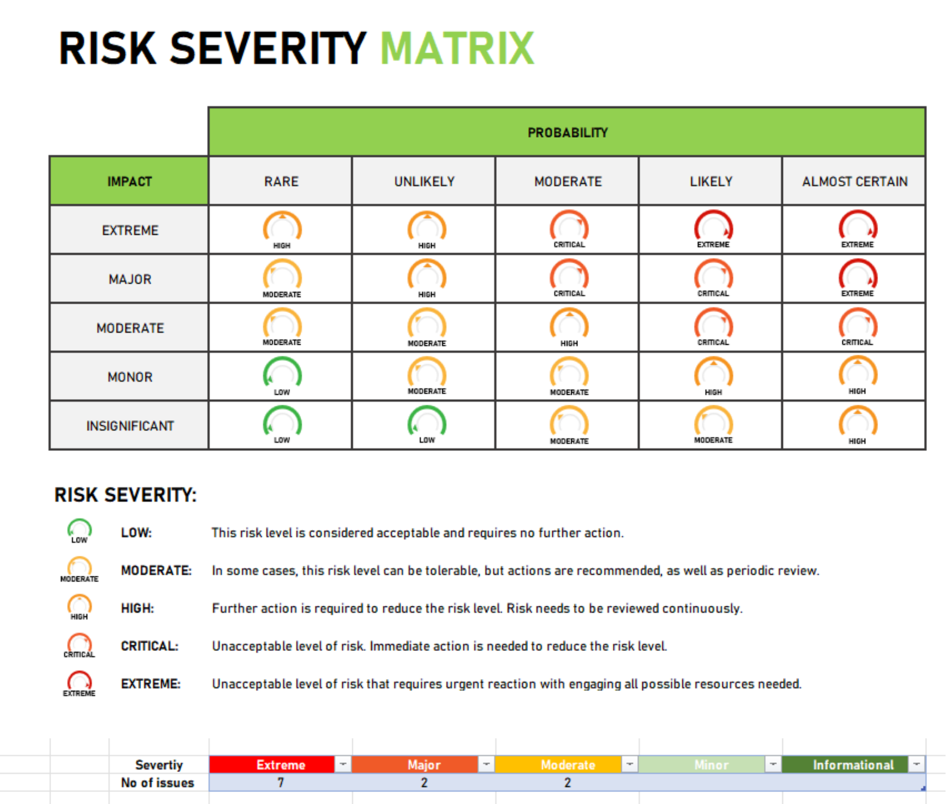
Executive Summary

This report was initiated by the client HTS to identify weakness within the customer side application and a number of structure/procedural issues found during the preliminary risk assessment. The report contains, detected poor methodology in both password protection and client-side access to the source code during the development of the application. The absence of best practice which will lead to vulnerability exploits by state, Insider threats and cyber criminals. Both ISO 27005 and 27001 were used to provide foundational structure to the penetration testing methodology.

Scope

Due to the continuous development of the applications and the potential exposure of the applications to potential threats, it was considered prudent to test the applications vulnerabilities. The initial phase was to identify any potential vulnerabilities around password storage/access, authentication and source code vulnerability. The identified concerns are used to build the report which includes mitigations recommendations to assist with the future stability and security of the platform from malicious actors. The table below identifies potential severity or likely hood of the exploitation occurring.



RISK SEVERITY MATRIX

	PROBABILITY				
IMPACT	RARE	UNLIKELY	MODERATE	LIKELY	ALMOST CERTAIN
EXTREME	HIGH	HIGH	CRITICAL	EXTREME	EXTREME
MAJOR	MODERATE	HIGH	CRITICAL	CRITICAL	EXTREME
MODERATE	MODERATE	MODERATE	HIGH	CRITICAL	CRITICAL
MONOR	LOW	MODERATE	MODERATE	HIGH	HIGH
INSIGNIFICANT	LOW	LOW	MODERATE	MODERATE	HIGH

RISK SEVERITY:

LOW	LOW:	This risk level is considered acceptable and requires no further action.
MODERATE	MODERATE:	In some cases, this risk level can be tolerable, but actions are recommended, as well as periodic review.
HIGH	HIGH:	Further action is required to reduce the risk level. Risk needs to be reviewed continuously.
CRITICAL	CRITICAL:	Unacceptable level of risk. Immediate action is needed to reduce the risk level.
EXTREME	EXTREME:	Unacceptable level of risk that requires urgent reaction with engaging all possible resources needed.

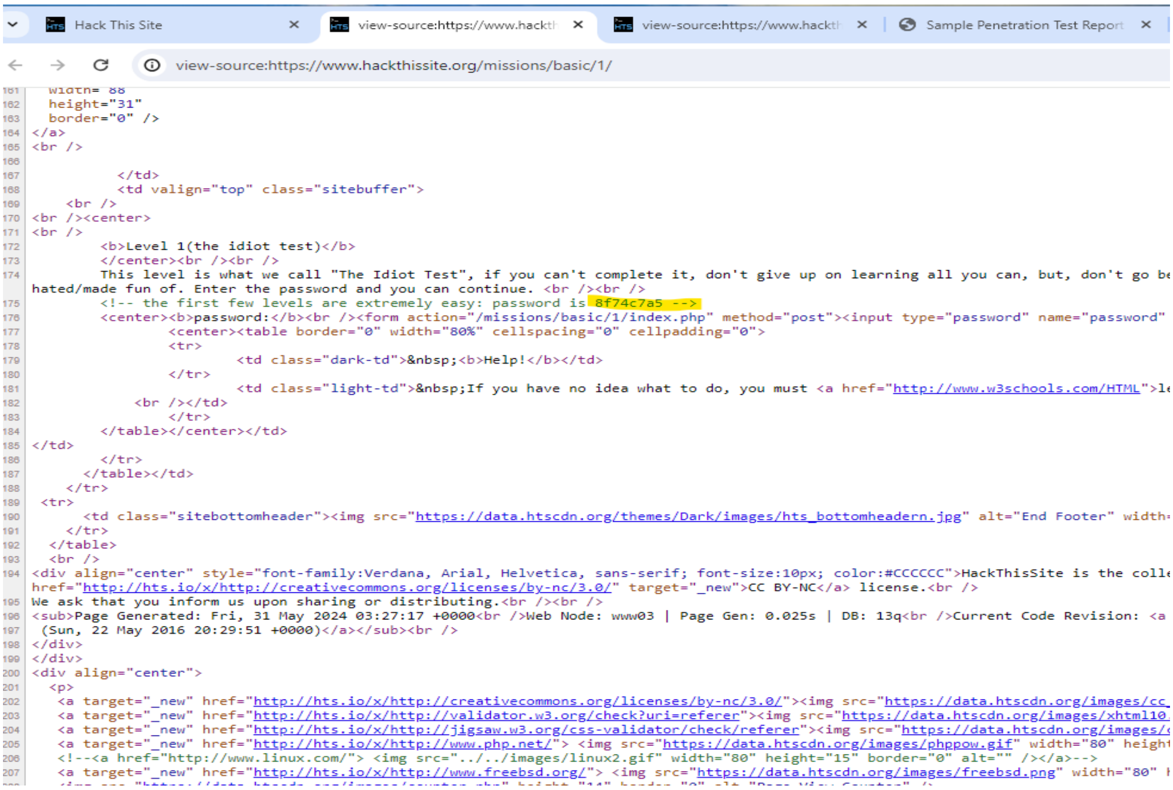
Severtiy	Extreme	Major	Moderate	Minor	Informational
No of issues	7	2	2		

Action Taken

Level 1 (Password HTML)

Action

Password identified within the web application source code which could indicate other sensitive information/notes could be available.


Hack This Site
view-source:https://www.hackth
view-source:https://www.hackth
Sample Penetration Test Report
view-source:https://www.hackthissite.org/missions/basic/1/

```
  width="88"
  height="31"
  border="0" />
</a>
<br />

          </td>
          <td valign="top" class="sitebuffer">
    <br />
<br /><center>
<br />
        <b>Level 1(the idiot test)</b>
        </center><br /><br />
        This level is what we call "The Idiot Test", if you can't complete it, don't give up on learning all you can, but, don't go be
hated/made fun of. Enter the password and you can continue. <br /><br />
        <!-- the first few levels are extremely easy: password is 8f74c7a5 -->
        <center><b>password:</b><br /><form action="/missions/basic/1/index.php" method="post"><input type="password" name="password"
                <center><table border="0" width="80%" cellspacing="0" cellpadding="0">
                <tr>
                        <td class="dark-td">&nbsp;<b>Help!</b></td>
                </tr>
                        <td class="light-td">&nbsp;If you have no idea what to do, you must <a href="http://www.w3schools.com/HTML">le
            <br /></td>
                </tr>
        </table></center></td>
</td>
        </tr>
      </table></td>
    </tr>
 <tr>
      <td class="sitebottomheader"><img src="https://data.htscdn.org/themes/Dark/images/hts_bottomheadern.jpg" alt="End Footer" width=
    </tr>
  </table>
  <br />
<div align="center" style="font-family:Verdana, Arial, Helvetica, sans-serif; font-size:10px; color:#CCCCCC">HackThisSite is the colle
href="http://hts.io/x/http://creativecommons.org/licenses/by-nc/3.0/" target="_new">CC BY-NC</a> license.<br />
We ask that you inform us upon sharing or distributing.<br /><br />
<sub>Page Generated: Fri, 31 May 2024 03:27:17 +0000<br />Web Node: www03 | Page Gen: 0.025s | DB: 13q<br />Current Code Revision: <a
 (Sun, 22 May 2016 20:29:51 +0000)</a></sub><br />
</div>
</div>
<div align="center">
  <p>
   <a target="_new" href="http://hts.io/x/http://creativecommons.org/licenses/by-nc/3.0/"><img src="https://data.htscdn.org/images/cc_
   <a target="_new" href="http://hts.io/x/http://validator.w3.org/check?uri=referer"><img src="https://data.htscdn.org/images/xhtml10
   <a target="_new" href="http://hts.io/x/http://jigsaw.w3.org/css-validator/check/referer"><img src="https://data.htscdn.org/images/
   <a target="_new" href="http://hts.io/x/http://www.php.net/"> <img src="https://data.htscdn.org/images/phppow.gif" width="80" height
   <!--<a href="http://www.linux.com/"> <img src="../../images/linux2.gif" width="80" height="15" border="0" alt="" /></a>-->
   <a target="_new" href="http://hts.io/x/http://www.freebsd.org/"> <img src="https://data.htscdn.org/images/freebsd.png" width="80" h
```

Recommendations

Before application is placed into a production environment the source code should be scrubbed for any information that can identify sensitive information, from passwords, notes, txt files and other related development documentation of the application.

Level 2 (Password Scripting/ Brute Force)

Action

Security of application related to poor password script application was identified. Due to the failure to upload the password file there was no requirement to meet any password requirements.

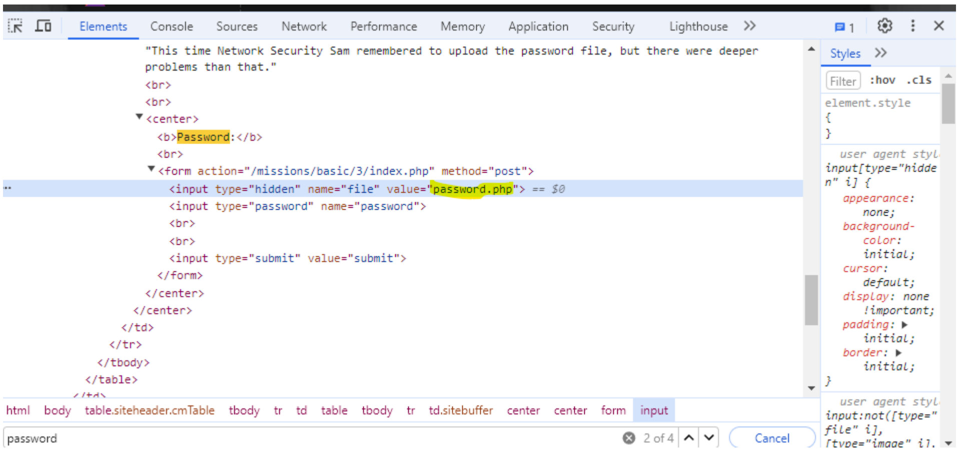
Recommendations

Improve password intricacy and length requirements, multi factor authentication and SHA-256 hashing algorithms.

Level 3 (Password Scripting/ PHP file access)

Vulnerability

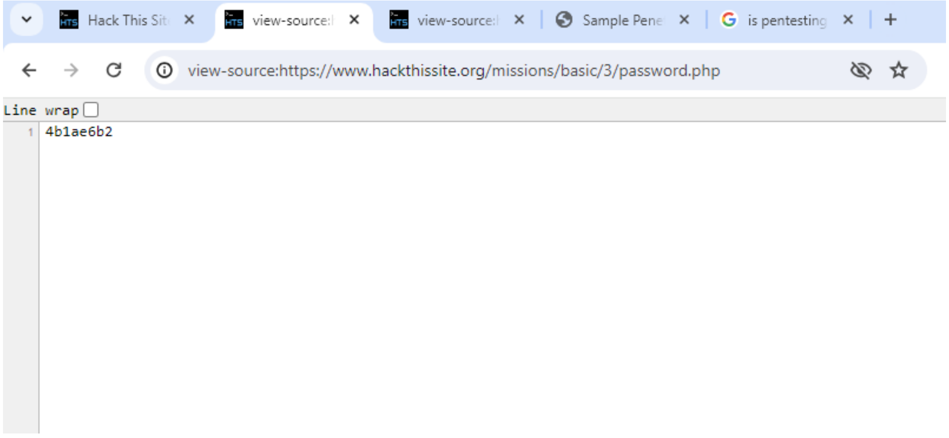
Able to identify a readable file (php) in the front end of the application gain the password.



```
"This time Network Security Sam remembered to upload the password file, but there were deeper
problems than that."
<br>
<br>
▼<center>
  <b>Password:</b>
  <br>
  ▼<form action="/missions/basic/3/index.php" method="post">
      <input type="hidden" name="file" value="password.php"> == $0
      <input type="password" name="password">
      <br>
      <br>
      <input type="submit" value="submit">
   </form>
  </center>
 </center>
</td>
</tr>
</tbody>
</table>
```

```
element.style {
}
user agent styl
input[type="hidden" i] {
  appearance: none;
  background-color: initial;
  cursor: default;
  display: none !important;
  padding: ▸ initial;
  border: ▸ initial;
}
user agent styl
input:not([type="file" i], [type="image" i],
```

password



view-source:https://www.hackthissite.org/missions/basic/3/password.php

Line wrap □

```
4b1ae6b2
```

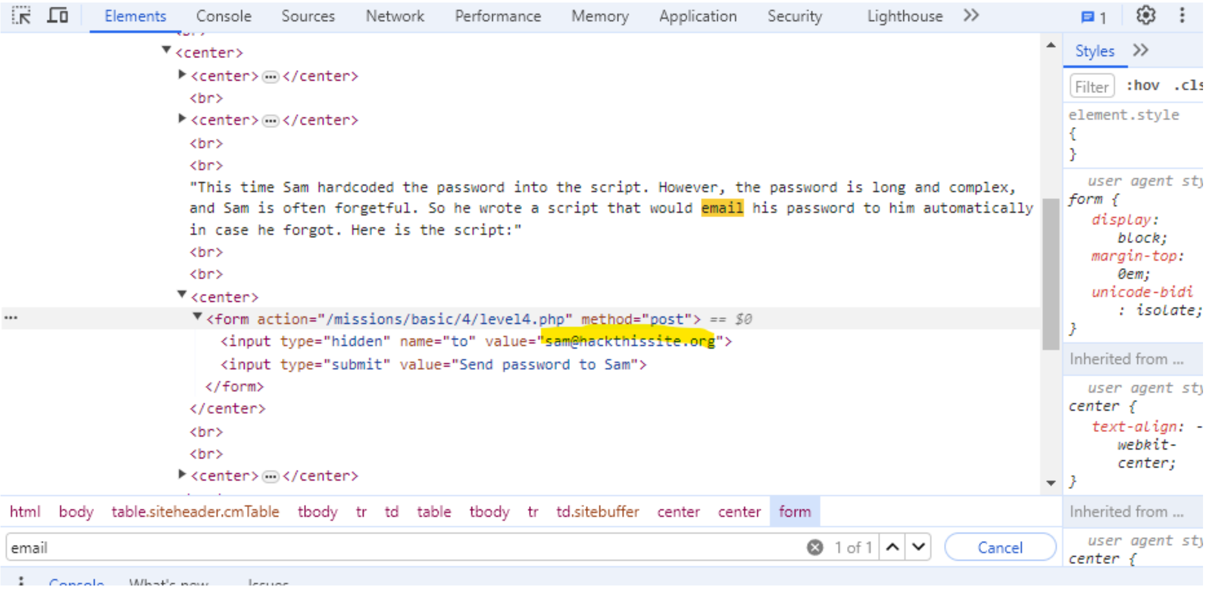
Recommendations

Secure readable files in directory via encryption that cannot be read by unauthorised users.

Level 4 breach (HTML coding)

Vulnerability

Able to identify notes, email and redirect to assist with password access via the source code.



```
▼<center>
  ▶<center>…</center>
  <br>
  ▶<center>…</center>
  <br>
  <br>
  "This time Sam hardcoded the password into the script. However, the password is long and complex,
  and Sam is often forgetful. So he wrote a script that would email his password to him automatically
  in case he forgot. Here is the script:"
  <br>
  <br>
  ▼<center>
    ▼<form action="/missions/basic/4/level4.php" method="post"> == $0
        <input type="hidden" name="to" value="sam@hackthissite.org">
        <input type="submit" value="Send password to Sam">
      </form>
  </center>
  <br>
  <br>
  ▶<center>…</center>
```

```
element.style
{
}
user agent sty
form {
  display:
    block;
  margin-top:
    0em;
  unicode-bidi
    : isolate;
}
user agent sty
center {
  text-align: -
    webkit-
    center;
}
user agent sty
center {
```

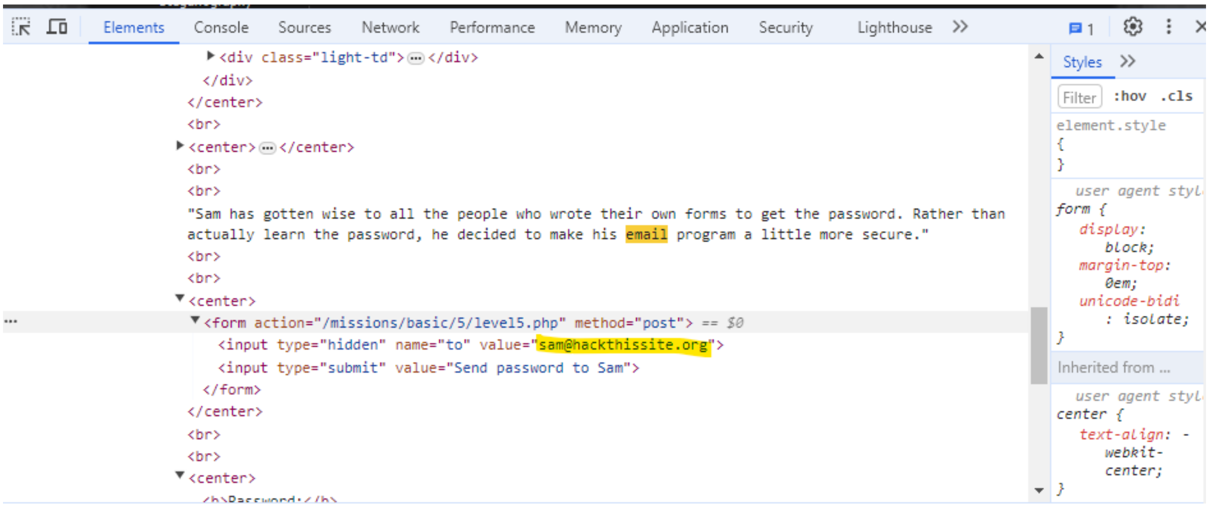
Recommendations

Remove hard code in HTML Java web applications use PHP as it is on the server side of the application and end users are unable to access it directly.

Level 5 breach (HTML tampering)

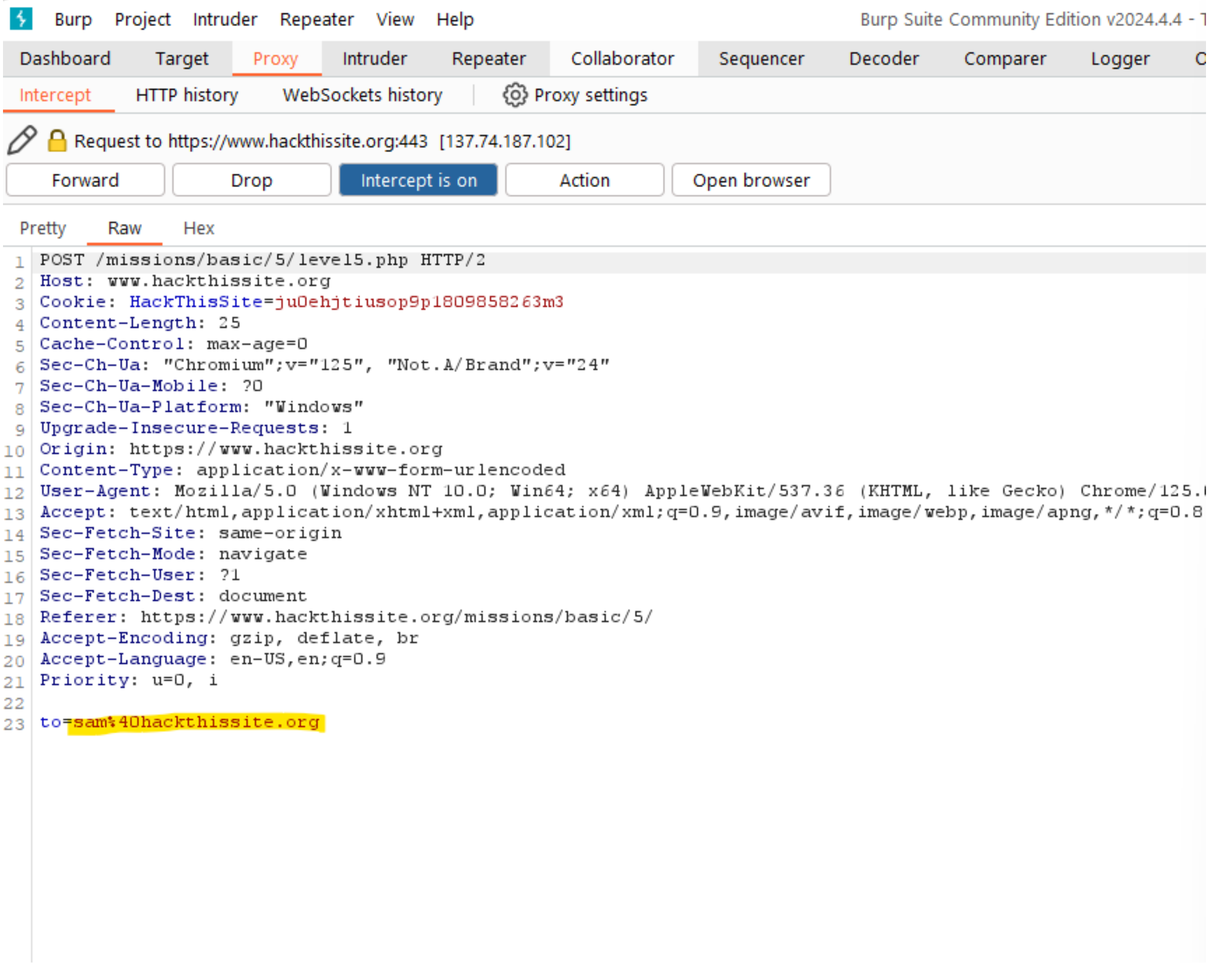
Vulnerability

The web application raw HTML code has information related to the user and is vulnerable to malicious actors who use the information for further.



```
    <div class="light-td">…</div>
    </div>
  </center>
  <br>
  <center>…</center>
  <br>
  <br>
  "Sam has gotten wise to all the people who wrote their own forms to get the password. Rather than
  actually learn the password, he decided to make his email program a little more secure."
  <br>
  <br>
  <center>
    <form action="/missions/basic/5/level5.php" method="post"> == $0
      <input type="hidden" name="to" value="sam@hackthissite.org">
      <input type="submit" value="Send password to Sam">
    </form>
  </center>
  <br>
  <br>
  <center>
```

```
element.style
{
}
user agent styl
form {
  display:
    block;
  margin-top:
    0em;
  unicode-bidi
    : isolate;
}
Inherited from ...
user agent styl
center {
  text-align: -
    webkit-
    center;
}
```



Request to https://www.hackthissite.org:443 [137.74.187.102]

```
POST /missions/basic/5/level5.php HTTP/2
Host: www.hackthissite.org
Cookie: HackThisSite=ju0ehjtiusop9p18098582 63m3
Content-Length: 25
Cache-Control: max-age=0
Sec-Ch-Ua: "Chromium";v="125", "Not.A/Brand";v="24"
Sec-Ch-Ua-Mobile: ?0
Sec-Ch-Ua-Platform: "Windows"
Upgrade-Insecure-Requests: 1
Origin: https://www.hackthissite.org
Content-Type: application/x-www-form-urlencoded
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/125.
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8
Sec-Fetch-Site: same-origin
Sec-Fetch-Mode: navigate
Sec-Fetch-User: ?1
Sec-Fetch-Dest: document
Referer: https://www.hackthissite.org/missions/basic/5/
Accept-Encoding: gzip, deflate, br
Accept-Language: en-US,en;q=0.9
Priority: u=0, i

to=sam%40hackthissite.org
```

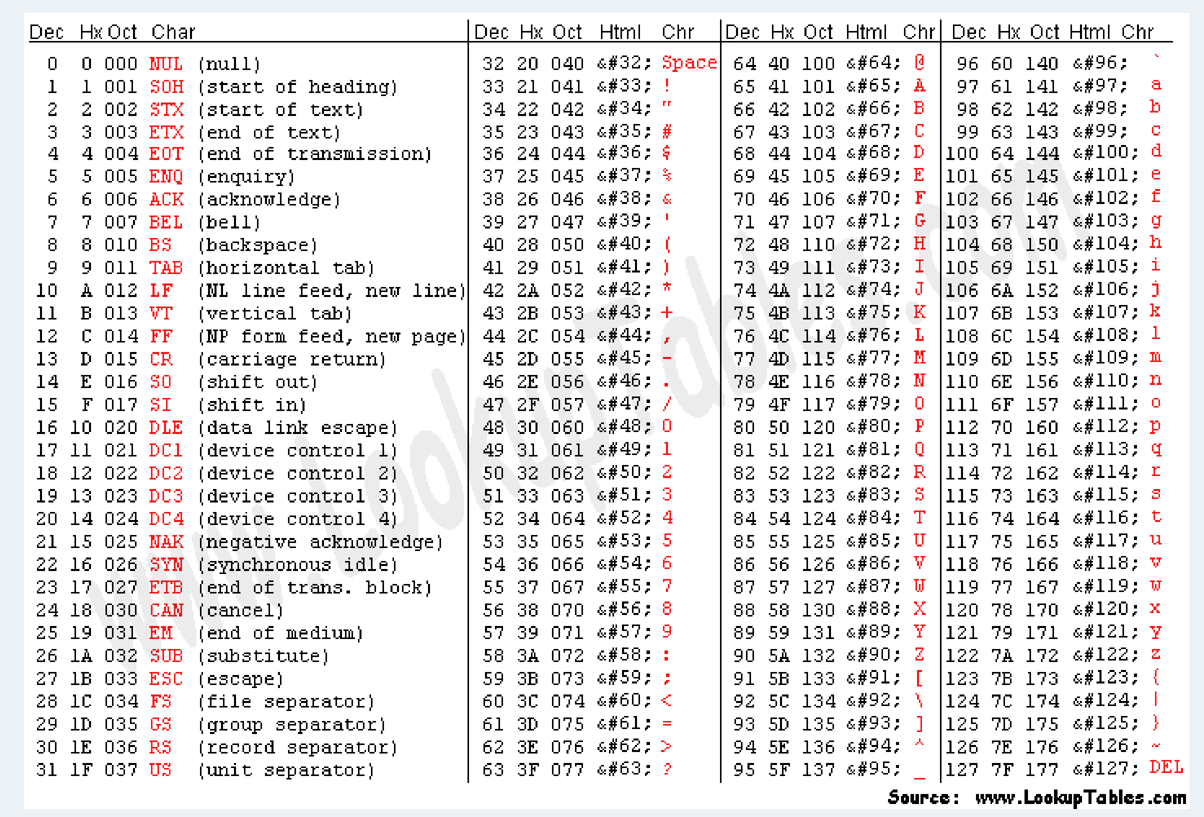
Recommendations

The application should be scrub for any sensitive data on the client side. hard code sensitive information as client-side code as vulnerable to attack. Server side. To further protect passwords held in within a directory, the use of .htaccess can assist with the protection of sensitive data.

Level 6 (Cryptography)

Vulnerability

The password was encrypted using a type of Caesar cypher. In this case a group of letters, numbers and symbols was used. The cypher complexity was then increased with the codes first identifier being the same then each following identifier was increase by 1 and then incrementally increased. This type of encryption is very simplistic and easy for attackers to break and gain access to the application.



Dec	Hx	Oct	Char	Dec	Hx	Oct	Html	Chr	Dec	Hx	Oct	Html	Chr	Dec	Hx	Oct	Html	Chr
0	0	000	NUL (null)	32	20	040	&#32;	Space	64	40	100	&#64;	@	96	60	140	&#96;	`
1	1	001	SOH (start of heading)	33	21	041	&#33;	!	65	41	101	&#65;	A	97	61	141	&#97;	a
2	2	002	STX (start of text)	34	22	042	&#34;	"	66	42	102	&#66;	B	98	62	142	&#98;	b
3	3	003	ETX (end of text)	35	23	043	&#35;	#	67	43	103	&#67;	C	99	63	143	&#99;	c
4	4	004	EOT (end of transmission)	36	24	044	&#36;	$	68	44	104	&#68;	D	100	64	144	&#100;	d
5	5	005	ENQ (enquiry)	37	25	045	&#37;	%	69	45	105	&#69;	E	101	65	145	&#101;	e
6	6	006	ACK (acknowledge)	38	26	046	&#38;	&	70	46	106	&#70;	F	102	66	146	&#102;	f
7	7	007	BEL (bell)	39	27	047	&#39;	'	71	47	107	&#71;	G	103	67	147	&#103;	g
8	8	010	BS (backspace)	40	28	050	&#40;	(	72	48	110	&#72;	H	104	68	150	&#104;	h
9	9	011	TAB (horizontal tab)	41	29	051	&#41;	)	73	49	111	&#73;	I	105	69	151	&#105;	i
10	A	012	LF (NL line feed, new line)	42	2A	052	&#42;	*	74	4A	112	&#74;	J	106	6A	152	&#106;	j
11	B	013	VT (vertical tab)	43	2B	053	&#43;	+	75	4B	113	&#75;	K	107	6B	153	&#107;	k
12	C	014	FF (NP form feed, new page)	44	2C	054	&#44;	,	76	4C	114	&#76;	L	108	6C	154	&#108;	l
13	D	015	CR (carriage return)	45	2D	055	&#45;	-	77	4D	115	&#77;	M	109	6D	155	&#109;	m
14	E	016	SO (shift out)	46	2E	056	&#46;	.	78	4E	116	&#78;	N	110	6E	156	&#110;	n
15	F	017	SI (shift in)	47	2F	057	&#47;	/	79	4F	117	&#79;	O	111	6F	157	&#111;	o
16	10	020	DLE (data link escape)	48	30	060	&#48;	0	80	50	120	&#80;	P	112	70	160	&#112;	p
17	11	021	DC1 (device control 1)	49	31	061	&#49;	1	81	51	121	&#81;	Q	113	71	161	&#113;	q
18	12	022	DC2 (device control 2)	50	32	062	&#50;	2	82	52	122	&#82;	R	114	72	162	&#114;	r
19	13	023	DC3 (device control 3)	51	33	063	&#51;	3	83	53	123	&#83;	S	115	73	163	&#115;	s
20	14	024	DC4 (device control 4)	52	34	064	&#52;	4	84	54	124	&#84;	T	116	74	164	&#116;	t
21	15	025	NAK (negative acknowledge)	53	35	065	&#53;	5	85	55	125	&#85;	U	117	75	165	&#117;	u
22	16	026	SYN (synchronous idle)	54	36	066	&#54;	6	86	56	126	&#86;	V	118	76	166	&#118;	v
23	17	027	ETB (end of trans. block)	55	37	067	&#55;	7	87	57	127	&#87;	W	119	77	167	&#119;	w
24	18	030	CAN (cancel)	56	38	070	&#56;	8	88	58	130	&#88;	X	120	78	170	&#120;	x
25	19	031	EM (end of medium)	57	39	071	&#57;	9	89	59	131	&#89;	Y	121	79	171	&#121;	y
26	1A	032	SUB (substitute)	58	3A	072	&#58;	:	90	5A	132	&#90;	Z	122	7A	172	&#122;	z
27	1B	033	ESC (escape)	59	3B	073	&#59;	;	91	5B	133	&#91;	[	123	7B	173	&#123;	{
28	1C	034	FS (file separator)	60	3C	074	&#60;	<	92	5C	134	&#92;	\	124	7C	174	&#124;	|
29	1D	035	GS (group separator)	61	3D	075	&#61;	=	93	5D	135	&#93;	]	125	7D	175	&#125;	}
30	1E	036	RS (record separator)	62	3E	076	&#62;	>	94	5E	136	&#94;	^	126	7E	176	&#126;	~
31	1F	037	US (unit separator)	63	3F	077	&#63;	?	95	5F	137	&#95;	_	127	7F	177	&#127;	DEL

Source: www.LookupTables.com

Recommendations

Use of modern encryption –

Single Secret Key – encryption and decryption and the same key is shared between both the sender and receiver.

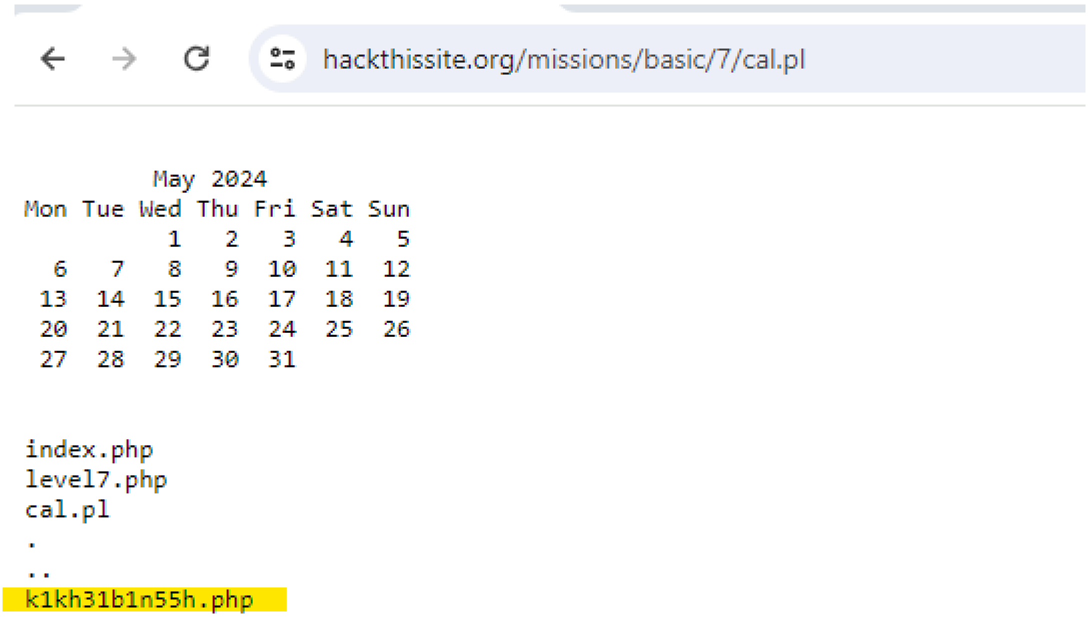
Hashing- using SHA-2 AES 256 (Advanced encryption standards)

The use of security password protection tool such as PageCrypt, or StatiCrypt.

Task 7 breach (Remote code execution)

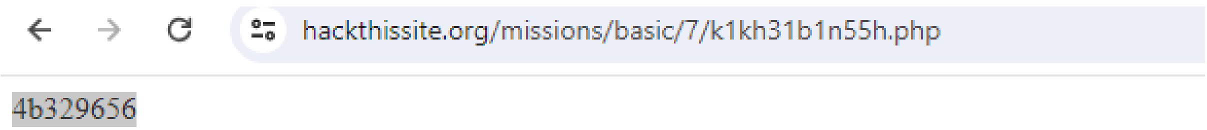
Vulnerability

User can access a file contain data in this case the password to access the system using the special characters semi-colon which are used to in code command lines. The attacker can brute force this vulnerability and access sensitive file and or command lines in the server.



```
         May 2024
Mon Tue Wed Thu Fri Sat Sun
          1   2   3   4   5
  6   7   8   9  10  11  12
 13  14  15  16  17  18  19
 20  21  22  23  24  25  26
 27  28  29  30  31


index.php
level7.php
cal.pl
.
..
k1kh31b1n55h.php
```



```
4b329656
```

Recommendations

Sanitisation of identifiable items such as cookies

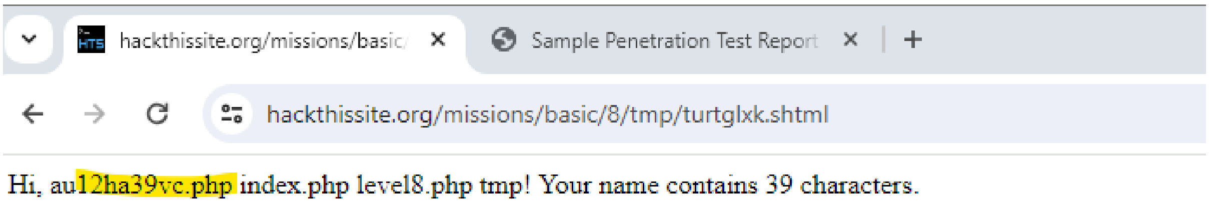
Remove direct injection type commands from the user input form.

Use of Firewalls to stop SQL injections

Task 8 (web form security/SSI injection)

Vulnerability

The use of the following command **<!--#exec cmd="ls ../" -->** within the application sign-in form allows malicious actors access to the server side of the application.


hackthissite.org/missions/basic/
Sample Penetration Test Report
hackthissite.org/missions/basic/8/tmp/turtglxk.shtml

Hi, au12ha39vc.php index.php level8.php tmp! Your name contains 39 characters.

Recommendations

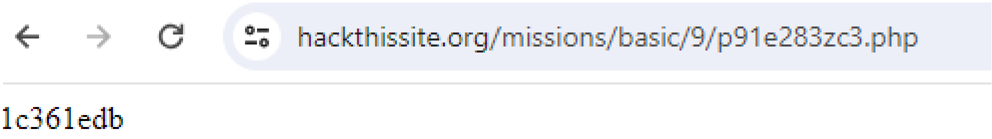
Clean treat untrusted data

To assist refer to OWASP for further advice (https://owasp.org/www-project-top-ten/)

Task 9 (Script; server-side attack)

Vulnerability

The attacker was able to move through the directories to obtain access to the server the following script was executed on through server 8 **<!--#exec cmd="ls ../../9" -->** to gain the hard coded password for serve 9 via a .php file.



hackthissite.org/missions/basic/9/p91e283zc3.php

```
1c361edb
```

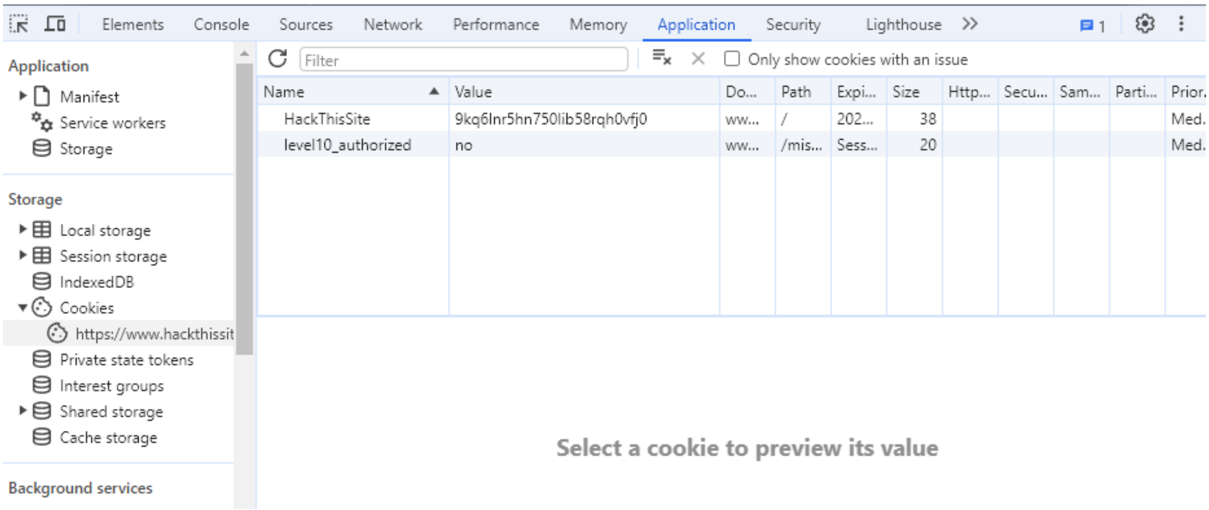
Recommendations

To ensure that user input is sanitised and the removal of hard coded password in .php files

Task 10 (Java Script)

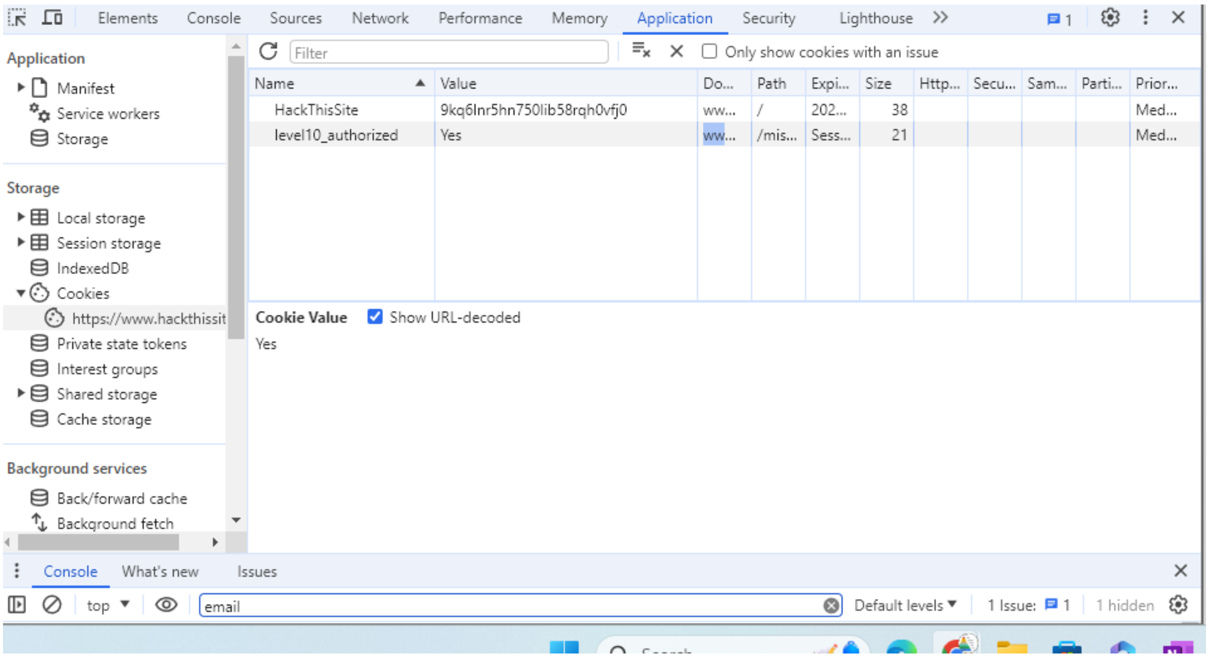
Vulnerability

The malicious actor was able to access to the application via the web dev ops tool and hijack the session cookie that related to the login session as the login session had not been cleared.



Name	Value	Do...	Path	Expi...	Size	Http...	Secu...	Sam...	Parti...	Prior.
HackThisSite	9kq6lnr5hn750lib58rqh0vfj0	ww...	/	202...	38					Med.
level10_authorized	no	ww...	/mis...	Sess...	20					Med.

Select a cookie to preview its value



Name	Value	Do...	Path	Expi...	Size	Http...	Secu...	Sam...	Parti...	Prior...
HackThisSite	9kq6lnr5hn750lib58rqh0vfj0	ww...	/	202...	38					Med...
level10_authorized	Yes	ww...	/mis...	Sess...	21					Med...

Cookie Value ☑ Show URL-decoded

Yes

email

Recommendations

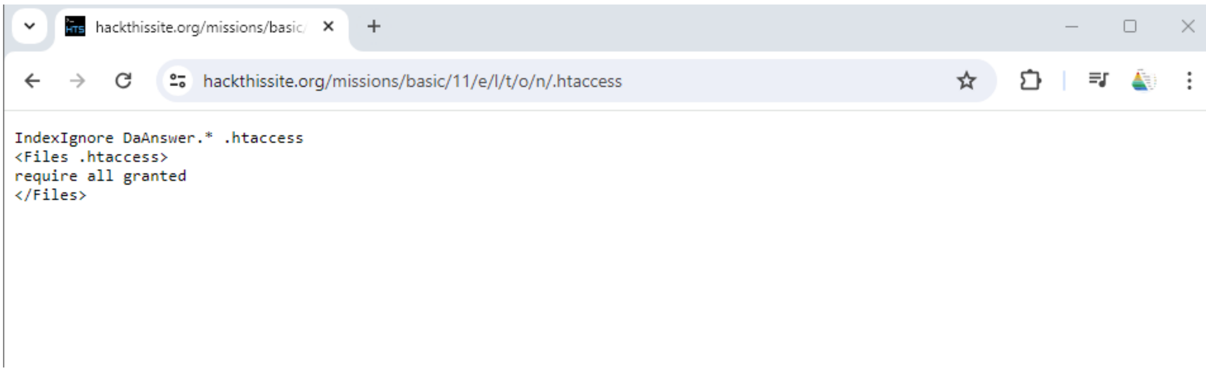
Ensure that cookies are cleared after the browser is closed. Maintain naming conventions that do not reflect the actual use of the session and ensure there is encryption of the session.

Maintain control accessing a session by ensuring that login details are authenticated before the user receives a session (cookie)

Task 11 (Directory Listing)

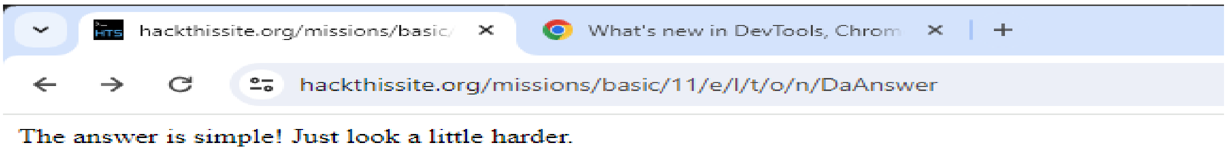
Vulnerability

The login form and password although hidden from direct access were still accessible on the user side of the application.



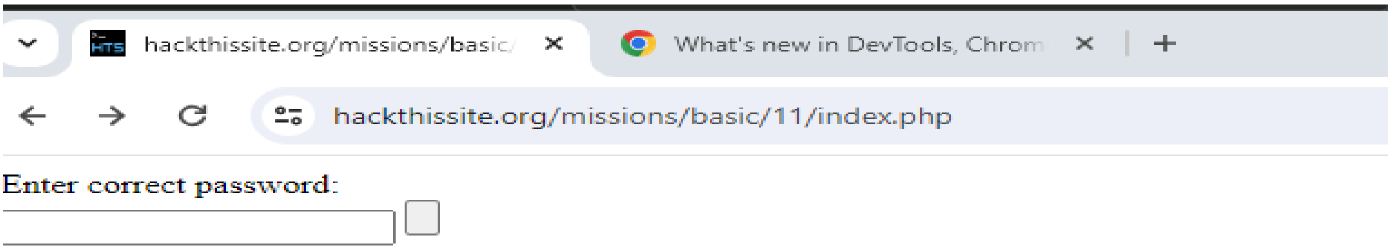
hackthissite.org/missions/basic/11/e/l/t/o/n/.htaccess

```
IndexIgnore DaAnswer.* .htaccess
<Files .htaccess>
require all granted
</Files>
```


hackthissite.org/missions/basic/
What's new in DevTools, Chrom

hackthissite.org/missions/basic/11/e/l/t/o/n/DaAnswer

The answer is simple! Just look a little harder.


hackthissite.org/missions/basic/
What's new in DevTools, Chrom

hackthissite.org/missions/basic/11/index.php

Enter correct password:

Recommendations

The Directory listing should not be enabled and password authentication should be in the main server activity file.

Conclusion

During the investigation, activities were conducted to test the security of password structure and it was found that there were numerous procedural issues lacking industry standards. There website contain vulnerabilities related to Hypertext markup language (HTML) source code via the user end of the application. It is suggested that training enforcement of coding and security procedures be conducted.