

[Full-disclosure] CAN-2004-1073 not fixed

Source: <http://www.derkeiler.com/Mailing-Lists/Full-Disclosure/2005-03/0895.html>

From: Santosh Eraniase (santosh.e_at_ap.sony.com)

Date: 03/29/05

Date: Tue, 29 Mar 2005 16:39:48 +0530
To: full-disclosure@lists.grok.org.uk

Hi,

There was a vulnerability disclosure on Nov 10, 2004
http://isec.pl/vulnerabilities/isec-0017-binfmt_elf.txt

These issues have been given the following candidate status,
CAN-2004-1070
CAN-2004-1071
CAN-2004-1072
CAN-2004-1073

Various vendors have released a patch which is similar to
the following bkbites patch:

["http://linux.bkbites.net:8080/linux-2.4/cset@1.1448.1.49?nav=index.html|src|src/fs/related/fs/binfmt_elf.c"](http://linux.bkbites.net:8080/linux-2.4/cset@1.1448.1.49?nav=index.html|src|src/fs/related/fs/binfmt_elf.c)
["http://linux.bkbites.net:8080/linux-2.4/cset@1.1448.69.10?nav=index.html|src|src/fs/related/fs/binfmt_elf.c"](http://linux.bkbites.net:8080/linux-2.4/cset@1.1448.69.10?nav=index.html|src|src/fs/related/fs/binfmt_elf.c)

The vendor advisorie's claim that all the CAN's listed above is fixed.
But on checking the patch, we notice the following:

The vulnerability disclosure and the CAN-2004-1073 report
the vulnerability to be in open_exec function in exec.c file.

```
---
5) bug in the common execve() code in exec.c: vulnerability in
open_exec() permitting reading of non-readable ELF binaries, which can
be triggered by requesting the file in the ELF PT_INTERP section:
541:     interpreter = open_exec(elf_interpreter);
        retval = PTR_ERR(interpreter);
        if (IS_ERR(interpreter))
            goto out_free_interp;
        retval = kernel_read(interpreter, 0, bprm->buf,
BINPRM_BUF_SIZE);
-----
We do not see any patch to exec.c. Instead there is a check on the
return value of the kernel read.
--
@@ -616,8 +628,11 @@
        if (IS_ERR(interpreter))
            goto out_free_interp;
        retval = kernel_read(interpreter, 0, bprm->buf,
BINPRM_BUF_SIZE);
```

Full-Disclosure: [Full-disclosure] CAN-2004-1073 not fixed

```
-         if (retval < 0)
+         if (retval != BINPRM_BUF_SIZE) {
+             if (retval >= 0)
+                 retval = -EIO;
+                 goto out_free_dentry;
+         }
+         /* Get the exec headers */
+         loc->interp_ex = *((struct exec *) bprm->buf);
```

Checking the return value seems to be a fix for the CAN-2004-1070 issue. But the bkbits patch makes no such claim of fixing the open_exec related vulnerability. It says the following are fixed

```
# Make sure kernel reads full size of elf data. Error out if mmap
# fails when mapping any sections of the executable. Make sure
# interpreter string is NULL terminated.
```

```
#
```

These statements correspond to CAN-2004-1070/1071/1072

On executing the PoC given in the disclosure document, as user on 2.4.20 x86 kernel, we get a coredump. The coredump contains the suid executable we have given as an argument to the PoC. The executable has the following permission

```
-rws--x--x  1 root      root          28628 Jan 25  2003 /bin/ping
```

Even though there is no read permission for group and others we can still read the executable using the PoC.

On executing the PoC on 2.4.29 and 2.6.11 kernel we initially get no core dump. The following code introduced to fix another bug, caused the loading of the interpreter to fail.

```
# ChangeSet
```

```
# 2004/11/23 08:01:10-02:00 barryn@pobox.com
# [PATCH] binfmt_elf.c fix for 32-bit apps with large bss
```

```
#
```

```
# This is a 2.4.27-2.4.28 port of this patch:
```

```
#
```

```
# > [PATCH] binfmt_elf.c fix for 32-bit apps with large bss
```

```
# >
```

```
diff -Nru a/fs/binfmt_elf.c b/fs/binfmt_elf.c
--- a/fs/binfmt_elf.c 2005-01-06 10:17:32 -08:00
+++ b/fs/binfmt_elf.c 2005-01-06 10:17:32 -08:00
```

```
.....
```

```
@@ -727,6 +739,19 @@
```

```
        k = elf_ppnt->p_vaddr;
        if (k < start_code) start_code = k;
        if (start_data < k) start_data = k;

+
+         /*
+          * Check to see if the section's size will overflow the
+          * allowed task size. Note that p_filesz must always be
+          * <= p_memsz so it is only necessary to check p_memsz.
+          */
+         if (k > TASK_SIZE || elf_ppnt->p_filesz >
elf_ppnt->p_memsz ||
+         elf_ppnt->p_memsz > TASK_SIZE ||
+         TASK_SIZE - elf_ppnt->p_memsz < k) {
+             /* set_brk can never work. Avoid overflows. */
+             send_sig(SIGKILL, current, 0);
+             goto out_free_dentry;
+         }
+     }
```

The condition `elf_ppnt->p_filesz > elf_ppnt->p_memsz` fails.

So we modified the PoC with the following patch

```
--- poc73.c.bak 2005-03-22 17:35:45.000000000 -0500
+++ poc73.c     2005-03-24 12:47:17.000000000 -0500
@@ -135,6 +135,7 @@
```

Full-Disclosure: [Full-disclosure] CAN-2004-1073 not fixed

```
    eph.p_type = PT_LOAD;
    eph.p_offset = 4096;
    eph.p_filesz = 4096;
+   eph.p_memsz = 4097;
    eph.p_vaddr = 0x0000;
    eph.p_flags = PF_R|PF_X;
    write(fd, &eph, sizeof(eph) );
```

With this change, on executing the PoC on 2.4.29 and 2.6.11, the core dump contains the suid executable.

We used the strings command to check if the strings in the suid is present in the core.

So we find that the vulnerability of reading non-readable binaries exist in the latest kernel and the vendor provided patch for CAN-2004-1073 does not fix this vulnerability.

Please let me know your opinion.

Regards,

--

Santosh Eraniouse

Full-Disclosure - We believe in it.

Charter: <http://lists.grok.org.uk/full-disclosure-charter.html>

Hosted and sponsored by Secunia - <http://secunia.com/>