

# VULNERABILITY ASSESSMENT AND PENETRATION TESTING REPORT

## Table of contents

1. Executive Summary.....	2
2. Methodology.....	3
● Information gathering.....	3
● Findings and Exploitation.....	4
Testhtml5.vulnweb.com.....	4
Testphp.vulnweb.com.....	6
Testasp.vulnweb.com.....	11
3. Risk Assessment.....	12
Testhtml5.vulnweb.com.....	12
Testphp.vulnweb.com.....	14
Testasp.vulnweb.com.....	15
4. Mutual Findings and Recommendations.....	17
5. Conclusion.....	18

## ◆ EXECUTIVE SUMMARY

The purpose of this vulnerability scan is to gather data on vulnweb site for testing acunetix vulnerability scanner. The audit was conducted on three of the live host that was present at the site.

Of the 3 hosts identified, had a wide variety of services where active and scanned. Critical, High, medium and low severity vulnerabilities were found all three hosts. The details are as follows.

### ● testhtml5.vulnweb.com

Vulnerability Risk	Unique Count
Critical severity vulnerability	1
High severity vulnerability	4
Medium severity vulnerability	60
Low severity vulnerability	7

The vulnerabilities found on this host was consist of outdated and unsupported Unix operating system. System was no longer supported which means the system is not applicable for future security patches. As a result, it is likely to contain security vulnerabilities.

Out of 13 services detected on the host most of the was lacking support or critical patches from past which made them vulnerable to basic attacks. The host was also vulnerable to click jacking attacks as the host did not set an X-Frame-Options response header or a Content-Security-Policy. This could possibly lead the site to UI redress attack or click jacking as mentioned earlier.

Lack of proper validation and sanitation of the input field made it possible to inject SQL queries into the filed compromising admin account in the site. Basic Nmap NSE script scan could reveal a plethora of vulnerabilities which show cases the SSL/TLS protocol detection. The service encrypts traffic using a protocol with know weakness.

It is our recommendation that immediate action be taken to resolve these vulnerabilities by applying patches and adjusting system configurations as necessary. In addition, a patch and configuration management process should be implemented to continually assess system risk level as vulnerabilities are discovered. This will ensure relevant security patches and configurations are applied in a timely manner.

### ● testphp.vulnweb.com

Vulnerability Risk	Unique Count
Critical severity vulnerability	3
High severity vulnerability	23
Medium severity vulnerabilities	77
Low severity vulnerabilities	8

This particular host had Apache, PHP and MySQL services running them. The PHP version 5.3.x < 5.3.15 had multiple vulnerabilities detected. According to its banner, An unspecified overflow vulnerability exists in the function '\_php\_stream\_scandir' in the file 'main/streams/streams.c'. (CVE-2012-2688). An unspecified error exists that can allow the 'open\_basedir' constraint to be bypassed.(CVE-2012-3365)

This particular version of PHP is also unsupported, hence there can be no new security patches be installed. As a result, its likely to contain security vulnerabilities.

By providing specially crafted parameters to CGIs, Nessus was able to get an error from the underlying database. This error suggests that the CGI is affected by a SQL injection vulnerability. An attacker may exploit this flaw to bypass authentication, read confidential data, modify the remote database, or even take control of the remote operating system.

Tested the host with SQLmap and found that 4 links in the site was prone to SQL injection and was able to view edit and manipulate the databases. due to lack of validation and sanitation, XSS attacks was also possible. The site has also exposed the sensitive data which was the initial login credentials which was username was "test" and password was "test". Security configuration also paved the way for Insecure Direct Object reference attacks. Which helped to buy products in site without paying a single penny.

#### ● testasp.vulnweb.com

Vulnerability Risk	Unique Count
Critical severity vulnerabilities	0
High severity vulnerabilities	1
Medium severity vulnerabilities	15
Low severity vulnerabilities	3

This host had IIS, ASP and Microsoft SQL service running. Initial scan shows wide variety of vulnerabilities. As we can see this particular host in Windows bases. The host is likely vulnerable to CSRF(Cross Site Request Forgery), with was show in initial nmap NSE scan. This host is prone to coolie injection attacks, at least one script that fails to adequately sanitize request strings with malicious JavaScript. Leveraging this issue, the attacker may able to inject arbitrary cookies. Depending on the structure of the web application, it may be possible to "session fixation" attack using this mechanism. During the crawl with SQLmap i was able to detect two links where sql injection was possible, and also view and manipulate the database tables.

The remote version of the Remote Desktop Protocol Server (Terminal Service) is vulnerable to a man-in-the-middle (MiTM) attack. The RDP client makes no effort to validate the identity of the server when setting up encryption. An attacker with the ability to intercept traffic from the RDP server can establish encryption with the client and server without being detected. A MiTM attack of this nature would allow the attacker to obtain any sensitive information transmitted, including authentication credentials.

## ◆ METHODOLOGY

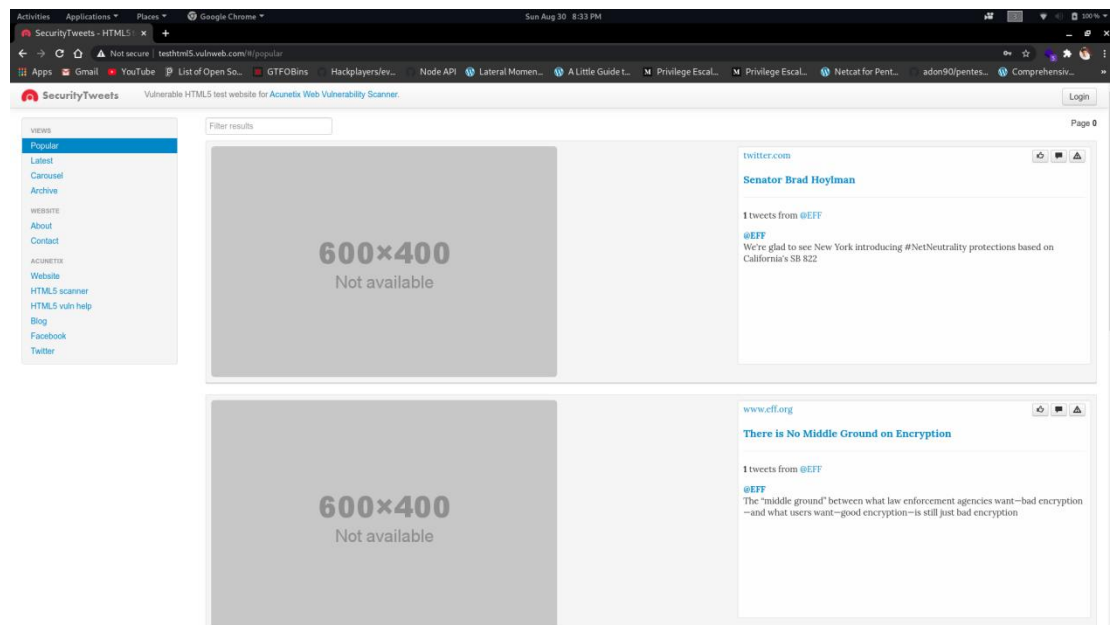
### 1. INFORMATION GATHERING.

Enumeration was done with help of certain tools. Nmap was used to understand the initial status of the host. Nmap detected the services on the host which helped to understand basics of the host. As all three host was deployed to be web server, also did some directory busting using gobuster. Nikto was used to understand the insight into the web server. Did some manual recon over the sites itself where i discovered multiple SQL injections, IDOR attacks and Cross-site scripting attacks. Finally i used Nessus Essentials on each of the three hosts to completely understand and give a wider scope to further Research and exploitation.

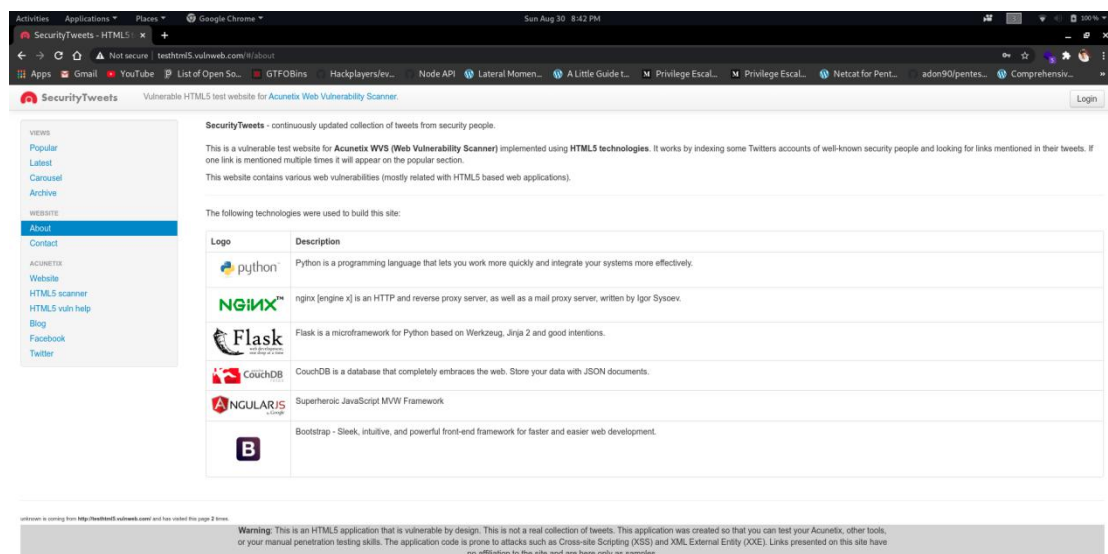
## 2. FINDINGS AND EXPLOITATION

### ● Testhtml5.vulnweb.com

According to the initial scan results, there are many outdated and unsupported services running on the host. What is interesting is, there is web site solely running on the host which is build as the name suggests HTML5. the site does every basic things a site suppose to do. This particular site gives a reference to the tweets in twitter. After crawling through the site found a search bar and a login page for users to login.



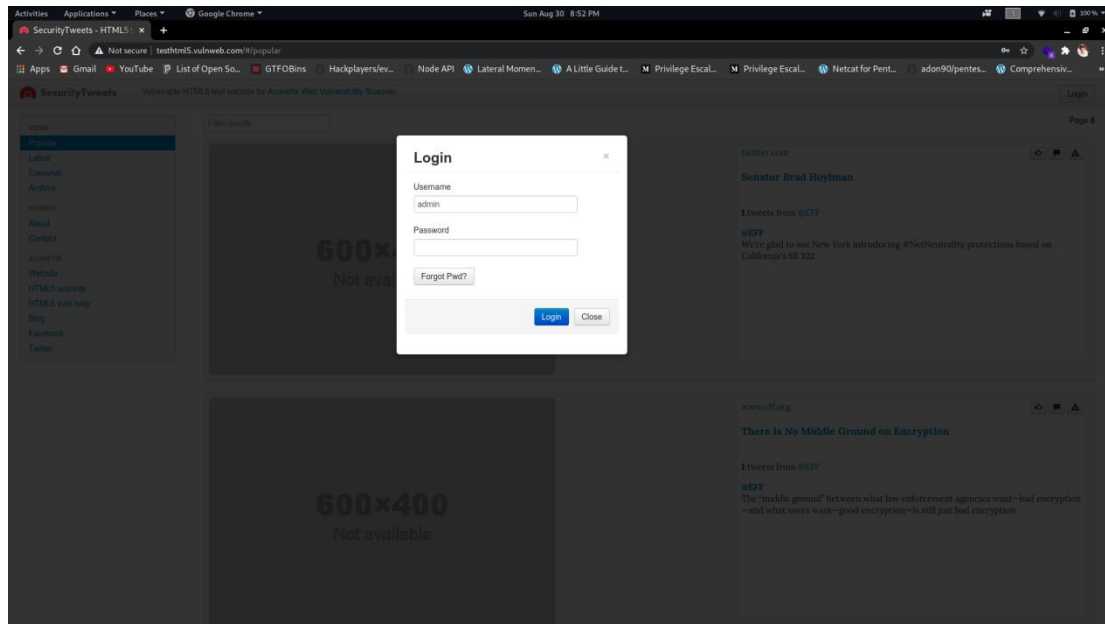
Another interesting finding was, there was a tab for the site which was called About. It particularly show the technologies used to create the site. Even without Recon with Third party tools we could understand on what was the site build on. That is partially sensitive information to give out, in right hands that information could be lethal.



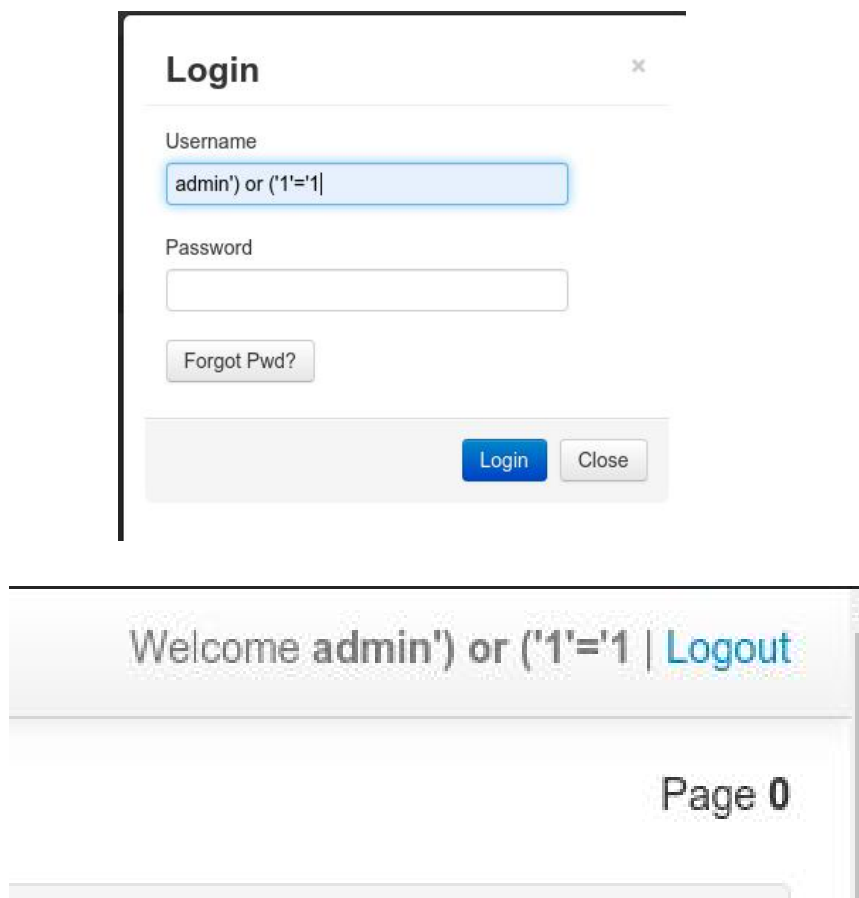
© Acunetix Ltd. 2019

twitter.github.io/bootstrap

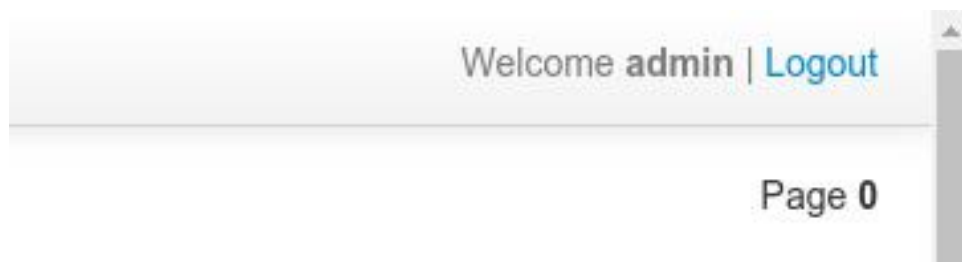
Interestingly enough, when we click the login button, we are presented with a basic login form, which has the username filled with default user with “admin”. Again, we found that admin is a user and by what the username suggests it is Administrator of the site. This is a exposed sensitive information. Without any third party tools we are able to find a valid user for the site.



When we inject the fields with basic SQL queries, outputs is rather interesting.



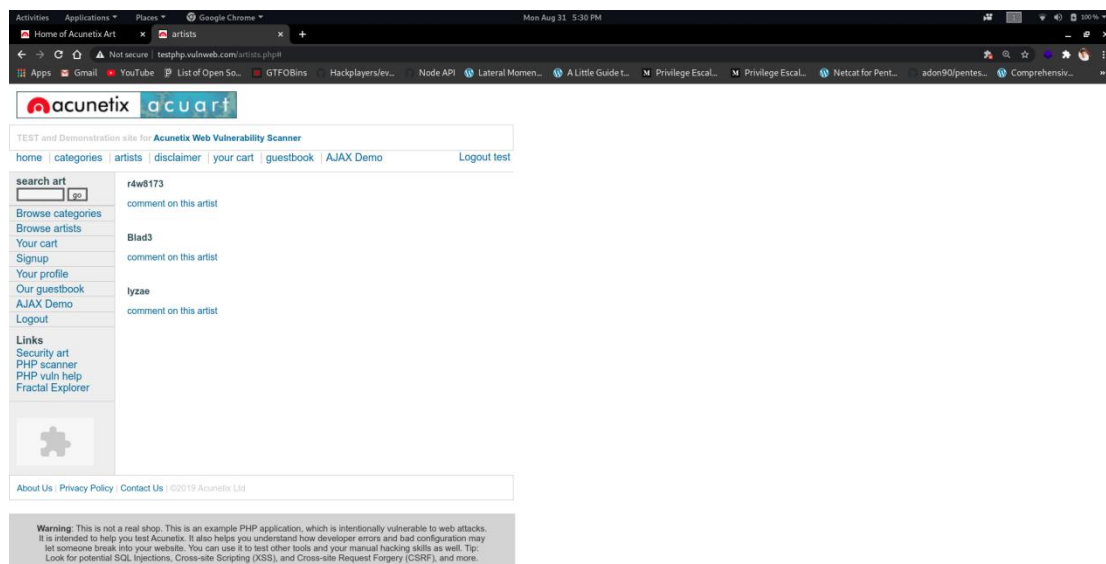
Even though there isn't any such account i was able to log in to the site. When i tried the same query which is "admin') or ('1'='1 " in the password field and admin itself in username field i was able to log into admin account without any tools.



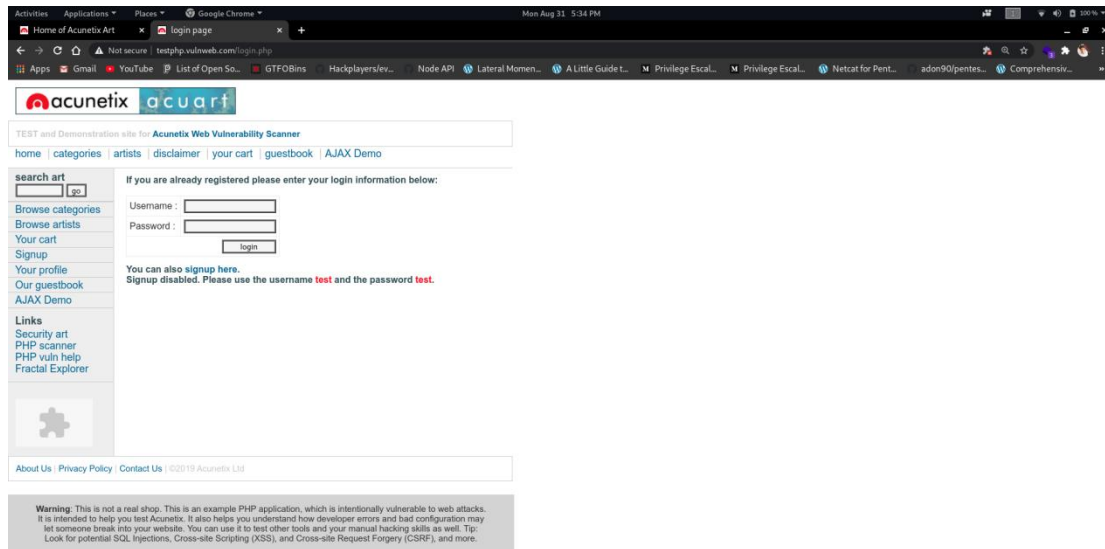
the site had gaping security flaws that could be exploited and give out the root credentials or at least expose the sensitive data which paves the way for ultimate compromise. It is recommended that the developers use proper validation and sanitation in all of the fields present in the site. Users are able to type in queries and get the admin account.

### ● Testphp.vulnweb.com

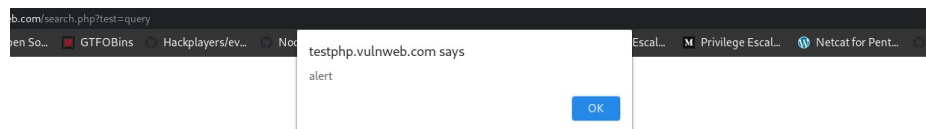
In the initial scans we were able to get a basic understanding of the structure of the host. It is a web server and a site hosted on the same. The website runs on Apache server, PHP on the back end and MySQL on the side of database. As the previous host, this host also has unsupported Operating system which is flagged as a critical threat to the system. According to the version number the Linux kernel is no longer supported. This will affect the integrity of the host. OS is something that is considered to be a base to build something upon. Lack of support implies that no new security patches for the product will be released by the vendor. As a result it is likely to contain security vulnerabilities. With some observation on the site, it was found that the site was programmed for online sales of various kinds of art named "acuart".



The site has basic functionality such as browsing through categories of art and browsing. Can browse through different artists and comment on them. Interestingly enough, there was a login page for the customers for logging in to buy art of the site. Which gave a test login credentials for the site. This deals with the exposure of sensitive information. That means anybody could log in using these credentials any time without revealing their identity.

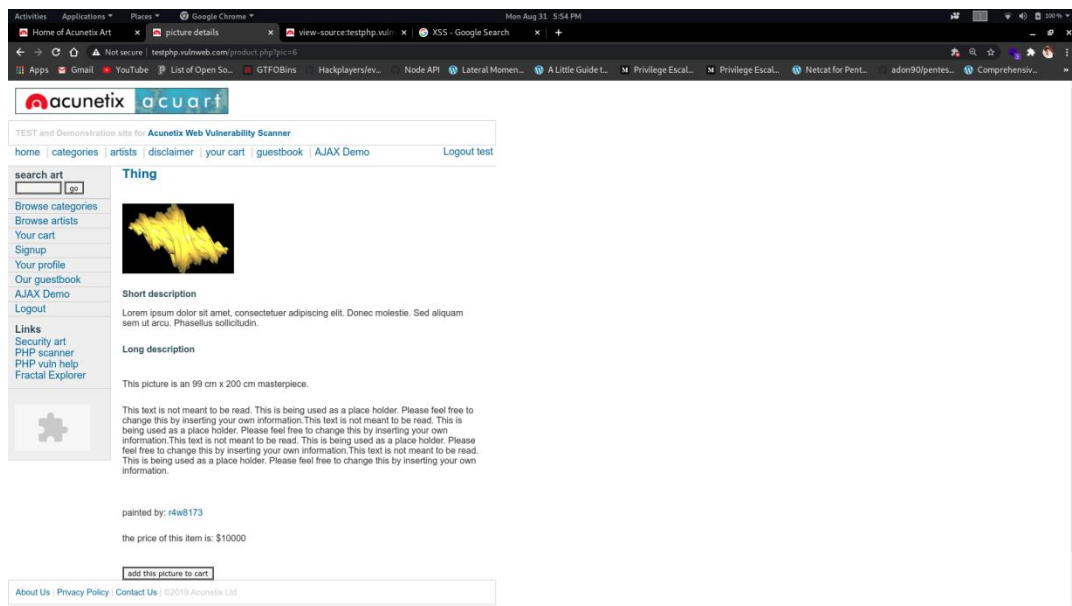


Signing in with test enables us to buy different items that are available to the customers anonymously. We can add products to our cart. Lack of validation and sanitation in some of the text fields in site proves that Cross-Site Scripting is possible. **XSS** attacks enable attackers to inject client-side scripts into web pages viewed by other users. A **cross-site scripting** vulnerability may be used by attackers to bypass access controls such as the same-origin policy. When a simple script is run on the search bar, the site gives an output.

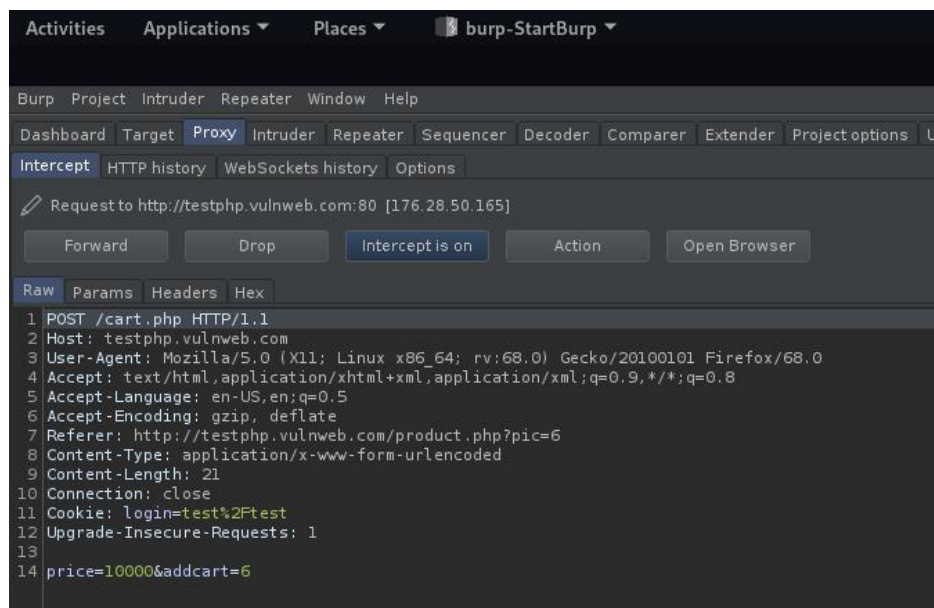


Script request was gives to show the tag alert on the page “<script>alert(“alert”)</script>”. on click of search button, the alert box was popped up. This can be lethal flaw for on whom can exploit. What else caught my mind was we could add the paintings the browsing page to our inbuilt cart.





We can intercept the request by a third party tool called Burp Suite. And change the amount for the painting, thus we can buy the painting at the cost we provide in burp suite.



Here i changed the price from 10001 to 0 and forwarded the request. As expected it was added to the cart worth of 0.

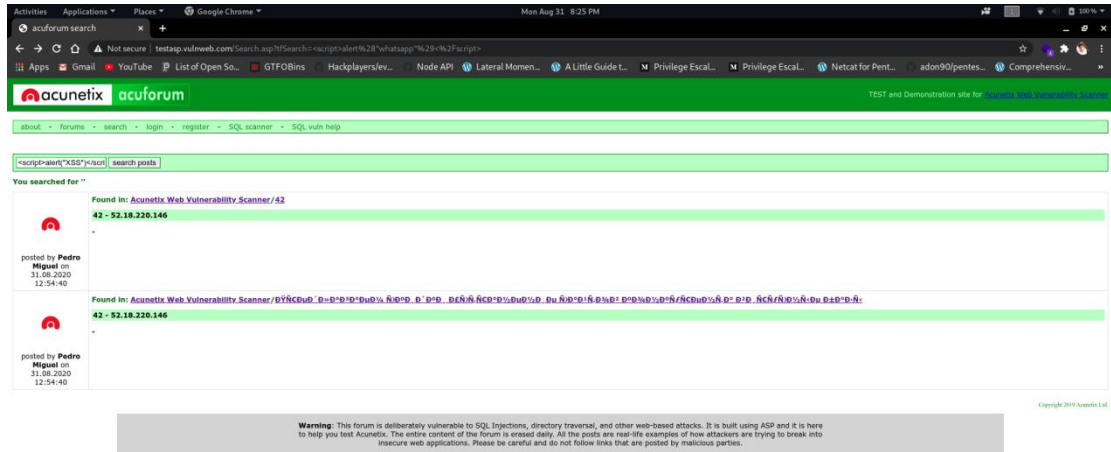




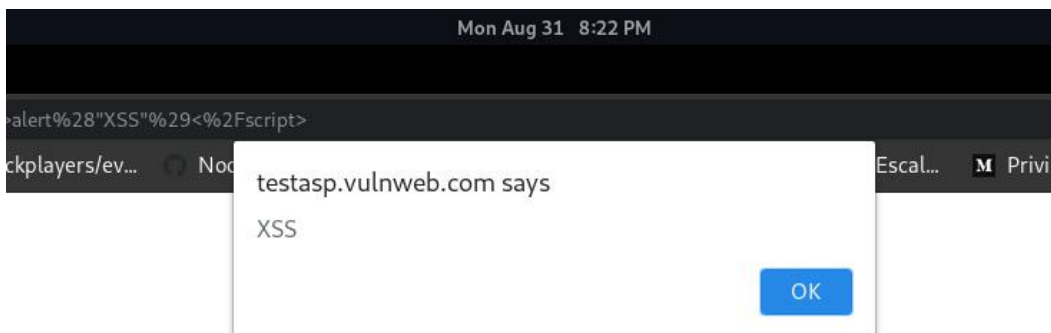
- testasp.vulnweb.com

During my initial scan, found out that the host is based on IIS, ASP and Microsoft SQL Server. Based of that, findings was its based of Microsoft. The technology used itself didn't had any critical flaws over the base systems. However, going through the page i found out that there was no validation and sanitation done to most of the field the user inputs. Which lead to XSS attacks. Attacker can enter malicious codes that can compromise the web application.

The site is looks and feels like typical forum. When toggling into the search tab of the search bar.

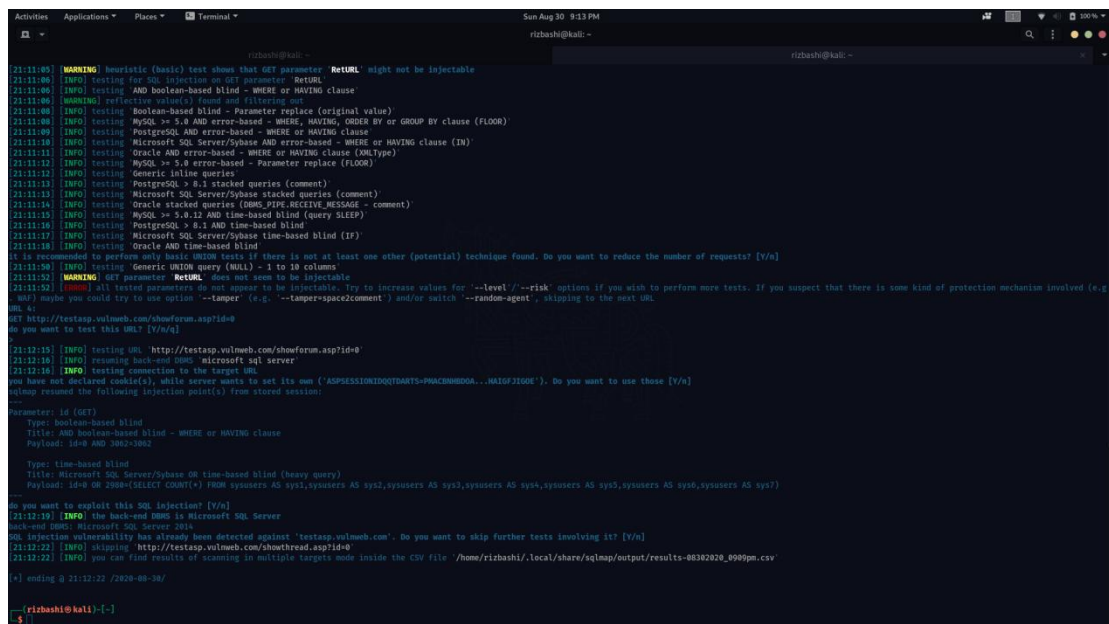


After injecting a script into the search, the output was



which proved the XSS attacks.

I also used third party tools for further investigation and recon over the site. One being SQLmap and i could find at least two vulnerable links from the normal crawl.



```
21:11:05 [WARNING] heuristic (basic) test shows that GET parameter RetURL might not be injectable
21:11:06 [INFO] testing for SQL injection on GET parameter RetURL
21:11:06 [INFO] testing AND boolean-based blind - WHERE or HAVING clause
21:11:06 [WARNING] reflective value(s) found and filtering out
21:11:08 [INFO] testing Boolean-based blind - Parameter replace (original value)
21:11:08 [INFO] testing MySQL >= 5.0 AND error-based - WHERE, HAVING, ORDER BY or GROUP BY clause (FLOOR)
21:11:09 [INFO] testing PostgreSQL AND error-based - WHERE or HAVING clause
21:11:10 [INFO] testing Microsoft SQL Server/MySQL AND error-based - WHERE or HAVING clause (IN)
21:11:11 [INFO] testing Oracle AND error-based - WHERE or HAVING clause (XMLType)
21:11:12 [INFO] testing MySQL >= 5.0 error-based - Parameter replace (FLOOR)
21:11:12 [INFO] testing Generic inline queries
21:11:13 [INFO] testing PostgreSQL >= 8.1 stacked queries (comment)
21:11:13 [INFO] testing Microsoft SQL Server/MySQL stacked queries (comment)
21:11:14 [INFO] testing Oracle stacked queries (DBMS_PIPE.RECEIVE_MESSAGE - comment)
21:11:15 [INFO] testing MySQL >= 5.0.12 AND time-based blind (query SLEEP)
21:11:16 [INFO] testing PostgreSQL >= 8.1 AND time-based blind
21:11:17 [INFO] testing Microsoft SQL Server/MySQL time-based blind (IF)
21:11:18 [INFO] testing Oracle AND time-based blind
21:11:18 [INFO] It is recommended to perform only basic UNION tests if there is not at least one other (potential) technique found. Do you want to reduce the number of requests? [Y/n]
21:11:19 [INFO] testing Generic UNION query (NULL) - 1 to 10 columns
21:11:20 [WARNING] GET parameter RetURL does not seem to be injectable
21:11:21 [WARNING] all tested parameters do not appear to be injectable. try to increase values for '--level/--risk' options if you wish to perform more tests. If you suspect that there is some kind of protection mechanism involved (e.g. WAF) maybe you could try to use option '--tamper' (e.g. '--tamper=spacecomment') and/or switch '--random-agent' skipping to the next URL.
GET http://testasp.vulnweb.com/showforum.asp?id=0
do you want to test this URL? [Y/n]
21:12:15 [INFO] testing URL http://testasp.vulnweb.com/showforum.asp?id=0
21:12:16 [INFO] resuming back-end dbms - microsoft sql server
21:12:16 [INFO] testing connection to the target URL.
21:12:17 [INFO] testing Microsoft SQL Server/MySQL time-based blind (IF)
21:12:18 [INFO] testing Oracle AND time-based blind
21:12:18 [INFO] It is recommended to perform only basic UNION tests if there is not at least one other (potential) technique found. Do you want to reduce the number of requests? [Y/n]
21:12:19 [INFO] testing Generic UNION query (NULL) - 1 to 10 columns
21:12:20 [WARNING] GET parameter RetURL does not seem to be injectable
21:12:21 [WARNING] all tested parameters do not appear to be injectable. try to increase values for '--level/--risk' options if you wish to perform more tests. If you suspect that there is some kind of protection mechanism involved (e.g. WAF) maybe you could try to use option '--tamper' (e.g. '--tamper=spacecomment') and/or switch '--random-agent' skipping to the next URL.
GET http://testasp.vulnweb.com/showthread.asp?id=0
do you want to test this URL? [Y/n]
21:12:19 [INFO] the back-end DBMS is Microsoft SQL Server
21:12:20 [INFO] testing connection to the target URL.
21:12:21 [INFO] testing Microsoft SQL Server/MySQL time-based blind (heavy query)
21:12:22 [INFO] skipping http://testasp.vulnweb.com/showthread.asp?id=0
21:12:22 [INFO] you can find results of scanning in multiple targets mode inside the CSV file /home/rihazhi/.local/share/sqlmap/output/results-00302020_0909pm.csv
[+] ending @ 21:12:22 / 2020-08-30/

--
rihazhi@kali:~$
```

with the flag “--dbs” we could see 6 databases. We can get to each and every databases. By which we can find credentials for the site which is stored in the database. And the web server is prone to cookie injection attacks. At least on CGI script failed to adequately sanitize request strings with malicious Java script.

## ◆ RISK ASSESSMENT

The results from the scan are listed below. It is important to note that all the given hosts was scanned and gone through third party tools. All three hosts were scanned on the bases of hosts and their web application part.

This report identifies security risks that could have a significant impact on mission critical application used in this mock up sites. For instance, testphp.vulnweb.com was pretty much a real life scenario if there is no substantial validation and sanitation. All it takes is a malicious script or peculiar SQL query to gain access to the site. It was comparative easy to Critical Severity Vulnerability

use some of automated tools for the crawling. These risks are quantified according to their likelihood of occurrence and the potential damage if they occur. Risk factors are combined to form an overall risk index for each system, allowing you to prioritize your remediation activities accordingly.

### Critical Severity Vulnerability

#### ● Testhtml5.vunweb.com

As the host the scan result showed large amount of medium level threats and some critical threats. They are relatively easy for attackers to exploit and will provide them with full control of the affected systems.

A list of the most frequent vulnerabilities are given below.



Vulnerability	Description	Solution
Unsupported Software and firmwares	This particular host was found with an outdated version of the operating system itself. According to Nessus scan result, the UNIX operating system running on the base is unsupported for future security patches.	The solution is to upgrade to a version of UNIX operating system that is currently in support. I would rather recommend to upgrade to the latest version as they have all the new security patches and won't go out of commission and support anytime soon.
Sensitive Data Exposure	Sensitive Data Exposure vulnerabilities can occur when a web application does not adequately protect sensitive information from being disclosed to attackers. This can include information such as mentioning the technologies used in developing the host with known vulnerabilities.	The solution is to try maximum not to include any such contents while developing the site. With basic search through the net we could find the vulnerable technologies and the version for the technologies that was used. Even if the developer is in the position for to use, such information on the page, please ensure every security patches are up to date and conduct regular risk assessment.
SQL Injection	I was able to log in to the admin account by injecting a basic SQL query into the password field. And gain access to the admin account. The user name was already filled out as admin, which shows the above said vulnerability as per. There was even no need to enumerate the users.	The solution is to validate and sanitize the user input field while developing the site. It is said that to never trust users input.

● Testphp.vulnweb.com

The Nessus scan gave up most of the details regarding the host vulnerabilities. As usual it was found that there were many critical vulnerabilities that threaten this particular system. This host is particularly easy to compromise as it is set in that way.

A list of severe and frequent vulnerabilities are given below.

Vulnerability	Description	Solution
Unsupported Software and firmwares	This particular host was found with an outdated version of the operating system itself. According to Nessus scan result, the UNIX operating system running on the base is unsupported for future security patches.	The solution is to upgrade to a version of UNIX operating system that is currently in support. I would rather recommend to upgrade to the latest version as they have all the new security patches and won't go out of commission and support anytime soon.
Sensitive Data Exposure	Sensitive Data Exposure vulnerabilities can occur when a web application does not adequately protect sensitive information from being disclosed to attackers. The initial credentials were given to log in to the site where username was "test" and the password was "test" which means anyone could log in anonymously and do what a credible user could do.	The solution is to try maximum not to include any such contents while developing the site. With basic search through the net we could find the vulnerable technologies and the version for the technologies that was used. Even if the developer is in the position for to use, such information on the page, please ensure every security patches are up to date and conduct regular risk assessment.
SQL Injection	I used a third party tool this time to crawl and look for SQL injection possibilities. It turns out that, there was more than one link in the site which was vulnerable to SQL Injection. I was able to see their entire databases linked to site. In no time I could access their databases and find credentials or what that seems confidential	The solution is to validate and sanitize the user input field while developing the site. It is said that to never trust users input.
IDOR(Insecure Data Object references)	IDOR (Insecure Direct Object Reference) is a common vulnerability that occurs when a reference to an internal implementation object is exposed without any other access control. The vulnerability is often easy to discover and allows attackers to access unauthorized data. Here, however, we can add items to the cart by browsing the art and its artists, when I requested some product to be added to the cart, I could intercept that particular traffic and change the amount of the art. And the very art can be bought with the price we decide.	Here the solution is rather simple, the developers should avoid displaying private object references such as keys of file name. Validation of parameters should be properly implemented. Tokens should be generated in such a way that it should be only mapped to the user and should not be public.

Cross-site scripting	Cross-Site Scripting (XSS) attacks are a type of injection, in which malicious scripts are injected into otherwise benign and trusted websites. XSS attacks occur when an attacker uses a web application to send malicious code, generally in the form of a browser side script, to a different end user. In this site, i was able to inject a simple script that proved the space for XSS attacks.	The solution for these kind of attacks are to filter input on arrival, encode data on output and use appropriate response headers.
----------------------	--	--

● Testasp.vulnweb.com

Comparatively this particularly is least vulnerable system among the list of hosts. This host has entirely different technologies used over the host. This is a Microsoft based host where IIS, ASP and Microsoft SQL server. The base of OS of the system is Windows Server 2012 R2. Particular OS is newer than the other two hosts scanned and tested.

Here are the list of severe threats found in this host.

Vulnerability	Description	Solution
SQL Injection	I again used a third party tool this time to crawl and look for SQL injection possibilities. It turns out that, there was more than two links in the site which was vulnerable to SQL Injection. I was able to see their entire databases linked to site. In no time i could access their databases and find credentials or what that seems confidential	The solution is to validate and sanitation the user input field while developing the site. It is said that to never trust users input.
Cross-site scripting	Cross-Site Scripting (XSS) attacks are a type of injection, in which malicious scripts are injected into otherwise benign and trusted websites. XSS attacks occur when an attacker uses a web application to send malicious code, generally in the form of a browser side script, to a different end user. In this site, i was able to inject a simple script that proved the space for XSS attacks.	The solution for these kind of attacks are to filter input on arrival, encode data on output and use appropriate response headers.



CGI Generic Cookie Injection Scripting	<p>The remote web server hosts at least one CGI script that fails to adequately sanitize request strings with malicious JavaScript.</p> <p>By leveraging this issue, an attacker may be able to inject arbitrary cookies.</p> <p>Depending on the structure of the web application, it may be possible to launch a 'session fixation' attack using this mechanism.</p>	<p>Restrict access to the vulnerable application.</p> <p>Contact the vendor for a patch or upgrade.</p>
Microsoft Windows Remote Desktop Protocol Server Man-in-the-Middle Weakness	<p>The remote version of the Remote Desktop Protocol Server (Terminal Service) is vulnerable to a man-in-the-middle (MiTM) attack. The RDP client makes no effort to validate the identity of the server when setting up encryption. An attacker with the ability to intercept traffic from the RDP server can establish encryption with the client and server without being detected. A MiTM attack of this nature would allow the attacker to obtain any sensitive information transmitted, including authentication credentials. This flaw exists because the RDP server stores a hard-coded RSA private key in the mstlsapi.dll library. Any local user with access to this file (on any Windows system) can retrieve the key and use it for this attack.</p>	<p>The solution for this vulnerability is to force the use of the SSL as a transport layer for this service if supported and select the "Allow connections only from computers running Remote Desktop with Network Level Authentication" settings if its Available.</p>

## ◆ MUTUAL FINDINGS AND RECOMENDATIONS

Among all three host scanned, all three host scans gave us all the information needed to compromise. We could bust through the directories present at the host. I could see that what all services running on them. This was an easy task.

### ◆ Testhtml5.vulnweb.com

With a third party tool i was able to crawl through the directories in the host, among them most were denied the permission to access the directory.

```
/cgi-bin (Status: 403)
/admin (Status: 301)
/comment (Status: 200)
/report (Status: 200)
/examples (Status: 200)
/logout (Status: 302)
/samples (Status: 200)
/like (Status: 200)
```

These were the directories that i found. Properly controlling access to web content is crucial for running a secure web server. The directory /samples had two files in them like wise in every directory that was tried redirected to either a directory in the webserver or page in the site.

### ◆ Testphp.vulnweb.com

As done for the first host, the directories as follows.

```
/images (Status: 301)
/cgi-bin (Status: 403)
/admin (Status: 301)
/pictures (Status: 301)
/Templates (Status: 301)
/Flash (Status: 301)
/ CVS (Status: 301)
/AJAX (Status: 301)
/secured (Status: 301)
```

All of the directories found was live. Which was particularly interesting was the directory /pictures. /pictures directory had other files than images which was credentials.txt, wp-config.bak (which seems to be the backup file and a log file) and the log file WS\_FTP.LOG. By the name of it was log it is FTP log. Credential seems a confidential file that was not supposed to be shown to the users with out proper authentication.

### ◆ Testasp.vulnweb.com

```
/images (Status: 301)
/cgi-bin (Status: 301)
```

```
/templates (Status: 301)
/html (Status: 301)
/Images (Status: 301)
/t (Status: 301)
/avatars (Status: 301)
/HTML (Status: 301)
/T (Status: 301)
/Templates (Status: 301)
/IMAGES (Status: 301)
/Html (Status: 301)
/Avatars (Status: 301)
/jscripts (Status: 301)
/CGI-BIN (Status: 301)
```

This host was comparatively safe than other two. Because there was proper permission authentication for each directory in this host. The most effective way to prevent file path traversal vulnerabilities is to avoid passing user-supplied input to filesystem APIs altogether. Many application functions that do this can be rewritten to deliver the same behavior in a safer way.

If it is considered unavoidable to pass user-supplied input to filesystem APIs, then two layers of defense should be used together to prevent attacks:

- The application should validate the user input before processing it. Ideally, the validation should compare against a white list of permitted values. If that isn't possible for the required functionality, then the validation should verify that the input contains only permitted content, such as purely alphanumeric characters.
- After validating the supplied input, the application should append the input to the base directory and use a platform file system API to canonicalize the path. It should verify that the canonicalized path starts with the expected base directory.

## ◆ CONCLUSION

Among three of the hosts the third was less vulnerable to conventional attacks, as their firmwares was newer than of other two. The host which was using HTML5 was easiest to get into admin account. the host with PHP back end had so many vulnerabilities. Most of them was classified as critical severity by nessus. And was also manually. Mostly first two hosts had outdated and unsupported technologies used in the build. Its always to good be up to date.

