**Part 1: Q&A**

**The URL Cruise Missile**

The URL is the gateway to the web, providing the user with unrestricted access to all available online resources. In the wrong hands can be used as a weapon to launch attacks.

Use the graphic below to answer the following questions:

| Protocol | Host Name | Path | Parameters |
|---|---|---|---|
| http:// | `www.buyitnow.tv` | /add.asp | ?item=price#1999 |

1. Which part of the URL can be manipulated by an attacker to exploit a vulnerable back-end database system?

**Answer: The Parameters**

2. Which part of the URL can be manipulated by an attacker to cause a vulnerable web server to dump the `/etc/passwd` file? Also, name the attack used to exploit this vulnerability.

**Answer: The Path–Path and Directory Traversal (dot-dot-slash /../../../../)**

3. Name three threat agents that can pose a risk to your organization.

**Answer: Cross-Site Scripting - Broken Injection - Using Components with Known Vulnerabilities, DDOS**

4. What kinds of sources can act as an attack vector for injection attacks?

**Answer: Clickjacking – Content Spoofing – Drive By Download – Phising**

5. Injection attacks exploit which part of the CIA triad?

**Answer: Confidentiality, Integrity and Availability. The potential is that the entire Triad could be become comprised dependent on the depth and degree if the compromise.**

6. Which two mitigation methods can be used to thwart injection attacks?

**Answer: Input Sanitation and Input Validation**

---

**Web Server Infrastructure**

Web application infrastructure includes sub-components and external applications that provide efficiency, scalability, reliability, robustness, and most critically, security.

- The same advancements made in web applications that provide users these conveniences are the same components that criminal hackers use to exploit them. Prudent security administrators need to be aware of how to harden such systems.

Use the graphic below to answer the following questions:

| Stage 1 | Stage 2 | Stage 3 | Stage 4 | Stage 5 |
|---|---|---|---|---|
| Client | Firewall | Web Server | Web Application | Database |

1. What stage is the most inner part of the web architecture where data such as, customer names, addresses, account numbers, and credit card info, is stored?

**Answer: Database**

2. Which stage includes online forms, word processors, shopping carts, video and photo editing, spreadsheets, file scanning, file conversion, and email programs such as Gmail, Yahoo and AOL.

**Answer: Web Application**

3. What stage is the component that stores files (e.g. HTML documents, images, CSS stylesheets, and JavaScript files) that's connected to the Internet and provides support for physical data interactions between other devices connected to the web?

**Answer: Web Server**

4. What stage is where the end user interacts with the World Wide Web through the use of a web browser?

**Answer: Client**

5. Which stage is designed to prevent unauthorized access to and from protected web server resources?

**Answer: Firewall**

---

**Server-Side Attacks**

In today's globally connected cyber community, network and OS level attacks are well defended through the proper deployment of technical security controls such as, firewalls, IDS, Data Loss Prevention, EndPoint and security. However, web servers are accessible from anywhere on the web, making them vulnerable to attack.

1. What is the process called that cleans and scrubs user input in order to prevent it from exploiting security holes by proactively modifying user input.

**Answer: Input Sanitation**

2. Name the process that tests user and application-supplied input. The process is designed to prevent malformed data from entering a data information system by verifying user input meets a specific set of criteria (i.e. a string that does not contain standalone single quotation marks).

**Answer: Input Validation**

3. **Secure SDLC** is the process of ensuring security is built into web applications throughout the entire software development life cycle. Name three reasons why organization might fail at producing secure web applications.

**Answer: Cost of implementation – Lack of Support from Management, as a result lack of Quality Assurance Testing - Reliance of Firewall, IDS, IPS Tools**

4. How might an attacker exploit the `robots.txt` file on a web server?

**Answer: This information can be used for the planning and gathering of data for attacks.**

5. What steps can an organization take to obscure or obfuscate their contact information on domain registry web sites?

**Answer: They can privatize their information of WHOIS and update with vague information or data.**

6. True or False: As a network defender, `Client-Side` validation is preferred over `Server-Side` validation because it's easier to defend against attacks.
   - Explain why you chose the answer that you did.

**Answer: False, it's a Risk and a Vulnerability. Currently to secure an environment, there are several Multi-Factor and layers for authentication validation.**

---

**Web Application Firewalls**

WAFs are designed to defend against different types of HTTP attacks and various query types such as SQLi and XSS.

WAFs are typically present on web sites that use strict transport security mechanisms such as online banking or e-commerce websites.

1. Which layer of the OSI model do WAFs operate at?

**Answer: Layer 7 Application**

2. A WAF helps protect web applications by filtering and monitoring what?

**Answer: HTTP/HTTPS Traffic between the web application and the internet**

3. True or False: A WAF based on the negative security model (Blacklisting) protects against known attacks, and a WAF based on the positive security model (Whitelisting) allows pre-approved traffic to pass.

**Answer: True**

---

**Authentication and Access Controls**

Security enhancements designed to require users to present two or more pieces of evidence or credentials when logging into an account is called multi-factor authentication.

- Legislation and regulations such as The Payment Card Industry (PCI) Data Security Standard requires the use of MFAs for all network access to a Card Data Environment (CDE).
- Security administrators should have a comprehensive understanding of the basic underlying principles of how MFA works.

1. Define all four factors of multifactor authentication and give examples of each:
   - **Factor 1: Knowledge - Need to know, what confidential information does a certain employee know**
   - **Factor 2: Possession – What do I possess that's of valuable like security token, passcodes, etc**
   - **Factor 3: Inherent – maybe something like a fingerprint**
   - **Factor 4: Location – Where are the servers located, client machines**
2. True or False: A password and pin is an example of 2-factor authentication.

**Answer: True**

3. True or False: A password and `google authenticator app` is an example of 2-factor authentication.

Answer: **True**

4. What is a constrained user interface?

**Answer: Need to know, limits user access based upon what's actually required for you to complete your responsibilities.**

---

---

**Part 2: The Challenge**

In this activity, you will assume the role of a pen tester hired by a bank to test the security of the bank's authentication scheme, sensitive financial data, and website interface.

**Lab Environment**

We'll use the **Web Vulns** lab environment. To access it:

- Log in to the Azure Classroom Labs dashboard.
- Find the card with the title **Web Vulns** or **Web Vulnerability and Hardening**.
- Click the monitor icon in the bottom-right.
- Select **Connect with RDP**.
- Use Credentials (azadmin:p4ssw0rd*)
- The lab should already be started, so you should be able to connect immediately.
- Refer to the lab setup instructions for details on setting up the RDP connection.

Once the lab environment is running, open the HyperV manager and make sure that the OWASPBWA and Kali box is running.

- Then, login to the Kali VM and navigate to the IP address of the OWASPBWA machine.
- Click the option for 'WebGoat' and start the WebGoat app.
- Use the credentials: `guest:guest`

On the bottom of the left side of the screen, click on `Challenge` and then choose `The Challenge`.

**Note:** A common issue with this lab is the Challange activity failing to start successfully. Hit the `Restart the Lesson` button in the top right if you get an error starting the activity.

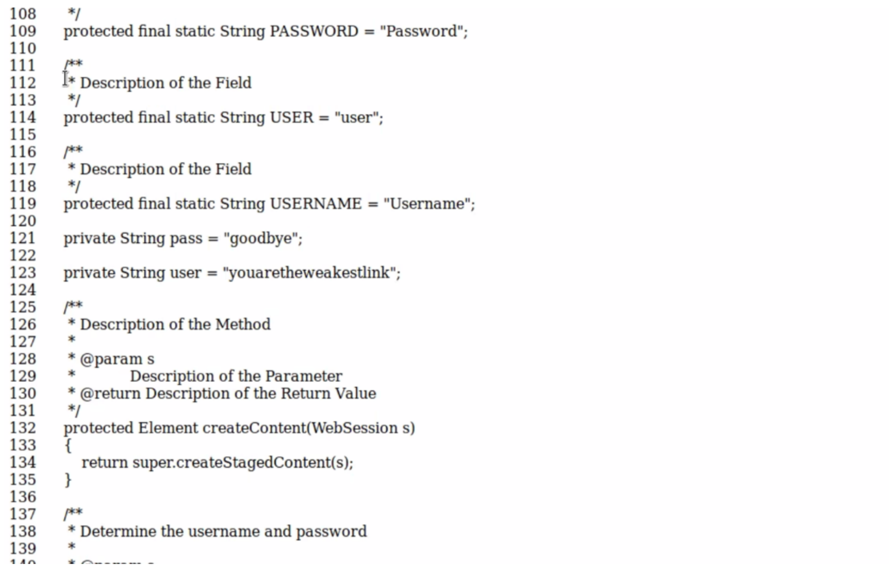
**The Challenge Instructions**

**Challenge #1**

Your first mission is to break the authentication scheme. There are a number of ways to accomplish this task.

- **Hint #1**: Sometimes, form fields are shy!
- **Hint #2**: Find the hidden JavaScript.
- **Hint #3**: You can appened `source?source=true` to the URL to read the source code.

Please include a screenshot here of the hidden JavaScript:



```
 */
protected final static String PASSWORD = "Password";

/**
 * Description of the Field
 */
protected final static String USER = "user";

/**
 * Description of the Field
 */
protected final static String USERNAME = "Username";

private String pass = "goodbye";

private String user = "youaretheweakestlink";

/**
 * Description of the Method
 *
 * @param s
 *           Description of the Parameter
 * @return Description of the Return Value
 */
protected Element createContent(WebSession s)
{
   return super.createStagedContent(s);
}

/**
 * Determine the username and password
 *
```

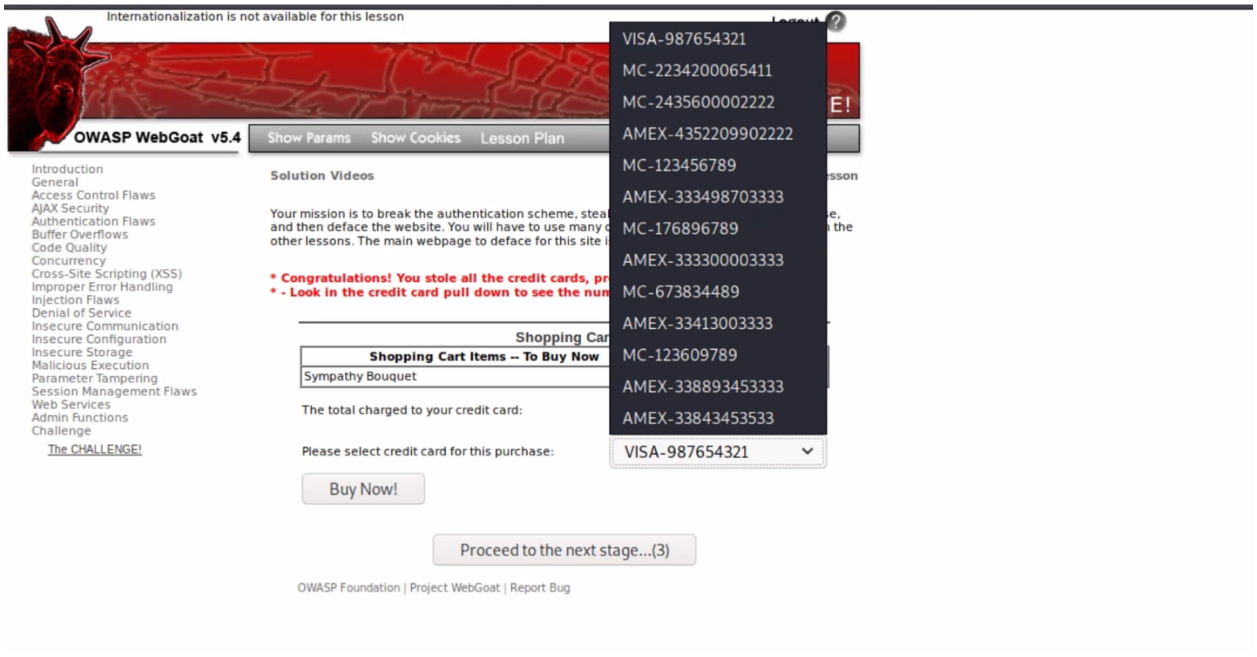
After completing the first challenge, you will be provided with an option to continue to the next challenge.

**Challenge #2**

Next, steal all of the credit card numbers from the database.

- **Hint #1**: Sometimes cookies wear different clothes to change their appearances.
- **Hint #2**: Break your way into the conversation and inject your own ideas.

Please include a screenshot here of all the credit card numbers from the database.



Internationalization is not available for this lesson

OWASP WebGoat v5.4

Your mission is to break the authentication scheme, steal and then deface the website. You will have to use many other lessons. The main webpage to deface for this site i

* Congratulations! You stole all the credit cards, pr
* - Look in the credit card pull down to see the num

Shopping Car

| Shopping Cart Items -- To Buy Now |
|---|
| Sympathy Bouquet |

The total charged to your credit card:

Please select credit card for this purchase: VISA-987654321

```
VISA-987654321
MC-2234200065411
MC-2435600002222
AMEX-4352209902222
MC-123456789
AMEX-333498703333
MC-176896789
AMEX-333300003333
MC-673834489
AMEX-33413003333
MC-123609789
AMEX-338893453333
AMEX-33843453533
```

Buy Now!
Proceed to the next stage...(3)
OWASP Foundation | Project WebGoat | Report Bug

After completing the second challenge, you will be provided with an option to continue to the next challenge.

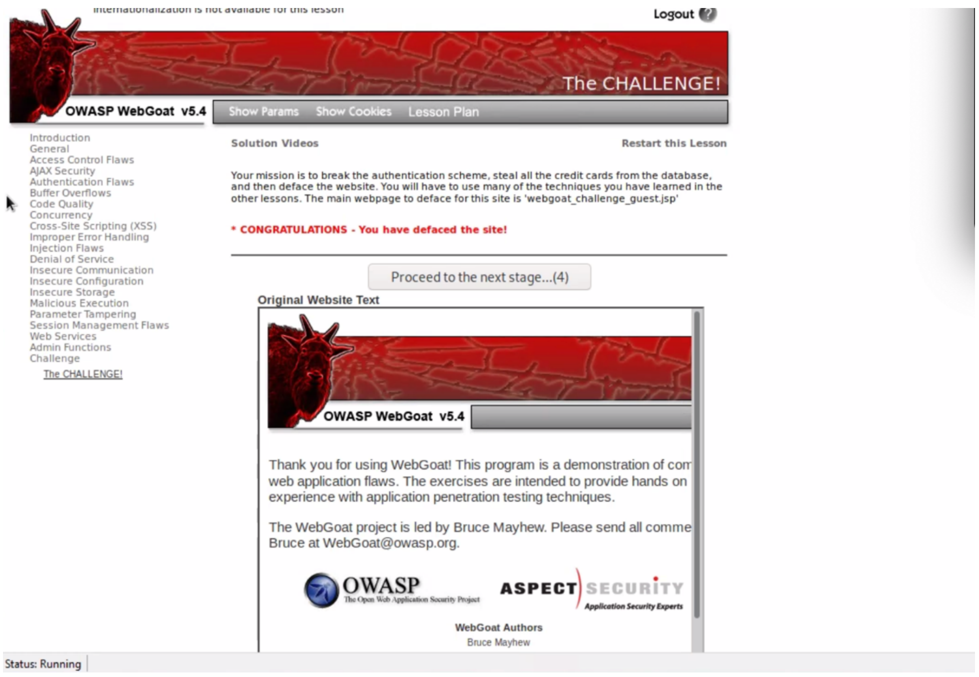
**Challenge #3**

Your final act is to deface the website using command injection. Follow the walkthrough below to help you get started.

**Now it's your turn**

- Now that we know where the webpage is, your task will be to deface the website. Keep in mind the following:
    - Use **WebScarab** to perform command injection.
    - When performing command injection, you will need to select a field that WebScarab can return commands to. These fields are typically located in a drop down.
    - You will also need to locate and edit the the webpage's source code: `webgoat_challenge_guest.jsp`
    - Your final command will:
        - Change to the location of the `webgoat_challenge_guest.jsp` file.
        - **and** echo `You've been hacked by...` followed by your name, to the `webgoat_challenge_guest.jsp` file.

Please include a screenshot of the defaced website.



Your mission is to break the authentication scheme, steal all the credit cards from the database, and then deface the website. You will have to use many of the techniques you have learned in the other lessons. The main webpage to deface for this site is 'webgoat_challenge_guest.jsp'

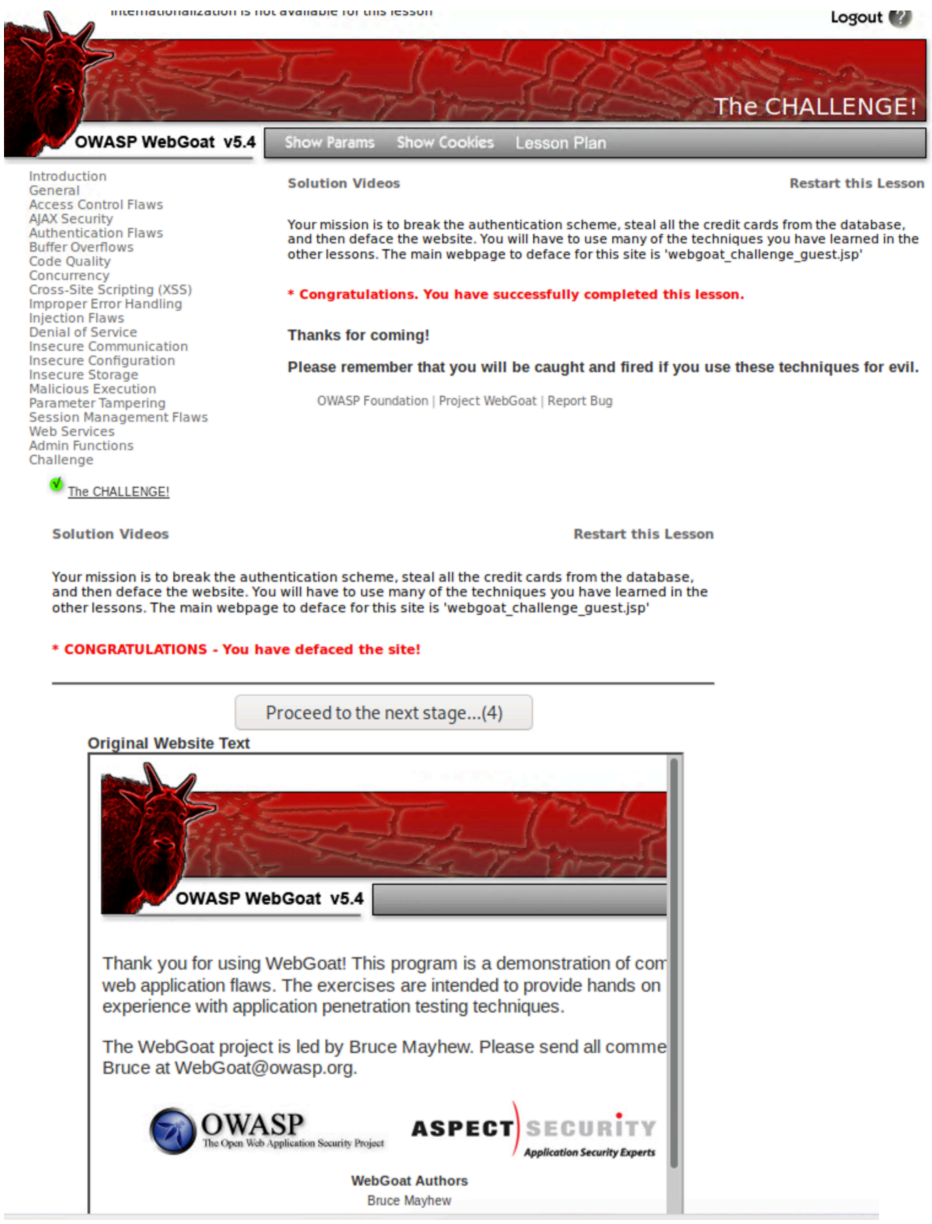
* CONGRATULATIONS - You have defaced the site!

Original Website Text

Thank you for using WebGoat! This program is a demonstration of com
web application flaws. The exercises are intended to provide hands on
experience with application penetration testing techniques.

The WebGoat project is led by Bruce Mayhew. Please send all comme
Bruce at WebGoat@owasp.org.

WebGoat Authors
Bruce Mayhew



Your mission is to break the authentication scheme, steal all the credit cards from the database, and then deface the website. You will have to use many of the techniques you have learned in the other lessons. The main webpage to deface for this site is 'webgoat_challenge_guest.jsp'

* Congratulations. You have successfully completed this lesson.

Thanks for coming!

Please remember that you will be caught and fired if you use these techniques for evil.

OWASP Foundation | Project WebGoat | Report Bug

Your mission is to break the authentication scheme, steal all the credit cards from the database, and then deface the website. You will have to use many of the techniques you have learned in the other lessons. The main webpage to deface for this site is 'webgoat_challenge_guest.jsp'

* CONGRATULATIONS - You have defaced the site!

Thank you for using WebGoat! This program is a demonstration of com
web application flaws. The exercises are intended to provide hands on
experience with application penetration testing techniques.

The WebGoat project is led by Bruce Mayhew. Please send all comme
Bruce at WebGoat@owasp.org.

OWASP
The Open Web Application Security Project
ASPECT SECURITY
Application Security Experts
WebGoat Authors
Bruce Mayhew