The questions shown below belong to previous iterations of the course. We include the answers exactly as they were. While these quizzes can help you for preparing yourself, we will not discuss the answers nor argue about their correctness. They are provided as-is. Thank you for your understanding.

1 Intro/Cyber Threats

1. Which of the following threats is an example of a *commodity threat*? (2020)
   - ◯ Mass-mailing e-mails falsely pretending to have hacked the recipient and asking for money.
   - ◯ Hacking web servers to display political messages.
   - ◯ Exploiting an SQL injection on an e-banking web site.
   - ◯ Exploiting a buffer overflow of the *ping* command to become administrator of a machine.
2. Which of the following statements is true about commodity threats? (2022)
   - ◯ Their main form of cyber-attacks is Denial-of-Service (DoS)
   - ◯ They target organizations such as nations and large-scale international businesses
   - ◯ They are based on ready-to-use tools with little or no customization
   - ◯ They are politically motivated

2 Crypto Basics

3. Which of the following statements is true? (2020)
   - ◯ Symmetric crypto algorithms do not solve the problem of key exchange.
   - ◯ A message authentication code is an asymmetric algorithm.
   - ◯ Symmetric crypto algorithms are usually much slower than asymmetric ones
   - ◯ All symmetric encryption algorithms are block ciphers.
4. Which of the following statements is false regarding cryptographic hash functions? (2022)
   - ◯ A cryptographic hash function is deterministic
   - ◯ It is infeasible to find any two messages with the same hash value
   - ◯ For any given message, it is fast to compute its hash
   - ◯ It is feasible to find a message that yields the same hash value as that of another given message
5. HSTS (HTTP Strict Transport Security) implements what protection? (2022)
   - ◯ Prevents use of insecure/deprecated cipher suites when connecting using TLS
   - ◯ Automatically changes all URLs from http://website.com to https://website.com
   - ◯ Prevents cross-domain cookie leakage attacks
   - ◯ Prevents MitM downgrade attackers from presenting a HTTPS website as HTTP
   - ◯ Validates the server config by checking the config file for vulnerabilities
   - ◯ Uses AES-256 encryption to provide confidentiality for a connection to a server
6. Which of the following techniques can help an attacker perform a MitM attack and sniff packets on a LAN network? (2022)
   - ◯ Acting as a DHCP server and replying with configurations including a attacker-controlled DNS server
   - ◯ Sending fake reverse-ARP requests to spoof the gateway

- ◯ Creating a fake website (e.g., faceb00k.com instead of facebook.com)
- ◯ Enabling HSTS
- ◯ All of the above

7. In the context of a TLS handshake: (2022)
   - ◯ The server provides a list of cipher suites and the client chooses one
   - ◯ The Server Hello message contains the server's certificate along with the intermediate and root CA certificates
   - ◯ The server signs its ephemeral Diffie-Hellman public key with its long-term private key
   - ◯ The client is always authenticated by a certificate

3 Access Control

8. In Linux, an user (non-root) wants to list all the files he or she can execute. To do it, the user has to: (2022)
   - ◯ Read the Linux index that lists all the files a subject (the user) can execute
   - ◯ Check the execution access rights for every accessible file in the filesystem
   - ◯ The user can do any of the two actions stated above
   - ◯ The user cannot list the files he or she can execute because the access control system does not allow normal users to check executable access to files
9. When does HMAC not protect against cookie tampering? (2022)
   - ◯ If the private key (one-time pad/password) is leaked
   - ◯ If the cookie session expires
   - ◯ If the cookie is forged for a session
   - ◯ If the cookie pool overflowed and is rewritten when accessing a privileged resource
10. Which statement is FALSE regarding the Kerberos protocol? (2022)
    - ◯ An authenticator contains the client's identifier and a timestamp encrypted under the session key with the destination server
    - ◯ The Kerberos protocol relies on securely distributed symmetric keys
    - ◯ The Ticket Granting Server (TGS) provides the Ticket Granting Ticket (TGT) to the client
    - ◯ With pre-authentication, a client proves to the Authentication Server (AS) that it knows the password by generating an authenticator
11. You are "user", group "user". File "f1" has permissions `-rwxr-sr--` and is owned by "root", group "root". (2020)
    - ◯ "f1" currently has the setuid bit set **and** you can execute "f1".
    - ◯ "f1" currently has the setuid bit set **and** you cannot execute "f1".
    - ◯ "f1" currently does not have the setuid bit set **but** if it was set, you could execute "f1".
    - ◯ "f1" currently does not have the setuid bit set **but** if it was set, you could not execute "f1".
12. To prevent malicious modifications to a cookie, a website can use the following techniques: (2020)
    - ◯ Sign the cookie
    - ◯ Encrypt the cookie with a key only known to the website
    - ◯ All of the above answer

- ◯ None of the above answers

13. Client-side verification of a password, when used to authenticate a client to a remote service: (2020)
    - ◯ Is never secure if done in software
    - ◯ Is never secure in Javascript, but would be secure in a compiled language
    - ◯ Can be secure in Javascript if using an appropriate one-time-pad
    - ◯ Can be secure in Javascript if the remote service signs the Javascript source
14. If your device is in position to man-in-the-middle a connection between a client and a server, what is *not* doable: (2020)
    - ◯ Block altogether TLS connections
    - ◯ Replace the answer of a DNS query with an another IP address
    - ◯ Replace elements of a webpage served over HTTP
    - ◯ Replace elements of a webpage served over HTTPS
15. A *secure* Nginx configuration should: (2020)
    - ◯ Also serve the website over HTTP (*e.g.*, for legacy clients)
    - ◯ Have a certificate (to provide TLS) signed by any CA authority
    - ◯ **Not** send the HSTS header
    - ◯ Have a certificate (to provide TLS) signed by the server of hw4ex3 (Nginx server)

4 Data Security

16. Assume a web service using a database to store user data and a malicious actor with access to the database host. Due to a misconfiguration, the malicious actor has access to an unprivileged user account on that host that can read any file on the filesystem. Which of the following measures can protect the confidentiality of the data? (2022)
    - ◯ Filesystem encryption
    - ◯ Encryption at the web service layer
    - ◯ Encryption at the database layer
    - ◯ All of the above
17. Assume a web service is using a database to store and retrieve user data. Which of the following are **true** if the data is encrypted at the database layer? (2022)
    - ◯ The root user of the database host cannot read plaintext user data
    - ◯ The data is protected against physical theft, i.e., someone breaking into the server room and stealing the hard disk
    - ◯ The data is protected against software theft, i.e., someone taking full control of the web service and reading data from the database
    - ◯ The communication channel between the web service and the database needs to be protected separately
18. Which type of password is less likely to be cracked by a dictionary attack similar to the one done in Homework 4? (2022)
    - ◯ A password made of ten digits numbers (e.g., 8234561232)
    - ◯ A password made of the latin names of plants (e.g, chamaenerion-angustifolium-sequoiadendron)

- ◯ A password made of the name of three cities in Asia (Ch'angch'un-Pleiku-Suphanburi)
- ◯ A password made of the name of three fruits (apple-cherry-tomato)

19. Which one of these statements is TRUE? (2020)
    - ◯ Salt makes the use of a lookup table difficult, i.e., far less efficient.
    - ◯ Salt is always required to prevent brute force attacks
    - ◯ Salt needs to be placed at the end of a password to be efficient
    - ◯ Dictionary attack with salt reduces to a brute force attack on a password of length: $len(password)+2$

20. Which of the following inputs is likely to return results from an SQL injection for a parameter similar to HW 1.1? (2022)
    - ◯ – AND 1 = 1
    - ◯ ‘ AND 1 = 1
    - ◯ ‘ AND 1 = 1 ‘ –
    - ◯ ‘ AND ‘1’ = ‘1

21. You input the name of your cousin, James O'Brien, in a form on a website. Which behavior might indicate the presence of an SQL injection on this website? (2022)
    - ◯ My input returns the name as O%27Brien
    - ◯ My input returns valid data
    - ◯ My input returns a generic server error

22. What is the purpose of the UNION operator in an SQL injection? (2022)
    - ◯ Make the WHEN condition always true
    - ◯ Create a prepared SQL query
    - ◯ To execute an additional arbitrary SQL query
    - ◯ To leak the layout of the database

23. In a Homework you implemented the Secure Remote Password (SRP), an augmented Password-Authenticated Key Exchange (PAKE) protocol. Which of the following statements is TRUE? (Assume that the cryptographic primitives are securely instantiated following best practices.) (2022)
    - ◯ An eavesdropper does not learn information that helps to guess the password
    - ◯ If the server gets hacked, the password hash is not revealed
    - ◯ The resulting key agreed by the 2 parties depends on the client's password
    - ◯ All previous answers are true

24. Which one of these statements is *false*? (2020)
    - ◯ A dictionary attack is only as good as the dictionary used
    - ◯ A dictionary attack is usually slower than a ed brute force attack
    - ◯ A dictionary attack is equivalent to a brute force attack on a smaller keyspace
    - ◯ A dictionary attack typically reduces the search space

25. Which one of these statements is *true*? (2020)
    - ◯ Salt is always required to prevent brute force attacks
    - ◯ Salt needs to be placed at the end of a password to be efficient

- ◯ Dictionary attack with salt reduces to a brute force attack on a password of length LEN(password) +2
- ◯ Salt makes the use of a lookup table difficult, i.e., far less efficient.

26. Consider a password of length N, and a dictionary of size D. Which one of these statements is *true*? Compared to a brute-force attack, the search time using this dictionary... (2020)
    - ◯ is reduced by a factor $(N \cdot D)$
    - ◯ is reduced by a factor (N/D)
    - ◯ is reduced by a factor (D/N)
    - ◯ is unrelated to N

5 PL security (compartments)

27. Static analysis (SA) has which of the following properties (choose all that apply): (2022)
    - ◯ Completeness: SA checks correctness in all cases
    - ◯ Incompleteness: SA does not assure correctness in all cases
    - ◯ Soundness: Failed SA checks always indicate actual bugs
    - ◯ Unsoundness: Failed SA checks can be false positives
28. What does a STRICT STATIC TYPE SYSTEM do that a STRICT DYNAMIC TYPE SYSTEM does not? (2022)
    - ◯ Finds all places where a variable is assigned the wrong type of value
    - ◯ Raises an error if a program performs an out-of-bounds memory reference
    - ◯ Requires type declarations on all program variables
    - ◯ Prevents a developer from writing functions that accept more than one type of argument
29. Which of the following statements are correct arguments why a company should or should not rewrite all of its C/C++ code in a safe language such as Java or Rust (choose all that apply)? (2022)
    - ◯ It would eliminate buffer overrun errors
    - ◯ The cost of a complete rewrite is likely to be much larger than the benefits
    - ◯ The new programs will still have security vulnerabilities
    - ◯ The new programs will run slower
30. Which of the following statements about automatic storage reclamation is correct? (2022)
    - ◯ Using malloc/free ensures a program runs much faster than using automatic techniques
    - ◯ Garbage collection cannot reclaim doubly linked lists because of cycles
    - ◯ Rust's ownership and borrowing performs all storage reclamation in Rust programs
    - ◯ Memory safety requires some form of automatic storage reclamation

6 OWASP and buffer overflows

31. Consider the following snippet of C code: (2022)

```
#include <stdio.h>
#include <stdlib.h>
#include <unistd.h>
int main(int argc, char *argv[]) {
    const char target[] = "/bin/sh";
    char *const args[] = {target, NULL};
    execve(target, args, NULL);
    /* execve(const char *path, char *const argv[], char *const envp[]) */
   return 0;
}
```

Which of the following statements is true?

- ◯ The executable crashes because of a stack buffer overflow
- ◯ If the executable is owned by root and the suid bit is set, the binary spawns a root shell
- ◯ If the executable is owned by root and the suid bit is set, the binary spawns a user shell
- ◯ If the executable is owned by root and the suid bit is set, the binary spawns a root shell only if the LD_LIBRARY_PATH environment variable has not been set
- ◯ The execve call does not do anything because the envp argument has been set to NULL and the executable thus simply exits

32. Consider the following snippet of C code:

```
void foo(char* argv[]) {
    char buf[4];
    strcpy(buf, argv[1]);
    /* strcpy(out, in) */
}
```

You are running it on a 32-bit x86 machine, the return address is 0x08049806. If you input "012345678" (the quotation marks are not part of the input), what address will foo return to (assuming the compiler does not add any padding)? (2022)

- ◯ 0x38049806
- ◯ 0x38009806
- ◯ 0x08049838
- ◯ 0x08040038

33. Which type of variables in C has a chance of causing a stack overflow? (2022)

- ◯ Global variables
- ◯ Local variables
- ◯ Static variables
- ◯ Constant variables

34. What could be the reason a stack canary always ends with a null byte? (2022)

- ◯ It is an optimization to save time when checking its integrity
- ◯ It prevents leakage through string-based functions (e.g. printf)
- ◯ It prevents too much usage of system entropy by randomizing only the necessary bytes
- ◯ To make sure the program crashes if the canary is erroneously used as return address

35. In your exploits, you probably added a `NOP` sled in the attack buffer. Which of the following is true? (2020)

- ◯ Using a NOP sled, the attacker needs to approximately but not precisely guess the address of the shellcode
- ◯ Adding NOPs is necessary to overflow a buffer
- ◯ To execute the shellcode, the NOP sled can be placed either before or after the shellcode, as long as the control flow jumps somewhere in the NOP sled
- ◯ NOP sleds bypass ASLR (address space layout randomization)

36. You are targeting the C function `foo`, defined as follows: (2020)

```
void foo(char* argv[]) {
  char buf[240];
  strcpy(buf, argv[1]); /* strcpy(out, in) */
}
```

You are running this program in a debugger on a 32-bit machine and have put the breakpoint at the end of this function. This is the (simplified) output of the `info register` command, and the `print` command:

```
> info register
esp 4800
ebp 5064
eip 72

> print &buf;
$1 = (char (*)[240]) 4824
```

(Addresses are shortened and converted to decimal for your convenience). If the shellcode takes 45 bytes, which of the following ranges of addresses could contain a NOP sled?

- ◯ 4800–5060
- ◯ 4824–5105
- ◯ 4824–5018
- ◯ 5019–5064

37. The size of both `int` and `unsigned long` types is 4 bytes; `int` is signed. Consider the following snippet of C code: (2020)

```
char *p;
int size = (int)strtoul(argv[1], &p, 10);
```

Which numeric value of the command-line argument (`argv[1]`) will result in the `count` variable containing a negative number?

- ◯ $2^{31} + 1$
- ◯ $2^{31} - 1$
- ◯ $2^{15} - 1$
- ◯ $2^{15} + 1$

7 Fuzzing

38. Which type of sanitizer is most likely to detect a buffer overflow? (2022)
    - ◯ ThreadSanitizer
    - ◯ AddressSanitizer
    - ◯ MemorySanitizer
    - ◯ UndefinedBehaviorSanitizer
39. Testing (unit testing/regression testing) has which of the following properties (choose all that apply): (2022)
    - ◯ Completeness: Testing checks correctness in all cases
    - ◯ Incompleteness: Testing does not assure correctness in all cases
    - ◯ Soundness: Testing failures generally indicate actual bugs
    - ◯ Unsoundness: Testing failures can be false positives
40. Which of the following is true for mutation-based input generation for fuzzing? (2022)
    - ◯ Specification is needed for input structure
    - ◯ Mutations are applied on a binary input blob
    - ◯ Application-specific operations must be applied, making input generation slow
    - ◯ Can only use purely random mutations for generating inputs
41. Which of the following is/are true: (2022)
    - ◯ Fuzzing makes static analysis obsolete, which is good since static analysis has too many false positives
    - ◯ A fuzzer can only find bugs in program regions covered during fuzzing
    - ◯ A fuzzer can satisfy some hard control-flow and data-flow restrictions (e.g., magic-byte matching)
    - ◯ A fuzzer running a sanitized program will detect all bugs in the code regions executed
    - ◯ Fuzzing eliminates manual efforts for bug identification

8 Network security and Operational security

42. Which statement is FALSE about intrusion detection systems (IDS)? (2022)
    - ◯ IDS introduce operational costs as they may generate a lot of false positives
    - ◯ Signature-based IDS can detect new unknown network attacks
    - ◯ Anomaly-based IDS are calibrated with benign network traffic
    - ◯ A matched signature in a signature-based IDS does not necessarily mean that the system is under attack

9 Mobile Security

43. Given an Android app written in Java, which of the following statements are true regarding the app's Dalvik bytecodes? (2022)
    - ◯ The bytecodes can be disassembled to Smali

- ◯ The bytecodes contains the original variable names and developer comments
- ◯ The bytecodes can be executed on the Java virtual machine
- ◯ The bytecodes can (often) be decompiled to Java

44. Which of the following are true about Android-based smartphones? (2022)
    - ◯ Any system software running on an Android smartphone is fully open source thanks to the AOSP (Android Open Source Project)
    - ◯ You can easily (in comparison to iOS devices) install a different/custom OS version
    - ◯ Android apps are written in Java/Kotlin and attackers consequently also need to develop their malware in Java/Kotlin to deploy it on Android smartphones
    - ◯ Each application is executed in a sandbox, effectively preventing it from influencing other app's execution

10 TEEs and Side channels

45. Suppose that a website implements password checking using the function shown below. Note that the server developer fixed the issue in the homework, where the server returned as soon as the first different character was found, instead ensuring that all characters of the guess are checked (for guesses of the correct length).

    Can this function be used to leak the correct password efficiently using a timing side-channel, and why? Efficiently means that the number of attempts increases linearly with the size of the password. Choose all relevant options (for why).

```
check_password(guess, pass):

        if(len(guess) != len(pass))
                return false

        correct = true
        for i in range(len(pass)):
                if guess[i] != pass[i]
                        correct = false
                        log("Incorrect password at time {}", time())

        return correct
```

    (2022)
    - ◯ No, because the developer correctly fixed the timing behavior by checking all characters
    - ◯ No, because the function logs all incorrect guesses, and can detect an active attacker
    - ◯ Yes, because the check fails when the attacker's guess does not have the same length as the password, creating a timing channel for guessing the correct length
    - ◯ Yes, because the 'log' function is called for every character that is different, making the runtime quicker for every character guessed correctly
    - ◯ No, because the password can be hashed with a unique salt
    - ◯ Yes, because the 'len' function takes time proportional to the length of the password
    - ◯ Guessing the password is possible but it is not efficient

46. If an attacker rewrites the BIOS boot block, is it still trustworthy to use the TPM? (2022)

- ◯ Yes
- ◯ No, the posterior TPM_Extend operations may produce arbitrary results
- ◯ No, the attacker can disable the interface to operate the TPM remotely
- ◯ No, the BIOS always performs a TPM_PCRRead over the boot lock code

47. Which of the following is NOT a weakness when choosing a security solution based on trusted execution environment as compared to a solution based on software (e.g., homomorphic encryption or secure multi-party computation)? (2020)
    - ◯ Risk of side-channel attacks
    - ◯ Lock-in by a single vendor
    - ◯ Difficulty/cost of system retrofitting in case of identified vulnerabilities
    - ◯ Computation performance

11 Privacy

48. You are going to anonymize and publish the following medical data set for research purposes: (2022)

Identifier Name	Quasi-Identifier Age	Sensitive Attribute Zip Code	Problem
Alice	25	1024	Flu
Bob	31	1025	Covid
Claire	29	1024	Flu
David	39	1025	Headache
Elia	33	1025	Flu

    How can you achieve 2-anonymity without over-reducing the information released?
    - ◯ Remove the "Zip Code" column
    - ◯ Remove the "Age" column
    - ◯ Remove the "Problem" column
    - ◯ Map the actual age to range (20, 30] and (30, 40] accordingly

49. Recall the *averaging attack* against the Laplace mechanism: an adversary sends multiple semantically equivalent queries and averages the responses to obtain an estimate of the actual response without the Laplace noise. Which of the following is true? (2020)
    - ◯ Adding the same noise value to queries that are equivalent WILL NOT prevent the attack
    - ◯ Giving no response if an equivalent query was asked before WILL NOT prevent the attack
    - ◯ Drawing fresh Laplace noise for every query WILL NOT prevent the attack
    - ◯ Any of these options will prevent this attack

50. Suppose there are k independent differentially-private mechanisms. Each of these mechanisms ensures ϵ_i differential privacy, where 1 ¡= i ¡= k. Suppose we design a function f, which internally queries the k independent mechanisms defined above. Then f is: (2021)
    - ◯ $sum_{i=1}^{k}\epsilon_i$ differentially private
    - ◯ $sum_{i=1}^{k}(\epsilon_i)^2$ differentially private
    - ◯ $max(\epsilon_1, \ldots, \epsilon_k)$ differentially private
    - ◯ $min(\epsilon_1, \ldots, \epsilon_k)$ differentially private

51. Consider the differentially private query interface for your database. An analyst has been allotted with a total privacy budget of B. They successfully made a query with ϵ_1 on the first day, and another query with ϵ_2 on the second day. What is the output of REMAINING_BUDGET method after the second day? (2020)

- ◯ $B - \epsilon_1 - \epsilon_2$
- ◯ $B + \epsilon_1 + \epsilon_2$
- ◯ $\epsilon_1 + \epsilon_2$
- ◯ $\epsilon_1 * \epsilon_2$

52. Which of the following statements is TRUE? (2021)

- ◯ Federated learning aims at concealing the identity of each data providerpp
- ◯ When differential privacy is used, federated learning completely prevents privacy attacks
- ◯ When differential privacy is used, federated learning is affected by the privacy vs. utility tradeoff
- ◯ Federated learning cannot be protected by cryptographic-based solutions

53. Let Enc, Dec be the Paillier encryption and decryption algorithms under some key pair and x and y be two integer plaintext messages. Which of the following equalities holds? (2020)

- ◯ $Enc(x) * Enc(y) = Enc(x + y)$
- ◯ $Dec(y * Enc(x)) = Dec(Enc(xy))$
- ◯ $Dec(Enc(x) + y) = Dec(Enc(x + y))$
- ◯ $DecryptionDec(Enc(x)Enc(y)) = x + y$

54. As in the homework, a service wants to homomorphically compute a simple linear regression $y = a*x+b$ on input x, given the client public-key pk and the ciphertext $ct = Enc(pk, x)$. How can the service compute $Enc(pk, y)$? (2021)

- ◯ $ct^a * Enc(pk, b)$
- ◯ $ct^E nc(pk, a) * Enc(pk, b)$
- ◯ $ct * a + Enc(pk, b)$
- ◯ $ct^{Enc(pk,a)} + Enc(pk, b)$

55. A company has implemented 5-anonymity in its database. Select the correct answer: (2022)

- ◯ An adversary can de-anonymize 5 data entries by knowing all the other de-anonymized database data entries beforehand (e.g., if the database has 1000 data entries, the adversary can de-anonymize 5 values by knowing the other 995)
- ◯ If 5 database data entries have the same QID and attributes, the lack of diversity of the sensitive attributes leaks information
- ◯ Similar entries will be considered as the same data entry when anonymizing, (i.e., two data entries with same QID and attributes are counted as a single one in the final anonymized set, so the set should have at least other 4 data entries)
- ◯ The database implements homomorphic encryption to protect against leaks

56. Alice uses a smartwatch made by the company BoBWatch that tracks her calories burnt daily. Her smartwatch is synchronized with an Android app that queries BobWatch's database for population statistics, which show Alice that she burns more calories than 80% of the people of her age and sex in her area. What protection mechanism must BoBWatch implement in order to prevent Alice from inferring the calories burnt by her flatmate of the same age/sex, Eve, who also wears a BoBWatch? (2022)

◯ Homomorphic encryption of BoBWatch's database

◯ Adversarial machine learning on Alice's BoBWatch

◯ Adversarial machine learning on Eve's BoBWatch

◯ Differential privacy for queries on BoBWatch's database E) BoBWatch must run the server processing user data in a Trusted Execution Environment (TEE)

12 ML security and privacy

57. We define the adversarial budget "ϵ" as the max allowed p-norm of the perturbation: s.t. $||\sigma||p < \epsilon$, where σ is the perturbation and $||\sigma||p = (\sum(|\sigma * i|^p))(1/p)$, i is the pixel index of σ. Which of the following statements is FALSE about the adversarial budget for manipulating an 8-bit (pixel values vary from 0 to 255) gray-scale natural image? (2021)

    ◯ Adding a small value, e.g., 1, to all pixels costs a small budget if p=infinity

    ◯ Changing one pixel value from 0 to 255 costs a small budget if p=infinity

    ◯ Affine-transform does not always cost a small budget when p = 2

    ◯ Changing one pixel value from 0 to 255 costs a large budget if p=infinity

58. In Machine Learning, researchers observed that when we train a pre-trained model (on task A) on a different task (task B), the model might forget task A, e.g., performing as badly as untrained models. This is called catastrophic forgetting.Which of the following statements about catastrophic forgetting and privacy-preserving machine learning is FALSE? (2021)

    ◯ Catastrophic forgetting harms the availability of the Machine Learning model on task A

    ◯ Catastrophic forgetting might make it harder to infer task A-relevant privacy information

    ◯ Overfitting is a common enemy of privacy and utility in privacy-preserving machine learning

    ◯ Catastrophic forgetting fulfills the two goals of privacy-preserving machine learning

59. A prediction-as-a-service platform deployed a non-linear regression model as below: $y = w2(\sigma(w1 * x1 + w2 * x2 + b1)) + b2$ where $x1$ and $x2$ are two inputs and y is the output. $w1$ and $w2$ are two weights and $b1$ is the bias. σ is non-linearity where $\sigma(x) = x$, for $x > 0$, and $\sigma(x) = 0$, for $x <= 0$. Given $b1 < 0$, how many queries do you need at most to determine $b2$? (2022)

    ◯ 0

    ◯ 1

    ◯ 2

    ◯ 3

    ◯ 4

    ◯ 5

60. Which one of the following statements is true? (2020)

    ◯ Federated learning preserves the confidentiality of the training data of participants.

    ◯ When combined with homomorphic encryption, federated learning cannot preserve the confidentiality of the intermediate values that are exchanged between the participants.

    ◯ Privacy-conscious data sharing is only possible via hardware-based solutions as there is no possible attack that is proven to be successful against secure hardware.

    ◯ In federated learning, differential privacy can degrade the utility of the training data.

61. Is it possible to create an ML system that detects and blocks network attacks with a 100% True Positive rate? (2022)

- ◯ Yes but it would not be usable at all
- ◯ Yes and likely to be usable
- ◯ No, as new attacks are not yet known
- ◯ No, as adversarial machine learning could find attacks not yet present in our rules

62. Consider an adversary performing an attack against a machine learning-as-a-service (MLaaS) system. What capabilities does the adversary have in a grey-box scenario? (2022)
    - ◯ The adversary knows the algorithm underlying the model
    - ◯ The adversary knows the parameters of the model (e.g., weights and biases)
    - ◯ The adversary has access to the training dataset
    - ◯ The adversary can run and inspect the trained model locally

63. Which one of the following statements about the security and privacy of Machine Learning is CORRECT? (2022)
    - ◯ For images, the p-norms are always consistent with human perception of image similarity
    - ◯ Applying a certified defense implies that a model is robust against all adversarial attacks
    - ◯ In federated learning, clients contribute their data to other clients while not directly sharing their data set
    - ◯ In model stealing, stealing a linear model with three-dimensional inputs (x has three dimensions) needs at least three input-output pairs

64. Which of the following is/are true about adversarial machine learning? (2022)
    - ◯ It is possible to create inputs between which the difference is imperceptible to humans, yet the machine learning model outputs wildly different predictions
    - ◯ It is possible to create inputs between which the difference is perceptible to humans without completely/to a very large part replacing the input in order to fool a machine learning model
    - ◯ Adversarial machine learning can be used to trick facial recognition software based on machine learning predictors
    - ◯ There exist sophisticated defenses against adversarial machine learning that are effective under all sorts of adversarial conditions
    - ◯ All of the above

65. Consider a prediction-as-a-service to which you send a query of x with the dimensionality d and the service outputs the prediction as $p = x'w$ where w is the model coefficient and the dimensionality of x' is d' with $d' < d$. Assume that you have the prior information on the dimensionality reduction algorithm, i.e., you know exactly how the server reduces d to d'. How many queries are needed to steal the model coefficients? (2021)
    - ◯ d
    - ◯ d'
    - ◯ d'+1
    - ◯ d+1

66. Combining decentralized datasets to train a machine learning model while protecting privacy is doable in various ways. Which of the following techniques is *not* appropriate to achieve this goal? (2020)
    - ◯ Add noise to partial aggregates.
    - ◯ Send partial aggregates to a trusted execution environment.
    - ◯ Rely on a combination of homomorphic encryption and secure multiparty computation.
    - ◯ Combine partial aggregates with as many adversarial examples.

13 Answers

1. ANSWER: A
2. ANSWER: C
   Explanation: Commodity threats consist of non-targeted and automated attacks that aim at short-term financial gains. Typical examples of commodity threats are malware-infected spam, extorsion spam and malicious advertisements. Unlike hacktivism they are not politically motivated nor aim at service disruption (e.g., DoS). Finally, they are different from advanced persistent threats that are very targeted (e.g., to large-scale business, nations), sophisticated, with a priority on long-term goals.
3. ANSWER: A
4. ANSWER: D
   Explanation: A cryptographic hash function is a mathematical algorithm that takes as input a message of arbitrary length and produces a fixed-length output (i.e., the hash of the message). Informally, it has the following properties:
   - It is a one-way function or in other words it is practically infeasible to revert a hash back to the message that produced it.
   - It is practically infeasible to find a message that yields the same hash value as that of another given message (2nd pre-image resistance).
   - It is practically infeasible to find any two messages with the same hash value (collision resistance).
   - A hash function is a deterministic algorithm, i.e., the same message always results in the same hash (recall that we use it to ensure integrity of transmitted messages or password storage) and is by-design fast to compute.
5. ANSWER: D
   Explanation: Definition of HSTS basically
6. ANSWER: A, B
   Explanation: DHCP responses are not authenticated, and configuring an attacker-controlled DNS will allow the attacker to spoof the IP of any website with an attacker-controlled website that can MitM the actual server. Spoofing the gateway with reverse-ARP causes the target to send all packets to the attacker. By itself, making a fake website does not send the victim's traffic there. Additional steps such as phishing are required. HSTS is irrelevant, and as a defense does not help the attacker.
7. ANSWER: C
   Explanation: A TLS handshake is triggered by a client sending a 'ClientHello' message to the server; this message contains the list of cipher suites that the client supports (along with some random value the client generates for the session). The server responds with the 'ServerHello' message that contains its cipher suite choice (and the server's random value for the session). Subsequently, the server sends the 'Certificate' message that contains its certificate (along with the certificates of the intermediate and root Certificate Authorities (CAs). The 'ServerKeyExchange' message contains the server's ephemeral (or session) Diffie-Hellman public key which is signed with the server's long-term private key (along with some fresh information, e.g., session ids) to prove its identity (i.e., the server proves that it knows the private key behind the public key of the certificate). After the 'ServerHelloDone' message and if requested by the server, a client may be authenticated by sending its certificate.
8. ANSWER: B
   Explanation: The user can find these files (e.g., by using the find command). In this case, the Linux File System uses Discretionary Access Control (DAC) for the metadata (access rights). This data is stored with an Access Control List (ACL), meaning that the file itself states the access rights per user, group and everyone. Then the user has to check the rights for its user, the groups it belongs to and everyone to check if it can execute the file. There is no index in Linux for this ability (capabilities).

9. ANSWER: A
   Explanation: With HMAC, the attacker cannot arbitrarily generate a cookie. A one-time pad leak permits an attacker to recreate the cookies on his or her own. Any other answer either causes the server to provide a new cookie or to have an invalid cookie (as the private key is unknown).

10. ANSWER: C
    Explanation: Kerberos is an authentication and authorization protocol whose security relies on symmetric keys shared between the clients and the Authentication Server (AS) — i.e., the password, AS and the Ticket Granting Server (TGS), and TGS and the various services. To get access to a service, a client first obtains the ticket granting ticket (TGT) from the AS, which it then presents to the TGS in order to get the appropriate service ticket. In all client interactions, an authenticator containing its identifier and a timestamp is generated and encrypted with the session key of the destination server (e.g., TGS or the final server). When pre-authentication is used, the client presents to AS an authenticator encrypted using the hashed password (and salt/iterations).

11. ANSWER: D

12. ANSWER: C

13. ANSWER: A

14. ANSWER: D

15. ANSWER: B

16. ANSWER: B, C
    Explanation: Filesystem encryption is transparent to any user on the corresponding host, it does consequently not protect against the malicious actor with read access to the filesystem since the database writes plaintext data to the files. Both encrypting at the web service or the database layer leads to encrypted data being written by the database to files in the file system, which prevents the malicious actor from reading the plaintext user data.

17. ANSWER: B, D
    Explanation: The database host's root user can still read plaintext user data because they can dump the database process's memory and extract data before it is encrypted. It is true that the data is encrypted when it is stored on the filesystem, so only being able to access the database files is not sufficient to read the data. This also holds true for a thief stealing the hard disk even if they can mount the filesystem on the stolen disk and read all the files on it, the files backing the database's storage only contain encrypted data. The user data is thus protected against access to a thief stealing the physical hard disk. A malicious entity taking full control over the web application can issue arbitrary requests to the database and thus read out all of its data, which is decrypted by the database before being sent to the web service. As encryption only happens in the database, the data in motion between the web service and the database is not encrypted and needs to be taken care of separately (e.g., by using a TLS-encrypted channel).

18. ANSWER: A
    Explanation: The dictionary proposed in the homework is made of all the words in Wikipedia. It will contain the name of all fruits, cities in Asia, and chemical elements. It is, however, unlikely to contain every ten digits number.

19. ANSWER: A

20. ANSWER: D

21. ANSWER: C

22. ANSWER: C

23. ANSWER: D
    Explanation: Eavesdroppers can not learn anything from the values sent through the communication channel, in particular: they can not brute-force the password. The resulting key depends on a, b and x (x depends on the password). If the server gets hacked the password hash is not revealed as the server stores g^x.

24. ANSWER: B

25. ANSWER: D

26. ANSWER: D

27. ANSWER: B, D
    Explanation: Static analysis must conservatively approximate some aspects of dynamic behavior such as aliasing or type casting. For example, if an object is typecasting from (void *), static analysis might have no clue about the original type of the object. Static analysis tries to be conservative in order to account for incomplete knowledge (such as dynamic behavior), and consequently can flag errors which cannot occur in runtime. For example, a static analyzer might tag an unchecked array access as possible buffer overflow even if the actual program will never execute a path that produces an out-of-bound value.

28. ANSWER: A
    Explanation: A static type systems will find at compile time all places where the type of a variable is inconsistent with a value that it is assigned. A dynamic system will only find the incorrect assignments that are executed in a run of a program. Both type systems raise an error on an out-of-bounds memory reference. Neither type system forces type declarations on all variables in languages with type inference (e.g. Java v8 and greater). Neither prevents a developer from writing polymorphic functions that accept more than one type of argument.

29. ANSWER: A, B, C
    Explanation: A safe language will ensure that all buffer overruns are detected at runtime and an error is raised. The cost of rewriting a non-trivial program in a safe language is very large, both in terms of human effort and in incompatibilities between the old and new version. The benefits in terms of safety, except for the most critical applications, are likely to be smaller. And, the rewritten program will still have security vulnerabilities! However, there is no reason why the rewritten program should run appreciably slower than the C/C++ program if performance is a requirement and developers are careful.

30. ANSWER: D
    Explanation: Developers may feel that malloc/free give them more control over storage allocation and are not costly, but that is not always true since object lifetimes are complex and interactions between memory allocation and the memory system (caches and TLB) have performance effects. Automatic storage reclamation can perform comparably with manual storage reclamation and sometimes better. Reference counting, not garbage collection, cannot reclaim cyclic structures. Rust ownership and borrowing handles some storage reclamation, but the language still provides reference; counting for data structures. Safe languages include automatic storage reclamation since manual control introduces possible errors around the lifetime of objects such as double frees, uses-after-free, etc.

31. ANSWER: C
    Explanation: The suid bit only allows the executable to change the UID, it does not automatically change the ID on its own. A call to `setuid` is still required in the executable. In this case (and as seen in the homeworks), `setuid(0)` at the start of the `main` function would actually result in a root shell being spawned.

32. ANSWER: D
    Explanation: The input consists of nine ASCII characters and one '\0'. After copying the input to

buf[], four bytes will overflow to the stored value of ebp and the remaining two bytes (\x38 and \x00) will overflow the return address. The x86 processors use little-endian byte ordering so that the least significant byte will be stored at the lowest address. Given that the direction of overflowing is towards the high address, the two least significant bytes of the return address (\x06 and \x98) will be overflowed by \x38 and \x00 separately. Therefore, the address value will become 0x08040038.

33. ANSWER: B
    Explanation: Among all the four types of variables, only local variables are stored in the stack, the others are stored in the data segment of the memory

34. ANSWER: B
    Explanation: A printf-like function can be used to leak memory using a %s like format specifier. This will cause printf to print bytes from memory until a NULL byte is encountered, including possibly the stack canary. Leaking the stack canary can help the attacker craft a buffer-overflow payload that overwrites the stack canary with correct bytes. When the stack canary ends with a NULL byte (i.e. the LSB is NULL), the first byte of the canary in memory will cause printf to stop printing, thereby protecting the remaining bytes from leakage. Note that this works for a little-endian machine. On a big-endian machine, the MSB must be NULL.

35. ANSWER: A

36. ANSWER: C

37. ANSWER: A

38. ANSWER: B
    Explanation: ThreadSanitizer looks for races between threads, which are not directly linked to buffer overlfows. Similarly, UBSan is looking for undefined language behavior while MemorySanitizer is looking for uninitialized variables. The redzones in ASAN help detect buffer overflows.

39. ANSWER: B, C
    Explanation: Testing is incomplete since even a large set of test cases may not cover all edge cases or trigger all code paths. Testing is sound, however, since a failed test case runs the actual code and finds unintended behavior.

40. ANSWER: B
    Explanation: Input structure and application-specific structure are necessary for grammar-based fuzzing, not mutation-based fuzzing. Mutation-based fuzzing can, nonetheless, rely on mutations other than random bit flipping.

41. ANSWER: B, C
    Explanation: Static analysis is still useful, because it detects possible bugs in all parts of a program, usually at low cost, often during compilation. Of course a fuzzer can only trigger a bug if it runs the buggy code region. A fuzzer can satisfy some constraints through brute force, and harder ones with the addition of tools such as constraint solvers. Sanitizers can have false negatives. Also, sanitizers cannot detect all bugs, e.g., logic bugs. Crashes while fuzzing still commonly require manual effort to triage bugs, determine exploitability and to patch/fix.

42. ANSWER: B
    Explanation: IDS introduce operational costs to an organization as they usually require expert personnel to monitor them, analyze their alarms, and take appropriate action if needed. Anomaly-based IDS are trained using normal network traffic so that they can “learn” the normal conditions and possibly flag network anomalies in the future. Signature-based IDS require previous knowledge of an attack to create a signature. Nonetheless, a matched signature does not mean a successful attack (e.g., a network vulnerability might be effective for Windows-based systems while the system under protection is a Linux one).

43. ANSWER: A, D
    Explanation: Dalvik bytecode can be disassembled to smali disassembly. Original variable names and comments are lost in the compilation process and are, thus, no longer present in the bytecode. Dalvik bytecode is incompatible with the Java virtual machine. Dalvik bytecode contains most of the original type information and can often be decompiled to source code (Java).

44. ANSWER: B, D
    Explanation: Not all system software running on an Android smartphone is open source. Vendors typically add proprietary binary blobs to their OS release for hardware support or user interface customizations. "Easily" installing a different or custom OS version on an Android smartphone depends on your definition of easy but it is generally much easier than on an iOS device. E.g., big communities for custom Android ROMs exist and provide both tutorials and software for this task. Even though a big portion of an Android app's logic is written in one of the JVM-compatible languages to then execute on top of the Dalvik VM, apps can include native libraries and call into them using the Java Native Interface (JNI). Malware can consequently also be developed in another language that is compiled to a native binary and executed through calls from the Dalvik VM into the native code. Applications are executed in a sandbox (with its security guarantees provided by the Linux kernel underlying Android), preventing them from influencing another app's execution (under the assumption that an attacker is not able to actually cross this kernel boundary and exploit the kernel).

45. ANSWER: C, D
    Explanation: The attacker can guess the correct length by trying different lengths of random passwords. Once the correct length is found, the second part runs and therefore the function takes longer to execute. In the second phase, the attacker can systematically vary each character, with the correct character being indicated by a shorter request time since the `log` function does not run.

46. ANSWER: B
    Explanation: If the PCR values are not reset to zero by the BIOS boot block, the TPM_Extend uses the values left in them and will produce different, and incorrect, results in different executions.

47. ANSWER: D

48. ANSWER: D
    Explanation: Mapping the actual age to the range of (20, 30] and (30, 40] will get two equivalence classes {Alice, Claire} and {Bob, David, Elia}, where 2-anonymity holds. Removing the "Age" column can also achieve 2-anonymity, but it conceals the "Age-Problem" correlation in the data set and is an over-reduction. While removing the "Zip Code" does not increase anonymity, the attacker can still identify a person with the age. The "Problem" column is the key data to publish, and cannot be removed.

49. ANSWER: C

50. ANSWER: A

51. ANSWER: A

52. ANSWER: C

53. ANSWER: D

54. ANSWER: A

55. ANSWER: B
    Explanation: It is the principle behind an homogeneity attack the same QID and attributes leak information (then invalidating the 3rd answer, as it increments diversity but not anonymity). The adversary cannot deanonymize an anonymity set out-of-the-blue he/she will know the QIDs, but not the attributes (consider the same QIDs but different attributes). Homomorphic encryption does not have anything to do with this question.

56. ANSWER: D
    Explanation: Differential privacy, to prevent leakage of personally identifiable data about Eve from queries to a database containing Eve's data. Homomorphic encryption will not help since Alice's app does not have direct access to the database. Adversarial machine learning is irrelevant. Running BoBWatch's server in a TEE helps protect data against the cloud provider, not Alice.

57. ANSWER: B

58. ANSWER: D

59. ANSWER: B

60. ANSWER: D

61. ANSWER: A
    Explanation: We could overfit a system so that it would block all queries. It would, however, be useless since all non-malicious queries would be flagged as false positives.

62. ANSWER: A
    Explanation: The parameters and, thus, the training data are unknown in a grey-box scenario. Without the parameters, the adversary does not have access to a trained model that can be run and inspected locally.

63. ANSWER: C
    Explanation: P-norms are not always consistent with perceptual similarity. One example is rotating an image might have a large norm, but it is a small change perceptually. Certified defenses only ensure robustness within a radius of norm. A three-dimensional model has four (3+1) parameters, thus needing four input-output pairs.

64. ANSWER: A, B, C
    Explanation: We have seen examples in the lecture where adversarial machine learning was used to introduce imperceptible changes (panda $\rightarrow$ gibbon) or perceptible changes (Reese Witherspoon $\rightarrow$ Russel Crowe) to an input. The latter was also used as an example on how to bypass facial recognition software, just by equipping the actress with frames for glasses that changed the model's classification, even though a human would still recognize the actress correctly. Adversarial machine learning can be defended against, e.g., by adversarial training or by detecting suspicious query patterns to the trained model (if it is hosted by the defender). How effective those defenses are in the general case is yet to be shown but they can at least thwart certain attacks.

65. ANSWER: B

66. ANSWER: C