شناخت کاربردی Iptable

« پروژه آزمایشگاه مدار منطقی »

استاد راهنما:

مهندس مجید اسدی شهمیرزادی

تهیه کنندگان:

مریم فلسفی ، لیلا ساروخانی، مهناز علی نژاد

Iptable شناخت کاربردی

Iptable Tutorial

شناخت کاربردی Iptable

شما باید نسخه ای از این جواز را در این آیین نامه بیابید که تحت بخش جواز عمومی GNU آمده است . در غیر این صورت با موسسه به آدرس زیر تماس بگیرید .

اهداهای مربوطه

ابتدا می‌خواهم این سند را به دوست دختر خود Ninel اهدا کنم . او بیش از آنچه که تصور می‌کردم حامی من بود . من امیدوارم که بتوانم با این اهدا شما را نیز خوشحال کنم . دوم آنکه مایل هستم این اثر را به تمام موسسه دهندگان linux تقدیم کنم . این افراد سیستم عامل جالب را طراحی کرده اند .

شناخت کاربردی Iptable

فهرست مطالب:

شناخت کاربردی Iptable

در مورد محقق

من دارای کامپیوترهای فراوان هستم . من یک کامپیوتر LAN دارم و تمام ماشین ها بر اینترنت وصل هستند ولی بایدLAN ایمن حفظ شود . iptables جدید یک نسخه ارتقا یافته از ipchain می‌توانید یک شبکه ایمن بسازید و این امر با حذف بسته های آتی عملی است . با این وجود FTP انفعالی و یا DGE در TRC مسائلی به دنبال دارند . مسائل دندانه ای کردن در کد iptables در سطح آغازین حل نشده اند . امروزه هر کس را که از آنها استفاده می‌کند به سوی نسخه تولید کامل راهنمایی می‌کنم و ipfwadm را ارتقا دارم . این که فعلی می‌تواند در صورت نیاز استفاده شود .

چگونگی خواندن :

این سند به درستی نوشته شده است . بنابراین می‌توانید به نکات جالب iptables پی ببرید . این به معنای اطلاعات خاصی در مورد اشکالات ایمنی خاص در iptables یا Netfilter نیست اگر اشکالات خاص و رفتارهایی را در iptables و هر زیر مولف یافتید با لیت پست Netfilter تماس بگیرید و سپس خواهید توانست اشکالات واقعی را بشناسید و حل کنید اشکالات ایمنی واقعی در iptable و Netfilter فراوان هستند و یک یا دو اشکال در

شناخت کاربردی Iptable

آن واحد گزارش شده است . آنها در صفوف اصلی Netfilter آمده اند باید اطلاعات را در مورد موضوع ارائه کنند .

می‌توان گفت مجموعه قوانین موجود در این سند پیرامون اشکالات واخل Netfilter نمی‌باشند . هدف اصلی توصیف چگونگی نصب قوانین در یک حالت ساده است به طوری که بتوان مسائل داخل کرد به عنوان مثال این سند نشان نمی‌دهد که چگونه HTTP PORT به دلایل مفاد بسته می‌شود آنطور که Apache در نسخه ۱۲-۲-۱ گزارش کرد. این سند برای هر کس قابل استفاده است و می‌تواند نشان دهد که چگونه با iptable می‌توان کار را آغاز کرد ولی در آن واحد یک روند تکامل را نیز پیچیده است . این جا اهداف و هماهنگی ها در patch - matic بیان نمی‌شوند . ارتقا نیاز است . اگر اطلاعات بیشتر در مورد این ارتقا لازم دارید باید patch - matic و دیگر اسناد را در صفوف اصلی Netfilter مطالعه کنید .

شرط لازم :

این سند به معلومات در مورد linux linux ، دست نوشته لایه ای ، و چگونگی کامپایل کردن kernel و بخش درونی آن نیاز دارد .

من سعی کرده ام تا تمام شرایط را قبل از توصیف این سند بررسی کنم ولی نمی‌توان تمام اطلاعات قبلی را مطرح کرد .

شناخت کاربردی Iptable

نهادهای مورد استفاده در این سند : این نهادهای در سند زمان دستورات ، فایل و دیگر اطلاعات خاص استفاده می‌شوند .

- گلچین کد و خروجی دستور این چنین است و تمام خروجی ها در فونت پهنای ثابت و دستور مکتوب کاربر به صورت bold هستند:
- تمام دستورات و نام های برنامه در این سند در حالت bold هستند:
- تمام item سیستم مانند سخت افزار و مولف درونی kernel و item سیستم انژای مانند روابط حلقه ای که در حالت italic آمده اند
- خروجی کامپیوتر به شکل this way در متن ظاهر می‌شوند .
- نام فایل و مسیر در سیستم فایل به شکل ...

شناخت کاربردی Iptable

فصل ۱ : مقدمه

۱-۱ : چرا این سند نوشته شد :

من فضای خالی بزرگی را در HOW TOS یافتم و در آنجا اطلاعات در مورد iptable و نقش Netfilter در linux 1/4x kernel جدید کافی نبود . در بین آنها می‌خواهم به سوالاتی پاسخ دهم که در مورد احتمال جدید مانند مطابق حالت بوده اند . بخش اعظم آن به صورت فایل rc... است که در دست نوشته etc.... آمده است . این فایل اصولاً بر اساس ارتقای idowt. است .

یک دست نوشته کوچک در مورد اجرای برنامه و راه اندازی سیستم به صورت rc.... در دسترس است .

۱-۲ : چگونگی نوشتن :

من با مارک باکر و دیگران در سیستم Netfilter مشورت کردم . بسیاری از آنها در این اثر مستند به من کمک نموده اند و در این جا سعی کرده ام از سایت Fr.zentux.Net خود استفاده کنم . این سند فرآیند نصب را مرحله به مرحله نشان می‌دهد و در مورد طرح iptables است . من مثالهای فایل re.Firewall و مثالهای یادگیری استفاده از iptables را نیز ارائه کرده ام در این جا باید تابع زنجیره های اصولی باشید و آنها را در کنار هم قرار دهید

. به این ترتیب سند می‌تواند آماده شود و روش منطقی تر را نشان دهد . هر زمان که این درک مشکل باشد به این سند مراجعه کنید .

۳-۱ : اصطلاحات مورد استفاده در سند :

این سند دارای اصطلاحاتی است که به تشریح نیاز دارند قبل از اینکه آنرا مطالعه کنید . در این بخش توصیف آنها و چگونگی انتخاب آنها در سند آمده است .

ترجمه آدرس شبکه مقصد - DNAT - روش ترجمه آدرس IP مقصد در بسته اشتاره دارد و یا تغییر آنرا نشان می‌دهد . این با SNAT استفاده شد و به کاربرها در به اشتراک گذاری آدرس IP اینترنت کمک می‌کند و خدمات سرویس دهنده را ارائه می‌نماید. این روند با تعیین port متفاوت با آدرس IP عملی است و مسیر linux را برای ارسال ترافیک نشان می‌دهد .

جریان- این اصطلاح به روابط ارسال و دریافت بسته ها در رابطه با روش جدید اشاره دارد . اصولاً این اصطلاح برای یک نوع ارتباط به کار می‌رود که دو یا چند بسته را در دو جهت ارسال می‌کند. در TCP این به معنای ارتباطی است که یک SYN را ارسال می‌کند و سپس با SYN/ACK جواب می‌دهد ولی ارتباط ارسالی SYN و جواب میزبان ICMP نیز مهم است . به عبارت دیگر از این اصطلاح به خوبی استفاده نمی‌کنم .

شناخت کاربردی Iptable

SNAT - ترجمه آدرس شبکه منبع این اصطلاح به روش ترجمه آدرس منبع به دیگری اشاره دارد . این برای چندین میزبان وجه اشتراک گذاری آدرس IP اینترنت به کار میرود زیرا نقص آدرس IP در IP 74 مطرح است (IP 74 این را حل می‌کند)

حالت - این اصطلاح بحالت بسته طبق پروتکل کنترل انتقال RFC 793 اشاره دارد و در Netfilter/iptable به کار می‌رود . توجه کنید که حالات درونی و خارجی تابع ویژگی RFC 793 است . دلیل اصلی آن است که Netfilter باید چندمین فرضیه در مورد ارتباط و بسته ها ارائه کند . فضای کاربر - این اصطلاح به هر چیز که در خارج kernel روی دهد اشاره دارد به عنوان مثال تحریک iptable-h در خارج kernel است ولی iptable - A Forward p - tcp jAce Ept در kernel است زیرا قانون جدید به مجموعه قوانین اضافه می‌شود .

فضای kernel - این نکاتش برخلاف فضای کاربر است این اقدامات را در kernel نشان می‌دهد و همه در خارج آن .

قلمرو کاربر - به فضای کاربر مراجعه کنید .

شناخت کاربردی Iptable

فصل ۲ : آماده سازی : این فصل توصیف درک نقش Netfilter و iptable در linux است . در این جا باید کار با آزمایش آغاز شود و نصب عملی شود . با زمان کافی می‌توانید آنرا دقیقاً اجرا کنید .

۲-۱ : کجا به iptable دست یابیم :

بسته فضای کاربر iptables می‌تواند از آدرس زیر down load شود :

http.....

این بسته iptable تسهیلات خاص فضای kernel را نشان می‌دهد که در طی تولید سیستم طراحی شده اند مراحل لازم به تفصیل بررسی خواهند شد .

۲-۲ :نصب kernel :

برای اجرای اساس iptable باید گزینه ها در kernel راه اندازی شوند و سیستم با دستورات مربوط آماده شود .

CONFIG- PAKET- این گزینه تولید برنامه را برای ارتباط کاری سیستم با ابزار شبکه نشان می‌دهد . نمونه ها به صورت tcpdump و snort است .

شناخت کاربردی Iptable

CONFIG- PAKET- یک نیاز برای عملکرد iptable نیست بلکه دارای موارد استعمال فراوان است . در این جا باید طرح در نظر گرفته شود . اگر آنرا نخواهید حذف می‌کنید .

CONFIG - NETFILER - این گزینه در صورتی نیاز است که بخواهید از کامپیوتر به عنوان ورود به اینترنت استفاده کنید . به عبارت دیگر این روند برای هر چیز در سند نیاز است زیرا طرح اصولی هستند . البته باید ابزار درست برای رابط اضافه شوند مانند آداپتور اینترنت ، ppp و رابط SLIP . این موارد اصول iptable هستند . شما نمی‌توانید چهارچوب را به kernel اضافه کنید . اگر از گزینه ها در iptable استفاده کنید باید نصب سیستم را در kernel انجام دهید . در این جا گزینه ها در kernel ۹-۴-۲ آمده اند.

CONFIG -NF - FTP - این مدل در صورتی نیاز است که بخواهید ارتباط را بر FTP برقرار کنید . جون ارتباطات FTPبه راحتی در موارد طبیعی اجرا نمی‌شوند باید از helper استفاده کنید . اگر این مدل را اضافه نکنید . نمی‌توانید FTP را به درستی اجرا کنید .

CONFIG- IP - NF - FPTAHLE - این گزینه در صورتی نیاز است که یک نوع Filter یا NAT نیاز باشد . در این جا چهارچوب شناخت

شناخت کاربردی Iptable

iptable به kernel اضافه می‌شود . بدون این امر نمی‌توانید با iptable کاری کنید .

CONFIG -IP- NF - MATCH - LIMIT - این مدل دقیقاً نیاز نیست ولی در مثالهای ....rcopir آمده است این گزینه هماهنگی LIMIT را نشان می‌دهد . باید احتمال کنترل بسته ها در نظر گرفته شود و قانون توسعه یابد . به عنوان مثال m.limit- lin 3 / lin.ld یک هماهنگی ۳ بسته را در هر دقیقه نشان می‌دهد . این مدل می‌تواند برای جلوگیری از علامت انکار خدماتی استفاده شود.

CONFIG - IP - NF - MATCH- MAC - این گزینه هماهنگی بسته را بر اساس آدرس MAC نشان می‌دهد . هر آداپتور اینترنت دارای آدرس MAC خاص خود است . ما بسته های بلوکی در آدرس MAC داریم و از بلوک خاص برای آدرس MAC استفاده می‌کنیم . ما از این گزینه در نمونه ....dc.fire استفاده نمی‌کنیم .

CONFIG - IP - NF..... - این گزینه در استفاده از تطابق MARK مفید است . به عنوان مثال اگر از MARK هدف برای بسته ها استفاده شود بسته به این که آیا بسته ها در جدول هستند یا خیر می‌توانیم هماهنگی را

شناخت کاربردی Iptable

برقرار کنیم . این گزینه هماهنگی واقعی MARK است و می‌تواند توصیف هدف واقعی MARK باشد .

CONFIG - IP - NF - MATCH - MULTIPORT - این روش به ما در تطابق بسته ها با یک سری port مقصد و منبع کمک می‌کند . این ویژگی غیرممکن است ولی هماهنگی برقرار خواهد شد .

CONFIG -IP - NF- MATCH- TOP - با این هماهنگی می‌توان بسته ها را بر اساس فیلد TOS در تطابق قرار دارد . TOP به type of service اشاره دارد TOS می‌تواند با قوانین خاص در جدول قرار گیرد و از طریق دستور ipltc اجرا شود .

CONFIG - IP- NF -MATCH- TCP MSS- این گزینه احتمال هماهنگی بسته های TCP را در فیلد MSS نشان می‌دهد .

CONFIG - IP- NF-MATCH - STALE - این یکی از بزرگترین اخبار در مقایسه با ipchain است . با این مدل می‌توان هماهنگی بسته ها را عملی ساخت . به عنوان مثال اگر شاهد ترافیک در دو بعد ارتباط TCP هستیم بسته به صورت ESTABLISHED ظاهر می‌شود . این مدل در مثال ....rc استفاده می‌شود .

شناخت کاربردی Iptable

CONFIG- IP- NF- MATCH- UNCLEAN – این مدل احتمال هماهنگی IP ، TCP، UDP ، ICMP را نشان می‌دهد و تطابق با یک نوع اعتبار ندارد . ما می‌توانیم این بسته ها را حذف کنیم ولی نمی‌دانیم که آیا این کار درست است . توجه کنید که این هماهنگی تجربی است و نمی‌توان در تمام موارد کامل باشد.

CONFIG - IP - NF - FILTER – این مدل اساس جدول Filter است و می‌تواند فیلتر IP را انجام دهد در این فیلتر می‌توان زنجیره Input , Forward و Output را یافت . این مدل در صورتی نیاز است که طراحی بسته برای ارسال و دریافت انجام شود .

CONFIG - IP - NF- TARGET - REJECT – این هدف به شما کمک می‌کند تا مشخص کنید که پیام خطای ICMP باید در پاسخ نسبت به بسته‌های بعدی ارسال شود در این جا حذف مهم است . در خاطر داشته باشید که روابط TCP برخلاف UDP ICMP , می‌توانند انکار با بسته TCPRSS باشند.

CONFIG -IP -NF- TARGET- MIRROR – این به بسته ها در برگشت به موسسه با بسته کمک می‌کند . به عنوان مثال اگر یک هدف MIRROR بر مقصد HTTP نصب شود و یا زنجیره INPUT مشخص

شناخت کاربردی Iptable

گردد دسترسی به port عملی است و باید بسته ها دوباره به صفوف home برگردند و در آنجا رویت شوند .

CONFIG - IP - NF- NAT - این مدل به ترجمه آدرس شبکه یا NAT در اشکال متفاوت کمک می‌کند این گزینه دسترسی به iptable را عملی می‌سازد . این گزینه در صورتی نیاز است که بخواهیم ارسال و تغییر port را انجام دهیم. توجه کنید که این گزینه برای Firealling و ارسال LAN نیاز نیست بلکه باید در اختیار باشد مگر اینکه بتوانیم آدرس خاص IP را برای تمام میزبان ها ارسال کنیم . بنابراین این گزینه برای دست نوشته rc.fire.... به کار می‌رود و در ثورتی شبکه مشخص خواهد شد که توانایی آدرس IP از بین برود.

CONFIG- IP - NF - TARGET - MASQUERAD - این گزینه هدف MASQUERAD را اضافه می‌کند. به عنوان مثال اگر ندانید که IP چه باید انجام دهد بهتر است که آنرا بدست آورید و از DNAT یا SNAT استفاده کنید . به عبارت دیگر اگر از DHCP , PPP یا SLIP یا دیگر روابط تعیین کننده IP استفاده می‌کنید باید از این هدف به جای SNAT استفاده شود Masquerading بارباراتر را بر کامپیوتر نسبت به NAT نشان می‌دهد ولی باید IP آدرس را از قبل نشان دهد .

شناخت کاربردی Iptable

CONFIG-IP-NF-TARGET-NEPIRECT - این هدف با prong برنامه مفید است . به جای این که بسته عبور کند باید به جعبه داخلی ببریم . به عبارت دیگر احتمال prong شفاف وجود دارد .

CONFIG-IP-NF-TARGET-LOG - این برنامه هدف LOGو نقش آنرا به iptables اضافه می‌کند . از این گزینه برای LOG بسته های خالی استفاده می‌شود و باید دید در بسته چه اتفاقی می‌افتد این برای و شکای زدایی دست نوشته که آنرا می‌نویسیم نیاز است .

CONFIG- IP- NF- TARGET- TCPMSS - این گزینه می‌تواند برای شمارش ارائه دهنده خدماتی اینترنت و سرویس دهنده هایی استفاده شود که بسته های تجزیه ICMP را کاهش داده اند این خود باعث می‌شود صفحات وب در دسترس قرار گیرند . پت کوچک نیز با بخش بزرگتر عملی است مانند SCP بعد از اینکه طرح معرفی شد ما از هدف TCPMSS برای غلبه بر این مسئله استفاده خواهیم کرد و باید MSS (اندازه قطعه ماکزیمم) به DMTU اضافه شود (واحد انتقال ماکزیمم مسیر) . این روش می‌تواند کنترل Netfilter را نشان دهد و تابع ISP و سرویس دهنده خدماتی در سیستم kernel فراخوانی شود .

شناخت کاربردی Iptable

CONFIG- IP- NF- COMPAT- IPCHAINS - این گزینه یک مد سازگاری را با ipchains اضافه می‌کند . به این مورد توجه نکنید و راه حل بلند مدت را در حل حرکت از linux 2/2 به 2/4 ارائه کنید . این جا kernel 2/6 استفاده خواهد شد .

همان طور که می‌توان دید یک سری گزینه معرفی شده است . در این جا باید دید چه نوع رفتارهای مازادی از مدلها حاصل می‌شوند اینها تنها گزینه ها در linux 2,4,9, kernel هستند . اگر بخواهید به این گزینه ها توجه کنید باید تابع patch - o- matic در Netfilter را در نظر داشته باشید و به گزینه ها در kernel توجه کنید . تثبیت POM نیز می‌تواند در kernel عملی شود و باید بتوان به kernel دست یافت . این توابع باید در آینده اضافه شوند ولی هنوز ساخته نشده اند . این خود دلایل متفاوت دارد مانند مسیر غیر ثابت به linux torvalds که نمی‌تواند حفظ شود و یا مسیر به kernel جریان اصلی که هنوز آزمایشی است .

شما به گزینه های زیر به صورت کامپایل در kernel نیاز دارید تا دست نوشته ....rc را اجرا کنید . اگر به گزینه هایی نیاز دارید که در دیگر دست نوشته ها نیاز هستند به بخش دست نوشته زیر مراجعه کنید .

شناخت کاربردی Iptable

برنامه

در میزان حداقل باید از دست نوشته rc.Firewall.txt استفاده شود . در دیگر مثالها نشان می‌دهم که چه نوع شرطی در این بخش نیاز است . اکنون باید دست نوشته اصلی مورد مطالعه در نظر باشد .

۳-۲ : نصب محل کاربر :

اول از همه باید دید که چگونه بسته های iptable کامپایل می‌شوند باید دید که در بیشتر سیستم ها و کامپایل iptable باید سیستم و کامپایل kernel صورت گیـــــرد . تـــــوزیع خاص نیز در بسته iptable عملی است و یکی از آنها RED HAT است . با این وجود RED HAT در هر پیش فرضی غیر فعال است . اکنون باید دید که چگونه این توزیعات در این فصل بررسی خواهند شد.

۱-۳-۲ : کامپایل برنامه های قلمرو کاربر :

نخست آنکه باید بسته های iptables آزاد شوند . در این جا از iptables a 1/26 و vanilla 2/4 kernel استفاده می‌شود . باز کردن بسته با استفاده از ....,hzip 2 صورت می‌گیرد (این می‌تواند با tar - xjvf.... عملی شود که باید یک دستور مانند دستور اول داشته باشد و با این وجود این با نسخه tar دیده نشده است).طرح باید به درستی در دایرکتوری به نام

شناخت کاربردی Iptable

iptables…. قرار گیرد. برای اطلاعات بیشتر باید iptables… را بتوانید که حاوی اطلاعات مفید در مورد کامپایل و دسترسی به اجرای برنامه است .

بعد از این باید گزینه طراحی و نصب مدل اضافی برای kernel در نظر گرفته شود . مرحله مورد توصیف کنترل و نصب بخش های استاندارد است که در kernel در نظر گرفته شده اند. این روند با کمک طرح اجرای عملی است .

بعضی از این طرح ها آزمایشی هستند و می‌توانند برای نصب مفید باشند . با این وجود باید هماهنگی جالب و اهداف در مرحله نصب صورت گیرد و این خود به روند خاص نیاز دارد برای انجام این مرحله باید این شکل از بسته iptables معرفی شود .

تولید patch فوری KERNEL- DIR/ USR/ SRC/ linux

متغییر KERNEL- DIR باید به محل واقعی اشاره کند که منبع kernel شما در آن واقع است . به طور طبیعی این به صورت usr/src… است ولی می‌تواند متغییر باشد و شاید در این منبع kernel در اختیار باشد .

این خود patch خاص را در مورد ورود به kernel گزارش می‌کند . patch بیشتر نیز توسط توسعه دهنده Netfilter می‌تواند به kernel اضافه شود .

این روند به طور واقعی انجام می‌شود یک روش نصب دستور زیر است .

شناخت کاربردی Iptable

برنامه

دستور فوق در مورد نصب قطعات دنیای Netfilter است که patch - o- metic نام دارد ولی حذف patch اضافی نیز عملی است .

توجه کنید که این بدان علت است که این دستورات واقعی اجرا می‌شوند آنها قبل از این که چیزی در منبع kernel تغییر کند مورد سوال قرار گیرند برای نصب تمام patch - o- metic باید دستور زیر اجرا شود .

برنامه

فراموش نکنید که هر patch را بطور کامل قبل از روند اجرا مطالعه کنید . بعضی از آنها patch دیگر را تخریب می‌کنند ولی مابقی kernel را تخریب می‌کنند در صورتی که با patch در patch - o- metic استفاده شوند.

نکته :شما می‌توانید مراحل فوق را نادیده بگیرد در صورتی که نخواهید kernel را جمع کنید و به عبارت دیگر لازم نیست موارد فوق را انجام دهید . با این وجود چیزهای جالب در patch - o- metic وجود دارد که می‌توانید جستجو کنید . بنابراین دستور اجرایی غلط نیز در نظر گرفته می‌شود .

بعد از این باید قطعه patch - o- metic نصب کامل شود و شما اکنون می‌توانید یک kernel جدید را کامپایل کنید و از patch جدید برای اضافه

شناخت کاربردی Iptable

کردن به منبع استفاده کنید سیستم باید طراحی شود و در این جا باید گزینه ها اضافه شوند . باید صبر کنید تا کامپایل کامل شود تا این که iptable برنامه اجرا شود.

کار را با کامپایل برنامه iptable ادامه دهید . برای کامپایل iptable یک دستوری مشابه دستور زیر صادر می‌شود.

برنامه

برنامه قلمرو کاربر باید به درستی کامپایل شود . در غیر این صورت باید بتوانید در لیت پست Netfilter عضو شوید . در این جا شانس کمک فراوان است . چندین چیز اشتباه در مورد نصب iptable وجود دارد و بنابراین اگر اقدام صورت نمی‌گیرد نگران نباشید . سعی کنید تا به روش منطقی اشکال را بشناسید و از کسی کمک بخواهید . اگر همه چیز درست است آماده نصب هستند . برای این کار باید دستور زیر ارسال شود .

برنامه

همه چیز باید در برنامه اجرا شود . برای استفاده از هر برنامه iptable باید اکنون نصب مجدد سیستم و kernel را انجام دهید در صورتی که قبلاً آن را انجام نداده اید . برای اطلاعات بیشتر در مورد نصب برنامه کاربر از منبع بر

فایل INSTALL در منبع مراجعه کنید که حاوی اطلاعات خارجی در مورد موضوع نصب است .

۲-۳-۲ : نصب بر Red Hat 7/1

Red Hat از قبل با یک kernel ۴/۲xنصب می‌شود که دارای Netfilter و iptable کامپایل شده در آن است . این طرح دارای تمام برنامه های کاربر و فایل سیستم برای اجرا است . با این وجود افراد Red Hat همه چیز را با استفاده از سیستم ipchain هماهنگ ناتوان کرده اند . برای رفع مشکل باید لیست متفاوت بستی در نظر گرفته شد. و به این دلیل است که iptable عمل نمی‌کند بنابراین باید دید که سیستم ipchain چگونه عمل می‌کند نصب iptable چگونه است .

نکته : امروزه نصب Red Hot 7/1 پیش فرض با نسخه قدیمی برنامه فضای کاربر عملی است و بنابراین ممکن است بخواهید تا نسخه جدید برنامه را کامپایل کنید و یک kernel کامپایل شده را نصب کنید قبل از این که به طور کامل از iptable استفاده شود. ابتدا باید طرح ipchains را غیر فعال کنید . بنابراین در آینده لجرا نمی‌شود برای این کار باید نام فایل را در ساختار دایرکتوری /etc/rcod تغییر دهید . دستور زیر باید استفاده شود .

برنامه

شناخت کاربردی Iptable

به این ترتیب به روابط نرم و رسم کردن دست نوشته... Htcl در k 92 ipchain اشاره دارند . اولین مرحله که در هر پیش فرض ۶ است اولین دست نشته را نشان می‌دهد . با تغییر این حالت به این k می‌گوییم که خدمات را kill کنید و یا این که آنرا در صورتی که قبلاً اجرا نشده است اجرا نکنید . اکنون این خدمات در آینده اجرا نخواهد شد .

با این وجود برای توقف خدمات از حالت اجرایی باید دستور دیگر را اجرا کنیم. این یک دستور خدماتی است که می‌تواند برای اجرای فعلی استفاده شود این دستور برای توقف خدمات ipchain ارائه می‌شود.

دستور

در نهایت برای راه اندازی خدمات iptable اقدام خواهد شد . ابتدا باید بدانیم کدام سطح اجرایی را می‌خواهیم اجرا کنیم این خود در سطح ۲، ۳، ۵ قرار می‌گیرد این سطوح اجرایی برای موارد زیر استفاده می‌شوند .

- Multiuser 2 بدون NFS و یا ۳ در صورتی که شبکه وجود نداشته باشند .
- ۳ مد چند کاربر کامل یعنی سطح اجرایی طبیعی
- xll ۵ این در صورتی استفاده می‌شود که به طور خودکار در x windows باشید.

شناخت کاربردی Iptable

برای تولید iptable در این سطوح اجرایی به دستور زیر نیاز است :

برنامه

دستورات فوق به عبارت دیگر اجرای iptable را در سطح ۲، ۳، ۵ باعث می‌شوند . اگر مایل هستید خدمات iptable در دیگر سطوح اجرا شوند باید که دستور صادر شود . با این وجود هیچ یک از سطوح اجرایی استفاده نمی‌شوند و بنابراین نباید آنرا برای سطوح اجرایی فعال کرد . سطح ۱ برای حالت کاربر واحد است یعنی وقتی که یک جعبه نشبیت می‌شود سطح ۴ باید غیر استفاده باشد و سطح ۶ برای shut down کامپیوتر است. برای فعال کردن خدمات iptable دستور زیر اجرا می‌شود.

دستور

هیچ قانونی در دست نوشته iptable وجود ندارد . برای اضافه کردن قوانین به یک جعبه Red Hot 7/1 دو روش وجود دارد .نخست آنکه باید دست نوشته …etc/rc را ویرایش کنید . این اثر غیر مطلوب حذف تمام قوانین را دارد در صورتی که بسته iptable یا RPM ارتقا یافته باشد . این روش به صورت 10 ad کردن مجموعه قوانین و ذخیره با دستور iptable- SNC است پس باید 10 ad خودکار یا rcod صورت گیرد.

شناخت کاربردی Iptable

ابتدا باید دید که چگونه نصب iptable با cut و past کردن iptable initod عملی است . برای اضافه کردن قوانینی که زمان شروع خدمات کامپیوتر اجرا می‌شوند باید آنها را تحت عنوان در تابع starto اضافه کنید . توجه کنید که اگر تحت بخش start عمل کنید تابع starto اجرا میشود ویرایش بخش (stop) تغییر می‌تواند نشان دهد که کدام دست نوشته و در چه زمانی اجرا می‌شود و یا چه چیز وارد سطح اجرایی می‌شود که به iptables نیاز ندارد باید بخش restart و (ondrestarte) کنترل شود . توجه کنید که تمام این اقدامات در صورتی کنترل می‌شوند که دارای شبکه Red Hot خودکار برای ارتقای بسته‌ها باشید . این می‌تواند با ارتقا از بسته iptable RPM حاصل شود .

روش دوم :نصب مستلزم این موارد است : ابتدا باید یک مجموعه قانون در فایل دست نوشته لایه ای نوشته شود و با تصمیم در iptable نوشته شود که نیاز شما را برآورد این بار آزمایش را فراموش نکنید . وقتی یک نصب بدون مسئله صورت گیرد همان طور که بدون اشکال دیده شد از دستور iptables - save استفاده کنید . شما م/یتوانید از آن به طور طبیعی استفاده کنید و یا این که iptables - saves.... را استفاده کنید که ذخیره مجموعه قوانین در فایل etc/sy.... است . این فایل به طور خودکار

شناخت کاربردی Iptable

توسط دست نوشته iptable rcd برای احیای مجموعه قوانین در آینده استفاده خواهد شد روش دیگر ذخیره دست نوشته با service iptable save خودکار ….etc/syscn است بعداً کامپیوتر دوباره راه اندازی می‌شود و دست نوشته iptable rcd از دستور iptable -restart برای ذخیره مجموعه قوانین از فایل save ….etc/ استفاده می‌کند .

این دو روش را با هم ترکیب نکنید زیرا ممکن است به هم آسیب برسانند و سیستم Firewall را تبدیل کنند .

وقتی تمام این مراحل کامل شد می‌توانید نصب ipchain و بسته iptable را کنسل کنید . این بدان علت است که نمی‌خواهید سیستم ما برنامه جدید iptable کاربر را با برنامه قدیمی از پیش نصب شده iptable ترکیب کند.

این مرحله در صورتی نیاز است که بخواهید iptable را از بسته منبع نصب کنید .غیر معمول نیست که بسته جدید و قدیم مخلوط شوند زیرا نصب مبنای rpm می‌تواند بسته را در محل غیر استاندارد نصب کند و ما با نصب بسته iptable می‌توانیم عمل نوشتن را تکرار کنیم . برای این کار از دستور زیر استفاده می‌شود . **برنامه**

چرا ipchains در صورتی که از آن استفاده نمی‌کنید در اطراف باقی می‌ماند . حذف آن مانند روش بانیری های iptable است.

شناخت کاربردی Iptable

برنامه

بعد از این که کار کامل شد با ارتقای بسته از منبع روبرو شده اید و دستوالعمل نصب منبع دنبال خواهد شد هیچ یک از بانیری های قدیمی ، کتابخانه ها و یا فایل شمول نباید در اهداف باشند .

فصل ۳ : عبور از جداول و زنجیره ها

در این فصل عبور بسته ها از زنجیره های متفاوت و با نظم خود توصیف خواهند شد همچنین نظم عبور جداول بررسی می‌شود و خواهیم دید که این روند بعداً تا چه حد با ارزش خواهد بود در صورتی که قوانین خاص را می‌نویسیم.همچنین نقاط ورود مولف های وابسته به kernel به تصویر را بررسی خواهیم کرد تصمیمات ردیابی نیز بررسی می‌شوند . این امر در صورتی مهم است که بخواهیم قوانین iptable را که تغییر الگوی ردیابی را برای بسته ها نشان دهند بنویسیم . باید دید که چرا و چگونه این بسته ها ردیابی می‌شوند . و نمونه های عالی DNAT و SNAT است بیت های Tos نباید فراموش شوند .

۳-۱ : کلیات :

وقتی که بسته ها ابتدا وارد Firewall شود با سخت افزار برخورد می‌کند و سپس وارد درایو وسیله مناسب در kernel می‌شود . پس بسته از یک مجموعه مراحل در kernel عبور خواهد کرد قبل از این که به برنامه صحیح ارسال شود و یا به میزبان دیگر برود. ابتدا بسته ای را بررسی می‌کنیم که برای میزبان داخلی مفروض است . از چندین نرحله باید عبور کرد قبل از اینکه به برنامه دریافت کننده برسیم .

شناخت کاربردی Iptable

توجه کنید که این بار بسته به جای عبور از زنجیره Fireward از INPUT عبور خواهد کرد این کاملاً منطقی است شاید تنها چیز منطقی در مورد عبور جداول و زنجیره ها در چه سیستم شما در ابتدا همین باشد ولی اگر به آن فکر کنید آنرا مفید تر خواهید دید.

در این مثال فرض می‌کنیم که بسته برای میزبان دیگر بر شبکه دیگر مفروض است . بسته مراحل متفاوت را به شرح زیر طی خواهد کرد .

همان طور که می‌توان دید یک سری مراحل فراوان وجود دارد . بسته می‌تواند در هر زنجیره iptableمتوقف شود در صورتی که شکل درست نداشته باشد . با این وجود ما به جوانب iptable توجه داریم . هیچ زنجیره خاصی از جداول برای روابط متفاوت و یا مشابه آن وجود ندارد . Fireward همیشه عبور با تمام بسته هایی است که به این مسیر و Firewall به جلو می‌آیند .

نکته :از زنجیره INPUT برای فیلتر سناریوی قبل استفاده نکنید . INPUT فقط برای بسته ها در میزبان داخلی شما به کار می‌رود که در اطراف مقصد دیگر نیست . اکنون می‌توان دید که چگونه زنجیره های متفاوت در این ۳ سناریوی جداگانه عبور می‌کنند اگر یک الگوی درست از طرح ارائه شود به شرح زیر است .

شناخت کاربردی Iptable

برای تشریح این تصویر باید این طرح را در نظر بگیرید . اگر یک بسته در تصمیم ردیابی اول داشته باشیم که برای خود ماشین داخلی مفروض نیست مسیر از زنجیره Forward است . اگر بسته برعکس برای آدرس IP باشد که ماشین داخلی به آن گوش می‌کند بسته از زنجیره ارسال می‌شود و به ماشین داخلی می‌رسد.

همانطور که گفته شد بسته ها می‌توانند برای ماشین داخلی در نظر گرفته شوند ولی آدرس مقصد می‌تواند از زنجیره PREUTING یا اقدام NAT تغییر کند . چون این امر قبل از اولین ردیابی تصمیم عملی است بسته بعد از این تغییر جستجو خواهد شد . به این علت ردیابی تصمیم تغییر میکند توجه کنید که تمام بسته ها از یک یا چند مسیر در تصویر عبور میکنند . اگر DNATبسته به یک شبکه صورت گیرد حرکت از مابقی زنجیره انجام خواهد شد تا این که به شبکه برسیم .

نکته :اگر احساس می‌کنید که اطلاعات بیشتر نیاز دارید می‌توانید از دست موشته آزمایشی باید قوانین آزمایشی چکونگی عبور جداول در زنجیره ها را ارائه کند .

۲-۳ : جدول ترکیبی

این جدول همانطور که گفته شد برای ترگیب بسته ها استفاده می‌شود . به عبارت دیگر به آسانی از تطابق ترکیبی استفاده می‌شود که می‌تواند تغییر فیلد TOSباشد .

نکته :شما نباید از این جدول برای فیلتر استفاده کنید و یا هر DNAT , SNAT ویا Masquerading را در این جدول استفاده نمایید .

اهدافی که در جدول ترکیبی معتبر عبارتند از MARK - TTL - TOS هرف TOSبرای تنظیم و تغییر فیلد نوع خدماتی در بسته استفاده می‌شود . این می‌تواند برای نصب خواستنی ها بر شبکه در مورد چگونگی مسیر یابی بسته و غیر استفاده شود . توجه کنید که این کامل نشده است و واقعاً بر اینترنت اجرا نمی‌شود و بیشتر ردیاب ها به ارزش در فیلد توجه ندارند و گاهی اوقات آنها می‌توانند عملکرد را ارائه کنند در این جا بسته ها به اینترنت می‌روند مگر این که بخواهید تصمیم را بر آن یا iprute 2 اراده کنید .

هدف TTL برای تغییر TTL (زمان عصر) در بسته استفاده می‌شود ما می‌توانیم بگوییم که بسته ها دارای یک TTL خاص هستند . یک دلیل برای آن این است که نمی‌خواهیم خود را از ارائه دهنده خدمات اینترنت در کنیم . بعضی از ارائه دهندگان دوست ندارند که کاربر ما چند کامپیوتر را به یک اتصال واحدتر راه اندازی کنند و ارائه دهندگان به دنبال تولید ارزش های متفاوت TTL توسط میزبان هستند و این خود نشانه اتصال کامپیوترها به یک ارتباط واحد است.

هدف MARK برای نصب ارزش مارک خاص در بسته استفاده می‌شود . این مارک ها می‌توانند توسط برنامه iproute 2 برای مسیرهای متفاوت بر بسته شناخته شوند و این خود به علامت بستگی دارد ما می‌توانیم حد پهنای باند و ردیف بندی طبقاتی را بر اساس این علائم داشته باشیم.

۳-۳ : جدول nat

این جدول باید فقط برای NAT (ترجمه آدرس شبکه) به بسته های متفاوت استفاده شود . به عبارت دیگر این جدول برای ترجمه فیلد منبع بسته و یا فیلد مقصد به کار می‌رود توجه کنید که همان طور که گفته شد فقط بسته اول در جریان وارد این زنجیره می‌شود بعد از این مابقی بسته ها

شناخت کاربردی Iptable

به طور خودکار یک اقدام مانن بسته اول دارند . اهداف واقعی که با این امور انجام می‌شوند عبارتند از : MASQUERADE - SNAT - DNAT

هدف DNAT عمدتاً در مواردی استفاده می‌شود که یک IPعمومی وجود دارد و میخواهیم تا دسترسی را بهFIREWALL و دیگر میزبانها آسان کنید (مانند DMZ) به عبارت دیگر ما آدرس مقصد را در بسته تغییر می‌دهیم . آنرا به میزبان برمی‌گردانیم

SNAT برای تغییر آدرس منبع بسته ها استفاده می‌شود . در بیشتر موارد شبکه های داخلی یا DMZ پنهان می‌شوند . یک مثال در واقع Firewall معلومات در خارج از آدرس IP است و باید شماره IP شبکه داخلی با شماره Firewall عوض شود با این هدف Firewall به طور خودکار بسته ها را SNAT و DE-SNAT می‌کند و بنابراین می‌توان اتصالات را از LAN به اینترنت برقرار کرد اثر شبکه شما از 192/168/…/netmask استفاده می‌کند بسته ها هرگز از اینترنت برنمی‌گردند زیرا LAN این شبکه ها را به صورت خصوصی و یا برای استفاده در LAN جداگانه تنظیم کرده است .

هدف MASRUERADE مانند SNAT استفاده می‌شود ولی هدف مربوطه کلی جای بیشتر در کامپیوتر نیاز دارد دلیل آن این است که هر زبان که هدف MASRUERADE با بسته در برخورد است به طور خودکار

آدرس IP را برای استفاده کنترل می‌کند آنرا مانند هدف SNAT استفاده خواهد کرد . این امر با آدرس واحد IP عملی است . این هدف کار درست با آدرس Dynamic DHCP IP را که ISP آنرا برای اتصال ppp ، pppoe یا SLIP به اینترنت ارائه کرده است راه اندازی می‌کند .

۳-۴ : جدول فیلتر :

این جدول برای فیلتر بسته ها استفاده می‌شود . ما می‌توانیم بسته ها را هماهنگ سازیم و آن ها را هر گونه که می‌خواهیم فیلتر کنیم . اینجا اقدام مخالف بسته ها عملی است و خواهیم دید که چه چیز در آنها وجود دارد و بسته به محتویات DROP یا ACCEPT را مشخص می‌کنیم البته ممکن است فیلتر قبلی صورت گرفته باشند با این وجود این جدول محل تعیین فیلتر برای طراحی است . تقریباً تمام اهداف در این زنجیره استفاده می‌شوند . ما در مورد جدول فیلتر مهارت داریم و می‌دانیم که این جدول محل درست فیلتر اصلی است .

فصل ۴ : ماشین حالت

این فصل در مورد ماشین حالت و جزئیات آن است . بعد از خواند فعل باید بدانید که چگونه این ماشین عمل خواهد کرد ما یک مجموعه مثال در مورد چگونگی عملکرد آن ارائه خواهیم کرد . همچنین یک مجموعه مثال از چگونگی بررسی این ماشین ارائه شده است اینها روند کاربردی را توجیه می‌کنند .

۴-۱ : مقدمه :

ماشین حالت یک بخش خاص در iptable است که نباید واقعاً ماشین حالت خوانده شود زیرا واقعاً یک ماشین ردیابی اتصال است . با این وجود بیشتر افراد آنرا با این نام می‌شناسند در تمام این فصل از این نام ها به صورت مترادف استفاده شده است . این نباید ابهام ایجاد می‌کنند .ردیابی اتصال در چهارچوب Netfilter صورت می‌گیرد Firewall هایی که این را اجرا می‌کنند Firewall حالتی نام دارند یک نوع از این Firewall ایمن تر از Firewall غیر حالتی است زیر لبه ها در نوشتن قوانین سخت تر کمک می‌کند .

در iptable بسته ها با اتصالات ردیابی شد در ۴ حالت ارتباط دارند . آنها عبارتند از: INVALIP , RELATED , ESTABLISHED ,

شناخت کاربردی Iptable

- NEW . ما مرتبه را به تفصیل بررسی خواهیم کرد . با هماهنگی STATE می‌توان نشان داد که چه کسی جلسات جدید را آغاز می‌کند .

بنابراین ردیابی اتصال توسط چهارچوب خاص در kernel به نام conntrack صورت می‌گیرد . conntrack می‌تواند به صورت مدول یا بخش درونی خود kernel باردار شود در بیشتر موارد ردیابی اتصال خاص نسبت به موتور پیش فرض نیاز است . به این علت قطعات خاص از آن وجود دارند که پروتکل TCP , UDP , ICMP را کنترل می‌کنند این مدل ها اطلاعات خاص از بسته ها ارائه می‌کنند به طوری که بتوانند مسیر جریان داده ای را دنبال کنند این اطلاعات برای تعیین حالت جهان استفاده می‌شوند به عنوان مثال جریان UDP با آدرس IP مقصد شناخته می‌شود و آدرس IPمنبع و port مقصد و منبع در آن نقش دارد .

در kernel قبلی احتمال خاموش و روشن کردن فرآیند قطعه زدایی وجود داشته است با این وجود چون iptable و Netfilter معرفی می‌شوند و ردیابی اتصال صورت می‌گیرد این گزینه رها می‌شود دلیل آن این است که ردیابی اتصال نمی‌تواند به درستی بدون قطعه زدایی بسته ها عمل کند و بنابراین قطعه زدایی با Conntrack ترکیب می‌شوند و این به طور خودکار

صورت می‌گیرد . این نمی‌تواند با خاموش کردن اتصال غیرفعال شود . تصمیم زدایی همیشه در صورتی عملی است که ردیابی اتصال فعال شود.

تمام ردیابی اتصال در زنجیره PREROUTING کنترل می‌شود به جز بسته های داخلی که در زنجیره OUT PUT کنترل می‌شوند . این بدان معناست که iptable تمام محاسبات حالات را در زنجیره PREROUTING نشان می‌دهد. اگر بسته اولیه را در یک جریان ارسال کنیم حالت در زنجیره ESTABLISHED - PREROUTING تغییر می‌کند اگر اولین بسته فعال نشود حالت جدید در زنجیره PREROUTING قرار می‌گیرد بنابراین تمام تغییرات و محاسبات در PREROUTING و یا زنجیره out put جدول nat صورت می‌گیرد .

۲-۴ : ورودی Conntrack :

اکنون ورودی Conntrack و چگونگی خواندن آنرا در ..../proc بررسی خواهیم کرد . این لیست ورودی جریان در پایگاه Conntrack است اگر یک مدل ip - conntrack فعال شود یک cat از ..../proc/net به شرح زیر است :

برنامه

شناخت کاربردی Iptable

این مثال شامل تمام اطلاعاتی است که مدل Conntrack نیاز دارد تا برای نوع اتصال به کار برد ابتدا یک پروتکل به صورت tcp مطرح است . پس یک ارزش در کد گذاری اعشاری مطرح است بعد از آن باید دید که چه مدت این ورودی زنده است این رقم در ۱۱۷ ثانیه مشخص خواهد شد این رقم پیش فرض بوده است . اکنون حالت واقعی در ورودی منظور است و باید حالت SYN- SENY در نظر گرفته شود . ارزش درونی اتصال با ارزش مورد استفاده به خارجی iptable تفاوت دارد ارزش SYN- SENY می‌گوید که ما اتصال را در بسته TCP SYN در یک جهت قرار داده ایم سپس آدرس منبع IP ، آدرس مقصد IP ، port منبع و مقصد را بررسی خواهیم کرد . در این جا کد کلید وار مفروض کرد که برگشت ترافیک را در این اتصال نشان می‌دهد . می‌توان دید که از این بسته ها چه چیز مفروض است اطلاعات آدرس مقصد و منبع IP را نشان می‌دهند این امر برای port منبع و مقصد مفروض است این ارزش ها مرد توجه ما است .

ورودی ردیاب اتصال می‌تواند بر مجموعه متفاوت ارزشها باشد که همگی در سر عنوان conntrack در فایل ....linux قرار گرفته اند . این ارزشها به نوع پروتکل فرعی IP مورد استفاده بستگی دارد پروتکل ICMP , UDP , TCP پیش فرض متفاوت ارزش مانند LINUX دارد در این جا هر

شناخت کاربردی Iptable

پروتکل را بررسی خواهیم کرد با این وجود از آنها در این فصل زیاد استفاده نخواهیم کرد زیرا آنها در خارج از عوامل درونی conntrack نمی‌باشند همچنین بسته به چگونگی تغییر حالت ارزش پیش فرض زمان تا موفعی که این ارتباط تخریب شود تغییر خواهد کرد .

اخیراً یک patch جدید در patch - o- matic در دسترس است که ردیابی پنجره ای tcp نام دارد این patch می‌تواند تمام متغیرهای Syctl را اضافه کند این بدان معناست که آنها در زمان حرکت تغییر می‌کنند اگر چه سیستم هنوز در حالت اجرایی است بنابراین این خود کامپایل مجدد kernel را در زمان تغییر برنامه غیر ضروری می‌سازد . اینها از طریق فراخوانی سیستم موجود در دایرکتوری ..../proc/ تغییر می‌کند شما باید به متغییر ..../proc/ مراجعه کنید.

وقتی یک اتصال یک ترافیک در دوجهت دارد ورودی conntrack می‌تواند علامت [UNREPLED] را پاک کند و آنرا resetکند ورودی می‌گوید که اتصال شاهد ترافیک در دوجهت سند است و علامت [ASSURED] عوض می‌شود علامت [ASSURED] می‌گوید که این ارتباط تضمین شده است و اگر به ماکزیمم ارتباط برسیم پاک نمی‌شود بنابراین ارتباط به صورت [ASSURED] است و پاک نخواهد شد برخلاف اتصال به صورت

[ASSURED] مقدار اتصالات جدول ردیابی به متغییر نصب شده در تابع kernelدرip- sysctl جدید بستگی دارد مقدار پیش فرض با این ورودی بسته به مقدار حافظه تغییر خواهد کرد بر 128mb از RAM می‌توان به ۸۱۹۲ ورودی دست یافت در 256 MBازRAM به ۱۶۳۷۶ ورودی دست می‌یابیم شما می‌توانید تنظیمات را از طریق .../proc/ بخوانید و نصب کنند .

۳-۴ : حالات بخش کاربر

همانطور که دیدید بسته ها می‌توانند حالات متفاوت در kernel داشته باشند و این امر به نوع پروتکل مورد بحث بستگی دارد . با این وجود در خارج از kernel فقط ۴ حالت است که توصیف شد این حالات می‌تواند به همه تطابق استفاده شوند که بر اساس حالت ردیابی فعلی معرفی شد. حالات معتبر عبارتند از , INWALIP , RELATEP , ESTABLISHED NEW جدول زیر هر یک را نشان می‌دهد .

این حالات می‌توانند با state - برای هماهنگی بر اساس روبط استفاده شوند این خود ماشین حالت را قوی و کارآمد می‌سازد ما باید تمام port را بالای lt 4 قرار دهیم تا ترافیک به شبکه داخلی برگردد با این وجود ماشین حالت

لازم نیست تا محل طولانی حاصل شود زیرا می‌توانیم Firewall را در ترافیک برگشتی قرار دهیم که برای تمام ترافیک ها نیاز نیست

۴-۴ : اتصالات TCP

در این بخش و بخش بعدی بررسی حالات و چگونگی کنترل هر یک از ۳ پروتکل TCP , UDP , ICMP آمده است همچنین خواهیم دید که چگونه اتصالات در هر پیش فرض کنترل می‌شوند در صورتی که به صورت این ۳ پروتکل طبقه بندی نشده باشند کار را با پروتکل TCP آغاز می‌کنیم زیرا حالتی است و جزئیات جالب با توجه ب ماشین حالت در iptable دارد . این نوع ردیابی اتصال مانند انواع کلی اتصال مفید است در اینجا خواهیم دید که حالت جریان ورودی در طی مراحل متفاوت اتصال چگونه است همان طور که دیدید ردیابی واقعاً جریان اتصال TCP را از دیدگاه کاربر نشان نمی‌دهد . وقتی این جا بسته(SYN) گزارش شد اتصال به صورت NEW است وقتی بسته به صورت SYN/ACK باشد یم اتصال بصورت ESTABLISHED مطرح است اگر اینها را در درجه دوم در نظر بگیرید دلیل را خواهید فهمید با این اجرای خاص می‌توانید به NEW و ESTABLISHED اجازه دهید تا شبکه داخلی را رها کنند و فقط اتصال ESTABLISHED برمی‌گردد . برعکس اگر ماشین ردیابی اتصال کل سیستم را به صورت NEW نشان دهد

شناخت کاربردی Iptable

نمی‌توان اتصال خارجی را در داخل متوقف کرد و NEW برمی‌گردد برای پیچیده کردن موضوع یک سری حالات درونی برای tcp مطرح می‌شود و در قلمرو کاربر به کار میرود آن منابع استاندارد93 RFCV به پروتکل کنترل انتقال صفحه ۲۱-۲۳ هستند این ها را به شرح زیر بررسی خواهیم کرد.

همانطور که دیدید این واقعاً از دیدگاه کارب رساده است با این وجود با بررسی کلی ساختار از دیدگاه KERNEL روند کار مشکل می‌شود اکنون یک نمونه خواهد آمد باید دید که چگونه حالات در ..../proc/ تغییر میکند اولین حالت دریافت اولین بسته SYN در یک اتصال است.

برنامه

همانطور که از ورودی فوق می‌توان دید ما یک حالت دقیق داریم که در آن بسته SYN ارسال شده است (علامت SYN - SENT تنظیم می‌شود) و هیچ جوابی برای آن ارسال نشده است (نمایش علامت UNREDLLED) این حالت درونی بعدی زماین حاصل می‌شود که یک بسته دیگر در دیگر جهت باشد.

برنامه

اکنون یک SYN/ACK برگشتی و دریافت خواهد شد وقتی این بسته دریافت شد تغییر حالت روی می‌دهد و این زمان ...SYN- REC

شناخت کاربردی Iptable

می‌گوید که SYN اصلی به درستی ارائه شده است و بسته برگشتنی SYN/ACK بغیر از Firewall به درستی حرکت کرده است به علاوه این ورودی ردیاب اتصال اکنون در دو جهت ترافیک دارد و بنابراین کاربرد داشته است این روند صریح است ولی علامت [UNREPLED] ظاهر میشود مرحله نهایی زمانی حاصل میشود که ACK نهایی در بخش ۳ مسیری باشد.

برنامه

در مثال آخر دارای ACK نهایی در بخش ۳ مسیری هستیم و اتصال به حالت ESTABLISHED رسیده است مادامیکه مکانیزم درونی IPTABLE آگاه باشد بعد از چند بسته بیشتر اتصال دوباره [ASSURED] خواهد شد همانطور که در مقدمه آمده است.

وقتی یک اتصال TCP بسته شد به روش زیر صورت خواهد گرفت و حالت زیر ار دارد همانطور که میتوان دید اتصال هرگز بسته نمیشود تا این که آخرین ACK نیز ارسال شود این روند در توصیف چگونگی بسته شدن تحت شرایط طبیعی کاربرد ندارد یک اتصال می‌تواند با ارسال یک RST (reset) بسته شود و این در صورتی است که انکار شده باشد در این حالت یک اتصال بعد از زمان از پیش معلوم بسته خواهد شد .

شناخت کاربردی Iptable

وقتی اتصال TCP بسته شد وارد حالت TIME NAIT میشود که در پیش فرض ۲ دقیقه است این خود به گونه ای استفاده می‌شود که تمام بسته هایی که از نظم خارج شده اند بتوانند در حالت تنظیم قرار گیرند حتی بعد از این که اتصال بسته شد این یک نوع زمان با من است به طوری که برای مسیر دیگر حفظ شود و یا به پایانه اتصال برسد .

اگر اتصال توسط بسته RST به صورتreset درآید حالت به close تبدیل دستور این بدان معنا است که اتصال در هر پیش فرض ۱۰ ثانیه قبل از بسته شدن فعال شود . بسته های RSTدر هر مفهوم تایید می‌شوند و اتصال را مستقیماً نقض می‌کنند یک حالت دیگر نیز وجود دارد در این جا لیست کامل حالات ممکن که در جریان TCP دیده می‌شوند و ارزش زمانی آنها آمده است

این ارزش ها به صورت مطلق مشخص نمی‌شوند آنها با تجدید kernelتغییر می‌کنند و از طریق سیستم فایل proc عوض خواهند شد به متغیر /proc/... تبدیل می‌شوند .

مقادیر پیش فرض باید عملاً تثبیت شوند این مقادیر در jiffites قرار گیـــرنـــد (۱۱۱۰۰ بخش از ثانیه) و بنابراین ۳۰۰۰ به معنای ۳۰ ثانیه است

شناخت کاربردی Iptable

نکته :توجه کنید که بخش کاربر در این ماشین شاهد علامت TCPدر بسته TCP نیست این روند مفید نیست زیرا ممکن است بخواهید بسته ها را در حالت NEW قرار دهید تا از Firewall عبور کنید ولی علامت NEW مشخص می‌شوند و در بیشتر موارد به معنای بسته های SYNاست این ویژگی با اجرای حالت فعلی دیده نمی‌شود حتی یک بسته بودن تنظیم بیت علامت ACK یک NEW است در صورتی که هماهنگی بر بسته NEW دیده شود این برای Firewalling استفاده می‌شود ولی معمولاً در شبکه خانگی مفید نیست در این جا یک Firewallواحد وجود دارد برای رسیدن به این رفتار از دستور State NEW packets ar / N . SYN bit set در ضمیمه سوالات و مسائل رایج استفاده می‌شود روش دیگر نصب توسعه tcp - window - tracking از patch - o- matic است که یک Firewallرا در ردیابی حالات بسته به نصب ویندوی tcp کمک می‌کند .

۵-۴ : اتصالات UDP

این اتصالات به تنهایی اتصالات حالتی نمی باشند بلکه بدون حالت هستند چندین دلیل برای آن وجود داردو مهمترین انها این است که آنها دارای محل اتصال و یا محل بسته شدن نمی‌باشند . در بین آنها توالی دیده نمی‌شود دریافت ۲ تا UDP DATA STAR در یک نظم خاص چیزی در مورد

شناخت کاربردی Iptable

نظم و محل ارسال نمی‌گوید با این وجود می‌توان حالات را بر اتصالات در kernel نصب کرد این جا چگونگی ردیابی اتصال و نمایش آن در conntrack آمده است.

همانطور که می‌توان دید اتصال دقیقاً مانند اتصال TCP رشد می‌کند از دیدگاه کاربر این روند عملی است از نظر درونی اطلاعات conntrack کلی متفاوت هستند ولی جزئیات یکی است اول ورودی را بعد از ارسال اولین بسته UDP بررسی خواهیم کرد :

برنامه

از این حالت سرویس دهنده یک پاسخ به اولین بسته ارسال کرده است و اتصال اکنون به صورت ESTABLISHED است این حالت و در ردیابی اتصال دیده نمی‌شود همانطور که می‌توان دید اختلاف اصلی آن است که علامت [UNREPLED] اکنون دیده نمبشودبه علاوه زمان پیش فرض به ۱۸۰ ثانیه رسیده است ولی در این مثال به ۱۷۰ ثانیه در زمان ۱۰ ثانیه ای رسیده است و ۱۶۰ ثانیه خواهد شد یک نکته در این جا نادیده گرفته شده است و می‌تواند به بیت اشاره کند و [ASSURED] است برای این علامت و تنظیم آن بر اتصال ردیابی شد باید یک مقدار ترافیک بر آن اتصال وجود داشته باشد .

برنامه

در این حالت اتصال مورد تایید است اتصال دقیقاً مانند مثال قبل است به جز آن برای علامت [ASSURED] اگر این اتصال برای ۱۸ ثانیه استفاده نشود زمان طولانی در ظاهر شدن دارد ۱۸۰ ثانیه یک مقدار کم است ولی باید برای بیشتر استفاده ها کافی باشد این رقم برای مقدار کامل در هر بسته حاصل می‌شود تا هماهنگی با یک ورودی دیده شود این حالت را درونی گویند

۶-۴ : اتصالات ICMP

بسته های ICMP از جریان حالتی دور هستند زیرا فقط برای کنترل استفاده میشوند و هرگز نباید اتصال را برقرار کنند ۴ نوع ICMP وجود دارد که بسته های برگشتی تولید کی‌کند و اینها ۲ حالت متفاوت دارند این نیازهای ICMP میتواند حالتNEW و ESTABLISHED داشته باشند انواع ICMP عبارتند از تقاضا وپاسخ ECHO ، تقاضا وپاسخ it imestamp تقاضا وپاسخ INFORMATION و تقاضا وپاسخ آدرس . و بین آنها تقاضای timestamp و Information دور هستند و میتوانند حذف شوند با این وجود پیام ECHO و چندین نصب مانند میزبان pinging استفاده می‌شود تقاضای آدرس استفاده نمی‌شود ولی می‌تواند در

شناخت کاربردی Iptable

زمان خاص مفید باشد برای بررسی چگونگی این حالت به تصویر زیر مراجعه کنید .

همانطور که میتوان دید میزبان یک تقاضای ECHO به هدف ارسال می‌کند که در واقع از سوی Firewall به صورت NEW در نظر گرفته میشود این هدف پاسخ به تقاضای ECHO است که ESTABLISHED خواهد بود وقتی اولین تقاضا ECHO دیده شد ورودی زیر به ip - conntrack میرود.

برنامه

این ورودی کلی متفاوت با حالات استاندارد TCP و UDP است پروتگل در این جا ظاهر می‌شود و آدرس مقصد و منبع مشخص میگردد این مسائل بعداً ظاهر خواهند شد ما دارای ۳ فیلد جدید به نام type , code , id هستیم آنها خاص هستند و فیلد type دارای نوع ICMP است و code دارای کد

شناخت کاربردی Iptable

ICMP است اینها در ضمیمه ICMP TYPE آمده اند وفیلد نهایی id دارای ICMP ID است هر بسته ICMP دارای id است و در این جا پیام ICMP دریافت می‌شود و یک id در پیام جدید ICMP قرار می‌گیرد به طوری که فرستنده جواب را بشناسند و بتواند با تقاضای ICMP ارتباط برقرار کند .

فیلد بعدی به صورت [UNREDLTIED] است که قبلاً دیده نشده در این جا علامت می‌گوید که ردیابی اتصالچگونه است ترافیک در یک جهت صورت می‌گیرد توقعات جواب برای بسته ICMP دیده شده است و این خود وارونگی آدرس IP منبع و مقصد است اینها مانند کد و نوع به مقادیر صحیح بسته برگشتی تبدیل می‌شوند بنابراین هر تقاضای ECHO به جواب ECHO تبدیل میشودICMP ID از بسته تقاضا حفظ میشود .

بسته جواب ESTABLISHED است با این وجود بعد از جواب ICMP قطعاً هیچ ترافیک قانونی در یک اتصال وجود ندارد به این علت ورودی ردیابی اتصال در زمانکد است پاسخ از ساختار Netfilter تخریب می‌شود .

در هر یک از این موارد تقاضا NEW است اگر چه پاسخ ESTABLISHED است اکنون آنها را بررسی خواهیم کرد وقتی

شناخت کاربردی Iptable

Firewall یک بسته تقاضا دارد یک NEW فرض می‌شود وقتی میزبان بسته پاسخ را به تقاضا ارسال میکند ESTABLISHED است .

تقاضای ICMP دارای زمان پیش فرض ۳۰ ثانیه است که میتواند در ورودی PROC/... تغییر کند این بطور کل یک مقدار دست زمانی است زیرا میتواند در بسته های را در حالت انتقال قرار دهی بخش دیگر ICMP نشان میدهد که این حالت برای توصیف و نوع اتصال udp و tcp و یا دیگر اتصالات برای میزبان استفاده میشود به این دلیل پاسخ های ICMP به صورت RELATED در اتصال اصلی دیده میشوند یک مثال ساده میزبان ICMP و یا غیر قابل دسترس بودن شبکه ICMP است اینها باید همیشه به میزبان برگردند. و این در حالتی است که یک اتصال ناموفق با میزبان دیگر برقرار شود ولی شبکه یا میزبان میتواند پایین باشد و بنابراین آخرین ردیاب به محل برسد و یا پیام ICMP پاسخ دهد در این حالت پاسخ ICMP بسته RELATED است تصویر زیر چگونگی آنرا نشان میدهد در مثال فوق یک بسته syn به آدرس خاص ارسال میشود این یک اتصال NEW است

با این وجود شبکه بسته باید قابل دسترسی شود و بنابراین ردیاب یک خطای RELATED نشان می دهد و بنابراین پاسخ ICMP به درستی به

سرویس گیرنده ارسال میشود . در ضمن Firewall ورودی ردیابی اتصال را تخریب کرده است زیرا میداند که این یک پیام خطا بوده است .

همین رفتار فوق در مورد اتصالات UDP دیده شده است در صورتیکه در هر مسئله کاربرد داشته باشد تمام پیام ICMP ارسال شد در پاسخ به اتصالات UDPبه صورت RELATED است تصویر زیر را در نظر بگیرید.

این بار یک بسته UDP به میزبان ارسال میشود این اتصالات UDP را NEW می خوانند با این وجود شبکه با ردیاب و Firewall از نظر اجرایی ممنوع میشود بنابراین Firewall یک ICMP Network prohibited را در برگشت دریافت میکنند Firewall میداند که این پیام خطای ICMP با اتصال UDP ارتباط دارد و آنرا به صورت بسته RELATED به سرویس گیرنده ارسال میکند در این حالت Firewall ورودی ردیابی اتصال را تخریب میکند و سرویس گیرنده پیام ICMP را دریافت میکند و باید منسوخ شود.

۷-۴ : اتصالات پیش فرض

در موارد خاص ماشین conntrack نمی‌داند که عبور پروتکل خاص را کنترل کند این در صورتی است که ماشین نداند که پروتکل چگونه است و چگونه کار میکند در این موارد باید به رفتار پیش فرض روی آورد این رفتار بر NETBIT , MUX , EGP استفاده میشود و مانند ردیابی اتصال UDP است اولین بسته NEW است و ترافیک پاسخ ESTABLISHED است وقتی رفتار پیش فرض استفاده شد تمام این بسته ها یک ارزش زمانی پیش فرض دارد این ارزش از طریق متغییر ..../proc/ تنظیم میشود مقدار پیش فرض در این جا ۶۰۰ ثانیه یا ۱۰ دقیقه است .بسته به تلاش ترافیک در ارسال برلینک که از ردیابی اتصال پیش فرض استفاده می کند این روند نیاز به تغییر دارد . اگر ترافیک از طریق ماهواره کنترل شود زمان طولانی نیاز است .

۸-۴ : پروتکل پیچیده و ردیابی اتصال

پروتکل های خاص پیچیده تر هستند . این روند در مورد ردیابی اتصال به آن معنا است که این پروتکل ها میتوانند سخت تر ردیابی شوند . نمونه های از آنها JJCB ، TRC و FIP است . هر یک از انها دارای اطلاعاتی درباره داده ای واقعی بسته است و بنابراین به helper نیاز دارد تا درست عمل کند .

شناخت کاربردی Iptable

اکنون پروتکل FIP را به عنوان مثال بررسی کنیم. پروتکل FIP ابتدا یک اتصال را به نام جلسه کنترل FIP باز می کند . وقتی دستور از طریق این جلسه صادر شد دیگر port ما باز می شود تا مابقی داده مربوط به دستور خاص را حمل کند این اتصالات می توانند به دو روش فعال وغیر فعال باشند . وقتی اتصال فعال باشد سرویس گیرنده FIP یک port به سرویس دهنده ارسال میکند و آدرس IP در ارتباط است . بعد از آن سرویس گیرنده FIP باید port را باز کند و سرویس دهنده از port 20 به آن وصل شود ۰ (به نام FTP-DATA) و داده ها را ارسال کند .

مسئله آن است که Firewall در مورد این اتصالات چیزی نمیداند زیرا آنها درباره داده ای پروتکل قرار گرفته اند . به این علت Firewall نمی تواند اتصال سرویس دهنده را برقرار کند . حل این مسئله اضافه کردن helper خاص به بخش ردیابی اتصال است که از طریق داده ها در اتصال کنترل برای نحو و اطلاعات خاص اسکن شود.وقتی اطلاعات درست باشند به صورت RELATED بیان می شوند و سرویس دهنده میتواند اتصال را ردیابی کند و این با ورود RELATED عملی است . تصویر زیر حالات را زمان اتصال FIP در سرویس گیرنده نشان می دهد .

شناخت کاربردی Iptable

FIP غیر فعال به روش مخالفت عمل میکند . سرویس گیرنده FIP به سرویس دهنده میگوید که میخواهد داده های خاصی داشته باشد که با آدرس IP جواب می دهند .سرویس گیرنده با دریافت این داده ها یا port خاص در ارتباط است و از port 20 می‌تواند داده ها را ارسال کند (ETPDATA.rd). اگر یک سرویس دهنده FIP پشت Firewall دارید باید این سیستم را در حالت iptable استاندارد قرار دهید و سپس اتصال اینترنتی را به سرویس دهنده FTP برقرار کند . همین امر در صورتی که چند کار بردارید صادق است و اگر بخواهید به سرویس دهنده HTIP و FTP برسید می توانید port دیگر را غیر فعال کنید تصویر زیر در مورد passive FTP آمده است .

تعدادی conntrack helper در خود kernel در دسترس است پروتکل FTP و IRC دارای helper برای نوشتن آن است . اگر نمیتوانند آنها را برای kernel بیابید باید به درخت patch -o -matic در iptable کاربر مراجعه کنید . این درخت دارای helperبیشتر است مانند n talk و پروتکل lt . 323 اگر آنها در درخت در دسترس نباشند یک سری گزینه دارید. شما میتوانید به منبع cvs در iptables مراجعه کنید و یا اینکه با لیت پست Netfilter - dnel تماس بگیرید . در غیر اینصورت طرح افزایشی وجود

شناخت کاربردی Iptable

ندارد و ابزار به جا می مانند و باید به ....RUSTY RUSSEL مراجعه کنید که در ضمیمه لیک و منابع دیگر آمده است .

conntrack helper با کامپایل آماری در kernel و یا به صورت مدل تولید می شود اگر آنها مدل باشند با دستور زیر باردار می شوند.

برنامه

ردیابی اتصال با NAT کار ندارند و بنابراین در صورتی که اتصالات NATing در دسترس باشند به مدل بیشتر نیاز است به عنوان مثال اگر می‌خواهید به NAT، FTP دست یابید به مدل NAT بیشتر نیاز است .

تمام NAT helper با ip nat آغاز شده اند و بنابراین NAT ، FTP به صورت in - nat ftpاست و مدل ICR به صورت in nat irc است . اینها تابع یک دستور نامگذاری هستند و بنابراین IRC به صورت ip.... است و FTP به صورت ip contrack است .