# IP 네트워크 scanning 기법

2002. 3. 18

해킹바이러스상담지원센터/박현미, hmpark@certcc.or.kr
해킹바이러스상담지원센터/오은숙, esoh@certcc.or.kr
해킹바이러스상담지원센터/이동련, ryuni@certcc.or.kr

해킹을 시도하기 위한 사전 단계로 foot printing, scanning, enumeration 과정을 통하여 target 네트워크와 시스템에 대한 정보를 수집한다. 본 문서에서는 이러한 과정에서 수집하는 정보의 내용과 수집 방법들을 소개하고자 한다.

일반적으로 해킹을 시도하기 위해 target 시스템의 정보를 수집하는 다음과 같은 3단계를 수행한다.

① foot printing
② scanning
③ enumeration

Foot printing은 해커가 조직의 보안 상태 정보를 알아내기 위한 단계로 다음과 같이 인터넷에서 손쉽게 얻을 수 있는 정보를 수집하는 단계이다.

(1) 이름, 전자우편주소, 전화번호, 팩스 번호 등
(2) IP 블럭
(3) DNS 서버
(4) 메일 서버

Foot Printing 단계를 통하여 수집된 1차 정보를 가지고 Scanning 단계에서는 해당 시스템이 살아 있는지 탐지한다. ping sweeps, 포트 scan, 운영체제 확인 등이 있으며 이러한 단계에서는 수집하는 정보들은 다음과 같다.

(1) 각 시스템의 실행되고 있는 TCP/UDP 서비스
(2) 시스템 기종(Sparc, Alpha, x86)
(3) 접속 가능한 IP 주소
(4) 운영체제 타입

이러한 scan 과정은 예를 들어 도둑이 물건을 훔치기 위해 집의 창문과 문을 조사하는 것과 비교할 수 있다.

Enumeration 단계는 시스템의 resource name을 알아내기 위해 사용하는 것으로 시스템에 접속할 수 있는 정보들을 수집한다. foot printing과 scan 단계를 통한 정보를 바탕으로 보

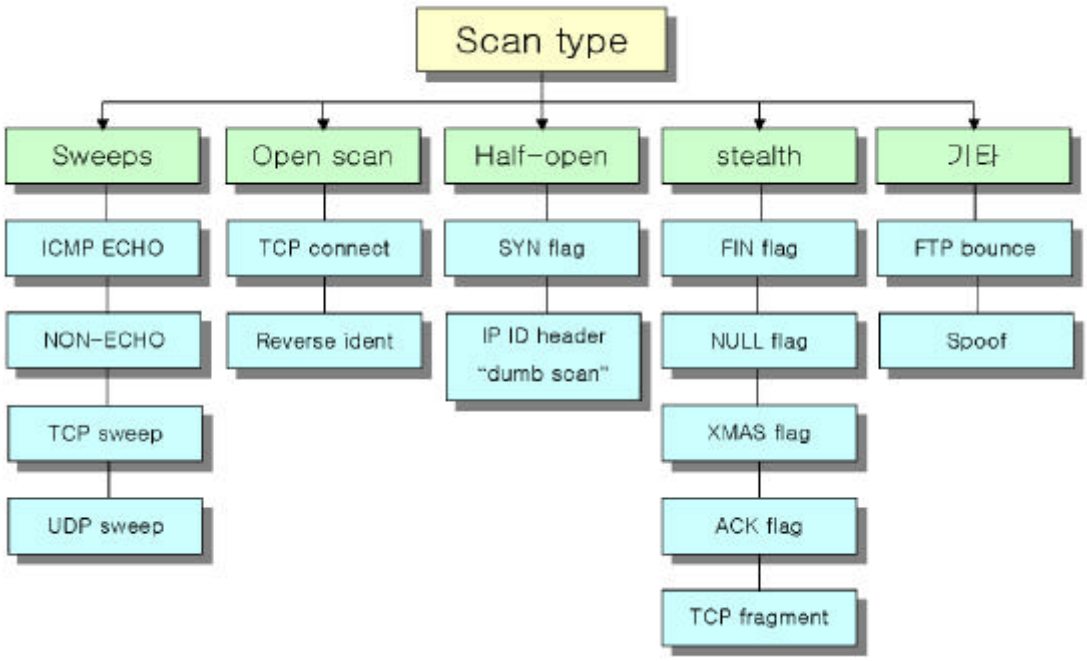
다 더 실용적인 정보를 얻는데 enumeration을 통하여 다음과 같은 정보를 수집한다.

(1) 사용자 및 그룹 이름
(2) 운영체제
(3) 라우팅 테이블
(4) SNMP 정보

Enumeration은 scan의 한가지로 볼 수도 있으며 일반적으로 scan 후 컴퓨터의 정보를 얻기 위해 사용한다.

## 1. Scanning 단계

Target 네트워크 내에서 실행되고 있는 살아 있는 TCP 네트워크 서비스를 식별하기 위한 효과적인 방법이 TCP 포트 scan이다. 해커에 의해 이용되는 TCP 포트 scanning 기술은 다음과 같이 몇 개의 그룹으로 분류할 수 있다.




이러한 각각의 scanning 기법은 target 네트워크를 원격으로 조사하기 위해, 열려 있거나 닫혀진 또는 필터링 되는 포트들을 식별하기 위해 사용되나 target 호스트와 네트워크에서 적용하는 네트워크 topology나 보안 메커니즘 등과 같은 환경에 영향을 받는다

네트워크를 scan하는 경우 쉽게 로깅되고 탐지될 수 있으나 열린 포트, 닫혀있거나 필터링 되는 포트에 대한 정확한 결과를 얻을 수 있다. 이에 stealth scan 방법은 IDS나 firewall과 같은 보안 도구들을 우회하기 위해 패킷의 플래그를 교묘하게 설정하는 등의 방법을 이용하

는데 이는 target 네트워크 환경에 따라 성공하기도 하며 경우에 따라서는 실패한다.

## 1) PING Sweeps

### 1.1 ICMP ping (ICMP ECHO requests)

Target 기관에서 사용하거나 소유하고 있는 IP 주소와 네트웍 범위를 알기 위해 ICMP ping-sweep를 사용하여 접근 가능한 살아 있는 호스트들을 찾는 방법으로 많은 시간이 소요되는 scanning 방법이다. ping은 목표 시스템이 활성화중이라는 것을 나타내는 ICMP ECHO_REPLY (Type 0)의 응답을 얻기 위해 목표시스템으로 ICMP ECHO(Type 8) 패킷을 전송하는데 사용되는 전통적인 툴로 ICMP ECHO 의 응답을 받은 경우 목표시스템이 살아 있다는 것을 의미한다. 이러한 ping은 기업용 네트워크와 같은 소규모 네트워크에서는 사용할 수 있으나 Class A와 같은 커다란 범위에서는 많은 시간이 소요되기 때문에 비효율적이다.

이러한 scanning을 위해서 사용되는 도구로는 정확한 결과를 준다고 알려졌으며 응답을 받은 후에야 다음 호스트로 이동하는 전통적인 ping과 달리 라운드 로빈 방식으로 여러 개의 ping을 요청하여 훨씬 빠른 시간 내에 많은 IP를 얻을 수 있는 fping, 윈도우 계열에서 사용 가능한 ping sweep 으로 대표적인 Pinger와 같은 것이 있다. 또한 scanning에 사용하는 강력한 도구인 Nmap의 '-sP'옵션을 이용하여 ping sweep을 할 수 있는데 nmap을 사용한 예는 다음과 같으며 자세한 정보는 http://www.certcc.or.kr/tools/Nmap.html를 참조한다.

```
[penguin:root]:/> nmap -sP 172.16.14.1-40

Starting nmap V. 2.3BETA15 by fyodor@insecure.org ( www.insecure.org/nmap/ )
Host  (172.16.14.1) appears to be up.
Host  (172.16.14.2) appears to be up.
Host  (172.16.14.31) appears to be up.
Host    (172.16.14.31) seems to be a subnet broadcast address (returned 1 extra pings).
Still scanning it.
Host  (172.16.14.32) appears to be up.
Host    (172.16.14.32) seems to be a subnet broadcast address (returned 1 extra pings).
Still scanning it.
Host  (172.16.14.33) appears to be up.
Host  (172.16.14.35) appears to be up.
Host mars.kcve.or.kr (172.16.14.40) appears to be up.
Nmap run completed -- 40 IP addresses (7 hosts up) scanned in 16 seconds
[penguin:root]:/>
```

위에서 보는 결과와 같이 nmap의 ping sweep scan을 이용하여 살아 있는 호스트를 식별하고 subnet broadcast 주소를 얻을 수 있는데 이는 해커에게 아주 유용하게 이용될 수 있는 정보이다.

이러한 ICMP ping-sweep에 대한 대응 방법으로는 ECHO, ECHO_REPLY 등외에 여러 가지 유형의 ICMP 트래픽을 주의 깊게 관찰해야한다. 대부분의 사이트에서는 직접적으로 인터넷에 연결된 시스템에게 ICMP 트래픽을 요청하지 않으며 파이어월에서 몇몇 ICMP 패킷을 필터링한다.

### 1.2 Non-ECHO ICMP

시스템에 대한 다양한 정보를 모으기 위해 TIMESTAMP(Type 13), ADDRESS MASK REQUEST(Type 17)과 같은 non-ECHO ICMP 프로토콜을 사용할 수 있다.
ICMP timestamp은 시스템의 현재 시간에 대해 처리하고 ICMP address mask는 subnet mask를 얻기 위해 사용되므로 특정 디바이스의 netmask를 요청하는데 사용할 수 있다. 많은 파이어월에서 ICMP ECHO 트래픽만 차단하므로 non-ECHO ICMP 방법을 이용할 수 있으나 이 트래픽 또한 차단될 수 있다.

### 1.3 TCP Sweeps

TCP 연결 확립 절차를 "3-way handshake"라하며 다음과 같은 과정으로 이루어진다.

① 클라이언트에서 원하는 포트로 초기 sequence 번호를 설정하여 SYN를 보낸다.
② 클라이언트의 SYN를 받은 서버의 해당 포트 서비스가 활성화 중인 경우 서버의 초기 sequence 번호와 함께 클라이언트의 SYN에 대한 응답으로 SYN+1 값의 ACK를 보낸다.
서버의 해당 포트가 닫혀있는 경우에는 연결을 재 설정하라는 RESET을 보낸다.
③ 서버로부터 SYN/ACK를 받은 클라이언트에서는 서버의 SYN에 대한 응답으로 다시 ACK를 보냄으로써 연결이 성립된다.

TCP Sweep 방법은 target 네트워크로 ICMP ECHO 요청 패킷 대신에 TCP ACK나 TCP SYN 패킷을 보낸다. 포트번호는 필요에 따라 선택하는데 대부분의 파이어월 등과 같은 차단 시스템에서 허가될 수 있는 21, 22, 23, 25, 80번 포트가 적당하다. 이에 대한 응답이 있는 경우 target 시스템이 살아 있다는 것을 말하며 이는 target 시스템의 운영체제나 차단 시스템 설정 등과 같은 환경에 영향을 받는다.

### 1.4 UDP Sweep

UDP(User Datagram Protocol)은 connectionless한 프로토콜로 TCP가 너무 복잡하고 느리거나 필요하지 않을 경우에 유용하게 사용될 수 있다. 대부분의 네트워크에서 열려진

UDP 포트를 찾는 것은 어려운데 현재 가장 많이 이용되는 방법으로 UDP Scan이라 하기도 한다. UDP 포트의 ICMP PORT UNREACHABLE 메시지를 이용하는데 UDP 데이터그램을 target 시스템의 UDP 포트로 보낸 후 ICMP PORT UNREACHABLE 메시지를 받지 못한 경우에 해당 포트가 열려진 것으로 판단한다. 그러나 이 UDP Sweep 방법은 다음과 같은 몇 가지 이유에서 완전하게 신뢰되지 못한다.

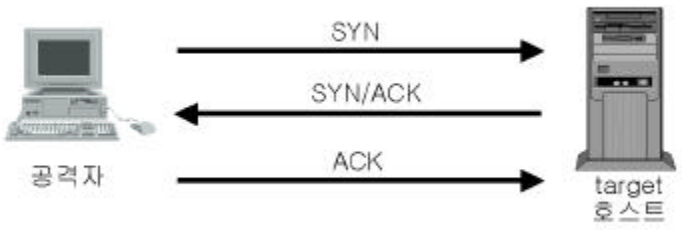
- 라우터에서 DUP 패킷이 drop 될 수 있다.
- 많은 올바른 UDP 서비스에서 응답을 하지 않는다.
- 대부분의 파이어월에서 UDP 패킷을 drop 하도록 설정되어 있다.
- ICMP PORT UNREACHABLE 메시지로 살아 있지 않은 UPD 포트의 응답에 의존하는 방법이다.

## 2) Open Scan

### 2.1 Standard TCP connect() scanning

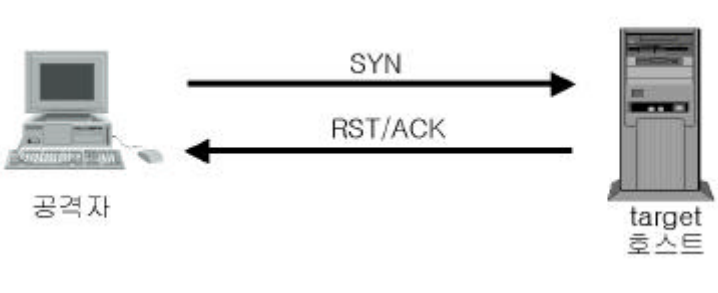
가장 간단한 유형의 포트 scan 방법으로 stealth 요소는 포함되지 않으며 TCP/IP 3-way handshake 연결을 확립한다.

Target 호스트의 해당 포트가 열려 있는 경우는 다음과 같은 패킷이 전송된다.




① target 호스트로 연결을 요청하는 SYN 패킷을 보낸다.
② target 호스트의 포트가 열려 있는 경우 target 호스트에서 연결 요청을 받아들이는 SYN/ACK 패킷을 보낸다.
③ ACK 패킷을 보냄으로 3-way handshake를 확립한다.

Target 호스트의 해당 포트가 닫혀 있는 경우에는 다음과 같이 패킷이 전송된다.



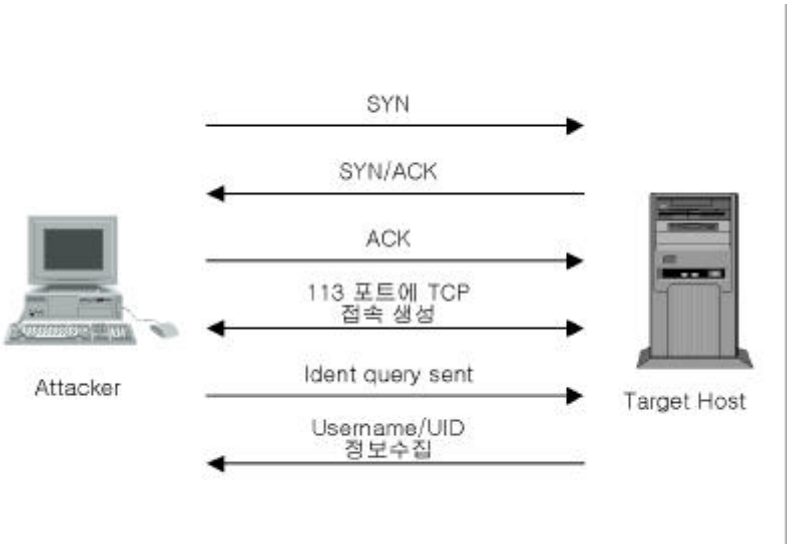

① target 호스트로 연결을 요청하는 SYN 패킷을 보낸다.
② target 호스트의 포트가 닫혀 있는 경우 연결 요청을 받아들이지 못하므로 target 호스트로부터 RST/ACK 패킷을 받는다.

이러한 방법에 의해 접근 가능한 TCP 네트워크 서비스를 식별하는 것은 아주 신뢰할 수 있는 정확한 결과를 얻을 수 있고 Unix-based 시스템 하에서 super-user의 권한 없이도 가능하나 쉽게 로깅되며 탐지될 수 있는 가장 원시적인 scanning 유형이다.

### 2.2 Revers Ident scanning

Revers ident 포트 쿼리는 접근 가능한 TCP 포트에서 실행되는 프로세스의 소유자에 대한 정보를 얻을 수 있다. 공격자가 이 scanning을 수행하는 목적은 root로 실행되는 데몬을 찾기 위함인데 공격자는 scanning 결과를 보고 취약한 오버플로우가 있는지 찾을 수 있고 또 해당 포트에 대한 악의적인 행동을 할 수 있다.

포트가 open되어 있을 때 reverse ident scan은 해당 포트에서 실행되는 프로세스 소유자의 username나 UID를 얻을 수 있는데 그 동작과정은 다음과 같다.



### 3) Half-open Scan

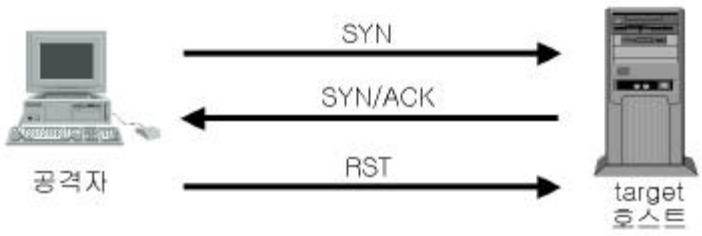
#### 3.1 SYN TCP flag scanning

이 방법은 TCP/IP 연결을 완전히 성립하지 않아 로깅 기법을 교묘하게 피할 수 있는 stealth scanning이다. 보통 3-way handshake는 두 호스트 사이의 연결을 초기화하기 위해 SYN 플래그를 설정하여 패킷을 보내고 포트가 열려 있는 경우 연결을 받아들이는 SYN/ACK 패킷으로 응답하면 이에 대한 ACK를 마무리로 연결을 완성하게 된다.

half-open SYN 포트 scanning 방법에서는 target 호스트로부터 SYN/ACK 패킷을 받은 경우 해당포트가 열려 있는 것만 확인하고 연결을 확립하기 위한 마지막 ACK 패킷 대신에 RST 패킷을 보냄으로써 완전한 연결을 성립하지 않아 로그가 남지 않게 한다.

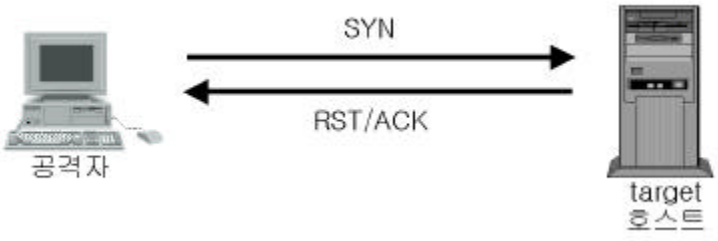
그러나 대부분의 IDS와 다른 보안 시스템에서 이와 같은 stealth half-open scan의 시도는 쉽게 탐지하므로 이를 우회하기 위해서는 FIN이나 TTL-based scanning 방법과 같은 다른 기술들을 이용하는 것이 좋다.

Target 호스트의 해당 포트가 열려 있는 경우는 다음과 같은 패킷이 전송된다.




① target 호스트로 연결을 요청하는 SYN 패킷을 보낸다.
② target 호스트의 포트가 열려 있는 경우 target 호스트에서 연결 요청을 받아들이는 SYN/ACK 패킷을 보낸다.
③ RST 패킷을 보냄으로 연결을 reset 한다

Target 호스트의 해당 포트가 닫혀 있는 경우에는 다음과 같이 패킷이 전송된다.

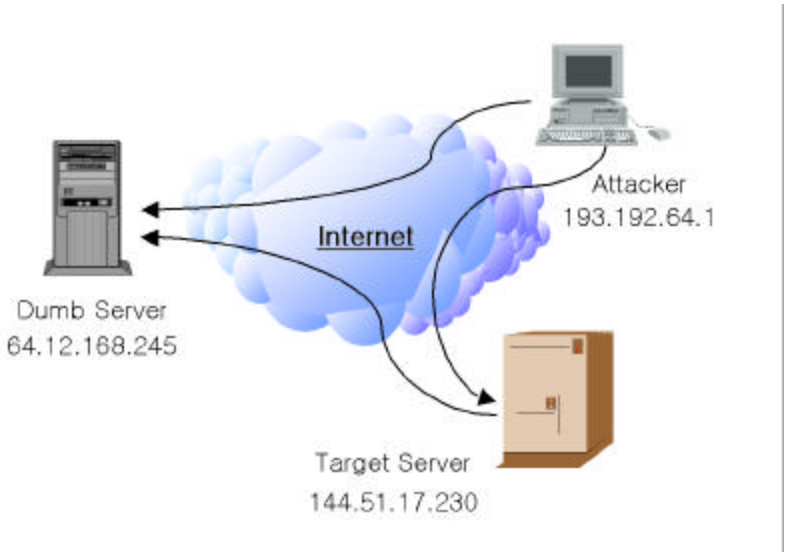



① target 호스트로 연결을 요청하는 SYN 패킷을 보낸다.
② target 호스트의 포트가 닫혀 있는 경우 연결 요청을 받아들이지 못하므로 target 호스트로부터 RST/ACK 패킷을 받는다.

이러한 방법은 TCP 네트워크의 살아 있는 서비스를 식별하는데 빠르고 안전한 방법이며 로깅되는 것을 어느 정도 피할 수 있으나 super-user 권한을 획득하여 네트워크 소켓의 raw access가 요구된다.

### 3.2 IP ID header TCP scanning

Antirez에 의해 처음으로 발견된 IP ID header TCP scanning 기법은 dumb 호스트라는 세 번째 호스트 필요한 것을 제외하면 SYN scan 방법을 반영한 기법이다. 이 scanning 방법에서 필요한 세 호스트는 아래의 그림에서 보는 바와 같이 attack 호스트(A)와 dumb 호스트(B), target 호스트(C)이다.




여기에서 "dumb"이라고 불리는 호스트의 역할이 중요한데, 뜻 그대로 어떤 트래픽을 보내지도 받지도 않는 silent하고 dumb한 호스트를 말한다. 왜 dumb 호스트가 필요한지는 dumb 호스트가 scanning 동안에 역할을 살펴보면 알 수 있다.

먼저 공격 시나리오를 살펴보면 다음과 같다.

① 호스트 A는 ID 필드를 분석하기 위하여 호스트 B에 연속적으로 ping을 보낸다. dumb 호스트는 ping에 대한 응답을 ID를 증가시키면서 보낸다.

```
60 bytes from BBB.BBB.BBB.BBB: seq=1 ttl=64 id=+1 win=0 time=96 ms
60 bytes from BBB.BBB.BBB.BBB: seq=2 ttl=64 id=+1 win=0 time=88 ms
60 bytes from BBB.BBB.BBB.BBB: seq=3 ttl=64 id=+1 win=0 time=92 ms
```

② 호스트 A는 dumb 호스트의 소스 주소를 사용하여 호스트 C에 spoofed SYN 패킷을 보낸다. 여기에서 사용되는 포트는 공격자가 open되었는지 close되었는지 테스트하기 위해 임의의 포트(1-65535)를 사용한다. 그러면 호스트 C는 SYN 패킷에 대해 dumb 호스트에 응답을 하는데(spoof된 dumb 호스트의 소스 주소를 사용했기 때문에) 호스트 C에서 open되거나 close된 포트에 대하여 다음과 같이 다른 결과를 얻게 된다.

- SYN/ACK 응답은 해당 포트가 open되어 있는 LISTENING 포트라는 것을 나타낸다. 그러면 dumb 호스트 RST 비트 플래그된 패킷으로 응답한다.(dumb 호스트는 실제로 이 SYN/ACK를 알지 못하므로)
- RST/ACK 응답은 NON-LISTENING 포트라는 것을 나타낸다. dumb 호스트는 이 패킷에 대하여 어떠한 응답도 보내지 않는다.

호스트 C의 open 된 포트에 대한 정보를 얻기 위해서는 호스트 C에서 dumb 호스트로 간 플래그 된 패킷을 알아야 하는데 이 패킷들은 이미 dumb 호스트로 보내져 호스트 A가 알 수 없다. 그러나 호스트 C에서 spoofed SYN/ACK 패킷이 dumb 호스트로 보내지는 동안에, dumb 호스트에서 호스트 A로 보내는 연속된 ping 패킷을 보고 가정할 수 있다. 즉 원래 ID 필드의 증가값은 1인데 SYN/ACK 비트 패킷을 보낸 호스트 C에게 dumb 호스트는 응답을 해야 하므로 ID 필드가 증가하는 것이다.

```
60 bytes from BBB.BBB.BBB.BBB: seq=25 ttl=64 id=+1 win=0 time=92 ms
60 bytes from BBB.BBB.BBB.BBB: seq=26 ttl=64 id=+3 win=0 time=80 ms
60 bytes from BBB.BBB.BBB.BBB: seq=27 ttl=64 id=+2 win=0 time=83 ms
```

close된 포트는 dumb 호스트에게 어떤 것을 보내라는 요청을 하지 않기 때문에 ping의 ID 필드가 증가하지 않는다.

```
60 bytes from BBB.BBB.BBB.BBB: seq=30 ttl=64 id=+1 win=0 time=90 ms
60 bytes from BBB.BBB.BBB.BBB: seq=31 ttl=64 id=+1 win=0 time=88 ms
60 bytes from BBB.BBB.BBB.BBB: seq=32 ttl=64 id=+1 win=0 time=87 ms
```

**여기에서 dumb 호스트가 필요한 이유를 다시 말하면 최대한 false-positive를 줄이기 위해서 가능한 지나다니는 트래픽이 없도록 해야하기 때문이다.**

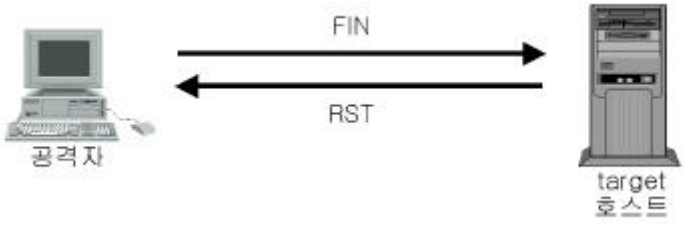
### 4) Stealth Scan

#### 4.1 Inverse TCP flag scanning

**firewall이나 IDS와 같은 보안 시스템들과 필터링들은 중요한 포트에 대한 SYN 패킷을 가지고 검사를 한다. 이에 보안시스템을 통과하기 위해서 probe 패킷은 다른 플래그를 설정하는데 3가지 방법이 있다. 이 세 가지 방법은 모두 BSD 계열에서만 적용된다. 포트가 열려 있으면 패킷이 drop이 되어 응답이 없고 닫혀 있으면 RST 응답이 전송된다.**

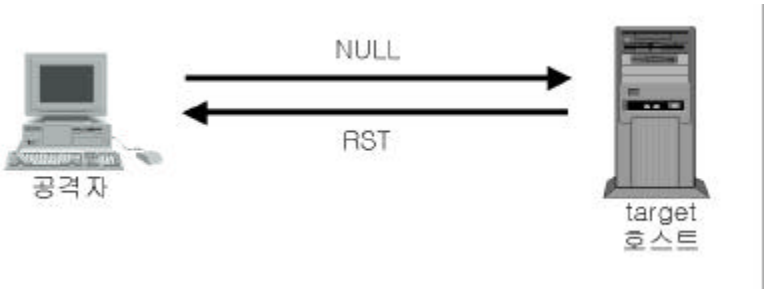
**가. FIN flag probe scanning**

**포트를 닫을 때 사용하는 FIN 플래그를 설정하여 probe 패킷을 전송한다.**

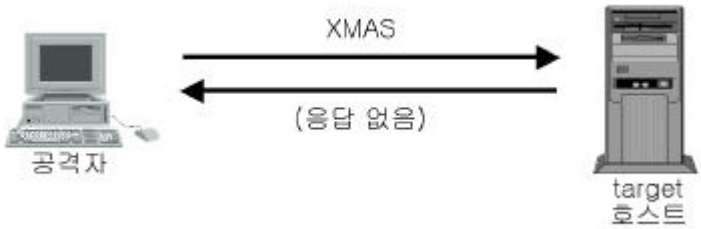

**나. NULL flag probe scanning**

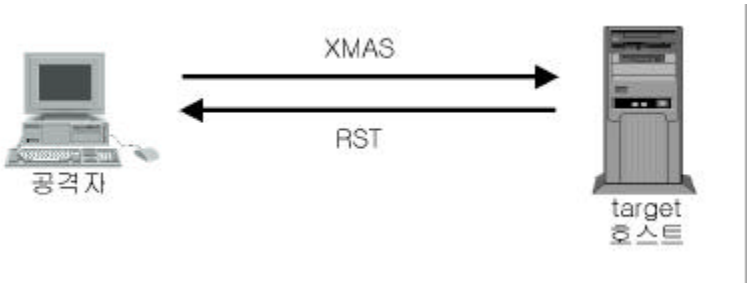
**어떠한 플래그(ACK, FIN, RST, SYN, URG, PSH)도 설정하지 않고 NULL 상태로 패킷을 전송한다.**


**다. XMAS flag probe scanning**

NULL 플래그 scan**과 반대로 모든 플래그**(ACK, FIN, RST, SYN, URG, PSH)**를 설정하여 패킷을 전송한다.**




**이러한** Inverse TCP Flag scanning **방법은 보안 시스템들을 우회하기 위해 사용되므로** IDS**와 같은 보안 장비들과 로깅을 피할 수 있는 장점이 있으나 네트워크 소켓의 하위부분을 접근해야 하며** super-user**의 권한이 필요하고** BSD **계열의** TCP/IP**에서만 효과가 있다.**

(마이크로소프트의 윈도우에서는 효과가 없다.)

### 4.2 ACK flag probe scanning

ACK 플래그를 설정하여 probe TCP 패킷을 만들어서 응답으로 오는 RST 패킷의 TTL과 WINDOW를 분석한다. 이 공격은 BSD 계열의 TCP/IP의 스택 취약점을 이용한다. ACK scan의 기술은 두 가지가 있다.

#### 가. Analysis of the TTL field of received packets

다른 TCP 포트에 ACK 패킷을 수천개 보내고 그 응답 패킷의 TTL 플래그를 분석한다. TTL의 값이 64이하이면 열려 있는 포트이다.


```
1: host 192.168.0.12 port 20: F:RST -> ttl: 70 win: 0
2: host 192.168.0.12 port 21: F:RST -> ttl: 70 win: 0
3: host 192.168.0.12 port 22: F:RST -> ttl: 40 win: 0
4: host 192.168.0.12 port 23: F:RST -> ttl: 70 win: 0
```

#### 나. Analysis of the WINDOW field of received packets

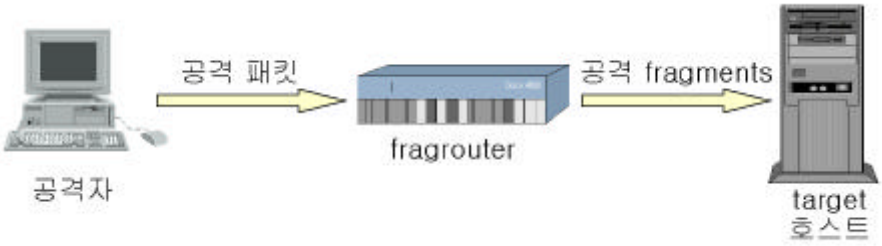
다른 TCP 포트에 ACK 패킷을 수천개 보내고 그 응답 패킷의 WINDOW 플래그를 분석한다. win의 값이 0이 아니면 열려 있는 포트이다.

```
1: host 192.168.0.12 port 20: F:RST -> ttl: 64 win: 0
2: host 192.168.0.12 port 21: F:RST -> ttl: 64 win: 0
3: host 192.168.0.12 port 22: F:RST -> ttl: 64 win: 512
4: host 192.168.0.12 port 23: F:RST -> ttl: 64 win: 0
```

이러한 방법은 log가 매우 다양하기 때문에 IDS에서 탐지되지 않을 수 있으나 분석하는데 시간이 소비되며 TCP/IP의 버그를 기대해야 하고 최근의 플랫폼에서는 적용되지 않는다.

### 4.3 TCP fragmentation scanning

하나의 패킷을 수 천 조각으로 나눠서 보낸다. 보안시스템에 탐지되지는 않으나 재조합하고 패킷에 접근하는데 시간이 걸린다. IDS를 테스트하기 위해 fragrouter을 만들었다. 이것은 ip 패킷을 받기 위해 별도의 시스템이 필요하다. 공격자는 공격 패킷을 fragrouter로 전송하면 fragrouter는 공격자가 정의한 크기로 패킷을 fragment해서 target호스트에 전송한다.




이 TCP fragmentation scanning 방법은 최근의 IDS와 네트워크 필터링을 피할 수 있다. 그러나 fragrouter를 설치해야 하고 패킷의 조각을 분실할 가능성이 있어 완전히 신뢰할 수 없으며 target 시스템에서 재 조합하는 동안에 네트워크와 커널에서의 로드 문제가 있다.

다음 기술문서 http://www.certcc.or.kr/paper/tr2001/tr2001-03/IP%20Fragmentation.html 에서 fragrouter에 대한 자세한 사항을 참조한다.

## 5) 기타

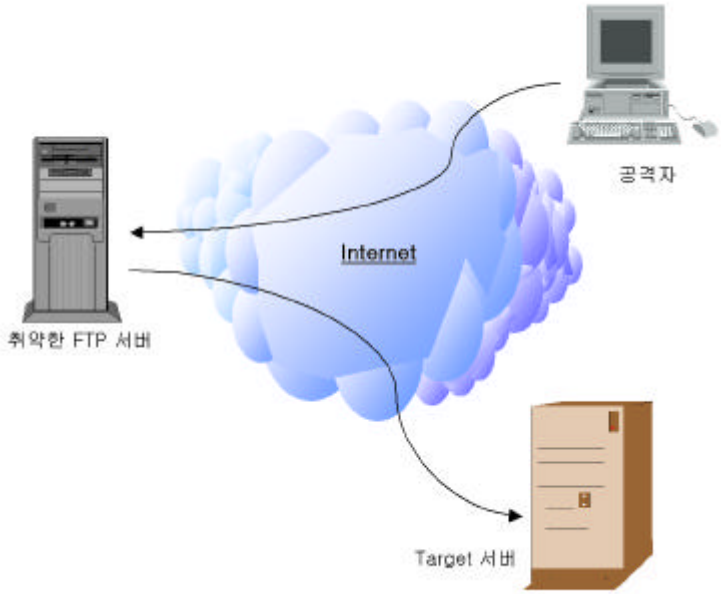
### 5.1 FTP bounce scanning

FTP bounce scan은 FTP 소프트웨어의 취약점으로 예전의 시스템에서만 가능하다.

FreeBSD 2.1.7 이하
HP-UX 10.10 이하

SunOS 5.5.1 이하
SunOS 4.1.4 이하
SCO OpenServer 5.0.4 이하
SCO UnixWare 2.1 이하
IBM AIX 4.3 이하
Caldera Linux 1.2 이하
Redhat 4.2 이하
Slackware 3.3 이하
WU-FTP 2.4.2-BETA-16 이하 버전을 실행하는 리눅스

쓰기 권한이 있는 디렉토리가 존재하면 더 효과적으로 공격할 수 있다. command 서비스나 파일에 다른 데이터를 추가할 수 있으면 이 데이터는 PORT command를 이용하여 타켓 호스트의 특별한 포트에 보낼 수 있다. 포트가 열려 있으면 150, 226 응답이 오고 닫혀 있으면 425 응답이 온다.




scanning 과정은 다음과 같다.

① 공격자는 FTP bounce 공격이 가능한 FTP 서버에 접속한다. passive 모드로 변환한 후 특정 호스트의 특정 포트로 데이터를 보낸다.

② PORT 명령으로 target 서버의 특정 포트로 접근한다.

| 200 PORT command successful |
|---|

③ LIST 명령을 보낸다.

- target 서버의 포트가 열려 있는 경우

```
LIST
150 Opening ASCII mode data connection for file list
226 Transfer complete.
```

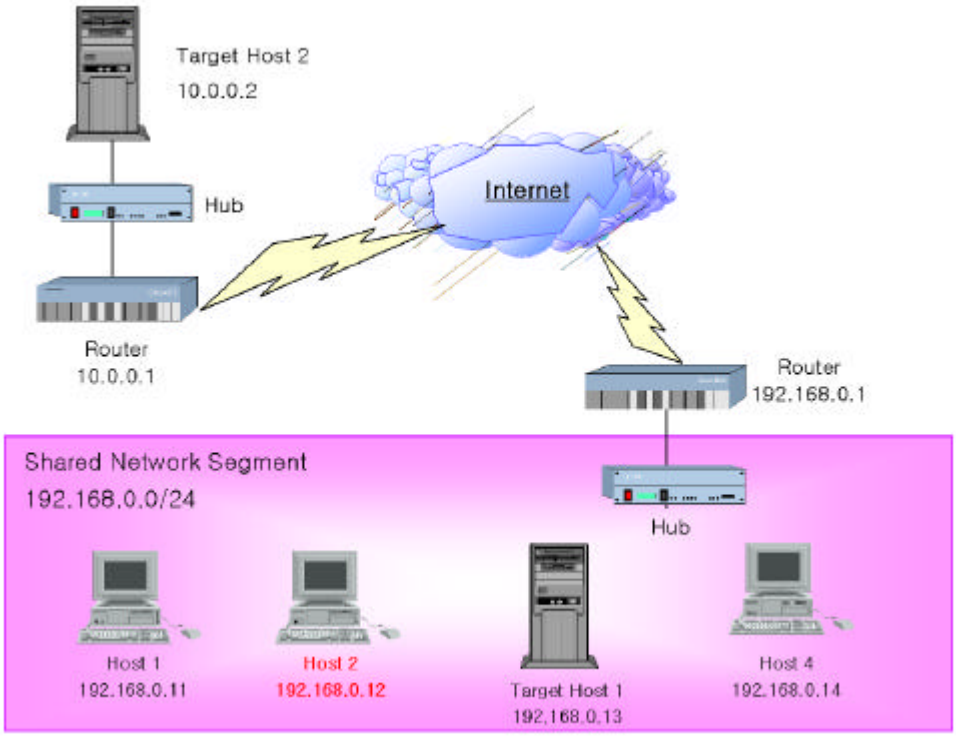
- target 서버의 포트가 닫혀 있는 경우

```
LIST
425 Can't build data connection : Connection refused.
```

### 5.2 Sniffer-based spoofed TCP scanning

이 scan 방법은 공격자의 IP를 노출시키지 않고 scan하기 위해 사용되는 방법으로 공격 IP를 spoof하여 scan을 시도하므로 피해자는 공격자를 확인할 수 없다. 즉, 공격자는 같은 네트워크 세그먼트 내에 있는 다른 호스트의 IP 주소에서 SYN 패킷을 spoof 한다. spoof한 패킷을 target 호스트에 보내고 target 호스트로부터 오는 포트 정보(어떤 포트가 액티브 상태인지)를 sniff한다. 이러한 방법을 이용한 툴은 spoofscan.c 등이 있는데 spoofed SYN scanning을 수행하기 위해서 먼저 다음의 사항 중의 하나를 만족해야 한다.

- 소스 IP 주소를 spoof할 호스트와 공유된 같은 네트워크 세그먼트에 존재한다.
- target이 되는 호스트와 공유된 같은 네트워크 세그먼트에 존재한다.
- target 호스트를 직접적으로든 간접적으로든 연결하는 router나 gateway 호스트와 같은 네트워크 세그먼트에 존재한다.




Spoofed TCP scanning 시나리오

이러한 scan을 동작하는 시나리오는 다음과 같은 3가지가 있다.

① 첫 번째 시나리오
- 공격 호스트(host 2)
- Spoof 할 호스트(host 1)
- target 호스트(target host 2) 인 경우,

공격 호스트(host 2)는 공유되어 있는 네트워크 세그먼트 내의 호스트로부터 spoofed SYN probe 패킷(spoofscan을 이용하여)을 target 호스트(target host 2)에 보낸다. target 호스트에서 온 응답을 sniff하여 target 호스트의 포트정보를 얻을 수 있다.

② 두 번째 시나리오
- 공격 호스트(host 2)
- Spoof 할 호스트(어떤 IP 주소 : 1.2.3.4, 1.3.3.7)
- target 호스트(target host 1)인 경우,

공격 호스트와 같은 네트워크 세그먼트에 있는 target 호스트(target host 1)를 포트 scan하고자 하는 경우에 소스 IP 주소를 1.2.3.4나 1.3.3.7과 같은 위조된 IP로하여 공격할 수 있다. 이 방법은 portsentry나 pro-active 보안 시스템이 올바르게 설치되거나 설정되지 않은 경우 DOS 공격에 이용될 수 있다.

③ 세 번째 시나리오

Spoofed port scan을 수행하는 더 이론적인 방법은 192.168.0.0과 10.0.0.0 네트워크 사이의 정적인 루트에 있는 한 호스트에 접근하는 것이다. 이 방법은 호스트가 있는 위치의 네트워크 상황에 따라 다양한 가능성이 있다.

## 2. Enumeration 단계

### 1) Banner Grabbing

어떤 서비스를 이용하면 해당 시스템의 OS를 알 수 있는데, 대부분의 시스템에서 지원하는 TELNET 서비스는 telnet 접속하였을 경우 welcome banner에서 OS를 알 수 있다. 이러한 이유로 요즘에는 배너를 삭제하거나 위조하기도 한다.

```
[hmpark@Boa_JJANG hmpark]$ telnet 172.16.5.37
Trying 172.16.5.37...
Connected to 172.16.5.37 (172.16.5.37).
Escape character is '^]'.
WOWLINUX Release 7.0 (AllLiEs)
```

### 2) DNS HINFO Record

DNS의 호스트 정보 레코드(HINFO Record)는 호스트의 H/W 타입과 OS를 나타내는 스트링 쌍으로 많은 DNS 쿼리를 통해서 보여지는데, 대부분의 관리자들이 이 레코드를 사용하지 않아 최근에는 많이 사용되지 않는 기술이다.

```
www    IN      HINFO   "Sparc Ultra 5"   "Solaris 2.6"
```

### 3) Operating System Guessing Techniques

많은 OS는 어떤 타입의 패킷을 보내고 받을 때 IP 레벨 표준의 설명을 가지고 있는데 이런 설명들을 분석하여 공격자는 target 호스트의 운영 플랫폼을 추측할 수 있다. 이러한 기술을 IP fingerprinting 이라고 하고 다음과 같은 기술이 있다.

- TCP FIN probe와 bogus flag probe
- TCP sequence number sampling

- TCP WINDOW와 ACK value sampling
- ICMP message quoting
- ICMP ECHO integrity
- Responses to IP fragmentation
- IP TOS(type of service) sampling

fingerprinting을 위한 Cheops와 Queso와 같은 툴이 개발되었고 sirc3는 BSD 계열의 TCP 스택, Windows TCP 스택과 Linux TCP 스택을 간단하게 탐지하는 툴이다.

## 3. 네트워크 Scanning 도구

| | |
|---|---|
| ① Nmap | Unix 기반 TCP/UDP 포트 scanning 툴로 bounce 공격, OS 추측과 네트워크 쿼리 등 주요 포트 scanning 타입을 탐지할 수 있다. |
| ② Spoofscan.c | jsbach에의해 개발된 TCP 기반 Spoofing/Sniffing 포트 스캐너로 Nmap 에서는 이 형식의 scanning을 수행하지 않는다. |
| ③ Fragrouter | IDS와 로깅을 우회하기 위한 툴로 다양한 모드로 패킷을 단편화 (fragment) 한다. |
| ④ IDES.c | IDS 우회 시스템으로 IDS 메커니즘을 혼동시키기 위해 데이터 스트림에 다양한 RST과 다른 패킷을 추가한다. |
| ⑤ Firewalk | ACL assessment 툴로 필터링되거나 필터링되지 않는 포트를 인식하기 위해 패킷의 TTL 기반 분석을 이용한다. |
| ⑥ HPING | 특별한 TCP 패킷 타입을 보내고 분석하는 툴 |
| ⑦ SING | 특별한 ICMP 패킷 타입을 보내고 분석하는 툴 |

## 4. 참고문헌


[1] Matta Security Limited, IP Network Scanning & Reconnaissance, 2002
[2] Publicaom, Network Scanning Techniques, 1999, 11
[3] http://www.synnergy.net/downloads/papers/portscan.txt
[4] Ron Gula, How to Handle and Identify Network Probes
[5] Fyodor, The Art of Port Scanning
[6] Joel Scambray, Hacking Exposed 2nd