COMP3632– Final Exam

Samuel Petit
20683298

Question 1
(a) (1 point) The most effectiveness indicator of a static security analysis is code coverage.
TRUE
(b) (1 point) Non-inference defines that observable behavior of the program should not depend on confidential data.
TRUE
(c) (1 point) It is very difficult to fool a biometrics authentication system into making a false positive.
FALSE
(d) (1 point) Attack toward RSA timing side channel needs a precise timer.
TRUE
(e) (1 point) Spoofed email attack with SMTP is due to the lack of authentication scheme.
TRUE
(f) (1 point) Network discovery and port scanning are typical prerequisite of a network attack.
TRUE
(g) (1 point) Machine learning model integrity can be tampered via adversarial examples
TRUE

Question 2
(a) (2 points) Which of the following task cannot be addressed by taint analysis?
A. Data Lifetime Analysis: keep track of data usage and its lifetime in the memory.
B. Boost Fuzz Testing: study which data byte can cause software crash.
C. Information Flow Analysis: study how private keys are used in the software.
D. Malicious Behavior Analysis: perform fine-grind analysis on whether sensitive data flows to network APIs.

(b) (2 points) Which of the following aspects are not mainly considered when depicting a side channel attack?
A. Secret dependent program information flow.
B. Information flow affects physical environment.
C. Program public data affect program information flow.
D. Physical environment is exploitable by adversarial.

(c) (2 points) Suppose a Bitcoin user, Alice, attempts to change the global transaction ledger and convince miners to remove a previous transaction she paid. Why is she likely to fail?
A. Alice’s signature will be rejected because it is invalid for the new transaction ledger.
B. Alice will not have enough computational power to find enough proofs of work.
C. Other users will reject it after a vote because they know Alice is lying.
D. Miners will reject it because it is different from the consensus.

(d) (2 points) The passphrase method of choosing passwords is better than the traditional password choosing method because:
A. Password crackers don't usually use an English dictionary.
B. Even if the attacker knows you are using the passphrase method, it is still difficult to guess your password.
C. It is easier to remember a shorter password.
D. We substitute letters into numbers in a typical passphrase.

3. Due to your newfound cybersecurity expertise, the company running COMP3632.com has hired you to defend their website against network attacks. You are given a diagnosis of each issue and a suggested solution. For each diagnosis, (1) determine the type of attack, (2) explain how the attack works, and (3) determine and explain if the solution is effective.

(a) (5 points) Diagnosis: The web server for COMP3632.com keeps running out of memory and crashing. It tries to open and maintain too many ports all of a sudden, though there is no reason for there to be a sudden spike in the number of visitors. Solution: Use a network firewall to drop certain packets.

> The attack is a "denial of service attack". It happens because the server is receiving many new connections. Upon each connection request (or SYN packet received), the server "remembers" this connection & keeps it available for future communication. However when the server receives too many of these connections and tries to keep them all open, eventually the server will run out of memory and crash. Using a network firewall is in fact an effective way to protect against DoS attacks since it can, depending on how it is setup, follow different policies for accepting inbound and outbound packets, the firewall can act as an in-between actor between the server and user and that way the firewall is in a position to monitor packets and make a denial of service less likely.

(b) (5 points) Diagnosis: COMP3632.com sees almost zero inbound traffic. Customers have called to complain that they cannot reach the site after typing COMP3632.com into the address bar or clicking links. However, there seems to be no other network issues, and the web server for COMP3632.com can be reached by pinging the IP. COMP3632.com is not a banned site.
Solution: Setup a package filter to detect malicious network attacks.

> This attack could be a DNS poison : where a DNS server has received a false translation and caches the **poisoned** translation and supplies a wrong IP address to clients. Thus no directing any traffic to the website even though it is up and running. Setting up a package filter to detect network attacks won't work since the DNS is on the application layer and so detecting attacks on a network level won't help against DNS poisoning.

4. A well-known smart contract attack is called "Failed External Calls".

(a) (2 points) Give a sample contract that is vulnerable toward “failed external calls” attack. Who is the attacker in this case?

```
mapping(address => uint) public refunds;

function refundAllAddresses() {
  address[] refundAddresses; // assuming this value is initialised.
  for(unit current in refundAddresses) {
    // refund all addresses on failure.
    refundAddresses.send(refunds[refundAddresses[x]]);
  }
}
```

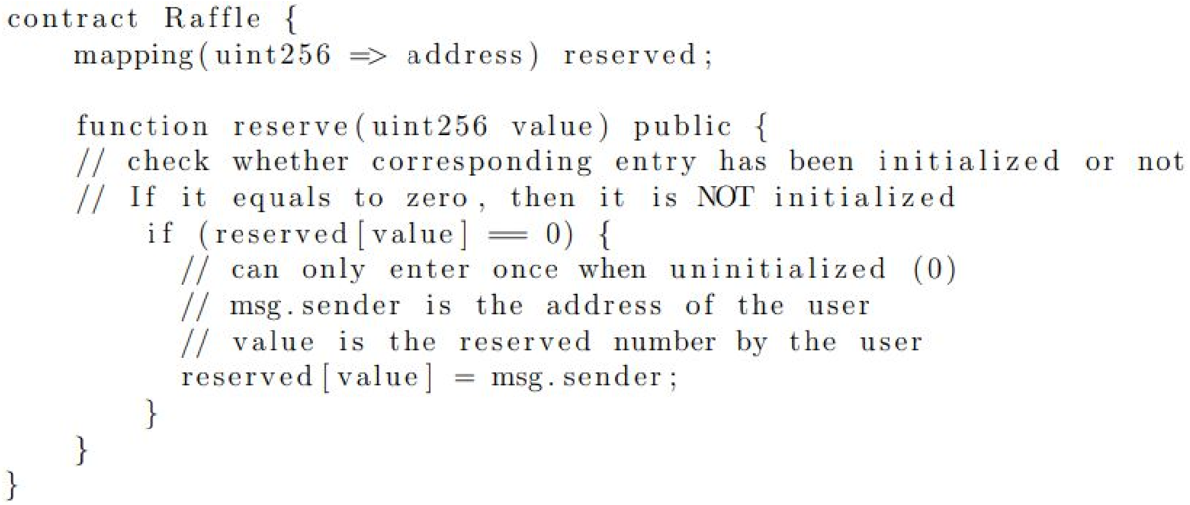
In this case the attacker is a bidder (user) who is using a smart contract which has a fallback function that will revert any payment he chooses and thus he can ensure to win any auction this way.

(b) (6 points) A common strategy to mitigate this attack is called “favor pull over push”. Explain this strategy with simple code fragments as an example. Why this can be used to mitigate “failed external calls”?
Favor pull over push is a method which favorises isolating each external call into its own transaction that can be initiated by the recipient of the call. This way a user cannot have access to other transactions and thus is not able to refund them unexpectedly.

```
class transaction {
  address highestBidderAddress;
  uint highestBidAmount;
  mapping(address => uint) refunds;

  function bid() {
    if(msg.value < highestBid) return;

    if(isInitialised(highestBidderAddress)) {
      // keep track of the amount to refund each bidder.
      refunds[highestBidderAddress] = refunds[highestBidderAddress] + highestBidAmount;
    }

    highestBidAmount = msg.value;
    highestBidderAddress = msg.address;
  }

  function refund() {
    uint refundAmount = refund[msg.sender];
    refunds[msg.sender] = 0;
    sendRefund(msg.sender, refundAmount);
```

```
 }
}
```

5. Consider the following contract, which acts as part of a typical Raffle game, where users transfer certain amount of Ether to function reserve and also reserve a number as the function input.



```
contract Raffle {
    mapping(uint256 => address) reserved;

    function reserve(uint256 value) public {
    // check whether corresponding entry has been initialized or not
    // If it equals to zero, then it is NOT initialized
        if (reserved[value] == 0) {
          // can only enter once when uninitialized (0)
          // msg.sender is the address of the user
          // value is the reserved number by the user
          reserved[value] = msg.sender;
        }
    }
}
```

(a) (2 points) Explain the potential “Transaction-Ordering-Dependence” bug of the above smart contract. (hint: note that a normal user when using this Raffle contract, will need to create a transaction including the transaction fee and the paid Ether. In that sense, can attackers somewhat trap a normal user to waste his money?)

> A transaction ordering dependence is an attack where the attacker will change the price during the processing of a transaction. Here because we receive information about the user before we reserve (initialise) the transaction, then it becomes possible for an attacker to use a different value than the user intended.

(b) (6 points) To defend Transaction-Ordering-Dependence attacks, one common practice is to use a “pre-commit” scheme. Explain how this scheme can be used to improve the implementation of the Raffle contract. From what aspect each “pre-commit” is more secure?

> A pre-commit scheme works by calculating a hash and sending that hash. The attacker can send the same hash but that will not prevent the legitimate user from registering the hash. Then that hash is put on the contract and then the hashed values are revealed to make the reservation. At this stage it is no longer possible to make a reservation thus, as long as the attacker cannot brute for the hash the transaction is secure.

6. We discuss how real-world RSA implementations can be exploited by cache-based side channels and timing-based side channels.

(a) (3 points) Cache-based side channel has been demonstrated on real-world commercial cloud computing services. Why are cloud computing services so vulnerable toward cache based side channel attacks? Explain your answer.
In an era where cloud computing services are used everywhere, virtual machines have never been more popular. It is the case that many VMs share the same cache and thus cache-based side channel attacks can be very dangerous.

(b) (6 points) Two RSA vulnerable implementations are discussed in the course, which enables cache-based side channels and timing based side channels. Explain these two vulnerable implementations.
The timing side channel attack is possible since the computation time for a 0 bit is much faster than a 1 then, if the attacker finds himself in a position where he can monitor the time used for each of those computations then he can find the key pretty easily.

The cache-based side channel attack is possible because accessing secret dependent values stored in memory will lead to secret cache access. Due to methods such as flush & reload, an attacker can then access such secret information and thus figure out what the secret key is.

(c) (9 points) Explain the general idea behind “Flush & Reload” attack toward cache-based side channels. In particular, 1) what is the typical attack scenario? 2) what is the capability of the attacker? and 3) what are the attack procedures?
hint: in case you are going to search relevant materials online, you might notice (or event get confused) about the prerequisite of this attack, for instance requiring read-only memory pages to be shared. You are not required to understand those technical details in order to answer this question. Just discuss high-level steps to launch the attack is sufficient.

If an attacked can determine which entries are stored in the cache over a period of time, he can then determine which cache lines may be interesting to him and then potentially launch an attack based on this. One of the ways this can be done is with the flush & reload attack, by clearing the cache & timing how long it takes to read the cache. If such time is fast then the attacker can determine that this piece of memory is being used, if it is slow then it is not in use. Over time while monitoring this, the attacker can obtain a pretty significant amount of information about a program the user is currently running.

(d) (9 points) Another commonly performed attack toward cache-based side channel is called “Prime & Probe”. 1) What is the typical attack scenario? 2) What is the capability of the attacker? 3) What are the attack procedures? hint: The same hint for question (c) also applies here.

Prime & probe is also a cache-based side attack. In this case the attacker also has control over the cache. This attack consists of 3 stages, the first one being the attacker filling the cache (or part of the cache) with his own data. The the process he is trying to attack would run, afterwards the attacker tries to access the cache he filled. If some of the data was evicted from the cache then a cache miss would happen and the attacker knows that this specific address was used by the program he is trying to launch an attack on. So similarly to flush & reload, the attacker potentially can have access to any piece of information from the program that is stored on the cache, this could be private information such as private keys...

7. Static and dynamic security analysis.

(a) (2 points) Explain the difference of static security analysis and dynamic security analysis.

Static security analysis performs “whole-program” analysis by parsing and analyzing the whole program as a tree or graph (i.e., Abstract Syntax Tree or Control-flow Graph) for interpretation and comprehension. There is no need to execute the program. Dynamic analysis, on the other hand, typically must run the program and see if certain security property violation can be identified or inferred.

(b) (3 points) Can taint analysis be implemented as a “dynamic analysis”? Explain your answer.

Yes. Basically we can perform taint analysis during program execution, by tracking the taint propagation dynamically on the execution trace.

(c) (4 points) Can dynamic security analysis be implemented as a “sound” analysis? Explain your answer.

No, it is not possible to implement sound analysis via dynamic security analysis. A sound analysis requires no false positives and dynamic analysis contains false positives by definition thus it is not possible.

(d) (4 points) Can dynamic security analysis be implemented as a “complete” analysis? Explain your answer.

Yes, it is possible to implement a complete analysis via dynamic security analysis. One would do so by running the programming while monitoring for errors and making sure that we execute all possible paths of the code base (ie have a code coverage of 100%). Note : we may get false positives that need to be confirmed manually.

8. Blockchain security.

(a) (6 points) Explain what is DNS cache poisoning attack. Can the Bitcoin Blockchain be attacked by DNS cache poisoning? Explain your answer.

As we’ve seen in a previous question, DNS cache poisoning is where a DNS server has received a false translation and caches the **poisoned** translation and thus supplies a wrong IP address to clients. In the case of bitcoin, there is no single location to poison since there are many many miners that would need to be poisoned in order it is not easily done however in theory it is possible that bitcoin could be attacked by a DNS cache poisoning attack.

(b) (6 points) Explain what is 51% attack toward Blockchain. What are the possible implications of this attack? Can 51% attack further enable double-spending attack? Explain your answer. What could be the reason such that the theoretically-possible 51% attack become practically impossible?

A 51% attack toward blockchain is the scenario where if a single entity controls 51% of all computational power, then this entity makes more proofs of transaction than anyone and thus could in theory make up any transaction they want valid since they have a majority. A 51% attack can very well enable a double spending attack since a digital token consists of a digital file which can be falsified or duplicated, a 51%

attacks enables the double-spending attack since a single entity now verifies all transactions.

A 51% becomes practically impossible if there are a sufficient amount of different entities controlling all the computational power such that it is now feasible for a single entity to obtain a majority in all of the computational power used by the blockchain.

9. We discussed information flow analysis (taint analysis) in the lecture.

(a) (4 points) Can we use information flow analysis to find cache-based side channel vulnerabilities in a RSA decryption procedure? If so, what are the corresponding taint source and taint sink points? If not, explain your answer.

Yes we can use information flow analysis to find cache-based side channel vulnerabilities in a RSA decryption procedure. One would do so by making the taint sink source the private key used in the decryption procedure. The taint source points for this specific scenario can be marked as any source of data which comes from the user (any untrusted data). We can then detect such vulnerabilities by checking if any of those taint source points make their way to the cache. We can then find potential vulnerabilities by analysing the cache.

(b) (4 points) Can we use information flow analysis to find Smart Contract vulnerabilities due to the mis-use of block.timestamp? If so, what are the corresponding taint source and taint sink points? If not, explain your answer.

It is possible to use information flow analysis to find smart contract vulnerabilities due to mis-use of block.timestamp, one would do so by marking block.timestamp as taint source and any value that is used when computing this value as taint sink points. By analysing the information flow of the block.timestamp variables we would be able to identify potential vulnerabilities when generating its value.

(c) (6 points) Can we use information flow analysis to find Smart Contract vulnerabilities due to the “Unexpected Revert Attack”? If so, what are the corresponding taint source and taint sink points? If not, explain your answer.

It would not be possible to use information flow analysis to find smart contract vulnerabilities due to the fact that the program will refund users as part of its normal functioning thus making it not possible to detect if the refund is normal or being executed by an attacker using information flow analysis.

In short, the attack is due to an external function call failing, not any flow of sensitive or untrusted information.

(d) (6 points) Can we use information flow analysis to find adversarial examples toward machine learning models? If so, what are the corresponding taint source and taint sink points? If not, explain your answer.

No it is not possible for information flow analysis to find adversarial examples toward machine learning models as it is not possible to use information flow to determine if data used on a machine learning model is valid or not. In this case it would be wiser to use information flow analysis try and detect vulnerabilities which could then help prevent adversarial examples from making it to the dataset.

10. A Generative Adversarial Network (GAN) is an important type of machine learning systems.
(a) (6 points) Briefly explain the design of GAN. What is the purpose of GAN? And what is the typical usage scenario of GAN?
It is a type of machine learning system which, from a given data set will generate new data with the same statistics as the training set. It is trained by giving it samples of the dataset until it can generate a new dataset which is accurate enough such that it is an acceptable data set. This way, in theory this type of model can generate as much training data as we need in order to train other models.

(b) (9 points) During the lecture, we discussed adversarial example attacks and membership attacks. In Homework 4 you are asked to study data poisoning attack. Can GAN be attacked by any of these three strategies? Explain your answer.

The GAN model may be trained with adversarial examples which will result in false data generation from the GAN model. Data poisoning can also work successfully against GANs, since training a model with wrong data will lead the model to give wrong answers in the end. Finally, although it is very challenging, membership attacks can still happen if the attacker re-trains a local copy of the GAN and check its response to inputs, although once again, this one is more challenging to execute.