@lizfrenz /git

STARTING W. OWASP VULNERABILITY MANAGEMENT GUIDE (OVMG)

Vulnerability Management is:

A **continuous** risks-reducing governance of information technology relevant to software and infrastructure.

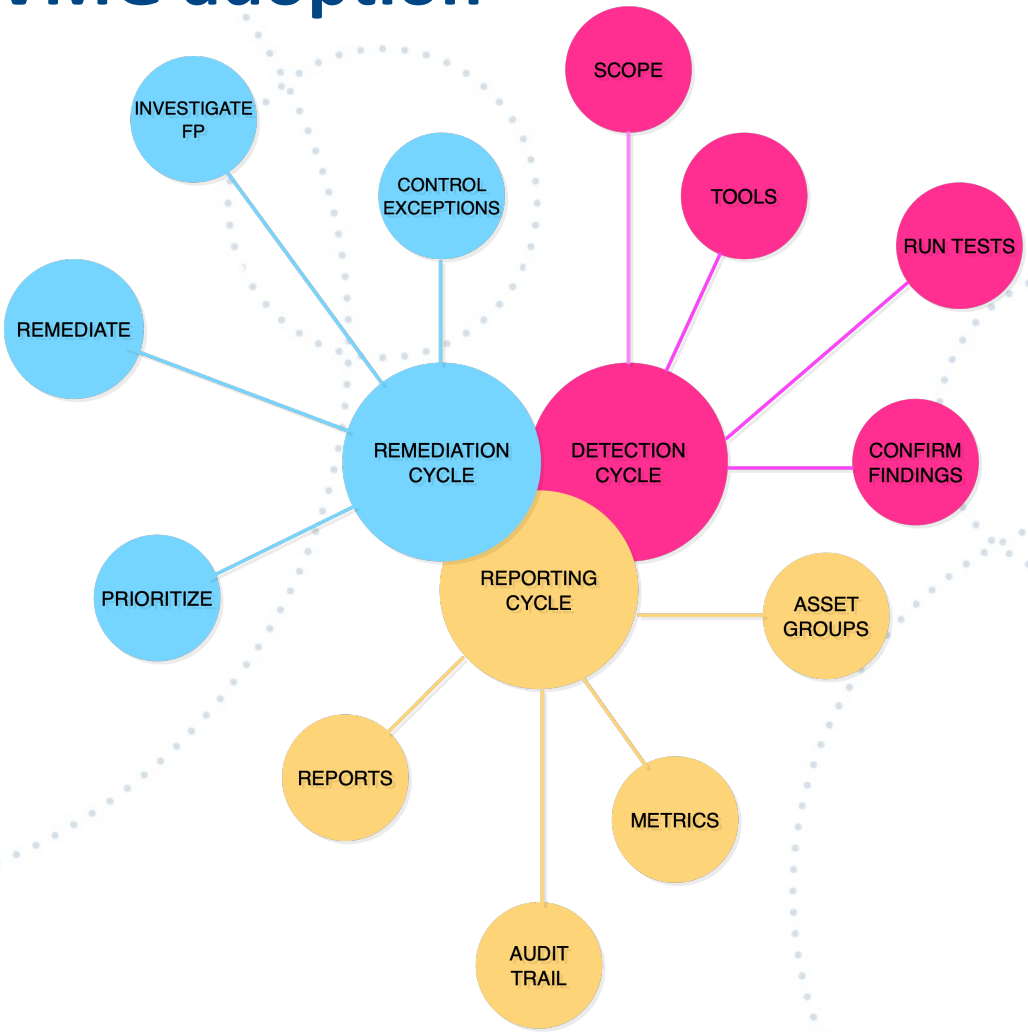
Not a Vulnerability Management:

- SOC Audits
- Applying current patches
- Periodic reporting
- Writing exploits and POC
- Writing exception for remediation
- Running a network or DAST/SAST scanners
- Pen testing

*All is a good stuff, keep doing these!

Reasons to consider OVMG adoption

- Concise
- Agnostic
- Task Oriented



How To Read

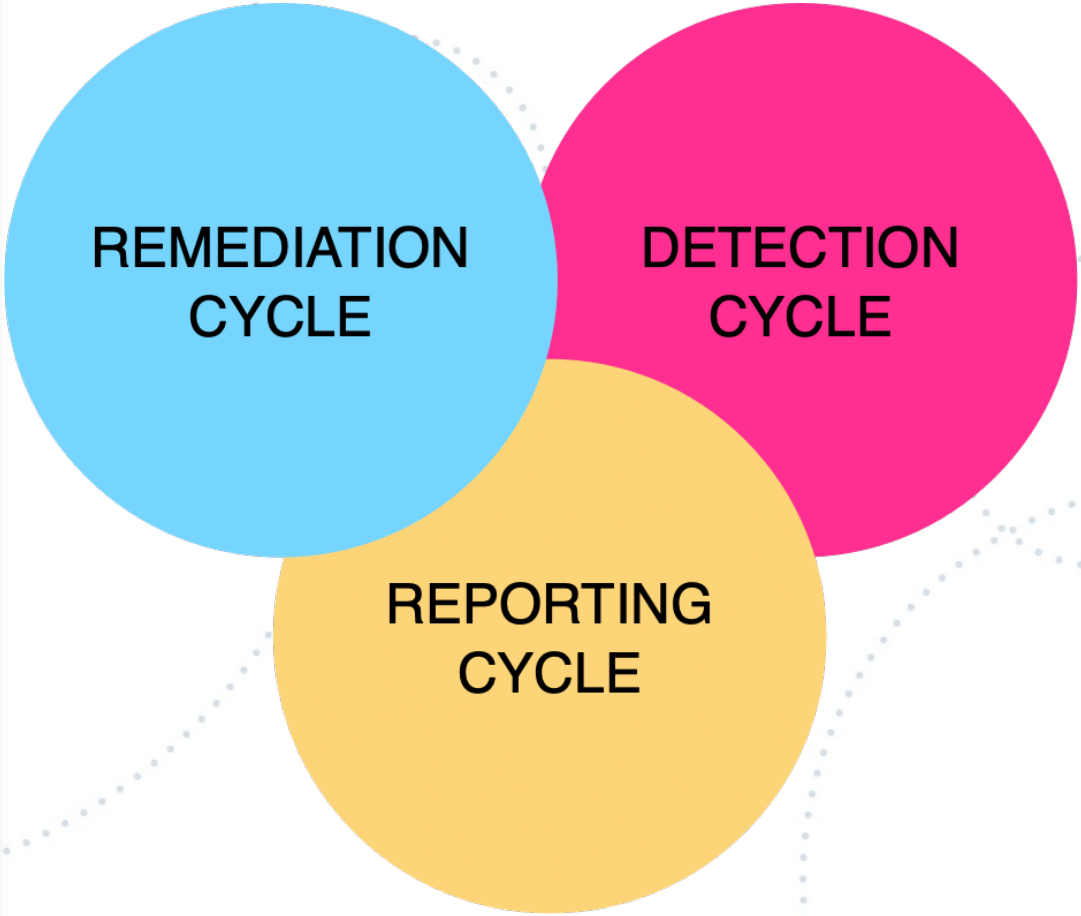
- Cycle – a building block of VM program, a process, color coded
- Task – is actionable item that feeds into other processes
- Input/Output – shows interdependencies
- To Do – aids to the checklist (v.0)
- Why – reasoning, this way you can make your own judgement call
- End goal – a short statement of expected result at the end of each task

1.1 TASK		INPUT	OUTPUT
Define/Refine scope		2.4 Reports 3.4 Exceptions	1.2 Tools 2.1 Assets Groups 2.2 Metrics 2.4 Reports 3.1 Prioritize 3.4 Exceptions
#	**TO DO**	**WHY**	
1.1.1	Know the enterprise risks	Whether your organization does or doesn't have risk registry, you have to ur th	

End Goal: your management should give you sign-off on a specific vulnerability test in writing. Ideally, you have to have a vulnerability management policy ready, but that might happen after you complete several rounds of OVMG. By completing the Scope task, you should be able to explain to your management and your peers why vulnerability testing is needed and how it benefits the business. You should be able to outline the next steps. You also should understand the boundaries of vulnerability tests.

OVMG TIPS

- Understand the end goal of each process
- Skip some steps
- Plan your work in cycles:
  - Cycle – is a domain
  - Process – is a task
- You must keep repeating all cycles



Detection

"Scope" End Goal

- Get approval/s
- Know your dos and donts
- Communicate VM policy and your process to all involved

"Tools" End Goal

- Understand tools limitations
- Optimize your tools in use to fulfill the objectives of the vulnerability tests

"Run Tests" End Goal

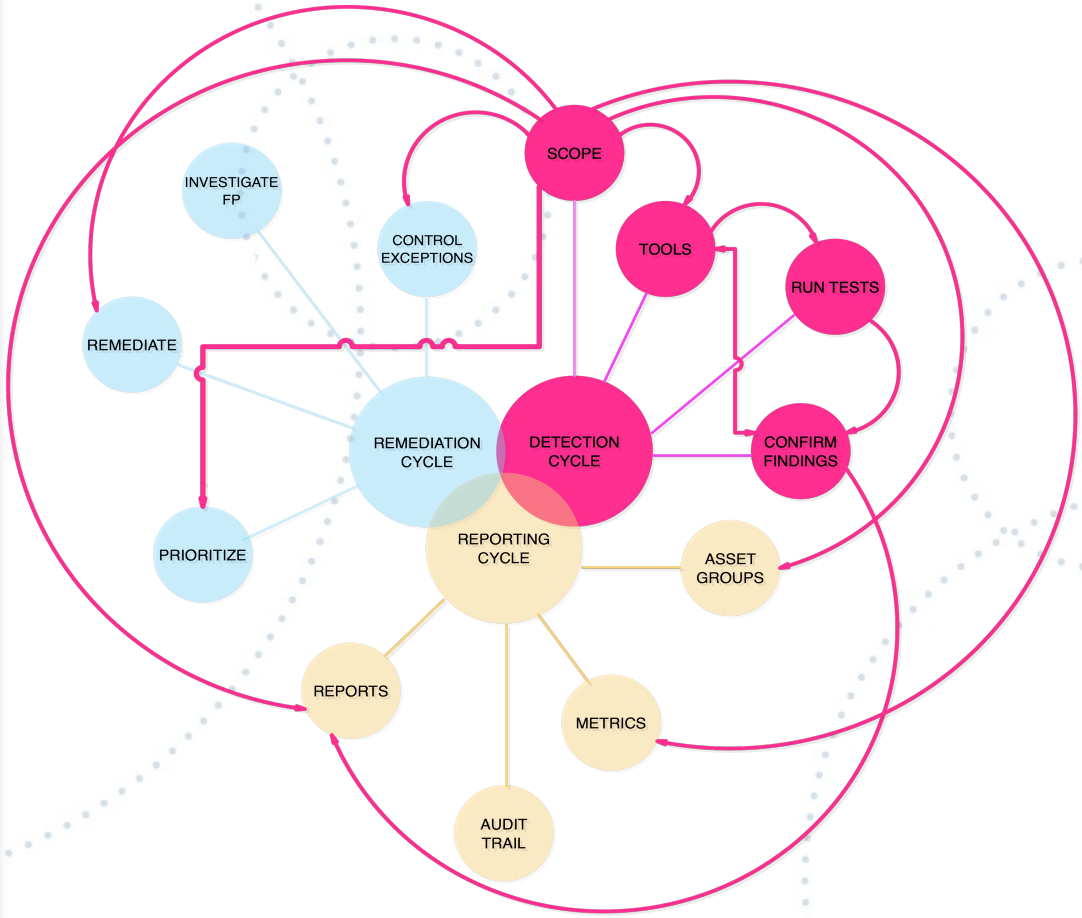
- run vulnerability tests

"Confirm Findings" End Goal

- Apply due-diligence to ensure integrity of results

Detection Tips

- Understand the dependencies
- Understand the importance of the “Scope”
- Planning is in focus
- Setting ground rules is in focus
- Gathering information is in focus for this cycle



Reporting

“Assets Groups” End Goal

- Create broad enough categories that could serve as atomic parts of your metrics: endpoints, OS or applications (environment, delivery, modules/components) e.g.

“Metrics” End Goal

- Quantify data, cross-reference with asset groups, emphasize what is important. Quantify by amount, percentage, share.

“Audit Trail” End Goal

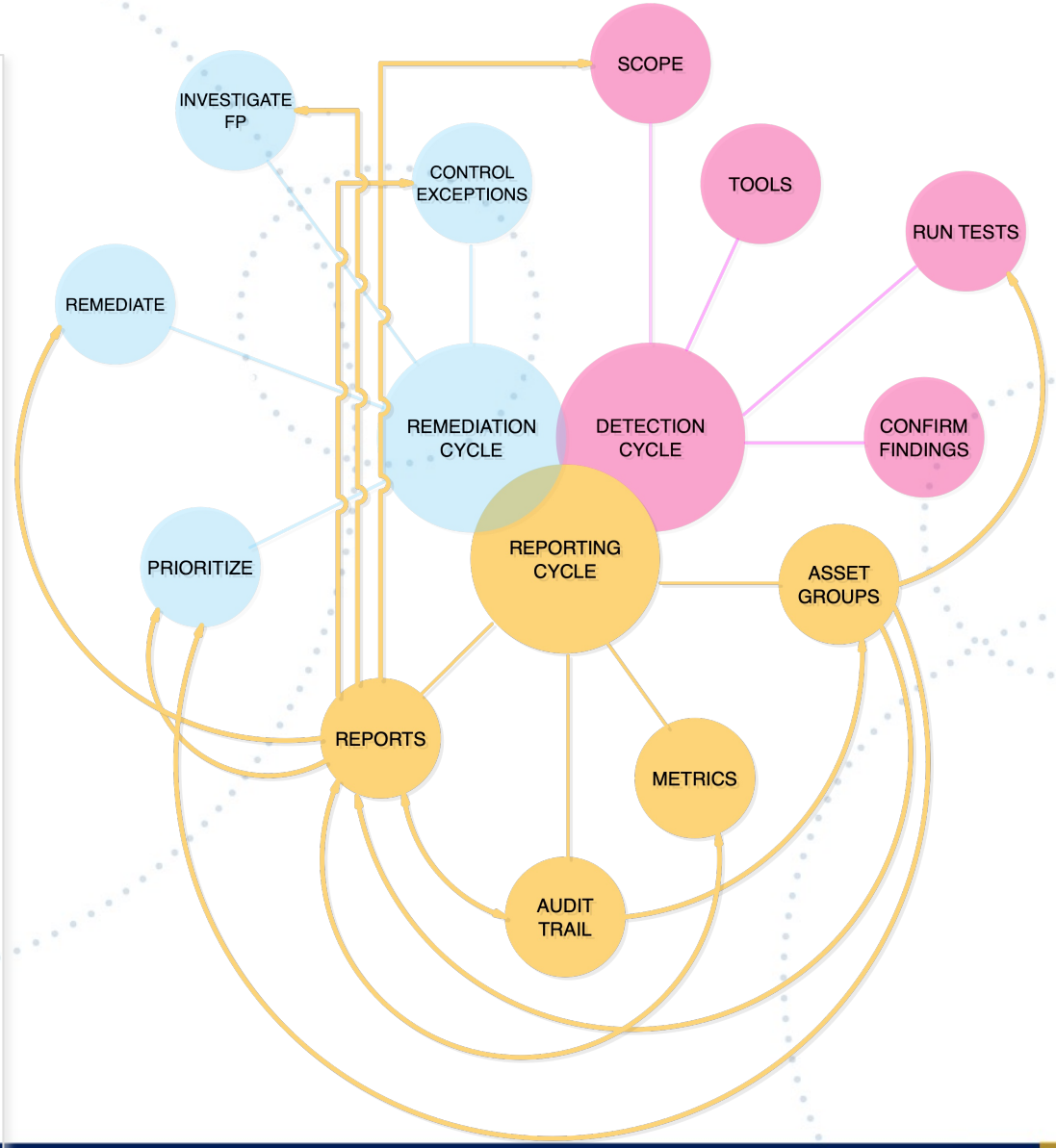
- create an audit trail to track a workload for teams or individuals who are responsible for vulnerability remediation

“Reports” End Goal

- summarize security scanning results in a concise form

Reporting Tips

- Get creative with Asset Groups
- Make sure that your metrics could be understood by others
- Check your (your tool) math
- Present the significance of your findings: big numbers are not necessarily big, and small are not relatively small if you have an exploit
- Do not anticipate that everybody has the same understanding of the results
- Create a ticket to track remediation
- Make sure that the ticket is assigned to the right stakeholders (and they know about it).
- Be consistent with your reporitng



Remediation

Prioritize

- evidence based reasoning why some vulnerabilities should be remediated asap, in a week, in a month, e.g.

Remediate

- remediation work done by the assigned teams/individuals. Remediation should not be assumed until checked by the next round of tests.

Investigate FP

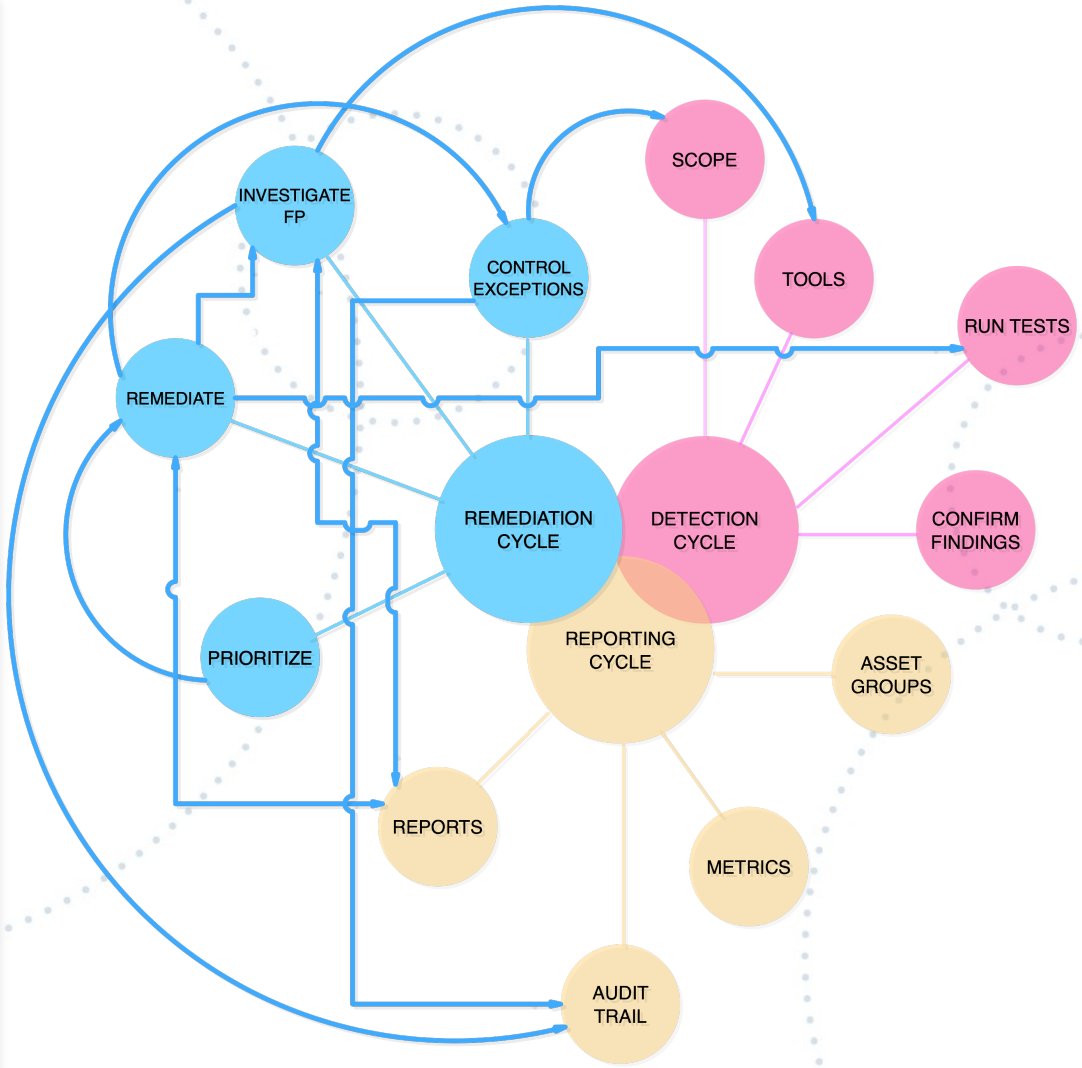
- review the evidence that suggests that an instance is not a vulnerability. Tune the tools as needed.

Control Exceptions

- ensure that all non-compliance is documented and approved by senior management.

Remediation Tips

- Communicate in non-antagonistic manner with the teams (constructive comm)
- Collect evidence for FP and share it with all who need to know
- Create a process for tracking exceptions including the rules for their review
- Validate remediation work (restart the server)
- Do not discard information as a noise, log it instead.
- Keep auditable trail of your work



Thoughts?

- Emergent threats
- Add on CTI
- Supply chain compromise
- Third party exposure
- Exclude SE
- CVSS rating vs. Exploitability