목차

DPI(Deep Packet Inspection) 우회 3
관련프로그램 3
MTU 조정 방법 3
notsodeep을 이용한 방법 3
라이브러리 의존성 설치 3
바이너리 빌드 3
iptables 규칙 적용 4
firewalld rule 4
서비스등록 형식 4
zapret을 이용한 방법 5
zapret 프로젝트를 clone받음 5
라이브러리 의존성 설치 5
바이너리 빌드 5
설치 5
실제 우회 규칙 적용하기 6

DPI(Deep Packet Inspection) 우회

관련프로그램

- https://secretsni.kilho.net/ - SecretSNI(길호넷)
- https://github.com/ValdikSS/GoodbyeDPI - GoodbyeDPI
- https://github.com/Include-sys/GUI-for-GoodbyeDPI - GoodbyeDPI-GUI
- https://safevisit.org/ko - SafeVisit

내용출처 : https://terzeron.com/confluence/pages/viewpage.action?pageId=34734083

- Windows에서는 GoodbyeDPI나 GUI for GoodbyeDPI 패키지를 이용하면 손쉽게 DPI를 우회할 수 있음
- Linux에서는 쉽게 DPI를 우회할 수 있는 방법이 없으나, MTU를 조정하는 방법, zapret을 이용한 방법과 notsodeep을 이용한 방법 등이 존재하므로 세 가지 중 한 가지를 선택해서 사용하면 됨
- notsodeep과 zapret 두 가지 방법 모두 iptables의 라우팅 규칙을 이용하고 있음
- zapret의 경우 기술문서가 러시아어로 작성되어 있어서 적용에 주의가 필요함

MTU 조정 방법

- 이 방법은 시스템에 충격을 줄 수 있기 때문에 권장하는 방법이 아님
- 우선 MTU 크기를 220 또는 400 정도의 값으로 조정하는 방법으로 우회 가능한지 확인해볼 것
  - https://www.clien.net/service/board/cm_mac/13158762에 언급된 내용에 따르면 브라우저마다 MTU 값이 다르게 요구된다고 하는데, 가장 성능이 낮게 나오긴 하지만 220으로 설정하면 어지간한 DPI는 우회 가능함
- 다음 문서를 참조할 것
  - 강제로 네트워 인터페이스의 MTU 조정하기

notsodeep을 이용한 방법

참고: https://github.com/farukuzun/notsodeep

라이브러리 의존성 설치

```
# Debian 계열
sudo apt-get install libnetfilter-queue-dev libc6-dev

# RedHat 계열
yum install libnetfilter_queue-devel compat-glibc
```

바이너리 빌드

```
git clone https://github.com/farukuzun/notsodeep.git
cd notsodeep
make
sudo nohup ./notsodeep &
```

iptables 규칙 적용

iptables로 TCP connection handshake의 패킷들을 NFQUEUE 응용프로그램인 notsodeep에서 처리할 수 있도록 넘기는 규칙을 적용함

```
sudo iptables -A INPUT -p tcp --tcp-flags SYN,ACK SYN,ACK --sport 443 -j
NFQUEUE --queue-num 200 --queue-bypass
sudo iptables -t raw -I PREROUTING -p tcp --sport 80 --tcp-flags SYN,ACK
SYN,ACK -j NFQUEUE --queue-num 200 --queue-bypass
```

특정 서버와의 통신인 경우에는, -s 옵션을 이용하여 서버 IP를 적어주면 그 서버와의 통신에 대해서만 적용됨 특정 서버를 notsodeep으로 필터링하지 않는 예외 규칙을 적용하려면 다음 명령을 사용할 수 있음

```
sudo iptables -I INPUT -s <제외할IP> -p tcp --tcp-flags SYN,ACK SYN,ACK --
sport 443 -j ACCEPT
```

-A 옵션 대신 -I 옵션을 사용하여 규칙 테이블 맨 위에 등록해두는 것이 중요함

firewalld rule

```
firewall-cmd --permanent --direct --passthrough ipv4 -I INPUT 1 -p tcp --
tcp-flags SYN,ACK SYN,ACK --sport 443 -j NFQUEUE --queue-num 200 --queue-
bypass
firewall-cmd --permanent --direct --passthrough ipv4 -t raw -I PREROUTING 1
-p tcp --sport 80 --tcp-flags SYN,ACK SYN,ACK -j NFQUEUE --queue-num 200 --
queue-bypass

firewall-cmd --reload
```

서비스등록 형식

```
cd /tmp
git clone https://github.com/farukuzun/notsodeep.git
cd notsodeep
make
cd ..
```

```
sudo su
cp -R notsodeep /opt
cp /opt/notsodeep/notsodeep.service /etc/systemd/system/
systemctl enable notsodeep.service
iptables -A INPUT -p tcp --tcp-flags SYN,ACK SYN,ACK --sport 443 -j NFQUEUE
--queue-num 200 --queue-bypass
iptables -t raw -I PREROUTING -p tcp --sport 80 --tcp-flags SYN,ACK SYN,ACK
-j NFQUEUE --queue-num 200 --queue-bypass
iptables-save > /etc/iptables/iptables.rules
systemctl enable iptables
systemctl start iptables
systemctl start notsodeep.service
```

zapret을 이용한 방법

링크 : https://github.com/bol-van/zapret 러시아어로 되어있음

zapret 프로젝트를 clone받음

```
git clone https://github.com/bol-van/zapret
```

라이브러리 의존성 설치

```
apt install lsb-core libnetfilter-queue-dev ipset
```

바이너리 빌드

```
cd zapret
cd nfq
make
cd tpws
make
```

설치

```
sudo cp -r zapret /opt
cd /opt/zapret
sudo cp /opt/zapret/init.d/debian7/zapret /etc/init.d/
sudo /etc/init.d/zapret start
sudo /opt/zapret/nfq/nfqws --daemon --qnum=200 --wsize=4 --hostspell=HoSt --
hostdot --host-tab --hostnospace ...
```

실제 우회 규칙 적용하기

- iptables.txt를 살펴보면 다양한 방법들이 제시되어 있음
  - https://raw.githubusercontent.com/bol-van/zapret/master/iptables.txt
- iptables 명령에 대한 설명은 다음 문서를 참조
  - iptables 명령 소개

window size 축소

- TCP window size를 줄이면 HTTP 요청이 여러 개의 TCP 세그먼트로 나뉘어 전송되기 때문에 DPI를 우회할 수 있음 (DPI 시스템에 따라 가능할 수도 있고 아닐 수도 있음)

```
sudo nfqws
iptables -t raw -I PREROUTING -p tcp --sport 80 --tcp-flags SYN,ACK SYN,ACK
-j NFQUEUE --queue-num 200 --queue-bypass
iptables -t raw -I PREROUTING -p tcp --sport 80 --tcp-flags SYN,ACK SYN,ACK
-m set --match-set zapret src -j NFQUEUE --queue-num 200 --queue-bypass
```

Host: 헤더 항목 변경

- Host: 헤더 항목의 이름을 host: 로 변경하면 DPI를 우회할 수 있음 (DPI 시스템에 따라 가능할 수도 있고 아닐 수도 있음)

```
sudo nfqws
iptables -t mangle -I POSTROUTING -p tcp --dport 80 -j NFQUEUE --queue-num
200 --queue-bypass
iptables -t mangle -I POSTROUTING -p tcp --dport 80 -m set --match-set
zapret dst -j NFQUEUE --queue-num 200 --queue-bypass
iptables -t mangle -I POSTROUTING -p tcp --dport 80 -m set --match-set
zapret dst -m connbytes --connbytes-dir=original --connbytes-mode=packets --
connbytes 1:5 -j NFQUEUE --queue-num 200 --queue-bypass
```

TPROXY 사용하는 방법

```
sysctl -w net.ipv4.ip_forward=1
iptables -t nat -A POSTROUTING -o eth0 -j MASQUERADE

ip -f inet rule add fwmark 1 lookup 100
ip -f inet route add local default dev lo table 100

# prevent loop
iptables  -t filter -I INPUT -p tcp --dport 1188  -j REJECT
iptables  -t mangle -A PREROUTING -i eth1 -p tcp --dport 80 -j MARK --set-
mark 1
```

```
iptables  -t mangle -A PREROUTING -i eth1 -p tcp --dport 80 -j TPROXY --
tproxy-mark 0x1/0x1 --on-port 1188

iptables  -t mangle -A PREROUTING -i eth1 -p tcp --dport 80 -m set --match-
set zapret dst -j MARK --set-mark 1
iptables  -t mangle -A PREROUTING -i eth1 -p tcp --dport 80 -m mark --mark
0x1/0x1 -j TPROXY --tproxy-mark 0x1/0x1 --on-port 1188
```

DNAT 사용하는 방법

```
# run tpws as user "tpws". its required to avoid loops.
sudo -u twps
sysctl -w net.ipv4.conf.eth1.route_localnet=1
iptables -t nat -I PREROUTING -p tcp --dport 80 -j DNAT --to 127.0.0.1:1188
iptables -t nat -I OUTPUT -p tcp --dport 80 -m owner ! --uid-owner tpws -j
DNAT --to 127.0.0.1:1188
```