

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

Takeaway *Research output is a decision - what to fix, what to defend, what to budget for.*

Vulnerability Assessment - Objective

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

Identify and categorise issues that may be exploited, or that constitute a risk to the normal operation of the entity

Takeaway *Assessment is analysis of your own entities against risk - neither compliance nor exploitation.*

Definitions: the primitives

Asset: anything of value to protect - data, systems, people, reputation, availability

Vulnerability: a weakness of an asset or control that can be exploited (code flaw, misconfiguration, missing control)

Threat: a potential cause of harm. Threat agent: who might do it, with what motive, opportunity and capability

Attack: use of a technique to exploit a vulnerability.

Exploit: the code or procedure that performs it

Exposure: an unmitigated vulnerability that a threat can reach (asset reachable from that vantage point)

Definitions: the primitives

Control / countermeasure: mechanism that prevents, detects or recovers; compensating control: an alternative that limits exposure

Severity: intrinsic property of the vulnerability.

Likelihood: probability of exploitation.

Impact: consequence

Risk = $f(\text{threat, likelihood, asset value, impact})$

Vulnerability severity is NOT risk: two systems with the same CVE can carry opposite risk

- Controls: Filter packets, connections

Takeaway *If asset, threat, exposure and impact are not separate ideas, risk collapses into a CVSS score.*

Assessment vs Audit

Audit: determines compliance with 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 improve systems
- Done before the audit, to identify any loopholes
- Done after the audit, to measure how effective the audit was

SANS Institute, "Scoping Security Assessments: A Project Management Approach" (white paper 33673), sans.org/white-papers/33673

Takeaway *Audit asks 'were the rules followed?'. Assessment asks 'are we actually exposed?'. You need both.*

Assessment vs Penetration Test

A penetration test focuses on infrastructure and systems, from an outside-in and inside-out perspective

- Outside: outside the domain (another 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 may it have reached
- What data/systems were impacted
- Was data exfiltrated?

Takeaway *A pentest proves one path exists; an assessment counts how many paths you have - and their cost.*

Why assess? (1) The organisational landscape

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

Takeaway *You cannot assess what you have not enumerated: asset inventory is the first assessment task.*

Why assess? (2) Defences alone are not enough

Standard defensive measures are not enough

- They help create and operate 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 deployed as perimeter control devices; visibility inside the network depends on deployment (micro-segmentation)
 - Inspects packets in the clear, or publicly visible metadata (ports, IP addresses, protocol), but struggles with encrypted traffic (TLS)
- **WAF:** Filter HTTP requests
 - Matches profiles of known attacks (deny list) or of allowed requests (allow list), but can be circumvented
- **IDS: Network- and Host-based Intrusion Detection Systems monitor network traffic or OS changes**
 - Matches profiles of known attacks, but can be circumvented
 - An IDS detects and alerts; it does not stop what already happened (blocking requires an inline IPS)

Takeaway *Firewalls, WAF and IDS answer the questions you expected; assessment asks what they will miss.*

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)

Too narrow: mimics a focused attack

- Cheap, fast, repeatable
- May miss easily found issues
 - Like hardening the bulletproof front door while leaving a glass window in the wall

Takeaway *Scope is a risk decision, not a technical one: too wide never ends, too narrow misleads.*

Limitations

Assessment is only valid at a given point in time

- Other vulnerabilities may exist before or after the assessment

The researcher must be aware of the latest vulnerabilities

- Risk of false negatives

Limited to the scope, location and methods used

- Different domains may have different firewall 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

Takeaway *Read the small print first: as of today, within this scope, by these methods.*

Before you probe: legal and ethical constraints (1/2)

Written authorisation before any testing: who, what systems, what techniques, what time window

Rules of Engagement (RoE): depth of testing, credentials provided, out-of-scope systems, escalation contact

Scope boundaries are physical and legal: a wireless signal leaves the building; cloud/SaaS/CDN may forbid testing

- Third-party services often require prior approval (pen-test gates, provider terms)

Data handling: minimise collection, no copying of personal data, secure storage and destruction (GDPR)

Before you probe: legal and ethical constraints (2/2)

Portuguese / EU context: unauthorised access is a criminal offence; act on contract, not on good intentions

When testing disrupts production: stop, notify immediately, agree a rollback path

Disclosure: how and when findings become known

- co-ordinate with the vendor;
- no public naming during remediation

If you find data or credentials accidentally: stop reading, report at once, document the incident

Takeaway *Authorisation and Rules of Engagement precede the first probe - an illegal finding is an incident.*

Types of assessment (company scale)

Active

Passive

External

Internal

Host-Based

Network

Application

Wireless

Takeaway *These types are not alternatives: a working programme runs several of them continuously.*

Four independent axes (why the 'types' overlap)

The eight types mix three different dimensions.

Describe an engagement as a coordinate instead:

- **Method:** active (probe the target) vs passive (observe what it says)
- **Vantage point:** external (no access) vs internal (some access, privileged or not)
- **Target scope:** network, host, wireless, application
- **Knowledge:** black box, gray box, white box (what the tester is given)
 - Example: "active, external, network-based, gray-box, credentialed for 3 host classes"

Each axis trades coverage, realism, disruption risk and credentials required

Benefit: no more arguing whether a test is 'host-based or internal' - it is both

Takeaway *Name the coordinate, not the type: active, external, network, gray-box. The argument ends there.*

Type: Active

Runs software to discover network hosts

- Send probes
- Checks information repositories

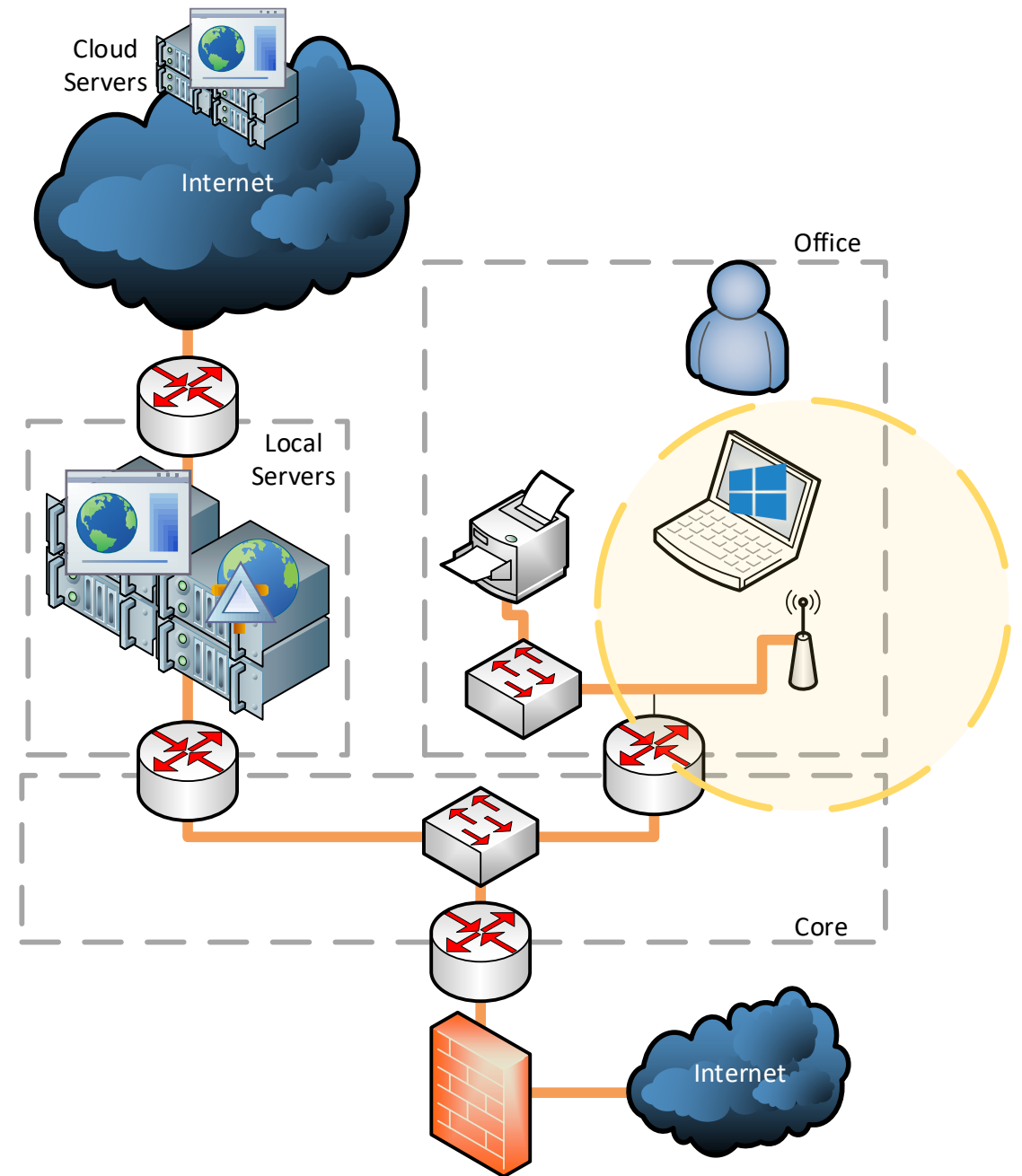
Runs tools to actively test software/systems

- Sends crafted arguments, payloads, packets
- Simulates attack techniques
- MiTM, DoS, etc...

May disrupt systems! Run in agreed windows, with rate limits, exclusions and a rollback plan

- Detection of vulnerability may have impact

Takeaway *Active probing trades certainty for production risk - schedule it like a change, with rollback.*



Type: Passive

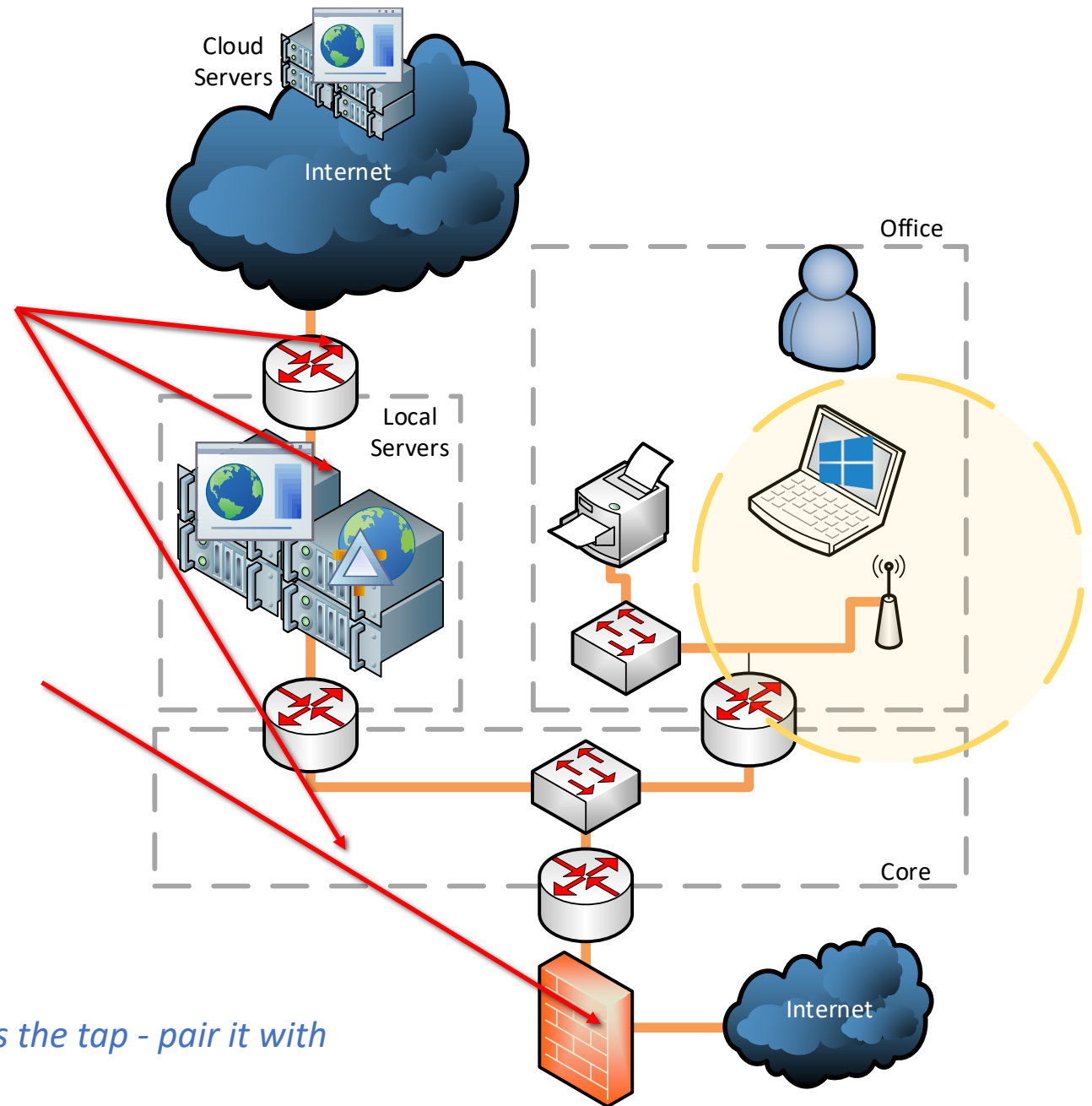
Runs software to eavesdrop on traffic

Observes logs and dumps

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

Minimal impact - but blind to what does not cross the tap (silent/patched hosts, encrypted payloads)

Takeaway *Passive is safe but silent about what never crosses the tap - pair it with credentialed checks.*



Type: External

Focus on the public exposition

- External attackers

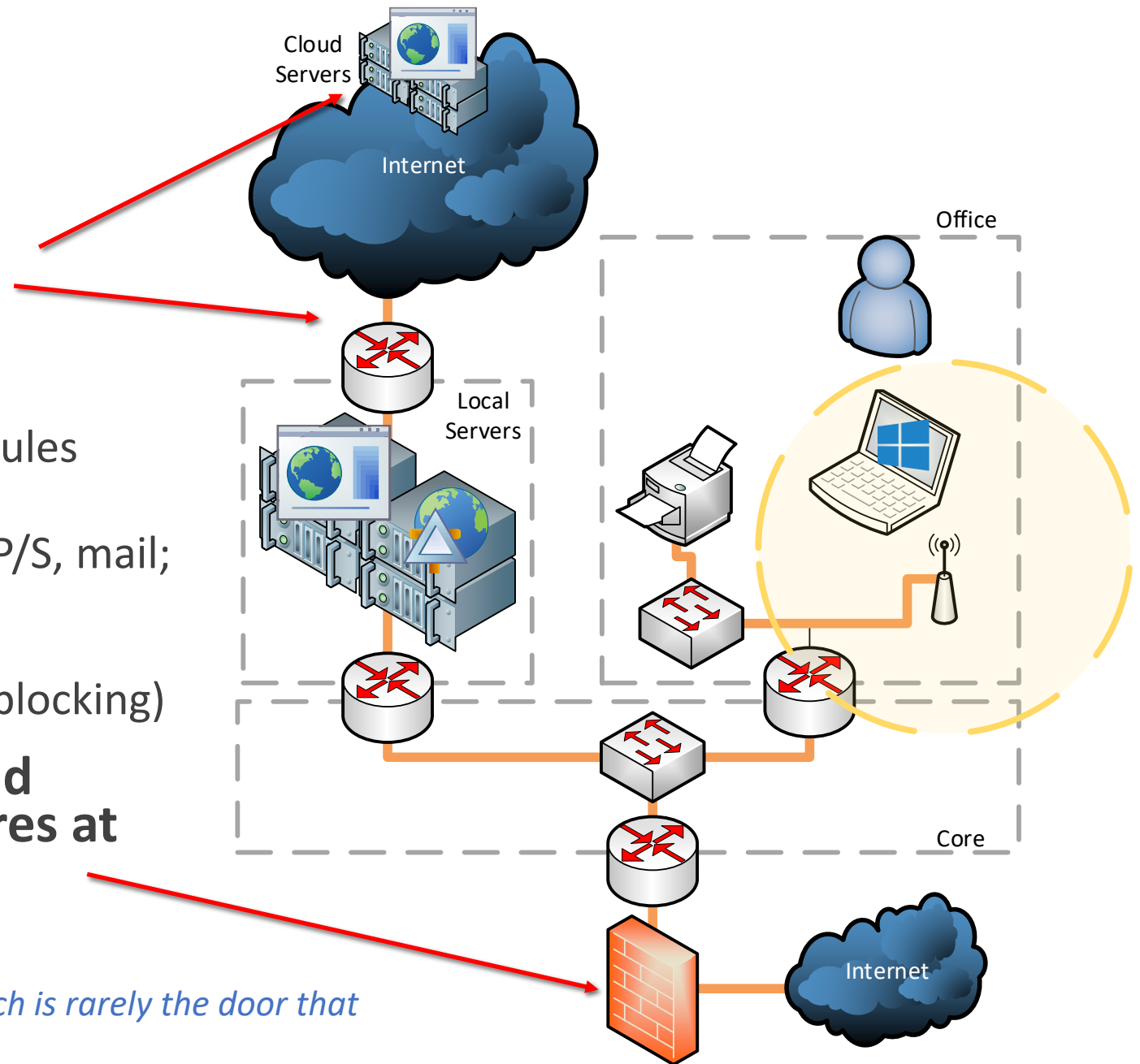
Targets:

- Publicly exposed routers and firewall rules
- Publicly available IP Ports
- Public services and records (DNS, HTTP/S, mail; zone transfers, SPF/DMARC)
- Information exposed to the public
- Security mechanisms (throttling, TLS, blocking)

It allows finding vulnerabilities and enables deploying countermeasures at the firewalls

- For assessment and exploitation

Takeaway *External shows your worst public door, which is rarely the door that gets you breached.*



Type: Internal

Run from inside the perimeter, with normal user credentials (or none)

Covers what the outside never sees

- admin panels, databases, printers, internal APIs, dev hosts

Reveals flat networks and lateral-movement paths

- SMB, RPC, shares, weak internal ACLs

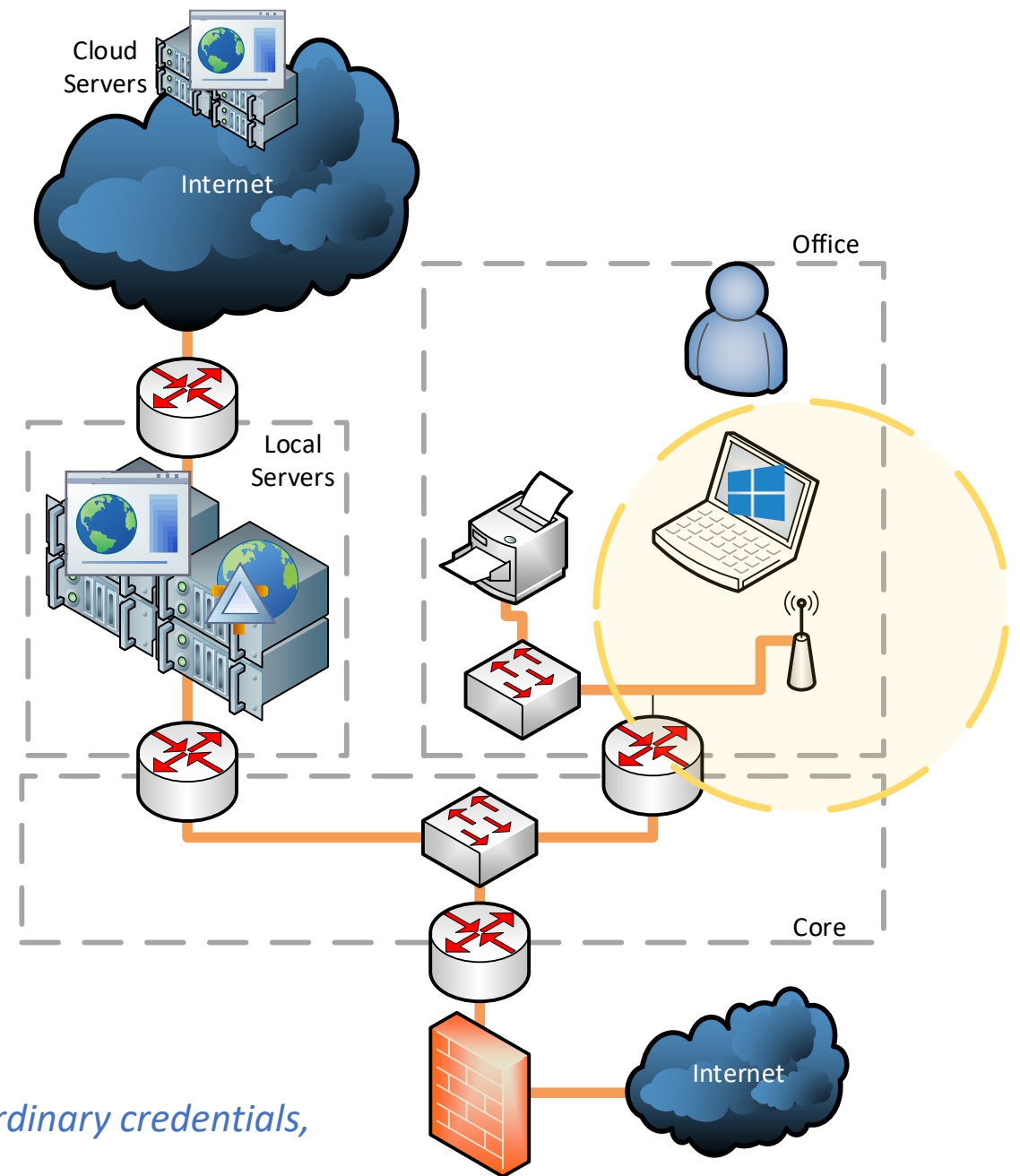
Most real incidents end here

- initial access outside, then expansion inside

Needs strict scope control

- internal systems usually hold production data

Takeaway *Most real incidents end here: inside the perimeter, with ordinary credentials, moving sideways.*



Type: Network

Focus on the communications of the network infrastructure

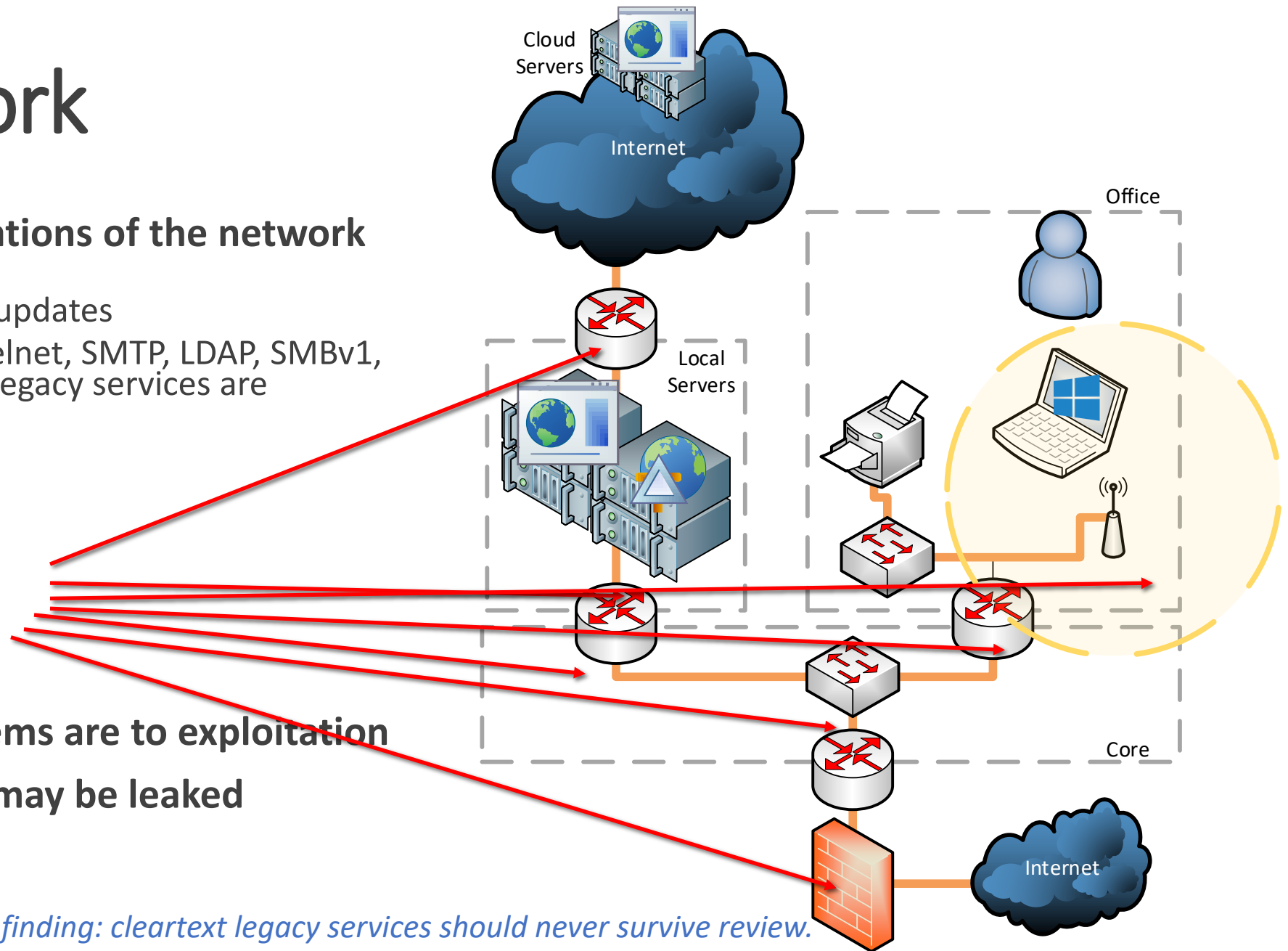
- Rules, misconfigurations, updates
- Individual services (FTP, Telnet, SMTP, LDAP, SMBv1, SNMP v1/v2c) - cleartext legacy services are themselves a finding

Targets:

- Communication links
- Networking Gear

Finds how exposed systems are to exploitation

Finds what information may be leaked



Takeaway *Reachability is itself a finding: cleartext legacy services should never survive review.*

Type: Wireless

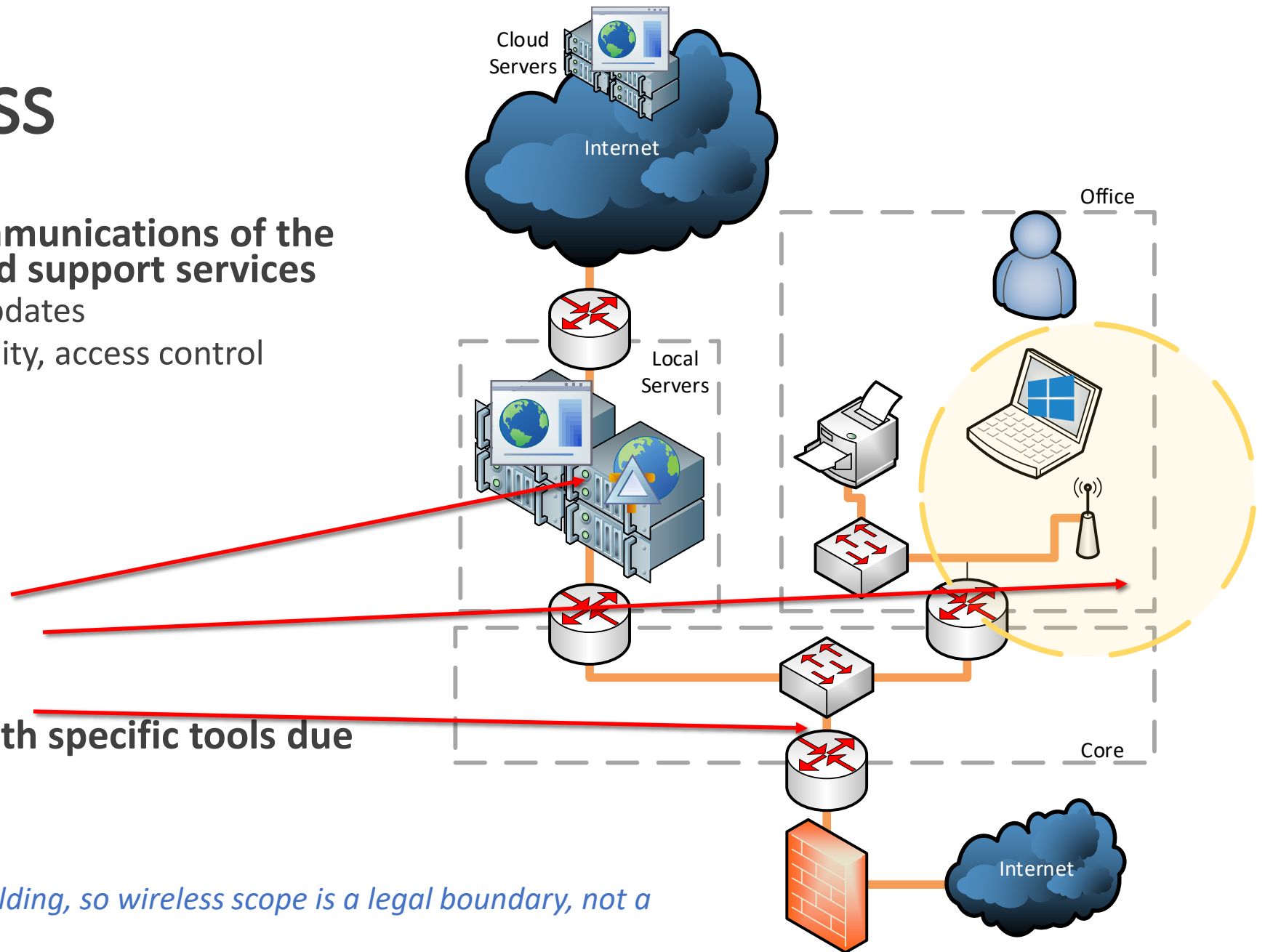
Focus on the wireless communications of the network infrastructure and support services

- 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



Takeaway *The signal leaves the building, so wireless scope is a legal boundary, not a technical one.*

Type: Application

Focus on a single application

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

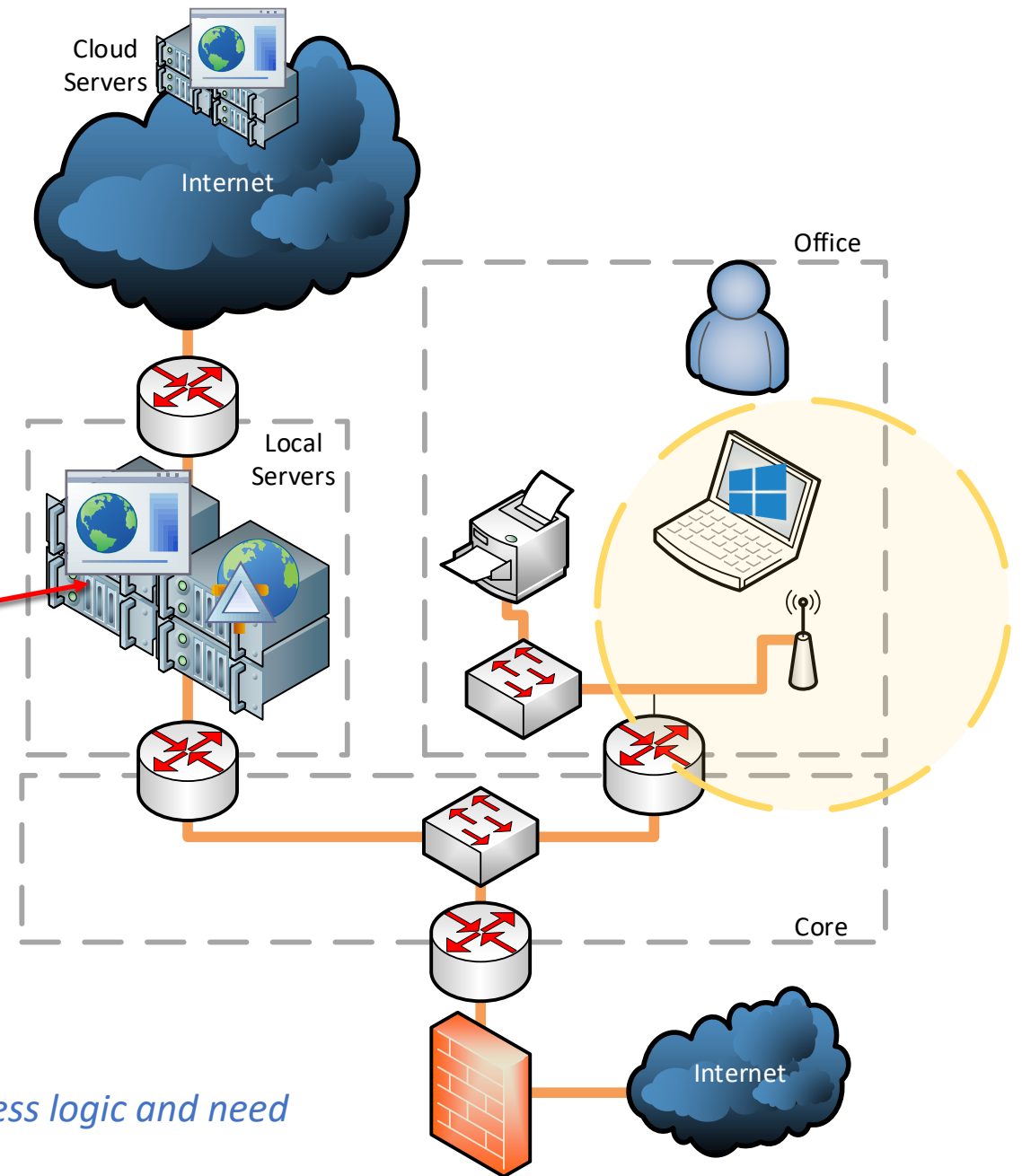
Targets:

- Application
- Service

Finds software vulnerabilities in the targeted application

- Bugs or flaws

Takeaway *Scanners find the shell; application flaws live in the business logic and need human reasoning.*



What the classic taxonomy misses (1/2)

Cloud: misconfiguration and posture (CSPM), identity and permissions, exposed storage, image/AMI hygiene

Containers and Kubernetes: vulnerable images, cluster RBAC, secrets in env/files, runtime policy

CI/CD and build chain: build-agent compromise, dependency confusion, unsigned artefacts, provenance

APIs: broken object-level authorisation, mass assignment, shadow APIs

- OWASP API Top 10

What the classic taxonomy misses (2/2)

Mobile and MDM: app hardening, enrolment and certificate distribution, jailbreak/root detection

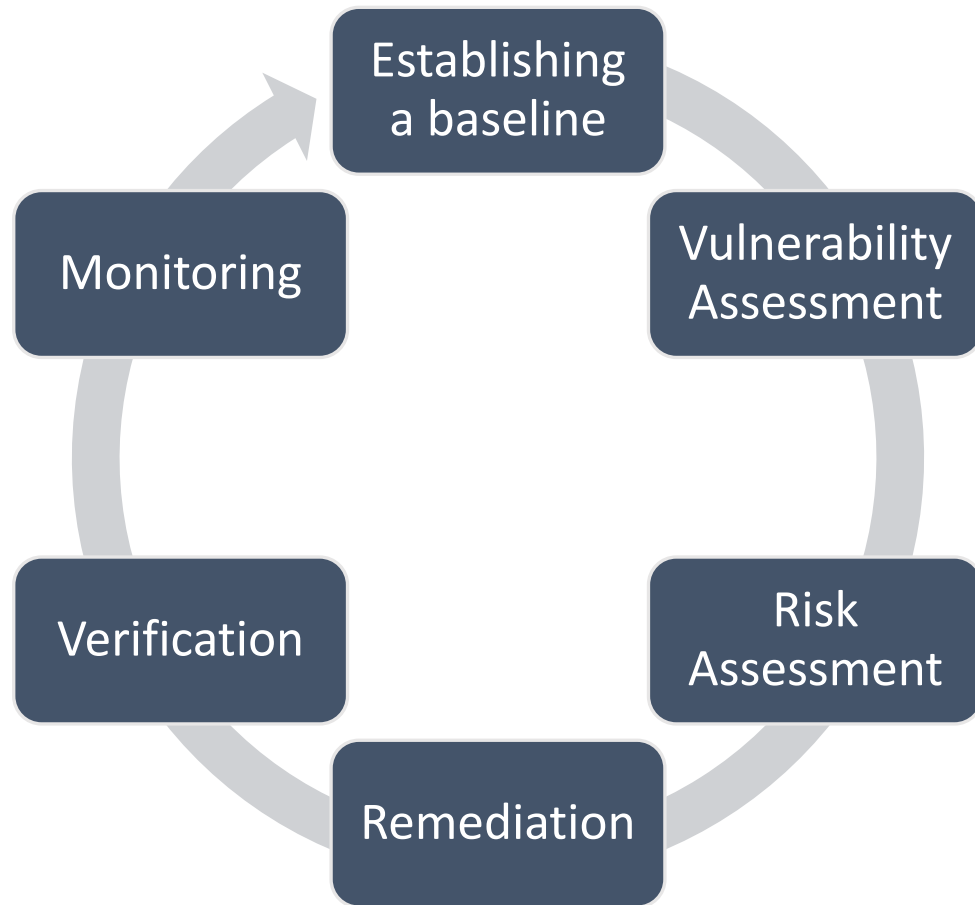
IoT and OT/ICS: long-lived, unpatchable systems - active testing often prohibited, so passive + lab replication

Continuous attack-surface management (ASM/CAASM/EASM): inventory first, point-in-time scanning second

Same life cycle applies to all of them - only the tools and constraints change

Takeaway *New surfaces change the tools, not the life cycle: inventory, assess, remediate, verify, monitor.*

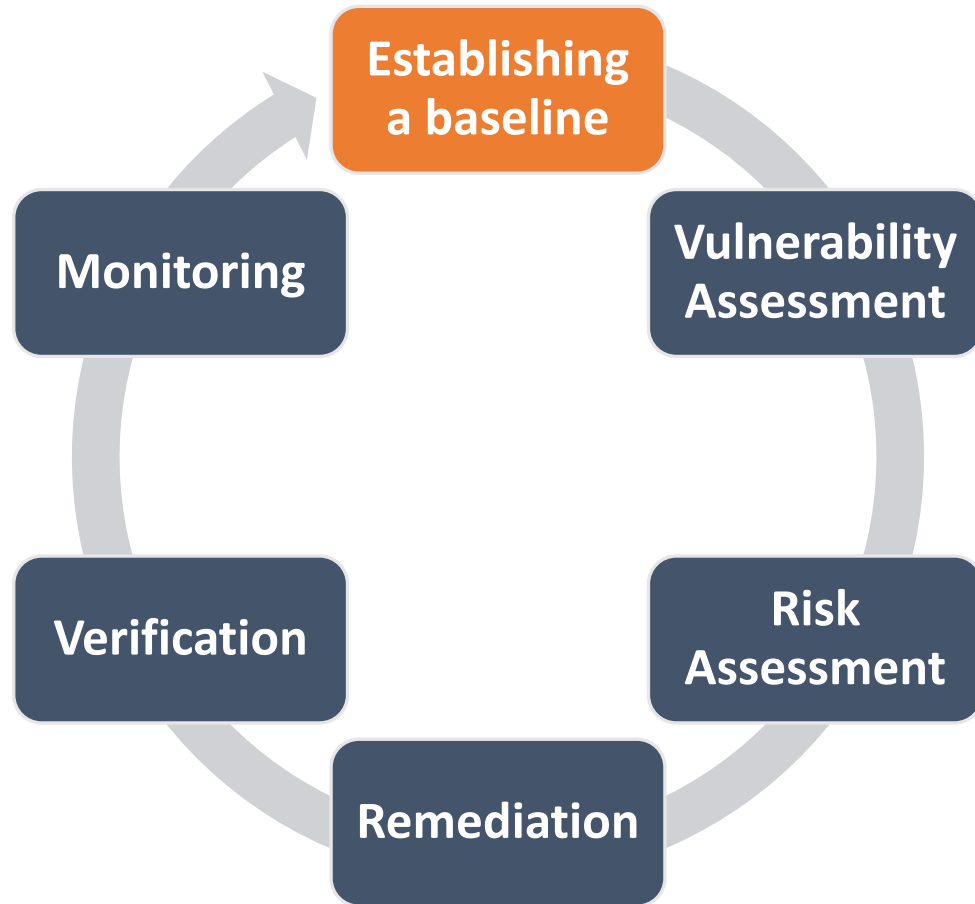
Vulnerability Management Life Cycle



Takeaway *Assessment is stage 2 of six. Skip the other five and findings become paperwork.*

Vulnerability Management Life Cycle - 1/6

Establishing a baseline



Establish a Baseline

Select the assets to be assessed and define their priorities

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

Characterize the systems/software state

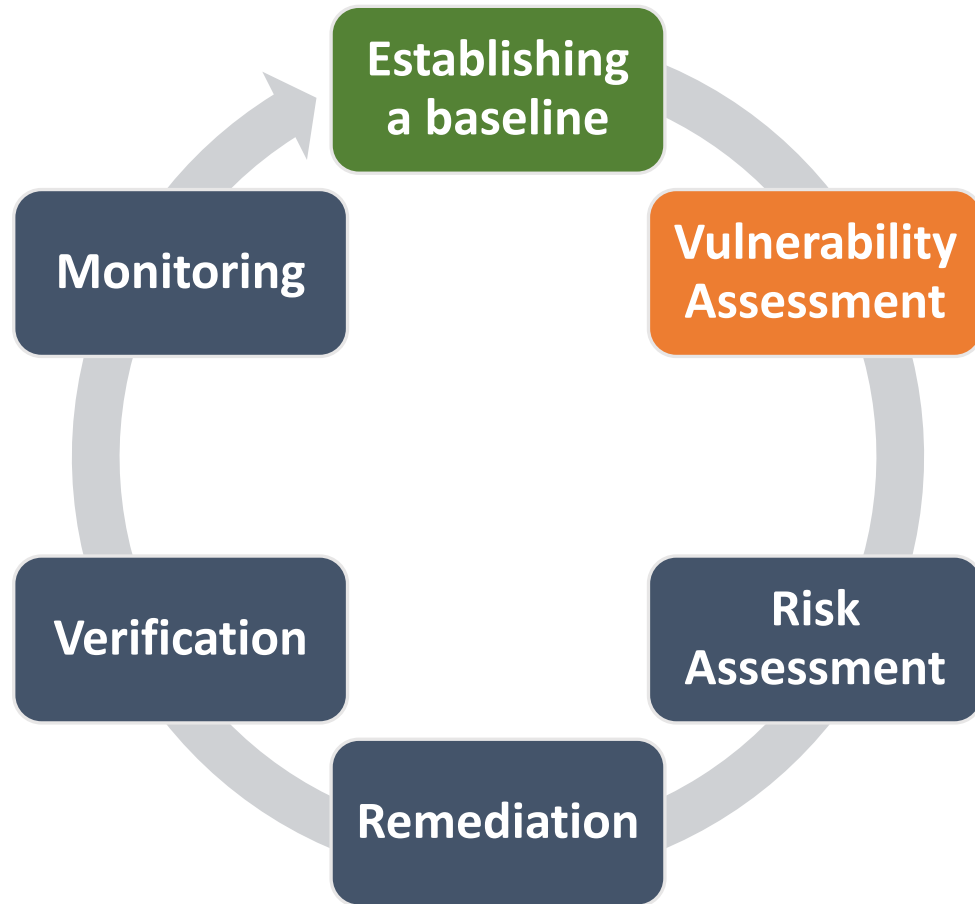
Determine what is known and what must be assessed

- Known vulnerabilities may be ignored from the assessment

Takeaway *Decide what you will not scan, and why - and put an owner on every asset you do scan.*

Vulnerability Management Life Cycle - 2/6

Vulnerability Assessment



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

An assessment usually does not exploit the vulnerability or build an exploit chain

- It's not a penetration test

Takeaway *Report finding, affected entities and the fix; resist the temptation to exploit it yourself.*

Assessment Methods

Close to software testing, but with a focus on security-related impact

- Extensively studied in the Robust Software course

Highly dependent on the scope of the assessment

- Application: SAST, DAST, IAST, SCA (software composition analysis), fuzzing, API testing
- Network entity: Protocol, message, authentication, authorization analysis
- Processes/Companies: OSINT, Social Engineering

Takeaway *Method follows scope: code, network, people. Choose the technique that can actually see the risk.*

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

Aims to mimic assessments from outside attackers

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

Limits what can be found (reduced coverage)

- Compensating controls (e.g. a firewall) mask the vulnerability -> false negatives: it is still present

Takeaway *Black box is realistic but shallow - its silence is not evidence of absence.*

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

- Knowledge available to the tester is maximal; the vantage point is chosen separately
- 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

Takeaway *White box finds everything, including what no attacker could ever reach: price and triage it.*

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

Takeaway *Credentials are the cheapest coverage you can ask for; grey box is the pragmatic default.*

How good is an assessment? Coverage and error

False positive: reported but not real - wastes remediation effort and erodes trust

False negative: real but not reported - the risk you never see; it is invisible by construction

Coverage: which assets, credentials, layers and attack paths were actually examined

- An assessment over 40 % of the inventory cannot support a statement about the other 60 %
- Credentialed scans raise coverage and cut false positives (patch/configuration state is read, not inferred)

Black box: probably more realistic, low coverage.

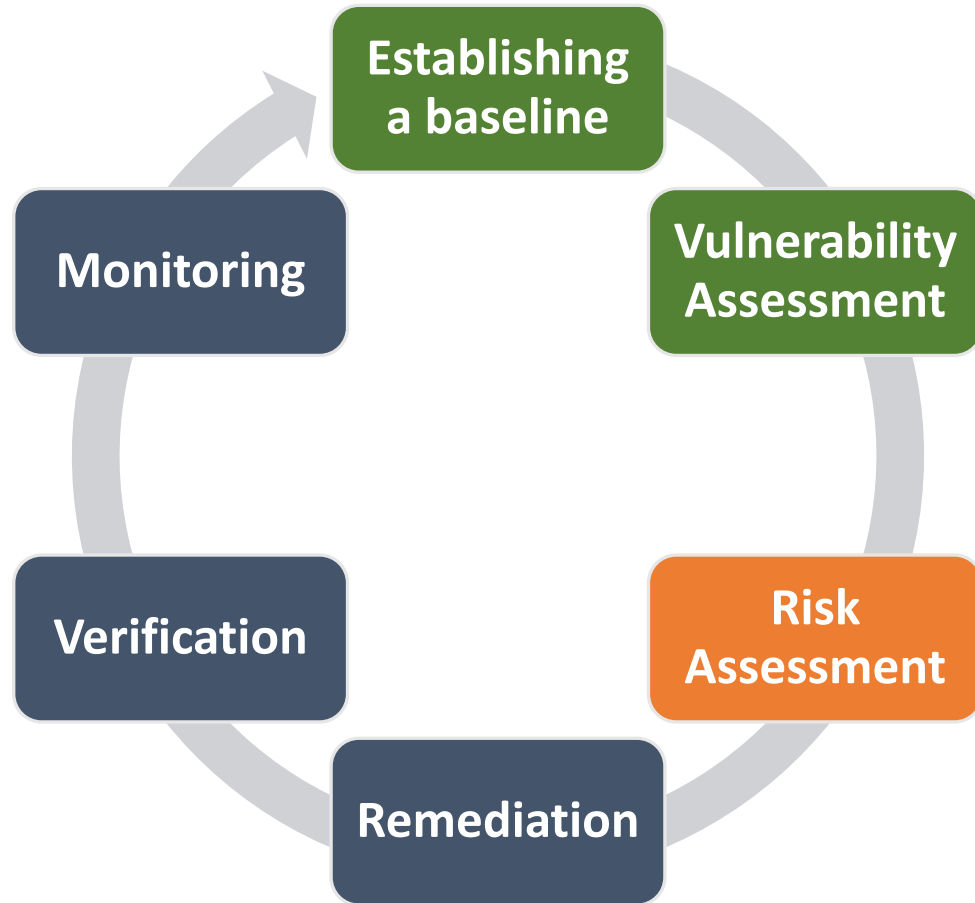
White box: high coverage, over-states what an attacker can do

Ask before any report: what could not be looked at, and why?

Takeaway *Quality is coverage plus false-positive/false-negative history - ask what could not be examined.*

Vulnerability Management Life Cycle - 3/6

Risk Assessment



Risk Assessment

Company takes in consideration the report and assess the risk

- For every asset with vulnerabilities
- Assigns risk levels (typically 3-4): risk combines likelihood and impact, not just vulnerability severity

Risk assessment may take all vulnerabilities found into consideration

- Individual vulnerabilities may be combined into an exploit chain with higher impact

Takeaway *Same CVE, opposite risk: this is where your context, assets and exploit chains are priced in.*

Documentation

Researchers should carefully document assessments

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

To understand how the vulnerability was exploited/what the impact may be

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

Takeaway *Document rationale, evidence and the fix; the tone of the report is part of the deliverable.*

Anatomy of a finding (1/2)

ID and title

Risk level and the rationale (CVSS vector + exposure + asset value)

Affected assets: hostname/IP, CPE, owner, environment

- production, test, development

Description: the weakness itself, linked to a CWE so similar findings aggregate

Evidence: the commands/requests and their output that reproduce it

- Sanitized Output!

Anatomy of a finding

Impact: what an attacker gains, and the preconditions they need first

Remediation information

- The fix
- Any workaround
- Dependencies
- The change window required

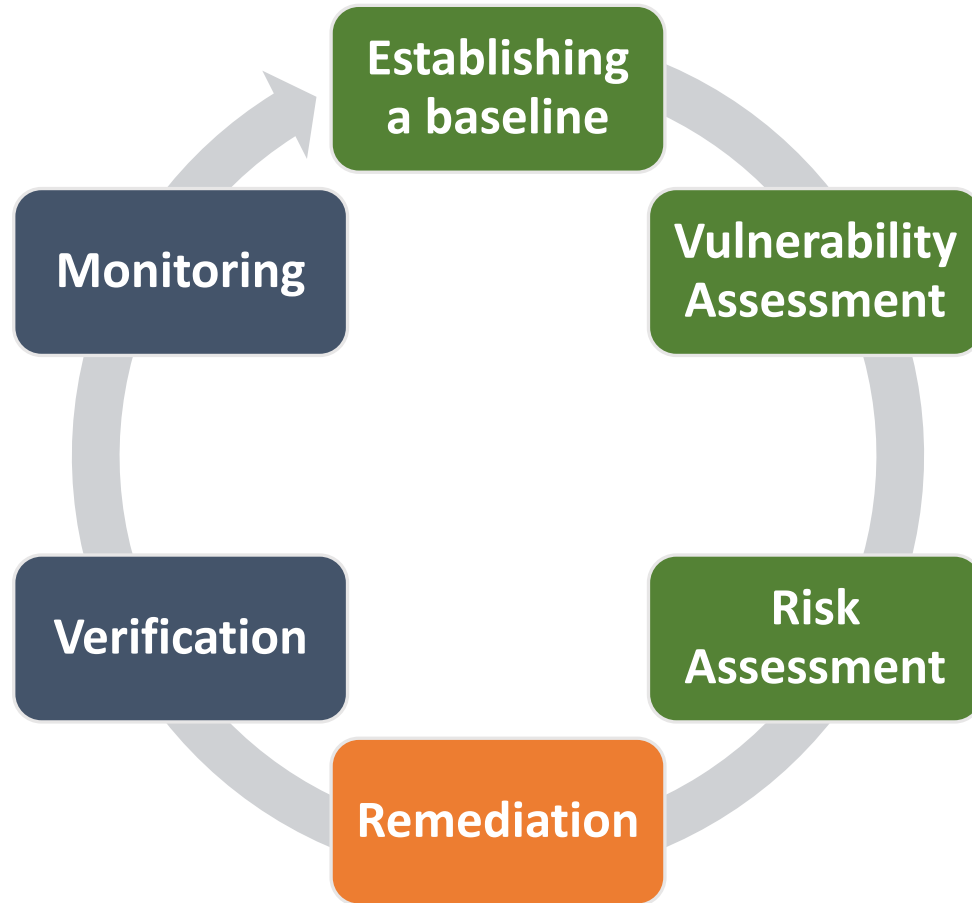
Status: open / pending retest / closed, with retest date and who verified

Write for a decision-maker: state the impact, not only the score

Takeaway *A finding you cannot re-derive or re-test is a rumour - evidence, owner and status make it real.*

Vulnerability Management Life Cycle - 4/6

Remediation



Remediation

The company implements measures to increase the security of its assets

May fix the vulnerability

- Correct software bugs or flaws, update it, change configs
- 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 failover

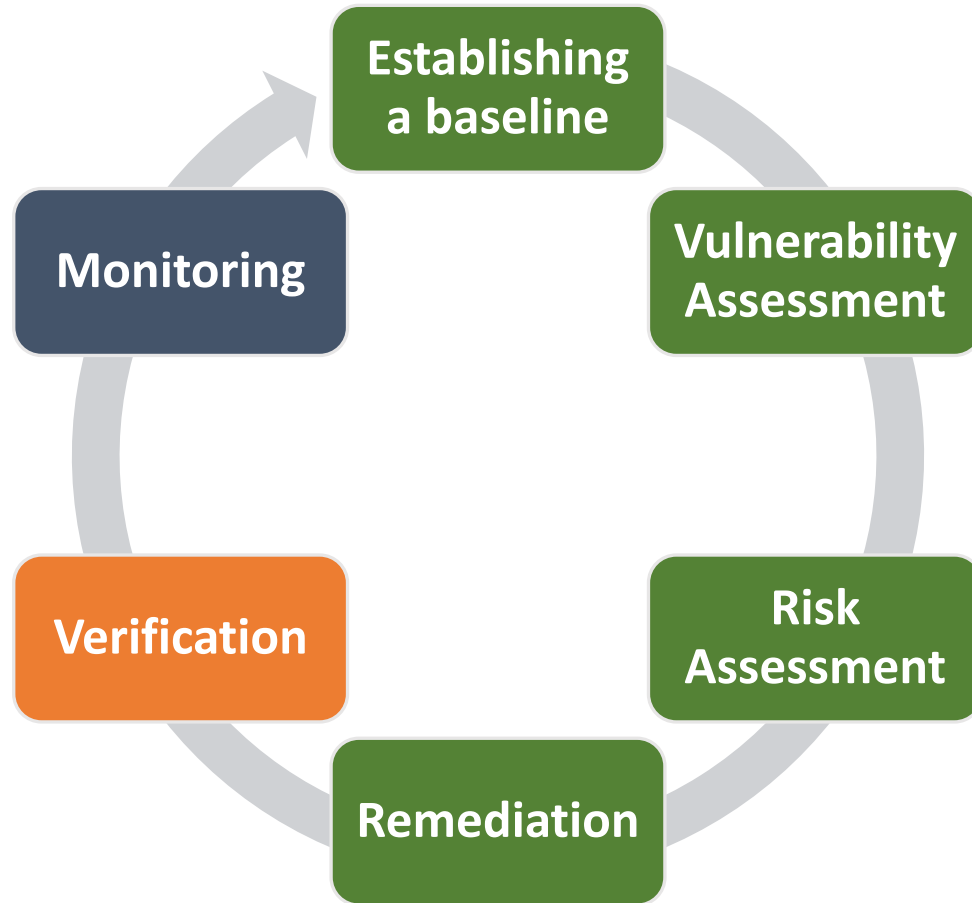
May increase the cost of exploiting the vulnerability

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

Takeaway *If you cannot patch, buy time: isolate, add redundancy, raise the cost of exploitation.*

Vulnerability Management Life Cycle - 5/6

Verification



Verification

Verifies the effectiveness of the remediation

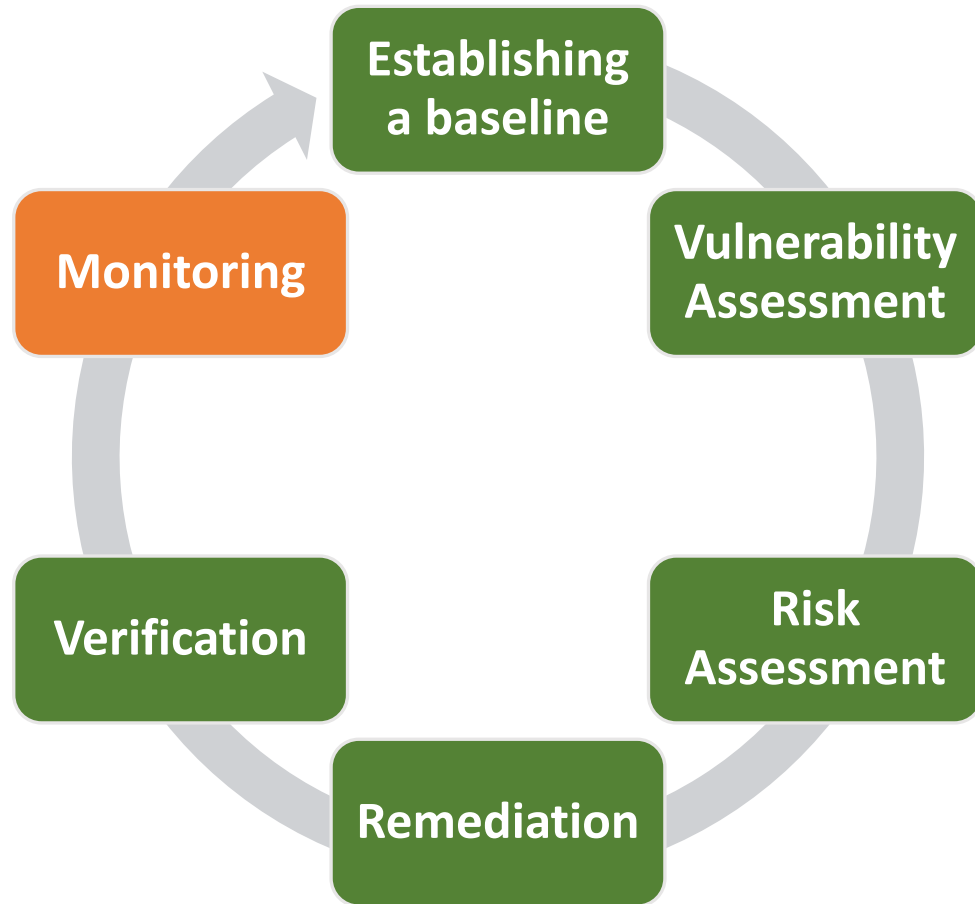
Involves assessing the existence and risk of the vulnerabilities found

- Using the same scope!
- The risk may remain similar if the vulnerability is exploited from another perspective
 - E.g. External vs Internal actor

Takeaway *Retest at the same scope and vantage point, or the fix may only have moved the hole.*

Vulnerability Management Life Cycle - 6/6

Monitoring



Monitoring

Deploys mechanisms to detect attempts to exploit the vulnerability

- May consider variations

Involves configuring firewalls, log-analysis systems, IDS (network- and host-based) and profilers

Takeaway *Assume the vulnerability is being tried right now - detection is what closes the loop.*

Life cycle: exit criteria and metrics

Baseline is done when:

- inventory agreed
- There is one owner per asset
- Risk tier assigned
- Scan windows and credentials agreed

Assessment: coverage of in-scope inventory (%), credential coverage, age of the last scan

Risk assessment: every finding has an owner, a risk level and a target date

Remediation: mean time to remediate per risk level; SLA breaches; fixes vs temporary workarounds

Verification: retest pass rate; reopened-case rate (measures whether fixes actually hold)

Monitoring: recurrence rate, time to detect, share of assets sending telemetry

SCAP – Security Content Automation Protocol

A suite of interoperable specifications (and a process) to automate the assessment of a system's security status

- Adoption driven mainly by public-sector procurement; vendor coverage is uneven (strongest on servers/OS configuration)

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

Takeaway *SCAP makes results comparable and automatable; check vendor coverage before you commit to it.*

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 (FIRST; v3.1 still deployed, v4.0/v4.1 current)

Takeaway *Enumeration, languages, metrics: know which SCAP piece a vendor claim actually refers to.*

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?

Takeaway *SCAP only pays off through content: know whose definitions your engine actually consumes.*

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

Naming convention to describe and enumerate platforms (applications, operating systems and hardware)

- The NVD CPE list currently holds over 1 million platform names
- Small amount of information: name, title, references, metadata (not a description)
- Format (CPE 2.3):

```
<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>
```

<language>:<sw_editi
S, h: hardware}. The
everything else.

CCE – Common Configuration Enumeration

Clearly identifies the configuration issues of a platform (CPE)

- That is, the configurations

Publicly available in many cases, but not always.

- Maintained by the content author (e.g. OS vendor); the MITRE/NIST CCE dictionary is no longer expanded - CCE IDs survive as legacy identifiers inside vendor and checklist content

Content

- **CCE Identifier Number** - “CCE-2715-1”
- **Description** - a human-readable 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

Takeaway *Configuration findings need stable IDs; the CCE dictionary is legacy, the practice is not.*

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 with 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 an operating system

Takeaway *XCCDF turns policy into something executable:*

profile, rule, fix - one platform at a time.

```
<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 referenced by the XCCDF rules that implement 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

```
<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">
    <criteria test_ref="oval:mil.disa.stig.windows:tst:17700" comment="BitLocker must be enabled on all fixed drives." />
  </criteria>
</definition>
```

Takeaway *OVAL answers 'is it true on this host?'; XCCDF decides whether that matters for your policy.*

OCIL - Open Checklist Interactive Language

For checks that cannot be decided automatically: a question, the expected answer and evaluation logic

Structured as a questionnaire: questions, answer types, decision logic, outcome

- Outcomes: pass / fail / review

Complements XCCDF and OVAL: OVAL decides what is machine-observable; OCIL handles everything else

Used in National Checklist content for procedural and interview-style controls

- Example: "is multi-factor authentication enforced for all administrators?" with no reliable artefact to query
- Supported by SCAP engines (e.g. OpenSCAP can evaluate OCIL questionnaires and produce results)

Practical value: it makes human verification auditable instead of anecdotal

Takeaway *Not everything is automatable - OCIL makes human verification auditable instead of anecdotal.*

SCAP related repositories

CVE: IDs assigned by CNAs (vendors, projects, research groups);

- MITRE is one CNA and a CVE Listing partner;
- Programme sponsored by CISA;
- NVD/NIST aggregates and enriches

CVSS: standard owned and maintained by FIRST.org; NVD publishes scores for some CVEs, but no longer enriches every one (selective triage since 2025/26)

CPE: Provided by NIST

CCE: legacy identifiers carried by vendor/checklist content; the dictionary itself is no longer maintained

XCCDF: military, government and regulated entities (e.g. HIPAA) imposing security requirements

OVAL: provided by OS vendors (Red Hat, SUSE, Debian, Ubuntu) and commercial vendors, via MITRE

Takeaway *Trust the source: who assigns, who enriches, how fresh. Write it into the procurement.*

Vulnerability data in 2026: what changed

CVE assignment is decentralised

NVD stopped universal enrichment

- 2024 funding gap, selective triage 2025/26
- Many CVEs now arrive without CVSS or CPE
- Prioritisation cannot be outsourced to NVD - use CNA data, vendor advisories, KEV and EPSS

CPE is the bridge to SBOM matching; the CCE dictionary is no longer maintained (legacy IDs only)

SCAP baseline is 1.3 (SP 800-126r3 + 126Ar3)

- SP 800-126r4 and 126Ar4 (SCAP 1.4) are initial public drafts (Dec 2025)

OSCAL is the emerging representation for machine-readable controls - watch it

Ask any supplier: which feed, how fresh, who enriches it, and what happens when it breaks?

Use Case: one finding, end to end

Asset: mail server, CPE cpe:2.3:a:vendor:product:1.2.3:*:*:*:*:*:*

CVE-2024-XXXX affects that version: the vendor CNA publishes CVSS v3.1 9.8 (AV:N/AC:L/PR:N/UI:N/...)

- KEV lists it; EPSS percentile is high -> escalate above the quarterly queue

Baseline content: an XCCDF rule references an OVAL definition -> automated check across every instance

Report: affected hosts, evidence, risk (internet-facing, production mail), fix, workaround, owner, SLA

Verification: same scope, same tool version, open until evidences shows it is fixed

Monitoring: periodic telemetry confirms all instances report

This closes the loop: CPE -> CVE -> CVSS -> KEV/EPSS -> XCCDF/OVAL -> remediation -> verification

Takeaway *The loop is always the same: identify, score, prioritise, fix, verify, watch - in that order.*

Key takeaways

An assessment answers "what is exposed?": it is neither an audit nor a penetration test

- Results are valid at a point in time, while coverage and false negatives bound their quality

Define scope and the four axes (method, vantage point, target, knowledge) before anything else

Never confuse vulnerability severity with organisational risk

Remediation without verification is a hope, not a control

Standards make results comparable and automatable

Priorities move daily: KEV, EPSS, exposure and inventory change

- So monitoring is part of assessment

Get authorisation in writing and document the Rules of Engagement