Stop Messing Up Vulnerability Management

David Quartarolo

Vulnerability Management is NOT sexy

Boring

- No Artificial Intelligence (AI)
- No Machine Learning (ML)
- No Hyperautomation
- No Quantum Computing
- No Everything as a Service
- No Distributed Cloud
- No Multiexperience
- No Extended Reality (XR)
- No Blockchain
- No Metaverse/Multiverse

Technology Buzzword Total = 0

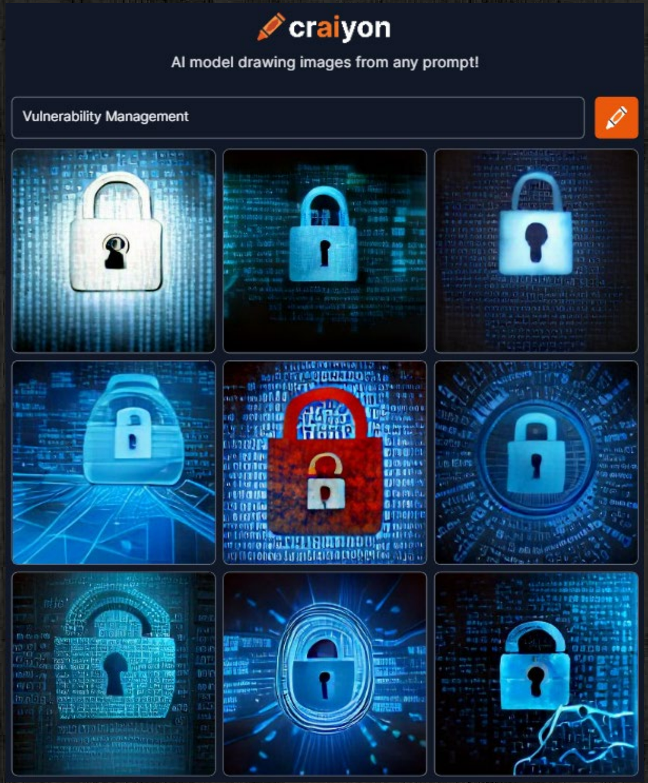
Hello!

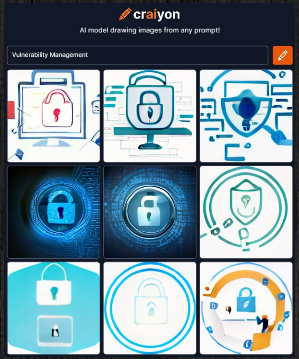
I am David Quartarolo

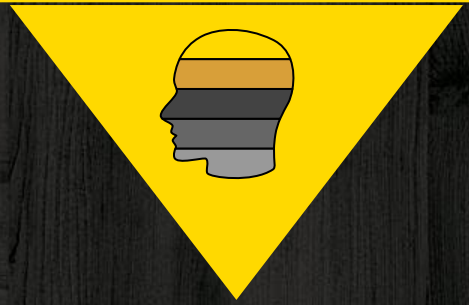
- Information Systems Security Engineer @ Sectigo
- Professional Open-Source Developer @ Active Countermeasures

Checkout:
github.com/activecm/smudge

What A.I. thinks Vulnerability Management looks like:







WHAT IS IT?

Cyclical practice of identifying, classifying, prioritizing, remediating, mitigating, and reporting vulnerabilities.

10

Not Having A Proper Inventory

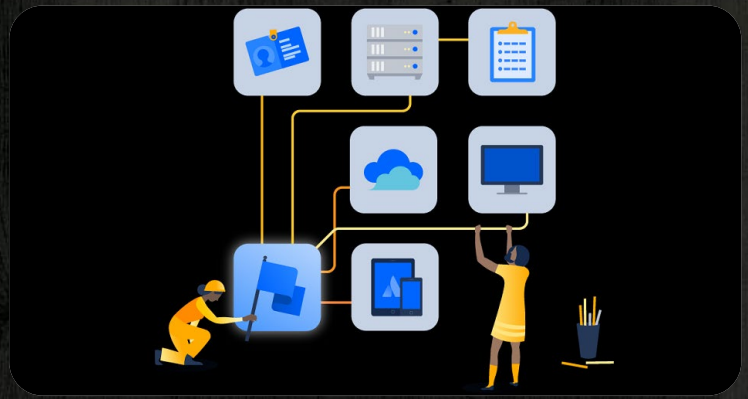
How can we assess the environment when no one knows what it is?

Don’t Screw Up Your Inventory

- Aligns with CIS Controls v8 Control 1
- Accessible to all IT Staff
- Consistantly Assessed

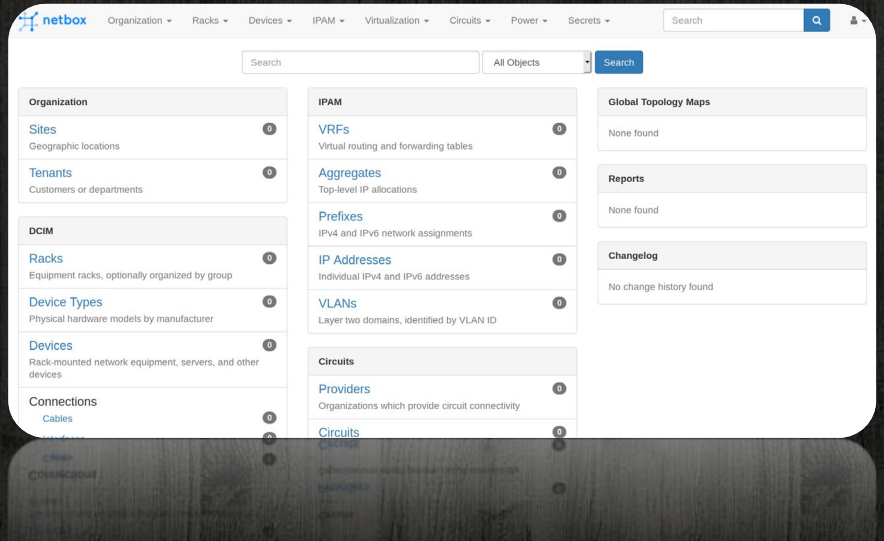
Bonus Points

- API
- Automation



Inventory?

- Physical Devices
- Virtual Machines
- Applications
- Libraries
- Containers?



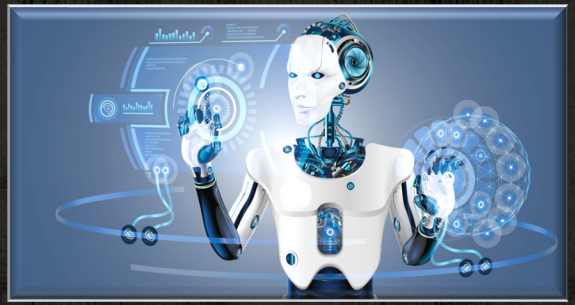
9

Not Having Adequate Automation

Vulnerability Management is filled with repetitive taks. Stop wasting your time!

Automation

- Automate the repetitive (especially remediation)
- Buy Software if you are not a dev
- Be Choosy
- Organization Size Dependent
- Standardization makes automation easier
- The bad guys automate everything!



8

Not Having Sufficient Metrics

Tell me again, HOW bad is it?

Metrics

- Evaluate Coverage
- Quantify Risk
- CVEs?
- VPR?
- Custom?

Key Driver	Description
Vulnerability Age	The number of days since the National Vulnerability Database (NVD) published the vulnerability.
CVSSv3 Impact Score	The NVD-provided CVSSv3 impact score for the vulnerability. If the NVD did not provide a score, Tenable.sc displays a Tenable-predicted score.
Exploit Code Maturity	The relative maturity of a possible exploit for the vulnerability based on the existence, sophistication, and prevalence of exploit intelligence from internal and external sources (e.g., Reversinglabs, Exploit-db, Metasploit, etc.). The possible values (**High**, **Functional**, **PoC**, or **Unproven**) parallel the CVSS Exploit Code Maturity categories.
Product Coverage	The relative number of unique products affected by the vulnerability: **Low**, **Medium**, **High**, or **Very High**.
Threat Sources	A list of all sources (e.g., social media channels, the dark web, etc.) where threat events related to this vulnerability occurred. If the system did not observe a related threat event in the past 28 days, the system displays **No recorded events**.
Threat Intensity	The relative intensity based on the number and frequency of recently observed threat events related to this vulnerability: **Very Low**, **Low**, **Medium**, **High**, or **Very High**.
Threat Recency	The number of days (0-180) since a threat event occurred for the vulnerability.

7

Not Defining An Exception Management Policy

The exception list hasn't been updated since I been here because no one knows where it is.

Exception Management

- Don't Skew reporting
- Review Often
- Approved by CISO / Department Head
- Exception List Accessible by Everyone
- Proper Exception List
  - Expiration
  - Approved Reason
  - Approval Authority

6

Having a Bootstrap Vulnerabilty Management Policy

Compliance said we needed a policy so we downloaded one.

Create a Vulnerability Policy

5

Not Having Appropriate Reporting

How much translates into new engineers or budget?

Reporting

- Metrics + Reporting mean nothing if not delivered.
- Metrics + Reporting mean nothing without your exception list.
- (Metrics + Exceptions) == Report -> Delivered to management team

This helps management understand organizational risk.

4

Not Having a Recurring VM Meeting

Time for politics!

VM Meetings

- Form a “Vulnerability Management” Committee
- Keep Meetings Short
- Review Monthly VM Report
- Socialize the VM Program’s Agenda

Vulnerability Management takes a village. This process merits a little time every month to ensure it’s success.

3

Not Defining A VM Scope

How much is a lot?

Scope

- Ensure your scope is clearly defined in your VM Policy
- Document exceptions PER Scope
- Needs to be constantly reevaluated
- Scanning needs to be defined PER Scope
- SLAs need to be defined PER Scope
- Scope needs to be defined by Business Process (Like everything else in InfoSec)
- Critical Systems Tiering Methodology?

2

Not Performing Baselining

Vulnerability Management != Patch Management

Baselining

Baselining (AKA Benchmarking)
Best practice for securely configuring systems.

- Tune baselines to your organization.
- Baselining should be done before and after patching.
- Get baselines from CIS (one way or another).
- Exceptions from baselines STILL NEED TO BE DOCUMENTED.

1

Not Having Transparency

The right people must be in the know!

Transparency

- Individuals that need information need to always have access to it!
- Don't lie or exaggerate on reporting or metrics.
- Use reporting and metrics to evaluate business risk!
- Keep other parts of the business in the know (if allowed).
- Context is EVERYTHING! Make sure you keep management apprised of real-world events.

I have more concerns about potential risks and vulnerabilities than most people.
- Nouriel Roubini

Thanks!

Any questions?

@d_quartarolo

/in/david-quartarolo/

but-i-am-dominator

Extra Resources

Resources

- Netbox
- Ansible
- VPR
- CIS Benchmarking