فایل متنیxinetd.conf که همانند اکثر فایل های پیکربندی تحت دایرکتوری etc میباشد، شامل اطلاعات کلی برای سرویس دهی تحت شبکه بوده و تحت دایرکتوی/etc/xinetd.d نیز تعداد زیادی فایل متنی قرار داشته و به ازای هر سرویس (مثلا" telnet) میتوان فایل متنی با همان نام مشاهده نمود. مدیر سیستم با تغییر دادن در فایل های فوق می تواند کنترل بیشتری را بر روی سرویس دهی داشته باشد. در لینوکس اکثر فایل های پیکربندی text base هستند و با ویرایشگرهای متن مانند Vim قابل تغییر هستند ولی در ویندوز این نوع فایل ها داخل رجیستری قرار دارند و با regedit باید آنها را مورد تغییر قرار داد .

shell مانند یگ گارسون در رستوران است ، پشت صحنه عوامل بسیار زیادی فعالیت می کنند که آن رستوران میتواند سرویس بدهد به این عوامل Daemon می گویند که یک معنی آن پشت پرده است .xinetd هم خودش هیچ وقت شخصا برای سرویس دادن عمل نمی کند مانند سرپرست راننده هاست که اگر شخصی تماس بگیرد و ماشین می خواهد سرپرست هم برایش ماشین می فرستد.

در فایل xinetd.conf داخل {} مانند برنامه های c پارامترها ومقادیرشان تعریف شده اند که xinetd از روی اینها موقع بالا آمدن می فهمد که به چند نفر باید سرویس بدهد و مثلا صد نفر می توانند telnet کنند یا 50 نفر ftp کنند. در واقع xinetd فایلی دارد که از روی آن می تواند بخواند که چه کسی می تواند چه کسی نمی تواند و چه کسی نباید بتواند!.

همان طور که موقع ورود نگهبان از شما کارت تشخیص هویت می خواهد xinetd نیز برای هر ارتباطی identication طلب می کند و مجموعه این اطلاعات را نیز نگه داری می کند.

اگر کسی telnet کند xinetd هاستش (ip) را نگه می دارد و اینکه با چه کسی کار دارد. در نگهبانی نیز شماره شناسایی مراجعه کننده و شماره کارمندی کسی را که با او کار دارد ثبت می شود.حال اگر شخص شماره شناسایی نداشت log out failure اتفاق می افتد ولی سیستم هاست را ثبت می کند و همه این

کارها توسط xinetd انجام می شود. به همین دلیل است که اگر به سایتی حمله کنید شناسایی می شوید ؛ userID ندارید ولی ip شما ثبت می شود.در لینوکس های سری 6 به بعد خانواده ردهت سرویس xinetd نصب نیست بلکه باید نصب و پیکربندی شود.

مزایا و معایب xinetd

به طور مثال اگر روی سروری 100 سرویس داشته باشیم و هر سرویس به تنهایی بخواهد در حافظه قرار بگیرد مطمئنا آن سرور در سرویس دهی با مشکل جدی مواجه خواهد شد .

Xinetd سرویس هایی که ضروری نیستند را از حافظه خارج کرده و مدیریت درخواست های آنها را به عهده گرفته و به جای آنها در حافظه قرار می گیرد. و سرویس مورد نظر را به حالت standby می برد . به محض اینکه درخواستی برای آن سرویس برسد xinetd آن را فراخوانی کرده و سرویسش مورد نظر را وارد چرخه سرویس دهی می کند. لازم به ذکر است تمام سرویس ها را نمی توان تحت xinetd راه اندازی کرد ، مثلا سرویس های پر تراکنشی مثل dhcp و یا dns را باید به صورت standalone راه اندازی کرد. اما سرویس های مثل ssh و ftp و یا telnet که listen دارند را می توان تحت xinetd اداره کرد. سرویس های جدی و پرتراکنش ، حتما باید به صورت standalone راه اندازی شود.

با xinetd یک لایه امنیتی به سیستم اضافه می شود و دست admin برایاعمال محدودیت در سرویس ها باز می شود. اگر سرویسی که دارای تراکنش بالائی باشد را تحت xinetd قرار دهیم مطمئنا خود xinetd با مشکل جدی مواجه خواهد شد . کانفیگ xinetd مربوط به سرویس خاصی نسیت. این فایل در مسیر /etc/xinetd.conf قرار دارد و در /etc/xinetd.d/ هم فایل های کانفیگ سرویس هایی که تحت xinetd اداره می شوند و می خواهند یکسری خواص گلوبال را به ارث نبرند قرار دارد.

بررسی آپشن های موجود در یک فایل xinetd

در زیر تعدادی از آپشن هایی که در یک فایل xinetd میتوان نوشت را توضیح می دهیم جهت اطلاع بیشتر به man xinetd رجوع فرمائید. مقادیر ثبت شده در جلوی هر آپشن فرضی می باشد.

instances =60

این آپشن مشخص می کند سرویس به چند درخواست هم زمان پاسخ می دهد. instances =60 یعنی به بیش تر از 60 نفر را سرویس نمی دهد حالا اگر 45 نفر telnet کنند و 15 نفر ftp .نفر 61ام که تلاش کند وصل شود با پیغام connection refused مواجه می شود.این عدد را در یک سازمان مثلا برابر 600 می گذاریم و برای شبکه های خانگی و کوچک 2 یا 3.پس این عدد در سازمان ها برای سرویس هایی که stand alone نیستند باید عوض شوند.

log_type =SYSLOGautjpriv

این خط مشخص می کند تمام لاگها را تحویل syslog سرور بدهد و log هایی را ثبت می کند که از نوع احراض هویت باشند.

log_on_success =HOSTPID

اگر کلاینت موفق شود از سرویس مورد نظر استفاده کند از id پروسس های آن log گرفته می شود.

log_on_failure =HOST

اگر کلاینت موفق به سرویس گرفتن نشد از IP یا نام سیستم log برداری می کند.

cps =25 30

در جلوی این آپشن دو عدد وجود دارد اولین عدد مشخص می کند چند درخواست هم زمان را در ثانیه بپذیرد و دومین عدد هم بیان می کند اگر ارتباط برقرار نشد چند ثانیه بعد مجدا retry کند. عدد دوم همان زمان time out است در این خط مشخص کرده ایم در ثانیه 25 درخواست بیشتر نمی توانند به سیستم

وصل شوند حال اگر درخواست 26 رسید باید 30 ثانیه صبر کند تا اتصال برقرار شود. این کار برای جلوگیری از حملات DDOS می باشد. عدد cps باید متناسب با instances باشد. در مورد cps یا connection per second باید توجه شود که اگر 10000 کاربر ظرفیت داشته باشیم و همه با هم وصل شوند مثل این است که 10000 نفر یک دفعه وارد یک اتاق شوند پس منطقی این است که هر دفعه (ثانیه) مثلا 25 نفر وارد شوند .اکثر سرویس های موجود در دایرکتوری xinetd (xinetd.d) تحت مدیریت xinetd اجرا می شوند مانند telnet ، ftp و chargen و بقیه سرویس ها نیز شبیه این سه سرویس کار می کنند. به ازای هر سرویسی در دایرکتوری xinetd.d یک فایل کانفیگ داریم.

disable =yes or no

اگر yes باشد سرویس غیر فعال می شود و اگر no باشد بر عکس آن را اجرا می کند.

wait =yes or no

این آپشن مشخص می کند نرم افزار مربوطه Multi Thread باشد یا Single Thread . اگر yes باشد باید ابتدا به درخواست رسیده پاسخ دهد سپس به درخواست بعدی رسیدگی کند، اما اگر no باشدمی تواند هم زمان به چند درخواست پاسخ دهد.

server =/usr/sbin/sshd

مقابل این گزینه باید آدرس اسکریپت باینری فایل اجرایی سرویس مورد نظر را وارد کنیم. این فایل حتما باید باینری باشد یعنی اگر یک سرویس نوشتید باید فایل آن را تبدیل به باینری کنید.

user =root

مشخص میکند اسکریپت اجرایی سرویس توسط چه یوزری اجرا می شود.

sock_type =stream or dgram

اگر مقدار این خط بر روی stream تنظیم شود یعنی پکتها از نوع tcp می باشند و ار بر روی dgram باشد یعنی پکتهای این سرویس از نو udp خواهند بود.

server_args =/etc/ssh/sshd_conf

هر سرویسی یکسری ارگومان دارد که ممکن است xinetd آنها را نشناسد. مثلا در یک سرویس می توانیم کاری کنیم که اگر یوزر یک دایرکتوری یا فایل ایجاد کرد با پرمیژن خاصی ذخیره شود . این می تواند قابلیت داخلی یک سرویس باشد ولی ممکن است در xinetd وجود نداشته باشد. اگر بخواهیم این قابلیت ها را به xinetd معرفی کنیم باید در جلوی این آپشن آدرس مسیر فایل کانفیگ سرویس مورد نظر را وارد می کنیم چون معمولا ارگومانها و فیچرها در فایل کانفیگ یک سرویس تعریف می شوند.

bind =192.168.1.1

این آپشن معلوم میکند این سرویس درخواست های رسیده از کدام کارت شبکه را پاسخ دهد.

access_time =19:12-19:16

در این خط مشخص میکنیم کاربران در چه زمانهای بتوانند از این سرویس استفاده کنند.

نمونه ای از فایل ایجاد شده در Xinetd برای سرویس Vsftpd

سرویس vsftpd به صورت standalone کار می کند حال اگر بخواهیم این سرویس زیر نظر xinetd اداره شود باید گزینه ???=listen را در فایل کانفیگ آن برابر YES شده و در زیر دایرکتوری xinetd یک فایل کانفیگ بسازیم. در زیر نمونه ای از فایل آورده شده است :

```
Servicevsftp

   {

socket_type        =stream      یعنی پکتها از نوع tcp می باشند
user               =root  مشخص میکند اسکریپت اجرایی سرویس توسط چه یوزری اجرا می شود
server             =/usr/sbin/vsftpd  آدرس اسکریپت باینری فایل اجرایی سرویس مورد نظر
server_args        =/etc/vsftod/vsftpd.conf  آدرس فایل کانفیگ جهت شناختن آرگومانها
nice               =10
disable            =no   نشان می دهد سرویس فعال است
flags              =ipv4

}
```

ضمیمه یک :

اضافه کردن Repository به لینوکس

اضافه کردن Repository به لینوکس

1- اضافه کردن repository اینترنتی

مواقعی پیش می آید نیاز به نرم افزاری دارید ولی نام آن را نمی دانید مثل نرم افزارهای پلیر یا pdf .برای پیدا کردن نرم افزار مورد نیاز می توانید از yum و مخازن اینترنتی لینوکس استفاده کنید به طور مثال برای پیدا کردن نرم افزار نمایش pdf از دستور جستجو yum به صورت زیر استفاده می کنیم .

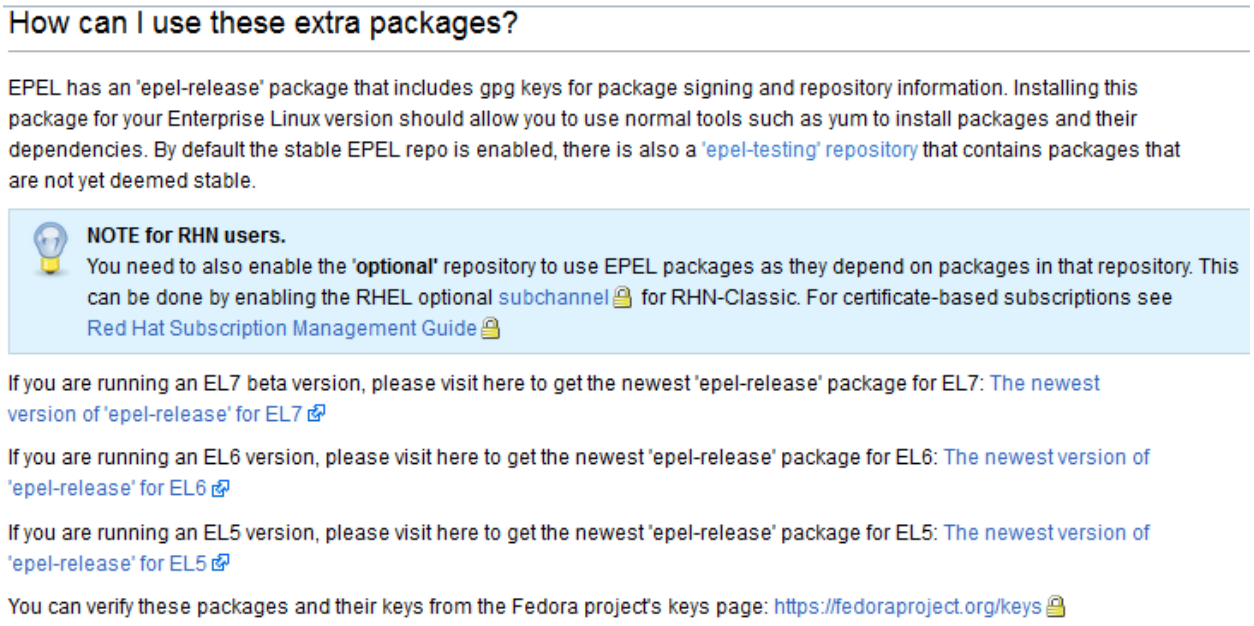
 # yum search pdf

با تایپ این دستور ، yum ابتدا دیتابیس خود را که متشکل از لیست rpm مخازن مشخص شده است را به روز می کند و سپس لیست تمام پکیج هایی که کلمه pdf در آن وجود دارد را نمایش می دهد.
مخازن اینترنتی مکانهای ذخیره سازی هستند که نرم افزارهای لینوکس در آنجا نگهداری می شوند. لیست این مخازن در سیستم عامل قابل تغییر است.

بر روی اینترنت مخازن زیادی وجود دارد ولی نباید همه و یا تعداد زیادی از آنها را بر روی سیستم add کرد ، چون هم می تواند خطر امنیتی ایجاد کرده و هم عملکرد پروسه yum را به شدت سنگین می کند .
از مهمترین و بهترین مخازن لینوکسی ، می توان به مخزن base و epel اشاره کرد.آدرس مخزن base بر روی تمام توزیع های مبتنی بر ردهت وجود دارد ولی epel به صورت پیش فرض بر روی سیستم نصب نیست و باید آن را به صورت دستی نصب کنیم . برای نصب آن ابتدا به آدرس زیر می رویم :

https://fedoraproject.org/wiki/EPEL

در پائین صفحه باز شده قسمتی به نام how can I use these extra packages? وجود دارد .



How can I use these extra packages?

EPEL has an 'epel-release' package that includes gpg keys for package signing and repository information. Installing this package for your Enterprise Linux version should allow you to use normal tools such as yum to install packages and their dependencies. By default the stable EPEL repo is enabled, there is also a 'epel-testing' repository that contains packages that are not yet deemed stable.

NOTE for RHN users.
You need to also enable the '**optional**' repository to use EPEL packages as they depend on packages in that repository. This can be done by enabling the RHEL optional subchannel for RHN-Classic. For certificate-based subscriptions see Red Hat Subscription Management Guide

If you are running an EL7 beta version, please visit here to get the newest 'epel-release' package for EL7: The newest version of 'epel-release' for EL7

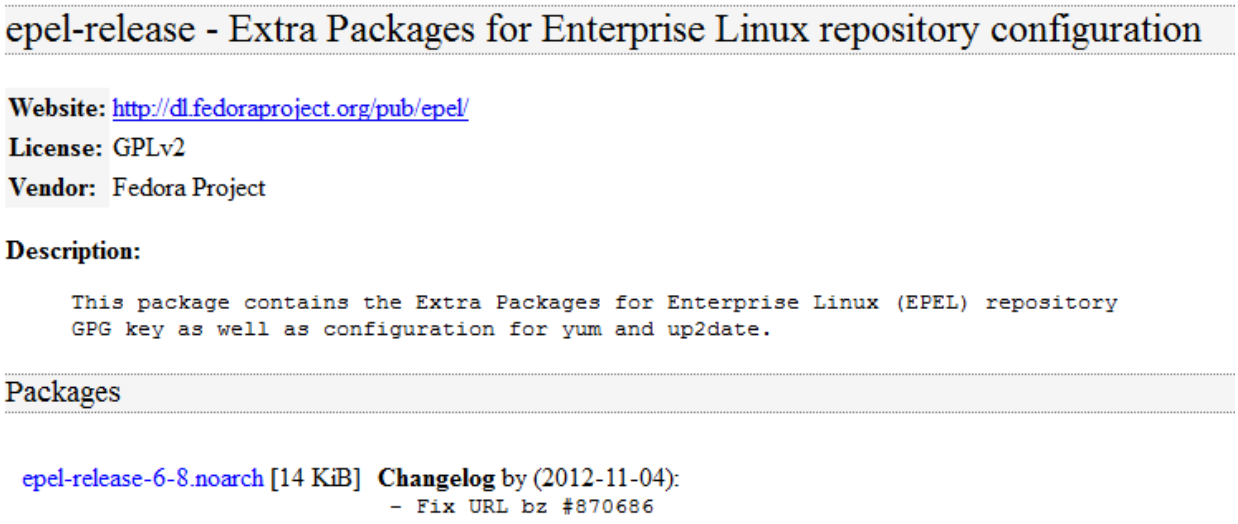
If you are running an EL6 version, please visit here to get the newest 'epel-release' package for EL6: The newest version of 'epel-release' for EL6

If you are running an EL5 version, please visit here to get the newest 'epel-release' package for EL5: The newest version of 'epel-release' for EL5

You can verify these packages and their keys from the Fedora project's keys page: https://fedoraproject.org/keys

با توجه به سری سیستم عامل بر روی یکی از لینکها کلیک کرده و وارد صفحه بعد می شویم :

http://mirror.switch.ch/ftp/mirror/epel/6/i386/repoview/epel-release.html



epel-release - Extra Packages for Enterprise Linux repository configuration

Website: http://dl.fedoraproject.org/pub/epel/
License: GPLv2
Vendor: Fedora Project

Description:

```
This package contains the Extra Packages for Enterprise Linux (EPEL) repository
GPG key as well as configuration for yum and up2date.
```

Packages

epel-release-6-8.noarch [14 KiB] **Changelog** by (2012-11-04):
- Fix URL bz #870686

با ورود به صفحه جدید به قسمت Packages و بر روی لینک دانلود رفته آدرس آن را کپی کرده و سپس در صفحه terminal توسط دستور yumو یا rpm آن را نصب می کنیم .

 # rpm -Uvh http://mirror.switch.ch/ftp/mirror/epel/6/i386/epel-release-6-8.noarch.rpm

سیستم به صورت اتوماتیک پکیج مورد نظر را دانلود و نصب می کند. لیست ریپازیتوری ها در مسیر /etc/yum.repo.d/قرار دارند که پسوندشان repo. می باشد.

داخل این پکیج 5 فایل قرار دارد که با دستور زیر قابل مشاهده است :

 # rpm -qlp epel-release-6-8.noarch.rpm

و با دستور زیر می توانیم اطلاعات کاملی در مورد این پکیج بدست آوریم :

 # rpm -qip epel-release-6-8.noarch.rpm

2- اضافه کردن repository محلی

در بعضی از سرورها دسترسی به اینترنت برای نصب پکیجها و یا هر بسته ای امکان پذیر نیست و همانطور که می دانید نصب rpm بدون نصب نیازمندی ها بسیار مشکل است. لذا باید کاری کنیم yum بدون وابستگی به اینترنت و از روی یک مخزن محلی ، نرم افزار مورد نظر را به همراه نیازمندی های آن نصب کند. برای این کار یا باید تمام پکیج های مورد نیاز را جمع آوری کرده و در قسمتی از هارد ذخیره کنیم و یا تمامی محتویات dvd نصب سیستم عامل را در مکانی از هارد کپی کرده و طبق روش زیر ابتدا به مسیر /etc/yum.repos.d می رویم :

 # mkdir -p /opt/dvd/pkg
 # cp -rf /media/dvdrm-name /opt/dvd
 # cp -rf /media/dvdrm-name /RPM-GPG-KEY* /opt/dvd/gpg

 # cd /etc/yum.repo.d/
 # mv CentOS_Media.repo dvd.repo

 # vi dvd.repo

```
[dvd]
name=dvd
baseurl=file:///opt/dvd/
gpgcheck=1
enabled=1
gpgkey=file:///opt/dvd/gpg/RPM-GPG-KEY-CentOS6.0
```

در صورت استفاده در شبکه ، می تواین تنظیمات زیر را در سیستم های مورد نیاز اعمال کنیم:

```
[dvd]
name=dvd
baseurl=http://ip-address/ opt/dvd/
gpgcheck=1
enabled=1
gpgkey=http://ip-address/pkg/gpg/RPM-GPG-KEY-fedora
```

تغییرات را ذخیره کرده و فایل را می بندیم. سپس باید لیست repoهای سرویس yum را به روز کنیم تا منبع جدید را بشناسد ، لذا با دستور زیر دیتابیس yum را به روز کرده و repo جدید را وارد cache سرویس yum می کند :

```
# yum   repolist
# yum   clean   metadata
```

سپس با دستور زیر هر پکیجی را که بخواهید می توانید با دستور yum بدون نیاز به اتصال به اینترنت نصب کنید.

```
# yum  -y  install  bind  --disablerepo=*  --enablerepo=dvd
```

منابع :

راهنمایی های استاد مهدوی فر

مطالب متفرقه منتشر شده در سایتهای فارسی زبان

مطالب دوره های RHCE و LPIc2

سایت centos .org

ایمیل نگارنده skywan13@yahoo.com