



Vulnerability Management Tool
Research Manual

Student Name: Síne Doheny
Supervised by: Richard Butler
Student ID: C00226237

Table of Contents

Abstract	4
Vulnerabilities	5
What is a Vulnerability	5
Where do Vulnerabilities Come From.....	5
Types of Vulnerabilities.....	5
Most Common High-Risk Vulnerabilities	5
Anatomy of a Cyber-Attack.....	7
Ontology of a Cyber-Attack and the Contribution of a Vulnerability in the Cyber-Attack relationship.....	7
Application Security	8
Types of Application Security	8
Where to Apply Application Security	9
Vulnerability Management	10
Why do we need vulnerability management?.....	10
Anatomy of a Vulnerability Management Life Cycle	10
Vulnerability Types	11
Vulnerability Assessment	11
Vulnerability Assessment Types	11
Findings	12
Vulnerability Management Team.....	12
Vulnerability Management Programs	14
Top Vulnerability Management Programs and Their Tools	14
Tools that are currently on the market and a listing of programs that have these tools implemented:.....	18
Tools that I would like to implement in my application and what languages you can create these tools with:.....	19
Development Perspective	21
Types of Applications.....	21
Factors to Consider.....	21
Development Languages.....	22
Web Application Architecture	22
Presentation Layer	22
Presentation Layer Development Languages.....	22
Business Logic Layer	22
Business Logic Layer Development Languages	22

Data layer	23
Types of Databases	23
Database Software's	24
Composition of a Vulnerability Management Application	25
Types of Security Concerns	25
Discover	25
Inventory management	25
Analyse	26
Vulnerability Assessments	26
Vulnerability Scanners	27
Scans	27
Performing a Vulnerability Scan	28
Scope of a Vulnerability Assessment	28
Negative Aspects of Vulnerability Scanning	29
Prioritise	29
Risk Management	29
Threat Modelling Process	29
Grouping and Prioritising Assets	30
Calculating Asset Value	30
Risk Assessment of a Vulnerability	32
Common Vulnerability Scoring System	32
DREAD	32
Vulnerability Management	34
Triage Vulnerability Group	34
Reports	34
Remediate	35
Patch Management	35
Managing a Vulnerability Management Application	35
Conclusion	37
Bibliography	38

Abstract

Bugs, flaws, errors, holes, weaknesses all equally related, all characterised by one name: a vulnerability. A vulnerability is a weakness which creates opportunity for a cyber-attack to take place. On average there is a cyber-attack every thirty nine seconds in today's world. An effective remedy for preventing exploit of a vulnerability is through vulnerability management. With an estimate of over thirty-one billion connected devices in today's world there is a mass of flaws and vulnerabilities introduced into enterprise environments waiting to be exploited. The purpose of this research manual is to assess the various factors that contribute to a successful, safe and secure vulnerability management tool which in turn will provide a basis of research to begin implementation of a management interface to facilitate all aspects of vulnerability management for an organisation. The aim of this project is to gather an understanding of a vulnerability, it's position in both an I.T. environment and a cyber- attack and to research the methodology of vulnerability management in ideology of discovering how to effectively store and efficiently mitigate a vulnerabilities presence.

Vulnerabilities

What is a Vulnerability

According to NIST a vulnerability is a “weakness in an information system, system security procedures, internal controls, or implementation that could be exploited or triggered by a threat source.” (Vulnerability - Glossary | CSRC, 2020). Errors, mistakes and poor design choices in software and systems all lead to vulnerabilities, weaknesses, flaws, faults, bugs and holes which are the names of entities that attackers exploit to gain access into or retrieve information out of a system.

Where do Vulnerabilities Come From

Vulnerabilities are often the result of a lack of multiple factors which include errors and mistakes in how software is written, flaws in user management, the response of an application to data input and erroneous or ill-advised use of application defense techniques. On the other spectrum sometimes these errors or gaps in the logic are a result of factors such as conscious decisions of development teams to abandon vulnerabilities due to a lack of time and pressure to produce and deliver an application and also the disincentive for developers to build security into an application, especially if the organization does not promote a culture that makes security everyone’s responsibility (Anatomy-of-an-Application-Security-Weakness.pdf, 2020). Schedules with tight deadlines and no room for security often leave developers with no time to attempt to achieve aspired security levels, primarily due to the fact that security is often not a priority, as accompanying it arises longer testing hours which leads to delayed finished products and higher development costs resulting in higher selling prices which some organisations cannot afford or do not want. Some organisations feel sufficient with a minimum viable product that has just enough features and value to gain traction with customers (Where Do Security Vulnerabilities Come From? - Dark Reading, 2020).

Types of Vulnerabilities

There are many types of computer security vulnerabilities. Adversaries will search for these weaknesses to exploit and completely take over a system to steal data or take advantage of and perform unauthorised actions in a computer system. There is no definitive list of the most common security vulnerabilities as technology is ever growing and so are vulnerabilities however, OWASP has provided a top ten most common high risk vulnerabilities list and MITRE produced the top twenty-five common weaknesses enumeration list which are what most companies will follow. These hybrid lists cover a range of software environments, including web applications and mobile applications which accounts for most enterprise applications.

There are many criteria which account for how a type of vulnerability is chosen such as how common a threat is, how easy a threat is to detect and remediate and a threats potential technical and business impact.

Most Common High-Risk Vulnerabilities

According to OWASP the top ten most common high-risk vulnerabilities are as follow:

1. Injection

Injection weaknesses are exploited when untrusted data is sent in a request as part of a command or query. This type of attack tricks the targeted system into executing

something that the application was not designed or programmed to do, which could result in access to protected/sensitive data.

2. Broken Authentication

Authentication and session management are often implemented incorrectly in applications, allowing attackers to compromise passwords, keys, or session tokens, or to exploit other implementation flaws to steal other users' identities temporarily or permanently.

3. Sensitive Data Exposure

Data such as healthcare data, financial data, credentials, and personally identifiable data is known as sensitive data. Many web applications and API's do not properly protect such data which often results in the sensitive data being compromised and data theft. Extra protection of this data can be carried out with encryption at rest or in transit, requiring special precautions when exchanged with the browser.

4. XML External Entities (XXE)

Older or poorly configured XML processors evaluate external entity references within XML documents. External entities can be used to disclose internal files using the file URI handler, internal file shares, internal port scanning, remote code execution, and denial of service attacks.

5. Broken Access Control

Enforcing restrictions on what authenticated users can do is essential to prevention of broken access control. With this vulnerability attackers can exploit these flaws to access unauthorized functionality and/or data, such as access to other user's accounts, viewing of sensitive files, modification of other user's data and changing access rights, etc.

6. Security Misconfiguration

This is commonly a result of insecure default configurations, incomplete or ad hoc configurations, open cloud storage, misconfigured HTTP headers, and verbose error messages containing sensitive information. Not only must all operating systems, frameworks, libraries, and applications be securely configured, but they must be patched/upgraded in a timely fashion.

7. Cross Site Scripting (XSS)

XSS vulnerabilities occur whenever an application includes untrusted data in a new web page without proper validation or escaping, or updates an existing web page with user-supplied data using a browser API that can create HTML or JavaScript. XSS allows attackers to execute scripts in the victim's browser which can hijack user sessions, deface web sites, or redirect the user to malicious sites.

8. Insecure Deserialization

Insecure deserialization often leads to remote code execution which can be used to perform attacks such as replay attacks, injection attacks, and privilege escalation attacks.

9. Using Components with Known Vulnerabilities

Components, such as libraries, frameworks, and other software modules, run with the same privileges as the application. If a vulnerable component is exploited, such an attack can facilitate serious data loss or server takeover. Applications and APIs using components with known vulnerabilities may undermine application defence's and enable various attacks and impacts.

10. Insufficient Logging and Monitoring

Insufficient logging and monitoring, coupled with missing or ineffective integration with incident response, allows attackers to further attack systems, maintain persistence, pivot to more systems, and tamper, extract, or destroy data.

(OWASP Top Ten Web Application Security Risks | OWASP, 2020)

Anatomy of a Cyber-Attack

Nowadays adversaries have become highly sophisticated and organised. To prevent a cyber-attack, it is essential that we understand how the theoretical evolution of an attack is formed. The following are the stages of a cyber-attack:

Stage One: Reconnaissance

Cyber criminals in this stage will search the web for any piece of publicly available data that will allow them access to compromise a company's network. This could be through the form of a company's intellectual property or vulnerabilities and potential entry points for example sensitive data such as an employee's email which could be used to gain entry to the company network through the email to exploit the network, a file, or an application vulnerability. This stage is used to gain access to the organisation.

Stage Two: Compromise

Once inside the network an adversary will "poke around", while remaining undetected searching for vulnerabilities that exist in the current system and application infrastructure and will often also look for pathways through a company's third party businesses to involve those businesses easily in the attack creating an operational effort.

Stage Three: Attack

This is the stage where data is stolen, or malicious software is planted in a company's system. It is the stage where the company network escalates from compromised to breached. A company may not be aware this event is taking place in the background and often it is not uncommon for a company to first hear about the attack from a customer or supplier who had been used in the attack.

Stage Four: Response

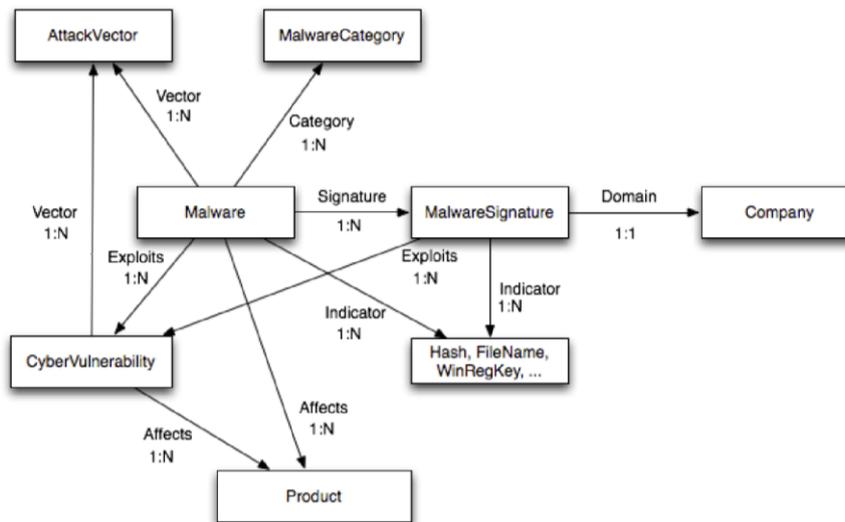
The final stage of the cyber-attack is when evidence of the attack which can be found is removed by the organisation but often hidden holes which could not be uncovered as they are so well hidden in the system are left behind and an adversary will return through them at a later stage again if successful.

Ontology of a Cyber-Attack and the Contribution of a Vulnerability in the Cyber-Attack relationship.

Cyber events are found to be a repercussion of entities and their cyber ontology. Cyber events are the result of different relationships between entities, with each leading to a further stage in a cyber-attack. An entity is something and anything with a distinct existence. In the following ontology recorded future have used the entity type "Malware" which holds an association with the entities "AttackVector", "MalwareCategory", "MalwareSignature" and

“CyberVulnerability”. The “MalwareSignature” entity has an association of technical indicators such as hashes, file names, Windows registry keys, etc. Each of these entities form a relationship which affects the entity “Product”.

In the following example we can see the role a vulnerability plays in a cyber-attack. This allows us to better understand where in the cyber-attack anatomy a vulnerability forms its existence and is used as the key step to carry out the rest of the relationships and the attack in general.



(Anatomy of Cyber Threats, Vulnerabilities, and Attacks, 2020)

Application Security

Application security is the process of developing, adding, and testing security features within applications to prevent security vulnerabilities against threats such as unauthorized access and modification (Application Security, 2020).

Application security is important due to applications now being available and connected to various networks and the cloud leaving them open to attacks and exploitation through any vulnerabilities present. Application security testing can prevent these attacks.

Types of Application Security

The security of application can be enhanced by including features such as: authentication, authorisation, encryption, and logging.

Authentication:

Authentication ensures that only authorised users who require access, can gain access to an application. This feature can be accomplished by requiring a user to login to the application using credentials. This feature can be enhanced by requiring multi factor authentication whereby a user will use credentials and then a thumb print etc.

Authorisation:

This feature allows a system to validate if a user has permission to access the application by comparing the users identity with a current list of authorised users. Authentication must occur

prior to authorisation so that the application matches only validated user credentials to the authorised user list.

Encryption:

After authentication, encryption can be used to protect sensitive data from being seen or used by a cybercriminal. In web-based applications where traffic and data travels between the database and the application, traffic can be encrypted so that it can render unreadable or usable and protected.

Logging:

Application log files provide a time-stamped record of which aspects of an application were accessed and by whom, this ensures that in the event of a security breach of an application, logging will easily identify who had access to the data and how.

Where to Apply Application Security

Cloud Based Application Security

Data in cloud-based applications is at an elevated risk due to the data being transmitted across the internet from the user to the application and back. This means extra care must be taken to ensure application security protects the sensitive data from a breach.

Mobile Application Security

Mobile devices also transmit and receive data across the internet leaving them susceptible to vulnerability. An organisation can add a layer of security through the use of a virtual private network for employees who have access to the organisation services remotely. Companies can also enforce security policies before allowing an employee access to the corporate network remotely.

Web Application Security

Information is transmitted to and from a user using an application that is accessed through a browser interface over the Internet. Business's that host web applications and services over the internet can prevent intrusion using a web application firewall. A web application firewall works by inspecting and if necessary, blocking data packets that are considered harmful to the application/network.

Vulnerability Management

Vulnerability management is defined as the process in which vulnerabilities in operating systems, enterprise applications, browsers and end user applications are identified, categorised, prioritised, and remediated. A vulnerability management program evaluates the identified risk of vulnerabilities. This evaluation leads to remediation and correction of vulnerabilities or a formal risk acceptance by an organisation whereby the cost of correction would outweigh the possible damages of the vulnerability.

Vulnerability management can often be confused with vulnerability scanning. Vulnerability scanning detects and classifies system weaknesses in computers, networks, and communications equipment. Whereas vulnerability management is the process surrounding vulnerability scanning, it is an ongoing cyclical organisational effort which not only scans for vulnerabilities but takes into account asset management, information management, vulnerability assessment, risk assessment of vulnerabilities, vulnerability reports and remediation.

Why do we need vulnerability management?

Organisations are continuously seeking to develop and employ security best practices to protect their assets. These Assets are tangible and intangible items that can be assigned a value. Risk is a chance that something unpredictable may happen to these items. Risk is the function of a threat exploiting a vulnerability. A vulnerability management program is essential to control these information security risks by enabling an organisation to have a continuous overview of vulnerabilities in their environment allowing them to control, identify and mitigate such vulnerabilities, preventing attackers from using them to gain access to an organisations systems and data.

Anatomy of a Vulnerability Management Life Cycle

There are six key steps to a successful vulnerability management life cycle:



(Vulnerability Management Lifecycle, 2020)

Discover: Inventory all assets across the network and identify host details including operating system and open services to identify vulnerabilities. Develop a network baseline. Identify security vulnerabilities on a regular automated schedule.

Prioritize Assets: Categorize assets into groups or business units and assign a business value to asset groups based on their criticality to your business operation.

Assess: Determine a baseline risk profile so you can eliminate risks based on asset criticality, vulnerability threat, and asset classification.

Report: Measure the level of business risk associated with your assets according to your security policies. Document a security plan, monitor suspicious activity, and describe known vulnerabilities.

Remediate: Prioritize and fix vulnerabilities in order according to business risk. Establish controls and demonstrate progress.

Verify: Verify that threats have been eliminated through follow-up audits. (Vulnerability Management Life Cycle | NPCR | CDC, 2020)

Vulnerability Types

Categories of vulnerability types include the following:

Operating system vulnerabilities: These vulnerabilities may be exploited to gain access to an asset the operating system is installed on or to cause damage. Examples include hidden backdoor programs and default super user accounts.

Network vulnerabilities: Issues on a networks hardware and software that expose it to possible intrusion by an outsider. Examples include insecure Wi-Fi access points and poorly configured firewalls.

Human vulnerabilities: User error can expose sensitive data and create exploitable access points for attackers.

Process vulnerabilities: Some vulnerabilities can be created by specific process controls. One example would be the use of weak passwords (which may also fall under human vulnerabilities).

Software security vulnerabilities: Software vulnerabilities range from errors in code to flaws in how the code responds to requests.

Vulnerability Assessment

A vulnerability assessment begins by identifying an organization's computer networks, hardware, software, and applications. It then engages in either penetration testing or vulnerability scans to determine the information security risk associated with the IT assets, including but not limited to network security and web application security (6 Strategies for Creating Your Cyber Vulnerability Assessment, 2020).

Vulnerability Assessment Types

There are several types of vulnerability assessments which can be found as follows:

Host assessment: The assessment of servers.

Network and wireless assessment: The assessment of private or public networks and network accessible resources to prevent unauthorised access.

Database assessment: The assessment of databases for vulnerabilities and misconfigurations.

Application scans: Identifying security vulnerabilities in web applications and their source code.

Findings

A report/database of findings from a vulnerability assessment should include the following:

- The name of the vulnerability
- The date of discovery
- The risk score, based on Common Vulnerabilities and Exposures (CVE) databases talk about rating risks associated with that dread matrix
- A detailed description of the vulnerability
- Details regarding the affected systems
- recommended mitigation strategies
- A proof of concept (PoC) of the vulnerability for the system (if possible) (what are the steps to reproduce vulnerability)
- A blank field for the owner of the vulnerability, the time it took to correct, the next revision and countermeasures between the final solution

Armed with this basic list when performing a vulnerability assessment, the recommendations phase will reflect a complete understanding of the security posture in all the different aspects of the process (A Step-By-Step Guide to Vulnerability Assessment, 2020).

In the database that captures vulnerabilities I would use the following parameters: ID, Version, Severity, Type, Family(Windows or Linux), Description, Asset affected, Remediation process, POC, Time, Age, Verified, Published, Remediated.

Vulnerability Management Team

A vulnerability management tool is only one part to the equation of remediation. When building a vulnerability management team, the following roles should be allocated and designated:

Security Officer: This is the owner of the vulnerability management process. This employee will ensure the program is implemented as it was designed to be.

Vulnerability Engineer: is responsible for configuring the vulnerability scanner and scheduling scans.

Asset Owner: is responsible for the asset that is scanned by the tool, this role decides whether identified vulnerabilities are mitigated, or the risk is accepted.

IT System Engineer: The IT system engineer is responsible for the remediation of identified vulnerabilities and ensuring that IT resources follow the organisations standard configuration.

Data controller:

Monitoring roles: These personnel should be responsible for analysing vulnerability severity, vulnerability information inventories and alerting the remediation team.

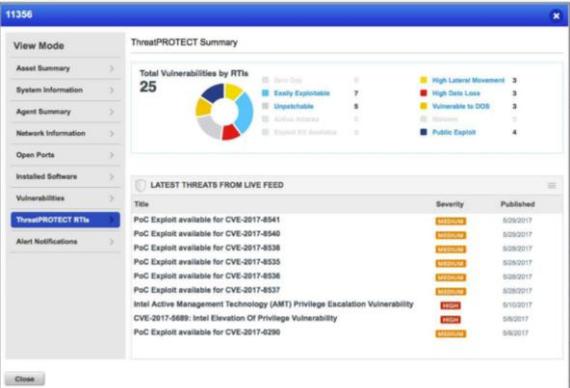
Remediation roles: Perform patching and ensure that patching and remediation was successful.

Authorisation roles: This is the management team personnel who are responsible for granting permission for certain actions.

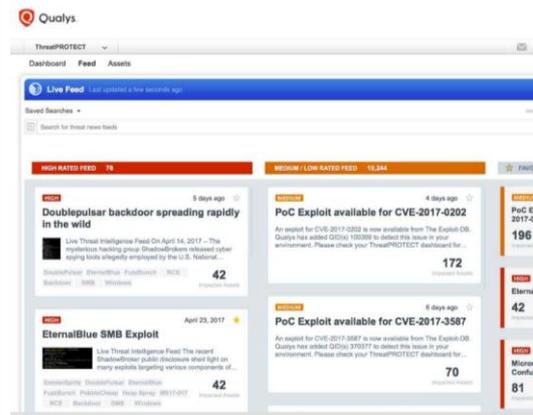
Vulnerability Management Programs

There are several vendors providing vulnerability management programs. Some of these vendors include: Qualys, Rapid7, Tenable, F-Secure, tripwire, GFI and Breachlock.

Top Vulnerability Management Programs and Their Tools

Program	Tools
<p>Qualys</p>	<p>Pre-Attack/Prevention Solutions: Asset discovery & inventory, certificate inventory, cloud inventory, asset tagging, ransomware attack prevention, anti-malware, exploit blocking, anti-phishing, behaviour based attack detection, file-less attack prevention, volume encryption, vulnerability management, misconfiguration assessment, patch management.</p> <p>Breach/Detection Solutions: MITRE (ATT&CK™) techniques and tactics driven detections, file, process & registry events, memory scan, threat intelligence enrichments, proprietary event risk scoring, insights into open ports, running processes, running services & installed software, file integrity monitoring, exploitable vulnerabilities detection, exploitable misconfigurations.</p> <p>Post Breach/Response Solutions: file quarantine/deletion, process termination, endpoint containment, user-defined PowerShell response, restoration of files/ systems, configuration/remediation, vulnerability patching, unwanted software, and service removal.</p> <p>Other features:</p> <p>Threat protection search engine: Craft ad hoc queries with multiple variables and criteria – such as asset class, vulnerability type, RTI, tag and operating system. Save any search, download results, and share them. Turn the queries you run regularly into permanent dashboard widgets whose information is dynamically updated in real time.</p>  <p>Live feed of vulnerability disclosures: Threat Protection’s Live Threat Intelligence Feed keeps organizations up to date on the latest vulnerabilities and news, so you are informed about new disclosures and about existing bugs whose risk severity has increased. This feature displays how many of your IT assets are impacted by each disclosure, with the use of the product’s powerful data correlation capabilities. It also allows you to click on feed entries and drill down into details and</p>

more granular information of a particular vulnerability and of the affected IT assets.



Continuous visibility: Always on platform sensors giving you continuous 2-second visibility of all IT assets.

Continuous Monitoring service: proactively address potential threats whenever new vulnerabilities appear, with real-time alerts to notify you immediately.

(Qualys Multi-Vector EDR | Qualys, Inc., 2020)

**Rapid7
(insightvm)**

Key features:

Lightweight endpoint agent: automatically collects data from all your endpoints, even those from remote workers and sensitive assets that cannot be actively scanned, or that rarely join the corporate network.

Live dashboards: InsightVM Live Dashboards are live and interactive by nature. You can easily create custom cards and full dashboards for anyone—from system admins to CISOs—and query each card with simple language to track progress of your security program.

Real risk prioritisation: Prioritises vulnerabilities on a 1-1000 scale based on the likeliness of an attacker exploiting the vulnerability in a real attack.

IT integrated remediation projects: Assign and track remediation duties in real time, providing continuous visibility into how well issues are being fixed. Take it one step further by integrating InsightVM directly with IT's ticketing systems to fold remediation seamlessly into the daily workload.

Cloud and virtual infrastructure assessment: full visibility into risk across your physical, virtual, and cloud infrastructure.

Attack surface monitoring with project sonar: InsightVM directly integrates with project sonar, a Rapid7 research project that regularly scans the public internet to gain insights into global exposure to common vulns. By leveraging Attack Surface Monitoring with Project Sonar, you can be confident that you have a pulse on all of your external-facing assets, both known and unknown.

Container security: InsightVM integrates with your CI/CD tools, public container repositories, and private repositories to assess container images for vulnerabilities during the build process—before they are deployed.

Integrated threat feeds: dynamic view show threats that are most relevant to environment, to better protect against current, impending threats and react quickly to critical, named vulnerabilities.

	<p>Goals and SLAS: Helps keep track and on top of goals.</p> <p>Easy to use RESTful API</p> <p>Policy assessment</p> <p>Automated assisted patching: Streamline remediation and when approved by sysadmin apply patches. Automatically re-assess impacted assets to verify successful patching.</p> <p>Automated containment: With Automated Containment, you can decrease exposure from vulnerabilities by automatically implementing temporary (or permanent) compensating controls via your Network Access Control (NAC) systems, Firewalls, and Endpoint Detection and Response tools; these can act as both stopgaps or long term solutions to reduce exposure.</p> <p>(Vulnerability Assessment Tool: InsightVM Features, 2020)</p>
Tenable.sc	<p>SLA Progress tracking: array of sensors automatically gather and analyse security and compliance data, the stages of vulnerabilities (unmitigated, mitigated) and CSC inventory of authorized and unauthorized software.</p> <p>Continual assessment of network: assess unknown assets and vulnerabilities, and monitor network changes, users, applications, and cloud infrastructure.</p> <p>Vulnerability summary: Three-month trend of vulnerabilities.</p> <p>Customizable dashboards: Leverage more than 350 pre-built, highly customizable dashboards and reports to get clear, actionable insight into the data you need to easily identify threats.</p> <p>Streamline compliance monitoring: Easily demonstrate adherence with predefined checks against industry standards and regulatory mandates.</p> <p>Gain operation technology (OT) visibility: Gain complete visibility, security and control over the OT threats that put your organization at risk with Tenable.ot integration.</p> <p>(Tenable.sc, 2020)</p>
F-Secure	<p>Endpoint security: automated solutions to cover all devices and servers for physical, virtual and cloud environments.</p> <p>Managed Detection and Response: detection, threat intelligence and incident response services with F-Secure countercept integration which uses advanced managed threat hunting service to detect and respond 24/7 to skilled human adversaries conducting live, hands-on keyboard attacks.</p> <p>Collaboration Protection: Complements native security capabilities by mitigating the risks in uploaded files, links and emails in Salesforce and Microsoft Office 365 cloud environments.</p> <p>Security Vulnerabilities: Behavioural science-based security awareness to prevent attacks exploiting human vulnerabilities. Tailored to achieve greatest possible improvement in security culture and reduction in risk from employee behaviour.</p> <p>Vulnerability Management: Integrated F-Secure Radar which scans the deep web, fights brand exploit, and reduces total costs.</p> <p>PCI DSS compliance: requirements are filled with F-Secure Radar reducing the risk of potential data loss</p> <p>(Vulnerability scanning and management platform - Radar, 2020)</p>

<p>tripwire</p>	<p>File Integrity & Change Monitoring: Automatically assesses and prioritizes detected changes with real-time data and security automation</p> <p>Vulnerability & Risk Management: Identify risk that needs a response by combining business context with vulnerability intelligence.</p> <p>Cloud Security: Deploy consistent security controls across physical, virtual, private, and public cloud infrastructures.</p> <p>DevOps Security: Secure DevOps workflow without increasing release cycle times.</p> <p>File Analyzer: Automated analysis of the file and executable behaviour.</p> <p>Configure & Harden Systems: Continuous system hardening through Security Configuration Management.</p> <p>Cybersecurity For Industrial Control Systems: Ensure the security, safety, and availability of industrial environments from malicious attacks and operational errors.</p> <p>Network Asset Discovery: Discovering what is on your network enables you to harden your IT infrastructure and monitor the integrity of those systems, improving system availability and uptime.</p> <p>Incident Detection & Investigation: Real time intelligence used as an indicator for detecting an incident and assessing its scope.</p> <p>Log Management: Centralized log management.</p> <p>CIS Critical Security Controls: Prioritised approach to security Controls.</p> <p>(Cybersecurity Compliance and Industry Needs Tripwire, 2020)</p>
<p>GFI</p>	<p>Patch management across multiple operating systems.</p> <p>Patch management for multiple third-party apps.</p> <p>Patch management for multiple web browsers.</p> <p>Network security scanning with a recommended course of action.</p> <p>Web based reporting interface.</p> <p>Vulnerability assessment database with information from BugTraq, SANS Corporation, OVAL, CVE, and others.</p> <p>Vulnerability check on networked devices.</p> <p>Network auditing.</p> <p>Security audits.</p> <p>Works in virtual environments.</p> <p>PCI DSS regulation compliance.</p> <p>Reporting in formats such as PDF, HTML, XLS, XLSX, RTF and CSV, and can be scheduled and sent by email.</p> <p>Run agent less or agent-based mode. Agent technology enables automated network security audits and distributes the scanning load across client machines.</p> <p>(GFI,2020)</p>
<p>Breachlock</p>	<p>Penetration testing as a service: Web application pen test, network pen test, mobile app pen test, OWASP based methodology, PCI DSS, HIPAA, SOC2 compliant, online support with security experts.</p> <p>Rata web app scanning: AI-enabled deep web scans, OWASP top 10 coverage, authenticated scans, fast and accurate crawler, API security scanning,</p> <p>Rata network scanning: AI-enabled network scans, port scanning and enumeration, scan external and internal network, patch validation.</p>

	DevOps Centric: On demand and scheduled scans, on demand penetration tests, JIRA, Slack, Trello integration, patch validation. Findings listed inside BreachLock SaaS (How It Works, 2020)
--	--

Tools that are currently on the market and a listing of programs that have these tools implemented:

Tools	Programs						
	Qualys	Rapid7	Tenable	F-Secure	Tripwire	GFI	Breachlock
Asset inventory	✓	✓	✓	✓	✓	✓	✓
Network asset discovery	✓	✓	✓	✓	✓	✓	✓
Certificate inventory	✓	✓	✓	✓	✓	✓	
Asset grouping	✓	✓	✓	✓	✓	✓	
Asset summary	✓	✓	✓	✓	✓		
Various vulnerability assessment scan tool	✓	✓	✓	✓	✓	✓	✓
Authenticated scan	✓	✓	✓	✓	✓	✓	✓
Unauthenticated scan	✓	✓	✓	✓	✓	✓	✓
File integrity scan	✓	✓	✓	✓	✓	✓	✓
Misconfiguration detection	✓	✓	✓	✓	✓	✓	✓
Various forms of attack detection such as XSS, SQL injection	✓	✓	✓	✓	✓	✓	✓
Behaviour based attack detection	✓	✓		✓			
Vulnerability inventory	✓	✓	✓	✓	✓	✓	✓
Ability to add in vulnerabilities that the vulnerability scan does not find	✓		✓	✓			
Vulnerability categorisation	✓	✓	✓	✓	✓	✓	
Vulnerability history	✓	✓	✓		✓		

Vulnerability age	✓	✓	✓		✓		
Risk and prioritisation calculation	✓	✓	✓	✓	✓	✓	✓
Threat intelligence news feed	✓	✓	✓	✓	✓	✓	
Indicators of compromise	✓	✓	✓	✓	✓		
Patch management	✓	✓	✓	✓	✓	✓	✓
Ability to assign a vulnerability to a user	✓	✓	✓	✓			✓
Patch catalogue	✓						
Failed patch install inventory	✓	✓	✓	✓	✓	✓	
Report dashboard	✓	✓	✓	✓	✓	✓	✓
Ability to distribute a report to a team through the application	✓	✓	✓	✓	✓	✓	✓
Live vulnerability remediation progress dashboard	✓	✓	✓	✓		✓	
Quarantined vulnerability database	✓	✓	✓	✓			

Tools that I would like to implement in my application and what languages you can create these tools with:

Tools	Development Languages					
	C#	Java	JavaScript	Python	PHP	Ruby
Asset inventory	✓	✓	✓	✓	✓	✓
Network asset discovery	✓	✓	✓	✓	✓	✓
Certificate inventory	✓	✓	✓	✓	✓	✓
Asset grouping	✓	✓	✓	✓	✓	✓
Asset summary	✓	✓	✓	✓	✓	✓
Vulnerability inventory	✓	✓	✓	✓	✓	✓

Ability to add in vulnerabilities that the vulnerability scan does not find	✓	✓	✓	✓	✓	✓
Vulnerability categorisation	✓	✓	✓	✓	✓	✓
Vulnerability history	✓	✓	✓	✓	✓	✓
Vulnerability age	✓	✓	✓	✓	✓	✓
Risk and prioritisation calculation	✓	✓	✓	✓	✓	✓
Threat intelligence news feed	✓	✓	✓	✓	✓	✓
Patch management	✓	✓	✓	✓	✓	✓
Ability to assign a vulnerability to a user	✓	✓	✓	✓	✓	✓
Failed patch install inventory	✓	✓	✓	✓	✓	✓
Report dashboard	✓	✓	✓	✓	✓	✓
Ability to distribute a report to a team through the application	✓	✓	✓	✓	✓	✓
Live vulnerability remediation progress dashboard	✓	✓	✓	✓	✓	✓
Quarantined vulnerability database	✓	✓	✓	✓	✓	✓

Development Perspective

When developing this program there are two types of application, a desktop application, and a web-based application.

Types of Applications

A desktop application comes with the requirement that it must be installed on a computer before it can be used. It is a computer program and does not need internet to be used.

A web application requires internet connection and does not need to be installed on a machine, as it runs in your web browser.

Factors to Consider

The following factors must be considered when deciding whether to develop a web-based application or a desktop application:

Access: Web applications are accessible from any location where there is an internet connection. Whereas desktop applications are confined to the machine in which they are installed and a physical location.

Maintenance: Web applications never need to be updated. Desktop applications still require manual download and update and install regularly. Also, web applications only need to be installed once comparing to desktop applications need to be installed separately on every computer.

Security: Standalone desktop applications are protected from various vulnerabilities through regular updates. Web based applications are continually exposed to a large number of users on the internet creating a greater amount of risk and threats.

System Requirements: Web applications solely rely on the internet thus system requirements are not a necessity. Desktop applications on the other hand have requirements such as memory, updated systems, specific system types etc.

Cost: Desktop applications cost a lot to develop. Web based applications are often cheaper to make and take less time to build.

Connectivity: Web applications rely significantly on internet connectivity and speed. Performance issues may be present with poor internet connectivity. Desktop applications are standalone and do not face any issues relating to internet connectivity.

I would like to create a web application as personally I feel there are much more advantages and ease for creating a web application. I would like my application to be accessible from anywhere. Another feature would be the ability to connect to devices and assess them and I can do so from my location and test any device anywhere in the world once there is internet connectivity. If I want to deploy anything it will go straight to every device or company that is connected to my network, I will not have to install each deployment separately on every device. It would be an overall more efficient choice for my application.

Development Languages

Web Application Architecture

Web application architecture is considered in three layers. These are the presentation layer, business logic layer and data layer.

Presentation Layer

This layer is also known as the front-end layer. It portrays and creates the look and feel of the application. It is the practice of creating user interfaces for a user to interact with. This layer contains the client-side code which is the code in the browser that responds to user input. The client side refers to the representation of the web applications functionality that the end user interacts with. It consists of the languages HTML, CSS, and JavaScript.

Presentation Layer Development Languages

HTML: Hyper-text markup language is the standard markup language for creating static web pages. It describes the structure of a web page and consists of a series of elements which tell the browser how to display the content. It is the foundation of the web page defining elements such as headings, paragraphs and embedding media. HTML is what sends data back and forth to the browser/clients through Hypertext Transfer Protocol (HTTP).

CSS: Cascading Style Sheets describes how HTML elements are to be displayed on the screen by specifying the documents style such as colours, fonts, and page layouts. While HTML is the foundation of a web page and CSS interacts with these elements creating the design and aesthetic on top of the foundation.

JavaScript: is a scripting or programming language which allows you to implement complex features on web pages. It allows the creation of dynamically updating content, implementation of multimedia control and much more. It can store values in variables, run code in response to events and run operations on pieces of text. Application Programming Interfaces (API's) work on top of client-side JavaScript and can be used to implement and expose data from surrounding computer environment (What is JavaScript?, 2020). Third Party API's are not built into the browser, but their code can be grabbed and inserted into your browser to add more features to your web page.

Business Logic Layer

The business logic layer is also known as the application layer of a web application. This layer is the back-end layer which implements the core functionality of the system and is responsible for encapsulating the relevant business logic (What are the three general layers that make up a web application architecture? - Skillset, 2020). It is the server-side code which is the code on the server that responds to the HTTP requests. This can be thought of as the plug between the web page and the database, that responds to a request from a user using the web page and then performs its middle layer code to retrieve data from the database and then carry out the request functionality with the data back in the web page. It is where the business and presentation logic work together to deliver a response back to the user.

Business Logic Layer Development Languages

Server code is written in the following languages:

C#: is a general purpose, object-oriented component-oriented programming language. It is used for a variety of purposes for example backend services, windows applications, website

development, mobile development, and game development. The ASP.NET windows framework uses C# as a programming language at the server side, used for creation of web applications, dynamic web sites on the .NET platform, web services or open source software. C# was created by Microsoft to adopt the best features of both Java and C++ which created a huge library providing more high-level functionality than Java and C++.

Java: Java is an object-oriented programming language which gives clear structure to programs and allows code reuse. The Java compiler is architecture neutral with no implementation dependent aspects allowing it to run on a variety of platforms including Windows, Mac, Linux, and Raspberry Pi. Java is used to develop mobile, desktop and web applications, web and application servers, games, and database connection. It is considered more dynamic than C or C++ due to its design to adapt to an evolving environment.

JavaScript: is an interpreted scripting language. It is commonly used as part of web pages to create and control dynamic web content and is the functionality which allows client-side script to interact with the user of a web page. The client-side mechanism provides many advantages comparing to CGI server scripts. JavaScript is mainly used to add interactivity to websites, develop mobile applications and browser-based games and is also used for back end development.

Python: is a general purpose, high level programming language that is interpreted and object-oriented. It eases of use design philosophy emphasises code readability. With high-level built-in data structures, dynamic typing and dynamic binding allow it to be used for web development, data science and software prototypes.

PHP: Hypertext PreProcessor is a widely used scripting language. It is often used for application or web development in data heavy websites. It is used to manage, and process data and its scripts are only executed on the server. It is compatible with almost all servers and runs on various platforms including Windows, Mac, Linux, and Unix.

Ruby: is a dynamic object-oriented programming language. It is similar to C and Java and runs on platforms such as Windows, Linux, and Mac. It is used for web applications.

Data layer

The data layer is also known as the database storage layer. It is a centralized location that receives data calls and stores all persistent data relating to the application. This layer provides access to the data hosted for the application in the database. The data layer works closely with the business logic layer where the logic responds to requests from the presentation layer and then knows which database to talk to and retrieve data from. The technologies used in the data layer are database management systems. A database stores an organised collection of structured information otherwise known as data. There are many different types of databases with different uses for data. The best database for a specific organisation will depend on how the data is intended to be used by the organisation.

Types of Databases

The following are the various types of databases:

Relational databases: Data stored in a relational database is organised into tables which can then be linked or create relationships between the data. This technology provides the most efficient way to access structured information.

Object-oriented databases: Data in an object-oriented database is represented in the form of objects.

Distributed databases: A distributed database contains the ability to store two or more files located in different sites. Therefore, data may be stored on multiple computers in the same physical location or scattered over different networks (What is a database?, 2020).

Data warehouses: This form of database is designed for fast query and analysis.

NoSQL databases: Otherwise known as a nonrelational database, NoSQL allows unstructured and semi structured data to be stored and manipulated.

Graph databases: A graph database stores data in terms of entities and the relationships between entities (What is a database?, 2020).

OLTP databases: An OLTP database is a speedy, analytic database designed for large numbers of transactions performed by multiple users.

Open source databases: Open source databases are where the source code of the database system is open source. They can use either SQL or NoSQL.

Cloud databases: A cloud database is a collection of data, either structured or unstructured, that resides on a private, public, or hybrid cloud computing platform (What is a database?, 2020).

Multimodel database: Just like the name suggests these databases combine different database models into a single integrated model to accommodate various data types.

Document/JSON database: Modern way to store data in JSON format.

Self-driving databases. The newest form of database are self-driving databases, they are cloud-based and use machine learning to automate database tuning, security, backups, updates, and other routine management tasks traditionally performed by database administrators (What is a database?, 2020).

Database Software's

The most popular database software's include the following:

MYSQL: MYSQL is an open source relational database that is used to store up to a single record of data or a whole inventory of data used for web applications.

Microsoft Access: Microsoft Access is a database management system owned by Microsoft which combines the relational Microsoft Jet Database Engine with a graphical user interface and software development tools.

SQL Server: Microsoft SQL Server is a relational database management system with the primary function of storing and managing data.

Oracle: Oracle database is a multi-model database management system. It is used for various actions such as online transaction processing, data warehousing and mixed database model workloads.

A database query language must be used in conjunction with the database management system for capability of maintaining the stored data. SQL is the database query language used for

maintaining a database. It allows you to access data in the database, gather data from different databases and update the data stored in the database.

Large enterprises today support very complex queries and with this comes a lot of challenges. When creating a database I will have to employ a variety of methods to improve performance and maintain the database so that it does not run into the following challenges: data volume overload, data breaches, fast paced demand of data, data scalability.

Composition of a Vulnerability Management Application

The differentiation between a good and great vulnerability management program is a strong integration of both including risk and prioritising endpoint groups but also integrating with other key business and technical systems and processes (Viegas, 2020). Multiple factors contribute to a vulnerability management program however the four components of the vulnerability management lifecycle are key in effect remediation. In this section I am going to research the four areas in depth and discuss the multiple factors that relate to these which are Discover, Analyse, Prioritise, Remediate.

Types of Security Concerns

There are three crucial areas of security which are regarded as impacting factors towards an organisations information system which must be monitored closely, these include:

Vulnerabilities: A vulnerability is a hole or weakness in an application which if left unapproached may be used by a malicious entity to cause harm towards a stakeholder.

Remediations: Remedial action is the act of correcting something that has been corrupted or that is deficient.

Threats: A threat is any potential negative action or event which may harm or cause unwanted impart towards a computer system or application.

In order to mitigate these pivotal security concerns a vulnerability management program requires the following areas of management:

Discover

Inventory management

Evaluating all of the assets of a company is critical to providing vulnerability management.

If an asset is not in the inventory it cannot be patched and therefore a vulnerability will be present and able to lurk through a company's system, all from that one missing device. The inventory system will create an organised structure of the company for all of the processes following this step, it allows the various endpoints to be grouped into various classifications such as servers and workstations. This classification will result in an organised infrastructure with the ability to easily determine the hardware equipment, operating systems and software applications used in the organization. Having a system inventory enables effective management when monitoring for vulnerabilities, patches, and threats. An inventory of assets sets an evaluation baseline for the application and should include all hardware, software, applications, services, and configurations used by an organisation. For accurately stored data it is recommended that the location of the asset and details regarding the sensitivity level of the data are also stored. An accurate inventory ensures accurate patching and remediation.

The following list is the scope of items to test for regarding company resources (an asset) and include in the inventory:

1. Associated system name
2. Property number
3. Owner of the IT resource (i.e., main user)
4. System administrator
5. Physical location
6. Connected network port
7. Software configuration
 - a. Operating system and version number
 - b. Software packages and version numbers
 - c. Network services
 - d. Internet Protocol (IP) address (if it is static)
8. Hardware configuration
 - a. Central processing unit
 - b. Memory
 - c. Disk space
 - d. Ethernet addresses (i.e., network cards)
 - e. Wireless capability
 - f. Input/output capability (e.g., Universal Serial Bus, Firewire)
 - g. Firmware versions

(Mell, Bergeron, and Henning, 2005).

Analyse

Vulnerability Assessments

Vulnerability identification is performed through vulnerability assessments. Vulnerability assessments are performed using industry standard scanning tools and systems or manual assessment. Performing vulnerability scans will help ensure that known vulnerabilities in your system are identified and addressed in a timely manner, reducing your organization's risk exposure to an acceptable level (Vulnerability Assessment, 2020). There are two approaches to assessing an information system, these are through automated assessment or manual assessment.

Automated Assessment may be carried out with the use of a tool to assess the security posture of an information system using pre simulated attacks/scripts against the system. The findings of automated assessment will be presented in a report upon completion.

Manual Assessment assesses all the features of the chosen test subject and is performed manually by a human instead of a script.

Vulnerability Scanners

Vulnerability scanners are used by many organisations to assess two key areas in their information system, their hosts, and networks. They carry an array of capabilities to identify multiple aspects of a subject's situation ranging from active hosts on networks and vulnerable ports on hosts, to vulnerabilities on a discovered operating system and compliance testing of host application security policies. A vulnerability scanner can identify both vulnerabilities and misconfigurations on the organisations network and hosts with the following:

Network Scanners:

Network scanners are used to map an organisations network and its topology. The topology mapped out will allow an organisation to locate and test the numerous hosts attached to its network. Network scanners also perform probes of network services, operating systems, routers, switches mail servers, web servers and firewalls to identify open ports, vulnerable software, and misconfigured services.

Host Scanners:

Host scanners perform assessment of specific hosts operating systems, installed applications and programs to identify vulnerabilities, misconfigurations, and constituents of systems such as out of date software versions.

Upon choosing which environment you will be assessing, there are various types of scans to assess infrastructure and ensure compliance with regulations such as the PCI Security Standards Council regulations, which may be carried out such as internal and external scans, authenticated and unauthenticated scans, network based scans, host based scans, wireless scans, application scans and database scans.

Scans

Internal and External Scans

Internal scans assess the inner perimeter of the organisation to find vulnerabilities that may be exploited by attackers who have gained access through the external perimeter of the network. The vulnerabilities may also be exploited by insider threats already inside the network.

It is essential to conduct external scans in prevention of attackers gaining access to the organisations network. External scans scan the perimeter of the organisations network in search of open ports or flaws in the application firewall.

Authenticated and Unauthenticated Scans

Scanning should be carried out with both authenticated and non-authenticated scans.

Authenticated scans are performed with higher permission or an administrator account. This form of scan is a deeper analysis of assets which can determine software configurations and versions.

Unauthenticated scans are scans performed while logged out of the system. They can determine limited system information such as basic configurations and open ports. Unauthenticated scans

represent the type of analysis that would be performed by an attacker that has not yet obtained valid credentials, but is attempting to identify vulnerabilities in a system to exploit (Hallmarks of Successful Threat & Vulnerability Management Programs, 2020).

Network Based Scans

Network scans identify possible weaknesses in the network security infrastructure that may be attacked and the vulnerable systems on the network.

Host Based Scans

Host based scans identify and locate vulnerabilities in workstations, servers, and other network hosts, providing visibility into exploitable configuration settings and previous patch history of already scanned systems.

Wireless Scans

Wireless scans establish rogue access points and verifies if a company's network is securely configured.

Application Scans

Application scanning detects known software vulnerabilities and erroneous configurations in network or web applications.

Database Scans

Database scans identify vulnerabilities and weak points in a database.

Performing a Vulnerability Scan

Vulnerability scans should be performed at least regularly if not continuously. According to PCI standards vulnerability scans should be performed "at least quarterly" in order to remain compliant with regulations (How Often Should You Run a Vulnerability Scan? | PCI Pal, 2020).

Scope of a Vulnerability Assessment

When assessing assets for vulnerabilities the following list of attributes should be considered:

- The name of the vulnerability
- The date of discovery
- The time of discovery
- Type of vulnerability
- A detailed description of the vulnerability
- Payload of vulnerability
- Details regarding the affected systems
- Details regarding the process to correct the vulnerability
- A proof of concept (PoC) of the vulnerability for the system (if possible)

- The risk score based on Common Vulnerabilities and Exposures (CVE) databases.

Negative Aspects of Vulnerability Scanning

Two errors that may occur with vulnerability scanning are:

False positives: an erroneous result indicating that there is a vulnerability when there is not.

False negatives: an erroneous result indicating there is no vulnerability present when in actuality there is a vulnerability present.

Comparing found vulnerabilities against an inventory is a procedure to avoid these errors. False positives can be avoided by matching known vulnerabilities against industry standard vulnerability databases such as Common Vulnerabilities and Exposures database or the NIST National Vulnerability Database. Another advisory to match against is CERT advisories.

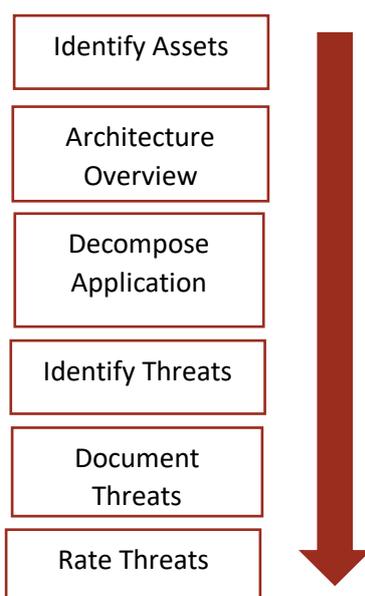
Prioritise

Risk Management

Risk is defined as the potential impact and consequences of an event and the probability of the event happening. Risk management is the act of prioritising vulnerabilities and assets in relation to their impact upon exploitation against an organisation. In order to appropriately prioritise risk and the security of assets, threat modelling must be applied to any asset in the organisation. Threat modelling works to identify, communicate, and understand threats and mitigations within the context of protecting something of value (Application Threat Modelling | OWASP, 2020). Forming a threat model for a vulnerability management application is vital to systematically identify and rate the threats that are most likely to affect an organisation. The identification and rating of threats based on an understanding of a systems architecture, the threats can be mitigated with appropriate measures in a logical order, starting with the threats that present the greatest risk (Threat Modelling, 2020).

Threat Modelling Process

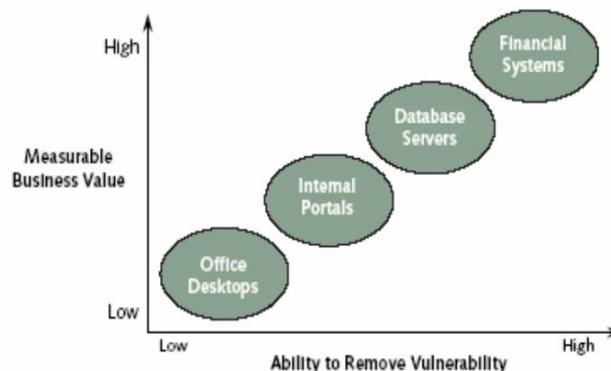
An effective threat modelling process is the following:



(Threat Modelling, 2020)

Grouping and Prioritising Assets

When creating a system inventory grouping and prioritising of assets for scans will make it not only easier to separate systems and know which asset belongs in which area but also prioritising assets is essential for assessing risk towards systems. Assigning priority levels of assets is used in the patch management stage of a vulnerability management program to prioritise which assets will require priority remediation based on the ideology that an identified threat will potentially cause harm and impact toward the business. It is extremely important to group assets which have a direct impact on business risk. For example, the server hosting your web application is much more important than the desktop in your office. Assets must be prioritised based on their business value. This ensures that if a vulnerability is found on a critical asset which will cause more impact towards the business than a vulnerability found on an asset that will cause less impact, the higher risk asset will be prioritised first for remediation to potentially save further cost for the organisation than if it were left to handle a lower impact risk asset first. The following diagram represents an information system infrastructure risk level measured by business value as per the recommendation of Qualys.



(Prioritizing assets by business value, 2020)

Assets of higher priority are also often grouped into the following categories:

- Assets essential for system operation (e.g., servers)
- Assets used for security management
- Assets residing on the organization's network boundary
- Assets that contain information of higher importance
- Assets that are accessible to external users.

(Creating a Patch and Vulnerability Management Program, 2020)

Calculating Asset Value

The Federal Information Security Management Act defines three security objectives for information and information systems, these are:

Confidentiality

“Preserving authorized restrictions on information access and disclosure, including means for protecting personal privacy and proprietary information...” [44 U.S.C., Sec. 3542]

A loss of confidentiality is the unauthorised disclosure of information (FIPS 199, Standards for Security Categorization of Federal Information and Information Systems, 2004).

Integrity

“Guarding against improper information modification or destruction, and includes ensuring information non-repudiation and authenticity...” [44 U.S.C., Sec. 3542]

A loss of integrity is the unauthorized modification or destruction of information (FIPS 199, Standards for Security Categorization of Federal Information and Information Systems, 2004).

Availability

“Ensuring timely and reliable access to and use of information...” [44 U.S.C., SEC. 3542]

A loss of availability is the disruption of access to or use of information or an information system (FIPS 199, Standards for Security Categorization of Federal Information and Information Systems, 2004).

Asset prioritisation and impact analysis can be further calculated through the loss rating of confidentiality, integrity, and availability with data classification. This is weighted in the following measures:

Asset Value	Severity Description
Catastrophic (5.0)	Severe impact to operations, extended outage, permanent loss of resource, triggers business continuity and/or public relations procedures, complete compromise of information, damage to reputation and/or significant cost to repair with continuity of business in jeopardy.
Major (4.0)	Serious impact to operations, considerable system outage, compromise of a large amount of information, loss of connected customers, lost client confidence with significant expenditure of resources required to repair.
Moderate (3.0)	Some impact to operations, tarnished image, and loss of member confidence with significant effort to repair.
Minor (2.0)	Small but tangible harm, may be noticeable by a limited audience, some embarrassment, with repair efforts absorbed into normal operations.
Insignificant (1.0)	Insignificant impact to operations with minimal effort required to repair, restore, or reconfigure.

(Kouns, 2013)

Asset Value is calculated with the following formula:

Asset Value = Impact to the asset from a breach in: Confidentiality (value weight) + Integrity (value weight) + Availability (value weight) = Result ÷ 3 (Asset Value Score)

For example:

Asset Name	Data Classification	Impact to the Asset from a Breach in Confidentiality 5.0 Very High; 4.0 High; 3.0 Medium; 2.0 Low; 1.0 Very Low	Impact to the Asset from a Breach in Integrity 5.0 Very High; 4.0 High; 3.0 Medium; 2.0 Low; 1.0 Very Low	Impact to the Asset from a Breach in Availability 5.0 Very High; 4.0 High; 3.0 Medium; 2.0 Low; 1.0 Very Low	Asset Value SCORE (AV)
Web Server	Sensitive	3.0	4.0	5.0	4.0

(Kouns,2004)

Risk Assessment of a Vulnerability

Risk assessment ties in with vulnerability assessment and is used to calculate the impact of a found vulnerability towards an enterprise and its level in priority regarding remediation. Risk assessment allows an organisation to determine which vulnerabilities are the most dangerous and how to approach remediating them rapidly and effectively.

There are two types of risk analysis these include:

Quantitative

Quantitative risk is the analysis of relevant and verifiable data to produce a numerical value for example a mathematical model which assigns a monetary value to an asset in order to calculate the cost of a threat in reality and the expected loss.

Qualitative

Qualitative risk analysis is a scenario-based model which evaluates risk using potential and probability as opposed to quantitative risk when reliable data is not available to obtain or is too costly to obtain.

For the purpose of the project I will be focusing on qualitative risk analysis method.

Common Vulnerability Scoring System

One of the most common forms of calculating a risk score is the CVSS Common Vulnerability Scoring System. It is an open framework which calculates a numerical score reflecting the severity rating of a vulnerability based on its characteristics. The numerical score provided by CVSS is then translated into a qualitative analysis result to aid organisations in prioritising vulnerability remediation. A CVSS score is composed with three elements of metrics: base, temporal and environmental metrics. Base metrics consider the characteristics of the vulnerability that do not change over time for example the attack vector, attack complexity, privileges required, user interaction and scope. Temporal metrics are the characteristics of a vulnerability that evolve over time such as exploit code maturity, remediation level and report confidence. Environmental metrics capture the characteristics of a vulnerability that relate to system distribution and the network environment. CVSS provides a score for a vulnerability in great complexity.

Another method of calculating risk analysis is with the DREAD matrix.

DREAD

The DREAD model is another form of qualitative risk analysis which can be used to calculate a vulnerability risk rating score. A vulnerability risk score is calculated by taking into account the following factors and answering these questions:

- **Damage potential:** If the vulnerability is exploited how bad is the damage?
- **Reproducibility:** How easy is it to reproduce the attack?
- **Exploitability:** Can the attack be launched easily?
- **Affected users:** How many users will be affected?
- **Discoverability:** How easy is it to discover the vulnerability?

Answering the above items will rate each threat. The DREAD matrix is a simple system in which you can rate a threat on a scale of one being low, two being medium and three being high. The value for each item can be decided using the following criteria created by Microsoft:

	Rating	High (3)	Medium (2)	Low (1)
D	Damage Potential	The attacker can subvert the security system; get full trust authorization; run as administrator; upload content.	Leaking sensitive information.	Leaking trivial information.
R	Reproducibility	The attack can be reproduced every time and does not require a timing window.	The attack can be reproduced, but only with a timing window and a particular race situation.	The attack is very difficult to reproduce, even with knowledge of the security hole.
E	Exploitability	A novice programmer could make the attack in a short time.	A skilled programmer could make the attack, then repeat the steps.	The attack requires an extremely skilled person and in-depth knowledge every time to exploit.
A	Affected Users	All users, default configuration, key customers.	Some users, non-default configuration.	Very small percentage of users, obscure feature; affects anonymous users.
D	Discoverability	Published information explains the attack. The vulnerability is found in the most commonly used feature and is very noticeable.	The vulnerability is in a seldom-used part of the product, and only a few users should come across it. It would take some thinking to see malicious use.	The bug is obscure, and it is unlikely that users will work out damage potential.

(Threat Modelling, 2020)

The risk rating is then acquired by adding the value calculated for each item dividing the total by five and assigning the total average risk value to a discovered rating level. The formula

looks as follows: $D+R+E+A+D/5$ = Average risk rating level. The rating level can be classified by assigning overall ratings of 12–15 as High risk, 8–11 as Medium risk, and 5–7 as Low risk.

For the purpose of this project while CVSS evaluates a more complex rating for risk assessment, I have chosen to primarily focus on the DREAD methodology as I feel it is straightforward in application of the methodology and interpretation of results and focuses on key priority components of a vulnerability to calculate and rate the level of risk associated with the vulnerability.

Vulnerability Management

Triage Vulnerability Group

After completing a vulnerability assessment, it is recommended that a triage vulnerability group should meet to triage all vulnerabilities discovered. The triage group should consist of staff with knowledge of cyber security and business risk and further IT estate management. When performing a vulnerability assessment, a severity rating is calculated for the vulnerability but scan it does not take into account the business risk. The triage team will assess all issues based on all available information. The group is essentially a follow on in risk management. The triage process should involve three sections fix, acknowledge, and investigate.

Fix

These are the issues which after risk assessment will be carried forward to be patched and mitigated. These issues are the prioritised issues and should be assigned a date by which a remediation should be complete.

Acknowledge

These are the issues which after decision will not be immediately patched. The triage group will be required to state reasons as for not doing so immediately at present.

Investigate

These issues are assigned investigate as a temporary status until further investigation. They could be predicted false positives or the cost of resolving the issue could not be known. The time scale designated to this category will depend on the calculated severity.

The triage group helps to identify risk calculation with ease. The group will come to an efficient calculated conclusion based on all calculated efforts and make valid business decision for which risk category a vulnerability should be placed.

Reports

Upon Identifying a vulnerability after conducting a vulnerability assessment, a report is generated which contains detailed information and the context of the vulnerability. The aim of the report is to find efficient solutions towards alleviating the vulnerability and enhancing the organisations security posture. The report should be complex for security engineers, but understandable for both technical and non-technical stakeholders as it will be shared with regulatory bodies and management in the process. The first part to a report is the executive summary. This will outline the overall risk of the organisation and the severity of the discovered vulnerability. It should then include items such as the assessment scope, assessment findings, primary objectives, dates and times of the assessment, assessment details including

the scanned asset etc, The next section should be the assessment overview which should contain clear and concise information of the investigation and it's deliverables. The generated report should then include the following information regarding the vulnerability:

- Vulnerability Name
- Vulnerability Date of Discovery
- Vulnerability Score
- A detailed description of the vulnerability including the affected systems.
- Details of the process to remediate the vulnerability.
- Proof of concept for the vulnerability system.

If a report is weak, a response plan will be weak thus it is essential to generate a report that targets all the right points.

Remediate

Patch Management

The purpose of patch management is to classify and then prioritise patching of an asset. Patching can be automated or performed manually by a patch management team. Before applying a patch, the remediation should be tested and verified that it corrects the vulnerability without harming the business operations and applications. Organisations may sometimes delay vulnerability patching due to a variety of reasons however delaying can sometimes result in fatal complications. It is important to remediate as quickly as possible. When deploying a patch, it should be deployed enterprise wide to target remediating the discovered vulnerability everywhere it has been found. Sometimes it is often just the requirement of an update to patch a system thus an organisation should monitor third party vendors continuously for topical information regarding vulnerabilities, threat intelligence and remediations. While deploying a patch it is essential to ensure all affected assets are remediated, this can be identified by working with the system asset inventory. After deployment of patches, the patch management team must re-assess all IP connected assets to ensure patching and remediation was successful and that the patch did not cause malfunction to other devices, services, or applications. Verification of fixes and a newly created re-assessment report provides documentation for compliance with laws and regulations. A mature patch management process will make the organization more proactive than reactive regarding maintaining appropriate levels of security for their systems. The efficiency of patch automation combined with preventative maintenance should result in spending less time, resources, and money on incident response (Creating a Patch and Vulnerability Management Program, 2020).

Managing a Vulnerability Management Application

Managing a vulnerability management application requires dealing with various levels of sensitive information and also high risk situations and components of a business, such as altering assets and infrastructure for patching and dealing with vulnerabilities that could potentially wipe out an organizations system. Dealing with these levels of risk require authority, communication between management and stake holders and different levels of agreements and disclosures. The following items must be provided and agreed before any processes can proceed:

Policies: Policies outline the procedures and rules regarding course of action before an event such as for example a vulnerability assessment can take place. A policy validates the purpose of an event and herein the professionals who will be granted permission to participate in the event. It will then provide the agreed strategy and procedural protocol/ standards to be followed for the event to be conducted as per the authorisation and proposal of the organization or an individual in a managerial position.

Policy Disclaimer Statement: A disclaimer statement will assert that the organization will not be held responsible for something such as inaccuracies. Policy disclaimer statements are important in business correspondence.

Liabilities: Liabilities are the opposite of a disclaimer and state the responsibility of a subject in any action or event which may possibly have an adverse impact on business operations. Liabilities are for both the organization and for example the vulnerability assessment tester.

Nondisclosure Agreements: Any individual involved in any test or event and especially an external individual involved in any assessment or event occurring in the vulnerability management application process must sign a nondisclosure agreement before proceeding. Confidentiality and nondisclosure agreements are extremely important to ensure that the organizations information is treated with a high level of confidentiality, they also provide cover in the possibility of something becoming disruptive and areas such as negligence and liability in the event of mishaps.

Privacy Policy: A privacy policy is a statement or legal document that discloses and states the use of data and reasoning for collection of data. A privacy policy is an obligation in compliance with laws and regulations such as GDPR.

Stakeholder Communication: This document will validate the selected personnel in a stakeholder position which will be involved in any aspect of the vulnerability management process. It is important to keep this up to date and have all involved personnel approved by senior management.

Authorised Approvals and Signatures: It is necessary that all plans and procedures for processes are agreed upon and signed by relevant authority.

Rules of Engagement: This document will be the terms and conditions outlined for a process. For example, if a specific grouping of assets is to be excluded from a vulnerability assessment it must be outlined in a rules of engagement document.

Test Plans: All planning for tests must include items such as who will be conducting the test, purpose for the test, the tasks assigned in the test, the test schedule, a contact for disaster recovery should a disaster occur, methodology of test, assumptions, scope of test, and then the applicable laws, regulations, standards and guidelines which are required to be followed throughout the process of a test.

Each of these items should be involved and applied in the processes of a vulnerability management application to remain honest and compliant regarding laws and regulations and to maintain a level of professionalism in an organization.

Conclusion

I have chosen to approach this paper in the manner of first researching a vulnerability, how it can be exploited, its role and position in the exploitation process and I have then investigated how vulnerability management works. Following from so I have researched the different development technologies to create a vulnerability management application and finally I have decomposed the components of a vulnerability management tool so that I have a deeper understanding of the different factors and sections to take into consideration when developing my own application. For my project I hope to create a web application which facilitates vulnerability management. I would like to provide an organisation with a tool that contains both test management and vulnerability management. The ideology of my tool is that it will track a test process and vulnerability process from start to finish, for example it will manage the test from setting up a test, to ensuring all legal factors and policies are complied with, then further a vulnerability test will be given the go ahead and my tool will track its progress in a system from start to finish. It will provide information about a test, when it was or is to be conducted and then I will provide a vulnerability dashboard to display the vulnerabilities found in the associated asset in which the test was conducted, the application will then further show the prioritisation levels of a vulnerability or asset and it will be placed in a category of risk before patching. I will then have a dashboard where a manager can log in and assign a vulnerability to a team or user globally, the final section of the application will display a live feed of the progress of vulnerability remediation. The owner of the vulnerability will be able to update this feed in accordance with the vulnerabilities progress. In doing so I am providing an organisation with a management tool for vulnerabilities on their system, essentially providing them with an area that states what needs fixing and how important and immediately the vulnerability must be patched. It is a vulnerability management tool which conveys the process of the audit from beginning to finish, an application which is not particularly used today and if it is used by an application it is only used in part. In organisations today, sensitive data and assets connected to the internet can be exploited at any minute, as stated earlier in this paper a cyber-attack occurs almost every thirty nine seconds in the world today, that's almost enough time for two cyber-attacks per minute and with one thousand four hundred and forty minutes in one day alone you can see how dangerous this is considering one cyber-attack can wipe a whole system. A vulnerability management tool can provide an interface for managing assets and data to track their vulnerabilities and remediate just in time to prevent cyber-attacks from taking place, if we fix what we know as soon as possible it can save time before an attacker takes the time to find and exploit a vulnerability.

Bibliography

2004. *FIPS 199, Standards For Security Categorization Of Federal Information And Information Systems*. [ebook] National Institute of Standards and Technology. Available at: <<https://csrc.nist.gov/csrc/media/publications/fips/199/final/documents/fips-pub-199-final.pdf>> [Accessed 10 November 2020].

2020. *Anatomy Of Cyber Threats, Vulnerabilities, And Attacks*. [ebook] pp.4,5. Available at: <<https://go.recordedfuture.com/hubfs/white-papers/cyber-anatomy.pdf>> [Accessed 29 October 2020].

2020. *Anatomy-Of-An-Application-Security-Weakness.Pdf*. [ebook] Synopsys, p.1. Available at: <<https://www.synopsys.com/blogs/software-security/types-of-security-vulnerabilities/>> [Accessed 28 October 2020].

2020. *Creating A Patch And Vulnerability Management Program*. [ebook] National Institute of Standards and Technology. Available at: <<https://csrc.nist.gov/library/alt-SP800-40v2.pdf>> [Accessed 10 November 2020].

2020. *Prioritizing Assets By Business Value*. [image] Available at: <https://www.qualys.com/docs/guide_vulnerability_management.pdf> [Accessed 10 November 2020].

Breachlock.com. 2020. *How It Works*. [online] Available at: <<https://www.breachlock.com/how-it-works/>> [Accessed 2 November 2020].

Bsigroup.com. 2020. *Vulnerability Assessment*. [online] Available at: <https://www.bsigroup.com/en-IE/our-services/cybersecurity-information-resilience/Services/Vulnerability-Assessment/?creative=464620825882&keyword=vulnerability%20assessment&matchtype=p&network=g&device=c&gclid=CjwKCAiA4o79BRBvEiwAjteoYMagRdXZtNm-pPADNB-JtWYN0JtgxsN0w7sCYau-LfHbYgvy-tPnBhoCsjkQAvD_BwE> [Accessed 5 November 2020].

Cdc.gov. 2020. *Vulnerability Management Life Cycle | NPCR | CDC*. [online] Available at: <<https://www.cdc.gov/cancer/npcr/tools/security/vmlc.htm>> [Accessed 21 October 2020].

Csrc.nist.gov. 2020. *Vulnerability - Glossary | CSRC*. [online] Available at: <<https://csrc.nist.gov/glossary/term/vulnerability>> [Accessed 19 October 2020].

Dark Reading. 2020. *Where Do Security Vulnerabilities Come From? - Dark Reading*. [online] Available at: <<https://www.darkreading.com/partner-perspectives/f5/where-do-security-vulnerabilities-come-from/a/d-id/1329951>> [Accessed 28 October 2020].

Docs.microsoft.com. 2020. *Threat Modeling*. [online] Available at: <[https://docs.microsoft.com/en-us/previous-versions/msp-n-p/ff648644\(v=pandp.10\)?redirectedfrom=MSDN](https://docs.microsoft.com/en-us/previous-versions/msp-n-p/ff648644(v=pandp.10)?redirectedfrom=MSDN)> [Accessed 10 November 2020].

Fireeye, 2015. [image] Available at: <https://www.fireeye.com/blog/executive-perspective/2015/11/hallmarks_of_success.html> [Accessed 5 November 2020].

FireEye. 2020. *Hallmarks Of Successful Threat & Vulnerability Management Programs*. [online] Available at: <https://www.fireeye.com/blog/executive-perspective/2015/11/hallmarks_of_success.html> [Accessed 5 November 2020].

F-secure.com. 2020. *Vulnerability Scanning And Management Platform - Radar*. [online] Available at: <<https://www.f-secure.com/en/business/solutions/vulnerability-management/radar>> [Accessed 2 November 2020].

Gfi.com. 2020. Available at: <<https://www.gfi.com/products-and-solutions/network-security-solutions/gfi-languard/specifications>> [Accessed 2 November 2020].

Kouns, 2004. [image] Available at: <https://rvasec.com/slides/2013/Kouns-Risk_Assessment.pdf> [Accessed 10 November 2020].

Kouns, B., 2013. [ebook] Available at: <https://rvasec.com/slides/2013/Kouns-Risk_Assessment.pdf> [Accessed 10 November 2020].

MDN Web Docs. 2020. *What Is Javascript?*. [online] Available at: <https://developer.mozilla.org/en-US/docs/Learn/JavaScript/First_steps/What_is_JavaScript> [Accessed 3 November 2020].

Mell, P., Bergeron, T. and Henning, D., 2005. *Creating A Patch And Vulnerability Management Program*. [ebook] Available at: <<https://csrc.nist.rip/library/alt-SP800-40v2.pdf>> [Accessed 5 November 2020].

Oracle.com. 2020. *What Is A Database?*. [online] Available at: <<https://www.oracle.com/ie/database/what-is-database.html>> [Accessed 4 November 2020].

Owasp.org. 2020. *Application Threat Modeling | OWASP*. [online] Available at: <https://owasp.org/www-community/Application_Threat_Modeling> [Accessed 5 November 2020].

Owasp.org. 2020. *OWASP Top Ten Web Application Security Risks | OWASP*. [online] Available at: <<https://owasp.org/www-project-top-ten/>> [Accessed 29 October 2020].

PCI Pal. 2020. *How Often Should You Run A Vulnerability Scan? | PCI Pal*. [online] Available at: <<https://www.pcipal.com/en/knowledge-centre/news/often-run-vulnerability-scan/#>> [Accessed 10 November 2020].

Qualys.com. 2020. *Qualys Multi-Vector EDR | Qualys, Inc.*. [online] Available at: <<https://www.qualys.com/apps/endpoint-detection-response/>> [Accessed 2 November 2020].

Rapid7. 2020. *Vulnerability Assessment Tool: Insightvm Features*. [online] Available at: <<https://www.rapid7.com/products/insightvm/features/>> [Accessed 2 November 2020].

Security Intelligence. 2020. *A Step-By-Step Guide To Vulnerability Assessment*. [online] Available at: <<https://securityintelligence.com/a-step-by-step-guide-to-vulnerability-assessment/>> [Accessed 21 October 2020].

Securityscorecard.com. 2020. *6 Strategies For Creating Your Cyber Vulnerability Assessment*. [online] Available at: <<https://securityscorecard.com/blog/strategies-for-creating-cyber-vulnerability-assessment>> [Accessed 21 October 2020].

Skillset.com. 2020. *What Are The Three General Layers That Make Up A Web Application Architecture?* - Skillset. [online] Available at: <<https://www.skillset.com/questions/what-are-the-three-general-layers-that-make-up-a-web-application-architecture#:~:text=A%20general%20architectural%20design%20of,layer%2C%20and%20a%20data%20layer.>> [Accessed 3 November 2020].

Tenable®. 2020. *Tenable.Sc.* [online] Available at: <<https://www.tenable.com/products/tenable-sc>> [Accessed 2 November 2020].

Tripwire.com. 2020. *Cybersecurity Compliance And Industry Needs / Tripwire.* [online] Available at: <<https://www.tripwire.com/solutions>> [Accessed 2 November 2020].

Viegas, G., 2020. *How To Build A Top-Notch Vulnerability Management Program.* [online] CSO Online. Available at: <<https://www.csoonline.com/article/3027570/taking-the-vulnerability-management-program-from-good-to-great.html>> [Accessed 5 November 2020].

VMware. 2020. *Application Security.* [online] Available at: <<https://www.vmware.com/topics/glossary/content/application-security>> [Accessed 29 October 2020].

Vulnerability Management Lifecycle. www.coalfire.com/the-coalfire-blog/june-2018/cyber-engineering-primer-vulnerability-mgmt. [Accessed 11 Nov. 2020.]