Unit 15 Homework: Web Vulnerabilities and Hardening

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?

    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.

    Path can be manipulated by an attacker. The attack is called Path Traversal.

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

    Threat agents/actors would be anyone that can cause, carry, transmit or support a threat (vulnerability) where as Parameter Tampering, Path Traversal, and XSS are classifications of vulnerabilites.

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

    Improper sanitization of servers, Web page caches, URL Parameter and Path vulnerable to tampering. Misconfigurations within a web server, web app, network, or operating system, Bugs in a web server from unpatched systems, Improper permissions that allow overly lenient access to directory and file permissions.

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

All three aspects. Confidentiality is exposed when a client information is leaked through command injection and accessing the system files like /etc/passwd and gaining user info. It can become an Integrity issue when the exploit gives unauthorized remote access to the system that hosts the vulnerable application. Availability issue from the standpoint of the client seeing a defaced web page.

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

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?

Stage 5 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.

Stage 3 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?

Stage 4 Web Server

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

Stage 1 Client

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

Stage 2 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.

Input sanitization

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).

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.

1. High implementation costs
2. Ractie Security mentality
3. Lack of management support and standardization

5. What steps can an organization take to obscure or obfuscate their contact information on domain registry web sites?
Some of the Steps the Organization can take may include;

- Use a proxy
- private domain registration service
- Have a WAF in place
- Have Input sanitization and validation in place
- Whitelisting and Blacklisting IPs
- Restricting character sequences and parameters

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.

False
The benefit of doing Server-side validation over Client-side validation is that client-side validation can be bypassed/manipulated since the end user could have javascript switched off and data could be sent directly to your server by someone who's not even using your site, with a custom app designed to do so. There maybe a Javascript error on your page (caused by any number of things) could result in some, but not all, of your validation running.

Client-side validation is prefered over Server-side because it is less complicated and directly impacts the client the most, so ususally client-side validation is done before server-side validation. So validate input on the client-side first because you can give better feedback to the average user, if they enter an invalid email address and move to the next field, you can show an error message immediately. That way the user can correct every field before they submit the form. If you only validate on the server, when they 'submit' the form, they get the error message, and then look for the problem.
Using a Server-Side validation to protect against the malicious user, who can easily bypass your JavaScript and submit dangerous input to the server.

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?
    Layer 7, Application Layer

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

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.

    False, Whitelisting only accepts traffic from sources that are known and trusted and Blacklisting uses preset signatures to block malicious web traffic

---

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 Standard login inputs (password, PIN, cognitive questions)

  - Factor 2 Physical keys (smartcard, hard token)

  - Factor 3 Biometrics (iris/retina scan, hand geometry)

  - Factor 4 Location (GPS detection, callback to a home phone number)

2. True or False: A password and pin is an example of 2-factor authentication. False

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

4. What is a constrained user interface?

It restricts what users can see and do based on their privileges.

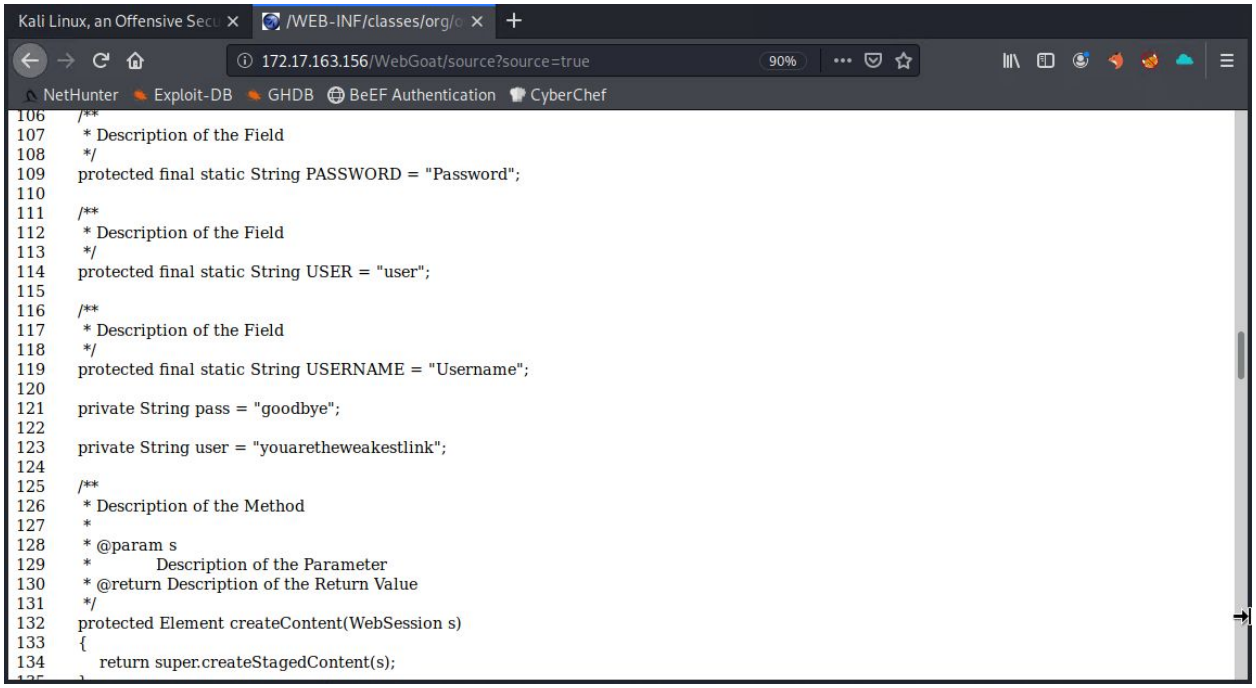
Part 2: The Challenge

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.

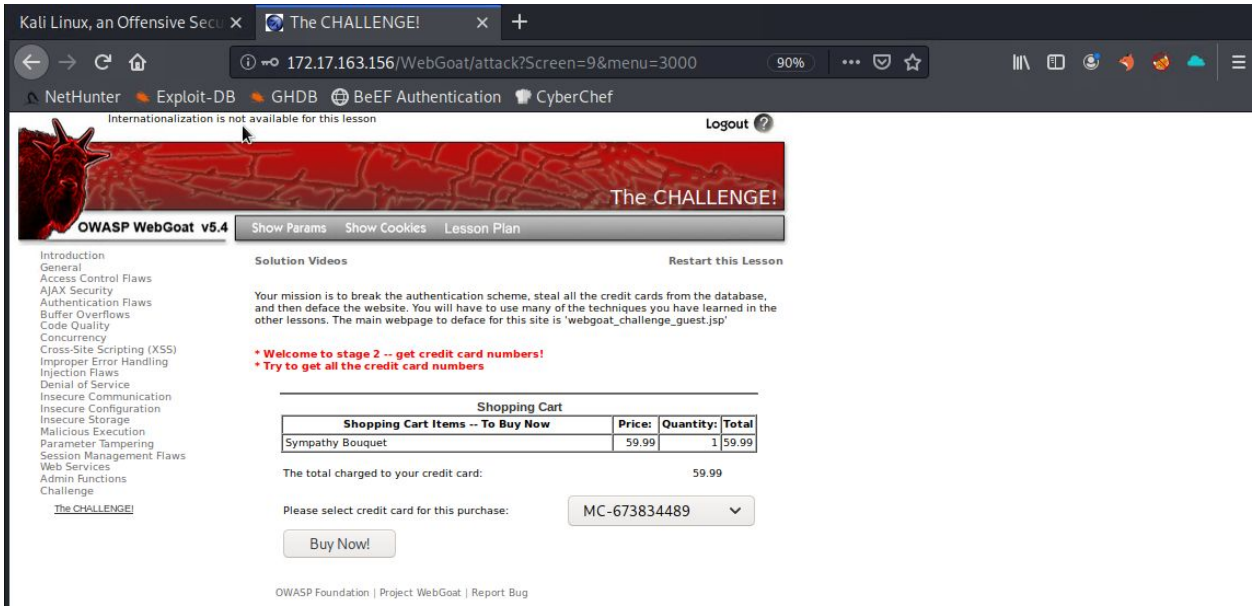
Challenge #1 Breaking the authentication stream

Change the URL to reveal the underlying Javascript. Change after WebGoat/, /WebGoat/source?source=true`

As we can see on lines 121 and 123 we get the username and password revealed to us on the Javascript.

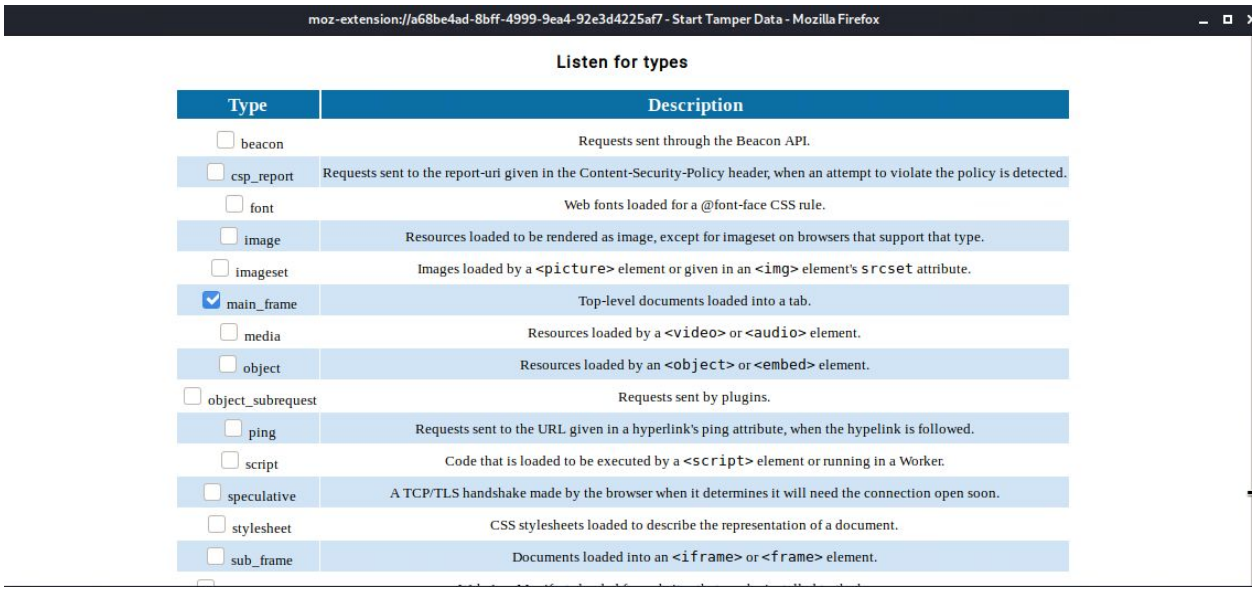


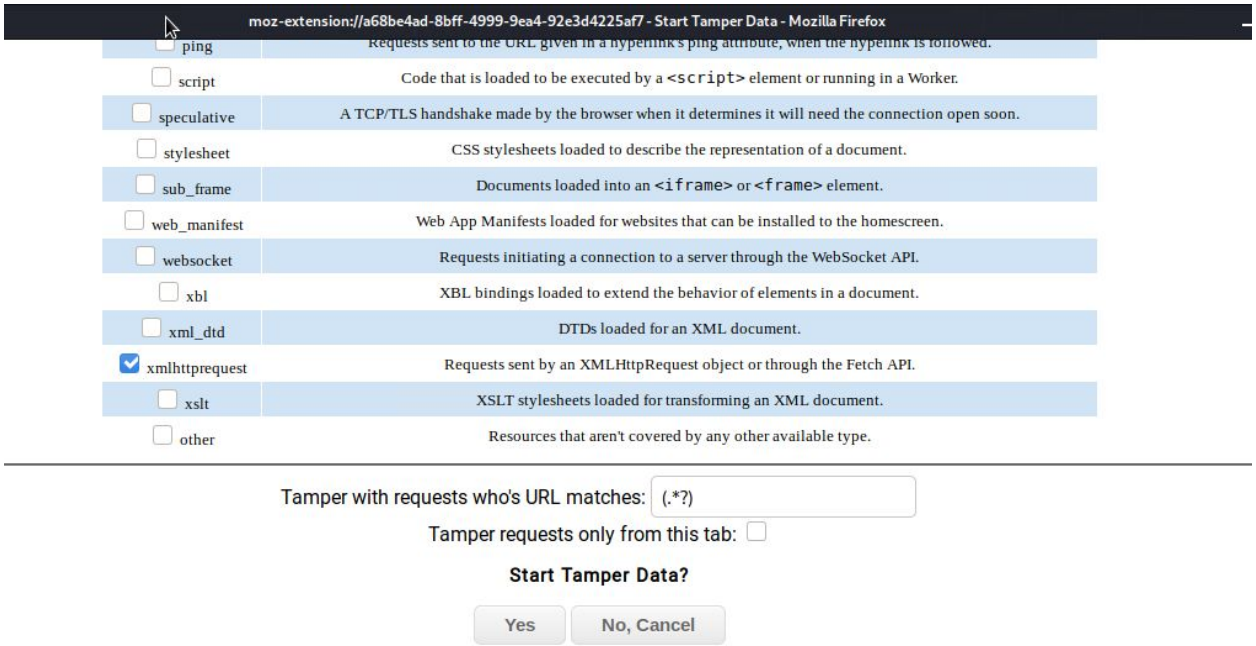
Put them in and challenge 1 is complete



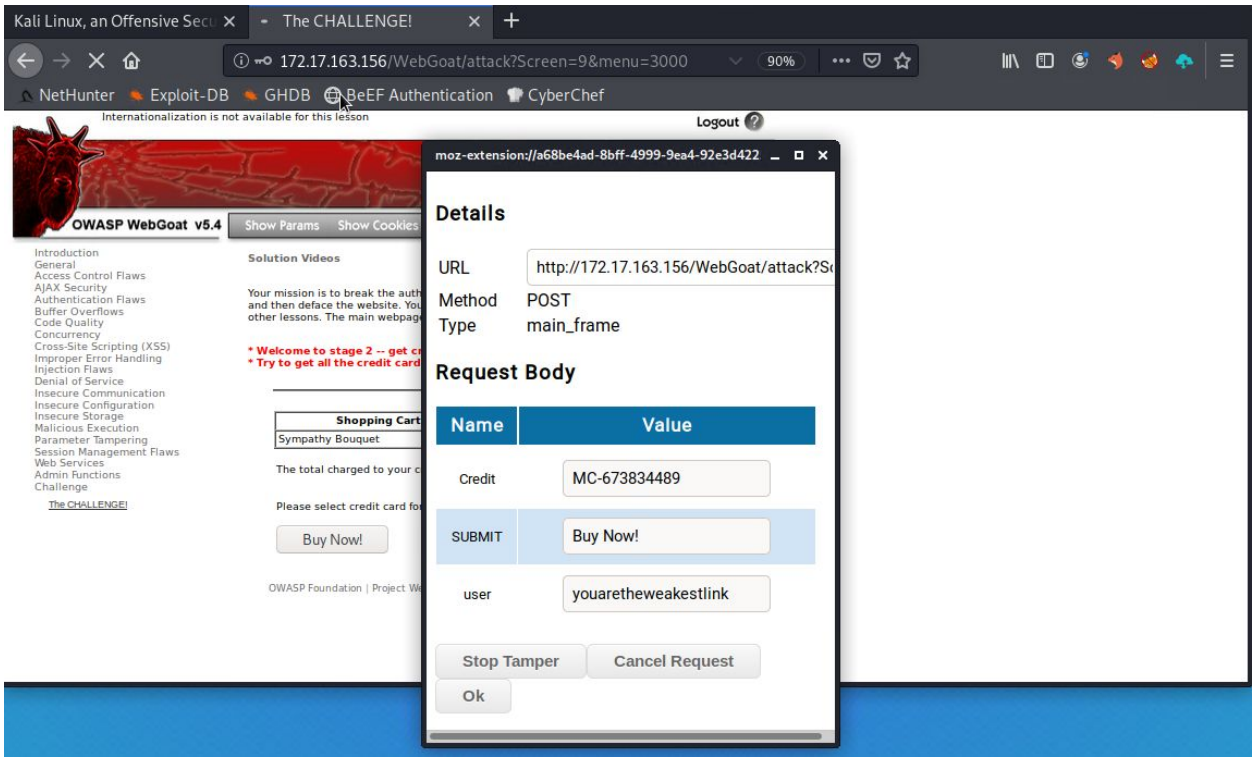
Challenge #2 Steal Credit Card numbers

Start Tamper Data by clicking the icon on extensions tab, Click Yes on the Start Tamper Data Window

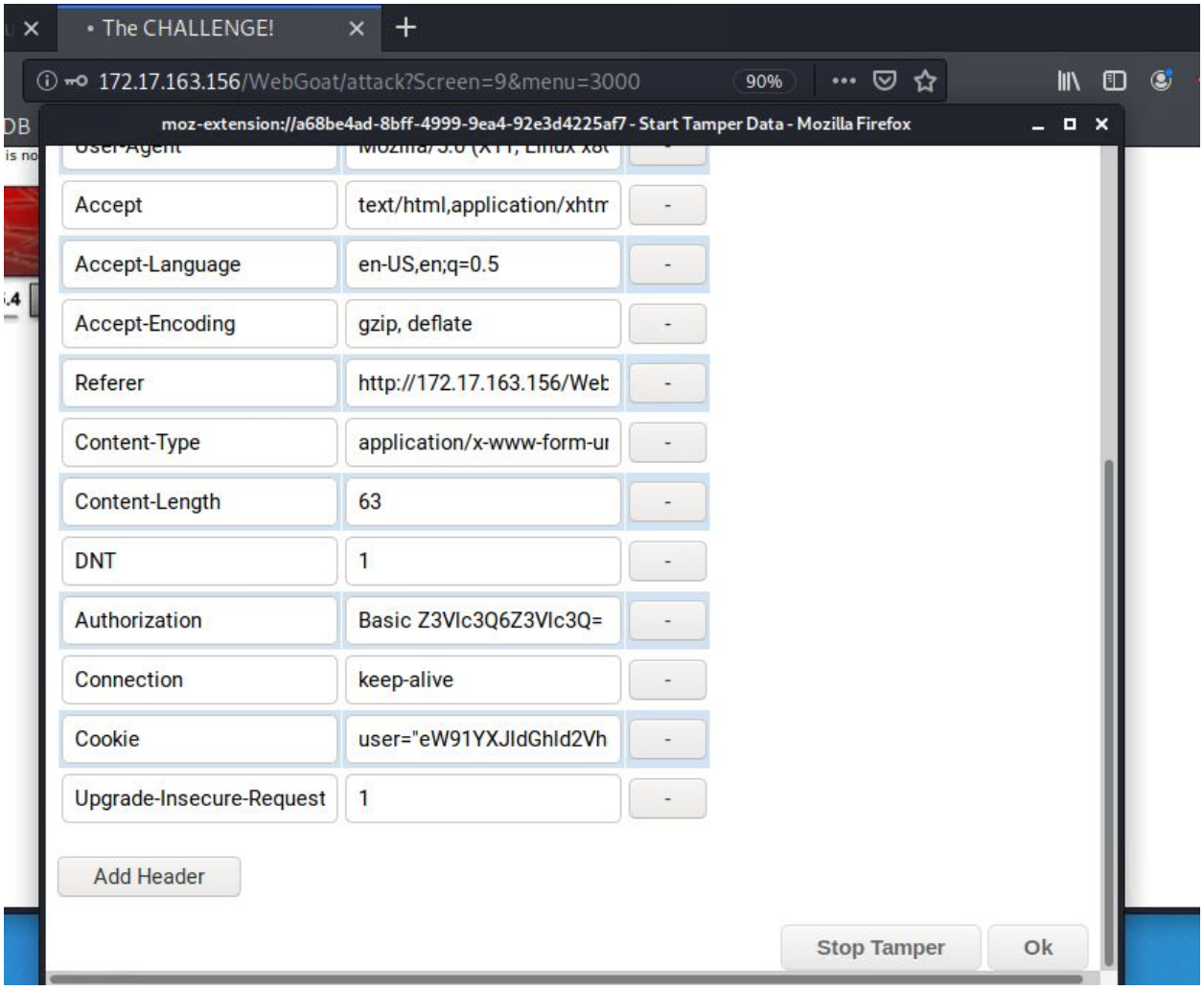




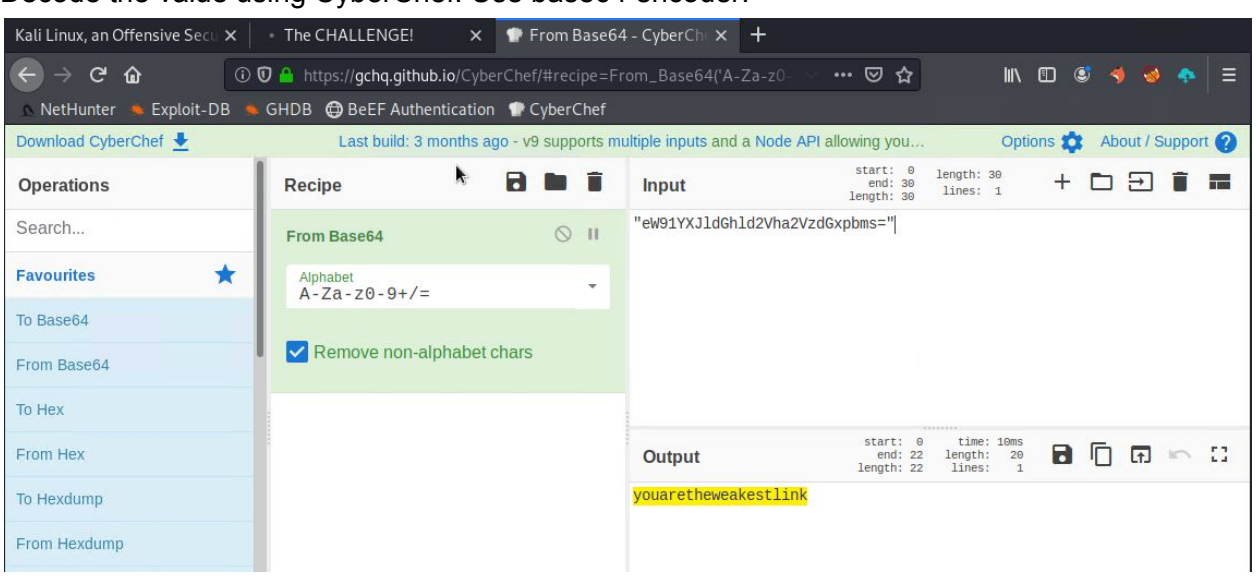
Next Press Buy Now! And then OK in the Tamper Data Request Box



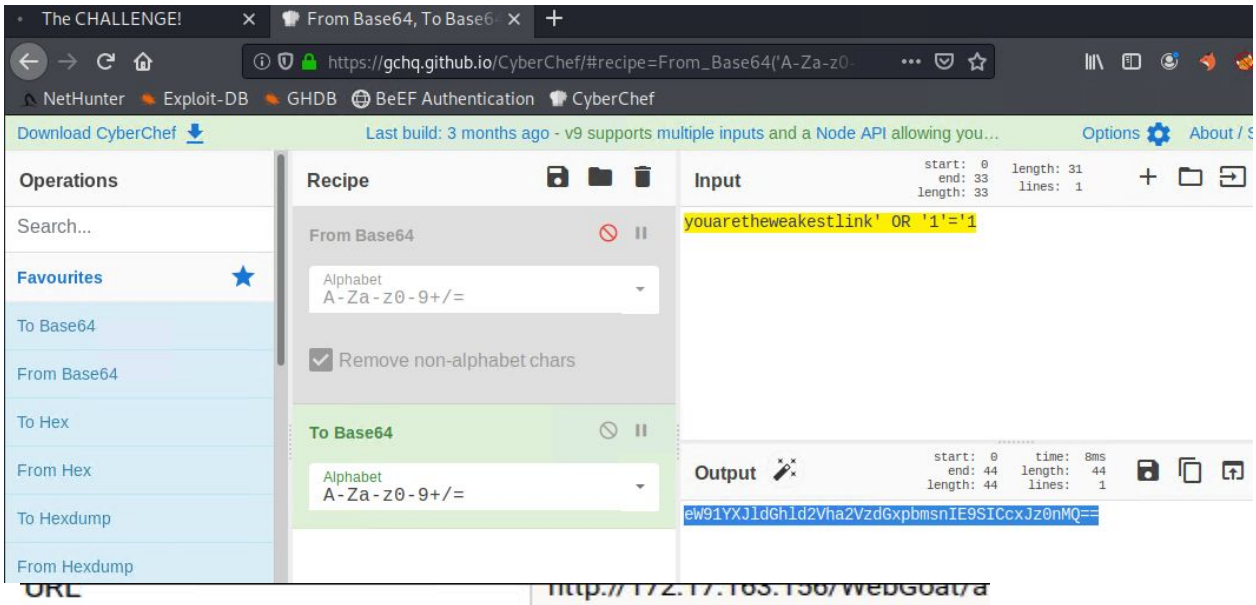
In Tamper Data window, scroll down to the Authorization box and copy the user value from the cookie section

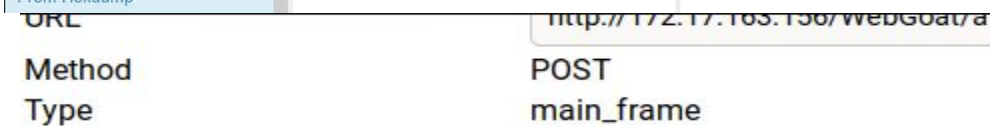


Decode the value using CyberChef. Use base64 encoder.

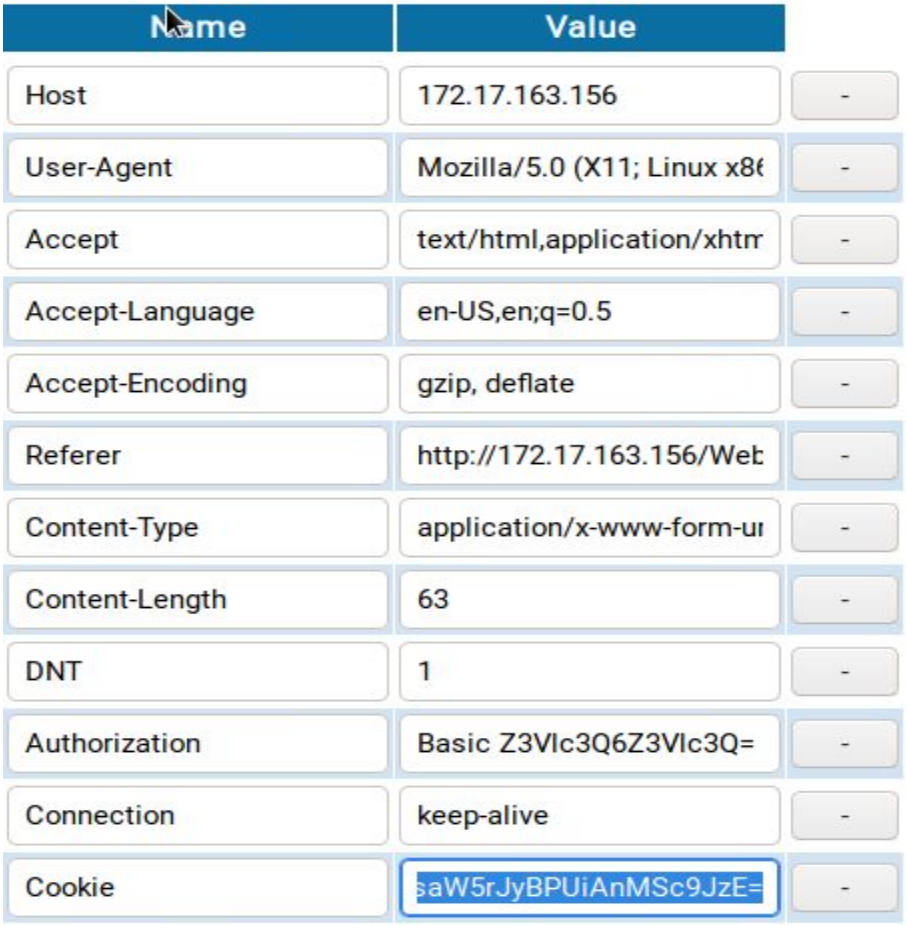


The decoded value is yourethweakestlink now we can use this value with the SQLi command OR '1'='1, it becomes yourethweakestlink OR '1'='1, we encode this to Base64 and Go back to Tamper data and inject the encoded value into the cookie value field between the quotations.



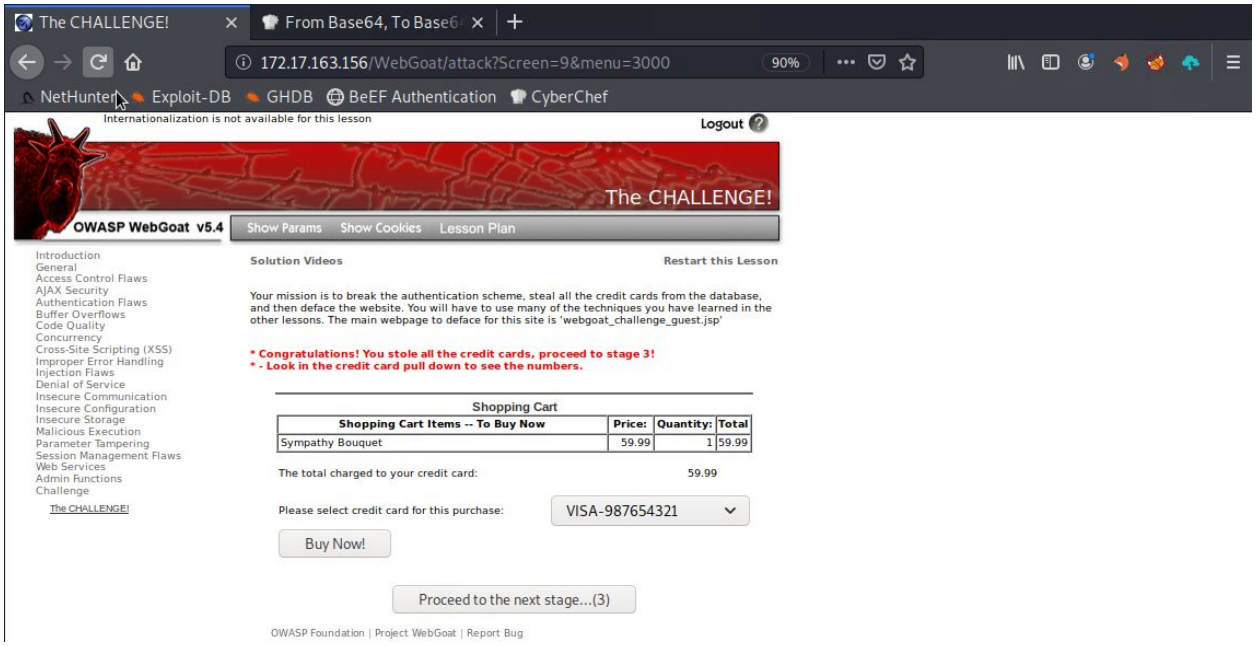


Headers



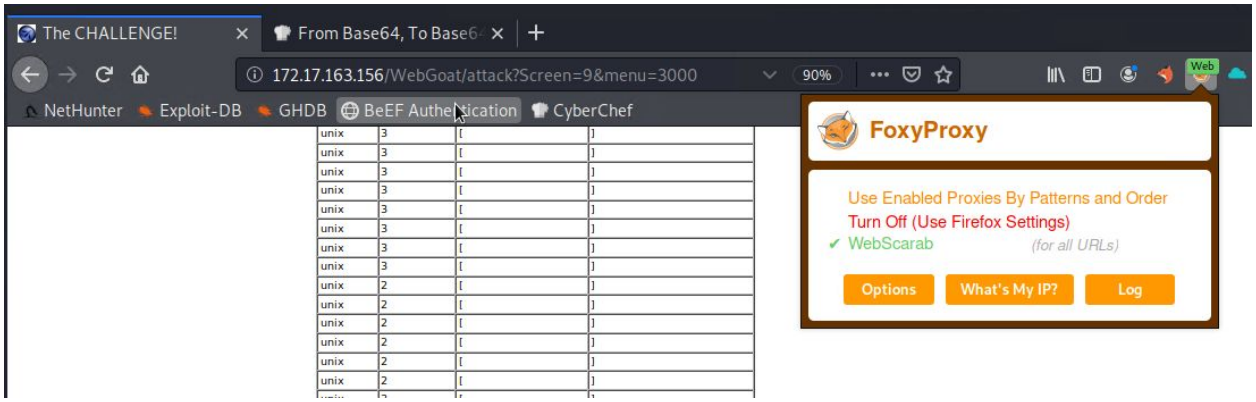
Name	Value	
Host	172.17.163.156	-
User-Agent	Mozilla/5.0 (X11; Linux x8	-
Accept	text/html,application/xhtm	-
Accept-Language	en-US,en;q=0.5	-
Accept-Encoding	gzip, deflate	-
Referer	http://172.17.163.156/Web	-
Content-Type	application/x-www-form-u	-
Content-Length	63	-
DNT	1	-
Authorization	Basic Z3Vlc3Q6Z3Vlc3Q=	-
Connection	keep-alive	-
Cookie	saW5rJyBPUiAnMSc9JzE=	-

After the request is forwarded we get the credit card numbers and congratulations we have completed challenge #2.

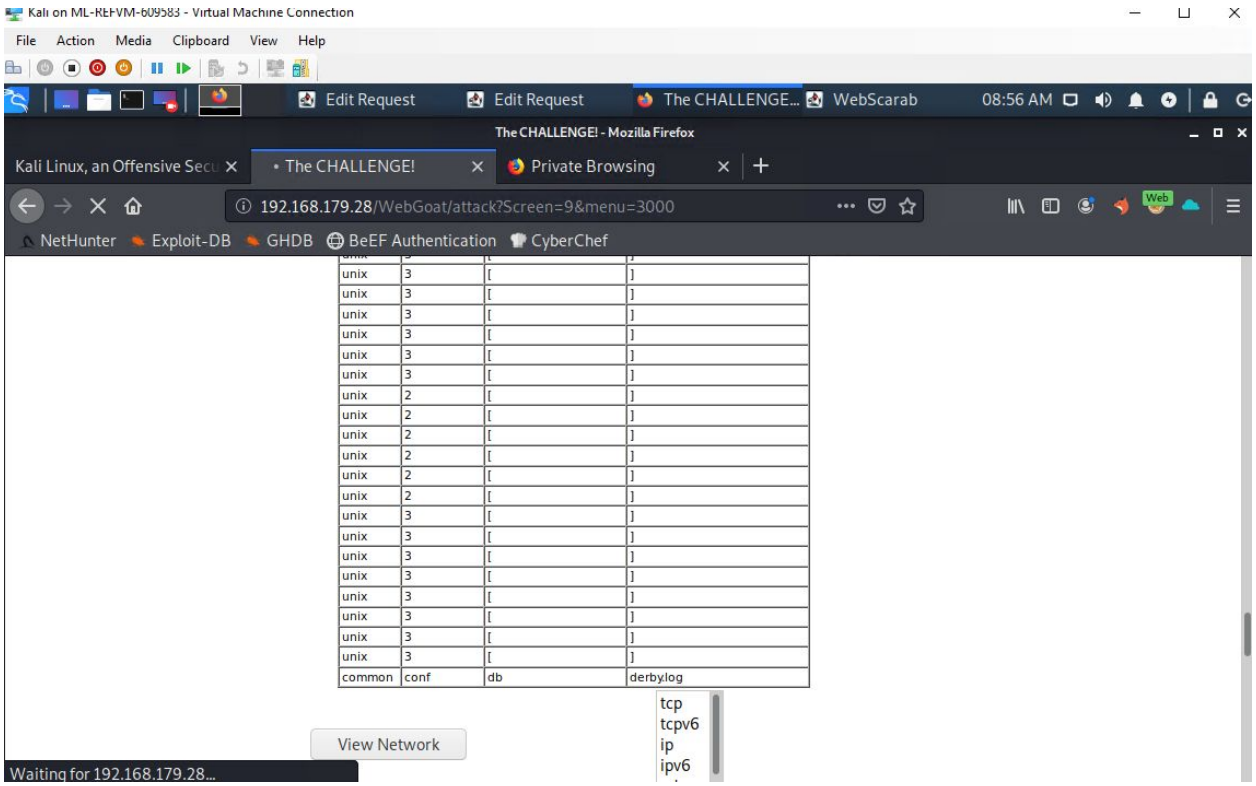


Challenge #3 Deface the website using command injection

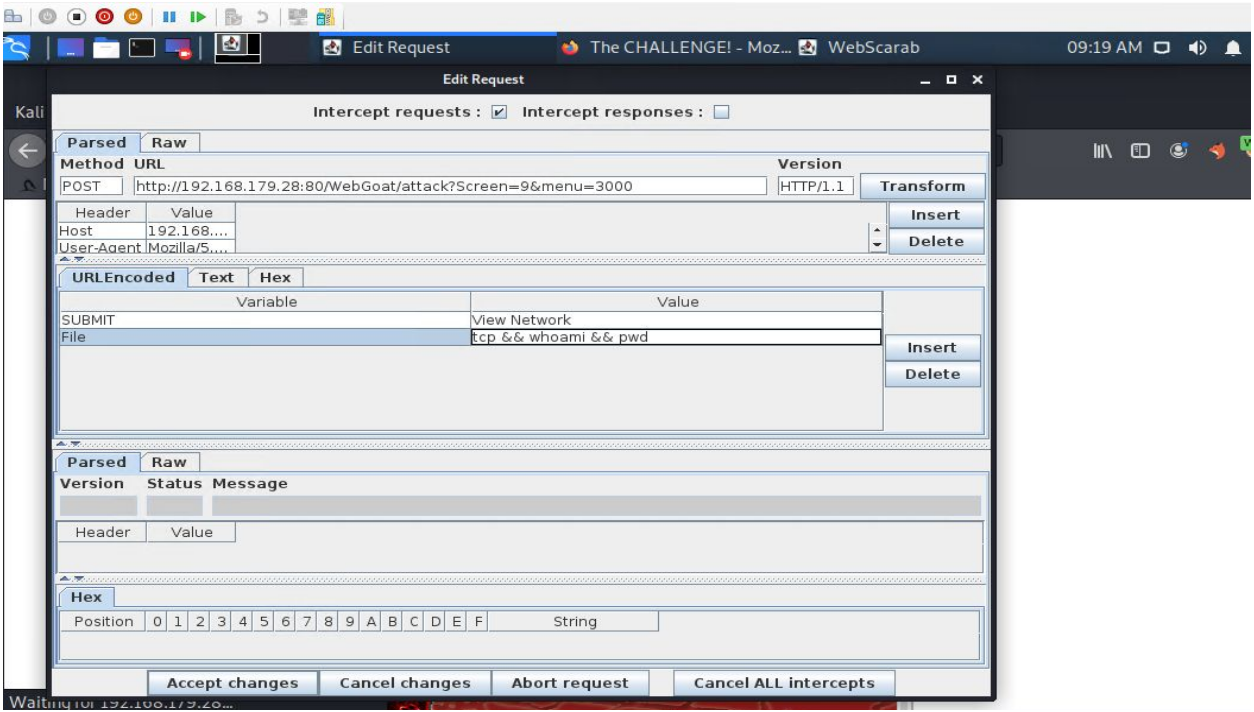
After completing Challenge #2 we get the option of proceeding to the next stage as shown above.
In this challenge we need to use command injection to get root access then locate the webgoat_challenge_guest.jsp file and inject it with code.
Firstly we need to turn on WebScarab and then turn on FoxyProxy to send GET/POST requests from the FireFox browser to WebScarab.



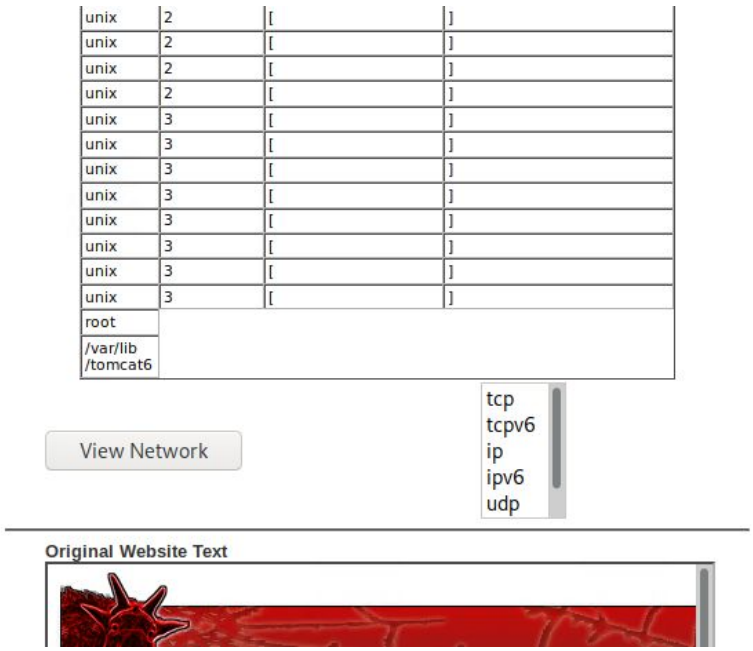
We select tcp and then hit the view network button, this will send a request to WebScarab.



The WebScarab Edit request window will pop up, here we will edit the Value (tcp) of Variable - File, with a simple command injection to check which directory we are in and if we can actually get a result. So we edit the Value field with “tcp && whoami && pwd” (this will tell the present

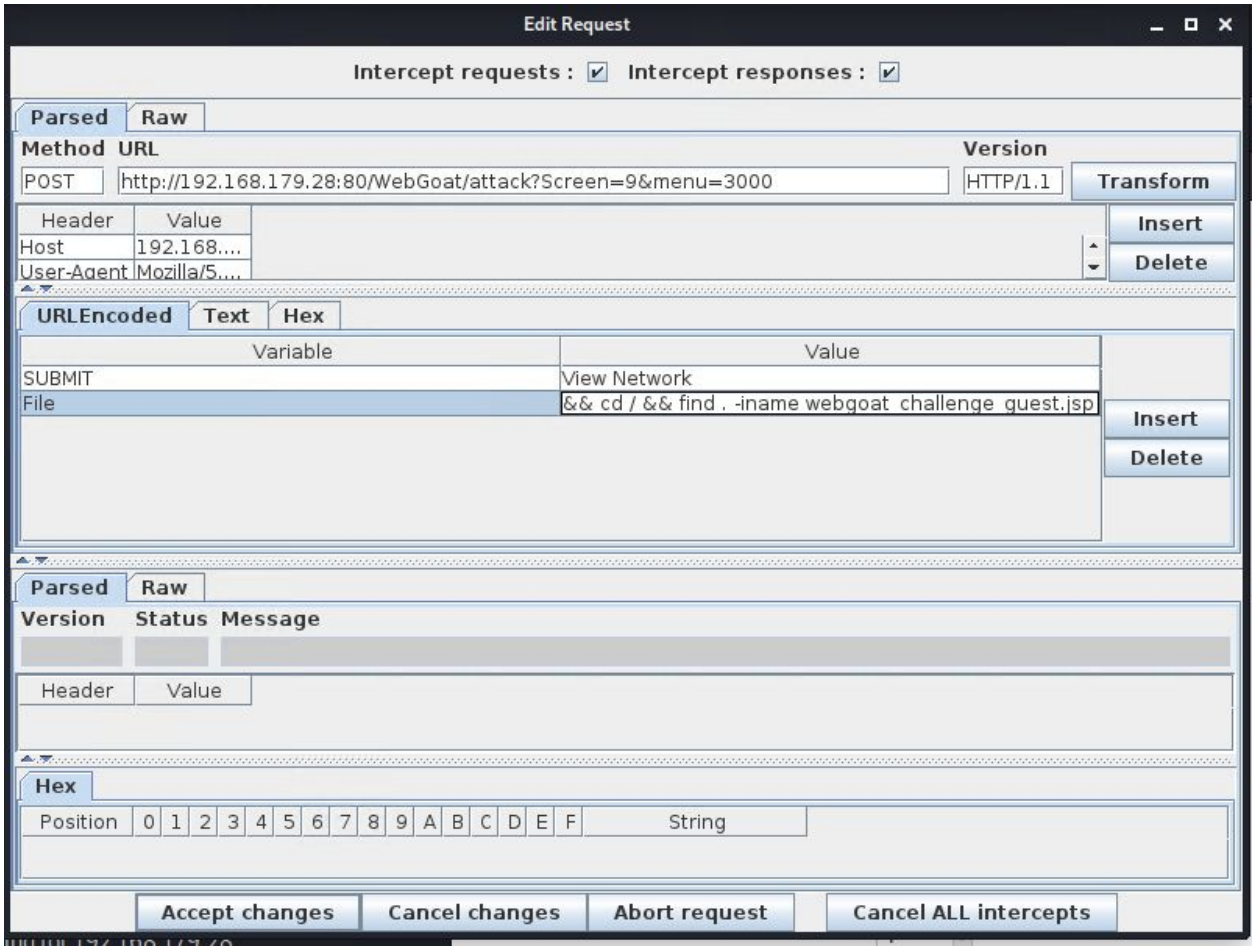


working directory we are in) . Then we hit accept changes and scroll down to see the results for the commands on the Network status screen, as shown above. It tells us that our current working directory is /var/lib/tomcat6, this proves that this website is vulnerable and that we hae root access since /var is a root level directory.

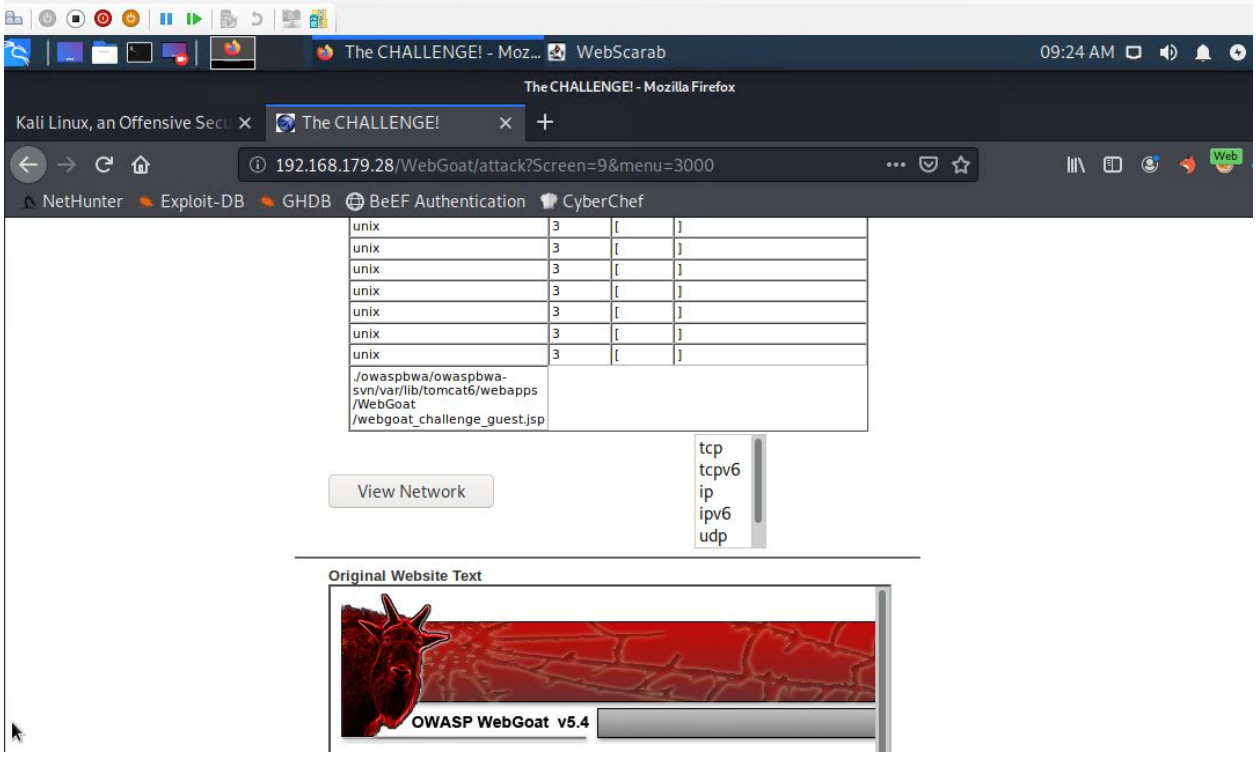


Now its time to locate the webgoat_challenge_guest.jsp file. We again click tcp and then view network to get to the Edit Request screen in Webscarab. Then input the command to locate the

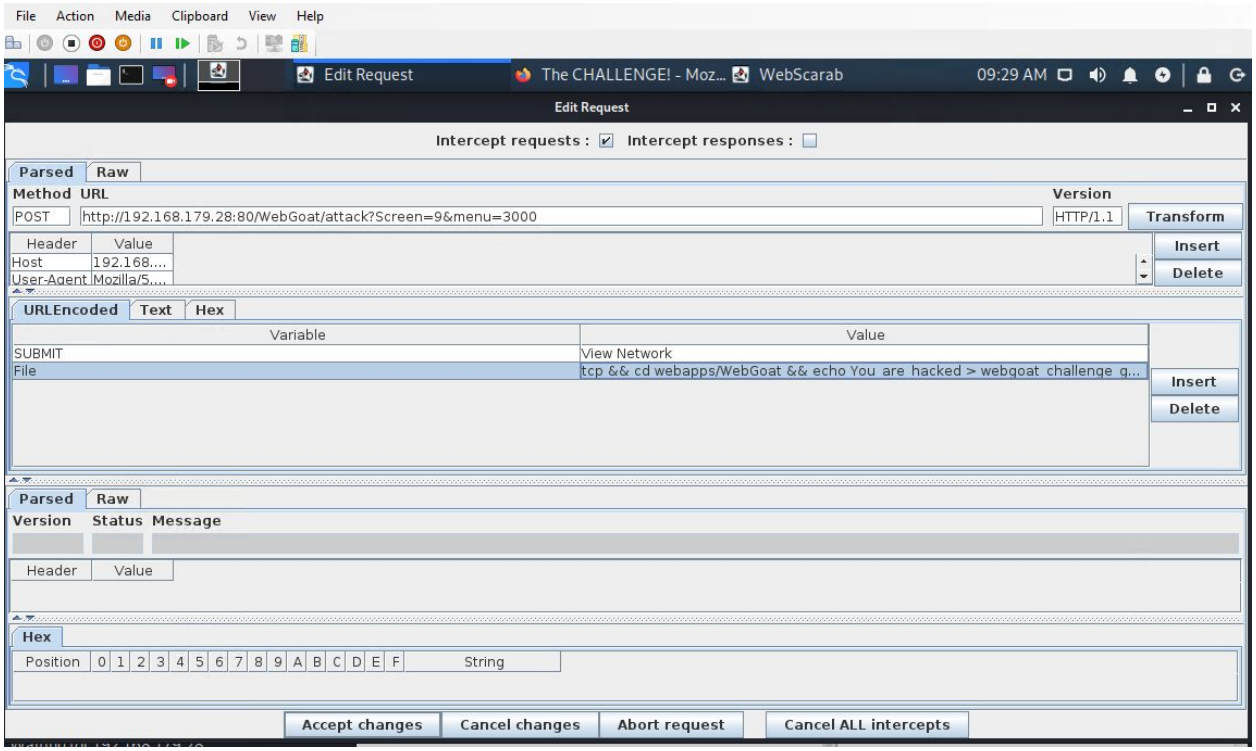
file, “tcp && cd / && find . -iname webgoat_challenge_guest.jsp”



This command will find the path for where the webgoat_challenge_guest.jsp file is located. As show below the absolute path is
./owaspbwa/owaspbwa-svn/var/lib/tomcat6/webapps/WebGoat/webgoat_challenge_guest.jsp



Now all we need to do is edit the webgoat_challenge_guest.jsp with a message that your



website has been hacked. So we again go to the Edit request page and inject in the value field this SQLi command, tcp && cd webapps/WebGoat && echo You_are_hacked > webgoat_challenge_guest.jsp

We have now successfully defaced the website using command injection.