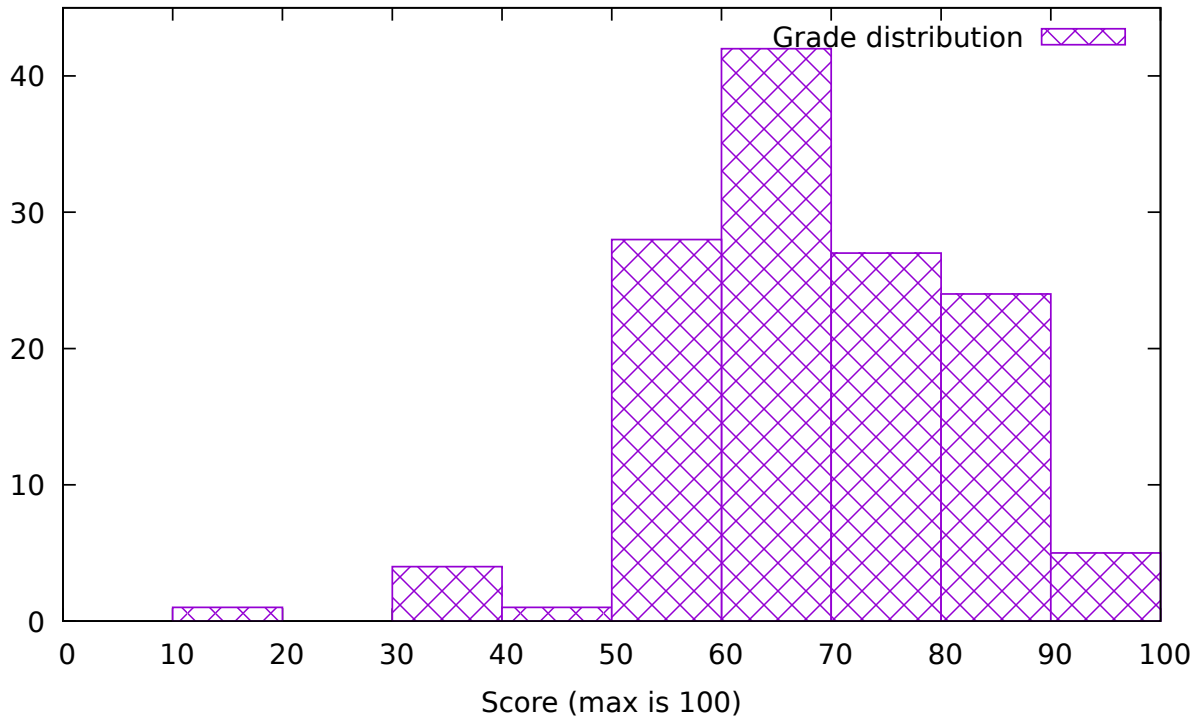
Department of Electrical Engineering and Computer Science

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

6.858 Spring 2018

Quiz II Solutions

Mean 68.3 Standard deviation 13.5



I Multiple-choice questions

For all of the multiple choice questions, please mark **all** choices that apply.

1. [4 points]: Based on the paper *"SoK: SSL and HTTPS: Revisiting past challenges and evaluating certificates trust model enhancements"*, which of the following statements are true?

A. Valid DV certificates provide more confidence to a user that she is connecting to the intended party than valid EV certificates.

B. OCSP stapling allows a server to prove to a browser that its certificate hasn't been revoked.

C. DANE makes it difficult for an adversary to launch a SSL stripping attack.

D. Server key-pinning makes it harder for an adversary to convince a CA to mint a certificate for a site and launch an MITM attack on that site.

Answer: B, C, D.

2. [4 points]: Based on the paper *"Click Trajectories: End-to-End Analysis of the Spam Value Chain"*, which of the following statements are true? "Spammers" here refer to operators of various parts of the "spam value chain."

A. Spammers run their spam-advertised web sites on compromised user machines that are part of a botnet.

B. Spammers need to register domain names in order for their spam-based advertisements to be effective.

C. Credit card network operators, such as Visa, perform random purchases to check whether transactions are correctly coded.

D. There is a high cost for spammers to switch acquiring banks.

Answer: B, D. We did not consider A to be correct: spammers do use botnets for sending spam, but largely avoid them for hosting their web site (though we accepted answers that said spammers use them as proxies). We accepted answers that said B was false because spammers could rely on URL redirectors like bit.ly.

3. [4 points]: Based on Mark Silis and Jessica Murray's guest lecture, which of the following statements are true?

A. Portions of MIT's `18.*.*.*` address space are announced via BGP by Akamai to protect against denial-of-service attacks.

B. The biggest security impact of selling parts of `18.*.*.*` address space to Amazon was that AFS had hard-coded permissions allowing access from any net-18 address.

C. MIT uses a single ISP, allowing IS&T to outsource firewall and intrusion detection work to them.

D. MIT IS&T relies on backups to defend against malware that encrypts user data and holds it hostage until the user pays a ransom fee.

Answer: A, D. We accepted an answer that said D is false because MIT relies on backups for *recovering* rather than *defending* against such attacks.

4. [3 points]: Based on the paper *"LAVA: Large-scale Automated Vulnerability Addition"*, which of the following statements are true?

A. The code added by LAVA is easy to detect.

B. Even if a bug can be detected, some bugs introduced by LAVA are unreachable (i.e., no input causes the buggy code to execute).

C. Even if a bug can be reached, some bugs introduced by LAVA are not exploitable (i.e., no input allows an adversary to execute arbitrary code).

Answer: A, C. We accepted answers that said A is false because existing tools have trouble locating these bugs, although the bugs are easy to find by looking for the exact code that LAVA inserts.

II Symbolic/concolic execution

Ben writes the following test case for lab 3:

```
def f(n):
   i = 0
   while i < n:
      print "1: Iteration"
      i += 1
   print "2: Done"
   return i

def test_f():
   v = f(fuzzy.mk_int("n"))

f_results = fuzzy.concolic_execs(test_f)
```

5. [6 points]: How many times would the `print "2: Done"` statement be executed? Explain where this number comes from, or why it's impossible to tell.

Answer: The code prints "2: Done" every time that `f()` is executed. Lab 3's concolic execution system runs `f()` once for every value of n that it tries (since n is specified to be a symbolic value).

Since n is a 32-bit signed integer, there are 2^{31} different execution paths through `f()`. All values of n that are zero or negative yield the same execution path (one where the loop executes zero times). Every positive value of n yields its own distinct execution path.

Lab 3's concolic execution system tries to find different inputs (in this case, the value of n) that lead to distinct execution paths. In this case, in principle, the concolic execution system might explore all of these 2^{31} paths. However, lab 3 contains a limit on the number of iterations that it will try, `max_iter`. In lab 3, this is set to 100, and as a result, the code will print "2: Done" 100 times.

We also accepted the answer 2^{31} or 2^{32} if accompanied by a good explanation along the lines of the above.

6. [6 points]: How many times would the `print "1: Iteration"` statement be executed? Explain where this number comes from, or why it's impossible to tell.

Answer: Every time `f()` is invoked with some argument n, the code prints "1: Iteration" n times (or zero times if n is negative). As a result, the exact number of times "1: Iteration" will be printed depends on the value of n that the concolic execution system chooses to pass to `f()`.

In lab 3's concolic execution system, the first choice for a concolic integer is 0, but subsequent choices are determined by Z3 when it is asked to give an example value of n that contradicts the existing path constraints. Z3 gives no guarantees about the value it chooses: for example, when asked to give a counterexample for $n > 0$, it could return $n = 1$, $n = 200$, or any other value $n > 0$. Thus, we were looking for the answer that it is impossible to determine.

We also accepted the answer 2451, because this happens to be the sum of values returned by Z3 for the above code snippet.

III Web security

Ben Bitdiddle proposes a new approach to protect against cookie-stealing attacks, like the ones you did in lab 4. Ben's idea is to use the `path` attribute of the cookie to limit which pages can access the cookie through `document.cookie`. In particular, a page can access a cookie only if its URL matches the cookie's path. Ben also modifies Zoobar to set the login cookie to `/zoobar/index.cgi/login`.

7. [6 points]: Can an attacker still obtain a victim's cookie, as in lab 4, with Ben's new defense mechanism and Ben's modified Zoobar? Either explain why this is not possible, or describe a specific attack, including snippets of HTML and Javascript code the attacker could use.

Answer: Inject the following code into some other page (like the profile page) via an XSS attack:

<iframe id="l" src="/zoobar/index.cgi/login"></iframe>

<script>

document.getElementById("l").appendChild("<script>alert(document.cookie);</script>");

</script>

If you described this in detail without code, we gave you full credit. However, if you just mentioned an XSS attack with little detail and no code, we only gave partial credit.

We also gave credit to those who explained that you can exploit an XSS bug in the login form.

IV TCP hijacking

As described in the paper *A look back at "security problems in the TCP/IP protocol suite"*, the TCP/IP protocol for establishing a connection uses the following messages:

A. C $\rightarrow$ S: SYN(ISN_C), SRC=C

B. C $\leftarrow$ S: SYN(ISN_S), ACK(ISN_C), SRC=S

C. C $\rightarrow$ S: ACK(ISN_s), SRC=C

D. C $\rightarrow$ S: byte0, ISN_C, SRC=C, ACK(ISN_s)

E. C $\rightarrow$ S: byte1, $ISN_C + 1$, SRC=C, ACK(ISN_s)

F. ...

TCP uses sequence numbers (as shown in message D and E) to detect duplicate bytes and deliver bytes to S in the order they were sent by C. As the paper describes, S increases the initial ISN by 128 every second and by 64 per new connection.

As described in the paper, this protocol is vulnerable to hijacking: an attacker can pretend to be C by suppressing message B and guessing the initial sequence number that S is proposing to C.

8. [5 points]: Assuming S isn't busy, how can an adversary guess the sequence number that it must send in message C efficiently? (Briefly describe.)

Answer: The adversary can open a regular connection to S, which will cause S to send the last ISN to the adversary. The adversary can then add 64 to that ISN value as a guess for the ISN that S will choose for the next connection from any client.

9. [5 points]: To defend against connection hijacking, Ben proposes to modify the TCP hand-shake protocol, replacing the initial ISN_S and ISN_C (in messages A and B) with random numbers, but still incrementing them sequentially as data is transmitted. Would this make hijacking harder? (Briefly explain.)

Answer: Yes, the adversary has a hard time guessing the next sequence number to use. (Although we did not ask for whether this is a good idea, it happens to not be one: TCP sequence numbers must increase slowly in order for TCP to detect duplicate connection requests.)

V Secure handshake

Alisa layers the following protocol over TCP/IP to set up a secure channel that provides confidentiality and integrity:

A. C $\rightarrow$ S: connect with TCP

B. C $\leftarrow$ S: PK_t, Sign(SK_A, {S: PK_S}), Sign(SK_S, {PK_t})

C. C $\rightarrow$ S: Encrypt(PK_t, {K, SN})

D. C $\rightarrow$ S: c_1 = Encrypt(K, {msg_1, SN}), t = MAC(K, c_1)

E. C $\rightarrow$ S: c_2 = Encrypt(K, {$msg_2, SN+1$}), t = MAC(K, c_2)

F. ...

Public keys are denoted as PK, and their corresponding secret key as SK. A private key for symmetric ciphers is denoted as K. "SN" is a unique sequence number. { and } mark a message.

You can assume that everyone knows the PK_A of the certificate authority. Furthermore, as soon S doesn't need SK_t in the protocol anymore (i.e., right after processing message C), S deletes SK_t from memory.

10. [6 points]: Ben doesn't like certificate authorities so he suggests replacing message B with:

C $\leftarrow$ S: PK_t, Sign(SK_S, {PK_t}).

That is, the server sends C a signed public key, but without the certificate. Ben also requires client C to store the PK_S in C's file system on the first successful connection to S. He further modifies the protocol to check the PK_S in message B against the stored key on subsequent connections.

What attack could an adversary launch? (Briefly describe the attack.)

Answer: An adversary could launch a man-in-the-middle attack (i.e., impersonate the server S using the adversary's version of PK_S and SK_S), on the first time that C connects to S. If the adversary did not intercept the first connection, subsequent connections from C to S are not vulnerable to man-in-the-middle attacks because C remembers the correct value of PK_S.

11. [6 points]: If the protocol doesn't include a unique *SN* in message C and D, what attack could an adversary launch? What concrete steps could an adversary take, which messages would he need to receive or send, and what bad thing would happen as a result?

Answer: The adversary could replay message D, without the server being able to determine whether the client really sent another copy, or whether this is a replay attack. This might cause the server to process message msg_1 twice.

12. [5 points]: Ben proposes to simplify the protocol by replacing PK_t with PK_s in message B and C. That is, Ben's protocol omits using PK_t. What attacks is Ben's protocol vulnerable to that Alissa's protocol isn't? What concrete steps could an adversary take, which messages would he need to receive or send, and what bad thing would happen as a result?

Answer: Ben's protocol doesn't provide forward secrecy. If an attacker manages to steal SK_s, he will be able to decrypt K and all communication.

VI Side channels

Consider the Spectre example implementation shown in Append A of the paper "Spectre attacks: exploiting speculative execution" by Kocher et al. In this example, the code attempts to learn the secret "secret" through a side-channel by running the function `victim_function`.

Suppose that you changed the value "256" to "128" on lines 49 and 81.

13. [6 points]: Would this code still print out the same secret?

Answer: Yes. This changes the attack to consider only byte values 0 through 127. The attack still works because ASCII strings consist of byte values below 128.

14. [6 points]: Suppose that the secret value is an arbitrary AES key (recall that AES keys are 128-bit values). Would this attack work to recover the AES key?

Answer: 128-bit AES keys are stored in memory as 16 8-bit bytes. Each byte value ranges from 0 to 255. Since this question's modified Spectre attack considers byte values from 0 to 127, it would be unable to recover half of the bytes in the AES key, on average.

VII Bitcoin

At the beginning of the semester, Alyssa P. Hacker got 1 Bitcoin and challenged Ben Bitdiddle to steal it. Alyssa tells Ben the transaction in which she acquired 1 Bitcoin (and her public key), but Ben does not know Alyssa's private key (and Alyssa stores it in a way that Ben cannot obtain it).

Ben has been working hard on this problem, and found a friend that can lend him some chips that compute SHA-256 hashes much faster than existing Bitcoin miners—so much so that they give him about 60% of the "mining power" in Bitcoin! Ben's friend is willing to lend these chips to Ben for a few months.

15. [6 points]: Suppose Alyssa transfers her Bitcoin to her friend Charlie. Can Ben transfer Alyssa's Bitcoin to himself? Give a precise attack or explain why not.

Answer: No, because Ben cannot generate a transaction signed by Alyssa's private key. Ben doesn't have enough time to fork the blockchain from before Alyssa got the Bitcoin.

16. [6 points]: Can Ben ensure that Alyssa cannot give her Bitcoin to someone else? Describe how, or explain why not.

Answer: Yes, Ben can prevent Alyssa's transactions from going into Ben's blocks, and out-mine any other blocks that include Alyssa's transaction.

VIII Tor

Alyssa's nosy neighbor, Norbert, runs two popular web sites. Alyssa is visiting one of Norbert's web sites through Tor, but Norbert doesn't know which one. The only thing Norbert can observe is Alyssa's encrypted WiFi transmissions (he does not know Alyssa's WiFi password, or which Tor nodes she happens to use).

17. [6 points]: Describe an attack by which Norbert can determine which of Norbert's two sites Alyssa is visiting.

Answer: Many possibilities. Turn off one web site and see whether Alyssa still has significant traffic. Change the size of the pages on the two web sites to be very different, and observe how much data Alyssa is receiving. Correlate timing between Alyssa's packets and requests at Norbert's web servers.

IX Secure messaging

The 6.858 course staff design the following messaging protocol, which is a variation of the protocol discussed in lecture. Each user maintains a long-term signing key, *SK*, and knows the other users' corresponding public keys, *PK*. To start a conversation, each user chooses a fresh Diffie-Hellman key (say, a for Alice) and sends the corresponding public key (say, g^a for Alice) to the other user. Each user signs their public key (with their long-term signing key), to prevent MITM attacks, and then signs the other user's public key, to acknowledge its receipt:

A. A $\rightarrow$ B: g^a, "Alice", Sign(SK_{Alice}, $\{g^a\}$)

B. A $\leftarrow$ B: g^b, "Bob", Sign(SK_{Bob}, $\{g^b\}$)

C. A $\rightarrow$ B: "ACK", Sign(SK_{Alice}, $\{g^b\}$)

D. A $\leftarrow$ B: "ACK", Sign(SK_{Bob}, $\{g^a\}$)

The users (Alice and Bob in this example) then derive a shared secret key, g^{ab}, using Diffie-Hellman, and exchange messages by using `Seal()`, as described in the lecture notes. `Seal()` provides authenticated encryption: an adversary cannot determine the contents of the message, or construct another valid message, without knowing the shared key. Messages must unseal correctly (specifically, `Seal`'s MAC tag must match) before they are displayed to the recipient.

18. [6 points]: Describe how David, one of the 6.858 TAs, can send the message "Lecture is canceled" to Nickolai as if the message was sent by Frans. Assume there are no software bugs, the cryptographic primitives (signatures, Diffie-Hellman, and `Seal`) are secure, and David does not have physical access to Nickolai's or Frans's computer.

Answer: David should pick x and send g^x to Frans, as if establishing a new connection. When Frans eventually responds with an ACK of g^x, David will have "Sign(SK_{Frans}, $\{g^x\}$)".

Now, David establishes a connection to Nickolai, sending this g^x value along with Frans's signature on that value. This will cause Nickolai's client to believe that Frans is establishing a connection with public key g^x (i.e., step A).

To send the acknowledgment of Nickolai's g^n (in step C), David needs to open a second connection to Frans, sending Nickolai's g^n (from Nickolai's step B) to Frans, and similarly wait for "Sign(SK_{Frans}, $\{g^n\}$)", which he can send in step C to Nickolai.

We also gave some credit for solutions that claimed to replay past messages between Nickolai and Frans that said "Lecture is canceled", as well as solutions that mis-spelled the names of the parties in question.

We did not give credit for solutions that required stealing the private keys from Frans, bribing him, using a phishing attack, etc.

X 6.858

We'd like to hear your opinions about 6.858. Any answer, except no answer, will receive full credit.

19. [2 points]: Are there any papers in the second part of the semester that you think we should definitely remove next year? If not, feel free to say that.

Answer: 35x LAVA. 20x Tangled Web. 11x Security economics. 9x SoK papers. 8x Messaging. 5x Bitcoin. 5x TCP. 4x SSL. 2x Tor. 3x Spectre. 2x EXE. 1x Capsicum.

20. [2 points]: Are there topics that we didn't cover this semester that you think 6.858 should cover in future years?

Answer: 11x Network security: firewalls, DoS, worms, hands-on. 10x Kerberos. 10x IoT/hardware security. 6x More side-channel attacks; lab? 4x More crypto. 4x Cryptocurrencies, blockchains. 4x Real-world, recent attacks/exploits. 4x Botnets. 3x Penetration testing. 3x Ur/Web. 3x Reverse engineering. 2x Hacking ethics. 2x Database security. 2x SUNDR. 2x MIT/Athena security. 2x More sandboxing. 2x Static analysis / formal methods. 1x More lectures by David. 1x Cover material in lecture that's not in the readings. 1x UI security. 1x Social engineering. 1x Signal paper instead of Messaging. 1x Vulnerability classification. 1x Password managers. 1x Corporate security from management perspective. 1x Link-layer network security. 1x BitTorrent. 1x How to defend against lab4 attacks? 1x ML security. 1x Stuxnet. 1x NSA attacks. 1x Control flow integrity. 1x ForceHTTPS.

End of Quiz