UPnP 홈네트워크 보안 취약점에 관한 연구

(A Study for Vulnerability of Security of the UPnP Home-Networking)

오임걸*, 이종일**
(Im-Geol Oh), (Jong-Il Lee)

요 약 UPnP는 TCP/IP와 같은 인터넷 표준과 기술을 기반으로 한 SSDP, UDP와 동일한 표준 프로토콜을 사용하고 있으며, 다른 물리적 네트워킹 제품에 독립적이다 . 그러나 UPnP의 구조가 SSDP, UDP같은 프로토콜 위에서 동작하기 때문에 홈네트워킹 기술에 대한 보안 대책에서 취약점을 가지고 있다. 그러므로 본 논문에서는 UPnP 취약점을 이용한 웜 바이러스가 네트워크 기반의 모든 장비들의 공격 및 홈네트워킹 장비내의 정보를 삭제하거나 대량의 데이터를 전송하는 DoS 공격에 대해서 분석 · 보고한다.

핵심주제어 : UPnP SSDP, UDP, 웜바이러스 DoS공격,

Abstract The UPnP uses the same standard protocol as SSDP and UDP based on standard internet and technology like the TCP/IP, and is independent of other physical networking product. But the structure of the UPnP has the vulnerability to the security countermeasure for home-networking technology since it is operated on the same protocol as the SSDP and UDP. In this paper, we analyze and report against the DoS attack, where the worm virus, using the vulnerability to the UPnP, eliminates the attack of all equipments that are based on networking and eliminates the information belonging to the equipments of the home-networking or transmits the massive data.

Key Words : UPnP, SSDP, UDP, worm Virus, DoS Attack

1. 서 론

홈네트워킹은 네트워킹 기술, 기반 S/W, 그리고 정보 가전기기의 발전에 따라 급속히 확산되고 있다 . 간단히 '가정 내 정보화'라고 표현할 수 있는 홈네트워킹은 멀티PC를 갖고 있는 가정이 증가하면서 그 중요성이 대두되기 시작했다[1].

홈 네트워크는 가정 내의 인터넷 정보단말기기와 초고속 인터넷 등, 가입자 네트워크를 연결하여 데이터 송수신, 멀티미디어 제어 등의 기능을 제공한다. 홈네트워킹 표준화 경쟁은 홈네트워킹의 제어시스템을 어떤 방식으로 할 것인가를 놓고 IEEE1394[2]에 기반

* 한서대학교 인터넷공학과
** 한서대학교 대학원 정보보호공학과 박사과정

을 둔 HAVi(Home Audio-Video Interoperability)[3] 진영과 Microsoft의 OS를 기반으로 한 UPnP (Universal Plug and Play)[4][5] 진영이 치열한 경쟁을 벌이고 있다.

UPnP 는 공통의 인터페이스를 가진 UPnP가 탑재된 멀티기기들이 다른 기기들과 연결 및 통신할 수 있도록 고안된 소프트웨어 아키텍처이다. 이는 새로 기기 설정을 하거나 추가적 소프트웨어를 설치할 필요 없이 여러 정보기기가 홈 네트워크에 장착될 수 있도록 고안되었다.

TCP/IP를 포함한 HTTP 및 XML과 같은 인터넷 표준과 기술을 기반으로 한 SSDP(Simple Service Discovery Protocol)와[4], UDP같은 표준 프로토콜을 사용하고 있으며 다른 물리적 네트워킹 제품에 독립적 이다[6].

홈네트워킹은 정보가전 제품을 손쉽게 연결 사용할 수 있게 함과 동시에 사생활 침해 및 개인정보 보호 대책도 함께 마련되어야 한다. 홈네트워킹 기술에 대한 보안 대책은 홈 네트워크가 가전제품, 계측기뿐만 아니라 잠금, 감시, 경보장치 등을 유·무선으로 연결하고 있어 더욱 중요하다.

이러한 중요함에도 불구하고, UPnP의 구조가 SSDP, UDP같은 프로토콜 위에서 동작하기 때문에 여전히 보안상의 취약점을 가지고 있다.

그러므로 본 논문에서는 UPnP 취약점을 이용한 웜 바이러스가 네트워크 기반의 모든 장비들의 공격 및 홈네트워킹 장비내의 정보를 삭제하거나 대량의 데이터를 전송하는 DoS 공격에 대해서 분석 보고한다.

2. 관련연구

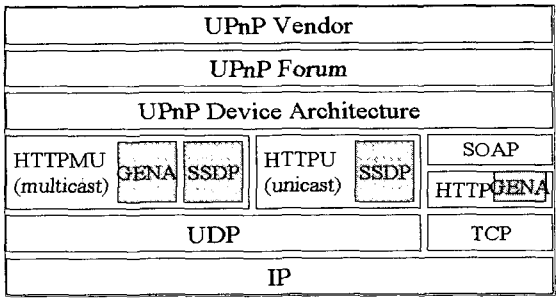
2.1 UPnP 프로토콜 구조

UPnP는 댁내망의 여러 디바이스들을 단대단(peer-to-peer) 방식으로 연결시켜 주는 미들웨어 구조이다. UPnP는 특정 운영체제나 프로그래밍 언어, 미디어와 독립적으로 네트워크상의 디바이스 간에 명령과 제어를 가능하게 한다. 댁내 망이나 SOHO 환경의 네트워크에 단순하고 유연하며 표준에 기반을 둔 연결성을 제공함으로써, UPnP는 사용자가 직접 네트워크 설정, 유지 관리를 하지 않고도 쉽게 디바이스와 서비스의 연결성을 제공한다[7].

UPnP는 TCP/IP, HTTP, XML같은 개방형 표준 프로토콜을 사용한다. 그러나 비용 요건, 기술 요건 및 레거시 시스템에 대한 지원 등 여러 가지 이유로 인하여 네트워크에 다른 기술들이 사용될 수도 있다. 여기에는 HAVi, CeBus, LonWorks, EIB 또는 X10 같은 네트워킹 기술들이 포함된다.

또한 이 기술들은 UPnP 브리지나 프록시를 통하여 UPnP 네트워크에 활용 할 수 있다.

<그림 1>은 UPnP의 프로토콜 구조를 나타낸 그림이다. SSDP는 네트워크상의 서비스를 찾기 위한 프로토콜이며, GENA(General Event Notification Architecture)는 한 디바이스의 상태가 변했을 때 이를 다른 디바이스에게 알리기 위한 프로토콜이며, SOAP(Simple Object Access Protocol)은 한 디바이스가 다른 디바이스에게 제어 명령을 보내기 위해 사용하는 프로토콜이다[8].



<그림 1> UPnP 프로토콜 스택

2.2 UPnP에 사용되는 프로토콜

2.2.1 TCP/IP

TCP/IP 네트워킹 프로토콜 스택은 나머지 모든 UPnP 프로토콜을 구축하는 기반 역할을 한다. 널리 사용되는 이 표준 TCP/IP 프로토콜을 사용함으로써 UPnP는 다른 물리적 매체를 수용하는 능력을 최대한 활용하여 다양한 공급자들이 제공하는 제품들 사이의 상호 운용성을 보장한다.

UPnP장치들은 TCP/IP 서비스(DHCP, DNS 등)뿐만 아니라 TCP, UDP IGMP, ARP, IP 등 TCP/IP 스택에 있는 많은 프로토콜을 사용한다.

2.2.2 SSDP

SSDP는 네트워크 서비스를 네트워크상에서 검색하는 방법을 정의한다. HTTPU 및 HTTPMU 기반 위에 구축되며, 제어 포인트가 네트워크 상에서 원하는 리소스를 검색하는 방법 및 장치들이 네트워크상에서 자신들이 가용상태에 있음을 알리는 방법을 정의한다. 검색 요청 및 가용성을 알리는 방법을 정의함으로써 SSDP는 이 두 가지 방법 중에서 하나만 사용할 경우에 요구되는 오버헤드를 없애준다. 그 결과, 네트워크상의 모든 제어 포인트는 네트워크 트래픽을 많이 발생시키지 않으면서도 네트워크 상태에 관한 정보를 완벽하게 파악하게 된다[4].

제어 포인트 및 장치 모두가 SSDP를 사용한다. UPnP 제어 포인트는 부팅이 되자마자 SSDP 검색 요청(HTTPMU을 사용함)을 보내서 네트워크에서 활용 가능한 장치와 서비스를 검색한다. 제어 포인트는 검색 결과 데이터를 정리하여 특정 서비스 또는 특정 장치만을 원하는 대로 선별할 수도 있다. UPnP 장치는 멀티캐스트 포트 정보를 수신한다. 검색 요청을 수신하자마자 장치는 일치 여부를 확인하기 위하여 검색 조건을 점검한다. 만약 일치된 것이 발견되면 유니캐스트 SSDP(HTTPU를 사용) 응답이 포인트로 전송된다.

이와 유사하게, 장치는 네트워크에 연결되자마자 지원하는 서비스의 사용 여부를 알리기 위하여 여러 개의 SSDP 가용상태 알림정보(Presence Announcements)를 전송한다.

가용상태 알림정보 및 유니캐스트 장치 응답 메시지는 모두 장치 설명서의 위치 포인터를 포함하며, 여기에는 이 장치의 기본정보 및 제공 서비스에 관한 정보가 들어 있다.

그 외에도 SSDP는 장치 및 장치 관련 서비스가 네트워크와의 연결을 원활하게 끊는 방법을 포함하고 있으며, 또한 자체적인 문제 해결을 위하여 유해 정보를 정화하는데 사용되는 캐시 타임아웃(cache timeouts)을 포함하고 있다.

3. 웜 공격에 대한 UPnP 취약점 조사 분석

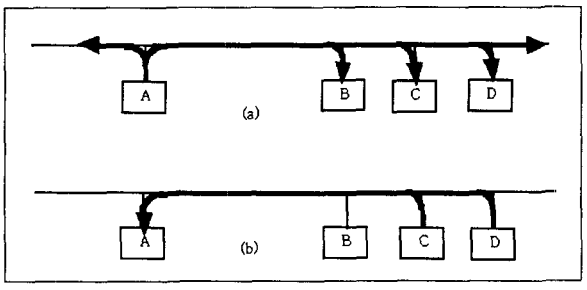
3.1 장치의 주소지정

새로운 장치가 해야 할 첫 번째 작업은 네트워크에 참여하기 위한 주소를 확보하는 것이다. 각각의 장치는 DHCP 클라이언트를 가지고 있어, 장치가 네트워크에 처음 연결됐을 때 DHCP 서버를 검색한다. 만약 장치의 DHCP 클라이언트가 서버로부터 응답을 받지 못하면, 서버가 응답을 할 수 있는지 확인하기 위하여 다시 시도한다. 네트워크가 DHCP 서버를 운영하지 않는다면, 장치는 알맞은 주소를 선택하기 위하여 자동 IP 주소지정(Auto-IP) 기능을 사용한다.

Auto-IP 기능을 사용하여 장치는 169.254/16 범위내에서 IP 주소를 자동으로 선택한다. 이 범위내의 처음과 마지막 256 주소들은 계속 유지되어야 하며 사용되어서는 안 된다. 주소가 선택된 다음에는 이미 그 주소가 사용 중인지 아닌지를 검사한다. 만약 주소가 다른 장치용으로 사용 중이면, 다른 주소를 선택하여 테스트한다.

네트워크가 사용 가능한 DHCP 서버를 가지고 있다면 이러한 모든 절차를 완료하는데 1 초도 걸리지 않는다. 그러나 가지고 있지 않다면 Auto-IP 기능을 사용할 수 있는 장치가 필요하고 절차도 조금 오래 걸린다. Auto-IP 기능을 사용하여 주소를 지정하면 장치는 다른 장치들과의 연결을 계속 유지하기 위하여, 네트워크상의 DHCP 서버가 사용 가능한지 정기적으로 검사한다.[4]

이 때, 장치는 DHCP 서버에 의하여 할당된 주소를 받든지(네트워크의 모든 기타 장치들은 동일 서브넷에 주소를 가짐) 아니 면 Auto-IP 주소를 가지게 된다. 어떤 경우에도 장치는 TCP/IP를 사용하여 네트워크상의 다른 장치들과 통신할 수 있다.



<그림 2> 장치의 주소지정

<그림 2>의 (a) : 기계 A는 목적지를 자신으로 표시한 주소 요청을 브로드캐스트한다.

<그림 2>의 (b) : 주소를 제공하기로 된 기계(C,D)들이 직접 A에게 응답한다.

장치가 적합한 네트워크용 IP 주소를 가지게 되면, 그 주소를 통해서 네트워크상에서 검색 및 조회가 가능하다. 이 경우에는 IP 주소보다는 편한 이름으로 장치를 설정하는 것이 사용하기에 훨씬 용이하다. 그러나 이름을 사용하여 매핑 주소를 지정하기 위해서 DNS를 사용하는 것은 UPnP 기능범주에 속하지 않는다.

3.2 UPnP 보안 취약점

3.2.1 원격제어

UPnP에 존재하는 첫 번째 취약점은 해커들의 remote SYSTEM level로의 접속할 수 있는 최상위 레벨이다. UPnP에 조작된 광고를 다양한 속도로 전송하면 공격 대상 시스템에 접근규약 위반이 발생한다. 아래의 예를 참조하면 프로토콜, 포트, URL의

```
NOTIFY * HTTP/1.1
HOST: 239.255.255.250:1900
CACHE-CONTROL: max-age=10LOCATION:
http://IPADDRESS:PORT/<buffer>.xml
NT:urn:schemas-UPnP-org:device:InternetGatewayDev
ice:1
NTS:ssdp:alive
SERVER:EEYE/2001 UPnP/1.0 product/1.1
USN:uuid:EEYE
```

uri 필드의 버퍼가 증가하면서 10,000ms의 주기로 세션을 전송하면 접근 규약 위반이 일어나게 되는 것을 볼 수 있다.

3.2.2 The DoS and DDoS Attack

UPnP 기능이 있는 장치가 네트워크상에 설치가 되면, 컴퓨터, 네트워크 디바이스, 가전제품 등은 자신의 종류를 구분하기 위한 콘트롤 위치를 확인하기 위한 통지를 보내게 된다. XP의 기본설치 시에는 디바이스 컨트롤이 설치되지 않는다.

MS가 "InternetGatewayDevice"를 기본적으로 지원함에도 불구하고 앞의 네트워크에서 스니퍼가 돌고 있다면 감지가 된다. 이 기능의 지원목적은 UPnP 지원 기능의 "gateway devices"를 제조하는 네트워크 하드웨어 개발자들을 돕기 위해 지원이 되었다.

SSDP 통고가 들어 있는 위장된 UDP 패킷을 보냄으로써 XP/ME 클라이언트를 특정한 주소로 접속을 유도 할 수 있으며 HTTP/HTTPS 요청을 보낼 수 있다.

```
NOTIFY * HTTP/1.1
HOST: 239.255.255.250:1900
CACHE-CONTROL: max-age=1
LOCATION:URL
NT:urn:schemas-UPnP-org:device:InternetGatewayDev
ice:1
NTS:ssdp:alive
SERVER:EEYE/2001 UPnP/1.0 product/1.1
USN:uuid:EEYE
```

위의 패킷 데이터는 XP/ME 시스템의 1900 포트로 UDP 패킷이 전송되어야 한다. XP 시스템이 이러한 요청(request)을 받게 되면 LOCATION header entity에 나오는 URL을 번역하게 된다. 이 URL이 그대로 Windows Internet Services API의 함수들로 전달된다. 결국 문자열은 소멸되고 새로운 세션이 생성하게 된다.

이 방법으로 해커들은 원격에서 XP사용자들을 유도하여 read/malloc의 루프로 빠뜨려 CPU의 자원을 100% 소모시키게 할 수가 있으며 고갈 시킬 수 있는 포인트로 할당 시킬 수가 있다. 이 공격을 당하면 사용자들은 물리적으로 시스템을 꺼야한다. 이 공격법을 다른 XP 시스템을 조정하는 용도에도 사용한다. 유니코드 공격으로 유도할 수가 있으며 더블 디코드(double decode), 랜덤 CGI 공격 등에 이용할 수가 있다.

또 DDoS 공격이 가능하게 된 이유는 SSDP 통보가 브로드캐스트 주소와 멀티캐스트로 보내질 수가 있기 때문에 결과적으로 하나의 UDP 패킷이 전체 시스템으로 보내지게 되는 것이다. 또한 UPnP 서비스 중의 일부가 UDP로서 구현이 되었기 때문에 이러한 공격이 완벽하게 이루어 질수가 있고 추적도 불가능하게 된다.

3.3 SSDP의 문제점

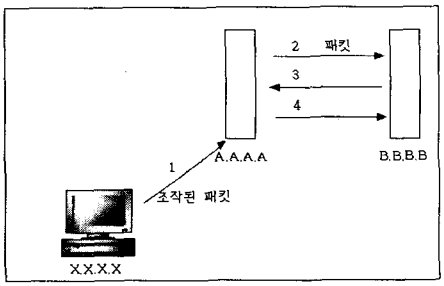
3.3.1 네트워크 트래픽 증가

네트워크의 트래픽을 증가시켜 실제로 시스템의 사용에는 문제가 없지만 이 시스템이 속한 네트워크

를 마비시킴으로써 시스템의 네트워크 서비스를 올바르게 수행하지 못하게 하는 것이다. 예를 들어 시스템의 서비스 중 echo 서비스는 이름 그대로 7번 포트에 데이터를 보내면 그 보낸 데이터를 다시 보내어 주는 기능을 하는 서비스다. 즉 만일 해커가 패킷을 조작함에 있어서 출발지의 주소를 B.B.B.B. 포트를 7번으로 하고 목적지의 주소를 A.A.A.A.라는 포트를 7번으로 하여 보낸다면, 보내어진 패킷은 먼저 A.A.A.A, 라는 echo 서비스에게 보내어지게 되지만, 출발지의 주소와 포트가 B.B.B.B.의 7번 포트로 이 패킷을 되돌려 주게 된다. 이때 되돌려지는 패킷의 출발지와 포트는 A.A.A.A와 7번으로 설정되어 있으므로 결국 이 패킷은 A.A.A.A와 B.B.B.B 호스트를 계속 왔다 갔다 하게 된다. 만일 해커가 이러한 패킷을 하나가 아니라 계속해서 보내게 되면 엄청나게 많은 패킷들이 이 호스트 사이를 계속 왔다 갔다 하게 된다. 만일 해커가 이러한 패킷을 하나가 아니라 계속해서 보내게 되면 엄청나게 많은 패킷들이 이 호스트 사이를 계속해서 왕복하게 되어 네트워크 부하는 기하급수적으로 증가하게 된다. 이 공격에서 중요한 점은 통신을 위해 두 호스트가 연결을 맺을 필요가 없다는 것이다. 그러므로 해커는 A.A.A.A에게 패킷을 보내면서 자기가 B.B.B.B라고 쉽게 속일 수 있다.

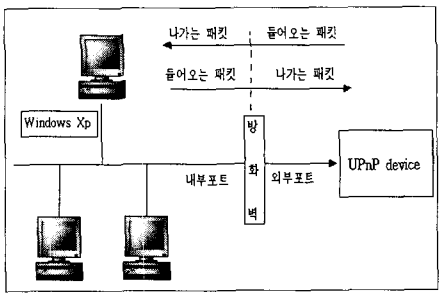
3.3.2 SSDP 프로토콜 보안 취약점

내부 네트워크로 들어오고 나가는 모든 트래픽은 방화벽을 통과한다. 대부분의 방화벽은 필터링을 위한 옵션을 제공한다. 옵션의 실행으로 방화벽은 방화벽뿐만 아니라 필터가 된다. 필터는 몇몇 데이터 그



<그림 3> 네트워크 트래픽 공격

램은 방화벽을 통해 통과시키고, 다른 데이터 그램은 여과시킨다. 간단한 예로 필터는 모든 UDP 세그먼트와 모든 텔넷 연결을 막도록 설정될 수 있다. 이러한 설정은 외부의 텔넷을 이용한 내부 호스트로의 로깅, 내부자의 텔넷을 이용한 외부 호스트로의 로깅, 그리고 이상한 UDP트래픽이 내부 네트워크로 들어오거나 나가는 것을 막는다. 방화벽은 IP 프로토콜 필드가 UDP 에 일치하는 모든 데이터 그램을 막음으로써 UDP 트래픽을 여과한다.

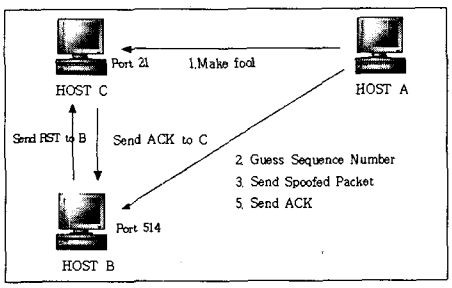


<그림 4> 방화벽의 위치 선정

내부근원지 IP주소 즉, 내부 호스트에서 들어오기 위해 요청하지만 실제로는 외부로부터 오는 패킷을 갖는 모든 데이터 그램을 거부하는 것은 매우 권장할 만하다. 이러한 패킷들은 흔히 공격자가 내부기계에서 오는 것처럼 하는 주소를 속이는 공격의 일부이다.

TCP/IP 프로토콜은 구현시의 정확성에도 불구하고 그 설계의 결점으로 인하여 보안상에 큰 취약점을 가지고 있다. 호스트에 대 한 인증을 IP의 출발지 주소만으로 수행한다. 공격할 대상의 IP주소와 포트번호는 알려져 있고 출발지의 호스트의 포트는 Ephemeral 포트번호를 사용하므로 출발지호스트의 IP 주소만을 속일 수 있다면 다른 호스트에 연결을 맺을 수 있다.

<그림 5>는 외부의 호스트 A에서 내부의 호스트 B로의 주소 속임을 이용한 연결과정을 보여준다.



<그림 5> 주소 속임을 이용한 연결 과정

1. A -> C : SYN(ISN) : 호스트 A는 호스트 C의 포트 21번에 계속 보냄으로써 호스트 C의 포트 21번을 마비(Flood)시킨다.
2. A -> B : SYN(ISN) : 호스트 A는 호스트 B에서 발생시키는 순서번호를 알아내기 위해 호스트 B의 514번 연결 요청을 여러 번 보낸다. 순서번호는 초당 128,000씩 증가하므로 패킷의 왕복 시간(Round Trip Time)만 알아내면 다음 순서번호를 알아낼 수 있다.
3. A -> B : SYN(ISN), SRC=C : 호스트 A는 호스트 B에 자기가 호스트 C인 것처럼 가장하여 연결 요청을 한다. 즉, 자신의 출발지 IP주소 항목에 호스트 C의 IP주소를 써넣은 패킷을 만들어 호스트 B로 보낸다.
4. B -> C : SYN(ISN), ACK(ISN) : 호스트 B는 들어온 연결 요청이 호스트 C로부터 온 것인 줄 알고 호스트 C에 ACK를 보낸다. 그런데 호스트 C의 포트는 이미 마비되었으므로 대답할 수 없다. 호스트 C의 포트가 마비되지 않았다면 자신이 호스트 B에 연결 요청을 한 적이 없으므로 RST 패킷을 보내어 연결이 이루어지지 않는다.
5. A -> B : ACK(ISN), SRC=C : 이 때 호스트 A가 호스트 C인 것처럼 가장하여 호스트 B의 SYN에 대한 ACK를 보낸다. 이렇게 하여 호스트 A와 호스트 B와의 연결이 이루어진다[9].

이와 같이 공격자가 프로토콜의 보안 취약점을 이용해 특정 호스트에 위조된 연결 설정 세그먼트를 보내 내부 네트워크로의 연결이 가능하다.

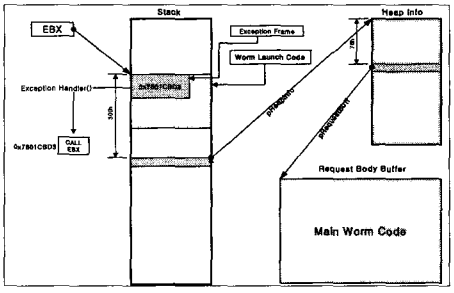
UPnP의 모든 측면은 HTTP 및 이의 파생 프로토콜 기반 위에 구축되며 HTTPU(HTTPMU 포함)는 HTTP 파생 프로토콜로서 TCP/IP가 아닌 UDP/IP를 기반으로 한 메시지 전달을 정의 한다. SSDP는 이러한 HTTPU 및 HTTPMU 기반 위에서 구축되었다. 따라서 UPnP는 TCP/IP, UDP가 갖는 보안 취약점을 SSDP 프로토콜의 사용으로 웜 바이러스 및 DoS등 다양한 공격이 가능하다.

3.4 웜 바이러스 공격

시스템에서 프로그램이 실행되어지고 있다는 의미는 그 프로세스가 이용하고 있는 메모리 영역이 존재한다는 뜻이며 이 영역은 원칙적으로 보호되는 구역과 보호되지 않는 구역으로 구분된다. 문제는 이 보호되지 않는 영역이 존재하고, 이 때문에 이 영역을 잘 활용하면 프로그램이 원래의 목적을 벗어난 이상 동작을 할 수 있다는 것이다.

Stack은 함수를 부를 때 ret(rn Address)등을 Push하여 스택에 저장하고, 이 후 함수가 끝났을 경우 위의 값들을 Pop하고 ret로 돌아가게 된다. main함수에서 function을 호출 하면 스택에 ret가 저장되고 sfp(stack frame pointer)가 저장된다. fp(frame pointer)는 실행중인 함수의 위치를 나타내고 sp(stack pointer)는 메모리의 끝을 나타낸다.

여기에서 경계를 명확하게 점검하지 않는 시스템 함수가 불리워질 때 여기에 버퍼의 크기보다 훨씬 큰 데이터를 넣어서 ret를 프로그램이 원하는 위치로 변경하는 것이다.



<그림 6> 웜 바이러스 구조

<그림 6>의 디코드는 Ox7801CBD3의 DWORD와 같다. 예외주소취급자는 DWORD인 Ox7801CBD3를 설치하고, EBX와 함께 첫 번째 DWORD인 CALL ECX에 포인팅한다.

예외 조정자를 부를 때, 그것은 CALL EBX를 요청하고 예외 블록 위에 쓰여진 첫 번째 바이트로 제어를 전환하고 코드를 해석할 때 버퍼내의 메인 웜 코드로 제어를 바꾼다. 이 코드는 EBX 의 값으로부터 Ox300바이트의 스택(pHeapinfo)을 가리킨다. pHeapinfo는 요청자의 본체에 GET를 포함하는 힙 버퍼 (pRequestBuff)를 포인터하고 메인 웜 코드를 포함한다[10].

스택에 모아진 제어 결과는 메인 웜 코드를 실행하는 코드를 집어넣음으로서 프로그램의 실행과 함께 메인 웜 코드를 수행 한다.

웜에 감염된 경우 인터넷 상의 임의의 주소를 선택하여 감염을 시도하기 때문에 네트워크의 트래픽이 증가하게 되며, 이로 인해 네트워크의 속도 저하 및 시스템 성능 저하를 초래하게 된다. 또한 이와 같이 웜이 무작위로 인터넷에 연결된 시스템에 연결을 시도함으로 인해 인터넷에 연결된 시스템은 '서비스 거

부 공격(DoS)'에 노출되는 것과 같은 위험에 놓인다.

4. 결 론

본 논문에서는 UPnP 취약점을 이용한 웜 바이러스의 공격에 대하여 알아보았다.

Peer-To-Peer 방식의 UPnP는 네트워크 연결을 통해 지능형제품, 무선 디바이스 등을 제공하기 위해 설계되었다. 또한 집이나 사무실 기타의 장소에서 근접통신이 가능하고, 데이터 전송을 용이하도록 하기 위해 TCP/IP, HTTP 및 XML과 같은 인터넷 기술을 기반으로 하는 SSDP, UDP 같은 프로토콜을 사용하여 여러 장치들이 서로 자동 연결되고, 더욱 많은 사람들이 네트워킹을 통해 함께 사용할 수 있다.

보안문제가 인터넷 사회에 끼치는 영향이 커짐에 따라 현재의 보안모델의 취약점을 공격하려는 시도가 많아지고 있다. 최근에는 이러한 전통적인 공격 형태를 이용하여 다른 시스템을 공격한 후에 스스로 다른 취약점이 있는 시스템들을 탐지하는 인터넷 웜(Internet Worm) 공격이 등장하였으며, 이러한 Internet Worm은 점점 더 다기능을 갖고 복잡해지고 있다. 뿐만 아니라 특정시스템의 취약점이 공개되면, 이 취약점을 이용한 Internet Worm이 수주일 또는 수개월 내에 등장하여 많은 시스템 및 네트워크에 피해를 주고 있다. 이러한 웜의 특성을 알고 미리 대비하는 보안의식이 필요하다. 끝으로 본 연구가 가진 제한점을 극복하고 계속 새로 발전하는 웜 바이러스의 활동 모형 분석과 이에 대한 연구가 계속되어야 할 것이다.

참 고 문 헌

[1] 국가정보화 백서, 한국전산원 2001.

[2] http://www.1394ta.org/Technology/About/1394a_and_b_whitepaperrevised.PDF

[3] http://www.havi.org

[4] http://www.microsoft.com/korea/windowsxp/pro/techinfo/planning/upnp/protocol.asp

[5] http://www.upnp.org

[6] 정보통신산업동향, 정보통신정책연구원, 2001.

[7] 조충래, 박광로, "UPnP 기술 표준화 현황", 주간 기술동향 통권 1075호, 2002년 12월.

[8] A.M. Brent, N. Toby, T. Charlie, and D.W. Mark, "Home networking with Universal Plug and Play," IEEE Communications Magazine, Vol.39, No.2, Dec. 2001, pp.104-109.

[9] PLUS 저, Security PLUS for Unix, 영진.com, 2001

[10] Bruce McCorkendale & Peter Szor, "Code Red Buffer Overflow," Symantec, Virus Bulletin, 2001.9

오 임 걸 (Im-Geol Oh)

- 정회원
- 1983년 2월 : 인하대학교 수학과 (이학사)
- 1986년 2월 : 인하대학교 수학과 응용수학전공(이학석사)
- 1993년 8월 : 인하대학교 통계학과 (이학박사)
- 2000년 8월 ~ 현재 : 인하대학교 컴퓨터공학과 박사과정
- 1995년 3월 ~ 현재 : 한서대학교 인터넷공학과 부교수
- 관심분야 : 컴퓨터 보안, 암호학, 컴퓨터 통신

이 종 일 (Jong-Il Lee)

- 정회원
- 2000년 2월 : 한서대학교 물리학과 (이학사)
- 2002년 8월 : 한서대학교 대학원 정보보호공학과 (공학석사)
- 2003년 2월 ~현재 : 한서대학교 대학원 정보보호공학과 박사과정
- 2002년 8월 ~ 2003년 12월 : 한서대학교 인터넷공학과 강사
- 관심분야 : 정보보호, 암호학, 무선 인터넷 보안