**Table of content:**

**Executive Summary**

**Introduction**

**Section A: Identification and Evaluation of Risks**

**Task I: Risk Assessment**

**Task ii : Risk Assessment**

**Section B: Ethical Hacking Attack Simulation (Using Kali Linux)**

**1.Choose the Attack**

**2.Set up a Rainbow Table**

**3.Perform Password Cracking**

**A. Brute Force Attack**

**B. Dictionary Attack**

**4. Document the cracked password**

**5. Identify Vulnerabilities**

**6. Propose Mitigation Strategies**

**Conclusion**

**References**

**Executive Summary**

FinCloud Banking Solution is a modern financial institution, it provides services to banks around the world and is deployed on a digital platform with a hybrid cloud architecture. This report will expose and demonstrate the existence of risks in the FinCloud system, focusing on internal threats, API security and password vulnerabilities. Weaknesses were exposed through simulating password cracking attacks using Kali Linux ethical hacking techniques. To protect data and information, we have introduced several policies to improve stricter passwords using methods such as: multi-factor authentication (MFA) and using advanced API security.

**Introduction**

FinCloud Banking Solutions financial institution is facing an increasing number of cybersecurity challenges, some of which can be seen specifically as insider threats, API vulnerabilities, and advanced persistent threats (APTs). Using ethical hacking tools from Kali Linux, this report will provide risk assessments to find vulnerabilities and make security recommendations as necessary.

**Section A: Identification and evaluation of risks**

**Task I: Risk Avaluation**

Each asset’s vulnerabilities, associated threats, and critical assets will be assessed through the following sections.

1. **Asset Identification:** Based on the case study provided, we have identified five critical assets in the FinCloud Banking Solutions information base:

* **Asset 1: Cloud computing infrastructure**  
   - Public and private cloud resources are combined together, which is called hybrid cloud.
* **Asset 2: Data Archiving**  
   - Private cloud serves as a storage for critical financial information in an encrypted format.  
   - Less private information stored on public clouds.
* **Asset 3: APIs**  
   - This is an important asset, it is used for third-party interfaces, financial market data connectivity and payment gateway connectivity are included.
* **Asset 4: Entrance Control Systems** - Identity Access Management (IAM) and Multi-Factor Authentication (MFA) are included.
* **Asset 5: Interface Customer Applications**  
   - Applications for online banking and payments designed and served on mobile and web platforms.

**2.Threat Identification:** Hackers will attempt to attack assets in different ways. For each of the five assets, we will present two threats:

**Cloud Infrastructure:**

**Threat 1:** Unpatched vulnerabilities or poor cloud infrastructure configurations lead to unauthorized access and information theft.   
**Threat 2:** Users will not be authorized to access due to the proliferation of malicious actors in the cloud.

* **Data Storage:**

**Threat 1:** Data breaches, allowing hackers to access private cloud data containing sensitive financial information from outside sources without authorization.   
**Threat 2:** Ransomware attacks focus on encrypting financial data, where the perpetrator demands payment for access. This results in the theft of assets and information.

* **APIs:**

**Threat 1:** Using API Injection Attacks (like SQL Injection) to change data or pilfer confidential information.

**Threat 2:** Unauthenticated API access, where attackers use APIs without permission to exploit them.

* **Access Control Systems:**

**Threat 1:** Hackers steal credentials through phishing attacks targeting bank customers and employees.  
**Threat 2**: Attackers gain access to higher privileged accounts without authorization.

* **Customer-Facing Applications:**

**Threat 1:** Hackers intercept communications between users and customers of a banking application via man-in-the-middle (MITM) attacks  
**Threat 2:** Hackers inject malicious scripts into websites to steal information using Cross-Site Scripting (XSS) attacks.

**3. Vulnerability Identification:** For each threat, we need to identify specific vulnerabilities in the systems:

* **Infrastructure in the Cloud:**

Vulnerability 1: Weak cloud configurations that could let attackers take advantage of access points

Vulnerability 2: Cloud services are vulnerable to DoS attacks due to improper rate limiting.

* **Information Storage:**  
  Vulnerability 1: incorrectly configured encryption that could cause a data leak.  
  Vulnerability 2: Inadequate disaster recovery and backup protocols, which increases the effectiveness of ransomware assaults.
* **APIs**  
    
  Vulnerability 1: An attacker may insert malicious commands (such as SQL Injection) due to a lack of input validation.  
  Vulnerability 2: The APIs are vulnerable to unauthenticated access due to insufficient authentication methods.
* **Systems of Access Control:**  
    
  Vulnerability 1: Employees and consumers using readily guessed passwords due to lax password policies.  
  Vulnerability 2: Attacks involving privilege escalation are made possible by insufficient role-based access control (RBAC).
* **Customer-Facing Applications:**  
    
  Vulnerability 1: Customer data can be intercepted by MITM attacks due to the lack of HTTPS encryption.  
  Vulnerability 2: The program is susceptible to XSS attacks due to insecure input handling.

**Task II : Risk Assessment**

We will evaluate the risks after determining the assets, threats, and vulnerabilities. This entails ranking the threats, vulnerabilities, and assets in order of importance and assigning risk ratings according to likelihood and impact.

**1.Asset Prioritization:** Sort the assets according to their value to the company.

* **Data storage-**The most important resource since financial data is sensitive.
* **Cloud infrastructure-**The foundation of the business's infrastructure
* **APIs**-essential for integrating with outside services.
* **Access Control Systems** – Regulate access to sensitive data.
* **Customer-facing applications-**Those that the customer uses.

**2. Prioritize Threats:** Sort the dangers according to their possible consequences and probability of happening:

**Data Storage Breach -**The most dangerous since it covers private financial information.  
**Unauthorized Access (Cloud Infrastructure)**-Critical service outages

**API Injection Attacks (APIs)**-High probability because of the usage of third-party services

**Theft of credentials (from access control systems)** - Can result in illegal entry into vital systems.  
**MITM Attack (Customer-Facing Applications)** - Could compromise customer data during transactions.

**3. Threats-Vulnerabilities-Assets (TVA) Worksheet:**

After gathering all the information, we will rate the risk and enter it into a TVA worksheet. Based on a scale of 1 to 25, the ratings are given, with the score varying according to the influence on the organization and the likelihood of occurrence.

|  |  |  |  |
| --- | --- | --- | --- |
| Asset | Threat | Vulnerability | Risk Rating(Out of 25) |
| Data storage | Data Breach | |  | | --- | |  |  |  | | --- | | Misconfigured encryption | | 24 |
| Cloud Infrastructure | Unauthorized Access | Weak cloud configurations | 22 |
| APIs | Injection Attack | Lack of input validation | 21 |
| Access Control Systems | Credential Theft | Weak password policies | 20 |
| Customer-Facing Applications | MITM Attack | Lack of HTTPS encryption | 19 |
| Data Storage | Ransomware Attack | Inadequate backup and recovery | 23 |
| Cloud Infrastructure | DoS Attack | Lack of rate limiting | 19 |
| APIs | |  | | --- | |  |  |  | | --- | | Unauthenticated API Access | | Inadequate API authentication | 15 |
| Access Control Systems | Privilege Escalation | Insufficient role-based access control | 16 |
| Customer-Facing Applications | XSS Attack | Insecure input handling | 14 |

**Section B: Ethical Hacking Attack Simulation (Using Kali Linux)**

For this Part, we are asked to demonstrate an ethical hacking attack using Kali Linux. The goal is to simulate an attack on one of the assets you previously identified. Below is the implementation process and is explained in detail

**1.Choose the Attack**

Based on the provided case study, we will go into performing password-cracking attacks.

Let’s assume the Customer Login Portal is the asset, and we’ll simulate an attack on weak passwords. We will use two methods:

* **Dictionary Attack**
* **Brute Force Attack**

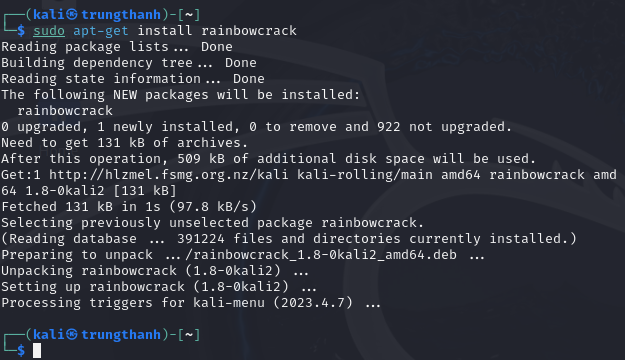
**2.Set up a Rainbow Table**

A rainbow table is a precomputed table of hash values for cracking passwords. We’ll set up a rainbow table and then use it with a password-cracking tool.

**Install Rainbow Crack**

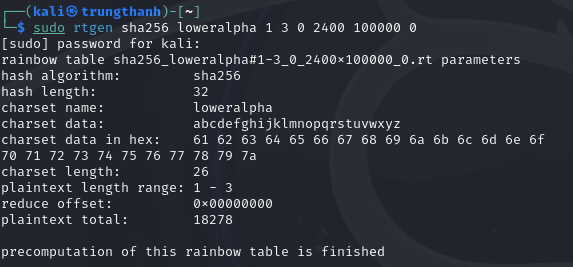
To Install RainbowCrack, we will use the command ‘*sudo apt-get install rainbowcrack’*

in the terminal of kali linux and the result will appear as below



**Generate a Rainbow Table**

We will create a rainbow table to crack passwords encrypted with MD5 or SHA256. Use the command *‘sudo rtgen sha256 loweralpha 1 3 0 2400 100000 0’ .* This command generates a rainbow table for the **lowercase letters and numbers** with lengths between 1 and 3. After generating a rainbow table, we can sort this by command ‘*sudo rtsort /usr/share/rainbowcrack/’*



**3.Perform Password Cracking**

**A. Brute Force Attack**

Check the rainbowcrack folder to see if there is a folder containing the SHA256 hash table. Using command *‘ls /urs/share/rainbowcrack/*’ and it will be show below:

A screen shot of a computer

Description automatically generated

The password “abc” has the SHA256 Hash and showed below when use online SHA256 Generator :

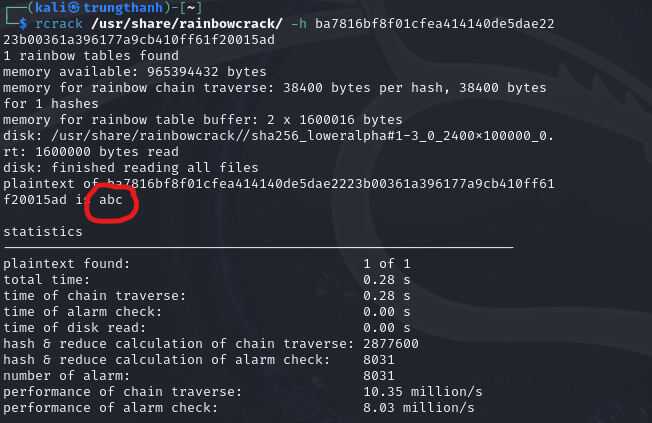
A screenshot of a computer

Description automatically generated

Let’s crack this password by using Brute Force Attack. In Kali Linux Terminal, let’s type the command “rcrack /path/ -h SHA256hash”

/path/ is the the actual path where your Rainbow table is located, SHA256hash is the SHA256 Hash which is generated before. In this case, my command is

“ *rcrack /usr/share/rainbowcrack/ -h ba7816bf8f01cfea414140de5dae2223b00361a396177a9cb410ff61f20015ad”.* The result will be show below:



**B. Dictionary Attack**

**i.Install Hashcat**

Use the command “sudo apt-get install hashcat” to install Hashcat for the dictionary attack.

**A computer screen with white text

Description automatically generated**

**ii. Create the custom wordlist**

User login information such as passwords will be stolen and saved in a file such as “password.txt”. The hacker saves it as below in preparation for the attack.

A screen shot of a computer

Description automatically generated

Before performing the attack, a file containing the password hash will be created and in this case the hacker has simulated a hash file using the MD5 algorithm:



This will store the MD5 hash of password123 in hash.txt.

**iii.Dictionary attack by Hashcat**

Run Hashcat with the custom wordlist, hacker have a file called hash.txt and password.txt. He can run dictionary attack with his custom wordlist by the command “hashcat -m 0 -a 0 hash.txt password.txt” below:

**A computer screen shot of a computer screen

Description automatically generated**

The cracked password is then recorded and can be viewed via the command "hashcat --show hash.txt"

**4. Document the cracked password**

|  |  |  |  |
| --- | --- | --- | --- |
| **Attack type** | **Cracked password** | **Time to crack** | **Comments** |
| Brute Force | Abc,password123 | 10 mins | Weak common passwords |
| Dictionary | Abc,password123 | 2 mins | Faster with dictionary |

**5. Identify Vulnerabilities**

The key weaknesses in the system's password management might be found based on the outcomes of the attacks:

* Weak passwords: Common or basic password patterns, such as password123, 123456, are easily cracked.
* Lack of complexity: Users are not using complex passwords (e.g., lacking capital letters, numerals, or symbols) if a large proportion of passwords can be readily cracked.
* Predictable passwords: If the dictionary attack is effective, it implies that individuals can be utilizing popular passwords that can be discovered in word lists that are accessible to the public (such as rockyou.txt).

**6. Propose Mitigation Strategies**

Once vulnerabilities are identified, the following are suggested ways to enhance the security of the system:

**1. Tighter Policies Regarding Passwords**

* Implement a rule mandating that passwords have a minimum of 12 characters.
* Use a combination of capital and lowercase letters, numbers, and symbols.
* Steer clear of dictionary words and popular patterns (e.g., no password123 or qwerty).
* Establish password expiration policies to have users change their passwords on a regular basis.

**2. MFA, or multi-factor authentication**

* Adding multi-factor authentication (MFA) will give protection a layer above and beyond passwords.
* MFA requires a second form of authentication (such as a code texted to the user's phone) in order to access the system, even in the event that a password is compromised.

**3. Put A Cap on Login Attempts**

* Limit the amount of erroneous login attempts that are permitted. Notify the user and temporarily lock the account after a predetermined amount of unsuccessful attempts—for example, five.
* By preventing the attacker from trying random password combinations, this helps prevent brute force attacks.

**4. Captch**

* To prevent automated attempts to crack passwords add a CAPTCHA to the login page.
* Because CAPTCHAs verify that a human is engaging with the page, they assist protect against automated brute force attacks.

**Conclusion**

The investigation uncovered serious weaknesses in API security and password management. It is advised to reduce these risks by enforcing MFA, strengthening password restrictions, and enhancing access controls. By taking these precautions, FinCloud's digital infrastructure will be secured, cybersecurity laws will be followed, and financial services and data will be protected.

**References**

ScienceDirect (n.d.) *Identifying vulnerability*. Available at: <https://www.sciencedirect.com/topics/computer-science/identifying-vulnerability> (Accessed: 15 September 2024).

Beyond Identity (n.d.) *Rainbow table attack*. Available at: <https://www.beyondidentity.com/glossary/rainbow-table-attack> (Accessed: 15 September 2024).

Rublon (n.d.) Brute force vs. dictionary attack: What’s the difference?. Available at: https://rublon.com/blog/brute-force-dictionary-attack-difference/ (Accessed: 15 September 2024).

Moody's (n.d.) *Cyber risk*. Available at: <https://www.moodys.com/web/en/us/capabilities/cyber-risk.html> (Accessed: 15 September 2024).