A Project Report on

Unveiling Insights: Effective Threat Mitigation with QRadar

By

Y Ashok – 21358008021

Long term intership

SRI SANKANRAS DEGREE COLLEGE, KURNOOL

ABSTRACT

In the timeless saga of cybersecurity, organizations embark on an enduring journey through a labyrinth of evolving threats, requiring steadfast defense mechanisms. IBM QRadar emerges as an enduring bastion, offering a multifaceted Security Information and Event Management (SIEM) platform. This abstract embarks on an epic odyssey to unveil the timeless efficacy of QRadar in fortifying threat mitigation strategies. QRadar's resilience transcends epochs, as it adeptly aggregates an eclectic ensemble of security data sources, harnesses advanced analytics for robust threat detection, orchestrates real-time alerting and nimble incident response, seamlessly integrates with external threat intelligence feeds, and provides perennial vigilance alongside comprehensive compliance reporting. By embracing the timeless legacy of QRadar's capabilities, organizations embark on a heroic quest, forging an indomitable shield that vigilantly identifies and vanquishes threats while steadfastly upholding regulatory standards in the ever-evolving cosmos of cybersecurity.

INTRODUCTION

In the ever-evolving landscape of cybersecurity, organizations find themselves navigating a perpetually shifting terrain of threats and challenges. From the dawn of digitalization to the present day, the quest for effective defense mechanisms against cyber threats has been a constant endeavor. In this enduring journey, IBM QRadar emerges as a stalwart companion, offering a multifaceted Security Information and Event Management (SIEM) platform that stands the test of time.

As organizations strive to safeguard their digital assets and sensitive information, the need for robust threat mitigation strategies becomes increasingly paramount. In this context, QRadar serves as a beacon of resilience, providing organizations with the tools and capabilities to proactively detect, respond to, and mitigate cyber threats in real-time.

This paper embarks on a comprehensive exploration of the efficacy of QRadar in bolstering threat mitigation efforts. By delving into QRadar's timeless features and functionalities, we aim to elucidate its enduring value as a cornerstone of modern cybersecurity practices. From its ability to aggregate diverse security data sources to its advanced analytics for threat detection, QRadar represents a formidable ally in the ongoing battle against cyber adversaries.

Through this examination, we seek to shed light on how organizations can harness the power of QRadar to fortify their security posture, navigate the complexities of the cyber landscape, and emerge victorious in the face of ever-present threats. Join us on this journey as we unveil the insights and capabilities that make QRadar a timeless guardian of digital resilience.

IBM QRadar Vulnerability Manager

Improve security and compliance by identifying security gaps and risks for resolution

For many organizations, managing network vulnerabilities and risks is a lesson in frustration. Vulnerability scans are typically conducted in response to compliance mandates, and they can reveal up to tens of thousands of exposures—depending upon network size. Scan results are often a complex puzzle of misconfigured devices, unpatched software, and outdated or obsolete systems. And security administrators must struggle to quickly identify and remediate or mitigate the exposures that pose the greatest risk. At the same time, security breaches are dramatically increasing for all kinds of organizations. From e-commerce and social-networking giants to healthcare, universities, banks, governments and gaming sites, the breadth of breach targets is vast. While the number of disclosed vulnerabilities continues to rise, the number of incidents that result in the loss, theft of exposure of personally identifiable information has been increasing at an alarming rate.

IBM QRadar Vulnerability Manager can help organizations minimize the chances of a network security breach by using a proactive approach to finding security weaknesses and minimizing potential risks. It uses a proven vulnerability scanner to collect up-to-date results, but unlike other solutions, it leverages the capabilities of IBM QRadar Security Intelligence Platform to present the data within the overall context of the network usage, security and threat posture. Designed to consolidate results from multiple vulnerability scanners, risk management solutions and external threat intelligence resources, QRadar Vulnerability Manager operates like a centralized control center to identify key security weaknesses that need to be addressed to help thwart future attacks. It also correlates network topology information using data from IBM QRadar SIEM including asset configurations, network events and flow patterns. This provides valuable insights revealing, for example, which assets and vulnerabilities are causing the most risk, so IT staff can prioritize their remediation tasks.

QRadar Vulnerability Manager can also help identify firewall and intrusion prevention (IPS) misconfigurations that may allow attackers into the network and create inefficiencies in devices. Many network attacks succeed simply due to inconsistent network and security configuration practices, highlighting the need for automated network configuration

monitoring and alerts for policy breaches. QRadar Vulnerability Manager offers an integrated, automated, policy-based approach that can greatly improve an organizations' ability to assess information security risk through a single console shared with QRadar SIEM. It leverages a broad range of risk indicators including asset, network and security configuration data, network activity data, network and security events, and vulnerability scan results. It also provides other key capabilities that include the assignment of risk scores, vulnerability risk assessment, and correlation of known vulnerabilities with network topologies. It helps deliver a prioritized list of vulnerabilities to better assess which systems are most vulnerable to attack and should be remediated first.

QRadar Vulnerability Manager also delivers advanced threat modeling , and the simulation and visualization of the potential spread of threats through the network by leveraging vulnerability, network topology and connection data. QRadar Vulnerability Manager helps security teams identify resource configuration issues, understand the impact of software patching schedules, coordinate with intrusion prevention systems to block open connections, and establish continuous monitoring of systems that can't otherwise be remediated—all from a single, integrated dashboard.

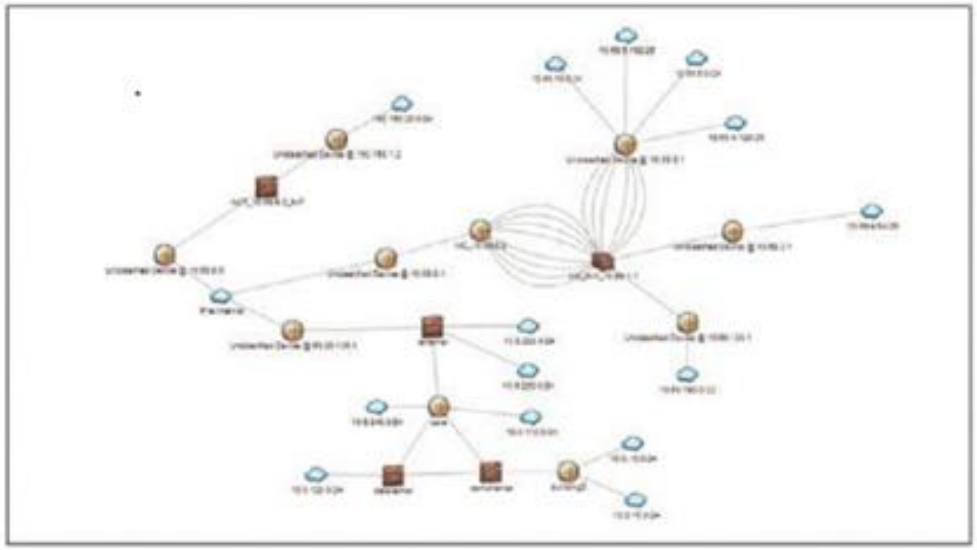
By correlating vulnerability data with QRadar SIEM event and threat analysis, device configuration and network traffic analysis, and external databases, including IBM XForce® threat intelligence, QRadar Vulnerability Manager can help organizations build actionable plans for deploying their often constrained IT staffing resources. And since it is already integrated with QRadar Security Intelligence Platform, security teams have one less system to install, configure and manage. Get a single, prioritized view of potential vulnerabilities

• Select a dashboard view and click through related tabs to review security offenses, log events, network flows, asset statuses and configurations, reports, risks and vulnerabilities

• Create, edit and save asset searches and scans for more intelligent monitoring

• Make faster, more informed decisions with a prioritized, consolidated view of scan data

• Help coordinate patching and virtual patching activities, and direct intrusion prevention systems (IPSs) to block potential attack paths for maximum impact



The QRadar Vulnerability Manager topology viewer enables users to view network devices and relationships, including subnets and links

QRadar Vulnerability Manager includes an embedded scanning engine that can be set up to run both dynamic and periodic scans, providing near real-time visibility of weaknesses that could otherwise remain hidden. Leveraging the passive asset discovery capabilities of IBM QRadar QFlow and Log Collector appliances, any new asset appearing on the network can be immediately scanned. As a result, organizations can reduce their exposure to advanced threats between regular scanning cycles and help ensure compliance with the latest security regulations.

Using the same rules-based approach as QRadar SIEM, QRadar Vulnerability Manager helps minimize false positives and filters out vulnerabilities already classified as nonthreatening. For example, applications may be installed on a server, but they may be inactive, and therefore not a security risk; devices that appear exposed may actually be protected by a firewall; or endpoints that have vulnerabilities may already be scheduled for patching. QRadar Vulnerability Manager maintains a current network view of all discovered vulnerabilities, including details such as when the vulnerabilities were found, when they were last seen, what scan jobs reported the vulnerabilities, and to whom the vulnerability is assigned for remediation or mitigation. The software also presents historic views of daily, weekly and monthly trends, and it can produce long-term trending reports, such as the month-by-month trend of Payment Card Industry (PCI) failure vulnerabilities discovered over the past year.

Stand-alone, independent vulnerability-scanning solutions can take considerable time to scan large address spaces for assets, servers and services, and their scan results can be out of date quickly. These point solutions also require additional infrastructure and include different technologies for network, application and database scanning—all requiring additional administration. And after identifying an often incomplete sea of vulnerabilities, the point solutions do not include any contextual information for helping security teams prioritize their tasks for remediation.

Thwart Advanced Threats

Unlike the random, brute-force attacks of the past, today's organizations must guard against "advanced persistent threats"—that is, a complex series of attacks that often take place over a prolonged timeframe. Using a range of tactics from zeroday exploits to custom malware to simply trolling for unpatched systems, these attackers consistently probe their targets using a "low-and-slow" approach until they find a security gap. Organizations can use more intelligent tools like QRadar Vulnerability Manager to improve their defenses by regularly scanning and addressing as many high-impact vulnerabilities as possible. Most vulnerability scanners simply identify large numbers of exposures and leave it up to security teams to understand the severity of risks. These tools are often not integrated with the existing security infrastructure and require additional manual effort to align with the current network topology, usage information and security processes. Many of these tools are used simply for compliance, rather than as an integral part of a threat and security management program.

Address Compliance Mandates

Regulatory requirements are forcing organizations of all sizes to develop vulnerability management programs to help ensure proper control of sensitive IT assets. QRadar Vulnerability Manager helps organizations facilitate compliance by conducting regular network scans and maintaining detailed audit trails. It categorizes each vulnerability with a severity rating and an exposure score. In addition to scanning assets both internally and externally, QRadar Vulnerability Manager enables security teams to create tickets to manage remediation activities and specify exceptions with a full audit trail.

Extend Your Security Intelligence

QRadar Vulnerability Manager combines the real-time security visibility of QRadar Security Intelligence Platform with the results of proven vulnerability-scanning technology. As part of the QRadar SIEM architecture, QRadar Vulnerability Manager can be deployed quickly and security teams do not need to learn a new interface. They can simply generate reports from within the familiar QRadar family user interface.

Apply Proactive Security

• High-speed internal scanning, which helps preserve network performance and availability
• Support for discovery, non-authenticated, uthenticated and Open Vulnerability Assessment Language (OVAL) scans
• External scanning capabilities to see the network from an attacker's viewpoint and help facilitate compliance
• Single-click investigations from dashboard screens and deep, rules-based, rapid searching capabilities to learn more about specific events or identify long-term trends
• Suppression of acceptable, false positive or otherwise non-mitigated vulnerabilities from ongoing reporting
• Vulnerability assignment and remediation lifecycle management
• Full audit trail for compliance reporting.

Why IBM?

IBM Security offers one of the most advanced and integrated portfolios of enterprise security products and services. The portfolio, supported by world-renowned X-Force research and development, provides security intelligence to help organizations holistically protect their people, infrastructures, data and applications, offering solutions for identity and access management, database security, application development, risk Page 3 of 5 management, endpoint management, network security and more. These solutions enable organizations to effectively manage risk and implement integrated security for mobile, cloud, social media and other enterprise business architectures. IBM operates one of the world's broadest security

research, development and delivery organizations, monitors 13 billion security events per day in more than 130 countries, and holds more than 3,000 security patents.

OWASP

The Open Web Application Security Project, or OWASP, is an international non-profit organization dedicated to web application security. One of OWASP's core principles is that all of their materials be freely available and easily accessible on their website, making it possible for anyone to improve their own web application security. The materials they offer include documentation, tools, videos, and forums. Perhaps their best-known project is the OWASP Top 10.

What is the OWASP Top 10?

The OWASP Top 10 is a regularly-updated report outlining security concerns for web application security, focusing on the 10 most critical risks. The report is put together by a team of security experts from all over the world. OWASP refers to the Top 10 as an 'awareness document' and they recommend that all companies incorporate the report into their processes in order to minimize and/or mitigate security risks.

1. Injection

Injection attacks happen when untrusted data is sent to a code interpreter through a form input or some other data submission to a web application. For example, an attacker could enter SQL database code into a form that expects a plaintext username. If that form input is not properly secured, this would result in that SQL code being executed. This is known as an SQL injection attack.

Injection attacks can be prevented by validating and/or sanitizing user-submitted data. (Validation means rejecting suspicious-looking data, while sanitization refers to

cleaning up the suspicious-looking parts of the data.) In addition, a database admin can set controls to minimize the amount of information an injection attack can expose.

2. Broken Authentication

Vulnerabilities in authentication (login) systems can give attackers access to user accounts and even the ability to compromise an entire system using an admin account. For example, an attacker can take a list containing thousands of known username/password combinations obtained during a data breach and use a script to try all those combinations on a login system to see if there are any that work.

Some strategies to mitigate authentication vulnerabilities are requiring two-factor authentication (2FA) as well as limiting or delaying repeated login attempts using rate limiting.

3. Sensitive Data Exposure

If web applications don't protect sensitive data such as financial information and passwords, attackers can gain access to that data and sellor utilize it for nefarious purposes. One popular method for stealing sensitive information is using an on-path attack.

Data exposure risk can be minimized by encrypting all sensitive data as well as disabling the caching* of any sensitive information. Additionally, web application developers should take care to ensure that they are not unnecessarily storing any sensitive data.

*Caching is the practice of temporarily storing data for re-use. For example, web browsers will often cache webpages so that if a user revisits thosepages within a fixed time span, the browser does not have to fetch the pages from the web.

4. XML External Entities (XEE)

This is an attack against a web application that parses XML* input. This input can reference an external entity, attempting to exploit a vulnerability in the parser. An

‘external entity’ in this context refers to a storage unit, such as a hard drive. An XML parser can be duped into sending data to an unauthorized external entity, which can pass sensitive data directly to an attacker.

The best ways to prevent XEE attacks are to have web applications accept a less complex type of data, such as JSON**, or at the very least to patch XML parsers and disable the use of external entities in an XML application.

*XML or Extensible Markup Language is a markup language intended to be both human-readable and machine-readable. Due to its complexity and security vulnerabilities, it is now being phased out of use in many web applications.

**JavaScript Object Notation (JSON) is a type of simple, human-readable notation often used to transmit data over the internet. Although it was originally created for JavaScript, JSON is language-agnostic and can be interpreted by many different programming languages.

5. Broken Access Control

Access control refers a system that controls access to information or functionality. Broken access controls allow attackers to bypass authorization and perform tasks as though they were privileged users such as administrators. For example a web application could allow a user to change which account they are logged in as simply by changing part of a url, without any other verification.

Access controls can be secured by ensuring that a web application uses authorization tokens* and sets tight controls on them.

*Many services issue authorization tokens when users log in. Every privileged request that a user makes will require that the authorization token be present. This is a secure way to ensure that the user is who they say they are, without having to constantly enter their login credentials.

6. Security Misconfiguration

Security misconfiguration is the most common vulnerability on the list, and is often the result of using default configurations or displaying excessively verbose errors. For

instance, an application could show a user overly-descriptive errors which may reveal vulnerabilities in the application. This can be mitigated by removing any unused features in the code and ensuring that error messages are more general.

7. Cross-Site Scripting

Cross-site scripting vulnerabilities occur when web applications allow users to add custom code into a url path or onto a website that will be seen by other users. This vulnerability can be exploited to run malicious JavaScript code on a victim's browser. For example, an attacker could send an email to a victim that appears to be from a trusted bank, with a link to that bank's website. This link could have some malicious JavaScript code tagged onto the end of the url. If the bank's site is not properly protected against cross-site scripting, then that malicious code will be run in the victim's web browser when they click on the link.

Mitigation strategies for cross-site scripting include escaping untrusted HTTP requests as well as validating and/or sanitizing user-generated content. Using modern web development frameworks like ReactJS and Ruby on Rails also provides some built-in cross-site scripting protection.

8. Insecure Deserialization

This threat targets the many web applications which frequently serialize and deserialize data. Serialization means taking objects from the application code and converting them into a format that can be used for another purpose, such as storing the data to disk or streaming it. Deserialization is just the opposite: converting serialized data back into objects the application can use. Serialization is sort of like packing furniture away into boxes before a move, and deserialization is like unpacking the boxes and assembling the furniture after the move. An insecure deserialization attack is like having the movers tamper with the contents of the boxes before they are unpacked.

An insecure deserialization exploit is the result of deserializing data from untrusted sources, and can result in serious consequences like DDoS attacks and remote code execution attacks. While steps can be taken to try and catch attackers, such as monitoring deserialization and implementing type checks, the only sure way to protect

against insecure deserialization attacks is to prohibit the deserialization of data from untrusted sources.

9. Using Components With Known Vulnerabilities

Many modern web developers use components such as libraries and frameworks in their web applications. These components are pieces of software that help developers avoid redundant work and provide needed functionality; common example include front-end frameworks like React and smaller libraries that used to add share icons or a/b testing. Some attackers look for vulnerabilities in these components which they can then use to orchestrate attacks. Some of the more popular components are used on hundreds of thousands of websites; an attacker finding a security hole in one of these components could leave hundreds of thousands of sites vulnerable to exploit.

Component developers often offer security patches and updates to plug up known vulnerabilities, but web application developers don't always have the patched or most-recent versions of components running on their applications. To minimize the risk of running components with known vulnerabilities, developers should remove unused components from their projects, as well as ensuring that they are receiving components from a trusted source and ensuring they are up to date.

10. Insufficient Logging And Monitoring

Many web applications are not taking enough steps to detect data breaches. The average discovery time for a breach is around 200 days after it has happened. This gives attackers a lot of time to cause damage before there is any response. OWASP recommends that web developers should implement logging and monitoring as well as incident response plans to ensure that they are made aware of attacks on their applications.

THE CWE/ SANS TOP 25 SECURITY VULNERABILITIES

The **CWE/ SANS top 25 vulnerabilities** are created through multiple surveys and individual interviews with developers, senior security analysts and researchers. It is a condensed list of the most common and severe software errors that can lead to serious software vulnerabilities that are typically simple to identify and exploit.

What Is CWE/ SANS Top 25?

The CWE/ SANS top 25 most dangerous software flaws is a list of the **most dangerous flaws** because they let attackers gain entire control of the software, steal data and information from it, or prohibit it from functioning at all.

The SANS top 25 is a versatile starting point that can be used by almost any organization, regardless of size, industry, geography or **government/ commercial** status.

The controls are prioritized to protect the organization's infrastructure and data by strengthening the organization's defense system through continuous automated protection and monitoring. They were developed and maintained by an international group of **organizations, government agencies, and security experts**.

How Does SANS Top 25 Work And Why Is It Important?

The SANS top 25 is a list created to give one the most bang for the buck when it comes to enhancing the risk posture against real-world risks. The Common Vulnerabilities and Exposures Team generated the list using publicly available data, CWE mappings from the **National Vulnerability Database (NVD), and CVSS** scores for each CWE.

A scoring algorithm was then used to determine the severity of each fault. This data-driven method can be used to generate a CWE Top 25 list of security vulnerabilities on a regular basis.

List Of SANS Top 25

1. Out-of-bounds Write

2. Improper Neutralization of Input During Web Page Generation (‘Cross-site Scripting’)
3. Out-of-bounds Read
4. Improper Input Validation
5. Improper Neutralization of Special Elements used in an OS Command (‘OS Command Injection’)
6. Improper Neutralization of Special Elements used in an SQL Command (‘SQL Injection’)
7. Use After Free
8. Improper Limitation of a Path name to a Restricted Directory (‘Path Traversal’)
9. Cross-Site Request Forgery (CSRF)
10. Unrestricted Upload of File with Dangerous Type
11. Missing Authentication for Critical Function
12. Integer Overflow or Wraparound
13. Deserialization of Untrusted Data
14. Improper Authentication
15. NULL Pointer Dereference
16. Use of Hard-coded Credentials
17. Improper Restriction of Operations within the Bounds of a Memory Buffer
18. Missing Authorization
19. Incorrect Default Permissions
20. Exposure of Sensitive Information to an Unauthorized Actor
21. Insufficiently Protected Credentials
22. Incorrect Permission Assignment for Critical Resource
23. Improper Restriction of XML External Entity Reference
24. Server-Side Request Forgery (SSRF)
25. Improper Neutralization of Special Elements used in a Command (‘Command Injection’)

Over the years, the list of the **SANS top 25** has undergone considerable changes. One noticeable change is the transition of the list from a more abstract description and version of the weaknesses to a more specific format. The **remapping** and **change** in the rankings of the weaknesses determine their intensity, functionally and severity.

As the community improves its mappings to more exact weaknesses, this movement is projected to continue in the next few years. More particular CWEs have risen to fill the place with these high-level classes as class-level weaknesses have declined.

The **Top 25 Team** feels that Base-level flaws are more instructive to stakeholders than **Class-level weaknesses.** Therefore further movement will substantially help users who are striving to comprehend the true concerns that affect today's systems **(CWE).**

Codegrip Follows SANS Top 25

Codegrip is an automated code review tool that automates the code review process. It helps in building an error-free and **smell-free code** by making the process of **reviewing code** frictionless and smooth.

Codegrip ensures that the codebase does not include any **vulnerabilities and bugs,** and for the same it uses the SANS top 25. The availability of the list of common vulnerabilities helps in avoiding those.

A general understanding of the SANS top 25 makes it convenient for the **automated tool** to identify and highlight the problems and provide a detailed report on the same. The SANS top 25 list makes the task of automated code review more versatile, easy and fast.

How SANS 25 Ensures Code Security?

The SANS top 25 list is constantly evolving and expanding. With regular updates and changes, it becomes critical to continuously spread awareness about the most frequent programming security flaws and this is exactly where the list comes in handy.

All persons involved in software development read and understand this, the security of the programs will **undoubtedly increase significantly.**

These programming samples available in the list helps in ensuring the **security and stability** of the codebase. Despite constant evolution and changes in the making and functioning of the list, the main goal of the list remains the same, that is to spread awareness about the common vulnerabilities that can make the code base lose and unhealthy.

Conclusion(SANS)

The health of the code marks for its security and the SANS top 25 list is capable of providing all the common ailments and diseases that the code may commonly suffer from.

The list gives critical recommendations for **software developers** in order to eliminate software **security flaws** in their products.

Accessing the list while **automating the code review process** can ensure a stricter and in-depth examination of the code which is what **Codegrip with the help of SANS top 25** does to ensure the **code security** of the software.