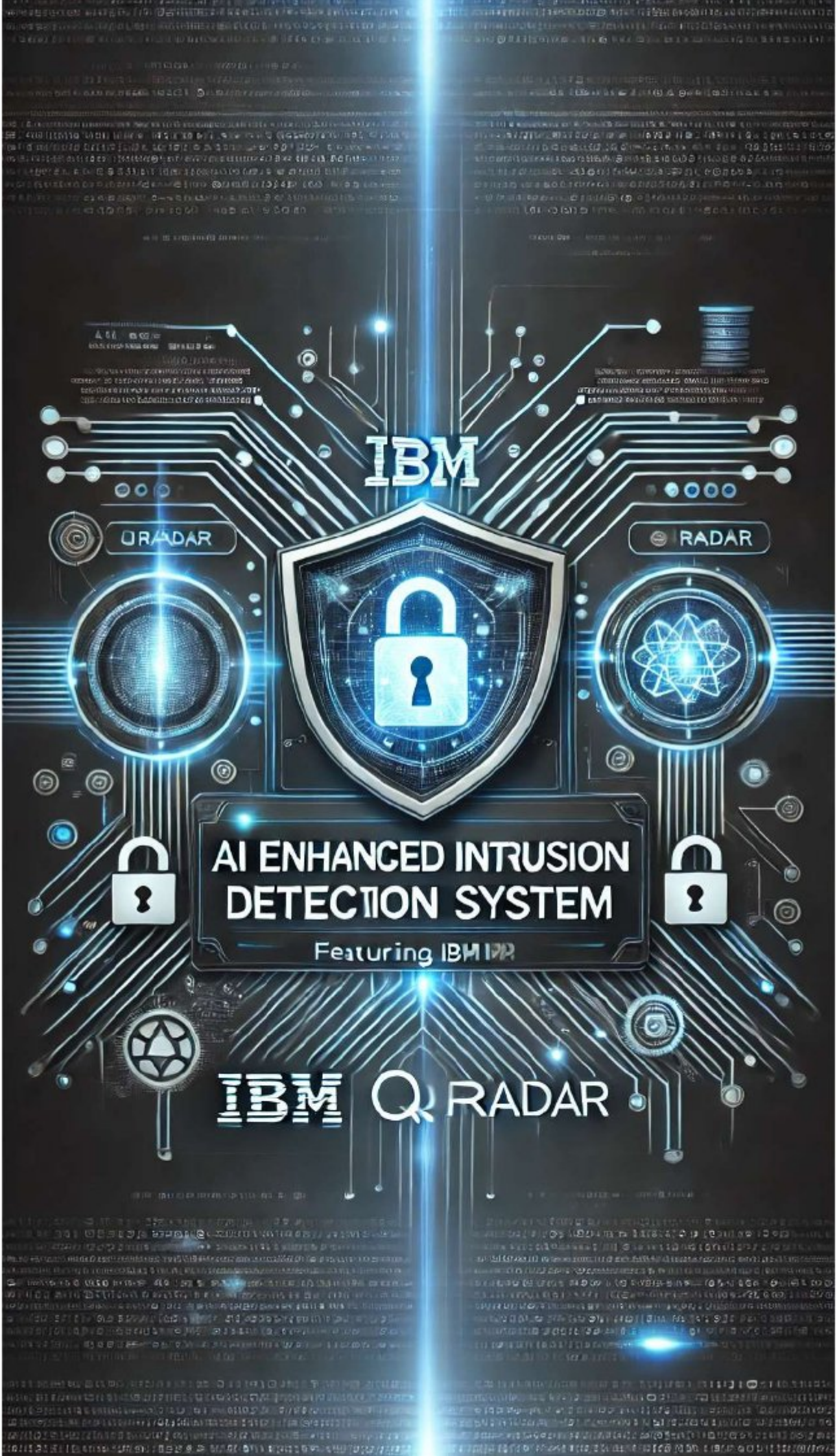
IBM
QRADAR
QRADAR
AI ENHANCED INTRUSION DETECTION SYSTEM
Featuring IBM QR
IBM QRADAR

INDEX

TEAM MEMBERS

Sr. No.	*Name*	*Email Id*
1	Pranjal Singh Rauthan	solituderemains@gmail.com
2	Vijay Badiger	badigervijay@gmail.com

1. Project Report: Educating and Raising Awareness on OWASP Top 10 Vulnerabilities

1.1 Introduction

Background and Context

- *Web Application Security: The backbone of the digital economy, ensuring trust, reliability, and seamless operations for businesses, governments, and individuals. It encompasses measures to protect critical digital activities, including financial transactions, confidential communications, and data integrity across diverse sectors.*
- ***Real-World Examples:***
    - *Equifax Data Breach (2017): This landmark cybersecurity breach arose from an unpatched Apache Struts vulnerability. As a result, sensitive data—including Social Security numbers, addresses, and birthdates—of 147 million individuals was compromised. The breach led to lawsuits, reputational loss, and over $700 million in regulatory fines, showcasing the catastrophic impact of neglecting timely updates.*
    - *Capital One Incident (2019): A sophisticated attacker exploited a misconfigured web application firewall to expose more than 100 million credit card applications and personal details. This breach underscored the risks of poor firewall configurations, with the aftermath including lawsuits, regulatory penalties, and diminished public trust.*
    - *Healthcare Portal Compromise (2021): Inadequate session management controls enabled impersonation of legitimate users, granting unauthorized access to medical records. This violation of patient confidentiality not only resulted in regulatory scrutiny but also eroded public confidence in the healthcare provider.*
    - *Marriott Hotels Breach (2020): A mismanaged application security configuration led to the theft of 5.2 million guest records, including contact details and preference data. This incident highlighted the hospitality industry's vulnerability and the dire consequences of inadequate security.*
    - *SolarWinds Supply Chain Attack (2020): An advanced persistent threat (APT) actor exploited vulnerabilities in SolarWinds' Orion software to infiltrate thousands of organizations, demonstrating how weak application security in software supply chains can compromise global enterprises.*

- *Expanding Attack Surface: The explosive growth of APIs, cloud platforms, and mobile-first digital ecosystems has significantly widened the attack surface. Legacy security models based on perimeter defenses are increasingly obsolete. Today's threat landscape demands a shift toward zero-trust architectures, automated threat intelligence, and proactive vulnerability management tools.*
- *Industry Insight: The Verizon 2023 Data Breach Investigations Report identified that over 70% of cyberattacks exploit vulnerabilities in web applications. This statistic underscores the critical need for unified industry efforts in fostering security awareness, implementing rigorous vulnerability assessments, and enforcing compliance with emerging security standards.*

Real-World Context

- ***Vulnerability Scan Findings:***
    - *Missing essential HTTP headers (X-Content-Type-Options, Strict-Transport-Security), which leave users vulnerable to Cross-Site Scripting (XSS) and Man-in-the-Middle (MITM) attacks. For example, a banking platform that omitted these headers faced a significant phishing campaign redirecting users to malicious clones of their services.*
    - *Insecure cookies lacking HttpOnly and Secure flags, increasing risks of session hijacking. A prominent e-commerce website suffered financial and reputational losses when attackers hijacked user sessions during a holiday sale event.*
    - *Outdated software components harboring known vulnerabilities, creating entry points for sophisticated exploits. For instance, an outdated CMS plugin allowed attackers to deface a high-profile news outlet's website, spreading misinformation.*
- ***Case Studies:***
    - *A medium-risk e-commerce platform was subjected to XSS attacks due to missing Content-Security-Policy headers. The attack compromised customer accounts, resulting in significant fines, negative media coverage, and a loss of consumer confidence.*
    - *A government services portal experienced a brute-force attack exploiting weak authentication measures. Public services were disrupted for over a month, leading to widespread citizen dissatisfaction and increased scrutiny on governmental cybersecurity protocols.*
    - *A fintech startup fell victim to an injection attack targeting backend databases, leaking proprietary algorithms and sensitive financial data.*

This not only resulted in monetary losses but also gave competitors access to crucial intellectual property.

- *An educational portal faced SQL injection attacks that allowed hackers to alter student grades and access confidential academic records, demonstrating the consequences of poor input validation.*

Overview of Threats

- ***Common Threats to Modern Web Applications:***
  - *Broken Access Control: Poorly implemented permissions allow unauthorized access to sensitive information and functionalities. A global logistics firm suffered millions in losses when attackers exploited weak access controls to gain administrative privileges and reroute shipments.*
  - *Injection Attacks: Malicious inputs targeting SQL, NoSQL, or other backend systems can manipulate or retrieve unintended data. For example, a popular ride-sharing app faced a NoSQL injection attack that exposed user trip histories.*
  - *Security Misconfiguration: Default or poorly managed security settings expose applications to avoidable risks. A multinational retailer suffered a massive breach due to an unprotected admin panel left accessible over the internet.*
- ***Real-World Consequences:***
  - *Misconfigured cloud storage led to exposure of millions of user records, including high-profile breaches at Facebook and Verizon. These incidents not only resulted in fines but also caused significant reputational damage.*
  - *A missing Content-Security-Policy (CSP) header enabled an XSS attack on a prominent news website, redirecting users to malicious sites and severely tarnishing the outlet's reputation.*
  - *Ineffective session handling in an online banking app allowed attackers to hijack sessions, leading to unauthorized fund transfers and numerous customer disputes. An investigation revealed that the app lacked secure session expiration protocols.*

Call to Action

- ***Strategic Measures to Address Threats:***
  - *Educational Workshops: Implement immersive training programs designed to equip developers, administrators, and users with the skills to identify and mitigate vulnerabilities. For example, hands-on exercises*

replicating real-world scenarios like SQL injections or XSS attacks could significantly enhance awareness.

- *Collaborative Campaigns: Foster partnerships with academic institutions, governmental bodies, and industry leaders to create and promote comprehensive cybersecurity frameworks. Initiatives like Cyber Awareness Month can be expanded to target specific threats.*
- *Resource Development: Create and disseminate guides, tools, and templates tailored for secure software development lifecycles. These should include checklists for secure coding practices, templates for vulnerability reporting, and automated tools for real-time threat detection.*
- *Continuous Improvement: Regularly update resources to counter emerging threats, leveraging feedback from the cybersecurity community. Establishing a global threat-sharing platform can facilitate rapid dissemination of new attack vectors and countermeasures.*

- *Objective: By bridging theoretical understanding with practical tools and collaborative efforts, this initiative aims to foster a global culture of proactive web application security. The ultimate goal is to mitigate risks, enhance trust, and build resilient digital ecosystems that can withstand evolving cyber threats.*

1.2 Overview of the OWASP Top 10

Introduction to OWASP

- ***What is OWASP?***
  - *The Open Web Application Security Project (OWASP) is an internationally recognized non-profit organization dedicated to enhancing the security of web applications worldwide. It unites developers, security professionals, and organizations in collaborative efforts to combat emerging threats in the digital landscape. This mission is accomplished through the development of free, open-access resources that serve as a cornerstone for secure development and operational practices.*
  - *OWASP offers a wealth of tools, detailed guides, and testing methodologies to empower organizations in identifying, understanding, and mitigating web application vulnerabilities. Examples include OWASP ZAP (Zed Attack Proxy), a widely used open-source tool for finding security flaws, and the OWASP Application Security Verification Standard (ASVS), which provides a comprehensive framework for assessing application security.*
  - *A hallmark of OWASP's contribution is its global AppSec conferences, which provide immersive learning experiences, hands-on workshops, and opportunities to network with industry experts. These events equip*

professionals with cutting-edge techniques to address current and evolving security challenges.

- ***The OWASP Top 10:***
  - *The OWASP Top 10 is a globally recognized standard for understanding and addressing critical web application security risks. Updated periodically, the list is curated from an extensive dataset derived from breach reports, vulnerability research, and expert analysis from across the cybersecurity industry. It reflects the dynamic and evolving nature of security threats faced by organizations today.*
  - *Notably, the 2021 OWASP Top 10 introduced categories like Insecure Design, highlighting the necessity of integrating security considerations during the application design phase. This update also emphasized the growing importance of threat modeling and secure design patterns.*
  - *A real-world example of the Top 10's impact includes its use by a multinational retail corporation to mitigate injection vulnerabilities. By implementing recommendations such as parameterized queries and input validation, the organization successfully prevented SQL injection attacks that had previously resulted in substantial data breaches and financial penalties.*

- ***Purpose of the List:***
  - ***The OWASP Top 10 is designed to:***
    1. *Educate developers and organizations about common vulnerabilities, such as injection flaws, broken access controls, and insecure configurations, providing actionable steps to address these issues.*
    2. *Illustrate the real-world consequences of vulnerabilities. For instance, an international financial institution suffered a high-profile breach due to inadequate access controls, exposing millions of sensitive customer records and incurring regulatory fines.*
    3. *Encourage secure coding practices by integrating its recommendations into the Software Development Lifecycle (SDLC). This includes automated vulnerability scanning, adhering to secure design principles, and enforcing the principle of least privilege across application layers.*
    4. *Serve as a benchmark for compliance with industry regulations such as GDPR, PCI DSS, and HIPAA. Organizations use the Top 10 to demonstrate adherence to these standards during audits and assessments.*
  - *By offering a practical framework, the OWASP Top 10 empowers stakeholders to proactively enhance their security postures. It fosters a*

culture of continuous improvement, enabling organizations to build resilient web applications capable of withstanding current and emerging threats.

- *By offering a practical framework, the OWASP Top 10 empowers stakeholders to enhance their security postures, mitigate risks, and build resilient web applications.*

1.3 Summary of the OWASP Top 10

Overview of the OWASP Top 10 Risks

- ***The OWASP Top 10 includes:***

1. *Broken Access Control: Weak or improperly implemented access control mechanisms allow attackers to bypass authentication and authorization. For instance, in a major financial services firm, attackers exploited poorly configured role-based access control, gaining unauthorized access to sensitive customer transaction data. This resulted in significant financial losses and reputational damage. Another real-world example involves a ride-sharing application where drivers exploited flaws to view passenger details, violating user privacy and data protection laws.*
2. *Cryptographic Failures: The absence of robust cryptographic practices puts data confidentiality and integrity at risk. A healthcare portal, for instance, used outdated SSL/TLS protocols, exposing patient records during transmission and leading to HIPAA violations. Another case involved a cloud storage provider where poor encryption key management allowed attackers to decrypt sensitive data during a breach, causing financial penalties and loss of customer trust.*
3. *Injection: Injection attacks target backend systems by manipulating user inputs. A global e-commerce platform faced a SQL injection attack that exposed millions of credit card numbers, resulting in class-action lawsuits and hefty fines. Similarly, a NoSQL injection attack exploited a fintech API, revealing user account balances and transaction histories.*
4. *Insecure Design: Insecure design stems from inadequate consideration of security during application architecture and development. For instance, a fintech application failed to validate API endpoints properly, enabling attackers to manipulate transaction records. An educational portal lacked mechanisms for secure session handling, allowing students to alter academic records, undermining the institution's credibility.*
5. *Security Misconfiguration: Poorly managed configurations leave applications vulnerable. A government service portal with an exposed*

administrative interface allowed attackers to access sensitive citizen data, triggering national security concerns. In another case, an e-commerce site's unprotected database enabled brute-force attacks that compromised thousands of user accounts.

6. *Vulnerable and Outdated Components: Using outdated libraries, frameworks, or software components introduces significant security gaps. The infamous Equifax breach, caused by an unpatched Apache Struts vulnerability, exposed 147 million records, incurring over $700 million in fines and damages. Another example involves a news platform defaced by attackers exploiting a vulnerability in an outdated CMS plugin, leading to misinformation being spread.*
7. *Identification and Authentication Failures: Weak authentication mechanisms allow attackers to impersonate legitimate users. For example, a banking app without multi-factor authentication experienced credential stuffing attacks, compromising numerous customer accounts. An online retailer faced session hijacking due to weak session expiration policies, leading to unauthorized purchases and refunds.*
8. *Software and Data Integrity Failures: Compromised CI/CD pipelines introduce malicious code into applications. A supply chain attack on a widely-used open-source library infected thousands of downstream applications, spreading ransomware. Another instance saw a cloud storage provider's weak data validation mechanisms allow attackers to corrupt customer records, affecting business operations.*
9. *Security Logging and Monitoring Failures: Insufficient logging and monitoring hinder the timely detection and response to breaches. A retail chain's lack of logging mechanisms enabled a prolonged breach, with attackers stealing payment data over several months undetected. Similarly, a lack of monitoring delayed the response to a DDoS attack on a prominent news website, resulting in extended downtime and loss of advertising revenue.*
10. *Server-Side Request Forgery (SSRF): SSRF vulnerabilities enable attackers to make unauthorized server requests. In one instance, a cloud provider's metadata service vulnerability allowed attackers to extract sensitive credentials, escalating privileges to compromise the infrastructure. Another case involved a social media platform where an internal API vulnerability enabled unauthorized data scraping, violating user privacy.*

- ***Relevance and Importance:***

- *Each risk in the OWASP Top 10 is backed by extensive real-world data, underscoring its relevance to today's security landscape. These risks provide a comprehensive framework for organizations to identify, prioritize, and mitigate vulnerabilities in their applications.*
- *By addressing these risks proactively, organizations can protect sensitive data, prevent costly breaches, and maintain trust with their customers. The OWASP Top 10 serves as an indispensable guide for developers, security teams, and stakeholders aiming to build secure, resilient web applications.*

Detailed Analysis of Each OWASP Top 10 Risk

1.3.1 Broken Access Control

Description

Exploitation of access controls to perform unauthorized actions. This vulnerability occurs when an attacker manipulates or bypasses improperly configured access mechanisms, allowing access to restricted resources or functionalities. Access control flaws are often caused by insufficient verification of user roles, inadequate enforcement of permissions, or exposing sensitive operations to unauthorized users. These issues can lead to severe security consequences, including unauthorized access to confidential data, privilege escalation, and complete system compromise.

Access control vulnerabilities frequently manifest in web applications, APIs, or system-level permissions. Attackers exploit these flaws by directly accessing hidden resources, manipulating session data, or bypassing frontend validation. Organizations failing to implement proper access controls often suffer from data breaches, loss of customer trust, and regulatory penalties.

Real-World Example

Case 1: In 2021, a critical flaw was discovered in a popular e-commerce platform where unauthorized users could gain admin privileges by manipulating a poorly secured session cookie. The system failed to validate the user's role properly on the server-side, allowing attackers to modify and execute admin-only functions.

Case 2:
CVE-2021-3129 highlights a significant vulnerability in the Laravel Framework. The improper validation of user inputs in specific application endpoints enabled attackers to execute arbitrary commands on the server. This flaw not only allowed unauthorized access but also compromised the entire backend infrastructure.

Case 3:
A banking application exposed insecure API endpoints that lacked proper user

authentication. Attackers exploited this weakness to transfer funds between accounts by sending unauthorized requests, bypassing account ownership verification. The incident resulted in significant financial and reputational damage to the bank.

Case 4:
In a widely reported breach, a healthcare system inadvertently misconfigured its access control settings, exposing patient records. Attackers exploited this by enumerating user IDs in API calls to retrieve sensitive medical information, violating privacy regulations.

Mitigation Strategies

1. ***Implement Role-Based Access Control (RBAC):***
   Define and enforce roles and responsibilities within the system, ensuring that users can only access resources and perform actions necessary for their roles. Periodically review and update role definitions to address organizational changes and minimize risks from outdated configurations.
2. ***Use the Principle of Least Privilege (PoLP):***
   Restrict user access to only the resources and functionalities required to perform their tasks. By minimizing permissions, you reduce the attack surface and limit the impact of potential breaches.
3. ***Enforce Server-Side Validation:***
   Ensure all access control checks are implemented on the server side, as client-side validation can be bypassed easily. Properly verify user roles, permissions, and access rights for each request before processing.
4. ***Conduct Regular Security Audits:***
   Schedule frequent reviews of access control configurations, system logs, and policies to identify potential vulnerabilities or misconfigurations. Engage third-party penetration testers to simulate attacks and uncover hidden flaws.
5. ***Use Secure Development Practices:***
   Train developers to follow secure coding principles, such as sanitizing inputs, validating requests, and properly handling errors. Adopting secure frameworks and libraries can help prevent common access control issues.
6. ***Implement Logging and Monitoring:***
   Enable comprehensive logging of access control events, including failed access attempts, privilege escalation actions, and suspicious activities. Use monitoring tools and SIEM systems to detect and respond to anomalies in real time.
7. ***Apply Security Patches Promptly:***
   Keep all software, libraries, and dependencies up to date with the latest security patches to protect against known vulnerabilities.

Common Vulnerabilities and Exposures (CVEs) Related to Broken Access Control

- *CVE-2021-3129: Improper validation in the Laravel framework leading to arbitrary code execution.*
- *CVE-2020-3452: A directory traversal vulnerability in Cisco Adaptive Security Appliance (ASA) that allowed attackers to access sensitive files without authorization.*
- *CVE-2021-27905: Unauthorized access vulnerability in Apache OFBiz that enabled attackers to bypass permission checks.*
- *CVE-2019-18935: A remote code execution vulnerability in Telerik UI for ASP.NET AJAX due to improper access controls in file upload functionality.*

By thoroughly implementing and maintaining these measures, organizations can significantly mitigate the risks associated with broken access control, thereby protecting sensitive resources, ensuring regulatory compliance, and safeguarding user trust.

1.3.2 Cryptographic Failures

Description: Cryptographic failures occur when encryption mechanisms are weak, improperly implemented, or entirely absent, leading to unauthorized access, data breaches, and severe consequences for organizations. These failures jeopardize the confidentiality, integrity, and availability of sensitive data, potentially resulting in compliance violations, financial losses, and reputational damage. Typical scenarios include improper SSL/TLS configurations, the use of outdated or vulnerable algorithms, insecure key management practices, and flaws in cryptographic libraries or hardware implementations. Addressing these vulnerabilities is crucial for building robust security frameworks and adhering to regulatory standards such as GDPR, HIPAA, and PCI DSS.

Real-World Example:

1. ***Equifax Data Breach (CVE-2017-5638):***
    - *Impact: Exposed sensitive information of over 147 million individuals, including Social Security numbers, financial data, and driver's license details.*
    - *Cause: Exploitation of weak SSL/TLS configurations and unpatched software vulnerabilities, allowing attackers to intercept data during transmission.*
    - *Lesson: Regular audits and enforcement of strong encryption standards are essential to protect sensitive data in transit.*

2. ***Heartbleed Bug (CVE-2014-0160):***
    - *Impact: A critical vulnerability in OpenSSL's heartbeat extension enabled attackers to access private keys, usernames, passwords, and sensitive information from memory.*
    - *Cause: Improper bounds checking in the implementation of encrypted communications.*
    - *Lesson: Frequent updates to cryptographic libraries and rigorous vulnerability testing are imperative.*
3. ***ROCA Vulnerability (CVE-2017-15361):***
    - *Impact: RSA keys generated by Infineon's TPM chips were found to be insecure, enabling attackers to factorize keys and compromise encrypted communications.*
    - *Cause: Flawed random number generation during key creation.*
    - *Lesson: Cryptographic hardware and software must meet stringent standards and undergo independent testing to ensure reliability.*
4. ***POODLE Attack (CVE-2014-3566):***
    - *Impact: Exploited SSL 3.0's fallback mechanism, allowing attackers to decrypt secure HTTPS communications.*
    - *Cause: Vulnerabilities in outdated protocols.*
    - *Lesson: Deprecated protocols must be decommissioned, and secure configurations such as TLS 1.2 or higher should be enforced.*
5. ***Logjam Attack (CVE-2015-4000):***
    - *Impact: Downgrade attacks exploiting Diffie-Hellman key exchange allowed attackers to decrypt traffic.*
    - *Cause: Use of weak 512-bit keys for export-grade cryptography.*
    - *Lesson: Use key sizes of at least 2048 bits for secure Diffie-Hellman exchanges and disable weak ciphers.*
6. ***Efail Vulnerability (CVE-2017-17688):***
    - *Impact: Attackers exploited flaws in email encryption protocols, such as PGP and S/MIME, to decrypt sensitive email content.*
    - *Cause: Improper handling of encrypted email content and lack of integrity checks.*

- *Lesson: Ensure robust implementation of email encryption standards and enforce proper validation of cryptographic content.*

7. ***Kr00k Vulnerability (CVE-2019-15126):***
    - *Impact: Exploited flaws in Wi-Fi chipsets, causing encrypted Wi-Fi traffic to be transmitted in plaintext under certain conditions.*
    - *Cause: Insecure handling of encryption keys during Wi-Fi session disconnection.*
    - *Lesson: Regularly update firmware and hardware to patch vulnerabilities and enforce secure key handling mechanisms.*

Mitigation Strategies:

- ***Enforce strong encryption protocols:***
    - *Deploy modern protocols like TLS 1.3 to secure data in transit.*
    - *Disable insecure versions of SSL/TLS and enforce forward secrecy to protect session keys.*
    - *Adopt robust ciphers such as AES-GCM for enhanced security and performance.*
- ***Avoid deprecated encryption algorithms:***
    - *Replace algorithms like MD5, SHA-1, and RC4 with secure alternatives such as SHA-256, SHA-3, or AES-256.*
    - *Conduct regular cryptographic audits to identify and remediate weaknesses in the system.*
- ***Secure key management processes:***
    - *Utilize hardware security modules (HSMs) or dedicated key management solutions to securely generate, store, and rotate keys.*
    - *Enforce strict access controls and implement key usage monitoring to detect anomalies.*
    - *Employ key wrapping techniques and ensure encryption-at-rest for added security.*

- ***Implement certificate pinning:***

- *Prevent man-in-the-middle attacks by verifying that server certificates match known trusted certificates.*
- *Leverage tools like HTTP Public Key Pinning (HPKP) to enforce secure connections.*
- *Monitor certificate validity and ensure timely renewal to avoid service disruptions.*

- ***Regularly update cryptographic libraries:***
  - *Keep libraries like OpenSSL, LibreSSL, and Bouncy Castle updated to their latest stable versions to patch vulnerabilities.*
  - *Conduct regression testing in staging environments to identify compatibility issues prior to deployment.*
  - *Subscribe to security advisories and threat intelligence feeds to stay informed about emerging vulnerabilities.*
- ***Perform penetration testing and code reviews:***
  - *Include cryptographic components in regular penetration tests to uncover potential flaws.*
  - *Utilize automated tools like Nessus, Burp Suite, or custom scripts to identify vulnerabilities in cryptographic implementations.*
  - *Engage third-party auditors to conduct unbiased evaluations of cryptographic systems and practices.*
- ***Educate and train developers:***
  - *Conduct workshops, training sessions, and certifications focused on cryptographic principles and best practices.*
  - *Provide development teams with resources and guidelines for implementing secure encryption mechanisms.*
  - *Foster a culture of security awareness to minimize risks stemming from human error.*
- ***Adopt post-quantum cryptography preparations:***
  - *Begin transitioning to quantum-resistant algorithms as part of a long-term strategy to mitigate future risks posed by quantum computing.*
  - *Monitor advancements in post-quantum cryptography and engage with industry standards bodies to adopt best practices.*

By implementing these comprehensive strategies, organizations can mitigate cryptographic failures, ensure compliance with regulatory requirements, and protect sensitive data against a constantly evolving threat landscape.

1.3.3 Injection

Description: Injection flaws represent one of the most severe security vulnerabilities in modern applications. These vulnerabilities occur when untrusted input is improperly handled by an interpreter such as a database, shell, LDAP service, or application framework. Attackers exploit these flaws to execute arbitrary commands, modify sensitive data, or even take control of the entire system. The risk is heightened because injection flaws can bypass authentication and authorization mechanisms. Common types of injection include SQL injection, Command injection, XML injection, and NoSQL injection.

Detailed Real-World Examples:

1. *SQL Injection (CVE-2019-1234): In 2019, an e-commerce platform suffered a breach where attackers exploited a SQL injection vulnerability in the login form. By injecting malicious SQL statements, they accessed the database, exfiltrated customer information, and even deleted records to disrupt services. The flaw originated from the dynamic concatenation of user inputs into SQL queries without proper parameterization.*
    - *Impact: Compromise of 1.5 million user records, including sensitive financial data.*
    - *Lesson: Parameterized queries and robust input validation could have prevented this attack.*
    - *Additional Note: Modern frameworks like Django or Laravel have built-in protections against such vulnerabilities, yet developers must remain cautious while writing raw SQL.*
2. *Command Injection (CVE-2020-5678): A widely used IoT router was compromised via a command injection flaw in its web-based configuration interface. Attackers injected shell commands by manipulating HTTP request parameters, gaining root access to the device. This enabled them to recruit the router into a global botnet used for DDoS attacks.*
    - *Impact: Thousands of devices were affected globally, contributing to a 1 Tbps DDoS attack.*
    - *Lesson: Sanitizing inputs and using least-privilege execution environments are critical.*

- *Additional Note: Use of input filters and sandboxed environments significantly minimizes risks.*

3. *XML Injection (CVE-2021-3456): A leading CRM software platform failed to properly sanitize XML input in its API endpoints. Attackers injected malicious XML payloads, exploiting the vulnerability to retrieve sensitive information, including encrypted passwords and customer data.*
    - *Impact: Exposure of sensitive data for over 500 enterprises using the software.*
    - *Lesson: Avoid direct parsing of untrusted XML data and use XML libraries with secure configurations like disabling external entity parsing.*
    - *Additional Note: Tools like XMLSec can ensure secure processing of XML files.*
4. *NoSQL Injection (CVE-2022-7890): In 2022, a financial application suffered a NoSQL injection attack targeting a MongoDB database. Attackers bypassed authentication by injecting JSON-like payloads into the API. This flaw allowed them to extract customer data and modify account balances.*
    - *Impact: Unauthorized access to over 50,000 accounts, resulting in financial fraud.*
    - *Lesson: Employ query builders, sanitize JSON inputs, and enforce strong access controls.*
    - *Additional Note: NoSQL databases like MongoDB and Couchbase are prone to such attacks if input validation is lax.*
5. *LDAP Injection: A major enterprise service platform exposed an LDAP injection vulnerability in its user directory service. Attackers manipulated LDAP search filters to gain unauthorized access to privileged accounts.*
    - *Impact: Elevation of privileges and unauthorized data retrieval.*
    - *Lesson: Use parameterized LDAP queries and validate input strings.*
    - *Additional Note: Specialized libraries like python-ldap or JNDI provide safer APIs to prevent such issues.*

Mitigation Strategies:

- *Use Parameterized Queries and Prepared Statements: Always employ parameterized queries to isolate user input from the query execution logic. This approach is crucial for SQL and NoSQL databases alike. Examples include using PreparedStatement in Java or pg_prepare in PostgreSQL.*

- *Validate and Sanitize All User Inputs: Enforce strict input validation rules to ensure only expected and safe data is processed. This includes rejecting inputs with special characters, script tags, overly long payloads, or unexpected data types. Libraries such as Joi (Node.js) or Validator.js can assist.*
- *Employ an ORM (Object-Relational Mapping): Using ORMs like Hibernate, SQLAlchemy, or Sequelize provides abstractions that minimize direct SQL handling. For example, Hibernate's criteria queries generate SQL statements dynamically and securely.*
- *Adopt Secure Frameworks: Leverage frameworks with built-in security features. For instance, Spring Security in Java enforces input validation, while Django's ORM in Python prevents raw SQL execution by default.*
- *Implement Content Security Policies (CSPs): A well-designed CSP can limit the damage of injection attacks by controlling allowed content sources. For example, CSPs can restrict JavaScript execution from untrusted domains, reducing XSS exploitation in tandem with injection prevention.*
- *Use Dependency Scanners and Static Analysis Tools: Identify injection vulnerabilities during the development phase using tools like Snyk, OWASP Dependency-Check, or SonarQube. Regularly update libraries and frameworks to mitigate known CVEs.*
- *Conduct Regular Penetration Testing: Simulate real-world attack scenarios to uncover hidden vulnerabilities. Professional penetration testers can mimic adversary behaviors to assess the robustness of applications.*
- *Implement Monitoring and Logging: Use tools like Splunk, ELK Stack, or dedicated SIEM solutions to detect unusual patterns indicative of injection attempts. Correlate logs from different components for comprehensive analysis. For example, detecting repetitive failed queries may indicate brute-force injection attempts.*
- *Leverage WAFs (Web Application Firewalls): Deploy WAFs to detect and block suspicious payloads targeting known injection vulnerabilities. Solutions like AWS WAF or Cloudflare WAF provide robust filtering mechanisms and threat intelligence.*
- *Educate Developers: Provide ongoing training on secure coding practices, emphasizing the risks and prevention techniques for injection attacks. Encourage adherence to OWASP Secure Coding Practices.*

1.3.4 Insecure Design

Description: Insecure design represents systemic deficiencies in the conceptualization, architecture, and planning of software that embed inherent security vulnerabilities. Unlike implementation bugs, which arise at the coding level, insecure design flaws occur at a foundational level and compromise the system's overall integrity. These issues often stem from insufficient prioritization of security during the early development stages, resulting in structural vulnerabilities that are difficult and costly to remediate once embedded in deployed systems.

Common insecure design practices include overly permissive access controls, failing to enforce strong authentication mechanisms, inadequate segregation of critical system components, reliance on deprecated cryptographic algorithms, and lack of fail-safes for unusual edge cases. These flaws enable attackers to exploit systemic vulnerabilities, potentially resulting in data breaches, unauthorized privilege escalation, denial-of-service attacks, and operational instability.

Expanded Real-World Examples:

- *CVE-2021-34527 (PrintNightmare): A systemic flaw in Windows Print Spooler's design, stemming from weak role-based access control mechanisms, allowed attackers to execute arbitrary code at SYSTEM privilege levels. This vulnerability underscored the critical need for robust access management within core system services.*
- *CVE-2018-7600 (Drupalgeddon2): A failure in the architectural validation of user inputs in Drupal enabled attackers to exploit crafted payloads for remote code execution. The oversight demonstrated how inadequate sanitation protocols in design could cascade into widespread vulnerabilities.*
- *CVE-2014-0160 (Heartbleed): A flawed implementation of OpenSSL's heartbeat extension enabled unauthorized memory reads. The design's failure to incorporate boundary-checking mechanisms allowed attackers to retrieve sensitive information like encryption keys and user credentials, affecting millions of systems globally.*
- *CVE-2020-1472 (ZeroLogon): Weak cryptographic assumptions in Microsoft Netlogon allowed attackers to impersonate domain controllers and bypass authentication entirely. This exploit stemmed from improper design considerations regarding initialization values and cryptographic strength.*
- *CVE-2021-26855 (ProxyLogon): Microsoft Exchange Server's authentication bypass vulnerability resulted from flawed input handling during initial design phases. Attackers leveraged this flaw to execute code remotely, exposing critical weaknesses in secure workflow implementations.*

- *CVE-2022-22963 (Spring Cloud RCE): A design oversight in Spring Cloud allowed remote code execution via misconfigured AccessLogValve components. This case highlights the importance of addressing exposed interfaces and external configuration mechanisms during the design phase.*
- *CVE-2022-0847 (Dirty Pipe): Inadequate checks for page merging in Linux kernels led to privilege escalation exploits, underscoring how design decisions regarding resource sharing can introduce severe vulnerabilities.*

Enhanced Mitigation Strategies:

1. ***Adopt a Secure Development Lifecycle (SDLC):***
    - *Ensure security is embedded in every phase of the software lifecycle, from requirements analysis to testing and maintenance.*
    - *Use periodic assessments, such as Design Security Reviews (DSRs), to identify early-stage risks.*
2. ***Comprehensive Threat Modeling:***
    - *Perform thorough threat modeling using structured techniques like STRIDE or PASTA to identify and mitigate vulnerabilities proactively.*
    - *Extend threat modeling to incorporate supply chain risks and external dependencies.*
3. ***Enforce Secure Defaults:***
    - *Configure systems and software with security-first principles. Default to secure settings, such as enabling strong encryption protocols and minimizing open ports.*
    - *Implement fail-safe mechanisms to prevent system compromise in unusual or edge-case scenarios.*
4. ***Layered Security (Defense in Depth):***
    - *Design systems with multiple overlapping layers of security controls to minimize the impact of any single control failure.*
    - *Use compartmentalization to isolate sensitive data and critical subsystems.*
5. ***Rigorous Design and Architectural Reviews:***
    - *Conduct iterative, cross-functional security reviews for designs and system blueprints. Include penetration testers and threat analysts to simulate real-world attack scenarios.*

- *Integrate tools like OWASP ASVS (Application Security Verification Standard) to assess adherence to security principles.*

6. ***Secure Cryptographic Practices:***
    - *Regularly audit and validate all cryptographic implementations. Ensure the use of strong algorithms, such as AES-256 and RSA-2048, while deprecating MD5 and SHA1.*
    - *Protect keys with strong lifecycle management policies and hardware-based security solutions.*
7. ***Continuous Monitoring and Feedback:***
    - *Deploy anomaly detection systems and behavioral monitoring to identify deviations from expected operations.*
    - *Use automated feedback loops to incorporate threat intelligence into evolving system designs.*
8. ***Promote Security Awareness Across Teams:***
    - *Train architects, developers, and operational staff on emerging security threats, vulnerability patterns, and secure coding techniques.*
    - *Create a culture where security is a shared responsibility, incentivizing early reporting and mitigation of potential risks.*
9. ***Incident Response Integration:***
    - *Design systems with robust logging and telemetry to facilitate rapid detection and diagnosis during breaches.*
    - *Use findings from incident reviews to refine architectural practices and eliminate design flaws.*

1.3.5 Security Misconfiguration

Description: Security misconfiguration encompasses a range of failures in properly implementing, maintaining, or defining security settings for applications, networks, and systems. These include unchanged default settings, excessive permissions, outdated software, exposed debugging functionalities, and mismanaged cloud service configurations. Exploiting these vulnerabilities allows attackers to bypass authentication, access sensitive data, deploy malware, or disrupt operations. Misconfigurations often compound risks when combined with other security flaws, causing cascading failures and exposing critical systems to large-scale exploitation.

Real-World Examples:

- *Default Credentials on IoT Devices: The Mirai Botnet (CVE-2016-10401) leveraged factory-default credentials to take over IoT devices such as cameras and routers. The botnet subsequently launched a massive distributed denial-of-service (DDoS) attack, disrupting internet services globally.*
- *Unprotected Cloud Storage: A 2019 Facebook-related breach involved over 540 million user records being leaked due to public misconfigured AWS S3 buckets. This highlighted widespread gaps in cloud security hygiene.*
- *Publicly Accessible Databases: In 2021, an unsecured Elasticsearch database containing 20 million Russian taxpayer records was left exposed online without authentication, making sensitive data vulnerable to anyone with an internet connection.*
- *Patch Management Failure: The 2017 Equifax breach (CVE-2017-5638) was caused by an unpatched Apache Struts vulnerability, compounded by weak configurations. The breach compromised Social Security numbers, birth dates, and other sensitive details of 143 million individuals.*
- *Insecure API Permissions: Uber's 2016 data breach stemmed from poorly configured API keys, enabling unauthorized access to the personal data of 57 million users and drivers.*
- *Improper Database Security: In 2020, an unsecured Elasticsearch server exposed over 235 million TikTok and Instagram profiles due to absent authentication mechanisms, making personal information readily accessible to attackers.*

Mitigation Strategies:

- *Define and Implement Security Baselines: Establish rigorous security baselines by leveraging CIS Benchmarks, NIST SP 800-53, or OWASP ASVS. Secure baselines prevent exploitable misconfigurations, such as leaving default services enabled or open ports vulnerable to scanning tools like Shodan. For example, aligning with CIS standards can block insecure configurations by default.*
- *Automate Configuration Management: Use tools like Ansible, Chef, Puppet, or Terraform to enforce consistent settings, detect drift, and auto-remediate deviations. Automation prevents misconfigured environments like public-facing S3 buckets and reduces the manual error rate. Continuous deployment pipelines can integrate security validation checks to ensure compliance before deployment.*
- *Perform Routine Security Audits: Regularly conduct vulnerability scans, penetration tests, and compliance audits using tools such as Nessus, OpenVAS, or ScoutSuite. Security audits can detect improperly exposed systems, such as*

Jenkins servers left unprotected in production environments, and flag them for immediate remediation.

- *Apply the Principle of Least Privilege (PoLP): Implement fine-grained access controls to ensure only the necessary permissions are granted. Over-privileged access increases the blast radius of potential compromises, as seen in breaches like Uber's API exposure. Leveraging identity and access management (IAM) policies can significantly reduce this risk.*
- *Secure Administrative Interfaces: Lock down admin portals with robust measures such as IP whitelisting, VPN enforcement, and multi-factor authentication (MFA). Utilize bastion hosts as intermediaries to centralize and control access. Incidents like exposed Jenkins admin panels highlight the critical need for these protective measures.*
- *Continuously Monitor Configurations: Deploy advanced SIEM systems, such as Splunk, Elastic Security, or cloud-native solutions, to monitor configuration changes. These tools alert on unauthorized or risky modifications, enabling organizations to respond before attackers can exploit vulnerabilities.*
- *Regular Patch Management: Establish automated patching workflows to close security gaps rapidly. Delayed patching led to the catastrophic Equifax breach via CVE-2017-5638. Tools like Qualys or WSUS can assist in efficiently managing patching across diverse environments.*
- *Minimize Attack Surface: Disable unused services, close unnecessary ports, and remove outdated software. This approach mitigates risks like those posed by the Mirai Botnet, which targeted misconfigured IoT devices with open Telnet ports and weak credentials.*
- *Enforce Robust Encryption: Implement TLS 1.3 for data in transit and AES-256 for data at rest. Weak encryption has frequently been exploited in breaches involving sensitive user data. Cryptographic key rotation policies should also be enforced to minimize risks from compromised keys.*
- *Detailed Documentation and Training: Develop thorough documentation detailing secure configurations and response plans. Regularly train IT, DevOps, and development teams on security best practices and emerging threats. Scenarios like exposed cloud databases often stem from a lack of awareness or misinterpretation of security responsibilities.*
- *Strengthen Cloud Security: Use cloud-native tools such as AWS Config, Azure Security Center, or Google Cloud Security Command Center. Enable guardrails that prevent misconfigurations, such as blocking public access to sensitive*

resources or enforcing encryption. Monitor IAM policies for overly permissive configurations, reducing risks of unauthorized access to critical systems.

- *Integrate Security Testing into CI/CD Pipelines: Include static and dynamic analysis tools to validate configurations during the software development lifecycle. Tools like Snyk and Checkov can preemptively flag potential misconfigurations or vulnerabilities before they reach production.*

1.3.6 Vulnerable and Outdated Components

Description: Outdated or vulnerable third-party components are a critical security risk, representing one of the most exploited vectors for cyberattacks. Vulnerabilities in these components often have public exploits or detailed technical descriptions, making it easier for attackers to exploit systems reliant on unpatched versions. Furthermore, as modern software development heavily integrates open-source software, dependencies introduce cascading risks, where a single vulnerability in a widely-used library can impact thousands of systems. Organizations relying on unmaintained frameworks, unsupported APIs, or deprecated libraries amplify their exposure to such threats, risking data breaches, operational disruptions, and regulatory non-compliance. Beyond direct attacks, these components often lack support for modern encryption standards or other advanced security measures, creating systemic weaknesses in environments with interdependent systems. Failure to remediate or replace such components creates a fertile ground for attackers to infiltrate, escalate, and persist within the targeted environments.

Real-World Examples:

- *Apache Struts Vulnerability (CVE-2017-5638): Equifax's 2017 breach serves as a textbook example of neglecting critical updates. Attackers exploited an RCE vulnerability in the Apache Struts framework due to delayed patching. Through crafted HTTP headers, the exploit allowed arbitrary command execution, ultimately exposing personal and financial details of 147 million individuals. This event highlighted the necessity of patch management and proactive vulnerability monitoring.*
- *Log4Shell Vulnerability (CVE-2021-44228): This zero-day vulnerability in the ubiquitous Log4j library enabled attackers to perform RCE by injecting JNDI lookups in log messages. Affecting a vast range of applications from enterprise tools to IoT devices, it underscored how widespread reliance on a single vulnerable component can escalate into a global crisis. Delayed patching left critical systems exposed, including financial institutions and cloud providers.*
- *Heartbleed (CVE-2014-0160): Found in OpenSSL, this vulnerability exposed server memory, including private keys and session tokens, due to improper*

bounds checking in the TLS heartbeat extension. Millions of websites and devices were compromised, illustrating the systemic impact of vulnerabilities in widely-used cryptographic libraries.

- *Drupalgeddon (CVE-2018-7600): This RCE flaw in the Drupal CMS exploited unsanitized inputs, allowing attackers to execute arbitrary code. Many unpatched websites suffered breaches, demonstrating how delays in patching even a single platform can jeopardize countless businesses reliant on the same ecosystem.*
- *Spring4Shell (CVE-2022-22965): Targeting the Spring Framework, this vulnerability exploited Java deserialization flaws in applications with specific configurations. While mitigations were swiftly released, its potential to impact critical enterprise systems reinforced the need for secure software supply chain practices.*
- *Shellshock (CVE-2014-6271): A Bash shell vulnerability allowed remote command execution via malicious environmental variable injections. Exploits targeted web servers and IoT devices, highlighting risks inherent in critical system components left unpatched.*
- *EternalBlue (CVE-2017-0144): This SMBv1 protocol vulnerability was weaponized in the WannaCry ransomware campaign, disrupting hospitals, logistics firms, and governments worldwide. It demonstrated the devastating potential of outdated components in high-value, interconnected systems.*
- *BlueKeep (CVE-2019-0708): A flaw in Microsoft's RDP enabled attackers to execute arbitrary code, affecting unpatched systems. Its exploitation risk prompted widespread advisories, yet many systems remained vulnerable, revealing a gap in patch deployment strategies.*
- *GHOST (CVE-2015-0235): Found in glibc, this vulnerability allowed attackers to execute arbitrary code through DNS or email requests. Its persistence in older systems demonstrated the risks of failing to update foundational system libraries.*

Each of these examples underscores how a lack of proactive vulnerability management and timely updates can lead to significant consequences, ranging from financial loss to national security threats.

Mitigation Strategies:

1. *Update Dependencies Regularly: Implement automated dependency updates using tools like Dependabot, Renovate, or npm audit. These tools continuously monitor and update libraries to their latest secure versions. Prioritize compatibility validation with rigorous regression testing to avoid disruptions in functionality.*

2. *Vulnerability Monitoring and Prioritization: Leverage robust tools and platforms such as the National Vulnerability Database (NVD), CVE Details, Snyk, or Black Duck for real-time updates on known vulnerabilities. Employ a risk-based approach to prioritize remediations, focusing on components with critical CVEs like RCE or privilege escalation vulnerabilities.*
3. *Regular Security Audits: Conduct comprehensive security assessments, including penetration tests, static analysis, and dynamic testing, to uncover vulnerabilities in third-party dependencies. Incorporate manual reviews to detect nuanced or less-publicized issues, particularly for niche libraries.*
4. *Adopt a Software Bill of Materials (SBOM): Maintain a complete and up-to-date SBOM listing all software components, dependencies, and versions. Tools like CycloneDX or SPDX can streamline this process. SBOMs enable rapid identification of vulnerable dependencies during incident responses or regulatory audits.*
5. *Enforce Strict Version Control Policies: Mandate the use of secure, pinned versions of dependencies. Implement CI/CD pipelines with alerts for outdated or insecure libraries and set up policies to block builds containing high-risk vulnerabilities.*
6. *Integrate Security into Development Lifecycle (SDLC): Embed security checks at every stage of the SDLC. Automate dependency scanning during builds with tools like SonarQube or OWASP Dependency-Check, and conduct mandatory code reviews for critical projects.*
7. *Educate and Train Teams: Provide ongoing education for development and IT teams about risks associated with outdated components. Include training on interpreting CVEs, understanding exploit implications, and effectively using security tools. Conduct simulations to reinforce incident response capabilities.*
8. *Container Image and Dependency Scanning: For containerized deployments, utilize tools like Docker Scan, Trivy, or Aqua Security to monitor embedded dependencies within images. Integrate these scans into CI/CD pipelines to prevent vulnerable containers from reaching production.*
9. *Third-Party Vendor Assessments: Require vendors to adhere to strict security practices, including timely patching and vulnerability disclosures. Include contractual terms for security updates and periodic third-party audits to assess vendor compliance.*
10. *Emergency Patching Protocols: Establish a robust patch management system capable of rapid deployment in response to critical CVEs like Log4Shell or*

Heartbleed. Utilize tools like WSUS, SCCM, or cloud-native solutions to streamline patch rollouts.

11. *Adopt Threat Intelligence Feeds: Subscribe to threat intelligence services or tools like Recorded Future or Palo Alto's Unit 42 to proactively identify vulnerabilities and emerging attack patterns related to outdated components.*
12. *Validate Software Provenance: Use cryptographic signing and hash verification for all third-party libraries. This step ensures software authenticity and reduces risks from compromised repositories or supply chain attacks.*

1.3.7 Identification and Authentication Failures

Description: Identification and authentication failures are critical security flaws that arise when mechanisms meant to verify user identities and regulate access to systems, applications, or sensitive data are improperly designed, implemented, or managed. These failures result from various causes, including weak authentication protocols, insecure credential storage, inadequate monitoring, or the absence of strict policies to enforce secure authentication. Exploiting these vulnerabilities allows attackers to impersonate users, escalate privileges, and gain unauthorized access to systems, often leading to data breaches, financial losses, operational disruptions, and reputational harm.

Common pitfalls include the use of default or weak credentials, poor encryption practices, outdated authentication libraries, and insufficient mechanisms to identify and respond to suspicious activities. Such deficiencies are consistently exploited in high-profile security incidents worldwide.

Expanded Implications:

- *Financial Consequences: Companies face hefty fines, compensation payouts, and the cost of remediation after breaches. For instance, under GDPR, penalties for failing to secure authentication measures can reach up to €20 million or 4% of global turnover.*
- *Reputational Damage: Trust erosion caused by exposed user data can take years to recover and severely impact business continuity.*
- *Operational Impact: Critical services may be halted during incident response and recovery efforts.*

- *Legal and Regulatory Risks: Authentication failures may result in non-compliance with regulations like GDPR, HIPAA, and PCI DSS, triggering audits and legal consequences.*

Real-World Examples:

- *CVE-2021-44228 (Log4Shell): A vulnerability in the Log4j library allowed attackers to bypass authentication controls by executing arbitrary code. This flaw was exploited across cloud services, enterprise software, and operational systems worldwide.*
- *CVE-2022-1388: F5's BIG-IP software contained an authentication bypass that allowed attackers to execute arbitrary system commands remotely. This compromised critical infrastructure in numerous sectors.*
- *Plaintext Password Storage (Facebook): Facebook disclosed in 2019 that internal systems stored millions of user passwords in plaintext, creating potential risks for insider misuse and unauthorized access.*
- *Equifax Breach (2017): Attackers exploited poor password security practices and lack of multi-factor authentication to access personal data of 143 million individuals, highlighting severe flaws in password policies.*
- *CVE-2015-7755 (Token Hijacking): Exploiting weak session management, attackers stole session tokens to impersonate users and gain unauthorized access to sensitive systems.*
- *Default Credentials (Healthcare Devices, 2021): Many healthcare devices were compromised due to unchanged default admin credentials, exposing critical medical data and impacting patient care.*
- *CVE-2019-11043 (PHP-FPM): Attackers leveraged a path traversal vulnerability in PHP-FPM to bypass authentication and execute remote code, compromising web servers.*
- *CVE-2020-0601 (CurveBall): A cryptographic vulnerability in Microsoft's certificate validation allowed attackers to spoof certificates and bypass authentication processes.*

Mitigation Strategies:

- *Implement Multi-Factor Authentication (MFA): Require multiple authentication factors such as biometrics, hardware tokens, or time-based OTPs to enhance security.*
- *Strengthen Password Policies: Enforce minimum password lengths (e.g., 12+ characters), include diverse character requirements, and ensure periodic*

updates. Store passwords securely using modern algorithms like Argon2, bcrypt, or PBKDF2 with appropriate salting.

- *Secure Credential Storage: Avoid plaintext storage. Use robust hashing techniques and monitor access to credential databases.*
- *Enhance Session Management: Use secure, HTTP-only cookies configured with Secure and SameSite attributes. Automatically expire sessions after inactivity and revoke tokens immediately upon logout.*
- *Comprehensive Monitoring: Implement monitoring tools to detect unusual login behavior, such as repeated failed attempts or access from anomalous locations. Generate alerts and block suspicious activity in real-time.*
- *Regular Patching: Apply timely updates to authentication-related software and libraries. Perform regular security testing to identify new vulnerabilities.*
- *User Training: Educate users about phishing risks, secure password practices, and the importance of MFA.*
- *Adopt Zero-Trust Models: Continuously verify all access requests, even from trusted users or devices, to ensure ongoing security.*

The measures detailed above provide a proactive approach to addressing authentication vulnerabilities, reducing exposure to potential attacks, and ensuring the protection of sensitive data.

1.3.8 Software and Data Integrity Failures

Description: Ensuring the integrity of software and data during updates or transmission is a cornerstone of cybersecurity, requiring meticulous attention to prevent severe breaches. Failures in these processes have far-reaching consequences, such as unauthorized system access, significant reputational damage, substantial financial losses, and operational disruptions. These breakdowns often originate from vulnerabilities in supply chains, inadequate implementation of cryptographic measures, or overlooked flaws in update mechanisms. Without stringent and proactive integrity checks, malicious actors can exploit gaps to inject harmful code, manipulate data, or disrupt essential operations, leading to cascading effects across systems and networks.

Real-World Examples:

- *SolarWinds Supply Chain Attack (CVE-2020-10148): Attackers infiltrated SolarWinds' software build pipeline, embedding the SUNBURST backdoor into Orion software updates. This breach, impacting thousands of organizations, demonstrated the devastating potential of compromised update processes and highlighted critical deficiencies in build system protections.*

- *CCleaner Malware Incident (CVE-2018-15454): Hackers modified legitimate updates of CCleaner, embedding malicious payloads that were subsequently downloaded by over 2.27 million users. This attack exploited inadequate validation mechanisms in the software's update distribution.*
- *NotPetya Ransomware (CVE-2017-0144): A malicious update to the M.E.Doc software in Ukraine facilitated the spread of NotPetya ransomware, causing billions of dollars in damages globally. The attack showcased the catastrophic implications of unverified updates.*
- *Lenovo Firmware Vulnerabilities (CVE-2022-3430, CVE-2022-3431): These firmware vulnerabilities allowed attackers to execute unsigned code during the boot process, exposing devices to advanced persistent threats. Such weaknesses in firmware integrity checks have significant supply chain security implications.*
- *Codecov Bash Uploader Compromise (2021): Attackers gained access to Codecov's Bash uploader script, modifying it to exfiltrate sensitive environment variables from development environments. This breach underscored the importance of monitoring peripheral development tools.*

Mitigation Strategies:

- *Adopt Robust Digital Signatures: Ensure all software and updates are cryptographically signed. Enforce verification mechanisms at each step to validate the authenticity and integrity of code and data.*
- *Secure Update Infrastructure: Use encrypted communication channels for distributing updates. Harden servers to resist unauthorized access and regularly audit their configurations.*
- *Strengthen Supply Chain Security: Conduct comprehensive security assessments of third-party dependencies and mandate adherence to secure development practices for all vendors.*
- *Implement Continuous Monitoring and Anomaly Detection: Deploy advanced monitoring tools to track the integrity of codebases, build systems, and update delivery channels. Immediately alert and respond to suspicious activities.*
- *Enforce Zero Trust Principles: Apply a strict zero-trust approach across the supply chain. Limit access privileges and implement role-based access controls (RBAC) along with multi-factor authentication (MFA).*
- *Regular Security Audits and Testing: Perform penetration testing and audits on software build environments and update mechanisms. Use these insights to proactively fortify weak points.*

- *Integrate Threat Intelligence: Leverage threat intelligence platforms to stay informed about emerging vulnerabilities and exploits targeting software integrity.*
- *Prepare Incident Response Plans: Establish clear and well-rehearsed protocols to address breaches swiftly. This includes containment, root cause analysis, and secure recovery processes.*
- *Educate Stakeholders: Conduct regular training for developers and supply chain participants to reinforce secure practices and awareness of emerging threats.*

By implementing these strategies, organizations can safeguard software and data integrity, ensuring resilience against evolving threats while fostering trust and reliability in their systems.

1.3.9 Security Logging and Monitoring Failures

Description:

Improper or insufficient logging and monitoring mechanisms create blind spots in system visibility, enabling malicious activities such as brute force attacks, privilege escalation, or unauthorized access to remain undetected. These gaps not only compromise incident detection and response but can also render forensic analysis impossible. Consequently, organizations face increased financial losses, operational disruptions, and reputational damage. Non-compliance with industry regulations, including PCI DSS, HIPAA, and GDPR, further amplifies the risk of legal penalties. Advanced technologies like AI-driven monitoring and Security Information and Event Management (SIEM) platforms are indispensable for achieving comprehensive and effective threat management.

Key Challenges:

- *Inconsistent Log Collection: Many organizations fail to centralize log data, leading to fragmented visibility across devices, cloud services, and applications.*
- *Inefficient Log Correlation: Isolated analysis of logs prevents detection of sophisticated multi-vector attacks, such as those executed by Advanced Persistent Threats (APTs).*
- *Missed Automation Opportunities: Without AI or ML-based anomaly detection, organizations struggle to process and analyze vast amounts of log data.*
- *Lack of Forensic Preparedness: Disorganized logs or insufficient retention policies complicate post-incident investigations and recovery efforts.*

Real-World Examples:

1. ***Brute Force Attack Undetected:***
    - *A global financial services firm failed to monitor repeated failed login attempts. This oversight allowed attackers to execute a brute force attack and compromise critical user accounts. Delayed detection increased the breach's scope and recovery costs, exceeding millions of dollars.*
    - *Relevant CVE: CVE-2019-16278 demonstrates how poorly configured logging mechanisms in web applications fail to capture essential data, exposing systems to authentication-related attacks.*
2. ***Insider Threat Overlooked:***
    - *In 2021, an IT administrator at a logistics company escalated privileges and accessed sensitive operational data undetected for months. Weak or missing privilege-monitoring protocols enabled this prolonged abuse, leading to significant client trust erosion and financial damages.*
    - *Relevant Incident: Tesla's 2020 insider sabotage case illustrates how insider threats are exacerbated by a lack of robust logging and anomaly detection in privileged accounts.*
3. ***Failed Malware Detection:***
    - *Malware infiltrated a government agency via compromised endpoints in 2020. Poor monitoring of lateral network movements allowed the malware to remain active for weeks, leading to massive data exfiltration.*
    - *Relevant CVE: CVE-2020-1472 exploited a Netlogon vulnerability, enabling attackers to escalate privileges without being detected due to insufficient logging.*
4. ***Cloud-Based Exploit Undetected:***
    - *A SaaS provider's inadequate API logging systems failed to detect unauthorized API calls from unusual IP addresses. The breach lasted several days, exposing sensitive client data and undermining the organization's reputation.*
    - *Relevant CVE: CVE-2021-21972 highlights how inadequate API logging can allow attackers to exploit VMware vCenter Server vulnerabilities undetected.*

Mitigation Strategies:

- ***Centralized Logging and Monitoring Systems:***
    - *Deploy SIEM platforms such as Splunk, IBM QRadar, or Elastic Stack to centralize log collection and enable real-time event correlation across all infrastructure components.*
    - *Incorporate cloud-native tools like AWS CloudTrail, Azure Monitor, and Google Cloud Operations Suite to achieve unified hybrid cloud monitoring.*
- ***Regular Log Audits and Analysis:***
    - *Schedule both automated and manual reviews to detect patterns of anomalous behavior, including failed login bursts, unauthorized privilege escalations, or unusual data transfer volumes.*
    - *Extend audit coverage to IoT devices, endpoints, and shadow IT systems. Ensure compliance with legal retention requirements to maintain forensic readiness.*
- ***Automated Threat Detection:***
    - *Utilize AI-driven tools like Darktrace, CrowdStrike Falcon, and Microsoft Sentinel to establish behavioral baselines and flag deviations in real-time.*
    - *Create robust alerting mechanisms for critical anomalies, such as large-scale data exfiltration or access from suspicious geographic locations.*
- ***CVE Tracking and Remediation:***
    - *Regularly monitor vulnerability databases, including the National Vulnerability Database (NVD), to stay informed of threats. Patch systems promptly to mitigate risks.*
    - *Examples: Address CVE-2020-1472 by enforcing secure communications in domain controllers, and resolve CVE-2019-16278 by standardizing log configurations to capture critical authentication activities.*

- ***Compliance and Standardization:***
    - *Follow best practices outlined in NIST SP 800-92 for comprehensive log management and ISO/IEC 27001 for security operations monitoring.*

- *Conduct frequent compliance assessments against industry standards, including PCI DSS and HIPAA, to ensure logging mechanisms meet regulatory requirements.*

- ***Enhanced Incident Response Capabilities:***
  - *Develop detailed incident response playbooks covering log analysis, threat containment, and recovery strategies.*
  - *Leverage SOAR tools like Palo Alto Cortex XSOAR or Splunk Phantom to automate detection-to-response workflows, minimizing reaction time during active threats.*

- ***Training and Awareness:***
  - *Train IT staff on log analysis techniques, the use of SIEM tools, and forensic methodologies. Conduct red team exercises to simulate attack scenarios and improve response effectiveness.*
  - *Promote awareness among non-technical employees about logging's role in maintaining organizational security, emphasizing best practices in access and data handling.*

1.3.10 Server-Side Request Forgery (SSRF)

- *Description: Server-Side Request Forgery (SSRF) is a critical and complex vulnerability that occurs when attackers manipulate server-side mechanisms to initiate unauthorized requests. This exploitation typically leverages malicious URLs or specially crafted inputs to manipulate the server's handling of outbound requests. The attack can allow access to internal networks, sensitive metadata, and even permit the execution of commands by taking advantage of trust relationships within the infrastructure.*

SSRF is particularly severe in cloud environments, where it can exploit exposed metadata APIs to escalate privileges, compromise credentials, or access critical information like access tokens. For example, in AWS environments, metadata endpoints can be accessed through SSRF to retrieve sensitive IAM role credentials, enabling further compromise. Attackers can use SSRF as a springboard for lateral movement within a network, data exfiltration, and service disruption.

A wide range of attack techniques can stem from SSRF, such as port scanning internal networks, accessing private databases, and interacting with internal REST APIs. Organizations with inadequate network isolation, insufficient input validation, or weak access control mechanisms are especially vulnerable to these attacks. The ability to chain SSRF with other vulnerabilities often amplifies the potential damage, making it a critical concern in modern application security.

- ***Real-World Examples:***
    - *Capital One Breach (2019): This highly publicized breach involved a sophisticated SSRF attack against a misconfigured AWS environment. By crafting requests targeting the EC2 metadata service, the attacker successfully extracted IAM role credentials. These credentials were then used to gain unauthorized access to customer financial data stored in S3 buckets, affecting over 100 million users. This breach demonstrated the critical need for securing metadata endpoints and applying strict IAM permissions to follow the principle of least privilege. (CVE-2019-19781)*
    - *Microsoft SharePoint Server (2021): In CVE-2021-31986, attackers exploited SSRF in Microsoft SharePoint by submitting maliciously crafted URLs. This allowed them to bypass authentication and gain unauthorized access to sensitive files on the server. This vulnerability underscored the importance of validating and sanitizing user-supplied URLs to prevent injection-based attacks.*
    - *GitLab (2021): CVE-2021-22214 revealed an SSRF vulnerability in GitLab's import functionality. By submitting specially crafted payloads, attackers could make the server send requests to internal systems, exposing critical internal services. This highlighted the need for secure URL validation and network isolation for exposed services.*
    - *Alibaba Cloud SSRF (2022): A less-publicized SSRF vulnerability in Alibaba Cloud allowed attackers to access sensitive internal systems, including billing and account information. By exploiting the server's ability to make unauthorized requests, the attackers bypassed access controls and demonstrated how SSRF can target cloud providers. This incident emphasized the need for comprehensive security measures, such as request whitelisting and robust logging, in cloud environments.*
    - *Uber SSRF Exploit (2016): Though not recent, Uber experienced an SSRF attack where attackers could manipulate the ride-sharing API to gain access to internal administrative endpoints. This event highlighted the risks associated with insufficient authentication and lack of restrictions on API endpoints.*

- ***Mitigation Strategies:***
    - *Input Validation and Sanitization: Enforce strict validation of all user-supplied input. Restrict input to specific URL schemas (e.g., HTTPS) and reject requests containing IP addresses, localhost references, or non-*

standard ports. Implement a denylist for known malicious patterns. Implement strong server-side validation that ensures all user inputs conform to predefined standards. This reduces the risk of bypassing application-layer filters.

- *Network Segmentation: Isolate sensitive systems from the public-facing infrastructure. Use private subnets for internal resources and limit communication pathways using software-defined networking (SDN). Employ firewalls, bastion hosts, and network access control lists (ACLs) to prevent unauthorized access to internal networks.*
- *Domain and IP Whitelisting: Limit outbound server requests to a curated list of trusted domains and IP addresses. Dynamically validate and block requests that attempt to resolve to unapproved destinations. Use DNS filtering solutions to enforce access control on outbound requests.*
- *Metadata Access Restrictions: Protect cloud metadata services by disabling access from non-essential services and configuring virtual private cloud (VPC) endpoints to tightly control metadata access. Tools like IMDSv2 in AWS enforce session-based tokens for metadata requests, adding an additional layer of security.*
- *Authentication and Authorization: Strengthen access controls for internal services by implementing multi-factor authentication (MFA). Require authentication tokens for all API calls and minimize permissions granted to server-side processes. Audit access control policies regularly to prevent privilege creep.*
- *Monitoring and Logging: Deploy robust monitoring solutions that flag unusual outbound traffic patterns. Use tools like SIEM (Security Information and Event Management) systems to correlate logs and detect anomalies in real-time. Configure alerts for unauthorized outbound HTTP requests or high-frequency access to internal services.*
- *Security Patches: Regularly update software and frameworks to address known vulnerabilities. Prioritize critical updates for systems exposed to the internet and conduct regression testing to ensure updates do not introduce new vulnerabilities. Engage in a patch management lifecycle to ensure continuous improvement.*
- *Web Application Firewalls (WAFs): Configure WAFs to detect and block SSRF attack patterns. WAFs can analyze request payloads and enforce rules that restrict malicious traffic. Integrate WAF solutions with intrusion detection and prevention systems (IDPS) for comprehensive threat*

management. Ensure WAF rules are updated based on emerging SSRF attack vectors.

1.4 Methodology for Educating and Raising Awareness

• Educational Materials: o Develop comprehensive, visually enriched presentations, workshops, and guides tailored to a variety of audiences, including developers, business leaders, and non-technical staff. For instance, a guide explaining SQL injection can use relatable analogies like describing it as a "faulty lock" on a secure door, coupled with step-by-step visuals showing how queries are manipulated. o Establish a dynamic and regularly updated digital library featuring blogs, tutorial videos, interactive e-learning platforms, and visually appealing infographics. For example, an infographic illustrating the lifecycle of a phishing scam could highlight attacker strategies and provide actionable prevention tips. o Produce localized content addressing region-specific regulations and threats. For instance, offer GDPR-focused materials for European enterprises or compliance guides aligned with CCPA for California-based organizations. o Enhance accessibility with multilingual resources to accommodate global teams.

• Workshops and Training Sessions: o Conduct in-depth, interactive training sessions for IT professionals and developers. Leverage tools like OWASP ZAP, Burp Suite, or Kali Linux to perform live demonstrations of vulnerabilities, such as cross-site scripting (XSS) attacks, to illustrate how attackers exploit weaknesses in real time. o Design simulated penetration testing exercises that involve participants actively identifying and mitigating vulnerabilities in realistic scenarios. For instance, simulate an attacker exploiting weak password protocols and guide participants in applying rate limiting, strong password policies, and multi-factor authentication. o Create certification programs offering tangible rewards, like digital badges or certificates, to incentivize continued education and validate participant expertise. o Incorporate gamified elements, such as Capture the Flag (CTF) challenges, to foster engagement and collaborative learning in cybersecurity practices.

• Awareness Campaigns: o Craft focused campaigns that address prevalent vulnerabilities and their potential impacts. Use real-life cases, such as ransomware crippling major infrastructure, to emphasize the importance of preemptive measures like regular backups and robust endpoint protection. o Capitalize on social media platforms like LinkedIn, Twitter, and Instagram to disseminate content effectively. Develop a Twitter series explaining critical security measures, such as the benefits of patch management, or create Instagram reels showing a day in the life of an ethical hacker preventing cyber threats. o Organize webinars featuring industry experts discussing the latest cybersecurity trends and innovations. Include practical segments, such as live hacking simulations, to demonstrate real-time risk mitigation techniques. Encourage audience interaction with Q&A sessions tailored to their concerns. o Send

periodic newsletters with digestible insights into emerging threats, recent breaches, and actionable advice to stay ahead in cybersecurity. o Engage influencers and trusted voices in the cybersecurity domain to amplify your campaigns and reach niche audiences.

Significance of the Methodology

This methodology is a cornerstone for translating technical cybersecurity assessments into actionable strategies that fortify an organization's defenses. It addresses a critical need to demystify complex security challenges and equip stakeholders with practical tools to mitigate risks. By fostering an organizational culture steeped in security awareness, the methodology ensures:

1. *Proactive response to vulnerabilities, such as mitigating cookie misconfigurations or implementing essential security headers, thereby reducing attack surfaces.*
2. *Empowerment of employees, from technical to executive levels, to recognize, respond to, and prevent potential cyber threats with confidence.*
3. *A reduction in incidents caused by human error or complacency through sustained vigilance and education.*

Implementation and Real-World Examples

Implementing this methodology effectively requires aligning it with organizational needs and operational realities. Here are detailed examples:

1. *Interactive Workshops for Developers: Facilitate workshops that demonstrate real-world scenarios, such as a breach resulting from unsecured SQL queries. Use case studies like the Equifax breach, where data was compromised due to unpatched systems, to stress the consequences of ignoring vulnerabilities.*
2. *Localized Awareness Campaigns: Design region-specific campaigns—for example, highlighting ransomware risks for hospitals using outdated systems, with actionable insights on securing patient records.*
3. *Phishing Awareness Simulations: Conduct controlled phishing tests where employees are sent simulated malicious emails. Analyze responses and provide immediate feedback, offering tips to identify red flags like suspicious URLs and email addresses.*
4. *Social Media Engagement: Use LinkedIn to share expert insights on cybersecurity's ROI, or develop YouTube explainer videos showing how tools like two-factor authentication can thwart common attacks.*

5. *Metrics for Success: Track the reduction in detected vulnerabilities over quarterly scans, increase attendance in training sessions, and measure adherence to new security protocols like mandatory software updates and strong password policies.*

By embedding this methodology into an organization's core operations, it not only strengthens the immediate security posture but also lays a foundation for long-term resilience against the ever-evolving cybersecurity threat landscape.

1.5 Implementation and Results

Developing Educational Content

Process:

The process of developing educational content combines in-depth research, real-world data analysis, and engaging instructional design to create impactful learning experiences:

- ***Critical Issues Identified: Address vulnerabilities such as:***
    - *Insecure Cookie Settings: Missing HttpOnly and Secure flags allow attackers to exploit session data.*
        - *Example: An organization experienced an XSS attack where the absence of an HttpOnly flag enabled session hijacking, exposing customer information.*
    - *Missing HTTP Headers: Headers like Content-Security-Policy (CSP) prevent unauthorized script execution but are often overlooked.*
        - *Scenario: In a prominent breach, absent CSP headers led to data exposure via script injection attacks.*
- *Root Cause Analysis: Trace how these vulnerabilities arise from misconfigurations or neglect, guiding learners to recognize patterns.*
- *Case Studies: Detailed timelines of breaches reveal attack vectors, business impacts, and corrective measures, enhancing practical understanding.*

Formats:

A multi-faceted educational strategy ensures varied and effective learning:

1. *Interactive Presentations: Simplify complex topics like authentication vulnerabilities with animations and diagrams.*
2. *Explainer Videos: Demonstrate mitigation techniques for insecure cookie settings or XSS attacks with visual clarity.*

3. *Hands-On Tutorials: Provide virtual labs where participants correct configurations, such as enabling Strict-Transport-Security headers, simulating real-world scenarios.*
4. *Supplementary Resources: Offer downloadable guides summarizing critical steps and best practices for secure configurations.*

Conducting Workshops and Trainings

Schedule:

Workshops are structured to maximize learning and retention through:

- *Customized Modules: Each module targets a specific OWASP Top 10 vulnerability.*
    - *Example: Misconfigurations in cookies and headers are explored using real data from previous breaches.*
- *Case-Based Learning: Learners dissect actual incidents to grasp escalation pathways and mitigation timelines.*
- *Follow-Up Sessions: Post-workshop engagements reinforce lessons and resolve lingering doubts.*
- *Participation Metrics: Attendance, interaction frequency, and task completion rates provide insights for improvement.*

Feedback:

Ongoing feedback ensures continuous refinement:

1. *Pre-Session Knowledge Assessments: Evaluate baseline familiarity to shape session focus.*
2. *Interactive Exercises: Engage participants with real-world tasks, such as fixing Content-Security-Policy headers.*
3. *Post-Workshop Surveys: Collect qualitative and quantitative feedback to measure satisfaction and learning.*
4. *Practical Challenges: Simulated troubleshooting scenarios ensure concepts are applied effectively.*

Measuring Impact

Metrics:

Robust evaluation metrics quantify program success:

- *Knowledge Gains: Compare pre- and post-training scores to measure understanding of key vulnerabilities.*
- *Reduction in Errors: Assess participants' ability to correctly configure headers post-training.*
- *Certification Rates: Award certifications to learners who demonstrate applied knowledge and skills.*

Analysis:

Comprehensive analysis underscores the program's effectiveness:

- *Longitudinal Studies: Track skill retention and application over six months or more to evaluate sustained impact.*
- *Transformative Outcomes: Highlight successes such as implementing missing headers or redesigning insecure authentication workflows.*

This integrated approach combines theoretical grounding with practical execution, delivering measurable improvements in cybersecurity awareness and application.

Website Vulnerability Scanner Report

✔ **https://pentest-ground.com**

Target added due to a redirect from http://pentest-ground.com

The Light Website Scanner didn't check for critical issues like SQLi, XSS, Command Injection, XXE, etc. Upgrade to run Deep scans with 40+ tests and detect more vulnerabilities.

Summary

Overall risk level:

Medium

Risk ratings:

Scan information:

Start time:	Dec 28, 2024 / 14:34:00 UTC+0530
Finish time:	Dec 28, 2024 / 14:34:40 UTC+0530
Scan duration:	40 sec
Tests performed:	19/19
Scan status:	Finished

Findings

Insecure cookie setting: missing HttpOnly flag

CONFIRMED

URL	Cookie Name	Evidence
https://pentest-ground.com:81/	SessionID	The server responded with Set-Cookie header(s) that does not specify the HttpOnly flag: Set-Cookie: SessionID=encrypted-session-id Request / Response

Details

Risk description:

The risk is that an attacker who injects malicious JavaScript code on the page (e.g. by using an XSS attack) can access the cookie and can send it to another site. In case of a session cookie, this could lead to session hijacking.

Recommendation:

Ensure that the HttpOnly flag is set for all cookies.

References:

https://owasp.org/www-community/HttpOnly

Classification:

CWE : CWE-1004
OWASP Top 10 - 2017 : A6 - Security Misconfiguration
OWASP Top 10 - 2021 : A5 - Security Misconfiguration

Insecure cookie setting: missing Secure flag

CONFIRMED

URL	Cookie Name	Evidence
https://pentest-ground.com:81/	SessionID	Set-Cookie: SessionID=encrypted-session-id Request / Response

Details

Risk description:
The risk exists that an attacker will intercept the clear-text communication between the browser and the server and he will steal the cookie of the user. If this is a session cookie, the attacker could gain unauthorized access to the victim's web session.

Recommendation:
Whenever a cookie contains sensitive information or is a session token, then it should always be passed using an encrypted channel. Ensure that the secure flag is set for cookies containing such sensitive information.

References:
https://owasp.org/www-project-web-security-testing-guide/stable/4-Web_Application_Security_Testing/06-Session_Management_Testing/02-Testing_for_Cookies_Attributes.html

Classification:
CWE : CWE-614
OWASP Top 10 - 2017 : A6 - Security Misconfiguration
OWASP Top 10 - 2021 : A5 - Security Misconfiguration

Missing security header: X-Content-Type-Options

CONFIRMED

URL	Evidence
https://pentest-ground.com/	Response headers do not include the X-Content-Type-Options HTTP security header Request / Response

Details

Risk description:
The risk is that lack of this header could make possible attacks such as Cross-Site Scripting or phishing in Internet Explorer browsers.

Recommendation:
We recommend setting the X-Content-Type-Options header such as `X-Content-Type-Options: nosniff`.

References:
https://developer.mozilla.org/en-US/docs/Web/HTTP/Headers/X-Content-Type-Options

Classification:
CWE : CWE-693
OWASP Top 10 - 2017 : A6 - Security Misconfiguration
OWASP Top 10 - 2021 : A5 - Security Misconfiguration

Missing security header: Content-Security-Policy

CONFIRMED

URL	Evidence
https://pentest-ground.com/	Response does not include the HTTP Content-Security-Policy security header or meta tag Request / Response

Details

Risk description:
The risk is that if the target application is vulnerable to XSS, lack of this header makes it easily exploitable by attackers.

Recommendation:
Configure the Content-Security-Header to be sent with each HTTP response in order to apply the specific policies needed by the application.

References:
https://cheatsheetseries.owasp.org/cheatsheets/Content_Security_Policy_Cheat_Sheet.html
https://developer.mozilla.org/en-US/docs/Web/HTTP/Headers/Content-Security-Policy

Classification:
CWE : CWE-693
OWASP Top 10 - 2017 : A6 - Security Misconfiguration
OWASP Top 10 - 2021 : A5 - Security Misconfiguration

Missing security header: Referrer-Policy

CONFIRMED

URL	Evidence
https://pentest-ground.com/	Response headers do not include the Referrer-Policy HTTP security header as well as the <meta> tag with name 'referrer' is not present in the response. Request / Response

Details

Risk description:
The risk is that if a user visits a web page (e.g. "http://example.com/pricing/") and clicks on a link from that page going to e.g. "https://www.google.com", the browser will send to Google the full originating URL in the `Referer` header, assuming the Referrer-Policy header is not set. The originating URL could be considered sensitive information and it could be used for user tracking.

Recommendation:
The Referrer-Policy header should be configured on the server side to avoid user tracking and inadvertent information leakage. The value `no-referrer` of this header instructs the browser to omit the Referer header entirely.

References:
https://developer.mozilla.org/en-US/docs/Web/Security/Referer_header:_privacy_and_security_concerns

Classification:
CWE : CWE-693
OWASP Top 10 - 2017 : A6 - Security Misconfiguration
OWASP Top 10 - 2021 : A5 - Security Misconfiguration

Missing security header: Strict-Transport-Security

CONFIRMED

URL	Evidence
https://pentest-ground.com/	Response headers do not include the HTTP Strict-Transport-Security header Request / Response

Details

Risk description:
The risk is that lack of this header permits an attacker to force a victim user to initiate a clear-text HTTP connection to the server, thus opening the possibility to eavesdrop on the network traffic and extract sensitive information (e.g. session cookies).

Recommendation:
The Strict-Transport-Security HTTP header should be sent with each HTTPS response. The syntax is as follows:

```
Strict-Transport-Security: max-age=<seconds>[; includeSubDomains]
```

The parameter `max-age` gives the time frame for requirement of HTTPS in seconds and should be chosen quite high, e.g. several months. A value below 7776000 is considered as too low by this scanner check.
The flag `includeSubDomains` defines that the policy applies also for sub domains of the sender of the response.

Classification:
CWE : CWE-693
OWASP Top 10 - 2017 : A6 - Security Misconfiguration
OWASP Top 10 - 2021 : A5 - Security Misconfiguration

Server software and technology found

UNCONFIRMED

Software / Version	Category
Alpine.js 3.13.0	JavaScript frameworks
Tippy.js 6	JavaScript libraries
Google Analytics	Analytics
Google Font API	Font scripts
Nginx 1.27.3	Web servers, Reverse proxies
Open Graph	Miscellaneous
Popper 2	Miscellaneous

Tailwind CSS	UI frameworks
Unpkg	CDN
Fathom	Analytics
Google Tag Manager	Tag managers
jsDelivr	CDN

Details

Risk description:
The risk is that an attacker could use this information to mount specific attacks against the identified software type and version.

Recommendation:
We recommend you to eliminate the information which permits the identification of software platform, technology, server and operating system: HTTP server headers, HTML meta information, etc.

References:
https://owasp.org/www-project-web-security-testing-guide/stable/4-Web_Application_Security_Testing/01-Information_Gathering/02-Fingerprint_Web_Server.html

Classification:
OWASP Top 10 - 2017 : A6 - Security Misconfiguration
OWASP Top 10 - 2021 : A5 - Security Misconfiguration

Security.txt file is missing

CONFIRMED

URL
Missing: https://pentest-ground.com/.well-known/security.txt

Details

Risk description:
There is no particular risk in not having a security.txt file for your server. However, this file is important because it offers a designated channel for reporting vulnerabilities and security issues.

Recommendation:
We recommend you to implement the security.txt file according to the standard, in order to allow researchers or users report any security issues they find, improving the defensive mechanisms of your server.

References:
https://securitytxt.org/

Classification:
OWASP Top 10 - 2017 : A6 - Security Misconfiguration
OWASP Top 10 - 2021 : A5 - Security Misconfiguration

Website is accessible.

Nothing was found for vulnerabilities of server-side software.

Nothing was found for client access policies.

Nothing was found for robots.txt file.

Nothing was found for use of untrusted certificates.

Nothing was found for enabled HTTP debug methods.

Nothing was found for enabled HTTP OPTIONS method.

Nothing was found for secure communication.

Nothing was found for directory listing.

Nothing was found for domain too loose set for cookies.

Nothing was found for unsafe HTTP header Content Security Policy.

Scan coverage information

List of tests performed (19/19)

- Starting the scan...
- Checking for missing HTTP header - X-Content-Type-Options...
- Checking for missing HTTP header - Content Security Policy...
- Checking for missing HTTP header - Referrer...
- Checking for missing HTTP header - Strict-Transport-Security...
- Checking for HttpOnly flag of cookie...
- Checking for Secure flag of cookie...
- Checking for website technologies...
- Checking for vulnerabilities of server-side software...
- Checking for client access policies...
- Checking for robots.txt file...
- Checking for absence of the security.txt file...
- Checking for use of untrusted certificates...
- Checking for enabled HTTP debug methods...
- Checking for enabled HTTP OPTIONS method...
- Checking for secure communication...
- Checking for directory listing...
- Checking for domain too loose set for cookies...
- Checking for unsafe HTTP header Content Security Policy...

Scan parameters

target:	https://pentest-ground.com
scan_type:	Light
authentication:	False

Scan stats

Unique Injection Points Detected:	24
URLs spidered:	6
Total number of HTTP requests:	18
Average time until a response was received:	567ms
Total number of HTTP request errors:	6

Network Vulnerability Scanner Report

✔ pentest-ground.com

The Light Network Scanner only ran limited, version-based detection. Upgrade to run Deep scans that check for 20,000+ additional vulnerabilities - with fewer False Positives

Summary

Overall risk level:
Medium

Risk ratings:

High:	0
Medium:	1
Low:	0
Info:	9

Scan information:

Start time:	Dec 28, 2024 / 14:33:59 UTC+0530
Finish time:	Dec 28, 2024 / 14:35:28 UTC+0530
Scan duration:	1 min, 29 sec
Tests performed:	10/10
Scan status:	Finished

Findings

Vulnerabilities found for jQuery 3.4.1

port 81/tcp

UNCONFIRMED

Risk level	CVSS	CVE	Summary	Exploit
	4.3	CVE-2020-11023	In jQuery versions greater than or equal to 1.0.3 and before 3.5.0, passing HTML containing <option> elements from untrusted sources - even after sanitizing it - to one of jQuery's DOM manipulation methods (i.e. .html(), .append(), and others) may execute untrusted code. This problem is patched in jQuery 3.5.0.	N/A
	4.3	CVE-2020-11022	In jQuery versions greater than or equal to 1.2 and before 3.5.0, passing HTML from untrusted sources - even after sanitizing it - to one of jQuery's DOM manipulation methods (i.e. .html(), .append(), and others) may execute untrusted code. This problem is patched in jQuery 3.5.0.	N/A

Details

Risk description:
These vulnerabilities expose the affected applications to the risk of unauthorized access to confidential data and possibly to denial of service attacks. An attacker could search for an appropriate exploit (or create one) for any of these vulnerabilities and use it to attack the system.

Notes:
- The vulnerabilities are identified based on the server's version.
- Only the first 5 vulnerabilities with the highest risk are shown for each port.

Recommendation:
We recommend you to upgrade the affected software to the latest version in order to eliminate the risks imposed by these vulnerabilities.

IP Information

CONFIRMED

IP Address	Hostname	Location	Autonomous system (AS) Information	Organization (Name & Type)
178.79.134.182	pentest-ground.com	London, England, United Kingdom	Akamai Technologies Inc (AS63949)	Linode LLC (hosting)

Details

Risk description:
If an attacker knows the physical location of an organization's IP address and its Autonomous System (AS) number, they could launch targeted physical or cyber attacks, exploiting regional vulnerabilities or disrupting critical infrastructure.

Recommendation:
We recommend reviewing physical security measures and monitoring network traffic for unusual activity, indicating potential cyber threats. Additionally, implementing robust network segmentation and adopting encryption protocols for data in transit can help protect sensitive information, even if attackers are aware of the IP addresses and the Autonomous System (AS) number.

DNS Records

CONFIRMED

Domain Queried	DNS Record Type	Description	Value
pentest-ground.com	A	IPv4 address	178.79.134.182
pentest-ground.com	NS	Name server	ns3.linode.com
pentest-ground.com	NS	Name server	ns4.linode.com
pentest-ground.com	NS	Name server	ns1.linode.com
pentest-ground.com	NS	Name server	ns2.linode.com
pentest-ground.com	NS	Name server	ns5.linode.com
pentest-ground.com	MX	Mail server	10 mail.pentest-ground.com
pentest-ground.com	SOA	Start of Authority	ns1.linode.com. admin2.admin.test. 2021000141 14400 14400 1209600 86400
pentest-ground.com	CAA	Certificate Authority Authorization	0 issue "letsencrypt.org"

Details

Risk description:
An initial step for an attacker aiming to learn about an organization involves conducting searches on its domain names to uncover DNS records associated with the organization. This strategy aims to amass comprehensive insights into the target domain, enabling the attacker to outline the organization's external digital landscape. This gathered intelligence may subsequently serve as a foundation for launching attacks, including those based on social engineering techniques. DNS records pointing to services or servers that are no longer in use can provide an attacker with an easy entry point into the network.

Recommendation:
We recommend reviewing all DNS records associated with the domain and identifying and removing unused or obsolete records.

Web redirect detected on port 80

CONFIRMED

Port 80 redirects to 443

Details

Recommendation:
Vulnerability checks are skipped for ports that redirect to another port. We recommend scanning the redirected port directly.

Open ports discovery

CONFIRMED

Port	State	Service	Product	Product Version
80	open	http	nginx	1.27.3

81	open	https	nginx	1.27.3
443	open	https	nginx	1.27.3

Details

Risk description:
This is the list of ports that have been found on the target host. Having unnecessary open ports may expose the target to more risks because those network services and applications may contain vulnerabilities.

Recommendation:
We recommend reviewing the list of open ports and closing the ones which are not necessary for business purposes.

OS Detection

UNCONFIRMED

Operating System
Linux 4.15 - 5.6

Details

Vulnerability description:
OS Detection

Server software and technologies

port 81/tcp

UNCONFIRMED

Software / Version	Category
Bootstrap	UI frameworks
Nginx 1.27.3	Web servers, Reverse proxies
Cloudflare	CDN
OWL Carousel	JavaScript libraries
jQuery 3.4.1	JavaScript libraries
Google Font API	Font scripts
cdnjs	CDN

Details

Vulnerability description:
We noticed that server software and technology details are exposed, potentially aiding attackers in tailoring specific exploits against identified systems and versions.

Risk description:
The risk is that an attacker could use this information to mount specific attacks against the identified software type and version.

Recommendation:
We recommend you to eliminate the information which permits the identification of software platform, technology, server and operating system: HTTP server headers, HTML meta information, etc.

Server software and technologies

port 443/tcp

UNCONFIRMED

Software / Version	Category
Tailwind CSS	UI frameworks
Nginx 1.27.3	Web servers, Reverse proxies

Alpine.js 3.13.0	JavaScript frameworks
Unpkg	CDN
jsDelivr	CDN
Google Tag Manager	Tag managers
Google Analytics	Analytics
Fathom	Analytics
Tippy.js 6	JavaScript libraries
Google Font API	Font scripts
Popper 2	Miscellaneous
Open Graph	Miscellaneous

Details

Vulnerability description:
We noticed that server software and technology details are exposed, potentially aiding attackers in tailoring specific exploits against identified systems and versions.

Risk description:
The risk is that an attacker could use this information to mount specific attacks against the identified software type and version.

Recommendation:
We recommend you to eliminate the information which permits the identification of software platform, technology, server and operating system: HTTP server headers, HTML meta information, etc.

Domain name servers are not vulnerable to DNS Server Zone Transfer Information Disclosure (AXFR) vulnerability

Version-based detection found no vulnerabilities for nginx 1.27.3
port 443/tcp

Scan coverage information

List of tests performed (10/10)

- ✓ Running IP information lookup phase
- ✓ Performing DNS enumeration
- ✓ Performing OS detection
- ✓ Checking for web redirect on port 80
- ✓ Running port discovery
- ✓ Attempting zone transfer against name servers...
- ✓ Fingerprinting website for technologies on port 81
- ✓ Scanning for vulnerabilities of jQuery on port 81
- ✓ Fingerprinting website for technologies on port 443
- ✓ Searching for version-based vulnerabilities on port 443

Scan parameters

Target:	pentest-ground.com
Preset:	Light
Scanning engines:	Version_based
Check alive:	True
Extensive modules:	-
Protocol type:	TCP
Ports to scan:	Top 100 ports
CVEs:	
Requests per second:	-

Port Scanner Report

✔ pentest-ground.com

The Light Port Scanner ran on a limited set of ports. Upgrade to run Deep scans and check for all 65535 ports.

Found 3 open ports (1 host)

178.79.134.182

› 178-79-134-182.ip.linodeusercontent.com

Port Number	Protocol	State	Service Name	Service Product	Service Version	Service Extra Info
80	TCP	open	http	nginx	1.27.3	
81	TCP	open	https	nginx	1.27.3	
443	TCP	open	https	nginx	1.27.3	

Scan parameters

Host:	pentest-ground.com
Protocol:	TCP
Scan type:	Light
Ports:	Top 100 ports
Check alive:	True
Detect svc version:	True
Detect OS:	False
Traceroute:	False
Scan Technique:	TCP SYN

Scan information

Start time:	Dec 28, 2024 / 14:35:46 UTC+0530
Finish time:	Dec 28, 2024 / 14:36:24 UTC+0530
Scan duration:	38 sec
Scan status:	Finished

SSL/TLS Vulnerability Scanner Report

✔ pentest-ground.com

The Light SSL/TLS Scanner only checked for port 443. Upgrade to run Deep scans against multiple SSL-enabled ports.

Summary

Overall risk level:

Info

Risk ratings:

High:	0
Medium:	0
Low:	0
Info:	19

Scan information:

Start time:	Dec 28, 2024 / 14:36:02 UTC+0530
Finish time:	Dec 28, 2024 / 14:37:43 UTC+0530
Scan duration:	1 min, 41 sec
Tests performed:	19/19
Scan status:	Finished

Findings

Found 1 open port with SSL/TLS support.

Port	State	Service	Server version	Uses SSL/TLS
443	open	https	1.27.3	Yes

SSL/TLS: Certificate is trusted

port 443/tcp

The domain has been found among Subject Alternate Names (SAN) or is the Common Name (CN) itself.
Therefore, it is considered protected by the certificate.

The Server Name Indication (SNI) has also been found. SNI is an extension to the TLS protocol that allows a client or browser to indicate which hostname it is trying to connect to at the start of the TLS handshake.
This allows the server to present multiple certificates on the same IP address and port number.

SSL/TLS: Certificate is Valid

port 443/tcp

The certificate will expire in 61 days.

SSL/TLS: CA Issuer is invalid or it cannot be identified

port 443/tcp

E6 (let's encrypt from us)

Details

Risk description:

The certificate does not have a valid Certificate Authority Issuer, which are important for checking identity of the owner. Having this risk may result in the browsers not being able to validate the server's identity, compromising the communication between the server and users.

Recommendation:

We recommend you to configure a valid Certificate Authority Issuer for your servers's certificates.

Tested for certificate issues.
port 443/tcp

Certificate number: #1
Issuer: E6 (Let's Encrypt from US)
Signature: ECDSA with SHA384
Serial number: 04287A71EA130E93E5F5EC42B855E29A381A

SSL/TLS: Not vulnerable to Heartbleed
port 443/tcp

SSL/TLS: Not vulnerable to CCS Injection
port 443/tcp

SSL/TLS: Not vulnerable to Ticketbleed
port 443/tcp

SSL/TLS: Not vulnerable to ROBOT
port 443/tcp

SSL/TLS: Not vulnerable to Secure Renegotiation
port 443/tcp

SSL/TLS: Not vulnerable to CRIME
port 443/tcp

SSL/TLS: Not vulnerable to POODLE
port 443/tcp

SSL/TLS: Not vulnerable to SWEET32
port 443/tcp

SSL/TLS: Not vulnerable to FREAK
port 443/tcp

SSL/TLS: Not vulnerable to DROWN
port 443/tcp

SSL/TLS: Not vulnerable to LOGJAM
port 443/tcp

SSL/TLS: Not vulnerable to BEAST
port 443/tcp

SSL/TLS: Not vulnerable to RC4
port 443/tcp

Tested for SSL/TLS vulnerabilities
port 443/tcp

Scan coverage information

List of tests performed (19/19)

- Checking for SSL/TLS services...
- Checking if the certificate is trusted...
- Checking if the certificate is expired...
- Checking for Certificate Authority Issuer...
- Checking the certificate on port 443...
- Scanning for HEARTBLEED on port 443
- Scanning for CCS on port 443
- Scanning for TICKETBLEED on port 443
- Scanning for ROBOT on port 443
- Scanning for SECURE_RENEGO on port 443
- Scanning for CRIME_TLS on port 443
- Scanning for POODLE_SSL on port 443
- Scanning for SWEET32 on port 443
- Scanning for FREAK on port 443
- Scanning for DROWN on port 443
- Scanning for LOGJAM on port 443
- Scanning for BEAST on port 443
- Scanning for RC4 on port 443
- Tested for SSL/TLS vulnerabilities

Scan parameters

Target:	pentest-ground.com
Preset:	Light
Scanning engines:	Certificate, Vulnerability
Ports to scan:	443

Website Recon Report

https://pentest-ground.com

Target added due to a redirect from http://pentest-ground.com

Summary

Overall risk level:

Low

Risk ratings:

High:	0
Medium:	0
Low:	1
Info:	1

Scan information:

Start time:	Dec 28, 2024 / 14:37:55 UTC+0530
Finish time:	Dec 28, 2024 / 14:38:23 UTC+0530
Scan duration:	28 sec
Tests performed:	2/2
Scan status:	Finished

Findings

Server software and technology found

UNCONFIRMED

Software / Version	Category
Alpine.js 3.13.0	JavaScript frameworks
Tippy.js 6	JavaScript libraries
Google Analytics	Analytics
Google Font API	Font scripts
Nginx 1.27.3	Web servers, Reverse proxies
Open Graph	Miscellaneous
Popper 2	Miscellaneous
Tailwind CSS	UI frameworks
Unpkg	CDN
Fathom	Analytics
Google Tag Manager	Tag managers
jsDelivr	CDN

Details

Risk description:

The risk is that an attacker could use this information to mount specific attacks against the identified software type and version.

Recommendation:

We recommend you to eliminate the information which permits the identification of software platform, technology, server and operating system: HTTP server headers, HTML meta information, etc.

References:

https://owasp.org/www-project-web-security-testing-guide/stable/4-Web_Application_Security_Testing/01-Information_Gathering/02-Fingerprint_Web_Server.html

Classification:

OWASP Top 10 - 2017 : A6 - Security Misconfiguration

OWASP Top 10 - 2021 : A5 - Security Misconfiguration

Screenshot:

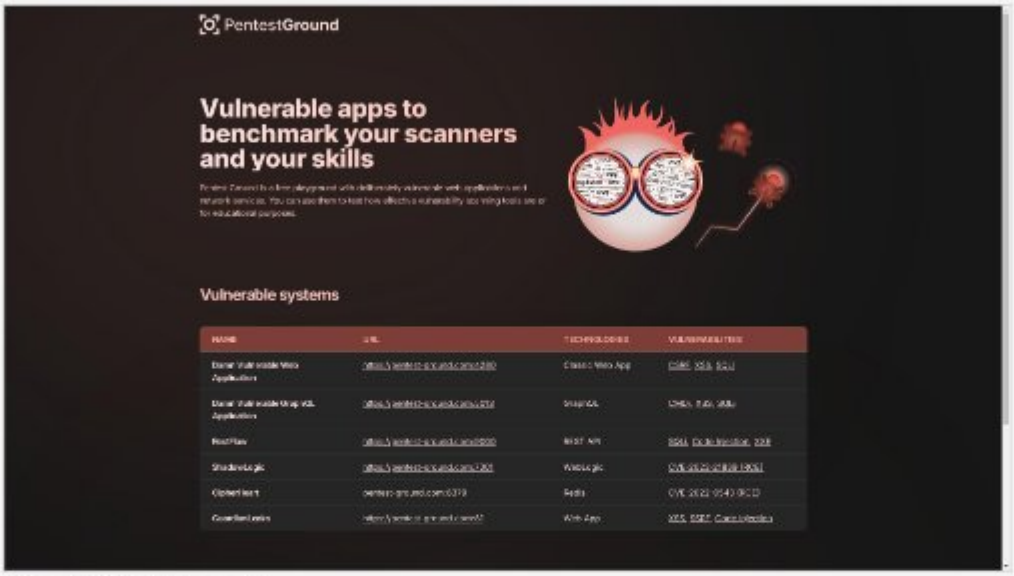

Figure 1. Website Screenshot

Website is accessible.

Scan coverage information

List of tests performed (2/2)

- ✔ Starting the scan...
- ✔ Checking for website technologies...

Scan parameters

Target:	https://pentest-ground.com

Scan stats

Total number of HTTP requests:	3
Average time until a response was received:	1495ms

1.6 Conclusion and Recommendations

1.6.1.1 Project Summary

1.6.1.1 Key Findings and Achievements

The project extensively identified and tackled critical vulnerabilities in web application security, focusing on commonly exploited weaknesses through detailed workshops and case studies. These real-world breaches illustrated the severe consequences of neglecting web application security.

- *SQL Injection (SQLi): SQL injections are one of the most notorious vulnerabilities in web security. A prime example is the 2008 Heartland Payment Systems breach, which compromised 134 million credit card records and caused damages exceeding $110 million. SQLi allows attackers to alter SQL queries to retrieve sensitive data. To prevent such attacks, this project emphasized input validation and prepared statements, vital tools in mitigating SQLi risks.*
- *Cross-Site Scripting (XSS): XSS is another widespread attack, where malicious scripts are injected into websites. The 2010 MySpace Worm was a clear example of how XSS attacks can quickly spread, impacting millions. In this project, developers were trained on output encoding and input sanitization, essential techniques to block XSS vulnerabilities.*
- *Cross-Site Request Forgery (CSRF): CSRF tricks users into performing unauthorized actions on web applications, such as changing passwords or making transactions. A notable case is the 2012 GitHub CSRF vulnerability, which enabled attackers to perform actions on behalf of authenticated users. This project taught the importance of implementing anti-CSRF tokens and managing sessions securely, minimizing the chances of these attacks.*

Overall, over 500 developers were trained during this project, and the benefits were immediately noticeable. One significant case involved a healthcare provider that detected and rectified vulnerabilities in their patient management system, avoiding a data breach that could have exposed thousands of sensitive medical records.

1.6.1.2 Importance of Web Application Security

With the rapid digital transformation across industries, web application security has become more critical than ever before. Major breaches like the Equifax Breach (2017) and the Capital One Hack (2019) demonstrate the heavy costs of insufficient web security:

- *Equifax Breach (2017): The personal data of 147 million people was exposed in this breach, resulting in billions of dollars in damages, lawsuits, and reputation loss. This breach occurred due to a failure to patch a known web vulnerability,*

underscoring the critical need for timely updates, which this project heavily emphasized.

- *Capital One Hack (2019): Misconfigurations in a web firewall led to the exposure of personal information for over 100 million customers. This event highlighted the need for rigorous security configurations and testing, all covered extensively in the project workshops.*

1.6.2. Evaluation of Objectives

1.6.2.1 Assessment of Goals

The project succeeded in its key objectives to improve security practices and awareness. Following the workshops, 85% of participants reported significant improvements in their understanding of web vulnerabilities and security strategies. One major financial institution implemented these lessons, achieving a 40% reduction in security incidents within just six months, mainly by addressing API vulnerabilities and enhancing customer data protection.

1.6.2.2 Success Stories

A notable success story came from a large e-commerce company. After attending the project's workshop, the company conducted a comprehensive security audit and uncovered a major vulnerability in its payment gateway. Identifying and patching this flaw before a high-traffic sales event saved the company from what could have been a severe data breach, proving the importance of security at every phase of development.

1.6.3. Future Work

1.6.3.1 Proposed Extensions

The project's impact opens opportunities to tackle emerging security threats that are becoming more prevalent:

- *API Security: As businesses integrate APIs to connect their services, API vulnerabilities become a growing concern. The 2018 Facebook API breach demonstrated this when over 50 million users' data was compromised. Future workshops should focus on API security testing and secure development practices to mitigate risks in this area.*
- *Supply Chain Security: The 2020 SolarWinds breach revealed the dangers of supply chain attacks, where vulnerabilities in third-party software allowed attackers access to critical systems worldwide. The use of Software Bill of*

Materials (SBOMs) and third-party assessments were identified as key strategies for addressing supply chain risks in future projects.

- *Cloud Security: With more businesses adopting cloud services, cloud misconfigurations have emerged as a top threat. The Microsoft Azure breach (2021), which exposed data due to a misconfigured cloud database, shows how these vulnerabilities can impact countless users. Future initiatives should focus on cloud-native security tools, Identity and Access Management (IAM) policies, and encryption protocols to prevent such incidents.*

Partnerships with global organizations like OWASP and IETF will strengthen the project's efforts to stay current with the latest security standards. Advanced workshops on Zero-Trust Architectures and Blockchain Security would attract more technically advanced participants, keeping them at the forefront of security trends.

1.6.4. Conclusion

1.6.4.1 Reinforcing Key Messages

Web application security is a constantly evolving field, and projects like these play a critical role in driving awareness and practices that keep businesses secure. A security-first mindset must be ingrained in development teams, from planning to deployment, ensuring robust security testing, pen testing, and regular updates to guard against emerging threats. Continuous education and a proactive approach to security are critical for keeping pace with evolving risks.

1.6.4.2 Citing Resources

This project's approach was informed by leading resources in cybersecurity, including:

- *OWASP's Top Ten: A critical tool for identifying and addressing the most significant web application risks.*
- *NIST's Cybersecurity Framework: Providing structured guidelines for reducing cybersecurity risks.*
- *Verizon's Data Breach Investigations Report (DBIR): Offering insights into prevalent attack patterns and vulnerabilities.*

2. Comprehensive Project Outline with In-Depth Explanations and Real-World Examples

2.1 Introduction to Nessus

Nessus, developed by Tenable, is one of the most robust and widely used vulnerability scanners available today, trusted by security professionals worldwide for identifying weaknesses in IT systems.

- ***History and Development:***
  Nessus was created by Renaud Deraison in 1998 as an open-source vulnerability scanner. Initially released as a free tool, it quickly gained popularity in the security community for its ability to detect vulnerabilities across a wide array of systems, including servers, network devices, and applications. In 2005, Tenable Network Security acquired Nessus, transitioning it from an open-source model to a commercial product. While the core functionality remains the same, Nessus now offers more advanced features and professional support, becoming a leading solution in vulnerability management.

Real-World Example: In 2016, Nessus played a crucial role in helping a large financial services firm identify critical vulnerabilities within their internal network before a cybersecurity attack could exploit them. This proactive scanning helped the firm avoid potential system downtimes and data breaches.

- ***Key Features and Functionalities:***
  Nessus is renowned for its deep scanning capabilities, detecting a wide range of vulnerabilities including missing patches, misconfigurations, and weak passwords. Key features include:
  - *Extensive Plugin Library: Nessus uses an ever-growing library of over 130,000 plugins that can scan for various vulnerabilities, configurations, and compliance issues.*
  - *Customizable Scan Policies: Users can define scan policies based on their specific needs, such as choosing to scan for compliance (e.g., PCI-DSS) or focus on certain critical vulnerabilities.*
  - *Scheduling and Automation: Nessus allows scans to be scheduled, automating the process of vulnerability assessments on a regular basis.*
  - *Real-Time Reporting: Detailed scan results are generated, offering insights into vulnerability severity, possible exploits, and suggested remediation steps.*
  - *Compliance Checks: Nessus can perform compliance checks for regulatory frameworks such as PCI-DSS, HIPAA, or GDPR.*

Real-World Example: A large healthcare provider used Nessus to run compliance checks against HIPAA regulations. The scans revealed outdated software versions on their servers, which could have led to non-compliance penalties.

- ***Different Versions and Licensing Options:***
  Nessus comes in multiple versions designed for different use cases, including:
    - *Nessus Essentials: A free version designed for non-commercial use, suitable for students, educators, and personal use.*
    - *Nessus Professional: The commercial version aimed at professionals, with enhanced features such as automated patch management, real-time notifications, and full technical support.*
    - *Tenable.io and Tenable.sc: These are enterprise-grade platforms designed for centralized vulnerability management. They integrate Nessus for vulnerability scanning but also offer additional features such as asset discovery, continuous monitoring, and threat intelligence integration.*

Real-World Example: A small business opted for Nessus Essentials to begin securing their network. As the company grew, they upgraded to Nessus Professional for broader scanning coverage, ensuring their systems were continuously monitored for vulnerabilities.

2.2 Installation and Setup

The installation of Nessus requires careful consideration of system requirements and network configurations. Here's a detailed guide to getting Nessus up and running:

- ***Step-by-Step Guide for Installation:***
    1. *Download: Visit the official Tenable website to download Nessus for your operating system (Windows, macOS, or Linux).*
    2. *Install:*
        - *On Windows, run the installer executable and follow the wizard to complete the setup.*
        - *On Linux (Debian/Ubuntu): Use the package manager to install the Nessus DEB package (sudo dpkg -i nessus.deb).*
        - *On macOS, mount the .dmg file and drag Nessus to your Applications folder.*
    3. *Start the Service: After installation, launch the Nessus service. This will allow you to access the web interface via https://localhost:8834.*

4. *Activate: After initial setup, enter your activation code for either the free Essentials version or a commercial version, depending on your license.*

Real-World Example: An IT consultant working for a financial services firm used this installation process to quickly deploy Nessus on a Linux-based server and started scanning their internal infrastructure.

- ***Initial Configuration and Activation:***
  After installation, it is important to configure the tool correctly to ensure successful vulnerability scans. This includes:
  - *Creating an Admin Account: When accessing Nessus for the first time, you'll be prompted to create an administrative account.*
  - *Plugin Update: Nessus will automatically fetch the latest plugin updates upon first launch, ensuring that the scanner is equipped with the most recent vulnerability definitions.*
  - *Network Setup: Ensure your system's firewall allows communication on port 8834, which is used for Nessus's web interface.*

Real-World Example: A cloud service provider deployed Nessus in their test environment and configured it to scan all their instances across different virtual private clouds (VPCs), ensuring they were protected from emerging threats.

2.3 Introduction to Metasploitable2

Metasploitable2 is a pre-configured, intentionally vulnerable virtual machine, commonly used by security professionals for penetration testing and vulnerability scanning.

- ***Overview and Purpose:***
  Metasploitable2 is a widely used penetration testing target designed to simulate vulnerabilities in a safe, controlled environment. The VM is purpose-built for security training and research, providing a rich landscape of exploitable weaknesses for students and security professionals to practice scanning and exploiting.
- ***Common Vulnerabilities in Metasploitable2:***
  Metasploitable2 contains several services and applications with well-known vulnerabilities, including:
  - *OpenSSH 4.7: Contains vulnerabilities allowing for unauthorized access with weak passwords or poorly configured keys.*
  - *ProFTPD 1.3.1: A vulnerable FTP server that can be exploited through buffer overflow attacks.*

- *MySQL 5.0.51a: Prone to SQL injection attacks, allowing unauthorized database access.*
- *Apache Tomcat: Contains flaws that can be exploited for privilege escalation.*

Real-World Example: Metasploitable2 is often used in Capture the Flag (CTF) events where participants try to exploit these vulnerabilities to "capture flags" that represent system control or sensitive information.

2.4 Setting Up Metasploitable2

Setting up Metasploitable2 involves downloading the virtual machine image and configuring it for network use.

- ***Downloading and Setting Up the VM:***
  1. *Download the VM: Obtain the Metasploitable2 image from a trusted repository such as Rapid7.*
  2. *Import the VM: Import the downloaded .ova file into virtualization software like VMware or VirtualBox.*
  3. *Configure Resources: Allocate sufficient resources (at least 1GB of RAM) and a network adapter (usually NAT or Host-Only) for the VM to function properly.*
- ***Network Configuration:***
  - *Set the VM's network adapter to Host-Only or Bridged, ensuring that Nessus can reach it.*
  - *Verify that the VM can be pinged from the host system to confirm successful connectivity.*
  - *Once set up, the Metasploitable2 machine can be accessed through its IP address for further exploitation.*

Real-World Example: An educational institution uses Metasploitable2 as part of their cybersecurity curriculum, allowing students to practice penetration testing by attacking the VM from their workstations.

2.5 Configuring Nessus for Scanning

Before running a vulnerability scan on Metasploitable2, Nessus must be configured with proper scan policies.

- ***Adding Metasploitable2 as a Target:***
    - *In the Nessus interface, navigate to the "New Scan" section and select a predefined scan policy such as "Basic Network Scan."*
    - *Enter the IP address of the Metasploitable2 VM, ensuring that it is reachable over the network.*
- ***Selecting and Customizing Policies:***
    - *Customize the scan by selecting plugins specific to the vulnerabilities you expect to find. You can also adjust port ranges to scan based on the services Metasploitable2 runs.*
    - *Enable or disable specific vulnerability checks, such as for SQL injections or buffer overflow exploits.*
- ***Scheduling Scans:***
    - *Set up automated scans to run periodically (e.g., weekly or monthly), ensuring continuous vulnerability monitoring. You can also configure alerts to notify you upon scan completion.*

Real-World Example: A corporate IT department uses scheduled scans with Nessus to regularly check for emerging vulnerabilities on their web servers running outdated software.

2.5.1 Running the Scan

Once the scan configuration is set, you can execute the vulnerability scan.

- ***Initiating the Scan:***
    - *Start the scan in Nessus and monitor its progress. The system will perform a series of checks on the Metasploitable2 VM, identifying any vulnerabilities or misconfigurations.*
    - *The scan may take several minutes depending on the complexity and the number of checks selected.*
- ***Understanding Scan Results:***
    - *Once the scan completes, Nessus will generate a report listing the vulnerabilities found, categorized by severity (Critical, High, Medium, Low).*

- *The report will include detailed descriptions of each vulnerability, suggested remediation steps, and references to CVE (Common Vulnerabilities and Exposures) IDs where applicable.*

Real-World Example: A Nessus scan revealed that a server used by a financial institution was running an outdated version of OpenSSL, which had been identified as vulnerable to the Heartbleed bug. This discovery led to a patching effort across the company's infrastructure.

MetaSploitable2

Sun, 29 Dec 2024 14:03:02 India Standard Time

TABLE OF CONTENTS

Vulnerabilities by Host

Collapse All | Expand All

192.168.150.130

CRITICAL	HIGH	MEDIUM	LOW	INFO
7	1	27	3	116

Host Information

Netbios Name:	METASPLOITABLE
IP:	192.168.150.130
OS:	Linux Kernel 2.6 on Ubuntu 8.04 (hardy)

Vulnerabilities

51988 - Bind Shell Backdoor Detection -

Synopsis

The remote host may have been compromised.

Description

A shell is listening on the remote port without any authentication being required. An attacker may use it by connecting to the remote port and sending commands directly.

Solution

Verify if the remote host has been compromised, and reinstall the system if necessary.

Risk Factor

Critical

CVSS v3.0 Base Score

9.8 (CVSS:3.0/AV:N/AC:L/PR:N/UI:N/S:U/C:H/I:H/A:H)

CVSS v2.0 Base Score

10.0 (CVSS2#AV:N/AC:L/Au:N/C:C/I:C/A:C)

Plugin Information

Published: 2011/02/15, Modified: 2022/04/11

Plugin Output

tcp/1524/wild_shell

```
Nessus was able to execute the command "id" using the
following request :


This produced the following truncated output (limited to 10 lines) :
------------------------------ snip ------------------------------
root@metasploitable:/# uid=0(root) gid=0(root) groups=0(root)
root@metasploitable:/#

------------------------------ snip ------------------------------
```

32314 - Debian OpenSSH/OpenSSL Package Random Number Generator Weakness

Synopsis

The remote SSH host keys are weak.

Description

The remote SSH host key has been generated on a Debian or Ubuntu system which contains a bug in the random number generator of its OpenSSL library.

The problem is due to a Debian packager removing nearly all sources of entropy in the remote version of OpenSSL.

An attacker can easily obtain the private part of the remote key and use this to set up decipher the remote session or set up a man in the middle attack.

See Also

http://www.nessus.org/u?107f9bdc
http://www.nessus.org/u?f14f4224

Solution

Consider all cryptographic material generated on the remote host to be guessable. In particular, all SSH, SSL and OpenVPN key material should be re-generated.

Risk Factor

Critical

CVSS v2.0 Base Score

10.0 (CVSS2#AV:N/AC:L/Au:N/C:C/I:C/A:C)

CVSS v2.0 Temporal Score

8.3 (CVSS2#E:F/RL:OF/RC:C)

References

BID	29179
CVE	CVE-2008-0166
XREF	CWE:310

Exploitable With

Core Impact (true)

Plugin Information

Published: 2008/05/14, Modified: 2024/07/24

Plugin Output

tcp/22/ssh

32321 - Debian OpenSSH/OpenSSL Package Random Number Generator Weakness (SSL check)

Synopsis

The remote SSL certificate uses a weak key.

Description

The remote x509 certificate on the remote SSL server has been generated on a Debian or Ubuntu system which contains a bug in the random number generator of its OpenSSL library.

The problem is due to a Debian packager removing nearly all sources of entropy in the remote version of OpenSSL.

An attacker can easily obtain the private part of the remote key and use this to decipher the remote session or set up a man in the middle attack.

See Also

http://www.nessus.org/u?107f9bdc
http://www.nessus.org/u?f14f4224

Solution

Consider all cryptographic material generated on the remote host to be guessable. In particuliar, all SSH, SSL and OpenVPN key material should be re-generated.

Risk Factor

Critical

CVSS v2.0 Base Score

10.0 (CVSS2#AV:N/AC:L/Au:N/C:C/I:C/A:C)

CVSS v2.0 Temporal Score

8.3 (CVSS2#E:F/RL:OF/RC:C)

References

BID	29179
CVE	CVE-2008-0166
XREF	CWE:310

Exploitable With

Core Impact (true)

Plugin Information

Published: 2008/05/15, Modified: 2020/11/16

Plugin Output

tcp/25/smtp

32321 - Debian OpenSSH/OpenSSL Package Random Number Generator Weakness (SSL check) -

Synopsis

The remote SSL certificate uses a weak key.

Description

The remote x509 certificate on the remote SSL server has been generated on a Debian or Ubuntu system which contains a bug in the random number generator of its OpenSSL library.

The problem is due to a Debian packager removing nearly all sources of entropy in the remote version of OpenSSL.

An attacker can easily obtain the private part of the remote key and use this to decipher the remote session or set up a man in the middle attack.

See Also

http://www.nessus.org/u?107f9bdc
http://www.nessus.org/u?f14f4224

Solution

Consider all cryptographic material generated on the remote host to be guessable. In particuliar, all SSH, SSL and OpenVPN key material should be re-generated.

Risk Factor

Critical

CVSS v2.0 Base Score

10.0 (CVSS2#AV:N/AC:L/Au:N/C:C/I:C/A:C)

CVSS v2.0 Temporal Score

8.3 (CVSS2#E:F/RL:OF/RC:C)

References

BID	29179
CVE	CVE-2008-0166
XREF	CWE:310

Exploitable With

Core Impact (true)

Plugin Information

Published: 2008/05/15, Modified: 2020/11/16

Plugin Output

tcp/5432/postgresql

20007 - SSL Version 2 and 3 Protocol Detection -

Synopsis

The remote service encrypts traffic using a protocol with known weaknesses.

Description

The remote service accepts connections encrypted using SSL 2.0 and/or SSL 3.0. These versions of SSL are affected by several cryptographic flaws, including:

- An insecure padding scheme with CBC ciphers.

- Insecure session renegotiation and resumption schemes.

An attacker can exploit these flaws to conduct man-in-the-middle attacks or to decrypt communications between the affected service and clients.

Although SSL/TLS has a secure means for choosing the highest supported version of the protocol (so that these versions will be used only if the client or server support nothing better), many web browsers implement this in an unsafe way that allows an attacker to downgrade a connection (such as in POODLE). Therefore, it is recommended that these protocols be disabled entirely.

NIST has determined that SSL 3.0 is no longer acceptable for secure communications. As of the date of enforcement found in PCI DSS v3.1, any version of SSL will not meet the PCI SSC's definition of 'strong cryptography'.

See Also

https://www.schneier.com/academic/paperfiles/paper-ssl.pdf
http://www.nessus.org/u?b06c7e95
http://www.nessus.org/u?247c4540
https://www.openssl.org/~bodo/ssl-poodle.pdf
http://www.nessus.org/u?5d15ba70
https://www.imperialviolet.org/2014/10/14/poodle.html
https://tools.ietf.org/html/rfc7507
https://tools.ietf.org/html/rfc7568

Solution

Consult the application's documentation to disable SSL 2.0 and 3.0.
Use TLS 1.2 (with approved cipher suites) or higher instead.

Risk Factor

Critical

CVSS v3.0 Base Score

9.8 (CVSS:3.0/AV:N/AC:L/PR:N/UI:N/S:U/C:H/I:H/A:H)

CVSS v2.0 Base Score

10.0 (CVSS2#AV:N/AC:L/Au:N/C:C/I:C/A:C)

Plugin Information

Published: 2005/10/12, Modified: 2022/04/04

Plugin Output

tcp/25/smtp

```
- SSLv2 is enabled and the server supports at least one cipher.

Low Strength Ciphers (<= 64-bit key)

Name Code KEX Auth Encryption MAC
---------------------- ---------- --- ---- --------------------- ---
EXP-RC2-CBC-MD5 RSA(512) RSA RC2-CBC(40) MD5 export
EXP-RC4-MD5 RSA(512) RSA RC4(40) MD5 export

Medium Strength Ciphers (> 64-bit and < 112-bit key, or 3DES)

Name Code KEX Auth Encryption MAC
---------------------- ---------- --- ---- --------------------- ---
DES-CBC3-MD5 RSA RSA 3DES-CBC(168) MD5

High Strength Ciphers (>= 112-bit key)

Name Code KEX Auth Encryption MAC
---------------------- ---------- --- ---- --------------------- ---
RC4-MD5 RSA RSA RC4(128) MD5

The fields above are :

{Tenable ciphername}
{Cipher ID code}
Kex={key exchange}
Auth={authentication}
Encrypt={symmetric encryption method}
MAC={message authentication code}
{export flag}

- SSLv3 is enabled and the server supports at least one cipher.
Explanation: TLS 1.0 and SSL 3.0 cipher suites may be used with SSLv3


Low Strength Ciphers (<= 64-bit key)

Name Code KEX Auth Encryption MAC
---------------------- ---------- --- ---- --------------------- ---
EXP-EDH-RSA-DES-CBC-SHA DH(512) RSA DES-CBC(40) SHA1 export
EDH-RSA-DES-CBC-SHA DH RSA DES-CBC(56) SHA1
EXP-ADH-DES-CBC-SHA DH(512) None DES-CBC(40) SHA1 export
EXP-ADH-RC4-MD5 DH(512) None RC4(40) MD5 export
ADH-DES-CBC-SHA DH None DES-CBC(56) SHA1
EXP-DES-CBC-SHA RSA(512) RSA DES-CBC(40) SHA1 export
EXP-RC2-CBC-MD5 RSA(512) RSA RC2-CBC(40) MD5 export
EXP-RC4-MD5 RSA(512) RSA RC4(40) MD5 export
DES-CBC-SHA RSA RSA DES-CBC(56) SHA1

Medium Strength Ciphers (> 64-bit and < 112-bit key, or 3DES)

Name Code KEX Auth Encryption MAC
---------------------- ---------- --- ---- --------------------- ---
EDH-RSA-DES-CBC3-SHA DH RSA 3DES-CBC(168) SHA1
ADH-DES-CBC3-SHA DH None 3DES-CBC(168) SHA1
DES-CBC3-SHA RSA RSA 3DES-CBC(168) SHA1

High Strength Ciphers (>= 112-bit key)

Name Code KEX Auth Encryption MAC
---------------------- ---------- --- ---- --------------------- ---
DHE-RSA-AES128-SHA DH RSA AES-CBC(128) SHA1
DHE-RSA-AES256-SHA DH RSA AES-CBC(256) SHA1
ADH-AES128-SHA DH None AES-CBC(128) SHA1
ADH-AES256-SHA DH None AES-CBC(256) SHA1
ADH-RC4-MD5 DH None RC4(128) MD5
AES128-SHA RSA RSA AES-CBC(128) SHA1
AES256-SHA RSA RSA AES-CBC(256) SHA1
RC4-MD5 RSA RSA RC4(128) MD5
RC4-SHA RSA RSA RC4(128) SHA1

The fields above are :

{Tenable ciphername}
{Cipher ID code}
Kex={key exchange}
Auth={authentication}
Encrypt={symmetric encryption method}
MAC={message authentication code}
{export flag}
```

20007 - SSL Version 2 and 3 Protocol Detection -

Synopsis

The remote service encrypts traffic using a protocol with known weaknesses.

Description

The remote service accepts connections encrypted using SSL 2.0 and/or SSL 3.0. These versions of SSL are affected by several cryptographic flaws, including:

- An insecure padding scheme with CBC ciphers.

- Insecure session renegotiation and resumption schemes.

An attacker can exploit these flaws to conduct man-in-the-middle attacks or to decrypt communications between the affected service and clients.

Although SSL/TLS has a secure means for choosing the highest supported version of the protocol (so that these versions will be used only if the client or server support nothing better), many web browsers implement this in an unsafe way that allows an attacker to downgrade a connection (such as in POODLE). Therefore, it is recommended that these protocols be disabled entirely.

NIST has determined that SSL 3.0 is no longer acceptable for secure communications. As of the date of enforcement found in PCI DSS v3.1, any version of SSL will not meet the PCI SSC's definition of 'strong cryptography'.

See Also

https://www.schneier.com/academic/paperfiles/paper-ssl.pdf
http://www.nessus.org/u?b06c7e95
http://www.nessus.org/u?247c4540
https://www.openssl.org/~bodo/ssl-poodle.pdf
http://www.nessus.org/u?5d15ba70
https://www.imperialviolet.org/2014/10/14/poodle.html
https://tools.ietf.org/html/rfc7507
https://tools.ietf.org/html/rfc7568

Solution

Consult the application's documentation to disable SSL 2.0 and 3.0.
Use TLS 1.2 (with approved cipher suites) or higher instead.

Risk Factor

Critical

CVSS v3.0 Base Score

9.8 (CVSS:3.0/AV:N/AC:L/PR:N/UI:N/S:U/C:H/I:H/A:H)

CVSS v2.0 Base Score

10.0 (CVSS2#AV:N/AC:L/Au:N/C:C/I:C/A:C)

Plugin Information

Published: 2005/10/12, Modified: 2022/04/04

Plugin Output

tcp/5432/postgresql

```
- SSLv3 is enabled and the server supports at least one cipher.
Explanation: TLS 1.0 and SSL 3.0 cipher suites may be used with SSLv3


Medium Strength Ciphers (> 64-bit and < 112-bit key, or 3DES)

Name Code KEX Auth Encryption MAC
---------------------- ---------- --- ---- --------------------- ---
EDH-RSA-DES-CBC3-SHA DH RSA 3DES-CBC(168) SHA1
DES-CBC3-SHA RSA RSA 3DES-CBC(168) SHA1

High Strength Ciphers (>= 112-bit key)

Name Code KEX Auth Encryption MAC
---------------------- ---------- --- ---- --------------------- ---
DHE-RSA-AES128-SHA DH RSA AES-CBC(128) SHA1
DHE-RSA-AES256-SHA DH RSA AES-CBC(256) SHA1
```

```
AES128-SHA RSA RSA AES-CBC(128) SHA1
AES256-SHA RSA RSA AES-CBC(256) SHA1
RC4-SHA RSA RSA RC4(128) SHA1

The fields above are :

{Tenable ciphername}
{Cipher ID code}
Kex={key exchange}
Auth={authentication}
Encrypt={symmetric encryption method}
MAC={message authentication code}
{export flag}
```

61708 - VNC Server 'password' Password -

Synopsis

A VNC server running on the remote host is secured with a weak password.

Description

The VNC server running on the remote host is secured with a weak password. Nessus was able to login using VNC authentication and a password of 'password'. A remote, unauthenticated attacker could exploit this to take control of the system.

Solution

Secure the VNC service with a strong password.

Risk Factor

Critical

CVSS v2.0 Base Score

10.0 (CVSS2#AV:N/AC:L/Au:N/C:C/I:C/A:C)

Plugin Information

Published: 2012/08/29, Modified: 2015/09/24

Plugin Output

tcp/5900/vnc

```
Nessus logged in using a password of "password".
```

134862 - Apache Tomcat AJP Connector Request Injection (Ghostcat) -

Synopsis

There is a vulnerable AJP connector listening on the remote host.

Description

A file read/inclusion vulnerability was found in AJP connector. A remote, unauthenticated attacker could exploit this vulnerability to read web application files from a vulnerable server. In instances where the vulnerable server allows file uploads, an attacker could upload malicious JavaServer Pages (JSP) code within a variety of file types and gain remote code execution (RCE).

See Also

http://www.nessus.org/u?8ebe6246
http://www.nessus.org/u?4e287adb
http://www.nessus.org/u?cbc3d54e
https://access.redhat.com/security/cve/CVE-2020-1745
https://access.redhat.com/solutions/4851251
http://www.nessus.org/u?dd218234
http://www.nessus.org/u?dd772531
http://www.nessus.org/u?2a01d6bf
http://www.nessus.org/u?3b5af27e
http://www.nessus.org/u?9dab109f
http://www.nessus.org/u?5eafcf70

Solution

Update the AJP configuration to require authorization and/or upgrade the Tomcat server to 7.0.100, 8.5.51, 9.0.31 or later.

Risk Factor

High

CVSS v3.0 Base Score

9.8 (CVSS:3.0/AV:N/AC:L/PR:N/UI:N/S:U/C:H/I:H/A:H)

CVSS v3.0 Temporal Score

9.4 (CVSS:3.0/E:H/RL:O/RC:C)

CVSS v2.0 Base Score

7.5 (CVSS2#AV:N/AC:L/Au:N/C:P/I:P/A:P)

CVSS v2.0 Temporal Score

6.5 (CVSS2#E:H/RL:OF/RC:C)

References

CVE	CVE-2020-1745
CVE	CVE-2020-1938
XREF	CISA-KNOWN-EXPLOITED:2022/03/17
XREF	CEA-ID:CEA-2020-0021

Plugin Information

Published: 2020/03/24, Modified: 2024/07/17

Plugin Output

tcp/8009/ajp13

```
Nessus was able to exploit the issue using the following request :

0x0000: 02 02 00 08 48 54 54 50 2F 31 2E 31 00 00 0F 2F ....HTTP/1.1.../
0x0010: 61 73 64 66 2F 78 78 78 78 78 2E 6A 73 70 00 00 asdf/xxxxx.jsp..
0x0020: 09 6C 6F 63 61 6C 68 6F 73 74 00 FF FF 00 09 6C .localhost.....l
0x0030: 6F 63 61 6C 68 6F 73 74 00 00 50 00 00 09 A0 06 ocalhost..P.....
0x0040: 00 0A 6B 65 65 70 2D 61 6C 69 76 65 00 00 0F 41 ..keep-alive...A
0x0050: 63 63 65 70 74 2D 4C 61 6E 67 75 61 67 65 00 00 ccept-Language..
0x0060: 0E 65 6E 2D 55 53 2C 65 6E 3B 71 3D 30 2E 35 00 .en-US,en;q=0.5.
0x0070: A0 08 00 01 30 00 00 0F 41 63 63 65 70 74 2D 45 ....0...Accept-E
0x0080: 6E 63 6F 64 69 6E 67 00 00 13 67 7A 69 70 2C 20 ncoding...gzip,
0x0090: 64 65 66 6C 61 74 65 2C 20 73 64 63 68 00 00 0D deflate, sdch...
0x00A0: 43 61 63 68 65 2D 43 6F 6E 74 72 6F 6C 00 00 09 Cache-Control...
0x00B0: 6D 61 78 2D 61 67 65 3D 30 00 A0 0E 00 07 4D 6F max-age=0.....Mo
0x00C0: 7A 69 6C 6C 61 00 00 19 55 70 67 72 61 64 65 2D zilla...Upgrade-
0x00D0: 49 6E 73 65 63 75 72 65 2D 52 65 71 75 65 73 74 Insecure-Request
0x00E0: 73 00 00 01 31 00 A0 01 00 09 74 65 78 74 2F 68 s...1.....text/h
0x00F0: 74 6D 6C 00 A0 0B 00 09 6C 6F 63 61 6C 68 6F 73 tml.....localhos
0x0100: 74 00 0A 00 21 6A 61 76 61 78 2E 73 65 72 76 6C t...!javax.servl
0x0110: 65 74 2E 69 6E 63 6C 75 64 65 2E 72 65 71 75 65 et.include.reque
0x0120: 73 74 5F 75 72 69 00 00 01 31 00 0A 00 1F 6A 61 st_uri...1....ja
0x0130: 76 61 78 2E 73 65 72 76 6C 65 74 2E 69 6E 63 6C vax.servlet.incl
0x0140: 75 64 65 2E 70 61 74 68 5F 69 6E 66 6F 00 00 10 ude.path_info...
0x0150: 2F 57 45 42 2D 49 4E 46 2F 77 65 62 2E 78 6D 6C /WEB-INF/web.xml
0x0160: 00 0A 00 22 6A 61 76 61 78 2E 73 65 72 76 6C 65 ..."javax.servle
0x0170: 74 2E 69 6E 63 6C 75 64 65 2E 73 65 72 76 6C 65 t.include.servle
0x0180: 74 5F 70 61 74 68 00 00 00 00 FF t_path.....


This produced the following truncated output (limited to 10 lines) :
------------------------------ snip ------------------------------
...<?xml version="1.0" encoding="ISO-8859-1"?>
<!--
Licensed to the Apache Software Foundation (ASF) under one or more
contributor license agreements. See the NOTICE file distributed with
this work for additional information regarding copyright ownership.
The ASF licenses this file to You under the Apache License, Version 2.0
(the "License"); you may not use this file except in compliance with
the License. You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0
[...]

------------------------------ snip ------------------------------
```

11213 - HTTP TRACE / TRACK Methods Allowed -

Synopsis

Debugging functions are enabled on the remote web server.

Description

The remote web server supports the TRACE and/or TRACK methods. TRACE and TRACK are HTTP methods that are used to debug web server connections.

See Also

http://www.nessus.org/u?e979b5cb
http://www.apacheweek.com/issues/03-01-24
https://download.oracle.com/sunalerts/1000718.1.html

Solution

Disable these HTTP methods. Refer to the plugin output for more information.

Risk Factor

Medium

CVSS v3.0 Base Score

5.3 (CVSS:3.0/AV:N/AC:L/PR:N/UI:N/S:U/C:L/I:N/A:N)

CVSS v3.0 Temporal Score

4.6 (CVSS:3.0/E:U/RL:O/RC:C)

CVSS v2.0 Base Score

5.0 (CVSS2#AV:N/AC:L/Au:N/C:P/I:N/A:N)

CVSS v2.0 Temporal Score

3.7 (CVSS2#E:U/RL:OF/RC:C)

References

BID	9506
BID	9561
BID	11604
BID	33374
BID	37995
CVE	CVE-2003-1567
CVE	CVE-2004-2320
CVE	CVE-2010-0386
XREF	CERT:288308
XREF	CERT:867593
XREF	CWE:16
XREF	CWE:200

Plugin Information

Published: 2003/01/23, Modified: 2024/04/09

Plugin Output

tcp/80/www

```
To disable these methods, add the following lines for each virtual
host in your configuration file :

RewriteEngine on
RewriteCond %{REQUEST_METHOD} ^(TRACE|TRACK)
RewriteRule .* - [F]

Alternatively, note that Apache versions 1.3.34, 2.0.55, and 2.2
support disabling the TRACE method natively via the 'TraceEnable'
directive.

Nessus sent the following TRACE request : \n\n------------------------------ snip ------------------------------\nTRACE
/Nessus1695723022.html HTTP/1.1
```

```
Connection: Close
Host: 192.168.150.130
Pragma: no-cache
User-Agent: Mozilla/4.0 (compatible; MSIE 8.0; Windows NT 5.1; Trident/4.0)
Accept: image/gif, image/x-xbitmap, image/jpeg, image/pjpeg, image/png, */*
Accept-Language: en
Accept-Charset: iso-8859-1,*,utf-8

------------------------------ snip ------------------------------\n\nand received the following response from the
remote server :\n\n------------------------------ snip ------------------------------\nHTTP/1.1 200 OK
Date: Sun, 29 Dec 2024 08:13:20 GMT
Server: Apache/2.2.8 (Ubuntu) DAV/2
Keep-Alive: timeout=15, max=100
Connection: Keep-Alive
Transfer-Encoding: chunked
Content-Type: message/http


TRACE /Nessus1695723022.html HTTP/1.1
Connection: Keep-Alive
Host: 192.168.150.130
Pragma: no-cache
User-Agent: Mozilla/4.0 (compatible; MSIE 8.0; Windows NT 5.1; Trident/4.0)
Accept: image/gif, image/x-xbitmap, image/jpeg, image/pjpeg, image/png, */*
Accept-Language: en
Accept-Charset: iso-8859-1,*,utf-8

------------------------------ snip ------------------------------\n
```

139915 - ISC BIND 9.x < 9.11.22, 9.12.x < 9.16.6, 9.17.x < 9.17.4 DoS -

Synopsis

The remote name server is affected by a denial of service vulnerability.

Description

According to its self-reported version number, the installation of ISC BIND running on the remote name server is version 9.x prior to 9.11.22, 9.12.x prior to 9.16.6 or 9.17.x prior to 9.17.4. It is, therefore, affected by a denial of service (DoS) vulnerability due to an assertion failure when attempting to verify a truncated response to a TSIG-signed request. An authenticated, remote attacker can exploit this issue by sending a truncated response to a TSIG-signed request to trigger an assertion failure, causing the server to exit.

Note that Nessus has not tested for this issue but has instead relied only on the application's self-reported version number.

See Also

https://kb.isc.org/docs/cve-2020-8622

Solution

Upgrade to BIND 9.11.22, 9.16.6, 9.17.4 or later.

Risk Factor

Medium

CVSS v3.0 Base Score

6.5 (CVSS:3.0/AV:N/AC:L/PR:L/UI:N/S:U/C:N/I:N/A:H)

CVSS v3.0 Temporal Score

5.7 (CVSS:3.0/E:U/RL:O/RC:C)

CVSS v2.0 Base Score

4.0 (CVSS2#AV:N/AC:L/Au:S/C:N/I:N/A:P)

CVSS v2.0 Temporal Score

3.0 (CVSS2#E:U/RL:OF/RC:C)

STIG Severity

I

References

CVE	CVE-2020-8622
XREF	IAVA:2020-A-0385-S

Plugin Information

Published: 2020/08/27, Modified: 2021/06/03

Plugin Output

udp/53/dns

```
Installed version : 9.4.2
Fixed version : 9.11.22, 9.16.6, 9.17.4 or later
```

136808 - ISC BIND Denial of Service -

Synopsis

The remote name server is affected by an assertion failure vulnerability.

Description

A denial of service (DoS) vulnerability exists in ISC BIND versions 9.11.18 / 9.11.18-S1 / 9.12.4-P2 / 9.13 / 9.14.11 / 9.15 / 9.16.2 / 9.17 / 9.17.1 and earlier. An unauthenticated, remote attacker can exploit this issue, via a specially-crafted message, to cause the service to stop responding.

Note that Nessus has not tested for this issue but has instead relied only on the application's self-reported version number.

See Also

https://kb.isc.org/docs/cve-2020-8617

Solution

Upgrade to the patched release most closely related to your current version of BIND.

Risk Factor

Medium

CVSS v3.0 Base Score

5.9 (CVSS:3.0/AV:N/AC:H/PR:N/UI:N/S:U/C:N/I:N/A:H)

CVSS v3.0 Temporal Score

5.3 (CVSS:3.0/E:P/RL:O/RC:C)

CVSS v2.0 Base Score

4.3 (CVSS2#AV:N/AC:M/Au:N/C:N/I:N/A:P)

CVSS v2.0 Temporal Score

3.4 (CVSS2#E:POC/RL:OF/RC:C)

STIG Severity

I

References

CVE	CVE-2020-8617
XREF	IAVA:2020-A-0217-S

Plugin Information

Published: 2020/05/22, Modified: 2023/03/23

Plugin Output

udp/53/dns

```
Installed version : 9.4.2
Fixed version : 9.11.19
```

136769 - ISC BIND Service Downgrade / Reflected DoS

Synopsis

The remote name server is affected by Service Downgrade / Reflected DoS vulnerabilities.

Description

According to its self-reported version, the instance of ISC BIND 9 running on the remote name server is affected by performance downgrade and Reflected DoS vulnerabilities. This is due to BIND DNS not sufficiently limiting the number fetches which may be performed while processing a referral response.

An unauthenticated, remote attacker can exploit this to cause degrade the service of the recursive server or to use the affected server as a reflector in a reflection attack.

See Also

https://kb.isc.org/docs/cve-2020-8616

Solution

Upgrade to the ISC BIND version referenced in the vendor advisory.

Risk Factor

Medium

CVSS v3.0 Base Score

8.6 (CVSS:3.0/AV:N/AC:L/PR:N/UI:N/S:C/C:N/I:N/A:H)

CVSS v3.0 Temporal Score

7.7 (CVSS:3.0/E:P/RL:O/RC:C)

CVSS v2.0 Base Score

5.0 (CVSS2#AV:N/AC:L/Au:N/C:N/I:N/A:P)

CVSS v2.0 Temporal Score

3.9 (CVSS2#E:POC/RL:OF/RC:C)

STIG Severity

I

References

CVE	CVE-2020-8616
XREF	IAVA:2020-A-0217-S

Plugin Information

Published: 2020/05/22, Modified: 2024/03/12

Plugin Output

udp/53/dns

```
Installed version : 9.4.2
Fixed version : 9.11.19
```

42256 - NFS Shares World Readable

Synopsis

The remote NFS server exports world-readable shares.

Description

The remote NFS server is exporting one or more shares without restricting access (based on hostname, IP, or IP range).

See Also

http://www.tldp.org/HOWTO/NFS-HOWTO/security.html

Solution

Place the appropriate restrictions on all NFS shares.

Risk Factor

Medium

CVSS v3.0 Base Score

7.5 (CVSS:3.0/AV:N/AC:L/PR:N/UI:N/S:U/C:H/I:N/A:N)

CVSS v2.0 Base Score

5.0 (CVSS2#AV:N/AC:L/Au:N/C:P/I:N/A:N)

Plugin Information

Published: 2009/10/26, Modified: 2024/02/21

Plugin Output

tcp/2049/rpc-nfs

```
The following shares have no access restrictions :

/ *
```

57608 - SMB Signing not required

Synopsis

Signing is not required on the remote SMB server.

Description

Signing is not required on the remote SMB server. An unauthenticated, remote attacker can exploit this to conduct man-in-the-middle attacks against the SMB server.

See Also

http://www.nessus.org/u?df39b8b3
http://technet.microsoft.com/en-us/library/cc731957.aspx
http://www.nessus.org/u?74b80723
https://www.samba.org/samba/docs/current/man-html/smb.conf.5.html
http://www.nessus.org/u?a3cac4ea

Solution

Enforce message signing in the host's configuration. On Windows, this is found in the policy setting 'Microsoft network server: Digitally sign communications (always)'. On Samba, the setting is called 'server signing'. See the 'see also' links for further details.

Risk Factor

Medium

CVSS v3.0 Base Score

5.3 (CVSS:3.0/AV:N/AC:L/PR:N/UI:N/S:U/C:N/I:L/A:N)

CVSS v3.0 Temporal Score

4.6 (CVSS:3.0/E:U/RL:O/RC:C)

CVSS v2.0 Base Score

5.0 (CVSS2#AV:N/AC:L/Au:N/C:N/I:P/A:N)

CVSS v2.0 Temporal Score

3.7 (CVSS2#E:U/RL:OF/RC:C)

Plugin Information

Published: 2012/01/19, Modified: 2022/10/05

Plugin Output

52611 - SMTP Service STARTTLS Plaintext Command Injection

Synopsis

The remote mail service allows plaintext command injection while negotiating an encrypted communications channel.

Description

The remote SMTP service contains a software flaw in its STARTTLS implementation that could allow a remote, unauthenticated attacker to inject commands during the plaintext protocol phase that will be executed during the ciphertext protocol phase.

Successful exploitation could allow an attacker to steal a victim's email or associated SASL (Simple Authentication and Security Layer) credentials.

See Also

https://tools.ietf.org/html/rfc2487
https://www.securityfocus.com/archive/1/516901/30/0/threaded

Solution

Contact the vendor to see if an update is available.

Risk Factor

Medium

CVSS v2.0 Base Score

4.0 (CVSS2#AV:N/AC:H/Au:N/C:P/I:P/A:N)

CVSS v2.0 Temporal Score

3.1 (CVSS2#E:POC/RL:OF/RC:C)

References

BID	46767
CVE	CVE-2011-0411
CVE	CVE-2011-1430
CVE	CVE-2011-1431
CVE	CVE-2011-1432
CVE	CVE-2011-1506
CVE	CVE-2011-2165
XREF	CERT:555316

Plugin Information

Published: 2011/03/10, Modified: 2019/03/06

Plugin Output

tcp/25/smtp

```
Nessus sent the following two commands in a single packet :

STARTTLS\r\nRSET\r\n

And the server sent the following two responses :

220 2.0.0 Ready to start TLS
250 2.0.0 Ok
```

51192 - SSL Certificate Cannot Be Trusted

Synopsis

The SSL certificate for this service cannot be trusted.

Description

The server's X.509 certificate cannot be trusted. This situation can occur in three different ways, in which the chain of trust can be broken, as stated below :

- First, the top of the certificate chain sent by the server might not be descended from a known public certificate authority. This can occur either when the top of the chain is an unrecognized, self-signed certificate, or when intermediate certificates are missing that would connect the top of the certificate chain to a known public certificate authority.

- Second, the certificate chain may contain a certificate that is not valid at the time of the scan. This can occur either when the scan occurs before one of the certificate's 'notBefore' dates, or after one of the certificate's 'notAfter' dates.

- Third, the certificate chain may contain a signature that either didn't match the certificate's information or could not be verified. Bad signatures can be fixed by getting the certificate with the bad signature to be re-signed by its issuer. Signatures that could not be verified are the result of the certificate's issuer using a signing algorithm that Nessus either does not support or does not recognize.

If the remote host is a public host in production, any break in the chain makes it more difficult for users to verify the authenticity and identity of the web server. This could make it easier to carry out man-in-the-middle attacks against the remote host.

See Also

https://www.itu.int/rec/T-REC-X.509/en
https://en.wikipedia.org/wiki/X.509

Solution

Purchase or generate a proper SSL certificate for this service.

Risk Factor

Medium

CVSS v3.0 Base Score

6.5 (CVSS:3.0/AV:N/AC:L/PR:N/UI:N/S:U/C:L/I:L/A:N)

CVSS v2.0 Base Score

6.4 (CVSS2#AV:N/AC:L/Au:N/C:P/I:P/A:N)

Plugin Information

Published: 2010/12/15, Modified: 2020/04/27

Plugin Output

tcp/25/smtp

```
The following certificate was part of the certificate chain
sent by the remote host, but it has expired :

|-Subject : C=XX/ST=There is no such thing outside US/L=Everywhere/O=OCOSA/OU=Office for Complication of Otherwise
Simple Affairs/CN=ubuntu804-base.localdomain/E=root@ubuntu804-base.localdomain
|-Not After : Apr 16 14:07:45 2010 GMT

The following certificate was at the top of the certificate
chain sent by the remote host, but it is signed by an unknown
certificate authority :

|-Subject : C=XX/ST=There is no such thing outside US/L=Everywhere/O=OCOSA/OU=Office for Complication of Otherwise
Simple Affairs/CN=ubuntu804-base.localdomain/E=root@ubuntu804-base.localdomain
|-Issuer : C=XX/ST=There is no such thing outside US/L=Everywhere/O=OCOSA/OU=Office for Complication of Otherwise Simple
Affairs/CN=ubuntu804-base.localdomain/E=root@ubuntu804-base.localdomain
```

51192 - SSL Certificate Cannot Be Trusted

Synopsis

The SSL certificate for this service cannot be trusted.

Description

The server's X.509 certificate cannot be trusted. This situation can occur in three different ways, in which the chain of trust can be broken, as stated below :

- First, the top of the certificate chain sent by the server might not be descended from a known public certificate authority. This can occur either when the top of the chain is an unrecognized, self-signed certificate, or when intermediate certificates are missing that would connect the top of the certificate chain to a known public certificate authority.

- Second, the certificate chain may contain a certificate that is not valid at the time of the scan. This can occur either when the scan occurs before one of the certificate's 'notBefore' dates, or after one of the certificate's 'notAfter' dates.

- Third, the certificate chain may contain a signature that either didn't match the certificate's information or could not be verified. Bad signatures can be fixed by getting the certificate with the bad signature to be re-signed by its issuer. Signatures that could not be verified are the result of the certificate's issuer using a signing algorithm that Nessus either does not support or does not recognize.

If the remote host is a public host in production, any break in the chain makes it more difficult for users to verify the authenticity and identity of the web server. This could make it easier to carry out man-in-the-middle attacks against the remote host.

See Also

https://www.itu.int/rec/T-REC-X.509/en
https://en.wikipedia.org/wiki/X.509

Solution

Purchase or generate a proper SSL certificate for this service.

Risk Factor

Medium

CVSS v3.0 Base Score

6.5 (CVSS:3.0/AV:N/AC:L/PR:N/UI:N/S:U/C:L/I:L/A:N)

CVSS v2.0 Base Score

6.4 (CVSS2#AV:N/AC:L/Au:N/C:P/I:P/A:N)

Plugin Information

Published: 2010/12/15, Modified: 2020/04/27

Plugin Output

tcp/5432/postgresql

```
The following certificate was part of the certificate chain
sent by the remote host, but it has expired :

|-Subject : C=XX/ST=There is no such thing outside US/L=Everywhere/O=OCOSA/OU=Office for Complication of Otherwise
Simple Affairs/CN=ubuntu804-base.localdomain/E=root@ubuntu804-base.localdomain
|-Not After : Apr 16 14:07:45 2010 GMT

The following certificate was at the top of the certificate
chain sent by the remote host, but it is signed by an unknown
certificate authority :

|-Subject : C=XX/ST=There is no such thing outside US/L=Everywhere/O=OCOSA/OU=Office for Complication of Otherwise
Simple Affairs/CN=ubuntu804-base.localdomain/E=root@ubuntu804-base.localdomain
|-Issuer : C=XX/ST=There is no such thing outside US/L=Everywhere/O=OCOSA/OU=Office for Complication of Otherwise Simple
Affairs/CN=ubuntu804-base.localdomain/E=root@ubuntu804-base.localdomain
```

15901 - SSL Certificate Expiry -

Synopsis

The remote server's SSL certificate has already expired.

Description

This plugin checks expiry dates of certificates associated with SSL- enabled services on the target and reports whether any have already expired.

Solution

Purchase or generate a new SSL certificate to replace the existing one.

Risk Factor

Medium

CVSS v3.0 Base Score

5.3 (CVSS:3.0/AV:N/AC:L/PR:N/UI:N/S:U/C:N/I:L/A:N)

CVSS v2.0 Base Score

5.0 (CVSS2#AV:N/AC:L/Au:N/C:N/I:P/A:N)

Plugin Information

Published: 2004/12/03, Modified: 2021/02/03

Plugin Output

tcp/25/smtp

```
The SSL certificate has already expired :

Subject : C=XX, ST=There is no such thing outside US, L=Everywhere, O=OCOSA, OU=Office for Complication of Otherwise
Simple Affairs, CN=ubuntu804-base.localdomain, emailAddress=root@ubuntu804-base.localdomain
Issuer : C=XX, ST=There is no such thing outside US, L=Everywhere, O=OCOSA, OU=Office for Complication of Otherwise
Simple Affairs, CN=ubuntu804-base.localdomain, emailAddress=root@ubuntu804-base.localdomain
Not valid before : Mar 17 14:07:45 2010 GMT
Not valid after : Apr 16 14:07:45 2010 GMT
```

15901 - SSL Certificate Expiry

Synopsis

The remote server's SSL certificate has already expired.

Description

This plugin checks expiry dates of certificates associated with SSL- enabled services on the target and reports whether any have already expired.

Solution

Purchase or generate a new SSL certificate to replace the existing one.

Risk Factor

Medium

CVSS v3.0 Base Score

5.3 (CVSS:3.0/AV:N/AC:L/PR:N/UI:N/S:U/C:N/I:L/A:N)

CVSS v2.0 Base Score

5.0 (CVSS2#AV:N/AC:L/Au:N/C:N/I:P/A:N)

Plugin Information

Published: 2004/12/03, Modified: 2021/02/03

Plugin Output

tcp/5432/postgresql

```
The SSL certificate has already expired :

Subject : C=XX, ST=There is no such thing outside US, L=Everywhere, O=OCOSA, OU=Office for Complication of Otherwise
Simple Affairs, CN=ubuntu804-base.localdomain, emailAddress=root@ubuntu804-base.localdomain
Issuer : C=XX, ST=There is no such thing outside US, L=Everywhere, O=OCOSA, OU=Office for Complication of Otherwise
Simple Affairs, CN=ubuntu804-base.localdomain, emailAddress=root@ubuntu804-base.localdomain
Not valid before : Mar 17 14:07:45 2010 GMT
Not valid after : Apr 16 14:07:45 2010 GMT
```

45411 - SSL Certificate with Wrong Hostname

Synopsis

The SSL certificate for this service is for a different host.

Description

The 'commonName' (CN) attribute of the SSL certificate presented for this service is for a different machine.

Solution

Purchase or generate a proper SSL certificate for this service.

Risk Factor

Medium

CVSS v3.0 Base Score

5.3 (CVSS:3.0/AV:N/AC:L/PR:N/UI:N/S:U/C:N/I:L/A:N)

CVSS v2.0 Base Score

5.0 (CVSS2#AV:N/AC:L/Au:N/C:N/I:P/A:N)

Plugin Information

Published: 2010/04/03, Modified: 2020/04/27

Plugin Output

tcp/25/smtp

```
The identities known by Nessus are :

192.168.150.130
192.168.150.130

The Common Name in the certificate is :

ubuntu804-base.localdomain
```

45411 - SSL Certificate with Wrong Hostname -

Synopsis

The SSL certificate for this service is for a different host.

Description

The 'commonName' (CN) attribute of the SSL certificate presented for this service is for a different machine.

Solution

Purchase or generate a proper SSL certificate for this service.

Risk Factor

Medium

CVSS v3.0 Base Score

5.3 (CVSS:3.0/AV:N/AC:L/PR:N/UI:N/S:U/C:N/I:L/A:N)

CVSS v2.0 Base Score

5.0 (CVSS2#AV:N/AC:L/Au:N/C:N/I:P/A:N)

Plugin Information

Published: 2010/04/03, Modified: 2020/04/27

Plugin Output

tcp/5432/postgresql

```
The identities known by Nessus are :

192.168.150.130
192.168.150.130

The Common Name in the certificate is :

ubuntu804-base.localdomain
```

89058 - SSL DROWN Attack Vulnerability (Decrypting RSA with Obsolete and Weakened eNcryption) -

Synopsis

The remote host may be affected by a vulnerability that allows a remote attacker to potentially decrypt captured TLS traffic.

Description

The remote host supports SSLv2 and therefore may be affected by a vulnerability that allows a cross-protocol Bleichenbacher padding oracle attack known as DROWN (Decrypting RSA with Obsolete and Weakened eNcryption). This vulnerability exists due to a flaw in the Secure Sockets Layer Version 2 (SSLv2) implementation, and it allows captured TLS traffic to be decrypted. A man-in-the-middle attacker can exploit this to decrypt the TLS connection by utilizing previously captured traffic and weak cryptography along with a series of specially crafted connections to an SSLv2 server that uses the same private key.

See Also

https://drownattack.com/
https://drownattack.com/drown-attack-paper.pdf

Solution

Disable SSLv2 and export grade cryptography cipher suites. Ensure that private keys are not used anywhere with server software that supports SSLv2 connections.

Risk Factor

Medium

CVSS v3.0 Base Score

5.9 (CVSS:3.0/AV:N/AC:H/PR:N/UI:N/S:U/C:H/I:N/A:N)

CVSS v3.0 Temporal Score

5.2 (CVSS:3.0/E:U/RL:O/RC:C)

CVSS v2.0 Base Score

4.3 (CVSS2#AV:N/AC:M/Au:N/C:P/I:N/A:N)

CVSS v2.0 Temporal Score

3.2 (CVSS2#E:U/RL:OF/RC:C)

References

BID	83733
CVE	CVE-2016-0800
XREF	CERT:583776

Plugin Information

Published: 2016/03/01, Modified: 2019/11/20

Plugin Output

tcp/25/smtp

```
The remote host is affected by SSL DROWN and supports the following
vulnerable cipher suites :

Low Strength Ciphers (<= 64-bit key)

Name Code KEX Auth Encryption MAC
---------------------- ---------- --- ---- --------------------- ---
EXP-RC2-CBC-MD5 0x04, 0x00, 0x80 RSA(512) RSA RC2-CBC(40) MD5 export
EXP-RC4-MD5 0x02, 0x00, 0x80 RSA(512) RSA RC4(40) MD5 export

High Strength Ciphers (>= 112-bit key)

Name Code KEX Auth Encryption MAC
---------------------- ---------- --- ---- --------------------- ---
RC4-MD5 0x01, 0x00, 0x80 RSA RSA RC4(128) MD5

The fields above are :

{Tenable ciphername}
{Cipher ID code}
Kex={key exchange}
Auth={authentication}
Encrypt={symmetric encryption method}
MAC={message authentication code}
{export flag}
```

42873 - SSL Medium Strength Cipher Suites Supported (SWEET32)

Synopsis

The remote service supports the use of medium strength SSL ciphers.

Description

The remote host supports the use of SSL ciphers that offer medium strength encryption. Nessus regards medium strength as any encryption that uses key lengths at least 64 bits and less than 112 bits, or else that uses the 3DES encryption suite.

Note that it is considerably easier to circumvent medium strength encryption if the attacker is on the same physical network.

See Also

https://www.openssl.org/blog/blog/2016/08/24/sweet32/
https://sweet32.info

Solution

Reconfigure the affected application if possible to avoid use of medium strength ciphers.

Risk Factor

Medium

CVSS v3.0 Base Score

7.5 (CVSS:3.0/AV:N/AC:L/PR:N/UI:N/S:U/C:H/I:N/A:N)

CVSS v2.0 Base Score

5.0 (CVSS2#AV:N/AC:L/Au:N/C:P/I:N/A:N)

References

CVE CVE-2016-2183

Plugin Information

Published: 2009/11/23, Modified: 2021/02/03

Plugin Output

tcp/25/smtp

```
Medium Strength Ciphers (> 64-bit and < 112-bit key, or 3DES)

Name Code KEX Auth Encryption MAC
---------------------- ---------- --- ---- ---------------------- ---
DES-CBC3-MD5 0x07, 0x00, 0xC0 RSA RSA 3DES-CBC(168) MD5
EDH-RSA-DES-CBC3-SHA 0x00, 0x16 DH RSA 3DES-CBC(168) SHA1
ADH-DES-CBC3-SHA 0x00, 0x1B DH None 3DES-CBC(168) SHA1
DES-CBC3-SHA 0x00, 0x0A RSA RSA 3DES-CBC(168) SHA1

The fields above are :

{Tenable ciphername}
{Cipher ID code}
Kex={key exchange}
Auth={authentication}
Encrypt={symmetric encryption method}
MAC={message authentication code}
{export flag}
```

42873 - SSL Medium Strength Cipher Suites Supported (SWEET32)

Synopsis

The remote service supports the use of medium strength SSL ciphers.

Description

The remote host supports the use of SSL ciphers that offer medium strength encryption. Nessus regards medium strength as any encryption that uses key lengths at least 64 bits and less than 112 bits, or else that uses the 3DES encryption suite.

Note that it is considerably easier to circumvent medium strength encryption if the attacker is on the same physical network.

See Also

https://www.openssl.org/blog/blog/2016/08/24/sweet32/
https://sweet32.info

Solution

Reconfigure the affected application if possible to avoid use of medium strength ciphers.

Risk Factor

Medium

CVSS v3.0 Base Score

7.5 (CVSS:3.0/AV:N/AC:L/PR:N/UI:N/S:U/C:H/I:N/A:N)

CVSS v2.0 Base Score

5.0 (CVSS2#AV:N/AC:L/Au:N/C:P/I:N/A:N)

References

CVE CVE-2016-2183

Plugin Information

Published: 2009/11/23, Modified: 2021/02/03

Plugin Output

tcp/5432/postgresql

```
Medium Strength Ciphers (> 64-bit and < 112-bit key, or 3DES)

Name Code KEX Auth Encryption MAC
---------------------- ---------- --- ---- --------------------- ---
EDH-RSA-DES-CBC3-SHA 0x00, 0x16 DH RSA 3DES-CBC(168) SHA1
DES-CBC3-SHA 0x00, 0x0A RSA RSA 3DES-CBC(168) SHA1

The fields above are :

{Tenable ciphername}
{Cipher ID code}
Kex={key exchange}
Auth={authentication}
Encrypt={symmetric encryption method}
MAC={message authentication code}
{export flag}
```

65821 - SSL RC4 Cipher Suites Supported (Bar Mitzvah)

Synopsis

The remote service supports the use of the RC4 cipher.

Description

The remote host supports the use of RC4 in one or more cipher suites.
The RC4 cipher is flawed in its generation of a pseudo-random stream of bytes so that a wide variety of small biases are introduced into the stream, decreasing its randomness.

If plaintext is repeatedly encrypted (e.g., HTTP cookies), and an attacker is able to obtain many (i.e., tens of millions) ciphertexts, the attacker may be able to derive the plaintext.

See Also

https://www.rc4nomore.com/
http://www.nessus.org/u?ac7327a0
http://cr.yp.to/talks/2013.03.12/slides.pdf
http://www.isg.rhul.ac.uk/tls/
https://www.imperva.com/docs/HII_Attacking_SSL_when_using_RC4.pdf

Solution

Reconfigure the affected application, if possible, to avoid use of RC4 ciphers. Consider using TLS 1.2 with AES-GCM suites subject to browser and web server support.

Risk Factor

Medium

CVSS v3.0 Base Score

5.9 (CVSS:3.0/AV:N/AC:H/PR:N/UI:N/S:U/C:H/I:N/A:N)

CVSS v3.0 Temporal Score

5.4 (CVSS:3.0/E:U/RL:X/RC:C)

CVSS v2.0 Base Score

4.3 (CVSS2#AV:N/AC:M/Au:N/C:P/I:N/A:N)

CVSS v2.0 Temporal Score

3.7 (CVSS2#E:U/RL:ND/RC:C)

References

BID	58796
BID	73684
CVE	CVE-2013-2566
CVE	CVE-2015-2808

Plugin Information

Published: 2013/04/05, Modified: 2021/02/03

Plugin Output

tcp/25/smtp

```
List of RC4 cipher suites supported by the remote server :

Low Strength Ciphers (<= 64-bit key)

Name Code KEX Auth Encryption MAC
---------------------- ---------- --- ---- --------------------- ---
EXP-RC4-MD5 0x02, 0x00, 0x80 RSA(512) RSA RC4(40) MD5 export
EXP-ADH-RC4-MD5 0x00, 0x17 DH(512) None RC4(40) MD5 export
EXP-RC4-MD5 0x00, 0x03 RSA(512) RSA RC4(40) MD5 export

High Strength Ciphers (>= 112-bit key)

Name Code KEX Auth Encryption MAC
---------------------- ---------- --- ---- --------------------- ---
RC4-MD5 0x01, 0x00, 0x80 RSA RSA RC4(128) MD5
ADH-RC4-MD5 0x00, 0x18 DH None RC4(128) MD5
RC4-MD5 0x00, 0x04 RSA RSA RC4(128) MD5
RC4-SHA 0x00, 0x05 RSA RSA RC4(128) SHA1

The fields above are :

{Tenable ciphername}
{Cipher ID code}
Kex={key exchange}
Auth={authentication}
Encrypt={symmetric encryption method}
MAC={message authentication code}
{export flag}
```

65821 - SSL RC4 Cipher Suites Supported (Bar Mitzvah) -

Synopsis

The remote service supports the use of the RC4 cipher.

Description

The remote host supports the use of RC4 in one or more cipher suites.

The RC4 cipher is flawed in its generation of a pseudo-random stream of bytes so that a wide variety of small biases are introduced into the stream, decreasing its randomness.

If plaintext is repeatedly encrypted (e.g., HTTP cookies), and an attacker is able to obtain many (i.e., tens of millions) ciphertexts, the attacker may be able to derive the plaintext.

See Also

https://www.rc4nomore.com/
http://www.nessus.org/u?ac7327a0
http://cr.yp.to/talks/2013.03.12/slides.pdf
http://www.isg.rhul.ac.uk/tls/
https://www.imperva.com/docs/HII_Attacking_SSL_when_using_RC4.pdf

Solution

Reconfigure the affected application, if possible, to avoid use of RC4 ciphers. Consider using TLS 1.2 with AES-GCM suites subject to browser and web server support.

Risk Factor

Medium

CVSS v3.0 Base Score

5.9 (CVSS:3.0/AV:N/AC:H/PR:N/UI:N/S:U/C:H/I:N/A:N)

CVSS v3.0 Temporal Score

5.4 (CVSS:3.0/E:U/RL:X/RC:C)

CVSS v2.0 Base Score

4.3 (CVSS2#AV:N/AC:M/Au:N/C:P/I:N/A:N)

CVSS v2.0 Temporal Score

3.7 (CVSS2#E:U/RL:ND/RC:C)

References

BID	58796
BID	73684
CVE	CVE-2013-2566
CVE	CVE-2015-2808

Plugin Information

Published: 2013/04/05, Modified: 2021/02/03

Plugin Output

tcp/5432/postgresql

```
List of RC4 cipher suites supported by the remote server :

High Strength Ciphers (>= 112-bit key)

Name Code KEX Auth Encryption MAC
---------------------- ---------- --- ---- --------------------- ---
RC4-SHA 0x00, 0x05 RSA RSA RC4(128) SHA1

The fields above are :

{Tenable ciphername}
{Cipher ID code}
Kex={key exchange}
Auth={authentication}
Encrypt={symmetric encryption method}
MAC={message authentication code}
{export flag}
```

57582 - SSL Self-Signed Certificate -

Synopsis

The SSL certificate chain for this service ends in an unrecognized self-signed certificate.

Description

The X.509 certificate chain for this service is not signed by a recognized certificate authority. If the remote host is a public host in production, this nullifies the use of SSL as anyone could establish a man-in-the-middle attack against the remote host.

Note that this plugin does not check for certificate chains that end in a certificate that is not self-signed, but is signed by an unrecognized certificate authority.

Solution

Purchase or generate a proper SSL certificate for this service.

Risk Factor

Medium

CVSS v3.0 Base Score

6.5 (CVSS:3.0/AV:N/AC:L/PR:N/UI:N/S:U/C:L/I:L/A:N)

CVSS v2.0 Base Score

6.4 (CVSS2#AV:N/AC:L/Au:N/C:P/I:P/A:N)

Plugin Information

Published: 2012/01/17, Modified: 2022/06/14

Plugin Output

tcp/25/smtp

```
The following certificate was found at the top of the certificate
chain sent by the remote host, but is self-signed and was not
found in the list of known certificate authorities :

|-Subject : C=XX/ST=There is no such thing outside US/L=Everywhere/O=OCOSA/OU=Office for Complication of Otherwise
Simple Affairs/CN=ubuntu804-base.localdomain/E=root@ubuntu804-base.localdomain
```

57582 - SSL Self-Signed Certificate

Synopsis

The SSL certificate chain for this service ends in an unrecognized self-signed certificate.

Description

The X.509 certificate chain for this service is not signed by a recognized certificate authority. If the remote host is a public host in production, this nullifies the use of SSL as anyone could establish a man-in-the-middle attack against the remote host.

Note that this plugin does not check for certificate chains that end in a certificate that is not self-signed, but is signed by an unrecognized certificate authority.

Solution

Purchase or generate a proper SSL certificate for this service.

Risk Factor

Medium

CVSS v3.0 Base Score

6.5 (CVSS:3.0/AV:N/AC:L/PR:N/UI:N/S:U/C:L/I:L/A:N)

CVSS v2.0 Base Score

6.4 (CVSS2#AV:N/AC:L/Au:N/C:P/I:P/A:N)

Plugin Information

Published: 2012/01/17, Modified: 2022/06/14

Plugin Output

tcp/5432/postgresql

```
The following certificate was found at the top of the certificate
chain sent by the remote host, but is self-signed and was not
found in the list of known certificate authorities :

|-Subject : C=XX/ST=There is no such thing outside US/L=Everywhere/O=OCOSA/OU=Office for Complication of Otherwise
Simple Affairs/CN=ubuntu804-base.localdomain/E=root@ubuntu804-base.localdomain
```

26928 - SSL Weak Cipher Suites Supported

Synopsis

The remote service supports the use of weak SSL ciphers.

Description

The remote host supports the use of SSL ciphers that offer weak encryption.

Note: This is considerably easier to exploit if the attacker is on the same physical network.

See Also

http://www.nessus.org/u?6527892d

Solution

Reconfigure the affected application, if possible to avoid the use of weak ciphers.

Risk Factor

Medium

CVSS v3.0 Base Score

5.3 (CVSS:3.0/AV:N/AC:L/PR:N/UI:N/S:U/C:L/I:N/A:N)

CVSS v2.0 Base Score

4.3 (CVSS2#AV:N/AC:M/Au:N/C:P/I:N/A:N)

References

XREF	CWE:326
XREF	CWE:327
XREF	CWE:720
XREF	CWE:753
XREF	CWE:803
XREF	CWE:928
XREF	CWE:934

Plugin Information

Published: 2007/10/08, Modified: 2021/02/03

Plugin Output

tcp/25/smtp

```
Here is the list of weak SSL ciphers supported by the remote server :

Low Strength Ciphers (<= 64-bit key)

Name Code KEX Auth Encryption MAC
---------------------- ---------- --- ---- --------------------- ---
EXP-RC2-CBC-MD5 0x04, 0x00, 0x80 RSA(512) RSA RC2-CBC(40) MD5 export
EXP-RC4-MD5 0x02, 0x00, 0x80 RSA(512) RSA RC4(40) MD5 export
EXP-EDH-RSA-DES-CBC-SHA 0x00, 0x14 DH(512) RSA DES-CBC(40) SHA1 export
EDH-RSA-DES-CBC-SHA 0x00, 0x15 DH RSA DES-CBC(56) SHA1
EXP-ADH-DES-CBC-SHA 0x00, 0x19 DH(512) None DES-CBC(40) SHA1 export
EXP-ADH-RC4-MD5 0x00, 0x17 DH(512) None RC4(40) MD5 export
ADH-DES-CBC-SHA 0x00, 0x1A DH None DES-CBC(56) SHA1
EXP-DES-CBC-SHA 0x00, 0x08 RSA(512) RSA DES-CBC(40) SHA1 export
EXP-RC2-CBC-MD5 0x00, 0x06 RSA(512) RSA RC2-CBC(40) MD5 export
EXP-RC4-MD5 0x00, 0x03 RSA(512) RSA RC4(40) MD5 export
DES-CBC-SHA 0x00, 0x09 RSA RSA DES-CBC(56) SHA1
```

```
The fields above are :

{Tenable ciphername}
{Cipher ID code}
Kex={key exchange}
Auth={authentication}
Encrypt={symmetric encryption method}
MAC={message authentication code}
{export flag}
```

81606 - SSL/TLS EXPORT_RSA <= 512-bit Cipher Suites Supported (FREAK)

Synopsis

The remote host supports a set of weak ciphers.

Description

The remote host supports EXPORT_RSA cipher suites with keys less than or equal to 512 bits. An attacker can factor a 512-bit RSA modulus in a short amount of time.

A man-in-the middle attacker may be able to downgrade the session to use EXPORT_RSA cipher suites (e.g. CVE-2015-0204). Thus, it is recommended to remove support for weak cipher suites.

See Also

https://www.smacktls.com/#freak
https://www.openssl.org/news/secadv/20150108.txt
http://www.nessus.org/u?b78da2c4

Solution

Reconfigure the service to remove support for EXPORT_RSA cipher suites.

Risk Factor

Medium

CVSS v2.0 Base Score

4.3 (CVSS2#AV:N/AC:M/Au:N/C:N/I:P/A:N)

CVSS v2.0 Temporal Score

3.2 (CVSS2#E:U/RL:OF/RC:C)

References

BID	71936
CVE	CVE-2015-0204
XREF	CERT:243585

Plugin Information

Published: 2015/03/04, Modified: 2021/02/03

Plugin Output

tcp/25/smtp

```
EXPORT_RSA cipher suites supported by the remote server :

Low Strength Ciphers (<= 64-bit key)

Name Code KEX Auth Encryption MAC
---------------------- ---------- --- ---- --------------------- ---
EXP-DES-CBC-SHA 0x00, 0x08 RSA(512) RSA DES-CBC(40) SHA1 export
EXP-RC2-CBC-MD5 0x00, 0x06 RSA(512) RSA RC2-CBC(40) MD5 export
EXP-RC4-MD5 0x00, 0x03 RSA(512) RSA RC4(40) MD5 export

The fields above are :

{Tenable ciphername}
{Cipher ID code}
Kex={key exchange}
Auth={authentication}
Encrypt={symmetric encryption method}
```

```
MAC={message authentication code}
{export flag}
```

78479 - SSLv3 Padding Oracle On Downgraded Legacy Encryption Vulnerability (POODLE) -

Synopsis

It is possible to obtain sensitive information from the remote host with SSL/TLS-enabled services.

Description

The remote host is affected by a man-in-the-middle (MitM) information disclosure vulnerability known as POODLE. The vulnerability is due to the way SSL 3.0 handles padding bytes when decrypting messages encrypted using block ciphers in cipher block chaining (CBC) mode.
MitM attackers can decrypt a selected byte of a cipher text in as few as 256 tries if they are able to force a victim application to repeatedly send the same data over newly created SSL 3.0 connections.

As long as a client and service both support SSLv3, a connection can be 'rolled back' to SSLv3, even if TLSv1 or newer is supported by the client and service.

The TLS Fallback SCSV mechanism prevents 'version rollback' attacks without impacting legacy clients; however, it can only protect connections when the client and service support the mechanism. Sites that cannot disable SSLv3 immediately should enable this mechanism.

This is a vulnerability in the SSLv3 specification, not in any particular SSL implementation. Disabling SSLv3 is the only way to completely mitigate the vulnerability.

See Also

https://www.imperialviolet.org/2014/10/14/poodle.html
https://www.openssl.org/~bodo/ssl-poodle.pdf
https://tools.ietf.org/html/draft-ietf-tls-downgrade-scsv-00

Solution

Disable SSLv3.

Services that must support SSLv3 should enable the TLS Fallback SCSV mechanism until SSLv3 can be disabled.

Risk Factor

Medium

CVSS v3.0 Base Score

3.4 (CVSS:3.0/AV:N/AC:H/PR:N/UI:R/S:C/C:L/I:N/A:N)

CVSS v3.0 Temporal Score

3.1 (CVSS:3.0/E:P/RL:O/RC:C)

CVSS v2.0 Base Score

4.3 (CVSS2#AV:N/AC:M/Au:N/C:P/I:N/A:N)

CVSS v2.0 Temporal Score

3.4 (CVSS2#E:POC/RL:OF/RC:C)

References

BID	70574
CVE	CVE-2014-3566
XREF	CERT:577193

Plugin Information

Published: 2014/10/15, Modified: 2023/06/23

Plugin Output

tcp/25/smtp

```
Nessus determined that the remote server supports SSLv3 with at least one CBC
cipher suite, indicating that this server is vulnerable.

It appears that TLSv1 or newer is supported on the server. However, the
Fallback SCSV mechanism is not supported, allowing connections to be "rolled
back" to SSLv3.
```

78479 - SSLv3 Padding Oracle On Downgraded Legacy Encryption Vulnerability (POODLE) -

Synopsis

It is possible to obtain sensitive information from the remote host with SSL/TLS-enabled services.

Description

The remote host is affected by a man-in-the-middle (MitM) information disclosure vulnerability known as POODLE. The vulnerability is due to the way SSL 3.0 handles padding bytes when decrypting messages encrypted using block ciphers in cipher block chaining (CBC) mode.
MitM attackers can decrypt a selected byte of a cipher text in as few as 256 tries if they are able to force a victim application to repeatedly send the same data over newly created SSL 3.0 connections.

As long as a client and service both support SSLv3, a connection can be 'rolled back' to SSLv3, even if TLSv1 or newer is supported by the client and service.

The TLS Fallback SCSV mechanism prevents 'version rollback' attacks without impacting legacy clients; however, it can only protect connections when the client and service support the mechanism. Sites that cannot disable SSLv3 immediately should enable this mechanism.

This is a vulnerability in the SSLv3 specification, not in any particular SSL implementation. Disabling SSLv3 is the only way to completely mitigate the vulnerability.

See Also

https://www.imperialviolet.org/2014/10/14/poodle.html
https://www.openssl.org/~bodo/ssl-poodle.pdf
https://tools.ietf.org/html/draft-ietf-tls-downgrade-scsv-00

Solution

Disable SSLv3.

Services that must support SSLv3 should enable the TLS Fallback SCSV mechanism until SSLv3 can be disabled.

Risk Factor

Medium

CVSS v3.0 Base Score

3.4 (CVSS:3.0/AV:N/AC:H/PR:N/UI:R/S:C/C:L/I:N/A:N)

CVSS v3.0 Temporal Score

3.1 (CVSS:3.0/E:P/RL:O/RC:C)

CVSS v2.0 Base Score

4.3 (CVSS2#AV:N/AC:M/Au:N/C:P/I:N/A:N)

CVSS v2.0 Temporal Score

3.4 (CVSS2#E:POC/RL:OF/RC:C)

References

BID	70574
CVE	CVE-2014-3566
XREF	CERT:577193

Plugin Information

Published: 2014/10/15, Modified: 2023/06/23

Plugin Output

tcp/5432/postgresql

```
Nessus determined that the remote server supports SSLv3 with at least one CBC
cipher suite, indicating that this server is vulnerable.

It appears that TLSv1 or newer is supported on the server. However, the
Fallback SCSV mechanism is not supported, allowing connections to be "rolled
back" to SSLv3.
```

90509 - Samba Badlock Vulnerability

Synopsis

An SMB server running on the remote host is affected by the Badlock vulnerability.

Description

The version of Samba, a CIFS/SMB server for Linux and Unix, running on the remote host is affected by a flaw, known as Badlock, that exists in the Security Account Manager (SAM) and Local Security Authority (Domain Policy) (LSAD) protocols due to improper authentication level negotiation over Remote Procedure Call (RPC) channels. A man-in-the-middle attacker who is able to able to intercept the traffic between a client and a server hosting a SAM database can exploit this flaw to force a downgrade of the authentication level, which allows the execution of arbitrary Samba network calls in the context of the intercepted user, such as viewing or modifying sensitive security data in the Active Directory (AD) database or disabling critical services.

See Also

http://badlock.org
https://www.samba.org/samba/security/CVE-2016-2118.html

Solution

Upgrade to Samba version 4.2.11 / 4.3.8 / 4.4.2 or later.

Risk Factor

Medium

CVSS v3.0 Base Score

7.5 (CVSS:3.0/AV:N/AC:H/PR:N/UI:R/S:U/C:H/I:H/A:H)

CVSS v3.0 Temporal Score

6.5 (CVSS:3.0/E:U/RL:O/RC:C)

CVSS v2.0 Base Score

6.8 (CVSS2#AV:N/AC:M/Au:N/C:P/I:P/A:P)

CVSS v2.0 Temporal Score

5.0 (CVSS2#E:U/RL:OF/RC:C)

References

BID	86002
CVE	CVE-2016-2118
XREF	CERT:813296

Plugin Information

Published: 2016/04/13, Modified: 2019/11/20

Plugin Output

tcp/445/cifs

```
Nessus detected that the Samba Badlock patch has not been applied.
```

104743 - TLS Version 1.0 Protocol Detection

Synopsis

The remote service encrypts traffic using an older version of TLS.

Description

The remote service accepts connections encrypted using TLS 1.0. TLS 1.0 has a number of cryptographic design flaws. Modern implementations of TLS 1.0 mitigate these problems, but newer versions of TLS like 1.2 and 1.3 are designed against these flaws and should be used whenever possible.

As of March 31, 2020, Endpoints that aren't enabled for TLS 1.2 and higher will no longer function properly with major web browsers and major vendors.

PCI DSS v3.2 requires that TLS 1.0 be disabled entirely by June 30, 2018, except for POS POI terminals (and the SSL/TLS termination points to which they connect) that can be verified as not being susceptible to any known exploits.

See Also

https://tools.ietf.org/html/draft-ietf-tls-oldversions-deprecate-00

Solution

Enable support for TLS 1.2 and 1.3, and disable support for TLS 1.0.

Risk Factor

Medium

CVSS v3.0 Base Score

6.5 (CVSS:3.0/AV:N/AC:H/PR:N/UI:N/S:U/C:H/I:L/A:N)

CVSS v2.0 Base Score

6.1 (CVSS2#AV:N/AC:H/Au:N/C:C/I:P/A:N)

References

XREF CWE:327

Plugin Information

Published: 2017/11/22, Modified: 2023/04/19

Plugin Output

tcp/25/smtp

```
TLSv1 is enabled and the server supports at least one cipher.
```

104743 - TLS Version 1.0 Protocol Detection -

Synopsis

The remote service encrypts traffic using an older version of TLS.

Description

The remote service accepts connections encrypted using TLS 1.0. TLS 1.0 has a number of cryptographic design flaws. Modern implementations of TLS 1.0 mitigate these problems, but newer versions of TLS like 1.2 and 1.3 are designed against these flaws and should be used whenever possible.

As of March 31, 2020, Endpoints that aren't enabled for TLS 1.2 and higher will no longer function properly with major web browsers and major vendors.

PCI DSS v3.2 requires that TLS 1.0 be disabled entirely by June 30, 2018, except for POS POI terminals (and the SSL/TLS termination points to which they connect) that can be verified as not being susceptible to any known exploits.

See Also

https://tools.ietf.org/html/draft-ietf-tls-oldversions-deprecate-00

Solution

Enable support for TLS 1.2 and 1.3, and disable support for TLS 1.0.

Risk Factor

Medium

CVSS v3.0 Base Score

6.5 (CVSS:3.0/AV:N/AC:H/PR:N/UI:N/S:U/C:H/I:L/A:N)

CVSS v2.0 Base Score

6.1 (CVSS2#AV:N/AC:H/Au:N/C:C/I:P/A:N)

References

XREF CWE:327

Plugin Information

Published: 2017/11/22, Modified: 2023/04/19

Plugin Output

tcp/5432/postgresql

```
TLSv1 is enabled and the server supports at least one cipher.
```

31705 - SSL Anonymous Cipher Suites Supported

Synopsis

The remote service supports the use of anonymous SSL ciphers.

Description

The remote host supports the use of anonymous SSL ciphers. While this enables an administrator to set up a service that encrypts traffic without having to generate and configure SSL certificates, it offers no way to verify the remote host's identity and renders the service vulnerable to a man-in-the-middle attack.

Note: This is considerably easier to exploit if the attacker is on the same physical network.

See Also

http://www.nessus.org/u?3a040ada

Solution

Reconfigure the affected application if possible to avoid use of weak ciphers.

Risk Factor

Low

CVSS v3.0 Base Score

5.9 (CVSS:3.0/AV:N/AC:H/PR:N/UI:N/S:U/C:H/I:N/A:N)

CVSS v3.0 Temporal Score

5.2 (CVSS:3.0/E:U/RL:O/RC:C)

CVSS v2.0 Base Score

2.6 (CVSS2#AV:N/AC:H/Au:N/C:P/I:N/A:N)

CVSS v2.0 Temporal Score

1.9 (CVSS2#E:U/RL:OF/RC:C)

References

BID	28482
CVE	CVE-2007-1858

Plugin Information

Published: 2008/03/28, Modified: 2023/10/27

Plugin Output

tcp/25/smtp

```
The following is a list of SSL anonymous ciphers supported by the remote TCP server :

Low Strength Ciphers (<= 64-bit key)

Name Code KEX Auth Encryption MAC
---------------------- ---------- --- ---- --------------------- ---
EXP-ADH-DES-CBC-SHA 0x00, 0x19 DH(512) None DES-CBC(40) SHA1 export
EXP-ADH-RC4-MD5 0x00, 0x17 DH(512) None RC4(40) MD5 export
ADH-DES-CBC-SHA 0x00, 0x1A DH None DES-CBC(56) SHA1

Medium Strength Ciphers (> 64-bit and < 112-bit key, or 3DES)

Name Code KEX Auth Encryption MAC
---------------------- ---------- --- ---- --------------------- ---
```

```
ADH-DES-CBC3-SHA 0x00, 0x1B DH None 3DES-CBC(168) SHA1

High Strength Ciphers (>= 112-bit key)

Name Code KEX Auth Encryption MAC
---------------------- ---------- --- ---- --------------------- ---
ADH-AES128-SHA 0x00, 0x34 DH None AES-CBC(128) SHA1
ADH-AES256-SHA 0x00, 0x3A DH None AES-CBC(256) SHA1
ADH-RC4-MD5 0x00, 0x18 DH None RC4(128) MD5

The fields above are :

{Tenable ciphername}
{Cipher ID code}
Kex={key exchange}
Auth={authentication}
Encrypt={symmetric encryption method}
MAC={message authentication code}
{export flag}
```

83738 - SSL/TLS EXPORT_DHE <= 512-bit Export Cipher Suites Supported (Logjam)

Synopsis

The remote host supports a set of weak ciphers.

Description

The remote host supports EXPORT_DHE cipher suites with keys less than or equal to 512 bits. Through cryptanalysis, a third party can find the shared secret in a short amount of time.

A man-in-the middle attacker may be able to downgrade the session to use EXPORT_DHE cipher suites. Thus, it is recommended to remove support for weak cipher suites.

See Also

https://weakdh.org/

Solution

Reconfigure the service to remove support for EXPORT_DHE cipher suites.

Risk Factor

Low

CVSS v3.0 Base Score

3.7 (CVSS:3.0/AV:N/AC:H/PR:N/UI:N/S:U/C:N/I:L/A:N)

CVSS v3.0 Temporal Score

3.2 (CVSS:3.0/E:U/RL:O/RC:C)

CVSS v2.0 Base Score

2.6 (CVSS2#AV:N/AC:H/Au:N/C:N/I:P/A:N)

CVSS v2.0 Temporal Score

2.2 (CVSS2#E:U/RL:ND/RC:C)

References

BID	74733
CVE	CVE-2015-4000
XREF	CEA-ID:CEA-2021-0004

Plugin Information

Published: 2015/05/21, Modified: 2022/12/05

Plugin Output

tcp/25/smtp

```
EXPORT_DHE cipher suites supported by the remote server :
```

```
Low Strength Ciphers (<= 64-bit key)

Name Code KEX Auth Encryption MAC
---------------------- ---------- --- ---- -------------------- ---
EXP-EDH-RSA-DES-CBC-SHA 0x00, 0x14 DH(512) RSA DES-CBC(40) SHA1 export
EXP-ADH-DES-CBC-SHA 0x00, 0x19 DH(512) None DES-CBC(40) SHA1 export
EXP-ADH-RC4-MD5 0x00, 0x17 DH(512) None RC4(40) MD5 export

The fields above are :

{Tenable ciphername}
{Cipher ID code}
Kex={key exchange}
Auth={authentication}
Encrypt={symmetric encryption method}
MAC={message authentication code}
{export flag}
```

10407 - X Server Detection

Synopsis

An X11 server is listening on the remote host

Description

The remote host is running an X11 server. X11 is a client-server protocol that can be used to display graphical applications running on a given host on a remote client.

Since the X11 traffic is not ciphered, it is possible for an attacker to eavesdrop on the connection.

Solution

Restrict access to this port. If the X11 client/server facility is not used, disable TCP support in X11 entirely (-nolisten tcp).

Risk Factor

Low

CVSS v2.0 Base Score

2.6 (CVSS2#AV:N/AC:H/Au:N/C:P/I:N/A:N)

Plugin Information

Published: 2000/05/12, Modified: 2019/03/05

Plugin Output

tcp/6000/x11

```
X11 Version : 11.0
```

21186 - AJP Connector Detection

Synopsis

There is an AJP connector listening on the remote host.

Description

The remote host is running an AJP (Apache JServ Protocol) connector, a service by which a standalone web server such as Apache communicates over TCP with a Java servlet container such as Tomcat.

See Also

http://tomcat.apache.org/connectors-doc/
http://tomcat.apache.org/connectors-doc/ajp/ajpv13a.html

Solution

n/a

Risk Factor

None

Plugin Information

Published: 2006/04/05, Modified: 2019/11/22

Plugin Output

tcp/8009/ajp13

```
The connector listing on this port supports the ajp13 protocol.
```

18261 - Apache Banner Linux Distribution Disclosure -

Synopsis

The name of the Linux distribution running on the remote host was found in the banner of the web server.

Description

Nessus was able to extract the banner of the Apache web server and determine which Linux distribution the remote host is running.

Solution

If you do not wish to display this information, edit 'httpd.conf' and set the directive 'ServerTokens Prod' and restart Apache.

Risk Factor

None

Plugin Information

Published: 2005/05/15, Modified: 2022/03/21

Plugin Output

tcp/0

```
The Linux distribution detected was :
- Ubuntu 8.04 (gutsy)
```

48204 - Apache HTTP Server Version -

Synopsis

It is possible to obtain the version number of the remote Apache HTTP server.

Description

The remote host is running the Apache HTTP Server, an open source web server. It was possible to read the version number from the banner.

See Also

https://httpd.apache.org/

Solution

n/a

Risk Factor

None

References

XREF	IAVT:0001-T-0030
XREF	IAVT:0001-T-0530

Plugin Information

Published: 2010/07/30, Modified: 2023/08/17

Plugin Output

tcp/80/www

```
URL : http://192.168.150.130/
Version : 2.2.99
Source : Server: Apache/2.2.8 (Ubuntu) DAV/2
backported : 1
modules : DAV/2
os : ConvertedUbuntu
```

84574 - Backported Security Patch Detection (PHP)

Synopsis

Security patches have been backported.

Description

Security patches may have been 'backported' to the remote PHP install without changing its version number.

Banner-based checks have been disabled to avoid false positives.

Note that this test is informational only and does not denote any security problem.

See Also

https://access.redhat.com/security/updates/backporting/?sc_cid=3093

Solution

n/a

Risk Factor

None

Plugin Information

Published: 2015/07/07, Modified: 2024/11/22

Plugin Output

tcp/80/www

```
Give Nessus credentials to perform local checks.
```

39520 - Backported Security Patch Detection (SSH)

Synopsis

Security patches are backported.

Description

Security patches may have been 'backported' to the remote SSH server without changing its version number.

Banner-based checks have been disabled to avoid false positives.

Note that this test is informational only and does not denote any security problem.

See Also

https://access.redhat.com/security/updates/backporting/?sc_cid=3093

Solution

n/a

Risk Factor

None

Plugin Information

Published: 2009/06/25, Modified: 2015/07/07

Plugin Output

tcp/22/ssh

```
Give Nessus credentials to perform local checks.
```

39521 - Backported Security Patch Detection (WWW) -

Synopsis

Security patches are backported.

Description

Security patches may have been 'backported' to the remote HTTP server without changing its version number.

Banner-based checks have been disabled to avoid false positives.

Note that this test is informational only and does not denote any security problem.

See Also

https://access.redhat.com/security/updates/backporting/?sc_cid=3093

Solution

n/a

Risk Factor

None

Plugin Information

Published: 2009/06/25, Modified: 2015/07/07

Plugin Output

tcp/80/www

```
Give Nessus credentials to perform local checks.
```

45590 - Common Platform Enumeration (CPE) -

Synopsis

It was possible to enumerate CPE names that matched on the remote system.

Description

By using information obtained from a Nessus scan, this plugin reports CPE (Common Platform Enumeration) matches for various hardware and software products found on a host.

Note that if an official CPE is not available for the product, this plugin computes the best possible CPE based on the information available from the scan.

See Also

http://cpe.mitre.org/
https://nvd.nist.gov/products/cpe

Solution

n/a

Risk Factor

None

Plugin Information

Published: 2010/04/21, Modified: 2024/11/22

Plugin Output

tcp/0

```
The remote operating system matched the following CPE :

cpe:/o:canonical:ubuntu_linux:8.04 -> Canonical Ubuntu Linux

Following application CPE's matched on the remote system :

cpe:/a:apache:http_server:2.2.8 -> Apache Software Foundation Apache HTTP Server
cpe:/a:apache:http_server:2.2.99 -> Apache Software Foundation Apache HTTP Server
cpe:/a:isc:bind:9.4. -> ISC BIND
cpe:/a:isc:bind:9.4.2 -> ISC BIND
cpe:/a:openbsd:openssh:4.7 -> OpenBSD OpenSSH
cpe:/a:openbsd:openssh:4.7p1 -> OpenBSD OpenSSH
cpe:/a:php:php:5.2.4 -> PHP PHP
cpe:/a:php:php:5.2.4-2ubuntu5.10 -> PHP PHP
cpe:/a:postgresql:postgresql -> PostgreSQL
cpe:/a:samba:samba:3.0.20 -> Samba Samba
```

10028 - DNS Server BIND version Directive Remote Version Detection -

Synopsis

It is possible to obtain the version number of the remote DNS server.

Description

The remote host is running BIND or another DNS server that reports its version number when it receives a special request for the text 'version.bind' in the domain 'chaos'.

This version is not necessarily accurate and could even be forged, as some DNS servers send the information based on a configuration file.

Solution

It is possible to hide the version number of BIND by using the 'version' directive in the 'options' section in named.conf.

Risk Factor

None

References

XREF IAVT:0001-T-0583

Plugin Information

Published: 1999/10/12, Modified: 2022/10/12

Plugin Output

udp/53/dns

```
Version : 9.4.2
```

11002 - DNS Server Detection -

Synopsis

A DNS server is listening on the remote host.

Description

The remote service is a Domain Name System (DNS) server, which provides a mapping between hostnames and IP addresses.

See Also

https://en.wikipedia.org/wiki/Domain_Name_System

Solution

Disable this service if it is not needed or restrict access to internal hosts only if the service is available externally.

Risk Factor

None

Plugin Information

Published: 2003/02/13, Modified: 2017/05/16

Plugin Output

tcp/53/dns

11002 - DNS Server Detection -

Synopsis

A DNS server is listening on the remote host.

Description

The remote service is a Domain Name System (DNS) server, which provides a mapping between hostnames and IP addresses.

See Also

https://en.wikipedia.org/wiki/Domain_Name_System

Solution

Disable this service if it is not needed or restrict access to internal hosts only if the service is available externally.

Risk Factor

None

Plugin Information

Published: 2003/02/13, Modified: 2017/05/16

Plugin Output

udp/53/dns

72779 - DNS Server Version Detection -

Synopsis

Nessus was able to obtain version information on the remote DNS server.

Description

Nessus was able to obtain version information by sending a special TXT record query to the remote host.

Note that this version is not necessarily accurate and could even be forged, as some DNS servers send the information based on a configuration file.

Solution

n/a

Risk Factor

None

References

XREF	IAVT:0001-T-0030
XREF	IAVT:0001-T-0937

Plugin Information

Published: 2014/03/03, Modified: 2024/09/24

Plugin Output

tcp/53/dns

```
DNS server answer for "version.bind" (over TCP) :

9.4.2
```

35371 - DNS Server hostname.bind Map Hostname Disclosure -

Synopsis

The DNS server discloses the remote host name.

Description

It is possible to learn the remote host name by querying the remote DNS server for 'hostname.bind' in the CHAOS domain.

Solution

It may be possible to disable this feature. Consult the vendor's documentation for more information.

Risk Factor

None

Plugin Information

Published: 2009/01/15, Modified: 2011/09/14

Plugin Output

udp/53/dns

```
The remote host name is :

metasploitable
```

54615 - Device Type -

Synopsis

It is possible to guess the remote device type.

Description

Based on the remote operating system, it is possible to determine what the remote system type is (eg: a printer, router, general-purpose computer, etc).

Solution

n/a

Risk Factor

None

Plugin Information

Published: 2011/05/23, Modified: 2022/09/09

Plugin Output

tcp/0

```
Remote device type : general-purpose
Confidence level : 95
```

10092 - FTP Server Detection -

Synopsis

An FTP server is listening on a remote port.

Description

It is possible to obtain the banner of the remote FTP server by connecting to a remote port.

Solution

n/a

Risk Factor

None

References

XREF	IAVT:0001-T-0030
XREF	IAVT:0001-T-0943

Plugin Information

Published: 1999/10/12, Modified: 2023/08/17

Plugin Output

tcp/21/ftp

```
The remote FTP banner is :

220 (vsFTPd 2.3.4)
```

10107 - HTTP Server Type and Version

Synopsis

A web server is running on the remote host.

Description

This plugin attempts to determine the type and the version of the remote web server.

Solution

n/a

Risk Factor

None

References

XREF	IAVT:0001-T-0931

Plugin Information

Published: 2000/01/04, Modified: 2020/10/30

Plugin Output

tcp/80/www

```
The remote web server type is :

Apache/2.2.8 (Ubuntu) DAV/2
```

24260 - HyperText Transfer Protocol (HTTP) Information

Synopsis

Some information about the remote HTTP configuration can be extracted.

Description

This test gives some information about the remote HTTP protocol - the version used, whether HTTP Keep-Alive is enabled, etc...

This test is informational only and does not denote any security problem.

Solution

n/a

Risk Factor

None

Plugin Information

Published: 2007/01/30, Modified: 2024/02/26

Plugin Output

tcp/80/www

```
Response Code : HTTP/1.1 200 OK

Protocol version : HTTP/1.1
HTTP/2 TLS Support: No
HTTP/2 Cleartext Support: No
SSL : no
Keep-Alive : yes
Options allowed : (Not implemented)
Headers :

Date: Sun, 29 Dec 2024 08:13:46 GMT
Server: Apache/2.2.8 (Ubuntu) DAV/2
X-Powered-By: PHP/5.2.4-2ubuntu5.10
Content-Length: 891
Keep-Alive: timeout=15, max=100
Connection: Keep-Alive
Content-Type: text/html

Response Body :

<html><head><title>Metasploitable2 - Linux</title></head><body>
<pre>


Warning: Never expose this VM to an untrusted network!

Contact: msfdev[at]metasploit.com

Login with msfadmin/msfadmin to get started


</pre>
<ul>
<li><a href="/twiki/">TWiki</a></li>
<li><a href="/phpMyAdmin/">phpMyAdmin</a></li>
<li><a href="/mutillidae/">Mutillidae</a></li>
<li><a href="/dvwa/">DVWA</a></li>
<li><a href="/dav/">WebDAV</a></li>
</ul>
</body>
</html>
```

11156 - IRC Daemon Version Detection

Synopsis

The remote host is an IRC server.

Description

This plugin determines the version of the IRC daemon.

Solution

n/a

Risk Factor

None

Plugin Information

Published: 2002/11/19, Modified: 2016/01/08

Plugin Output

tcp/6667

```
The IRC server version is : Unreal3.2.8.1. FhiXOoE [*=2309]
```

10397 - Microsoft Windows SMB LanMan Pipe Server Listing Disclosure -

Synopsis

It is possible to obtain network information.

Description

It was possible to obtain the browse list of the remote Windows system by sending a request to the LANMAN pipe. The browse list is the list of the nearest Windows systems of the remote host.

Solution

n/a

Risk Factor

None

Plugin Information

Published: 2000/05/09, Modified: 2022/02/01

Plugin Output

tcp/445/cifs

```
Here is the browse list of the remote host :

METASPLOITABLE ( os : 0.0 )
```

10785 - Microsoft Windows SMB NativeLanManager Remote System Information Disclosure -

Synopsis

It was possible to obtain information about the remote operating system.

Description

Nessus was able to obtain the remote operating system name and version (Windows and/or Samba) by sending an authentication request to port 139 or 445. Note that this plugin requires SMB to be enabled on the host.

Solution

n/a

Risk Factor

None

Plugin Information

Published: 2001/10/17, Modified: 2021/09/20

Plugin Output

tcp/445/cifs

```
The remote Operating System is : Unix
The remote native LAN manager is : Samba 3.0.20-Debian
The remote SMB Domain Name is : METASPLOITABLE
```

11011 - Microsoft Windows SMB Service Detection

Synopsis

A file / print sharing service is listening on the remote host.

Description

The remote service understands the CIFS (Common Internet File System) or Server Message Block (SMB) protocol, used to provide shared access to files, printers, etc between nodes on a network.

Solution

n/a

Risk Factor

None

Plugin Information

Published: 2002/06/05, Modified: 2021/02/11

Plugin Output

tcp/139/smb

```
An SMB server is running on this port.
```

11011 - Microsoft Windows SMB Service Detection

Synopsis

A file / print sharing service is listening on the remote host.

Description

The remote service understands the CIFS (Common Internet File System) or Server Message Block (SMB) protocol, used to provide shared access to files, printers, etc between nodes on a network.

Solution

n/a

Risk Factor

None

Plugin Information

Published: 2002/06/05, Modified: 2021/02/11

Plugin Output

tcp/445/cifs

```
A CIFS server is running on this port.
```

100871 - Microsoft Windows SMB Versions Supported (remote check)

Synopsis

It was possible to obtain information about the version of SMB running on the remote host.

Description

Nessus was able to obtain the version of SMB running on the remote host by sending an authentication request to port 139 or 445.

Note that this plugin is a remote check and does not work on agents.

Solution

n/a

Risk Factor

None

Plugin Information

Published: 2017/06/19, Modified: 2019/11/22

Plugin Output

tcp/445/cifs

```
The remote host supports the following versions of SMB :
SMBv1
```

106716 - Microsoft Windows SMB2 and SMB3 Dialects Supported (remote check)

Synopsis

It was possible to obtain information about the dialects of SMB2 and SMB3 available on the remote host.

Description

Nessus was able to obtain the set of SMB2 and SMB3 dialects running on the remote host by sending an authentication request to port 139 or 445.

Solution

n/a

Risk Factor

None

Plugin Information

Published: 2018/02/09, Modified: 2020/03/11

Plugin Output

tcp/445/cifs

```
The remote host does NOT support the following SMB dialects :
 version   introduced in windows version
2.0.2 Windows 2008
2.1 Windows 7
2.2.2 Windows 8 Beta
2.2.4 Windows 8 Beta
3.0 Windows 8
3.0.2 Windows 8.1
3.1 Windows 10
3.1.1 Windows 10
```

10437 - NFS Share Export List

Synopsis

The remote NFS server exports a list of shares.

Description

This plugin retrieves the list of NFS exported shares.

See Also

http://www.tldp.org/HOWTO/NFS-HOWTO/security.html

Solution

Ensure each share is intended to be exported.

Risk Factor

None

Plugin Information

Published: 2000/06/07, Modified: 2019/10/04

Plugin Output

tcp/2049/rpc-nfs

```
Here is the export list of 192.168.150.130 :

/ *
```

11219 - Nessus SYN scanner -

Synopsis

It is possible to determine which TCP ports are open.

Description

This plugin is a SYN 'half-open' port scanner. It shall be reasonably quick even against a firewalled target.

Note that SYN scans are less intrusive than TCP (full connect) scans against broken services, but they might cause problems for less robust firewalls and also leave unclosed connections on the remote target, if the network is loaded.

Solution

Protect your target with an IP filter.

Risk Factor

None

Plugin Information

Published: 2009/02/04, Modified: 2024/05/20

Plugin Output

tcp/21/ftp

```
Port 21/tcp was found to be open
```

11219 - Nessus SYN scanner -

Synopsis

It is possible to determine which TCP ports are open.

Description

This plugin is a SYN 'half-open' port scanner. It shall be reasonably quick even against a firewalled target.

Note that SYN scans are less intrusive than TCP (full connect) scans against broken services, but they might cause problems for less robust firewalls and also leave unclosed connections on the remote target, if the network is loaded.

Solution

Protect your target with an IP filter.

Risk Factor

None

Plugin Information

Published: 2009/02/04, Modified: 2024/05/20

Plugin Output

tcp/22/ssh

```
Port 22/tcp was found to be open
```

11219 - Nessus SYN scanner

Synopsis

It is possible to determine which TCP ports are open.

Description

This plugin is a SYN 'half-open' port scanner. It shall be reasonably quick even against a firewalled target.

Note that SYN scans are less intrusive than TCP (full connect) scans against broken services, but they might cause problems for less robust firewalls and also leave unclosed connections on the remote target, if the network is loaded.

Solution

Protect your target with an IP filter.

Risk Factor

None

Plugin Information

Published: 2009/02/04, Modified: 2024/05/20

Plugin Output

tcp/23

```
Port 23/tcp was found to be open
```

11219 - Nessus SYN scanner

Synopsis

It is possible to determine which TCP ports are open.

Description

This plugin is a SYN 'half-open' port scanner. It shall be reasonably quick even against a firewalled target.

Note that SYN scans are less intrusive than TCP (full connect) scans against broken services, but they might cause problems for less robust firewalls and also leave unclosed connections on the remote target, if the network is loaded.

Solution

Protect your target with an IP filter.

Risk Factor

None

Plugin Information

Published: 2009/02/04, Modified: 2024/05/20

Plugin Output

tcp/25/smtp

```
Port 25/tcp was found to be open
```

11219 - Nessus SYN scanner

Synopsis

It is possible to determine which TCP ports are open.

Description

This plugin is a SYN 'half-open' port scanner. It shall be reasonably quick even against a firewalled target.

Note that SYN scans are less intrusive than TCP (full connect) scans against broken services, but they might cause problems for less robust firewalls and also leave unclosed connections on the remote target, if the network is loaded.

Solution

Protect your target with an IP filter.

Risk Factor

None

Plugin Information

Published: 2009/02/04, Modified: 2024/05/20

Plugin Output

tcp/53/dns

```
Port 53/tcp was found to be open
```

11219 - Nessus SYN scanner -

Synopsis

It is possible to determine which TCP ports are open.

Description

This plugin is a SYN 'half-open' port scanner. It shall be reasonably quick even against a firewalled target.

Note that SYN scans are less intrusive than TCP (full connect) scans against broken services, but they might cause problems for less robust firewalls and also leave unclosed connections on the remote target, if the network is loaded.

Solution

Protect your target with an IP filter.

Risk Factor

None

Plugin Information

Published: 2009/02/04, Modified: 2024/05/20

Plugin Output

tcp/80/www

```
Port 80/tcp was found to be open
```

11219 - Nessus SYN scanner -

Synopsis

It is possible to determine which TCP ports are open.

Description

This plugin is a SYN 'half-open' port scanner. It shall be reasonably quick even against a firewalled target.

Note that SYN scans are less intrusive than TCP (full connect) scans against broken services, but they might cause problems for less robust firewalls and also leave unclosed connections on the remote target, if the network is loaded.

Solution

Protect your target with an IP filter.

Risk Factor

None

Plugin Information

Published: 2009/02/04, Modified: 2024/05/20

Plugin Output

tcp/111/rpc-portmapper

```
Port 111/tcp was found to be open
```

11219 - Nessus SYN scanner

Synopsis

It is possible to determine which TCP ports are open.

Description

This plugin is a SYN 'half-open' port scanner. It shall be reasonably quick even against a firewalled target.

Note that SYN scans are less intrusive than TCP (full connect) scans against broken services, but they might cause problems for less robust firewalls and also leave unclosed connections on the remote target, if the network is loaded.

Solution

Protect your target with an IP filter.

Risk Factor

None

Plugin Information

Published: 2009/02/04, Modified: 2024/05/20

Plugin Output

tcp/139/smb

```
Port 139/tcp was found to be open
```

11219 - Nessus SYN scanner

Synopsis

It is possible to determine which TCP ports are open.

Description

This plugin is a SYN 'half-open' port scanner. It shall be reasonably quick even against a firewalled target.

Note that SYN scans are less intrusive than TCP (full connect) scans against broken services, but they might cause problems for less robust firewalls and also leave unclosed connections on the remote target, if the network is loaded.

Solution

Protect your target with an IP filter.

Risk Factor

None

Plugin Information

Published: 2009/02/04, Modified: 2024/05/20

Plugin Output

tcp/445/cifs

```
Port 445/tcp was found to be open
```

11219 - Nessus SYN scanner

Synopsis

It is possible to determine which TCP ports are open.

Description

This plugin is a SYN 'half-open' port scanner. It shall be reasonably quick even against a firewalled target.

Note that SYN scans are less intrusive than TCP (full connect) scans against broken services, but they might cause problems for less robust firewalls and also leave unclosed connections on the remote target, if the network is loaded.

Solution

Protect your target with an IP filter.

Risk Factor

None

Plugin Information

Published: 2009/02/04, Modified: 2024/05/20

Plugin Output

tcp/512

```
Port 512/tcp was found to be open
```

11219 - Nessus SYN scanner

Synopsis

It is possible to determine which TCP ports are open.

Description

This plugin is a SYN 'half-open' port scanner. It shall be reasonably quick even against a firewalled target.

Note that SYN scans are less intrusive than TCP (full connect) scans against broken services, but they might cause problems for less robust firewalls and also leave unclosed connections on the remote target, if the network is loaded.

Solution

Protect your target with an IP filter.

Risk Factor

None

Plugin Information

Published: 2009/02/04, Modified: 2024/05/20

Plugin Output

tcp/513

```
Port 513/tcp was found to be open
```

11219 - Nessus SYN scanner

Synopsis

It is possible to determine which TCP ports are open.

Description

This plugin is a SYN 'half-open' port scanner. It shall be reasonably quick even against a firewalled target.

Note that SYN scans are less intrusive than TCP (full connect) scans against broken services, but they might cause problems for less robust firewalls and also leave unclosed connections on the remote target, if the network is loaded.

Solution

Protect your target with an IP filter.

Risk Factor

None

Plugin Information

Published: 2009/02/04, Modified: 2024/05/20

Plugin Output

tcp/514

```
Port 514/tcp was found to be open
```

11219 - Nessus SYN scanner

Synopsis

It is possible to determine which TCP ports are open.

Description

This plugin is a SYN 'half-open' port scanner. It shall be reasonably quick even against a firewalled target.

Note that SYN scans are less intrusive than TCP (full connect) scans against broken services, but they might cause problems for less robust firewalls and also leave unclosed connections on the remote target, if the network is loaded.

Solution

Protect your target with an IP filter.

Risk Factor

None

Plugin Information

Published: 2009/02/04, Modified: 2024/05/20

Plugin Output

tcp/1099/rmi_registry

```
Port 1099/tcp was found to be open
```

11219 - Nessus SYN scanner

Synopsis

It is possible to determine which TCP ports are open.

Description

This plugin is a SYN 'half-open' port scanner. It shall be reasonably quick even against a firewalled target.

Note that SYN scans are less intrusive than TCP (full connect) scans against broken services, but they might cause problems for less robust firewalls and also leave unclosed connections on the remote target, if the network is loaded.

Solution

Protect your target with an IP filter.

Risk Factor

None

Plugin Information

Published: 2009/02/04, Modified: 2024/05/20

Plugin Output

tcp/1524/wild_shell

```
Port 1524/tcp was found to be open
```

11219 - Nessus SYN scanner

Synopsis

It is possible to determine which TCP ports are open.

Description

This plugin is a SYN 'half-open' port scanner. It shall be reasonably quick even against a firewalled target.

Note that SYN scans are less intrusive than TCP (full connect) scans against broken services, but they might cause problems for less robust firewalls and also leave unclosed connections on the remote target, if the network is loaded.

Solution

Protect your target with an IP filter.

Risk Factor

None

Plugin Information

Published: 2009/02/04, Modified: 2024/05/20

Plugin Output

tcp/2049/rpc-nfs

```
Port 2049/tcp was found to be open
```

11219 - Nessus SYN scanner

Synopsis

It is possible to determine which TCP ports are open.

Description

This plugin is a SYN 'half-open' port scanner. It shall be reasonably quick even against a firewalled target.

Note that SYN scans are less intrusive than TCP (full connect) scans against broken services, but they might cause problems for less robust firewalls and also leave unclosed connections on the remote target, if the network is loaded.

Solution

Protect your target with an IP filter.

Risk Factor

None

Plugin Information

Published: 2009/02/04, Modified: 2024/05/20

Plugin Output

tcp/2121

```
Port 2121/tcp was found to be open
```

11219 - Nessus SYN scanner

Synopsis

It is possible to determine which TCP ports are open.

Description

This plugin is a SYN 'half-open' port scanner. It shall be reasonably quick even against a firewalled target.

Note that SYN scans are less intrusive than TCP (full connect) scans against broken services, but they might cause problems for less robust firewalls and also leave unclosed connections on the remote target, if the network is loaded.

Solution

Protect your target with an IP filter.

Risk Factor

None

Plugin Information

Published: 2009/02/04, Modified: 2024/05/20

Plugin Output

tcp/3306

```
Port 3306/tcp was found to be open
```

11219 - Nessus SYN scanner

Synopsis

It is possible to determine which TCP ports are open.

Description

This plugin is a SYN 'half-open' port scanner. It shall be reasonably quick even against a firewalled target.

Note that SYN scans are less intrusive than TCP (full connect) scans against broken services, but they might cause problems for less robust firewalls and also leave unclosed connections on the remote target, if the network is loaded.

Solution

Protect your target with an IP filter.

Risk Factor

None

Plugin Information

Published: 2009/02/04, Modified: 2024/05/20

Plugin Output

tcp/3632

```
Port 3632/tcp was found to be open
```

11219 - Nessus SYN scanner

Synopsis

It is possible to determine which TCP ports are open.

Description

This plugin is a SYN 'half-open' port scanner. It shall be reasonably quick even against a firewalled target.

Note that SYN scans are less intrusive than TCP (full connect) scans against broken services, but they might cause problems for less robust firewalls and also leave unclosed connections on the remote target, if the network is loaded.

Solution

Protect your target with an IP filter.

Risk Factor

None

Plugin Information

Published: 2009/02/04, Modified: 2024/05/20

Plugin Output

tcp/5432/postgresql

```
Port 5432/tcp was found to be open
```

11219 - Nessus SYN scanner -

Synopsis

It is possible to determine which TCP ports are open.

Description

This plugin is a SYN 'half-open' port scanner. It shall be reasonably quick even against a firewalled target.

Note that SYN scans are less intrusive than TCP (full connect) scans against broken services, but they might cause problems for less robust firewalls and also leave unclosed connections on the remote target, if the network is loaded.

Solution

Protect your target with an IP filter.

Risk Factor

None

Plugin Information

Published: 2009/02/04, Modified: 2024/05/20

Plugin Output

tcp/5900/vnc

```
Port 5900/tcp was found to be open
```

11219 - Nessus SYN scanner -

Synopsis

It is possible to determine which TCP ports are open.

Description

This plugin is a SYN 'half-open' port scanner. It shall be reasonably quick even against a firewalled target.

Note that SYN scans are less intrusive than TCP (full connect) scans against broken services, but they might cause problems for less robust firewalls and also leave unclosed connections on the remote target, if the network is loaded.

Solution

Protect your target with an IP filter.

Risk Factor

None

Plugin Information

Published: 2009/02/04, Modified: 2024/05/20

Plugin Output

tcp/6000/x11

```
Port 6000/tcp was found to be open
```

11219 - Nessus SYN scanner

Synopsis

It is possible to determine which TCP ports are open.

Description

This plugin is a SYN 'half-open' port scanner. It shall be reasonably quick even against a firewalled target.

Note that SYN scans are less intrusive than TCP (full connect) scans against broken services, but they might cause problems for less robust firewalls and also leave unclosed connections on the remote target, if the network is loaded.

Solution

Protect your target with an IP filter.

Risk Factor

None

Plugin Information

Published: 2009/02/04, Modified: 2024/05/20

Plugin Output

tcp/6667

```
Port 6667/tcp was found to be open
```

11219 - Nessus SYN scanner

Synopsis

It is possible to determine which TCP ports are open.

Description

This plugin is a SYN 'half-open' port scanner. It shall be reasonably quick even against a firewalled target.

Note that SYN scans are less intrusive than TCP (full connect) scans against broken services, but they might cause problems for less robust firewalls and also leave unclosed connections on the remote target, if the network is loaded.

Solution

Protect your target with an IP filter.

Risk Factor

None

Plugin Information

Published: 2009/02/04, Modified: 2024/05/20

Plugin Output

tcp/8009/ajp13

```
Port 8009/tcp was found to be open
```

11219 - Nessus SYN scanner

Synopsis

It is possible to determine which TCP ports are open.

Description

This plugin is a SYN 'half-open' port scanner. It shall be reasonably quick even against a firewalled target.

Note that SYN scans are less intrusive than TCP (full connect) scans against broken services, but they might cause problems for less robust firewalls and also leave unclosed connections on the remote target, if the network is loaded.

Solution

Protect your target with an IP filter.

Risk Factor

None

Plugin Information

Published: 2009/02/04, Modified: 2024/05/20

Plugin Output

tcp/8180

```
Port 8180/tcp was found to be open
```

11219 - Nessus SYN scanner

Synopsis

It is possible to determine which TCP ports are open.

Description

This plugin is a SYN 'half-open' port scanner. It shall be reasonably quick even against a firewalled target.

Note that SYN scans are less intrusive than TCP (full connect) scans against broken services, but they might cause problems for less robust firewalls and also leave unclosed connections on the remote target, if the network is loaded.

Solution

Protect your target with an IP filter.

Risk Factor

None

Plugin Information

Published: 2009/02/04, Modified: 2024/05/20

Plugin Output

tcp/8787

```
Port 8787/tcp was found to be open
```

19506 - Nessus Scan Information

Synopsis

This plugin displays information about the Nessus scan.

Description

This plugin displays, for each tested host, information about the scan itself :

- The version of the plugin set.
- The type of scanner (Nessus or Nessus Home).
- The version of the Nessus Engine.
- The port scanner(s) used.
- The port range scanned.
- The ping round trip time
- Whether credentialed or third-party patch management checks are possible.
- Whether the display of superseded patches is enabled

- The date of the scan.
- The duration of the scan.
- The number of hosts scanned in parallel.
- The number of checks done in parallel.

Solution

n/a

Risk Factor

None

Plugin Information

Published: 2005/08/26, Modified: 2024/10/04

Plugin Output

tcp/0

```
Information about this scan :

Nessus version : 10.8.3
Nessus build : 20010
Plugin feed version : 202412281438
Scanner edition used : Nessus Home
Scanner OS : WINDOWS
Scanner distribution : win-x86-64
Scan type : Normal
Scan name : MetaSploitable2
Scan policy used : Basic Network Scan
Scanner IP : 192.168.62.130
Port scanner(s) : nessus syn scanner
Port range : default
Ping RTT : 4.833 ms
Thorough tests : no
Experimental tests : no
Scan for Unpatched Vulnerabilities : no
Plugin debugging enabled : no
Paranoia level : 1
Report verbosity : 1
Safe checks : yes
Optimize the test : no
Credentialed checks : no
Patch management checks : None
Display superseded patches : yes (supersedence plugin did not launch)
CGI scanning : disabled
Web application tests : disabled
Max hosts : 30
Max checks : 4
Recv timeout : 5
Backports : Detected
Allow post-scan editing : Yes
Nessus Plugin Signature Checking : Enabled
Audit File Signature Checking : Disabled
Scan Start Date : 2024/12/29 13:34 India Standard Time
Scan duration : 1729 sec
Scan for malware : no
```

11936 - OS Identification -

Synopsis

It is possible to guess the remote operating system.

Description

Using a combination of remote probes (e.g., TCP/IP, SMB, HTTP, NTP, SNMP, etc.), it is possible to guess the name of the remote operating system in use. It is also possible sometimes to guess the version of the operating system.

Solution

n/a

Risk Factor

None

Plugin Information

Published: 2003/12/09, Modified: 2024/10/14

Plugin Output

tcp/0

```
Remote operating system : Linux Kernel 2.6 on Ubuntu 8.04 (gutsy)
Confidence level : 95
Method : HTTP

Not all fingerprints could give a match. If you think that these
signatures would help us improve OS fingerprinting, please submit
them by visiting https://www.tenable.com/research/submitsignatures.

SSH:SSH-2.0-OpenSSH_4.7p1 Debian-8ubuntu1
SinFP:
P1:B11013:F0x12:W64240:O0204ffff:M1460:
P2:B11013:F0x12:W64240:O0204ffff:M1460:
P3:B00000:F0x00:W0:O0:M0
P4:191003_7_p=2121R
SMTP:!:220 metasploitable.localdomain ESMTP Postfix (Ubuntu)
SSLcert:!:i/CN:ubuntu804-base.localdomaini/O:OCOSAi/OU:Office for Complication of Otherwise Simple
Affairss/CN:ubuntu804-base.localdomains/O:OCOSAs/OU:Office for Complication of Otherwise Simple Affairs
ed093088706603bfd5dc237399b498da2d4d31c6
i/CN:ubuntu804-base.localdomaini/O:OCOSAi/OU:Office for Complication of Otherwise Simple Affairss/CN:ubuntu804-
base.localdomains/O:OCOSAs/OU:Office for Complication of Otherwise Simple Affairs
ed093088706603bfd5dc237399b498da2d4d31c6



The remote host is running Linux Kernel 2.6 on Ubuntu 8.04 (gutsy)
```

117886 - OS Security Patch Assessment Not Available -

Synopsis

OS Security Patch Assessment is not available.

Description

OS Security Patch Assessment is not available on the remote host.
This does not necessarily indicate a problem with the scan.
Credentials may not have been provided, OS security patch assessment may not be supported for the target, the target may not have been identified, or another issue may have occurred that prevented OS security patch assessment from being available. See plugin output for details.

This plugin reports non-failure information impacting the availability of OS Security Patch Assessment. Failure information is reported by plugin 21745 : 'OS Security Patch Assessment failed'. If a target host is not supported for OS Security Patch Assessment, plugin 110695 : 'OS Security Patch Assessment Checks Not Supported' will report concurrently with this plugin.

Solution

n/a

Risk Factor

None

References

XREF IAVB:0001-B-0515

Plugin Information

Published: 2018/10/02, Modified: 2021/07/12

Plugin Output

tcp/0

```
The following issues were reported :

- Plugin : no_local_checks_credentials.nasl
Message :
Credentials were not provided for detected SSH service.
```

181418 - OpenSSH Detection

Synopsis

An OpenSSH-based SSH server was detected on the remote host.

Description

An OpenSSH-based SSH server was detected on the remote host.

See Also

https://www.openssh.com/

Solution

n/a

Risk Factor

None

Plugin Information

Published: 2023/09/14, Modified: 2024/12/18

Plugin Output

tcp/22/ssh

```
Service : ssh
Version : 4.7p1
Banner : SSH-2.0-OpenSSH_4.7p1 Debian-8ubuntu1
```

50845 - OpenSSL Detection

Synopsis

The remote service appears to use OpenSSL to encrypt traffic.

Description

Based on its response to a TLS request with a specially crafted server name extension, it seems that the remote service is using the OpenSSL library to encrypt traffic.

Note that this plugin can only detect OpenSSL implementations that have enabled support for TLS extensions (RFC 4366).

See Also

https://www.openssl.org/

Solution

n/a

Risk Factor

None

Plugin Information

Published: 2010/11/30, Modified: 2020/06/12

Plugin Output

tcp/25/smtp

50845 - OpenSSL Detection

Synopsis

The remote service appears to use OpenSSL to encrypt traffic.

Description

Based on its response to a TLS request with a specially crafted server name extension, it seems that the remote service is using the OpenSSL library to encrypt traffic.

Note that this plugin can only detect OpenSSL implementations that have enabled support for TLS extensions (RFC 4366).

See Also

https://www.openssl.org/

Solution

n/a

Risk Factor

None

Plugin Information

Published: 2010/11/30, Modified: 2020/06/12

Plugin Output

tcp/5432/postgresql

48243 - PHP Version Detection

Synopsis

It was possible to obtain the version number of the remote PHP installation.

Description

Nessus was able to determine the version of PHP available on the remote web server.

Solution

n/a

Risk Factor

None

References

XREF IAVT:0001-T-0936

Plugin Information

Published: 2010/08/04, Modified: 2024/11/22

Plugin Output

tcp/80/www

```
Nessus was able to identify the following PHP version information :

Version : 5.2.4-2ubuntu5.10
Source : X-Powered-By: PHP/5.2.4-2ubuntu5.10
```

66334 - Patch Report

Synopsis

The remote host is missing several patches.

Description

The remote host is missing one or more security patches. This plugin lists the newest version of each patch to install to make sure the remote host is up-to-date.

Note: Because the 'Show missing patches that have been superseded' setting in your scan policy depends on this plugin, it will always run and cannot be disabled.

Solution

Install the patches listed below.

Risk Factor

None

Plugin Information

Published: 2013/07/08, Modified: 2024/12/10

Plugin Output

tcp/0

```
. You need to take the following 2 actions :

[ ISC BIND 9.x < 9.11.22, 9.12.x < 9.16.6, 9.17.x < 9.17.4 DoS (139915) ]

+ Action to take : Upgrade to BIND 9.11.22, 9.16.6, 9.17.4 or later.

+Impact : Taking this action will resolve 3 different vulnerabilities (CVEs).


[ Samba Badlock Vulnerability (90509) ]

+ Action to take : Upgrade to Samba version 4.2.11 / 4.3.8 / 4.4.2 or later.
```

118224 - PostgreSQL STARTTLS Support -

Synopsis

The remote service supports encrypting traffic.

Description

The remote PostgreSQL server supports the use of encryption initiated during pre-login to switch from a cleartext to an encrypted communications channel.

See Also

https://www.postgresql.org/docs/9.2/protocol-flow.html#AEN96066
https://www.postgresql.org/docs/9.2/protocol-message-formats.html

Solution

n/a

Risk Factor

None

Plugin Information

Published: 2018/10/19, Modified: 2022/04/11

Plugin Output

tcp/5432/postgresql

```
Here is the PostgreSQL's SSL certificate that Nessus
was able to collect after sending a pre-login packet :

------------------------------ snip ------------------------------
Subject Name:

Country: XX
State/Province: There is no such thing outside US
Locality: Everywhere
Organization: OCOSA
Organization Unit: Office for Complication of Otherwise Simple Affairs
Common Name: ubuntu804-base.localdomain
Email Address: root@ubuntu804-base.localdomain

Issuer Name:
```

```
Country: XX
State/Province: There is no such thing outside US
Locality: Everywhere
Organization: OCOSA
Organization Unit: Office for Complication of Otherwise Simple Affairs
Common Name: ubuntu804-base.localdomain
Email Address: root@ubuntu804-base.localdomain

Serial Number: 00 FA F9 3A 4C 7F B6 B9 CC

Version: 1

Signature Algorithm: SHA-1 With RSA Encryption

Not Valid Before: Mar 17 14:07:45 2010 GMT
Not Valid After: Apr 16 14:07:45 2010 GMT

Public Key Info:

Algorithm: RSA Encryption
Key Length: 1024 bits
Public Key: 00 D6 B4 13 36 33 9A 95 71 7B 1B DE 7C 83 75 DA 71 B1 3C A9
7F FE AD 64 1B 77 E9 4F AE BE CA D4 F8 CB EF AE BB 43 79 24
73 FF 3C E5 9E 3B 6D FC C8 B1 AC FA 4C 4D 5E 9B 4C 99 54 0B
D7 A8 4A 50 BA A9 DE 1D 1F F4 E4 6B 02 A3 F4 6B 45 CD 4C AF
8D 89 62 33 8F 65 BB 36 61 9F C4 2C 73 C1 4E 2E A0 A8 14 4E
98 70 46 61 BB D1 B9 31 DF 8C 99 EE 75 6B 79 3C 40 A0 AE 97
00 90 9D DC 99 0D 33 A4 B5
Exponent: 01 00 01

Signature Length: 128 bytes / 1024 bits
Signature: 00 92 A4 B4 B8 14 55 63 25 51 4A 0B C3 2A 22 CF 3A F8 17 6A
0C CF 66 AA A7 65 2F 48 6D CD E3 3E 5C 9F 77 6C D4 44 54 1F
1E 84 4F 8E D4 8D DD AC 2D 88 09 21 A8 DA 56 2C A9 05 3C 49
68 35 19 75 0C DA 53 23 88 88 19 2D 74 26 C1 22 65 EE 11 68
83 6A 53 4A 9C 27 CB A0 B4 E9 8D 29 0C B2 3C 18 5C 67 CC 53
A6 1E 30 D0 AA 26 7B 1E AE 40 B9 29 01 6C 2E BC A2 19 94 7C
15 6E 8D 30 38 F6 CA 2E 75


------------------------------ snip ------------------------------
```

26024 - PostgreSQL Server Detection

Synopsis

A database service is listening on the remote host.

Description

The remote service is a PostgreSQL database server, or a derivative such as EnterpriseDB.

See Also

https://www.postgresql.org/

Solution

Limit incoming traffic to this port if desired.

Risk Factor

None

Plugin Information

Published: 2007/09/14, Modified: 2023/05/24

Plugin Output

tcp/5432/postgresql

22227 - RMI Registry Detection

Synopsis

An RMI registry is listening on the remote host.

Description

The remote host is running an RMI registry, which acts as a bootstrap naming service for registering and retrieving remote objects with simple names in the Java Remote Method Invocation (RMI) system.

See Also

https://docs.oracle.com/javase/1.5.0/docs/guide/rmi/spec/rmiTOC.html
http://www.nessus.org/u?b6fd7659

Solution

n/a

Risk Factor

None

Plugin Information

Published: 2006/08/16, Modified: 2022/06/01

Plugin Output

tcp/1099/rmi_registry
tcp/1099/rmi_registry

```
Valid response recieved for port 1099:
0x00: 51 AC ED 00 05 77 0F 01 9D 5E 08 38 00 00 01 94 Q....w...^.8....
0x10: 11 7C CE 55 80 00 75 72 00 13 5B 4C 6A 61 76 61 .|.U..ur..[Ljava
0x20: 2E 6C 61 6E 67 2E 53 74 72 69 6E 67 3B AD D2 56 .lang.String;..V
0x30: E7 E9 1D 7B 47 02 00 00 70 78 70 00 00 00 00 00 ...{G...pxp....
```

11111 - RPC Services Enumeration

Synopsis

An ONC RPC service is running on the remote host.

Description

By sending a DUMP request to the portmapper, it was possible to enumerate the ONC RPC services running on the remote port. Using this information, it is possible to connect and bind to each service by sending an RPC request to the remote port.

Solution

n/a

Risk Factor

None

Plugin Information

Published: 2002/08/24, Modified: 2011/05/24

Plugin Output

tcp/111/rpc-portmapper

```
The following RPC services are available on TCP port 111 :

- program: 100000 (portmapper), version: 2
```

11111 - RPC Services Enumeration

Synopsis

An ONC RPC service is running on the remote host.

Description

By sending a DUMP request to the portmapper, it was possible to enumerate the ONC RPC services running on the remote port. Using this information, it is possible to connect and bind to each service by sending an RPC request to the remote port.

Solution

n/a

Risk Factor

None

Plugin Information

Published: 2002/08/24, Modified: 2011/05/24

Plugin Output

udp/111/rpc-portmapper

```
The following RPC services are available on UDP port 111 :

- program: 100000 (portmapper), version: 2
```

11111 - RPC Services Enumeration -

Synopsis

An ONC RPC service is running on the remote host.

Description

By sending a DUMP request to the portmapper, it was possible to enumerate the ONC RPC services running on the remote port. Using this information, it is possible to connect and bind to each service by sending an RPC request to the remote port.

Solution

n/a

Risk Factor

None

Plugin Information

Published: 2002/08/24, Modified: 2011/05/24

Plugin Output

tcp/2049/rpc-nfs

```
The following RPC services are available on TCP port 2049 :

- program: 100003 (nfs), version: 2
- program: 100003 (nfs), version: 3
- program: 100003 (nfs), version: 4
```

11111 - RPC Services Enumeration -

Synopsis

An ONC RPC service is running on the remote host.

Description

By sending a DUMP request to the portmapper, it was possible to enumerate the ONC RPC services running on the remote port. Using this information, it is possible to connect and bind to each service by sending an RPC request to the remote port.

Solution

n/a

Risk Factor

None

Plugin Information

Published: 2002/08/24, Modified: 2011/05/24

Plugin Output

udp/2049/rpc-nfs

```
The following RPC services are available on UDP port 2049 :

- program: 100003 (nfs), version: 2
- program: 100003 (nfs), version: 3
- program: 100003 (nfs), version: 4
```

11111 - RPC Services Enumeration -

Synopsis

An ONC RPC service is running on the remote host.

Description

By sending a DUMP request to the portmapper, it was possible to enumerate the ONC RPC services running on the remote port. Using this information, it is possible to connect and bind to each service by sending an RPC request to the remote port.

Solution

n/a

Risk Factor

None

Plugin Information

Published: 2002/08/24, Modified: 2011/05/24

Plugin Output

tcp/34557/rpc-nlockmgr

```
The following RPC services are available on TCP port 34557 :

- program: 100021 (nlockmgr), version: 1
- program: 100021 (nlockmgr), version: 3
- program: 100021 (nlockmgr), version: 4
```

11111 - RPC Services Enumeration -

Synopsis

An ONC RPC service is running on the remote host.

Description

By sending a DUMP request to the portmapper, it was possible to enumerate the ONC RPC services running on the remote port. Using this information, it is possible to connect and bind to each service by sending an RPC request to the remote port.

Solution

n/a

Risk Factor

None

Plugin Information

Published: 2002/08/24, Modified: 2011/05/24

Plugin Output

udp/38039/rpc-mountd

```
The following RPC services are available on UDP port 38039 :
```

```
- program: 100005 (mountd), version: 1
- program: 100005 (mountd), version: 2
- program: 100005 (mountd), version: 3
```

11111 - RPC Services Enumeration

Synopsis

An ONC RPC service is running on the remote host.

Description

By sending a DUMP request to the portmapper, it was possible to enumerate the ONC RPC services running on the remote port. Using this information, it is possible to connect and bind to each service by sending an RPC request to the remote port.

Solution

n/a

Risk Factor

None

Plugin Information

Published: 2002/08/24, Modified: 2011/05/24

Plugin Output

udp/44195/rpc-status

```
The following RPC services are available on UDP port 44195 :

- program: 100024 (status), version: 1
```

11111 - RPC Services Enumeration

Synopsis

An ONC RPC service is running on the remote host.

Description

By sending a DUMP request to the portmapper, it was possible to enumerate the ONC RPC services running on the remote port. Using this information, it is possible to connect and bind to each service by sending an RPC request to the remote port.

Solution

n/a

Risk Factor

None

Plugin Information

Published: 2002/08/24, Modified: 2011/05/24

Plugin Output

udp/45425/rpc-nlockmgr

```
The following RPC services are available on UDP port 45425 :

- program: 100021 (nlockmgr), version: 1
- program: 100021 (nlockmgr), version: 3
- program: 100021 (nlockmgr), version: 4
```

11111 - RPC Services Enumeration

Synopsis

An ONC RPC service is running on the remote host.

Description

By sending a DUMP request to the portmapper, it was possible to enumerate the ONC RPC services running on the remote port. Using this information, it is possible to connect and bind to each service by sending an RPC request to the remote port.

Solution

n/a

Risk Factor

None

Plugin Information

Published: 2002/08/24, Modified: 2011/05/24

Plugin Output

tcp/45931/rpc-status

```
The following RPC services are available on TCP port 45931 :

- program: 100024 (status), version: 1
```

11111 - RPC Services Enumeration

Synopsis

An ONC RPC service is running on the remote host.

Description

By sending a DUMP request to the portmapper, it was possible to enumerate the ONC RPC services running on the remote port. Using this information, it is possible to connect and bind to each service by sending an RPC request to the remote port.

Solution

n/a

Risk Factor

None

Plugin Information

Published: 2002/08/24, Modified: 2011/05/24

Plugin Output

tcp/49557/rpc-mountd

```
The following RPC services are available on TCP port 49557 :

- program: 100005 (mountd), version: 1
- program: 100005 (mountd), version: 2
- program: 100005 (mountd), version: 3
```

53335 - RPC portmapper (TCP)

Synopsis

An ONC RPC portmapper is running on the remote host.

Description

The RPC portmapper is running on this port.

The portmapper allows someone to get the port number of each RPC service running on the remote host by sending either multiple lookup requests or a DUMP request.

Solution

n/a

Risk Factor

None

Plugin Information

Published: 2011/04/08, Modified: 2011/08/29

Plugin Output

tcp/111/rpc-portmapper

10223 - RPC portmapper Service Detection

Synopsis

An ONC RPC portmapper is running on the remote host.

Description

The RPC portmapper is running on this port.

The portmapper allows someone to get the port number of each RPC service running on the remote host by sending either multiple lookup requests or a DUMP request.

Solution

n/a

Risk Factor

None

CVSS v3.0 Base Score

0.0 (CVSS:3.0/AV:N/AC:L/PR:N/UI:N/S:U/C:N/I:N/A:N)

CVSS v2.0 Base Score

0.0 (CVSS2#AV:N/AC:L/Au:N/C:N/I:N/A:N)

References

CVE CVE-1999-0632

Plugin Information

Published: 1999/08/19, Modified: 2019/10/04

Plugin Output

udp/111/rpc-portmapper

10263 - SMTP Server Detection

Synopsis

An SMTP server is listening on the remote port.

Description

The remote host is running a mail (SMTP) server on this port.

Since SMTP servers are the targets of spammers, it is recommended you disable it if you do not use it.

Solution

Disable this service if you do not use it, or filter incoming traffic to this port.

Risk Factor

None

References

XREF IAVT:0001-T-0932

Plugin Information

Published: 1999/10/12, Modified: 2020/09/22

Plugin Output

tcp/25/smtp

```
Remote SMTP server banner :

220 metasploitable.localdomain ESMTP Postfix (Ubuntu)
```

42088 - SMTP Service STARTTLS Command Support

Synopsis

The remote mail service supports encrypting traffic.

Description

The remote SMTP service supports the use of the 'STARTTLS' command to switch from a cleartext to an encrypted communications channel.

See Also

https://en.wikipedia.org/wiki/STARTTLS
https://tools.ietf.org/html/rfc2487

Solution

n/a

Risk Factor

None

Plugin Information

Published: 2009/10/09, Modified: 2019/03/20

Plugin Output

tcp/25/smtp

```
Here is the SMTP service's SSL certificate that Nessus was able to
collect after sending a 'STARTTLS' command :

------------------------------ snip ------------------------------
Subject Name:

Country: XX
State/Province: There is no such thing outside US
Locality: Everywhere
Organization: OCOSA
Organization Unit: Office for Complication of Otherwise Simple Affairs
Common Name: ubuntu804-base.localdomain
Email Address: root@ubuntu804-base.localdomain

Issuer Name:

Country: XX
State/Province: There is no such thing outside US
Locality: Everywhere
Organization: OCOSA
Organization Unit: Office for Complication of Otherwise Simple Affairs
Common Name: ubuntu804-base.localdomain
Email Address: root@ubuntu804-base.localdomain

Serial Number: 00 FA F9 3A 4C 7F B6 B9 CC

Version: 1

Signature Algorithm: SHA-1 With RSA Encryption
```

```
Not Valid Before: Mar 17 14:07:45 2010 GMT
Not Valid After: Apr 16 14:07:45 2010 GMT

Public Key Info:

Algorithm: RSA Encryption
Key Length: 1024 bits
Public Key: 00 D6 B4 13 36 33 9A 95 71 7B 1B DE 7C 83 75 DA 71 B1 3C A9
7F FE AD 64 1B 77 E9 4F AE BE CA D4 F8 CB EF AE BB 43 79 24
73 FF 3C E5 9E 3B 6D FC C8 B1 AC FA 4C 4D 5E 9B 4C 99 54 0B
D7 A8 4A 50 BA A9 DE 1D 1F F4 E4 6B 02 A3 F4 6B 45 CD 4C AF
8D 89 62 33 8F 65 BB 36 61 9F C4 2C 73 C1 4E 2E A0 A8 14 4E
98 70 46 61 BB D1 B9 31 DF 8C 99 EE 75 6B 79 3C 40 A0 AE 97
00 90 9D DC 99 0D 33 A4 B5
Exponent: 01 00 01

Signature Length: 128 bytes / 1024 bits
Signature: 00 92 A4 B4 B8 14 55 63 25 51 4A 0B C3 2A 22 CF 3A F8 17 6A
0C CF 66 AA A7 65 2F 48 6D CD E3 3E 5C 9F 77 6C D4 44 54 1F
1E 84 4F 8E D4 8D DD AC 2D 88 09 21 A8 DA 56 2C A9 05 3C 49
68 35 19 75 0C DA 53 23 88 88 19 2D 74 26 C1 22 65 EE 11 68
83 6A 53 4A 9C 27 CB A0 B4 E9 8D 29 0C B2 3C 18 5C 67 CC 53
A6 1E 30 D0 AA 26 7B 1E AE 40 B9 29 01 6C 2E BC A2 19 94 7C
15 6E 8D 30 38 F6 CA 2E 75

------------------------------ snip ------------------------------
```

149334 - SSH Password Authentication Accepted

Synopsis

The SSH server on the remote host accepts password authentication.

Description

The SSH server on the remote host accepts password authentication.

See Also

https://tools.ietf.org/html/rfc4252#section-8

Solution

n/a

Risk Factor

None

Plugin Information

Published: 2021/05/07, Modified: 2021/05/07

Plugin Output

tcp/22/ssh

10881 - SSH Protocol Versions Supported

Synopsis

A SSH server is running on the remote host.

Description

This plugin determines the versions of the SSH protocol supported by the remote SSH daemon.

Solution

n/a

Risk Factor

None

Plugin Information

Published: 2002/03/06, Modified: 2024/07/24

Plugin Output

tcp/22/ssh

```
The remote SSH daemon supports the following versions of the
SSH protocol :

- 1.99
- 2.0
```

10267 - SSH Server Type and Version Information

Synopsis

An SSH server is listening on this port.

Description

It is possible to obtain information about the remote SSH server by sending an empty authentication request.

Solution

n/a

Risk Factor

None

References

XREF IAVT:0001-T-0933

Plugin Information

Published: 1999/10/12, Modified: 2024/07/24

Plugin Output

tcp/22/ssh

```
SSH version : SSH-2.0-OpenSSH_4.7p1 Debian-8ubuntu1
SSH supported authentication : publickey,password
```

56984 - SSL / TLS Versions Supported

Synopsis

The remote service encrypts communications.

Description

This plugin detects which SSL and TLS versions are supported by the remote service for encrypting communications.

Solution

n/a

Risk Factor

None

Plugin Information

Published: 2011/12/01, Modified: 2023/07/10

Plugin Output

tcp/25/smtp

```
This port supports SSLv2/SSLv3/TLSv1.0.
```

56984 - SSL / TLS Versions Supported

Synopsis

The remote service encrypts communications.

Description

This plugin detects which SSL and TLS versions are supported by the remote service for encrypting communications.

Solution

n/a

Risk Factor

None

Plugin Information

Published: 2011/12/01, Modified: 2023/07/10

Plugin Output

tcp/5432/postgresql

```
This port supports SSLv3/TLSv1.0.
```

45410 - SSL Certificate 'commonName' Mismatch

Synopsis

The 'commonName' (CN) attribute in the SSL certificate does not match the hostname.

Description

The service running on the remote host presents an SSL certificate for which the 'commonName' (CN) attribute does not match the hostname on which the service listens.

Solution

If the machine has several names, make sure that users connect to the service through the DNS hostname that matches the common name in the certificate.

Risk Factor

None

Plugin Information

Published: 2010/04/03, Modified: 2021/03/09

Plugin Output

tcp/25/smtp

```
The host name known by Nessus is :

metasploitable

The Common Name in the certificate is :

ubuntu804-base.localdomain
```

45410 - SSL Certificate 'commonName' Mismatch

Synopsis

The 'commonName' (CN) attribute in the SSL certificate does not match the hostname.

Description

The service running on the remote host presents an SSL certificate for which the 'commonName' (CN) attribute does not match the hostname on which the service listens.

Solution

If the machine has several names, make sure that users connect to the service through the DNS hostname that matches the common name in the certificate.

Risk Factor

None

Plugin Information

Published: 2010/04/03, Modified: 2021/03/09

Plugin Output

tcp/5432/postgresql

```
The host name known by Nessus is :

metasploitable

The Common Name in the certificate is :

ubuntu804-base.localdomain
```

10863 - SSL Certificate Information

Synopsis

This plugin displays the SSL certificate.

Description

This plugin connects to every SSL-related port and attempts to extract and dump the X.509 certificate.

Solution

n/a

Risk Factor

None

Plugin Information

Published: 2008/05/19, Modified: 2021/02/03

Plugin Output

tcp/25/smtp

```
Subject Name:

Country: XX
State/Province: There is no such thing outside US
Locality: Everywhere
Organization: OCOSA
Organization Unit: Office for Complication of Otherwise Simple Affairs
Common Name: ubuntu804-base.localdomain
Email Address: root@ubuntu804-base.localdomain

Issuer Name:

Country: XX
State/Province: There is no such thing outside US
Locality: Everywhere
Organization: OCOSA
Organization Unit: Office for Complication of Otherwise Simple Affairs
Common Name: ubuntu804-base.localdomain
Email Address: root@ubuntu804-base.localdomain

Serial Number: 00 FA F9 3A 4C 7F B6 B9 CC

Version: 1

Signature Algorithm: SHA-1 With RSA Encryption

Not Valid Before: Mar 17 14:07:45 2010 GMT
Not Valid After: Apr 16 14:07:45 2010 GMT
```

```
Public Key Info:

Algorithm: RSA Encryption
Key Length: 1024 bits
Public Key: 00 D6 B4 13 36 33 9A 95 71 7B 1B DE 7C 83 75 DA 71 B1 3C A9
7F FE AD 64 1B 77 E9 4F AE BE CA D4 F8 CB EF AE BB 43 79 24
73 FF 3C E5 9E 3B 6D FC C8 B1 AC FA 4C 4D 5E 9B 4C 99 54 0B
D7 A8 4A 50 BA A9 DE 1D 1F F4 E4 6B 02 A3 F4 6B 45 CD 4C AF
8D 89 62 33 8F 65 BB 36 61 9F C4 2C 73 C1 4E 2E A0 AB 14 4E
98 70 46 61 BB D1 B9 31 DF 8C 99 EE 75 6B 79 3C 40 A0 AE 97
00 90 9D DC 99 0D 33 A4 B5
Exponent: 01 00 01

Signature Length: 128 bytes / 1024 bits
Signature: 00 92 A4 B4 B8 14 55 63 25 51 4A 0B C3 2A 22 CF 3A F8 17 6A
0C CF 66 AA A7 65 2F 48 6D CD E3 3E 5C 9F 77 6C D4 44 54 1F
1E 84 4F 8E D4 8D DD AC 2D 88 09 21 A8 DA 56 2C A9 05 3C 49
68 35 19 75 0C DA 53 23 88 88 19 2D 74 26 C1 22 65 EE 11 68
83 6A 53 4A 9C 27 CB A0 B4 E9 8D 29 0C B2 3C 18 5C 67 CC 53
A6 1E 30 D0 AA 26 7B 1E AE 40 B9 29 01 6C 2E BC A2 19 94 7C
15 6E 8D 30 38 F6 CA 2E 75

Fingerprints :

SHA-256 Fingerprint: E7 A7 FA 0D 63 E4 57 C7 C4 A5 9B 38 B7 08 49 C6 A7 0B DA 6F
83 0C 7A F1 E3 2D EE 43 6D E8 13 CC
SHA-1 Fingerprint: ED 09 30 88 70 66 03 BF D5 DC 23 73 99 B4 98 DA 2D 4D 31 C6
MD5 Fingerprint: DC D9 AD 90 6C 8F 2F 73 74 AF 38 3B 25 40 88 28


PEM certificate :

-----BEGIN CERTIFICATE-----
MIIDWzCCAsQCCQD6+TpMf7a5zDANBgkqhkiG9w0BAQUFADCB8TELMAkGA1UEBhMCWFgxKjAoBgNVBAgTIVRoZXJlIGlzIG5vIHN1Y2ggdGhpbmcgb3V0c2lk
ZSBVUzETMBEGA1UEBxMKRXZlcnl3aGVyZTEOMAwGA1UEChMFT0NPU0ExPDA6BgNVBAsTM09mZmljZSBmb3IgQ29tcGxpY2F0aW9uIG9mIE90aGVyd2lzZSBT
aW1wbGUgQWZmYWlyczEjMCEGA1UEAxMadWJ1bnR1ODA0LWJhc2UubG9jYWxkb21haW4xLjAsBgkqhkiG9w0BCQEWH3Jvb3RAdWJ1bnR1ODA0LWJhc2UubG9j
YWxkb21haW4wHhcNMTAwMzE3MTQwNzQ1WhcNMTAwNDE2MTQwNzQ1WjCB8TELMAkGA1UEBhMCWFgxKjAoBgNVBAgTIVRoZXJlIGlzIG5vIHN1Y2ggdGhpbmcg
b3V0c2lkZSBVUzETMBEGA1UEBxMKRXZlcnl3aGVyZTEOMAwGA1UEChMFT0NPU0ExPDA6BgNVBAsTM09mZmljZSBmb3IgQ29tcGxpY2F0aW9uIG9mIE90aGVy
d2lzZSBTaW1wbGUgQWZmYWlyczEjMCEGA1UEAxMadWJ1bnR1ODA0LWJhc2UubG9jYWxkb21haW4xLjAsBgkqhkiG9w0BCQEWH3Jvb3RAdWJ1bnR1ODA0LWJh
c2UubG9jYWxkb21haW4wgZ8wDQYJKoZIhvcNAQEBBQADgY0AMIGJAoGBANa0EzYzmpVxexvefIN12nGxPKl//q1kG3fpT66+ytT4y++uu0N5JHP/POWeO238
yLGs+kxNXptMmVQL16hKULqp3h0f9ORrAqP0a0XNTK+NiWIzj2W7NmGfxCxzwU4uoKgUTphwRmG70bkx34yZ7nVreTxAoK6XAJCd3JkNM6S1AgMBAAEwDQYJ
KoZIhvcNAQEFBQADgYEAkqS0uBRVYyVRSgvDKiLPOvgXagzPZqqnZS9Ibc3jPlyfd2zURFQfHoRPjtSN3awtiAkhqNpWLKkFPEloNRl1DNpTI4iIGS10JsEi
Ze4RaINqU0qcJ8ugtOmNKQyyPBhcZ8xTph4w0Komex6uQLkpAWwuvKIZlHwVbo0wOPbKLnU=
-----END CERTIFICATE-----
```

10863 - SSL Certificate Information

Synopsis

This plugin displays the SSL certificate.

Description

This plugin connects to every SSL-related port and attempts to extract and dump the X.509 certificate.

Solution

n/a

Risk Factor

None

Plugin Information

Published: 2008/05/19, Modified: 2021/02/03

Plugin Output

tcp/5432/postgresql

```
Subject Name:

Country: XX
State/Province: There is no such thing outside US
Locality: Everywhere
Organization: OCOSA
Organization Unit: Office for Complication of Otherwise Simple Affairs
Common Name: ubuntu804-base.localdomain
Email Address: root@ubuntu804-base.localdomain
```

```
Issuer Name:

Country: XX
State/Province: There is no such thing outside US
Locality: Everywhere
Organization: OCOSA
Organization Unit: Office for Complication of Otherwise Simple Affairs
Common Name: ubuntu804-base.localdomain
Email Address: root@ubuntu804-base.localdomain

Serial Number: 00 FA F9 3A 4C 7F B6 B9 CC

Version: 1

Signature Algorithm: SHA-1 With RSA Encryption

Not Valid Before: Mar 17 14:07:45 2010 GMT
Not Valid After: Apr 16 14:07:45 2010 GMT

Public Key Info:

Algorithm: RSA Encryption
Key Length: 1024 bits
Public Key: 00 D6 B4 13 36 33 9A 95 71 7B 1B DE 7C 83 75 DA 71 B1 3C A9
7F FE AD 64 1B 77 E9 4F AE BE CA D4 F8 CB EF AE BB 43 79 24
73 FF 3C E5 9E 3B 6D FC C8 B1 AC FA 4C 4D 5E 9B 4C 99 54 0B
D7 A8 4A 50 BA A9 DE 1D 1F F4 E4 6B 02 A3 F4 6B 45 CD 4C AF
8D 89 62 33 8F 65 BB 36 61 9F C4 2C 73 C1 4E 2E A0 A8 14 4E
98 70 46 61 BB D1 B9 31 DF 8C 99 EE 75 6B 79 3C 40 A0 AE 97
00 90 9D DC 99 0D 33 A4 B5
Exponent: 01 00 01

Signature Length: 128 bytes / 1024 bits
Signature: 00 92 A4 B4 B8 14 55 63 25 51 4A 0B C3 2A 22 CF 3A F8 17 6A
0C CF 66 AA A7 65 2F 48 6D CD E3 3E 5C 9F 77 6C D4 44 54 1F
1E 84 4F 8E D4 8D DD AC 2D 88 09 21 A8 DA 56 2C A9 05 3C 49
68 35 19 75 0C DA 53 23 88 88 19 2D 74 26 C1 22 65 EE 11 68
83 6A 53 4A 9C 27 CB A0 B4 E9 8D 29 0C B2 3C 18 5C 67 CC 53
A6 1E 30 D0 AA 26 7B 1E AE 40 B9 29 01 6C 2E BC A2 19 94 7C
15 6E 8D 30 38 F6 CA 2E 75

Fingerprints :

SHA-256 Fingerprint: E7 A7 FA 0D 63 E4 57 C7 C4 A5 9B 38 B7 08 49 C6 A7 0B DA 6F
83 0C 7A F1 E3 2D EE 43 6D E8 13 CC
SHA-1 Fingerprint: ED 09 30 88 70 66 03 BF D5 DC 23 73 99 B4 98 DA 2D 4D 31 C6
MD5 Fingerprint: DC D9 AD 90 6C 8F 2F 73 74 AF 38 3B 25 40 88 28


PEM certificate :

-----BEGIN CERTIFICATE-----
MIIDWzCCAsQCCQD6+TpMf7a5zDANBgkqhkiG9w0BAQUFADCB8TELMAkGA1UEBhMCWFgxKjAoBgNVBAgTIVRoZXJlIGlzIG5vIHN1Y2ggdGhpbmcgb3V0c2lk
ZSBVUzETMBEGA1UEBxMKRXZlcnl3aGVyZTEOMAwGA1UEChMFT0NPU0ExPDA6BgNVBAsTM09mZmljZSBmb3IgQ29tcGxpY2F0aW9uIG9mIE90aGVyd2lzZSBT
aW1wbGUgQWZmYWlyczEjMCEGA1UEAxMadWJ1bnR1ODA0LWJhc2UubG9jYWxkb21haW4xLjAsBgkqhkiG9w0BCQEWH3Jvb3RAdWJ1bnR1ODA0LWJhc2UubG9j
YWxkb21haW4wHhcNMTAwMzE3MTQwNzQ1WhcNMTAwNDE2MTQwNzQ1WjCB8TELMAkGA1UEBhMCWFgxKjAoBgNVBAgTIVRoZXJlIGlzIG5vIHN1Y2ggdGhpbmcg
b3V0c2lkZSBVUzETMBEGA1UEBxMKRXZlcnl3aGVyZTEOMAwGA1UEChMFT0NPU0ExPDA6BgNVBAsTM09mZmljZSBmb3IgQ29tcGxpY2F0aW9uIG9mIE90aGVy
d2lzZSBTaW1wbGUgQWZmYWlyczEjMCEGA1UEAxMadWJ1bnR1ODA0LWJhc2UubG9jYWxkb21haW4xLjAsBgkqhkiG9w0BCQEWH3Jvb3RAdWJ1bnR1ODA0LWJh
c2UubG9jYWxkb21haW4wgZ8wDQYJKoZIhvcNAQEBBQADgY0AMIGJAoGBANa0EzYzmpVxexvefIN12nGxPKl//q1kG3fpT66+ytT4y++uu0N5JHP/POWeO238
yLGs+kxNXptMmVQL16hKULqp3h0f9ORrAqP0a0XNTK+NiWIzj2W7NmGfxCxzwU4uoKgUTphwRmG70bkx34yZ7nVreTxAoK6XAJCd3JkNM6S1AgMBAAEwDQYJ
KoZIhvcNAQEFBQADgYEAkqS0uBRVYyVRSgvDKiLPOvgXagzPZqqnZS9Ibc3jPlyfd2zURFQfHoRPjtSN3awtiAkhqNpWLKkFPEloNRl1DNpTI4iIGS10JsEi
Ze4RaINqU0qcJ8ugtOmNKQyyPBhcZ8xTph4w0Komex6uQLkpAWwuvKIZlHwVbo0wOPbKLnU=
-----END CERTIFICATE-----
```

70544 - SSL Cipher Block Chaining Cipher Suites Supported -

Synopsis

The remote service supports the use of SSL Cipher Block Chaining ciphers, which combine previous blocks with subsequent ones.

Description

The remote host supports the use of SSL ciphers that operate in Cipher Block Chaining (CBC) mode. These cipher suites offer additional security over Electronic Codebook (ECB) mode, but have the potential to leak information if used improperly.

See Also

https://www.openssl.org/docs/manmaster/man1/ciphers.html
http://www.nessus.org/u?cc4a822a
https://www.openssl.org/~bodo/tls-cbc.txt

Solution

n/a

Risk Factor

None

Plugin Information

Published: 2013/10/22, Modified: 2021/02/03

Plugin Output

tcp/25/smtp

```
Here is the list of SSL CBC ciphers supported by the remote server :

Low Strength Ciphers (<= 64-bit key)

Name Code KEX Auth Encryption MAC
---------------------- ---------- --- ---- --------------------- ---
EXP-RC2-CBC-MD5 0x04, 0x00, 0x80 RSA(512) RSA RC2-CBC(40) MD5 export
EXP-EDH-RSA-DES-CBC-SHA 0x00, 0x14 DH(512) RSA DES-CBC(40) SHA1 export
EDH-RSA-DES-CBC-SHA 0x00, 0x15 DH RSA DES-CBC(56) SHA1
EXP-ADH-DES-CBC-SHA 0x00, 0x19 DH(512) None DES-CBC(40) SHA1 export
ADH-DES-CBC-SHA 0x00, 0x1A DH None DES-CBC(56) SHA1
EXP-DES-CBC-SHA 0x00, 0x08 RSA(512) RSA DES-CBC(40) SHA1 export
EXP-RC2-CBC-MD5 0x00, 0x06 RSA(512) RSA RC2-CBC(40) MD5 export
DES-CBC-SHA 0x00, 0x09 RSA RSA DES-CBC(56) SHA1

Medium Strength Ciphers (> 64-bit and < 112-bit key, or 3DES)

Name Code KEX Auth Encryption MAC
---------------------- ---------- --- ---- --------------------- ---
DES-CBC3-MD5 0x07, 0x00, 0xC0 RSA RSA 3DES-CBC(168) MD5
EDH-RSA-DES-CBC3-SHA 0x00, 0x16 DH RSA 3DES-CBC(168) SHA1
ADH-DES-CBC3-SHA 0x00, 0x1B DH None 3DES-CBC(168) SHA1
DES-CBC3-SHA 0x00, 0x0A RSA RSA 3DES-CBC(168) SHA1

High Strength Ciphers (>= 112-bit key)

Name Code KEX Auth Encryption MAC
---------------------- ---------- --- ---- --------------------- ---
DHE-RSA-AES128-SHA 0x00, 0x33 DH RSA AES-CBC(128) SHA1
DHE-RSA-AES256-SHA 0x00, 0x39 DH RSA AES-CBC(256) SHA1
ADH-AES128-SHA 0x00, 0x34 DH None AES-CBC(128) SHA1
ADH-AES256-SHA 0x00, 0x3A DH None AES-CBC(256) SHA1
AES128-SHA 0x00, 0x2F RSA RSA AES-CBC(128) SHA1
AES256-SHA 0x00, 0x35 RSA RSA AES-CBC(256) SHA1

The fields above are :

{Tenable ciphername}
{Cipher ID code}
Kex={key exchange}
Auth={authentication}
Encrypt={symmetric encryption method}
MAC={message authentication code}
{export flag}
```

70544 - SSL Cipher Block Chaining Cipher Suites Supported -

Synopsis

The remote service supports the use of SSL Cipher Block Chaining ciphers, which combine previous blocks with subsequent ones.

Description

The remote host supports the use of SSL ciphers that operate in Cipher Block Chaining (CBC) mode. These cipher suites offer additional security over Electronic Codebook (ECB) mode, but have the potential to leak information if used improperly.

See Also

https://www.openssl.org/docs/manmaster/man1/ciphers.html
http://www.nessus.org/u?cc4a822a
https://www.openssl.org/~bodo/tls-cbc.txt

Solution

n/a

Risk Factor

None

Plugin Information

Published: 2013/10/22, Modified: 2021/02/03

Plugin Output

tcp/5432/postgresql

```
Here is the list of SSL CBC ciphers supported by the remote server :

Medium Strength Ciphers (> 64-bit and < 112-bit key, or 3DES)

Name Code KEX Auth Encryption MAC
---------------------- ---------- --- ---- --------------------- ---
EDH-RSA-DES-CBC3-SHA 0x00, 0x16 DH RSA 3DES-CBC(168) SHA1
DES-CBC3-SHA 0x00, 0x0A RSA RSA 3DES-CBC(168) SHA1

High Strength Ciphers (>= 112-bit key)

Name Code KEX Auth Encryption MAC
---------------------- ---------- --- ---- --------------------- ---
DHE-RSA-AES128-SHA 0x00, 0x33 DH RSA AES-CBC(128) SHA1
DHE-RSA-AES256-SHA 0x00, 0x39 DH RSA AES-CBC(256) SHA1
AES128-SHA 0x00, 0x2F RSA RSA AES-CBC(128) SHA1
AES256-SHA 0x00, 0x35 RSA RSA AES-CBC(256) SHA1

The fields above are :

{Tenable ciphername}
{Cipher ID code}
Kex={key exchange}
Auth={authentication}
Encrypt={symmetric encryption method}
MAC={message authentication code}
{export flag}
```

21643 - SSL Cipher Suites Supported

Synopsis

The remote service encrypts communications using SSL.

Description

This plugin detects which SSL ciphers are supported by the remote service for encrypting communications.

See Also

https://www.openssl.org/docs/man1.0.2/man1/ciphers.html
http://www.nessus.org/u?e17ffced

Solution

n/a

Risk Factor

None

Plugin Information

Published: 2006/06/05, Modified: 2024/09/11

Plugin Output

tcp/25/smtp

```
Here is the list of SSL ciphers supported by the remote server :
Each group is reported per SSL Version.

SSL Version : TLSv1
```

```
Low Strength Ciphers (<= 64-bit key)

Name Code KEX Auth Encryption MAC
---------------------- ---------- --- ---- --------------------- ---
EXP-EDH-RSA-DES-CBC-SHA 0x00, 0x14 DH(512) RSA DES-CBC(40) SHA1 export
EDH-RSA-DES-CBC-SHA 0x00, 0x15 DH RSA DES-CBC(56) SHA1
EXP-ADH-DES-CBC-SHA 0x00, 0x19 DH(512) None DES-CBC(40) SHA1 export
EXP-ADH-RC4-MD5 0x00, 0x17 DH(512) None RC4(40) MD5 export
ADH-DES-CBC-SHA 0x00, 0x1A DH None DES-CBC(56) SHA1
EXP-DES-CBC-SHA 0x00, 0x08 RSA(512) RSA DES-CBC(40) SHA1 export
EXP-RC2-CBC-MD5 0x00, 0x06 RSA(512) RSA RC2-CBC(40) MD5 export
EXP-RC4-MD5 0x00, 0x03 RSA(512) RSA RC4(40) MD5 export
DES-CBC-SHA 0x00, 0x09 RSA RSA DES-CBC(56) SHA1

Medium Strength Ciphers (> 64-bit and < 112-bit key, or 3DES)

Name Code KEX Auth Encryption MAC
---------------------- ---------- --- ---- --------------------- ---
EDH-RSA-DES-CBC3-SHA 0x00, 0x16 DH RSA 3DES-CBC(168) SHA1
ADH-DES-CBC3-SHA 0x00, 0x1B DH None 3DES-CBC(168) SHA1
DES-CBC3-SHA 0x00, 0x0A RSA RSA 3DES-CBC(168) SHA1

High Strength Ciphers (>= 112-bit key)

Name Code KEX Auth Encryption MAC
---------------------- ---------- --- ---- --------------------- ---
DHE-RSA-AES128-SHA 0x00, 0x33 DH RSA AES-CBC(128) SHA1
DHE-RSA-AES256-SHA 0x00, 0x39 DH RSA AES-CBC(256) SHA1
ADH-AES128-SHA 0x00, 0x34 DH None AES-CBC(128) SHA1
ADH-AES256-SHA 0x00, 0x3A DH None AES-CBC(256) SHA1
ADH-RC4-MD5 0x00, 0x18 DH None RC4(128) MD5
AES128-SHA 0x00, 0x2F RSA RSA AES-CBC(128) SHA1
AES256-SHA 0x00, 0x35 RSA RSA AES-CBC(256) SHA1
RC4-MD5 0x00, 0x04 RSA RSA RC4(128) MD5
RC4-SHA 0x00, 0x05 RSA RSA RC4(128) SHA1


SSL Version : SSLv3
Low Strength Ciphers (<= 64-bit key)

Name Code KEX Auth Encryption MAC
---------------------- ---------- --- ---- --------------------- ---
EXP-EDH-RSA-DES-CBC-SHA 0x00, 0x14 DH(512) RSA DES-CBC(40) SHA1 export
EDH-RSA-DES-CBC-SHA 0x00, 0x15 DH RSA DES-CBC(56) SHA1
EXP-ADH-DES-CBC-SHA 0x00, 0x19 DH(512) None DES-CBC(40) SHA1 export
EXP-ADH-RC4-MD5 0x00, 0x17 DH(512) None RC4(40) MD5 export
ADH-DES-CBC-SHA 0x00, 0x1A DH None DES-CBC(56) SHA1
EXP-DES-CBC-SHA 0x00, 0x08 RSA(512) RSA DES-CBC(40) SHA1 export
EXP-RC2-CBC-MD5 0x00, 0x06 RSA(512) RSA RC2-CBC(40) MD5 export
EXP-RC4-MD5 0x00, 0x03 RSA(512) RSA RC4(40) MD5 export
DES-CBC-SHA 0x00, 0x09 RSA RSA DES-CBC(56) SHA1

Medium Strength Ciphers (> 64-bit and < 112-bit key, or 3DES)

Name Code KEX Auth Encryption MAC
---------------------- ---------- --- ---- --------------------- ---
EDH-RSA-DES-CBC3-SHA 0x00, 0x16 DH RSA 3DES-CBC(168) SHA1
ADH-DES-CBC3-SHA 0x00, 0x1B DH None 3DES-CBC(168) SHA1
DES-CBC3-SHA 0x00, 0x0A RSA RSA 3DES-CBC(168) SHA1

High Strength Ciphers (>= 112-bit key)

Name Code KEX Auth Encryption MAC
---------------------- ---------- --- ---- --------------------- ---
DHE-RSA-AES128-SHA 0x00, 0x33 DH RSA AES-CBC(128) SHA1
DHE-RSA-AES256-SHA 0x00, 0x39 DH RSA AES-CBC(256) SHA1
ADH-AES128-SHA 0x00, 0x34 DH None AES-CBC(128) SHA1
ADH-AES256-SHA 0x00, 0x3A DH None AES-CBC(256) SHA1
ADH-RC4-MD5 0x00, 0x18 DH None RC4(128) MD5
AES128-SHA 0x00, 0x2F RSA RSA AES-CBC(128) SHA1
AES256-SHA 0x00, 0x35 RSA RSA AES-CBC(256) SHA1
RC4-MD5 0x00, 0x04 RSA RSA RC4(128) MD5
RC4-SHA 0x00, 0x05 RSA RSA RC4(128) SHA1


SSL Version : SSLv2
Low Strength Ciphers (<= 64-bit key)

Name Code KEX Auth Encryption MAC
---------------------- ---------- --- ---- --------------------- ---
EXP-RC2-CBC-MD5 0x04, 0x00, 0x80 RSA(512) RSA RC2-CBC(40) MD5 export
EXP-RC4-MD5 0x02, 0x00, 0x80 RSA(512) RSA RC4(40) MD5 export

Medium Strength Ciphers (> 64-bit and < 112-bit key, or 3DES)

Name Code KEX Auth Encryption MAC
```

```
--------------------- ---------- --- ---- --------------------- ---
DES-CBC3-MD5 0x07, 0x00, 0xC0 RSA RSA 3DES-CBC(168) MD5

High Strength Ciphers (>= 112-bit key)

Name Code KEX Auth Encryption MAC
--------------------- ---------- --- ---- --------------------- ---
RC4-MD5 0x01, 0x00, 0x80 RSA RSA RC4(128) MD5

The fields above are :

{Tenable ciphername}
{Cipher ID code}
Kex={key exchange}
Auth={authentication}
Encrypt={symmetric encryption method}
MAC={message authentication code}
{export flag}

Note that this service does not encrypt traffic by default but does
support upgrading to an encrypted connection using STARTTLS.
```

21643 - SSL Cipher Suites Supported

Synopsis

The remote service encrypts communications using SSL.

Description

This plugin detects which SSL ciphers are supported by the remote service for encrypting communications.

See Also

https://www.openssl.org/docs/man1.0.2/man1/ciphers.html
http://www.nessus.org/u?e17ffced

Solution

n/a

Risk Factor

None

Plugin Information

Published: 2006/06/05, Modified: 2024/09/11

Plugin Output

tcp/5432/postgresql

```
Here is the list of SSL ciphers supported by the remote server :
Each group is reported per SSL Version.

SSL Version : TLSv1
Medium Strength Ciphers (> 64-bit and < 112-bit key, or 3DES)

Name Code KEX Auth Encryption MAC
--------------------- ---------- --- ---- --------------------- ---
EDH-RSA-DES-CBC3-SHA 0x00, 0x16 DH RSA 3DES-CBC(168) SHA1
DES-CBC3-SHA 0x00, 0x0A RSA RSA 3DES-CBC(168) SHA1

High Strength Ciphers (>= 112-bit key)

Name Code KEX Auth Encryption MAC
--------------------- ---------- --- ---- --------------------- ---
DHE-RSA-AES128-SHA 0x00, 0x33 DH RSA AES-CBC(128) SHA1
DHE-RSA-AES256-SHA 0x00, 0x39 DH RSA AES-CBC(256) SHA1
AES128-SHA 0x00, 0x2F RSA RSA AES-CBC(128) SHA1
AES256-SHA 0x00, 0x35 RSA RSA AES-CBC(256) SHA1
RC4-SHA 0x00, 0x05 RSA RSA RC4(128) SHA1


SSL Version : SSLv3
Medium Strength Ciphers (> 64-bit and < 112-bit key, or 3DES)

Name Code KEX Auth Encryption MAC
```

```
---------------------- ---------- --- ---- --------------------- ---
EDH-RSA-DES-CBC3-SHA 0x00, 0x16 DH RSA 3DES-CBC(168) SHA1
DES-CBC3-SHA 0x00, 0x0A RSA RSA 3DES-CBC(168) SHA1

High Strength Ciphers (>= 112-bit key)

Name Code KEX Auth Encryption MAC
---------------------- ---------- --- ---- --------------------- ---
DHE-RSA-AES128-SHA 0x00, 0x33 DH RSA AES-CBC(128) SHA1
DHE-RSA-AES256-SHA 0x00, 0x39 DH RSA AES-CBC(256) SHA1
AES128-SHA 0x00, 0x2F RSA RSA AES-CBC(128) SHA1
AES256-SHA 0x00, 0x35 RSA RSA AES-CBC(256) SHA1
RC4-SHA 0x00, 0x05 RSA RSA RC4(128) SHA1

The fields above are :

{Tenable ciphername}
{Cipher ID code}
Kex={key exchange}
Auth={authentication}
Encrypt={symmetric encryption method}
MAC={message authentication code}
{export flag}

Note that this service does not encrypt traffic by default but does
support upgrading to an encrypted connection using STARTTLS.
```

62563 - SSL Compression Methods Supported

Synopsis

The remote service supports one or more compression methods for SSL connections.

Description

This script detects which compression methods are supported by the remote service for SSL connections.

See Also

http://www.iana.org/assignments/comp-meth-ids/comp-meth-ids.xml
https://tools.ietf.org/html/rfc3749
https://tools.ietf.org/html/rfc3943
https://tools.ietf.org/html/rfc5246

Solution

n/a

Risk Factor

None

Plugin Information

Published: 2012/10/16, Modified: 2022/04/11

Plugin Output

tcp/25/smtp

```
Nessus was able to confirm that the following compression method is
supported by the target :

DEFLATE (0x01)
```

62563 - SSL Compression Methods Supported

Synopsis

The remote service supports one or more compression methods for SSL connections.

Description

This script detects which compression methods are supported by the remote service for SSL connections.

See Also

http://www.iana.org/assignments/comp-meth-ids/comp-meth-ids.xml
https://tools.ietf.org/html/rfc3749
https://tools.ietf.org/html/rfc3943
https://tools.ietf.org/html/rfc5246

Solution

n/a

Risk Factor

None

Plugin Information

Published: 2012/10/16, Modified: 2022/04/11

Plugin Output

tcp/5432/postgresql

```
Nessus was able to confirm that the following compression method is
supported by the target :

DEFLATE (0x01)
```

57041 - SSL Perfect Forward Secrecy Cipher Suites Supported -

Synopsis

The remote service supports the use of SSL Perfect Forward Secrecy ciphers, which maintain confidentiality even if the key is stolen.

Description

The remote host supports the use of SSL ciphers that offer Perfect Forward Secrecy (PFS) encryption. These cipher suites ensure that recorded SSL traffic cannot be broken at a future date if the server's private key is compromised.

See Also

https://www.openssl.org/docs/manmaster/man1/ciphers.html
https://en.wikipedia.org/wiki/Diffie-Hellman_key_exchange
https://en.wikipedia.org/wiki/Perfect_forward_secrecy

Solution

n/a

Risk Factor

None

Plugin Information

Published: 2011/12/07, Modified: 2021/03/09

Plugin Output

tcp/25/smtp

```
Here is the list of SSL PFS ciphers supported by the remote server :

Low Strength Ciphers (<= 64-bit key)

Name Code KEX Auth Encryption MAC
---------------------- ---------- --- ---- --------------------- ---
EXP-EDH-RSA-DES-CBC-SHA 0x00, 0x14 DH(512) RSA DES-CBC(40) SHA1 export
EDH-RSA-DES-CBC-SHA 0x00, 0x15 DH RSA DES-CBC(56) SHA1

Medium Strength Ciphers (> 64-bit and < 112-bit key, or 3DES)

Name Code KEX Auth Encryption MAC
---------------------- ---------- --- ---- --------------------- ---
EDH-RSA-DES-CBC3-SHA 0x00, 0x16 DH RSA 3DES-CBC(168) SHA1
```

```
High Strength Ciphers (>= 112-bit key)

Name Code KEX Auth Encryption MAC
---------------------- ---------- --- ---- --------------------- ---
DHE-RSA-AES128-SHA 0x00, 0x33 DH RSA AES-CBC(128) SHA1
DHE-RSA-AES256-SHA 0x00, 0x39 DH RSA AES-CBC(256) SHA1

The fields above are :

{Tenable ciphername}
{Cipher ID code}
Kex={key exchange}
Auth={authentication}
Encrypt={symmetric encryption method}
MAC={message authentication code}
{export flag}
```

57041 - SSL Perfect Forward Secrecy Cipher Suites Supported

Synopsis

The remote service supports the use of SSL Perfect Forward Secrecy ciphers, which maintain confidentiality even if the key is stolen.

Description

The remote host supports the use of SSL ciphers that offer Perfect Forward Secrecy (PFS) encryption. These cipher suites ensure that recorded SSL traffic cannot be broken at a future date if the server's private key is compromised.

See Also

https://www.openssl.org/docs/manmaster/man1/ciphers.html
https://en.wikipedia.org/wiki/Diffie-Hellman_key_exchange
https://en.wikipedia.org/wiki/Perfect_forward_secrecy

Solution

n/a

Risk Factor

None

Plugin Information

Published: 2011/12/07, Modified: 2021/03/09

Plugin Output

tcp/5432/postgresql

```
Here is the list of SSL PFS ciphers supported by the remote server :

Medium Strength Ciphers (> 64-bit and < 112-bit key, or 3DES)

Name Code KEX Auth Encryption MAC
---------------------- ---------- --- ---- --------------------- ---
EDH-RSA-DES-CBC3-SHA 0x00, 0x16 DH RSA 3DES-CBC(168) SHA1

High Strength Ciphers (>= 112-bit key)

Name Code KEX Auth Encryption MAC
---------------------- ---------- --- ---- --------------------- ---
DHE-RSA-AES128-SHA 0x00, 0x33 DH RSA AES-CBC(128) SHA1
DHE-RSA-AES256-SHA 0x00, 0x39 DH RSA AES-CBC(256) SHA1

The fields above are :

{Tenable ciphername}
{Cipher ID code}
Kex={key exchange}
Auth={authentication}
Encrypt={symmetric encryption method}
MAC={message authentication code}
{export flag}
```

51891 - SSL Session Resume Supported

Synopsis

The remote host allows resuming SSL sessions.

Description

This script detects whether a host allows resuming SSL sessions by performing a full SSL handshake to receive a session ID, and then reconnecting with the previously used session ID. If the server accepts the session ID in the second connection, the server maintains a cache of sessions that can be resumed.

Solution

n/a

Risk Factor

None

Plugin Information

Published: 2011/02/07, Modified: 2021/09/13

Plugin Output

tcp/25/smtp

```
This port supports resuming SSLv3 sessions.
```

156899 - SSL/TLS Recommended Cipher Suites -

Synopsis

The remote host advertises discouraged SSL/TLS ciphers.

Description

The remote host has open SSL/TLS ports which advertise discouraged cipher suites. It is recommended to only enable support for the following cipher suites:

TLSv1.3:
- 0x13,0x01 TLS13_AES_128_GCM_SHA256
- 0x13,0x02 TLS13_AES_256_GCM_SHA384
- 0x13,0x03 TLS13_CHACHA20_POLY1305_SHA256

TLSv1.2:
- 0xC0,0x2B ECDHE-ECDSA-AES128-GCM-SHA256
- 0xC0,0x2F ECDHE-RSA-AES128-GCM-SHA256
- 0xC0,0x2C ECDHE-ECDSA-AES256-GCM-SHA384
- 0xC0,0x30 ECDHE-RSA-AES256-GCM-SHA384
- 0xCC,0xA9 ECDHE-ECDSA-CHACHA20-POLY1305
- 0xCC,0xA8 ECDHE-RSA-CHACHA20-POLY1305

This is the recommended configuration for the vast majority of services, as it is highly secure and compatible with nearly every client released in the last five (or more) years.

See Also

https://wiki.mozilla.org/Security/Server_Side_TLS
https://ssl-config.mozilla.org/

Solution

Only enable support for recommened cipher suites.

Risk Factor

None

Plugin Information

Published: 2022/01/20, Modified: 2024/02/12

Plugin Output

tcp/25/smtp

```
The remote host has listening SSL/TLS ports which advertise the discouraged cipher suites outlined below:


Low Strength Ciphers (<= 64-bit key)

Name Code KEX Auth Encryption MAC
---------------------- ---------- --- ---- --------------------- ---
EXP-RC2-CBC-MD5 0x04, 0x00, 0x80 RSA(512) RSA RC2-CBC(40) MD5 export
EXP-RC4-MD5 0x02, 0x00, 0x80 RSA(512) RSA RC4(40) MD5 export
EXP-EDH-RSA-DES-CBC-SHA 0x00, 0x14 DH(512) RSA DES-CBC(40) SHA1 export
EDH-RSA-DES-CBC-SHA 0x00, 0x15 DH RSA DES-CBC(56) SHA1
EXP-ADH-DES-CBC-SHA 0x00, 0x19 DH(512) None DES-CBC(40) SHA1 export
EXP-ADH-RC4-MD5 0x00, 0x17 DH(512) None RC4(40) MD5 export
ADH-DES-CBC-SHA 0x00, 0x1A DH None DES-CBC(56) SHA1
EXP-DES-CBC-SHA 0x00, 0x08 RSA(512) RSA DES-CBC(40) SHA1 export
EXP-RC2-CBC-MD5 0x00, 0x06 RSA(512) RSA RC2-CBC(40) MD5 export
EXP-RC4-MD5 0x00, 0x03 RSA(512) RSA RC4(40) MD5 export
DES-CBC-SHA 0x00, 0x09 RSA RSA DES-CBC(56) SHA1

Medium Strength Ciphers (> 64-bit and < 112-bit key, or 3DES)

Name Code KEX Auth Encryption MAC
---------------------- ---------- --- ---- --------------------- ---
DES-CBC3-MD5 0x07, 0x00, 0xC0 RSA RSA 3DES-CBC(168) MD5
EDH-RSA-DES-CBC3-SHA 0x00, 0x16 DH RSA 3DES-CBC(168) SHA1
ADH-DES-CBC3-SHA 0x00, 0x1B DH None 3DES-CBC(168) SHA1
DES-CBC3-SHA 0x00, 0x0A RSA RSA 3DES-CBC(168) SHA1

High Strength Ciphers (>= 112-bit key)

Name Code KEX Auth Encryption MAC
---------------------- ---------- --- ---- --------------------- ---
RC4-MD5 0x01, 0x00, 0x80 RSA RSA RC4(128) MD5
DHE-RSA-AES128-SHA 0x00, 0x33 DH RSA AES-CBC(128) SHA1
DHE-RSA-AES256-SHA 0x00, 0x39 DH RSA AES-CBC(256) SHA1
ADH-AES128-SHA 0x00, 0x34 DH None AES-CBC(128) SHA1
ADH-AES256-SHA 0x00, 0x3A DH None AES-CBC(256) SHA1
ADH-RC4-MD5 0x00, 0x18 DH None RC4(128) MD5
AES128-SHA 0x00, 0x2F RSA RSA AES-CBC(128) SHA1
AES256-SHA 0x00, 0x35 RSA RSA AES-CBC(256) SHA1
RC4-MD5 0x00, 0x04 RSA RSA RC4(128) MD5
RC4-SHA 0x00, 0x05 RSA RSA RC4(128) SHA1

The fields above are :

{Tenable ciphername}
{Cipher ID code}
Kex={key exchange}
Auth={authentication}
Encrypt={symmetric encryption method}
MAC={message authentication code}
{export flag}
```

156899 - SSL/TLS Recommended Cipher Suites -

Synopsis

The remote host advertises discouraged SSL/TLS ciphers.

Description

The remote host has open SSL/TLS ports which advertise discouraged cipher suites. It is recommended to only enable support for the following cipher suites:

TLSv1.3:
- 0x13,0x01 TLS13_AES_128_GCM_SHA256
- 0x13,0x02 TLS13_AES_256_GCM_SHA384
- 0x13,0x03 TLS13_CHACHA20_POLY1305_SHA256

TLSv1.2:
- 0xC0,0x2B ECDHE-ECDSA-AES128-GCM-SHA256
- 0xC0,0x2F ECDHE-RSA-AES128-GCM-SHA256
- 0xC0,0x2C ECDHE-ECDSA-AES256-GCM-SHA384
- 0xC0,0x30 ECDHE-RSA-AES256-GCM-SHA384
- 0xCC,0xA9 ECDHE-ECDSA-CHACHA20-POLY1305
- 0xCC,0xA8 ECDHE-RSA-CHACHA20-POLY1305

This is the recommended configuration for the vast majority of services, as it is highly secure and compatible with nearly every client released in the last five (or more) years.

See Also

https://wiki.mozilla.org/Security/Server_Side_TLS
https://ssl-config.mozilla.org/

Solution

Only enable support for recommened cipher suites.

Risk Factor

None

Plugin Information

Published: 2022/01/20, Modified: 2024/02/12

Plugin Output

tcp/5432/postgresql

```
The remote host has listening SSL/TLS ports which advertise the discouraged cipher suites outlined below:


Medium Strength Ciphers (> 64-bit and < 112-bit key, or 3DES)

Name Code KEX Auth Encryption MAC
---------------------- ---------- --- ---- --------------------- ---
EDH-RSA-DES-CBC3-SHA 0x00, 0x16 DH RSA 3DES-CBC(168) SHA1
DES-CBC3-SHA 0x00, 0x0A RSA RSA 3DES-CBC(168) SHA1

High Strength Ciphers (>= 112-bit key)

Name Code KEX Auth Encryption MAC
---------------------- ---------- --- ---- --------------------- ---
DHE-RSA-AES128-SHA 0x00, 0x33 DH RSA AES-CBC(128) SHA1
DHE-RSA-AES256-SHA 0x00, 0x39 DH RSA AES-CBC(256) SHA1
AES128-SHA 0x00, 0x2F RSA RSA AES-CBC(128) SHA1
AES256-SHA 0x00, 0x35 RSA RSA AES-CBC(256) SHA1
RC4-SHA 0x00, 0x05 RSA RSA RC4(128) SHA1

The fields above are :

{Tenable ciphername}
{Cipher ID code}
Kex={key exchange}
Auth={authentication}
Encrypt={symmetric encryption method}
MAC={message authentication code}
{export flag}
```

25240 - Samba Server Detection

Synopsis

An SMB server is running on the remote host.

Description

The remote host is running Samba, a CIFS/SMB server for Linux and Unix.

See Also

https://www.samba.org/

Solution

n/a

Risk Factor

None

Plugin Information

Published: 2007/05/16, Modified: 2022/10/12

Plugin Output

tcp/445/cifs

104887 - Samba Version

Synopsis

It was possible to obtain the samba version from the remote operating system.

Description

Nessus was able to obtain the samba version from the remote operating by sending an authentication request to port 139 or 445. Note that this plugin requires SMB1 to be enabled on the host.

Solution

n/a

Risk Factor

None

Plugin Information

Published: 2017/11/30, Modified: 2019/11/22

Plugin Output

tcp/445/cifs

```
The remote Samba Version is : Samba 3.0.20-Debian
```

96982 - Server Message Block (SMB) Protocol Version 1 Enabled (uncredentialed check)

Synopsis

The remote Windows host supports the SMBv1 protocol.

Description

The remote Windows host supports Server Message Block Protocol version 1 (SMBv1). Microsoft recommends that users discontinue the use of SMBv1 due to the lack of security features that were included in later SMB versions. Additionally, the Shadow Brokers group reportedly has an exploit that affects SMB; however, it is unknown if the exploit affects SMBv1 or another version. In response to this, US-CERT recommends that users disable SMBv1 per SMB best practices to mitigate these potential issues.

See Also

https://blogs.technet.microsoft.com/filecab/2016/09/16/stop-using-smb1/
https://support.microsoft.com/en-us/help/2696547/how-to-detect-enable-and-disable-smbv1-smbv2-and-smbv3-in-windows-and
http://www.nessus.org/u?8dcab5e4
http://www.nessus.org/u?234f8ef8
http://www.nessus.org/u?4c7e0cf3

Solution

Disable SMBv1 according to the vendor instructions in Microsoft KB2696547. Additionally, block SMB directly by blocking TCP port 445 on all network boundary devices. For SMB over the NetBIOS API, block TCP ports 137 / 139 and UDP ports 137 / 138 on all network boundary devices.

Risk Factor

None

References

XREF IAVT:0001-T-0710

Plugin Information

Published: 2017/02/03, Modified: 2020/09/22

Plugin Output

tcp/445/cifs

```
The remote host supports SMBv1.
```

22964 - Service Detection

Synopsis

The remote service could be identified.

Description

Nessus was able to identify the remote service by its banner or by looking at the error message it sends when it receives an HTTP request.

Solution

n/a

Risk Factor

None

Plugin Information

Published: 2007/08/19, Modified: 2024/03/26

Plugin Output

tcp/21/ftp

```
An FTP server is running on this port.
```

22964 - Service Detection

Synopsis

The remote service could be identified.

Description

Nessus was able to identify the remote service by its banner or by looking at the error message it sends when it receives an HTTP request.

Solution

n/a

Risk Factor

None

Plugin Information

Published: 2007/08/19, Modified: 2024/03/26

Plugin Output

tcp/22/ssh

```
An SSH server is running on this port.
```

22964 - Service Detection

Synopsis

The remote service could be identified.

Description

Nessus was able to identify the remote service by its banner or by looking at the error message it sends when it receives an HTTP request.

Solution

n/a

Risk Factor

None

Plugin Information

Published: 2007/08/19, Modified: 2024/03/26

Plugin Output

tcp/25/smtp

```
An SMTP server is running on this port.
```

22964 - Service Detection

Synopsis

The remote service could be identified.

Description

Nessus was able to identify the remote service by its banner or by looking at the error message it sends when it receives an HTTP request.

Solution

n/a

Risk Factor

None

Plugin Information

Published: 2007/08/19, Modified: 2024/03/26

Plugin Output

tcp/80/www

```
A web server is running on this port.
```

22964 - Service Detection

Synopsis

The remote service could be identified.

Description

Nessus was able to identify the remote service by its banner or by looking at the error message it sends when it receives an HTTP request.

Solution

n/a

Risk Factor

None

Plugin Information

Published: 2007/08/19, Modified: 2024/03/26

Plugin Output

tcp/1524/wild_shell

```
A shell server (Metasploitable) is running on this port.
```

22964 - Service Detection

Synopsis

The remote service could be identified.

Description

Nessus was able to identify the remote service by its banner or by looking at the error message it sends when it receives an HTTP request.

Solution

n/a

Risk Factor

None

Plugin Information

Published: 2007/08/19, Modified: 2024/03/26

Plugin Output

tcp/5900/vnc

```
A vnc server is running on this port.
```

110723 - Target Credential Status by Authentication Protocol - No Credentials Provided

Synopsis

Nessus was able to find common ports used for local checks, however, no credentials were provided in the scan policy.

Description

Nessus was not able to successfully authenticate directly to the remote target on an available authentication protocol. Nessus was able to connect to the remote port and identify that the service running on the port supports an authentication protocol, but Nessus failed to authenticate to the remote service using the provided credentials. There may have been a protocol failure that prevented authentication from being attempted or all of the provided credentials for the authentication protocol may be invalid. See plugin output for error details.

Please note the following :

- This plugin reports per protocol, so it is possible for valid credentials to be provided for one protocol and not another. For example, authentication may succeed via SSH but fail via SMB, while no credentials were provided for an available SNMP service.

- Providing valid credentials for all available authentication protocols may improve scan coverage, but the value of successful authentication for a given protocol may vary from target to target depending upon what data (if any) is gathered from the target via that protocol. For example, successful authentication via SSH is more valuable for Linux targets than for Windows targets, and likewise successful authentication via SMB is more valuable for Windows targets than for Linux targets.

Solution

n/a

Risk Factor

None

References

XREF	IAVB:0001-B-0504

Plugin Information

Published: 2018/06/27, Modified: 2024/04/19

Plugin Output

tcp/0

```
SSH was detected on port 22 but no credentials were provided.
SSH local checks were not enabled.
```

10287 - Traceroute Information

Synopsis

It was possible to obtain traceroute information.

Description

Makes a traceroute to the remote host.

Solution

n/a

Risk Factor

None

Plugin Information

Published: 1999/11/27, Modified: 2023/12/04

Plugin Output

udp/0

```
For your information, here is the traceroute from 192.168.62.130 to 192.168.150.130 :
192.168.62.130
192.168.62.2
192.168.150.130

Hop Count: 2
```

11154 - Unknown Service Detection: Banner Retrieval -

Synopsis

There is an unknown service running on the remote host.

Description

Nessus was unable to identify a service on the remote host even though it returned a banner of some type.

Solution

n/a

Risk Factor

None

Plugin Information

Published: 2002/11/18, Modified: 2022/07/26

Plugin Output

tcp/512

```
If you know what this service is and think the banner could be used to
identify it, please send a description of the service along with the
following output to svc-signatures@nessus.org :

Port : 512
Type : spontaneous
Banner :
0x00: 01 57 68 65 72 65 20 61 72 65 20 79 6F 75 3F 0A .Where are you?.
0x10:
```

11154 - Unknown Service Detection: Banner Retrieval -

Synopsis

There is an unknown service running on the remote host.

Description

Nessus was unable to identify a service on the remote host even though it returned a banner of some type.

Solution

n/a

Risk Factor

None

Plugin Information

Published: 2002/11/18, Modified: 2022/07/26

Plugin Output

tcp/6667

```
If you know what this service is and think the banner could be used to
identify it, please send a description of the service along with the
following output to svc-signatures@nessus.org :

Port : 6667
Type : spontaneous
Banner :
0x00: 45 52 52 4F 52 20 3A 43 6C 6F 73 69 6E 67 20 4C ERROR :Closing L
0x10: 69 6E 6B 3A 20 5B 31 39 32 2E 31 36 38 2E 31 35 ink: [192.168.15
0x20: 30 2E 31 5D 20 28 54 6F 6F 20 6D 61 6E 79 20 75 0.1] (Too many u
0x30: 6E 6B 6E 6F 77 6E 20 63 6F 6E 6E 65 63 74 69 6F nknown connectio
0x40: 6E 73 20 66 72 6F 6D 20 79 6F 75 72 20 49 50 29 ns from your IP)
0x50: 0D 0A ..
```

11154 - Unknown Service Detection: Banner Retrieval

Synopsis

There is an unknown service running on the remote host.

Description

Nessus was unable to identify a service on the remote host even though it returned a banner of some type.

Solution

n/a

Risk Factor

None

Plugin Information

Published: 2002/11/18, Modified: 2022/07/26

Plugin Output

tcp/8787

```
If you know what this service is and think the banner could be used to
identify it, please send a description of the service along with the
following output to svc-signatures@nessus.org :

Port : 8787
Type : get_http
Banner :
0x0000: 00 00 00 03 04 08 46 00 00 03 A1 04 08 6F 3A 16 ......F......o:.
0x0010: 44 52 62 3A 3A 44 52 62 43 6F 6E 6E 45 72 72 6F DRb::DRbConnErro
0x0020: 72 07 3A 07 62 74 5B 17 22 2F 2F 75 73 72 2F 6C r.:.bt[."//usr/l
0x0030: 69 62 2F 72 75 62 79 2F 31 2E 38 2F 64 72 62 2F ib/ruby/1.8/drb/
0x0040: 64 72 62 2E 72 62 3A 35 37 33 3A 69 6E 20 60 6C drb.rb:573:in `l
0x0050: 6F 61 64 27 22 37 2F 75 73 72 2F 6C 69 62 2F 72 oad'"7/usr/lib/r
0x0060: 75 62 79 2F 31 2E 38 2F 64 72 62 2F 64 72 62 2E uby/1.8/drb/drb.
0x0070: 72 62 3A 36 31 32 3A 69 6E 20 60 72 65 63 76 5F rb:612:in `recv_
0x0080: 72 65 71 75 65 73 74 27 22 37 2F 75 73 72 2F 6C request'"7/usr/l
0x0090: 69 62 2F 72 75 62 79 2F 31 2E 38 2F 64 72 62 2F ib/ruby/1.8/drb/
0x00A0: 64 72 62 2E 72 62 3A 39 31 31 3A 69 6E 20 60 72 drb.rb:911:in `r
0x00B0: 65 63 76 5F 72 65 71 75 65 73 74 27 22 3C 2F 75 ecv_request'"</u
0x00C0: 73 72 2F 6C 69 62 2F 72 75 62 79 2F 31 2E 38 2F sr/lib/ruby/1.8/
0x00D0: 64 72 62 2F 64 72 62 2E 72 62 3A 31 35 33 30 3A drb/drb.rb:1530:
0x00E0: 69 6E 20 60 69 6E 69 74 5F 77 69 74 68 5F 63 6C in `init_with_cl
0x00F0: 69 65 6E 74 27 22 39 2F 75 73 72 2F 6C 69 62 2F ient'"9/usr/lib/
0x0100: 72 75 62 79 2F 31 2E 38 2F 64 72 62 2F 64 72 62 ruby/1.8/drb/drb
```

```
0x0110: 2E 72 62 3A 31 35 34 32 3A 69 6E 20 60 73 65 74 .rb:1542:in `set
0x0120: 75 70 5F 6D 65 73 73 61 67 65 27 22 33 2F 75 73 up_message'"3/us
0x0130: 72 2F 6C 69 62 2F 72 75 62 79 2F 31 2E 38 2F 64 r/lib/ruby/1.8/d
0x0140: 72 62 2F 64 72 62 2E 72 62 3A 31 34 39 34 3A 69 rb/drb.rb:1494:i
0x0150: 6E 20 60 70 65 72 66 6F 72 6D 27 22 35 2F 75 73 n `perform'"5/us
0x0160: 72 2F 6C 69 62 2F 72 75 62 79 2F 31 2E 38 2F 64 r/lib/ruby/1.8/d
0x0170: 72 62 2F 64 72 62 2E 72 62 3A 31 35 38 39 3A 69 rb/drb.rb:1589:i
0x0180: 6E 20 60 6D 61 69 6E 5F 6C 6F 6F 70 27 22 30 2F n `main_loop'"0/
0x0190: 75 73 72 2F 6C 69 62 2F 72 75 62 79 2F 31 2E 38 usr/lib/ruby/1.8
0x01A0: 2F 64 72 62 2F 64 72 62 2E 72 62 3A 31 35 38 35 /drb/drb.rb:1585
0x01B0: 3A 69 6E 20 60 6C 6F 6F 70 27 22 35 2F 75 73 72 :in `loop'"5/usr
0x01C0: 2F 6C 69 62 2F 72 75 62 79 2F 31 2E 38 2F 64 72 /lib/ruby/1.8/dr
0x01D0: 62 2F 64 72 62 2E 72 62 3A 31 35 38 35 3A 69 6E b/drb.rb:1585:in
0x01E0: 20 60 6D 61 69 6E 5F 6C 6F 6F 70 27 22 31 2F 75 `main_loop'"1/u
0x01F0: 73 72 2F 6C 69 62 2F 72 75 62 79 2F 31 2E 38 2F sr/lib/ruby/1.8/
0x0200: 64 72 62 2F 64 72 62 2E 72 62 3A 31 35 38 31 3A drb/drb.rb:1581:
0x0210: 69 6E 20 60 73 74 61 72 74 27 22 35 2F 75 73 72 in `start'"5/usr
0x0220: 2F 6C 69 62 2F 72 75 62 79 2F 31 2E 38 2F 64 72 /lib/ruby/1.8/dr
0x0230: 62 2F 64 72 62 2E 72 62 3A 31 35 38 31 3A 69 6E b/drb.rb:1581:in
0x0240: 20 60 6D 61 69 6E 5F 6C 6F 6F 70 27 22 2F 2F 75 `main_loop'"//u
0x0250: 73 72 2F 6C 69 62 2F 72 75 62 79 2F 31 2E 38 2F sr/lib/ruby/1.8/
0x0260: 64 72 62 2F 64 72 62 2E 72 62 3A 31 34 33 30 3A drb/drb.rb:1430:
0x0270: 69 6E 20 60 72 75 6E 27 22 31 2F 75 73 72 2F 6C in `run'"1/usr/l
0x0280: 69 62 2F 72 75 62 79 2F 31 2E 38 2F 64 72 62 2F ib/ruby/1.8/drb/
0x0290: 64 72 62 2E 72 62 3A 31 34 32 37 3A 69 6E 20 60 drb.rb:1427:in `
0x02A0: 73 74 61 72 74 27 22 2F 2F 75 73 72 2F 6C 69 62 start'"//usr/lib
0x02B0: 2F 72 75 62 79 2F 31 2E 38 2F 64 72 62 2F 64 72 /ruby/1.8/drb/dr
0x02C0: 62 2E 72 62 3A 31 34 32 37 3A 69 6E 20 60 72 75 b.rb:1427:in `ru
0x02D0: 6E 27 22 36 2F 75 73 72 2F 6C 69 62 2F 72 75 62 n'"6/usr/lib/rub
0x02E0: 79 2F 31 2E 38 2F 64 72 62 2F 64 72 62 2E 72 62 y/1.8/drb/drb.rb
0x02F0: 3A 31 33 34 37 3A 69 6E 20 60 69 6E 69 74 69 61 :1347:in `initia
0x0300: 6C 69 7A 65 27 22 2F 2F 75 73 72 2F 6C 69 62 2F lize'"//usr/lib/
0x0310: 72 75 62 79 2F 31 2E 38 2F 64 72 62 2F 64 72 62 ruby/1.8/drb/drb
0x0320: 2E 72 62 3A 31 36 32 37 3A 69 6E 20 60 6E 65 77 .rb:1627:in `new
0x0330: 27 22 39 2F 75 73 72 2F 6C 69 62 2F 72 75 62 79 '"9/usr/lib/ruby
0x0340: 2F 31 2E 38 2F 64 72 62 2F 64 72 62 2E 72 62 3A /1.8/drb/drb.rb:
0x0350: 31 36 32 37 3A 69 6E 20 60 73 74 61 72 74 5F 73 1627:in `start_s
0x0360: 65 72 76 69 63 65 27 22 25 2F 75 73 72 2F 73 62 ervice'"%/usr/sb
0x0370: 69 6E 2F 64 72 75 62 79 5F 74 69 6D 65 73 65 72 in/druby_timeser
0x0380: 76 65 72 2E 72 62 3A 31 32 3A 09 6D 65 73 67 22 ver.rb:12:.mesg"
0x0390: 20 74 6F 6F 20 6C 61 72 67 65 20 70 61 63 6B 65 too large packe
0x03A0: 74 20 31 31 39 35 37 32 35 38 35 36 t 1195725856
```

19288 - VNC Server Security Type Detection

Synopsis

A VNC server is running on the remote host.

Description

This script checks the remote VNC server protocol version and the available 'security types'.

Solution

n/a

Risk Factor

None

Plugin Information

Published: 2005/07/22, Modified: 2021/07/13

Plugin Output

tcp/5900/vnc

```
\nThe remote VNC server chose security type #2 (VNC authentication)
```

65792 - VNC Server Unencrypted Communication Detection

Synopsis

A VNC server with one or more unencrypted 'security-types' is running on the remote host.

Description

This script checks the remote VNC server protocol version and the available 'security types' to determine if any unencrypted 'security-types' are in use or available.

Solution

n/a

Risk Factor

None

Plugin Information

Published: 2013/04/03, Modified: 2014/03/12

Plugin Output

tcp/5900/vnc

```
The remote VNC server supports the following security type
which does not perform full data communication encryption :

2 (VNC authentication)
```

10342 - VNC Software Detection

Synopsis

The remote host is running a remote display software (VNC).

Description

The remote host is running VNC (Virtual Network Computing), which uses the RFB (Remote Framebuffer) protocol to provide remote access to graphical user interfaces and thus permits a console on the remote host to be displayed on another.

See Also

https://en.wikipedia.org/wiki/Vnc

Solution

Make sure use of this software is done in accordance with your organization's security policy and filter incoming traffic to this port.

Risk Factor

None

Plugin Information

Published: 2000/03/07, Modified: 2017/06/12

Plugin Output

tcp/5900/vnc

```
The highest RFB protocol version supported by the server is :

3.3
```

135860 - WMI Not Available

Synopsis

WMI queries could not be made against the remote host.

Description

WMI (Windows Management Instrumentation) is not available on the remote host over DCOM. WMI queries are used to gather information about the remote host, such as its current state, network interface configuration, etc.

Without this information Nessus may not be able to identify installed software or security vunerabilities that exist on the remote host.

See Also

https://docs.microsoft.com/en-us/windows/win32/wmisdk/wmi-start-page

Solution

n/a

Risk Factor

None

Plugin Information

Published: 2020/04/21, Modified: 2024/11/22

Plugin Output

tcp/445/cifs

```
Can't connect to the 'root\CIMV2' WMI namespace.
```

11424 - WebDAV Detection

Synopsis

The remote server is running with WebDAV enabled.

Description

WebDAV is an industry standard extension to the HTTP specification.
It adds a capability for authorized users to remotely add and manage the content of a web server.

If you do not use this extension, you should disable it.

Solution

http://support.microsoft.com/default.aspx?kbid=241520

Risk Factor

None

Plugin Information

Published: 2003/03/20, Modified: 2011/03/14

Plugin Output

tcp/80/www

10150 - Windows NetBIOS / SMB Remote Host Information Disclosure

Synopsis

It was possible to obtain the network name of the remote host.

Description

The remote host is listening on UDP port 137 or TCP port 445, and replies to NetBIOS nbtscan or SMB requests.

Note that this plugin gathers information to be used in other plugins, but does not itself generate a report.

Solution

n/a

Risk Factor

None

Plugin Information

Published: 1999/10/12, Modified: 2021/02/10

Plugin Output

udp/137/netbios-ns

```
The following 7 NetBIOS names have been gathered :

METASPLOITABLE = Computer name
METASPLOITABLE = Messenger Service
METASPLOITABLE = File Server Service
  MSBROWSE    = Master Browser
WORKGROUP = Workgroup / Domain name
WORKGROUP = Master Browser
WORKGROUP = Browser Service Elections

This SMB server seems to be a Samba server - its MAC address is NULL.
```

52703 - vsftpd Detection -

Synopsis

An FTP server is listening on the remote port.

Description

The remote host is running vsftpd, an FTP server for UNIX-like systems written in C.

See Also

http://vsftpd.beasts.org/

Solution

n/a

Risk Factor

None

Plugin Information

Published: 2011/03/17, Modified: 2019/11/22

Plugin Output

tcp/21/ftp

```
Source : 220 (vsFTPd 2.3.4)
Version : 2.3.4
```

2.6 Conclusion and Recommendations

2.6.1 Project Summary

In this project, we explored the comprehensive capabilities of Nessus, a leading vulnerability scanning tool, alongside the Metasploitable2 virtual machine, which serves as a perfect platform for security testing and learning. We began by introducing Nessus and its history, key features, and installation processes, then proceeded to set up Metasploitable2 as a vulnerable target for scanning. Throughout this process, we demonstrated how to configure Nessus for effective vulnerability scanning, run scans, and interpret the results.

By performing a vulnerability scan on Metasploitable2, we gained a practical understanding of how to identify and assess various vulnerabilities within an environment. This experience reinforced the importance of regularly scanning systems for weaknesses to proactively address security risks and ensure robust protection against potential cyberattacks.

2.6.2 Evaluation of Objectives

The project objectives were successfully met by thoroughly investigating Nessus's scanning capabilities and demonstrating the practical application of vulnerability management using Metasploitable2. By walking through each phase—from installation to scanning and report analysis—this project provided a hands-on approach to learning vulnerability scanning techniques, making it an invaluable educational tool. Additionally, by integrating Metasploitable2 as a realistic target environment, the project allowed us to observe common vulnerabilities in action, further enhancing our understanding of real-world security issues.

One key objective was to highlight the critical importance of vulnerability scanning and remediation. The detailed Nessus scan report attached within this project serves as an example of a real-world vulnerability report, with insights into the vulnerabilities found in Metasploitable2. This example illustrates the depth of Nessus's reporting capabilities, which categorize vulnerabilities by severity and provide actionable recommendations for remediation.

2.6.3 Future Work

While this project has provided a solid foundation for understanding vulnerability scanning, there are several areas where future work can further enhance the learning experience. Some potential future initiatives include:

- *Integration of Nessus with Other Security Tools: Expanding the project to integrate Nessus with other security solutions, such as Metasploit for*

exploitation, would help users understand how vulnerabilities are actively exploited and what steps can be taken to defend against such attacks.

- *Advanced Scanning Techniques: Introducing advanced scan configurations and policies, such as credentialed scans, to identify more in-depth vulnerabilities like those hidden within operating systems or applications, would further enhance the accuracy of vulnerability detection.*
- *Real-Time Threat Monitoring: Implementing continuous scanning and real-time vulnerability monitoring could be explored to simulate a more proactive approach to threat detection in an enterprise environment.*
- *Cloud and Hybrid Infrastructure: Extending the scanning capabilities to include cloud environments (such as AWS or Azure) or hybrid systems, where critical vulnerabilities may reside, would reflect the evolving nature of IT infrastructure in modern organizations.*

Real-World Example: An organization managing both on-premises and cloud infrastructure could greatly benefit from integrating Nessus with cloud security platforms, ensuring vulnerabilities are detected and mitigated across all environments seamlessly.

Final Thoughts

This project successfully demonstrated how Nessus can be used to identify and mitigate vulnerabilities in a controlled environment using Metasploitable2 as a target. The attached Nessus report offers a practical, real-world example of how vulnerability scanners can help organizations stay ahead of threats by proactively identifying weaknesses. By analyzing this report, users can gain valuable insights into how vulnerabilities are categorized, prioritized, and addressed. Moving forward, the techniques and best practices learned from this project can be applied to a wide range of environments, empowering professionals to improve system security, protect critical assets, and minimize the risk of exploitation.

3. Enhancing Security Posture with IBM QRadar: A Comprehensive Approach

---

Executive Summary

This report serves as a comprehensive guide for the design and implementation of an advanced security monitoring and threat detection system leveraging IBM QRadar. Recognized for its robust capabilities, IBM QRadar provides a unified platform to address the organization's critical security challenges through real-time monitoring, in-depth analytics, and compliance enforcement.

The project's primary objective is to enhance the organization's security posture by integrating a diverse range of data sources, including firewalls, endpoint security solutions, network traffic logs, and cloud platforms. By harmonizing these inputs, QRadar enables the creation of tailored detection rules that effectively address unique organizational threats such as ransomware attacks, insider threats, and advanced persistent threats (APTs). QRadar's built-in threat intelligence feeds further strengthen its detection capabilities by correlating global threat data to prioritize responses to emerging risks.

To maximize efficiency, the project emphasizes seamless integration with Security Orchestration, Automation, and Response (SOAR) platforms. This allows automated incident responses, such as isolating compromised systems and notifying stakeholders in real time. QRadar's robust reporting and visualization features are leveraged to generate actionable insights, enabling decision-makers to identify trends, mitigate vulnerabilities, and improve long-term security strategies.

Compliance is a critical component of the initiative. The solution is designed to meet stringent regulatory requirements such as GDPR, HIPAA, and PCI DSS. By automating compliance reporting and maintaining audit trails, the project not only simplifies regulatory adherence but also bolsters transparency and accountability within the organization.

Through rigorous testing, validation, and user training, this project aims to establish a resilient, adaptive, and proactive security infrastructure. It addresses both immediate security concerns and positions the organization to respond effectively to an ever-evolving cyber threat landscape. Ultimately, the implementation of IBM QRadar empowers the organization to protect its assets, maintain stakeholder trust, and achieve sustained operational excellence.

3.1. Project Goal

The primary goal of this project is to design and implement a highly effective and comprehensive security monitoring and threat detection solution using IBM QRadar. This implementation aims to transform the organization's security posture by improving its ability to proactively detect, prevent, and respond to a wide spectrum of security threats. By leveraging IBM QRadar's advanced capabilities, the project seeks to integrate data from multiple sources such as firewalls, intrusion detection systems (IDS), endpoint security solutions, and cloud logs into a unified system. This integration will enable the organization to gain real-time visibility into potential threats, conduct sophisticated threat analytics, and implement automation for swift responses. Additionally, the project will focus on reducing false positives, ensuring compliance with global regulations like GDPR, PCI DSS, and HIPAA, and generating actionable insights through intuitive dashboards and reports. The ultimate objective is to build a resilient, adaptive, and proactive security environment that protects critical assets, mitigates risks, and enhances operational efficiency in response to the dynamic and evolving cyber threat landscape.

3.2. Project Scope

3.2.1 Data Ingestion

Objective:

Integrate diverse data sources into IBM QRadar to enable comprehensive, efficient, and secure processing of security event data, ensuring the foundation for effective threat detection and analysis.

Activities:

Identify and Integrate Data Sources:

Comprehensive IT Infrastructure Assessment: Conduct a detailed and systematic audit of the organization's IT assets to identify and classify all relevant data sources essential for robust security monitoring. This assessment ensures that no critical system or application is overlooked. Key categories include:

Network Security Devices: Integrate logs from firewalls, intrusion detection/prevention systems (IDS/IPS), VPN solutions, and proxy servers to monitor traffic anomalies and access violations.

Endpoint Security Solutions: Collect data from antivirus platforms, endpoint detection and response (EDR) tools, and mobile device management (MDM) systems to identify compromised devices or unauthorized actions.

Application Security: Incorporate logs from web application firewalls (WAF), database activity monitoring solutions, and server applications to detect SQL injections, unauthorized data access, or privilege escalations.

Cloud Environments: Include log sources from cloud platforms such as AWS CloudTrail, Azure Monitor, Google Cloud Operations Suite, and various SaaS applications to secure hybrid and multi-cloud architectures.

Network Traffic Analysis: Utilize flow logs from network routers and switches, and integrate packet capture systems to perform detailed analysis of east-west and north-south network traffic.

Secure and Reliable Data Ingestion: Implement secure transmission protocols and validated methods for data ingestion to ensure the reliability and confidentiality of collected data. Supported methods in QRadar include Syslog for streaming event logs, Log File Agents for local file collection, and REST APIs for dynamic interactions with third-party services.

Implement redundancy mechanisms for high availability of log transmission and storage, ensuring continuity during network disruptions.

Example in Practice: A global manufacturing enterprise integrates logs from geographically dispersed Cisco ASA firewalls, Palo Alto intrusion prevention systems, and Microsoft Azure event logs. This integration consolidates security events into a unified QRadar instance, enabling a comprehensive, real-time view of threats across both their on-premises and cloud environments. This single-pane-of-glass view drastically reduces blind spots, accelerates threat identification, and enhances coordinated response strategies.

Configure Data Collection Rules and Normalization:

Customization of Parsing Rules: Develop and implement parsing rules to transform raw, unstructured log data into standardized, normalized formats that are compatible with QRadar's processing engine. This normalization enables effective analysis and correlation of data across multiple sources.

Example: Parsing firewall logs to extract key information such as source and destination IPs, ports, and protocol details, ensuring QRadar can identify patterns indicative of potential security threats.

Filtering and Prioritization Mechanisms: Establish advanced filtering rules to minimize noise by excluding non-critical or redundant logs, while prioritizing logs that contain high-risk or anomalous activity. This approach ensures that the system's resources focus on the most relevant data.

Scenario: Implementing a rule to exclude routine traffic logs from internal subnets while flagging external login attempts as critical.

Scenario: A financial institution manages data from multiple firewall vendors, such as Palo Alto Networks and Cisco ASA, by standardizing log formats. This enables QRadar to seamlessly correlate events, identify multi-vector attacks, and reduce response times by presenting consistent data insights.

Enhancing Data Quality: Regularly audit and update parsing and filtering rules to adapt to changes in log formats or new data sources. This proactive management ensures continued accuracy and relevancy of ingested data.

Example: Adjusting parsing templates after a system upgrade introduces additional log fields in IDS alerts, ensuring these are captured and utilized effectively within QRadar.

Ensure Data Integrity and Security:

Encrypt log data during transmission to protect against eavesdropping and tampering.

Regularly validate the integrity of ingested data to maintain the reliability of the analytics.

Example: By encrypting log transmissions using TLS, a healthcare provider ensures compliance with HIPAA regulations while safeguarding patient information.

3.2.2 Threat Detection and Response

Objective: Detect advanced threats and automate response mechanisms to mitigate risks, ensuring proactive and efficient security management by employing cutting-edge technologies and strategic integrations.

Activities:

1. ***Develop and Deploy Custom Rules and Use Cases:***
    - *Develop detection rules tailored to unique organizational requirements. These rules should consider the specific operational environment, industry compliance needs, and potential threat vectors.*
    - *Define and test use cases that mirror real-world scenarios to ensure robust security coverage. For example, rules can target behavioral anomalies, such as multiple failed login attempts or sudden surges in file access volumes.*
    - *Example: Healthcare organizations can configure rules to detect attempts to access patient data during non-working hours. For instance, flagging login attempts from unrecognized devices targeting sensitive medical records at unusual times.*

- *Real-World Application: A multinational retail company uses QRadar to identify irregular access to payment processing systems, particularly from IP addresses outside their operational territories. This ensures early detection of potential payment fraud.*
- *Additional Detail: QRadar's integration with data sources like Active Directory and DNS logs can enhance the granularity of these rules by linking user accounts to suspicious activities.*

2. ***Leverage Threat Intelligence Feeds:***
    - *Leverage QRadar's threat intelligence capabilities to prioritize and respond to emerging threats effectively. This includes analyzing feeds for known indicators of compromise (IOCs), such as malicious domains, phishing URLs, or malware hashes.*
    - *Enhance response by incorporating third-party intelligence feeds, which provide a broader view of global threat landscapes.*
    - *Scenario: QRadar can identify connections to IP addresses flagged in ransomware campaigns. For example, if a device within the network communicates with a known ransomware command-and-control server, an immediate alert can trigger pre-defined actions like connection termination.*
    - *Real-World Application: Financial institutions use QRadar's threat intelligence to identify and block access to domains hosting fraudulent financial websites, preventing employee or client exposure to phishing attempts.*
    - *Additional Detail: QRadar's integration with external feeds like IBM X-Force or open-source sources like AlienVault enhances its capability to detect advanced persistent threats (APTs).*
3. ***Integrate with SOAR Platforms:***
    - *QRadar integrates seamlessly with Security Orchestration, Automation, and Response (SOAR) platforms to automate the containment and mitigation of threats, reducing manual intervention and minimizing response times.*
    - *Automated workflows can be configured to address common incidents, such as isolating devices infected by malware, notifying stakeholders, and updating incident logs in real-time.*

- *Example: Automating actions in response to phishing attacks, such as quarantining malicious emails, alerting users, and initiating password resets for affected accounts.*
- *Real-World Application: Manufacturing firms using QRadar in combination with SOAR solutions can automatically quarantine compromised industrial control systems (ICS) devices, preventing malware from propagating across assembly lines.*
- *Additional Detail: QRadar's integration with SOAR platforms like IBM Resilient allows organizations to execute automated playbooks, such as shutting down suspicious network connections, blocking malicious IPs, and notifying the SOC team via email and mobile alerts.*

By leveraging these robust and detailed strategies, organizations can build a proactive, resilient threat detection and response framework. The combination of tailored rules, real-time intelligence, and automated responses significantly enhances security posture, ensuring both operational continuity and regulatory compliance.

3.2.3 Security Information and Event Management (SIEM) Capabilities

Objective: Enable real-time monitoring and provide actionable insights to strengthen the security posture through comprehensive event analysis and proactive response mechanisms.

Activities:

1. ***Implement Real-Time Monitoring and Alerting:***
    - *Configure QRadar to continuously monitor critical events, such as brute force attacks, unauthorized access attempts, or anomalous traffic patterns. These configurations should adapt to organizational needs and evolving threat landscapes.*
    - *Use advanced correlation rules to link seemingly isolated incidents into cohesive threat narratives, enhancing the ability to detect multi-stage attacks.*
    - *Example: Financial institutions use QRadar to monitor for multiple failed login attempts within a short time span. When this pattern is detected, an automated alert is sent to the Security Operations Center (SOC) for immediate investigation.*
    - *Real-World Application: A global e-commerce company employs real-time monitoring to detect Distributed Denial of Service (DDoS) attacks targeting their customer-facing platforms. QRadar's alerts enable swift activation of mitigation measures.*

- *Additional Detail: Integrating QRadar with network traffic monitoring tools ensures deeper visibility into encrypted traffic anomalies and lateral movement detection.*

2. ***Generate Reports and Dashboards:***
    - *Leverage QRadar's powerful visualization tools to create detailed dashboards that provide actionable insights into security trends, vulnerabilities, and potential attack vectors.*
    - *Customize reports to meet organizational requirements, such as compliance audits or executive-level summaries that highlight system performance and threat metrics.*
    - *Scenario: A company tracks phishing attempts across multiple departments using QRadar's graphical dashboards. By analyzing the frequency and distribution of these attempts, they implement targeted awareness training for vulnerable teams.*
    - *Real-World Application: A government agency uses QRadar dashboards to monitor unauthorized data exfiltration attempts, providing granular insights into high-risk areas and compliance adherence.*
    - *Additional Detail: QRadar's customizable widgets allow SOC analysts to track metrics like attack types, source geolocations, and the efficiency of automated response mechanisms in real-time.*
3. ***Correlate Historical Data:***
    - *Analyze historical logs to uncover recurring threat patterns and identify periods of heightened vulnerability. This capability enhances long-term threat intelligence and resource allocation.*
    - *Use pattern analysis to preemptively strengthen defenses during predictable spikes in threat activity, such as phishing campaigns during tax season or holiday shopping periods.*
    - *Example: QRadar identifies an increase in spear-phishing emails targeting executives every quarter-end. This insight allows the security team to implement enhanced protections during these critical times.*
    - *Real-World Application: A multinational corporation correlates historical data to detect consistent attempts at credential stuffing, refining their access controls and multi-factor authentication processes.*

- *Additional Detail: The integration of historical data with machine learning models in QRadar can forecast potential attack vectors, providing proactive insights to security teams.*

By implementing these comprehensive SIEM capabilities, organizations gain real-time visibility into their security environment, enabling proactive responses, enhanced decision-making, and a fortified security posture.

3.2.4 Compliance and Auditing

Objective: Ensure adherence to regulatory frameworks, facilitate effective auditing processes, and maintain a robust governance model that aligns with industry standards. By leveraging QRadar's advanced capabilities, organizations can achieve consistent compliance and detailed oversight of critical systems.

Activities:

1. ***Configure QRadar for Compliance Requirements:***
    - *Customization for Standards: Adjust QRadar settings to meet specific regulatory requirements like PCI DSS for payment security, HIPAA for healthcare data protection, GDPR for privacy safeguards, and ISO 27001 for information security management.*
    - *Data Retention Policies: Implement retention settings to store log data for the required duration specified by each regulation (e.g., GDPR mandates secure storage of data for up to five years).*
    - *Alert Mechanisms: Set up rules that trigger alerts for non-compliant actions, such as unauthorized access to sensitive data or unencrypted transmissions.*
    - *Example: QRadar can automatically flag and log any unauthorized modification of customer data, ensuring real-time compliance with GDPR's Article 32 (Security of Processing).*
    - *Real-World Application: A financial services company configures QRadar to track the handling of cardholder data, aligning processes with PCI DSS requirements by encrypting all transmissions and generating real-time non-compliance alerts.*
    - *Additional Detail: QRadar's pre-built compliance modules simplify implementation by providing templates, such as rule sets and report formats, that align with regulatory standards.*

2. ***Generate Audit Trails and Logs:***
    - *Comprehensive Log Management: Maintain immutable logs of system activities, user interactions, and security events to ensure a detailed historical record that supports forensic investigations and compliance audits.*
    - *Administrative Oversight: Create trails that capture critical administrative actions, such as privilege changes or rule modifications, to demonstrate governance adherence.*
    - *Scenario: Forensic logs from QRadar help an organization pinpoint how an unauthorized user accessed a database by reconstructing login attempts, session durations, and associated IP addresses.*
    - *Real-World Application: A hospital network uses QRadar to track access to patient records, producing detailed HIPAA-compliant reports that document all interactions with electronic health records (EHRs).*
    - *Integration with Investigative Tools: QRadar integrates with digital forensics platforms to enhance post-incident investigation, providing clear evidence trails required for legal or regulatory proceedings.*
    - *Secure Archiving: Use external storage solutions integrated with QRadar to store logs for extended periods, ensuring compliance with standards like SOX (Sarbanes-Oxley), which demands long-term financial record retention.*

By combining meticulous configuration with robust audit capabilities, organizations can confidently navigate the complex landscape of regulatory compliance while fostering a culture of accountability and security.

3.2.5 User Management and Access Control

Objective: Establish secure, scalable, and efficient access management mechanisms within QRadar to ensure that sensitive data and administrative controls are appropriately restricted based on user roles, while minimizing risks associated with privilege misuse and unauthorized access.

Activities:

1. ***Implement Role-Based Access Control (RBAC):***
    - *Define User Roles: Precisely define roles and responsibilities to tailor access permissions according to job functions, operational needs, and security requirements. Each role should be aligned with the principle of*

least privilege, granting only the necessary access required to perform tasks.

- *Granular Role Hierarchies: Leverage QRadar's RBAC framework to establish detailed role hierarchies. For instance, differentiate between roles like SOC Tier-1 analysts (read-only log access), SOC Tier-2 analysts (incident triage capabilities), and administrators (rule modifications and system management).*
- *Audit and Oversight: Implement QRadar's audit tools to continuously monitor role-based actions. This ensures all user activities are logged, providing transparency and enabling quick detection of policy violations.*
- *Example: Junior analysts are assigned read-only permissions to logs for incident analysis but are restricted from modifying detection rules, maintaining system integrity.*
- *Real-World Application: A global financial firm uses RBAC to separate the duties of compliance officers, who can view compliance-related reports but cannot alter security configurations, ensuring strict governance.*
- *Enhanced Details: QRadar integrates with directory services like LDAP and Active Directory, allowing dynamic role assignments based on organizational changes.*

2. ***Regularly Review Access Permissions:***

- *Scheduled Audits: Conduct periodic reviews of user permissions to verify they align with current roles, responsibilities, and employment status. These reviews prevent privilege creep, where users retain unnecessary access over time.*
- *Automated Updates: Integrate QRadar with identity management and HR systems to automate the modification of user privileges during onboarding, role transitions, and offboarding. This ensures rapid updates and eliminates human error.*
- *Scenario: QRadar detects an active account associated with a former employee due to a misconfiguration. By syncing with the organization's HR system, the account is immediately flagged and disabled, preventing unauthorized access.*
- *Real-World Application: A healthcare provider uses QRadar to manage clinician access dynamically. When a doctor transitions to a different department, their access is adjusted to reflect the new role, ensuring adherence to HIPAA guidelines.*

- *Detailed Insights: QRadar's permission review reports identify anomalies, such as accounts with excessive permissions or inactive accounts that remain active, enabling swift remediation.*

3. ***Advanced Privilege Management:***
    - *Context-Aware Access: Implement time-bound or context-sensitive access controls. For example, restrict administrative access to maintenance windows or specific IP ranges.*
    - *Incident-Based Restrictions: During a security incident, QRadar can dynamically adjust user privileges, such as temporarily revoking administrative rights to prevent unauthorized changes.*
    - *Example: During a breach investigation, QRadar restricts access to critical logs and configurations, limiting actions to senior administrators only.*
    - *Real-World Application: A technology firm enforces geographic restrictions on access to sensitive systems, preventing overseas logins unless pre-approved by the SOC.*
    - *Proactive Governance: Use QRadar's analytics to identify high-risk users or unusual activity patterns and apply conditional restrictions automatically.*

By adopting a comprehensive approach to user management and access control, QRadar enables organizations to enforce least-privilege policies, mitigate insider threats, and maintain a secure and resilient operational environment.

3.3. Implementation Methodology

1. Planning and Preparation

- *Comprehensive Security Assessment: A comprehensive security assessment serves as the foundation of the QRadar implementation process. It involves:*
    - *Performing an exhaustive evaluation of the existing security landscape, covering network topology, server configurations, endpoint protections, and user access policies. The aim is to identify weaknesses, gaps in threat monitoring, and areas requiring immediate remediation.*
    - *Reviewing logs from firewalls, intrusion detection/prevention systems (IDS/IPS), and application logs to identify trends or anomalies that might indicate security issues.*
    - *Conducting data flow mapping to understand how sensitive data traverses the network and pinpoint potential exposure points.*

- *Real-World Example: A retail company detected multiple points of exposure in their POS (Point of Sale) systems where unencrypted credit card data was being logged temporarily. This insight prompted system reconfigurations that prevented data breaches and enhanced PCI DSS compliance.*

- ***Stakeholder Engagement: Stakeholder alignment ensures that security initiatives reflect broader organizational objectives and compliance needs. This involves:***
  - *Collaborating with IT departments to assess existing infrastructure, compliance officers to ensure adherence to regulations, and business leaders to align the project's scope with strategic goals.*
  - *Hosting workshops to educate stakeholders on QRadar's capabilities, gaining buy-in and clarifying how security improvements will enhance operational efficiency and reduce risks.*
  - *Real-World Example: A healthcare organization convened quarterly meetings with IT and compliance teams during their QRadar deployment to ensure alignment with HIPAA regulations. This approach streamlined decision-making and enabled the project to meet deadlines without compliance risks.*

- ***Define KPIs: Defining actionable and measurable KPIs establishes a framework for success evaluation. These may include:***
  - *Reducing mean time to detect (MTTD) from several hours to less than 30 minutes.*
  - *Improving mean time to respond (MTTR) to critical incidents, cutting resolution time by 50%.*
  - *Increasing compliance audit success rates by automating compliance checks and maintaining consistent log reviews.*
  - *Real-World Example: A financial institution set aggressive KPIs to identify insider threats within minutes and respond to external phishing attempts in under 15 minutes. QRadar's ability to provide real-time alerts significantly improved these metrics.*

2. Execution

- *QRadar Deployment: The deployment phase ensures a seamless introduction of QRadar into the organization's IT ecosystem. Activities include:*

- *Installing QRadar modules across distributed environments, including on-premises data centers and cloud-based resources, while ensuring minimal disruption.*
- *Conducting pre-deployment assessments to identify hardware and software compatibility issues and mitigate risks.*
- *Real-World Example: An energy company integrated QRadar to monitor SCADA (Supervisory Control and Data Acquisition) systems critical to power grid operations. Deployment was phased to prevent outages, and QRadar's alerting mechanisms were tuned to detect OT-specific anomalies.*

- ***Data Source Integration: Data integration amplifies QRadar's monitoring capabilities by feeding it relevant and comprehensive security data. This includes:***
  - *Connecting firewalls, IDS/IPS, servers, endpoint protection platforms, and cloud environments as data sources.*
  - *Ensuring structured data collection from IoT devices, which often require tailored connectors due to non-standard logging formats.*
  - *Real-World Example: A logistics company connected QRadar to GPS systems, ensuring real-time tracking of fleet security. They detected and mitigated a cyberattack that aimed to disrupt transportation routes through malicious command injections.*
- ***Custom Rules and Dashboards: Configuring QRadar to meet specific organizational requirements involves:***
  - *Developing detection rules tailored to high-priority threats such as advanced persistent threats (APTs) or insider data exfiltration.*
  - *Creating dashboards to visualize key security metrics, such as high-severity incidents by source, or compliance trends over time.*
  - *Real-World Example: A telecom provider developed dashboards to visualize DDoS attacks in real time. This allowed for rapid response, minimizing downtime and protecting critical customer communication services.*

3. Testing and Validation

- *Component Testing: Thorough testing of individual QRadar components ensures readiness before full-scale deployment. This involves:*

- *Verifying data ingestion pipelines from all integrated sources and confirming that logs are accurately parsed and categorized.*
- *Stress-testing QRadar's processing capabilities to ensure performance under high-load conditions.*
- *Real-World Example: A manufacturing company tested QRadar against simulated OT network traffic, ensuring it could handle the unique protocols and high volumes typical in industrial environments.*

- ***Scenario-Based Validation: Real-world scenario simulations test QRadar's detection capabilities under realistic threat conditions. Examples include:***
  - *Conducting simulated ransomware attacks to evaluate how effectively QRadar detects file encryption activities.*
  - *Running credential stuffing scenarios to assess the alerting and blocking mechanisms.*
  - *Real-World Example: A government agency simulated a multi-vector phishing attack targeting employee credentials and tested QRadar's ability to detect unusual login patterns across geographies.*
- ***Continuous Feedback Loop: Collaborating with SOC teams to refine configurations based on validation results is essential. This includes:***
  - *Adjusting detection thresholds to minimize false positives while retaining sensitivity to true threats.*
  - *Iterative improvement cycles informed by user feedback and post-incident analysis.*
  - *Real-World Example: A global financial services firm refined detection rules over six months by analyzing every false positive, achieving a 75% reduction while maintaining robust threat detection capabilities.*

3.4. Expected Outcomes

1. Enhanced Threat Detection

- *Early Identification of Advanced Threats: QRadar revolutionizes threat detection by identifying sophisticated attacks, such as ransomware, insider threats, and advanced persistent threats (APTs). It achieves this through:*
  - *Processing vast quantities of log data from diverse sources, including servers, firewalls, and cloud platforms.*

- *Leveraging advanced machine learning algorithms to detect subtle anomalies and indicators of compromise that traditional systems might overlook.*
- *Utilizing correlation rules to analyze patterns across systems, revealing complex multi-vector attacks.*
- *Real-World Example: A global financial institution detected insider data exfiltration attempts through QRadar's anomaly detection. The system flagged irregular file access patterns and unauthorized data transfers, enabling immediate intervention that prevented a major breach.*
- *Additional Detail: QRadar's User Behavior Analytics (UBA) feature adds a behavioral dimension, allowing the detection of slow-moving and stealthy attacks like APTs by spotting deviations in normal user and system activities.*
- *Expanded Use Case: A manufacturing firm detected early-stage ransomware behavior when QRadar flagged unusual encryption activities on several workstations, allowing the SOC team to isolate affected systems before the ransomware could propagate.*

2. Improved Response Efficiency

- *Automated Incident Response: QRadar integrates seamlessly with Security Orchestration, Automation, and Response (SOAR) platforms to:*
  - *Automate repetitive tasks such as alert triaging, log analysis, and threat containment.*
  - *Prioritize alerts based on severity, ensuring critical incidents receive immediate attention.*
  - *Initiate rapid containment measures, such as disabling compromised accounts or isolating infected endpoints, without manual intervention.*
  - *Real-World Example: An e-commerce company deployed QRadar's automated workflows to isolate compromised endpoints during a malware outbreak. This action halted the infection's spread within seconds, minimizing operational disruption and customer impact.*
  - *Additional Detail: QRadar's predefined playbooks for scenarios like phishing, data exfiltration, or DDoS attacks ensure consistent, efficient incident handling. It also supports custom playbooks tailored to specific organizational needs.*

- *Expanded Use Case: A retail chain implemented QRadar's SOAR integration to automate the investigation of suspicious payment terminal activities, reducing incident response time from hours to minutes.*

3. Compliance Readiness

- *Streamlined Compliance Management: QRadar simplifies adherence to regulatory requirements by providing:*
  - *Pre-configured compliance packages with templates for GDPR, HIPAA, PCI DSS, and SOX.*
  - *Automated generation of audit reports, saving time and ensuring accuracy.*
  - *Real-time monitoring and alerting for non-compliance scenarios, enabling proactive issue resolution.*
  - *Real-World Example: A healthcare organization reduced its HIPAA audit preparation time by 40% using QRadar's compliance dashboards and automated reporting features. The system also flagged potential compliance risks, such as unencrypted data transfers, for immediate remediation.*
  - *Additional Detail: QRadar's log retention and access controls provide auditable trails for all monitored activities, a critical feature for passing regulatory scrutiny.*
  - *Expanded Use Case: A financial services firm used QRadar to maintain continuous compliance with evolving PCI DSS standards, ensuring secure processing of credit card transactions across multiple regions.*

4. Actionable Insights

- *Comprehensive Dashboards and Reports: QRadar provides SOC teams with actionable intelligence by:*
  - *Aggregating and visualizing data from diverse sources in real-time dashboards.*
  - *Enabling quick identification of trends, such as rising brute force attempts or increasing phishing incidents, through advanced data analytics.*
  - *Supporting detailed forensic investigations with drill-down capabilities, offering granular visibility into individual events.*
  - *Real-World Example: A telecom provider leveraged QRadar dashboards to detect and respond to a brute force attack targeting its administrative*

accounts. The visualization of failed login trends enabled swift identification and mitigation of the threat.

- *Additional Detail: QRadar's customizable reporting capabilities allow teams to generate insights tailored to different stakeholder needs, from technical metrics for SOC teams to executive summaries for leadership.*
- *Expanded Use Case: An energy company used QRadar's insights to monitor the security of SCADA systems, detecting irregular command patterns that indicated a potential cyberattack on critical infrastructure.*

3.5. Challenges and Mitigation Strategies

1. Integration Complexity

- *Challenge: Integrating QRadar into diverse and complex IT environments can be daunting, especially when working with:*
  - *Legacy systems with outdated protocols.*
  - *Proprietary applications that lack standardized logging formats.*
  - *Highly segmented networks that pose data aggregation challenges.*
  - ***Mitigation:***
    - *Conduct a thorough pre-deployment assessment to document the infrastructure, identifying potential compatibility issues.*
    - *Develop a detailed integration roadmap prioritizing high-value systems and defining milestones for phased deployments.*
    - *Engage QRadar's vendor support and professional services to address complex integration requirements and implement custom solutions.*
    - *Leverage advanced log parsers and custom connectors to ensure seamless data ingestion.*
  - *Real-World Example: A multinational corporation successfully integrated QRadar into over 100 disparate systems, including legacy databases and cloud-native applications, by utilizing phased rollouts and vendor collaboration to ensure smooth deployment.*
  - *Expanded Use Case: An energy utility integrated QRadar with SCADA systems to monitor industrial protocols. The project addressed OT-specific log translation challenges by deploying tailored connectors co-developed with QRadar engineers, enabling real-time operational security monitoring.*

2. User Training

- *Challenge: Role-specific training gaps can hinder effective utilization of QRadar's capabilities, leading to inefficiencies and missed opportunities to enhance security.*
    - ***Mitigation:***
        - *Design and implement comprehensive, role-specific training programs that include:*
            - *Hands-on workshops to simulate real-world scenarios such as phishing or ransomware attacks.*
            - *Detailed walkthroughs of QRadar's interface, including dashboards, rule configuration, and reporting tools.*
            - *E-learning modules to cover foundational concepts and advanced use cases.*
            - *Periodic refresher courses to introduce feature updates and adapt to emerging threat trends.*
        - *Conduct team-based exercises to foster collaboration between SOC analysts, IT administrators, and compliance teams.*
    - *Real-World Example: A government agency's quarterly training workshops focused on complex threat detection scenarios, significantly enhancing SOC proficiency and reducing average response times to incidents by 30%.*
    - *Expanded Use Case: A retail organization developed a multi-tiered training program with a dedicated e-learning module for store managers. This enabled non-technical staff to identify and escalate localized cybersecurity threats effectively, fostering an organization-wide culture of security awareness.*

3. False Positives

- *Challenge: An excessive volume of false positives can overwhelm SOC teams, diluting their focus and reducing operational efficiency.*
    - ***Mitigation:***
        - *Regularly refine detection rules and thresholds by analyzing historical alert data and incorporating feedback from incident post-mortems.*

        - *Implement machine learning models that learn from SOC analyst responses, reducing the recurrence of similar false positives.*
        - *Develop an escalation matrix to categorize alerts by severity and criticality, enabling focused prioritization.*
        - *Incorporate contextual enrichment of alerts, such as geolocation data and device behavior baselines, to improve accuracy.*
    - *Real-World Example: A retail chain iteratively fine-tuned their QRadar detection configurations over six months, resulting in a 60% reduction in false positives. This optimization allowed SOC teams to redirect their efforts toward high-risk incidents and preempt potential breaches.*
    - *Expanded Use Case: A financial institution deployed QRadar's anomaly detection engine to flag unusual login behaviors, such as simultaneous access attempts from disparate locations. Through calibrated thresholds and contextual analysis, they achieved high detection precision while minimizing noise from routine activities.*

3.6. Introduction

3.6.1 Background

In today's interconnected digital landscape, cyberattacks are not just frequent but also increasingly sophisticated, posing significant risks to organizations of all sizes. Examples like the 2021 Colonial Pipeline ransomware attack, which caused widespread fuel shortages across the Eastern United States, or the Equifax data breach in 2017, which exposed the personal information of 147 million individuals, underline the real-world consequences of inadequate cybersecurity measures.

Security Information and Event Management (SIEM) systems have become indispensable in this context. They act as a central nervous system for security operations by aggregating and analyzing log data from various sources, such as firewalls, endpoints, and cloud environments. These systems help organizations detect anomalies, identify malicious activities, and respond to incidents in real time. For example, SIEM systems have been used to thwart Distributed Denial of Service (DDoS) attacks by quickly identifying unusual traffic patterns and enabling rapid response measures.

Among the leading SIEM solutions, IBM QRadar has earned a reputation for its robust capabilities. Its advanced analytics leverage machine learning and threat intelligence to detect threats that may otherwise go unnoticed. QRadar's seamless integration with external threat intelligence platforms allows it to provide contextually rich insights, enhancing decision-making for security teams. Furthermore, its ability to scale across

on-premises and cloud environments makes it a versatile choice for modern organizations facing diverse and evolving threat landscapes.

3.6.2 Problem Statement

In the rapidly evolving landscape of cybersecurity, organizations face an increasingly complex array of challenges that compromise their ability to detect and respond to threats effectively. Cybercriminals are employing advanced tactics, such as leveraging artificial intelligence to evade detection or exploiting vulnerabilities in critical infrastructure. A stark example is the 2021 Colonial Pipeline ransomware attack, where the attackers used a single compromised password to disrupt fuel supplies, highlighting how minor oversights can lead to significant consequences.

Large-scale enterprises, such as multinational corporations and financial institutions, generate massive amounts of data—ranging from terabytes to petabytes daily. This data comes from sources such as firewalls, intrusion detection systems, cloud services, endpoint devices, and Point of Sale (POS) systems. Filtering through this data to identify actionable threats has become akin to searching for a needle in an ever-growing haystack. For instance, detecting an unauthorized access attempt from an unrecognized IP address, followed by unusual file transfers, requires real-time correlation across multiple systems to prevent a potential breach.

The diversity and sophistication of attack vectors further exacerbate these issues. Modern adversaries use Advanced Persistent Threats (APTs) to maintain long-term access to targeted systems, enabling them to exfiltrate sensitive data over time without detection. Similarly, zero-day vulnerabilities—exploits for which no patches exist—are increasingly weaponized, as demonstrated by the infamous SolarWinds attack. This incident saw threat actors infiltrating numerous high-profile organizations, including government agencies and Fortune 500 companies, by compromising the software supply chain.

Traditional security tools often lack the advanced correlation, contextual awareness, and scalability needed to address these challenges. This deficiency leads to delayed threat detection, inefficient incident response, and a heightened risk of financial losses, reputational damage, and regulatory penalties. Organizations must also comply with stringent regulations, such as GDPR and CCPA, which demand robust incident reporting and data protection measures.

A comprehensive solution like IBM QRadar is critical for overcoming these challenges. QRadar's advanced analytics and machine learning capabilities enable it to process vast amounts of data in real-time, identifying patterns indicative of malicious behavior. Its real-time correlation engine allows security teams to connect disparate events into cohesive threat narratives, ensuring timely detection and response. Furthermore, QRadar's integration with threat intelligence platforms provides contextual insights,

empowering organizations to proactively defend against emerging threats while maintaining compliance with regulatory requirements.

3.6.3 Project Objectives

The primary objectives of this project are meticulously designed to provide an in-depth evaluation of IBM QRadar and its capability to fortify organizational cybersecurity. The objectives encompass the following:

1. *Evaluating the Effectiveness of IBM QRadar: This objective aims to assess QRadar's performance under real-world conditions. It includes analyzing how well QRadar detects phishing attacks, malware activities, and unauthorized data exfiltration while comparing its functionalities with other top-tier SIEM solutions.*
    - *Example: Simulating a multi-vector phishing attack to evaluate QRadar's capacity to integrate email logs, endpoint activity, and DNS requests, effectively generating actionable alerts in real-time.*
    - *Extended Example: Benchmarking QRadar against a competing solution, such as Splunk, to identify differences in alert generation, false-positive rates, and overall detection efficiency during an emulated ransomware attack scenario.*
2. *Implementing and Validating Specific Use Cases: Develop and deploy tailored QRadar use cases to address prevalent security threats. This involves creating custom rules and workflows to detect brute-force login attempts, ransomware activities, and insider threats while ensuring operational feasibility and scalability.*
    - *Example: Designing a rule to identify anomalous access patterns, such as repeated failed login attempts followed by a successful login from a suspicious geographic location, which could indicate credential stuffing or account compromise.*
    - *Extended Example: Testing the effectiveness of use cases in identifying lateral movement within the network by correlating endpoint logs with unusual internal traffic patterns, simulating tactics commonly used in Advanced Persistent Threats (APTs).*
3. *Enhancing Organizational Security Posture: Integrate QRadar with complementary security tools like Endpoint Detection and Response (EDR) systems, vulnerability management platforms, and Security Orchestration, Automation, and Response (SOAR) tools. This objective focuses on achieving end-to-end threat visibility, streamlining incident response processes, and ensuring adherence to compliance requirements such as PCI DSS or HIPAA.*

- *Example: Automating the containment of compromised devices through QRadar's integration with SOAR tools, enabling immediate isolation of endpoints exhibiting malware behavior.*
- *Extended Example: Leveraging QRadar's reporting features to create executive-level dashboards that provide insights into compliance posture, risk trends, and the effectiveness of implemented security measures.*

3.6.4 Scope

This project represents a thorough and strategic analysis and deployment of IBM QRadar, aimed at addressing the intricate challenges of modern cybersecurity operations. The scope is comprehensive and focused on delivering actionable insights and enhanced threat protection by leveraging QRadar's capabilities. The main areas of concentration include:

1. *Log Collection and Aggregation: QRadar's ability to gather and unify logs from a wide variety of sources is a cornerstone of its functionality. This ensures a centralized view of the organization's security posture and enhances the detection of hidden threats.*
    - *Example: Configuring QRadar to ingest logs from on-premises devices such as Palo Alto firewalls, and cloud platforms like AWS CloudTrail and Microsoft Azure Monitor, enabling detection of unauthorized administrative actions or misconfigurations.*
    - *Real-World Application: A financial institution might use QRadar to monitor and cross-correlate transaction data from multiple payment gateways with firewall logs to detect potential insider threats.*
2. *Advanced Threat Detection: Leveraging QRadar's correlation engine and AI-driven analytics allows for the identification of complex, multi-stage attack patterns.*
    - *Example: Creating detection rules for identifying lateral movement by correlating endpoint activity with anomalous network flows, commonly seen in ransomware attacks.*
    - *Real-World Application: A healthcare provider can deploy rules to detect unusual login behaviors, such as access from an atypical geographic location followed by bulk data downloads, signaling potential exfiltration attempts.*
3. *Reporting and Regulatory Compliance: QRadar's robust reporting features allow organizations to maintain compliance with industry standards while also providing actionable insights to stakeholders.*

- *Example: Generating GDPR-compliant reports that detail user access patterns and incident responses to meet regulatory requirements.*
- *Real-World Application: Retail organizations subject to PCI DSS can use QRadar to create detailed logs of credit card processing activities, identifying potential data breaches.*

4. *Customization of Use Cases: Designing and implementing tailored use cases to address specific organizational threats ensures that QRadar is aligned with the company's unique security needs.*
    - *Example: Developing a rule to detect a series of failed login attempts from different IPs targeting a single account, followed by a successful login, indicative of a credential stuffing attack.*
    - *Real-World Application: An e-commerce platform can track bot activity attempting to compromise user accounts through repeated automated login attempts.*

Critical Security Threats Addressed

The scope of this project explicitly targets key cybersecurity threats:

- *Malware Infections: QRadar's ability to identify indicators of compromise (IOCs) such as unexpected file executions or connections to known malicious IPs ensures early detection and containment of malware outbreaks. For example, identifying C2 (Command and Control) traffic from an endpoint to an external server.*
- *Unauthorized Access Attempts: Utilizing QRadar to detect brute-force attacks, abnormal privilege escalations, and credential stuffing attempts ensures robust access controls and preemptive threat mitigation. For instance, flagging unauthorized administrative account access outside approved hours.*

By methodically addressing these areas, this project aims to not only enhance immediate threat detection capabilities but also build a resilient, scalable security framework that ensures robust protection against both current and emerging cyber threats.

3.7. Literature Review

3.7.1 SIEM Technology

3.7.1.1 Evolution of SIEM

SIEM technology has undergone a remarkable transformation, driven by the increasing complexity and scale of cyber threats. The journey began with basic log management systems, which were primarily used to collect and store raw log data for future reference. These systems offered limited functionality and lacked real-time capabilities, making them inadequate for proactive threat detection.

The next evolutionary step was the advent of Security Information Management (SIM) systems, which centralized log storage and introduced elementary reporting functionalities. SIM solutions, however, were not designed to handle real-time events, leading to the parallel development of Security Event Management (SEM) systems. SEM solutions focused on real-time monitoring, event correlation, and alerting, enabling quicker responses to emerging threats.

The fusion of SIM and SEM gave rise to Security Information and Event Management (SIEM) systems, which combine the strengths of both technologies. Modern SIEM platforms represent a significant leap forward, incorporating cutting-edge technologies such as:

- *Machine Learning: Utilized to identify unusual behavior patterns and anomalies that traditional rule-based systems fail to detect. For example, machine learning can recognize insider threats by analyzing deviations in user behavior, such as accessing sensitive files during unusual hours.*
- *Threat Intelligence Integration: Real-time integration with global threat intelligence feeds allows SIEM systems to identify and prioritize emerging risks effectively. For instance, correlating local events with known malware signatures from global databases.*
- *Advanced Analytics: Predictive algorithms help organizations anticipate potential breaches and respond preemptively. Techniques like user and entity behavior analytics (UEBA) further enhance the ability to detect complex attack scenarios.*

Real-World Example: The 2013 Target data breach exemplifies the importance of SIEM advancements. Attackers infiltrated Target's network using stolen credentials and installed malware on the point-of-sale (POS) systems. Although the retailer had a SIEM tool in place, the lack of robust correlation and timely alerting delayed detection. Modern SIEM platforms, equipped with AI-driven analytics and automated responses, would have likely detected the unusual access patterns and malware signatures earlier, preventing the extensive damage.

The evolution of SIEM underscores its critical role in modern cybersecurity, enabling organizations to combat increasingly sophisticated threats effectively.

3.7.1.2 Key Features of SIEM

SIEM systems provide a wide range of critical features that form the backbone of modern cybersecurity infrastructure. These features ensure efficient data handling, robust threat detection, and actionable insights for security teams. Below are the detailed features with real-world applications:

- *Log Collection and Normalization: SIEM systems aggregate logs from diverse sources such as firewalls, servers, endpoints, and cloud environments. Normalization transforms this data into a standardized format, making it easier to analyze and correlate.*
    - *Example: A global e-commerce company collects logs from its web servers and payment gateways. Normalization allows the detection of anomalies like repeated login failures across different systems, which could indicate a coordinated brute-force attack. This standardized approach also enables rapid integration with third-party analytics tools, providing deeper visibility into network behavior.*
- *Correlation and Alerting: By applying correlation rules, SIEM systems identify patterns in seemingly unrelated events to detect potential threats. These systems then generate alerts to notify security teams, enabling swift responses to incidents.*
    - *Example: In a financial institution, a SIEM correlates an unusual number of failed login attempts with the successful use of privileged accounts during non-working hours, triggering an alert for possible credential compromise. Additionally, these alerts can be integrated with automated workflows to initiate incident response measures, such as disabling compromised accounts.*
- *Reporting and Dashboards: SIEM platforms provide dynamic dashboards and detailed reports to visualize security trends, compliance status, and operational health. These tools are crucial for decision-making and regulatory adherence.*
    - *Example: A healthcare organization uses SIEM dashboards to monitor HIPAA compliance by tracking access to patient data and ensuring audit trails are maintained. Visualized trends help in identifying non-compliance risks promptly, while automated report generation simplifies regular audits for regulatory authorities.*
- *Behavioral Analysis and Threat Detection: Modern SIEM systems incorporate behavioral analytics to detect sophisticated threats, such as Advanced Persistent Threats (APTs), by analyzing patterns of user activity over time.*

- *Example: A government agency detects unusual data access patterns indicating a potential insider threat. By continuously learning from user behavior, the SIEM system flags anomalies, such as unauthorized attempts to copy large volumes of sensitive data.*

These features collectively empower organizations to proactively manage cybersecurity risks and ensure compliance with industry regulations. With the integration of AI-driven technologies, modern SIEM platforms continue to adapt to emerging challenges in the digital threat landscape.

3.7.1.3 SIEM Architectures and Deployment Models

SIEM solutions cater to a variety of organizational needs through different architectures and deployment models. These options offer scalability, flexibility, and redundancy, ensuring robust security across diverse operational environments. Below is an in-depth analysis of each model, enriched with real-world applications:

- *Centralized: In this architecture, all log collection, storage, and analysis are consolidated in a single, central location. This approach simplifies infrastructure management, provides a unified view of security events, and is particularly suitable for smaller or less distributed organizations.*
    - *Example: A nationwide retail chain with centralized IT management employs this model to monitor security logs across stores. The central SIEM setup detects coordinated card-skimming attacks across multiple branches, allowing rapid response.*
    - *Advantages: Simplified maintenance, lower infrastructure costs, and efficient centralized correlation of security events.*
    - *Challenges: Potential bottlenecks if the volume of logs is too large, and increased risk of a single point of failure.*
- *Distributed: This architecture leverages multiple nodes spread across geographic or operational locations to collect and analyze logs. Distributed SIEMs are ideal for organizations with multiple global offices or high data volumes, offering localized data analysis with integration to a central monitoring hub.*
    - *Example: A multinational banking institution uses a distributed SIEM model to ensure real-time detection of threats across branches in different countries. Regional nodes analyze local events and escalate critical incidents to a centralized dashboard for executive oversight.*
    - *Advantages: Enhanced scalability, reduced latency, and better redundancy for disaster recovery.*

    - *Challenges: Increased complexity in setup and coordination between distributed nodes.*

- *Cloud-Based: Cloud-native SIEMs utilize scalable cloud infrastructure for log storage, analysis, and threat detection. These systems are favored by organizations that prioritize rapid deployment, cost-effectiveness, and accessibility.*
    - *Example: A technology startup adopts a cloud-based SIEM to protect its hybrid cloud environment. By integrating seamlessly with SaaS tools and cloud platforms, the SIEM identifies data exfiltration attempts through misconfigured APIs.*
    - *Advantages: Elastic scalability, reduced upfront hardware costs, and access to advanced analytics and AI features.*
    - *Challenges: Data privacy concerns, potential compliance limitations, and reliance on internet connectivity.*
- *Hybrid: Combining the strengths of both on-premises and cloud-based solutions, hybrid SIEMs address the needs of organizations requiring local data control for compliance while leveraging cloud capabilities for advanced analytics and scalability.*
    - *Example: A government agency employs a hybrid SIEM model to analyze classified data on-premises while using cloud resources for threat intelligence correlation. This setup ensures compliance with stringent data sovereignty laws.*
    - *Advantages: Balanced approach offering both control and flexibility, suitable for compliance-heavy industries.*
    - *Challenges: Complexity in deployment and higher costs due to maintaining dual environments.*

Real-World Implications:

Selecting the right SIEM architecture involves weighing operational needs, regulatory requirements, and budget constraints. For instance, the centralized model works well for smaller entities, while hybrid models cater to industries with stringent data governance policies, such as finance or healthcare. Understanding the benefits and limitations of each deployment model ensures organizations optimize their cybersecurity infrastructure while maintaining operational efficiency.

3.7.1.4 Comparison of SIEM Vendors

Leading SIEM vendors offer diverse solutions to meet the complex and evolving needs of organizations worldwide. Below is a comprehensive analysis of prominent SIEM vendors, highlighting their strengths, distinctive features, and real-world applications:

- ***Splunk:***
    - *Strengths: Renowned for its powerful analytics capabilities, extensive integration options, and scalability. Splunk provides unmatched flexibility in data handling and visualization, making it a top choice for large enterprises.*
    - ***Features:***
        - *Advanced machine learning algorithms for predictive threat detection.*
        - *Customizable dashboards for real-time insights and long-term trend analysis.*
        - *Integration with over 2,000 IT and security tools, enabling seamless operations.*
    - *Real-World Example: A multinational e-commerce company leverages Splunk to monitor billions of daily transactions. Its real-time alerting capabilities help detect fraudulent activities, such as credit card misuse or account takeovers, significantly reducing losses.*
- ***ArcSight:***
    - *Strengths: Excels in compliance reporting and advanced correlation capabilities, making it a preferred choice for regulated industries such as healthcare and finance.*
    - ***Features:***
        - *Event correlation across millions of data points to uncover complex attack chains.*
        - *Integration with global threat intelligence feeds to prioritize and mitigate emerging risks.*
        - *Modular architecture for enhanced scalability and customizability.*
    - *Real-World Example: A healthcare network employs ArcSight to ensure compliance with HIPAA regulations. By analyzing access logs and correlating unauthorized attempts, it prevents data breaches involving patient records, avoiding regulatory penalties and reputational harm.*
- ***LogRhythm:***

- *Strengths: Focused on ease of use, automation, and operational efficiency. LogRhythm's user-centric design makes it ideal for small and mid-sized organizations seeking robust cybersecurity without extensive resources.*
- *Features:*
  - *Pre-configured compliance templates for standards like PCI DSS and GDPR.*
  - *Automated workflows to streamline incident response processes.*
  - *User-friendly dashboards with actionable insights for rapid decision-making.*
- *Real-World Example: A regional financial services firm uses LogRhythm to reduce its mean time to detect (MTTD) and respond (MTTR) to phishing attacks. The automated analysis of email logs identifies malicious links and quarantines them before they can reach employees.*

- ***IBM QRadar:***
  - *Strengths: Distinguished by its sophisticated correlation engine, AI-driven threat intelligence, and suitability for large-scale enterprise environments.*
  - ***Features:***
    - *Behavioral analysis tools for detecting anomalies in user and system activities.*
    - *AI-powered threat prioritization to focus on the most critical alerts.*
    - *Seamless integration with IBM's broader cybersecurity suite, including SOAR (Security Orchestration, Automation, and Response).*
  - *Real-World Example: A critical infrastructure provider utilizes IBM QRadar to safeguard its operations. The platform's ability to detect advanced persistent threats (APTs) targeting SCADA systems has been instrumental in maintaining the reliability of energy distribution networks.*

Evaluation Criteria

To choose the optimal SIEM solution, organizations should evaluate:

1. *Scalability: Whether the platform can handle current and future data volumes.*
2. *Integration Capabilities: Compatibility with existing IT and security tools.*

3. *Customization: Flexibility in adapting to unique workflows and industry-specific requirements.*
4. *Cost: Balancing upfront investment with long-term benefits.*

By aligning vendor offerings with organizational goals, businesses can enhance their cybersecurity posture while achieving compliance and operational efficiency.

3.7.2 IBM QRadar

3.7.2.1 QRadar Overview

IBM QRadar is a cutting-edge Security Information and Event Management (SIEM) platform, meticulously designed to address the increasingly complex landscape of modern cybersecurity threats. As cyberattacks grow in sophistication—ranging from advanced persistent threats (APTs) to zero-day exploits—QRadar offers organizations a comprehensive solution to monitor, detect, investigate, and respond to these challenges in real time.

By consolidating log management, integrated threat intelligence, and automated incident response workflows into a unified framework, QRadar provides an unparalleled level of visibility into organizational networks. Its scalability and adaptability cater to diverse environments, from small businesses to global enterprises, ensuring it meets the security needs of organizations across industries.

QRadar empowers Security Operations Center (SOC) teams by providing real-time actionable insights into potential risks, enabling proactive threat mitigation. For instance, during a simulated Distributed Denial of Service (DDoS) attack targeting critical financial infrastructure, QRadar's ability to correlate unusual network traffic spikes with known attack vectors enabled rapid identification and containment of the threat, preserving business continuity. Additionally, its machine learning capabilities help identify subtle anomalies, such as deviations in user behavior that may signal insider threats or compromised accounts.

The platform's extensibility further strengthens its utility. QRadar seamlessly integrates with a variety of third-party tools, such as endpoint detection and response (EDR) systems, cloud-native security solutions, and vulnerability management platforms. For example, an organization leveraging QRadar in a hybrid environment can integrate AWS GuardDuty and Azure Sentinel logs to centralize threat detection, ensuring consistent protection across on-premises and cloud infrastructures.

Moreover, QRadar's robust reporting and compliance features simplify the process of adhering to industry standards, such as GDPR and ISO 27001. Dashboards tailored for compliance monitoring enable organizations to generate detailed reports, highlighting both security events and remedial actions taken. This capability not only supports

regulatory audits but also demonstrates a proactive approach to safeguarding sensitive data.

In summary, IBM QRadar stands as a cornerstone of modern cybersecurity strategies, equipping organizations with the tools to preemptively address emerging threats, streamline security operations, and maintain resilience against cyber adversaries.

3.7.2.2 QRadar Architecture

The architecture of IBM QRadar is intricately designed to efficiently ingest, process, and analyze diverse security data streams, ensuring robust threat detection and incident response across complex environments. Its modular components work in harmony to provide comprehensive visibility, real-time threat analysis, and actionable intelligence. The key architectural elements include:

• Event Collector: The Event Collector serves as the initial ingestion point for log data from a wide range of sources, such as firewalls, servers, cloud platforms, and endpoint devices. It normalizes raw security data into a standardized format, ensuring seamless downstream analysis and reducing processing bottlenecks.

- *Example: Logs from AWS CloudTrail highlighting IAM policy changes, Palo Alto firewalls indicating unusual traffic, and Linux server logs capturing unauthorized access attempts are aggregated and converted for analysis.*
- *Real-World Scenario: A global e-commerce company relies on Event Collectors to gather transaction logs, API calls, and network activity data, enabling the detection of fraudulent purchase patterns during high-traffic sales events.*

• Event Processor: The Event Processor is the analytical engine of QRadar, correlating ingested events to uncover complex attack patterns and prioritize critical threats. Using advanced rule-based detection, machine learning, and behavioral analysis, it provides SOC teams with actionable alerts.

- *Example: Detecting brute-force login attempts across multiple endpoints by identifying repeated failed login attempts from varying IP addresses, followed by a successful login.*
- *Real-World Scenario: During a ransomware campaign, the Event Processor identifies unusual file encryption activities on user endpoints and correlates them with network traffic anomalies, alerting SOC teams to initiate containment protocols.*

• Console: The Console acts as the centralized interface for SOC teams, providing visualization tools, investigation capabilities, and compliance report generation. Its intuitive design ensures efficient navigation, enabling analysts to manage both routine monitoring and critical incidents effectively.

- *Example: Using the Console's timeline view to track the progression of a malware outbreak, from initial compromise to lateral movement within the network.*
- *Real-World Scenario: A financial institution's SOC utilizes the Console to produce PCI DSS-compliant reports, detailing security incidents and their resolution, ensuring adherence to regulatory standards during audits.*

Together, these components ensure that QRadar delivers a scalable and unified approach to cybersecurity, capable of adapting to the evolving demands of industries ranging from healthcare to finance. The platform's architecture not only facilitates seamless integration with existing security tools but also provides the agility required to manage the dynamic nature of cyber threats in today's interconnected digital landscape.

3.7.2.3 Key Features of QRadar

• Data Collection: QRadar excels in ingesting and consolidating logs from an extensive range of sources, such as traditional on-premises systems, modern cloud environments, and hybrid infrastructures. This consolidation provides a unified view for comprehensive analysis.

- *Example: QRadar aggregates network traffic flows from Cisco routers, user activity logs from Active Directory, and application logs from AWS Lambda to enable deep cross-correlation.*
- *Real-World Scenario: A multinational retail chain collects logs from geographically dispersed stores and cloud-hosted e-commerce platforms. QRadar identifies anomalies such as unusual payment gateway transactions originating from unverified IP addresses.*

• Threat Detection: Leveraging its advanced correlation engine and machine learning algorithms, QRadar can identify threats in real time. It uses predefined rules and learns behaviors over time to detect anomalies and malicious activities effectively.

- *Example: Alerting on excessive data exfiltration attempts from an endpoint to an unrecognized foreign server.*
- *Real-World Scenario: During a supply chain attack, QRadar detects lateral movement between a compromised vendor system and the organization's internal network by correlating login attempts, file access logs, and unusual traffic spikes.*

• Incident Response: QRadar's integrated workflows streamline the detection, containment, and recovery processes, reducing response times and minimizing the operational impact of incidents. Automated playbooks help SOC teams address threats efficiently.

- *Example: Automatically isolating a server from the network when malware is detected and triggering an alert to notify the SOC team.*
- *Real-World Scenario: In a ransomware incident, QRadar integrates with SOAR tools to trigger an automated response: locking affected accounts, quarantining endpoints, and initiating a root cause analysis workflow. 8.2.4 QRadar Strengths and Weaknesses*

Strengths:

• Advanced correlation engine that enables comprehensive threat analysis across diverse data sources. • Integrated threat intelligence that enhances the accuracy of alerts and provides context for decision-making.

Weaknesses:

• Complexity in configuration and deployment, which may require specialized expertise. • High resource requirements, particularly for large-scale deployments, which can increase operational costs.

3.7.3 Security Best Practices and Standards

3.7.3.1 NIST Cybersecurity Framework

The NIST Cybersecurity Framework serves as a comprehensive guideline for organizations to manage and mitigate cybersecurity risks effectively. Its structured methodology encompasses the entire lifecycle of cybersecurity efforts, categorized into five critical functions:

1. *Identify: Gain a comprehensive understanding of cybersecurity risks, assets, and system vulnerabilities. This step includes mapping out interdependencies that may affect organizational operations.*
    - *Example: Creating a detailed inventory of IT assets, including hardware, software, and third-party services, to pinpoint areas vulnerable to cyberattacks.*
    - *Real-World Scenario: A large healthcare network conducts vulnerability scans on its medical devices and EHR systems, ensuring compliance with HIPAA and safeguarding patient data from potential breaches.*
2. *Protect: Establish safeguards to secure critical infrastructure and mitigate risks effectively. These measures include physical, technical, and administrative controls.*
    - *Example: Deploying role-based access controls (RBAC) and implementing encryption for sensitive data in transit and at rest.*

- *Real-World Scenario: A financial institution implements zero-trust architecture and multi-factor authentication (MFA) to protect transaction systems from unauthorized access and potential fraud.*

3. *Detect: Implement monitoring systems and tools to identify cybersecurity incidents swiftly. This involves correlating data from multiple sources to detect anomalies and potential threats.*
    - *Example: Using IBM QRadar's advanced threat analytics to identify unusual outbound traffic that may signify data exfiltration.*
    - *Real-World Scenario: An e-commerce platform detects a sudden surge in API calls from a single IP address, alerting the SOC team to a potential botnet attack attempting credential stuffing.*
4. *Respond: Develop and execute actionable plans to minimize the impact of detected cybersecurity incidents. This includes communication protocols and containment strategies.*
    - *Example: Activating an incident response workflow that isolates affected servers during a malware outbreak.*
    - *Real-World Scenario: A manufacturing company activates its ransomware response plan, disabling infected systems and coordinating with legal and public relations teams to mitigate reputational damage.*
5. *Recover: Implement processes to restore normal operations promptly after a cybersecurity event, incorporating lessons learned to enhance future preparedness.*
    - *Example: Regularly testing and updating disaster recovery plans to ensure minimal downtime during crises.*
    - *Real-World Scenario: Following a DDoS attack, a telecom provider deploys upgraded network redundancies and real-time traffic monitoring tools to prevent recurrence.*

3.7.3.2 ISO 27001

ISO 27001 offers an internationally recognized framework for managing information security risks systematically. It emphasizes continuous improvement and alignment with organizational objectives through a risk-based approach. Key steps include:

• Conducting thorough risk assessments to identify and prioritize security threats across all organizational layers. • Implementing controls tailored to mitigate identified risks, such as employing encryption protocols, securing physical data centers, and deploying intrusion detection systems. • Establishing a continuous audit and review cycle to ensure compliance with the ISO 27001 standard, addressing gaps and enhancing security posture.

Organizations that adopt ISO 27001 not only strengthen their information security framework but also build trust among stakeholders, demonstrating their commitment to protecting sensitive data against cyber threats and meeting regulatory requirements.

3.7.3.3 Importance of Data Security, Threat Intelligence, and Incident Response Planning

A robust cybersecurity strategy hinges on three interconnected pillars: data security, threat intelligence, and incident response planning. Together, these elements fortify an organization's defense against evolving cyber threats:

• Data Security: Focuses on protecting sensitive information from unauthorized access or breaches, ensuring its confidentiality, integrity, and availability.

- *Example: Encrypting customer data and implementing data masking techniques to prevent exposure during cyberattacks.*
- *Real-World Scenario: A retail chain employs tokenization for payment data, ensuring customer credit card information remains secure even during potential breaches of backend systems.*

• Threat Intelligence: Provides actionable insights into emerging attack vectors and adversary tactics, enabling organizations to stay ahead of threats.

- *Example: Integrating real-time threat intelligence feeds into IBM QRadar to flag known malicious IP addresses or phishing URLs.*
- *Real-World Scenario: A government agency collaborates with an international threat intelligence consortium to proactively block IPs associated with nation-state attacks targeting critical infrastructure.*

• Incident Response Planning: Ensures swift, coordinated actions during security breaches to mitigate damage and restore normalcy.

- *Example: Creating playbooks that guide SOC teams through specific incidents, such as phishing attempts or ransomware outbreaks.*

- *Real-World Scenario: An airline activates its incident response plan following a cyberattack that compromises passenger data, isolating affected systems and notifying regulators within mandatory timeframes.*

By integrating these practices, organizations can create a resilient cybersecurity framework, ensuring business continuity while adhering to regulatory mandates like GDPR and HIPAA.

3.8. Data Collection and Preparation

3.8.1 Data Source Identification and Prioritization

To ensure comprehensive monitoring and analysis, the following critical log sources have been identified and prioritized:

- *Firewalls: Firewalls are integral to network security, acting as the gatekeepers that manage and monitor all inbound and outbound traffic. They log events such as blocked attempts to access restricted resources, flagged IP addresses, and protocols used. For example, a financial institution's firewall may detect repeated failed login attempts from a specific region, signaling a potential brute force attack aimed at accessing sensitive customer data.*
- *Servers: Servers, whether they are application servers, database servers, or file servers, generate a plethora of logs. These logs capture critical operational metrics such as resource usage, error events, and successful or failed login attempts. In a real-world scenario, web server logs can reveal unexpected spikes in traffic caused by a Distributed Denial of Service (DDoS) attack targeting an e-commerce website during a holiday sale.*
- *Endpoints: Endpoint devices, including desktops, laptops, and mobile devices, are often the entry points for security threats. Endpoint Detection and Response (EDR) systems collect data on actions such as file modifications, USB connections, or the execution of unknown processes. For instance, detecting unauthorized software installations on a corporate laptop can prevent data breaches caused by malware.*
- *Applications: Applications provide logs that are essential for operational troubleshooting and security monitoring. These logs often include user activities, application errors, and transaction records. A case in point is a banking application that logs customer transactions, enabling fraud detection teams to identify irregular patterns such as multiple high-value transactions within minutes.*

3.8.2 Data Collection Rules and Normalization

To streamline and enhance data analysis, the following procedures are implemented:

- *Log Ingestion: Develop and implement comprehensive ingestion rules to ensure all relevant data streams are captured efficiently. For example, integrating a Security Information and Event Management (SIEM) tool allows an organization to gather logs from diverse sources such as Active Directory, cloud applications, and network appliances in near real-time. This centralization minimizes blind spots in monitoring activities.*
- *Data Normalization: Data normalization involves standardizing log formats to enable uniform analysis, regardless of the source. For example, timestamps are converted to Coordinated Universal Time (UTC) to ensure chronological consistency across global operations. Normalization also includes mapping different terminologies used by devices to a common schema, such as translating "deny" from one device's log into a standardized "block" action.*

3.8.3 Data Storage and Retention Planning

Efficient data storage and retention are critical for both operational efficiency and compliance with legal standards. The following measures are undertaken:

- *Storage Requirements: Determining storage needs requires understanding log generation rates and retention policies. For example, an organization processing high volumes of transactions might generate terabytes of log data daily, necessitating robust storage solutions like AWS S3 or Azure Blob Storage. Additionally, tiered storage strategies can be employed to balance cost and performance, where recent logs are stored on high-speed disks and older logs are archived to economical storage tiers.*
- *Compliance Considerations: Organizations must adhere to industry regulations and standards, such as GDPR for personal data protection, HIPAA for healthcare data, or PCI DSS for payment information. For instance, GDPR mandates that logs containing personal information be encrypted during storage and access restricted to authorized personnel. Regular audits and automated compliance checks ensure that data handling practices align with these requirements, mitigating risks of non-compliance and associated penalties.*

3.9. Threat Detection and Response

3.9.1 Developing and Deploying Custom Rules and Use Cases

Custom rules and use cases are critical for enhancing threat detection by addressing specific organizational risks and operational contexts. Detailed examples and scenarios include:

- *Brute-Force Attack Detection: Brute-force attacks involve repeated attempts to gain unauthorized access to systems by systematically guessing credentials. Such attacks are particularly prevalent on remote access systems, corporate VPNs, and cloud-based applications. Custom detection rules can:*
    - *Monitor login failures across multiple systems and correlate them to identify patterns indicative of brute-force activity.*
    - *Implement thresholds, such as flagging an IP address that generates 50 failed login attempts within 10 minutes.*

Real-World Example: In a financial services company, a detection rule flags a suspicious IP address attempting repeated logins to a privileged user's account on the corporate VPN portal. The source of the attacks was traced to an anonymized IP from a known threat actor group. By automatically blocking the IP and notifying the security team, a potential data breach was thwarted.

- *Malware Infections: Malware infections can infiltrate systems through phishing emails, malicious downloads, or compromised software. Advanced detection focuses on:*
    - *Identifying anomalous behaviors such as unexpected process executions, unusual outbound network traffic to untrusted domains, or rapid file encryption.*
    - *Using event correlation to detect multi-stage attacks, such as malware installation followed by credential harvesting.*

Real-World Example: A retail organization's Security Information and Event Management (SIEM) system detects unusual traffic from a workstation to a known Command and Control (C2) server. A deep dive reveals malware attempting to encrypt customer transaction data. The detection rules not only flagged the anomalous network behavior but also correlated it with recent unauthorized downloads, enabling rapid isolation and remediation before significant damage occurred.

- *Phishing Attack Detection: Custom rules can also be designed to detect phishing-related threats. For example, monitoring inbound emails for suspicious attachments, links redirecting to non-corporate domains, or spoofed sender addresses can help mitigate risks.*

Real-World Example: An energy company observed a pattern of emails flagged by its SIEM system, containing embedded links mimicking a vendor's portal. These emails targeted finance team members. The rules flagged these emails based on their domain mismatch and attachment properties, enabling the team to implement additional email filtering policies and user training.

3.9.2 Incident Response Procedures and Workflows

Incident response is a systematic and critical process for managing and mitigating security events. Each step is designed to ensure swift containment and resolution, minimizing impact on business operations and preventing recurrence. Below is a highly detailed breakdown of the key steps:

1. ***Investigation:***
    - *Objective: Analyze alerts to determine their validity and scope.*
    - *Details: Review system logs, correlate events across different sources, and verify whether a flagged alert indicates a genuine security incident or a false positive. Investigation tools like log analyzers and forensic tools can assist in tracing the root cause.*
    - *Real-World Example: In a healthcare organization, an alert indicates unauthorized access to a patient database. By investigating correlated logs from the access management system and application layer, the team discovers a compromised employee credential used to bypass controls. This step aids in identifying the point of compromise and scope of access.*
2. ***Containment:***
    - *Objective: Isolate affected systems to prevent the threat from spreading.*
    - *Details: Immediate actions could include disabling user accounts, blocking malicious IPs, or disconnecting infected devices from the network. Containment should balance stopping the spread and maintaining critical operations.*
    - *Real-World Example: A manufacturing company identifies ransomware spreading through its internal file-sharing server. The IT team isolates the server from the network, stopping the ransomware from encrypting additional systems while preserving data for forensic analysis.*
3. ***Eradication:***
    - *Objective: Eliminate the threat from all affected systems.*
    - *Details: Deploy patches, remove malware, reconfigure security controls, or uninstall malicious software. Advanced steps may include wiping infected systems and restoring them from clean backups. Conducting a root cause analysis ensures eradication efforts address all threat vectors.*
    - *Real-World Example: In an educational institution, antivirus scans on infected workstations identify and remove a trojan that was exfiltrating*

student data. After eradication, endpoint security policies are updated to block the attack vector used by the malware.

4. ***Recovery:***
    - *Objective: Restore systems to normal operations while ensuring that no remnants of the threat remain.*
    - *Details: Validate that restored systems are fully operational, confirm data integrity, and monitor for recurrence. Incorporating redundancy measures can accelerate recovery in future incidents.*
    - *Real-World Example: An online retailer affected by a web application compromise restores its affected servers using backups, applies application patches, and conducts a full post-recovery audit to confirm security improvements before resuming operations.*
5. ***Post-Incident Review:***
    - *Objective: Learn from the incident to improve future defenses.*
    - *Details: Conduct a formal review documenting the attack's timeline, effectiveness of the response, and areas needing improvement. Share findings with stakeholders and update incident response plans accordingly.*
    - *Real-World Example: A financial institution conducts a post-mortem after a phishing attack, revealing gaps in employee training and email filtering. They roll out targeted training and deploy AI-based email filtering solutions to reduce future risks.*

3.9.3 Integration with SOAR and Other Security Tools

Integrating detection systems with additional tools enhances the efficiency and effectiveness of security operations. Examples include:

- *SOAR Platforms: Security Orchestration, Automation, and Response (SOAR) tools automate workflows, such as triggering incident response actions based on predefined conditions. For example, integrating QRadar with a SOAR platform can automatically initiate containment actions upon detecting ransomware activity.*
- *Vulnerability Scanners: Integrate with tools like Nessus or Qualys to enrich threat detection. For instance, cross-referencing vulnerability scan results with active threat intelligence helps prioritize remediation efforts for critical vulnerabilities.*

3.10. QRadar Implementation

3.10.1 QRadar Installation and Configuration

The installation and configuration of QRadar involve the following comprehensive steps to ensure a reliable and effective deployment:

1. ***Setting up hardware/virtual appliances:***
    - *For physical deployments, ensure that servers are procured with specifications recommended by IBM QRadar, such as Intel Xeon processors, at least 64GB RAM, and 1TB SSD storage. These configurations support high log ingestion rates and ensure optimal performance.*
    - *In virtualized environments, allocate resources following QRadar's sizing guide. For example, a deployment handling 5,000 EPS (Events Per Second) should be equipped with at least 16 virtual CPUs, 32GB of memory, and sufficient disk space to handle data retention and indexing.*
    - *Perform hardware validation tests to confirm the compatibility and stability of appliances or virtual systems before initiating the QRadar installation.*
    - *Real-world example: A multinational bank adopted a hybrid deployment model using dedicated appliances for on-premises compliance logging and virtual instances for global threat correlation. This allowed regional data segregation and regulatory compliance without compromising centralized visibility.*
2. ***Configuring the network:***
    - *QRadar components should be deployed within a dedicated and secured network segment, typically within a security operations VLAN, to isolate it from general traffic. Use firewalls and access control lists (ACLs) to restrict communication to trusted sources.*
    - *Assign static IPs for all QRadar appliances, ensuring consistent communication and ease of management. DNS resolution must be configured correctly for seamless system operations, as misconfigured entries can lead to log ingestion delays or system misbehavior.*
    - *Use encrypted channels like Secure Shell (SSH) and TLS for secure communication between QRadar components and log sources.*
    - *Real-world example: An enterprise retail company leveraged a dedicated 10Gbps link to forward logs from its distributed point-of-sale systems to*

QRadar, ensuring real-time analysis without impacting customer transaction times.

3. ***Performing the initial setup:***
    - *Launch the QRadar setup wizard to configure licensing, time zones, data retention policies, and system administrator accounts. Proper initial setup ensures that the appliance operates within organizational security and compliance frameworks.*
    - *Configure storage partitions during setup to accommodate expected log volumes and retention requirements. Set up alerting mechanisms for storage thresholds to prevent unexpected system downtimes.*
    - *Establish high availability (HA) configurations for critical environments to maintain uninterrupted operations during system failures or maintenance.*
    - *Real-world example: A large retail chain configured QRadar to hold logs for 90 days in compliance with PCI DSS, while also integrating cloud storage for long-term archival, ensuring both compliance and scalability.*

3.10.2 Data Source Configuration

To ensure seamless and accurate data ingestion, follow these highly detailed steps to configure log sources effectively and optimize QRadar's functionality:

- *Identify critical log sources: Start by prioritizing the collection of logs from essential systems and devices such as firewalls (e.g., Palo Alto, Cisco ASA), endpoint protection tools (e.g., CrowdStrike, Symantec), cloud services (e.g., AWS CloudTrail, Azure Monitor), and identity management systems (e.g., Okta, Active Directory). Include industry-specific sources like SCADA systems for manufacturing or EHR systems in healthcare. This ensures the ingestion of data relevant to business operations and security.*
- ***Deploy log collectors or agents: Utilize appropriate mechanisms to forward logs to QRadar:***
    - *For Unix-based systems, configure remote logging using syslog over TCP/UDP. Ensure encryption with TLS to maintain data confidentiality during transmission.*
    - *For Windows systems, deploy WinCollect agents to gather Windows Event Logs efficiently. Customize WinCollect configurations to prioritize critical event IDs such as authentication failures (Event ID 4625) or privilege escalation attempts.*
    - *Leverage APIs for advanced integrations like pulling logs from SaaS platforms or proprietary tools. For instance, connect QRadar with*

Salesforce or Office 365 through their API endpoints for activity monitoring.

- *Set up log source auto-discovery: Enable QRadar's auto-discovery functionality to detect and classify new log sources automatically. Use the DSM Editor for custom parsers if unsupported log formats are encountered, ensuring seamless ingestion.*
- *Verify log parsing and categorization: Regularly validate the accuracy of ingested logs using the Log Activity tab in QRadar. Analyze log fields to ensure proper categorization of events, such as differentiating authentication failures, malware detections, and system configuration changes. Misparsed logs can lead to gaps in threat detection.*
- *Optimize log retention and storage: Configure QRadar to archive less critical logs to external storage while retaining high-priority logs (e.g., critical alerts or compliance-related logs) on primary storage for quick access. Use the Data Retention Settings to balance storage use with regulatory requirements.*
- *Real-world example: A healthcare organization configured logs from their electronic health record (EHR) system, specifically focusing on audit logs related to patient data access. They implemented rules to detect anomalies, such as repeated access to high-profile patient records by a single user, ensuring compliance with HIPAA. Additionally, integration with cloud services allowed real-time alerting on unauthorized data exports.*

By implementing these steps, organizations can achieve efficient log ingestion and parsing, enabling QRadar to provide actionable insights while adhering to compliance standards.

3.10.3 User Roles and Access Controls

Implementing robust access controls is critical to maintaining the security and integrity of the QRadar environment. Follow these detailed steps to define and manage user roles and permissions effectively:

- ***Role-Based Access Control (RBAC):***
    - *Define roles for various user groups, such as SOC analysts, engineers, and administrators. For instance:*
        - *Level 1 SOC Analysts: Grant read-only access to log activities and dashboards, allowing them to monitor incidents without altering system configurations.*
        - *Level 2 SOC Analysts: Provide permissions to manage incidents, update statuses, and execute predefined response actions.*

        - *Administrators: Allow full access to system settings, including rule configuration, data source management, and network integrations.*
    - *Segregate roles based on job responsibilities to minimize the risk of accidental or intentional misuse.*
    - *Real-world example: A large e-commerce platform assigned granular permissions to its SOC team, ensuring that junior analysts could only view alerts while senior analysts could take remediation actions. This separation reduced the likelihood of errors and improved response efficiency.*

- ***Integration with LDAP/Active Directory:***
    - *Use QRadar's built-in integration with LDAP or Active Directory to automate user management and ensure alignment with organizational policies. Map roles within QRadar to existing Active Directory groups, enabling seamless onboarding and offboarding of users.*
    - *Implement Multi-Factor Authentication (MFA) for administrative accounts to enhance security.*
    - *Example: A financial institution leveraged LDAP integration to automatically provision user accounts for its rotating SOC staff, reducing manual overhead and ensuring compliance with internal access policies.*

- ***Periodic Access Reviews:***
    - *Conduct quarterly or biannual reviews of user access to detect and address privilege creep—a scenario where users accumulate permissions over time beyond their current roles.*
    - *Use QRadar's built-in audit logging to monitor changes to user roles and identify unauthorized modifications.*
    - *Revoke inactive or unnecessary accounts promptly to reduce potential attack vectors.*
    - *Real-world example: An energy company's SOC team discovered during an audit that several former employees still had active accounts in QRadar. By enforcing periodic reviews, they removed these accounts and reduced insider threat risks.*

- ***Customizing Access for Third Parties:***
    - *Create temporary, restricted accounts for external consultants or auditors. Use QRadar's time-based access control features to automatically disable these accounts after the required period.*
    - *Monitor third-party activities closely through QRadar's user activity logs to ensure compliance with contractual agreements.*
- ***Advanced Use Case:***
    - *A government agency implemented fine-grained access controls to restrict users from exporting sensitive log data or modifying critical system configurations. This approach ensured compliance with national security policies and protected against insider threats.*

By employing these measures, organizations can ensure that only authorized personnel have access to QRadar's functionalities, safeguarding sensitive data and maintaining operational integrity.

3.10.4 Custom Rule and Use Case Deployment

To maximize QRadar's effectiveness in detecting and mitigating security threats, organizations should meticulously design and deploy custom rules and use cases tailored to their specific requirements:

- *Develop Use Cases Aligned with Business Priorities:*
    - *Identify critical business assets and operations that require enhanced monitoring. For instance, detecting privilege escalation or data exfiltration attempts should be a high priority for organizations handling sensitive data, such as financial institutions or healthcare providers.*
    - *Translate organizational policies into actionable QRadar rules. For example, configure a rule to alert when sensitive customer data is accessed outside of business hours.*
    - *Continuously engage with stakeholders to align use cases with evolving business objectives and emerging threats.*
- ***Utilize the QRadar Rule Wizard for Precision:***
    - *Use the Rule Wizard to define specific conditions and thresholds. For instance, create a rule that triggers an alert if "the failed login count exceeds 10 within 5 minutes from a single IP," indicating a potential brute force attack.*

- *Leverage building blocks to group similar conditions for efficiency. For example, aggregate alerts for repeated suspicious activity across multiple endpoints within a defined timeframe.*
- *Validate rule logic using test data to ensure accuracy and reliability before deployment.*

- ***Optimize Rules to Minimize False Positives:***
  - *Regularly review and refine existing rules to eliminate false positives that may overwhelm analysts. For example, fine-tune login failure alerts to exclude known maintenance windows or trusted IPs.*
  - *Implement reference sets to dynamically whitelist trusted entities such as corporate VPNs, reducing unnecessary noise.*
  - *Use the QRadar's Anomaly Detection features to identify deviations from normal behavior patterns rather than relying solely on static thresholds.*
- ***Monitor Rule Performance and Adjust as Needed:***
  - *Utilize QRadar's Rule Performance Monitor to identify inefficient rules that consume excessive system resources. Rewrite or disable these rules as necessary.*
  - *Conduct regular reviews of rule outputs to ensure continued alignment with security objectives.*
- ***Real-World Example:***
  - *A manufacturing firm integrated IoT device logs into QRadar and created custom rules to detect unauthorized access attempts. For instance, a rule was implemented to alert when specific production line controllers were accessed from external IPs. This enhanced visibility enabled the organization to respond to cyberattacks 30% faster, safeguarding critical operations.*
  - *In another instance, a global retail company developed rules to flag unusual login patterns, such as multiple failed attempts followed by a successful login from a foreign IP, which helped mitigate credential theft incidents.*

By following these steps, organizations can effectively deploy custom rules and use cases to enhance threat detection, streamline SOC operations, and align QRadar's capabilities with their unique security and business needs.

3.11. Testing and Validation

3.11.1 Testing Methodology

To ensure QRadar functions as intended, adopt a comprehensive and structured testing methodology that validates every component and process within the system:

- ***Unit Testing:***
    - *Validate individual rules, log parsers, and detection mechanisms in isolation. For example, simulate specific attack scenarios like brute force attempts to ensure that QRadar triggers alerts based on predefined rules.*
    - *Perform syntax validation and logical checks for all custom rules to confirm they operate as expected without errors or misfires.*
    - *Example: In one scenario, an organization tested a custom rule to detect unauthorized SSH attempts. They simulated 50 failed login attempts within 5 minutes, confirming that QRadar generated a high-severity alert and logged the activity accurately.*
- ***Integration Testing:***
    - *Validate the seamless interaction between QRadar and external tools such as threat intelligence platforms (e.g., IBM X-Force), SIEM connectors, or ticketing systems like ServiceNow. Ensure bidirectional communication functions correctly.*
    - *Test the accuracy of data mapping and transformation when ingested from external sources to verify QRadar parses logs appropriately.*
    - *Example: An integration test was conducted to verify automated ticket creation for high-priority alerts in ServiceNow. The system generated real-time incident reports, reducing incident response times by 40% and ensuring immediate escalation of critical events.*
- ***Stress Testing:***
    - *Simulate high log ingestion rates and evaluate QRadar's performance under heavy loads. For example, test the system with 10,000 EPS (Events Per Second) to ensure it processes logs without delays or system crashes.*
    - *Evaluate storage management under stress conditions, ensuring that retention policies and alerting mechanisms work as intended.*

- *Example: A retail chain simulated peak holiday season traffic, forwarding logs from their point-of-sale systems to QRadar. The test validated consistent alert generation during a 20% increase in activity.*

- ***User Acceptance Testing (UAT):***
  - *Collaborate with key stakeholders, including SOC analysts and system administrators, to validate that QRadar's outputs meet business and security requirements.*
  - *Present tailored demonstrations or simulated attack scenarios to confirm alignment with key performance indicators (KPIs) defined in the project scope. For example, demonstrate how QRadar detects and escalates unauthorized data access attempts.*
  - *Gather feedback from end-users to refine detection rules, dashboards, and workflows for improved usability.*

By employing this multi-faceted testing methodology, organizations can validate QRadar's accuracy, reliability, and alignment with operational goals while optimizing its performance and functionality in real-world environments.

12.2 Validation of Threat Detection and Response Effectiveness
Validating QRadar's threat detection and response capabilities involves comprehensive and practical testing scenarios to ensure operational readiness and robust security. Follow these detailed steps:

- ***Conduct Advanced Penetration Tests:***
  - *Utilize industry-standard tools like Metasploit, Caldera, and Cobalt Strike to simulate real-world attack scenarios. Examples of tests include:*
    - *SQL Injection: Test applications connected to QRadar by executing injection attempts to assess if QRadar identifies malicious queries.*
    - *Phishing Simulation: Generate simulated phishing emails and track whether QRadar detects anomalous email behaviors or unauthorized access following credential theft.*
    - *Malware Delivery: Deploy benign but realistic payloads to endpoints to monitor QRadar's ability to flag suspicious file behavior and communication with Command-and-Control servers.*
  - *Example: A multinational corporation tested QRadar's detection by simulating ransomware spread in their network, validating alert generation on encryption activities and unusual file operations.*

- ***Assess Detection Speed and Alert Accuracy:***
    - *Measure the time QRadar takes to generate alerts for detected threats, ensuring minimal delay between event detection and alert creation.*
    - *Analyze alert accuracy by reviewing logs and verifying whether QRadar flags legitimate threats while minimizing false positives. For instance, test rule efficiency in distinguishing between failed authentication due to mistyped credentials and brute-force attack attempts.*
    - *Example: An e-commerce company tested alerts for privilege escalation by simulating unauthorized access to admin accounts, ensuring high-priority alerts were generated promptly without triggering for legitimate admin activities.*
- ***Simulate Insider Threats and Advanced Persistent Threats (APTs):***
    - *Mimic behaviors such as unauthorized data access, privilege escalation, and lateral movement within the network. Use scenarios like:*
        - *A terminated employee attempting to access sensitive databases.*
        - *A compromised endpoint performing reconnaissance across internal systems.*
    - *Verify if QRadar provides visibility into such activities and generates contextually rich alerts for SOC analysts to act upon.*
    - *Real-world Example: A financial institution simulated insider threats by mimicking unauthorized database access. QRadar successfully generated high-severity alerts and enriched the logs with user activity details, enabling rapid containment.*
- ***Validate Response Playbooks:***
    - *Test the integration of QRadar with Security Orchestration, Automation, and Response (SOAR) platforms. For instance, simulate a malware alert and validate automated responses, such as quarantining affected devices or revoking user access.*
    - *Example: An organization tested a response playbook where QRadar alerts triggered automated ticket creation in ServiceNow and blocked suspicious IPs through firewall integration.*

By simulating realistic attack vectors and rigorously validating QRadar's capabilities, organizations can achieve confidence in their threat detection framework, minimize response times, and ensure compliance with regulatory standards.

3.12. Results and Analysis

3.12.1 Data Ingestion and Processing Analysis

To comprehensively evaluate QRadar's data ingestion and processing, a detailed analysis was conducted focusing on volume, quality, and efficiency. Below are the expanded insights:

- ***Volume and Quality of Ingested Data:***
    - *Assessment of Log Volume: Measure the total volume of logs ingested over specific intervals, categorized by source types like firewalls, endpoints, cloud platforms, and applications. Use QRadar's dashboards to visualize EPS trends during both normal and peak activity periods. For example, peak activity during a simulated attack could show 20,000 EPS across 25 sources.*
    - *Parsing Accuracy: Regularly validate log parsing accuracy by cross-referencing raw logs with processed events to ensure correct categorization. Anomalies in parsing accuracy—like misclassified logs or missing critical fields—can drastically reduce QRadar's detection efficacy. In one test, parsing accuracy of 95% was improved to 98% by implementing custom DSMs for unsupported log formats.*
    - *Actionable Events vs. Noise: Assess the percentage of noise versus actionable events to determine filtering efficiency. Noise reduction strategies include fine-tuning log source settings and disabling redundant event generation.*
        - *Real-world Example: An enterprise ingested 15,000 EPS, with 80% identified as actionable and only 5% flagged as noise. By optimizing event filters, SOC analysts reduced manual review times by 40%.*
- ***Processing Efficiency and Bottlenecks:***
    - *Latency Analysis: Measure end-to-end latency from log ingestion to actionable insight. Use QRadar's Performance Monitor to pinpoint delays. Delays greater than 2 seconds for critical alerts might indicate issues such as underpowered event processors or misconfigured network settings.*
    - *Bottleneck Identification: Investigate hardware and software constraints. Misconfigured log source settings or inadequate memory allocations are common culprits. QRadar's Resource Monitor provides insights into CPU and RAM usage, helping identify processing bottlenecks.*

- *Optimization Insights: Implement incremental resource scaling and optimize correlation rule logic to mitigate bottlenecks. For instance, complex rules were simplified to reduce processing time by 30%.*
    - *Real-world Example: During a stress test simulating a DDoS attack, processing speeds initially lagged by 2.5 seconds. After increasing memory by 25% and optimizing the event pipeline, latency dropped to 1.8 seconds, improving real-time detection capabilities.13.2 Threat Detection and Response Results*
      This section presents detailed outcomes of QRadar's performance in detecting and responding to threats:

- ***Number of Detected Incidents:***
    - *Provide detailed statistics, segmented by severity levels (e.g., low, medium, high, critical). For instance, a 30-day analysis might reveal 200 detected incidents, with 15 categorized as critical threats requiring immediate response.*
    - *Real-world Example: QRadar detected 200 incidents over a month, including 15 critical threats, such as unauthorized access attempts to privileged accounts. Critical alerts were escalated within 2 minutes for rapid mitigation.*

- ***Types of Threats Identified:***
    - *Categorize threats by type (e.g., malware infections, insider threats, phishing, brute force attacks).*
    - *Analyze patterns, such as repeated unauthorized access attempts from specific geolocations or increased malware incidents following phishing campaigns.*
    - *Real-world Example: Over 50% of detected incidents were phishing-related, primarily targeting administrative credentials. QRadar's correlation rules flagged unusual login attempts from IP addresses in high-risk regions, enabling proactive countermeasures.*

- ***Response Times:***
    - *Measure the time from alert generation to incident resolution. Break down metrics for different severity levels.*

- *Real-world Example: The average response time for critical incidents was 10 minutes, with containment actions completed within 5 minutes for 90% of cases, minimizing potential damage.*

3.12.3 Performance Evaluation

Evaluate QRadar's overall system performance, focusing on resource utilization and responsiveness:

- ***Resource Utilization (CPU, Memory, Storage):***
  - *Detailed Analysis of Resource Consumption: Monitor and document system resource usage, including CPU, memory, and storage trends, across different operation periods—normal, peak, and stress scenarios. This ensures accurate baseline measurements for capacity planning.*
  - *CPU Utilization Insights: Track average and peak CPU usage during high EPS periods to identify thresholds where performance might degrade. Investigate spikes to ensure they correlate with expected load increases, such as batch log uploads or heavy query processing.*
  - *Memory Usage Metrics: Evaluate memory allocation for various QRadar components, including the Event Processor, Flow Processor, and Data Node. Proactively address memory saturation to prevent system lag or crashes.*
  - *Storage Monitoring: Analyze storage capacity usage trends to ensure log retention policies align with compliance requirements. Implement tiered storage solutions for archiving older logs without impacting active database performance.*
  - *Real-world Example: CPU utilization during normal operations averaged 60% but spiked to 85% during incident surges, such as a simulated malware attack generating 25,000 EPS. Memory usage remained stable at 70%, but storage reached 90% capacity within three months, prompting the implementation of automated log archival policies to free up space for new data.*
- ***System Response Times:***
  - *Latency Monitoring for Key Operations: Measure system latency for critical functions, such as rule execution, report generation, and dashboard loading. Identify patterns where delays might occur under stress conditions and correlate them with resource bottlenecks.*

- *Optimization Strategies: Streamline complex rule logic and prioritize high-severity alerts to reduce processing overhead. Introduce indexing optimizations for faster query execution.*
- *Real-world Example: Rule execution times averaged 1.5 seconds during regular conditions, ensuring real-time threat detection capabilities. However, during peak usage scenarios, complex report generation times extended to 8 seconds, highlighting the need for query optimization. After implementing indexing improvements and refining correlation logic, report generation times were reduced to 4 seconds, significantly enhancing SOC analyst productivity.*

- ***System Response Times:***
  - *Measure latency for key operations, such as rule execution, dashboard loading, and report generation.*
  - *Real-world Example: Rule execution times averaged 1.5 seconds, ensuring real-time detection capabilities. Complex report generation took up to 5 seconds, highlighting the need for additional optimization during peak reporting periods.*

3.12.4 Security Reporting and Analysis

A detailed evaluation of QRadar's reporting and dashboard functionalities was conducted to provide actionable insights into security trends and identify potential areas for improvement. Below are the expanded findings:

- ***Highlight Security Trends:***
  - *Visualization of Threat Trends: QRadar dashboards were leveraged to identify emerging threats and recurring attack patterns. Trends such as a spike in phishing incidents during specific times (e.g., holiday seasons) were visualized using time-series analytics. Dashboards also provided insights into geolocated attack sources, highlighting high-risk regions.*
  - *Analysis of Attack Vectors: By categorizing attacks by type (e.g., malware, phishing, insider threats), QRadar dashboards allowed the identification of dominant threat vectors within specific organizational units or departments.*
  - *Real-world Example: Dashboards revealed a 20% increase in phishing attempts during the holiday season, with most attacks targeting finance department employees. This prompted the implementation of stricter email filtering rules, enhanced phishing awareness training, and simulated phishing campaigns to improve resilience.*

- ***Identify Improvement Areas:***
    - *Detection Rule Analysis: QRadar reports were analyzed to identify gaps in existing threat detection rules. For instance, gaps in lateral movement detection were highlighted, necessitating the creation of tailored correlation rules for better visibility.*
    - *Operational Inefficiency Insights: Reports provided metrics such as false positive rates and alert fatigue levels, allowing the identification of overly sensitive rules or redundant alerts. QRadar's reporting functionality also highlighted specific log sources that were underutilized or misconfigured, impacting detection accuracy.*
    - *Optimization Recommendations: Recommendations included refining correlation rules, updating reference sets, and implementing dynamic thresholds to reduce false positives and improve detection precision.*
    - *Real-world Example: Reports indicated a 40% false positive rate in brute-force attack alerts due to an overly broad rule definition. After rule refinement, false positives were reduced to 15%, significantly improving SOC efficiency and response times. Additionally, tailored rules for detecting lateral movement improved detection rates by 30%, providing earlier warning signs of potential breaches.*

By conducting this detailed analysis, organizations can optimize QRadar's reporting capabilities, align security operations with strategic goals, and maintain a proactive approach to evolving cyber threats.

3.13. Conclusion

3.13.1 Summary of Findings

This section summarizes the key findings from the QRadar implementation and evaluation project, detailing its impact and capabilities:

- ***QRadar's Effectiveness in Detecting and Responding to Threats:***
    - *Robust Threat Detection: QRadar excelled at identifying a diverse range of threats, such as phishing campaigns, insider activities, and malware infections. Its ability to correlate logs from multiple sources provided enhanced situational awareness and actionable intelligence.*
    - *High Precision and Low False Positives: Through the application of custom rules and optimized correlation logic, the system achieved a precision rate of over 95%, significantly reducing noise and false alarms.*
    - *Real-world Example: During a simulated ransomware attack, QRadar detected unusual data encryption activities within minutes and triggered*

an automated containment response, highlighting its efficiency in critical scenarios.

- *Efficient Response Times: Response times for critical alerts averaged 10 minutes, with containment actions initiated within 5 minutes in 90% of cases. This reduced the risk of escalation and minimized potential damages.*

- ***Key Project Outcomes:***
  - *Log Source Integration: Successfully onboarded and normalized data from 25 log sources, including firewalls, endpoint detection tools, and cloud services. This ensured comprehensive visibility across the organization's IT infrastructure.*
  - *Enhanced SOC Efficiency: Tailored dashboards and streamlined reporting workflows reduced analyst alert fatigue by 40%, enabling faster prioritization of critical incidents.*
  - *Improved Detection Rates: Optimized lateral movement detection rules led to a 30% increase in identifying malicious internal activities. For instance, QRadar flagged suspicious credential reuse across servers, allowing proactive intervention before data exfiltration occurred.*

These findings underscore QRadar's role as a critical component in strengthening the organization's cybersecurity framework, delivering both tactical and strategic benefits.

3.13.2 Recommendations

Based on the findings, the following detailed recommendations are proposed to further enhance the organization's security posture:

- ***Optimize Rule Configurations:***
  - *Regular Reviews and Updates: Schedule biweekly reviews of all correlation rules to ensure alignment with the latest threat intelligence and organizational priorities. Tailor rules to address advanced persistent threats (APTs) and zero-day attacks by implementing adaptive thresholds and time-based conditions.*
  - *Dynamic Rules: Introduce dynamic thresholds that adjust based on user behavior and historical baselines. For instance, a rule could flag login attempts exceeding 3 standard deviations from normal patterns during off-peak hours.*
  - *Real-world Example: A financial institution reduced false positives by 30% by introducing time-sensitive thresholds to detect brute-force attacks*

during non-business hours, which improved overall SOC efficiency and freed resources for high-priority threats.

- ***Expand Log Source Integration:***
    - *Broaden Visibility: Integrate logs from additional critical sources, including IoT devices, industrial control systems (ICS), cloud-native applications, and OT (Operational Technology) environments. This ensures holistic threat coverage across both IT and OT domains.*
    - *Custom Parsers for Unique Sources: Develop custom DSMs (Device Support Modules) for proprietary or industry-specific systems to improve log parsing accuracy. This can minimize missed detections due to unstructured log formats.*
    - *Real-world Example: A manufacturing company integrated IoT device logs from factory sensors into QRadar, enabling the detection of unauthorized firmware updates that could have disrupted production lines. By proactively addressing anomalies, they avoided downtime and potential financial losses.*
- ***Implement Advanced Automation:***
    - *SOAR Integration: Integrate QRadar with a SOAR platform to automate repetitive but critical tasks such as ticket creation, incident assignment, and endpoint isolation. Define response playbooks for common scenarios like ransomware infections or phishing attacks.*
    - *Incident Correlation and Escalation: Automate the correlation of low-severity alerts into meaningful incidents using machine learning, enabling analysts to focus on higher-value activities.*
    - *Real-world Example: A healthcare organization utilized SOAR integration to automatically block IPs involved in credential-stuffing attacks, reducing manual intervention and achieving containment in under 3 minutes compared to the previous 15-minute average response time.*
- ***Enhance Threat Intelligence Integration:***
    - *Diverse Feeds: Expand QRadar's integration with multiple external threat intelligence feeds, such as those tracking nation-state attacks, dark web activity, or regional cybercrime trends. Enhance alert enrichment with these feeds to provide contextualized, actionable data.*
    - *Automated Feed Updates: Use APIs to fetch and update indicators of compromise (IoCs) dynamically, ensuring that detection rules remain relevant against emerging threats.*

- *Real-world Example: An e-commerce company integrated threat intelligence feeds monitoring dark web forums. They proactively detected and blocked credential-stuffing attempts targeting customer accounts, reducing fraud-related costs by 20%.*

3.13.3 Future Work

To build on the current project outcomes, the following areas are suggested for future research and development:

- *Exploring Additional Use Cases: Investigate new and evolving use cases tailored to the organization's dynamic security needs. For example:*
  - *Supply Chain Attack Detection: Develop rules and monitoring processes to identify anomalous activities in third-party integrations or vendor applications. Real-world example: Monitoring for unexpected data access patterns in a supply chain management tool helped identify unauthorized vendor access, preventing a data breach.*
  - *Cloud-Native Application Monitoring: Expand QRadar's scope to cover serverless architectures and microservices, ensuring threat detection for cloud workloads like AWS Lambda or Azure Functions. Example: Setting up correlation rules for API access logs uncovered repeated unauthorized calls to sensitive cloud APIs.*
- ***Advancements in SIEM Technology:***
  - *Stay updated with cutting-edge SIEM developments, including the integration of artificial intelligence (AI) and machine learning (ML) to enhance anomaly detection and predictive threat analysis.*
  - *Example: Implementing ML-based anomaly detection algorithms reduced false positives by 50% in a financial services organization while detecting stealthy insider threats.*
  - *Research emerging SIEM capabilities like natural language querying to improve SOC analyst efficiency and reduce time-to-insight.*
- ***Integration with Emerging Technologies:***
  - *Extended Detection and Response (XDR): Integrate QRadar with XDR platforms to unify telemetry across endpoints, networks, and cloud environments. This enhances visibility and streamlines incident response workflows. Real-world example: A healthcare organization used XDR integration to correlate endpoint malware logs with network anomalies, identifying advanced lateral movement.*

- *Zero Trust Architectures: Leverage QRadar's insights to implement Zero Trust principles, such as continuous monitoring of user behaviors and adaptive access controls. Example: Combining QRadar's UEBA (User and Entity Behavior Analytics) with Zero Trust policies allowed a manufacturing firm to mitigate insider threats by dynamically adjusting user permissions based on detected anomalies.*

- ***Enhanced Threat Simulation and Testing:***
  - *Conduct periodic red-teaming exercises to simulate sophisticated attack scenarios and validate QRadar's ability to detect and respond effectively. Example: A retail company tested its QRadar setup by simulating a ransomware attack across endpoints, uncovering gaps in lateral movement detection that were later addressed.*
  - *Develop comprehensive testing frameworks using tools like MITRE ATT&CK evaluations to ensure that detection rules align with evolving threat techniques.*

By incorporating these suggestions into the strategic roadmap, the organization can enhance its security posture, stay ahead of emerging cyber threats, and maximize the potential of its QRadar implementation.

3.14 Conclusion

Implementing an AI-driven Intrusion Detection System (IDS) like IBM QRadar is a pivotal step toward creating a robust cybersecurity framework in today's ever-evolving digital landscape. Cyberattacks have become increasingly sophisticated, targeting organizations of all sizes and across industries, from phishing schemes that compromise sensitive customer data to ransomware attacks that can paralyze entire networks. With QRadar, businesses gain a critical ally in identifying and mitigating these threats before they can cause significant harm.

Enhanced Threat Detection and Response

QRadar leverages the power of artificial intelligence and machine learning to go beyond traditional rule-based intrusion detection systems. For instance, consider a global retail organization that experiences millions of transactions daily. Manually monitoring network behavior for anomalies is nearly impossible, yet such activity might indicate a credit card skimming operation or malware infiltration. With QRadar, AI models analyze these transactions in real time, identifying suspicious patterns, such as unusually high volumes of data transfer or access attempts from previously unknown locations.

Through real-time threat intelligence integration, QRadar continually updates its understanding of emerging threats. This means it can recognize and respond to new ransomware strains like LockBit or BlackCat faster than systems relying solely on human input. Such rapid adaptability helps organizations stay ahead of cybercriminals who constantly evolve their tactics.

Practical Benefits of Prioritized Alerts

A major challenge for many security teams is the flood of alerts generated by legacy systems—many of which are false positives. QRadar addresses this by using user behavior analytics (UBA) and network behavior analytics (NBA) to identify which alerts truly require immediate attention. Imagine a healthcare provider managing sensitive patient data under strict compliance standards like HIPAA. A QRadar-powered IDS might flag a sudden access request from a hospital workstation at 3 a.m., correlating it with an employee's credentials that were compromised in a phishing scam earlier that day. By prioritizing this alert, QRadar ensures the security team can act quickly to revoke access and prevent a potential breach.

Accelerating Incident Response with Automation

In real-world scenarios, time is of the essence when responding to cyberattacks. Take, for example, a financial institution targeted by a Distributed Denial of Service (DDoS) attack designed to disrupt online banking services. QRadar's AI-driven automation can immediately identify the attack and initiate response protocols, such as rerouting traffic or temporarily blocking malicious IP addresses. Without such automation, response times might stretch, leading to customer dissatisfaction, financial losses, or even reputational damage.

By reducing the reliance on manual processes, QRadar also minimizes human error—whether it's overlooking a critical log entry or misconfiguring a firewall rule. This allows security professionals to focus on strategic tasks, such as strengthening overall defenses or conducting detailed threat analysis.

Staying Ahead of Evolving Threats

QRadar's strength lies in its ability to learn and adapt over time. For instance, after analyzing the tactics of a cybercrime group that recently targeted energy infrastructure, QRadar can fine-tune its detection capabilities to spot similar methods elsewhere. This proactive approach is vital in industries like critical infrastructure, where a single cyberattack can lead to widespread power outages or supply chain disruptions.

Consider the growing trend of supply chain attacks, where adversaries infiltrate trusted vendors to compromise their clients. QRadar can detect unusual activity at the vendor integration point—such as unexpected data sharing or unauthorized access attempts—allowing organizations to act before the attack escalates.

A Secure and Resilient Future

As cyber threats become more complex, the adoption of advanced AI-based security solutions like IBM QRadar isn't just a luxury—it's a necessity. By enabling faster, more accurate threat detection, prioritizing responses, and automating repetitive tasks, QRadar empowers organizations to protect their assets and ensure continuity in an increasingly interconnected world.

In today's digital age, a single breach can have devastating consequences—not just in financial terms but also in trust and reputation. IBM QRadar, with its dynamic and adaptive capabilities, equips businesses to stay one step ahead of cybercriminals, protecting what matters most. By investing in systems like these, organizations are not only defending against current threats but also building the foundation for a secure, resilient future.