트래픽 분석을 통한 서비스거부공격 추적

2003. 1. 15

정현철, hcjung@kisa.or.kr
변대용, yong@kisa.or.kr

기술자문 : **최우형**(Cisco Systems Korea, Ltd), whchoi@cisco.com

보안 분야에서 늘 회자되고 있는 이슈 중의 하나가 2000년 초 야후, 아마존 등 세계 주요 인터넷 사이트에 대한 분산서비스거부공격으로 인한 피해일 것이다. 뿐만 아니라 지난 2002년 말경에도 전세계의 최상위 네임서버가 서비스거부공격을 당하는 등 서비스거부공격의 위협은 항상 우리 곁에 도사리고 있다.

하지만 서비스거부공격은 공격은 방어가 힘들뿐만 아니라 공격자의 추적 또한 대단히 힘들다. 서비스거부공격에 대한 방어와 추적이 힘든 가장 큰 이유 중의 하나가 IP를 Spoofing하여 공격하기 때문이다.

본 문서에서는 Spoofing된 서비스거부공격이 발생될 경우 라우터, NMS 장비, 기타 공개 도구 등을 이용하여 공격 근원지를 추적할 수 있는 방법을 기술하였다. 물론, 본 문서에서 기술한 방법에는 많은 제약조건이 따르지만, IPv4 인터넷 프로토콜이 Spoofing에 대한 근본적인 방지와 추적 방안이 없는 상황에서 최선의 방법이라 생각한다.

마지막으로 본 문서 작성을 위해 현장의 경험과 지식을 공유해 주시고, 많은 조언을 아끼지 않은 최우형님에게 감사의 말씀드립니다.

1. 개요

인터넷 보안 체계를 공격하는 최근의 경향은, 시스템 또는 네트워크 자원을 공격 대상으로 사용 가능한 자원을 모두 소비해서 실제 자원을 사용해야 하는 사용자가 자원을 사용할 수 없게 하는 서비스거부공격(DoS: Denial of Service)이 증가하고 있다는 것이다.

서비스거부공격은 대역폭, 프로세스 처리 능력, 기타 시스템 자원을 고갈시킴으로써 정상적인 서비스를 할 수 없도록 하는 공격 형태이다. 대역폭을 목표로 한 공격은 시스템에 대량의 TCP, UDP 또는 ICMP 패킷을 보내는 공격이며, 프로세스 처리 능력 등 시스템 자원 고갈을 목표로 하는 공격에는 TCP 옵션 변경, 비정상적인 패킷 사이즈 등 비정상적인 패킷을 송신하여 자원을 고갈시키거나 비정상적으로 시스템을 멈추게 한다. 또한 최근에는 다수의 시스템에 공격 Agent를 설치한 후, 동시에 공격함으로써 엄청난 파괴력을 지닌 분산서비스거부공격도 일반화되는 추세이다.

다음은 지금까지 잘 알려진 서비스거부공격들이다.

- o Smurf 공격 : ICMP echo 요청을 브로드캐스트 한다.
- o Fraggle 공격 : UDP echo 요청을 7번 포트로 브로드캐스트 한다.
- o TearDrop 공격 : 패킷들로 분리된 IP Datagram의 조합 시 필요한 각 패킷의 offset값을 겹치게 만들어 시스템이 충동을 일으키거나 재부팅되게 한다.
- o Land 공격 : 출발지와 목적지의 IP 주소를 동일한 SYN 패킷을 보낸다.
- o SYN Flooding : 공격대상 시스템에 연속적인 SYN패킷을 보낸다.
- o DDoS 공격 : 다수의 시스템에서 DoS 공격 패킷을 보낸다.

위의 서비스거부공격들에서 대부분 자신의 주소를 무작위로 속이거나 공격 목표 시스템으로 위장하여 공격하는 경향을 찾아 볼 수 있다. 일반적으로 서비스거부공격이 자신의 IP 주소를 Spoofing하는 특성상 추적은 상당히 힘들다. 추적이 어려운 원인에는 다음과 같은 것들을 들 수 있다.

- o 공격이 여러 개의 소스에서 이루어진다
- o 공격이 특정 소스에서 매우 적은 수의 패킷을 사용할 수 있기 때문에 공격을 받은 후 추적하는 것이 거의 불가능하다.
- o 대역폭 사용량에는 큰 영향을 주지 않는 SYN 집중 공격 같이 서버 자원을 소비하는 다양한 형태의 트래픽이 공격에 포함될 수 있다

- o 트래픽의 유형에 유효한 대상 서비스가 포함될 수 있지만, 이는 ISP의 DNS 서버 또는 인터넷에 연결된 조직에 보내는 HTTP 요청과 같이 비즈니스에 있어서 필수적인 서비스이기 때문에 차단할 수 없다.
- o 공격은 TCP, UDP, ICMP 또는 다른 IP 프로토콜을 포함한 다양한 형태의 패킷을 이용할 수 있다.

이러한 특성을 가진 서비스거부공격의 근원지를 추적하기 위한 패킷추적은 보안전문가들의 연구대상이었다. 예를 들어, 불법적인 IP 주소를 ingress filtering을 통하여 차단하여 유효한 IP 주소만 사용하게 함으로써 공격의 근원지 추적을 용이하게 하거나, 라우터의 패킷을 저장(logging)하고 data mining 기술을 이용하여 logging된 패킷정보를 종합 분석하여 공격의 근원지를 추적하는 기술, 거쳐간 라우터의 주소를 패킷정보에 추가하는 알고리즘 개발 등 몇몇 추적 방법이 발표되었다. 하지만 이들 방법 역시 자체적으로 공격자 추적을 위해 현실적으로 만족스러운 해결책이 되기에는 부족한 부분이 많다. 게다가 추적 작업을 수행하려면 네트워크 장비 및 자원에 액세스해야 하고 이러한 이유로 인해 추적 작업이 현재 네트워크 또는 ISP 네트워크 경계와 같은 특정 네트워크에 한정될 수밖에 없었다.

본 문서에서는 ISP 및 기관내 네트워크 관리자의 공조하에서 역추적을 실현하기 위해 현 시점에서 사용할 수 있는 방법에 대해 소개하고자 한다.

2. 트래픽 흐름 분석을 통한 추적

버퍼오버플로우 공격 등 공격 대상에 침입하기 위한 공격은 공격 대상 서버로부터 응답을 받아야 하므로 공격 근원지를 속이는 경우가 드물다. 그렇지만 서비스거부 공격의 경우는 공격 대상으로부터 응답 패킷을 받을 필요가 없고, 대부분의 서비스 거부공격자는 자신의 공격 출처를 속이기 위한 목적으로 IP spoofing 기법을 추가 사용하여 공격의 효과를 증대시킨다.

따라서, 공격을 받는 피해기관에서는 무작위로 Spoofing된 공격 트래픽에 근원지 추적 및 대응을 엄두도 못내고 속수무책 당하는 경우가 많다.

Spoofing된 공격 트래픽으로부터 IP소스 주소를 이용하여 공격자를 추적할 수는 없지만 각 게이트웨이(라우터) 단에서 네트워크 트래픽 량이나 Spoofing된 패킷의 흐름을 관찰함으로써 실제 공격이 발생된 근원지를 추적할 수 있다.

트래픽 흐름 분석(Traffic Flow Analysis)은 SYN 패킷으로 시작해서 세션 내의 최종 FIN ACK까지의 단일 TCP 세션과 같은 개별적인 네트워크 트래픽을 모니터링하는 데 사용하고, 이러한 네트워크 관리를 위한 흐름 분석을 표준화하려는 노력은 계속되어 왔다. 그 중 하나는 트래픽 흐름 측정 MIB(Management Information Base)를 통한 표준화 노력이고 다른 한 가지는 Cisco의 NetFlow를 통한 표준화 노력이다.

"트래픽 흐름 측정"(RFC 2720)은 작동(behavior) 모델, 네트워크 용량, 네트워크 성능, 서비스 품질 및 네트워크 사용량의 분배와 같은 네트워크 관리 정보를 시스템 관리자에게 제공할 수 있도록 설계되었다.

Cisco에서 개발한 NetFlow는 IP 트래픽 전용이고 Cisco 장비에 이미 사용되고 있다는 것만 제외하면 트래픽 흐름 측정 MIB와 비슷한 기능을 제공한다. 어떤 프로그램을 사용하건 소스 및 대상 주소, 각 흐름의 바이트 수 및 패킷 수, 트래픽의 시작 인터페이스 및 업스트림 피어 정보 등을 캡처 필터를 사용하여 모니터링할 수 있다. 업스트림 피어 정보는 Spoofing된 패킷을 추적하여 실제 근원지를 추적하는 작업에서 필수적인 정보이다.

NetFlow는 TCP 플래그는 물론이고 포트 정보에 대한 TCP 및 UDP 패킷도 검사한다. 이런 작업은 공격 트래픽을 추적할 때 매우 유용하다. 또한 NetFlow는 라우터

를 통해 해당 flow의 소스 인터페이스 같은 세부적인 피어 정보와 next-hop 및 AS 피어 정보 등의 라우팅 정보를 제공한다.

Spoofing된 서비스거부공격 근원지 네트워크는 트래픽 흐름 MIB를 사용한 SNMP 접근법과 Cisco 라우터의 NetFlow 접근법 등을 통하여 파악이 가능하다. 이렇게 공격 근원지 네트워크(기관)이 파악되면 해당 Subnet에서 어떤 호스트가 트래픽을 발생시키는지 찾아야 할 것이다. 이 과정은 내부 네트워크에서 공격 패턴을 가진 MAC 주소를 찾고 이 MAC 주소가 실제 어느 IP 주소인지, 또는 스위치에서 어느 포트에 연결되어 있는지를 찾음으로써 실제 공격 시스템까지 추적이 가능할 것이다. 마지막으로 해당 시스템에서 공격 프로세스와 공격에 이용되는 파일들은 기존의 시스템 분석 방법들을 활용할 수 있다.

지금까지 언급된 서비스거부공격의 인지와 추적은 다음과 같이 요약될 수 있다.

① SNMP 기반의 모니터링 툴이나 시스템의 모니터링 명령어를 통한 공격사실 인지
  - 모니터링 툴 : MRTG 등 SNMP 기반의 NMS 장비
  - 시스템 명령어 : snoop, tcpdump 등
② 각 지점별 라우터 네트워크 인터페이스에서의 트래픽 분석을 통한 경로 추적
  - NetFlow, CEF 기능 사용
③ 공격 근원지 Subnet 확인
④ 해당 Subnet 내의 트래픽 발생 시스템 추적
  - MAC 주소 추적
⑤ 해당 시스템에서 공격 도구 탐지 및 삭제
  - 프로세스 확인 등 시스템 분석

그러면 각각의 절차에 대해서 보다 상세하게 알아보도록 하자.

가. 네트워크 모니터링을 통한 공격사실 인지

서비스거부공격의 인지는 시스템이나 네트워크가 느려지거나 접속 불능 상태 등 직관적으로 인지할 수도 있지만 많은 기관에서 운용하고 있는 MRTG와 같은 SNMP 기반의 네트워크 모니터링 툴을 활용할 경우 보다 정확하고 계량적인 분석이 가능하다. 또한 직관적으로 시스템이나 네트워크의 이상징후를 감지하였을 경우 이를 확인하기 위해서 라우터에서 CPU나 메모리의 사용량 등으로 정확한 판단을 할 수 있다.

□ MRTG를 이용한 모니터링

o MRTG 개요

Muti Router Traffic Grapher (MRTG)는 현재 세계에서 트래픽모니터링 및 네트워크 관리를 위해서 사용되고 있는 가장 범용의 툴이다. MRTG는 트래픽관리서버(MRTG가 설치되어 운영되고 있는 서버)에서 주기적으로 실행된 결과를 GIF 및 Portable Network Graphics(PNG)의 그래픽파일을 포함한 HTML파일을 자동으로 생성하여 웹브라우저를 통해서 네트워크 트래픽을 분석, 관리할 수 있다.

MRTG가 네트워크 트래픽 모니터링이나 분석을 위한 도구로만 사용되는 것은 아니다. MRTG는 SNMP(Simple Network Monitoring Protocol)에서 지원되는 다양한 장비(서버, 라우터, 스위치 등)의 MIB(Management Information Base) 값을 가져와서 장비의 사용량 등을 분석할 수도 있다.

즉, MRTG로 가능한 여러 가지 작업들은 다음과 같다.

o 네트워크 트래픽 모니터링 및 분석
o 서버 트래픽 모니터링 및 분석
o CPU 사용량 모니터링 및 분석
o MEMORY 모니터링 및 분석
o DISK 사용량 모니터링 및 분석
o 기타 MIB에서 가져올 수 있는 다양한 정보의 모니터링 및 분석

물론, MRTG를 통해서 분석할 수 있는 가장 대표적인 목적은 트래픽을 분석하기 위한 모니터링이다. 하지만, MRTG는 SNMP라는 네트워크관리 프로토콜을 사용하며, SNMP는 MIB이라는 자원의 객체데이터베이스에 정의된 값들을 가져오거나 설정이 가능하기 때문에 MRTG는 이들을 통한 다양한 객체의 분석 및 모니터링이 가

능하다는 것이다. 또한 이러한 분석과 모니터링은 일, 주, 월, 연간으로 수행이 가능하다.

□ MRTG 설치

우선 최신 버전의 MRTG 설치 및 MRTG 관련 자료는 아래의 URL을 참조, 설치한다.

http://www.mrtg.co.kr/mrtg/mrtg_index.html
http://people.ee.ethz.ch/~oetiker/webtools/mrtg/

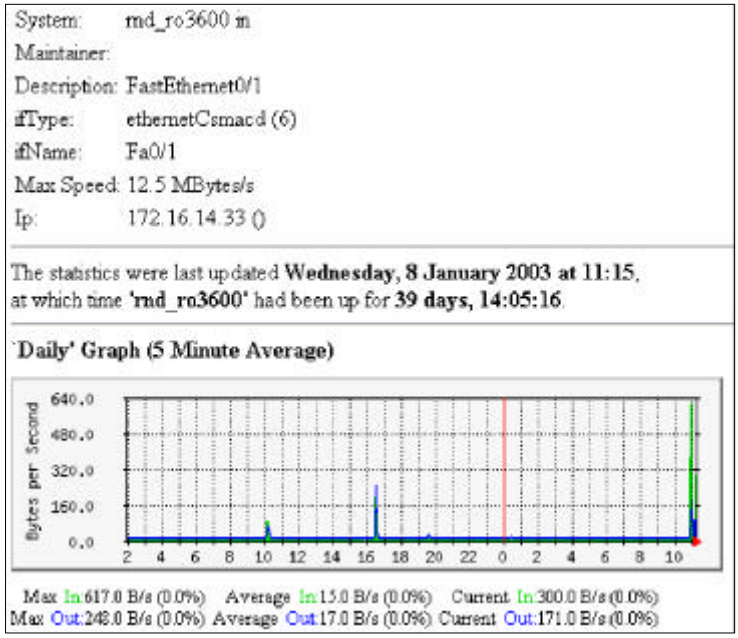
MRTG의 분석 대상이 될 네트워크 장비의 SNMP를 enable 시킨다.

시스코 사의 라우터 경우, 아래의 명령어를 사용하여 MRTG 서버와 대상 시스템(이 경우 라우터)간의 SNMP 통신을 위한 커뮤니티 이름을 설정하고 SNMP를 enable 한다.

```
rnd_ro3600(config)#snmp-server community [커뮤니티 이름]
※[커뮤니티 이름] : SNMP에서의 인증을 위한 이름 입력, 기타 정보는 '?' 입력
```

□ MRTG 모니터링

앞에서 언급한 바와 같이 네트워크 트래픽 모니터링은 아래의 그림처럼 MRTG의 그래프 지원을 받아 웹브라우저를 통해 가능하다.



그림의 상단부분은 SNMP 정보를 바탕으로 한 모니터링 대상의 간략한 설명이며, 그래프의 Y축은 BPS(Bytes per Second)값으로 CPU, Memory, DISK 사용량 등으로 설정 가능하다.

□ MRTG 대상 장비 분석

MRTG 등 네트워크 모니터링 툴에서 네트워크 트래픽의 이상 징후가 보이면, 해당 장비로 접속하여 관련 정보를 확인, 분석한다.

시스코 사의 라우터 경우, 아래에서 보이는 것과 같이 'show processes [cpu | memory]' 명령어를 사용하여 시스템의 CPU, memory 사용량을 확인한다.

```
rnd_ro3600#show processes cpu
CPU utilization for five seconds: 3%/2%; one minute: 2%; five minutes: 2%
 PID  Runtime(ms)  Invoked  uSecs   5Sec   1Min   5Min TTY Process
   1          80   684071      0  0.00%  0.00%  0.00%   0 Load Meter
   2         764      436   1752  0.00%  0.00%  0.04% 130 Virtual Exec
   3      933668   347360   2687  0.00%  0.02%  0.00%   0 Check heaps
   4           0        1      0  0.00%  0.00%  0.00%   0 Chunk Manager
   5          16       30    533  0.00%  0.00%  0.00%   0 Pool Manager
   6           0        2      0  0.00%  0.00%  0.00%   0 Timers
   7           0        2      0  0.00%  0.00%  0.00%   0 Serial Backgroun
   8        3096   683916      4  0.00%  0.00%  0.00%   0 ALARM_TRIGGER_SC
   9           0        1      0  0.00%  0.00%  0.00%   0 OIR Handler
  10          44   114005      0  0.00%  0.00%  0.00%   0 Environmental mo
  11      571788  2000308    285  0.00%  0.00%  0.00%   0 ARP Input
```

여기서 주의깊게 살펴보아야 할 부분은 파란색으로 표시된 부분이다. 평상시의 CPU 사용량을 고려해서 갑작스런 CPU 사용이 높을 경우 비정상적인 트래픽의 발생을 의심해 보아야 한다. SYN flooding과 같은 많은 서비스거부공격은 실제 대규모 패킷으로 대역폭을 점유하기 보다는 작은 패킷들을 다량으로 보내어 PPS(Packet Per Second)를 높이고 결과적으로 라우팅 장비나 대상 시스템에서 많은 CPU를 소모하게 되는 경우가 많다.

최근 5초 동안의 CPU 활용도(CPU utilization for five seconds)에서 처음 숫자는 전체 CPU 사용률을 명기하고 있고, 두 번째 숫자는 인터럽트 레벨에서 사용된 CPU 시간의 비율(캐쉬 사용 비율)을 의미한다. 정상적인 경우 두 숫자가 거의 비슷

하지만 앞의 숫자가 월등히 높을 경우, 즉 캐쉬 사용이 적은 경우는 IP Spoofing을 의심해 보아야 한다.

또한 MRTG에서 이상징후를 보인 인터페이스의 네트워크 트래픽 관련 정보를 확인한다.

```
rnd_ro3600#sh int fa0/0
FastEthernet0/0 is up, line protocol is up
  Hardware is AmdFE, address is 0002.4bb8.44c1 (bia 0002.4bb8.44c1)
  Internet address is 172.16.14.2/27
  MTU 1500 bytes, BW 100000 Kbit, DLY 100 usec,
     reliability 255/255, txload 1/255, rxload 1/255
  Encapsulation ARPA, loopback not set
  Keepalive set (10 sec)
  Full-duplex, 100Mb/s, 100BaseTX/FX
  ARP type: ARPA, ARP Timeout 04:00:00
  Last input 00:00:00, output 00:00:00, output hang never
  Last clearing of "show interface" counters never
  Queueing strategy: fifo
  Output queue 0/40, 0 drops; input queue 0/75, 532 drops
  5 minute input rate 124000 bits/sec, 258 packets/sec
  5 minute output rate 1000 bits/sec, 1 packets/sec
     2851233321 packets input, 3527917296 bytes
     Received 338930 broadcasts, 0 runts, 0 giants, 0 throttles
     388149 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored
     0 watchdog
     0 input packets with dribble condition detected
     15687666 packets output, 521854055 bytes, 0 underruns(0/0/0)
     0 output errors, 0 collisions, 3 interface resets
     0 babbles, 0 late collision, 0 deferred
     0 lost carrier, 0 no carrier
     0 output buffer failures, 0 output buffers swapped out
```

라우터에서 해당 인터페이스의 네트워크 트래픽 정보 확인으로 서비스거부공격이 진행되고 있다고 판단되면, 이제 공격에 사용되는 시스템을 추적해 보자.

나. NetFlow를 이용한 공격경로 추적

전통적인 MRTG, cricket 등과 같은 SNMP기반의 트래픽 모니터링 툴들은 비정상적인 트래픽 증가를 탐지해서 보안 사고가 발생 가능성을 경고할 수 있다. 하지만 SNMP 기반의 툴들은 트래픽 양의 변화와 수준에 대한 정보만을 제공한다. 보안 목적 특히 공격자를 추적하기 위해서는 이 정보만으로는 부족하다. 또한, Sniffer 등과 같은 툴들은 훨씬 자세한 정보를 제공하지만 너무 많은 데이터량으로 인해 분석이 곤란한 경우가 발생될 수도 있고 패킷 캡쳐링을 위해 많은 대역폭을 소모하여 문제가 발생될 수도 있다. 따라서, DoS와 같은 보안사고 발생시 네트워크 모니터링을 위해서 트래픽 량과 데이터 내용에 대한 어느 정도 수준의 정보를 제공할 수 있는 분석 툴이 필요하다.

IP Spoofing을 이용한 전형적인 서비스거부 공격이 발생되었을 경우 공격자를 추적하기 위해 가장 먼저 분석해야 할 장비가 라우터일 것이다. 특히, Cisco 라우터에서는 네트워크 흐름을 프로파일(profile)할 수 있는 NetFlow를 제공하고 있어 실제 공격자 추적에 매우 많은 도움을 준다.

□ CEF 및 NetFlow 설정

NetFlow는 라우터를 통하는 트래픽 흐름(flow)을 매핑할 수 있는 수단을 제공한다. 이 정보는 네트워크 대역폭이나 라우터의 성능 등을 업그레이드하는데 반영하거나 트래픽 패턴의 분석, 보안 검토 등에 활용될 수도 있다.
NetFlow 스위칭은 Cisco 라우터에서 사용가능한 스위칭 모드 중의 하나인데, 다른 스위칭 모드에 비해 추가적인 메모리와 CPU 자원을 소모한다. 따라서 NetFlow를 설정하기 전에 자신의 라우터에서 요구되는 자원들에 대해 이해하고 있어야 한다.
NetFlow 캐쉬의 크기는 필요에 따라 늘이거나 줄일 수 있지만 기본적으로 64K(65,535)개의 flow 캐쉬 엔트리를 유지한다. 각 캐쉬 엔트리가 약 64byte 정도의 스토리지를 사용하므로 기본 설정 상태일 경우 약 4MB의 DRAM이 필요하게 된다.

NetFlow는 CEF(Cisco Express Forwarding) 또는 dCEF(distributed Cisco Express Forwarding) 기능이 설정된 상태에서 사용가능하다.

CEF 기능은 7000 라우터에서는 IOS 11.1CC 이상, 그 외의 라우터에서는 IOS 12.0 이상에서 지원 가능하다. 또한 NetFlow도 7000 라우터에서는 11.1CA나 CC 이상, 그 외의 라우터에서는 12.0T 이상에서 지원가능하다.

CEF는 빠른 스위칭 기능뿐만 아니라 안정성도 보장해 주고 있어 일반적으로 효율

성을 높이기 위해 사용된다. 또한 CEF는 RPF(Reverse Path Forwarding)과 같은 보안 관점에서의 잇점도 제공한다. RPF는 내부 주소를 가지고 외부 네트워크 인터페이스에서 들어오는 트래픽을 차단할 수 있도록 한다. 이처럼 CEF는 효율성과 보안성 두가지 측면을 만족시켜 주고 있어 가능한 사용을 권고하고 있다.

CEF 또는 dCEF는 global config 모드에서 다음의 명령을 입력함으로써 enable 시킬 수 있다.

```
router(config)#ip cef
또는
router(config)#ip cef distributed
```

CEF가 정상적으로 설정되어 있는지 확인하기 위해서 다음과 같은 명령을 사용할 수 있다.

```
rnd_ro3600#sh ip cef
Prefix              Next Hop            Interface
0.0.0.0/0           172.16.14.1         FastEthernet0/0
0.0.0.0/32          receive
172.16.14.0/27      attached            FastEthernet0/0
172.16.14.0/32      receive
172.16.14.1/32      172.16.14.1         FastEthernet0/0
172.16.14.2/32      receive
172.16.14.31/32     receive
172.16.14.32/27     attached            FastEthernet0/1
172.16.14.32/32     receive
172.16.14.33/32     receive
172.16.14.34/32     172.16.14.34        FastEthernet0/1
172.16.14.36/32     172.16.14.36        FastEthernet0/1
172.16.14.37/32     172.16.14.37        FastEthernet0/1
172.16.14.38/32     172.16.14.38        FastEthernet0/1
```

이 결과는 각 인터페이스에 활성화된 IP 주소 또는 IP 블록을 알려주고, 해당 인터페이스의 Next Hop 주소를 명기하고 있다.

cef 또는 dcef가 정상적으로 설정되었으면 각 인터페이스별로 NetFlow를 설정하도록 한다.
먼저, NetFlow를 설정할 각 인터페이스의 설정 모드로 들어간 후 NetFlow 스위칭을 다음과 같이 enable 시킨다.

```
rnd_ro3600(config)#interface Fast1/0
rnd_ro3600(config-if)#ip route-cache flow
```

□ NetFlow **결과 확인**

이제 NetFlow **결과가 공격자 추적을 위해서 어떻게 해석될 수 있는지 알아보도록 하자**.

NetFlow**를 통한 네트워크 통계는** "show ip cache flow"**라는 명령을 통해서 확인할 수 있다**.

```
rnd_ro3600#show ip cache flow
IP packet size distribution (2755 16976 total packets):
    1-32   64   96  128  160  192  224  256  288  320  352  384  416  448  480
     .000 .982 .001 .003 .000 .000 .000 .000 .000 .000 .000 .000 .000 .000 .000

     512  544  576 1024 1536 2048 2560 3072 3584 4096 4608
     .000 .000 .000 .000 .010 .000 .000 .000 .000 .000 .000

IP Flow Switching Cache, 278544 bytes
  162 active, 3934 inactive, 15469135 added
  199136077 ager polls, 0 flow alloc failures
  Active flows timeout in 30 minutes
  Inactive flows timeout in 15 seconds
  last clearing of statistics 3w6d
Protocol          Total    Flows   Packets Bytes  Packets Active(Sec) Idle(Sec)
--------          Flows     /Sec     /Flow  /Pkt     /Sec     /Flow     /Flow
TCP-Telnet         8275      0.0        28    95      0.1      8.7      14.3
TCP-FTP             668      0.0         7    69      0.0      1.7       8.3
TCP-FTPD            218      0.0       297  1050      0.0      0.1       2.8
TCP-WWW          103650      0.0         2   397      0.0      0.1       4.9
TCP-SMTP            84      0.0         1    46      0.0      0.6       5.2
...
TCP-other      15020394      6.4        18    54    117.2      0.6      10.8
UDP-DNS           77345      0.0         1    66      0.0      0.1      15.3
...
UDP-other        255282      0.1         4   104      0.4      1.5      15.4
ICMP               2328      0.0         5   134      0.0      8.8      15.4
...
IP-other            329      0.0         1    20      0.0      0.4      15.7
Total:        15468973      6.6        17    55    117.9      0.6      10.9

SrcIf         SrcIPaddress    DstIf         DstIPaddress    Pr SrcP DstP  Pkts
Fa0/0         238.110.164.31  Fa1/0         172.16.14.94    06 AE27 0003    26
Fa0/0         78.0.215.80     Fa1/0         172.16.14.94    06 8D57 0004    26
Fa0/0         224.216.69.3    Fa1/0         172.16.14.94    06 81F6 0002    26
Fa0/0         58.47.188.84    Fa1/0         172.16.14.94    06 52B1 000A    25
Fa0/0         71.39.162.91    Fa1/0         172.16.14.94    06 21A5 0003    26
Fa0/0         157.239.174.83  Fa1/0         172.16.14.94    06 D763 0003    25
```

NetFlow를 통해서 패킷 분포, 프로토콜 분포, 현재의 flow 등 크게 3가지 정보를 얻을 수 있다.

먼저, 패킷 분포 부분에서는 각 패킷의 크기별로의 분포를 나타내고 있는데, syn flooding, Ping flooding 등 서비스거부공격이 발생될 경우 일반적인 분포와는 달리 특정 크기의 패킷 특히, 아주 작거나 큰 패킷의 분포가 비정상적으로 늘어나는 것을 확인할 수 있다. 앞의 예제에서는 64Byte 이하의 패킷들이 98.2%나 차지하고 있는 것을 볼 수 있다. SYN flooding 공격을 오랜 시간 동안 진행한 결과이다.

두 번째는 프로토콜 분포인데 여기서는 지금까지의 네트워크 서비스 종류별 분포를 보여주고 있다. 이 정보는 어떤 서비스가 얼마만큼의 대역폭을 사용하고 있는지 알 수 있어 향후 서비스별 대역폭 제한 등 QoS에서도 사용될 수 있을 것이다. 또한 공격 발생시 특정 서비스나 기타 알려지지 않은 서비스의 흐름이 급격하게 증가함을 알 수 있다. 예제에서 초당 flow 수가 6.6인데 그중 대부분인 6.4를 TCP-other가 차지하고 있다. 이는 잘 알려진 포트 이외의 TCP 포트로 트래픽이 집중되고 있음을 알 수 있다.

세 번째는 현재 맺어져 있는 네트워크 flow에 대한 정보이다.
이 정보는 근원지 인터페이스(SrcIf), 근원지 IP 주소(SrcIPaddress), 목적지 인터페이스(DstIf), 목적지 IP 주소(DstIPaddress), IP 프로토콜(TCP 6, UDP 17)(Pr), 근원지 포트(SrcP), 목적지 포트(DstP) 그리고 패킷 수로(Pkts) 이루어져 있다. IP 프로토콜, 근원지 포트, 목적지 포트는 16진수로 표현되어 있으므로 주의해야 한다. 예제에서 Fa0/0 인터페이스로부터 Spoofing된 것으로 의심되는 트래픽들이 유입되고 있는 것을 알 수 있다.

서비스거부공격이 진행중일 경우 Spoofing된 패킷과 관련된 flow를 살펴볼 필요가 있다. 만일 공격으로 의심되는 트래픽이 192.168.xxx.xxx 대역에서 발생된 것으로 판단될 경우 include 명령을 이용하여 이와 관련된 트래픽만 확인할 수 있다.(Cisco에서 include 명령은 유닉스의 grep 명령과 유사하다.)

```
rnd_ro3600#sh ip cache flow | include 192.168
```

그리고, 이 Spoofing된 트래픽들이 어느 네트워크 인터페이스 카드를 통해서 유입되고 있는지 SrcIf를 통해서 확인한다.

위의 명령은 대규모 스캔 및 공격을 동반하는 인터넷 웜의 트래픽 분석에도 사용될 수 있다. 가령, Netbios를 통해 전파되는 Opaserv, Funlove 등의 웜바이러스에 감염된 시스템들을 탐색할 때도 사용될 수 있다. Netbios에 감염될 경우 Netbios-ns(UDP 137)로의 스캔이 급증하므로 다음 명령을 사용할 수 있다.(137의 16진수값은 89이다.)

```
rnd_ro3600#sh ip cache flow | include 89
```

그리고, 현재 시점부터 다시 트래픽의 상황을 확인하고 싶을 경우 다음의 명령으로 기존의 NetFlow 스위칭 통계를 초기화할 수 있다.

```
rnd_ro3600#clear ip flow stats
```

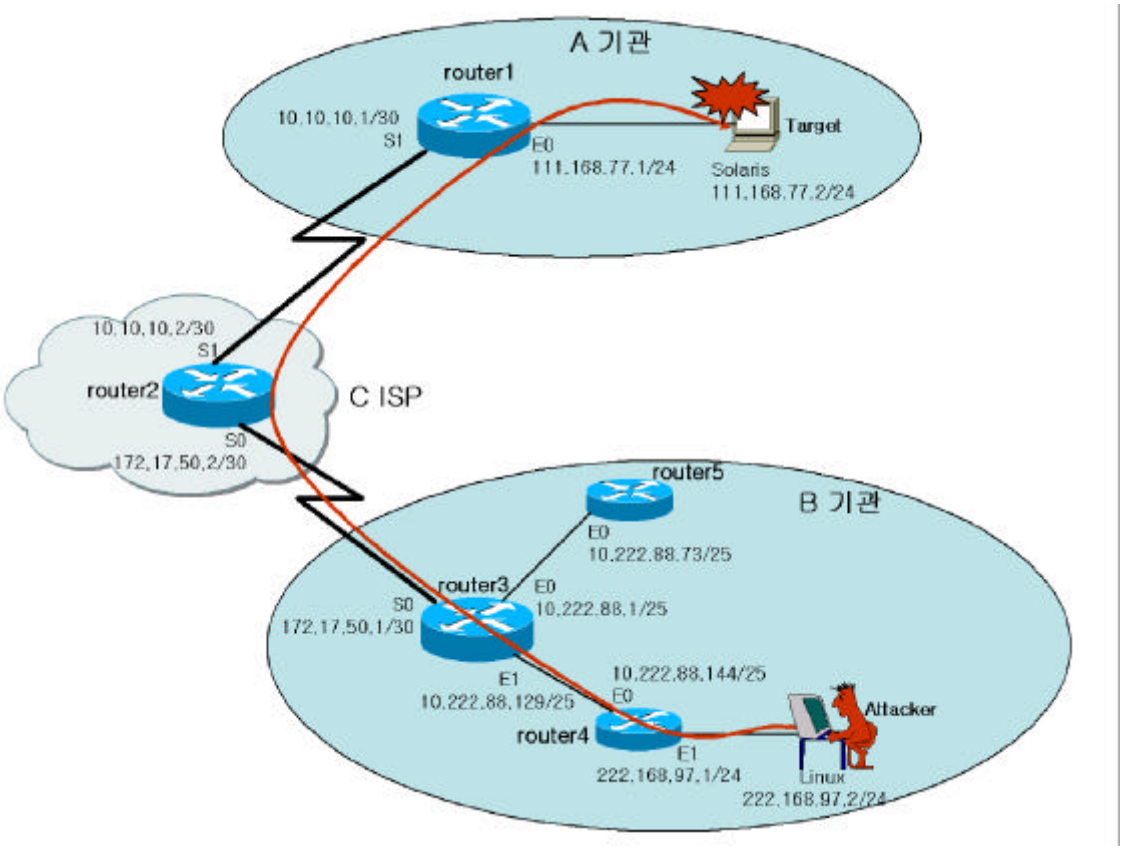
□ NetFlow를 이용한 공격경로 추적 시나리오

지금까지 살펴본 CEF와 NetFlow의 기능을 이용하여 IP Spoofing된 패킷의 실제 주소를 추적하는 시나리오를 생각해 보자.

시나리오에 따라 추적을 하기에 앞서 간단한 추적 개념을 말하고자 한다.
먼저 라우터에서 NetFlow를 이용하여 Spoofing된 패킷이 어느 네트워크 인터페이스에서 유입되고 있는지 살펴본다. 그리고 CEF 테이블을 이용하여 해당 네트워크 인터페이스에 연결되어 있는 활성화된 디바이스를 찾는다(Next Hop 추적). 각 디바이스에서 역시 NetFlow와 CEF를 이용하여 공격경로를 추적하는 과정을 반복하는 방식이다.

여기서 제약사항은 마지막 추적이 이루어질 때 까지 공격이 계속 진행되고 있어야 하며, 각 단계의 라우터가 Cisco에 한해서만 가능하다. 만일 Cisco가 아닌 다른 게이트웨이일 경우 NetFlow와 같은 명령이 지원되는지 알아보아야 할 것이다. 또한 분석을 위해서 각 단계의 라우터에 대한 접근 권한을 가지고 있어야 한다. 일반적으로 서비스거부공격은 여러기관 또는 ISP를 거쳐서 일어날 수 있으므로 기관간의 공동대응체계가 사전에 구축되어 있어야 신속한 추적이 가능하다. 특히, 네트워크 관리자는 자신의 기관이 연결된 ISP의 보안담당자 연락처를 확보하여 사고 발생시 신속하게 도움을 받을 수 있도록 하여야 한다. 그리고, 각 ISP에서도 고객사 지원과 더불어 ISP 간의 정보교류 채널도 확보되어야 할 것이다.

그러면, 다음의 공격 시나리오를 보자.



먼저 B 기관의 리눅스 서버(222.168.97.2)에서 A 기관의 웹서버(111.168.77.2, Solaris)를 대상으로 IP Spoofing된 서비스거부공격을 가한다고 하자. A 기관과 B 기관은 C ISP에 연결되어 있다.

A 기관의 웹서버 관리자는 MRTG를 통해 네트워크 트래픽을 모니터링하던 중 트래픽이 급증하고 있으며, 홈페이지 접속 속도 또한 상당히 느려졌음을 인지하였다. 웹서버 관리자는 공격받고 있는 Solaris 시스템에 접속하여 snoop 명령어를 통해서 IP Spoofing된 패킷들이 80번 포트로 수없이 들어오고 있는 것을 확인할 수 있다. 공격 패킷들은 96.170.xxx.xxx 대역에서 들어오고 있으며, 이들은 Spoofing된 것임을 알 수 있었다.

공격사실을 인지한 A 기관 관리자는 먼저 공격이 내부망에서 이루어지고 있는지 외부에서 이루어지고 있는지를 판단하기 위해 A 기관의 경계(border) 라우터에 로그인하여 트래픽을 살펴본다.

```
router1#sh ip cache flow | include 96.170
Se1             96.170.4.8      Et0             111.168.77.2    06 040C 0050    159
```

Spoofing된 패킷들의 근원지 인터페이스가 Serial1임을 확인할 수 있다. 즉, 경계 라우터 밖의 어디선가 공격이 가해지고 있음을 알 수 있다.
CEF 테이블에는 각 인터페이스별로 액티브한 모든 소스들을 가지고 있다. 따라서 Serial1에 연결된 next hop을 찾을 수 있다.

```
router1#sh ip cef se1
Prefix              Next Hop            Interface
0.0.0.0/0           10.10.10.2          Serial1
10.10.10.0/30       attached            Serial1
```

여기서는 next hop이 10.10.10.2(router2)만이 존재하는 것을 알 수 있다. 즉 router2를 통해서 공격이 들어오고 있는 것을 알 수 있다.
A 기관의 router1에서 NetFlow와 CEF를 이용하여 추적한 결과 공격은 C ISP의 router2를 통해서 이루어지고 있음이 명백해졌다.

A 기관 네트워크 담당자는 평소 알고 지내던 C ISP의 보안담당자에게 연락하여 router2에 대한 분석을 의뢰한다.
C ISP의 보안담당자는 router2에 로그인한 후 NetFlow와 CEF를 이용하여 어느 인터페이스를 통해 공격이 들어오고 있는지 추적해 보았다.

```
router2#sh ip cache flow | include 96.170
Se0             96.170.4.8      Se1             111.168.77.2    06 043C 0050    299

router2#sh ip cef se0
Prefix              Next Hop            Interface
172.17.50.0/30      attached            Serial0
222.168.97.0/24     172.17.50.1         Serial0
```

router2에서 NetFlow와 CEF를 통해서 Serial0 인터페이스에서 공격 트래픽이 유입되고 있으며, Serial0에 연결된 netxt hop은 172.17.50.1 즉, router3임을 알 수 있다. router3은 B 기관의 경계 라우터로써 즉시 B 기관의 네트워크 담당자에게 연락을 취해 B사로부터 공격이 발생되고 있음을 알린다.

B 기관의 네트워크 담당자는 router3에서 앞의 분석절차와 마찬가지로 분석하여 보았다.

```
router3#sh ip cache flow | include 96.170
Et1             96.170.4.8      Se0             111.168.77.2    06 053C 0050   3235

router3#sh ip cef et1
Prefix              Next Hop            Interface
10.222.88.128/25    attached            Ethernet1
10.222.88.144/32    10.222.88.144       Ethernet1
222.168.97.0/24     10.222.88.144       Ethernet1
10.222.88.73/32     10.222.88.73        Ethernet1
```

router3의 NetFlow 결과 Spoofing된 트래픽은 내부의 Ethernet1에서 발생되고 있는 것을 알 수 있다. 그런데, CEF 결과 두 개의 가능한 소스 즉 , 10.222.88.144(router4)와 10.222.88.73(router5)이 있는 것으로 나타났다. 따라서 이 두 라우터에서 각각 위와 같은 분석을 적용시켜 볼 필요가 있다. 먼저 router5에서 NetFlow를 통해 Spoofing된 트래픽이 보이는지 점검해 보자.

```
router5#sh ip cache flow | include 96.170
router5#
```

router5에서 NetFlow 결과 Spoofing된 트래픽이 나타나지 않고 있지만 실제 공격은 계속 진행되고 있었다. 따라서 나머지 하나의 주소 10.222.88.144(router4)에서 공격이 들어오고 있을 가능성이 높아졌다. router4에서 역시 NetFlow를 통해 확인해 보자.

```
router4#sh ip cache flow | include 96.170
Et1             96.170.4.8      Et0             111.168.77.2    06 053A 0050   6673
```

예상대로 router4의 Ethernet1 인터페이스로부터 Spoofing된 트래픽이 유입되고 있는 것을 확인할 수 있다. 그러면 이 트래픽들은 어떤 소스에서 유입되는지 Ethernet1의 CEF 테이블을 통해 확인해 보자.

```
router4#sh ip cef et1
Prefix              Next Hop            Interface
222.168.97.0/24     attached            Ethernet1
222.168.97.2/32     222.168.97.2        Ethernet1
```

CEF 테이블에서 현재 액티브한 IP 주소는 222.168.97.2 하나임을 알 수 있다. MAC 주소나 다른 방법으로 이 주소는 라우터가 아니라 일반 서버임을 알 수 있고, Sniffering 툴들을 이용해서 공격이 실제 발생되고 있는 서버가 222.168.97.2의 MAC 주소임을 확인할 수 있다.

이처럼 라우터에 하나의 시스템만이 연결되어 있을 경우는 대단히 다행스런 경우이다. 하지만 대부분 경우 해당 subnet 내에 다수의 PC와 서버들이 물려져 있을 것이다. 이 경우에는 지금까지와 마찬가지로 NetFlow와 CEF 만으로는 더 이상 추적이 힘들어 진다.

다. MAC 주소를 이용한 시스템 추적

앞서 시나리오의 예에서 만일 router4에 다수의 PC 또는 유닉스/ 리눅스 서버들이 액티브 상태일 경우 router4가 관리하는 subnet에서 실제 공격서버를 추적하는 작업이 별도로 필요할 것이다. 어떻게 이 많은 클라이언트와 서버 중에서 공격 트래픽을 발생시키는 것을 찾아 낼 것인가?

일반적으로 많은 관리자들은 워크그룹 스위치에서 포트를 하나씩 뽑아가며 공격 트래픽이 계속 발생되는지를 모니터링해서 그 포트를 찾아내는 방식을 사용하거나, 스위치 포트의 램프가 계속 깜빡이면서 뭔가 열심히 작업하는 포트를 우선적으로 뽑아서 확인하기도 한다.

이러한 방법으로도 찾을 수는 있겠지만 정상적으로 운영되어야 하는 다른 시스템들의 서비스도 잠시 중지시키지 않을 수 없게 된다.

그러면 포트를 뽑는 방법이외의 방법은 없을까?

□ 공격지 MAC 주소 추적

먼저 생각해 볼 수 있는 것은 대부분의 서비스거부공격이 IP만을 spoofing 할 뿐이지 자신의 MAC 주소까지 spoofing하지는 않는다. 따라서 공격 트래픽에서 MAC 주소를 알아내고 이 MAC주소를 가진 시스템을 찾을 수 있을 것이다.

공격지의 MAC 주소는 게이트웨이를 거치면서 변경되므로 NetFlow 등 앞에서 제시한 방법을 이용하여 Subnet까지 찾아와야만 한다.

그리고, 해당 Subnet에서 스위치에 미러링 포트를 연결하거나 아니면 게이트웨이에

서 더미 허브를 연결시켜서 Subnet을 모니터링 하도록 한다.

Tcpdump, snoop, 또는 윈도우즈 시스템의 네트워크 모니터링 도구를 이용하여 MAC 주소로 패킷을 모니터링하면서 기존에 보아온 공격 패킷들의 패턴을 가진 패킷의 근원지 MAC 주소를 찾아낸다.

하지만, tcpdump나 snoop의 경우 인터페이스가 썩 좋은 편은 못돼 공격지 MAC 주소(ethernet 주소)를 찾기가 불편하다. 따라서 공개 또는 상용 sniffeing 툴을 이용하면 보다 편리하게 모니터링 할 수 있을 것이다. 공개 또는 쉐어웨어 모니터링 툴은 다음 사이트에서 찾을 수 있다.

http:// packetstormsecurity.org/ sniffers/

다음은 편리한 GUI를 제공해 주는 sniffeing 툴 중의 하나인 spynet에서 캡쳐한 화면이다.

Time (h:m:s:ms)	MAC source addr	MAC dest. addr	Frame	Pro...	Addr. IP src	Addr. IP dest	Port src	Port dest
15:36:14:062	00 40 2B 1A 7C 88	00 50 56 41 44 43	IP	TCP	160.87.75.176	172.16.5.200	8785	7
15:36:14:062	00 40 2B 1A 7C 88	00 50 56 41 44 43	IP	TCP	156.86.87.248	172.16.5.200	58131	7
15:36:14:062	00 40 2B 1A 7C 88	00 50 56 41 44 43	IP	TCP	156.255.10.20	172.16.5.200	24032	7
15:36:14:062	00 40 2B 1A 7C 88	00 50 56 41 44 43	IP	TCP	167.5.171.222	172.16.5.200	21097	7
15:36:14:062	00 40 2B 1A 7C 88	00 50 56 41 44 43	IP	TCP	60.238.41.114	172.16.5.200	37967	7
15:36:14:062	00 40 2B 1A 7C 88	00 50 56 41 44 43	IP	TCP	249.77.22.116	172.16.5.200	33607	7
15:36:14:078	00 40 2B 1A 7C 88	00 50 56 41 44 43	IP	TCP	244.19.171.29	172.16.5.200	26840	7
15:36:14:078	00 40 2B 1A 7C 88	00 50 56 41 44 43	IP	TCP	168.77.6.110	172.16.5.200	14772	7
15:36:14:079	00 40 2B 1A 7C 88	00 50 56 41 44 43	IP	TCP	70.135.96.67	172.16.5.200	61666	7
15:36:14:078	00 40 2B 1A 7C 88	00 50 56 41 44 43	IP	TCP	238.0.239.186	172.16.5.200	28987	7
15:36:14:078	00 40 2B 1A 7C 88	00 50 56 41 44 43	IP	TCP	151.102.169.128	172.16.5.200	40399	7
15:36:14:078	00 40 2B 1A 7C 88	00 50 56 41 44 43	IP	TCP	117.117.44.119	172.16.5.200	48226	7

앞의 그림에서 공격지 IP 주소는 spoofing되어 여러 IP를 보이고 있으나, 실제 공격지 MAC 주소가 00:40:2B:1A:7C:88 하나로 고정되어 있는 것을 알 수 있다.

□ **해당 MAC 주소를 사용하는 시스템 추적**

이렇게 MAC 주소를 찾았으면 해당 MAC 주소가 어느 시스템에 물려 있는지 알아내야 할 것이다. 여기서 단순하게 RARP 프로토콜에서 이더넷 주소를 IP 주소로 변환하여 주면 간단히 해결될 문제이지만 이를 지원해 주지는 않는 듯 하다.(RARP는 일반적으로 디스크가 없는 시스템에서 부팅시 자신의 IP 주소를 동적으로 받기 위해 사용된다.)

따라서 MAC 주소를 이용한 추적은 스위치의 MAC 테이블을 활용하도록 하자.

Cisco Catalyst 2950 Switch 부터 기본 제공되는 l2trace 명령을 이용하여 해당 MAC

주소가 스위치의 어느 포트에 연결되어 있는지 찾아내는 방법이 있을 수 있다.

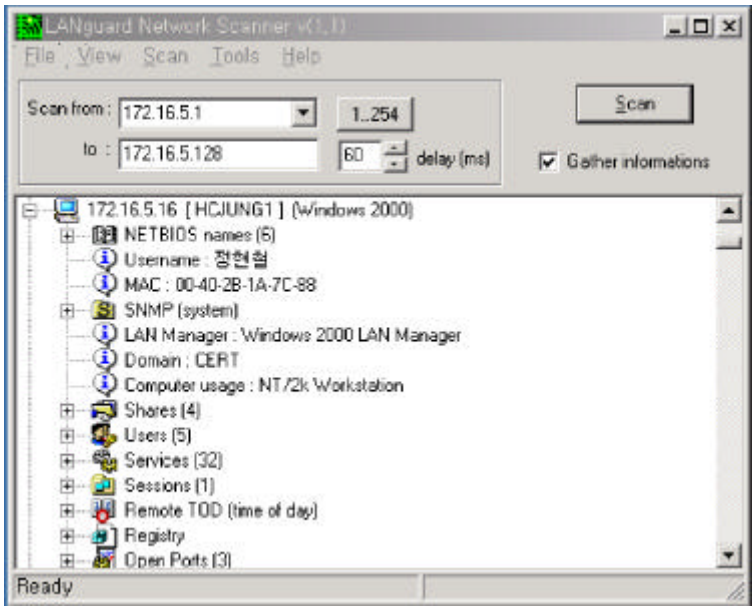
```
6509> (enable) l2trace 00-00-e8-34-00-01-e6-27 detail
Starting L2 Trace
l2trace vlan number is 222.
Attention: Source 00-00-e8-34-d2-96 is not directly attached to this system.
Source 00-00-e8-34- found in WS-C4006 : 100.248.2.254
WS-C4006 : cat4006 : 100.248.2.254: 4/27 10MB half duplex -> 2/1-2 1000MB full duplex
WS-C6509 : cat6509 : 100.248.117.78: 3/14,4/14 1000MB full duplex -> 8/44 10MB half duplex
Destination 00-01-e6-27- found in WS-C6509 named HSS_6509 on port 8/44 10MB half duplex
```

l2trace 명령이 지원되지 않을 경우는 MAC 테이블이나 ARP 테이블을 열람하도록 한다.

```
sw_r3#sh mac-address-table
Dynamic Address Count:                  4
Secure Address Count:                   0
Static Address (User-defined) Count:    0
System Self Address Count:              49
Total MAC addresses:                    53
Maximum MAC addresses:                  8192
Non-static Address Table:
Destination Address  Address Type  VLAN  Destination Port
-------------------  ------------  ----  --------------------
0002.4bb8.44d1       Dynamic          1  FastEthernet0/1
0003.ba02.220e       Dynamic          1  FastEthernet0/19
0800.1b41.2465       Dynamic          1  FastEthernet0/8
0800.1b41.318a       Dynamic          1  FastEthernet0/6

sw_r3#sh ip arp
Protocol  Address          Age (min)  Hardware Addr   Type   Interface
Internet  172.16.14.94            -   0003.e3c7.54c0  ARPA   VLAN1
Internet  172.16.14.65           30   0002.4bb8.44d1  ARPA   VLAN1
```

스위치를 이용하는 방법이외에 역시 공개 도구를 이용하여 내부 네트워크의 MAC 주소를 알아내 보도록 하자. MAC 주소를 알 수 있는 도구들은 주로 SNMP MIB 값을 이용하여 스캐닝을 하는 도구들인데 대표적인 도구가 Solarwinds나 LANGuard 등이 있을 수 있다. 다음은 LANGuard를 이용하여 내부 네트워크를 스캔하여 각 IP 들의 MAC 주소를 탐색하는 화면이다.



라. 시스템에서 공격도구 탐지 및 삭제

이제 공격 트래픽을 발생시키고 있는 시스템까지 찾았다. 하지만 대부분의 경우 공격시스템도 이미 해킹을 당한 피해 시스템인 경우가 많다. 이 경우 시스템 분석을 통해 공격자에 의해 생성된 각종 Artifact들(서비스거부공격 도구 포함) 을 찾아내어 제거하여야 한다. 하지만 시스템 내에는 수많은 파일들이 존재하고 많은 프로세스들도 실행되고 있어 어느 파일과 프로세스가 공격 트래픽을 발생시키고 있는지 찾아내는 것도 쉽지만은 않다. 다행히 시스템 분석 관련한 문서들은 쉽게 찾아 볼 수 있다.

Spoofing된 서비스거부공격은 유닉스 시스템과 윈도우즈 시스템 등 어디에서나 가능하다. 최근 개인용 PC들의 성능이 서버급 못지 않게 좋아지고 인터넷 접속이 일반화되면서 윈도우즈 시스템을 이용한 서비스거부공격이 늘고 있다.

시스템에서 서비스거부공격 도구를 포함한 공격툴이나 백도어 등 Artifact 분석을 위해서는 다음 문서를 참고하기 바란다.

Windows NT/ 2000 시스템 해킹 분석절차 :
http:/ / www.certcc.or.kr/ paper/ tr2002/ tr2002_11/ windows_server.pdf

UNIX 피해 시스템 분석 및 침입자 모니터링 Part I v1.0 :
http:/ / www.certcc.or.kr/ paper/ tr2001/ tr2001-03/ Scene-of-the-Crime.pdf

위의 문서들에 체계적인 분석절차가 기술되어 있는데, 서비스거부공격 도구는 일반적으로 많은 패킷을 발생시켜 시스템의 CPU 사용량을 많이 소모하므로, 이러한 프로세스와 관련 파일 및 프로그램을 추적하여야 할 것이다.

3. 결론

최근 각 기관이 많은 보안 장비를 도입하여 외부로부터의 침입에는 대응을 하고 있지만 막상 서비스거부공격에 대해서는 속수무책인 경우가 많다. 서비스거부공격에 대한 대응에 있어서 가장 큰 문제점은 대부분의 서비스거부공격이 IP 소스 주소를 속여서 공격을 수행하므로 공격지를 추적하기가 힘들다는 것이다.

본 문서에서는 지금까지의 이러한 어려움을 다소나마 극복하고자 라우터, NMS 등의 장비를 이용하여 추적할 수 있는 방안에 대해 기술하여 보았다. 하지만 앞서도 밝혔지만 이러한 추적에는 많은 제약사항이 따른다. 공격경로상의 모든 라우터가 Cisco 제품이고 접근권한을 가져야 한다. 또한 전화 발신 위치를 추적하는 것처럼 공격이 진행중일 때만이 분석이 가능하다.

서비스거부공격 패킷은 여러 ISP와 라우터를 거쳐서 도달하므로 각 라우터의 담당자와의 긴밀한 협조관계가 필수적이다. 각 ISP들에서는 이러한 문제가 발생되었을 경우 신속하게 분석하여 고객사의 피해를 최소화하고 공격자를 추적할 수 있는데 협력하여야 할 것이다. 이러한 취지에서 본 문서의 부록에 국내 주요 ISP별 보안담당자 연락처를 첨부하도록 한다. 분산서비스거부공격이 한창 이슈가 되었던 1999년 11월 미국 CERT/CC에서 전 세계 전문가들을 모아 놓고 개최한 분산서비스공격 대응 워크샵(Distributed Systems Intruder Tools Workshop)에서도 기술적인 대응 못지 않게 관리자, 시스템운영자, ISP, 침해사고대응팀(IRT) 간의 유기적인 협조체제의 중요성을 강조한 바 있다.

국내의 경우 서비스거부공격의 직접적인 공격 대상의 되는 경우보다는 내부 시스템이 해킹을 당한 후 서비스거부공격에 이용되는 경우가 더 흔하다. 이 경우도 공격 트래픽에 위해 내부 네트워크 대역폭과 라우터 CPU 등 내부 자원의 과다사용에 의한 장애가 발생되는 경우가 많다. 본 문서는 서비스거부공격을 당하고 있을 경우뿐만 아니라 내부 네트워크에서 공격에 이용되고 있는 시스템을 찾아내는데도 활용될 수 있으리라 생각된다.

참고문헌

Distributed Denial of Service Incident Handling : Real-Time Inter-Network Defense
http:/ / www.ietf.org/ internet-drafts/ draft-moriarty-ddos-rid-02.txt

Track the source of spoofed packets, by Rob Thomas
http:/ / www.cymru.com/ Documents/ tracking-spoofed.html

Null routing traffic and tracking DoS attacks, by Chris Morrow
http:/ / www.secsup.org/ Tracking/

Tackling Network DoS on Transit Networks
http:/ / www.dante.net/ pubs/ dip/ 42/ 42.html

Inferring Internet Denial-of-Service Activity
http:/ / www.caida.org/ outreach/ papers/ 2001/ BackScatter/ usenixsecurity01.pdf

Multi Router Traffic Grapher
http:/ / www.mrtg.co.kr/ mrtg/ mrtg_index.html
http:/ / people.ee.ethz.ch/ ~oetiker/ webtools/ mrtg/

Sniffering Tool
http:/ / packetstormsecurity.org/ sniffers/

[부록] ISP 보안담당자 연락처

영문서비스명	기관명	담당자	전 화 (주 · 야간)	전자우편
BORANET	데이콤	전승일	02-6220-7755 02-6220-0535	ipadm@bora.net
DREAMX	드림라인	정원철	02-2106-6172 02-2186-7000	ip@cjdream.com
ELIMNET	(주)엘림넷	노정현	02-3149-4941 02-3149-4999	abuse@elim.net
GNGIDC	지앤지네트웍스	Postmaster	02-2105-6075 02-2105-6130	abuse@gngidc.net
HANANET	(주)하나로통신	콜센타	02-106 02-106	info@hananet.net
ISSAN	(주)아이쎈	김혜진	02-789-9135 02-789-9114	issanadm@issan.net
KIDC	한국인터넷데이터센터	이민식	02-6440-2936 02-6440-2930	security@kidc.net
KOLNET	한국통신하이텔(주)	나진열	02-3289-2482 02-3289-4114	abuse@mail.hitel.net
KORNET	한국통신	김진원	02-3675-1499 02-3129-1411	abuse@kornet.net
KREONet	한국과학기술정보연구원	노민기	042-869-0554 042-869-0707	mknoh@hpcnet.ne.kr
KTNET	한국무역정보통신	오광진	02-6000-2170 02-6000-2091	domain@ktnet.co.kr
NETSGO	주식회사넷츠고	김용재	02-829-2953 02-829-2968	hl1mva@netsgo.com
NOWCOM	(주)나우콤	이정식	02-590-3951 02-590-3951	sulong@nownuri.net
PUBNET	한국통신-초고속국가망	오윤우	02-710-1416 02-710-1416	abuse@pubnet.ne.kr
SAEROUNNET	새로운넷주식회사	길우영	02-2102-3388 02-2102-3387	sanso@saeroun.co.kr
SHINBIRO	온세통신	김성인	031-738-6411 031-738-6413	ip@mgate.shinbiro.com
SKSpeedNet	에스케이텔레콤	남상훈	02-3709-0802 02-3709-0802	swnam@sktelecom.com
THRUNET	주식회사두루넷	갈영수	02-3488-8438 02-3488-8438	mailadmin@korea.com
KT-IDC	KT-IDC	고범석	031-788-0011 031-788-0011	abuse@kt-idc.com