Vulnerability Assessment of Networked Systems

JOÃO PAULO BARRACA

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

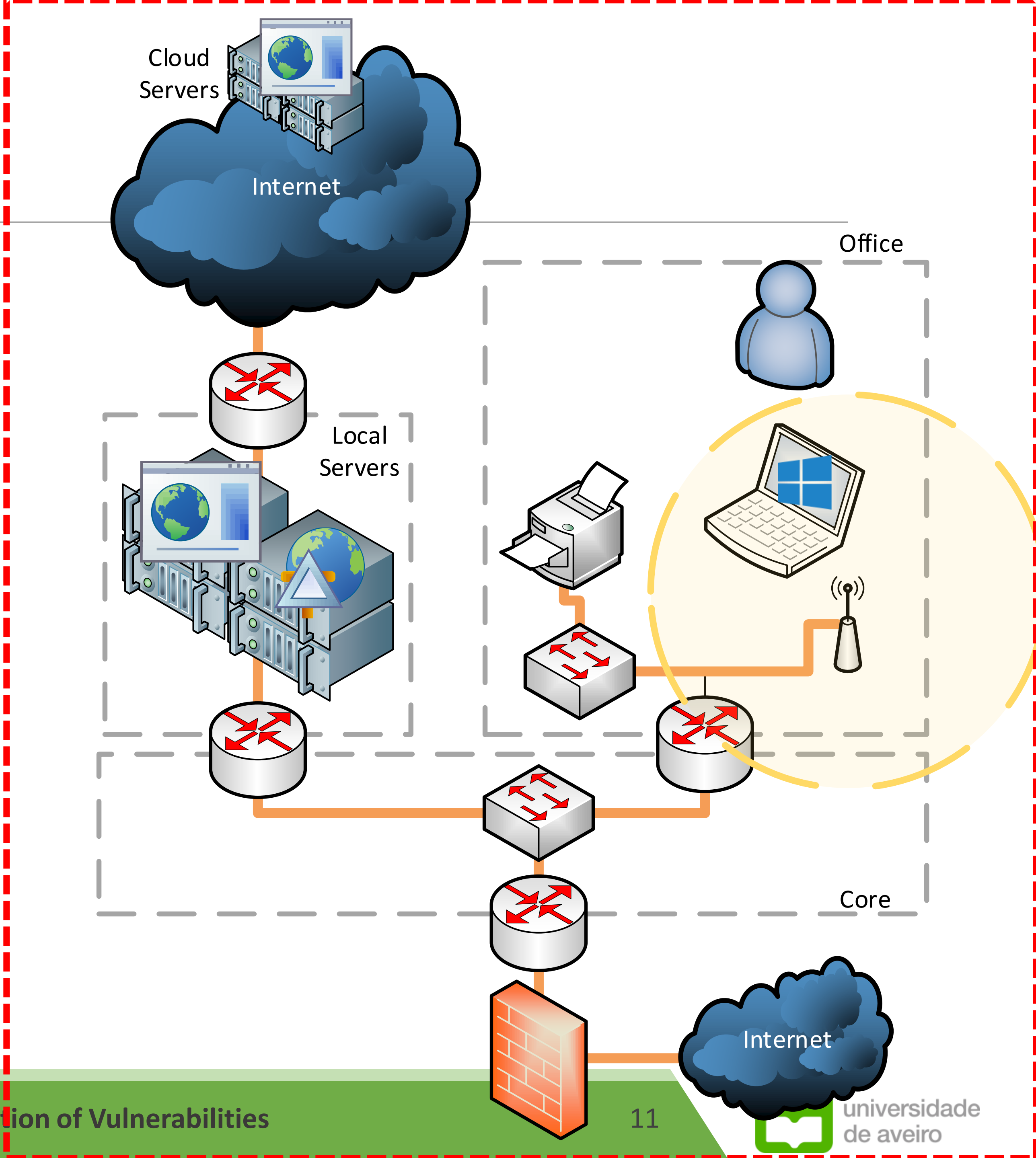
- Send probes
- Checks information repositories

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



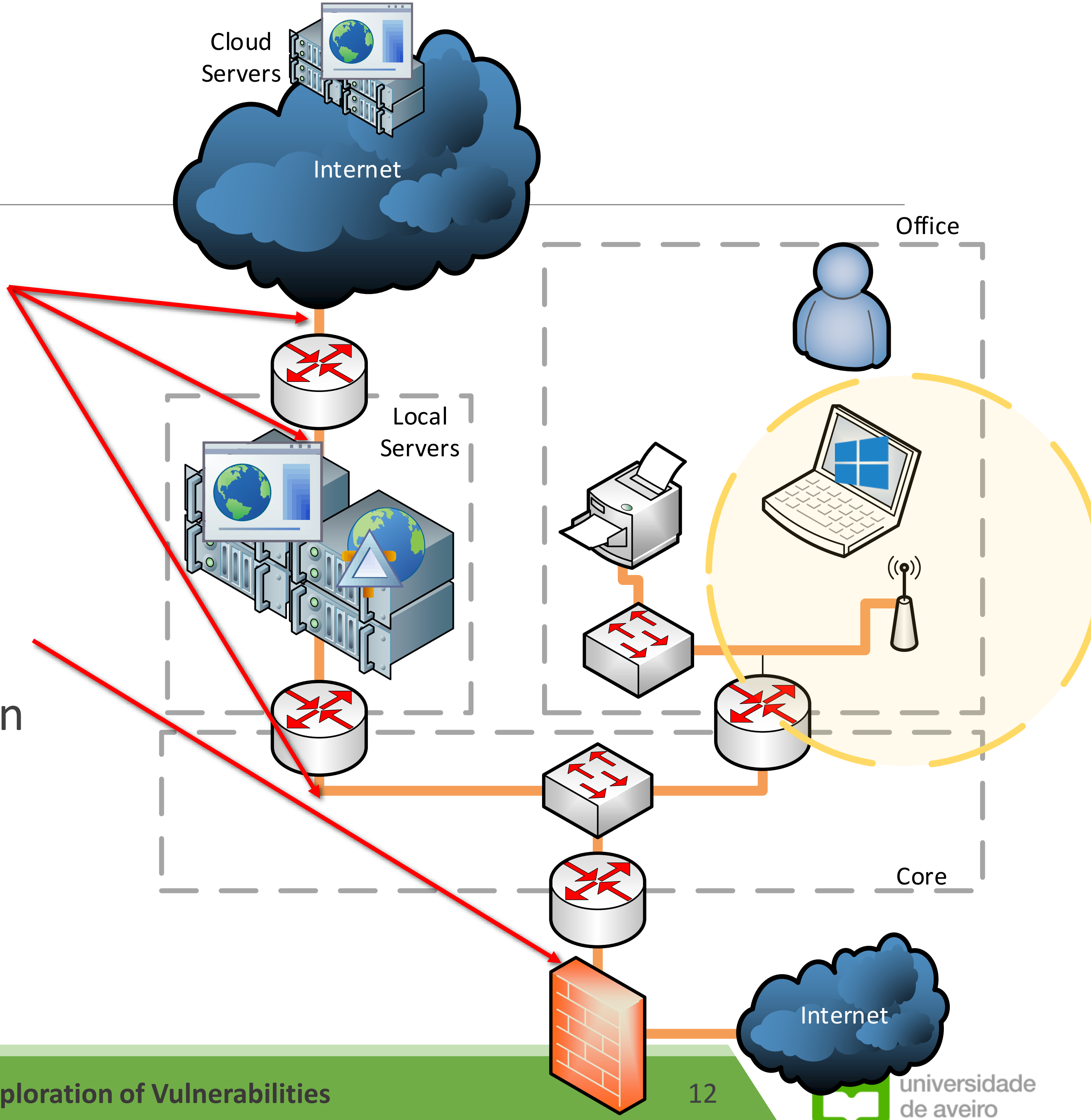
Type: Passive

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

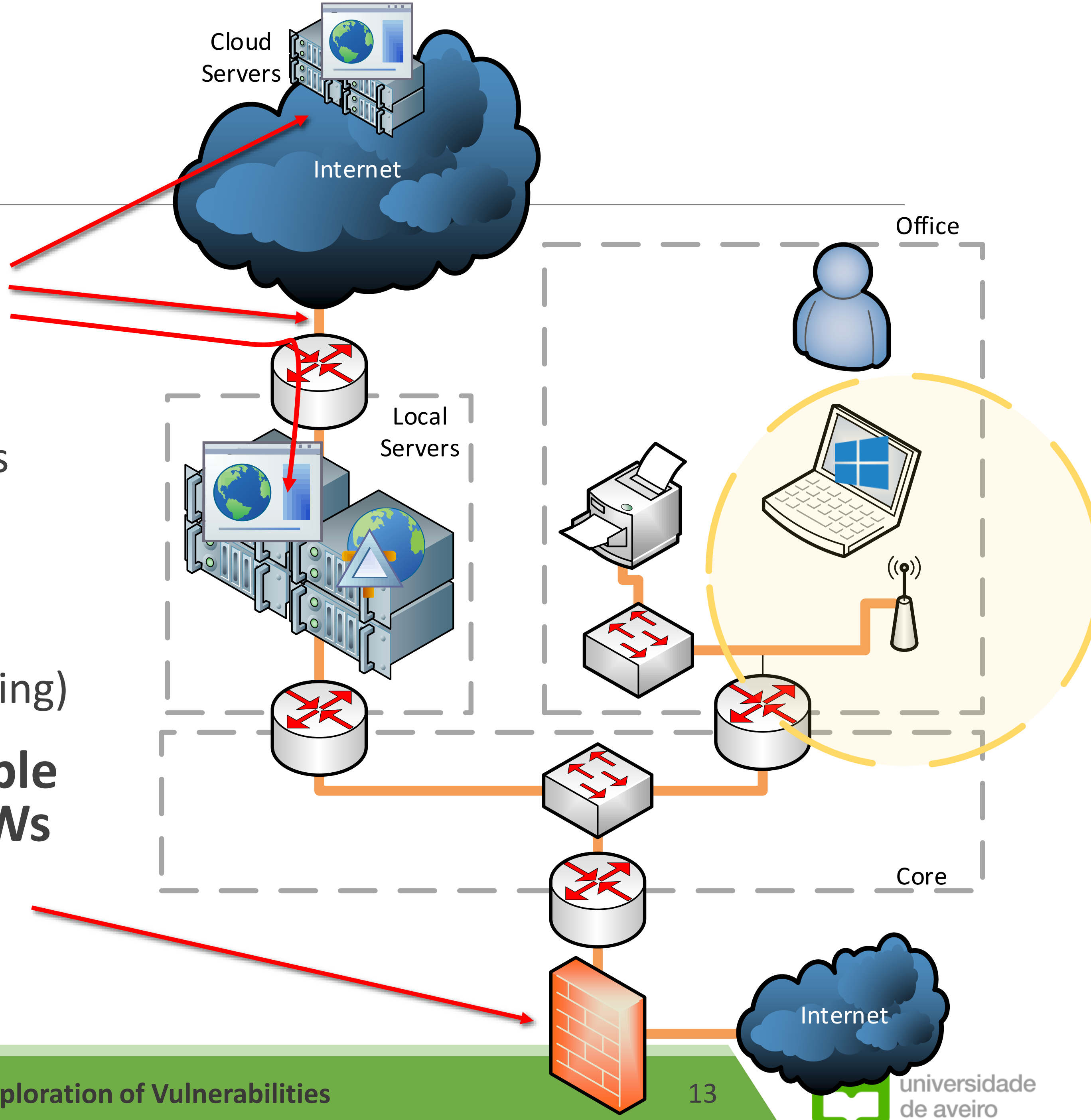
- External attackers

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



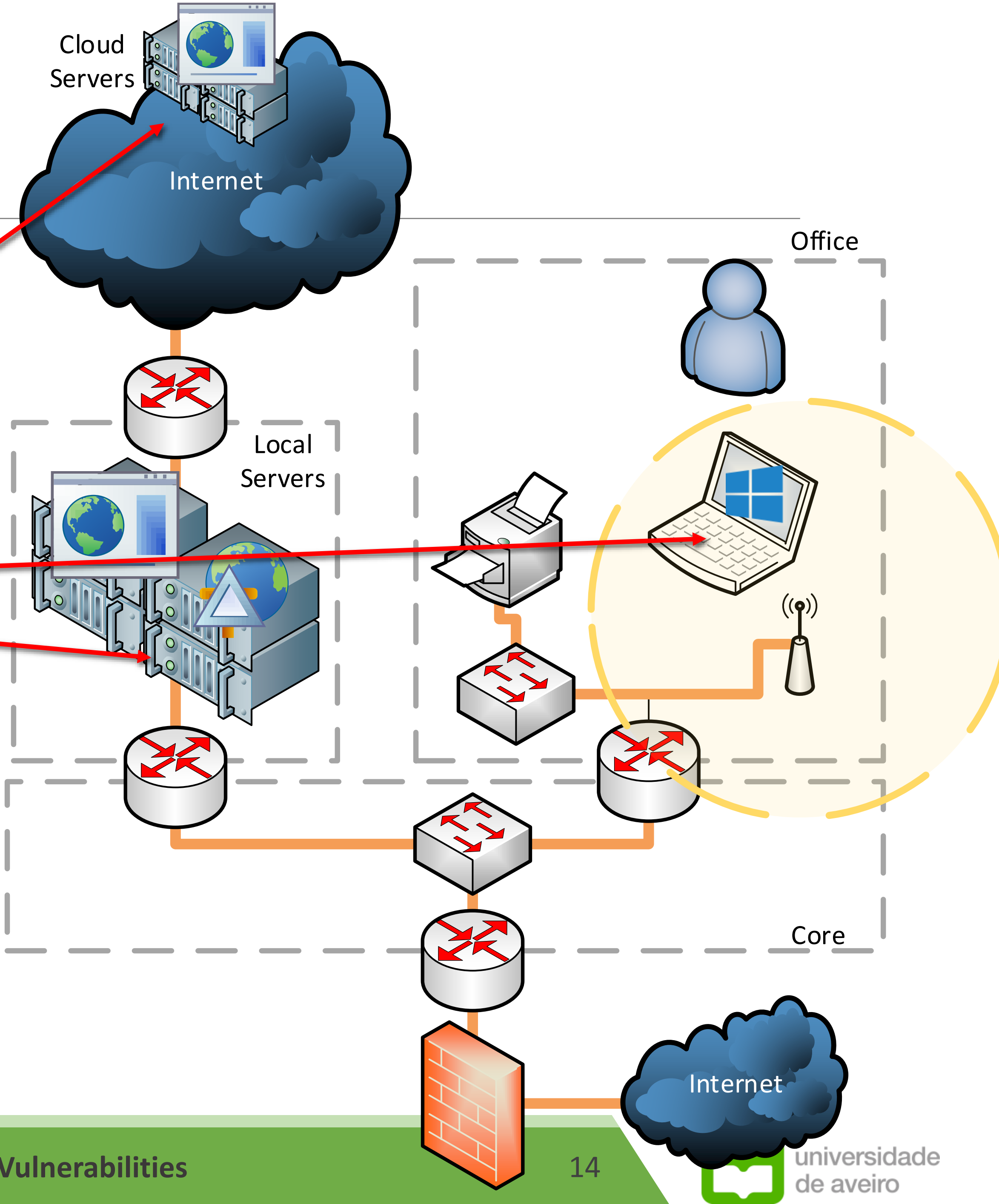
Type: Host Based

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



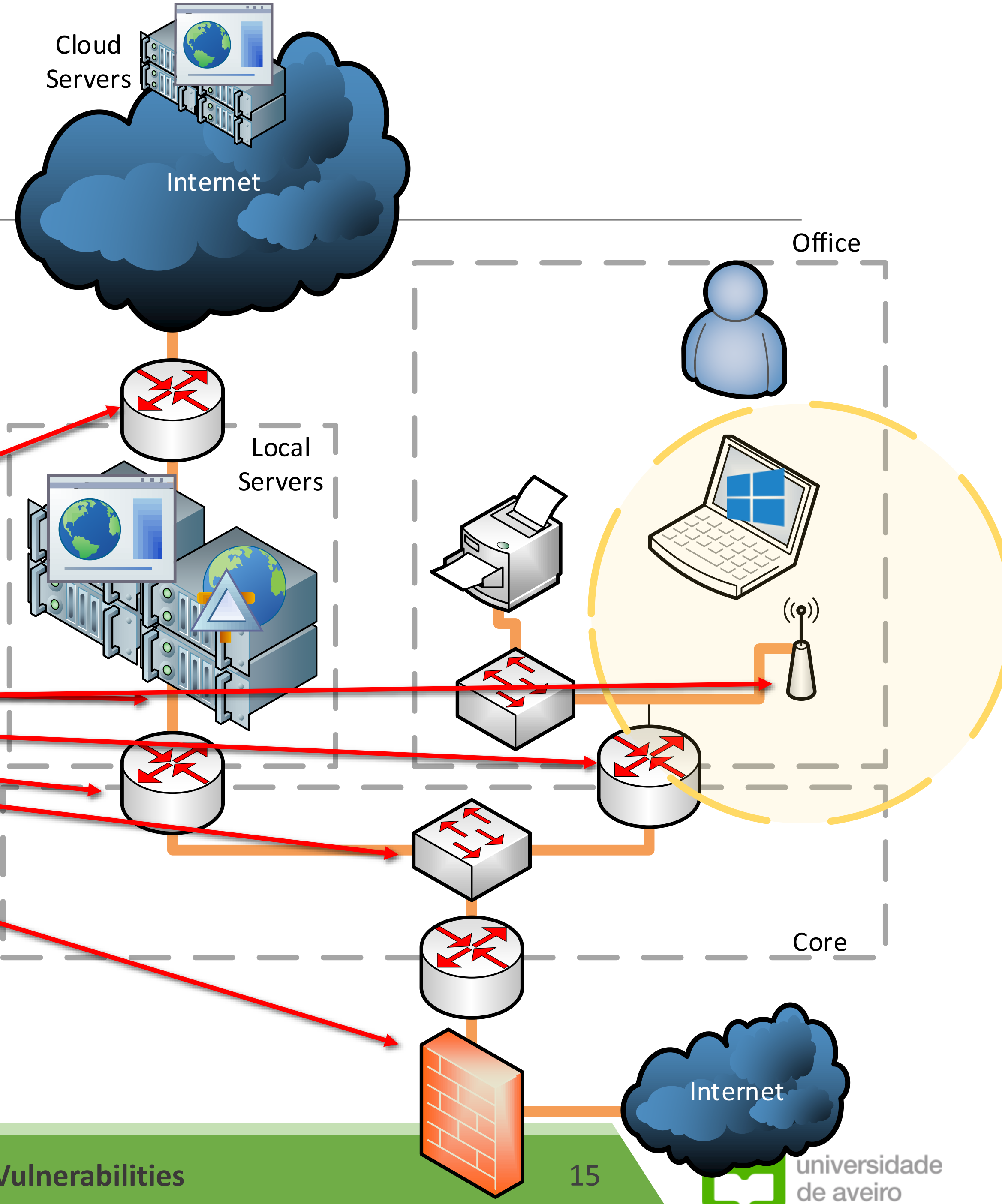
Type: Network

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

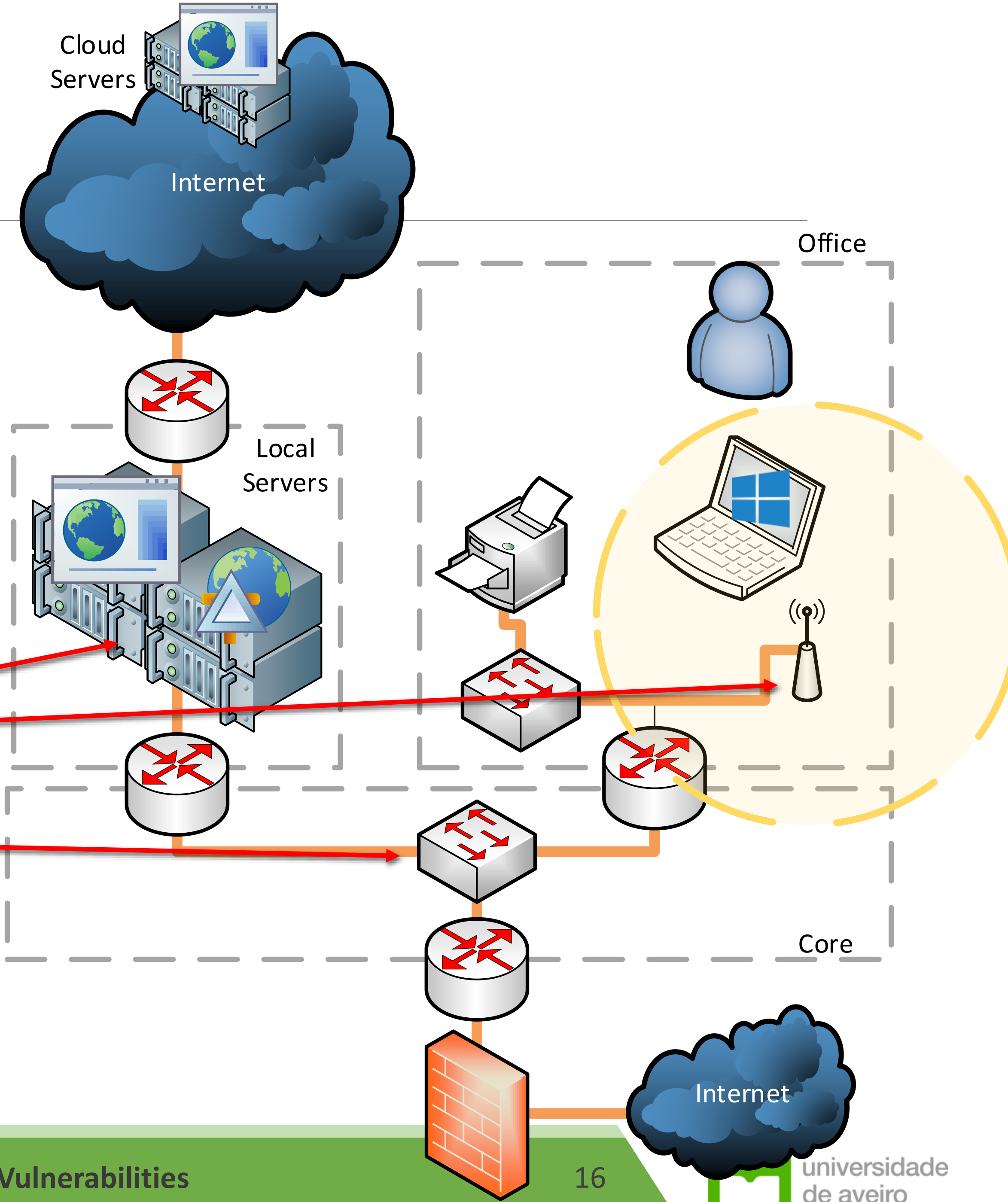
Type: Wireless

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

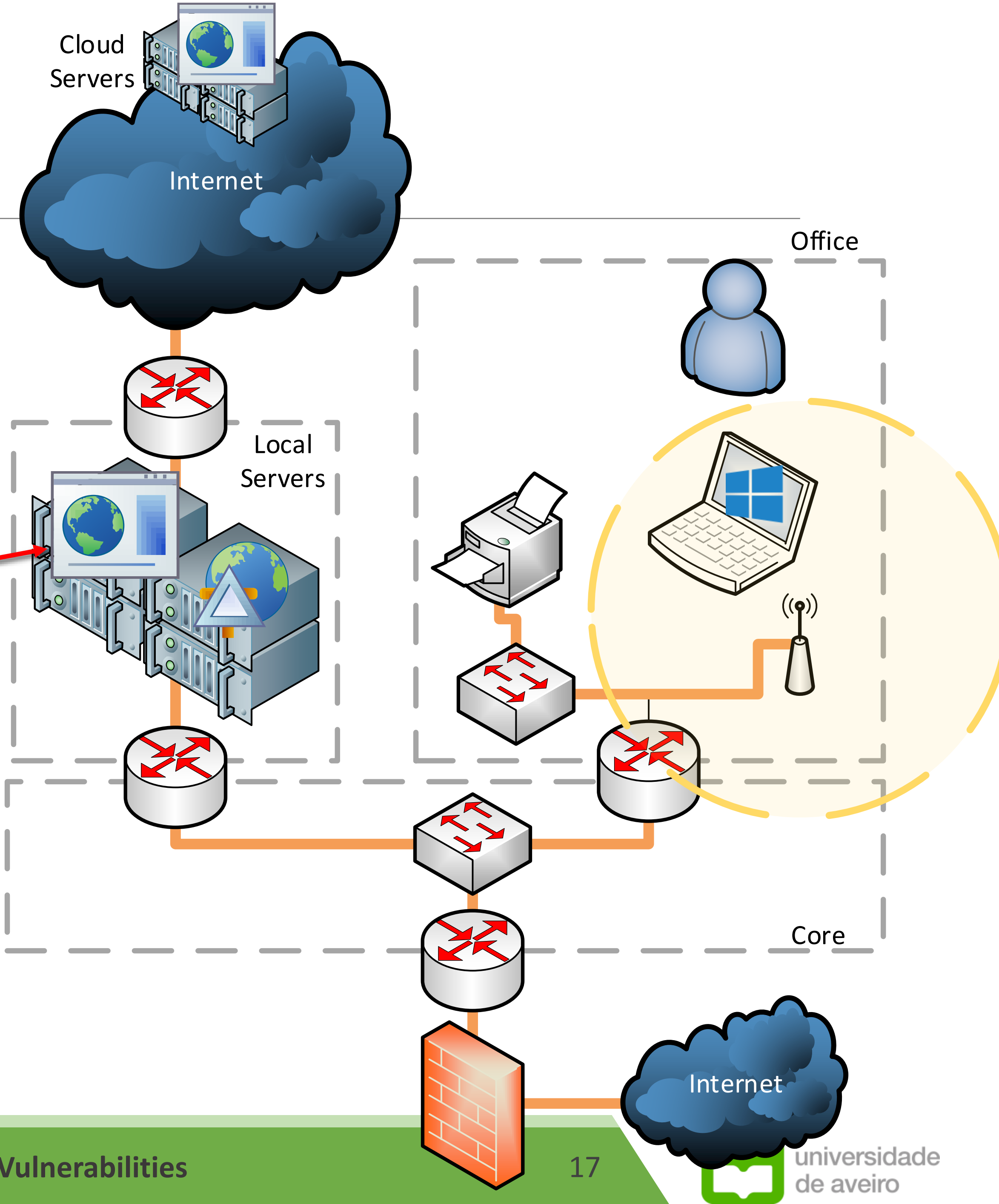
Type: Application

Focus on a single application

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

Targets:

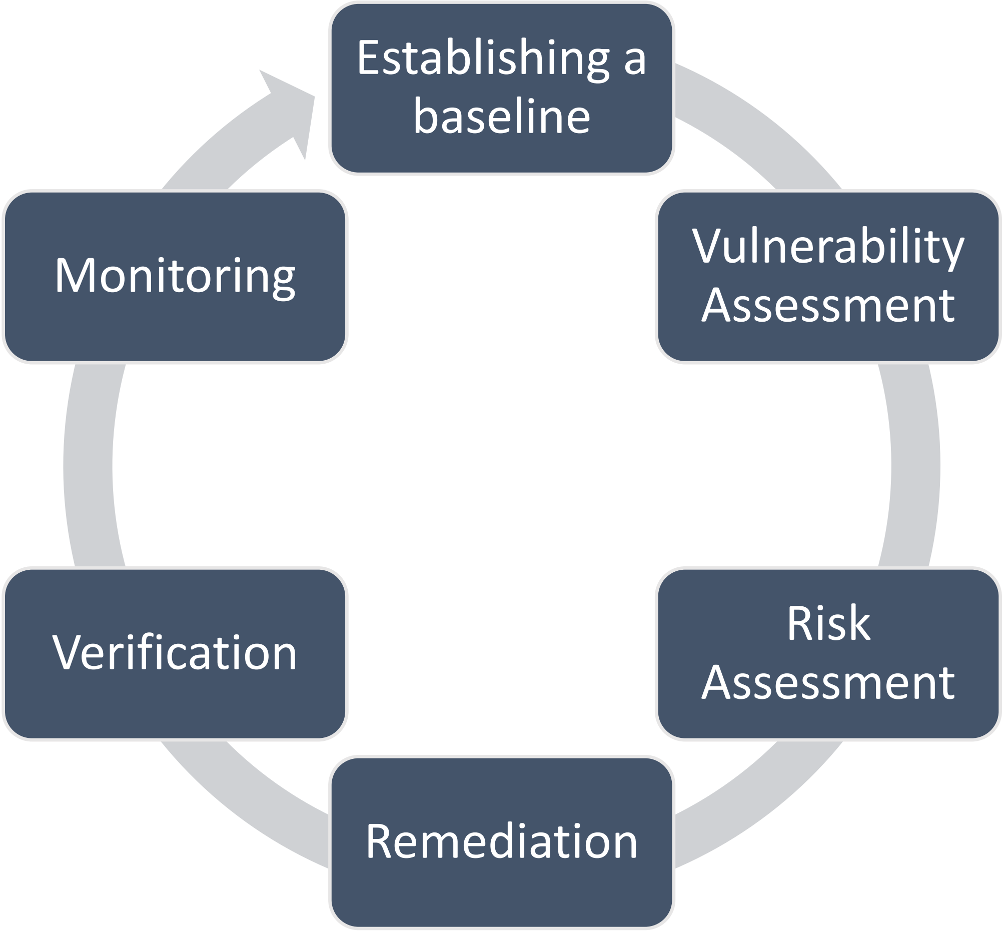
- Application
- Service



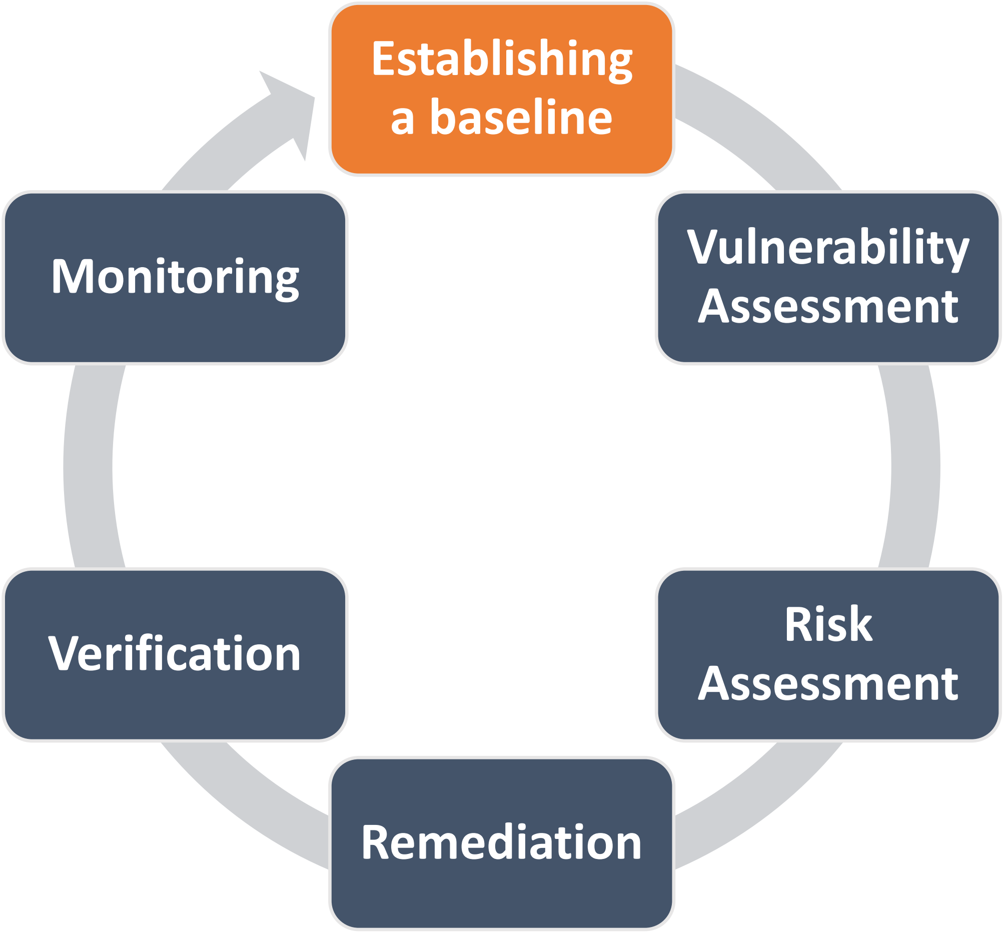
Finds software vulnerabilities in the targeted application

- 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

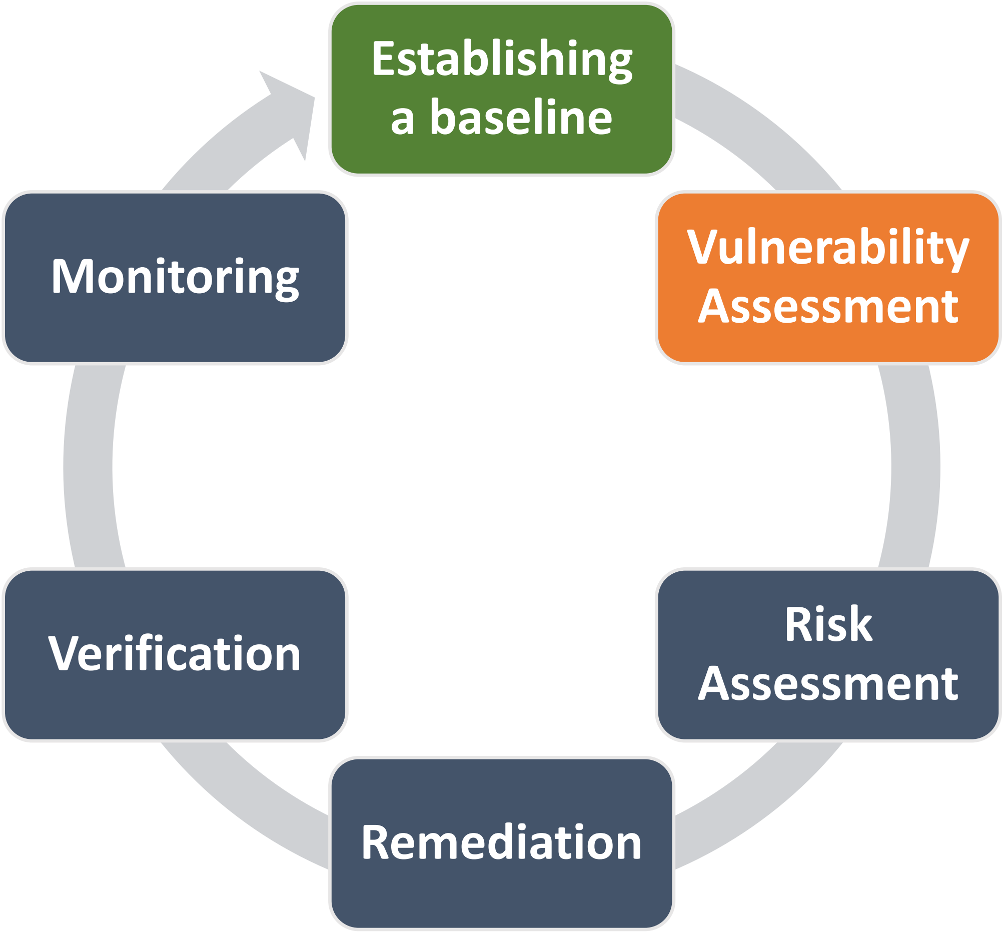
- Some assets may be excluded due to potential impact or cost

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 are the recommendations to handle it

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 with a limited scope
- 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

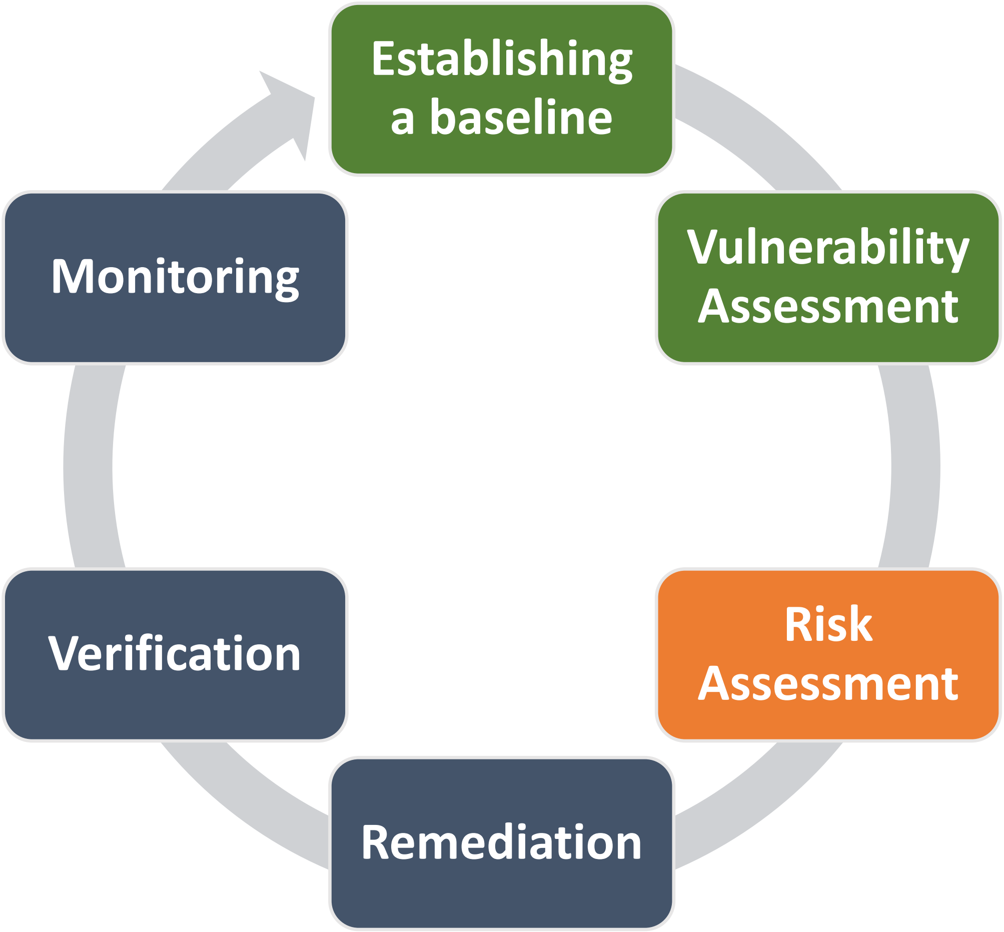
Some information is provided to researchers

- 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

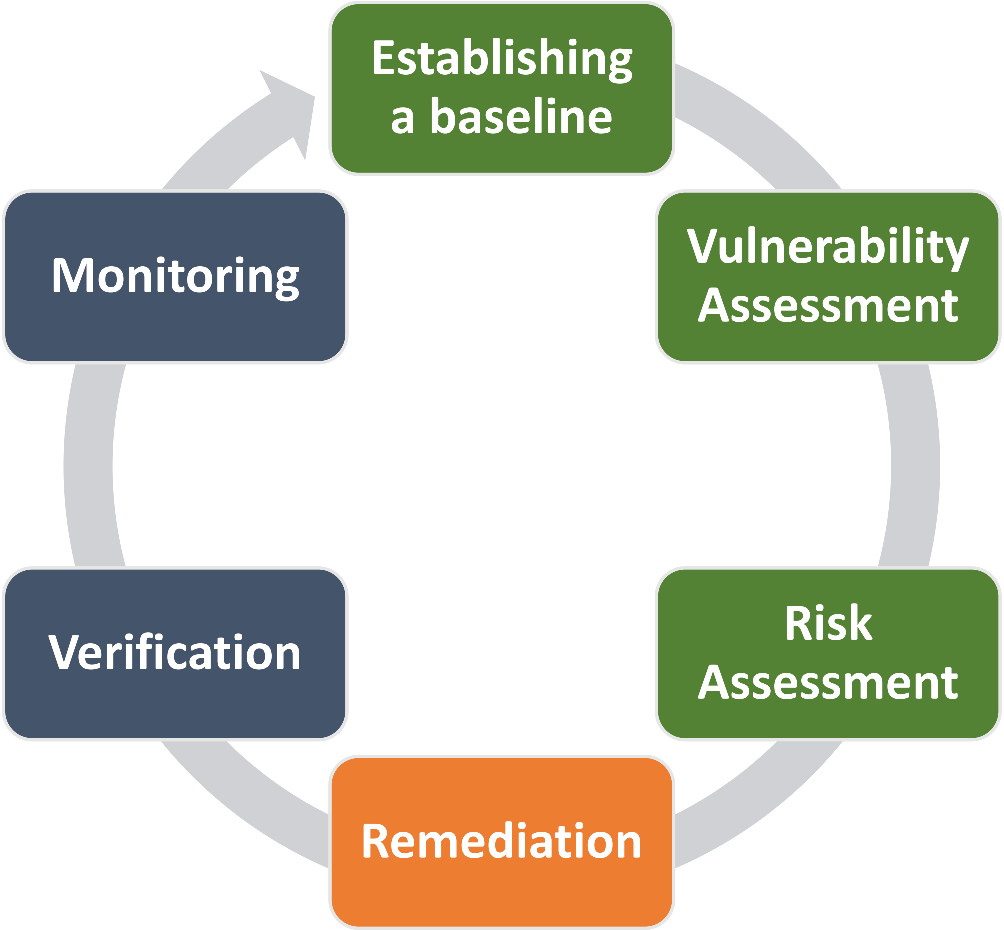
Researchers should carefully document assessments

- 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
- This capability is not always present

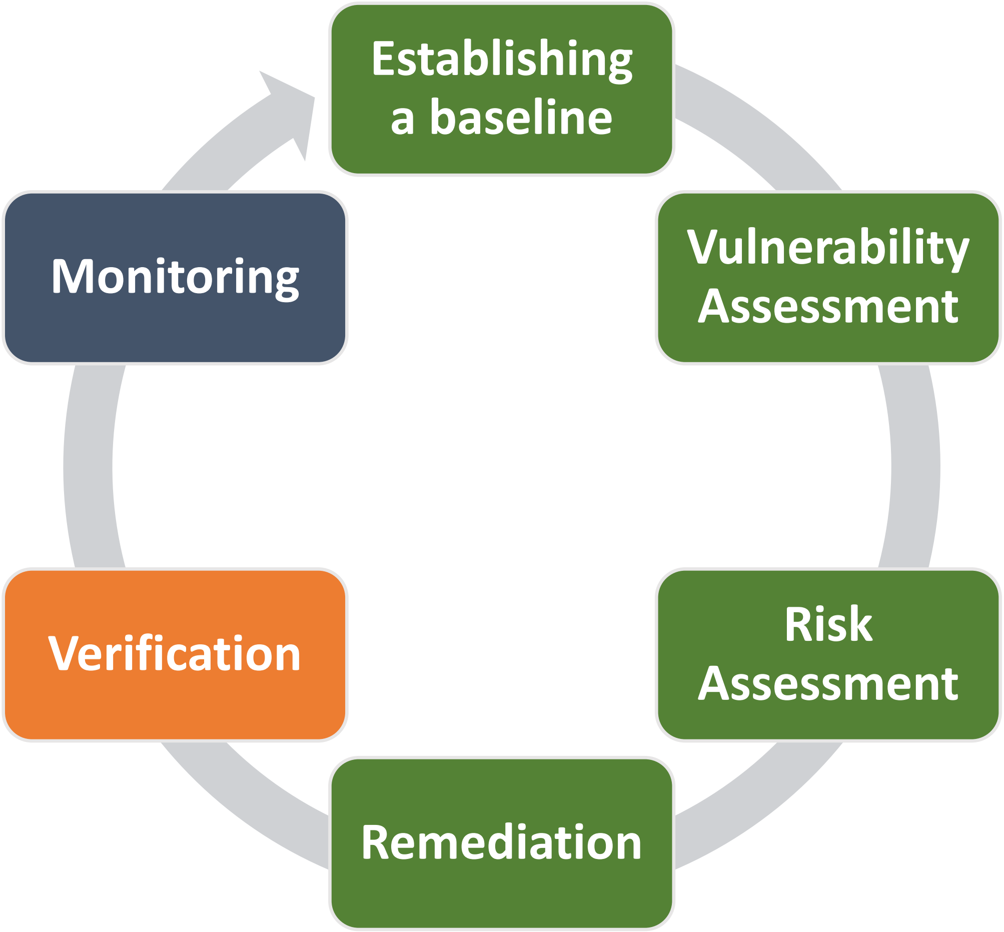
May reduce the impact of a successful exploitation

- 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



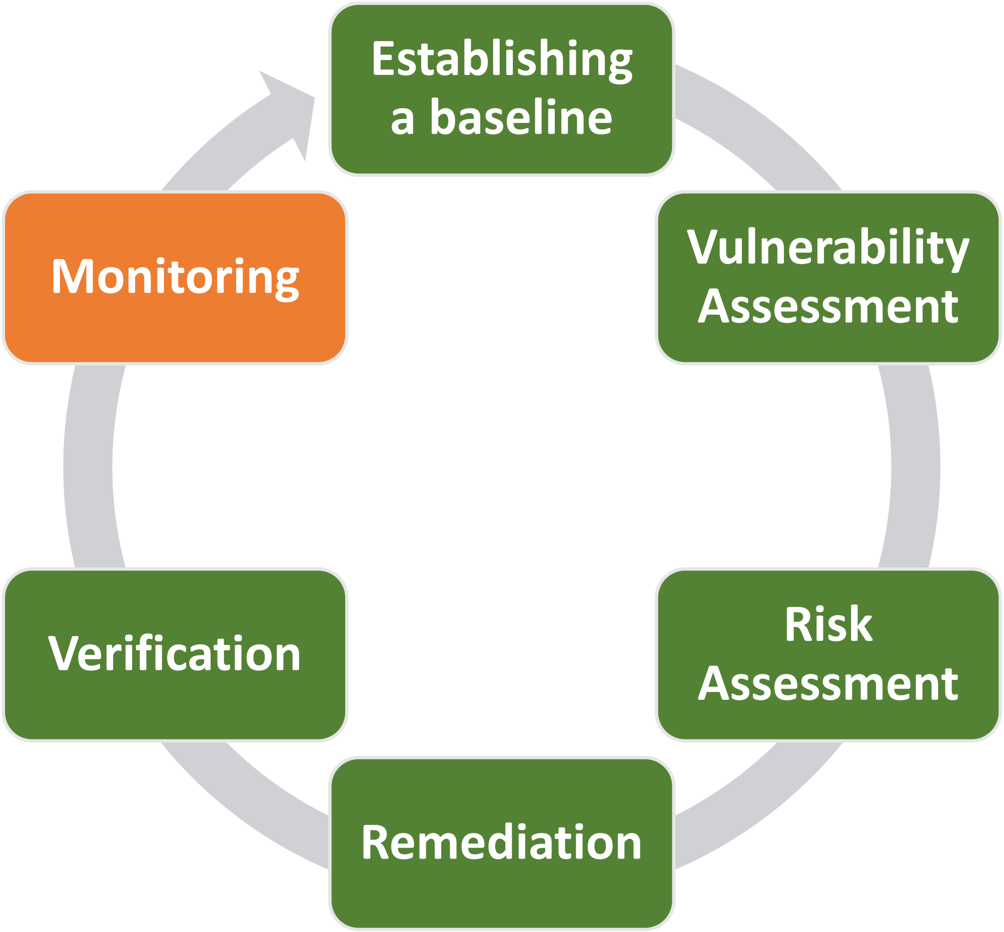
Verification

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

SCAP – Security Content Automation Protocol

Protocol to automatically assess the security status of a system

- Supported by all major system / OS providers

Some objectives:

- Track system status
- Identify vulnerabilities
- Monitor the system security policies
- Quantify the existing risks
- Common terminology across vendors and environments

Most common in environments with high policy compliance requirements

SCAP – Security Content Automation Protocol

Enumeration

- CVE: Common Vulnerabilities and Exposures
- CCE: Common Configuration Enumeration
- CPE: Common Platform Enumeration

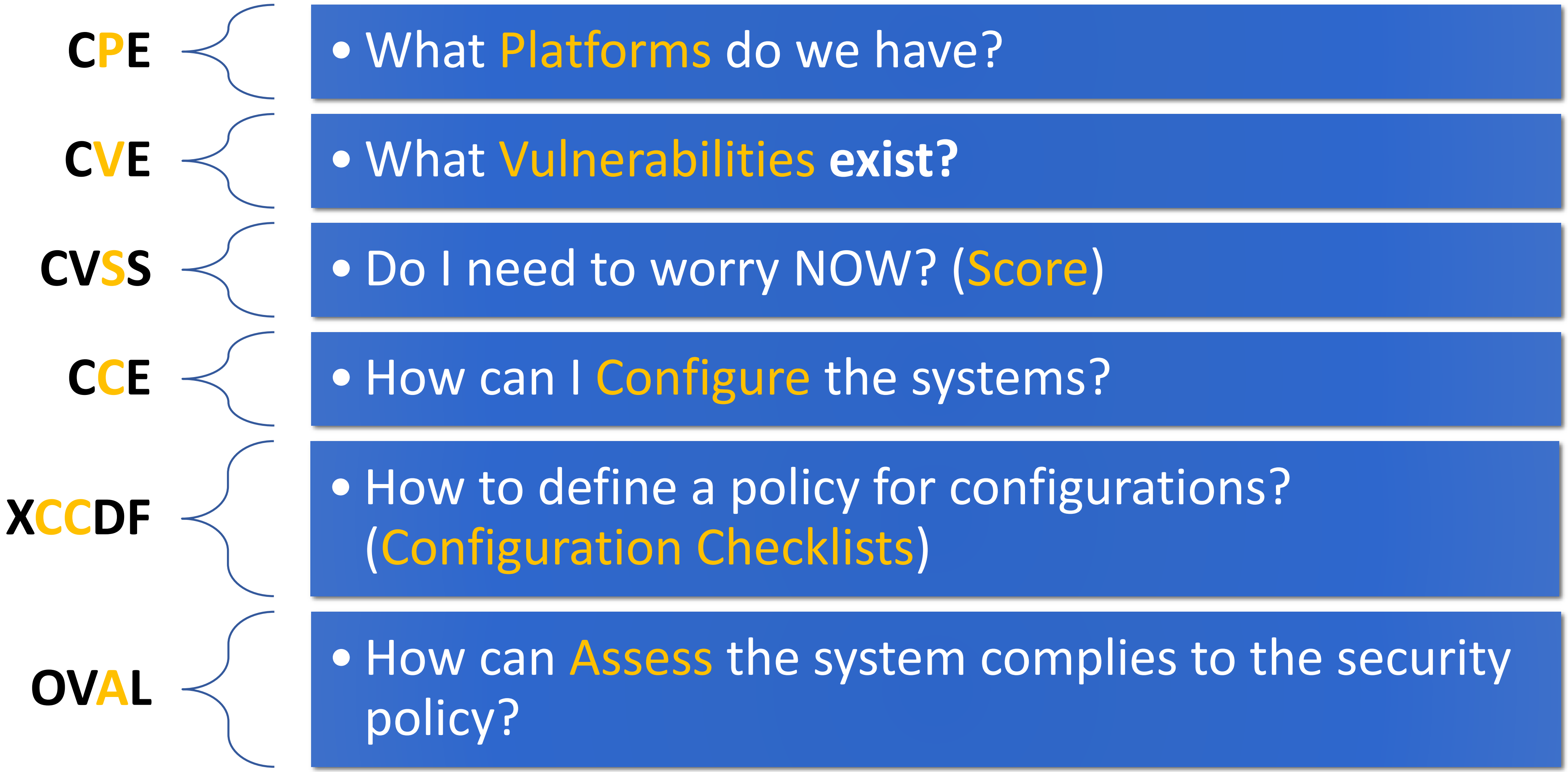
Languages

- OVAL: Open Vulnerability Assessment Language
- OCIL: Open Checklist Interactive Language
- XCCDF: eXtensible Configuration Checklist Description Format

Metrics

- CVSS: Common Vulnerability Scoring System

SCAP – Security Content Automation Protocol



CPE	What Platforms do we have?
CVE	What Vulnerabilities **exist?**
CVSS	Do I need to worry NOW? (Score)
CCE	How can I Configure the systems?
XCCDF	How to define a policy for configurations? (Configuration Checklists)
OVAL	How can Assess the system complies to the security policy?

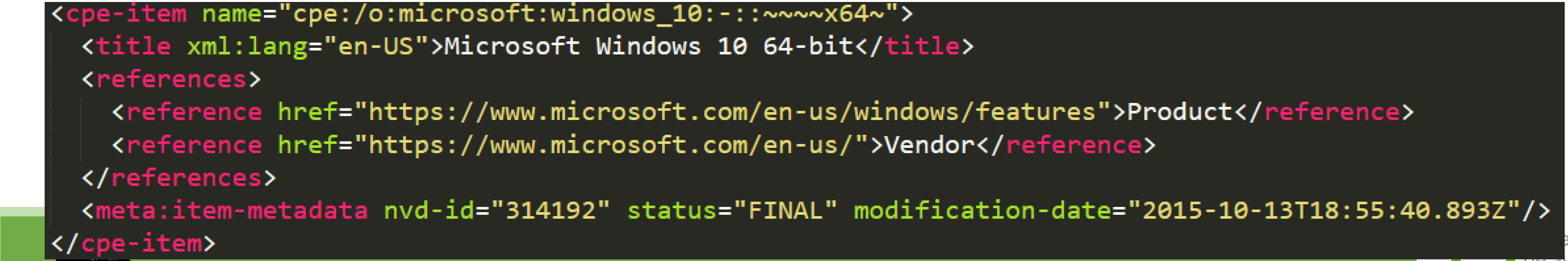
CPE – Common Platform Enumeration

Know what entities must be addressed in the scope of security

- Consider a company or campus: hundreds of computers with lots of software

XML based language to describe enumerate (software or firmware)

- Currently lists >550K entities
- Small amount of information: name, title, references, metadata (not a description)
- Format: `cpe:/{part}:{vendor}:{product}:{version}:{update}:{edition}:{language}`



```
<cpe-item name="cpe:/o:microsoft:windows_10:-::~~~~x64~">
  <title xml:lang="en-US">Microsoft Windows 10 64-bit</title>
  <references>
    <reference href="https://www.microsoft.com/en-us/windows/features">Product</reference>
    <reference href="https://www.microsoft.com/en-us/">Vendor</reference>
  </references>
  <meta:item-metadata nvd-id="314192" status="FINAL" modification-date="2015-10-13T18:55:40.893Z"/>
</cpe-item>
```

CCE – Common Configuration Enumeration

Clearly states the controls of an CPE

- That is, the configurations

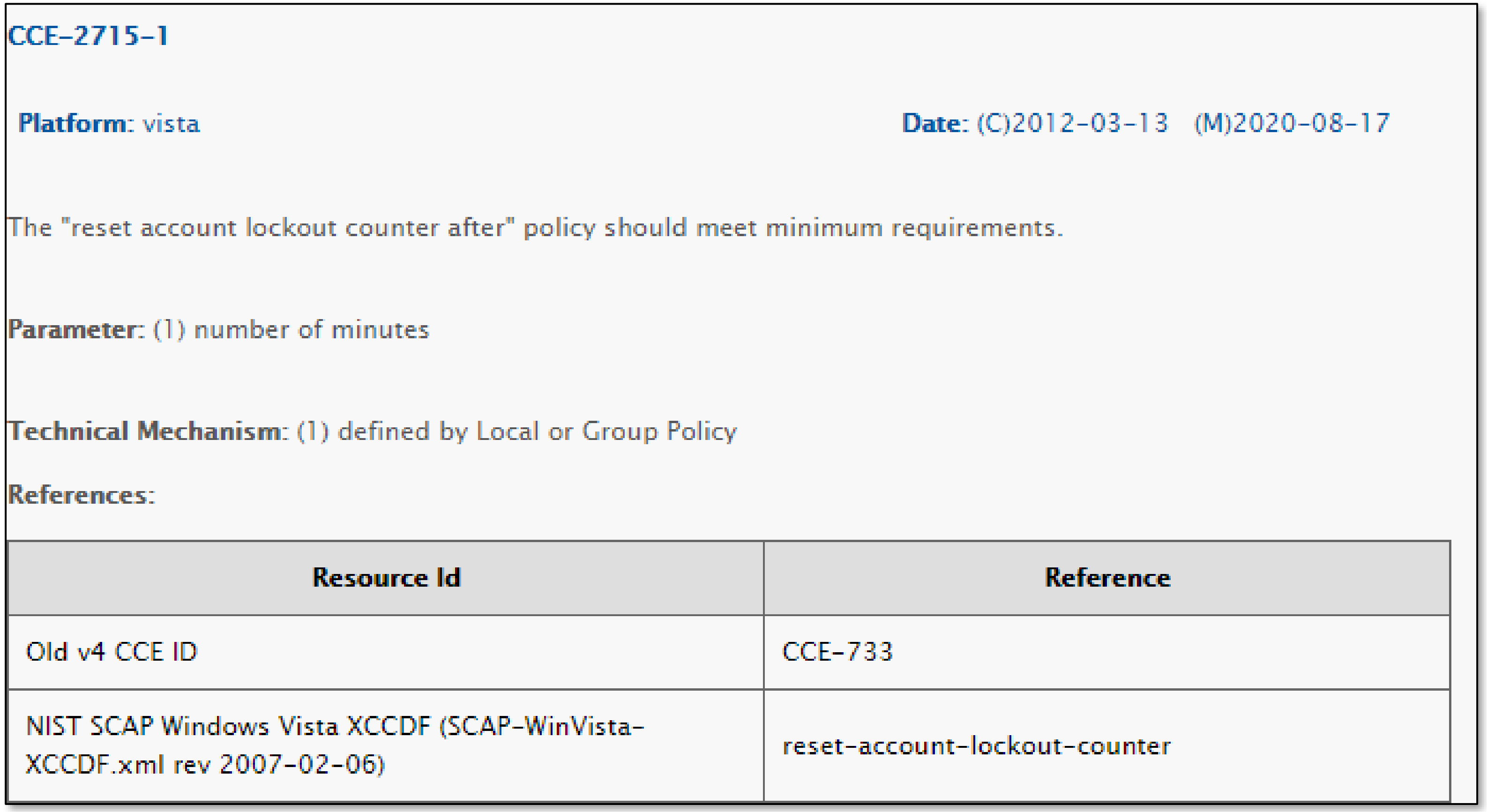
Publicly available in many cases, but not always.

- Managed by the vendor

Content

- **CCE Identifier Number** – "CCE-2715-1“
- **Description** – a humanly understandable description of the configuration issue
- **Conceptual Parameters** – parameters that would need to be specified in order to implement a CCE on a system
- **Associated Technical Mechanisms** – for any given configuration issue there may be one or more ways to implement the desired result
- **References** – pointers to the specific sections of the documents or tools in which the configuration issue is described in detail

CCE – Common Configuration Enumeration



CCE-2715-1

Platform: vista **Date**: (C)2012-03-13 (M)2020-08-17

The "reset account lockout counter after" policy should meet minimum requirements.

Parameter: (1) number of minutes

Technical Mechanism: (1) defined by Local or Group Policy

References:

Resource Id	Reference
Old v4 CCE ID	CCE-733
NIST SCAP Windows Vista XCCDF (SCAP-WinVista-XCCDF.xml rev 2007-02-06)	reset-account-lockout-counter

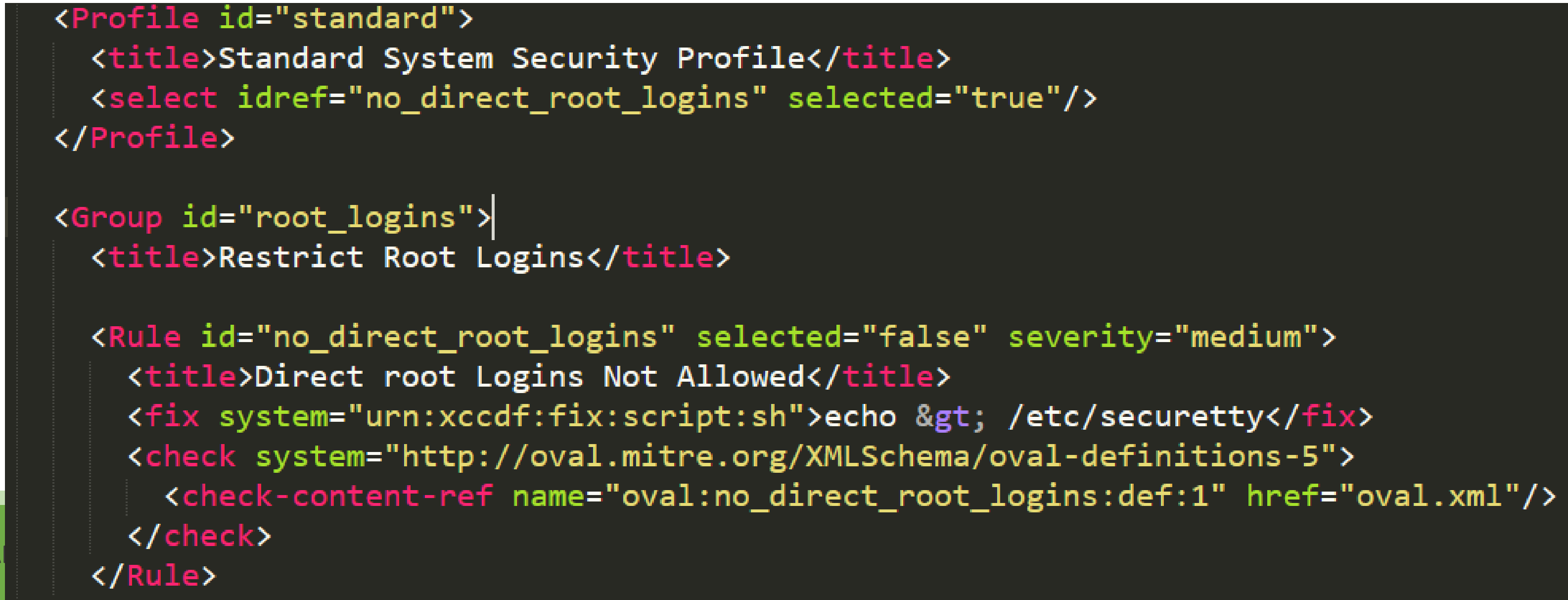
XCCDF - eXtensible Configuration Checklist Description Format

XML based language to define verifications and fixes associated to a profile

- Profile defines a set of policies and what needs to be verified
- Specific to a CPE (e.g. OS Distribution)

Fixes may include commands and validations

- Run scripts, call APIs
- Specific to a operating system



```
<Profile id="standard">
  <title>Standard System Security Profile</title>
  <select idref="no_direct_root_logins" selected="true"/>
</Profile>

<Group id="root_logins">
  <title>Restrict Root Logins</title>

  <Rule id="no_direct_root_logins" selected="false" severity="medium">
    <title>Direct root Logins Not Allowed</title>
    <fix system="urn:xccdf:fix:script:sh">echo &gt; /etc/securetty</fix>
    <check system="http://oval.mitre.org/XMLSchema/oval-definitions-5">
      <check-content-ref name="oval:no_direct_root_logins:def:1" href="oval.xml"/>
    </check>
  </Rule>
```

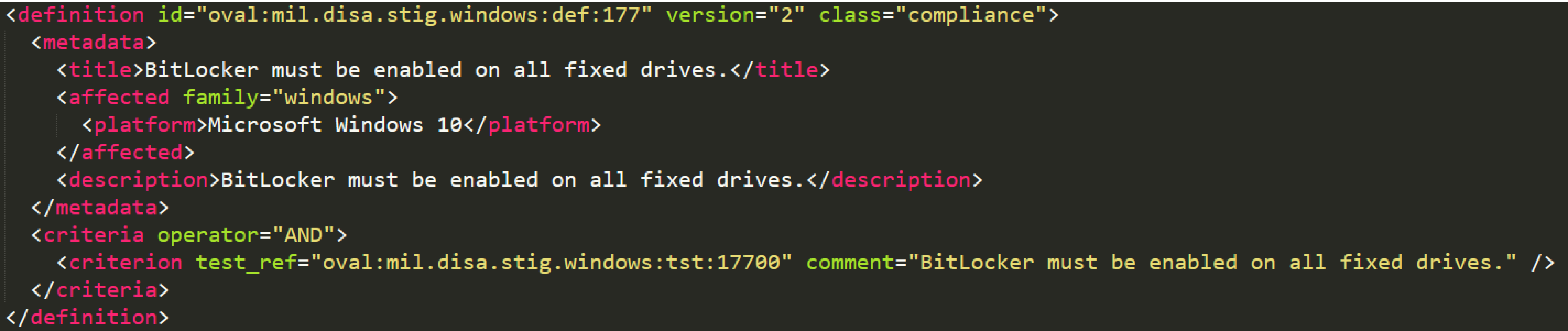
OVAL - Open Vulnerability And Assessment Language

XML based language with definitions to validate security controls

- Definitions are used by a XCCDF implementing a specific policy

Each definition states:

- What to assess: ex: The state of a CCE, or presence of a CPE
- How to assess: ex: How the state(s) is(are) checked
- How to report: ex: What message is provided



```xml
<definition id="oval:mil.disa.stig.windows:def:177" version="2" class="compliance">
  <metadata>
    <title>BitLocker must be enabled on all fixed drives.</title>
    <affected family="windows">
      <platform>Microsoft Windows 10</platform>
    </affected>
    <description>BitLocker must be enabled on all fixed drives.</description>
  </metadata>
  <criteria operator="AND">
    <criterion test_ref="oval:mil.disa.stig.windows:tst:17700" comment="BitLocker must be enabled on all fixed drives." />
  </criteria>
</definition>
```

SCAP related repositories

CVE: MITRE and NIST (NVD)

CVSS: Calculated by NIST

CPE: Provided by NIST

CCE: Provided by each software developer

XCCDF: MIL, GOV, HIPPA related entities imposing security requirements

OVAL: provided by software developers, other entities