**Level 1(the idiot test)**

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 begging to someone else for the answer, thats one way to get you hated/made fun of. Enter the password and you can continue.

The first stage is logging in and the is no user field which basically is a html code and upon inspecting the page source and using find option you can find the password for the page.

**Level 2**

Network Security Sam set up a password protection script. He made it load the real password from an unencrypted text file and compare it to the password the user enters. However, he neglected to upload the password file...

The passwords are nonexistent and nothing to compare as sam forgot to upload the passwords file. So just a simple enter button logs you in.

**Level 3**

This time Network Security Sam remembered to upload the password file, but there were deeper problems than that.

The page directory was not complete in the url. If you are familiar with directory transversal you can navigate to the required page. The password.php has the password and inserting it in the url gives the stored password.

**Level 4**

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:

Using the inspect element I inspected the e-mail and the replaced sams email with mine to get the password. The simple inspect element can replace the email address. Its

**Level 5**

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.

Using the inspect element I inspected the e-mail and the replaced sams email with mine to get the password. The input value is not secure. But instead of submitting just clicked send password to sam.

**Level 6  
  
Network Security Sam has encrypted his password. The encryption system is publically available and can be accessed with this form:**

**Please enter a string to have it encrypted.Top of Form**

**Bottom of Form**

**You have recovered his encrypted password. It is:  
  
b5h48>hk  
  
Decrypt the password and enter it below to advance to the next level.**

For b5h48>hk b4f149bd is the decryption. As abcdefgh encrypts to

Acegikmo which is n n+1 n+3 n+5 and so on… follow the Ascii table for decryption since its n n-1 n-3 n-5 and so on.

**Level 7**

This time Network Security sam has saved the unencrypted level7 password in an obscurely named file saved in this very directory.  
  
In other unrelated news, Sam has set up a script that returns the output from the UNIX cal command. Here is the script:  
  
Enter the year you wish to view and hit 'view'.

Just enter the year followed by semi colon and ls command to list all in unix. Youll find all the files in the directory eg: 2024; ls

All you need to know is linux commands to check the directory.

**Level 8**

Sam remains confident that an obscured password file is still the best idea, but he screwed up with the calendar program. Sam has saved the unencrypted password file in /var/www/hackthissite.org/html/missions/basic/8/  
  
However, Sam's young daughter Stephanie has just learned to program in PHP. She's talented for her age, but she knows nothing about security. She recently learned about saving files, and she wrote a script to demonstrate her ability.

In the Enter your name field use a SQL injection to gain access to the files and directories. Service side injection to execute a code <!--#exec cmd="ls ../" --> The ../ goes back in directory. Look for the .php file and replace it In the url to find the password.

**Lvl 9**

Network Security Sam is going down with the ship - he's determined to keep obscuring the password file, no matter how many times people manage to recover it. This time the file is saved in /var/www/hackthissite.org/html/missions/basic/9/.  
  
In the last level, however, in my attempt to limit people to using server side includes to display the directory listing to level 8 only, I have mistakenly screwed up somewhere.. there is a way to get the obscured level 9 password. See if you can figure out how...  
  
This level seems a lot trickier then it actually is, and it helps to have an understanding of how the script validates the user's input. The script finds the first occurance of '<--', and looks to see what follows directly after it.

Server side injection to execute a code <!--#exec cmd="ls /var/www/hackthissite.org/html/missions/basic/9/" --> replace the ../ with the path they already provided but the problem here is no field to submit the script. Go back to the lvl8 and do the same exact way and find the php file and copy the name and comeback to lvl 9 and paste it at the end in the url.

Lvl 10 using cookie

javascript:alert(document.cookie);

javascript:alert(document.cookie="level10\_authorized=yes");

lvl 11

Nothing to enter just a webpage with few lines of Text. It changes every reload.

look up apache hiding files .htaccess

every reload gave a different song.

look whats common in all the songs that the page is displaying.

It is Elton John

Tried Elton john and all possible combininations in the url

E worked and then gave a little hint. And then followed all the directories and it said the password is somewhere look a little harder. The password page should be somewhere in the directories so I navigated back to directory 11 and tried index.php and boom the login page showed up and the hint which is somewhere is the password.

Overall there are numerous Vulnerabilities that are exploited here in this exercise. Security is always a concern. There is a need for pen testing to find the vulnerabilities in any organization.