CS458 A1 - Written Responses

Mrugank Upadhyay
m4upadhy
20843597

1. **(6 marks) For each of the reported security breaches, please 1) identify which one(s) of the security CIA properties is (are) compromised, and 2) describe a possible attack approach leading to the breach.**
(Note:: There are various security flaws in Cryptopia's cyber infrastructure.)

(a) Cryptopia's internet-connected and smart production machines were functioning normally since their deployment; However, on the day Cryptopia's executives were meeting with an important investor for checking the facility, the employees found they lost control over all the machines.

The CIA property that is compromised is availability as the employees no longer have access to the machines. A possible attack vector for this case could be the use of an IoT botnet since these devices are connected to the internet, but rarely have much security built-in due to their (typical) lack of usefulness for bad actors considering they lack resources on board. Poor network security can also make it very easy to expose such devices and allow a bad actor to gain control. DOSing can be made easier if the network has no capability to rate limit.

(b) Cryptopia's marketing leader lost a bid for an important transaction to a competing company. An internal investigation shows that the bidding details were leaked when the marketing leader was uploading them to the auction portal.

The CIA property that is compromised is confidentiality since the bidding details are private to the company itself and should not be shown to any external parties. One possible attack vector for this case could be the use of a Man-in-the-Middle attack. It can specifically be done by phishing attacks where a bad actor has replicated the auction portal, and the marketing leader simply thinks they're on the correct site. If this particular type of attack is used, it can be argued that integrity is also violated as the user here is not receiving the correct data they expect (the correct webpage).

(c) The company's logs revealed that a user, with username "surprise", had tampered with the configuration of the production system to cause it to slow down. Further investigation revealed that user "surprise" had no password.

The CIA property that is compromised is confidentiality since the user "surprise" is not an authorized user that is given permission to view/modify the configuration data. One possible attack vector for this case would be the use of privilege escalation. In this case, a low level user can be created if the company doesn't have a proper user identity and access management system, and this low level user can then be elevated to have increases privileges (doesn't necessarily have to be root). One method could be via finding an unintentional flaw in a production environment entity and using it to elevate the permission of the "surprise" user, perhaps via a buffer overflow to setuid programs.

2. (6 marks) Cryptopia did not have a formalized way to review, test, and maintain the codebases powering its production systems. You have been tasked with formalizing the process to ensure proper security controls. Please state and explain a security control you would recommend during each of the following phases of development: Code Review/Testing/-Maintenance.

Code Review:

For code review, a security control that I'd recommend would be to use easter egg code reviews. This is primarily due to the fact that Cryptopia's security infrastructure is already poorly developed, and the added incentive to find hidden or difficult-to-find bugs with easter eggs would significantly aid in not only resolving future bugs but also finding past bugs that have slipped through.

Testing:

For testing, a security control that I'd recommend would be the use of fuzz testing to ensure that the CIA triad is not compromised. Fuzz testing will allow for more practical testing in terms of security as that is typically how bad actors will try to gain knowledge of the inner workings of a system (e.g. reviewing core dumps via string formatting vulnerability, etc). One additional option would then be to add white-box testing to ensure increased robustness of the software and infrastructure.

Maintenance:

For maintenance, a security control that I'd recommend would be the creation of standards, and implementation of processes and audits for those standards. These standards can include anything ranging from design philosophies/methodologies, to change management techniques. Especially with cybersecurity, correctly managing change is crucial in order to ensure that vulnerabilities are not introduced on an ex post facto basis. For example, if a new function is added to fix a bug, and this function receives a string, it is easy to introduce a string formatting vulnerability if the string provided can be controlled by a bad actor. This can be difficult to test and maintain if changes are not managed correctly (for example, with a change advisory board). The creation of processes and audits then ensures that the standards are adhered to, and improved over time.

3. (8 marks) After identifying the attack methods, you have been asked to reinforce the defence mechanisms. For each defence method studied in class: prevent, deter, deflect, detect, recover, discuss how they could be used to defend against any one of the attack approaches mentioned in question (1). More specifically, explain how the defence method would apply in the narrative context of the attack, and why it would help against it. (Note: You only need to explain the defences for one attack).

Example:

Prevent. In case of a Man-in-the-middle attack, encrypt bidding data sent to the auction portal. Encrypted bidding data would be of no use to the attacker.

(a) Deflect

In the case of privilege escalation attacks, one way to deflect the bad actor would be through the use of honeypots. One can create a legitimate environment with intentional security flaws to make it more attractive to bad actors due to the ease with which they can be targeted. In context, a fake production environment can be used, and when attacked can help cybersecurity specialists better understand how the attack was performed, and search and fix similar vulnerabilities in the real environment.

(b) Detect

Detection can be applied to privilege escalation attacks by not only looking for signs after the escalation has occurred but by trying to find unusual activity before escalation is done. In this case, a bad actor can be trying to find the inner workings of different programs for the production environment, and so repeated executions of programs, especially those that tend not to run often or aren't automated can be an indication of such an attack going on. Detecting privilege escalation attacks can significantly increase the reaction time to stopping and recovering from them. If escalation hasn't happened yet, detection can help find our if a particular user is compromised (if an attacker is trying to use another user simply for recon), or what the objectives and approaches the attacker has/takes.

(c) Recover

In the case of privilege escalation attacks, in this context, one way to recover from them is to delete the offending user, "surprise". Additionally, purging the modifications would allow the data to be restored, if backups were kept (or via a version control system). In this case, since it was simply a configuration file that was modified, rolling back to a previous version would help reduce the alleviate the issues caused by the attack.

(d) Deter

Detering privilege escalation can be done via the use of identity and access management systems. In this case, only allowing certain users/roles to have access to production environments and their associated software reduces the number of people that can be compromised (if the attacker uses a user with moderate access as a starting point), and makes it harder for a bad actor to access associated programs in the first place. This difficulty comes in part because more critical systems can be gated by multiple authentication requirements. For example, for production environments, having automated services is typically the norm, however, if the generic user accounts for services aren't managed properly, they can become an opening point for attackers. But if these are isolated and only employees have access to make certain changes (e.g. configuration settings) and they must be made manually, it narrows the options an attacker can leverage to perform the attack and also disincentivizes an attacker.

4. (6 marks) The IT department in Cryptopia proposed a list of custom two-factor authentication schemes that protect access to the production mobile controllers (company-owned smartphones used to control the machines). You have been asked to review these proposals. Indicate whether you would accept or reject the proposals below and explain the reason(s) behind your decision. If you reject a proposal, propose an alternative.

(a) The scheme unlocks a controller if a correct password and a correct PIN are entered.

This proposal should be rejected. The problem with this proposal is that it approaches two factor authentication, but uses the same class of authentication, in this case it's something the user knows. The issue here lies in the fact that both of these can be easily found out by a bad actor, perhaps via a social engineering attack, and thus don't provide a strong way of proving identity. It's a good start with using two factors, so a possible change could be to use two different classes, such as something the user knows (password) and something the user has (perhaps a hardware key such as a yubikey). This is much more difficult for a bad actor to overcome as they would now need to somehow physically steal the yubikey, or find a way to replicate the output of the hardware device.

(b) The scheme unlocks a controllers if a correct password is entered or a correct number (received via email) is entered.

This proposal can be accepted. It employes two factor authentication with two different classes: something the user knows (password/number), and something the user has (access to their email). The extra security comes from the fact that it is difficult to obtain the number if the bad actor doesn't have access to the user's email, or a way to intercept it. This can be strengthened if multifactor authentication is employed and gives the user options to choose different methods since this way the bad actor can't know which method will be used at that time to authenticate the user.

(c) The scheme unlocks a controller if the user enters a correct password exclusively within the company's premises.

This proposal can be accepted. Since the authentication is to access the company smartphones, it requires them to have something they know (password), as well as have some information about the user's context (location). When the location check is done correctly, this is a very strong method of authentication as it builds upon many other authentication systems. In order to gain access to the company premises (assuming it means they need to be **inside** a building), they probably need to have a badge, or some other authentication factor to even enter the building. This means that not only are different classes of authentication employed, they're stacked on top, decreasing the chances of giving bad actors access to the device.

5. (4 marks) Identify the type of the following pieces of malware – i.e., whether the malware is a worm, Trojan, Ransomware, and/or Logic Bomb. Give a brief description of how it spreads or how a computer becomes infected, and the resulting effect. (A malware may be classified into more than one type.)

(a) WannaCry

The WannaCry malware was ransomware. Typically, ransomware spreads via malicious files that are downloaded or obtained through attachments, mostly without the user's knowledge and in a similar fashion WannaCry was spread using a vulnerability in the Windows SMB (which allows Windows computers to share files) protocol. The result of this ransomware was the encryption of important files in a user's system and holding them ransom until a payment was made to decrypt them. WannaCry had a widespread impact, even after a patch from Microsoft as many systems had not update to the patch in time.

(b) Code Red

Code Red malware was a worm. Typically, worms are spread via security flaws in large systems where the offending worm begins to search for other entities to infect within a particular network. It can sometimes contain a payload that is triggered through different means. In the case of Code Red, a buffer overflow was exploited in Microsoft's IIS web server, and this allowed the worm to then continue infecting other web servers (even those that were not IIS). The result of this malware was the infection of more than 300,000 servers, in which it installed back doors onto systems and altered the home page to read a message. On certain systems, DOS attacks were launched as well.

Sploit Explanations

Sploit 1

For sploit1, the vulnerability use was a buffer overflow vulnerability when copying the filename for the seed file ("-e" option) into args.filename within the parse_args function. We see that pwgen uses strncpy, which is a very good choice for copying a string as it only copies a STR_SZ amount of data. The issue here, however, is that the buffer size for args.filename is supposed to only be 1024 bytes, however, the amount used is BUFF_SZ which is actually 2048. Due to this discrepancy, sploit1 takes advantage of the file name option to instead inject the shellcode that is defined above. The construction of the code is similar to what is shown in smashing the stack for fun and profit, where we pad the beginning of the buffer with NOPs and then add our shellcode. Then we get the address for the beginning of the buffer and append copies at the end so that it overwrites the return address after overflowing and points back to the beginning of the buffer. The NOPs will execute until we get to the shellcode that needs to be run, and as pwgen is run as root, the shell created will also be root. The injection mechanism here is via the -e option and using an environment variable to store and use buffer.

Considering this is a buffer overflow exploit, to specifically alter the pwgen code, there is one modification that should be made. The modification is to fix the mistake of an incorrect buffer size, by defining a new size for filenames (since all of the other filenames are also 1024 in the code, for consistency's sake). This will ensure that there are no magic numbers which can be accidently altered, and the incorrect size won't be used. Next is the issue of confirming a legitimate filename.

Sploit2

For sploit2, we employ a format string vulnerability to gain access to a root shell. This vulnerability is found within the print_usage function where it writes the program name (arg0) into the usage string using %s, and similar to sploit1, snprintf uses a buffer size of 2048 into a string buffer of only 1024. The exploit takes advantage of this by overflowing the string with our shellcode instead. This is done in a similar fashion to sploit1, with an alignment alteration for the address pointer (+5 instead of +2). We once again pad the front with NOPs and our realigned address after our shellcode. The injection mechanism is different, however, as previously we used the -e option to inject the payload from an environment variable, for this method we will instead rename arg0. Renaming arg0 can only be done by using the exec commands, and in this case, we use execvp to run pwgen for its first parameter, and pass an argument list which contains the payload as arg0, and the -h option gets pwgen to run the print_usage function.

Similar to sploit1, we can fix this issue by ensuring that the buffer size used is the same as the size of the buffer being written to. One way to ensure this is to simply create a buffer_sz variable within the function, and reference it instead of using a global definition since not all buffers will be of size 2048.