Software Vulnerabilities: Exploitation and Mitigation

Lecture 1
Principles of Computer Security, Software Development & Software Vulnerabilities

(18 Feb. 2019)

Alexandre Bartel
alexandre.bartel@uni.lu

Software

What is “Software”?

- 1850 document [1] :

> *Two other departments [among rubbish-tip pickers], called the “**soft-ware**” and the “hard-ware,” are very important. The former includes all vegetable and animal matters—everything that will decompose.*

[1] Shapiro, Fred R. "Origin of the term software: Evidence from the JSTOR electronic journal archive." (2000): 69-70.

https://web.archive.org/web/20030605004419/http://computer.org/annals/an2000/pdf/a2069.pdf

What is “Software”?

- Professor John W. Tukey in 1958 [1]
- In “The teaching of concrete mathematics” [2]

*Today the “**software**” comprising the carefully planned interpretive routines, compilers, and other aspects of automative programming are at least as important to the modern electronic calculator as its “hardware” of tubes, transistors, wires, tapes, and the like.*

Source: wikipedia.org

John Wilder Tukey (1915-2000)

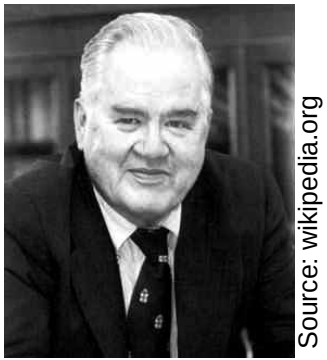
[1] Shapiro, Fred R. "Origin of the term software: Evidence from the JSTOR electronic journal archive." (2000): 69-70.

https://web.archive.org/web/20030605004419/http://computer.org/annals/an2000/pdf/a2069.pdf

[2] Tukey, John W. "The teaching of concrete mathematics." The American Mathematical Monthly 65.1 (1958): 1-9.

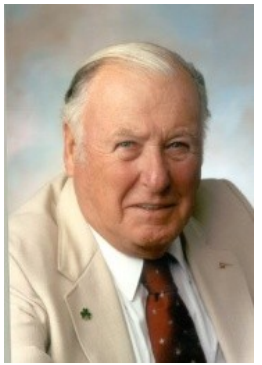
What is “Software”?

- Professor John W. Tukey
  - “software”
  - “bit” (a portmanteau of binary digit)
  - FFT (Fast Fourier Transform) algorithm
    - Together with James William Cooley
    - invented by Carl Friedrich Gauss 160 years before...



Source: wikipedia.org

John Wilder Tukey (1915-2000)



Source: computer.org

James William Cooley (1926-2016)

Source: wikipedia.org

Carl Friedrich Gauß (1777–1855)

What is “Software”?

- Today

  D1: *The programs and other operating information used by a computer.*

  D2: *(computer science) written programs or procedures or rules and associated documentation pertaining to the operation of a computer system and that are stored in read/write memory*

Source:
https://en.oxforddictionaries.com/definition/software
https://www.thefreedictionary.com/software

Software Development – Life Cycle

- Initial idea
- Requirements
- Design
- Implementation / Test
- Deployment / Maintenance
- Disposal

Software Development Model: “Waterfall”

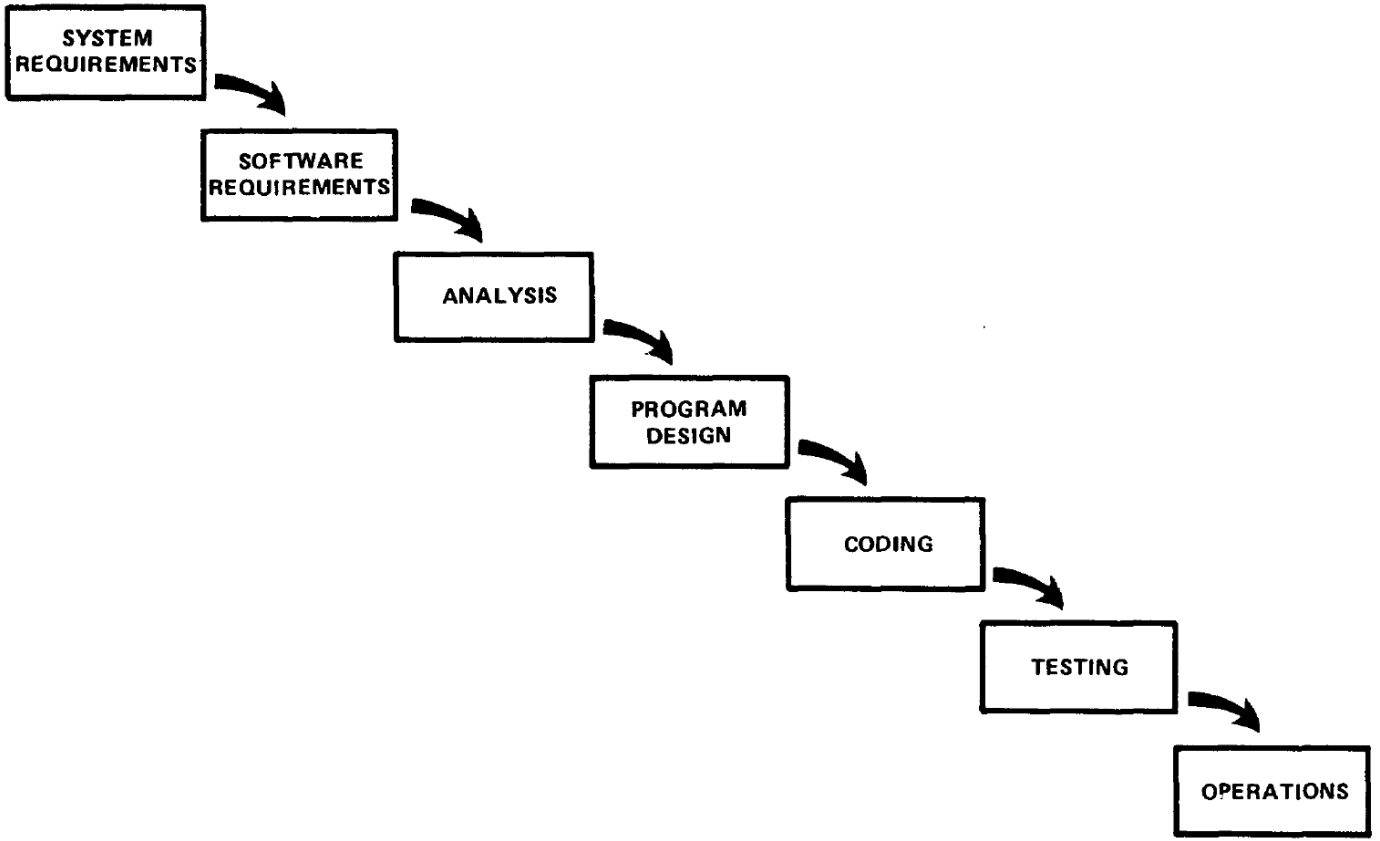


Figure 2. Implementation steps to develop a large computer program for delivery to a customer.

Source: wikipedia.org

Winston Walker Royce (1929 - 1995)

Source: Royce, Winston W. "Managing the development of large software systems: concepts and techniques." Proceedings of the 9th international conference on Software Engineering. IEEE Computer Society Press, 1987.

“Waterfall”?

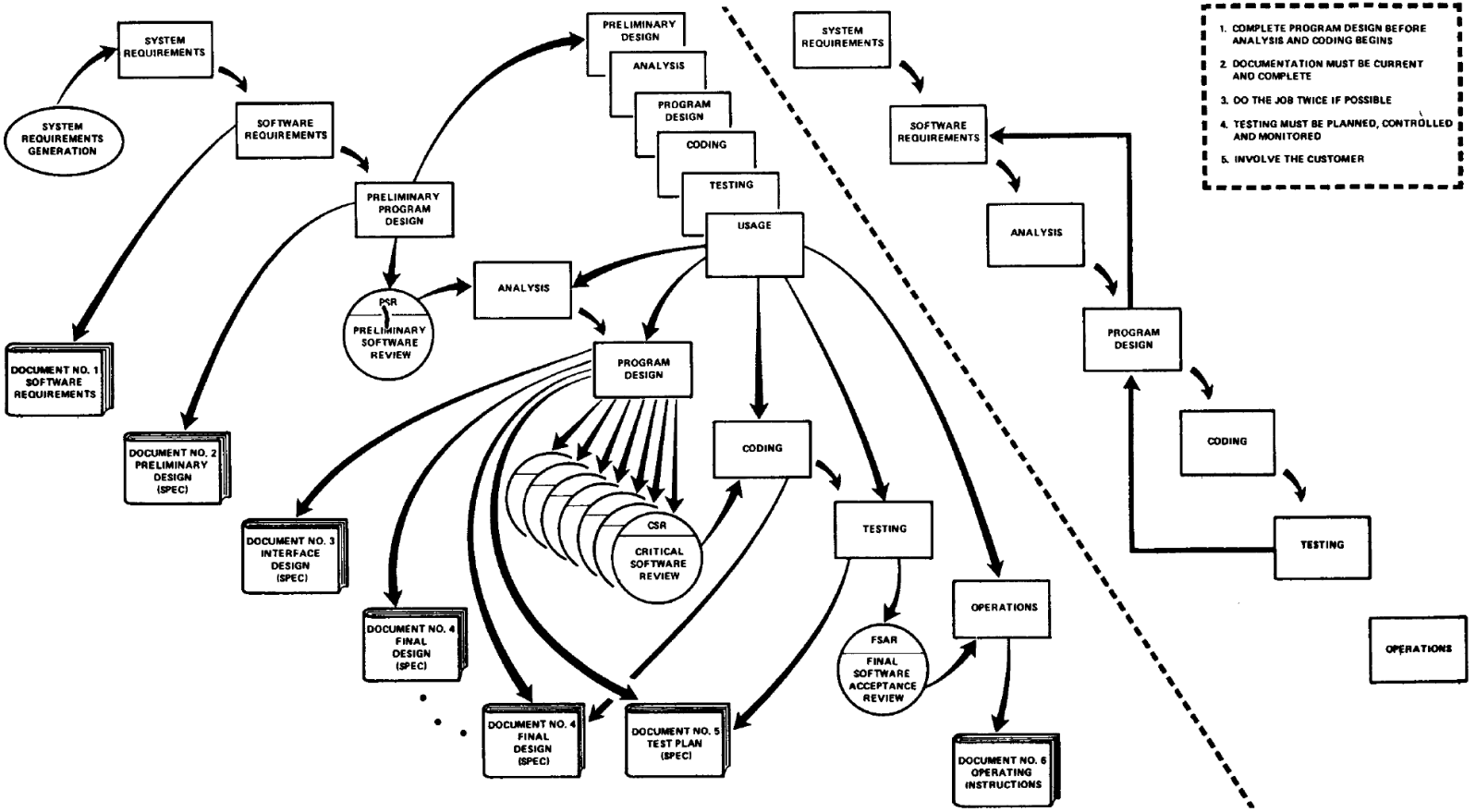
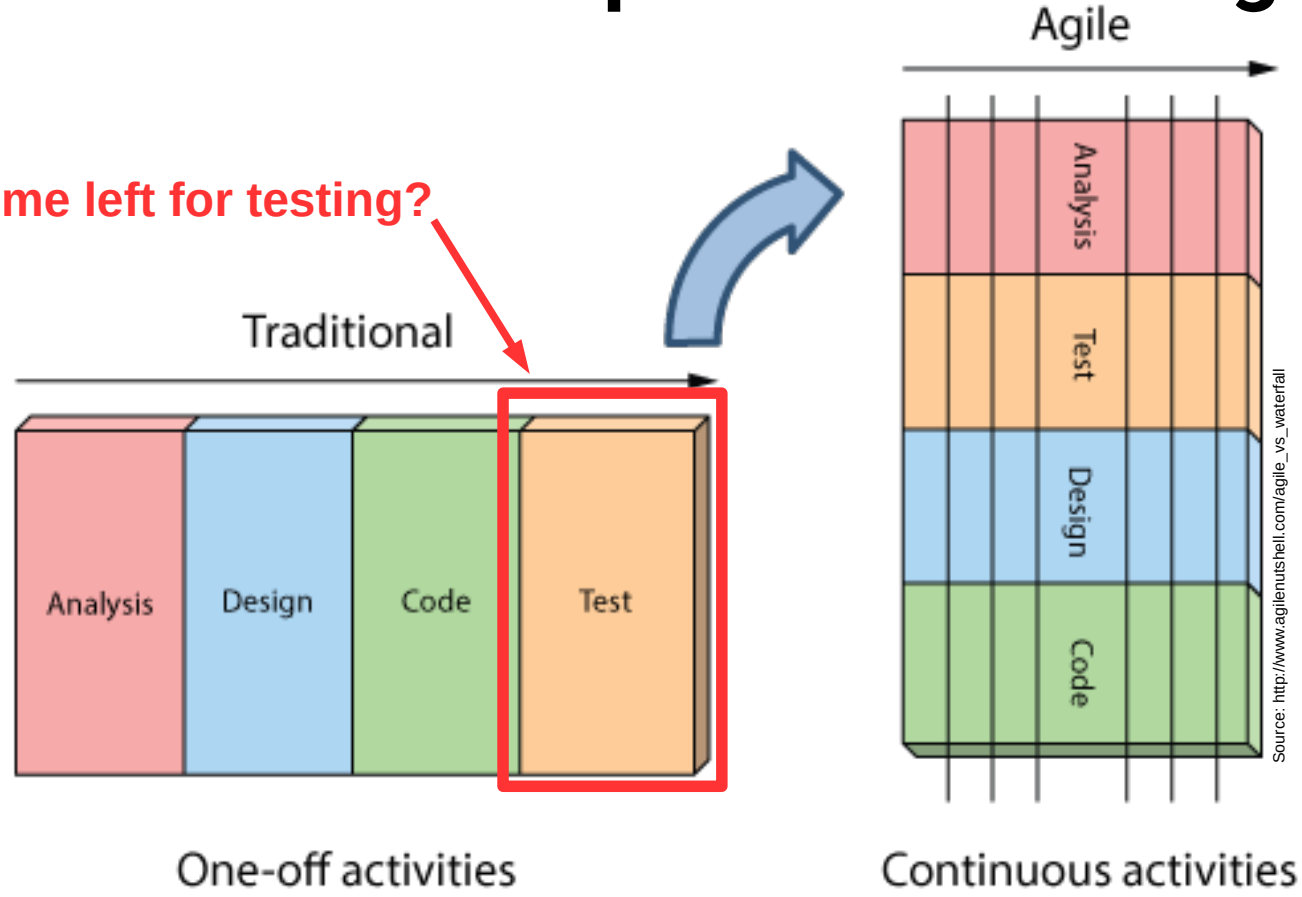


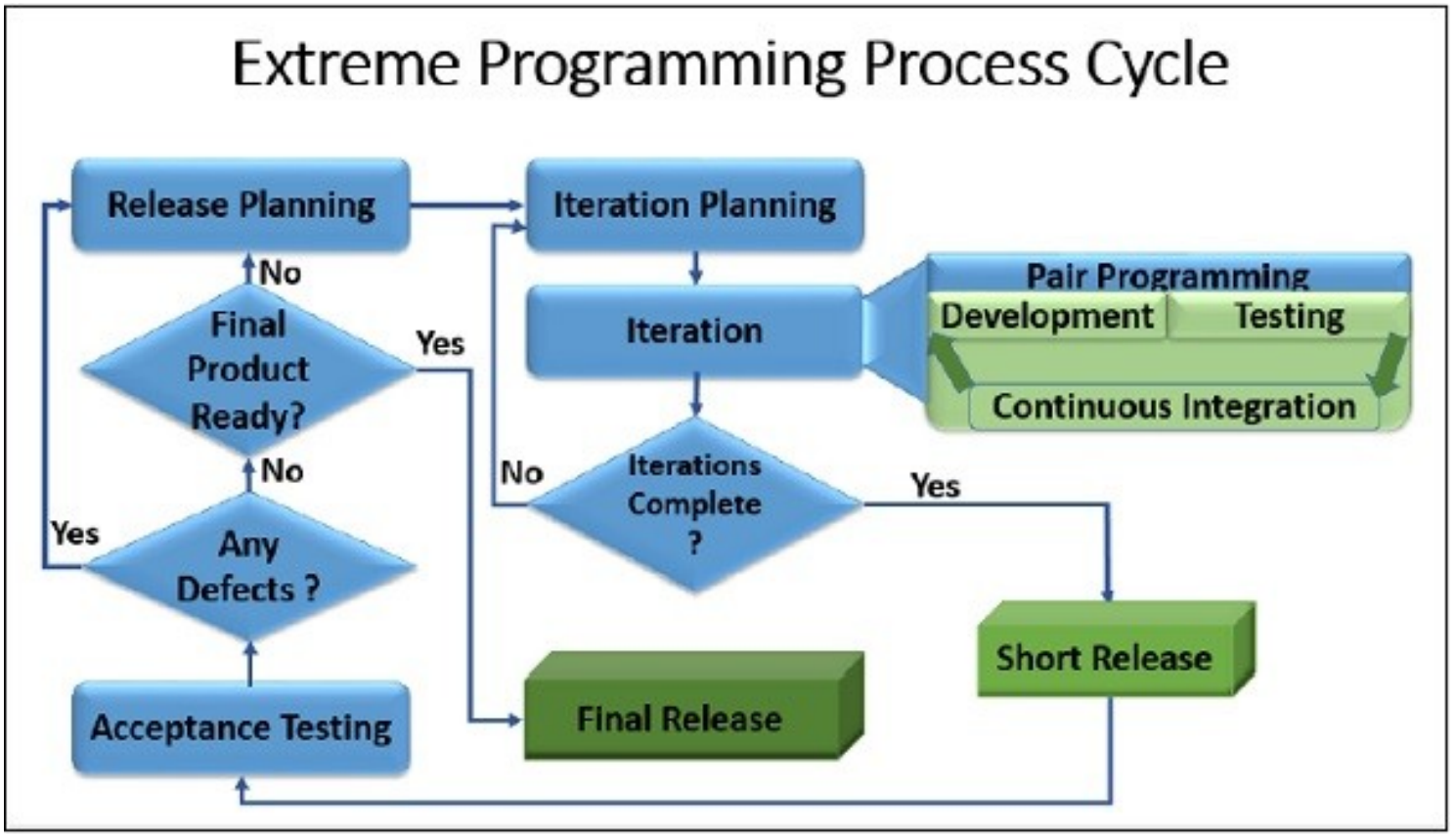
Figure 10. Summary

Software Development Model: Agile

What if not time left for testing?



Agile: “Extreme Programming”



Other Software Development Methodologies

- Rapid prototyping
- Spiral (focus on minimizing project risk)
- Chaos model (solve most important issue fist)
- ...

Goals of Software Methodologies

- Same time spent
- Less risk
- Better quality

Software Vulnerability

Vulnerability

“A flaw or weakness in a system's design, implementation, or operation and management that could be exploited to violate the system's security policy.” [1]

[1] Shirey, Robert. "RFC 2828: Internet security glossary." The Internet Society 13 (2000).

Vulnerability

- Initial idea
- Requirements
- Design
- Implementation / Test
- Deployment / Maintenance
- Disposal

Worst kind: difficult to correct once system deployed.
Ex: WEP

Security Policy

- A Security policy defines what it means to be secure for a software system

Security Attributes

- Software system attributes to maintain:
  - Confidentiality (ex: no read of files)
  - Integrity (ex: no modification of files)
  - Availability (ex: no interruption of service)
  - [Accountability (ex: no untraceable actions)]

Security Mechanisms

- Used to maintain system attributes (CIA + Accountability)
- Security measures:
  - Access control
  - Firewall
  - Intrusion detection systems
  - ...

Vulnerability

"**Most systems have vulnerabilities of some sort**, but this does not mean that the systems are too flawed to use. Not every threat results in an attack, and not every attack succeeds. Success depends on the degree of vulnerability, the strength of attacks, and the effectiveness of any countermeasures in use. If the attacks needed to exploit a vulnerability are very difficult to carry out, then the vulnerability may be tolerable. If the perceived benefit to an attacker is small, then even an easily exploited vulnerability may be tolerable. However, if the attacks are well understood and easily made, and if the vulnerable system is employed by a wide range of users, then it is likely that there will be enough benefit for someone to make an attack." [1]

[1] Shirey, Robert. "RFC 2828: Internet security glossary." The Internet Society 13 (2000).

Reporting a Vulnerability [1]

- A vulnerability is discovered (by a third party)
- Who to tell?
- What to tell?
- When to tell?

[1] Cencini, Andrew, Kevin Yu, and Tony Chan. "Software vulnerabilities: full-, responsible-, and non-disclosure." December 7 (2005), University of Washington.

Terminology / Actors

- Software product
- Flaw
- Vulnerability
- Exploit

- Vendor
- Discoverer
- Originator
- Customer / end user
- Black Hat
- White Hat
- Script Kiddie

Vulnerability Life Cycle

- Birth
- Discovery
- Disclosure
- Correction
- Publicity
- Scripting
- Death

Disclosure “Flavours”

- Non-disclosure
- Full-disclosure
- Responsible-disclosure

Non-Disclosure

- A researcher finds a software vulnerability
- The researcher keeps this vulnerability a secret
- Motivation
  - Malicious intent (e.g., black hats or governments)
  - Laziness (?)
  - Profit (selling information on the black market)

Full Disclosure

- A researcher finds a vulnerability
- The researchers publicly informs the community
- Pros:
  - Ethically correct to inform
  - May motivate software vendors to patch quickly (what if not responding?)
  - Immediate credit
- Cons:
  - Increases the risk of widespread exploitation

Responsible Disclosure

- Least risk for users
- Basic steps
  - Inform vendor and give it time to patch
  - Follow with full disclosure after some time (30 days?, more?, less?)
- **USE THIS APPROACH IF YOU FIND A VULNERABILITY**

Ex: Yahoo 2014

“Time is of the essence when we discover these types of issues: the more quickly we address the risks, the less harm an attack can cause. ***Today, we are committing to publicly disclosing on our security Tumblr the vulnerabilities we discover within 90 days****. By committing to this short time frame, we will help ensure that these vulnerabilities are patched as quickly as possible. We reserve the right to extend or shorten this timeline based on extenuating circumstances, including active exploitation, or known threats. We also commit to sharing the appropriate technical details so other parties can assess their risk and take appropriate action.”* [1]

[1] https://yahoopolicy.tumblr.com/post/104777538533/users-first-our-vulnerability-disclosure-policy

Ex: Google 2015

*“Project Zero **has adhered to a 90-day disclosure deadline**. Now we are applying this approach for the rest of Google as well. We notify vendors of vulnerabilities immediately, with details shared in public with the defensive community after 90 days, or sooner if the vendor releases a fix. We’ve chosen a middle-of-the-road deadline timeline and feel it’s reasonably calibrated for the current state of the industry.”* [1]

[1] https://googleprojectzero.blogspot.com/2015/02/feedback-and-data-driven-updates-to.html

Responsible Disclosure: HOWTO?

- In most cases, vulnerabilities are identified by a “CVE” number
- CVE: Common Vulnerability Exposure
  - Vulnerability description
  - External references
- The US-based National Cybersecurity FFRDC (federally funded research and development center), operated by the Mitre Corporation, maintains the system, with funding from the National Cyber Security Division of the United States Department of Homeland Security.

Who Maintains this list of CVEs?

- The US-based National Cybersecurity FFRDC (federally funded research and development center), operated by the Mitre Corporation, maintains the system, with funding from the National Cyber Security Division of the United States Department of Homeland Security.

Requesting a CVE

- https://cve.mitre.org/cve/request_id.html
- CVEs are assigned by a CVE Numbering Authority (CNA)
- There are two main types of CNAs:
  - The Mitre Corporation functions as Editor and Primary CNA
  - Various CNAs assign CVE numbers for their own products (e.g. Microsoft, Oracle, HP, Red Hat, etc.)

CVE Example: CVE-2018-2826

CVE-ID	
CVE-2018-2826	Learn more at National Vulnerability Database (NVD) • CVSS Severity Rating • Fix Information • Vulnerable Software Versions • SCAP Mappings • CPE Information
Description	
Vulnerability in the Java SE component of Oracle Java SE (subcomponent: Libraries). The supported version that is affected is Java SE: 10. Difficult to exploit vulnerability allows unauthenticated attacker with network access via multiple protocols to compromise Java SE. Successful attacks require human interaction from a person other than the attacker and while the vulnerability is in Java SE, attacks may significantly impact additional products. Successful attacks of this vulnerability can result in takeover of Java SE. Note: This vulnerability applies to Java deployments, typically in clients running sandboxed Java Web Start applications or sandboxed Java applets, that load and run untrusted code (e.g., code that comes from the internet) and rely on the Java sandbox for security. This vulnerability does not apply to Java deployments, typically in servers, that load and run only trusted code (e.g., code installed by an administrator). CVSS 3.0 Base Score 8.3 (Confidentiality, Integrity and Availability impacts). CVSS Vector: (CVSS:3.0/AV:N/AC:H/PR:N/UI:R/S:C/C:H/I:H/A:H).	

CVSS: Common Vulnerability Scoring System

CVE Example: CVE-2018-2826

References

Note: References are provided for the convenience of the reader to help distinguish between vulnerabilities. The list is not intended to be complete.

- CONFIRM:http://www.oracle.com/technetwork/security-advisory/cpuapr2018-3678067.html
- CONFIRM:https://security.netapp.com/advisory/ntap-20180419-0001/
- CONFIRM:https://help.ecostruxureit.com/display/public/UADCE725/Security+fixes+in+StruxureWare+Data+Center+Expert+v7.6.0
- UBUNTU:USN-3747-1
- URL:https://usn.ubuntu.com/3747-1/
- BID:103796
- URL:http://www.securityfocus.com/bid/103796
- SECTRACK:1040697
- URL:http://www.securitytracker.com/id/1040697

CVE Example: CVE-2018-2826

Assigning CNA	
Oracle	
Date Entry Created	
20171215	Disclaimer: The entry creation date may reflect when the CVE ID was allocated or reserved, and does not necessarily indicate when this vulnerability was discovered, shared with the affected vendor, publicly disclosed, or updated in CVE.
Phase (Legacy)	
Assigned (20171215)	

Attacks

Attack 1: Availability of a Web Server

- The web server allows users to upload images
- A bug in the software processing images reboots the web server when specific images are sent
- Attack?
- CIA?

Attack 2: Arbitrary code Execution

- A game allows users to play on custom “maps”
- A specially crafted map allows an attacker to execute arbitrary code on the machine running the game
- Attack?
- CIA?

Questions?

Next Lecture:
Buffer Overflows