Financira
Evropska unija
NextGenerationEU

THE RECOVERY
AND RESILIENCE
PLAN

VARNOST PROGRAMOV

Predavanja #10

Matevž Pesek

Od prejšnjič

- Kakšne tipe peskovnikov poznamo?
- Kaj so sistemski klici?
- Kaj je namen virtualizacije?

DANAŠNJE TEME

- Raz-zbiranje
- Razhroščevanje
- Fuzzing*

RAZ-ZBIRANJE

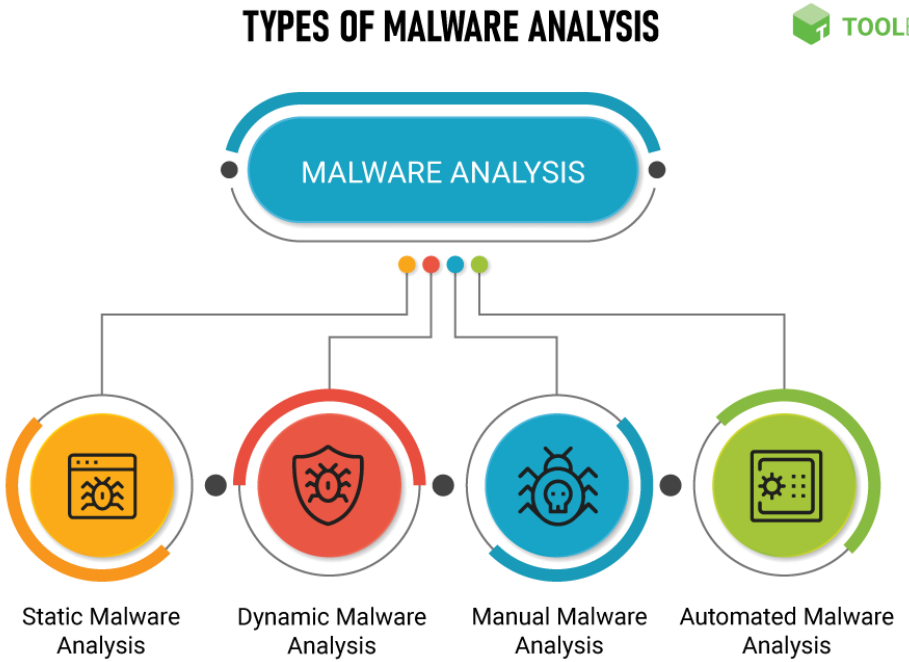
disassembling

Motivacija

- Predhodni koraki
  - Do sedaj ...
- Kaj naredimo, ko gre enkrat vse narobe?
  - Malware, virusi, trojanski konji, ...

Kako se borimo?

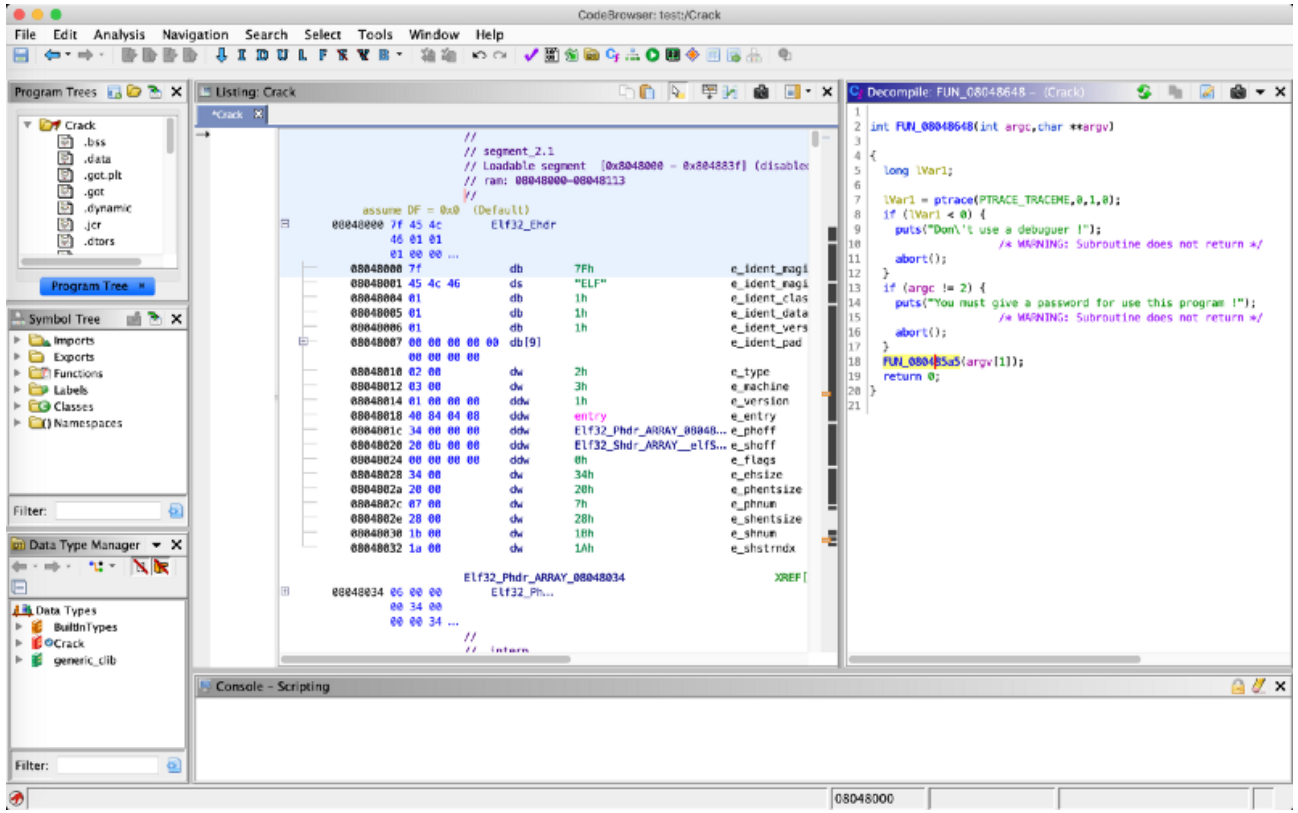
- Aktivno:
  - Antivirusni program
  - Požarni zid
- Pasivno:
  - Forenzika
  - Analiza zlonamerne kode
    - Testiranje in sledenje izvajanja v peskovniku
    - Raz-zbiranje kode in statična analiza



https://www.spiceworks.com/it-security/data-security/articles/what-is-malware-analysis-definition-types-stages-best-practices/#lg=1&slide=0

Raz-zbiranje

- Orodja:
  - dogbolt.org
  - Ghidra
  - IDA
- Vedno dobimo nazaj pomanjkljivo kodo
- Gledamo delovanje programa
  - Kako se izvaja
  - Kako se širi
  - Ali obstaja ”killswitch”
  - ...



PRIMERI

Disassembling

ArcaneDoor

- CVE-2024-20359: Line Runner in Line Dancer
  - Izvede lua skripto ob vklopu
  - Lua skripta se naloži zlonamerna koda
  - Ob pravem paketu se požene poljubna koda
  - Ostane samo v spominu
- CVE-2024-20353
  - DOS ob napačnih HTTP glavi
  - Ponovno zažene strežnik

```
{
long payload;
int iVar2;
uint len;
void *decoded_payload;
if (ip_pak != 0) {
payload = ip_pak + 0x20; iVar2 =
memcmp(s_[illegible]_55824e200200,ip_pak,0x2
0); if (iVar2 == 0) {
len = __wrap_strlen(payload); decoded_payload = malloc(len);
if (decoded_payload != (void *)0x0) {
base64_decode(payload,decoded_payload);
memcpy(CUS_shellcode_payload,decoded_payload,(ulong)len);
free(decoded_payload);
CUS_shellcode_payload();
} }
} processHostScanReply(param_1); return;
}
```

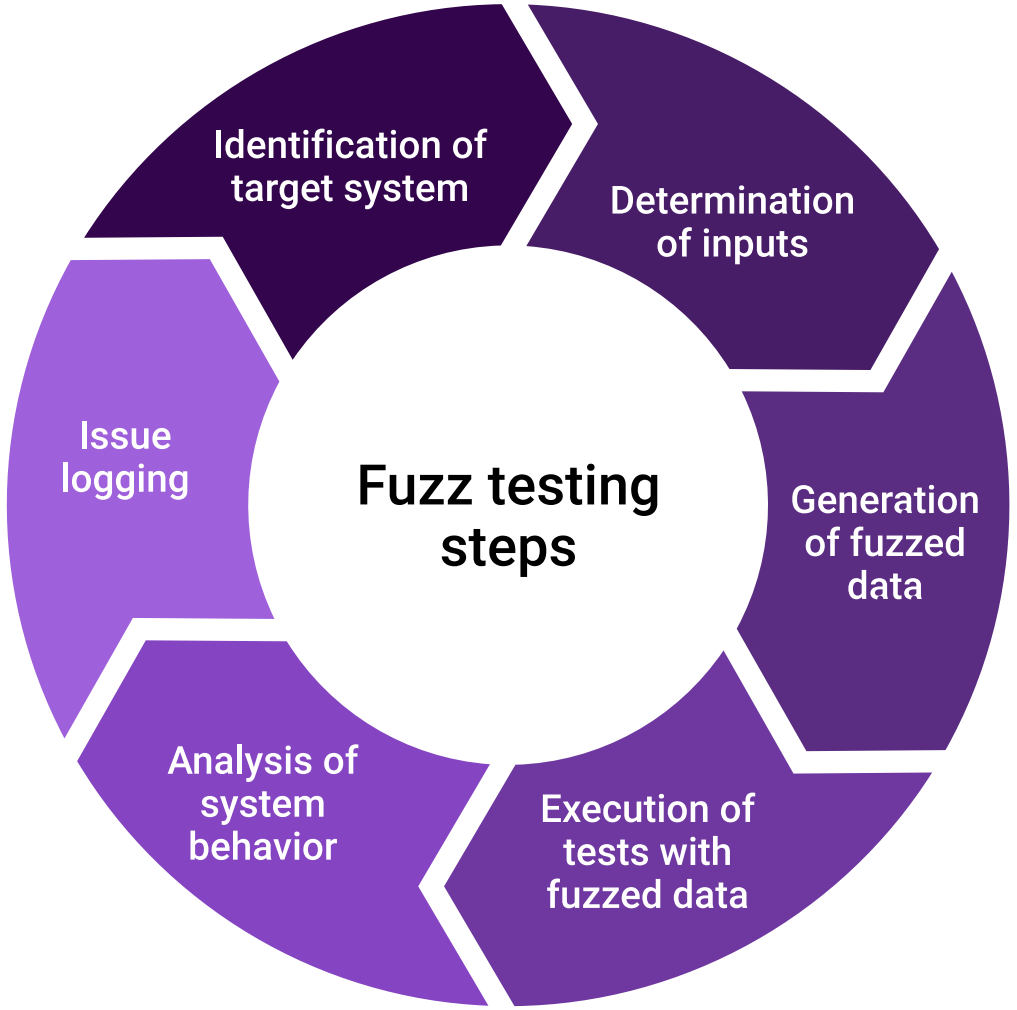
https://blog.talosintelligence.com/arcanedoor-new-espionage-focused-campaign-found-targeting-perimeter-network-devices/

FUZZING

Kako najdemo napake?

- Statična analiza kode in ročno preverjanje programa
- Ko pridemo do napake, odpremo program v razhroščevalniku in preverimo kaj je šlo narobe

- Ali lahko to kako automatiziramo?

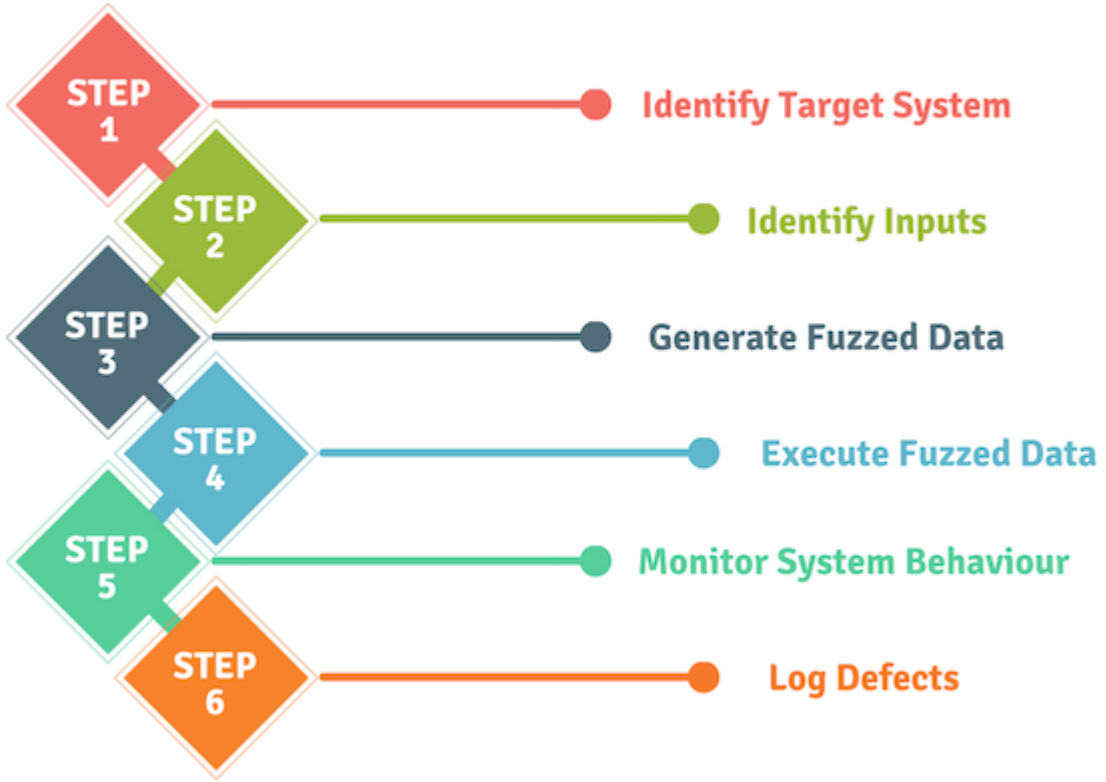


https://www.synopsys.com/glossary/what-is-fuzz-testing.html

Fuzzing

- “Fuzzing“ je tehnika automatiziranega testiranja
- Automatizirano iskanje vhodov v program, ki sledijo do napake
- Napake lahko nato analiziramo glede na vhode

- AFL++ [primer levo]

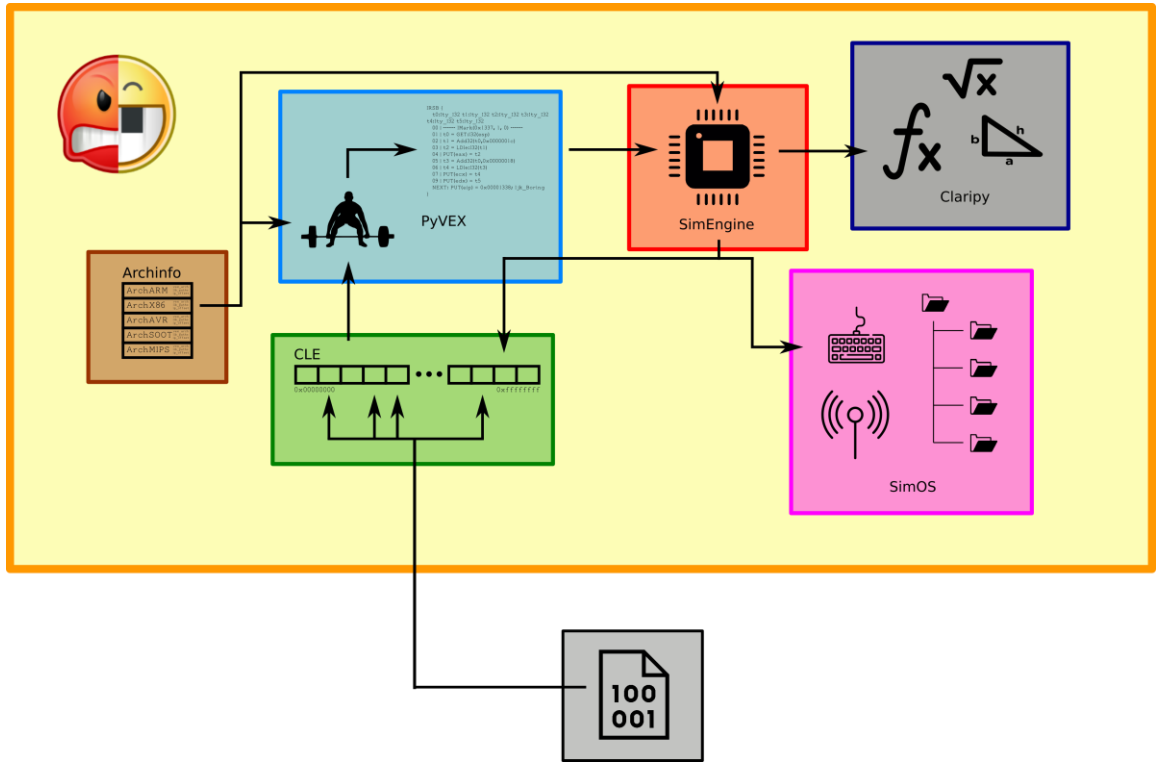


https://cyberhoot.com/cybrary/application-fuzzing/

Analiza binarnih programov

- Namesto naključnih vhodov analiziramo program in zgeneriramo vhod
- Rekonstrukcija grafa izvajanja
- Automatizirano tudi iskanje iz izkoriščanje varnostnih lukenj

- angr [primer levo]



https://angr.io/blog/throwing_a_tantrum_part_1/

PRIMERI

Fuzzing

Grub2 “back to 28”

- CVE-2015-8370
- GRUB vpis uporabniškega imena in gesla
- Zaradi slabega preverjanja dolžine se lahko ponovno zažene računalnik
- Med 17 in 20 vnosov se program začne izvajati od 0-tega naslova
- IVT lahko prepišemo
- Točno 28 vnosov se prepiše IVT tako, da zaženemo “grub rescue mode”
- Kaj lahko naredimo od tu?

```
while (1) {
 // ...
 if (key == '\b') {
  cur_len--;
  grub_printf ("\b");
  continue;
 }
 //...
 if (cur_len + 2 < buf_size) {
  buf[cur_len++] = key;
  grub_printf ("%c", key);
 }
}

grub_memset(buf+cur_len,0,buf_size-cur_len);
```

https://www.hmarco.org/bugs/CVE-2015-8370-Grub2-authentication-bypass.html

HVALA

Vaje

- Disassembling
- fuzzing