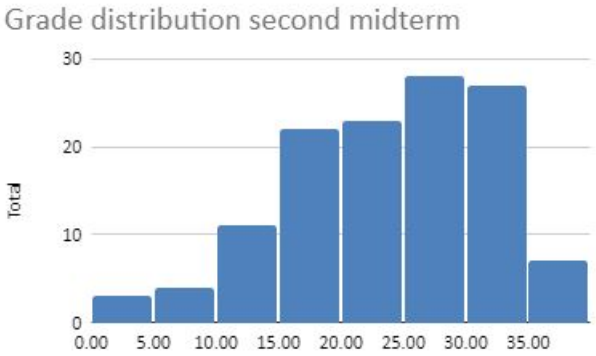
COM-301 - Second Midterm
Most Repeated Errors

November 27, 2021



General Advice

- Answer the question. Do not provide a mitigation to an attack if you are only asked about the attack. If the proposal is wrong it will result in negative points.
- Stick to the number of lines you are given. We can be lenient for couple of words past the limit or a little note. When the norm was clearly disregarded we did not consider the content, effectively reducing points.

Multi-choice Question: Password Security

Error: password hashing algorithms increase pre-image resistance. Hash algorithms designed to hash passwords (scrypt, bcrypt) have as goal to hamper offline attacks in which the adversary precomputes passwords hashes. To this end, these algorithms are *on purpose* made slow. They, however, still provide strong pre-image resistance – the only required property for securing passwords.

Multi-choice Question: Authentication

Error: In the token-based authentication scheme seen in the class, both token and server use the same hash function. Recall that the scheme in the class

the token authenticates by computing the same number as the server applying a function on a pre-agreed seed value. It is crucial that only the token can compute this value, thus, this must be a keyed function. Therefore, a hash function – that anyone could compute – is not adequate for this purpose (see slide 36 in the Authentication lecture).

Multi-choice Question: Applied Cryptography

Error: A MAC is a good choice to provide integrity only *when the symmetric key is signed.* Whether the fact that the symmetric key is signed or not is relevant for integrity depends on the communication protocol. In any case, it is not a pre-requisite for a MAC to be a good choice, so it is not the case that *only* if the key is signed we should use MAC.

MACs, however, are a good choice for integrity when the partners have a *pre-shared* symmetric key, as that guarantees each of them that only one of the two could have written the message. As the key is pre-shared and never communicated, an adversary cannot modify it or change it.

Multi-choice Question: Attacks

Error: Cross Site Request Forgery can be avoiding by sanitizing cookies. The cookie used in CRSF is the original cookie from the server and thus it is sanitized. The problem is that it is sent from another page while the user is not visiting the intended website .

Short Question: Software Security

```
int bonus[100] =  { 0 };                    /* array of 100 integers initialized to 0 */

char successDB(char* mission) {

    /* function that returns 0 for failed missions */
    /* and 1 for successful missions */
    /* if the mission does not exist, it crashes */

}

1: int AddMission(int minionID) {
2:      char success;                       /* mission name */
3:      char mission[30];                   /* one byte: 1 if mission succeeds, 0 otherwise */
4:
5:      gets(mission);                      /* reads mission name from keyboard */
6:      success = successDB(mission);       /* checks in the DB the success of the mission */
7:
8:      if (success == 1) { bonus[minionID] += 1 }; /* if mission succeeded, increase bonus */
9:
10:     return 0;
11: }
```

Error: Overwriting the return address by giving by inputting a long mission grants that you get bonus. What return address would you overwrite? where would it point to? Just saying that it overwrites the return address without explaining how it gets to the bonus is not enough. In any case, obtaining bonus by modifying the flow in this program is very difficult, as the return address you can overwrite with mission is reached after line 8, where the bonus is assigned.

Error: Overwriting `success` with 1 by inputting a long mission grants that you get bonus. Indeed, the code enables the adversary to overwrite `success` to 1 by inputting a long mission name. However, if you do that, the mission name will not be valid, and `successDB` in line 6 will crash. When it crashes, you never get to line 8 and there is no bonus.

Even if the program would not crash, `success` would be overwritten in line 6, so no guarantee of bonus.

Error: `char mission[30]` is a bug Line 3, by itself, is not a bug and cannot be exploited. It is the use of the `mission` variable in unsafe ways in lines 5 and 6 what generates the problem, not the declaration.

Short Question: Applied Cryptography

Error: To prevent modifications on the hardrive content the hash function needs to be collision resistant. Collision resistance means that it is hard to find *any* two inputs that hash to the same value. These are two *free* inputs. In the question, one input was fixed: The scientists make one hash of the content *before* passing the text.

This means that one input is fixed, and it should be hard to find another input that hashes to the same value. This is Second pre-image resistance.

Short Question: Software security

Error: Measure performance with bugs (or any ratio bugs to X). Indeed, finding bugs is the goal and more bugs is better. However, this is not a good way of measuring the success of the technique. The idea is to make sure that you have find as many vulnerabilities as possible. That means that you want to execute as much of the code as possible. In the lesson we learned that you need to take care of flow and data flows. The best (feasible) way to do that is to ensure that all branches are executed (branch coverage) with all possible values (data coverage). Note that "code coverage" is imprecise as it does not explain how to measure that coverage; and "statement coverage" is insufficient as explained in the lecture.

Short Question: Software security

Error: Code passed as a parameter can be executed when DEP is active. Data execution prevention means that pages in memory are either writable or executable. As the stack is writable, it cannot be executed. So this mechanism is not a good defense for this code because it will not let the code execute. Besides, DEP only protects against code inserted in the stack, but not control-flow attacks.

Short Question: Network Security

Error: HTTP provides confidentiality. HTTP without the S implies that the HTTP connection is not done over HTTPS. Thus, the content is not encrypted and the connection is not confidential.

Error: DNSSEC provides confidentiality. DNSSEC ensures that DNS recods are signed by authorized resolvers. As such, they cannot be poisoned at the recursive resolvers, nor they can be hijacked/spoofed. However, there is no encryption. Thus, DNSSEC does not provide confidentiality. Only integrity and origin authentication.

Error: BGP to reroute to the adversary's server In a BGP hijacking attack, the adversary can change the route packets will follow *between server and receiver*, but *it cannot change the receiver*. Recall, what the adversary does is corrupting a router to advertise cheaper routes so that traffic comes to this router and can be observed. This cannot change the destination of the communication. BGP hijacking, however, can be used to gain man-in-the-middle position to redirect packets using other attacks. The chain of attacks needs to be made explicit in the answer to get the points.

Important: The economy of mechanism principle also applies to adversaries!!! They will not deploy the most cumbersome, expensive attack, but they will do the simplest attack possible. By proposing a more complicated attack you run the risk of not being technically accurate – and thus you will not get full score in the question.

Short Question: Network Security

Error: Not explaining why the MITM can happen. HTTP without the S implies that the HTTP connection is not done over HTTPS. Thus, the content is not encrypted not authenticated. This is a necessary condition for the adversary to be able to perform the MITM. If it was HTTPS the adversary would not be able to read the content even if it would be re-routed!

Error: Using ARP spoofing to claim the MAC of the server. ARP spoofing is an attack to change the pairs (MAC, IP). However, the idea is not to claim the MAC of the server: if you have two machines with the same MAC address on a LAN, the LAN breaks. What one does in ARP spoofing is convince the sender that the IP of the server corresponds to the MAC of the adversary.