Fundamentos del Software: programación verificada y segura

Programación segura

Motivación

Roberto Blanco[†] & Ricardo J. Rodríguez[‡]

MAX PLANCK INSTITUTE
FOR SECURITY AND PRIVACY

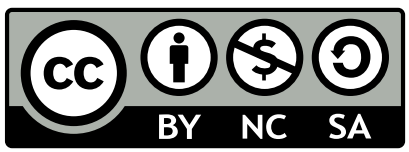
Universidad Zaragoza

[†]Max Plank Institute for Security and Privacy
Bochum, Alemania

[‡]Dpto. de Informática e Ingeniería de Sistemas
Universidad de Zaragoza (España)

Septiembre 2020

Universidad de Zaragoza



Los artefactos software son sistemas complejos

- **Fallos en diseño muy probables**
- **Fallos en el software (bugs): inevitables**
  - De 6 a 16 bugs por 1000 líneas de código (aproximadamente)

Defectos y bugs del software → vulnerabilidades

Introducción

HECHO: Los errores en el software son caros

One bug, one crash

`http://youtu.be/PK_yguLapgA?t=50s`

- **Primera prueba de vuelo del Ariane 5** (1996)
- European Space Agency (ESA)
- Pocos segundos después del lanzamiento, cambió abruptamente el curso de su trayectoria, activándose un mecanismo de autodestrucción
  - Causado por in **error numérico (desbordamiento)**: intento conversión de un número de 64 bits a un número de 16 bits
  - Más información en `https://around.com/ariane.html`

Universidad Zaragoza

Introducción

HECHO: Los errores software son peligrosos

Envejecimiento del software (*software aging*)

- Los sistemas software que están continúamente ejecutándose por un tiempo largo tienden a mostrar:
  - **Rendimiento degradado**
  - **Aumento en la ratio de ocurrencia de fallos**

Universidad Zaragoza

Introducción

HECHO: Los errores software son peligrosos

Envejecimiento del software (*software aging*)

- Los sistemas software que están continúamente ejecutándose por un tiempo largo tienden a mostrar:
  - **Rendimiento degradado**
  - **Aumento en la ratio de ocurrencia de fallos**

Ejemplo: sistema Patriot

- Error en el software del sistema de defensa de misiles Patriot, usado en los incidentes Scud de Dhahran (`https://gulflink.health.mil/scud_info/scud_info_s04.htm`)
- **Parámetros de velocidad y tiempo**: números reales
  - Guardados en registros enteros de 24 bits, contando las decenas de segundos
  - Uso continuado durante 19.4 días (aprox.) sin desbordamiento
  - Conversión a número real necesaria para interceptar
  - Imprecisiones proporcionales al tiempo de ejecución del sistema
- El 21 de febrero de 1991, la Patriot Project Office comunicó que: *"very long runtimes" could negatively affect the system's targeting*
  - El reinicio costaba cerca de 60≈90s (sistema inoperativo)

Introducción

HECHO: Los errores software son peligrosos

FAA warning all airlines to reboot Dreamliners regularly or risk pilots losing control in mid-flight

To prevent a catastrophic software glitch, airlines are ordered to reboot Boeing 787 Dreamliners regularly.

By Mary-Ann Russon

December 12, 2016 12:04 GMT

*"The **FAA is mandating** that operators of Boeing's 787 Dreamliner **periodically reset** the power on the airplane to **avoid a glitch** that **could cause all three computer modules** that manage the jet's flight control surfaces **to briefly stop working while in flight**."*

Introducción

HECHO: Los errores software son peligrosos

- Ejemplo de error en tiempo de ejecución: división entre cero
- *USS Yorktown*, 1998

Software glitches leave Navy Smart Ship dead in the water

By Gregory Slabodkin Jul 13, 1998

"On 21 September 1997, while on maneuvers off the coast of Cape Charles, Virginia, a crew member ***entered a zero*** *into a database field* ***causing an attempted division by zero*** *in the ship's Remote Data Base Manager, resulting in a buffer overflow which brought down all the machines on the network,* ***causing the ship's propulsion system to fail****."*

Universidad Zaragoza 1542

Introducción

HECHO: Los vulnerabilidades del software son valiosas

- **Intencionadas vs. no intencionadas**
- Exploit: aprovecharse de un defecto del software
  - **Detectar un fallo y comunicarlo** a las compañías!
  - **Bug bounty programs**
    - Echa un vistazo `https://bugcrowd.com/list-of-bug-bounty-programs`!

Universidad Zaragoza

Introducción

Bugs de interés ocurridos en los últimos años

Heartbleed

CVE-2014-0160

- **Bug en librería criptográfica OpenSSL**
- Validación de entrada de datos incorrecta en la implementación de protocolo *heratbeat* de TLS
- **Buffer over-read**: pueden leerse datos arbitrarios

Fuente: `https://en.wikipedia.org/wiki/Heartbleed`

Universidad Zaragoza

Introducción

Bugs de interés ocurridos en los últimos años

SHELLSHOCK (aka Bashdoor)

CVE-2014-6271

- **Bug en consola Bash**
- **Ejecución de comandos no intencionada** cuando los comandos se concatenan al final de una definición de función, guardadas en variables de entorno
- Vulnerabilidades relacionadas (CVE-2014-6277, CVE-2014-6278, CVE-2014-7169, CVE-2014-7186, y CVE-2014-7187)

Fuente: `https://en.wikipedia.org/wiki/Shellshock_(software_bug)`

Universidad Zaragoza

Introducción

Bugs de interés ocurridos en los últimos años

Dirty COW

Fuente: `https://en.wikipedia.org/wiki/Dirty_COW`

CVE-2016-5195

- **Vulnerabilidad del núcleo de GNU/Linux**
- **Escalada de privilegios local, explotando una condición de carrera** en la implementación del mecanismo de copy-on-write del gestor de memoria
- **Útil para rootear cualquier dispositivo Android** (hasta Android 7)

Universidad Zaragoza

Introducción

Bugs de interés ocurridos en los últimos años

Spectre

CVE-2017-5753, CVE-2017-5715

- **Afecta a los microprocesadores modernos que realizan predicción de saltos** (mecanismo de optimización en ejecución de código)
- **Ejecución especulativa**:
  - Un fallo de predicción de salto puede dejar elementos observables en la cache, que puede llevar a la revelación de datos privados a un atacante
- **Explotación remota de páginas web maliciosas** (i.e., JavaScript)
- Esto es un *ataque de canal lateral*

Fuente: `https://en.wikipedia.org/wiki/Spectre_(security_vulnerability)`

Universidad Zaragoza

Introducción

Bugs de interés ocurridos en los últimos años

CVE-2017-5754

MELTDOWN

- **Vulnerabilidad hardware** (Intel x86, IBM POWER, y algunos microprocesadores ARM)
- **Un proceso deshonesto puede leer toda la memoria, aunque no esté autorizado**
- **Condición de carrera** entre los accesos a memoria y comprobación de privilegios de acceso
  - Combinado con un ataque de canal lateral a caches, **un proceso no autorizado podría leer datos de cualquier dirección mapeada en el espacio de memoria del proceso**
  - Procesos y recursos del núcleo se mapean siempre en todos los procesos en algunos sistemas operativos

Fuente: `https://en.wikipedia.org/wiki/Meltdown_(security_vulnerability)`

Universidad Zaragoza

Introducción

Bugs de interés ocurridos en los últimos años

Protocolo Samba

CVE-2017-7494

- **Ejecución remota de código en el protocolo Samba**
- Conocido después de la (amplia) difusión del ransom-worm WannaCry
  - `EternalBlue` usado como mecanismo de difusión. Parcheado el 14 de marzo de 2017 (MS17-010)
  - `DoublePulsar` para escalada de privilegios local
  - Comienzo del ataque: 12 de mayo, 2017 – *do the maths, folks*

Fuente:

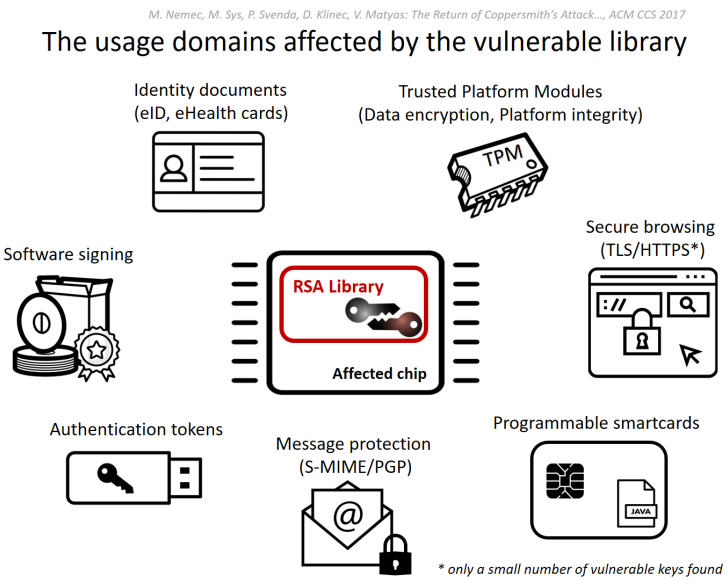
`https://nakedsecurity.sophos.com/2017/05/26/samba-exploit-not-quite-wannacry-for-linux-but-patch-anyway/`

Universidad Zaragoza

Introducción

Bugs de interés ocurridos en los últimos años

ROCA



Fuente: `https://en.wikipedia.org/wiki/ROCA_vulnerability`

CVE-2017-15361

- **Fallo criptográfico que permite recuperar la clave privada a través de su clave pública**
- **Error en la generación de claves RSA en la librería RSAlib** (de Infineon)
  - Chips existentes en multitud de tarjetas inteligentes e implementaciones *Trusted Platform Module* (TPM)

Universidad Zaragoza

Introducción

Bugs de interés ocurridos en los últimos años

CVE-2019-13456

- WPA3 propone un nuevo protocolo de handshake para solventar los problemas de WPA2: *Dragonfly handshake*
  - **Un atacante en el rango de la víctima puede recuperar la contraseña**

CVE-2019-13377

- Ataque de canal lateral (*side-channel*)
  - Existen diferencias observables en el tiempo y en patrones de acceso a cache cuando se usan curvas Brainpool
  - **Acceso a red Wi-Fi, sin conocer contraseña**
- Implementaciones de SAE y EAP-pwd en `hostapd` y `wpa_supplicant` 2.x–2.8

La implementación de Dragonfly sin fugas de canal lateral es sorprendentemente complicado

Fuente: `https://wpa3.mathyvanhoef.com/`

Fundamentos del Software: programación verificada y segura

Programación segura

Motivación

Roberto Blanco[†] & Ricardo J. Rodríguez[‡]

MAX PLANCK INSTITUTE
FOR SECURITY AND PRIVACY

Universidad Zaragoza

[†]Max Plank Institute for Security and Privacy
Bochum, Alemania

[‡]Dpto. de Informática e Ingeniería de Sistemas
Universidad de Zaragoza (España)

Septiembre 2020

Universidad de Zaragoza