Homework 3 CSIT 5730

Written Assignment

Some questions in this assignment come from the textbook: `Information Security Principles and Practice`. Nevertheless, you don’t need to read the textbook in order to solve the questions.

This assignment has in total 100 points. That will count 4% of your final grade.

1. (12pt) This problem deals with static and dynamic security analysis.

   (a) (4pt) Discuss the pros and cons of static security analysis and dynamic security analysis respectively.

   (b) (4pt) Can dynamic security analysis be implemented as a ”sound” analysis? Explain your answer.

   (c) (4pt) Can dynamic security analysis be implemented as a ”complete” analysis? Explain your answer.

   (d) (4pt) Can dynamic analysis (e.g., fuzz testing) be enhanced by “static analysis”? Explain your answer.

2. (14pt) Symbolic Execution is a very popular static program analysis technique. Consider the following code snippet:

```
int twice(int v){
     return 2*v;
}

void testme(int x, int y){
     z = twice(y);
     if (z == x){
       if (x > y+10)
          ERROR;
    }
}

int main(){
    x = sym_input();
    y = sym_input();
    testme(x, y);
    return 0;
}
```

   (a) (6pt) How many possible execution paths are there for this code? Give a possible input ”x” and ”y” to trigger ERROR.

(b) (8pt) Explain the path explosion problem in symbolic execution. Discuss how to alleviate the path explosion in practice. You can search online but do not directly copy paste answers.

3. (22pt) Side Channels.

   (a) (6pt) What are three key components that form a side channel attack?

   (b) (4pt) Can fuzzing analysis be used to detect software timing-based side channel vulnerabilities? Please explain why.

   (c) (6pt) We have presented the RSA vulnerable implementation that leads to timing-based side channels in the lecture. Clarify that implementation and why it is vulnerable.

   (d) (6pt) Please clarify the three-step procedure of the "Flush & Reload" attack with diagrams.

4. (18pt) This problem deals with biometrics.

   (a) (4pt) Describe the pros and cons of biometrics over passwords.

   (b) (6pt) What are the fraud rate and insult rate for biometrics? Explain briefly.

   (c) (8pt) Dolphin attack is a kind of attack to voice recognition systems. You can find more information in: https://dl.acm.org/doi/10.1145/3133956.3134052 (it won the best paper award in a highest-ranked security conference). Please briefly explain how dolphin attack works. Propose a viable solution to prevent dolphin attacks.

5. (10pt) In this course, we discussed three types of firewalls: packet filter, stateful packet filter, and application proxy.

   (a) (6pt) At which layer of the Internet protocol stack does each of these firewalls operate? For each of these firewalls, discuss two types of available information.

   (b) (4pt) Will Firewalk work for application proxy firewall? Why or Why not? Explain your answer.

6. (6pt) This question is about authentication.

   (a) (2pt) Describe an attacking scenario that exploits the lack of CLNT or SRVR in a client-server authentication protocol.

   (b) (2pt) Describe the importance of certificate authority (CA) in the context of authentication of a server.

   (c) (2pt) Describe man-in-the-middle attack to attack a client-server authentication protocol.

7. (18pt) Smart contract security.

   (a) (6pt) Explain the function ”withdraw” in the following smart contract.

```
pragma solidity ^0.8.0;

contract Donation {
    mapping (address => uint) public balances;
    address payable public owner;

    constructor() {
        owner = payable(msg.sender);
    }

    function donate() public payable {
        // Receive donations
        balances[msg.sender] += msg.value;
    }

    function withdraw(uint _amount) public {
        require(balances[msg.sender] >= _amount,
        ”Insufficient␣balance”);
        balances[msg.sender] -= _amount;
        msg.sender.transfer(_amount);
    }
}
```

   (b) (6pt) Suppose if we modify the order of transfer and balance reduction i.e., the contract becomes:

```
    function withdraw(uint _amount) public {
        require(balances[msg.sender] >= _amount,
        ”Insufficient␣balance”);
        msg.sender.transfer(_amount);
        balances[msg.sender] -= _amount;
    }
```

   Describe the additional vulnerability introduced by this change.

   (c) (6pt) Discuss three possible types of vulnerabilities in smart contract. Describe how these vulnerabilities can be exploited by attackers.

Submission Instructions

All submissions should be done through the Canvas system. You should submit a pdf document with your answers for each question. Please check out the late submission policies on the course website in case you didn't attend the first lecture.