**پروژه درس امنیت شبکه**

**عنوان**

# بررسی IPTABLE

**استاد : آقای دکتر فقیه ایمانی**

**مرضیه پورحجتی ثابت**

**۹۴۰۱۱۳۸۳۲**

ترم دوم سال تحصیلی ۹۵-۹۴


بسم الله الرحمن الرحيم

**مقدمه:**

شبکه‌های کامپیوتری مبتنی بر گنو/لینوکس به طور فزاینده‌ای رو به گسترش هستند. در این بین یکی از عمده مشکلات، مشکلات مربوط به امنیت شبکه می‌باشد. Iptables یکی از کاراترین و انعطاف پذیرترین نرم‌افزارهای تولید شده در این زمینه است. با استفاده از این نرم افزار شما می توانید ترافیک ورودی و خروجی سیستم و شبکه را تحت کنترل خود درآورید.

قدرت این نرم افزار به حدی است که از لایه ۲ تا لایه ۷ را می تواند کنترل کند.

این نرم افزار از سه جدول تشکیل شده است:

۱- Filter

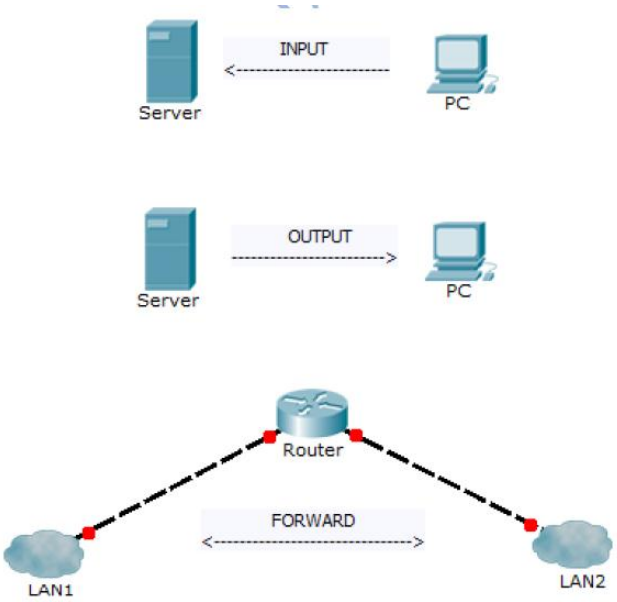
۲- NAT

۳- Mangle

**جدول FILTER:**

از سه قسمت تشکیل شده و هر قسمت آن را Chain می گویند:

| | |
|---|---|
| **INPUT** | زمانی استفاده می شود که مقصد packet سرور شما باشد |
| **OUTPUT** | زمانی استفاده می شود که مبدا packet سرور شما باشد |
| **FORWARD** | زمانی استفاده می شود که مقصد و مبدا packet هیچکدام سرور شما نباشد، به عبارتی در این حالت سرور شما بایستی حتما Router باشد. |



یک rule را می‌توان مجموعه دستوراتی در نظر گرفت که به firewall می‌گویند تا چگونه به بسته‌ها و اتصالات یک زنجیره‌ی خاص مجوز دهد. پس یک rule عبارتی است که کرنل برای پی بردن به اینکه با یک بسته چه باید بکند به آن نگاه می‌کند. اگر تمام ضوابط(match) درست بود، آنگاه دستورات target را اجرا می‌کند. نحوه‌ی استفاده از دستور iptables به صورت زیر است:

**iptables** [-t *table*] command [match] [target/jump]

با استفاده از انتخاب t- جدول مورد نظر را انتخاب می‌کنیم. اگر هیچ جدولی در این قسمت معین نگردد، به طور پیش فرض از جدول Filter استفاده می‌شود. سپس command را مشخص می‌کنیم. با استفاده از این قسمت به iptables می‌گوییم که با این rule چه باید بکند. به عنوان مثال می‌توان یک rule را اضافه، حذف و یا جایگزین کرد. انواع دستورات مختلف در ادامه تشریح خواهد شد.

match بخشی است که برای مشخص کردن خصوصیات خاص یک بسته که آن را از بقیه مجزا می‌کند، به کرنل فرستاده می‌شود. به طور مثال بسته‌های مربوط به یک شبکه‌ی خاص یا بسته هایی که از پروتکل خاصی استفاده می‌کنند و غیره .

در نهایت هدف بسته را مشخص می‌کنیم. در صورتی که تمام ضوابط مطابقت داشته باشند، به کرنل می‌گوییم که با بسته‌ی مورد نظر چه باید بکند. برای مثال می‌توانیم بسته را drop کنیم یا به جای دیگری بفرستیم یا به جدول دیگری ارسال کنیم.

قبل از آن که مثال ها را با هم بررسی می کنیم بایستی این نکته را مد نظر قرار دهیم که زمانی Rule های Firewall کار می کنند که سرویس iptables فعال باشد، به صورت زیر:

service iptables start

ذکر این نکته ضروریست که رفتار پیش فرض تمامی Chain ها در حالت ACCEPT است، به عبارت بهتر به صورت پیش فرض هیچ کدام از Chain ها بسته های مربوط به خود را DROP نمی کنند.

**دستورات:**

در این قسمت تمام دستورات ممکن را پوشش خواهیم داد.

| | |
|---|---|
| **-A, --append** | Command |
| **iptables -A INPUT ...** | Example |
| با استفاده از این دستور می‌توانید یک rule را به انتهای یک زنجیره بچسبانید. البته این را باید در نظر بگیرید که ruleها از بالا به پایین بررسی می‌شوند و اولین rule که مطابقت داشته باشد اعمال می‌شود. | Explanation |
| **-D, --delete** | Command |
| **iptables -D INPUT --dport 80 -j DROP, iptables -D INPUT 1** | Example |

| | |
|---|---|
| Explanation | بـا استفاده از این دستور می‌توانید یک rule را از زنجیره حذف کنید. این کار به دو طریق انجـــام می‌شود؛ می‌توان rule مورد نظر را عینا وارد کرد و یا اینکه شماره rule را به کار برد. Rule ها از بالا به پایین شـمـــاره گـــذاری می‌شـــوند. در مورد روش اول باید دقت داشته باشید که rule مورد نظر را درست وارد کنید، در غـــیـــر این صورت امکان دارد که rule دیگری را از زنجیره حذف کنید. |
| Command | **-R, --replace** |
| Example | **iptables -R INPUT 1 -s 192.168.0.1 -j DROP** |
| Explanation | بـــا اســـتفاده از ایـــن دستـــور می‌توانید یک rule را با rule دیگری جایگزین کنید. این کار مشابه حذف یک rule از زنجیره می‌باشد. |
| Command | **-I, --insert** |
| Example | **iptables -I INPUT 1 --dport 80 -j ACCEPT** |
| Explanation | بـا استفاده از دستور Insert می‌توانید تا یک rule را در مکان خاصی اضافه کنید. در این مورد هم باید تقدم ruleها را در نظر داشته باشید. |
| Command | **-L, --list** |
| Example | **iptables -L INPUT** |
| Explanation | با استفاده از ایـــن دستـــور می‌توانـیـد کـــل ruleهای یـک زنجیره را لیست بگیرید. توجه داشته باشید که اگر نام جدول را ذکر نکنید، به طور پیـــش فـرض از جدول filter لیست خواهد گرفت. خروجی این دستور تحت تاثیر انتخاب های مختلف دیگر از جمله **n**– و **v**– می باشد. |
| Command | **-F, --flush** |
| Example | **iptables -F INPUT** |
| Explanation | دستور **F**– بـــرای پـــاک کـــردن تـمـــام ruleهای یـــک زنجـــیره یا جدول و یا تمام iptables استفاده می‌شود. این دستور معادل پاک کردن تمام rule ها به صورت تک تک است. |
| Command | **-Z, --zero** |
| Example | **iptables -Z INPUT** |
| Explanation | ایـــن دستور به برنامه می‌گوید تا تمام شمارنده‌های یک زنجیره یا تمام آنها را پاک کند. اگر شما انتـــخـــاب **v**– را بـــه هـــمراه دســـتور **L**– به کار برید، می‌توانید شمارنده‌ها را در ابتدای هر فیلد مشاهده کنید. اگـر دستور **L**– و **Z**– را به همراه هم استفاده کنید، ابتدا تمام ruleها را برای شما لیست کرده و شمارنده‌های آنها را صفر می‌کند. |
| Command | **-N, --new-chain** |
| Example | **iptables -N allowed** |

| | |
|---|---|
| Explanation | با اســـتفـــاده از این دستور شما می‌توانید زنجیره‌های مخصوص را در جداول مشخصی تهیه کنید. در مثال بالا زنجیره‌ی **allowed** را در جـــدول filter ســاخته‌ایم. توجه داشته باشید که دو زنجیره‌ی هم نام نباید وجود داشته باشد. |
| Command | **-X, --delete-chain** |
| Example | **iptables -X allowed** |
| Explanation | با استفاده از ایـــن دستـــور می‌توانید یـــک زنجیره را از یک جدول پاک کنید. برای این کار نباید rule وجود داشته باشد تا به این زنجیره اشاره کند. اگـــر ایـــن دستـــور بـــدون هیچ انتخابی انجام شـــود، تمـــام زنجیره‌های جدول را به جز آنهاییکه مخصوص خود جـــدول هستند، پاک خواهد شد. |
| Command | **-P, --policy** |
| Example | **iptables -P INPUT DROP** |
| Explanation | با استفاده از این دستـــور می‌توانید به کرنل بگویید که برای یک زنجیره، هدف یا policy مخصوصی را به صورت پیش فرض در نـظـــر بـگیرد. تمام بسته هایی که با هیچ کدام از ruleهای موجود در زنجیره match نشوند مجبـــور خـواهند بود که از این policy استفاده کنند. Policyهای مجاز برای این قسمت **DROP** و **ACCEPT** هستند. |
| Command | **-E, --rename-chain** |
| Example | **iptables -E allowed disallowed** |
| Explanation | بـــا استفاده از ایـــن دستـــور شما می‌توانید تا نام یک زنجیره را تغییر دهید. همانطور که می بینید در مثال بالا زنـجـــیـــره‌ی allowed به disallowed تغییر نام یافته است. دقت داشته باشید ایـــن عمـل در عملکرد اصلی جدول هیچ تاثیری نخواهد داشت و فقط یک کار تزئینی محسوب می‌شود. |

بـه غیر از مواردی که می‌خواهید از help داخلی iptables استفاده کنید یا اینکه نسخه‌ی iptables را به دست آورید، باید همیشه یـــک دستور کـــامل را وارد کنید.برای یافتن شماره‌ی نسخه‌ی iptables باید از انتخاب **v**- و برای استفاده از help داخلی iptables باید از انتخاب **h**- استفاده کنید.

**مثال ۱)**

iptables -A INPUT –j DROP

این اسکریپت که در یک خط فرمان تایپ می شود از پنج قسمت تشکیل شده است:

| | |
|---|---|
| **iptables** | زمانی استفاده می شود که بخواهیم به تنظیمات Firewall دسترسی داشته باشیم .حتما بایستی حروف آن کوچک تایپشوند. |
| **-A** | زمانی استفاده می شود که بخواهیم به تنظیمات قبل یک قانون (Rule) اضافه کنیم که مخفف Append می باشد. حتما حرف A بایستی بزرگ تایپ شود. |
| **INPUT** | با توجه به توضیحات ارائه شده می توان دریافت که این Rule زمانی استفاده می شود که مقصد، سیستم جاری شما باشد .حتما تمامی حروف آن بایستی بزرگ تایپ شوند. |
| **-j** | تصمیم نهایی در مورد یک قانون به وسیله j گرفته می شود که مخفف jump می باشد .حتما بایستی j کوچک تایپ شود. |
| **DROP** | زمانی استفاده می شود که بخواهیم بسته ای را نابود کنیم و به فرستنده جوابی نفرستیم |

پس در اینجا در میابیم که به وسیله این اسکریپت تمامی بسته ها که به سمت Server ما روانه می شوند از بین می روند.

**نکته ) اگر در این اسکریپت به جای A– از D– استفاده کنیم، Rule مورد نظر پاک می شود .حرف D بایستی بزرگ تایپ شود D مخفف Delete می باشد.**

**نکته ) اگر بخواهیم تمامی Rule های مربوط به INPUT را پاک کنیم از دستور زیر استفاده می کنیم:**

**iptables –F INPUT**

**که در اینجا حرف F بایستی بزرگ تایپ شود و مخفف FLUSH می باشد.**

**مثال ۲)**

مثال ۱ را کمی تخصصی تر می کنیم به گونه ای که تمامی بسته هایی که از سمت مبدا 192.168.1.50 فرستاده می شوند نابود شوند:

iptables –A INPUT –s 192.168.1.50 –j DROP

این اسکریپت نسبت به مثال قبل دو قسمت جدید دارد:

| | |
|---|---|
| **-S** | زمانی استفاده می شود که بخواهیم مبدا بسته را مشخص کنیم .که مخفف source می باشد .و حرف s کوچک تایپ می شود. |
| **192.168.1.50** | IPفرستنده بسته می باشد. |

**نکته )تمامی Rule های موجود در همه Chain ها به وسیله دستور زیر قابل رؤیت می باشد:**

**iptables –nL**

**حرف n بایستی حتما کوچک و حرف L باستی حتما بزرگ تایپ شود، که در اینجا n مخفف numerical است یعنی به دور از هر گونه اسم DNS و L مخفف list می باشد.**

**ذکر این نکته ضروریست که اگر این دستور را بدون n بزنید نتیجه آن کندتر ظاهر می شود زیرا که به دنبال Resolve نمودن تمامی IP ها از سمت DNS یا فایل /etc/hosts می رود.**

**مثال ۳)**

مثال ۲ را کمی تخصصی تر می کنیم به گونه ای که اگر از سمت مبدا 192.168.1.50 به سمت سرور ما ping شود آن را DROP کند:

iptables –A INPUT –s 192.168.1.50 –p icmp –j DROP

| | |
|---|---|
| **-p** | زمانی استفاده می شود که بخواهیم پروتکل بسته را مشخص کنیم که مخفف protocol است و حتما بایستی حرف p کوچک تایپ شود. |
| **icmp** | اسم پروتکل را مشخص می کند .اسم پروتکل بایستی با حروف کوچک تایپ شود. |

همانطور که میدانیم ping از پروتکل icmp استفاده می کند.

**مثال ۴)**

مثال شماره ۳ را کمی تخصصی تر می کنیم اما قبل از آن بیایید یک بحث کوتاهی را با هم داشته باشیم.

همانطور که میدانیم icmp ، type های مختلفی دارند

اگر icmp مربوط به مبدا 192.168.1.50 ، DROP شود بسته های echo-request آن از بین میروند .خوب تا اینجا مطلوب ما می باشد اما در صورتی که بخواهیم به 192.168.1.50 ، ping کنیم بسته های echo-reply آن از بین می روند که این قسمت ماجرا مطلوب ما نیست .به عبارت بهتر ما می خواهیم به 192.168.1.50 ، ping کنیم اما 192.168.1.50 نتواند به ما ping کند.

مثال جاری این مشکل را حل می کند:

iptables -A INPUT -p icmp --icmp-type echo-request –j DROP

| | |
|---|---|
| **--icmp-type** | زمانی استفاده می شود که بخواهیم نوع پیام icmp را مشخص کنیم. |
| **echo-request** | ارسال پیام icmp می باشد. |

**مثال ۵)**

برای block کردن یک سرویس خاص بایستی به دو نکته توجه کنیم:

۱- Protocol

۲- Port number

به طور مثال می خواهیم سرویس web را برای 192.168.1.50 ببندیم .همانطور که میدانیم web از پروتکل tcp و شماره پورت ۸۰ استفاده می کند .بنابراین این مساله را به صورت زیر حل می کنیم:

iptables –A INPUT –s 192.168.1.50 –p tcp --dport 80 –j DROP

| | |
|---|---|
| **--dport** | زمانی استفاده می شود که بخواهیم پورت مقصد را مشخص کنیم و مخفف destination port می باشد و تمامی حروف آن بایستی کوچک تایپ شوند. |

**مثال ۶)**

بیایید سری به لایه 2 بزنیم، فرض کنید می خواهیم یک MAC-Address را block کنیم که نتواند به سرور ما telnet بزند.

iptables –A INPUT –m mac --mac-source 00:52:76:D0:00:01 –p tcp --dport 23 –j DROP

| | |
|---|---|
| **-m** | زمانی استفاده می شود که بخواهیم از یکی از option های iptablesاستفاده کنیم، که در این مثال از mac استفاده کردیم، مخفف match می باشد و حتما حرف m بایستی کوچک نوشته شود. |
| **--mac-source** | آدرس MAC مبدأ را بعد از آن مینویسیم، حتما تمامی حروف آن بایستی کوچک تایپ شوند. |

**مثال ۷)**

فرض کنید که می خواهیم به سیستم هایی که از طریق eth1 به سرور ما وصل شده اند، سرویس DNS را ارائه ندهیم.

iptables –A INPUT –p udp –dport 53 –i eth1 –j DROP

| | |
|---|---|
| **-i** | کارت شبکه ورودی را بعد از آن مشخص می کنیم که مخفف input interfaceمی باشد و حتما حرف i بایستی کوچک تایپ شود. |

**مثال ۸)**

فرض کنید که می خواهیم تمامی درخواست های مربوط به سرویس web را در سرور ثبت رخداد ( log ) کنیم.

iptables –A INPUT –p tcp --dport 80 –j LOG --log-prefix “WebRequest”

به وسیله این تکنیک تمامی درخواست های web در فایل /var/log/messages ثبت رخداد می شوند و شامل خطوطی می شوند که ابتدای آن WebRequest می باشد.

**مثال ۹)**

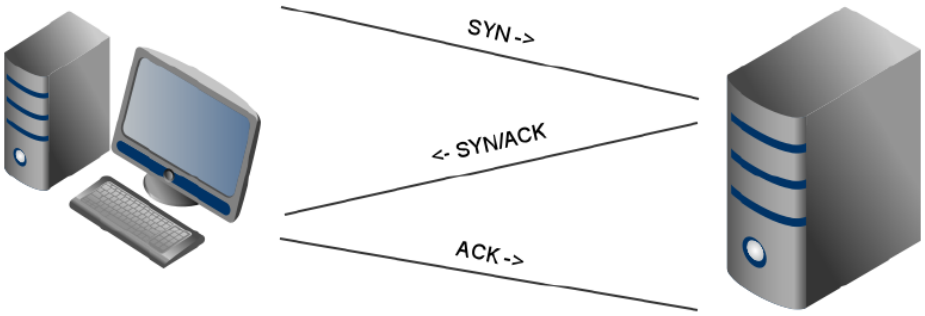
فرض کنید که می خواهیم سرویس web و telnet را برای یک IP خاص ببندیم.

iptables –A INPUT –p tcp –m multiport --dports 80,23 –s 192.168.1.50 –j DROP

| | |
|---|---|
| **multiport** | این بار بعد از m– از این option استفاده نمودیم با ذکر این نکته که تمامی حروف آن بایستی با حروف کوچک تایپ شوند. |
| **--dports** | به حرف s انتهای dports دقت کنید، تمامی حروف حتما بایستی کوچک تایپ شوند. |

**مثال ۱۰)**

یکی از سریعترین راهکارهای Block کردن در Firewall ها این است که به محض فرستادن یک Connection Request از سمت یک IP خاص ( SYN ) جلوی آن را بگیریم .به شکل زیر دقت کنید:




TCP Connection بعد از سه مرحله ذکر شده در بالا ایجاد می شود، سرعت را می توانیم بالا ببریم بدین صورت که Server دیگر درخواست های SYN/ACK و ACK را پردازش نکند.

فرض کنید که می خواهیم مثال شماره 5 را به گونه ای سریعتر بازنویسی کنیم..

iptables -A INPUT -p tcp --syn --dport 23 -j DROP

| | |
|---|---|
| **--syn** | درخواست SYN به وسیله syn-- مشخص می شود، که تمامی حروف آن بایستی کوچک تایپ شوند. |

REJECT همانند DROP یک ارتباط را Block می‌کند با این تفاوت که پیغامی را به سمت فرستنده ارسال می‌کند که محتوای این پیغام درون Rule مشخص می‌شود. این در حالی است که DROP صرفاً بسته را از بین می‌برد و هیچ پیغامی را به سمت فرستنده ارسال نمی‌کند.

پیغامهایی که در REJECT مشخص میشوند، به شرح زیر میباشد:

| | |
|---|---|
| icmp-port-unreachable | زمانی استفاده می‌شود که بخواهیم به فرستنده بگوییم پورت مورد نظر بر روی سرور باز نمی‌باشید و نرم افزاری روی سرور برای Listen کردن آن وجود ندارد. |
| icmp-net-unreachable | زمانی به کار میرود که بخواهیم به فرستنده بگوییم، شبکه IP مقصد در جدول مسیریابی سرور نمی‌باشد به عباتبی Router هیچ مسیری را برای ارتباط با مقصد پیدا نمیکند. |
| icmp-host-unreachable | زمانی به کار میرود که به فرستنده اعلام کنیم بسته را به سمت مقصد ارسال کرده ایم ولی جوابی را دریافت ننموده ایم. |
| icmp-proto-unreachable | زمانی به کار می‌رود که به فرستنده اعلام کنیم پروتکل مورد نظر را سرور پشتیبانی نمی‌کند. |
| icmp-net-prohibited | به منظور Block کردن یک شبکه استفاده می‌شود. به عبارتی به فرستنده می‌گوییم که شبکه مقصد مورد نظر Block شده است. |
| icmp-host-prohibited | به منظور Block کردن یک host استفاده می‌شود .به عبارتی به فرستنده می‌گوییم که IP مقصد مورد نظر Block شده است. |
| tcp-reset | برای فرستادن بسته RESET در جواب بستهSYN به منظور جلوگیری از connection استفاده میشود. |

همانطور که میدانیم، ICMP مخفف Internet Control Message Protocol می‌باشد، این پروتکل دقیقاً همانند یک فازمتر برای شبکه عمل میکند که توسط آن می‌توان شبکه را Troubleshoot کرد.

**مثال ۱۱)**

فرض کنید میخواهیم بسته هایی را که از مبدأ 10.10.10.10 به سمت پورت 22 پروتکل TCP (SSH) سرور ما می‌آیند بدین صورت ببندیم که به فرستنده اعلام عدم وجود سرویس را بدهیم و یا به عبارتی پیغام icmp-port-unreachable را بفرستیم.

iptables –A INPUT –s 10.10.10.2 –p tcp --dport 22 –j REJECT --reject-with icmp-port-unreachable

| | |
|---|---|
| **REJECT** | بعد از j– به کار میرود و زمانی استفاده میشود که بخواهیم از option های REJECT استفاده کنیم. بایستی تمامی حروف آن بزرگ تایپ شوند. |
| **--reject-with** | پیغامی را که میخواهیم به فرستنده ارسال شود، بعد از این عبارت می نویسیم .تمامی حروف این عبارت بایستی کوچک تایپ شوند. |

برای تست این Rule می توانیم از hping استفاده کنیم.

اگر روی کامپیوتر 10.10.10.2 دستور زیر را وارد کنیم:

hping 10.10.10.1 -S -V -p 22 -c 3

بعد از زدن این دستور از سمت کامپیوتر 10.10.10.2 با پیغام زیر روبرو میشویم:

ICMP Port Unreachable from ip=10.10.10.1 name=UNKNOWN
ICMP Port Unreachable from ip=10.10.10.1 name=UNKNOWN
ICMP Port Unreachable from ip=10.10.10.1 name=UNKNOWN

که جواب مورد انتظار ما میباشد.

**مثال ۱۲)**

به اسکریپت زیر دقت کنید:

iptables –A INPUT –p icmp –s 10.10.10.10 –j REJECT --reject-with icmp-port-unreachable

اگر کامپیوتر 10.10.10.10 به 10.10.10.1 ، ping کند با پیغام زیر مواجه میشود:

Pinging 10.10.10.1 with 32 bytes of data:
Reply from 10.10.10.1: Destination port unreachable.
Reply from 10.10.10.1: Destination port unreachable.
Reply from 10.10.10.1: Destination port unreachable.
Reply from 10.10.10.1: Destination port unreachable.

**مثال ۱۳)**

اگر اسکریپت زیر را در سرور وارد کنیم:

iptables –A INPUT –p tcp --dport 22 –s 10.10.10.2 –j DROP

می‌دانیم که تمامی بسته‌های پورت 22 را که از مبدأ 10.10.10.2 می‌آیند را نابود می‌کند .اما نکته‌ای که باید به آن توجه کرد این است که به راحتی می‌توان از طریق کامپیوتر 10.10.10.2 فهمید که پورت 22 باز است ولی برای یک سری مشتری خاص.

بدین صورت که اگر از یک پورت اسکنر مانند nmap استفاده کنیم با خروجی زیر مواجه میشویم:

```
nmap -p 22 10.10.10.1
PORT STATE SERVICE
22/tcp filtered ssh
```

اما اگر اسکریپت بالا را به صورت زیر تغییر دهیم، دیگر از سمت کامپیوترهای فیلتر شده نمی‌توان به وجود سرویس ssh (پورت22 ) در سرور پی برد:

```
iptables –A INPUT –p tcp --dport 22 –s 10.10.10.2 –j REJECT --reject-with tcp-reset
```

خروجی port scanner :

```
nmap -p 22 10.10.10.1
PORT STATE SERVICE
22/tcp closed ssh
```

**مثال ۱۴)**

در صورتی که بخواهیم یک سرویس را در یک بازه زمانی ببندیم، از تکنیک زیر استفاده می‌کنیم:

```
iptables -A INPUT -p tcp --dport 22 -m time --timestart 09:00 --timestop 18:00 –j REJECT --reject-with tcp-reset
```

در اسکریپت بالا سرویس SSH از ساعت ۹:۰۰ تا ۱۸:۰۰ در دسترس نمی‌باشد.

| | |
|---|---|
| **--timestart** | زمان شروع را مشخص میکند؛ تمامی حروف بایستی کوچک تایپ شوند. |
| **--timestop** | زمان پایان را مشخص میکند؛ تمامی حروف بایستی کوچک تایپ شوند. |

**مثال ۱۵)**

در صورتی که بخواهیم یک سرویس را در روزهای خاصی از هفته ببندیم، از تکنیک زیر استفاده می‌کنیم:

```
iptables –A INPUT –p tcp --dport 22 –m time --weekdays Sa,Su –j REJECT --reject-with tcp- reset
```

در اسکریپت بالا در روزهای شنبه و یکشنبه نمی‌توان از سرویس SSH استفاده کرد.

| | |
|---|---|
| **--weekdays** | روزهای هفته را می‌توان توسط این option مشخص کرد؛ تمامی حروف این option بایستی کوچک تایپ شوند. |

**مثال ۱۶)**

اگر بخواهیم یک سرویس را در یک بازه تاریخی ببندیم، از تکنیک زیر استفاده می‌کنیم:

```
iptables –A INPUT –p tcp –dport 22 –m time --datestart 2014-01-05 --datestop 2014-02-05 –j REJECT –reject-with tcp-reset
```

در مثال بالا سرویس SSH از روز پنجم ماه اول سال تا روز پنجم ماه دوم سال Block می‌شود.

| | |
|---|---|
| **--datestart** | تاریخ شروع اعمال Rule را مشخص میکند؛ تمامی حروف آن باید کوچک نوشته شوند. |
| **--datestop** | تاریخ انقضای Rule را مشص میکند؛ تمامی حروف آن بایستی کوچک نوشته شوند. |

**مثال ۱۷)**

در صورتی که بخواهیم سرویس web در روزهای خاصی از ماه block باشد، از تکنیک زیر استفاده می‌کنیم:

iptables –A INPUT –p tcp --dport 80 –m time --monthdays 5,9,13,17 –j REJECT --reject-with tcp-reset

معنای اسکریپت بالا بدین معناست که سرویس web فقط در روزهای 5 و 9 و 13 و 17 قابل استفاده نمی‌باشد.

| | |
|---|---|
| **--monthdays** | روزهای اعمال Rule را مشخص می‌کند، تمامی حروف آن بایستی به صورت کوچک نوشته شوند. |

**نکته ) در iptables علامت ! به معنای NOT میباشد .به طور مثال اسکریپت زیر:**

**iptables –A INPUT –p tcp --dport 80 –m time ! --monthdays 5,9,13,17 –j REJECT --reject- with tcp-reset**

**به این معنا است که سرویس web فقط در روزهای 5 و 9 و 13 و 17 هر ماه قابل استفاده میباشد.**

**مثال ۱۸ )**

فرض کنید میخواهیم بسته‌هایی که به سمت سرور ما فرستاده میشوند، دارای محدودیّت حجمی باشند:

iptables –A INPUT –p icmp –m length ! --length 0:1500 –j REJECT --reject-with icmp-host-unreachable

در اسکریپت بالا مشخص میکنیم در صورتی که طول بسته‌های ارسالی پروتکل ICMP از ۱۵۰۰ بایت بیشتر باشد، آن بسته را REJECT کند.

| | |
|---|---|
| **length** | به منظور اعمال محدودیت حجمی استفاده میشود و تمامی حروف آن باید کوچک تایپ شوند. |
| **--length** | بازه حجمی محدود شده توسط این option مشخص میشود و تمامی حروف آن باید کوچک تایپ شوند. |

**مثال ۱۹ )**

در صورتی که بخواهیم یک سری IP خاص را با هم برای یک Rule مشخص کنیم، از تکنیک زیر استفاده می‌کنیم:

iptables –A INPUT –p icmp –m iprange --src-range 10.10.10.12-10.10.10.17 –j REJECT

در اسکریپت بالا معیّن میکند که اگر IP مربوط به یک کامپیوتر 10.10.10.12 الی 10.10.10.17 باشد، قادر به ping کردن سرور ما نیست.

| | |
|---|---|
| **Iprange** | به منظور معرفی یک محدوده IP به کار می‌رود؛ تمامی حروف آن بایستی کوچک تایپ شوند. |
| **--src-range** | بازه IP را بعد از این option وارد می‌کنیم؛ تمامی حروف آن باید کوچک تایپ شوند. |

نکته )در صورتی که REJECT بدون reject-with-- به کار رود، به صورت پیش فرض پیغام icmp-port-unreachable را به سمت فرستنده می‌فرستد.

در این قسمت میخواهیم شما را با چند نکته جهت مدیریت هر چه بهتر iptables آشنا کنیم:

**نکته ۱)**

تمامی Rule های iptables باید در مسیر /etc/sysconfig/iptables/ ذخیره شوند تا بعد از ریسرتار شدن سرویس iptables و یا حتی ریستار شدن سرور، هیچکدام از Rule ها پاک نشوند. Rule ها را به صورت زیر ذخیره می‌کنیم:

```
iptables-save > /etc/sysconfig/iptables
```

**نکته ۲)**

اگر بخواهیم Rule هایی را که از قبل در یک فایل ذخیره کرده‌ایم روی سرور restore کنیم، از تکنیک زیر استفاده می‌کنیم:

```
iptables-restore < /server/backup/iptables-rules
```

بدین صورت تمامی Rule های موجود در فایل iptables-rules روی سرور Restore میشوند و می‌توان توسط دستور ذکر شده در نکته یک آنها را به صورت دائمی ذخیره کرد.

**نکته ۳)**

سرویس iptables تمامی Rule های هر chain را از ابتدا شروع به خواندن میکند، اگر با یکی از Rule ها منطبق شود آن را اجرا میکند و از خواندن بقیه Rule ها صرف نظر میکند .در صورتی که با هیچ کدام از Rule ها منطبق نشود به سراغ Policy از قبل تعریف شده خود میرود که به صور پیش فرض Accept میباشد.

اولویّت Rule ها بر اساس ترتیب ورود آنها مشخص میشود، در صورتی که از A- استفاده کنیم یک به انتهای Chain اضافه کرده‌ایم. و در صورتی که از I– استفاده کنیم یک Rule به ابتدای Chain اضافه کرده‌ایم.

به طور مثال اگر دو دستور زیر را یکی پس از دیگری وارد کنیم:

```
iptables –A INPUT –j DROP
iptables –A INPUT –p icmp –j ACCEPT
```

همانطور که میدانیم Rule اول تمامی بسته‌هایی که به سمت سرور می‌آیند را Block می‌کند و Rule دوم صرفاً به بسته‌های مربوط به پروتکل ICMP اجازه پردازش می‌دهد. اما سؤال اینجاست که اگر از کامپیوتر دیگر به این سرور ping کنیم آیا جواب( Reply) دریافت می‌کنیم یا خیر؟ جواب خیر می‌باشد زیرا که Rule اول اولویّت بالاتری دارد؛ پس هر بسته‌ای با آن منطبق می‌شود و به سراغ Rule های دیگر نمی‌رود.

حال به مثال زیر دقت کنید:

```
iptables –A INPUT –p icmp –j ACCEPT
```

iptables –A INPUT –j DROP

در این قسمت اگر از کامپیوتر دیگر به سرور ping کنیم جواب ( Reply ) دریافت می کنیم، زیرا که با Rule ابتدایی منطبق می‌شود و به سراغ Rule های دیگر نمی‌رود .اگر از کامپیوتر دیگر به سرور SSH بزنیم تمامی بسته‌های ما Block می‌شوند زیرا که با Rule ابتدایی منطبق نمی‌شود؛ بنابراین به سراغ Rule دوم می‌رود و تمامی بسته‌های SSH یا هر پروتکل دیگر به جز ICMP از بین می‌روند .حال فرض کنیم که در این حالت دستور زیر را وارد کنیم:

iptables –I INPUT –p tcp --dport 22 –j ACCEPT

این Rule به ابتدای مجموعه Rule ها وارد میشود و در این حالت تمامی بسته‌های مربروط به پروتکل SSH پردازش میشوند.

ذکر این نکته ضروری است که اگر بسته با هیچ کدام از Rule ها منطبق نشود، در صورتی که Policy پیش فرض را تغییر نداده باشیم آن بسته پذیرفته ( Accept ) میگردد.

**نکته ۴)**

در صورتی که بخواهیم کلیه Rule های مربوط به یک Table را مانیتور کنیم، از دستور زیر استفاده میکنیم:

iptables –nL –t <Table-Name>

به طور مثال اگر بخواهیم Rule های موجود در Table مربوط به Filter را ببینیم از دستور زیر استفاده میکنیم:

iptables –nL –t filter

که البته به صورت پیش فرض اگر t filter– را وارد نکنیم Rule های مربوط به filter را به ما نشان میدهد.

به عنوان مثالی دیگر اگر بخواهیم Rule های Table مربوط به NAT را ببینیم از دستور زیر استفاده میکنیم:

iptables –nL –t nat

**نکته ۵)**

در صورتی که بخواهیم Rule های مربوط به یک Table را به همراه شماره اولویّت آنها مشاهده کنیم، از دستور

زیر استفاده میکنیم:

iptables –nL –t <Table-Name> --line-numbers

به طور مثال اگر بخواهیم این کار را برای Rule های NAT انجام دهیم، از دستور زیر استفاده میکنیم:

iptables –nL –t nat --line-numbers

همانطور که میدانیم در صورت مشخص نکردن Table به صور پیش فرض filter بررسی میشود:

iptables –nL --line-numbers

**نکته ۶)**

در صورتی که بخواهیم Policy پیش فرض یک Chain را تغییر دهیم، از تکنیک زیر استفاده میکنیم:

iptables –P <Table-Name> <Action-Name>

به طور مثال:

iptables –P INPUT DROP
iptables –P FORWARD DROP
iptables –P OUTPUT ACCEPT

دستور اول و دوم به ترتیب Policy پیش فرض INPUT و FORWARD را DROP تنظیم میکند.

دستور سوم Policy پیش فرض OUTPUT را ACCEPT تنظیم میکند.

**نکته ۷)**

در صورتی که بخواهیم یک Rule را در محل خاصی از نظر اولویّت در یک Chain وارد کنیم، از تکنیک زیر استفاده میکنیم:
iptables –I INPUT 5 –p tcp --dport 22 –s 10.10.10.2 –j DROP

همانطور که مشاهده میکنید بعد از I INPUT– عدد 5 آمده است که به این معنا است که این Rule را در اولویّت شماره پنج Chainمربوط به INPUT قرار دهد .به طوری که با زدن دستور زیر:
iptables –nL --line-numbers

آن Rule را در اولویّت پنجم نمایش میدهد.

**نکته ۸)**

در صورتی که بخواهیم یک Rule را با توجه به شماره اولویّت آن پاک کنیم، از دستور زیر استفاده میکنیم:
iptables –D <Chain-Name> <Line-Number>
به طور مثال می خواهیم Rule شماره دو Chain مربوط به INPUT را پاک کنیم:
iptables –D INPUT 2
یا اینکه می خواهیم Rule شماره پنج Chain مربوط به FORWARD را پاک کنیم:
iptables –D FORWARD 5

**نکته ۹)**

به منظور Document نمود Rule ها جهت توضیح دادن علت وارد کرد یک Rule ، از تکنیک زیر استفاده میکنیم:
iptables –A INPUT –p tcp --dport 22 –m comment --comment “Blocking SSH” –j DROP

در مثال بالا می‌بینیم که برای یک Rule توانستیم یک توضیح معنا داری را وارد کنیم .در صورتی که Administrator دستور زیر را وارد کند:

iptables –nL

توضیح مربوط به هر Rule را میتواند مشاهده کند.

**نکته ۱۰)**

در صورتی که تصمیم به مانیتور کردن تعداد بسته‌ها و همچنین حجم داده‌هایی که با Rule های موجود منطبق شده باشند را داشته باشیم، از تکنیک زیر استفاده میکنیم:

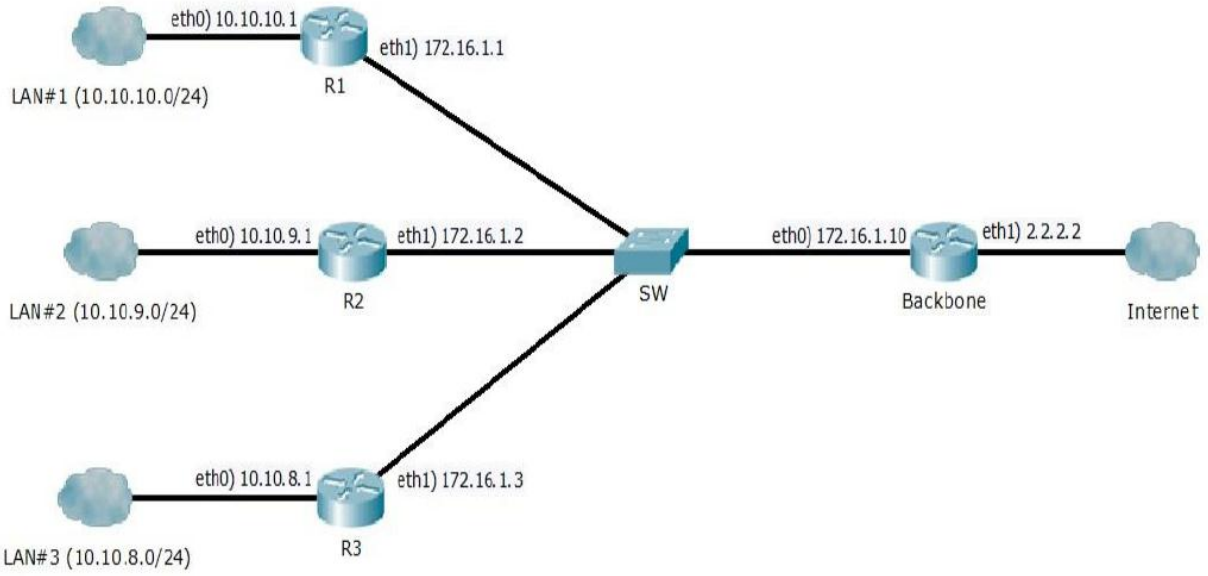
iptables –nvL

با استفاده از v– دو ستون اضافه تر ایجاد می‌شود که تعداد بسته‌های منطبق شده با Rule در یک ستون و مقدار بایت منطبق شده با Rule در ستونی دیگر نمایش داده میشود.

**FORWARD Chain**:

FORWARD Chain زمانی استفاده میشود که سرور شما نقش یک مسیر یاب ( Router ) را بازی کند. در علم شبکه به منتقل کردن بسته از سمت مبدا به طرف مقصد را Forwarding میگو ند.

قبل از ان که به سراغ مثالها برویم توسط یک دیاگرام بستر آزمایشی این مستند را مشخص می‌کنیم:




**مثال ۲۰)**

اگر بخواهیم مسیریاب R1 را به گونه‌ای پیکربندی نماییم که به نیمه دوم LAN#1 اجازه استفاده از اینترنت را ندهیم از تکنیک زیر استفاده می‌کنیم:

iptables –A FORWARD –s 10.10.10.128/25 –j DROP

| | |
|---|---|
| **FORWARD** | زمانی که بخواهیم عبور بسته‌ها از مبدا به سمت مقصد را کنترل کنیم از این Option استفاده می‌کنیم، تمامی حروف آن بایستی بزرگ تایپ شوند. |

**مثال ۲۱)**

فرض کنید که میخواهیم مسیریاب R2 را طوری پیکربندی نماییم که شبکه LAN#2 نتواند هیچ گونه ارتباطی را با LAN#3 برقرار کند.

iptables –A FORWARD –s 10.10.9.0/24 –d 10.10.8.0/24 –j DROP

**مثال ۲۲)**

اگر بخواهیم مسیریاب R3 را به نحوی پیکربندی نما ییم که اجازه ندهیم کسی به بیرون از شبکه VPN از نوع PPTP بزند از تکنیک زیر استفاده میکنیم:

iptables –A FORWARD –p tcp --dport 1723 –j DROP

**مثال ۲۳)**

در صورتی که بخواهیم تعداد کانکشنهای که به سرویس وب فرستاده میشود را محدود کنیم از تکنیک زیر استفاده میکنیم، به طور مثال فرض کنید که در LAN#2 به تعداد ۲۰۰ کامپیوتر موجود میباشد به ازای هر کامپیوتر 3 connection به سمت وب در آن واحد فرستاده شود جمعاً ۶۰۰ Connection اجازه عبور به سمت سرویس وب را دارند. بنابراین:

iptables –A FORWARD –s 10.10.9.0/24 –p tcp --dport 80 –m connlimit --connlimit-above 600 –j REJECT --reject-with tcp-reset

**مثال ۲۴)**

اگر بخواهیم مسیریاب Backbone را به صورتی پیکربندی کنیم که بسته‌های بزرگتر از ۱۵۰۰ بایت اجازه خارج شدن از شبکه را نداشته باشند از تکنیک زیر استفاده می‌کنیم:

iptables –A FORWARD –m length ! --length 0:1500 –j REJECT --reject-with tcp-reset

**مثال ۲۵)**

اگر بخواهیم مسیریاب Backbone را به صورتی پیکربندی کنیم که کل شبکه صرفاً در ساعتهای اداری اجازه استفاده از اینترنت را داشته باشند از تکنیک زیر استفاده میکنیم:

iptables –A FORWARD –m time --timestart 16:01 --timestop 7:29 –j REJECT

**مثال ۲۶)**

اگر بخواهیم R3 را به صورتی پیکربندی نما یم که LAN#2 اجازه استفاده از منابع LAN#3 را نداشته باشد از تکنیک زیر استفاده میکنیم:

iptables –A FORWARD –d 10.10.8.0/24 –i eth 1 –s 10.10.9.0/24 –j DROP

**مثال ۲۷)**

اگر بخواهیم کلیه فعالیتهای یک کامپیوتر را در نظر بگیریم از تکنیک زیر استفاده میکنیم، به طور مثال درخواستهای وب کامپیوتر 10.10.10.5 در LAN#1 را از طریق R1 در نظر بگیریم:

iptables –A FORWARD –s 10.10.10.5 –p tcp --dport 80 –j LOG --log-prefix "10.5 Activities"

**مثال ۲۸)**

در صورتی که بخواهیم مسیریاب Backbone را طوری پیکربندی کنیم که از سمت اینترنت نتوان هیچ گونه Connection به سمت شبکه داخلی داشت، از تکنیک زیر استفاده میکنیم:

iptables –A FORWARD –m state --state NEW –i eth1 –j DROP

| | |
|---|---|
| **state** | یکی از Option ها ی است که برای اعلام وضعیت Connection استفاده میشود.تمامی حروف آن بایستی کوچک تایپ شوند. |
| **--state** | وضعیت Connection را بعد از این option مشخص میکنیم، تمامی حروف آن بایستی کوچک تایپ شوند. |

**مثال ۲۹)**

در صورتی که بخواهیم شبکه LAN#2 را طوری محدود نماییم که صرفاً کامپیوترهای که Windows بر روی آنها نصب میباشد بتوانند از اینترنت استفاده کنند، از تکنیک زیر استفاده میکنیم:

iptables -A FORWARD -m ttl ! --ttl-eq 128 -j DROP

| | |
|---|---|
| **ttl** | زمانی استفاده میشود که بخواهیم محدودت را بر اساس TTL ایجاد نماییم، تمامی حروف آن بایستی کوچک تایپ شوند. |
| **--ttl-eq** | بعد از این Option مقدار TTL مجاز را قرار میدهیم، تمامی حروف آن بایستی کوچک تایپ شوند. |

## جدول NAT :

جهت پیاده سازی NAT در سیستم عامل لینوکس دو روش موجود است:

**روش اول SNAT )**

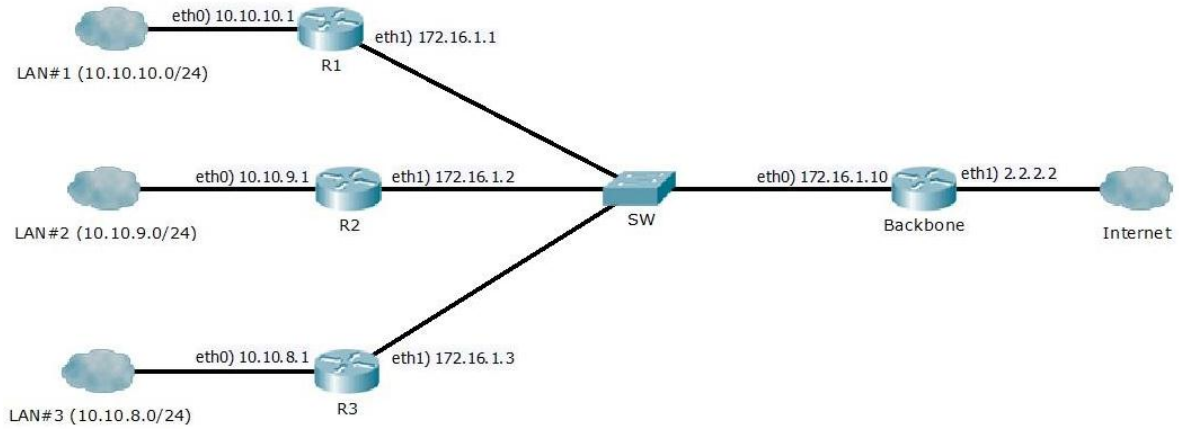
SNATمخف Source NAT می‌باشد و بدین معناست که یکی از Header های Source دچار تغییر شوند به طور مثال: Source IP و Source port .

ذکر این نکته ضروری است که در لینوکس به فرآیند ( PAT (Port Address Translation ، MASQUERADE می‌گویند.

**روش دوم DNAT )**

DNATمخفف Destination NAT می‌باشد و بدین معناست که یکی از Header ها Destination دچار تغییر شوند. به طور مثال Destination IPوDestination port

ابتدا بستر مربوط به مثال ها را مشخص می کنیم:




**مثال ۱)**

فرض کنید که می خواهیم تمامی کامپیوترهای موجود در LAN#1 در صورت استفاده از شبکه خارجی به NAT 172.16.1.1 شوند.

iptables –t nat –A POSTROUTING –o eth1 –j SNAT --to 172.16.1.1

| | |
|---|---|
| **-t nat** | جهت مشخص کردن Table از-t استفاده می کنیم که حرف t بایستی کوچک تایپ شود، عبارت nat مشخصه Table مربوط به NAT می باشد که تمامی حروف آن بایستی کوچک تایپ شوند. |
| **POSTROUTING** | زمانی که از SNAT استفاده می کنیم بایستی از زنجیره POSTROUTING (Chain) استفاده کنید و حتما تمامی حروف آن بزرگ تایپ شوند. |

| | |
|---|---|
| **–o** | جهت مشخص کردن کارت شبکه خروجی بسته اطلاعاتی به کار می رود، حتما با حروف کوچک تایپ شود. |
| **SNAT** | زمانی که از Source NAT استفاده می کنیم این عبارت را بعد از j– وارد می کنیم و بایستی حروف آن بزرگ تایپ شوند. |
| **--to** | جهت مشخص کردن یک IP استفاده می شود که می خواهیم بسته های اطلاعاتی مورد نظر به آن NAT شوند، حتماً تمامی حروف آن کوچک تایپ شوند. |

**مثال ۲)**

در صورتی که بخواهیم نیمه اول شبکه LAN#2 از طریق R2 به صورت NAT بسته های آنها به بیرون فرستاده شوند و نیمه دیگر بسته هایشان به صورت Route به بیرون فرستاده شوند، از تکنیک زیر استفاده می کنیم:

iptables –A POSTROUTING –s 10.10.9.0/25 –o eth1 –j SNAT --to 172.16.1.2

**مثال ۳)**

در صورتی که بخواهیم مسریاب Backbone را به صورتی پیکربندی کنیم که اگر بسته ای بخواهد از مسیریاب خارج شود،یکی از IP های 2.2.2.3-2.2.2.5 (از 2.2.2.3 تا 2.2.2.5) را به عنوان Source IP به خود اختصاص دهد:

iptables -t nat -A POSTROUTING -o eth1 -j SNAT --to 2.2.2.3-2.2.2.5

با این روش در آن واحد بیشتر از سه کامپیوتر نمی توانند از اینترنت استفاده کنند. پس بنابراین بهتر است که مسیریاب های R1 و R2 و R3 را به صورت SNAT ذکر شده در مثال ۱ پیکربندی کنیم.

به این نوع پیکربندی در سیسکو اصطلاحاً Pool می گویند.

**مثال ۴)**

جهت پیکربندی PAT برروی مسیریاب Backbone از تکنیک زیر استفاده می کنیم:

iptables –A POSTROUTING –o eth1 –t nat –j MASQUERADE

| | |
|---|---|
| **MASQUERADE** | جهت پیکربندی PAT از این عبارت بعد از j– استفاده می کنیم، حتماً تمامی حروف آن بزرگ تایپ شوند. |

**مثال۵)**

در صورتی که بخواهیم ارتباط ما با یک web server توسط یک IP خاص انجام شود (برروی Backbone) به عبارتی آن web server ما را با یک IP غیر از IP اصلی ببیند، از تکنیک زیر استقاده می کنیم:

iptables –t nat –A POSTROUTING –p tcp --dport 80 –d 216.239.32.20 –j SNAT --to 2.2.2.5

در این مثال سرور 216.239.32.20 به جای این که ما را با IP 2.2.2.2 ببیند، با IP 2.2.2.5 بسته های درخواست ما را مشاهده میکند.

**مثال ۶)**

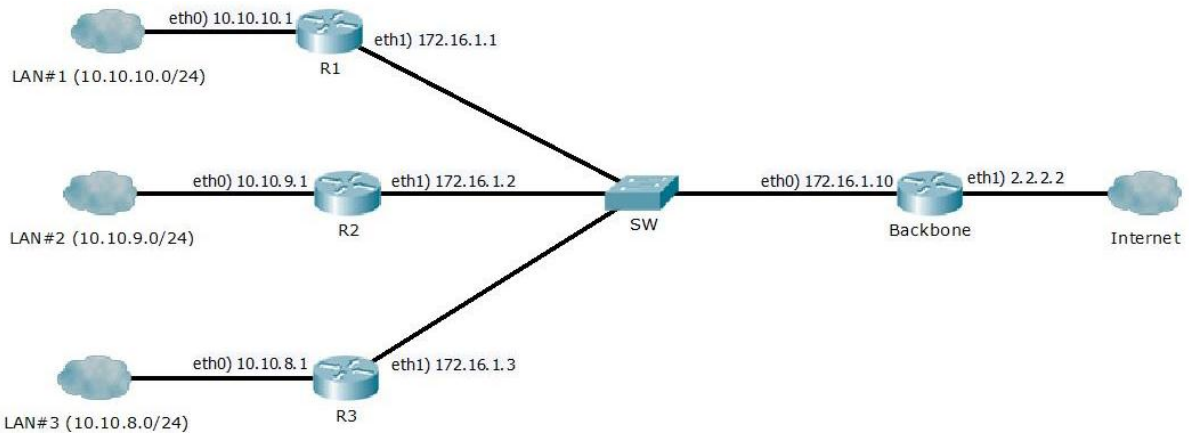
فرض کنید که سرور ما چندین IP داشته باشد و بخواهیم هر IP برای یک سرویس خاص باشد می توانیم از تکنیک زیر بهره بگیریم، به طور مثال 2.2.2.6 برای سرویس Web باشد و 2.2.2.7 را برای سرویس Mail در نظر گرفته باشیم، از تکنیک زیر استفاده می کنیم:

```
iptables –t nat –A POSTROUTING –p tcp --sport 80 –o eth1 –j SNAT --to 2.2.2.6
iptables –t nat –A POSTROUTING –p tcp --sport 25 -o eth1 –j SNAT --to 2.2.2.7
```

**:DNAT**

DNAT زمانی استفاده می شود که بخواهیم مقصد یک بسته را عوض کنیم با عنایت به این نکته که مقصد می تواند پورت و یا IPباشد.

باتوجه به دیاگرام شبکه زیر DNAT بسته ها را از یک شبکه Public مانند اینترنت به یک شبکه Private مانند اینترانت یا LAN هدایت می کند. این نوع NAT را در Cisco با نام Static NAT می شناسند.




**مثال ۱)**

فرض کنید که می خواهیم از طریق مسیریاب Backbone بسته های مربوط به 2.2.2.2 را به سمت مسیریاب R2 هدایت کنیم، برای حل این مسئله از تکنیک زیر استفاده می کنیم:

```
iptables –t nat –A PREROUTING --d 2.2.2.2 –j DNAT --to 172.16.1.2
```

| | |
|---|---|
| **PREROUTING** | زمانی از این Chain استفاده می شود که بخواهیم از DNAT استفاده کنیم. تمامی حروف آن بایستی بزرگ تایپ شوند. |
| **-j DNAT** | در انتهای تمامی Rule های مربوط به DNAT بعد از j– حتماً عبارت DNAT را با حروف بزرگ تایپ می کنیم. |
| **--to** | بعد از این Option حتماً IP که تمامی بسته ها قرار است به سمت آن هدایت شوند را مشخص می کنیم، تمامی حروف آن بایستی کوچک تایپ شوند. |

**مثال ۲)**

فرض کنید که بخواهیم توسط مسیریاب R3 تمامی بسته های مربوط به Web که به طرف 172.16.1.3 می آیند را به سمت سرور 10.10.8.5 هدایت کنیم از تکنیک زیر استفاده می کنیم:

iptables –t nat –A PREROUTING –p tcp --dport 80 –d 172.16.1.3 –j DNAT --to 10.10.8.5

**مثال ۳)**

فرض کنید که بخواهیم توسط مسیریاب R1 کلیه بسته هایی که به طرف 172.16.1.1 پورت 8080 می آیند را به سمت 10.10.10.5 پورت 80 هدایت کنیم، از تکنیک زیر استفاده می کنیم:

iptables –t nat –A PREROUTING –p tcp --dport 8080 –d 172.16.1.1 –j DNAT --to 10.10.10.5:80

**مثال ۴)**

در صورتی که بخواهیم کلیه درخواست های Web مربوط به LAN#2 را به وسیله R1 به سمت سرور 172.16.1.100 هدایت کنیم، از تکنیک زیر استفاده می کنیم:

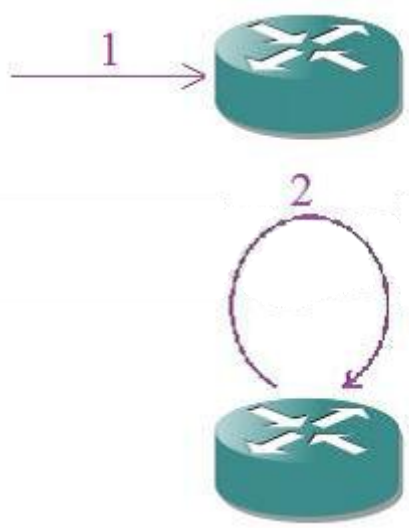
iptables –t nat –A PREROUTING –s 10.10.9.0/24 –p tcp --dport 80 –j DNAT --to 172.16.1.100:80

**مثال ۵)**

می توانیم مسیریاب Backbone را طوری پیکربندی کنیم که خود مسئولیت پاسخگویی به درخواست های Web کل شبکه را به عهده بگیرد:

iptables –t nat –A PREROUTING -p tcp --dport 80 –j REDIRECT --to-port 80

توسط این دستور تمامی درخواست های Web به سمت Apache Server مربوط به مسیریاب Backbone تغییر جهت می دهد.



| | |
|---|---|
| **REDIRECT** | زمانی استفاده می شودکه یک درخواست را به سمت سرور خودمان هدایت کنید و تفاوت آن با بقیه حالت های DNAT این است که بسته به سمت سرور دیگری تغییر جهت نمی دهد. تمامی حروف بایستی بزرگ تایپ شوند. |

## جدول Mangle:

در این قسمت درباره Mangle که یک جدول دیگر از iptable است صحبت کنیم، این جدول کنترل روند رد و بدل شدن اطلاعات را برعهده دارد، و در صورتی که به درستی پیکربندی شود در performance شبکه اثر محسوس و مؤثری دارد و در صورتی که به هر دلیلی پیکربندی درستی از این جدول نداشته باشیم اثر معکوسی بر بازده پیش فرض شبکه ایجاد می کند.

بهبود روند رد و بدل شدن اطلاعات بین دو نقطه QOS می گویند. که مخفف Quality Of Service می باشد. Mangle در لغت به معنای دستکاری کردن می باشد و در فن نیز برای بالا بردن QOS، بایستی Header های یک بسته را دستکاری کنیم.

**مثال ۱)**

ابتدا از یک مثال امنیتی شروع می کنیم، همانطور که می دانیم یکی از راه های کشف کردن سیستم عامل مقصد استفاده ازTTL برگشتی حاصل از Ping می باشد. به جدول زیر دقت کنید:

| سیستم عامل | **TTL** |
|---|---|
| **Linux** | ۶۴ |
| **Cisco** | ۲۵۵ |
| **Windows** | ۱۲۸ |
| **Mikrotik** | ۶۴ |
| **Android** | ۶۴ |
| **MAC** | ۶۴ |

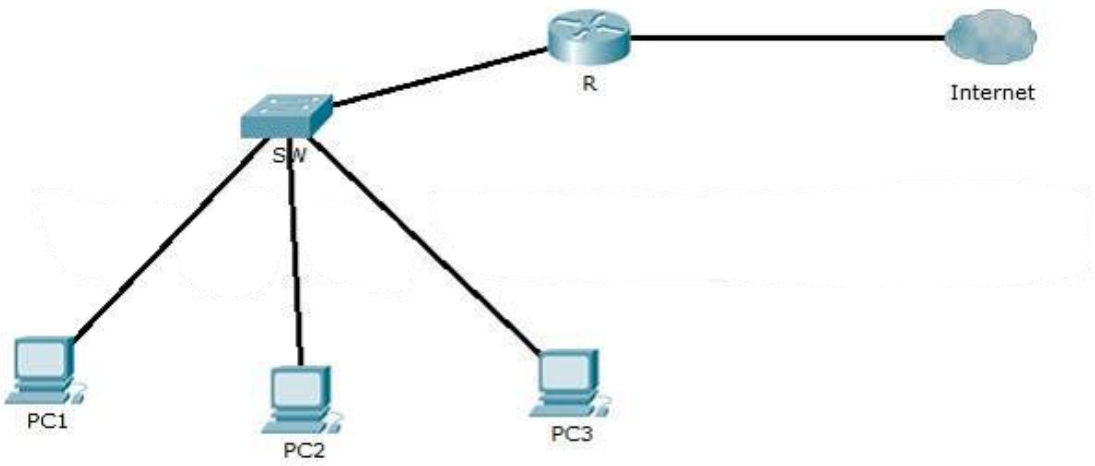
با توجه به جدول بالا در صورتی که بخواهیمLinux را طوری پیکربندی کنیم که در جواب Ping با TTL مربوط به Cisco پاسخ دهد و نفوذگر را گمراه کند که سیستم مورد هدف آن یک Cisco است و Linux نمی باشد، از تکنیک زیر استفاده می کنیم:

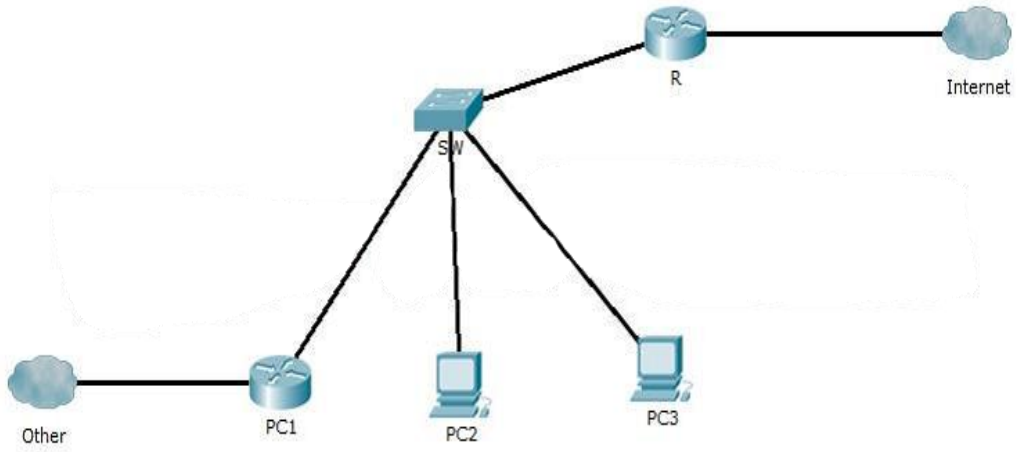
iptables -t mangle -A PREROUTING -j TTL --ttl-set 0

| | |
|---|---|
| **mangle** | در صورت استفاده از جدول mangle از این option استفاده می کنیم. تمامی حروف آن بایستی کوچک تایپ شوند. |
| **TTL** | برای تغییر دادن TTL بعد از j– از این option استفاده می شود و تمامی حروف آن بایستی بزرگ تایپ شوند. |
| **--ttl-set** | به منظور انتساب دادن عدد TLL از این optionاستفاده می کنیم. |

**مثال ۲)**

در صورتی که نخواهیم هیچ کلاینتی به عنوان مسیریاب، اینترنت را برای بقیه به اشتراک بگذارد از تکنیک زیر استفاده می کنیم( به دیاگرام های زیر دقت کنید):




iptables -t mangle -I POSTROUTING -o eth0 -j TTL --ttl-set 1

در این دستور کلیه بسته هایی که به کلاینت ها می فرستیم با مقدار TTL یک می باشند، و در صورتی که بخواهد از مسیریاب دیگر (به طور مثال ازPC1 در دیاگرام غیرقانونی) عبور کند، به علت صفر شدن مقدار TLL در اثر رد شدن از مسیریاب، آن بسته از بین می رود.

**مثال ۳)**

در صورتی که بخواهیم کسی نتواند Topology شبکه را پیدا کند به طور مثال به وسیله دستور tracert یا traceroute و یا نرم افزار Cheops و همچنین روش های Fire Walk، می توانیم از تکنیک زیر استفاده کنیم:

```
iptables -t mangle -I FORWARD -i eth0 -j TTL --ttl-inc 255
```

با وارد کردن دستور بالا در صورت اجرای هر گونه نرم افزار Topology Detector، مسیریاب های موجود بین مبدا و مقصد مشخص نمی شوند.

| | |
|---|---|
| **--ttl-inc** | صرفاً جهت اضافه نمودن مقدار TTL استفاده می شود. تمامی حروف آن بایستی کوچک تایپ شوند. |

**مثال ۴)**

جهت بهبود بازده شبکه از یک معیار به نام TOS(Type Of Service) استفاده می کنیم، تغییر دادن این معیار که یکی از Headerهای بسته اطلاعاتی می باشد بر اساس جدول زیر تعیین می شود:

| مقدار TOS | شرح |
|---|---|
| **16 (0x10)** | Minimize-Delay |
| **8 (0x08)** | Maximize-Throughput |
| **4 (0x04)** | Maximize-Reliability |
| **2 (0x02)** | Minimize-Cost |
| **0 (0x00)** | Normal-Service |

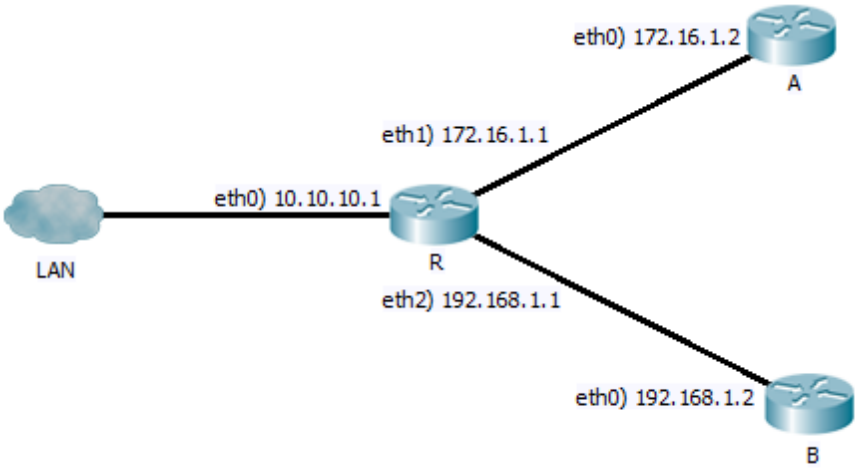
در صورتی که بخواهیم بسته های درخواست Web را با بالاترین اولویت نسبت به بسته های دیگر شبکه ارسال نماییم از تکنیک زیر استفاده می کنیم:

```
iptables -t mangle -A POSTROUTING -p tcp --dport 80 -j TOS --set-tos 8
```

**مثال ۵)**

یکی دیگر از موارد استفاده از Mangle، مارک کردن بسته ها می باشد (Packet Marking) که برای زیر نظر گرفتن بسته های اطلاعاتی استفاده می شود. به سناریو زیر دقت کنید:

فرض کنید که می خواهیم تمامی بسته های مربوط به Web را از طریق مسیریاب A و دیگر بسته های شبکه را از طریق مسیریاب B ارسال نماییم.




ابتدا بسته های Web را مارک می کنیم و دستور زیر را یر روی مسیریاب R وارد می نماییم:

iptables -A PREROUTING -i eth0 -t mangle -p tcp --dport 80 –j MARK --set-mark 1

توسط دستور بالا تمامی بسته های درخواست Web با مقدار 1 مارک می شوند. توجه داشته باشید که اعتبار این مارک در حد همان مسیریاب R می باشد و در صورتی که بسته از مسیریاب R بیرون برود هیچ اثری از آن مارک وجود ندارد.

| | |
|---|---|
| **MARK** | زمانی که بخواهیم بسته ای را مارک کنیم از این option بعد از j– استفاده می کنیم که تمامی حروف آن بایستی بزرگ تایپ شوند. |
| **--set-mark** | مقدار مارک بسته ها را بعد از option وارد می کنیم که تمامی حروف آن بایستی کوچک تایپ شوند. |

بعد از وارد نمودن دستور بالا توسط دستور زیر یک جدول مسیریابی جدید درون فایل /etc/iproute2/rt_tables اضافه می کنیم.

echo 201 web.out >> /etc/iproute2/rt_tables

سپس بسته های مارک 1 شده را به این جدول مسیریابی ارجاع می دهیم.

ip rule add fwmark 1 table web.out

درون جدول مسیریابی Web.out یک default route اضافه می کنیم که توسط آن کلیه بسته های Web به مسیریاب A هدایت می شوند.

ip route add default via 172.16.1.2 dev eth1 table web.out

و برای این که بقیه بسته ها را به مسیریاب B ارسال کنیم، یک default route به جدول مسیریابی اصلی سیستم اضافه می کنیم.

route add default gw 192.168.1.2