Netfilter 확장 HOWTO

MARIEFabrice

fabrice (at) celestix.com

도운이

정두기

doogie (at) darimtech.com

이 문서는 넷필터에 대한 iptables 확장을 어떻게 사용하고 설치하는 지를 설명한다.

고친 과정

고침 0.12002-01-28고친이 정두기 (darimtech.com)

최초 작성

1. 소 개

안녕. 이 문서는 넷필터를 사용하고, 버그 리포팅을 하며, 테스트를 하고, 개발하는 데 시간을 소모하는 모든 사람들한테 내가 감사할 기회이다.

이 하우투는 러스티의 Linux 2.4 패킷 필터링 하우투 (http://netfilter.samba.org/unreliable-guides/packet-filtering-HOWTO/index.html) 를 읽고 이해한다는 가정하에 쓰여졌다. 또한 당신이 커널을 적절히 설치하고 컴파일할 수 있다고 가정한다.

iptables 배포본은 일반 유저들을 위한 것뿐만 아니라 실험적인 유저들을 위한 확장을 포함하고 있으며, 또한 이 확장은 커널에 의존적이다. 원하지 않는다면, 이 확장은 보통 컴파일되지 않는다.

이 하우투의 목표는 넷필터 확장을 어떻게 설치하고 어떻게 기본적으로 다루는지를 시작하는 사람들에 대한 도움을 주는 것이다.

(C) 2001 Fabrice MARIE. GNU GPL 라이센스를 따른다.

2. Patch-O-Matic

2.1. Patch-O-Matic이란 무엇인가?

iptables 메이크파일은 'patch-o-matic' (또는 'p-o-m')이라 불리는 기능을 포함한다. p-o-m는 당신이 원하는 패치들을 선택하는 과정으로 안내해 주고, 당신을 위해 커널을 자동으로 패치한다.

처음, 당신은 가장 최근의 CVS 트리를 얻어야 하며, 가장 최근의 확장을 사용하고 있는지를 확인하라. 그렇게 하려면 다음과 같이 해야 한다 :

```
# cvs -d :pserver:cvs@pserver.samba.org:/cvsroot login
# cvs -z3 -d :pserver:cvs@pserver.samba.org:/cvsroot co netfilter
```

이것은 최상위 디렉토리 'netfilter/'를 만들 것이고, 내부의 파일들을 검사한다.

커널 소스가 '/usr/src/linux/'에 이미 있음을 확인하라. 의존성이 검사되었는지를 확인하고 그렇지 않다면 :

```
# cd /usr/src/linux/
# make dep
```

그 다음 'userspace/'에 있는 넷필터 디렉토리로 가서, p-o-m을 호출할수 있다.

2.2. Patch-O-Matic 실행하기

'userspace/' 디렉토리에서, p-o-m을 실행하라 :

```
# make patch-o-matic

Welcome to Rusty's Patch-o-matic!

Each patch is a new feature: many have minimal impact, some do not.
Almost every one has bugs, so I don't recommend applying them all!
-------------------------------------------------------

Already applied: 2.4.1 2.4.4
Testing... name_of_the_patch NOT APPLIED ( 2 missing files)
The name_of_the_patch patch:
Here usually is the help text describing what
the patch is for, what you can expect from it,
and what you should not expect from it.
Do you want to apply this patch [N/y/t/f/q/?]
```

p-o-m은 관련된 모든 패치를 실행한다. 이미 모두 적용되어 있다면, 첫줄에 'Already applied:'를 볼 수 있다. 그렇지 않다면, 패치 이름과 약간의 설명이 표시될 것이다. p-o-m은 어떻게 진행되는지를 설명한다 : 'NOT APPLIED (n missing files)'는 패치가 아직 적용되지 않았

음을 의미하고, 반면에 'NOT APPLIED (n rejects out of n hunks)'는 일반적으로 다음을 뜻한다:

1. 패치가 완벽하게 적용되지 않았거나...
2. ...또는 이미 패치하려는 커널에 포함되어 있는 경우

마지막으로 패치할 것인지 아닌지를 결정하게 하는 프롬프트가 보일 것이다.

- 패치하기를 원하지 않는다면 단순히 엔터키만 누른다.
- 패치를 적용하고 테스트하기 위해 p-o-m을 원한다면 'y'키를 누른다, 만약 실패한다면 다시 한번 확인을 위한 프롬프트를 보여줄 것이다. 그렇지 않다면, 패치는 적용된 것이고, 'Already Applied' 라인에서 패치의 이름을 볼 수 있다.
- 패치가 정상적으로 적용되는지를 테스트하려면 't'키를 누른다.
- 패키가 'p-o-m'에 강제로 적용되기를 원한다면 'f'키를 누른다.
- 마지막으로 p-o-m을 종료하기를 원한다면 'q'키를 누른다.

경험적으로 실제로 패치를 적용하기 전에 각 패치에 대한 약간의 설명을 조심스럽게 읽는 것이다. 현재 patch-o-matic에 대한 많은 공식 패치가 있기 때문에 (그리고 아마도 더 많은 비공식 패치도 있을 것이다), 모두를 적용하는 것은 추천하지 않는다 ! 비록 더많은 패치가 필요할 때 넷필터를 재컴파일하는 것을 의미할 지라고, 원하는 패치만 적용하는 것을 고려해야 한다.

이제 patch-o-matic의 새로운 형태가 만들어졌을 것이다. 이것은 단지 깨끗하게 적용되어진 패치들을 보여줄 뿐만 아니라, 적용되지 않은 다른 모든 패치도 보여준다. 이를 실행하기 위해서, 다음과 같이 하라 :

```
# make most-of-pom
```

이것은 관련 패치와 그리고 그와 상호작용을 하는 것에 관해서는 patch-o-matic과 정확하게 같은 방법으로 동작한다. 'developer-only' 패치를 사용하는 것은 피하라.

2.3. 그 다음은 무엇을 하는가 ?

적용하기를 원하는 모든 패치를 적용했다면, 다음 단계는 커널을 재컴파일하고 설치하는 것이다. 이 하우투는 그것을 어떻게 하는지에 대한 설명은 하지 않는다. 대신에 리눅스 커널 HOWTO (http://www.linuxdoc.org/HOWTO/Kernel-HOWTO.html) 를 보라.

커널을 재구성하는 동안, "Networking Options -> Netfilter Configuration"에서 새로운 옵션들을 볼 수 있다. 필요로 하는 옵션을 선택하고, 커널을 재컴파일하고 설치하라.

커널을 설치하면, 'userspace/' 디렉토리에 있는 "iptables" 패키지를 다음과 같이 컴파일하고 설치하라

```
# make all install
```

새로운 iptables 패키지가 설치된 것이다 ! 이제 새로운 기능을 사용할 시간이다.

3. 새로운 netfilter matches

이 절에서, 새로운 netfilter matches의 사용법을 설명할 것이다. 이 패치들은 알파벳 순서로 나타낼 것이다. 부가적으로, 영향을 미치는 다른 패치에 대한 설명은 없다. 이는 다음 버전 문서에 포함될 것이다.

일반적으로, 다음과 같이 하면 특별한 모듈로부터 힌트를 얻을 수 있다.

```
# iptables -m the_match_you_want --help
```

이는 보통의 iptables 도움말을 보여주고, 거기에 더해서 끝에 명시된 “원하는 match”에 해당되는 도움말을 보여준다.

3.1. ah-esp patch

이 패치는 Yon Uriarte <yon@astaro.de>에 의해 작성되었고 다음의 2가지 새로운 matches를 한 것이다 :

- “ah” : Security Parameter Index (SPI)에 기초한 AH 패킷을 match할 수 있도록 한다.
- “esp” : SPI에 기초한 ESP 패킷을 match할 수 있도록 한다.

이 패치는 SPI에 기초한 연결들을 구분짓고자 IPSEC을 사용하는 사람들에게 유용할 수 있다.

예를 들어, 다음과 같이 하면 500과 일치하는 SPI를 가지는 모든 AH 패킷을 드롭시킬수 있다.

```
# iptables -A INPUT -p 51 -m ah --ahspi 500 -j DROP

# iptables --list
Chain INPUT (policy ACCEPT)
target     prot opt source               destination
DROP       ipv6-auth--  anywhere             anywhere           ah spi:500
```

ah match가 지원하는 옵션은 다음과 같다 :

- --ahspi [!] spi[:spi] -> match spi (range)

esp match도 똑같이 작용한다.

```
# iptables -A INPUT -p 50 -m esp --espspi 500 -j DROP
```

```
# iptables --list
Chain INPUT (policy ACCEPT)
target     prot opt source               destination
DROP       ipv6-crypt--
anywhere             anywhere           esp spi:500
```

esp match가 지원하는 옵션은 다음과 같다 :

- --espspi [!] spi[:spi] -> match spi (range)

ah 또는 esp match를 사용할때, 또는 명백한 이유로 룰 첨가를 중단하고자 할때, "-p 50" 또는 "-p 51" (esp & ah 각각)을 통해 적절한 프로토콜을 명시하는 것을 잊지 말아야 한다.

3.2. iplimit patch

이 패치는 Gerd Knorr <kraxel@bytesex.org>에 의해 작성되었으며, 특정 호스트나 네트워크로부터의 TCP 연결 갯수를 어떻게 제한하는지에 대한 새로운 match를 한 것이다.

예를 들어, 한 IP 주소에 의한 HTTP 연결 갯수를 4개로 제한하려고 하면 :

```
# iptables -A INPUT -p tcp --syn --dport http -m iplimit --iplimit-
above 4 -j REJECT

# iptables --list
Chain INPUT (policy ACCEPT)
target   prot opt source    destination
REJECT   tcp  --  anywhere  any-
where     tcp dpt:http flags:SYN,RST,ACK/SYN #conn/32 > 4 reject-
with icmp-port-unreachable
```

또는 예를 들어 class A 전체의 연결 갯수를 제한하기를 원한다면 :

```
# iptables -A INPUT -p tcp --syn --dport http -m iplimit --iplimit-
mask 8 --iplimit-above 4 -j REJECT

# iptables --list
Chain INPUT (policy ACCEPT)
target   prot opt source    destination
REJECT   tcp  --  anywhere  any-
where     tcp dpt:http flags:SYN,RST,ACK/SYN #conn/8 > 4 reject-with icmp-
port-unreachable
```

iplimit patch가 지원하는 옵션은 다음과 같다 :

- [!] --iplimit-above n -> 현재 tcp 연결 갯수를 n개 이상으로 하려면 (하지 않으려면)

- --iplimit-mask n -> subnet mask를 사용하는 그룹 호스트들

3.3. ipv4options patch

이 패치는 Fabrice MARIE <fabrice@celestix.com>에 의해 작성되었으며, 설정된 IP 옵션에 의해 패킷을 match할 수 있게 할 수 있도록 새로운 match를 한 것이다.

예를 들어, IP 옵션에 설정된 record-route 또는 timestamp를 가진 모든 패킷을 드롭하려면 다음과 같이 설정한다 :

```
# iptables -A INPUT -m ipv4options --rr -j DROP
# iptables -A INPUT -m ipv4options --ts -j DROP

# iptables --list
Chain INPUT (policy ACCEPT)
target     prot opt source               destination
DROP       all  --  anywhere             anywhere            IPV4OPTS RR
DROP       all  --  anywhere             anywhere            IPV4OPTS TS
```

ipv4options match가 지원하는 옵션은 다음과 같다 :

- --ssrr -> strict source routing flag에 match되는.
- --lsrr -> loose source routing flag에 match되는.
- --no-srr -> source routing을 가지지 않는 패킷에 match되는.
- --rr -> record route flag에 match되는.
- [!] --ts -> timestamp flag에 match되는.
- [!] --ra -> router-alert option에 match되는.
- [!] --any-opt -> 적어도 하나의 IP 옵션(또는 !이 선택되지 않은 모든 IP 옵션) 을 가진 패킷에 match되는.

3.4. length patch

이 패치는 James Morris <jmorris@intercode.com.au>에 의해 작성되었으며, 길이에 기초한 패킷을 match할 수 있게 새로운 match를 한 것이다.

예를 들어, 85 바이트보다 큰 패킷 크기를 가진 모든 ping packet을 드롭하려면 다음과 같이 한다 :

```
# iptables -A INPUT -p icmp --icmp-type echo-request -m length --
length 85:0xffff -j DROP
```

```
# ptables --list
Chain INPUT (policy ACCEPT)
target     prot opt source               destination
DROP       icmp --  anywhere             anywhere           icmp echo-
request length 85:65535
```

length match에 대한 부가적인 옵션은 다음과 같다 :

- [!] --length length[:length] -> value 또는 value의 범위에 대한 패킷 길이에 해당하는.

표현되지 않은 value의 범위는 내포되어 있을 것이다. 내포된 value는 최소 0, 최고 65535이다.

3.5. mport patch

이 패치는 Andreas Ferber <af@devcon.net>에 의해 작성되었으며, TCP, UDP 연결에 대해 단일포트와 포트범위를 조합해서 포트를 명시할수 있도록 새로운 match를 한 것이다.

예를 들어, 한 라인에서 ftp, ssh, telnet, http를 막기를 원한다면, 다음과 같이 할 수 있다 :

```
# iptables -A INPUT -p tcp -m mport --ports 20:23,80 -j DROP

# iptables --list
Chain INPUT (policy ACCEPT)
target     prot opt source               destination
DROP       tcp  --
anywhere             anywhere           mport ports ftp-data:telnet,http
```

mport match에 대한 부가적인 옵션은 다음과 같다 :

- --source-ports port[,port:port,port...] -> source port(s)에 match된다.
- --sports port[,port:port,port...] -> source port(s)에 match된다.
- --destination-ports port[,port:port,port...] -> destination port(s)에 match된다.
- --dports port[,port:port,port...] -> destination port(s)에 match된다.
- --ports port[,port:port,port] -> source and destination port(s) 모두에 match된다.

3.6. nth patch

이 패치는 Fabrice MARIE <fabrice@celestix.com>에 의해 작성되었으며, 룰에 의해 받은 특정 N번째 패킷을 match할 수 있도록 새로운 match를 한 것이다.

예를 들어, 매 2번째 핑 패킷을 드롭하길 원한다면, 다음과 같이 할 수 있다 :

```
# iptables -A INPUT -p icmp --icmp-type echo-request -m nth --every 2 -
j DROP

# iptables --list
Chain INPUT (policy ACCEPT)
target     prot opt source               destination
DROP       icmp --  anywhere             anywhere           icmp echo-
request every 2th
```

이 패치는 Richard Wagner <rwagner@cloudnet.com>에 의해 확장되었는데, 이는 inbound와 outbound 연결에 대한 로드 밸런싱을 제공하는 쉽고 빠른 방법을 만들 수 있게 해준다.

예를 들어, 10.0.0.5, 10.0.0.6, 10.0.0.7의 3개 주소에 대한 로드 밸런싱을 원한다면, 다음과 같이 할 수 있다 :

```
# iptables -t nat -A POSTROUTING -o eth0 -m nth --counter 7 --every 3 --
packet 0 -j SNAT --to-source 10.0.0.5
# iptables -t nat -A POSTROUTING -o eth0 -m nth --counter 7 --every 3 --
packet 1 -j SNAT --to-source 10.0.0.6
# iptables -t nat -A POSTROUTING -o eth0 -m nth --counter 7 --every 3 --
packet 2 -j SNAT --to-source 10.0.0.7

# iptables -t nat --list
Chain POSTROUTING (policy ACCEPT)
target     prot opt source               destination
SNAT       all  --
anywhere             anywhere           every 3th packet #0 to:10.0.0.5
SNAT       all  --
anywhere             anywhere           every 3th packet #1 to:10.0.0.6
SNAT       all  --
anywhere             anywhere           every 3th packet #2 to:10.0.0.7
```

nth match가 지원하는 옵션은 다음과 같다 :

- --every Nth -> 모든 N번째 패킷과 일치
- [--counter] num -> 카운터 0-15 (디폴트값:0) 사용.
- [--start] num -> 0 대신 'num'으로 카운터를 초기화. 이 num은 0에서 (Nth-1) 사이여야 한다.
- [--packet] num -> 'num' 패킷과 일치. 0 ~ (Nth-1) 사이여야 한다. '--packet'이 카운터로 사용된다면 0에서 (Nth-1)사이의 모든 value를 처리하기 위해 --packet 룰에 N번째 number가 있어야 한다.

3.7. pkttype patch

이 패치는 Michal Ludvig <michal@logix.cz>에 의해 작성되었으며, 호스트/브로드캐스트/멀티캐스트 등 그 타입에 기초한 패킷을 match할 수 있도록 새로운 match를 한 것이다.

예를 들어, 모든 브로드캐스트 패킷을 조용히 드롭시키길 원한다면 :

```
# iptables -A INPUT -m pkttype --pkt-type broadcast -j DROP

# iptables --list
Chain INPUT (policy ACCEPT)
target     prot opt source               destination
DROP       all  --
anywhere             anywhere           PKTTYPE = broadcast
```

pkttype match가 지원하는 옵션은 다음과 같다 :

- --pkt-type [!] packettype -> 패킷 타입이 다음중 하나일 경우 패킷 타입을 일치시킨다.

  - host -> 모두
  - broadcast -> 전체
  - multicast -> 그룹

3.8. pool patch

Patrick Schaaf <bof@bof.de>.에 의해 패치되었고, Joakim Axelsson and Patrick에 의해 재작성중에 있다. 그러므로 이 부분은 곧 바뀔 것이다.

3.9. psd patch

이 패치는 Dennis Koslowski <dkoslowski@astaro.de>에 의해 작성되었으며, 포트 스캔을 탐지하는데 관한 새로운 match이다.

가장 간단한 형태로, psd match는 다음과 같이 사용될 수 있다 :

```
# iptables -A INPUT -m psd -j DROP

# iptables --list
Chain INPUT (policy ACCEPT)
target  prot opt source    destination
DROP    all  --  anywhere  anywhere    psd weight-threshold: 21 delay-
threshold: 300 lo-ports-weight: 3 hi-ports-weight: 1
```

psd match가 제공하는 옵션은 다음과 같다 :

- [--psd-weight-threshold threshold] -> Portscan 탐지 가중치
- [--psd-delay-threshold delay] -> Portscan 탐지 지연치
- [--psd-lo-ports-weight lo] -> well-known 포트(privileged port) 가중치
- [--psd-hi-ports-weight hi] -> user 포트(High ports) 가중치

3.10. random patch

이 패치는 Fabrice MARIE <fabrice@celestix.com>에 의해 작성되었으며, 주어진 확률에 기초한 패킷을 랜덤하게 계산할 수 있도록 하는 새로운 match이다.

예를 들어, 50%의 핑 패킷을 랜덤하게 드롭하기를 원한다면, 다음과 같이 할 수 있다 :

```
# iptables -A INPUT -p icmp --icmp-type echo-request -m random --
average 50 -j DROP

# iptables --list
Chain INPUT (policy ACCEPT)
target     prot opt source       destination
DROP       icmp --  anywhere     anywhere        icmp echo-
request  random 50%
```

random patch가 지원하는 옵션은 다음과 같다 :

- [--average] percent -> match %에 대한 확률 생략된다면, 50%의 확률이 세팅된다. 퍼센트는 1과 99사이의 숫자여야 한다.

3.11. realm patch

이 패치는 Sampsa Ranta <sampsa@netsonic.fi>에 의해 작성되었으며, 패킷 분류자에 나타나는 키와 유사한 기준과 일치하는 것으로써 라우팅 영역 키를 사용할 수 있도록 하는 새로운 match이다.

예를 들어, 10개의 영역에서 외부로 향하는 패킷을 모두 로그에 기록하려면, 다음과 같이 할 수 있다 :

```
# iptables -A OUTPUT -m realm --realm 10 -j LOG

# iptables --list
Chain OUTPUT (policy ACCEPT)
target     prot opt source               destination
LOG        all  --
anywhere             anywhere           REALM match 0xa LOG level warning
```

realm match가 지원하는 옵션은 다음과 같다 :

- --realm [!] value[/mask] -> 영역 일치

3.12. record-rpc patch

이 패치는 Marcelo Barbosa Lima <marcelo.lima@dcc.unicamp.br>에 의해 작성되었으며, 효과적인 RPC 필터링을 허용하기 위해 패킷 소스가 이전에 portmapper를 통해 포트를 요청했을 경우, 또는 portmapper에 대한 새로운 GET 요청일 경우 match하는 데 대한 새로운 match이다.

RPC 연결 추적 정보를 match하기 위해, 간단히 다음과 같이 할 수 있다 :

```
# iptables -A INPUT -m record_rpc -j ACCEPT

# iptables --list
Chain INPUT (policy ACCEPT)
target     prot opt source               destination
ACCEPT     all  --  anywhere             anywhere
```

record_rpc match는 어떠한 옵션도 가지지 않는다.

match 정보가 없다고 염려할 것은 없다. 이 match에 대한 print() function이 비어있기 때문에 이는 간단하다.

```
/* Prints out the union ipt_matchinfo. */
static void
print(const struct ipt_ip *ip,
const struct ipt_entry_match *match,
int numeric)
{
}
```

3.13. string patch

이 패치는 Emmanuel Roger <winfield@freegates.be>에 의해 작성되었으며, 패킷의 한 문자열을 match하는 것에 대한 새로운 match이다.

예를 들어, “cmd.exe” 문자열을 포함하고 있는 패킷을 match하고 userland IDS로 보내려면, 다음과 같이 할 수 있다 :

```
# iptables -A INPUT -m string --string 'cmd.exe' -j QUEUE

# iptables --list
Chain INPUT (policy ACCEPT)
target     prot opt source               destination
```

```
QUEUE      all  --
anywhere             anywhere           STRING match cmd.exe
```

조심스럽게 이 match를 사용해야 한다. 많은 사람들이 DROP taget에 따라서 웜 바이러스를 멈추기 위해 이 match를 사용하길 원한다. 이는 중요한 실수이다. 특정 IDS 침입 방법은 이를 무력화할수 있다.

유사한 경향으로, 많은 사람들은 POST 문자열을 포함하는 HTTP 패킷을 드롭함으로써 POST나 GET같은 HTTP의 특정 기능을 멈추기 위한 수단으로 이 match를 사용하기를 원했었다. 이러한 작업은 proxy를 필터링하는 것이 더 좋은 방법임을 이해하라. 부가적으로 POST란 단어를 가지고 있는 HTML content는 이전 방법(설정)에 의해 드롭될 것이다. 이 match는 더 좋은 분석을 위해 유저영역의 관심있는 패킷을 큐잉할수 있게 하기 위해 설계되었다. 이것이 전부이다. 이 방법에 의해 패킷을 드롭하는 것은 특정 IDS 침입 방법에 의해 무력화될 수 있다.

string match가 지원하는 옵션은 다음과 같다 :

- --string [!] string -> 패킷의 문자열을 일치시킨다.

3.14. time patch

이 패치는 Fabrice MARIE <fabrice@celestix.com>에 의해 작성되었으며, 출발 혹은 도착 (로컬에서 생성된 패킷) 시간에 기초한 패킷을 match할 수 있도록 하는 새로운 match이다.

예를 들어, 월요일부터 금요일까지 8:00부터 18:00까지 도착 시간을 가진 패킷을 허용하려면 다음과 같이 할 수 있다 :

```
# iptables -A INPUT -m time --timestart 8:00 --timestop 18:00 --
days Mon,Tue,Wed,Thu,Fri -j ACCEPT

# iptables --list
Chain INPUT (policy ACCEPT)
target     prot opt source               destination
ACCEPT     all  --  anywhere             any-
where        TIME from 8:0 to 18:0 on Mon,Tue,Wed,Thu,Fri
```

time match가 지원하는 옵션은 다음과 같다 :

- --timestart value -> 최소 HH:MM
- --timestop value -> 최대 HH:MM
- --days listofdays -> 적용되는 요일 리스트, (대소문자 구분)
  - Mon
  - Tue
  - Wed

- Thu
- Fri
- Sat
- Sun

3.15. ttl patch

이 패치는 Harald Welte <laforge@gnumonks.org>에 의해 작성되었으며, TTL에 기초한 패킷을 match할 수 있도록 하는 새로운 match이다.

예를 들어, TTL이 5보다 적은 패킷을 로그에 기록하려면, 당신은 다음과 같이 할 수 있다 :

```
# iptables -A INPUT -m ttl --ttl-lt 5 -j LOG

# iptables --list
Chain INPUT (policy ACCEPT)
target     prot opt source               destination
LOG        all  --  anywhere             any-
where           TTL match TTL < 5 LOG level warning
```

ttl match가 지원하는 옵션은 다음과 같다.

- --ttl-eq value -> time to live 값과 일치
- --ttl-lt value -> TTL < value 한 것과 일치
- --ttl-gt value -> TTL > value 한 것과 일치

4. 새로운 **netfilter** 타겟

이 장에서, 새로운 netfilter 타겟의 사용에 관해 설명할 것이다. 이 패치들은 알파벳 순서로 나타낼 것이고, 부가적으로 연관된 다른 패치에 대한 설명은 하지 않는다. 하지만 다음 버전에서는 추가될 것이다.

일반적으로 타겟에 관해, 다음의 내용처럼 특별한 모듈에 대한 힌트를 얻을 수 있다 :

```
# iptables -j THE_TARGET_YOU_WANT --help
```

이는 보통의 iptables 도움 메세지를 보여주고, 그 끝에 “THE_TARGET_YOU_WANT”타겟을 설명한다.

4.1. ftos patch

이 패치는 Matthew G. Marsh <mgm@paktronix.com>에 의해 작성되었으며, 임의의 값으로 TOS 패킷을 셋팅할 수 있도록 하는 새로운 match이다.

예를 들어, 15의 outgoing 패킷의 모든 TOS를 셋팅하려면, 다음과 같이 할 수 있다.

```
# iptables -t mangle -A OUTPUT -j FTOS --set-ftos 15

# iptables -t mangle --list
Chain OUTPUT (policy ACCEPT)
target     prot opt source               destination
FTOS       all  --  anywhere             anywhere            TOS set 0x0f
```

FTOS target이 지원하는 옵션은 다음과 같다 :

- --set-ftos value -> 패킷 헤더의 TOS field를 어떤 값으로 설정. 이 값이 10진수가 될 수 있고 (ex: **32**) 16진수로도 될 수 있다. (ex: **0x20**)

4.2. IPV4OPTSSTRIP patch

이 패치는 Fabrice MARIE <fabrice@celestix.com>에 의해 작성되었으며 IPv4 패킷의 모든 IP 옵션을 제거(strip)할 수 있도록 하는 새로운 target이다.

다음과 같이 가장 간단하게 로드할 수 있다 :

```
# iptables -t mangle -A PREROUTING -j IPV4OPTSSTRIP

# iptables -t mangle --list
Chain PREROUTING (policy ACCEPT)
target     prot opt source               destination
IPV4OPTSSTRIP  all  --  anywhere             anywhere
```

이 타겟은 어떠한 옵션도 지원하지 않는다.

4.3. NETLINK patch

이 패치는 Gianni Tedesco <gianni@ecsc.co.uk>에 의해 작성되었으며 netlink 소켓을 통해 유저 영역으로 드롭된 패킷을 보낼 수 있도록 하는 새로운 타겟이다.

예를 들어, 모든 핑 패킷을 드롭하고 유저영역의 netlink 소켓으로 패킷들을 보내려면, 다음과 같이 할 수 있다 :

```
# iptables -A INPUT -p icmp --icmp-type echo-request -j NETLINK --nldrop
```

```
# iptables --list
Chain INPUT (policy ACCEPT)
target     prot opt source               destination
NETLINK    icmp --  anywhere             anywhere           icmp echo-
request nldrop
```

NETLINK 타겟이 지원하는 옵션은 다음과 같다 :

- --nldrop -> 패킷을 드롭한다.
- --nlmark <number> -> 패킷을 표시한다.
- --nlsize <bytes> -> 패킷 크기를 제한한다.

netlink socket에 대한 더 많은 정보를 원한다면, Netlink Sockets Tour (http://www.skyfree.org/linux/kernel_network/netlink.html)를 참고하라.

4.4. NETMAP patch

이 패치는 Svenning Soerensen <svenning@post5.tele.dk>에 의해 작성되었으며 원래의 호스트 주소를 유지하는 동안 네트워크 주소와 정적으로 1:1 매핑을 만들게 할 수 있는 새로운 타겟이다.

예를 들어, 1.2.3.0/24에서 5.6.7.0/24로 향하는 incomming 연결의 목적지를 변경하길 원한다면, 다음과 같이 할 수 있다 :

```
# iptables -t nat -A PREROUTING -d 1.2.3.0/24 -j NETMAP --to 5.6.7.0/24

# iptables -t nat --list
Chain PREROUTING (policy ACCEPT)
target     prot opt source               destination
NETMAP     all  --  anywhere             1.2.3.0/24         5.6.7.0/24
```

NETMAP 타겟이 지원하는 옵션은 다음과 같다 :

- --to address[/mask] -> 매핑할 네트워크 주소

4.5. SAME patch

이 패치는 Martin Josefsson <gandalf@wlug.westbo.se>에 의해 작성되었으며 SNAT와 유사하고 각각 연결에 대해 한 클라이언트에 같은 주소를 부여할 수 있는 새로운 타겟이다.

예를 들어, 연결에 대한 소스 주소를 1.2.3.4-1.2.3.7로 변경하려면 다음과 같이 할 수 있다 :

```
# iptables -t nat -A POSTROUTING -j SAME --to 1.2.3.4-1.2.3.7

# iptables -t nat --list
Chain POSTROUTING (policy ACCEPT)
target     prot opt source               destination
SAME       all  --  anywhere             anywhere           same:1.2.3.4-
1.2.3.7
```

SAME target이 지원하는 옵션은 다음과 같다 :

- --to <ipaddr>-<ipaddr> -> 소스에 매핑된 주소. 아마도 다수의 영역에 대한 한번 이상 기술되었을 것이다.
- --nodst -> 소스 선택에 대해 도착 IP를 사용하지 말라.

4.6. tcp-MSS patch

이 패치는 Marc Boucher <marc+nf@mbsi.ca>에 의해 작성되었으며 연결에 대한 최대 크기를 제어할 수 있도록, TCP SYN 패킷의 MSS 값을 변경하고 검사할수 있도록 하는 새로운 target이다.

Marc 자신이 설명한 바에 의하면, 이것은 해킹인데(THIS IS A HACK), ICMP Fragmentation이 패킷을 요구하는 것을 막는 뇌사 상태의 ISP들 또는 서버들을 극복하는데 사용된다.

전형적인 사용방법은 다음과 같은 것이다 :

```
# iptables -A FORWARD -p tcp --tcp-flags SYN,RST SYN -j TCPMSS --clamp-
mss-to-pmtu

# iptables --list
Chain FORWARD (policy ACCEPT)
target     prot opt source               destination
TCPMSS     tcp  --  anywhere             any-
where           tcp flags:SYN,RST/SYN TCPMSS clamp to PMTU
```

tcp-MSS target이 지원하는 옵션은 다음과 같다 (상호 배제) :

- --set-mss value 특정값으로 MSS 옵션을 명백히 셋팅
- --clamp-mss-to-pmtu MSS 값을 자동으로 고정시킴 (path_MTU - 40)

4.7. TTL patch

이 패치는 Harald Welte <laforge@gnumonks.org>에 의해 작성되었으며, 주어진 값에 의해 IP 패킷의 TTL 값을 증가/감소시키거나 유저가 셋팅할 수 있도록 하는 새로운 target이다.

예를 들어, 모든 outgoing 연결의 TTL값을 126으로 셋팅하려고 한다면, 다음과 같이 할 수 있다 :

```
# iptables -t mangle -A OUTPUT -j TTL --ttl-set 126

# iptables -t mangle --list
Chain OUTPUT (policy ACCEPT)
target     prot opt source               destination
TTL        all  --  anywhere             anywhere           TTL set to 126
```

TTL target이 지원하는 옵션은 다음과 같다 :

- --ttl-set value -> TTL을 <value>로 셋팅
- --ttl-dec value -> TTL을 <value>만큼 감소
- --ttl-inc value -> TTL을 <value>만큼 증가

4.8. ulog patch

이 패치는 Harald Welte <laforge@gnumonks.org>에 의해 작성되었으며, 표준 LOG target보다 진보된 로깅 메커니즘을 제공하는 새로운 match이다. 'libiptulog/'는 ULOG 메세지를 받는 라이브러리를 포함한다.

Harald는 ULOG에 대한 적절한 문서를 포함하는 web page (http://www.gnumonks.org/projects/ulogd)를 보유한다. 그래서 여기서는 설명할 특별한 내용은 없다.

5. 새로운 연결 추적 패치들

이 단락에서, 사용가능한 연결 추적/NAT 패치를 볼 수 있고, 그것을 사용하기 위해, 단순히 해당 모듈 (필요하다면 옵션으로)을 로딩한다.

5.1. eggdrop-conntrack patch

이 패치는 Magnus Sandin <magnus@sandin.cx>에 의해 작성되었으며 네트워크 eggdrop bot에 대한 연결 추적을 지원한다.

5.2. ftp-fxp patch

이 패치는 Magnus Sandin <magnus@sandin.cx>에 의해 작성되었으며 ftp 연결 추적에 지원되는 FXP를 더한다. NAT된 ftp 데몬에 대한 FXP화는 작동하지 않는다. FXP 추적을 가능하게 하기 위해, 다음과 같이 하라 :

```
# modprobe ip_conntrack_ftp.o fxp=1
```

이 패치는 보안 경고를 언급한다 : WARNING, 이 패치를 적용하는 것과 **WILL**를 가능하게 하는 것은 FTP 연결 추적에 제공되는 보안을 감소시킨다. 주의해서 사용하라 (당신이 어떻게 할수 있다는 범위 하에서만)

5.3. irc-conntrack-nat patch

이 패치는 Harald Welte <laforge@gnumonks.org>에 의해 작성되었으며 NAT와 연결 추적을 통해 작동하는 DCC를 가능하게 한다.

5.4. record-rpc patch

이 패치는 Marcelo Barbosa Lima <marcelo.lima@dcc.unicamp.br>에 의해 작성되었으며 TCP와 UDP를 사용하는 portmapper 요청을 추적하기 위해 netfilter를 허용한다.

5.5. snmp-nat patch

이 패치는 James Morris <jmorris@intercode.com.au>에 의해 작성되었으며 기초적인 SNMP를 NAT할 수 있는 netfilter를 제공한다. 이는 “basic” SNMP-ALG 형태이고, RFC 2962 (http://www.faqs.org/rfcs/rfc2962.html)에 설명되어 있다. 이것은 IP 레이어 NAT 매핑을 일치시키는 SNMP 페이로드 내부의 IP 주소를 변경하는데 작동한다.

5.6. talk-conntrack-nat patch

이 패치는 Jozsef Kadlecsik <kadlec@blackhole.kfki.hu>에 의해 작성되었으며 talk 연결을 추적하고 NAT하는 netfilter를 제공한다. 디폴트로 otalk (UDP port 517), talk (UDP port 518) 모두 지원한다. otalk/talk는 ip_conntrack_talk, ip_nat_talk 모듈의 모듈 파라미터에 의해 선택적으로 가능/불가능하게 될 수 있다. 이 옵션은 다음과 같다 :

- otalk = 0 | 1
- talk = 0 | 1

주어진 프로토콜에서 0은 지원하지 않음, 1은 지원함을 의미한다.

5.7. tcp-window-tracking patch

이 패치는 Jozsef Kadlecsik <kadlec@blackhole.kfki.hu>에 의해 작성되었으며, Guido van Rooij의 Real Stateful TCP Packet Filtering in IP Filter (http://www.iae.nl/users/guido/papers/tcp_filtering.ps.gz) 에 따라 TCP 연결 추적을 netfilter에게 허용한다. 이것은 window 크기조절을 지원하며, 이미 연결된 연결을 핸들링할수 있도록 한다.

이 패치는 "ftp-fixes" 패치가 적용되는 것을 요구한다. 아마도 그것은 요즘 표준 커널의 일부분일 것이다 ...

6. 새로운 IPv6 netfilter matches

In this section, we will attempt to explain the usage of new netfilter matches. The patches will appear in alphabetical order. Additionally, we will not explain patches that break other patches. But this might come later.

Generally speaking, for matches, you can get the help hints from a particular module by typing :

```
# ip6tables -m the_match_you_want --help
```

This would display the normal ip6tables help message, plus the specific "the_match_you_want" match help message at the end.

6.1. agr patch

This patch by Andras Kis-Szabo <kisza@sch.bme.hu> adds 1 new match :

- "agr" : lets you match the IPv6 packet based on it's addressing parameters.

This patch can be quite useful for people using EUI-64 IPv6 addressing scheme who are willing to check the packets based on the delivered address on a LAN.

For example, we will redirect the packets that have a correct EUI-64 address:

```
# ip6tables -N ipv6ok
# ip6tables -A INPUT -m agr -j ipv6ok
# ip6tables -A INPUT -s ! 3FFE:2F00:A0::/64 -j ipv6ok
# ip6tables -A INPUT -j LOG
# ip6tables -A ipv6ok -j ACCEPT

# ip6tables --list
Chain INPUT (policy ACCEPT)
target     prot opt source               destination
ipv6ok     all      anywhere             anywhere           AGR
ipv6ok     all     !3ffe:2f00:a0::/64    anywhere
```

```
LOG        all      anywhere             any-
where           LOG level warning

Chain ipv6ok (2 references)
target     prot opt source               destination
ACCEPT     all      anywhere             anywhere
```

This match hasn't got any option.

6.2. ipv6header patch

This patch by Andras Kis-Szabo <kisza@sch.bme.hu> adds a new match that allows you to match a packet based on its extension headers.

For example, let's drop the packets which have got hop-by-hop, ipv6-route headers and a protocol payload:

```
# ip6tables -A INPUT -m ipv6header --header hop-by-hop,ipv6-
route,protocol -j DROP

# ip6tables --list
Chain INPUT (policy ACCEPT)
target     prot opt source               destination
DROP       all      anywhere             any-
where           ipv6header flags:hop-by-hop,ipv6-route,protocol
```

And now, let's drop the packets which have got an ipv6-route extension header:

```
# ip6tables -A INPUT -m ipv6header --header ipv6-route --soft -j DROP

# ip6ptables --list
Chain INPUT (policy ACCEPT)
target     prot opt source               destination
DROP       all      anywhere             any-
where           ipv6header flags:ipv6-route soft
```

Supported options for the length match are :

- --header [!] headers -> You can specify the interested headers with this option. Accepted formats:
  - hop,dst,route,frag,auth,esp,none,proto
  - hop-by-hop,ipv6-opts,ipv6-route,ipv6-frag,ah,esp,ipv6-nonxt,protocol
  - 0,60,43,44,51,50,59

- --soft -> You can specify the soft mode: in this mode the match checks the existance of the header, not the full match!

6.3. ipv6-ports patch

This patch by Jan Rekorajski <baggins@pld.org.pl> adds 4 new matches :

- "limit" : lets you to restrict the number of parallel TCP connections from a particular host or network.
- "mac" : lets you match a packet based on its MAC address.
- "multiport" : lets you to specify ports with a mix of port-ranges and single ports for UDP and TCP protocols.
- "owner" : lets you match a packet based on its originator process' owner id.

These matches are the ports of the IPv4 versions. See the main documentation for the details!

6.4. length patch

This patch by Imran Patel <ipatel@crosswinds.net> adds a new match that allows you to match a packet based on its length. (This patch is shameless adaption from the IPv4 match written by James Morris <jmorris@intercode.com.au>)

For example, let's drop all the pings with a packet size greater than 85 bytes :

```
# ip6tables -A INPUT -p ipv6-icmp --icmpv6-type echo-request -m length --
length 85:0xffff -j DROP

# ip6ptables --list
Chain INPUT (policy ACCEPT)
target     prot opt source               destination
DROP       ipv6-icmp --  anywhere             anywhere           ipv6-
icmp echo-request length 85:65535
```

Supported options for the length match are :

- [!] --length length[:length] -> Match packet length against value or range of values (inclusive)

Values of the range not present will be implied. The implied value for minimum is 0, and for maximum is 65535.

7. 새로운 IPv6 netfilter targets

In this section, we will attempt to explain the usage of new netfilter targets. The patches will appear in alphabetical order. Additionally, we will not explain patches that break other patches. But this might come later.

Generally speaking, for targets, you can get the help hints from a particular module by typing :

```
# ip6tables -j THE_TARGET_YOU_WANT --help
```

This would display the normal iptables help message, plus the specific "THE_TARGET_YOU_WANT" target help message at the end.

7.1. LOG patch

This patch by Jan Rekorajski <baggins@pld.org.pl> adds a new target that allows you to LOG the packets as in the IPv4 version of iptables.

The examples are the same as in iptables. See the man page for details!

7.2. REJECT patch

This patch by Harald Welte <laforge@gnumonks.org> adds a new target that allows you to REJECT the packets as in the IPv4 version of iptables.

The examples are the same as in iptables. See the man page for details!

8. 새로운 IPv6 연결 추적 패치들

The connection tracking hasn't supported, yet.

9. Contribution

9.1. 새로운 확장에 대한 공헌

Netfilter core-team은 항상 새로운 확장/버그 픽스를 환영한다. 이 부분에서 우리는 새로운 확장이 patch-o-matic에 포함되어 쉽게 패키징될 수 있는가에 대해서는 관심이 없다. 하지만 이는 이 하우투의 다음 버전에 포함될 것이다.

먼저, 새로운 확장/버그 픽스를 하려는 사람은 Netfilter Hacking HOWTO (http://netfilter.samba.org/unreliable-guides/netfilter-hacking-HOWTO/index.html)에 익숙해져야 할 것이다.

Rusty는 netfilter에 대한 새로운 패치를 어떻게 하는가에 대한 가이드라인을 썼다. 이는 여기에서 볼 수 있다:

```
/path/to/netfiltercvs/netfilter/userspace/patch-o-matic/NEWPATCHES
```

또는 온라인상의 최신 버전은 여기에 있다 : NEWPATCHES (http://www.samba.org/cgi-bin/cvsweb/~checkout~/netfilter/userspace/patch-o-matic/NEWPATCHES).

마지막으로, netfilter-devel 메일링 리스트에 참가하는 것은 좋은 아이디어다. 어떻게 참가하는가에 대한 더 많은 정보는 netfilter 홈페이지에서 볼 수 있다.

9.2. 이 하우투에 대한 공헌

이 하우투를 업데이트하는 것은 환영한다. 그렇게 하기 위해서 추천되는 방법은 netfilter-devel 메일링 리스트로 이 문서의 SGML 관리자에게 패치를 보내는 것이다.

9.3. 역자의 말 (-_-;)

처음으로 DocBook으로 만들어 보는 문서라 허접함이 많습니다. 또 번역에도 스스로 만족이 안되는 부분이 곳곳에 보입니다(실력이 딸려서 --;). 번역에 이상이 있거나 수정할 점이 있으면 제 메일로 보내주시길 바랍니다. 이상 DeepBlue의 허접한 번역이었습니다 -_-;;