PHANTOM SRL

REPORT

Risk Management

Prepared For :
EPIC EDUCATION srl

By Phantom srl

Alessio D'Ottavio
Davide Di Turo
Francesco Pio Scopece
Giuseppe Pignatello
Luca Iannone
Manuel Di Gangi
Marco Fasani

PHANTOM SRL

REPORT

Risk Management

Prepared For :
EPIC EDUCATION srl

By Phantom srl

Alessio D'Ottavio
Davide Di Turo
Francesco Pio Scopece
Giuseppe Pignatello
Luca Iannone Manuel
Di Gangi Marco Fasani

PHANTOM s.r.l
IMPOSSIBLE IS OUR TARGET

INDICE

PHANTOM s.r.l
IMPOSSIBLE IS OUR TARGET

INDEX

TRACE

Using Adam Shostack's threat modeling framework, identify a threat to a software development company. What are we working on? What can go wrong? What are we going to do about it? Did we do a good job? Repeat the process, performing a gap analysis to find points for improvement.

NIST SP 800-53 Rev. 5. controls can assist in threat modeling:
NIST SP 800-53 Rev. 5-Security and Privacy Controls for Information Systems and Organizations NIST SP 800-53A Rev. 5 -Assessing Security and Privacy Controls in Information Systems and Organizations

Adam Shostack's threat modeling framework is a structured method for identifying and managing security threats in a system or process. This framework provides a structure for understanding and addressing potential threats proactively, rather than reacting only after security breaches have occurred.

To identify threats we consider the following scenario:

What are we working on?

The company is developing a home banking platform available for both web browsers and applications on mobile devices.

What can go wrong?

Vulnerability analysis
Vulnerabilities that may be present include:

- Code and Design Vulnerabilities:
  - Code security vulnerability
  - Unsafe architecture
  - Using insecure libraries or frameworks
  - Coding errors, such as invalidated input, which could make the system vulnerable to SQL injection or XSS attacks.

- Session Management and Authentication:
  - Poor management of user sessions
  - Failure to manage third-party authentications
  - Client-side session management issues
  - Weaknesses in access and authentication policies, which could allow unauthorized users to access the system.

- Data Protection and Encryption:
  - Incorrect implementation of encryption
  - Exposure of sensitive information
  - Lack of encryption of data transmitted between the local client and the server, exposing the information to potential interception during communication.

- Management and Maintenance:
  - Safety maintenance
  - Poor management of software updates
  - Failure to manage third-party vulnerabilities

- Resource Management and Performance:
  - Unmanaged resources
  - Performance issues
  - Absence of frequency limits and rate control

- Testing and Monitoring:
  - Failure to manage files uploaded by users
  - Failure to manage logs and monitoring
  - Failure to manage security tests

- Interaction with External Components:
  - Vulnerability in interaction with external components
  - Poor management of security updates

- Other Risks:
  - Failure to handle errors
  - Privacy violations
  - Failure to perform backups
  - Environmental disasters

Threat analysis

Using the STRIDE framework, we can identify the following threats:

- Spoofing: An attacker could pose as an authorized user to gain illegitimate access to sensitive data.
- Tampering: Manipulation of data during transmission between the external server and the database, compromising the integrity of the data.
- Repudiation: An attacker may deny having performed malicious or unauthorized actions, complicating accountability and incident management.
- Information Disclosure: Possible exposure of sensitive data due to insecure communication between the external server and the database.
- Denial of Service (DoS): Targeted attack to disrupt or degrade system performance, making services unavailable to legitimate users.
- Elevation of Privilege: Attempt to gain access to higher privileges than required to perform unauthorized operations on the system or database.

What are we going to do about it?

Preventive and threat mitigation measures

Using the DREAD mitigation framework, we can identify the following preventative measures:

- Defend: Implement rigorous access and authentication controls to ensure only authorized users can access the system.
- Remedy: Validate incoming data to prevent injection attacks and use encryption to protect communication between the external server and the database.
- Avoidance: Use of encrypted connections and secure protocols to minimize the risk of data interception or manipulation during transmission.
- Early detection and response: Implementation of continuous monitoring systems to promptly detect anomalies or suspicious activities and quickly respond to security incidents.

NIST SP 800-53 Rev. 5 Controls Provides a detailed set of security and privacy controls that can be implemented to protect information systems and organizations from a wide range of threats.

1. Threat Identification: The threat involves the compromise of sensitive customer data during software development, testing, or deployment.
2. Vulnerability Analysis: Considering NIST SP 800-53 Rev. 5 Security Controls:
   - AC-2 Account Management: Verification of procedures for managing credentials and authorized access to data.
   - SI-7 Software, Firmware, and Information Integrity: Protecting software from unauthorized modification during development and distribution.
   - SA-11 Developer Security Testing and Evaluation: Performing security testing during development to identify and fix vulnerabilities.
   - CM-7 Least Functionality: Ensure your software contains only the features you need to reduce your attack surface.
   - PL-4 Rules of Behavior: Implementing policies and procedures to properly manage personal information.
3. Corrective action planning:
   - Implement credential management and access control procedures to ensure authorized access to sensitive data (AC-2).
   - Use digital signatures or integrity checks to protect software from unauthorized modification (SI-7).

- Integrate security testing throughout the development cycle to identify and fix vulnerabilities (SA-11).
- Apply the "Least Privilege" principle to limit access to necessary functionality (CM-7).
- Apply stringent rules regarding the processing of users' personal data (PL-4).

4. Evaluation of the work performed:
   - Conduct periodic assessments to ensure the effectiveness of corrective actions and the adequacy of implemented security controls.

Did we do a good job?

NIST SP 800-53A Rev. 5 Controls

Provides detailed guidelines for evaluating the effectiveness of implemented security and privacy controls.

1. Identification of Security Controls: All relevant security controls implemented in the system have been identified and documented, in accordance with the requirements of NIST SP 800-53 Rev. 5.
2. Evaluation of the effectiveness of implemented security controls:
   - RA-03 Risk Assessment: Assessing the risk associated with compromising sensitive data during software development.
   - CA-07 Continuous Monitoring: Continuous monitoring of the development environment to identify and respond to security threats.
   - RA-05 Vulnerability Scanning: Performing periodic scans to identify and fix vulnerabilities in software under development.
   - SA-12 Supply Chain Protection: Assessing and mitigating risks associated with security in the software supply chain.
   - AC-01 Policy and procedures: Privacy-based access policy control, ensuring that personal information is accessible only to those who have the right to access it.
3. Control planning:
   - Conduct periodic risk assessments to identify new threats and update security countermeasures (RA-3).
   - Implement a continuous monitoring system to detect and respond to security threats during software development (CA-7).
   - Use automated tools to perform periodic software scans and fix identified vulnerabilities (RA-5).
   - Work with suppliers to assess and mitigate risks associated with supply chain security (SA-12).
   - Evaluation of work performed: Regularly evaluate the effectiveness of implemented security controls and make continuous improvements based on risk assessments (AC-01).
4. Evaluation of the work performed: Regularly evaluate the effectiveness of the implemented security controls and make continuous improvements based on risk assessments.

GAP Analysis

Gap analysis compares the current state with the desired objective, highlighting discrepancies and guiding corrective actions.

Unmet goals to address - What can go wrong bi.:

- Failure to manage third-party authentications;
- Incorrect implementation of encryption;
- Exposure of sensitive information;
- Safety maintenance;
- Failure to manage third-party vulnerabilities;
- Unmanaged resource;
- Vulnerability in interaction with external components;
- Failure to handle errors;
- Failure to perform backups;
- Environmental disasters.

NIST SP 800-53 Rev. 5 Controls - What we will do about it encore.
It provides a detailed set of security and privacy controls that can be implemented to protect information systems and organizations from a wide range of threats.

1. Threat Identification: The threat involves the compromise of sensitive customer data during software development, testing, or deployment.
2. Vulnerability Analysis: Considering NIST SP 800-53 Rev. 5 Security Controls:
    - IA-7 Identification and Authentication: Implementation of identification and authentication mechanisms to manage access to information systems and resources by third parties, such as external vendors, business partners or contractors.
    - SC-28 System and communication protection: Correct and secure implementation of encryption to protect sensitive information and communications from unauthorized or compromised access.
    - RA-5 Risk Assessment: Periodic evaluation of security controls to identify and mitigate vulnerabilities, including privacy-related ones, as specified by PL-4.
    - CM-2 Baseline Configuration: Define, document, and enforce a secure baseline configuration for devices, applications, and services, and monitor and manage configuration changes to ensure system security and integrity.
    - CM-9 Configuration management plan: manages the system configuration and establishes guidelines to ensure that all external components are properly managed and integrated into the system to ensure information security.

- RA-3 Risk Assessment: evaluation of the risks associated with errors, omissions and accidental or deliberate actions that could compromise the security of information or the integrity of systems.
- CP-9 System backup: Planning and executing backups of critical data and information systems to ensure the availability and integrity of information in the event of failures or disasters.
- CP-2 Contingency Plan: Development of business continuity and disaster recovery plans to address a wide range of catastrophic events, including natural disasters.

3. Planning of corrective actions:
   - During the evaluation, it will be necessary to verify whether the identification and authentication mechanisms for third parties are implemented correctly and whether they meet the criteria established in the security controls. The absence of such mechanisms or their incorrect implementation could lead to a negative compliance assessment. (IA-7)
   - During the evaluation, you will need to review the implementation of encryption to protect sensitive information and communications. If the implementation is deficient or inadequate, it could be identified as an area of non-compliance and could indicate a significant vulnerability. (SC-28)
   - Risk assessment is a key element in compliance assessments. If risk assessments are not conducted regularly or comprehensively, a lack of compliance and potential exposure to security risks may be reported. (RA-5)
   - During the evaluation, you will need to examine whether there is a documented baseline configuration and whether it has been applied correctly. The absence of a secure core configuration or lack of monitoring of changes could be considered a significant vulnerability. (CM-2)
   - Having an effective configuration management plan is critical to ensuring that configurations are managed appropriately and securely. The absence of a plan or its inadequate implementation could be identified as areas of non-compliance during the assessment. (CM-9)

4. Evaluation of the work performed: Regularly evaluate the effectiveness of the implemented security controls and make continuous improvements based on risk assessments.

Did we do a good job? BIS.

To evaluate whether we have done a good job, we will need to conduct regular security assessments of the software and the system as a whole. If we can effectively identify and mitigate risks and maintain a high level of data security, we can consider ourselves to have done a good job in protecting our software and our customers

Codice Web App Home banking


```
from flask import Flask, render_template, request, redirect, url_for import sqlite3

app = Flask(HOME BANKING)

DATABASE = 'database.db'

def get_db_connection():
  conn = sqlite3.connect(DATABASE) conn.row_factory = sqlite3.Row return conn

@app.route('/login', methods=['GET', 'POST']) def login():
  if request.method == 'POST':
    username = request.form['username'] password = request.form['password']

    conn = get_db_connection() cursor = conn.cursor() cursor.execute('SELECT * FROM users WHERE username = ?
AND password = ?', (username, password)) user = cursor.fetchone() conn.close()

    if user:
      return redirect(url_for('dashboard')) else:
      return render_template('login.html', error="Invalid credentials. Please try again.")

  return render_template('login.html')

@app.route('/dashboard') def dashboard():
  # Recupera le informazioni dell'utente dal database e passale al template conn = get_db_connection() cursor =
conn.cursor() cursor.execute('SELECT * FROM users WHERE username = ?', ('JohnDoe',)) user_info = cursor.fetchone()
conn.close()

  if user_info:
    return render_template('dashboard.html', user=user_info) else:
    return redirect(url_for('login'))

if __name__ == '__main__':
  app.run(debug=True)
```

Server side code Home banking

```
#include <iostream> #include <string> #include <winsock2.h> #include <ws2tcpip.h> #include
<random> #include <chrono> #include <thread>

#pragma comment(lib, "ws2_32.lib")

using namespace std;
using namespace chrono;

// Structure to store user information struct User { string id;
    string passwordHash;
    string sessionToken;
};

// Maximum time to wait for authentication (2 minutes) constexpr int MAX_AUTHENTICATION_TIME = 2
* 60;

// Function to generate a random session token string generate_session_token() { random_device rd;
    mt19937 gen(rd());
    uniform_int_distribution<> dis(0, 255);

    string token;
    for (int i = 0; i < 32; ++i) { token.push_back(dis(gen));
    } return token;
}

// Function to generate a 6-digit OTP (One-Time Password) code string generate_otp_code() {
random_device rd;
    mt19937 gen(rd());
    uniform_int_distribution<> dis(100000, 999999);

    return to_string(dis(gen));
}

}
```

```
// Function to authenticate user with two-factor authentication bool authenticate_user_two_factor(const string& id, const string& password, User& user) { // Basic credential authentication (ID and password) // If credential authentication basically fails, return false

    // Verify the OTP (One-Time Password) code string otp;
    cout << "Enter the OTP code: ";
    cin >> otp;

    // Let's simulate the verification of the OTP code string generated_otp = generate_otp_code();
    if (otp == generated_otp) { // Generate and store a new session token user.id = id;
        user.passwordHash = password;
        user.sessionToken = generate_session_token();

        // Update session token in database (simulated)

        return true;
    } else { cerr << "Invalid OTP code. Connection closed." << endl;
        return false;
    } }

// Function to handle a single connection void handle_connection(SOCKET clientSocket) { try { string id, password;
        cout << "Enter ID: ";
        cin >> id;
        cout << "Enter password: ";
        cin >> password;

        User user;
        auto start_time = steady_clock::now();
        if (authenticate_user_two_factor(id, password, user)) { auto end_time = steady_clock::now();
            auto elapsed_seconds = duration_cast<seconds>(end_time - start_time).count();
            if (elapsed_seconds > MAX_AUTHENTICATION_TIME) { cerr << "Maximum waiting time exceeded. Connection closed." << endl;
                // Blocca il software per 5 minuti this_thread::sleep_for(minutes(5));
                // Send a notification to the user's phone number (use Twilio or similar service)
```

```
            } else {
                cout << "Autenticazione riuscita. Token di sessione: " << user.sessionToken << endl;
            }
        } else {
            cerr << "Credenziali non valide. Connessione chiusa." << endl;
        }
    } catch (exception& e) {
        cerr << "Eccezione durante la gestione della connessione: " << e.what() << endl;
    }

    closesocket(clientSocket);
}

// Funzione per accettare le connessioni in arrivo
void start_accept(SOCKET serverSocket) {
    while (true) {
        sockaddr_in clientAddr;
        int clientAddrLen = sizeof(clientAddr);
        SOCKET clientSocket = accept(serverSocket, reinterpret_cast<sockaddr*>(&clientAddr),
&clientAddrLen);
        if (clientSocket != INVALID_SOCKET) {
            thread(handle_connection, clientSocket).detach();
        } else {
            cerr << "Errore durante l'accettazione della connessione: " << WSAGetLastError() << endl;
        }
    }
}

int main() {
    WSADATA wsaData;
    int iResult = WSAStartup(MAKEWORD(2, 2), &wsaData);
    if (iResult != 0) {
        cerr << "Errore durante l'inizializzazione di Winsock: " << iResult << endl;
        return 1;
    }

    SOCKET serverSocket = socket(AF_INET, SOCK_STREAM, IPPROTO_TCP);
    if (serverSocket == INVALID_SOCKET) {
        cerr << "Errore nella creazione del socket: " << WSAGetLastError() << endl;
        WSACleanup();
        return 1;
    }

    sockaddr_in serverAddr;
    serverAddr.sin_family = AF_INET;
    serverAddr.sin_addr.s_addr = htonl(INADDR_ANY);
    serverAddr.sin_port = htons(8080);
```

```
    if (bind(serverSocket, reinterpret_cast<sockaddr*>(&serverAddr), sizeof(serverAddr)) == SOCKET_ERROR) {
cerr << "Errore nell'associazione del socket: " << WSAGetLastError() << endl;
        closesocket(serverSocket);
        WSACleanup();
        return 1;
    }

    if (listen(serverSocket, SOMAXCONN) == SOCKET_ERROR) { cerr << "Error listening to the socket: " <<
WSAGetLastError() << endl;
        closesocket(serverSocket);
        WSACleanup();
        return 1;
    }

    start_accept(serverSocket);

    closesocket(serverSocket);
    WSACleanup();

    return 0;
```

The program :

This program is a server that manages user authentication via a two-factor process, using an OTP (One-Time Password) code. Start by initializing the Winsock library to allow socket communication. Next, it creates a TCP socket to accept incoming connections from clients and binds it to a specific port and local IP address. The server then starts listening for incoming connections using the listen function and accepts the connections using the accept function. Once it accepts a connection, it starts a new thread to handle the connection.

In the connection management thread, the server requests the user to enter the ID and password and then also requests the OTP code. If authentication is successful for both the ID and password and the OTP code, the server generates a session token and returns it to the client. Otherwise, it closes the connection. The server uses special functions to generate the random OTP code and session token.

The server has a maximum waiting time for authentication (2 minutes). If this time is exceeded, the server blocks access for 5 minutes and sends a notification to the user's phone number. When the server terminates, it closes the socket and releases the used resources.

Conclusions

The company implements several measures to address the threats, but there are still some points of improvement identified through this gap analysis. Focusing on these points can help the company further strengthen its security and better manage emerging threats in the context of financial software.

THANK YOU

BY

PHANTOM SRL

Prepared For :
EPIC EDUCATION srl

PHANTOM s.r.l
IMPOSSIBLE IS OUR TARGET