Answer 1

1) Already solved as example.

2)

a) "/><script>alert(document.domain);</script>

b) <input type="text" name="p1" size="50" value="">

3)

a) <script>alert(document.domain);</script>

b) <select name="p2"> <option>Japan</option> <option>Germany</option> <option>USA</option> <option>United Kingdom</option> </select>

4)

a) &#34> <script>alert(document.domain);</script>

b) <input type="hidden" name="p3" value="">

5)

a) "><script>alert(document.domain);</script>

b) <input type="text" name="p1" maxlength="15" size="30" value="">

Answer 2

a) The account of a legitimate user can be blocked by a hacker or merely another normal user. This is a violation of principle of least common mechanism as attacks could block the account of a legitimate users resulting in a denial of service.

b) Yes I agree with the argument. The system identified the unauthorized access due to failed password attempts and hence, it does not provide access to the account. Hence only the subject given the access is allowed

---

Answer 3

a) Thou shall not use a computer to steal.
b) Thou shall not use or copy commercial software for which you have not paid.
c) Thou shall not use a computer to harm other people.
d) Thou shall not use other people's computer resources without authorization.
e) Thou shall not snoop around in other people's files.

Answer 4

- Broken Access Control: No. After viewing the bug reports filed for our group we found that uploaded pdfs by seller can be viewed by simply typing the name of the pdf file in the url. We could have implemented a functionality to restrict the access of static files/images.

- Cryptographic Failures: Yes. We used sha256 for hashing the passwords which was provided by django internally and https for our website.
- Injection: Yes. Django comes with internal API which avoids using the interpreter entirely, provides a parameterized interface.
- Insecure Design: Before even writing a single character of code we made UMLs, held meetings regarding the design because changing the code afterwards would have been much difficult.
- Security Misconfiguration: Yes. We sent security directives to clients, https and password hashing with sha256 and added recaptcha for preventing bots.
- Vulnerable and Outdated Components: Yes. We used latest versions available for add libraries and removed unused dependencies, unnecessary features, components, files, and documentation. Scanned the website with Nessus and Zap.
- Identification and Authentication Failures: No. We used captcha, cryptography etc. but we failed to recognise the brute forcing otps for password resets. The otps we were sending were not expiring. We could add an expiration and limit the attempts for entering otps to prevent their brute force.
- Software and Data Integrity Failures: Yes. Installed the libraries form trusted sources and used digital signatures to verify the integrety of the files.
- Security Logging and Monitoring Failures: Yes. Django already provides logs but on top of that we implemented our own logs for other important features.
- Server-Side Request Forgery: Yes. Validated all client side input data, enforced deny by default firewall policies, not sending raw responses to clients, disabled https redirections etc.

Answer 5

Hiding the password hashing algorithm does not appreciably add to the security of the system because even after knowing the algorithm the attacker cannot decrypt the hash. Hiding the password by hashing will add appreciably to the system due to the same reason as stated in first part (as it is a one way function).

Answer 6

a) Such a policy would result in loss of privacy of the students of IIITD. If the data is leaked this could result in social embarassment of the students which could harm the reputation of the institute aswell.

b) I suggest the following changes: -

- The data should be irreversebly anonymitized such that no individual could be linked to the data and analysing the data should not reveal that it was taken from IIITD.
- The users should be given an option whether to use their data for this purpose or not.
- They should be allowed to review, edit and delete the data collected.

Answer 7

Although encipherment and decipherment keys are different, breaking the Caeser Cipher is very easy. There are only 26 possibilies we need to try by decrypting the cipher text and checking which one makes sense. A public key cryptosystem should not be possible to break by brute force, if it is

possible to break via brute force then it should take many many years and not just seconds as the case with Caeser Cipher.

Answer 8

a) Checksums are commonly used to detect transmission errors in network communications. We can use symmetric-key encryption algorithms to generate cryptographic checksums to authenticate data. An identity function is not a good cryptographic checksum function because unlike a good crypto checksum function it is easily revertible, it does not produce a fixed size bit string as a result and a small change in the input leads to a small change in the output.
b) The sum program XOR's all words in its input to generate a one-word output is not a good cryptographic checksum function because sum program is insecure**.** It is vulnerable to a MITM attack**.** XOR checksum is a bad hashing function because it does not separate anagrams.

Answer 9

1) In simple sanitization, only the intruder's commands will be deleted, this means the rest of the data is vulnerable. Hence the anonymity is very weak in this level. Due to the only the commands removal it can be readily automated and requires very little to none human intervention.
2) In Information-tracking although sensitive data is replaced, still it is not secure because if the key is found out then the sensitive data will be at risk. Here anonymity is a bit better than in simple sanitization case. Automation requires sensitive data to be identified and encrypted by human.
3) Format sanitization has better anonymity than Information-tracking. The automation would be difficult and require more human intervention than the previous level.
4) Comprehensive sanitization is the most secured out of all four, the unique identifier and encodes the identifier thus created. So, this gives a highly secured environment from hacking or restoring data. This level will be the hardest to automate because all the sensitive data needs to be identified and verified that none exists after sanitization and thus would require the most human intervention.

Answer 10

a) min = n-1, max = (n-1)*(n-2)/2

Expected number of messages = {n-1 + (n-1)*(n-2) / 2 } / 2 = n*(n-1)/4

b) If we assume that the service uses TCP-IP protocol, then the messages can be tracked by verifying the ACKs corresponing to each of the n-1 messages. We would know the ACKs of the messages we sent if we encounter n-1 new ACKs then we can determine if all the messages are sent.