Case Study: **Vulnerability Assessment of a Protected Environment**

Assignment Description

> *Apriorit participates in development of a protected operating meta-environment, which enables corporate users to securely work with their local operating systems and securely exchange data.*

Research Type

> ***The client contracted Apriorit security testing experts to perform vulnerability assessment and penetration testing of the current beta-version of the environment. Both black box and white box testing were to be performed.***

Testing Goals and Plan

The team formulated security testing goals: to analyze possible vectors of both external and internal attacks aiming to steal or corrupt data as well as research the possibility of DDoS attacks and their potential impact on the environment performance.

Testing plan included such activities:

- ✓ Data at rest protection
- ✓ Data in motion protection: between system components and via network
- ✓ Authentication and authorization reliability
- ✓ System component protection: security of interfaces, installers, system data
- ✓ System code security and protection
- ✓ DDoS attack vectors and impact

At each stage of vulnerability assessment, limited penetration testing was performed using the discovered potential security breaches.

The team used such tools as:

> Metasploit tools, Kali Linux tools, nmap, Wireshark, PacketSender, netcat, SQLite Browser, WinHex, tcpdump, static code analyzers, Valgrind, Capstone, LOIC.

Testing Activities

The team set up a dedicated testing lab for this assignment where they deployed the tested environment and created needed testing data to simulate an operating instance.

Vulnerability assessment was made step by step according to the testing plan using two perspectives: a malicious hacker trying to target the system from outside and a malicious insider trying to elevate the authorized privileges / use other's credentials to access data outside their reach or perform otherwise unauthorized actions.

When researching data at rest protection, experts first of all focused on the proper encryption mechanisms and default permissions of system users and applications.

Vulnerability assessment of data in motion aspect required traffic encryption research, port scanning, network sniffing, and subsequent man-in-the-middle attacks and MAC spoofing attacks emulation.

When performing penetration testing via the authentication channel, the team tested authentication mechanisms and analyzed possible ways to violate the access restrictions. The storage of keys and passwords were properly researched, then common methods of stealing and using of authorized identity were checked.

Apriorit specialists performed vulnerability assessment of the solution architecture and implementation: the goal was to check that implemented software components cannot be used to disable or bypass product security features or to gain an unauthorized access. This part included analysis of available interfaces, unencrypted resources, and possible ways to manipulate with the installation and boot processes.

During the code security analysis, the testing team checked the code with the static code analyzers, researched possibilities to reverse engineer the solution binaries, and checked the vulnerabilities of the used third-party tools.

Finally, the system software and hardware interfaces were checked against various types of DDoS attacks. The goal was to detect and localize corresponding vulnerabilities and estimate the extend of potential problems with the performance and functionality.

Detected Vulnerabilities

Each stage of the vulnerability assessment discovered potential security problems. Those vulnerabilities were researched, prioritized and described in the report together with the corresponding recommendations. The most important recommendations concerned, among others, such topics:

- Minimize default user / process permissions
- Add validation for all actions that may cause data/operability loss
- Encrypt data at each stage of interaction and within each used network protocol
- Drop unencrypted network traffic
- Use strong encryption mechanisms, including proper "salt" usage
- Keep the unused ports closed
- Add protected storage for encryption keys and certificates
- Protect system components from debug and reversing
- Add system component integrity validation, especially at the installation stage
- Use the newest versions of the third-party tools and block the unused functionality of them to minimize potential vulnerability scope

- Add advanced input data validation, load balancers support and data migration mechanisms to deal with the potential DDoS attacks.

All provided recommendations were described in detail with the nuances of possible implementation.

Results

The vulnerability assessment report was delivered to the client and discussed within the joint team of developers and security experts. As the result, the team approved the solution refactoring plan to remove all the detected potential security problems.

This security audit with preliminary research and planning as well as post-research report discussions and consultations took 100 man-hours to be completed.