SRI SHANKARA'S DEGREE COLLEGE,KURNOOL
RAYALASEEMA UNIVERSITY

A PROJECT REPORT ON

UNVEILING INSIGHTS: EFFECTIVE THREAT MITIGATION WITH QRADAR

By

Shaik Reena Tasleem

BSc(Mpcs)

21360008019

Long term internship

UNVEILING INSIGHTS: EFFECTIVE THREAT MITIGATION WITH QRADAR

Team id : LTIP2024TMID14085

Team size : 5

Team leader : Boya Anitha

Team member : Shaik Reena Tasleem

Team member : Yerrala Ashok

Team member : Yerravati Ramudu

Team member : Yerramu Prathibha

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.

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 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.

* 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

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. 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

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.

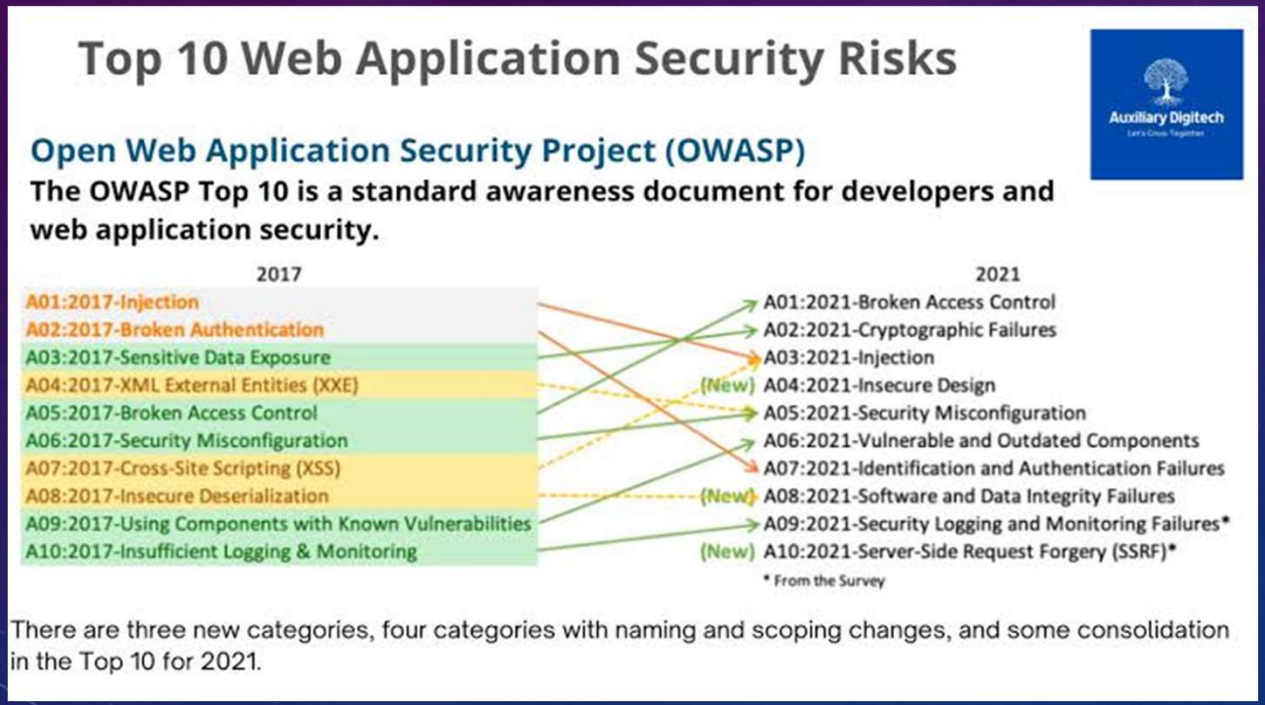
IBM operates one of the world's broadest securityresearch, development and delivery organizations, monitors 13 billion security events per day in more than 130 countries, and holds more than 3,000 security patents.

OWASP

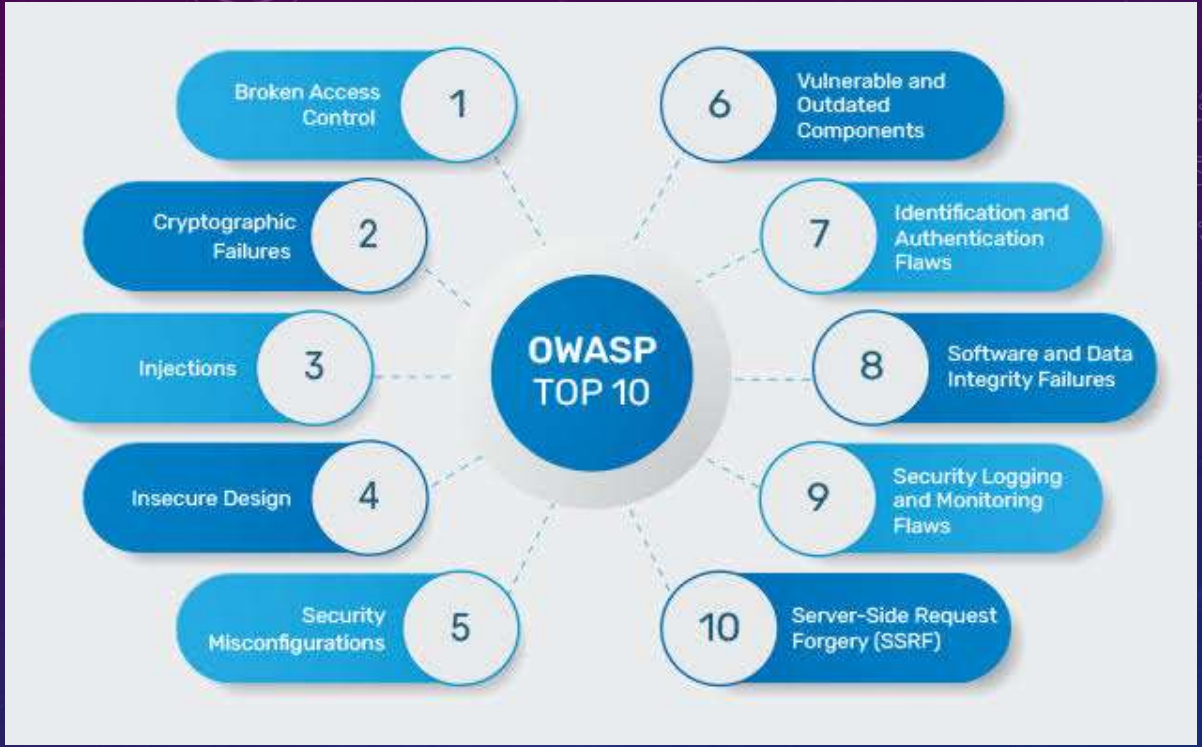
The Open Web Application Security Project, or OWASP, is an international non-profit organization dedicated to web application security. One of OWASP's core principles is that all of their materials be freely available and easily accessible on their website, making it possible for anyone to improve their own web application security.

The materials they offer include documentation, tools, videos, and forums. Perhaps their best-known project is the OWASP Top10.

WHAT IS THE OWASP TOP 10 ?

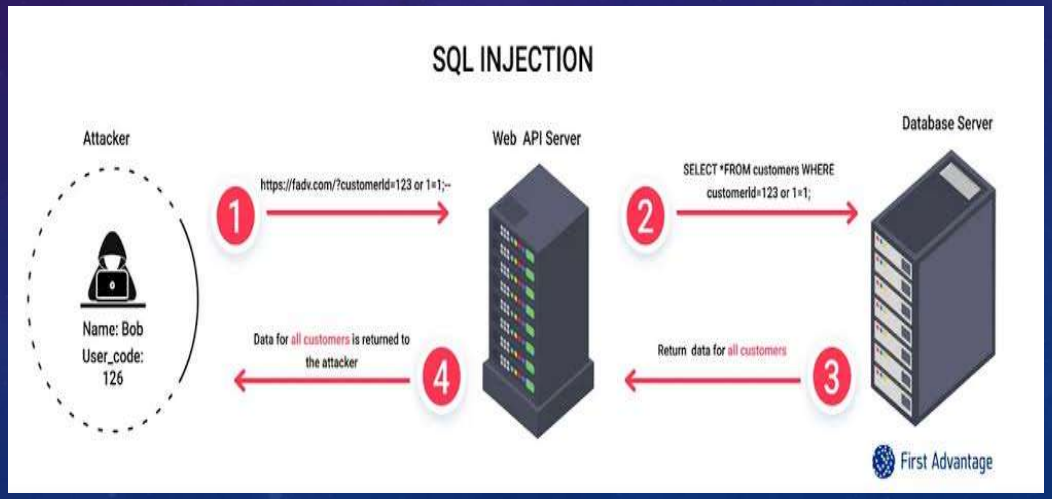


The OWASP Top 10 is a regularly-updated report outlining security concerns for web application security, focusing on the 10 most critical risks.

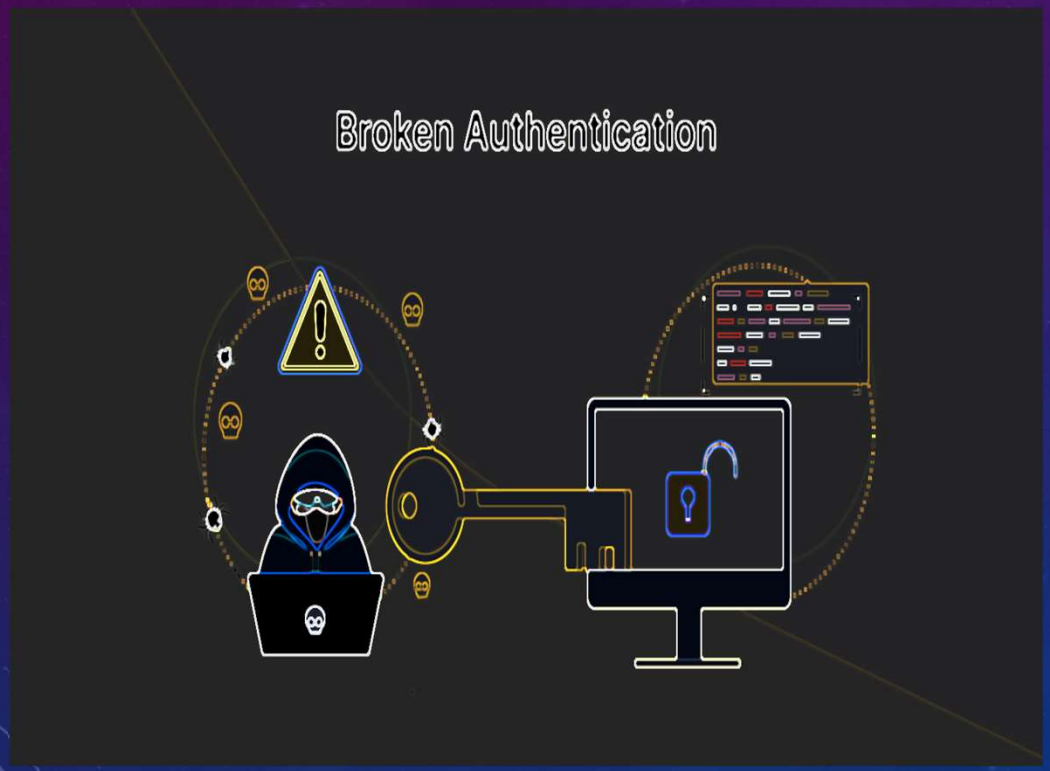


The report is put together by a team of security experts from all over the world. OWASP refers to the Top 10 as an 'awareness document' and they recommend that all companies incorporate the report into their processes to minimize and/or mitigate security risks.

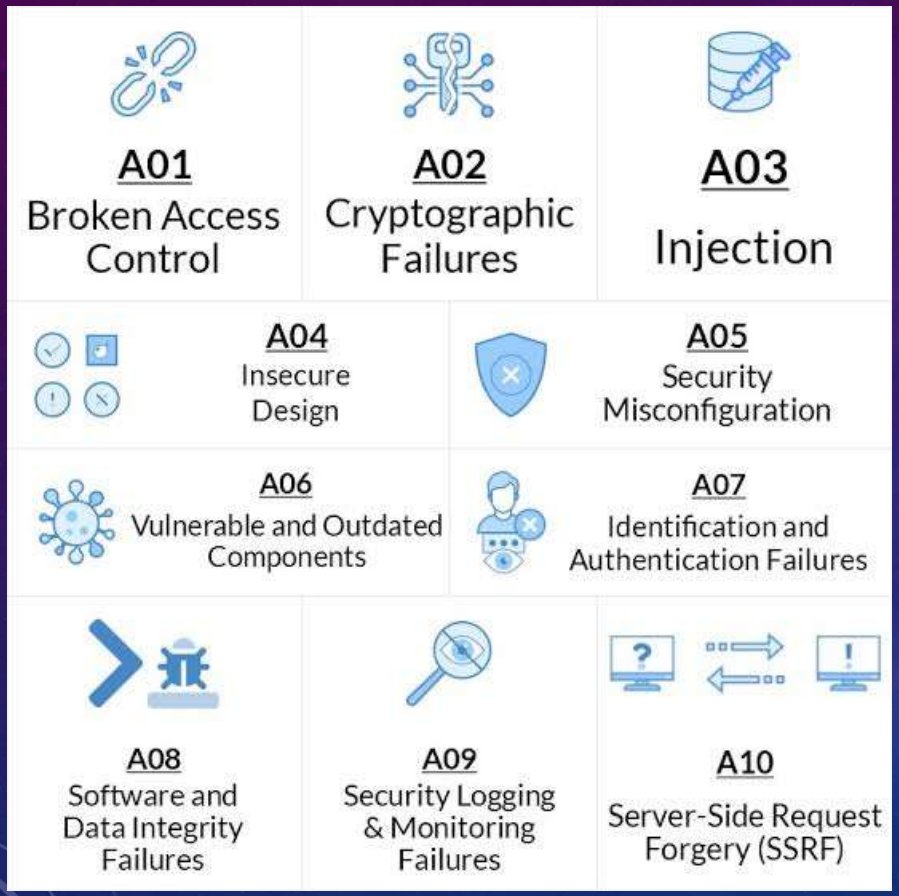
Injection Injection attacks happen when untrusted data is sent to a code interpreter through a form input or some other data submission to a web application. For example, an attacker could enter SQL database code into a form that expects a plaintext username. If that form input is not properly secured,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



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

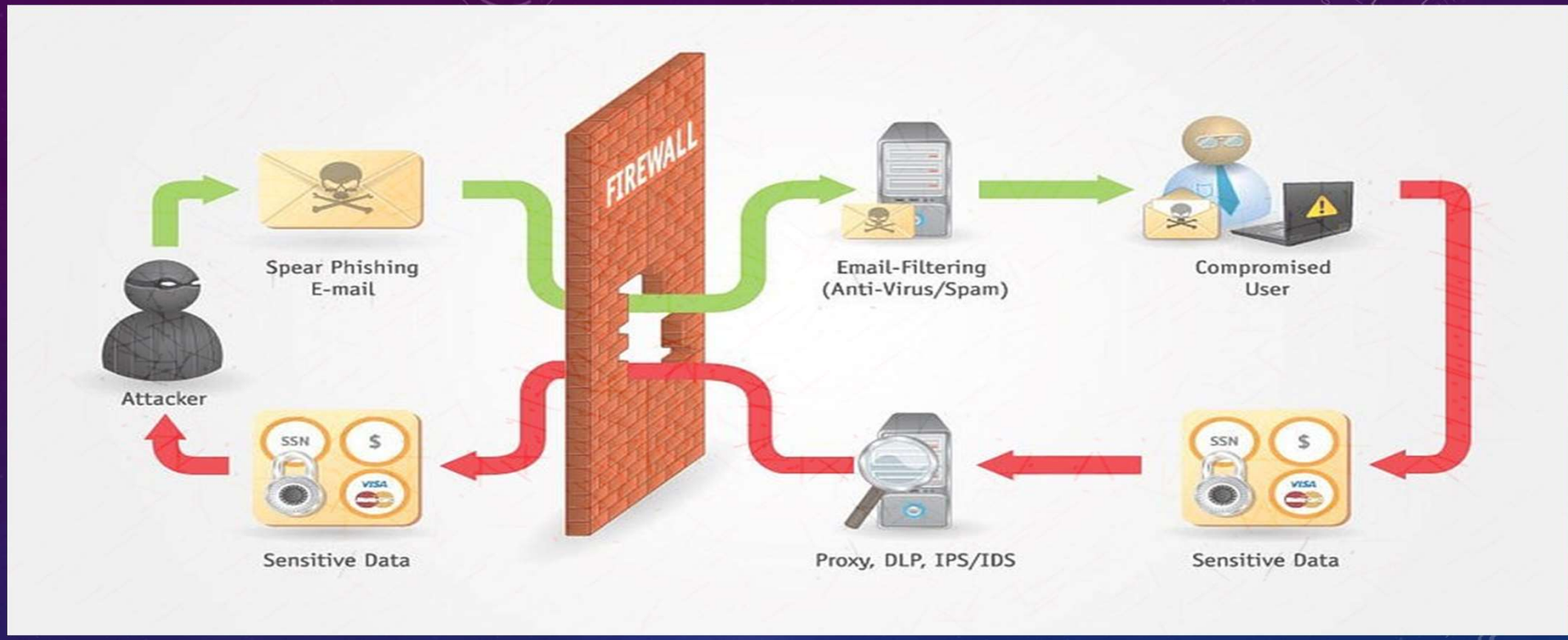


combinations obtained during a data breach and use a script to try all those combinations on a login system to see if there are any that work.

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

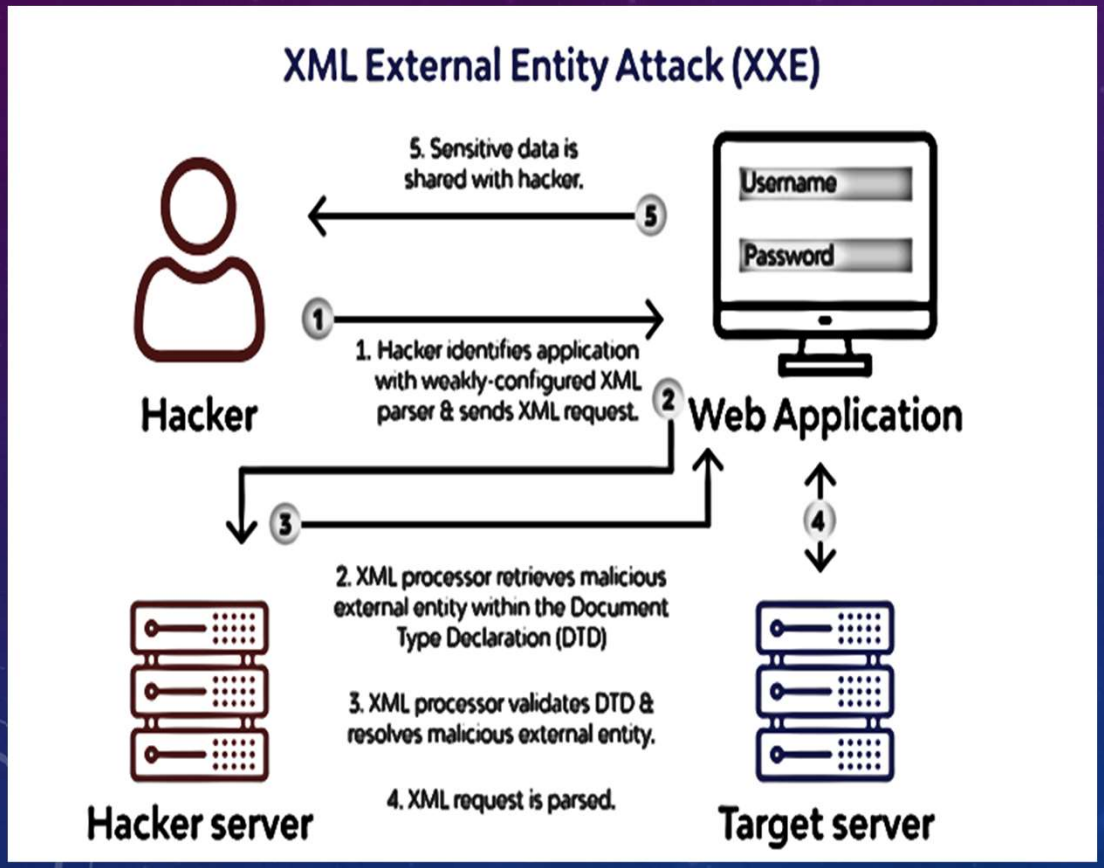
Data Exposure

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.



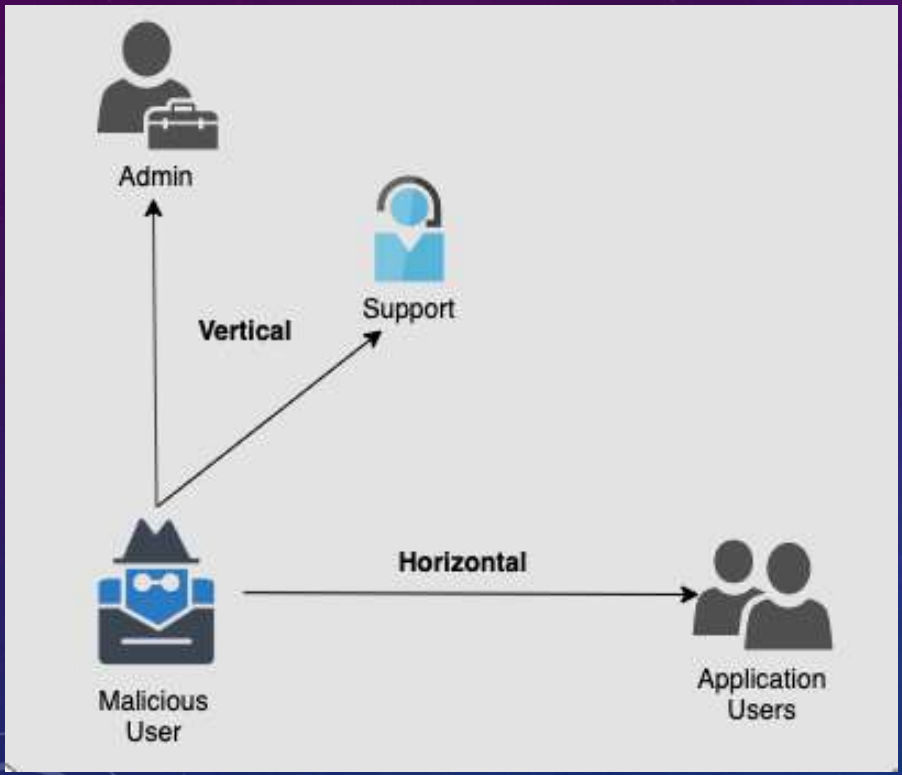
Caching is the practice of temporarily storing data for re-use. For example, web browsers will often cache webpages so that if a user revisits thosepages within a fixed time span, the browser does not have to fetch the pages from the web.

XML External Entities (XEE)



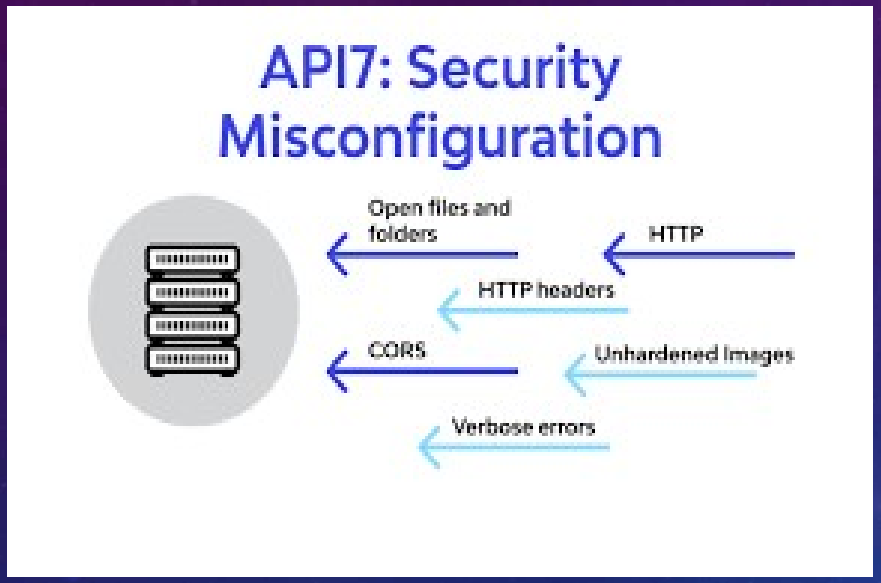
This is an attack against a web application that parses XML* input. This input can reference an external entity, attempting to exploit a vulnerability in the parser. An 'external entity' in this context refers to a storage unit, such as a hard drive. An XML parser can be duped into sending data to an unauthorized external entity, which can pass sensitive data directly to an attacker.

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.

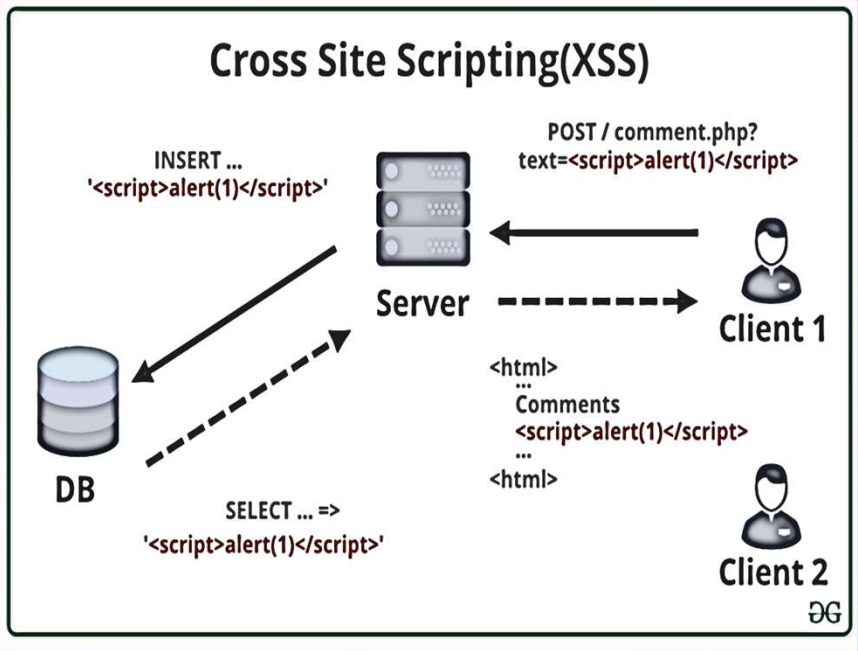
SECURITY MISCONFIGURATION



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.

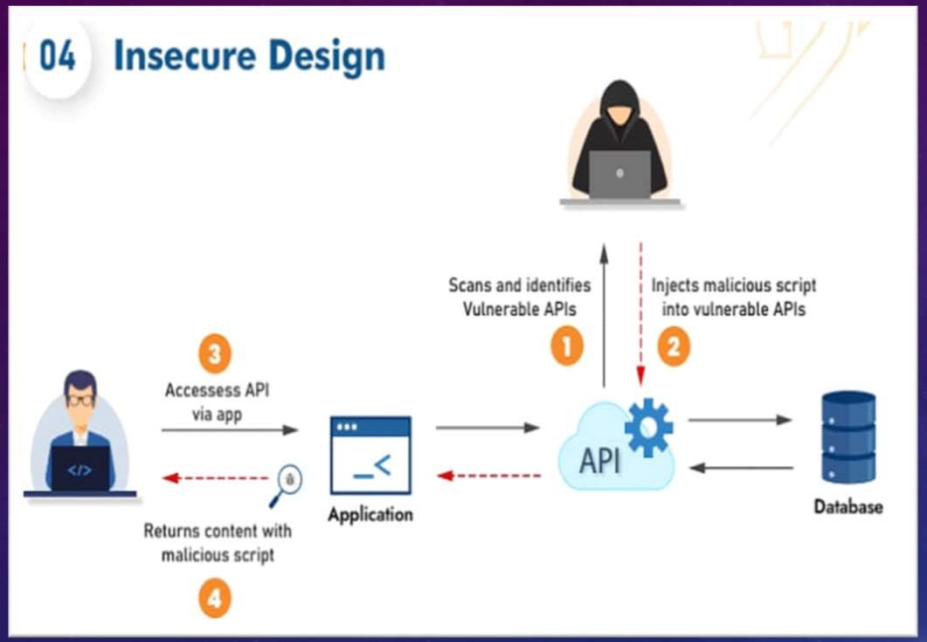
CROSS-SITE SCRIPTING



- 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.

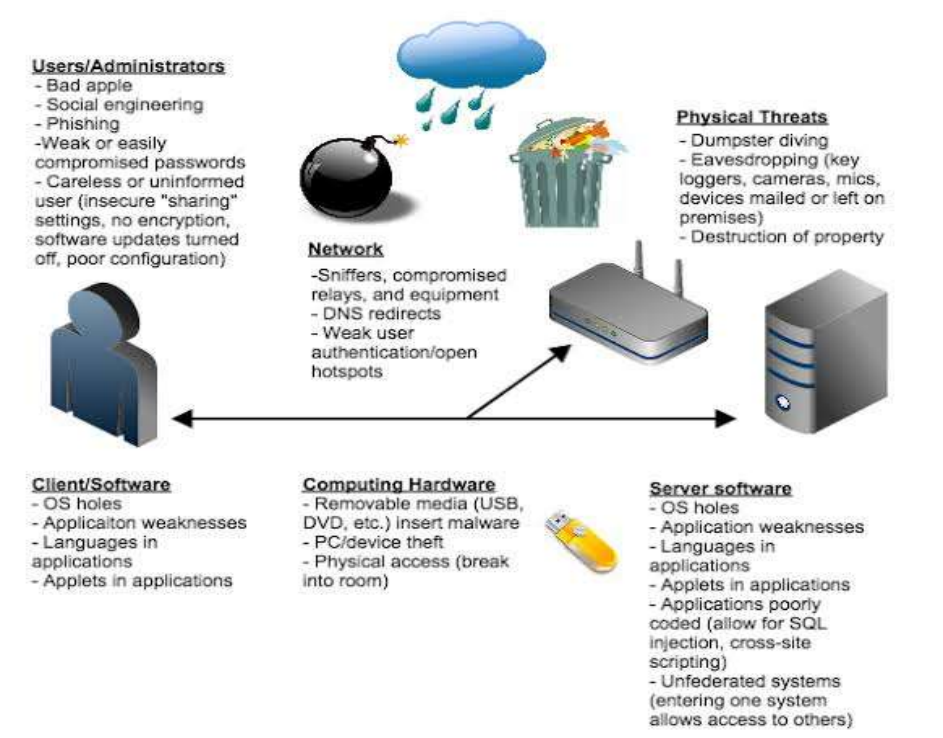
INSECURE DESERIALIZATION



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.

USING COMPONENTS WITH KNOWN VULNERABILITIES

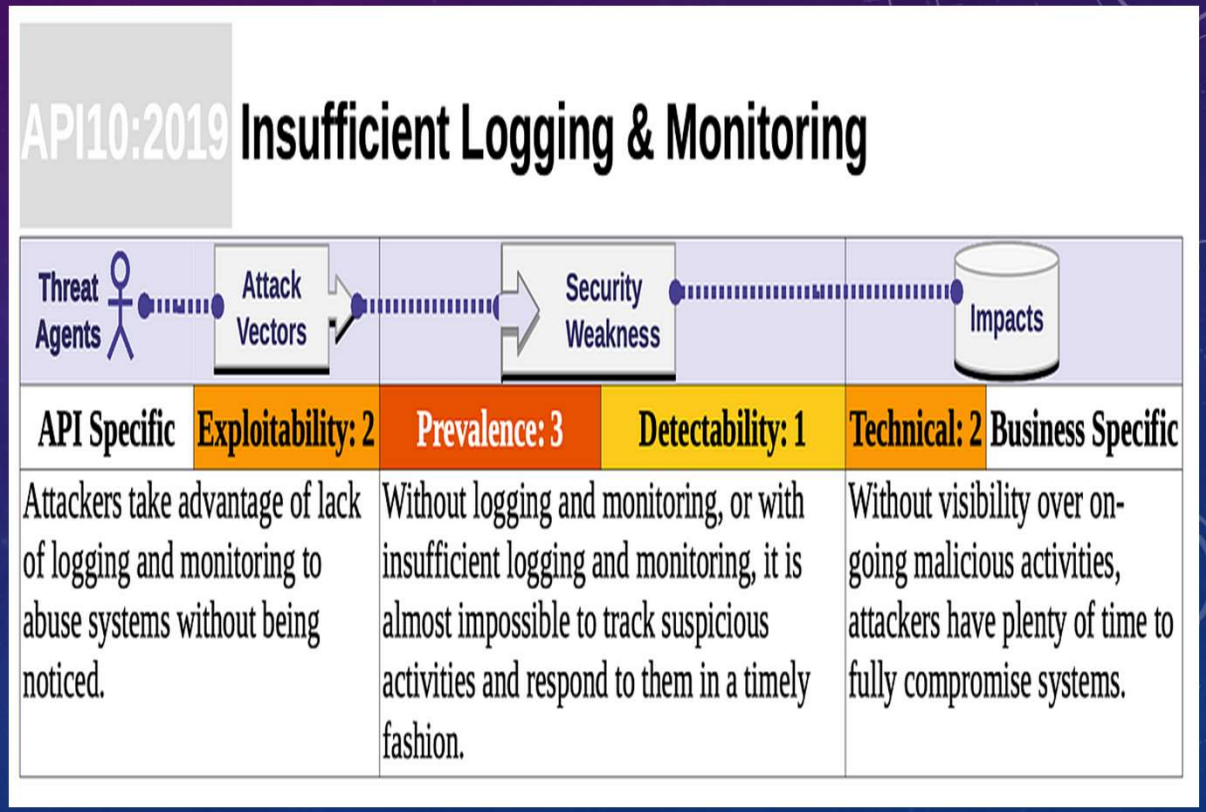


- Many modern web developers use components such as libraries and frameworks in their web applications. These components are pieces of software that help developers avoid redundant work and provide needed functionality; common examples include front-end frameworks like React and smaller libraries that are 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.

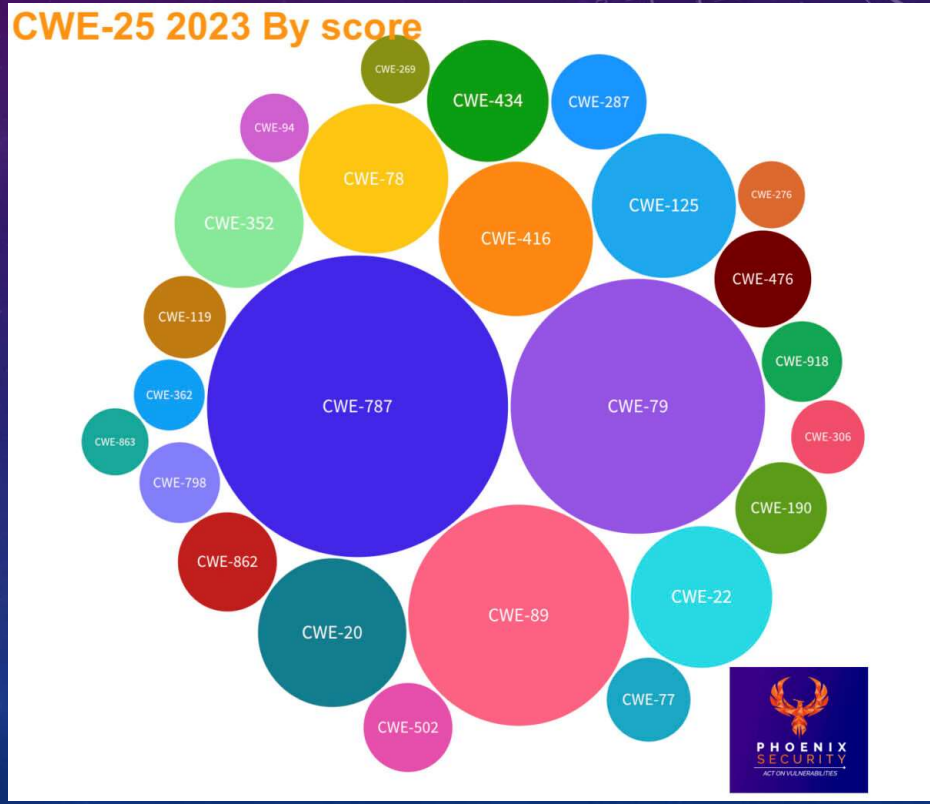
INSUFFICIENT LOGGING AND MONITORING

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**.

LIST OF SANS TOP 25

Rank	ID	Name
[1]	CWE-89	Improper Neutralization of Special Elements used in an SQL Command ('SQL Injection')
[2]	CWE-78	Improper Neutralization of Special Elements used in an OS Command ('OS Command Injection')
[3]	CWE-120	Buffer Copy without Checking Size of Input ('Classic Buffer Overflow')
[4]	CWE-79	Improper Neutralization of Input During Web Page Generation ('Cross-site Scripting')
[5]	CWE-306	Missing Authentication for Critical Function
[6]	CWE-862	Missing Authorization
[7]	CWE-798	Use of Hard-coded Credentials
[8]	CWE-311	Missing Encryption of Sensitive Data
[9]	CWE-434	Unrestricted Upload of File with Dangerous Type
[10]	CWE-807	Reliance on Untrusted Inputs in a Security Decision
[11]	CWE-250	Execution with Unnecessary Privileges
[12]	CWE-352	Cross-Site Request Forgery ('CSRF')

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.

Rank	ID	Name
[13]	CWE-98	Improper Control of Filename for Include/Require Statement in PHP Program ('PHP File Inclusion')
[14]	CWE-129	Improper Validation of Array Index
[15]	CWE-754	Improper Check for Unusual or Exceptional Conditions
[16]	CWE-209	Information Exposure Through an Error Message
[17]	CWE-190	Integer Overflow or Wraparound
[18]	CWE-131	Incorrect Calculation of Buffer Size
[19]	CWE-306	Missing Authentication for Critical Function
[20]	CWE-494	Download of Code Without Integrity Check
[21]	CWE-732	Incorrect Permission Assignment for Critical Resource
[22]	CWE-770	Allocation of Resources Without Limits or Throttling
[23]	CWE-601	URL Redirection to Untrusted Site ('Open Redirect')
[24]	CWE-327	Use of a Broken or Risky Cryptographic Algorithm
[25]	CWE-362	Concurrent Execution using Shared Resource with Improper Synchronization ('Race Condition')

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.

IBM QRADAR DASHBOARD

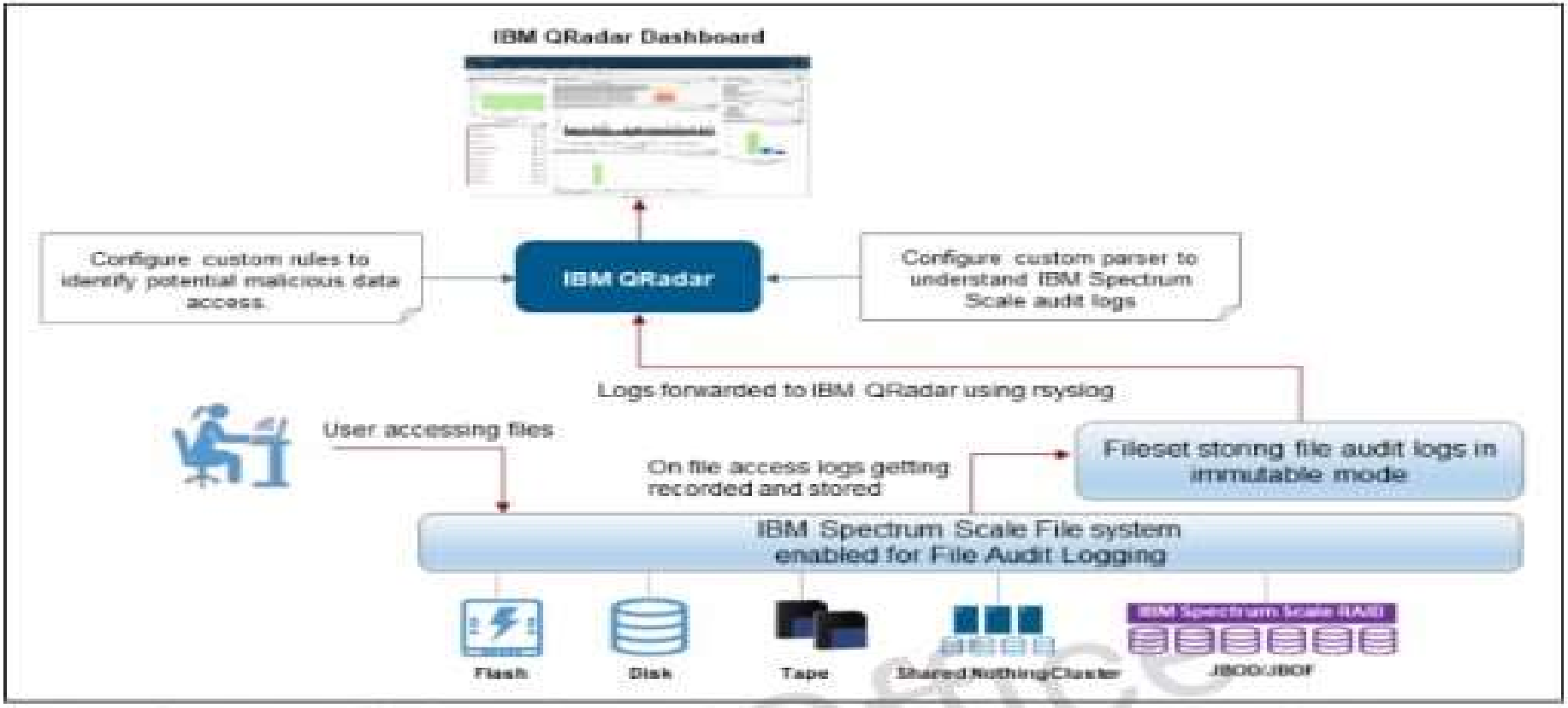


Figure 4 High-level setup: IBM Spectrum Scale and IBM QRadar for storing and analyzing file audit logs

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.