Masquerading Made Simple HOWTO

John Tapsell

tapselj0 (at) cs.man.ac.uk

Thomas Spellman

thomas (at) resonance.org

Matthias Grimm

DeadBull (at) gmx.net

도운이

서정민

jms (at) kldp.org

모든 필자들은 irc.opensource.net 서버의 #debian 채널에 있다.

John Tapsell(JohnFlux)가 이 문서의 공식 메인테이너이다.

의문점, 열정, 의견, 날짜 등에 관한 어떤 것이든, John Tapsell에게 이메일을 보내기 바란다.

David Ranch의 문서를 뻔뻔스럽게 도용한 점을 밝힌다 - `<dranch(at)trinnet.net>`.

이 문서는 IP-Masquerading HOWTO 문서에 대한 대체가 아니다. 보충하는 문서이고, 두 문서는 함께 볼 것을 권한다. 이 문서는 그 HOWTO 문서에서 거론된 것을 다시 기술하지 않고, 그 문서를 설명하지도 않는다. 더 자세한 설명서로는 http://ipmasq.cjb.net과 the sandard Masq-HOWTO를 볼 것을 바란다.

주어진 리눅스 호스트에 IP Masquerade를 어떻게 사용하는지에 대해 기술한다. IP Masq는 NAT(Netwrk Address Translation)의 한 형태로써,등록된 인터넷 IP를 없는 내부 네트워크에 연결된 컴퓨터로 하여금, 단 하나의 인터넷 IP 주소를 가진 리눅스 박스를 이용하여 인터넷을 사용할 수 있도록 한다.

고친 과정

고침 번역 v0.952001-09-28고친이 jms

고침 번역 v0.9 2001-09-21고친이 jms

고침 0.05 2001-09-07고친이 jpt

1. 소개

이 문서는 일부러 짧고 이해하기 쉽게 구성되었다.

다음과 같은 외부와 연결시키고 싶은 네트워크가 있다고 하자.

```
 ______________
/              \      외부 IP        __________      내부 IP
|  Internet     |   123.12.23.43    | linux    | 192.168.0.1
|  google.com   |<---[외부 랜카드]-| machine |--[내부 랜카드]
\______________/                     ---------        |
                                                     |
                  ______________         ______|_______
                 | 192.168.0.2 |        /              \
                 |   내부      |-------|  192.168.0.0   |
                 |   컴퓨터    |       |    인트라넷     |
                  -------------         \______________/
```

2. 요약 : (필자는 요약을 먼저 하는 것을 선호한다)

외부 랜카드는 eth0, 외부 IP는 123.12.23.43, 그리고 내부 랜카드는 eth1라고 가정한다면:

```
$> modprobe ipt_MASQUERADE # 만약 실패하더라도 다음을 계속하시오.
$> iptables -F; iptables -t nat -F; iptables -t mangle -F
$> iptables -t nat -A POSTROUTING -o eth0 -j SNAT --to 123.12.23.43
$> echo 1 > /proc/sys/net/ipv4/ip_forward
```

Dial-up 연결의 경우:

```
$> modprobe ipt_MASQUERADE # 만약 실패하더라도 다음을 계속하시오.
$> iptables -F; iptables -t nat -F; iptables -t mangle -F
$> iptables -t nat -A POSTROUTING -o ppp0 -j MASQUERADE
$> echo 1 > /proc/sys/net/ipv4/ip_forward
```

그리고 보안을 위해:

```
$> iptables -A INPUT -m state --state ESTABLISHED,RELATED -j ACCEPT
$> iptables -A INPUT -m state --state NEW -i ! eth0 -j ACCEPT
$> iptables -P INPUT DROP  # 위의 두 줄이 성공했을 경우에만
$> iptables -A FORWARD -i eth0 -o eth0 -j REJECT
```

Dial-up 연결의 경우 (eth0을 내부랜카드로 이용시):

```
$> iptables -A INPUT -m state --state ESTABLISHED,RELATED -j ACCEPT
$> iptables -A INPUT -m state --state NEW -i ! ppp0 -j ACCEPT
$> iptables -P INPUT DROP  # 위의 두 줄이 성공했을 경우에만
$> iptables -A FORWARD -i ppp0 -o ppp0 -j REJECT
```

끝이다! "**iptables -t nat -L**"로 규칙을 볼 수 있다.

3. 자세한 설명

2.4.x 대의 커널 컴파일하는 경우:

다음의 커널 지원이 필요하다.

- Networking Options 란에서
  - Network packet filtering (CONFIG_NETFILTER)
- Networking Options->Netfilter Configuration 란에서
  - Connection tracking (CONFIG_IP_NF_CONNTRACK)
  - FTP Protocol support (CONFIG_IP_NF_FTP)
  - IP tables support (CONFIG_IP_NF_IPTABLES)
  - Connection state match support (CONFIG_IP_NF_MATCH_STATE)
  - Packet filtering (CONFIG_IP_NF_FILTER)
    - REJECT target support (CONFIG_IP_NF_TARGET_REJECT)

- Full NAT (CONFIG_IP_NF_NAT)
  - MASQUERADE target support (CONFIG_IP_NF_TARGET_MASQUERADE)
  - REDIRECT target support (CONFIG_IP_NF_TARGET_REDIRECT)
- Packet mangling (CONFIG_IP_NF_MANGLE)
- LOG target support (CONFIG_IP_NF_TARGET_LOG)

먼저, iptable과 masq 모듈이 커널에 컴파일, 설치가 되지 않았지만 모듈로만 존재하는 경우, 설치할 필요성이 있다. 만약 'insmod ipt_MASQUERADE"한다면, ip_tables, ip_conntrack 그리고 iptable_nat을 로드한다.

```
$> modprobe ipt_MASQERADE
```

인트라넷이 크거나, 두세대 이상의 컴퓨터를 인터넷에 연결할려고 하더라도 별로 큰 차이는 없다.

자, 다른 룰을 가지고 있지 않다는 가정하에, 다음을 하라.

```
$> iptables -F; iptables -t nat -F; iptables -t mangle -F
```

만약 iptables를 찾을 수 없다는 에러가 날 경우, iptables를 찾아 설치하라. 만약 nat과 같은 테이블이 없다면, nat 지원을 넣고 다시 커널을 컴파일하라. mangle이란 테이블이 없다고 하면, 걱정하지 마라, 마스커레이딩에 필요한 것이 아니다. 만약 iptables가 커널과 맞지 않다고 하면(incompatible), 2.4 이상의 커널을 구하여 iptables 지원을 하여 컴파일하라.

그리고, 고정 아이피인 경우 (예: DHCP가 아닌 네트워크):

```
$> iptables -t nat -A POSTROUTING -o eth0 -j SNAT --to 123.12.23.43
```

유동 아이피인 경우 (예: 모뎀 - 전화접속을 먼저 하라):

```
$> iptables -t nat -A POSTROUTING -o ppp0 -j MASQUERADE
```

마지막으로 커널에 확답을 주기 위해, 패킷포워딩을 한다 (매 리부팅시 한번만 실행하면 되지만, 여러번 해도 상관없다) :

```
$> echo 1 > /proc/sys/net/ipv4/ip_forward
```

이 모든 것이 작동될 경우, 내부 네트워크에서의 마스커레이딩만 허용하기를 바란다(다음장 참고). - 인터넷을 통한 접속을 원치 않을 것이기 때문이다.

먼저, 모든 연결을 허락한다. (e.g. FTP 서버 연결)

```
$> iptables -A INPUT -m state --state ESTABLISHED,RELATED -j ACCEPT
```

여기에서 에러가 난다면, 커널이 state tracking을 지원하지 않을 가능성이 크다. 재컴파일할 것을 권한다. 그리고, 이제 인트라넷(로컬, 내부 네트워크)으로부터의 새로운 연결만 허락한다. ppp0를 eth0나 자신의 외부 디바이스로 대체하라. (!는 '외에 어떤 것이든' 란 뜻이다)

```
$> iptables -A INPUT -m state --state NEW -i ! ppp0 -j ACCEPT
```

이제 다른 모든 연결은 막는다:

```
$> iptables -P INPUT DROP  # 위의 두 줄이 성공했을 경우에만
```

처음 두 룰 중에 하나라도 실패한다면, 이 마지막 룰은 마스커레이딩하는 것을 막을 것이다. 이 룰을 되돌리기 위해선, "**iptables -P INPUT ACCEPT**"을 하면 된다.

4. 설치 후를 위한 설명

이제 모든 것이 작동될 것이다. 다음을 잊지 마라:

- 내부 네트워크에 있는 모든 클라이언트의 게이트웨이 값에 내부 IP 주소를 설정한다. (윈도우에서는 오른쪽 클릭하여 네트워크 환경->등록정보->게이트웨이에 가서 값을 리눅스 게이트웨이 내부 IP를 설정한다)
- ISP의 HTTP 프락시가 있다면 설정하거나, 투명 프락시를 사용하게 모든 클라이언트들을 설정하고, (경고 - 큰 네트워크에서는 투명 프락시가 아주 느리다는 것을 자주 들었다) 혹은 새 리눅스 게이트웨이에 squid를 실행한다. (이것은 옵션이지만, 큰 네트워크에서 선호한다)
- 이제 보안을 생각해 보자. 먼저 일반적으로 포워딩을 끄자: "**iptables -P FORWARD DROP**", 그리고 iptables와 `/etc/hosts.allow`과 `/etc/hosts.deny` 사용법을 익힌다. 경고 - 마스커레이딩이 작동될 때까지는 이 iptalbes 룰을 시도하지 마라. 마지막 룰을 DENY할려면, 원하는 모든 패킷을 명백하게 통과하도록 해야 한다. (되돌릴려면, "**ipatbles -P FORWARD ACCEPT**")
- 이제 인터넷으로 접속할 수 있는 서비스를 허락한다.

  예를 들어, 웹서버 접속을 위해선 다음을 한다:

  ```
  $> iptables -A INPUT --protocol tcp --dport 80 -j ACCEPT
  $> iptables -A INPUT --protocol tcp --dport 443 -j ACCEPT
  ```

  ident서비스(예: irc 접속)를 위해서는 다음을 한다:

  ```
  $> iptables -A INPUT --protocol tcp --dport 113 -j ACCEPT
  ```

테스트할려면:

- 웹 접속이 가능한 클라이언트에서 접속을 시도한다. 구글 IP는 216.239.33.100(많은 구글 IP 중 하나)이고 접속이 가능해야 할것이다. 예: "**ping 216.239.33.100**" "**lynx 216.239.33.100**".
- 도메인 네임으로의 접속이 가능한지도 시도한다. 예: "**ping google.com**" "**lynx google.com**" 혹은 IE/Netscape로의 접속

eth0는 외부 인터넷 카드이고, 123.12.23.43은 외부 IP이다.

5. FAQ's

- 내가 이용한 룰들을 볼려면?

  - 다음을 시도한다.

  ```
  $> iptables -L
  $> iptables -t nat -L
  ```

- 안된다! iptables/NAT/SNAT/MASQ 모두 안된다.

  - 최신의 커널을 구하여 iptables와 NAT 옵션 지원으로 컴파일하라.

- 안된다! 마스커레이딩이 전혀 작동을 안한다.

  - Try **echo 1 > /proc/sys/net/ipv4/ip_forward**

- 안된다! 네트워크 자체가 안된다.

  - 다음을 시도한다.

  ```
  $> iptables -F
  $> iptables -t nat -F
  $> iptables -t mangle -F
  ```

  모든 룰이 안된다면, 다른 iptables 룰들을 재실행한다.

  - **iptables -P FORWARD ACCEPT**를 시도한다.

- 아직 작동을 안한다!

  - "**dmesg | tail**"는 에러나는가? 아니면 " **cat /var/log/messages | tail**"는 어떠한가?

- 이해가 안간다, 그냥 작동을 안한다!

  - 모르겠다. 그러나 다음이 가능해야 한다:

  ```
  1) 게이트웨이에서 외부로 핑이 간다.
  2) 게이트웨이에서 내부 클라이언트에 핑이 간다.
  3) 내부 클라이언트에서 게이트웨이로 핑이 간다.
  ```

  이것은 마스커레이딩을 시도하기 전에 가능해야 한다.

- 이것을 어디다가 다 넣는가?

  - `/etc/network/interfaces` 파일이나 firewall.rc에 넣는다. interfaces 파일에 넣는다면, 외부 인터페이스의 앞에 올리고, "**iptables -t nat -F**"를 나중에 내리는 것으로 설정한다.

- 요청시 ppp만 어떻게 가능하게 하는가? How do I get it to only bring the ppp up on demand?

  - ISP 게이트웨이의 IP가 23.43.12.43이라고 할 때, 다음 한줄을 `/erc/ppp/peers/provider`의 끝에 추가하라.

  :23.43.12.43

(이것은 유동 IP를 위한 것이다 - 고정 IP는 my.**external.ip.number:23.43.12.43**라고 될 것이다)

그리고 파일의 끝에 라인 하나를 추가한다:

demand

pppd는 연결이 끊길 경우, "**ifdown ppp0**" 혹은 "**poff**" 명령이 있을 때까지 백그라운드에서 재다이얼을 시도한다. 이것은 "**nopersist**" 옵션을 선택하지 않을 경우이고, 선택한다면 연결이 끊기자마자, pppd는 멈출 것이다. 10분을 보류한 뒤 연결을 끊고 싶다면, "**idle 600**"란 새로운 라인을 추가하면 된다.

- 연결이 계속 죽는다!

  - 첫번째, 다이얼링을 요구하는가? 그냥 원래 그렇게 되도록 되어있는건 아닌가? /etc/ppp/peers/provider를 체크하라, 그리고 마스커레딩하기 전에 다이얼링이 되는지 확인하라.

  - 두번째, 만약 위의 문제가 아니라면, 나같이 엉뚱하게 돌아가고 있을지도 모른다. 커널 2.4.3 버전으로 돌아가서 작동되는지 확인해봐라. (나도 왜 그런지 잘 모른다)

- 나 이런 방법이 싫다! 미리 만들어진 스크립트나 GUI나 그런것들을 원한다.

  - 여기를 참고하라: http://shorewall.sourceforge.net/ (http://shorewall.sourceforge.net/)

- 케이블모뎀은 고정 IP인가 동적 IP인가?

  - 좋은 질문이다. 동적이라고 생각하는게 나을 것이다.

- DHCP 네트워크는 고정 IP인가 동적 IP인가?

  - 동적이다.

- 서비스는 어떻게 하는가?

  - IP 포트들을 포워딩하거나 리다이렉트하라 - 또 한번 거론하지만, 방어벽이 필요하다면 설치하라.

- 클라이언트에서 리눅스 게이트웨이의 외부 IP로 핑은 되는데 인터넷은 안된다.

  - "**rmmod iptable_filter**"를 해봐라. - 다음 버전에서 더 자세히 언급하겠다.

  - **routed**나 **gated**를 작동하지 않고 있는지 확인하라. - 체크를 위해서는, "**ps aux | grep -e routed -e gated**".

  - 여기를 참고하라. http://ipmasq.cjb.net

- 설정된 연결들을 어떻게 보는가? 예를 들어, netstat..

  - cat /proc/net/ip_conntrack 을 시도하라.

- squid 정보와 라우팅 등에 관해 더 알고 싶다!

- Advanced Routing HOWTO를 볼 것을 권한다. http://www.linuxdoc.org/HOWTO/Adv-Routing-HOWTO.html

- 이 HOWTO 문서는 쓰레기이다! 이 문서를 쓴 사람들한테 어떻게 따질 수 있는가?

  - irc.opensource.net의 #debian 방에 가서 JohnFlux를 찾아라. - 번역에 관한 것이면, irc.hanirc.org의 #debian 방에 가서 hutbaram을 찾아라.

- 이 HOWTO 문서는 쓰레기이다! 더 나은 버전은 어디서 구하는가?

  - http://ipmasq.cjb.net

  - LDP의 Masq-HOWTO 문서를 참고하라.