**Penetration Testing**

**Santhosh Reddy Miriyala *Prof. Dadmehr Rahbari***

***MSc*. *Information Assurance Adujunct Lecturer***

***and Cybersecurity Compute and Information Science***

***College of Engineering and Business Gannon University***

***Gannon University***

**Abstract**

Virtual Machines (VMs) are operating systems that may be installed on computer hardware to enable the simultaneous operation of various operating systems and software applications. Untrusted data delivered to interpreters, code injection vulnerabilities, and sensitive data exposures are only a few of the areas where this research focuses on VM-related security weaknesses and concerns. The significance of security setups, logging, and monitoring in preventing attacks is also explored in depth throughout the project. The importance of finding security flaws and fixing them is emphasized, as is the use of various penetration testing (Antunes & Vieira, 2012) techniques. The literature study looks at the steps and different kinds of penetration testing, paying special attention to finding vulnerabilities, managing configurations, testing permissions, checking input data, and verifying web services. Software upgrades, secure coding techniques, and the significance of incident response structures are all recommended as ways to address vulnerabilities. In order to reduce potential dangers and safeguard sensitive information, the project highlights the value of proactive security measures and constant monitoring.

# 1.0Introduction

## 1.1 Objective

The objective is that both penetration testing (Kermani & Esford, 2022) and incident response for web applications work towards the same goal: strengthening the safety of websites and other web-based systems from malicious intrusion. The goal of penetration testing is to fortify an application's defenses against cyber-attacks by uncovering flaws in advance, gauging the severity of those flaws, and offering advice on how to fix them. However, the goal of incident response is to quickly and effectively identify, isolate, investigate, and recover from security incidents in order to limit the damage caused by breaches, maintain the reliability and availability of web applications, and enhance security in the future. When put together, these goals form an all-encompassing, preventative strategy for protecting the assets and data of web applications from the ever-evolving cyber security risks they face.

## 1.2 Scope

Penetration testing for online applications examines APIs and client-side components, among other things, to systematically detect and analyze vulnerabilities in areas like authentication, authorization, input validation, session management, and data security. Its purpose is to identify potential security flaws and offer solutions for fixing them. In contrast, the goal of incident response for web applications is to quickly and effectively identify and address security incidents. This includes all steps necessary to do so, such as identification, categorization, isolation, investigation, notification, elimination, recovery, documentation, and post-incident analysis. Integrity and availability of online services rely on both penetration testing (Halfond et al., 2009) and incident response, with the former proactively finding flaws and the latter reacting to and limiting the effects of security breaches.

## 1.3 Testing Objectives

* Identify vulnerabilities in network infrastructure.
* Evaluate the effectiveness of access controls.
* Assess the risk of unauthorized access.

Penetrating testing is a cyber attacker's way of seeing how vulnerable your system is. It's a form of testing that can help pinpoint vulnerabilities before an attacker does any damage. It can be managed with any program. Depending on the situation, this can also be adjusted manually [3]. The procedure entails researching the target beforehand, finding all available entry points, breaking in (either physically or virtually), and then reporting the results. The primary objective of this type of testing is to identify potential security flaws in a website or security poster, as well as to gauge the level of support for the company's security policy, evaluate the preparedness of employees to deal with any potential threats, and predict how vulnerable the website is to potential disasters. Although the overarching goal is to safeguard against and identify any attempted attacks on the program or app. This testing provides valuable input to a company in the form of a report. This information can also be used to help prioritize a company's planned security investments. These reports also have the added benefit of guiding programmers toward more bulletproof code. If designers learn how their security system was hacked or misused, they can direct their specialists to make those points much harder for hackers to exploit in the future. It is quite unusual for a whole system to have all of its support properly organized. When an entry analyst has a good understanding of the system and its weaknesses, he or she can exploit a security hole and gain unauthorized access with the help of a pen test. However, specialists in the field of security don't just focus on frameworks. Clients on a system are a common target for a pen tester, who may contact them via phishing emails, pre-content phone calls, or even physical social engineering. Security professionals can find vulnerabilities in complex system designs, unique apps, and web-based administrations with the help of this testing. These penetration testing tools and services let you gain immediate insight into high-risk areas, which is crucial for developing effective security budgets and strategies.

The accomplished work introduced a vulnerability associated with encryption. This can be summarized briefly as follows:

Encoding data at rest or in transit is one method that businesses frequently use to ensure the security of their communications. Standard protocols such as Secure Shell (SSH), Secure Sockets Layer (SSL), and Transport Layer Security (TLS) are used to encrypt data so that it cannot be read without a special key. However, there have been times when businesses have used less secure encryption systems, and in those instances, hackers have been able to crack the code. Most Wi-Fi connections are protected by the WPA2 protocol, which was shown to be extremely vulnerable in October 2016. Sometimes developers will try to prevent messages from being sent in order to trick verification systems that are supposed to check the digital identity of senders. As a result, they will be able to fend off so-called "man in the middle" attacks. You may learn how safe your data exchanges and data storage procedures are by conducting infiltration testing.

**2.0 Literature Review**

## 2.1 Planning Stage

This is the stage where the assignment's parameters are defined. Strategy risks, an initial timetable, and other preparatory measures for security are all part of this stage. These are required for the release and require the application of unique methods and procedures. VM has made a long-term commitment to ensuring the following security measures are taken during development: Safe coding practices, numerous security tools, and an understanding of security concepts are all part of this. Additionally, a threat model would be created to expose the application's security holes and poor design. Modeling potential risks early on in the development process is essential. This provides enough time to rethink the application's architecture and address any security concerns [6].

In general, there are a few reasons to do penetration tests (or "pen tests"). Above all else, entrance testing can aid in ensuring the safety of sensitive client data, identifying security flaws, locating framework escape clauses, and gauging the overall quality of current barrier devices. In addition, regular entry testing can help a company be alert for any new piece of software that is released. Due to the ever-changing nature of security threats, personal and financial data must be checked and double-checked at regular intervals before being transferred between different endpoints. Similarly, there are a few significant benefits to using an entrance exam [3]. It's a great way for MSPs to demonstrate their expertise and keep tabs on security issues before they become a problem. It saves businesses money by eliminating the need to schedule individual meetings. MSP's customers can comply with regulations and avoid penalties with the use of entrance testing procedures. In the end, it's a crucial tool for protecting the standing, reputation, and customer loyalty of managed service providers. Although pen testing and vulnerability assessments both pertain to cyber security, they are not interchangeable terms. The goal of a security assessment is to identify vulnerabilities within a company. After analyzing the weak spots in cyber security and data storage, a list of problems can be compiled. In contrast, an infiltration test approaches cyber security through a systematic, objective, and attack-like environment. The goal of the evaluation is to achieve a specific outcome, such as the success or failure of a database, stockpiling plan, or designated record. A pen test's output is more than simply a list; it's also a technique and guidance for pinpointing where weaknesses lie [3].

# 3.0 Methodology

The penetration testing process involves several phases:

## 3.1 Reconnaissance

The Reconnaissance is the first step since it specifies that the primary goal of an admission test for a web application is to learn as much as possible about the target application. One of the first and most fundamental steps of an application pen test is observation, often known as information gathering. Open-source tools (web indexes), scanners, plain HTTP requests, and custom-made requests are all used for this purpose. Thus, it is possible to coerce the program to leak information, such as by revealing error messages or revealing the variants and technologies used. Fuzzing, error analysis in code and fingerprinting applications are all part of it.

## 3.2 Configuration Management

Understanding the server and infrastructure configuration is vital, as weaknesses in the system can compromise the application. Mistakes like using insecure HTTP methods or keeping old and backup files can jeopardize security.

## 3.3 Authentication Testing

It's the steps taken in an effort to confirm the sender's high moral standing before responding to their message. The logon process is probably the best-known example of such a method. Verification testing involves observing the verification process and using that knowledge to circumvent the validation infrastructure. Brute-force attacks and user enumeration are two of the primary types of tests conducted here.

## 3.4 Session Management

It's defined as the rules that govern the web application's tasteful interactions with the user; therefore, it affects everything from the login procedure to the user's final exit from the app.

## 3.5 Authorization Testing

Authorization testing examines the system's authorization and authentication procedures to look for holes that could be exploited to gain unauthorized access or boost a user's privileges. It entails learning the inner workings of the approval procedure and then applying that knowledge to the approval instrument. Given that approval is contingent upon having been effectively verified, the pen examiner will double-check this detail only after having collected considerable certificates in relation to a comprehensively outlined set of responsibilities and advantages. Therefore, it is important to look into the possibility of going against the approved plan, discovering a way to get around the lack of protection, or creating means to boost the benefits.

## 3.6 Data Input Validation

Most security vulnerabilities result from improper or incomplete input validation. To protect against threats like cross-site scripting and SQL injection, testing for this is essential. The inability to reliably authenticate user or server-generated content is often seen as the biggest security flaw in web applications. Cross-page scripting, SQL injection (Clarke, 2009), interpreter mixing, geographical/Unicode assaults, archival structure ambushes, and so on are all caused by this errors, and they rely on it for their success. The next step we'll go over is entirely up to the discretion of the user doing the test.

## 3.7 Denial of Service

The purpose of a Denial of Service (DoS) attack is to prevent its intended audience from accessing the attacked resource. DoS vulnerabilities at the application level can be triggered by hackers using code defects. An attack that aims to prevent legitimate users from accessing a resource is called denial of service. Traditional methods of resisting administrative attacks have relied on elaborate systems of coordination, such as the flooding of a neutral machine with so much traffic as to render it incapable of handling its usual users. A malicious client may be able to render some functionality unavailable, however, due to weaknesses in the applications themselves. These problems originate from code defects and are triggered by malicious or unexpected user input. In this round of testing, we will focus on attacks against accessibility at the application layer that can be launched by a single malicious client on a single system. Therefore, it may not always be a part of an infiltration test, as not all clients have a desire to be tested.

## 3.8 Web Services

The final phase of web application penetration testing assesses vulnerabilities in web services, similar to other application vulnerabilities.

# 4.0 Implementation stage

Vulnerability in a web application is a security gap in a program. They have persisted for a long period due to factors such as inadequate support for or cleaning of structure inputs, incorrectly configured web servers, and improper application setup, and they can be exploited to address security concerns. These weaknesses are out of scale with more common types, like framework or asset weaknesses. Web apps are on the upswing because of the increased openness required to connect with a wide range of users across a variety of platforms.

Firstly, it involves SQL assault, which is so commonly used to manage and coordinate data on applications at the moment that developers have devised methods to sneak their SQL instructions into the database. These commands can alter, remove, or delete data, and in some cases can even give the programmer access to the system's core. A computer language used for communicating with databases, SQL (authoritatively articulated, yet typically articulated "spin-off") stands for structured query language. The majority of the servers that host essential data for websites and administrations utilize SQL to manage database content. These servers are the primary target of SQL implantation ambush attacks, which use malicious code to steal information from the server. This is especially dangerous if the server also saves sensitive information about the website's or web application's (Wilhelm, 2013) users, such as credit card numbers, login credentials, or other information that might be used to commit fraud or identity theft [2]. Unprotected software is more likely to fall victim to a compelling SQL implantation ambush because it fails to adequately clean information sources given by the customer, namely by stripping away anything that looks to be SQL code. If a website is vulnerable to a mix ambush, for instance, an attacker might potentially enter code into the site's request box instructing the SQL server to dump all of the site's stored user names and passwords. Cross-site scripting is another form of cyberattack in which an attacker targets a specific website in order to steal its private information, such as client credentials or financial details. In any case, a cross-site scripting ambush may be chosen if the attacker prefers to legitimately focus on the site's visitors. Similar to a SQL injection attack, this method involves secretly inserting malicious code into a website or software program. However, in this situation, the customer's application may run the malicious code the attacker has implanted when the customer visits the booby-trapped website, and it may legitimately pursue the visitor. It's commonly believed that an attacker can plant a cross-site scripting trap by inserting malicious code into a data field, which would then be executed whenever a user visited the compromised page. A comment left on a blog, for instance, could contain a link to malicious JavaScript [2].

A website's credibility might take some damage from cross-site scripting attacks since they put customers' personal information at risk without leaving any obvious trace of the attack. Cross-site scripting can be used to steal credentials, credit card details, and other personal information sent to a website or application, with or without the knowledge of the site's or application's owners. Penetration testing for a website uses the following methods: Reconnaissance, configuration management, permission testing, data input validation, and online services are just few of the stages included.

## 4.1 Recommendations

The following are some suggestions for improving security and addressing weaknesses in organizations:

1. Keep Software Up-to-Date: Maintaining up-to-date server and client software is essential for preventing security issues.
2. Secure Coding Practices: Encouraging developers to use secure coding methods throughout application development can greatly reduce the likelihood of security flaws.
3. Error Handling: Make sure attackers can't learn too much about the system's architecture from error messages.
4. Incident Response Framework: A vulnerability discovery, prioritization, and mitigation strategy should be set up as part of the

# 4.2 Exploitation

## 4.3 Practical Implementation

In order to map out the potential attack surface, we looked at the megacorpone.com domain's name servers (Figure 1).

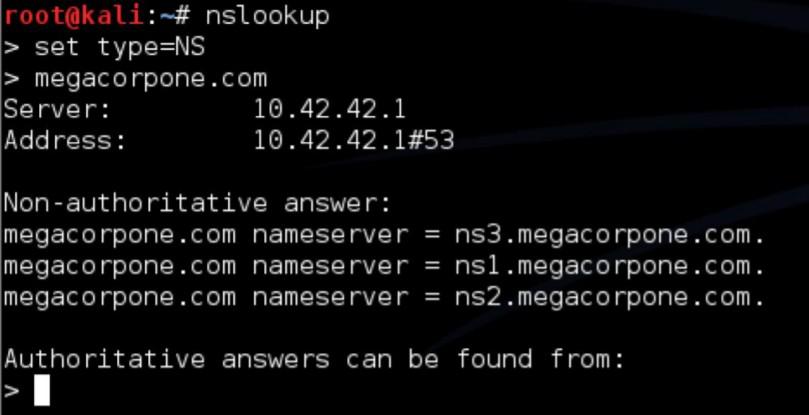
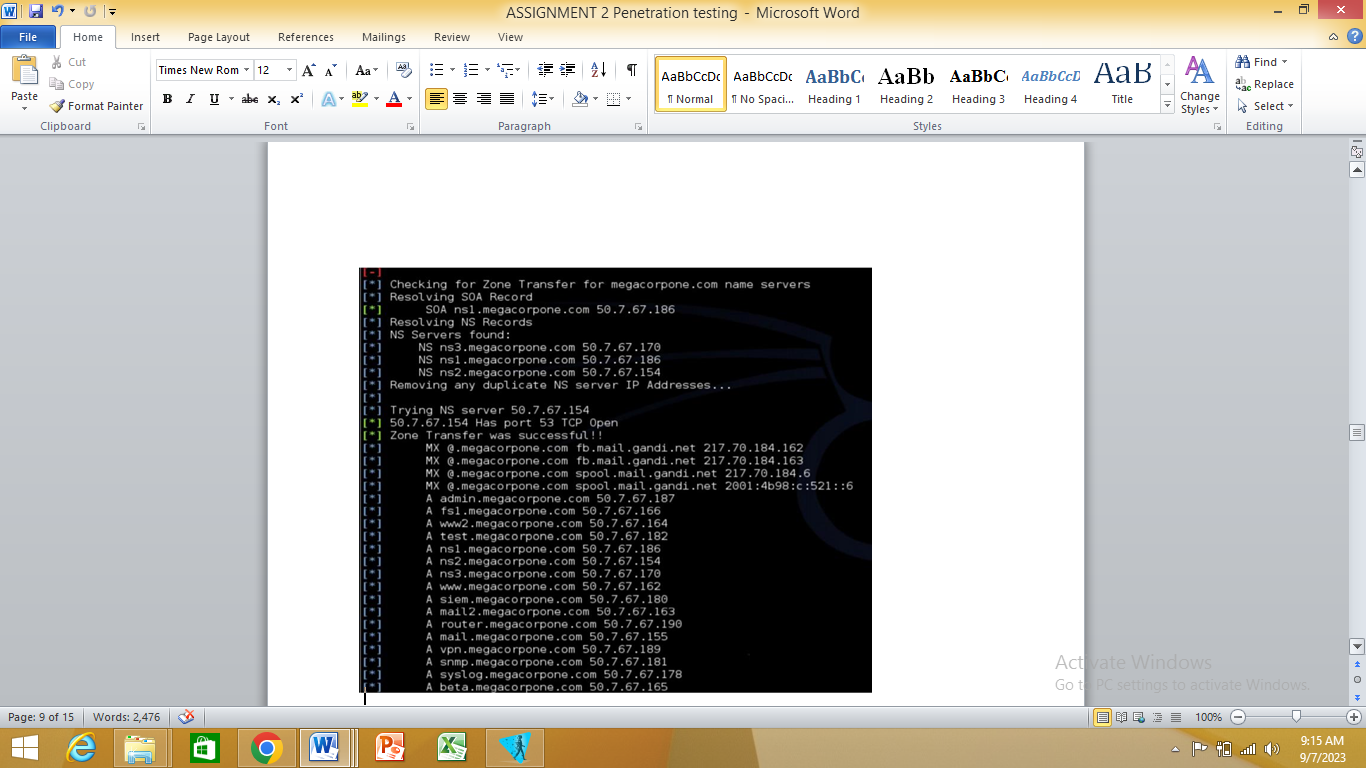


Figure 1: Collecting Data



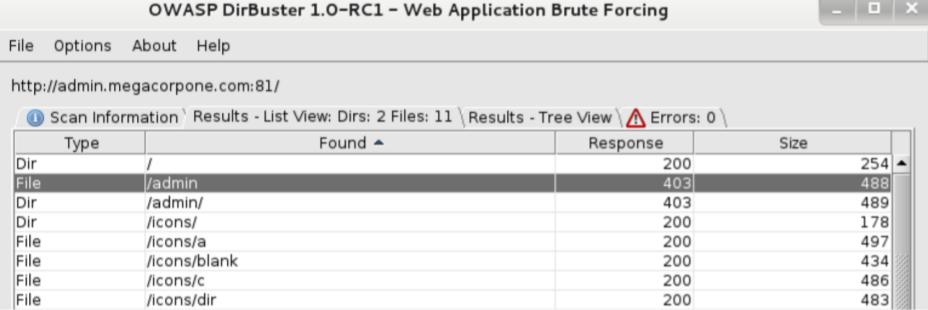


Figure 2: DNS Zone



Figure 3: A compromise in interface

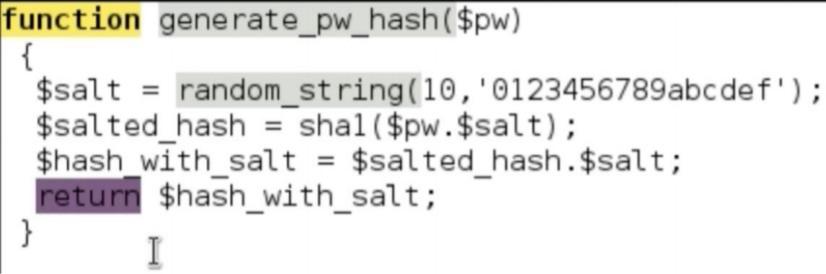


Figure 4: SQ Lite Manager



Figure 5: Source Code

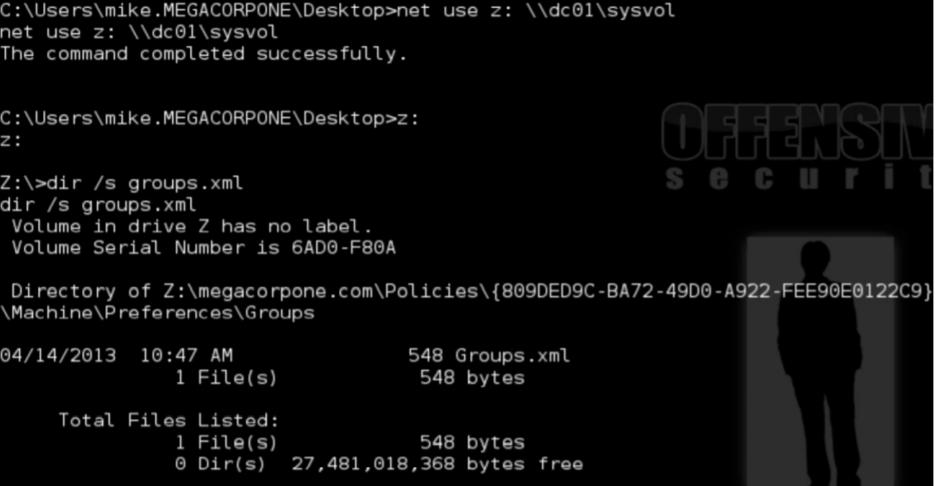


Figure 6: Screen output of SQL lite availability

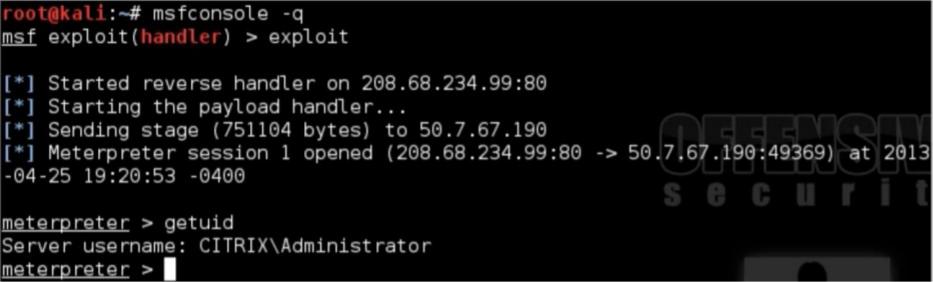


Figure 7: Screen output of showing Gained access of newly retriev

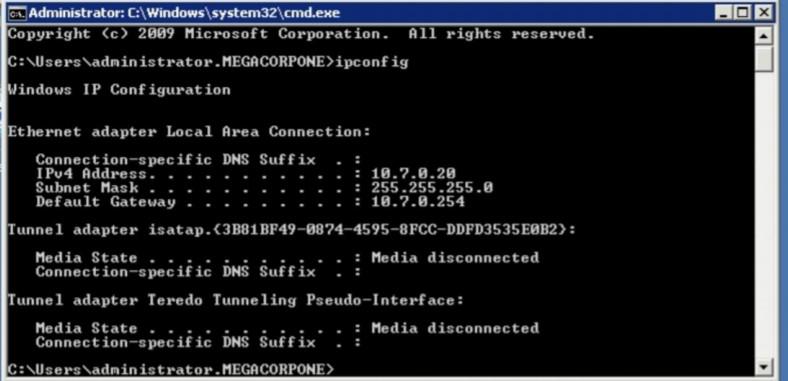


Figure 8: Screen output showing Critix Server that has been compromised

## 4.4 Potential impact assessment

Penetration testing plays a crucial role in a comprehensive cyber security strategy, and the proficient implementation of these tests can provide significant insights and advantages for an organization. The subsequent points underscore the importance of penetration testing system (Kang et al., 2022) and illustrate its efficacy based on the presented information.

1. Identification of Vulnerabilities: The main purpose of a penetration test is to find security threats on the target website. The fact that errors were uncovered during testing proves the experiment was successful. Data security, session management, and the authentication and authorization processes are all at risk. By isolating these problems, the test successfully revealed vulnerabilities that bad actors could have exploited [4].
2. Risk Assessment: The process of identifying vulnerabilities is crucial to conducting a thorough risk assessment. Gaining insight into the potential ramifications of these vulnerabilities is a fundamental objective of the examination. This test's efficacy in illuminating the vulnerability of the website to potential attacks is indicative of its effectiveness. Moreover, it furnishes a foundation upon which educated decisions may be made pertaining to risk mitigation strategies [6].
3. Data Security and Privacy: Penetration testing has proven useful by revealing previously unknown vulnerabilities in data protection. It stresses the significance of protecting personal information. The fact that these problems were identified indicates that the test was successful in preventing breaches of confidentiality involving sensitive data.
4. Operational Impact: The testing was so comprehensive because it took into account potential effects on operations. This includes situations where the website is temporarily unavailable or performs poorly. By considering these factors, the test demonstrates a realistic approach, increasing confidence that the organization is ready to efficiently manage interruptions [8].
5. User Experience and Trust: The fact that penetration testing can have a negative effect on user experience and trust only serves to increase the practice's popularity. It's crucial to properly execute the test with as few interruptions as possible, as this example shows. It also emphasizes the importance of users' confidence and trust as factors in the testing procedure.
6. Compliance and Legal Obligations: The possibility that compliance problems and legal repercussions will emerge during the test is evidence of penetration testing's thoroughness. The test is a tool for discovering and fixing such problems beforehand, ensuring that the business follows all applicable rules and is protected from legal action.
7. Communication and Reporting: The importance of getting the test results to the right people cannot be overstated. Sharing results, hazards, and recommended corrective actions clearly ensures that all parties involved are on the same page. This openness is critical for making well-informed choices and taking the right steps [4].
8. Remediation Efforts: The acknowledgement that organizations must devote resources and promptly address vulnerabilities highlights the practical implications of the test. The achievement of a favorable result is contingent upon the timely resolution of security vulnerabilities, thereby bolstering the overall security stance of the website

## 4.5 Incident Response Plan

Six well-defined protocols help IT professionals and staff recognizes and responds to a cyber-incident like a data breach or digital attack. Maintaining an effective incident response strategy calls for regular reviews and rehearsals. Due to the predetermined nature of the process, special preparation is needed. Establish a procedure to refresh and manage the occurrence reaction plan in light of industry and hierarchical changes; designate workers to be available 24/7 to manage frequencies; train employees on their episode reaction responsibilities; establish alarms from interruption discovery and interruption avoidance; and record trustworthiness checking frameworks; and test the occurrence reaction plan at least once a year.

Firstly, a CSIRT's greatest strength lies in the hands of its decision-makers. Therefore, your team must consist of high-ranking executives or even company representatives. It is common for the chief information security officer, chief information officer (CIO), or chief innovation officer to steer an incident response team. However, the group may also comprise the chief executive officer and the chief operating officer (CEO), depending on the association's structure. Decisions must be taken quickly and cannot be tested during an emergency response because time is of the essence.

Second, the data collection methods implemented here makes use of a wide variety of IT security instruments that could prove valuable in spotting an anomaly. Some of these include an intrusion detection system (IDS) for detecting actual or potential intrusions, a vulnerability scanner for identifying actual or potential vulnerabilities (though you should use one routinely by and by), and manual devices for entrance testing to confirm helplessness, along with other risk areas such as the internet, network security, organizational security, and a security data and event board.

Thirdly, once specialized assets have determined the nature and scope of the incident, the next step is containment, during which you'll decide which assets, if any, need to be quarantined. Despite the temptation to eliminate the threat as soon as possible, establishing control is essential and cannot be skipped. If the attacker isn't stopped, they may still be communicating with your team and can keep infecting new systems. In order to exert control, it is necessary to isolate the affected resources from those that are not. However, this is rarely done (even accidentally) because it can make destruction more difficult if the components are no longer attached. The decision made at the end of the stage of control ensured the segregation of resources and the elimination of the threat.

Another stage is removal, during which your specialist assets begin eliminating the effects of the occurrence once the affected resources have been confined. This includes tasks such as removing malware, patching security holes, resetting frameworks with reliable backups, and so on. At the conclusion of the disposal phase, a decision is made to eliminate all specialized consequences of the event and guarantee the framework.

The recovery phase has arrived, at which point all the guaranteed infrastructure must be brought back online, reconnected to external resources, and normalized technical and business operations resume. The final step in the recovery phase is a determination that the entire technical infrastructure is operating as smoothly as it did before the incident. Keep in mind that the recuperation phase also involves the completion of tasks by your communications and legitimate or consistent resources. The outcome of this phase is for higher-ups to declare the incident over and for employees to go back to business as usual. This action is unnecessary once the event has concluded, which is the last phase and a lesson learned. The incident response team's secret weapons, including the chiefs, should be reassembled at some point following the event so that lessons learned can be analyzed. If certain team members didn't do as well as expected, the method may be restarted from the beginning to incorporate new members, reallocate responsibilities, or provide extra training.

## 4.6 Vulnerability Details & Mitigation

The prioritization of exploited vulnerabilities, as per the guidelines outlined in NIST SP 800-30, involves assessing their probability and potential consequences in order to ascertain the overall level of risk.

## 4.7 Weak Credentials

Rating: The externally accessible administration interface is inadequately secured with a weakened password.

Impact: The administrative password for the SQLite Manager web interface can be retrieved using standard enumeration and brute-force techniques. All user password hashes recorded in the underlying database can be obtained because no additional authentication mechanisms are in place. If the plaintext passwords are retrieved and password reuse is discovered, the targeted system may be further exploited.

**Remediation:** It is imperative to implement robust security measures by employing intricate passwords or passphrases for safeguarding all administrative interfaces. Please refrain from utilizing frequently used or commercially oriented vocabulary, which may be readily located or readily formulated with the assistance of a lexicon.

Rating: The misconfiguration of a DNS server results in unlimited zone transfers.

**Impact:** If a DNS server is set up to allow zone transfers to any other DNS server, then the server has the ability to leak critical information regarding to the assets of the firm as well as the configurations of the network.

**Remediation:** The restriction of DNS zone transfers should be limited exclusively to servers that have been pre-approved.

# 5.0 Discussion

When companies undergo penetration testing, they gain a wealth of knowledge in a short amount of time. Numerous insights and lessons occur as security professionals meticulously study and examine systems, networks, and applications. The tests reveal, first and foremost, where the organization’s infrastructure has security holes. These security holes can be caused by anything from faulty code to incorrectly set permissions, providing a complete list of problem spots.

In addition, by mimicking the methods used by cybercriminals, penetration testing reveals valuable information about how they operate. When security teams are aware of common attack vectors, they are better able to prepare for and repel actual attacks.

The company's security measures are also in-depth examined. It assesses the state of current security measures and detects flaws that need fixing right away. A company's ability to detect, respond to, and mitigate security problems can be gauged by evaluating the effectiveness of its incident response protocols.

In the context of social engineering, penetration testing (Wilhelm, 2013) can serve as a valuable lesson in user consciousness. It highlights the need for strong training and awareness programmers to enhance the human element of security by making staff more resistant to approaches like phishing.

Lessons gained help priorities remedial efforts in addition to identifying vulnerabilities. Organizations can better deploy resources if they have a clear awareness of which vulnerabilities offer the greatest concern.

The testing procedure also serves to demonstrate the company's security culture. It reveals whether a security-first mentality is already deeply embedded in the company's DNA or whether changes are required to the company's culture.

Penetration testing identifies vulnerabilities that could result in noncompliance for businesses that are required to maintain regulatory compliance. Eliminating these discrepancies will keep your company in compliance with regulations and lessen the likelihood of fines [3].

The necessity of always striving to do better is probably the most important takeaway. Penetration testing is not a one-and-done activity but rather an iterative procedure. Each test results in a more refined threat model and security strategy that helps organizations remain ahead of the curve. In this approach, important assets and data are protected since penetration testing not only identifies vulnerabilities but also encourages a culture of resilience and continuous improvement in cyber security practices.

# 6.0 Conclusion

Data on newly discovered vulnerabilities that are identified after the test will not be provided by the entrance exam. The infiltration analyzer won't have enough framework information. In comparison with defenselessness assessments, which can identify more problems than infiltration testing using symptomatic surveys with all servers being equal, the entry test isn't the best way to identify the flaws. Infiltration testing is a representation of an association and its system security, whereas entrance testing has limited time to analyze and identify weaknesses. Expanding the scope of infiltration testing is necessary. The entrance exam window is extremely constrained. Expanding the entry testing time breaking point will enable testing groups to identify additional vulnerabilities and guarantee the association's system security. It is necessary to take additional action against vulnerabilities that were identified through the infiltration test.

**References**

[1]. Antunes, N., & Vieira, M. (2012). Evaluating and improving penetration testing in web services. 2012 IEEE 23rd International Symposium on Software Reliability Engineering. <https://doi.org/10.1109/issre.2012.26>

[2]. Clarke, J. (2009). Testing for SQL injection. SQL Injection Attacks and Defense, 29-93. <https://doi.org/10.1016/b978-1-59749-424-3.00002-5>

[3]. Halfond, W. G., Choudhary, S. R., & Orso, A. (2009). Penetration testing with improved input vector identification. 2009 International Conference on Software Testing Verification and Validation. <https://doi.org/10.1109/icst.2009.26>

[4]. Kang, H., Kwon, O., Shin, C., Seo, J., Jang, I., & Dong-Woo, M. (2022). Development of free fall cone penetration testing system. Cone Penetration Testing 2022, 164-169. <https://doi.org/10.1201/9781003308829-17>

[5]. Kermani, M., & Esford, F. (2022). Consolidation settlement prediction using cone penetration testing. Cone Penetration Testing 2022, 985-990. <https://doi.org/10.1201/9781003329091-148>

[6]. Metasploit toolkit for penetration testing, exploit development, and vulnerability research. (2007). <https://doi.org/10.1016/b978-1-59749-074-0.x5000-4>

[7]. Risk mitigation, vulnerability management and resilience under disasters. (2022). <https://doi.org/10.3390/books978-3-0365-3862-4>

[8]. Wilhelm, T. (2013). Web application attack techniques. Professional Penetration Testing, 339-355. <https://doi.org/10.1016/b978-1-59749-993-4.00013-6>