情 報 保 護 學 會 誌
第 14 券 第 2 號, 2004. 4

인터넷 트래픽의 정보은닉 기법 분석

손 태 식*, 박 종 성*, 문 종 섭*

요 약

은닉채널에 관한 연구는 1980년대 이전부터 진행되어 왔으며, 최근에는 멀티미디어 데이터에 대한 스테가노그래피에 대한 관심이 집중되고 있다. 하지만, 본 논문에서는 현재 스테가노그래피나 정보은닉에서 다루는 동영상 데이터에 대한 은닉채널이 아닌, 인터넷 환경의 근간을 이루는 TCP/IP 네트워크 트래픽에 존재하는 은닉채널에 대한 연구를 수행하였다. 먼저 은닉채널 개념 및 기존 연구동향을 분석하였으며, 그 후 TCP/IP를 구성하는 각 프로토콜에 생성 가능한 은닉채널을 분석하여 향후 연구 방향을 제시하였다.

Ⅰ. 서 론

네트워크 환경의 급속한 발달과 함께 네트워크를 통해 전달되는 정보 혹은 네트워크 환경에 포함된 정보의 양은 점점 더 증가하고 있다. 이러한 정보의 증가는 네트워크 자원에 대한 보안 문제로 연결되며 지금 이 순간에도 우리는 네트워크의 취약성을 이용한 여러 공격이나 컴퓨터 바이러스에 노출되어 있다. 하지만, 앞서 열거된 공격들에 대한 대응을 위한 해결책으로 방화벽, IDS, 안티-바이러스 솔루션 등의 대응방안이 제안되고 있지만, 이러한 솔루션은 물론이고 현재 네트워크 환경의 근간인 TCP/IP 프로토콜의 내재된 취약성을 이용한 공격에 대해서는 특별한 대응책을 가지고 있지 못하다. TCP/IP 프로토콜은 기본적으로 효율중심의 프로토콜로서 개발 당시에는 보안적인 측면에 대한 고려가 전무했다. 그러므로 TCP/IP를 구성하는 여러 프로토콜들 및 그 응용 서비스들을 이용하여 일반 네트워크 트래픽처럼 보이는 은닉 채널을 생성이 가능하다. 즉, 보안 솔루션들로 보호되는 네트워크일지라도 은닉채널을 통해 내부로부터의 정보 누출이 가능한 것이다. 여기서 은닉 채널이란 시스템의 보안 정책을 위반하는 어떤 방법을 사용하여 정보를 전송하는 프로세스에 의해 이용될 수 있는 임의의 통신 채널로서 정의 할 수 있다. 근본적으로 이와 같은 통신 채널은 일반적인 컴퓨터 설계상의 통신 수단이 아니며 보통 특정 정보에 접근하는 것이 허락되지 않는 프로세스나 사용자들에게 정보를 전송하기 위한 수단으로서 사용된다[1].

본 논문에서는 은닉채널이라 불리는 두 호스트간 기밀통신 채널의 개념과 관련 연구 동향을 파악한 후, TCP/IP 프로토콜에서 생성 가능하다고 알려진 여러 은닉 채널 형성 방법을 분석하였다. 본 논문의 구성은 다음과 같다 2장에서는 은닉채널에 관한 개요 및 관련 연구를 3장에서는 TCP/IP 프로토콜에 따른 은닉채널 생성 기법에 대해서 분석하였으며, 4장에서는 TCP/IP 트래픽에 은닉채널을 생성하는 기법에 따라 분류하였다. 마지막으로 5장과 6장에서는 향후 연구 방향과 결론을 제시하였다.

Ⅱ. 은닉 채널

1. 은닉 채널 소개

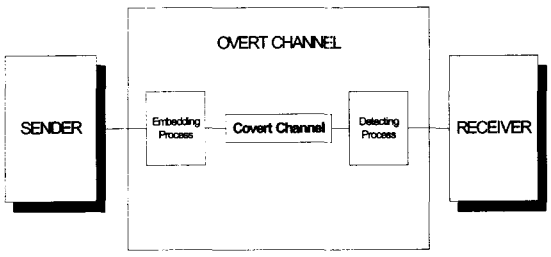
은닉채널 기법을 분석하기 전에 먼저 은닉채널의 개념과 종류 등에 대해서 알아보겠다. 은닉채널은 그림 1과 같이 공개채널(Overt Channel)과 구분되어 정의 될 수 있다[2]. 은닉채널에서는 데이터를 전송하기 위해 일반적인 통신채널 즉, 공개채널을 사용하지

† 본 연구는 정보통신부 대학 IT연구센터 육성·지원사업의 연구결과로 수행되었습니다.
* 고려대학교 정보보호대학원, 정보보호기술연구센터 ({743zh2k, p19j78s}@korea.ac.kr)
* 고려대학교 전자 및 정보공학부 교수 (jsmoon@korea.ac.kr)

만 이때 일반 통신채널 속에 포함된 특정 데이터에 대한 은닉화 기법은 은닉채널의 생성자 외에는 알 수 없다. 하지만, 공개채널에서는 은닉채널과 마찬가지로 일반적인 통신채널을 이용하지만, 데이터의 기밀성을 유지하기 위한 기법이 공개되거나 쉽게 구분이 될 수 있는 차이점을 가지고 있다.

은닉채널이란 용어는 Lampson의 [3]에서 처음으로 소개되었으며, 그 개념에 대한 정의는 1985년의 미국 DOD의 기술문서 [1]에 잘 나타나 있다. 그 정의는 "Any communication channel that can be exploited by a process to transfer information in a manner that violates the systems security policy"이며 유사한 의미로 steganography, information hiding, subliminal channel 등이 사용된다. 또 다른 정의로는 [3]에서 언급하는 것과 같이 "데이터를 전송하기 위해 의도되거나 설계되지 않은 통신 채널", [4]에서의 "리소스들의 상태를 나타내는 변수값들을 스토리지에 의해 전달하는 통신 채널" 그리고 [5]에서와 같이 "은닉채널은 한 주체에서 다른 주체로 정보를 전달하는 일반적인 데이터 개체들로 보이지 않는 엔터티들을 사용한다"등과 같은 것들이 있다.

은닉채널은 보통 은닉저장채널(Covert Storage Channel)과 은닉시간채널(Covert Timing Channel)로 구분한다. 은닉저장채널은 임의의 프로세스가 특정 오브젝트의 값을 변경하게 되면 다른 프로세스는 그러한 오브젝트 값 변경의 결과를 관찰하여 특정 정보를 알아낼 수 있는 임의의 통신 채널을 말한다. 또한 은닉시간채널은 임의의 프로세스가 CPU, I/O 등 시스템 자원에 대한 어떤 효과를 유발하게 되면 그 결과가 상대 프로세스에 의해 관찰되고 이렇게 관찰된 시간을 바탕으로 특정 정보를 알아낼 수 있는 채널이다[4][5]. 은닉채널 자체에 대한 보다 자세한 소개는 J. McHugh의 [6]을 참고하면 된다.



[그림 1] 공개 채널과 은닉채널과의 관계(Overt Channel and Covert Channel)

2. 은닉채널 연구 동향

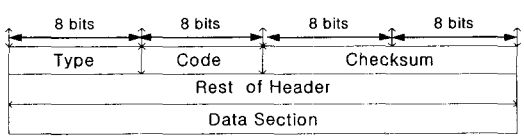
은닉채널에 관한 연구는 1970~80년대에 [1-6]과 같이 은닉채널의 개념 형성 및 정의에 관한 연구를 시작으로 최근에는 멀티미디어 데이터에 대한 정보은닉(Information Hiding)분야에서 활발하게 그 연구가 진행되고 있다[7][8]. 하지만, 본 논문에서 다루려는 네트워크 트래픽에 대한 은닉채널 형성 기법은 ICMP 프로토콜을 이용한 Daemon9의 [9]와 같은 연구, TCP 및 IP 헤더를 이용한 Rowland의 [10]과 같은 연구, Brinkhoff의 HTTP 프로토콜을 이용한 연구 [11], 타임스탬프 메시지를 이용한 John Giffin의 [12]와 같은 연구 그리고 Christopher Abad의 [13]와 같은 IP 체크섬을 이용한 연구 등이 이루어지고 있다. 또한 단지 이론 연구뿐만이 아니라 [9-22]와 같이 TCP/IP 프로토콜에서 은닉 채널을 형성할 수 있는 도구들이 개발되고 있다. 하지만 상대적으로 은닉채널 공격기법의 연구 외에 탐지에 관한 연구 [23]는 미미한 실정이다. 또한 최근에는 [24][25]와 같이 은닉 채널의 수학적 모델 분석 및 생성에 관한 연구도 이루어지고 있다.

III. 정보은닉 기법 분석 - 프로토콜별 분류

본 논문에서는 기존의 은닉채널 생성 기법들이 주 대상으로 삼았던 화상, 음성, 동영상 등의 멀티미디어 데이터에 대한 은닉채널 생성이 아닌 일반적인 TCP/IP 네트워크 환경의 네트워크 트래픽을 대상으로하여 생성 가능한 은닉채널 기법에 대하여 분석하였다. 이러한 은닉채널 형성 기법의 분석은 생성된 은닉채널의 기능적 측면에 대한 분석이 아닌 은닉채널을 형성하는 대상 TCP/IP 프로토콜 각각의 특성을 이용한 은닉채널 생성기법에 초점을 맞추었다.

1. ICMP 프로토콜

ICMP 프로토콜은 보통 다른 호스트나 게이트웨이와 연결된 네트워크에 문제가 있는지 확인하기 위하여 주로 사용된다. ICMP를 이용한 네트워크 진단 프로



[그림 2] ICMP 헤더 구조

그램으로는 ping 프로그램이 있다. 이 프로그램을 사용하여 특정한 게이트웨이, 호스트, 라우터 등이 제대로 작동을 하고 있는지 등을 조사하며, ICMP 요청에 대한 응답시간을 검사함으로써 네트워크 상태도 어느 정도 확인 할 수 있다. 그래서 일반적으로 대부분의 네트워크는 ICMP를 사용하는 ping 패킷에 대한 접근을 허용한다. 즉, 이러한 ICMP 패킷의 네트워크 접근 가능성을 이용하여 ICMP 패킷의 페이로드 부분에 특정 정보를 포함(은닉)하여 방화벽이나 IDS와 같은 보안 장비를 보호되고 있는 네트워크에 대해서 은닉채널을 형성 할 수 있다.

ICMP 은닉채널 패킷은 기본적으로 IP 페이로드를 이용해서 보내어지며, IP 헤더 포맷의 9번째 필드(프로토콜 타입)가 ICMP를 나타내는 1로 설정되어 있다. ICMP 페이로드에 특정 데이터를 삽입하여 위장이 가능한 이유는 ICMP 페이로드 부분이 가변적인 크기를 가지며 이때 ICMP 페이로드의 값은 ICMP 프로토콜을 생성하는 운영체제에 따라 다르게 설정 되기 때문이다. 또한 정상적인 ICMP 패킷의 경우 이러한 ICMP 페이로드에는 의미 없는 값들이 채워지거나 아니면 NULL 상태를 유지한다. 그러므로 이러한 ICMP 페이로드를 관찰한다 할지라도 그 의미를 파악하는 것은 쉽지 않다. 이러한 ICMP의 특성을 이용한 은닉채널 생성 도구로는 daemon9와 alhambra가 개발한 Loki[9], edi와 teso의 ICMP tunneling tool 그리고 CodeZero team에서 개발한 ICMP backdoor[20]등이 있다.

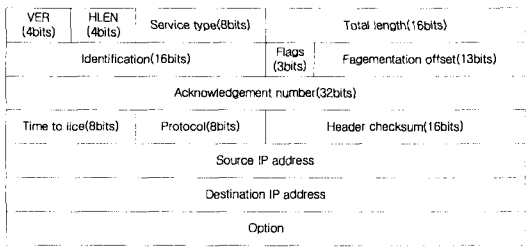
〔표 1〕 ICMP 페이로드의 특성

type / OS	ICMP 페이로드
Null 패킷	0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000
Windows 패킷	0900 6162 6364 6566 6768 696a 6b6c 6d6e 6f70 7172 7374 7576 7761 6263
Solaris 패킷	50ec f53d 048f 0700 0809 0a0b 0c0d 0e0f 1011 1213 1415 1617 1819
Linux 패킷	9077 063e 2dbd 0400 0809 0a0b 0c0d 0e0f 1011 1213 1415 1617 1819

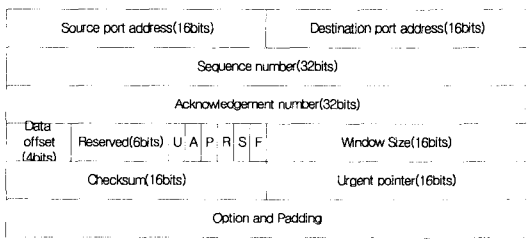
2. IP 프로토콜과 TCP 프로토콜

기존의 많은 연구를 통해서 TCP와 IP 프로토콜의 헤더에는 은밀한 방식으로 원격지에 있는 호스트에게 정보를 저장하거나 전송할 수 있는 많은 필드들을 가지고 있다는 것이 알려져 있다. 특히 그림 3, 4의 TCP, IP 헤더를 살펴보면, 각 헤더 내에는 Option 필드와 같이 프로토콜의 다양한 기능 설정을 위해서 사용될 뿐 통신과정에는 사용되지 않거나 또는 송신자의 필요로 인해서만 설정되는 필드들과 ID, SEQ 등 통신 과정에 필수적으로 사용되는 필드들로 크게 나누어 은닉채널을 형성 할 수 있음을 알 수 있다. 하지만 이렇게 데이터를 은닉하여 전송할 수 있는 공간들이 있을지라도, 전송과정에 있어서 보다 필수적인 필드에 데이터를 포함시키는 것이 은닉채널을 이용한 공격 성공률 및 탐지의 복잡성을 높일 수 있으며, 또한 이러한 전송과정에서 필수적으로 사용되는 필드들은 일반적인 TCP/IP 옵션 필드들이 패킷 필터링이나 쪼개고 재조립하는 과정에서 때때로 변화되거나 노출될 수 있다는 문제점을 가진 것에 비해 전송 도중의 변화 및 노출 가능성을 가지지 않는다는 장점을 지니고 있다.

본 장에서는 TCP, IP 헤더의 여러 필드 중에서도 IP 헤더의 ID 필드와 TCP 헤더의 SEQ 필드를 이용한 은닉채널을 생성 기법에 대해서 분석하였다. 위의 이러한 필드를 이용한 은닉채널 형성기법은 〔10〕에 나타나있으며, IP 헤더의 ID 필드의 경우 ID 필드를 임의의 숫자로 생성하여 특정 ASCII 값의 배가 되도록 인코딩하는 방법을 사용한다. 이러한 방법은 원격의 호스트에게 단순히 ID 필드를 읽는 것만으로 쉽게 특정 정보를 전달할 수 있게 해준다. 즉, ID 필



〔그림 3〕 IP 헤더 구조



〔그림 4〕 TCP 헤더 구조

드의 경우 'H' 값을 은닉하기 위해서는 'H'의 ASCII 값인 72를 공격 이전에 일반적인 패킷에 포함되어 있는 ID 값들의 범주와 비슷한 숫자를 생성할 수 있도록 임의의 숫자 256배를 하여 계산한 값인 18432 (72*256)를 ID 필드로 생성하는 것이다. 이후 이러한 패킷을 특정 포트로 수신하는 은닉채널 서버를 구성하고, 은닉채널 서버는 특정 포트로 수신된 패킷의 ID 필드 값을 256으로 나누어 원하는 데이터를 일반적인 TCP/IP 헤더의 ID 필드를 이용하여 얻을 수 있게 되는 것이다.

TCP 헤더의 SEQ 필드 역시 ID 필드의 은닉화와 같은 방법이나 SEQ 필드(32bytes)에 올 수 있는 값의 범위는 ID 필드(16bytes)의 값보다 훨씬 크기 때문에 'H'의 경우 72*256*65536의 값으로 ASCII 값을 생성한다. 이렇게 특정 값으로 ASCII 값을 인코딩하는 것은 앞서 설명한 바와 같이 일반적으로 TCP/IP를 이용한 통신과정에서 쓰이는 패킷들이 포함하는 ID 또는 SEQ 필드 값과 유사하게 만들기 위한 것이다.

하지만 이렇게 조작된 ID, SEQ 필드를 가지고 있는 TCP/IP 헤더는 정상적인 TCP/IP 패킷의 헤더와는 다음과 같은 차이점을 가진다. 먼저, 조작된 ID, SEQ 필드의 값이 정상적인 패킷의 ID, SEQ 필드 값이 가지는 값과는 비슷할지라도 많은 패킷들을 패턴화 시킬 경우 일반적인 패킷들과는 상이한 패턴을 가지게 된다. 또한 각 패킷은 TCP 연결시도와 같은 형태로 은닉 패킷을 전달하기 위하여 syn 플래그와 같이 특정 제어 플래그가 설정되어 있거나, IP fragment 필드 그리고 offset 설정 필드에 있어 일반적인 TCP/IP 패킷과는 다른 점을 가지게 된다. 비록 이러한 차이점이 있다고 할지라도 IDS의 시그너쳐 또는 패킷 트래픽의 관찰자의 직관만으로는 구분하는 것이 어려운 것이 현실이다.

Source port address(16bits)	Destination port address(16bits)
UDP Total Length(16bits)	CheckSum(16bits)

(그림 5) UDP 헤더 포맷

3. UDP + DNS 프로토콜

UDP 프로토콜은 통신과정에서의 효율을 중점을 둔 프로토콜로서 TCP 프로토콜에서 제공하는 것과 같은 다양한 전송관련 기능을 제공하지 않는다. 그림 x에 나타나듯이 UDP 헤더 포맷은 단지 4가지의 요소를 가지는 단순한 구조을 가지고 있으며, 따라서 IP나 TCP 헤더와 달리 통신과정에서 필수적인 헤더 필드를 통해 데이터를 은닉화 시키는 것이 쉽지 않다. 하지만, UDP 프로토콜은 네트워크 환경에서 반드시 사용해야 하는 DNS 서비스에 기반을 제공한다는 특징을 가지고 있으며 또한 DNS 서비스 중에서도 DNS 질의/응답은 매우 일반적인 트래픽으로서 어떤 보안 솔루션일지라도 패킷의 흐름을 허용하는 것이 일반적이다. 즉, DNS 서비스는 내부적으로 UDP 프로토콜을 통하여 서비스를 제공하게 되고, 공격자는 이러한 점에 착안하여 DNS 질의/응답 메시지를 은닉채널을 생성하는 바탕으로 사용할 수 있으며, 이렇게 생성된 은닉채널을 통해 원격 서버에서 쉘을 실행시켜 시스템 정보를 얻어 오거나 미리 약속된 상대 시스템의 동조자로부터 특정 정보를 얻어 올 수 있다. 이러한 UDP 기반의 DNS 은닉채널을 생성하는 방법을 UDP 쉘(UDP Shell) 또는 DNS 터널링(DNS Tunneling) 기법이라고 일컬으며, 실제 예로서는 Fryxar가 만든 tunnelshell, Oskar Pearson가 만든 DNS Tunnel 그리고 THC' svan Hauser에 의해 만들어진 Daemonshell 등의 공개 도구가 있다.

이러한 은닉채널의 형성이 가능한 이유는 앞서 설명과 같이 네트워크 환경에 있어 UDP 포트 53번을 사용하는 DNS 서비스의 경우 일반적으로 방화벽과 같은 보안 솔루션에 의해 필터링 되지 않기 때문이며, 그러므로 DNS 질의/응답 메시지를 사용하는 은닉채널의 경우 주의 깊게 메시지의 내용을 관찰하지 않는 한 그 탐지가 어려우며, DNS 질의/응답 메시지를 통한 내부 데이터의 유출을 막는 것은 어렵다.

4. TCP + HTTP 프로토콜

현재의 인터넷과 거의 동의어로 사용되는 웹서비스는 HTTP라는 프로토콜을 기반으로 동작한다. 이러한 HTTP를 사용하는 웹서비스는 인터넷 트래픽의 거의 대부분을 차지하는 것은 물론이고 어떤 네트워크 환경에서도 반드시 서비스되어지는 것이 현실이다. 따라서, 이러한 HTTP 프로토콜을 이용한 은닉채널 형성 기법은 TCP/IP 트래픽을 이용한 은닉채널 형성 방법 중 가장 널리 사용되는 방법 중의 하나이며 다양한 공격 도구들이 나와있다. 또한 이 HTTP 프로토콜은 TCP 서비스를 기반으로 하기 때문에 HTTP를 이용한 은닉채널의 형성은 항상 TCP 전송 메커니즘을 염두해 두어야 한다. HTTP 프로토콜을 이용하여 은닉

```
GET http://www.somehost.com/cgi-bin/board.cgi?view=12121212 / HTTP/1.0
Host: www.somehost.com
User-Agent: Mozilla/5.0 (12121212)
Accept: text/html
Accept-Language: en,fr,en,fr,en,en,en,en
Accept-Encoding: gzip,deflate,compress
Accept-Charset: ISO-8859-1,utf-8,ISO-1212-1
CONNECTION: close
Proxy-Connection: close
X-Microsoft-Plugin: unexpected error #12121212
```

〔그림 6〕 HTTP 헤더 내 데이터 숨김 예제

채널을 형성하는 방법은 HTTP가 지원하는 GET, POST, HEAD 메소드 요청에 데이터를 은닉화하여 일반 HTTP 트래픽과 구분하게 어렵게 만드는 방식 또는 공격자가 공격 대상으로 여기는 네트워크 내부의 주소를 가지고 나오는 HTTP 트래픽에 특정 정보를 삽입하여 외부로 정보를 유출하는 방법 등 다양하게 존재한다. 특히 HTTP를 이용한 은닉채널은 HTTP 프락시 서버를 이용하여 내부 네트워크에 존재하는 특정 서버에 접근을 허용해주기도 한다. 이러한 TCP 기반의 HTTP 은닉채널을 생성하는 방법을 HTTP 쉘(HTTP Shell), HTTP 터널링(HTTP Tunneling), web 터널링(Web Tunneling) 기법이라고 일컬으며, 실제 예로서는 THC의 van Hauser에 의해 만들어진 Rwwwshell, Brinkhoff에 의해 제작된 GNU httptunnel, Jos Visser의 proxy-tunnel 그리고 Alex Dyatlov에 의한 web shell 등의 공개 도구가 있다.

이러한 HTTP를 이용한 은닉채널의 형성이 가능한 이유는 앞서 설명된 것과 같이 네트워크 환경에 있어 HTTP를 이용하는 웹서비스는 필수적인 요소이고, TCP 포트번호 80을 이용하는 HTTP 서비스의 경우 일반적으로 방화벽과 같은 보안 솔루션에 의해 필터링 되지 않기 때문이다. 하지만, 이러한 HTTP 은닉채널의 공격이 가능할지라도 HTTP 트래픽을 분석하는 것은 HTTP 트래픽이 네트워크 트래픽 양에 차지하는 비율을 고려할 때 쉽지 않은 일이다. 특히 내부 네트워크로부터 나오는 HTTP 트래픽의 경우 은닉채널을 생성하였다면 그 탐지는 더더욱 어렵다.

5. HTTP 메시지

비밀 데이터를 감추기 위한 방법은 HTTP request/response 메시지 헤더(header)와 내용(body) 부분에 적용된다[26].

5.1 HTTP 헤더 내 데이터 숨김

클라이언트 측에서 보내고자하는 비밀 데이터는 HTTP 헤더 내에 숨겨져 전송된다. 적은 양의 비밀 데이터를 전송한다고 가정하고 비밀 데이터를 숨기기 위한 방법들을 생각해 보자. 그림 6은 HTTP proxy를 통한 실제 브라우저 GET 요청이다. 물론, 전송해야 할 비밀 데이터의 크기가 크다면 데이터의 길이를 제한해야 할 것이다. 스테가노그래피를 사용한 방법에서는 은밀 데이터가 일반 데이터의 크기보다 훨씬 더 크다고 가정한다. 그렇지 않다면 HTTP 요청의 많은 수 때문에 은닉채널이 보호되지 못한다. 감시자를 혼란시키기 위한 방법은 다음과 같다.

(1) URI 스트링은 공개 웹 서비스와 동작하는 것처럼 보인다. 하지만 데이터는 추가적인 질의 문자열에 의해 부호화 되어있다.
"view=12121212"

(2) "User-Agent" 항목으로 부호화(encoded)된 데이터 "(12121212)".

(3) 가짜 문자열 요청(fake charset request)로 부호화된 데이터 "ISO-1212-1".

(4) 확장 헤더 문자열 내로 부호화된 데이터
"X-Microsoft-Plugin: unexpected error #12121212"

여기서 사용되는 스테가노그래피 기법들은 다음과 같다.

(1) 문자열 명령의 형태 차이를 이용해 정보를 숨길 수 있다.
"Accept: test/html", "Accept-Language: en"=0
"Accept-Language: en", "Accept: text/html"=1

(2) 특정 항목 또는 정보의 존재 여부에 따라 숨길 수 있다.
"Accept-Encoding" 항목이 존재하면 = 0

"Accept-Encoding" 항목이 존재하지 않으면=1

(3) "en" = 0 이고 "fr" = 1 이라고 가정하면, 아래와 같은 항목에서 1 바이트를 숨길 수 있다.
Accept-Language: en, fr, en, fr, en, en, en, en
= 01010000 = 0x50

(4) 필드 이름의 민감한 특징을 이용하면 정보를 숨길 수 있다.
(ex) 대소문자의 차이 이용
"Connection: close" = 0
"CONNECTION: close" = 1

위의 예제를 통해 정보를 숨기기 위한 몇 가지 방법을 보았다. 이러한 방법들은 HTTP 프로토콜을 관점에서 모두 유효하며 숨겨진 각 정보의 조합을 통한 정보의 이용도 가능하다. HTTP proxy나 server와 같은 제3의 허위 프로그램을 사용하는 경우에는 스테가노그래피 부호화의 방법이 훼손될 수 있다. 하지만 HTTPd 서버 모델 또는 CGI 서버 모델과 함께 위 예제의 (2)번 방법이 사용된다면 스테가노그래피는 멋지게 적용된다.

이러한 스테가노그래피를 이용한 방법은 비밀 데이터의 양이 많을 때 부적당하며, 은닉채널을 위한 채널 생성의 초기에 사용되는 것이 적당하다. 일례로 서버 측에서의 발생하는 "0"과 "1"이라는 정보는 은닉채널 클라이언트 프로그램에게 채널 형성을 위한 각 단계의 성공 여부에 대한 정보가 된다.

5.2 HTTP body 내 데이터 숨김

HTTP 몸체(body)는 헤더 부분에 비해 공간이 크기 때문에 비밀 데이터를 숨길 가능성은 더욱 크다. HTTP 몸체 부분에서의 데이터 숨김 기법도 이전의 HTTP 헤더에 적용된 기법과 같다. 감시자는 거의 실제 요청으로 혼돈하기 쉽다. 하지만 실제로는 스테가노그래피 기법에 의해 비밀 데이터가 숨겨져 있다. 그림 7은 HTTP proxy를 경유하여 보낸 실제 브라우저의 POST 요청이다. 감시자를 혼란시키기 위한 방법은 다음과 같다.

(1) 요청 메시지 몸체를 데이터처럼 볼 수 있다. 은닉 데이터는 패스워드 값으로 여겨진다.
"..pass=12121212.."

(2) 요청 메시지는 이미지처럼 여겨질 수도 있고 "Content-type" 항목은 "image/gif"이 된다. 은닉 데이터는 다음과 같이 보여 진다.
"GIF89121212121212.."
은닉 채널 서버와 클라이언트는 데이터를 이미지로서 어떻게 처리할지 알고 있기 때문에 실제 전송 형태에서 Content-type이 multimedia type 메시지로 나타나는 것은 문제가 되지 않는다.

(3) 응답 메시지는 표준을 따르건 따를지 않건 간에 HTML 페이지내에 inbound 데이터를 실어 보낸다.
〈121212〉〈/121212〉 또는
〈img src=".." alt="121212〉

여기서 사용되는 스테가노그래피 기법들은 다음과 같다.

(1) 숨겨진 메시지는 text 메시지 부분 내에 놓일 것이다. Mike에 의해 제출된 이 메시지는 "^&*(%" 또는 단어들 사이의 공백, 대·소문자에 의해 부호화될 수 있고 임의의 text 메시지와 은밀 데이터는 사전 파일에 의해 부호화 될 수 있다.

(2) 요청/응답 메시지는 하나 또는 그 이상의 이미지를 지닌 은닉 데이터를 포함한다. 이 이미지는 감시자가 텍스트 내용을 혼돈하고 있는 동안 데이터의 덮개로 사용될 수 있다.

(3) 응답 메시지는 HTML 페이지 내에 부호화된 inbound 데이터를 포함한다. 부호화된 inbound 데이터에 대한 스테가노그래피 기법으로는 코드 정

```
POST http://www.somehost.com/cgi-bin/linuxchat.cgi?action=newmsg HTTP/1.0
Host: www.somehost.com
User-Agent: Mozilla/5.0
Content-Type: multipart/form-data
Content-Length: 75
Connection: close
Proxy-Connection: close

name=Mike&pass=12121212&message="_ editor is great and _ is a ^&*(% !!!"
```

〔그림 7〕 HTTP body 내부의 데이터 숨김

보를 표준과 다른 명령 아래 태그 속성으로 변환하는 기법을 제시한다. 최대 300byte의 내용이 16Kb 크기의 HTML 데이터로 부호화 된다.

ex) 〈img src="../" alt="../"〉 = 0
〈img alt="../", src="../"〉 = 1

IV. 정보은닉 기법 분석 - 은닉기법별 분류

본 장에서는 인터넷 트래픽에서의 정보은닉 기법을 TCP/IP 프로토콜에 따른 분류가 아닌 대표적으로 알려진 정보은닉 기법 별로 분류하여 분석하겠다. 좀 더 세부적인 내용은 [2]의 논문을 참고하면 된다.

1. 패킷 헤더 필드 조작

TCP/IP 프로토콜의 헤더 조작을 통한 은닉 채널 생성기법은 앞서 소개된 프로토콜별 은닉기법 분석에서 가장 널리 사용되는 기법 중의 하나이며, 이미 설명된 HTTP, TCP, UDP, IP, ICMP 프로토콜에 이외에도 IGMP와 같은 프로토콜로도 가능하다고 알려져 있다. 또한 본 절에서는 그 중에서도 3장에서 다룬 IP 프로토콜 기법에서 설명된 것 보다 좀 더 세부적인 방법으로서 IPv4 패킷의 헤더필드를 이용한 몇 가지 은닉채널 생성 시나리오에 대해서 알아보겠다.

첫번째로 IP 헤더의 DF 필드를 이용한 은닉기법은 DF 필드와 MTU(Maximum Transfer Unit)간의 관계를 이용하여 정상적인 IP 패킷 내부에 한 비트의 숨겨진 통신 채널을 생성하게 된다. 간단하게 예를 들어 MTU가 1000bytes인 통신 채널을 지나는 1000bytes 미만의 IP 패킷들은 모두 DF 필드 값이 1로 설정될 것이다. 즉, DF 필드의 조작을 통해 은닉채널을 형성하려는 사용자는 1000bytes 보다 작은 특정 크기의 패킷을 사용하여 DF 필드에 임의의 값을 전송 할 수 있다. 이것은 간단한 은닉저장채널(Covert Storage Channel)의 예가 될 수 있다. 단, 이러한 은닉저장채널을 형성하기 위해서는 은닉채널을 형성하려는 두 사용자 간에 두 사용자가 이용하는 통신 채널의 MTU와 그러한 MTU 보다 작은 특정 크기의 패킷 사이즈를 정의하는 것이 요구된다.

첫번째 설명된 DF 필드를 이용한 은닉채널형성 기법은 MTU와 패킷 사이즈를 아는 두 당사자만이 사용 가능한 방법이었다. 하지만, 첫 번째 은닉채널 형성 기법에 Identification 필드를 함께 사용하게 되면 one-to-one이 아닌 one-to-many로 은닉채널을 생성할 수 있다. 즉, Identification 필드의 패킷 식별 특성을 이용하여 생성 여러 은닉채널을 구분하는데 Identification 필드를 사용하는 것이다.

지금까지의 방법은 같은 네트워크 안에서 MTU 사이즈를 미리 알아야 은닉채널 형성이 가능하다는 근본적인 문제점을 가지고 있다. 하지만 IP 헤더의 버전 필드(Versions)와 헤더 길이 필드(Internet Header Length) 그리고 Identification 필드의 XOR 값을 이용하여 사전 지식 없이 어떠한 네트워크 환경에서도 사용 가능한 은닉채널 기법을 알아보겠다. 하지만, 이 기법 역시 IP 헤더가 옵션 필드를 사용하지 않아야 한다는 제한점은 가지고 있다. 특히 여러 필드들의 XOR 값을 이용하는 이 방법은 보통의 이상 패킷 탐지 방법들이 단편적인 패킷 필드 각각을 조사하는 취약성을 이용하여 여러 필드들의 XOR 조합을 이용하는 특징을 가지고 있다. 은닉채널을 생성하는 방법은 아래와 같다.

Encoding

1. 가정 : IPv4 헤더를 사용(단, 옵션 필드는 설정되지 않은 패킷들만을 이용)
2. 1의 가정에 따르면 Versions = 0100, IHL = 0101의 값을 항상 갖는다. 이때 각 비트들을 연속하여 다음과 같이 정의한다.
   $\{h_1, h_2, \ldots, h_8\} = \{0, 1, 0, 0, 0, 1, 0, 1\}$
3. 16bits Identification 필드는 $i_1, i_2, \ldots i_{16}$으로 정의한다. 또한 이 16bits 필드에 은닉된 값이 저장될 것이다.
4. 16bits Identification 필드의 첫 8bits에는 은닉 데이터(Ci)와 hi 데이터의 XOR 값이 저장되며, 나머지 8bits에는 임의로 생성된 8bits의 값이 저장된다.
   즉, $i^*_l = h_l \oplus C_l$, $l=1,2, \ldots ,8$
   $i^*_9, i^*_{10}, \ldots, i^*_{16} = \{$임의로 선택된 8bits$\}$
5. 예제)
   버전 및 IHL 필드값 :
   $\{h_1, h_2, \ldots, h_8\} = \{0, 1, 0, 0, 0, 1, 0, 1\}$
   은닉 데이터(영문 "A"의 ASCII 값)
   $\{c_1, c_2, \ldots, c_8\} = \{0, 1, 0, 0, 0, 0, 0, 1\}$
   Identification 필드의 첫 8bits
   $\{i^*_1, i^*_2, \ldots, i^*_8\}$
   $= \{0,1,0,0,0,1,0,1\} \oplus \{0,1,0,0,0,0,0,1\}$
   $= \{0,0,0,0,0,1,0,0\}$
   Identification 필드의 마지막 8bits(임의 선택)
   $\{i^*_8, i^*_9, \ldots, i^*_{16}\} = \{0,1,0,1,0,1,0,0\}$

완성된 은닉 데이터를 가진 Identification 필드
$\{i^*_1,i^*_2, \ldots, i^*_{16}\}=\{0,0,0,0,0,1,0,0,0,1,0,1,0,1,0,0\}$

Decoding

1. 수신측에서는 버전 필드와 IHL 필드 8bits와 Identification 필드의 첫 8bits의 XOR 연산을 통하여 손쉽게 은닉화 된 데이터를 복호화 할 수 있다.

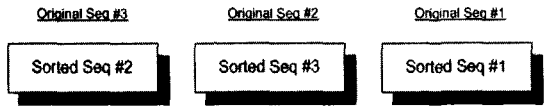
$\{c_1, c_2, \ldots, c_8\}$
$= \{h_1, h_2, \ldots, h_8\} \oplus \{i^*_1,i^*_2, \ldots, i^*_8\}$
$= \{0,1,0,0,0,1,0,1\} \oplus \{0,0,0,0,0,1,0,0\}$
$= \{0,1,0,0,0,0,0,1\}$ -> ASCII 값 "A"

이와 같은 XOR 연산을 통한 패킷 조합 은닉화 기법은 앞서 서술된 것처럼 단편적인 패킷 필드 검사에서 드러나지 않을뿐더러, 패킷 필터링 방화벽에도 강점을 가진다. 또한 Identification 필드 마지막 8bits의 무작위성으로 인하여 보다 일반적인 통신 패킷으로 보이기 때문에 상태 조사 방화벽 등에서도 장점을 가진다고 할 수 있다.

2. 패킷 정렬화

패킷 정렬화를 통한 정보은닉 기법은 기존의 TCP/IP 헤더 필드 조작 기법과 같이 특정 헤더나 페이로드의 내용에 의존적이지 않다는 장점을 가지고 있다. 이러한 패킷 정렬화를 통한 정보은닉 기법은 앞서 패킷 필드 조작이나 프로토콜별 정보은닉 기법 분석에서 사용되었던 필드에는 적용되지 않는다. 먼저 TCP 헤더에는 패킷 정렬에 관련된 Sequence Number 필드와 Acknowledge Number 필드가 있다. 하지만 이 두 필드들은 직접적으로 패킷의 정렬 순서에 관련이 없을뿐더러 전송과정 중 IP 계층에서의 분할/재조립에 영향을 받는다. 또한 IP 헤더의 경우 패킷 정렬에 관련된 Identification 필드가 존재한다. 하지만 Identification 필드 역시 패킷 정렬에 직접적인 연관성이 없으며 또한 송/수신 ip 주소와 ip가 전송하는 프로토콜에 의존적인 특성을 가지고 있다. 따라서 본 논문에서는 패킷 정렬화를 통한 인터넷 트래픽에서의 정보은닉 기법을 분석하기 위하여 IPSec 프로토콜의 anti-replay 메커니즘에 사용되는 Sequence Number 필드를 사용한다. 이 필드의 주 목적 replay attack을 탐지하는 것이다. 그러므로 이 필드는 직접적으로 패킷의 순서에 연관성을 가지고 있다. anti-replay 메커니즘이란 수신된 패킷이 중복되어 수신되었는지를 결정해주는 기능을 말하는 것으로서 IPSec 프로토콜에서는 Sequence Number 필드를 사용하여 통신 과정 중 SA(Security Association) 협상 과정에서 Sequence Number 필드를 0으로 설정하고 그 후 1씩 증가시켜 패킷간 정렬 기능을 제공할 수 있다.

앞서 설명한 것과 같은 Sequence Number 필드를 이용한 패킷 은닉 기법은 다음의 그림 x와 같이 세 개의 연속된 패킷들이 있다고 가정하며, 연속된 패킷은 실제 패킷간의 원래 순서를 가지며, 또한 그림 x의 아래와 같이 임의로 정렬된 순서를 가질 수 있다. 즉, 표 x와 같이 세 개의 패킷간 정렬 기법을 이용해서는 6개의 가능한 메시지를 전달할 수 있는 것이다.



〔그림 8〕 임의로 정렬된 IPSec 패킷 모델

〔표 2〕 세 개의 패킷을 이용한 가능한 메시지 조합

No.	정렬된 조합	가능한 메시지
1	123	001
2	132	010
3	231	011
4	213	100
5	321	101
6	312	110

이번에는 실제로 패킷 정렬화를 통한 은닉채널 형성에 대해서 알아보겠다. 먼저 이러한 은닉채널을 형성하기 위해서는 두 당사자간에 main key, sub key를 공유하고 있어야 하며, 수신측에서는 수신되는 sequence를 매핑 할 수 있는 표 x와 같은 은닉 데이터 메시지 조합을 알고 있어야 한다. 먼저 원본 sequence를 정의하면,

- $O=(X_1,X_2, \ldots, X_K)$: O는 원본 sequence의 집합이며, K는 main key이다. 또한 X_i는 패킷 순서 중 i번째 패킷을 나타낸다.
- $P=S(1), S(2), \ldots S(N)$: P는 원본 sequence들로 구성될 수 있는 전체 sequence의 집합이며, 즉, P의 원소 각각의 상대에게 전달되어 임의의 메

시지를 나타내게 된다. 하지만 여기서 N≠K!이다. 즉, main key K의 조합으로 나올 수 있는 모든 sequence의 수와 N의 수가 동일하지는 않다. 그러므로 P는 chaotic sequence structure의 성질을 가진다.

· $S(i)=(X'_1, X'_2, \ldots, X'_K)$ where $S(i)\in P$ and $I=1,2, \ldots, N$: 여기서 S(i)는 전체 패킷의 정렬 집합 P의 원소 중 하나로서 임의로 정렬된 패킷의 sequence 중 하나를 나타낸다. 이와 같은 S(i)의 계산은 다음과 같다.

Sorting sequence
= S(i)
= S(X'(M-1)K+x)
= (M-1)K + O(x) : x≤K

여기서 M은 K의 곱을 의미하며, 만약 정렬하려는 패킷의 수가 K의 1배수 안에 포함된다면, 그때 M은 1의 값을 가지면 만약 K의 2배수 안에 포함된다면 그때 M의 값은 2가 된다. 또한 O(x)는 원본 sequence를 나타내며 main key보다 작거나 같은 것을 가진다. 또한 여기서 재 정렬이란 결국 sorting된 값을 다시 원본 sequence로 변환하는 것과 같은 의미라는 것을 알 수 있다.

R=S(i) : 여기서 R은 재 정렬 된 즉, 수신된 패킷을 나타낸다. 그러므로 R의 의미를 파악해내기 위해서는 다음과 같은 방법을 사용한다.

Resorting sequence
= R(i)
= R(X"(M-1)K+x)
= (M-1)K+O(X"(M-1)K+x-(M-1)K) : x≤K

Example

가정 : main key는 8, 패킷의 수는 16, subkey는 1, sequence number의 값은 1, 16 패킷에 대한 정렬값은 다음의 표 x와 같다.

〔표 3〕 가정에 의해 정렬된 sequence

O: X_i	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1
S(1): X'_i	14	11	16	13	10	15	12	9	6	3	8	5	2	7	4	1

R(1) = (1-1)8 + O(1) : x≤8
= O(1)

R(2) = (1-1)8 + O(4) : x≤8
= O(4)
..... (이하 생략)

R(16) = (2-1)8 + O(16-8) : x≤8
= 8 + O(8) = 8 + 6 = O(14)

〔표 4〕 재정렬된 sequence

R: X''_i	14	11	16	13	10	15	12	9	6	3	8	5	2	7	4	1
O: X'_i	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1

V. 향후 연구 방향

기존의 은닉채널에 관한 연구는 주로 음성, 화상, 동영상 등의 멀티미디어 데이터에 대한 정보은닉 기법에 대해서 주로 이루어졌다. 하지만, 인터넷 환경의 프로토콜로서 사실상 표준(de facto)으로 불리는 TCP/IP 프로토콜을 이용한 인터넷 트래픽에 대한 은닉채널 형성기법 또한 활발히 연구 되고 있는 중이며, 은닉채널 형성을 통한 공격 기법 자체에 대한 연구 역시 많은 연구가 수행중이다. 그러나, 이런 은닉채널 형성 기법들을 사용한 공격 기법이나 공격 도구에 의한 대응 방안 연구는 매우 미미하다고 할 수 있다. 은닉채널 형성을 통한 공격은 현재까지 알려진 일반적인 트래픽 분석 등과 같은 방안을 사용해서는 일반 네트워크 트래픽과 은닉채널이 형성된 인터넷 트래픽간의 차이점을 발견하는 것이 매우 어렵다. 따라서 향후에는 은닉채널공격에 대한 Support Vector Machine, 신경망등과 같은 기계 학습기법을 은닉채널 탐지에 적용하여 일반적인 인터넷 트래픽과 은닉채널이 형성된 트래픽간의 판별에 이용되어야 할 것이다.

VI. 결 론

은닉채널이란 매개체를 이용하여 특정 정보를 은닉해주는 통신채널로 쉽게 간주할 수 있다. 이러한 은닉채널에 대한 위험성은 보통 멀티미디어 데이터에 대한 스테가노그래피 형태로 알려져 왔으나, 현재에는 네트워크 트래픽을 이용한 공격 방안 역시 그 위험성이 대두 되고 있으며, 이러한 인터넷 트래픽을 이용한 은닉채널 공격은 현재 널리 사용되고 있는 시스템 및 네트워크 취약성을 이용한 해킹 공격들보다 훨씬 큰 잠재적 위험성을 내포하고 있다.

본 논문에서는 현재 인터넷 환경의 사실상 표준인 TCP/IP 프로토콜에 초점을 맞추어 각 프로토콜 특성에 따른 은닉채널 형성 기법과 형성 기법에 따른 은닉채널을 각각 분석하여 향후 인터넷 트래픽 은닉채널 탐지 및 대응에 기반을 마련하였다.

참 고 문 헌

[1] Department of defence trusted computer system evaluation criteria, Tech. Rep. DOD 5200.28-ST, Department of Defence, December 1985. Supersedes CSC-STD-001-83.

[2] K. Ahsan and D. Kundur, "Practical Data Hiding in TCP/IP" Proc. Workshop on Multimedia Security at ACM Multimedia '02, 7 pages, French Riviera, December 2002.

[3] B. W. Lampson, A note on the confinment problem, in Proc. of the Communications of the ACM, no. 16:10, pp. 613 615, October 1973.

[4] P.A. Porras and R.A. Kemmerer. Analyzing covert storage channels. In Proc. 1991 Symposium on Research in Security and Privacy, pages 36--51, Oakland, CA, May 1991. IEEE Computer Society.

[5] J.C. Wray. An analysis of covert timing channels. In Proc. 1991, Symposium on Research in Security and Privacy, pages 2--7, Oakland, CA, May 1991. IEEE Computer Society.

[6] J. McHugh, Covert Channel Analysis, Technical Memorundum 5540:080A, Naval Research Laboratory, Washington D.C., 1995. A Chapter of the Handbook for the Computer Security Certification of Trusted Systems.

[7] Neil F. Johnson et al, Information Hiding: Steganography and Watermarking - Attacks and Countermeasures, Kluwer Academic Publishers, 2000.

[8] Fabien A. P. Petitcolas, editor. Information hiding. Proceedings of the 5th international workshop on information hiding, vol. 2578 of Lecture Notes in Computer Science, Noordwijkerhout, The Netherlands, 7.9 Oct 2002.

[9] daemon9 and alhambra. "Project Loki: ICMP Tunnelling." Phrack Magazine, Volume Seven, Issue 49, File 6 of 16, August 1996. URL: http://phrack.infonexus.com/search.phtml?view&article=p49-6

[10] C. H. Rowland, Covert channels in the TCP/IP protocol suite, Tech. Rep. 5, First Monday, Peer Reviewed Journal on the Internet, July 1997. URL:http://www.psionic.com/papers/covert/covert.tcp.txt

[11] Brinkhoff, Lars. "GNU httptunnel." August 31, 2000,ttp://www.nocrew.org/software/httptunnel.html(November, 2000).

[12] John Giffin, Covert Messaging Through TCP Timestamps, PET2002

[13] Christopher Abad, IP Checksum covert channels and selected hash collision, www.securityfocus.com, 2001

[14] Oskar Pearson, "DNS Tunnel - through bastion hosts", ICON, http://www.icon.co.za/~wosp/

[15] Fryxar, tunnelshell, http://www.geocities.com/fryxar/

[16] van Hauser., rwwwshell, http://www.thc.org/releases/rwwwshell-2.0.pl.gz

[17] Jos Visser, proxytunnel, www.josvisser.nl/proxytunnel/

[18] Alex Dyatlov, web shell, http://www.entreelibre.com/simsim/wsh/

[19] edi and teso. "ICMP tunneling tool," July 10, 1999. URL: http://teso.scene.at/releases.php3, http://teso.scene.at/releases/itunnel-1_2.tar.gz

[20] BiT. "ICMP backdoor client and server." Confidence Remains High,

Issue 9, May 11, 1998. URL: http://www.hackersclub.com/km/magazines/crh/crh009.txt

[21] van Hauser. "Placing Backdoors Through Firewalls," v1.5, May 1999. URL: http://thc.pimmel.com/files/thc/fw-backd.htm

[22] Gina Fisk,Eliminating Steganography in Internet Traffic with Active Wardens, F.A.P. Petitcolas (Ed.): IH 2002, LNCS 2578, pp. 18 35, 2003. Springer- Berlin Heidelberg 2003

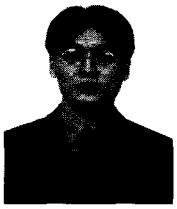
[23] Sohn TaeShik, Seo Jung-Taek, Moon Jong-Sub, "A Study on the Covert Channel Detection of TCP/IP Header using Support Vector Machine", ICICS 2003, LNCS 2836, Springer-Verlag Berlin Heidelberg 2003

[24] Alexander Grusho, "Mathematical Models of the Covert Channels", V.I. Gorodetski et al. (Eds.): MMM-ACNS 2001, LNCS 2052, pp. 15-20, 2001. Springer-Verlag Berlin Heidelberg 2001

[25] Alexandre Grusho and ElenaTimonina, "Construction of the covert channels", V.I. Gorodetski et al. (Eds.): MMM-ACNS 2003, LNCS 2776, pp.428-431, 2003. Springer-Verlag Berlin Heidelberg 2001"

[26] Alex Dyatlov, Simon Castro, Exploitation of data streams authorized by a network access control system for arbitrary data transfers : tunneling and covert channels over the HTTP protocol.v1.0, June 2003, http://www.gray-world.net

〈著 者 紹 介〉



손 태 식 (TaeShik Shon)

2000년 2 : 월아주대학교 정보 및 컴퓨터 공학부 졸업(공학사)
2002년 2월 : 아주대학교 정보통신공학과 졸업(공학석사)
2002년 3월~2004년 2월 : 고려대학교 정보보호대학원 박사수료
2002년 8월~현재 : 고려대학교 정보보호기술연구센터
2003년 5월~12월 : ICU 부설 정보통신교육원,서경대, KISEC 강사
2004년 2월~현재 : 방문연구원, University of Minnesota
〈관심분야〉 네트워크 · 시스템보안, 인터넷프로토콜 보안

박 종 성 (JongSeong Park)

2002년 2월 : 경남대학교 졸업(공학사)
2003년 2월~현재 : 고려대학교 정보보호대학원 석사과정
〈관심분야〉 네트워크 · 시스템보안, 컴퓨터 포렌식

문 종 섭 (JongSub Moon)

1981년 2월 : 서울대학교 계산통계학과 학사
1983년 2월 : 서울대학교 계산통계학과 석사
1992년 2월 : Illinois Institute of Technology 박사
1993년~현재 : 고려대학교 전자 및 정보공학부 교수
고려대학교 정보보호대학원 겸임 교수
〈관심분야〉 IDS, 신경망, 생체인식, 운영체제