COM-301 - Mini-Exam 4
Most Repeated Errors

November 27, 2021

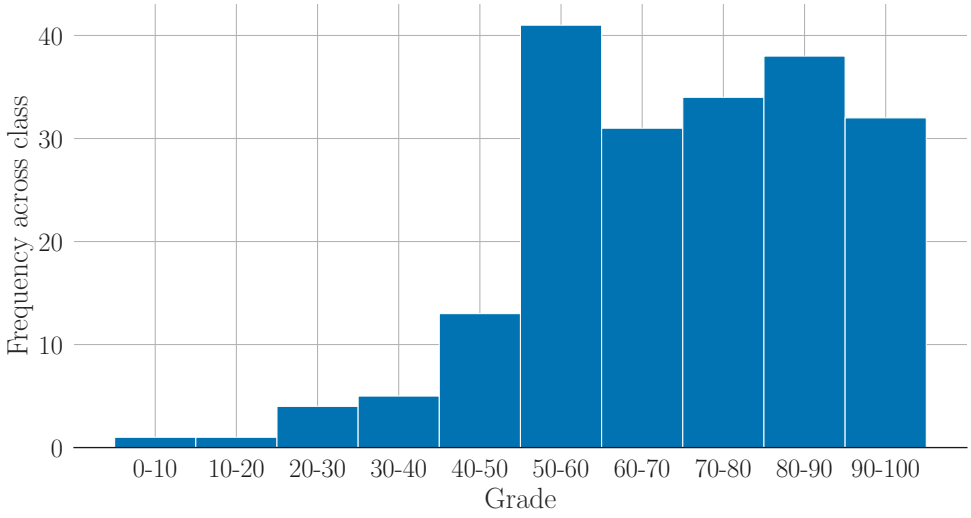

Figure 1: Distribution of grades across class

General Advice

- Read the question carefully and try to answer exactly what you are asked for. If the question asks you to describe **one** attack, describe just one attack not multiple. If the question asks whether an attack works, explain that, not an alternative attack. If the question **does not** ask you to facilitate a communication, don't provide means to facilitate the communications.
  Extending your answer beyond what you have been asked for is risky. If the additional details given are incorrect it will make you lose points. We need to grade everything that has been written in your answer, we can-

not just select the parts that are correct (think of this as marking several answers in a multiple choice question).

- If the question uses some names or specific terminology, stick to the terminology in the question. This facilitates the grading as it leaves no room for misinterpretation.

Question: Sanitizers and Symbolic

Error 1: Using address sanitizers in deployment. Address sanitizers are too costly (high performance hit makes the programs too slow) to be used in deployment. Alice can use address sanitizer in the testing phase to find bugs, but she cannot use it as a mitigation on the election machine. We give points to students who reasoned that this approach is very costly but as the security is critical this defense should be deployed.

Error 2: Using symbolic analysis in deployment. Symbolic analysis is a static method. This means that symbolic analysis is used before running the code and applying this protection on the election machine does not make sense. Besides the static nature, symbolic execution can only scale to small programs and cannot handle election code.

Question: Ricco's cafe

Error 1: HTTPS can be man-in-the-middled or it's IP changed. HTTPS uses TLS to secure the communication. As such, the server is authenticated, i.e., if Rita gets to https://cutestkittens.com, Rita knows that she is in the correct domain. A MITM would not have the certificate and would not be able to sign. Protection against MITM is one of the main features of TLS
A DNS attack that forwards Rita to other IP address would not work, as the host in that IP address would not have the server certificate of https://cutestkittens.com.

Error 2: Try ARP spoofing on a device connected to the internet. Once a device has started a connection outside of the LAN, this devices has the MAC address of the gateway. Therefore, this device will not launch a new ARP request and cannot be attacked with ARP spoofing.

Question: TA Chat

Error 1: ARP spoofing to change the MAC of a packet. Many students tried to use ARP spoofing to change the MAC of a packet. ARP spoofing only convinces other machines of an incorrect MAC-IP association, but it **does not** change the MAC of the packets.
Also, it **cannot** change a white list of MACs.

Error 2: ARP spoofing on the server. The server does not initiate connections, and therefore it does not lauch ARP requests. Since it does not use ARP,

you cannot spoof the ARP responses and manipulate the server MAC/IP cache.

Error 3: ARP spoofing remotely via VPN. ARP is a protocol to discover the MAC address corresponding to an IP in a local area network. One cannot launch ARP requests in other LAN, nor see the ARP responses of other LAN.

Error 4: MAC spoofing remotely via VPN. This attack cannot be performed remotely. If your machine is not local you cannot spoof the MAC of packets. For machines inside the remote LAN, the MAC of your packets will be the MAC of the machine through which you access the LAN via VPN. You can't change the MAC of this machine from your computer (or you need to describe how this would be done).

Error 5: DNS hijacking/poisoning to read the history. Getting the TAs to connect to the adversary instead of the server does not provide the adversary with the history. The history is sent by the real server. If the server is not involved in the communication you cannot receive the history.

Error 6: Man-in-the-middle to spoof. A man in the middle attack does *not* relay link-layer packets. If you have a man in the middle you are changing the MAC address. Thus, the server will not accept your connections.
(Those that specified that the MITM would maintain the MAC address received partial points)

Question: Bobby & Sawit

Error 1: Providing an (incorrect) attack instead of answering whether Bobby's setup works. Some answers provided attacks (e.g. ARP spoofing) that the IT team could launch to intercept Bobby's communication. However, this does not answer the question of whether Bobby is protected, and in almost all cases, the proposed attack does not lead to a desired objective. For example, intercepting an ARP request does not allow the IT team to learn that Bobby is posting memes, or even that he is visiting Sawit (the ARP requests are for machines in the same LAN and are used to obtain an IP-MAC mapping for machines in the LAN). Similarly, running a DDoS does not help the IT team learn about Bobby's activities on Sawit.

Error 2: Confusing DNS hijacking and cache poisoning. A DNS cache posioning corrupts the DNS resolvers with fake domain/IP pairings. A DNS hijack is a man in the middle attack where the responses from a DNS resolver are corrupted. Some answers mentioned that the IT team was performing a cache poisoning even though the question described a DNS hijack.

Question: Flipagram

Error 1: Not stating assumptions for an attack or describing attacker capabilities required. If an answer uses assumptions beyond what is stated in the question,

those assumptions should also be stated. Also, the capabilities required for an attacker to carry out the attacker should be specified. For example, if a TCP SYN flood attack is performed by a single user, the answer needs to explain why the photo upload limit is bypassed. Another example is if an attacker is performing an MitM, the answer has to state what parties are being attacked, where the victims/attacker are located.

Question: getPassword

The errors below are listed based on the following:
(1) 'gets' does not stop after the first null terminator (in fact, it only stops at EOF and newline, which are distinct characters from null terminators),
(2) the stack layout is as in class, i.e. an overflow in the password array overwrites the isCorrect value,
(3) the fuzzing strings are composed of only null-terminator characters (which is, in fact, a possible fuzzing strategy),
(4) size of int and memory pointers is 4 bytes.
Basing off different considerations, e.g., assuming 'gets' stops after the first null terminator, on its own did *not* result in penalties. That is, we graded considering your assumptions.

Error 1: Saying that the fuzzing strategy does not uncover the vulnerability only because strcmp always returns 0. This statement provides only half of the reasoning, and did not receive full points. The fuzzer will cause an overflow of the password array that overwrites the isCorrect variable. However, the overflowed value of isCorrect is still 0, resulting in no detectable unexpected behavior. Thus, the fuzzer does not uncover the vulnerability also because the fuzzing input arrays only contain zeros, *independently of the strcmp outputs.*

Error 2: Saying that the fuzzing strategy will uncover the vulnerability "because of the overflow"/"because the overflow causes a crash." As mentioned in the exercises, not any overflow is detectable or causes a crash. In this case of testing with strings of length up to 15, an overflow can only overwrite the value of the isCorrect variable, which cannot cause a crash.

Error 3: Not explaining why allowing characters other than null terminator uncovers the vulnerability. This is especially relevant if the response assumed that 'gets' stops at the first null terminator, thus did not have to elaborate on the issue with overwriting isCorrect in the first part of the answer. It is important to correctly justify why exactly allowing non-zero characters uncovers the vulnerability: If the fuzzing input size is up to 15, it overwrites isCorrect with a non-zero character, which is detectable. If the fuzzing input size is more than 16 (size of password and isCorrect) + 4 (saved base pointer), it is likely to cause a crash/segmentation fault.

Question: PDFuzz

Error 1: Using a dumb fuzzer is enough to cover the code In this scenario, using

a dumb fuzzer is a very bad idea, as the overwhelming majority of randomly generated inputs will not even be able to pass format checks. Thus, dumb fuzzers will be mostly useless when it comes to testing the functionalities of the PDF reader and will not be able to cover much of the code at all. Generation-based fuzzer is way more adapted in this case, where the input format is a public and well-known. It is important to note that we discuss coverage in a limited execution time. If we give infinite time to a dumb fuzzer, it will eventually try all binary inputs. however, some checks will take a long time, e.g. get past a simple corruption check like CRC.

Error 2: White-Box fuzzing being necessary to have good coverage of the code Since the PDF Protocol is public, knowing it gives you enough information to test the program thoroughly. Of course, white-box fuzzing will lead to better results, but it is not necessary to get good coverage on the code. Since many companies would not want to share their code, it's important to ask yourself if having access to the code (required for white-box fuzzing) really is necessary to do good fuzzing.

Question: Censoring WeaselNews

Error 1: BGP Hijacking allows the Swiss government to redirect traffic to their own server. BGP Hijacking **itself** does **not** allow the Swiss government to redirect traffic to their own server. Through BGP hijacking the Swiss government could only achieve that requests to WeaselNews' servers get dropped and never reach their destination. To achieve redirection, the Swiss government need to chain the BGP hijacking with other kind of spoofing to achieve the redirection. Alternatively, the government could deploy DNS Spoofing to redirect traffic.

Error 2: Recommend to use a "firewall" without specifying how this would be implemented. Answers that just recommended to use a "firewall" were counted as incomplete. We expected the answer to explain what it would require to implement a firewall on a national level (i.e., monitoring all entry/exit points of the Swiss network). Without any detail it is not possible to assess properly the class learnings.