2001년도 대한전기학회 하계학술대회 논문집 2001.7.18-20

Network Packet 의 Eavesdropping 에 대한 보안 대책에 관한 연구

최 영 . 이 승 철
중앙대학교 전자전기 공학부

A Study on the Security Measures against the Eavesdropping of the Network P

Young Choi , Seung-Chul Lee
School of EE, Chung-Ang University

Abstract - 각종 행정 업무나 금융 업무 등 많은 분야의 업무를 Network로 처리할 수 있도록 Internet 환경이 구축된 현실에서 개개인의 신상정보 또는 금융정보 등과 같은 누설되어서는 안 되는 다량의 정보들이 Internet을 통해 전송 되어지고 있다. 거미줄처럼 얽혀있는 Internet망을 통한 통신 중 어디 한곳에서라도 정보가 손실된다면 큰 혼란을 가져 올 것이다. 또한 누군가가 Internet망을 통해 전송되는 정보들을 들여다 볼 수 있다면 Encryption되지 않은 Data는 모두 누설되고 말 것이다. 이러한 위험요소들은 기술적으로 구현이 가능하며 실제적으로도 암암리에 행해지고 있는 것으로 알려져 있다. 현재 Internet Security를 위한 많은 Encryption Protocol이 존재하고 있지만 일관된 Encryption Protocol의 부재, 사용의 어려움, Cryptography Application의 부재로 인하여 Encryption을 사용할 수 없는 경우가 많이 있다. 본 논문에는 Eavesdropping의 원리를 이해하고 이에 대한 대책을 제시하였다.

1. 서 론

TCP/IP Protocol의 결함에 대해서는 이미 1985년에 Robert T. Morris의 논문 "A Weakness in the 4.2 BSD UNIX TCP/IP Software"에서 언급되었고 1995년 KevinMitnick이 이 이론을 실제화 하여 San Diego Supercomputer Center에 공격을 시도하다가 체포된 사건이 있었다. 이 사건 이후로 Kevin Mitnick이 사용한 기술은 IP spoofing라는 용어로 불리게 되었고 현재까지 TCP/IP 약점을 이용한 여러 가지의 기법이 지속적으로 나오고 있다. TCP/IP Protocol에서는 Datagram 자체가 Encryption되지 않고 전송되게 되어 있다. 따라서, Network를 이동하는 Packet을 모아서 순서에 맞게 재조합을 하면 원래의 Data를 얻을 수가 있다. 이러한 방법을 Sniffing이라고 한다. 이 방법으로 Remote Host에서 Monitoring하고 Packet을 재 조합해서 사용자의 ID와 Password를 얻을 수 있다. Data는 Network에서 여러 Host를 대상으로 Source Host에서 뿌려지게된다(Broadcasting). 그럼 각 Host들은 자신의 Data가 아니면 그냥 흘려보내고, 자신의 Data면 그것을 받아서 처리하게된다. 원래 System은 기본으로 자신의 Data만 받도록 설정되어 있지만 Sniffer(Sniffing Program)를 돌리게 되면 자신의 System Device의 Interface가 열리게 되고, Destination이 자신이 아닌 Packet도 받아들이게 된다. 이러한 모드를 Promiscuous Mode 라고 한다.

2. 본 론

2.1 Packet 의 구조

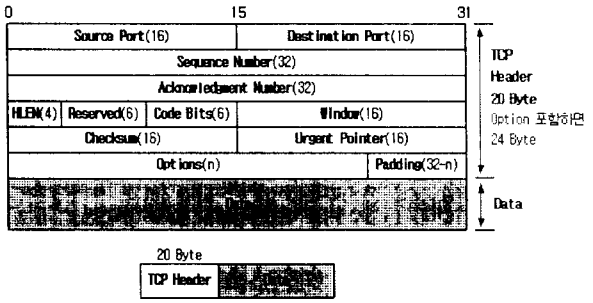
Network상의 Packet을 수집하기 위하여 Tcpdump를 사용하였다. Tcpdump는 주어진 조건식을 만족하는 Network Interface를 거치는 Packet들의 Header들을 출력해 주는 Program으로 실행하는 사람은 반드시 Network Interface에 대한 읽기 권한이 있어야만 한다.

```
1: Packet Capture Number
0000: 00 30 80 05 00 2B 00 10 5A 22 BD 6D 08 00 45 00
0010: 00 3D DB 36 00 00 80 11 03 BD A5 C2 0F 37 A5 C2
0020: 01 01 0A A0 00 35 00 29 64 A9 03 49 01 00 00 01
0030: 00 00 00 00 00 00 03 77 77 77 05 79 61 68 6F 6F
0040:  02 63 6F 02 6B 72 00 00 01 00 01

2: Packet Capture Number
0000: 00 10 5A 22 BD 6D 00 30 80 05 00 2B 08 00 45 00
0010: 00 AC 69 25 40 00 FD 11 B8 5E A5 C2 01 01 A5 C2
0020: 0F 37 00 35 0A A0 00 98 42 B7 03 49 81 80 00 01
0030: 00 02 00 02 00 02 03 77 77 77 05 79 61 68 6F 6F
0040: 02 63 6F 02 6B 72 00 00 01 00 01 C0 0C 00 05 00
0050: 01 00 00 01 11 00 10 02 72 63 05 79 61 68 6F 6F
0060: 02 63 6F 02 6B 72 00 C0 2D 00 01 00 01 00 00 03
0070: 8A 00 04 D3 20 77 97 C0 30 00 02 00 01 00 00 18
0080: 33 00 05 02 6E 73 C0 30 C0 30 00 02 00 01 00 00
0090: 18 33 00 06 03 6E 73 31 C0 30 C0 59 00 01 00 01
00A0: 00 00 07 BC 00 04 D3 20 77 47 C0 6A 00 01 00 01
00B0: 00 00 0A 47 00 04 D3 20 77 17

3: Packet Capture Number
0000: 00 30 80 05 00 2B 00 10 5A 22 BD 6D 08 00 45 00
0010: 00 30 DB 37 40 00 80 06 1F DF A5 C2 0F 37 D3 20
0020: 77 97 0A A1 00 50 4B F0 49 43 00 00 00 00 70 02
0030: 40 00 A3 49 00 00 02 04 05 B4 01 01 04 02

4: Packet Capture Number
0000: 00 10 5A 22 BD 6D 00 30 80 05 00 2B 08 00 45 00
0010: 00 2C 39 86 40 00 38 06 09 95 D3 20 77 97 A5 C2
0020: 0F 37 00 50 0A A1 3A 5D ED AC 4B F0 49 44 60 12
0030: 44 70 8B C5 00 00 02 04 05 B4 00 00

5: Packet Capture Number
0000: 00 30 80 05 00 2B 00 10 5A 22 BD 6D 08 00 45 00
0010: 00 28 DB 39 40 00 80 06 1F E5 A5 C2 0F 37 D3 20
0020: 77 97 0A A1 00 50 4B F0 49 44 3A 5D ED AD 50 10
0030: 44 70 A3 82 00 00 00 00 00 00 00 00
```

(표1) Packet Capture의 예

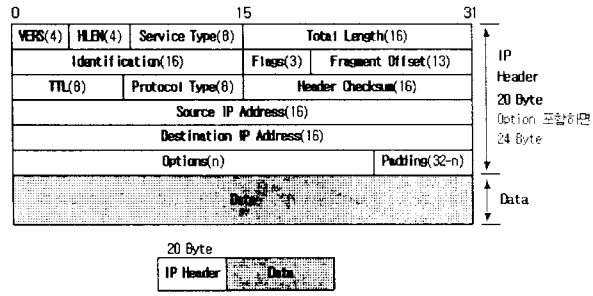
위의 (표1)은 일정시간동안 본 연구실내에서 의도적으로 Broadcasting된 Packet을 Capture한 Data중 일부분을 보인 것이다. 그 중에서 4번 Packet을 자세히 들여다보면 아래와 같은 구조를 하고 있는 것을 알 수 있다. Packet Data는 HEX Code 형태를 하고 있고 (그림 1,2)을 참조해서 보면 쉽게 구조를 파악할 수 있다.

MAC Source Address:	00-30-80-05-00-2B
MAC Dest Address:	00-10-5A-22-BD-6D
Frame:	IP
Protocol:	TCP->WWW-HTTP
Source IP Address:	211.32.119.151
Dest IP Address:	165.194.15.55
Source Port:	80
Destination Port:	2721
SEQ:	979234220
ACK:	1274038596
Packet Size:	60



HLEN	TCP Protocol Header의 길이(HLEN:Header Length)
Reserved	사용하지 않고 예약된 부분, 모두가 0 으로 되어있음
Code Bits	Segment 타입을 나타냄, URG:긴급, ACK:확인, PHS:강제전송, RST:리셋 SYN:동기, FIN:종료
Window	Segment 전송중의 송수신 Buffer Size를 지정하기 위해 사용
Checksm	TCP Packet의 내용이 바르게 교환되었는지를 점검
Urgent Pointer	Sequence Number에 의한 연속된 Data보다도 우선하는 긴급 Data의 교환시 Data의 어디까지가 긴급 Data인지를 나타냄
Options	최대의 Segment Size등을 지정함
Padding	TCP Protocol Header를 32bit 에 맞추기 위해 붙여짐

(그림1) TCP Header Format



VERS	Internet Protocol Version으로 현재 IP는 Ver. 4
Identification	상위층으로부터 각 IP Datagram을 분별하기 위한 식별번호
Flags	IP Datagram이 분할(Fragment)에 관한 정보를 나타냄
Format Offset	각 Fragment의 원래 Data에 있어서의 위치를 Byte 단위로 나타냄
TTL	Time To Live의 약자로 통과 가능한 Router의 남은 수를 나타냄. Router를 경유할 때마다 이 값이 하나씩 줄어든다.
Protocol Type	Data에 포함되는 상위 Protocol(TCP :6, UDP :17)의 종류를 나타냄

(그림2) IP Header Format

MAC(Medium Access Control) Address는 48bit (6byte)로 구성된 Hardware Address로 각각의 Device는 모두 다른 주소를 갖고 있다. 6byte중 앞의 3byte는 Hardware 제조사를 가리키는 고유 번호이고 뒤의 3byte는 제조사에서 정한 Serial Number이다. 위에서 보인 MAC Source Address에서 00-30-80은 CISCO Systems 사의 고유번호라는 것을 http://standards.ieee. org/에서 확인 할 수 있다. 또 MAC Dest Address에서 00-10-5A는 3 COM사의 제품이라는 것을 알 수 있다.

2.2 TCP 연결 체계의 취약점

2.2.1 IP Spoofing

1.	UDP->DNS	165.194.15.55	165.194.1.1	2720	53
	---	---	75		
2.	UDP->DNS	165.194.1.1	165.194.15.55	53	2720
	---	---	186		
3.	TCP->WWW-HTTP	165.194.15.55	211.32.119.151		2721
	80	1274038595	0	62	
4.	TCP->WWW-HTTP	211.32.119.151	165.194.15. 55		80
	2721	979234220	1274038596	60	
5.	TCP->WWW-HTTP	165.194.15.55	211.32.119.151		2721
	80	1274038596	979234221	60	
6.	TCP->WWW-HTTP	165.194.15.55	211.32.119.151		2721
	80	1274038596	979234221	380	
7.	TCP->WWW-HTTP	211.32.119.151	165.194.15.55		80
	2721	979234221	1274038922	517	
8	TCP->WWW-HTTP	211.32.119.151	165.194.15.55		80
	2721	979234684	1274038922	60	

(표2) Packet Filtering

(표2)는 Capture한 Packet 중에서 Target Host를 지정하여 Filtering 한 결과이다. 여기서 알 수 있듯이 3-Way Handshake를 한 후에 Data를 전송함을 볼 수 있다. 1과 2는 Domain Name Server에서 kr.yahoo.com을 UDP Connection을 사용하여 받아오고 3(Syn(a)), 4(Syn(b), Ack(Syn(a)+1)), 5(Syn(Syn(a)+1), Ack(Syn(b)+1))에서 TCP 3-Way Handshaking이 이뤄졌다.

FTP와 Telnet 연결에서도 마찬가지이므로 TCP의 취약점을 살펴보면 두 호스트간에 Syn와 Ack Number를 주고받으며 연결이 이뤄지기 때문에 임의의 Host(C)가 연결하고자 하는 Host(A)에서 보낼 Syn Number를 추측 할 수 있다면 두 Host간의 Connection을 가로채거나 임의의 Host(C) 자신의 IP Address를 Host(A)와 신뢰관계에 있는 Host(B) IP Address로 위장하여 연결하는 것이 가능하다는 것이다. Syn Number를 추측하는 방법은 임의의 Host(C)가 연결하고자 하는 Host(A)에 TCP 3-Way Handshake중 위 (표2)에서 3번, 4번만 행하고 5번에서는 Reset 신호를 보내는 방법을 여러 번 행하여 돌아오는 Sequence Number를 모니터링 해 보면 ISN(Initial Sequence Number)의 일정한 증감을 알 수 있다. 이렇게 얻어진 증감 Number와 Packet의 왕복시간(Round Trip Time)을 계산해 내면 ISN을 알 수 있다. 계산된 ISN과 위장된 IP(Host(C)가 Host(B)의 IP Address를 도용함) Address로 신뢰관계의 Host만이 접근 할 수 있는 Network에 연결이 가능하다.

TCP/IP Protocol은 Host의 Authentication을 Host의 Source IP Address만으로 판별하므로 Host(B)의 Port를 마비시킨 상태에서 Host(A)에 Syn을 보내면 Host(A)는 Host(B)에게 ISN을 보내게 되지만 Host(B)는 응답이 없으므로 그 틈을 타서 Host(C)는 계산된 ISN에 1을 더해서 Host(A)에게 보내게 되면 Host(A)는 위조된 IP Address를 사용해 접속한 Host(C)를 Host(B)로 믿고 인증과 함께 Connection이 이뤄진다.[1]

2.2.2 Connection Hijacking

이렇듯 초기의 Connection도 가능하지만 두 Host간에 이미 Connection이 이뤄진 상태에서도 접근이 가능하다. 두 Host간의 Connection이 이뤄진 상태에서 연결 가로채기(Connection Hijacking)는 초기의 Connection보다 간단하게 구현이 가능하다.

먼저, B라는 Host에서 A(Server)로 접속하여 사용하고 있을 때, C(Attacker)에서 Host B로부터 Reset을 보낸 것처럼 Packet을 보낸다. 그러면, A(Server)는 자동으로 Connection 을 끊게 되고, Host B는 A(server)와 연결이 아직도 유효하다고 생각하게 된다. 그런 다음

C(Attacker)는 Host B에서 Connection 을 새로 설정 하는 것처럼 Syn Packet을 A(Server)에게 보낸다. 이렇게 되면 A(Server)는 Host B와 Connection이 없는 상태에서 다시 연결을 시도하는 것으로 생각하게된다. 따라서 A(Server)는 자동으로 Syn + Ack 패킷을 Host B에게 보내게 된다. 그러나 Host B는 이전의 Connection이 유효한 상황으로 생각하고 있으므로 이 Syn + Ack Packet을 이해하지 못해 받아들이지 않는다. 이때 C(Attacker)는 이 Packet을 가로채서 Syn + Ack Packet에 같이 첨부된 Sequence Number에 따라 Ack를 보내준다. 첨부된 Sequence Number와 자신이 보냈던 Sequence Number를 참조하여 Packet을 Host B 에서 보낸 것처럼 A(Server) 에 보내게 되면 A(Server)는 Host B 와 Connection이 이루어진 것으로 생각하게 된다. 또한 Host B 도 여전히 Connection이 이루어진 상태로 생각하게 되는 것이다. 이제 A(Server)에서 Host B로 가는 Packet을 가로채어 Host B에 연계를 시켜주고 Host B에서 A(Server)로 가는 Packet 또한 C(Attacker)가 가로채어 A(Server)로 넘겨주게 된다. 이러한 과정에서 A(Server)에서 Host B로 어떤 Data를 요구하는 경우 C(Attacker)가 보내준 Packet을 받아서 Host B에 연결하여주면 Host B는 그것에 답하게된다. 그리고 이러한 방법은 Sequence Number에 의하여 이루어지기 때문에 A(Server)에서 보내는 Request에 대하여는 Host B는 받아보지 못하는 것이다. 즉 처음에 A(Server)와 Host B 사이의 통신에 사용하던 Sequence Number가 변하기 때문이다. 즉 A에서 보낸 Packet이 Host B로 가더라도 Host B가 그 Packet을 받을 수 없고, 그 반대도 마찬가지이다. 이러한 방법으로 C(Attacker)는 원하는 Packet을 A(Server)에게 보낼 수 있게 된다. 예로 들어 여기서 설명한 Host B와 A(Server)의 Connection이 Telnet이나 FTP Connection이었다고 한다면 이미 Host B가 ID나 Password를 입력한 이후에 Connection Hijacking이 이뤄지기 때문에 A(Server)에 접속할 ID나 Password를 몰라도 Host B가 실행할 수 있는 모든 권한을 그대로 갖게 된다.

2.3 Sniffer 탐지 방법

2.3.1 Ping Method

대부분의 Sniffer는 일반 TCP/IP Stack상에서 동작하기 때문에 Request를 받으면 그에 해당하는 Response를 전달하게된다. ping을 이용한 Sniffer 탐지 방법은 의심이 가는 시스템에게 Ping을 보내는데 MAC 주소를 위장하여 보내는 방법이다. MAC 주소를 위조하여(Local Network에 존재하지 않는 MAC 주소 사용)하여 Ping(ICMP Echo Request)을 다른 시스템에게 보낸다. 만약 Ping Reply(ICMP Echo Reply)를 받게되면, 해당 호스트가 Sniffing을 하고 있는 것이다. 왜냐하면 존재하지 않는 MAC 주소를 사용했기 때문에 Sniffing을 하지 않는 호스트는 누구도 Ping Request를 볼 수 없게되며 Reply를 하지 않게 된다.

2.3.2 DNS Method

일반적으로 Sniffing Program은 사용자의 편의를 위하여 Sniffing한 시스템의 IP 주소를 보여주지 않고 Domain Name을 보여주기 위하여 Inverse-DNS lookup을 수행하게 된다. 따라서 DNS Traffic을 감시하여 Sniffer를 탐지할 수도 있다. 이 방법은 Remote 또는 Local Network 모두에서 할 수 있는 방법이다. Remote에서 테스트 대상 Network로 Ping Sweep을 보내고, 들어오는 Inverse-DNS lookup을 감시하여 Sniffer를 탐지할 수 있다. Local Network에서 할 경우에는 위조된 IP 주소로 IP Datagram을 보내고 이에 대한 DNS Lookup이 있는지 감시하여 Sniffer를 탐지할 수 있다.

2.3.3 Host Method

Host단위에서 Promiscuous Mode를 확인하는 방법으로 "ifconfig -a" 명령을 이용하여 확인할 수 있다. [2]

3 Sniffing 방지 대책

가장 좋은 방법은 Data를 Encryption 하는 것이다. Data를 Encryption 하게되면 Sniffing을 당하더라도 내용을 알아 볼 수 없게 된다. 또한 Switching 환경의 Network를 구성하여(비록 Sniffing이 가능하기는 하지만) 되도록 Sniffing이 어렵도록 하여야 한다.

3.1 Web Countmeasure : SSL

Encryption Web Surfing을 가능하게 해주는 SSL (Secure Sockets Layer)은 많은 Web Server와 Browser에 구현되어 있다. 그리고 대부분의 전자상거래 사이트에 접속하여 신용카드 정보를 보낼 때 사용된다.

3.1 E-Mail Countmeasure : PGP & S/MIME

전자메일(E-mail) 또한 많은 방법으로 Sniffing되고 있다. Internet은 여러 곳에서 Monitoring될 수도 있으며, 잘못 전달될 수도 있다. 전자메일을 보호하기 위한 가장 안전한 방법은 메일을 Encryption하는 방법이며, 가장 대표적인 방법은 PGP와 S/MIME을 사용한다.

3.1 Telnet, FTP Countmeasure : SSH

SSH(Secure Shell)은 유닉스 시스템에 Encryption된 Login을 제공하는 Tool로서 사실상 표준으로 사용되고 있다. Telnet 대신에 반듯이 SSH을 사용하여야 한다.[3]

3.1 VPN

VPN(Virtual Private Networks)은 Internet에서 Encryption Traffic을 제공한다. [4]

3. 결 론

본 논문에서는 TCP/IP의 구조적인 결함과 Internet상에서 수집된 packet에 대한 분석과 더불어 Encryption되지 않은 data의 취약점을 알아보고 이를 이용한 IP Spoofing, Connection Hijacking의 원리를 이해하고 이에 대한 해결책으로서 Switching Network구성과 SSL, PGP & S/MIME, SSH, VPN과 같은 Encryption Application을 제시하였다. 현재 본 연구실에서는 Sniffing Auto Detector를 개발 중에 있다.

본 연구는 (주)네트 인텔리젠스(NIC)사의 지원에 의하여 수행되었음

[참 고 문 헌]

[1] Robert T. Morris "A Weakness in the 4.2 BSD UNIX TCP/IP Software" , 1985
[2] Brian Hatch, James Lee, George Kurtz "Hacking Linux Exposed: Linux Security Secrets & Solutions," Osborne:McGraw-Hill, 2001.
[3] StuartMcClure Joel Scambray, George Kurtz "Hacking Exposed: Network Security Secrets & Solutions," 2rd Edition, Osborne:McGraw-Hill, 2000.
[4] StuartMcClure Joel Scambray, George Kurtz "Hacking Exposed: Network Security Secrets & Solutions," Osborne:McGraw-Hill, 1999.