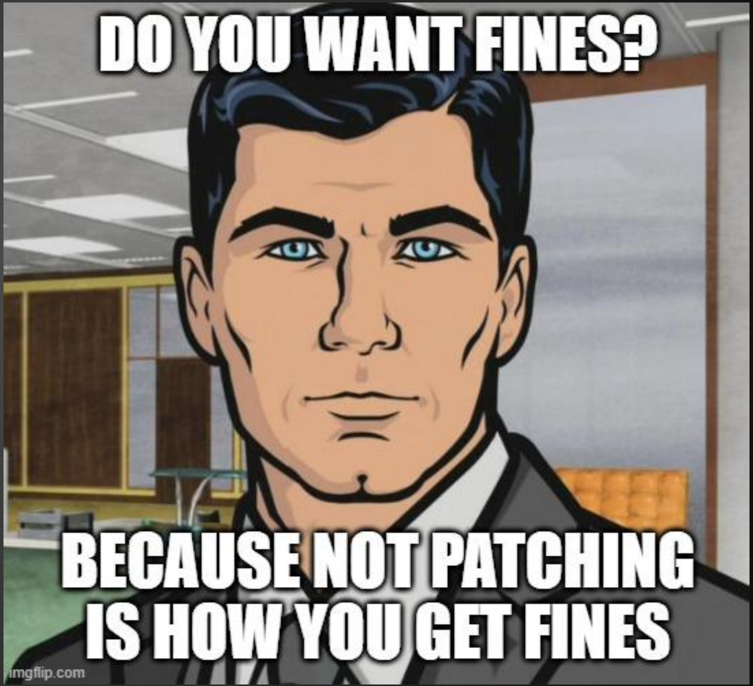
Building a Vulnerability Management Program

Avoiding pitfalls, managing risk, and mastering CYA

- Why you need a Vulnerability Management program
- How to build one
- Navigating the Vulnerability Management lifecycle

Why do I need a vulnerability management program?

- Vulnerability management is risk management
- PCI/HIPAA/DISA/etc says so
- Hackers don’t want you to patch



How do you build a vulnerability management program?

1) Get management buy-in
2) Write (or fix) your vulnerability management and/or patch management policies
3) Get the right tools and develop procedures for scanning and reporting
4) CYA (document, document, document)

Red Flags:

“We’d rather pay the fines than patch.”

“We’ll just accept the risk”

“We patch annually/quarterly/never because we can’t have downtime.”

“We don’t have the tools to patch so we don’t.”

“What’s the point in patching? There’ll just be more patches next month.”

Choose your weapon

- You don’t need an expensive tool (but it helps)
- Plan your attack
    - Document requirements
    - Pick the vendors you want to assess
    - Try to stay objective
- Assess your vendors
    - How is the product licensed? Are compliance and vulnerability scanning licensed separately? What about web app scanning?
    - Plan for the future - do they do container scanning? Do they have an API that can be leveraged for automation?
    - How well does it do reporting?
    - Address **all** of the requirements on your list
    - POC if you have a chance

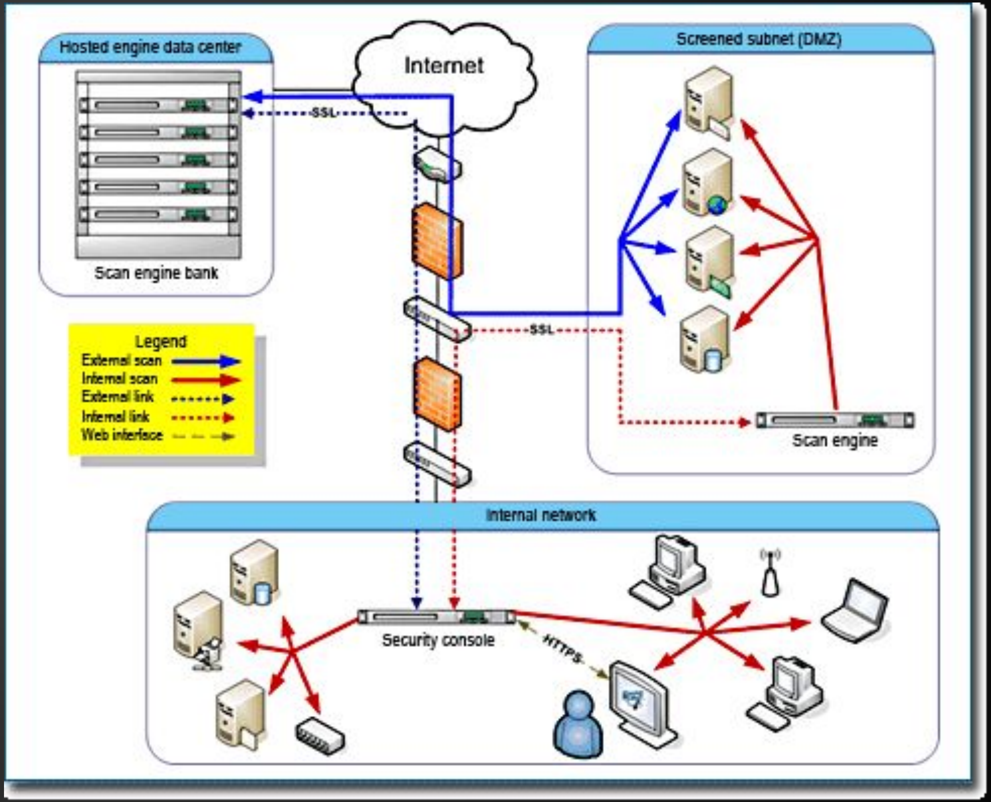
Requirement	Weight (1-5, 5 is most critical)	Vendor A (1-5, 5 fills requirement completely)	Vendor B (1-5, 5 fills requirement completely)
Scans must complete in a timely manner.	4	1 (individual assets can take hours to scan)	4 (individual assets take 5-10 minutes to scan)
Scanner must offer an API for automation	2	5	5
Reporting interface must be simple and easy to use	5	1 (interface is slow, basic reports take hours to run)	3 (interface is easy to navigate but reports can take hours to run)
Vendor must offer an on-premise central console	4	5	1 (vendor only has a cloud offering at this time)
	Weighted Score	**39**	**45**

Plan your deployment

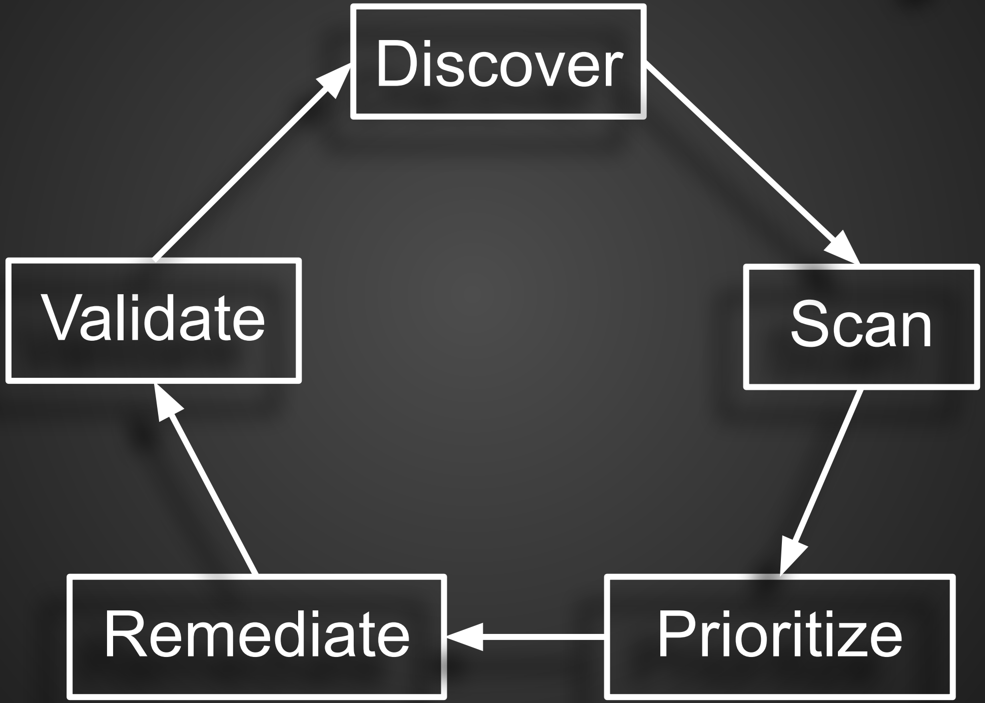
- What does your network look like?
  - Do you have any impediments like low latency links or firewalls in between sites?
  - Do the math - how many assets do you need to scan, and in what amount of time?
  - Be prepared to add more resources to your scan engines, or more scan engines to your environment
- Credential Management
  - Do you have a password manager? How will you reduce risk from cached credentials?
  - Test extensively and track authentication failures

Plan your deployment

- Keep your scanners close to the systems they're scanning
- Avoid opening up firewalls more than you have to
- Be mindful of shared resource limitations on virtual machines



Enter the Lifecycle


Discover

Scan

Prioritize

Remediate

Validate

Discover

- How do you decide what to scan?
  - Discover everything, vulnerability scan only what you're licensed for
  - Talk to your network engineers to see if you can maximize efficiency
  - What are you scanning, when?
- Run your discovery scans
  - Use the OS detection
  - Your endpoint firewall will shank you
  - Cull the herd - watch out for ghosts, VIPs, dead DNS names
- Build your vulnerability scan lists
  - How do you want to group assets?
  - Make sure your credentials are assigned correctly

WHAT DO YOU MEAN I CAN'T SCAN THE WHOLE CLASS B?!
IT'S FOR SECURITY!
Foto: Th. Reinhardt / pixelio.de
MemeCenter.com

Scan

- Avoid resume generating events
  - CYA - work with the business on scheduling, follow the change control processes, notify everyone every time you do something
  - Proactively test things that are likely to break
  - You will break things! Be available and ready to kill scans if needed
- Avoid letting the business dictate your requirements
  - No open ended questions - offer options for them to choose from
  - Don’t take “you can’t scan it, it’ll break” as the final answer
  - No one wants you to scan during business hours in case there’s an outage, but if you cause an outage during non-business hours, no one is around to fix it.
- Pick a schedule and stick to it

Double check your scan results!

- “The scans came back clean!” Or your credentials were wrong...
- The patch management system says there’s no patches required but the scanner says otherwise - who’s right?
- Check with your system admins before you hand them hundreds of tickets - automated ticketing is bad and wrong

Prioritize - How do you eat the elephant?

- Two types of findings - the patch all the systems are missing, and the system missing all of the patches
- Not every finding is “patchable”
- It can be overwhelming - find one thing that seems manageable and start there

Prioritizing Based on Cost and Risk

- Cost = User impact + Operational cost + Monetary cost
  - Operational cost includes packaging and deploying patches, manual patching, and potential downtime resulting from a patch that breaks business processes
  - Monetary cost might be implementing third party patching solution, professional services, etc
- Risk = Vulnerability + Threat + Exposure
  - Vulnerability - CVSS score, vendor's recommendations
  - Threat - what are your threats? Depending on your industry the threat level can be very high or very low
  - Exposure - where do the assets with the vulnerabilities sit? How protected are they?
  - See if you can reduce the risk - what mitigations are available?

Prioritizing

- This is an example of a prioritization matrix - this may look different for you and your company depending on your assets and tools
- Lower prioritization doesn’t mean it shouldn’t be done!

“Cost” \ Risk	**Low**	**Medium**	**High**
Low	Medium	High	Highest
Medium	Low	Medium	High
High	Lowest	Low	Medium

Top Priorities

- Windows OS and Browsers
- The unholy trinity - Acrobat Reader, Flash, and Java
- Unsupported/Out of date operating systems and software
- Edge devices (routers, firewalls, VPNs)
- Systems containing sensitive info (domain controllers, database servers, HR/Payroll)

Remediate

- Manage business uptime requirements - have a regular cadence so they know when the reboots are coming
- Leverage any centralized patch management
- How do you handle mobile systems? Legacy systems?
- Have a plan to document exceptions
- Anticipate problems

YO DAWG, I HEARD YOU
LIKE PATCH MANAGEMENT
SO WE PUT SOME BUGS IN YOUR PATCHES
SO YOU CAN PATCH YOUR PATCHES
imgflip.com

Validate

- Remediate, Rescan, Repeat
- APIs and self-service -- give the admins access to do validation scans themselves
- Track the time it takes to remediate
- Can you ever really hit 100% compliance? (spoiler alert: probably not)

Final thoughts...

- Track your progress - use your report data to build graphs showing progress and risk reduction
- Give credit where credit is due - a little gratitude can go a long way
- Don’t give up - collaborate, try different angles of attack
- Be proactive - make vulnerability scans part of the system build process to make sure they’re up to date before they’re brought online

Vulnerability management in nutshell

- Every organization is different
- Be prepared to change your approach
- Don’t let perfect be the enemy of good
- Keep fighting

TL;DR -- Vulnerability management is about people, not patching

Thank you!

Questions? Comments? Let me know!
@megan_b in Discord