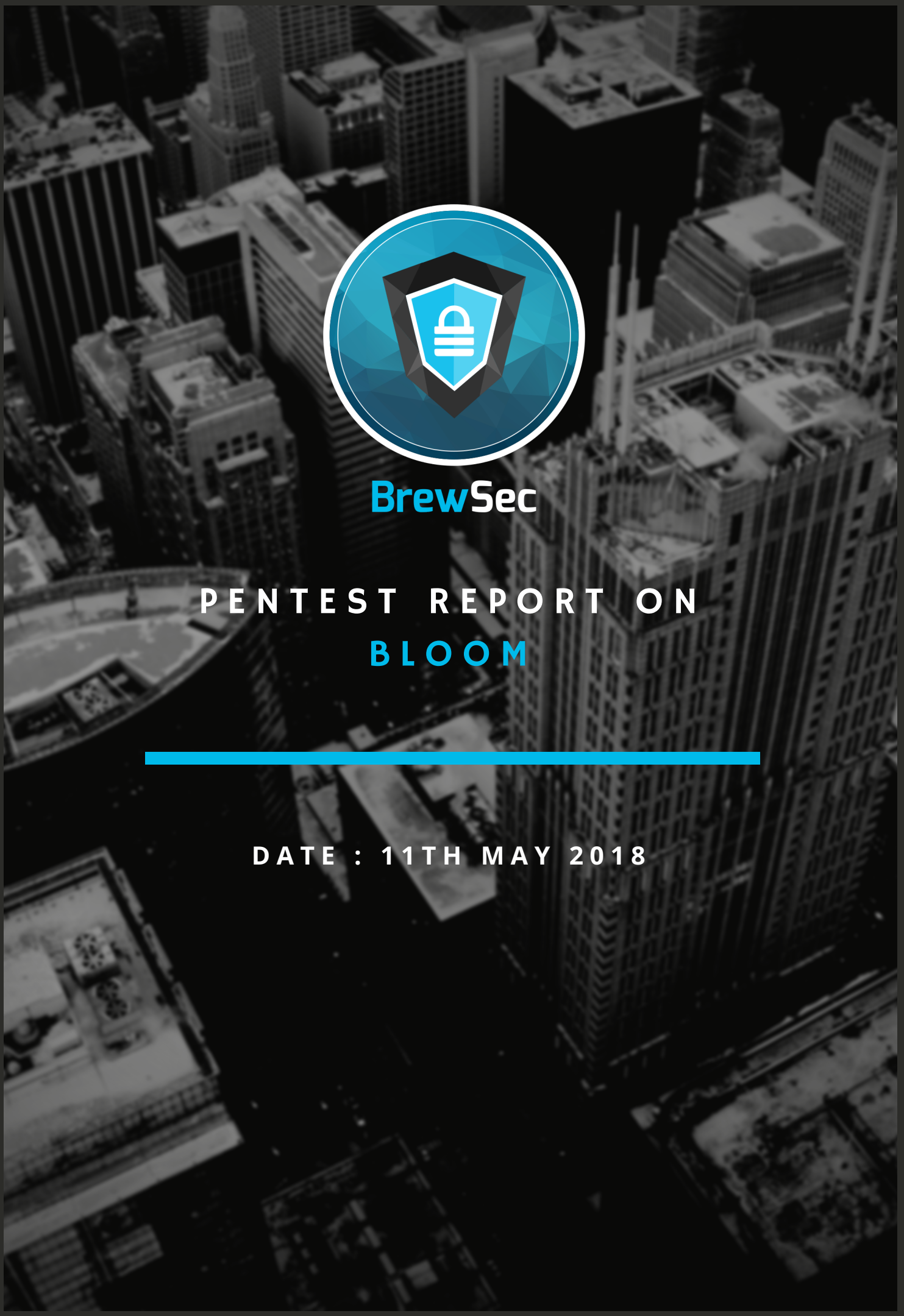


BrewSec

PENTEST REPORT ON BLOOM

DATE : 11TH MAY 2018

BrewSec

Table Of Contents

BrewSec

1. Introduction

Damn Vulnerable Web App (DVWA) is a web application built using PHP and is hosted on Apache server running on Ubuntu. We were given a website with login credentials and were asked to find the vulnerabilities affecting the website.

BrewSec is group of security researchers with more than 3 years of experience in the field of cybersecurity. We focus on different areas of cybersecurity like web application security, mobile application security, reverse engineering, system security exploitation, cryptography, malware analysis and digital forensics. We provide different cybersecurity services and solutions like advanced hands-on trainings, penetration testing, vulnerability assessment and disk imaging.

1.1 Scope

Below are the two web applications that were requested to be pentested:

- http://192.168.43.171/DVWA

2. Test Summary

BrewSec has completed pentesting both the given IP addresses provided. We focused on testing the authentication of the website following a black box approach. We found that the website was vulnerable from many security vulnerability such as SQL Injection, Command Injection, Cross site scripting. This is a serious breach of confidentiality and integrity of registered users. We also found the websites are not using SSL. This means that an attacker within the network can perform sniffing and the chances of leaking important information such as login credentials is very high.

BrewSec

2.1 Top 5 Results

SNO	Finding	Rating (CVSS2 Score)	Severity
1	Command Injection	8.8	HIGH
2	SQL Injection	8.6	HIGH
3	Stored Cross Site Scripting	8.1	HIGH

2.2 Risk Assessment Methodology

The rating of the findings is based on the Common Vulnerabilities Scoring System (CVSS 2.0). The severity of the finding is rated between 0 and 10. We provide severity rankings of "Low," "Medium," and "High" in addition to the CVSS scores. The mapping is:
Vulnerabilities are labeled "Low" severity if they have a CVSS score of 0.0-3.9.
Vulnerabilities will be labeled "Medium" severity if they have a CVSS score of 4.0-6.9.
Vulnerabilities will be labeled "High" severity if they have a CVSS score of 7.0-10.0.
Compared to other rating methodologies, CVSS is well defined, transparent and comprehensible. The particular metrics used in CVSS were identified as the best compromise between completeness, ease-of-use and accuracy.

3. Testing Environment

- Burp Suite
- Nessus
- SQLmap
- Nmap
- masscan
- Dirsearch for brute forcing directories
- Bfac
- Google cloud and Amazon AWS for running tools

4. Identified Vulnerabilities

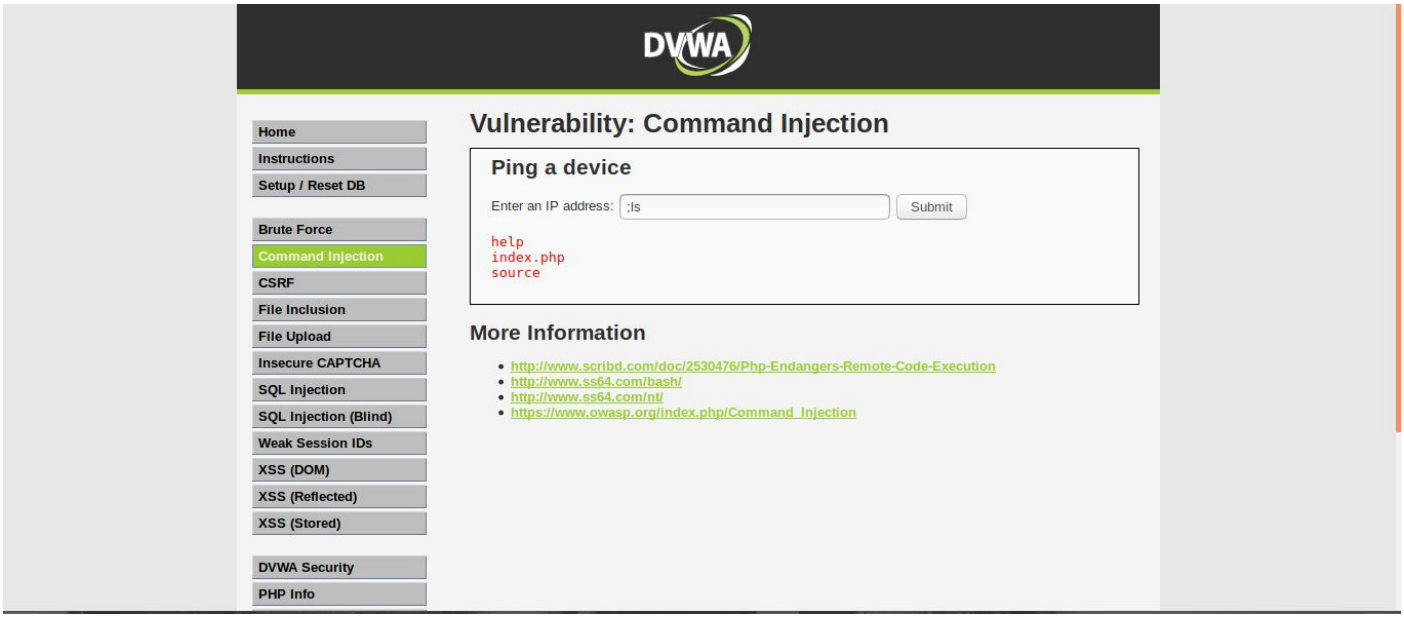
4.1 DVWA-01 : OS command injection(HIGH)

4.1.1 Description

Command injection is an attack in which the attacker can make the server to execute the arbitrary shell commands through a vulnerable web application. Command injection attacks are possible when an application passes unsafe user supplied data (forms, cookies, HTTP headers etc.) to a system shell. In this attack, the attacker-supplied operating system commands are usually executed with the privileges of the vulnerable application. Command injection attacks are possible largely due to insufficient input validation.

4.1.2 Proof Of Concept

- Attacker visits the vulnerable endpoint http://192.168.43.171/DVWA/vulnerabilities/exec/#
- Attacker injects the payload ';ls' into the Ip address input field
- Attacker can execute any linux shell command on the company server, below picture shows the result of ls command executed in the DVWA server. This security issue can help the attacker to take full control of the server.



4.1.3 Rating

CVSSv2 Score : 8.8
Base Score : AV:N/AC:L/PR:L/UI:N/S:U/C:H/I:H/A:H
Temporal Score : E:F/ RL:OF / RC:C
Environmental Score : CDP:H / TD:H / CR:H / IR:H / AR:H
Overall rating : HIGH

4.1.4 Mitigation

- he user data should be strictly validated
- Only alphanumeric strings and dot should be accepted
- The application should use command APIs that launch a specific process via its name rather than functions like eval, system which directly executes the shell commands

4.2 DVWA-02 : Stored Cross Site Scripting (HIGH)

4.2.1 Description

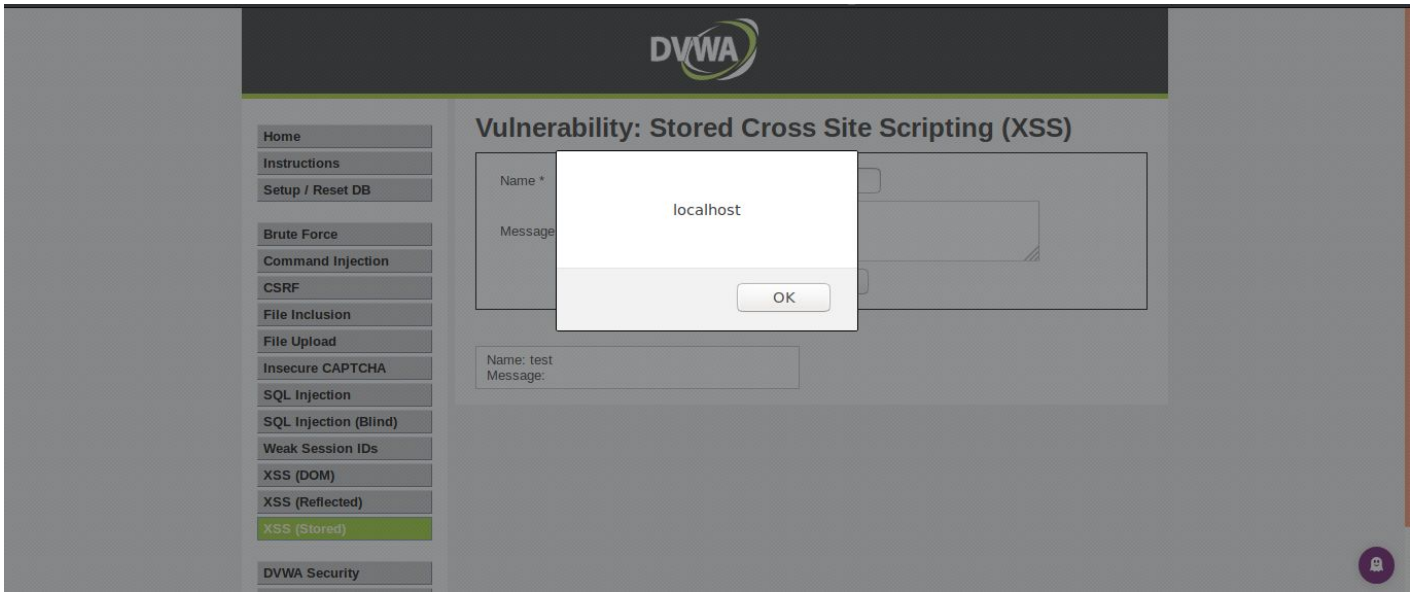
Stored Cross Site Scripting (XSS) is a persistent XSS where an attacker is able to insert malicious code into a database or file until it is removed from the database. XSS attacks occur when an attacker uses a web application to send malicious code, generally in the form of a browser side code, to a different end user. The end user's browser has no way to know that the code should not be trusted, and will execute the code. Because the end user thinks the script came from a trusted source, the malicious code can access any cookies, session tokens, or

other sensitive information retained by the browser. This code can even rewrite the content of the HTML page. This can lead to compromising accounts of users and admins, which could lead to reputational loss.

4.2.2 Proof Of Concept

CASE 1

- Attacker visits the vulnerable endpoint http://192.168.43.171/DVWA/vulnerabilities/xss_s/
- Attacker injects the payload <script>alert(document.domain)</script> into the "**Message"** field
- Attacker clicks the sign Guestbook button. This javascript(which could be malicious and can read cookies etc) is now saved to the database
- A legitimate user(generally this page is accessed by the administrator) visits the URL http://192.168.43.171/DVWA/vulnerabilities/xss_s/ , the javascript code saved by the attacker above will get executed. And this could lead the attacker to get access to confidential information such as cookies, saved passwords, deface the website etc. With the confidential information obtained the attacker could potentially use it for carrying further attacks.



4.2.3 Rating

CVSSv2 Score : 9.0
Base Score : AV:N / AC:L / Au:N / C:C / I:C / A:P

Temporal Score : E:F/ RL:OF / RC:C
Environmental Score : CDP:MH / TD:H / CR:H / IR:H / AR:H
Overall rating : HIGH

4.2.4 Mitigation

The following is recommended to remediate XSS vulnerabilities:

- Sanitize and validate user inputs in both server and client side
- HTML escape before inserting untrusted data into HTML element content
- Use whitelists in place for Black lists for input filtering

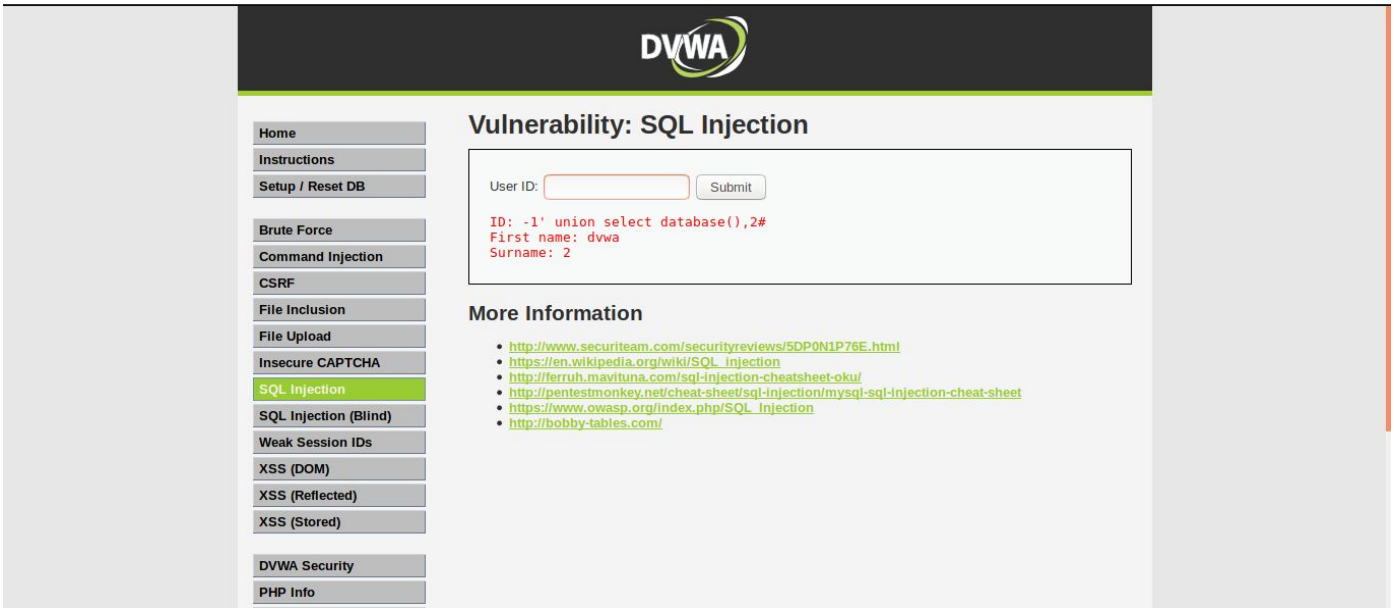
4.3 DVWA-03 : SQL Injection(HIGH)

4.3.1 Description

A SQL injection attack consists of insertion or "injection" of a SQL query via the input data from the client to the application. A successful SQL injection exploit can read sensitive data from the database, modify database data (Insert/Update/Delete), execute administration operations on the database (such as shutdown the DBMS), recover the content of a given file present on the DBMS file system and in some cases issue commands to the operating system. SQL injection attacks are a type of injection attack, in which SQL commands are injected into data-plane input in order to effect the execution of predefined SQL commands

4.3.2 Proof Of Concept

- Attacker login's to the web site and visits http://localhost/DVWA/vulnerabilities/sqli/
- Attacker can inject sql query through GET parameter id
- Attacker when inputs single quote to the id parameter, the application throws mysql error
- Further, Attacker can get more information about the database by visiting this URL (http://localhost/DVWA/vulnerabilities/sqli/?id=%27+union+select+database(),2--+-&Submit=Submit#)



4.3.3 Rating

CVSSv2 Score : 8.1
Base Score : AV:N / AC:L / Au:N / C:P / I:P / A:P
Temporal Score : E:F/ RL:OF / RC:C
Environmental Score : CDP:LM / TD:H / CR:H / IR:H / AR:H
Overall rating : HIGH

4.3.4 Mitigation

The following is recommended to remediate XSS vulnerabilities:

- Use of Prepared Statements (Parameterized Queries)
- Use of Stored Procedures
- Never trust user input, Escaping all User Supplied Input