동작 메커니즘 진단을 기반으로 한 백도어(backdoor) 행동분석 방법 모델 연구★

나상엽* · 노시춘*

요 약

백도어의 침투공격 형태는 "trapdoor" 침투점을 이용하여 보안기능을 우회하고 데이터에 직접 접근을 허용하게 한다. 백도어는 소스코드 수정 없이도 코드 생성이 가능하고 심지어 컴파일 후에 수정도 가능하다. 이같은 방식은 컴파일러를 다시 작성하여 소스코드를 컴파일 할 때 특정 부분에 백도어를 삽입시키는 방법을 사용하기 때문에 가능하다. 백도어 방역작업에는 백도어 기본적 구조나 특성에 따라 피해영역이나 차단방법이 조금씩 다를 수 있다. 본 연구는 백도어 동작 메커니즘 진단을 기반으로 하여 백도어 행동분석 방법 모델을 도출하였다. 연구의 목적은 백도어에 대한 구조 및 감염형태를 파악하고 행동방식을 분석함으로서 앞으로 악성코드 대응과 해킹공격에 유용하게 활용할 수 있는 정보를 확보하는 데 있다.

A Study of Action Research Analysis Methods Model of Backdoor Behavior based on Operating Mechanism Diagnosis

SangYeob Na* · SiChoon Noh*

ABSTRACT

Form of backdoor penetration attacks "trapdoor" penetration points to bypass the security features and allow direct access to the data. Backdoor without modifying the source code is available, and even code generation can also be modified after compilation. This approach by rewriting the compiler when you compile the source code to insert a specific area in the back door can be due to the use of the method. Defense operations and the basic structure of the backdoor or off depending on the nature of the damage area can be a little different way. This study is based on the diagnosis of a back door operating mechanism acting backdoor analysis methods derived. Research purposes in advance of the attack patterns of malicious code can respond in a way that is intended to be developed. If we identify the structures of backdoor and the infections patterns through the analysis, in the future we can secure the useful information about malicious behaviors corresponding to hacking attacks.

Key words : Backdoor Behavior, Diagnosis, Vulnerability, Defense Mechanism, Malicious Code

접수일(2014년 3월 4일), 수정일(1차: 2014년 3월 16일), 게재확정일(2014년 3월 24일)

★ 본 논문은 남서울대학교 학술연구비 지원에 의해 연구되었음

* 남서울대학교 컴퓨터학과

1. 서 론

오늘날 악성코드는 기술적으로 더욱 지능화되고 있고 파괴력 또한 더욱 강화되는 추세이며, 형태적으로는 바이러스(virus), 웜(worm), 트로이 목마(tryan)의 다양한 조합형으로 진화하고 있다. 날로 강력해지는 악성코드는 고속화, 글로벌화한 초고속 네트워크 인프라를 타고 순식간에 확산되므로 파괴력과 위험성은 가공할 수준에 이르고 있다. 악성코드 침투 시 일반적 대처는 먼저 바이러스 백신을 개발하여 바이러스 월(viruswall)상에 장착된 engine을 update한다. 이 방법이 현재까지 기술적으로 거의 유일한 방안이나 이 방법이 문제를 원천적으로 해결하는 것은 아니다. 본 연구는 백도어 방역을 위한 하나의 정보 분석 모델로서 백도어 동작 메커니즘 진단을 기반으로 백도어 행동 분석 방법을 제안한다. 백도어는 그 자체의 위험성과 함께 해킹공격의 필수도구인 트로이목마 활동에서의 백도어의 역할에도 주목해야 한다. 본 방법론 제안이유는 해킹공격에 사용되는 악성코드의 대표가 바로 트로이목마이며 이 트로이목마가 대개는 백도어를 활용하여 타겟 시스템에 침투한다는 사실이다. 백도어 행동을 분석한다면 제반 악성코드의 재 침입이나 해킹 공격에 대응할 수 있는 정보를 획득할 수 있고 아울러 악성코드 대응 시 개선된 방역전술 기본 자료로도 활용할 수 있다. 논문 기술순서는 서론, 백도어 동작 메커니즘 진단, 백도어 방역 시 위험성 사례, 백도어 행동 분석 방법 모델, 결론의 순서이다.

2. 백도어 동작 메커니즘 진단

2.1 백도어 Infection 메커니즘

백도어의 원래 의미는 운영체제나 프로그램을 생성할 때 정상적인 인증 과정을 거치지 않고, 운영체제나 프로그램 등에 접근할 수 있도록 만든 일종의 통로로서 다른 말로 Administrative hook이나 트랩 도어(Trap door)라고도 한다. 백도어는 백 채널(back channel), 불법 서버(illicit server)로도 분류된다. 백도어의 능동적인 침투행태는 "trapdoor" 침투점을 이용하여 보안기능을 우회하고 데이터에 직접 접근을 시도하는 방법이다. 소스코드 수정 없이도 백도어를 만드는 것과 심지어 컴파일 후에 수정하는 방법으로 백도어를 제작하는 것도 가능하다. 이는 컴파일러를 다시 작성하여 소스 코드를 컴파일 할 때 특정 부분에 백도어를 삽입시키는 것 이다. 사용자가 문서화 되지 않은 특정 입력을 제공하면 어떤 프로그램의 기능에 접근할 수 있게 된다. 시스템이 Trusting Trust 컴파일러와 같은 백도어 또는 트로이목마에 영향을 받으면 "정당한" 사용자가 시스템의 제어권을 되찾는 것은 매우 어렵다. 광대역 망에 연결된 PC로 불안전한 마이크로소프트 윈도와 아웃룩을 실행하는 PC가 표적이 된다[1][2][5].

2.2 백도어 대응절차

사용자 PC 환경에서의 백도어에 대한 일반적 보안대책은 악성코드에 대처하는 방법과 동일하게 백도어를 진단하고 파일을 삭제한다. 백도어 방어 에서는 차별화되는 방법은 바이러스 치료와 같은 기생형 악성코드 치료방법을 사용하지 않으며 일반적으로 프로세스를 중지시키거나 또는 특정의 백도어 유형에서는 레지스트리 삭제 등 다음과 같은 방법을 사용한다[4][6][8].

▪ 특정 백도어의 경우에는 프로세스를 중지시키는 방법이다. 예를들면 2368 프로세스를 'Kill Process Tree(Shift Del)'로 중지시키는 방식이다.

▪ 또 다른 특정한 예는 백도어 파일의 삭제방법 이다. 이 방법은 Bifrost 폴더에서 확인한 파일을 삭제하는 경우이다.

-C: \Windows \system32 폴더에서 확인한 파일을 삭제한다.

-C: \Program Files\MisoFile 폴더에서 확인한 파일을 삭제한다.

▪ 레지스트리 삭제

또 다른 경우는 시작 프로그램에서 확인한 사항을 삭제한다. 예를들면 'regedit'로 레지스트리 경로를 확인할 수 있는데, 레지스트리에서 해당 항목을 삭제하는 경우이다.

3. 백도어 방역 시 위험성 사례

▪ 백도어의 첫 번째 위험성은 백도어 감염 시 정상적인 로그인 절차를 거치지 않고 트로이 목마를 침투시켜 백도어로 이용하는 경우 시스템 침입 사실이 은폐된다는 점이다. 국내에서 발견된 Win - Trojan/MircPack 변형들도 엄밀히 말하면 GTBot에 포함될 수 있다. 이 같은 행태는 사후 분석 작업을 매우 어렵게 한다.

▪ 백도어의 두 번째 위험성은 백도어 앞에서 설명한 원리로 인하여 백도어는 시스템 관리자(system manager)의 보안 관리를 우회하여 동작 한다는 점이다. 따라서 관리자가 루트(root) 권한을 사용하여 수시 패스워드 갱신 등 아무리 안전하게 관리하더라도 백도어는 언제든지 시스템에 침입할 수 있다.

▪ 세 번째 위험성은 백도어는 재 침입을 위한 백도어 설치 등이 가능하기 때문에 다른 악성코드 보다 위험성이 특히 높다. 이 위험성으로 인하여 트로이목마 등 타 악성코드가 시스템에 재 침입하는 기능을 제공하게 된다.

▪ 네 번째 위험성은 대부분 백도어 프로그램은 시스템 침입시간이 짧고 로그흔적을 남기지 않고 침입한다는 점이다(wtmp, utmp, lastlog). 이점은 로그분석과 포렌식에 의한 증거채집 유효시간을 벗어나게 함으로서 대응을 어렵게 한다[7][8][9].

4. 백도어 행동분석 방법 모델

이상과 같은 악성코드로서 백도어의 위험성에 대응하기 위해 백도어 행동을 분석한다면 제반 악성코드의 재 침입이나 해킹 공격에 대응하는 정보를 획득할 수 있다. 백도어 방역작업에서는 백도어 기본적 구조나 특성에 따라 피해영역이나 차단방법이 조금씩 다를 수 있다. 백도어 행동분석 방법 모델은 <표1>과 같이 행동분석 절차 4단계 설계와 행동분석 10개 단위기능 테스트 방법으로 설계하였다.

<표1> 백도어 행동분석 방법 모델

행동분석 절차 설계	백도어 행동분석 기능 테스트 방법
모의 백도어 제작 환경 구성	주소를 인자로 넘기는 API 테스트
백도어 구현 시나리오 작성	Temp의 경로 테스트
	레지스트리 등록 테스트
	C:\a.txt를 예외 등록
모의 백도어 코드 작성	전송 문자열 비교
	준비된 루틴으로 이동
백도어 행동분석 기능 테스트	“w”와 “C:\a.txt” 인자로 함수 테스트,
	주소 파일 확인
(4개 단계)	파일내용 버퍼로 이동
	버퍼 내용 전송
	(10개 기능)

4.1 행동분석 절차 설계

행동분석 절차는 모의 백도어 제작환경 구성, 백도어 구현 시나리오 작성, 모의 백도어 코드 작성, 백도어 실행 테스트, 백도어 행동분석을 다음과 같이 차례대로 실시하는 모델로 구성된다.

4.1.1 모의 백도어 제작 환경구성

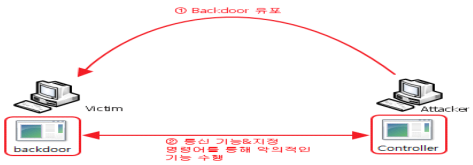
악성코드 동작 메커니즘 진단을 참조하여 백도어 행동방식을 분석을 위한 첫 번째 과제로 모의 백도어 코드를 작성한다. 이때 백도어 제작 및 테스트 구현 환경은 Host PC, Guest PC 1(Attacker), Guest PC 2(Victim)가 필요하다. 본 논문에서 연구목적의 실제 환경구성은 <표3>과 같이 Host PC는 Windows 7, Guest PC 1(Attacker)는 Windows xp sp3, Guest PC 2(Victim)는 Windows xp sp3를 사용한다.

<표2> 모의 백도어 제작환경 사양

Operating System	Version
Host PC	Windows 7
Guest PC 1(Attacker)	Windows xp sp3
Guest PC 2(Victim)	Windows xp sp3

4.1.2 4단계의 백도어 구현 시나리오 작성

<표2>의 모의 백도어 제작환경 사양을 활용하여 백도어 구현 시나리오는 attacker가 유포한 백도어를 victim에서 실행 시켰을 때 attacker가 문자열 형태로 명령을 전송하여 악의적 기능을 수행 하도록 하는 4단계의 순서이다. 이 시나리오 진행 프로세스는 ① victim(Server) - attacker (Client)간 통신 기능을 수행한다. ② “MyBack” 이라는 명칭으로 victim 의 방화벽에 특정파일 예외 등록을 한다. 향후 Dropper 제작 시에 활용하기 위한 기능을 제작한다. ③ 부팅 시 백도어 자동 실행을 위해 VMINFO라는 명칭으로 레지스트리에 등록한다. 향후 Dropper 제작 시에 활용하기 위한 기능을 제작한다. ④ “PS” 명령어 전송 시 victim의 현재 프로세스 목록을 attacker 화면에 출력한다. 향후 Keylogger 제작 시 활용하기 위한 기능을 작성한다.



(그림1) 모의 백도어 구현 시스템 구성도

4.1.3 모의 백도어 코드 작성

“4.1.2 백도어 구현 시나리오”를 바탕으로 하여 모의 백도어 코드를 작성한다. 완전한 테스트를 위해서는 victim(server) - attacker(client) 간 통신기능 수행을 통해 client 측 백도어 프로그램이 준비되어야 한다. 코드는 소켓프로그램의 비교적 단순하고 정형화된 로직을 활용하여 백도어 구현 시나리오의 프로세스를 차례대로 수행할 수 있는 애플리케이션을 작성한다. 애플리케이션은 특히 윈도우 환경에서 사용할 수 있도록 윈도우 소켓 winsock2.h 버전 을 사용한다. 윈도우 소켓 활용 프로그램은 먼저 헤더파일을 사용하여 윈도우소켓 함수를 호출(#include)하여야 한다. 이 기능을 위해 헤더화일 <winsock2.h>, <stdlib.h>, <stdio.h> 호출이 필요하고 라이브러리를 링크하기 위해서는 #pragma comment(lib,″ ″)헤더화일 호출이 사용 되어야 한다. 또한 파일 전송을 위해서 버퍼사이즈 recvBuffer BUFSIZE 크기를 확보한다. 코드 내용은 에러 확인 함수, 윈속 초기화, 생성된 socket을 서버에 연결하기 전에 접속할 서버에 대한 내용을 구조체를 이용하여 기록한다. 해당 소켓의 메모리 초기화, 프로토콜 패밀리 TCP를 사용하며 소켓이 생성된다. 생성된 socket을 서버에 연결하기 전에 접속할 서버에 대한 내용을 기록한다. 사용할 포트번호, 접속할 서버의 IP 주소를 설정한다. 작성정보와 소켓을 연결하고 버퍼내용을 수신하여 retval에 저장하며 사용한 후 이를 제거한다. 본 연구는 일련의 클라이언트-서버 프로그램이 작성되었지만 전체 프로그램 공개 시 보안 문제를 고려하여 <표3>의 client 프로그램 일부만을 예시한다.

<표3> Client 프로그램 예제(일부)

```
#include<winsock2.h>
#include<stdlib.h>
#include<stdio.h>
#pragma comment(lib,"ws2_32.lib")
#define BUFSIZE 1024

void error_exit(char *msg)
{
	LPVOID lpMsgBuf;
	FormatMessage(
		FORMAT_MESSAGE_ALLOCATE_BUFFER|
		FORMAT_MESSAGE_FROM_SYSTEM,
		NULL, WSAGetLastError(),
		MAKELANGID(LANG_NEUTRAL,
SUBLANG_DEFAULT),
		(LPTSTR)&lpMsgBuf, 0, NULL);
	MessageBox(NULL, (LPCTSTR) lpMsgBuf, msg,
MB_ICONERROR);
	LocalFree(lpMsgBuf);
	exit(-1);
}
                        “중 략”

		{
			char buf2[100000];
			// recv() :
			retval = recv(tcp_sock,buf2,100000,0);
			printf("%s\n",buf2);
		}
	}

	closesocket(tcp_sock);
	WSACleanup();
	return 0;
}
```

4.1.4 백도어 실행 테스트

백도어 실행 환경설정으로서 ①attacker, victim 간 통신 기능 ②“MyBack” 이라는 명칭으로 victim의 방화벽에 특정 파일 예외 등록 ③자동 실행 레지스트리에 VMINFO라는 명칭으로 등록 ④attacker에서 “PS” 문자열 전송 시 victim의 현재 프로세스 목록을

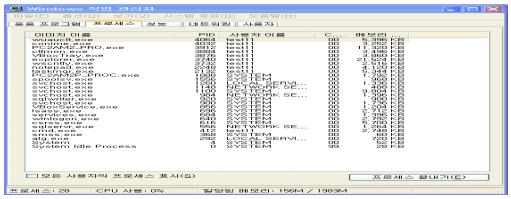
attacker의 모니터에 출력 ⑤Victim의 작업 관리자를 통해 프로세스 목록 확인절차를 순서대로 진행한다. 백도어 실행은 attacker, victim 간 통신기능 구현을 순서대로 실행해보는 것 이다. 통신기능을 실행하는 이유는 악성코드의 실행이 이루어지기 위해 관련된 노드 간의 통신이 선행되어야하기 때문이다. 백도어 실행 테스트 결과 시나리오대로 attacker에서 “PS” 문자열 송신 시 victim의 프로세스 목록을 받아와서 출력하는 것을 확인할 수 있다. 이상의 프로세스는 다음과 같이 이루어 졌다.

○ attacker, victim 간 통신 기능 확인을 위해 attacker에서 문자열을 전송하였을 때 victim의 모니터에 MessageBox 출력을 구현한다.

○ “MyBack” 이라는 명칭으로 victim의 방화벽 에 특정 파일 예외 등록한다.

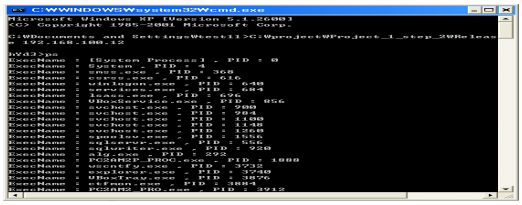
○ 자동 실행 레지스트리에 VMINFO라는 명칭 으로 등록한다.

○ attacker에서 “PS” 문자열 전송victim의 현재 프로세스 목록을 attacker의 모니터에 출력한다.



(그림2) Attacker의 모니터에 victim의 현재 프로세스 목록

○ victim의 작업 관리자를 통해 프로세스 목록을 확인한다. 시나리오대로 attacker에서 “PS” 문자열 송신 시 victim의 프로세스 목록을 받아와서 출력하는 것 을 확인할 수 있다.



(그림3) victim의 작업 관리자 프로세스 목록

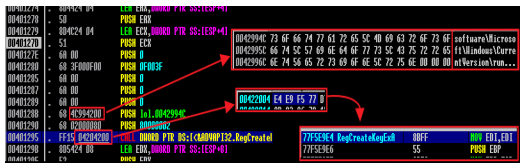
4.2 백도어 행동분석 기능 테스트 방법

백도어 행동분석 기능 테스트 도구는 Immunity Debugge를 활용한다. Immunity Debugge를 사용하여 백도어 행동분석을 실험한 이유는 이 도구는 GUI기반이며 command line을 가지고 있다. Immunity Debugger는 파이썬(Python) API를 플러그인 형태로 지원하고 있어서 이 2개를 연동 시 Reverse Engineering 환경을 구축할 수 있는데 이를 활용하면 악성코드를 분석하고, 이진파일을 반전할 수 있다. 백도어 행동분석 테스트는 주소를 인자로 넘기는 API 테스트, Temp의 경로 테스트, 레지스트리 등록 테스트, C:\a.txt를 예외 등록, 전송 문자열 비교, 준비된 루틴으로 이동, “w”와 “C:\a.txt” 인자로 함수 테스트, 주소 파일 확인, 파일내용 버퍼로 이동, 버퍼 내용 전송 등 10개 단위기능의 테스트를 실시한다.

<표4> Immunity Debugge 버전

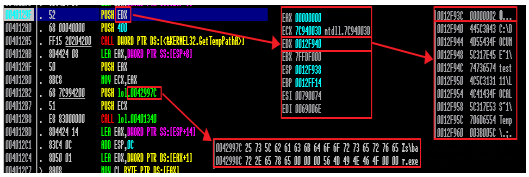
Tool	Version
Immunity Debugger	v 1.85

행동분석 기능 테스트는 Immunity Debugge v 1.85를 활용하여 백도어 행동패턴을 API 단위로 Part를 나누어 (그림4) 에서 (그림13)까지 순서대로 실행해 본다.



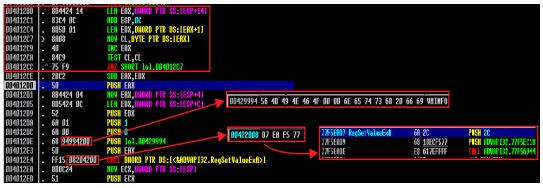
(그림4) 주소를 인자로 넘기는 API

(그림4)의 API에서 주소를 인자로 넘기는 부분을 분석해 보기 위해 첫째로 0x42994C의 주소를 추적해 보면 software\Microsoft\Windows\Current Version\run을 통해 레지스트리를 이용하고 있음을 확인할 수 있으며 둘째로 함수 호출부를 확인해 보면 Reg CreateKeyEx를 호출하므로 레지스트리 키를 생성한다는 것을 확인할 수 있다.

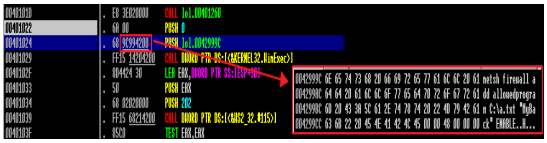


(그림5) Temp의 경로 테스트

(그림5)를 통해 Temp의 경로를 받아왔으며 그림의 하단(바로아래)에서\backdoor.exe와 합쳐져 특정 폴더를 가리키고 있고 반복문으로 문자열의 길이를 측정한 후 VNINFO라는 이름으로 Reg SetValueEx를 통해 레지스트리를 등록하는 것을 아래의 (그림6)으로 확인할 수 있다.

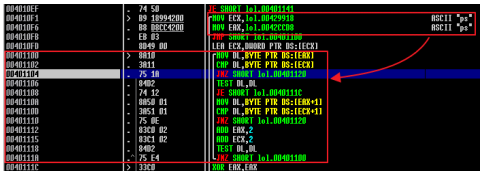


(그림6) 레지스트리 등록



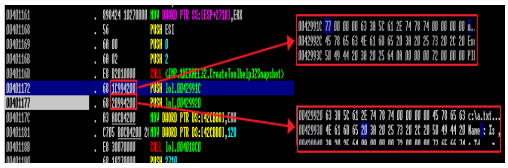
(그림7)C;\a.txt를 예외 등록

(그림7)에서는 인자로 받은 문자열을 통해 netsh 명령어를 이용하며 MyBack이라는 이름으로 C;\a.txt를 예외 등록하는 것을 확인할 수 있다.



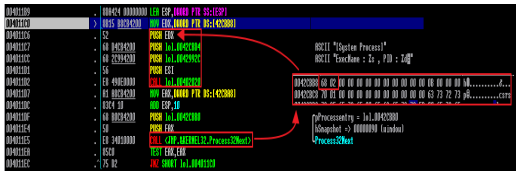
(그림8) 전송 문자열 비교

(그림8)에서 기존에 있는 문자열 ps와 recv()의 결과로 받은 문자열을 비교하여 ps가 맞으면 다음 준비된 루틴으로 이동하게 된다.



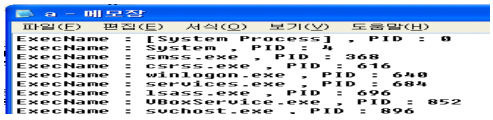
(그림9) 준비된 루틴으로 이동

문자열 “w”와 “C;\a.txt”를 인자로 갖고 함수를 호출하는 것으로 보아 fopen을 추측할 수 있다.

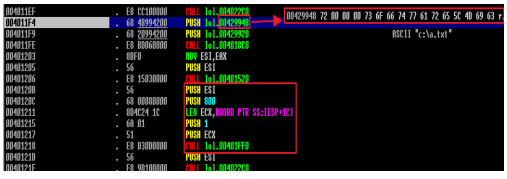


(그림10) “w”와 “C;\a.txt” 인자로 함수테스트

(그림10)의 반복문을 몇 번 수행시키면 Process 32Next의 결과 값이 0x42CBB8의 값을 변경시키고 이 값 포함 Process Name의 문자열과 출력용 문자열 그리고 마지막으로 파일 포인터를 인자를 받으며 함수를 호출하므로 해당 함수는 fprintf 이며 해당 파일에 프로세스의 목록과 PID를 입력 하는 것임을 추측할 수 있다. 이때 기존에 알게 된 주소파일을 확인해 보면 (그림11)과 같다.



(그림11) 주소 파일 확인



(그림12) 파일내용 버퍼로 이동

(그림11),(그림12)를 통해 위의 함수는 “r”로 C:\a.txt를 여는 fopen임을 알 수 있으며 아래의 함수에서는 파일 포인터와 2048 사이즈의 버퍼를 이용하

여 파일의 내용을 버퍼로 이동시키고 있음을 추측할 수 있다. 그리고 그 직 후 send()를 통해 해당 버퍼의 내용을 전송하는 것을 (그림13) 을 통해 확인할 수 있다.

(그림13) 버퍼 내용 전송

5. 결 론

백도어 방역작업에는 백도어 기본적 구조나 특성에 따라 피해영역이나 차단방법이 조금씩 다를 수 있다. 본 연구는 백도어 동작 메커니즘 진단을 기반으로 하여 backdoor 행동분석 방법 을 제안하였다. 백도어의 능동적인 침투행태는 "trapdoor" 침투점을 이용하여 보안기능을 우회 하고 데이터에 직접 접근을 시도한다. 소스코드 수정 없이도 백도어를 만드는 것과 심지어 컴파일 후에 수정하는 방법으로 백도어를 제작하는 것도 가능하다. 이와 같은 악성코드로서 백도어 의 위험성에 대응하기 위해 백도어 행동분석 방법을 도출하면 제반 악성코드의 재 침입이나 해킹 공격에 대응하는 정보를 획득할 수 있다. 백도어 행동분석 방법으로 모의 백도어 제작 환경, 백도어 구현 시나리오, 모의 백도어 코드 작성, 백도어 실행 테스트, 백도어 행동분석 과정이 필요하다. 본 논문에서는 백도어 행동 분석 방법 모델로 <표1>과 같이 행동분석 절차 4단계 설계와 행동분석 10개 단위기능 테스트 방법을 제안했다. 악성코드를 항구적으로 퇴치 할 수 있는 근본적 기술과 방법 이 개발되지 못하고 있는 환경에서 악성코드의 구조 및 감염 형태 식별과 백도어 행동양식을 분석한다면 해킹 공격을 차단할 수 있는 정보를 획득할 수 있다. 백도어 방역의 성공여부는 악성코드에 대한 지속적 연구와 대응 메커니즘 개발에 달려 있음을 본 연구를 통해 강조하고자 한다.

참고문헌

[1] Willam Stallings, “Network and Internetwork Security”.Prentice Hall, 1995.

[2] J.Hruska,“Computer Virus and Anti-virus Warfare” Ellis Horwood, 1992

[3] P.Denning“Computer Under Attack Intruders, Worms, and Virus”, Addison-Wesley,1990

[4] F.Cohen,“A short Course on Computer Viruses”. ASP Press, 1990 5 .1986.Bruce Schneier, “Applies Cryptography”, Wiley, Second Edition, 1996

[5] OWASP, CSRF Guard, http://www.owasp.org /index.php/CSRF_Guard

[6] David Gourley and Brian Totty, “HTTP: The Definitive Guide”, O'Reilly Media, 2002.

[7] SiChoon, Noh, Multi-Level Protection Building for Virus Protection Infrastructure, Springer Springer LNCS 3036,2004

[8] SiChoon, Noh, Protection Structure Building for Malicious Traffic Protecting in Intranet Systems,Springer LNCS 3981

[9] Stepen Cost, An Introduction to SQL Injection Attacks,for Oracle develops, 2007.3

[10] 류재철(충남대학교), 악성코드의 새로운 보금자리 ‘스마트폰’, 2011.06

[11] 김정녀, 스마트폰 보안 기술 동향 및 전망, 2010.11

[저 자 소 개]

나 상 엽 (SangYeob Na)

1992년 : 동국대학교
전자계산학과(학사)
1995년 : 동국대학교
컴퓨터공학과(석사)
2001년 : 동국대학교
컴퓨터공학과(박사)
2005년 : Carnegie Mellon University MSIT(Master Of Information Technology)
1996년~현 재 : 남서울대학교
컴퓨터학과 교수

email : nsy@nsu.ac.kr

노 시 춘 (SiChoon Noh)

1987년 : 고려대학교
경영정보학(석사)
2005년 : 경기대학교
정보보호기술(박사)
2002년 : KT 시스템보안부장
2004년 : KT 충청전산국장
2005년~현 재 : 남서울대학교
컴퓨터학과 교수
2011년~현 재 : 남서울대학교
IT융합연구소 연구위원

email : nsc321@nsu.ac.kr