

Descubriendo vulnerabilidades en ejecutables con SEA

Gustavo Grieco



SciPyCon Argentina 2013



Bugs y Exploits

Sea un programa **P** y una entrada **i**,

❶ Si $P(i)$ produce un resultado incorrecto o inesperado:

- ▶ **P** tiene un **bug**.
 - ▶ **i** es un **caso de prueba**.
-

❷ Si produce un resultado incorrecto o inesperado, y se puede sacar algún provecho

- ▶ **P** tiene una **vulnerabilidad**.
- ▶ **i** es un **exploit**.

Bugs y Exploits

¿Que sucede cuando alguien descubre bugs o vulnerabilidades?

1

2

3

Bugs y Exploits

¿Que sucede cuando alguien descubre bugs o vulnerabilidades?

- 1 Se **reporta** al fabricante.
- 2
- 3

Bugs y Exploits

¿Que sucede cuando alguien descubre bugs o vulnerabilidades?

- ① Se **reporta** al fabricante.
- ② El fabricante **soluciona** el bug.
- ③

Bugs y Exploits



¿Que sucede cuando alguien descubre bugs o vulnerabilidades?

- 1 Se **reporta** al fabricante.
- 2 El fabricante **soluciona** el bug.
- 3 Todo el mundo **está** contento.

Desgraciadamente..

Desgraciadamente..



Stuxnet

Desgraciadamente..



Stuxnet



Aurora

Desgraciadamente..



Stuxnet



Aurora



El dinero en el dinero..



- Google
 - ▶ Vulnerabilidad en **Chrome** o **Chromium**
- Pwn2own
 - ▶ Vulnerabilidad en **Adobe Acrobat Reader**
 - ▶ Vulnerabilidad en **Internet Explorer 10**

El dinero en el dinero..



- Google

- ▶ Vulnerabilidad en **Chrome** o **Chromium**

60.000 USD

- Pwn2own

- ▶ Vulnerabilidad en **Adobe Acrobat Reader**
- ▶ Vulnerabilidad en **Internet Explorer 10**

El dinero en el dinero..



- Google

- ▶ Vulnerabilidad en **Chrome** o **Chromium**

60.000 USD

- Pwn2own

- ▶ Vulnerabilidad en **Adobe Acrobat Reader**
- ▶ Vulnerabilidad en **Internet Explorer 10**

70.000 USD

El dinero en el dinero..



- Google

- ▶ Vulnerabilidad en **Chrome** o **Chromium**

60.000 USD

- Pwn2own

- ▶ Vulnerabilidad en **Adobe Acrobat Reader**
- ▶ Vulnerabilidad en **Internet Explorer 10**

70.000 USD

100.000 USD

¿Reinventando la rueda ..?



- !Exploitable [?] (Microsoft Corp.)
- Automatic Generation of Control Flow Hijacking Exploits for Software Vulnerabilities [?] (Universidad de Oxford)
- Unleashing MAYHEM on Binary Code [?]

¿Reinventando la rueda ..?



- !Exploitable [?] (Microsoft Corp.)
- Automatic Generation of Control Flow Hijacking Exploits for Software Vulnerabilities [?] (Universidad de Oxford)
- Unleashing MAYHEM on Binary Code [?]

¿Reinventando la rueda ..?



- !Exploitable [?] (Microsoft Corp.)
- Automatic Generation of Control Flow Hijacking Exploits for Software Vulnerabilities [?] (Universidad de Oxford)
*"Unfortunately, I have **no idea** where that code is anymore..."*
- Unleashing MAYHEM on Binary Code [?]

¿Reinventando la rueda ..?



- !Exploitable [?] (Microsoft Corp.)
- Automatic Generation of Control Flow Hijacking Exploits for Software Vulnerabilities [?] (Universidad de Oxford)
*"Unfortunately, I have **no idea** where that code is anymore..."*
- Unleashing MAYHEM on Binary Code [?]
*"We currently have **no plans** of releasing the source code of Mayhem"*

Objetivos del proyecto



x86 / arm \Rightarrow Reverse
Engineering
Intermediate
Language

add
sub
mul
div
mod

x86 / arm \Rightarrow Reverse
Engineering
Intermediate
Language

x86 / arm \Rightarrow

Reverse
Engineering
Intermediate
Language

add
sub
mul
div
mod
or
and
xor
bsh
bisz

x86 / arm \Rightarrow

Reverse
Engineering
Intermediate
Language

add
sub
mul
div
mod
or
and
xor
bsh
bisz
str
stm
ldm

x86 / arm \Rightarrow

Reverse
Engineering
Intermediate
Language

add
sub
mul
div
mod
or
and
xor
bsh
bisz
str
stm
ldm
jcc
nop
undef
unkn

x86 / arm \Rightarrow

Reverse
Engineering
Intermediate
Language

add
sub
mul
div
mod
or
and
xor
bsh
bisz
str
stm
ldm
jcc
nop
undef
unkn

Lecturas complementarias I



A. Autor.

Manual de Lo que sea.

Editorial, 1990.



S. Alguen.

Sobre esto y aquello.

Revista Esto y Aquello. 2(1):50–100, 2000.