# HackMiami Web Application Scanner 2013 PwnOff

## An Analysis of Automated Web Application Scanning Suites

James Ball, Alexander Heid, Rod Soto
http://www.HackMiami.org

### Overview

Web application scanning suites have become commonplace within the information security industry. There are many open-source and free scanning suites available, as well as a wide array of commercially licensed scanning suites.  Often these suites are marketed as automated and simple to use. The notion is that a user can point the tool at a URL and the software will rip the site apart, seeking out vulnerabilities such as SQL injections, Cross Site Scripting (XSS), and other common web application security issues.

Successful exploitation of vulnerabilities such as SQLi and XSS can lead to the compromise of data. The impact of the compromise can be minimal to catastrophic. Even the reputational impact of minimal breaches can still be significant to an organization.

This document is an analysis of the performance of five common web application scanners, which were put against three different types of web applications. The document will provide as an evaluation of the web application scanner suites from installation to the completion of the scan, and will rate the suites on multiple criteria.

The Web Application PwnOff was a live event that took place at the HackMiami 2013 Hackers Conference in Miami Beach Florida. There were three target web applications, one PHP based, one JSP based and one .NET based. The scans consisted of a single pre-authentication scan, and a single post-authentication scan against each user level. Rating scores will be on a scale of 1 (lowest) to 5 (highest).

### Meet the Contestants

Figure 1.1 displays the security suites that were used during the HackMiami 2013 Web App PwnOff.

**Fig 1.1 – Security Suite Details**

| Name | Version Used | License Used | OS | Price |
|---|---|---|---|---|
| Acunetix | 8 | Commercial | Windows 7 | $1,400 - $13,000 |
| Appscan | 8.5 | Commercial | Windows 7 | $20,000 |
| BURP | 1.5.11 | Professional | Windows 7 | $299 |
| Nexpose | 5.6 | Enterprise | VM | $20,000 |
| NTO Spider | 6.0 |  | Windows 7 | $10,000 |

**Types of Scans**

Each testing suite conducted a total of 9 scans.

**Pre-authentication scans –** This scan is where the user inputs the target URL and the tool begins attacking pages that a user is able to access without logging in. Scans are performed after initial spidering. These scans usually will not find anything on the surface if the web application only has the login page accessible. However if XSS or SQL Injection is found on these scans, exploitation could lead to catastrophic damage.

**Post-authentication (User) scan –** These scans go further in depth of the application using login credentials for regular level users.

**Post-authentication (Admin) scan –** These scans go further in depth of the application using login credentials for administrative users.

## ACUNETIX

### PHP APPLICATION SCAN RESULTS

Figures 1.2 – 1.4 displays the results of the Acunetix scan against the PHP web application.

**Fig 1.2 - PHP Pre-Authentication**

| Vulnerability | Detected | False Positives | False Negatives |
|---|---|---|---|
| XSS | **Yes (2)** | **0** | **0** |
| SQLi | **No (0)** | **0** | **1** |
| Blind SQLi | **Yes (1)** | **1** | **0** |
| Traversal | **No** | **0** | **0** |
| CSRF | **Yes (3)** | **3*** | **0** |
| Command Injection | **No** | **0** | **0** |
| Application Error | **Yes(4)** | **0** | **0** |
| Bruce Force | **YES(1)** | **0** | **0** |
| Bad HTTP Methods | **No** | **0** | **0** |
| HTTP Response Splitting | **Yes(1)** | **1** | **0** |
| Directory Listing | **YES(8)** | **0** | **0** |
| Informational** | **YES(10)** | **-** | **-** |

**Fig 1.3 - PHP Post-Authentication (User)**

| Vulnerability | Detected | False Positives | False Negatives |
|---|---|---|---|
| XSS | Yes (6) | 0 | 0 |
| SQLi | Yes (2) | 0 | 1 |
| Blind SQLi | Yes (2) | 1 | 0 |
| Traversal | Yes (3) | 2 | 0 |
| CSRF | Yes (6) | 3* | 0 |
| Command Injection | No | 0 | 0 |
| Application Error | Yes (8) | 0 | 0 |
| Bruce Force | YES (1) | 0 | 0 |
| Bad HTTP Methods | No | 0 | 0 |
| Directory Listing | YES (8) | 0 | 0 |
| Session/ Cookie flags | YES (2) | 0 | 0 |
| Session fixation | No | 0 | 1 |
| Weak Cookie | No | 0 | 0 |
| Sensitive Info In URL | Yes (12) | 6 | 0 |

**Fig 1.4 – PHP Post-Authentication (Admin)**

| Vulnerability | Detected | False Positives | False Negatives |
|---|---|---|---|
| XSS | Yes (6) | 0 | 0 |
| SQLi | Yes (2) | 0 | 1 |
| Blind SQLi | Yes (2) | 1 | 0 |
| Traversal | No | 0 | 0 |
| CSRF | Yes (9) | 3* | 0 |
| Command Injection | No | 0 | 1 |
| Application Error | Yes (9) | 0 | 0 |
| Bruce Force | YES (1) | 0 | 0 |
| Bad HTTP Methods | No | 0 | 1 |
| Directory Listing | YES (9) | 0 | 0 |
| Session/ Cookie flags | YES (2) | 0 | 0 |
| Session fixation | No | 0 | 1 |

| | | | |
|---|---|---|---|
| Weak Cookie | **No** | **0** | **1** |
| Sensitive Info In URL | **Yes (15)** | **7** | **0** |

## ACUNETIX JSP APPLICATION SCAN RESULTS

Figures 1.5 – 1.7 displays the results of the Acunetix scan against the JSP web application.

### Fig 1.5 - JSP Pre-Authentication

| **Vulnerability** | **Detected** | **False Positives** | **False Negatives** |
|---|---|---|---|
| XSS | **No** | **0** | **0** |
| SQLi | **No** | **0** | **0** |
| Blind SQLi | **Yes (1)** | **1** | **0** |
| Traversal | **No** | **0** | **0** |
| CSRF | **Yes (1)** | **1*** | **0** |
| Command Injection | **No** | **0** | **0** |
| Application Error | **Yes (4)** | **0** | **0** |
| Bruce Force | **YES (1)** | **0** | **0** |
| Bad HTTP Methods | **No** | **0** | **0** |
| HTTP Response Splitting | **Yes (1)** | **1** | **0** |
| Directory Listing | **YES (2)** | **0** | **0** |
| Informational** | **YES (14)** | **-** | **-** |

### Fig 1.6 - JSP Post-Authentication (User)

| **Vulnerability** | **Detected** | **False Positives** | **False Negatives** |
|---|---|---|---|
| XSS | **Yes (3)** | **0** | **0** |
| SQLi | **Yes (2)** | **0** | **1** |
| Blind SQLi | **Yes (2)** | **1** | **0** |
| Traversal | **Yes (3)** | **2** | **0** |
| CSRF | **Yes (3)** | **1*** | **0** |
| Command Injection | **No** | **0** | **0** |
| Application Error | **Yes (8)** | **0** | **0** |
| Bruce Force | **YES (1)** | **0** | **0** |
| Bad HTTP Methods | **No** | **0** | **0** |
| Directory Listing | **YES (2)** | **0** | **0** |

| | | | |
|---|---|---|---|
| Session/ Cookie flags | **YES (2)** | **0** | **0** |
| Session fixation | **No** | **0** | **0** |
| Weak Cookie | **No** | **0** | **0** |
| Sensitive Info In URL | **Yes (3)** | **1** | **0** |

### Fig 1.7 - JSP Post-Authentication (Admin Level)

| **Vulnerability** | **Detected** | **False Positives** | **False Negatives** |
|---|---|---|---|
| XSS | **Yes (4)** | **0** | **0** |
| SQLi | **Yes (2)** | **0** | **1** |
| Blind SQLi | **Yes (2)** | **1** | **0** |
| Traversal | **No** | **0** | **0** |
| CSRF | **Yes (4)** | **1*** | **0** |
| Command Injection | **No** | **0** | **1** |
| Application Error | **Yes (9)** | **0** | **0** |
| Bruce Force | **YES (1)** | **0** | **0** |
| Bad HTTP Methods | **No** | **0** | **1** |
| Directory Listing | **YES (3)** | **0** | **0** |
| Session/ Cookie flags | **YES (2)** | **0** | **0** |
| Session fixation | **No** | **0** | **0** |
| Weak Cookie | **No** | **0** | **1** |
| Sensitive Info In URL | **Yes (3)** | **1** | **0** |

## ACUNETIX .NET APPLICATION SCAN RESULTS

Figures 1.8 – 1.10 displays the results of the Acunetix scan against the .NET web application.

### Fig 1.8 - .NET Pre-Authentication

| Vulnerability | Detected | False Positives | False Negatives |
|---|---|---|---|
| XSS | **No** | **0** | **0** |
| SQLi | **No** | **0** | **0** |
| Blind SQLi | **No** | **0** | **0** |
| Traversal | **No** | **0** | **0** |
| CSRF | **Yes (1)** | **1** | **0** |
| Command Injection | **No** | **0** | **0** |
| Application Error | **Yes (2)** | **0** | **0** |
| Bruce Force | **Yes (1)** | **0** | **0** |
| Bad HTTP Methods | **No** | **0** | **1** |
| Unencrypted View State | **Yes (1)** | **0** | **0** |
| HTTP Response Splitting | **No** | **0** | **0** |
| Directory Listing | **Yes (2)** | **0** | **0** |
| Informational** | **Yes (10)** | **-** | **-** |

Fig 1.9 - .NET Post-Authentication (USER)

| Vulnerability | Detected | False Positives | False Negatives |
|---|---|---|---|
| XSS | Yes (4) | 0 | 0 |
| SQLi | Yes (1) | 0 | 1 |
| Blind SQLi | Yes (0) | 1 | 0 |
| Traversal | No | 0 | 0 |
| Command Injection | No | 0 | 0 |
| Application Error | Yes(8) | 0 | 0 |
| Bruce Force | YES(1) | 0 | 0 |
| Bad HTTP Methods | No | 0 | 1 |
| Directory Listing | YES(2) | 0 | 0 |
| Session/ Cookie flags | YES(2) | 0 | 0 |
| Session fixation | No | 0 | 0 |
| Unencrypted View State | Yes(1) | 0 | 0 |
| Weak Cookie | Yes(1) | 0 | 0 |
| Sensitive Info In URL | Yes(3) | 1 | 0 |

Fig 1.10 - .NET Post-Authentication (Admin)

| Vulnerability | Detected | False Positives | False Negatives |
|---|---|---|---|
| XSS | Yes (6) | 0 | 0 |
| SQLi | Yes (2) | 0 | 1 |
| Blind SQLi | Yes (2) | 1 | 0 |
| Traversal | No | 0 | 0 |
| Command Injection | No | 0 | 0 |
| Application Error | Yes(9) | 0 | 0 |
| Bruce Force | YES(1) | 0 | 0 |
| Bad HTTP Methods | No | 0 | 1 |
| Directory Listing | YES(3) | 0 | 0 |
| Session/ Cookie flags | YES(2) | 0 | 0 |
| Session fixation | No | 0 | 0 |
| Unencrypted View State | Yes(1) | 0 | 0 |
| Weak Cookie | Yes(1) | 0 | 0 |
| Sensitive Info In URL | Yes(3) | 1 | 0 |

## ACUNETIX RATING SCORES

### Interface

The interface was simple. It used a commander style windows interface that was intuitive to navigate and use. Navigating configurations for the scanner was simple and straightforward. It was also lightweight compared to other testing suites, and did not seem to be a resource hog.

For ease of use, easy installation, and straightforward interface Acunetix gets **5/5**

### Vulnerability Detection

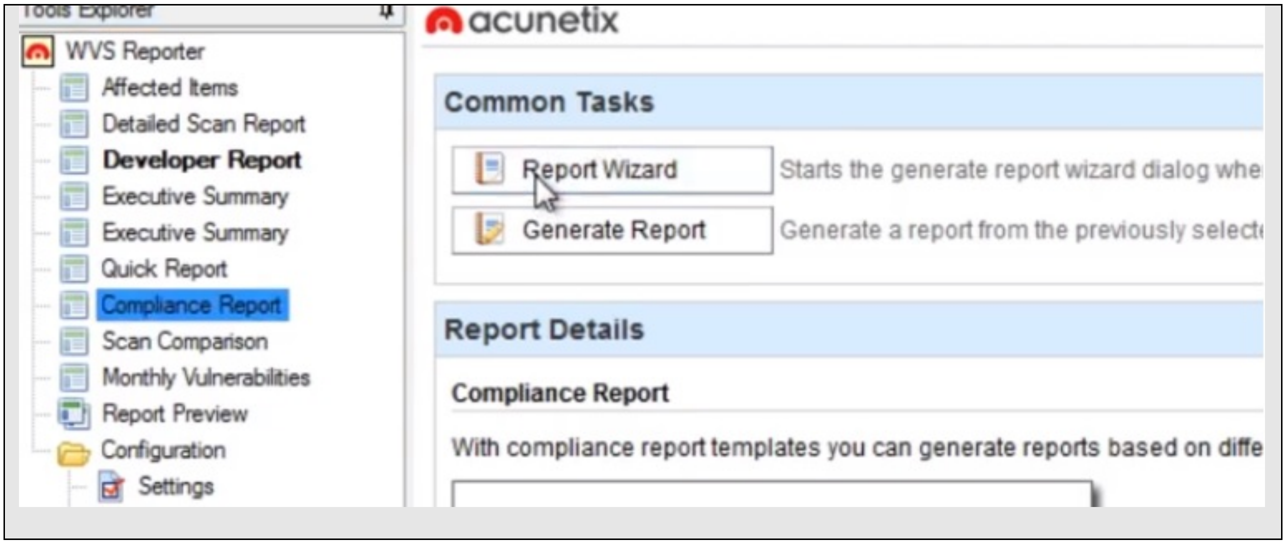
Acunetix was the best at finding Cross-Site Scripting (XSS) vectors. Acunetix was able to detect specific vulnerabilities catered to the language it was scanning against.  With few false positives and false negatives, Acunetix gets **4/5**

### Reporting

Acunetix has a great reporting feature for discovered findings. There is a report generator that lets the user select the format of the report. A personal favorite is developer's report, which shows where the vulnerabilities lie in the request and response, as well as in the code. With the consultant edition, a user can place custom logos on top of the reports.

The ease of report generation and rich features scores Acunetix gets **4.5/5**

### Fig 1.12 - Acunetix Report Generator



Starts the generate report wizard dialog whe

Generate a report from the previously selecte

With compliance report templates you can generate reports based on diffe

### Overall Value

Acunetix has licenses that cater to many different needs. They have a license for small businesses, which costs $1,445 per site, or you can try the Enterprise Edition ($3,195-$9,995) or Consultant Editions ($3,995-$12,995) with Unlimited Websites. These have the option to be perpetual licenses instead of yearly licenses. However, these prices do not cover support. The Enterprise and Consultant editions limit the user on the amount of scans per computer at the same time. This restriction also applies to Virtual Machines, too. A security consultant would be most advised to make use of the Consultant Edition: Unlimited Website for $6,350. The array of licensing options and relative costs scores Acunetix **4/5**.

## IBM RATIONAL APPSCAN STANDARD

### PHP APPLICATION SCAN RESULTS

Figures 2.1 – 2.3 displays the results of the IBM Rational AppScan Standard scan against the PHP web application.

**Fig 2.1 – PHP Pre-authentication scan**

| **Vulnerability** | **Detected** | **False Positives** | **False Negatives** |
|---|---|---|---|
| XSS | **Yes (1)** | **0** | **1** |
| SQLi | **No** | **0** | **0** |
| Blind SQLi | **No** | **0** | **0** |
| Traversal | **No** | **0** | **0** |
| CSRF | **Yes (3)** | **3*** | **0** |
| Command Injection | **No** | **0** | **0** |
| Application Error | **Yes (4)** | **0** | **0** |
| Bruce Force | **Yes (1)** | **0** | **0** |
| Bad HTTP Methods | **No** | **0** | **0** |
| HTTP Response Splitting | **No** | **0** | **0** |
| Directory Listing | **Yes (8)** | **0** | **0** |
| Informational** | **Yes (10)** | **-** | **-** |

**Fig 2.2. PHP Post-authentication (user)**

| Vulnerability | Detected | False Positives | False Negatives |
|---|---|---|---|
| XSS | **Yes (3)** | **0** | **3** |
| SQLi | **Yes (2)** | **0** | **0** |
| Blind SQLi | **Yes (1)** | **0** | **0** |
| Traversal | **Yes (1)** | **0** | **0** |
| CSRF | **Yes (9)** | **6*** | **0** |
| Command Injection | **No** | **0** | **0** |
| Application Error | **Yes (8)** | **0** | **0** |
| Bruce Force | **Yes (1)** | **0** | **0** |
| Bad HTTP Methods | **No** | **0** | **0** |
| Directory Listing | **Yes (8)** | **0** | **0** |
| Session/ Cookie flags | **Yes (2)** | **0** | **0** |
| Session fixation | **Yes** | **0** | **0** |
| Weak Cookie | **No** | **0** | **0** |
| Sensitive Info In URL | **Yes (26)** | **14** | **0** |

**Fig 2.3 - PHP Post-authentication scan (Administrator)**

| Vulnerability | Detected | False Positives | False Negatives |
|---|---|---|---|
| XSS | **Yes (3)** | **0** | **3** |
| SQLi | **Yes (3)** | **0** | **0** |
| Blind SQLi | **Yes (1)** | **0** | **0** |
| Traversal | **Yes (1)** | **0** | **0** |
| CSRF | **Yes (14)** | **8*** | **0** |
| Command Injection | **Yes (1)** | **0** | **0** |
| Application Error | **Yes (9)** | **0** | **0** |
| Bruce Force | **Yes (1)** | **0** | **0** |
| Bad HTTP Methods | **Yes (1)** | **0** | **0** |
| Directory Listing | **Yes (9)** | **0** | **0** |
| Session/ Cookie flags | **Yes (2)** | **0** | **0** |
| Session fixation | **Yes** | **0** | **0** |
| Weak Cookie | **Yes** | **0** | **0** |
| Sensitive Info In URL | **Yes (37)** | **19** | **0** |

## IBM RATIONAL APPSCAN JSP APPLICATION SCAN RESULTS

Figures 2.4 – 2.6 displays the results of the IBM Rational AppScan Standard scan against the JSP web application.

### Fig 2.4 - JSP Pre-Authentication

| Vulnerability | Detected | False Positives | False Negatives |
|---|---|---|---|
| XSS | No | 0 | 0 |
| SQLi | No | 0 | 0 |
| Blind SQLi | No | 0 | 0 |
| Traversal | No | 0 | 0 |
| CSRF | No | 0 | 0 |
| Command Injection | No | 0 | 0 |
| Application Error | Yes(4) | 0 | 0 |
| Bruce Force | No | 0 | 1 |
| Bad HTTP Methods | No | 0 | 0 |
| HTTP Response Splitting | No | 0 | 0 |
| Directory Listing | YES(2) | 0 | 0 |
| Informational** | YES(14) | - | - |

### Fig 2.5 – JSP Post-Authentication (User)

| Vulnerability | Detected | False Positives | False Negatives |
|---|---|---|---|
| XSS | Yes (2) | 0 | 2 |
| SQLi | Yes (2) | 0 | 1 |
| Blind SQLi | Yes (1) | 0 | 0 |
| Traversal | Yes (1) | 0 | 0 |
| CSRF | Yes (4) | 3* | 0 |
| Command Injection | No | 0 | 0 |
| Application Error | Yes (8) | 0 | 0 |
| Bruce Force | Yes (1) | 0 | 0 |
| Bad HTTP Methods | No | 0 | 0 |
| Directory Listing | Yes (2) | 0 | 0 |
| Session/ Cookie flags | Yes (2) | 0 | 0 |
| Session fixation | No | 0 | 0 |
| Weak Cookie | No | 0 | 0 |

| | | | |
|---|---|---|---|
| Sensitive Info In URL | **Yes (4)** | **2** | **0** |

### Fig 2.6 – JSP Post-Authentication (Administrator)

| **Vulnerability** | **Detected** | **False Positives** | **False Negatives** |
|---|---|---|---|
| XSS | **Yes (2)** | **0** | **2** |
| SQLi | **Yes (2)** | **0** | **1** |
| Blind SQLi | **Yes (3)** | **2** | **0** |
| Traversal | **Yes (1)** | **0** | **0** |
| CSRF | **Yes(6)** | **3*** | **0** |
| Command Injection | **Yes (1)** | **0** | **0** |
| Application Error | **Yes (9)** | **0** | **0** |
| Bruce Force | **Yes(1)** | **0** | **0** |
| Bad HTTP Methods | **Yes(1)** | **0** | **0** |
| Directory Listing | **Yes (3)** | **0** | **0** |
| Session/ Cookie flags | **Yes (2)** | **0** | **0** |
| Session fixation | **No** | **0** | **0** |
| Weak Cookie | **Yes (1)** | **0** | **0** |
| Sensitive Info In URL | **Yes (5)** | **3** | **0** |

## IBM RATIONAL APPSCAN .NET APPLICATION SCAN RESULTS

Figures 2.7 – 2.9 displays the results of the IBM Rational AppScan Standard scan against the .NET web application.

### Fig 2.7 – .NET Pre-Authentication

| Vulnerability | Detected | False Positives | False Negatives |
|---|---|---|---|
| XSS | **No** | **0** | **0** |
| SQLi | **No** | **0** | **0** |
| Blind SQLi | **No** | **0** | **0** |
| Traversal | **No** | **0** | **0** |
| CSRF | **No** | **0** | **0** |
| Command Injection | **No** | **0** | **0** |
| Application Error | **Yes (2)** | **0** | **0** |
| Bruce Force | **No** | **0** | **1** |
| Bad HTTP Methods | **No** | **0** | **1** |
| Unencrypted View State | **Yes (1)** | **0** | **0** |
| HTTP Response Splitting | **No** | **0** | **0** |
| Directory Listing | **Yes (2)** | **0** | **0** |
| Informational** | **Yes (6)** | **-** | |

### Fig 2.8 – .NET Post-Authentication (Users)

| Vulnerability | Detected | False Positives | False Negatives |
|---|---|---|---|
| XSS | Yes (2) | 0 | 2 |
| SQLi | Yes (2) | 0 | 0 |
| Blind SQLi | No | 0 | 0 |
| Traversal | No | 0 | 0 |
| Command Injection | No | 0 | 0 |
| Application Error | Yes(8) | 0 | 0 |
| Bruce Force | Yes (1) | 0 | 0 |
| Bad HTTP Methods | Yes (1) | 0 | 0 |
| Directory Listing | YES(2) | 0 | 0 |
| Session/ Cookie flags | YES(2) | 0 | 0 |
| Session fixation | No | 0 | 0 |
| Unencrypted View State | Yes(1) | 0 | 0 |
| Weak Cookie | Yes(1) | 0 | 0 |
| Sensitive Info In URL | Yes(2) | 0 | 0 |

### Fig 2.9 – .NET Post-Authentication (Administrator)

| Vulnerability | Detected | False Positives | False Negatives |
|---|---|---|---|
| XSS | Yes (3) | 0 | 3 |
| SQLi | Yes (3) | 0 | 0 |
| Blind SQLi | Yes (1) | 0 | 0 |
| Traversal | No | 0 | 0 |
| Command Injection | No | 0 | 0 |
| Application Error | Yes (9) | 0 | 0 |
| Bruce Force | Yes (1) | 0 | 0 |
| Bad HTTP Methods | Yes (1) | 0 | 0 |
| Directory Listing | Yes (3) | 0 | 0 |
| Session/ Cookie flags | Yes (2) | 0 | 0 |
| Session fixation | No | 0 | 0 |
| Unencrypted View State | Yes (1) | 0 | 0 |
| Weak Cookie | Yes (1) | 0 | 0 |
| Sensitive Info In URL | Yes (5) | 2 | 0 |

## IBM RATIONAL APPSCAN RATING SCORES

### Ease of Interface

IBM Rational AppScan Standard has a clean interface that is simple to navigate. However, some of the more important features are not prominently displayed. It is also very resource intensive when the scan is running. For example, running two concurrent AppScan sessions will slow down a computer with 8 GB of RAM considerably. AppScan also had an issue with session detection on the JSP and .NET applications, and the users had to use the login manager a few times to get a completed scan. With the clean interface and simplicity of adjusting configuration settings, AppScan gets a score of **3.5/5**

### Vulnerability Detection

AppScan has a very good vulnerability database and can pick up even some of the trickiest and most obscure SQL injections. It also detects and accurately rates command injection and directory traversal. With few false negatives and and high SQLi catch rate, AppScan gets a score of **4/5**

### Reporting

AppScan also has a report generation wizard. The license allows users to customize the logo and headers on the report. Report generation is simple, and the user can have AppScan produce several separate PDF files, or have it produce a larger single file by combining reports. The Report Wizard was intuitive, and since it comes natively with the ability to customize the reports Appscan gets a score of **4.8/5**

### Overall Value

Appscan comes at a price of $20,300 per license per year. This does come with free support for the first year. This product is better suited for Enterprise class customers. A Security Consultant for hire will still be able to get a Single Install license for $9,540 but if you have two computers you want to put it on it will be better to go with the $20k license. While it does a fantastic job the price is high especially for an individual. For value Appscan gets a score of **3/5**

# PORTSWIGGER BURP

## PHP APPLICATION SCAN RESULTS

Figures 3.1 – 3.3 displays the results of the Portswigger BURP scan against the PHP web application.

### Fig 3.1 – PHP Pre-authentication scan

| Vulnerability | Detected | False Positives | False Negatives |
|---|---|---|---|
| XSS | Yes (2) | 0 | 0 |
| SQLi | No | 0 | 0 |
| Blind SQLi | No | 0 | 0 |
| Traversal | No | 0 | 0 |
| CSRF | No | 0 | 0 |
| Command Injection | No | 0 | 0 |
| Application Error | Yes(4) | 0 | 0 |
| Bruce Force | No | 0 | 1 |
| Bad HTTP Methods | No | 0 | 0 |
| HTTP Response Splitting | No | 0 | 0 |
| Directory Listing | YES(8) | 0 | 0 |
| Informational** | YES(6) | - | - |

### Fig 3.2. PHP Post-Authentication (user)

| Vulnerability | Detected | False Positives | False Negatives |
|---|---|---|---|
| XSS | Yes (3) | 0 | 3 |
| SQLi | Yes (2) | 0 | 0 |
| Blind SQLi | Yes (1) | 0 | 0 |
| Traversal | Yes(1) | 0 | 0 |
| CSRF | Yes (9) | 6* | 0 |
| Command Injection | No | 0 | 0 |
| Application Error | Yes(8) | 0 | 0 |
| Bruce Force | YES(1) | 0 | 0 |
| Bad HTTP Methods | No | 0 | 0 |
| Directory Listing | YES(8) | 0 | 0 |
| Session/ Cookie flags | YES(2) | 0 | 0 |
| Session fixation | Yes | 0 | 0 |
| Weak Cookie | No | 0 | 0 |
| Sensitive Info In URL | Yes(26) | 14 | 0 |

### Fig 3.3 - PHP Post-Authentication scan (Administrator)

| Vulnerability | Detected | False Positives | False Negatives |
|---|---|---|---|
| XSS | **Yes (3)** | **0** | **3** |
| SQLi | **Yes (3)** | **0** | **0** |
| Blind SQLi | **Yes(1)** | **0** | **0** |
| Traversal | **Yes (1)** | **0** | **0** |
| CSRF | **Yes (14)** | **8*** | **0** |
| Command Injection | **Yes(1)** | **0** | **0** |
| Application Error | **Yes(9)** | **0** | **0** |
| Bruce Force | **YES(1)** | **0** | **0** |
| Bad HTTP Methods | **YES(1)** | **0** | **0** |
| Directory Listing | **YES(9)** | **0** | **0** |
| Session/ Cookie flags | **YES(2)** | **0** | **0** |
| Session fixation | **Yes** | **0** | **0** |
| Weak Cookie | **Yes** | **0** | **0** |
| Sensitive Info In URL | **Yes(37)** | **19** | **0** |

## PORTSWIGGER BURP JSP APPLICATION SCAN RESULTS

Figures 3.4 – 3.6 displays the results of the Portswigger BURP scan against the JSP web application.

### Fig 3.4 - JSP Pre-Authentication

| Vulnerability | Detected | False Positives | False Negatives |
|---|---|---|---|
| XSS | **No** | **0** | **0** |
| SQLi | **No** | **0** | **0** |
| Blind SQLi | **No** | **0** | **0** |
| Traversal | **No** | **0** | **0** |
| CSRF | **Yes (1)** | **1*** | **0** |
| Command Injection | **No** | **0** | **0** |
| Application Error | **Yes (4)** | **0** | **0** |
| Bruce Force | **Yes (1)** | **0** | **0** |
| Bad HTTP Methods | **No** | **0** | **0** |
| HTTP Response Splitting | **No** | **0** | **0** |
| Directory Listing | **Yes (2)** | **0** | **0** |
| Informational** | **Yes (14)** | **-** | **-** |

### Fig 3.5 – JSP Post-Authentication (User)

| Vulnerability | Detected | False Positives | False Negatives |
|---|---|---|---|
| XSS | **Yes (2)** | **0** | **1** |
| SQLi | **Yes (3)** | **0** | **0** |
| Blind SQLi | **Yes (1)** | **0** | **0** |
| Traversal | **Yes (1)** | **0** | **0** |
| CSRF | **Yes (3)** | **1*** | **0** |
| Command Injection | **No** | **0** | **0** |
| Application Error | **Yes (8)** | **0** | **0** |
| Bruce Force | **Yes (1)** | **0** | **0** |
| Bad HTTP Methods | **No** | **0** | **0** |
| Directory Listing | **Yes (2)** | **0** | **0** |
| Session/ Cookie flags | **Yes (2)** | **0** | **0** |
| Session fixation | **No** | **0** | **0** |
| Weak Cookie | **No** | **0** | **0** |
| Sensitive Info In URL | **Yes (4)** | **2** | **0** |

### Fig 3.6 – JSP Post-Authentication (Administrator)

| Vulnerability | Detected | False Positives | False Negatives |
|---|---|---|---|
| XSS | **Yes (2)** | **0** | **2** |
| SQLi | **Yes (3)** | **0** | **0** |
| Blind SQLi | **Yes (1)** | **0** | **0** |
| Traversal | **Yes (1)** | **0** | **0** |
| CSRF | **Yes (4)** | **1*** | **0** |
| Command Injection | **Yes (1)** | **0** | **0** |
| Application Error | **Yes (9)** | **0** | **0** |
| Bruce Force | **Yes (1)** | **0** | **0** |
| Bad HTTP Methods | **Yes (1)** | **0** | **0** |
| Directory Listing | **Yes (3)** | **0** | **0** |
| Session/ Cookie flags | **Yes (2)** | **0** | **0** |
| Session fixation | **No** | **0** | **0** |
| Weak Cookie | **Yes (1)** | **0** | **0** |
| Sensitive Info In URL | **Yes (5)** | **3** | **0** |

## PORTSWIGGER BURP .NET APPLICATION SCAN RESULTS

Figures 3.7 – 3.9 displays the results of the IBM Rational AppScan Standard scan against the .NET web application.

### Fig 3.7 – .NET Pre-Authentication

| Vulnerability | Detected | False Positives | False Negatives |
|---|---|---|---|
| XSS | No | 0 | 0 |
| SQLi | No | 0 | 0 |
| Blind SQLi | No | 0 | 0 |
| Traversal | No | 0 | 0 |
| CSRF | Yes (1) | 1 | 0 |
| Command Injection | No | 0 | 0 |
| Application Error | Yes (2) | 0 | 0 |
| Bruce Force | Yes (1) | 0 | 0 |
| Bad HTTP Methods | Yes (1) | 0 | 0 |
| Unencrypted View State | Yes (1) | 0 | 0 |
| HTTP Response Splitting | No | 0 | 0 |
| Directory Listing | Yes (2) | 0 | 0 |
| Informational** | Yes (12) | - | - |

### Fig 3.8 – .NET Post-Authentication (User)

| Vulnerability | Detected | False Positives | False Negatives |
|---|---|---|---|
| XSS | Yes (2) | 0 | 2 |
| SQLi | Yes (2) | 0 | 0 |
| Blind SQLi | No | 0 | 0 |
| Traversal | No | 0 | 0 |
| Command Injection | No | 0 | 0 |
| Application Error | Yes(8) | 0 | 0 |
| Bruce Force | YES(1) | 0 | 0 |
| Bad HTTP Methods | Yes (1) | 0 | 0 |
| Directory Listing | YES(2) | 0 | 0 |
| Session/ Cookie flags | YES(2) | 0 | 0 |
| Session fixation | No | 0 | 0 |
| Unencrypted View State | Yes(1) | 0 | 0 |

| | | | |
|---|---|---|---|
| Weak Cookie | **Yes(1)** | **0** | **0** |
| Sensitive Info In URL | **Yes(3)** | **1** | **0** |

### Fig 3.9 – .NET Post-Authentication (Administrator)

| Vulnerability | Detected | False Positives | False Negatives |
|---|---|---|---|
| XSS | **Yes (3)** | **0** | **3** |
| SQLi | **Yes (3)** | **0** | **0** |
| Blind SQLi | **Yes (1)** | **0** | **0** |
| Traversal | **No** | **0** | **0** |
| Command Injection | **No** | **0** | **0** |
| Application Error | **Yes(9)** | **0** | **0** |
| Bruce Force | **YES(1)** | **0** | **0** |
| Bad HTTP Methods | **Yes (1)** | **0** | **0** |
| Directory Listing | **YES(3)** | **0** | **0** |
| Session/ Cookie flags | **YES(2)** | **0** | **0** |
| Session fixation | **No** | **0** | **0** |
| Unencrypted View State | **Yes(1)** | **0** | **0** |
| Weak Cookie | **Yes(1)** | **0** | **0** |
| Sensitive Info In URL | **Yes(6)** | **3** | **0** |

## PORTSWIGGER BURP RATING SCORES

### Ease of Interface

Portswigger's BURP is written in JAVA, making it cross platform and simple to install on all operating systems. However, because it is written in JAVA, it is also very resource intensive. Navigation through BURP is simple and clean. Some features could have been more prominently displayed to the user, many of the more useful features hide behind a right click in particular spots of the suite. BURP is also a fantastic tool for manual testing, as it has a proxy feature. BURP also provides simple and quick ways to generate mass amounts of custom tests, such as with its "Intruder" tool. The automated scanner is simple to use as well, and it has no problems maintaining a valid session while spidering. Another advantage to its use of JAVA is that it allows for easy integration of custom plug-ins. BURP log files files can also be used with other tools such as NTOSpider and SQLmap. For ease of interface, Burp scores **4/5**

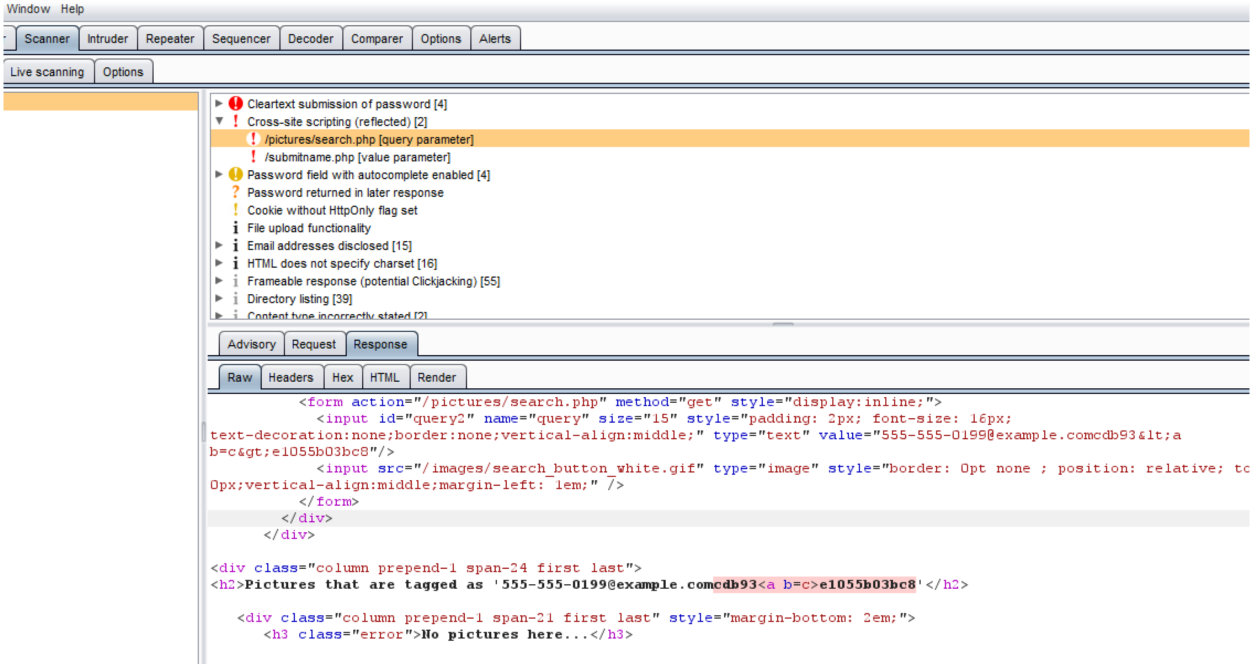
## Vulnerability Detection

BURP's Active Scanner and Passive Scanner does a very good job in discovering vulnerabilities as the site is browsed. Burp is able to detect major vulnerabilities such as Cross Site Scripting (XSS) and SQLi with ease. One feature Burp has is the ability to compare two responses side-by-side and line-by-line on the same split window. This feature makes it easier in detecting false positives. For vulnerability detection, Burp gets a score of **4/5**

## Reporting

BURP currently does not have a feature that will generate PDF reports from the data collected. BURP only report its vulnerabilities from the scanner on the Results tab. The results tab accurately describes the finding and will highlight the issue in the request and response. Furthermore, BURP can produce HTML reports. However, without a PDF generation feature, BURP gets a Reporting score of **3.5/5**

**Fig 3.10 - Screenshot showing BURP reporting on progress of scan.**



▶ ❗ Cleartext submission of password [4]
▼ ! Cross-site scripting (reflected) [2]
  ! /pictures/search.php [query parameter]
  ! /submitname.php [value parameter]
▶ ❗ Password field with autocomplete enabled [4]
? Password returned in later response
! Cookie without HttpOnly flag set
i File upload functionality
▶ i Email addresses disclosed [15]
▶ i HTML does not specify charset [16]
▶ i Frameable response (potential Clickjacking) [55]
▶ i Directory listing [39]
▶ i Content type incorrectly stated [2]

```
          <form action="/pictures/search.php" method="get" style="display:inline;">
            <input id="query2" name="query" size="15" style="padding: 2px; font-size: 16px;
text-decoration:none;border:none;vertical-align:middle;" type="text" value="555-555-0199@example.comcdb93&lt;a
b=c&gt;e1055b03bc8"/>
            <input src="/images/search_button_white.gif" type="image" style="border: 0pt none ; position: relative; tc
0px;vertical-align:middle;margin-left: 1em;" />
          </form>
        </div>
      </div>

<div class="column prepend-1 span-24 first last">
<h2>Pictures that are tagged as '555-555-0199@example.comcdb93<a b=c>e1055b03bc8'</h2>

   <div class="column prepend-1 span-21 first last" style="margin-bottom: 2em;">
      <h3 class="error">No pictures here...</h3>
```

## Overall Value

At a cost of only $300 per user, this is a great value. Burp is great to have even if you do not use the Active or Passive Scanners with it. The cross platform nature, extendibility, and its affordability gets BURP gets an Overall Value alue score of **5/5**

## RAPID 7 NEXPOSE

### PHP APPLICATION SCAN RESULTS

Figures 4.1 – 4.3 displays the results of the Rapid7 Nexpose scan against the PHP web application.

### Fig 4.1 – PHP Pre-authentication scan

| Vulnerability | Detected | False Positives | False Negatives |
|---|---|---|---|
| XSS | Yes (2) | 0 | 0 |
| SQLi | No | 0 | 0 |
| Blind SQLi | No | 0 | 0 |
| Traversal | No | 0 | 0 |
| CSRF | Yes(3) | 3* | 0 |
| Command Injection | No | 0 | 0 |
| Application Error | Yes(4) | 0 | 0 |
| Bruce Force | Yes(1) | 0 | |
| Bad HTTP Methods | No | 0 | 0 |
| HTTP Response Splitting | No | 0 | 0 |
| Directory Listing | YES(8) | 0 | 0 |
| Informational** | YES(12) | - | - |

### Fig 4.2. PHP Post-authentication (user)

| Vulnerability | Detected | False Positives | False Negatives |
|---|---|---|---|
| XSS | Yes (6) | 0 | 0 |
| SQLi | Yes (2) | 0 | 0 |
| Blind SQLi | Yes (1) | 0 | 0 |
| Traversal | Yes (1) | 0 | 0 |
| CSRF | Yes (6) | 3* | 0 |
| Command Injection | No | 0 | 0 |
| Application Error | Yes (8) | 0 | 0 |
| Bruce Force | Yes (1) | 0 | 0 |
| Bad HTTP Methods | No | 0 | 0 |
| Directory Listing | Yes (8) | 0 | 0 |
| Session/ Cookie flags | Yes (2) | 0 | 0 |
| Session fixation | Yes | 0 | 0 |
| Weak Cookie | No | 0 | 0 |
| Sensitive Info In URL | Yes (8) | 0 | 0 |

### Fig 4.3 - PHP Post-authentication scan (Administrator)

| Vulnerability | Detected | False Positives | False Negatives |
|---|---|---|---|
| XSS | **Yes (6)** | **0** | **0** |
| SQLi | **Yes (3)** | **0** | **0** |
| Blind SQLi | **Yes (1)** | **0** | **0** |
| Traversal | **Yes (1)** | **0** | **0** |
| CSRF | **Yes (9)** | **6*** | **0** |
| Command Injection | **Yes (1)** | **0** | **0** |
| Application Error | **Yes (9)** | **0** | **0** |
| Bruce Force | **Yes (1)** | **0** | **0** |
| Bad HTTP Methods | **Yes (1)** | **0** | **0** |
| Directory Listing | **Yes (9)** | **0** | **0** |
| Session/ Cookie flags | **Yes (2)** | **0** | **0** |
| Session fixation | **Yes** | **0** | **0** |
| Weak Cookie | **Yes** | **0** | **0** |
| Sensitive Info In URL | **Yes (9)** | **0** | **0** |

## RAPID7 NEXPOSE JSP APPLICATION SCAN RESULTS

Figures 4.4 – 4.6 displays the results of the Rapid7 Nexpose scan against the JSP web application.

### Fig 4.4 - JSP Pre-Authentication

| Vulnerability | Detected | False Positives | False Negatives |
|---|---|---|---|
| XSS | **No** | **0** | **0** |
| SQLi | **No** | **0** | **0** |
| Blind SQLi | **No** | **0** | **0** |
| Traversal | **No** | **0** | **0** |
| CSRF | **Yes (1)** | **1*** | **0** |
| Command Injection | **No** | **0** | **0** |
| Application Error | **Yes (4)** | **0** | **0** |
| Bruce Force | **Yes (1)** | **0** | **0** |
| Bad HTTP Methods | **No** | **0** | **0** |
| HTTP Response Splitting | **No** | **0** | **0** |
| Directory Listing | **Yes (2)** | **0** | **0** |
| Informational** | **Yes (14)** | **-** | **-** |

## Fig 4.5 – JSP Post-Authentication (User)

| Vulnerability | Detected | False Positives | False Negatives |
|---|---|---|---|
| XSS | **Yes (6)** | **2** | **0** |
| SQLi | **Yes (3)** | **0** | **0** |
| Blind SQLi | **Yes (1)** | **0** | **0** |
| Traversal | **Yes (1)** | **0** | **0** |
| CSRF | **Yes (3)** | **1*** | **0** |
| Command Injection | **No** | **0** | **0** |
| Application Error | **Yes (8)** | **0** | **0** |
| Bruce Force | **Yes (1)** | **0** | **0** |
| Bad HTTP Methods | **No** | **0** | **0** |
| Directory Listing | **Yes (2)** | **0** | **0** |
| Session/ Cookie flags | **Yes (2)** | **0** | **0** |
| Session fixation | **No** | **0** | **0** |
| Weak Cookie | **No** | **0** | **0** |
| Sensitive Info In URL | **Yes (4)** | **2** | **0** |

## Fig 4.6 – JSP Post-Authentication (Administrator)

| Vulnerability | Detected | False Positives | False Negatives |
|---|---|---|---|
| XSS | **Yes (6)** | **2** | **0** |
| SQLi | **Yes (3)** | **0** | **0** |
| Blind SQLi | **Yes (1)** | **0** | **0** |
| Traversal | **Yes (1)** | **0** | **0** |
| CSRF | **Yes (4)** | **1*** | **0** |
| Command Injection | **Yes (1)** | **0** | **0** |
| Application Error | **Yes (9)** | **0** | **0** |
| Bruce Force | **Yes (1)** | **0** | **0** |
| Bad HTTP Methods | **Yes (1)** | **0** | **0** |
| Directory Listing | **Yes (3)** | **0** | **0** |
| Session/ Cookie flags | **Yes (2)** | **0** | **0** |
| Session fixation | **No** | **0** | **0** |
| Weak Cookie | **Yes (1)** | **0** | **0** |
| Sensitive Info In URL | **Yes (5)** | **3** | **0** |

## RAPID7 NEXPOSE .NET APPLICATION SCAN RESULTS

Figures 4.7 – 4.9 displays the results of the Rapid7 Nexpose scan against the .NET web application.

### Fig 4.7 – .NET Pre-Authentication

| Vulnerability | Detected | False Positives | False Negatives |
|---|---|---|---|
| XSS | No | 0 | 0 |
| SQLi | No | 0 | 0 |
| Blind SQLi | No | 0 | 0 |
| Traversal | No | 0 | 0 |
| CSRF | No | 0 | 0 |
| Command Injection | No | 0 | 0 |
| Application Error | Yes (2) | 0 | 0 |
| Bruce Force | Yes (1) | 0 | 0 |
| Bad HTTP Methods | Yes (1) | 0 | 0 |
| Unencrypted View State | Yes (1) | 0 | 0 |
| HTTP Response Splitting | No | 0 | 0 |
| Directory Listing | Yes (2) | 0 | 0 |
| Informational** | Yes (12) | - | - |

### Fig 4.8 – .NET Post-Authentication (User)

| Vulnerability | Detected | False Positives | False Negatives |
|---|---|---|---|
| XSS | Yes (5) | 1 | 2 |
| SQLi | Yes (2) | 0 | 0 |
| Blind SQLi | No | 0 | 0 |
| Traversal | No | 0 | 0 |
| Command Injection | No | 0 | 0 |
| Application Error | Yes(8) | 0 | 0 |
| Bruce Force | YES (1) | 0 | 0 |
| Bad HTTP Methods | Yes (1) | 0 | 0 |
| Directory Listing | YES (2) | 0 | 0 |
| Session/ Cookie flags | YES (2) | 0 | 0 |
| Session fixation | No | 0 | 0 |
| Unencrypted View State | Yes (1) | 0 | 0 |
| Weak Cookie | Yes (1) | 0 | 0 |
| Sensitive Info In URL | Yes (3) | 1 | 0 |

### Fig 4.9 – .NET Post-Authentication (Administrator)

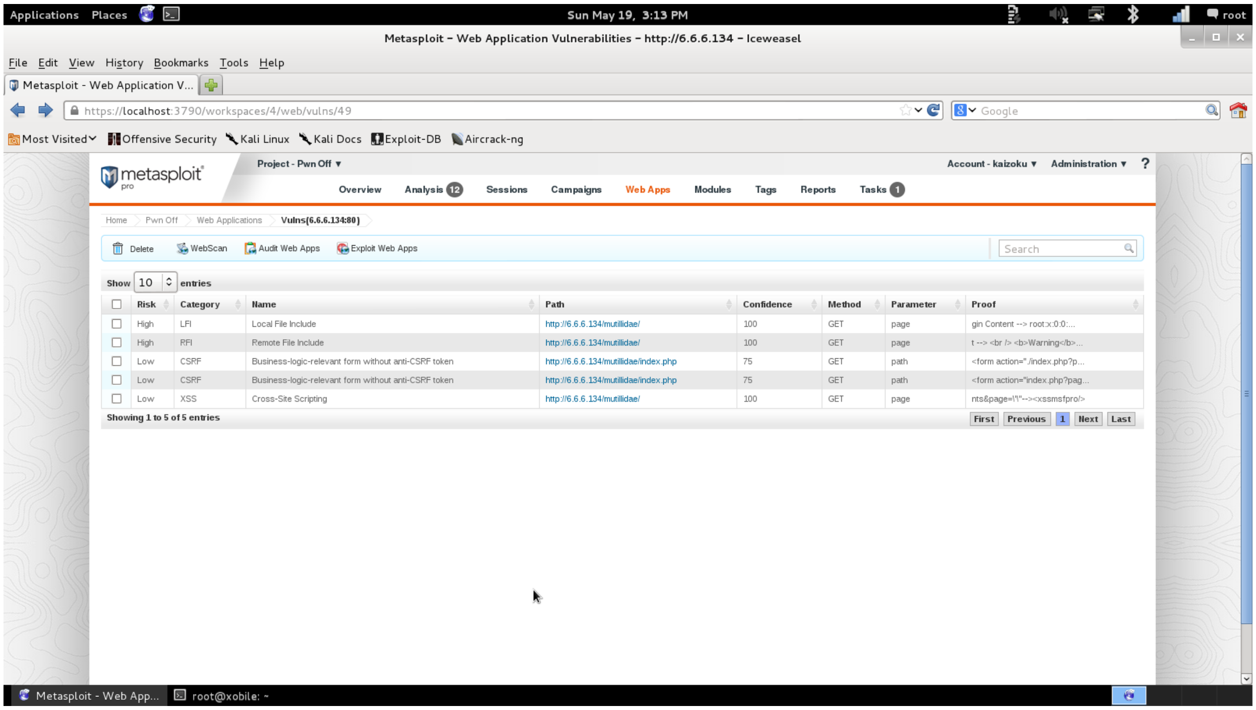
| Vulnerability | Detected | False Positives | False Negatives |
|---|---|---|---|
| XSS | **Yes (7)** | **1** | **0** |
| SQLi | **Yes (3)** | **0** | **0** |
| Blind SQLi | **Yes (1)** | **0** | **0** |
| Traversal | **No** | **0** | **0** |
| Command Injection | **No** | **0** | **0** |
| Application Error | **Yes (9)** | **0** | **0** |
| Bruce Force | **Yes (1)** | **0** | **0** |
| Bad HTTP Methods | **Yes (1)** | **0** | **0** |
| Directory Listing | **Yes (3)** | **0** | **0** |
| Session/ Cookie flags | **Yes (2)** | **0** | **0** |
| Session fixation | **No** | **0** | **0** |
| Unencrypted View State | **Yes (1)** | **0** | **0** |
| Weak Cookie | **Yes (1)** | **0** | **0** |
| Sensitive Info In URL | **Yes (4)** | **1** | **0** |

## RAPID7 NEXPOSE RATING SCORES

### Ease of Interface

Rapid 7's Nexpose interface is lightweight, and is accessed through a web browser. For the purpose of this paper, we are focusing on the web application testing portion of Nexpose. Nexpose is part of a suite of tools from Rapid 7 that does more than web application testing, such as Metasploit. Rapid 7 also distributes Nexpose as in its own VM. Running Nexpose was streamlined and it did not lag or take up a lot of memory. For ease of interface, Nexpose gets a score of **5/5**

### Fig 4.10 – Nexpose Web Interface



| □ | Risk | Category | Name | Path | Confidence | Method | Parameter | Proof |
|---|---|---|---|---|---|---|---|---|
| □ | High | LFI | Local File Include | http://6.6.6.134/mutillidae/ | 100 | GET | page | gin Content --> root:x:0:0:... |
| □ | High | RFI | Remote File Include | http://6.6.6.134/mutillidae/ | 100 | GET | page | t --> <br /> <b>Warning</b>... |
| □ | Low | CSRF | Business-logic-relevant form without anti-CSRF token | http://6.6.6.134/mutillidae/index.php | 75 | GET | path | <form action="./index.php?p... |
| □ | Low | CSRF | Business-logic-relevant form without anti-CSRF token | http://6.6.6.134/mutillidae/index.php | 75 | GET | path | <form action="index.php?pag... |
| □ | Low | XSS | Cross-Site Scripting | http://6.6.6.134/mutillidae/ | 100 | GET | page | nts&page=\'\"--><xssmsfpro/> |

### Vulnerability Detection

Nexpose has a comprehensive knowledgebase and a great engine for detecting vulnerabilities. During the PwnOff, Nexpose yielded no false negatives. Nexpose was able to find vulnerabilities in web applications just as well as it was able drop shells on WindowsXP boxes. Nexpose gets a vulnerability detection score of **5/5**

### Reporting

Nexpose has a reporting feature where it will export findings into comprehensive PDF file. For some findings, Nexpose will print out the entire Response Body for each variant, which can lead the report being verbose.

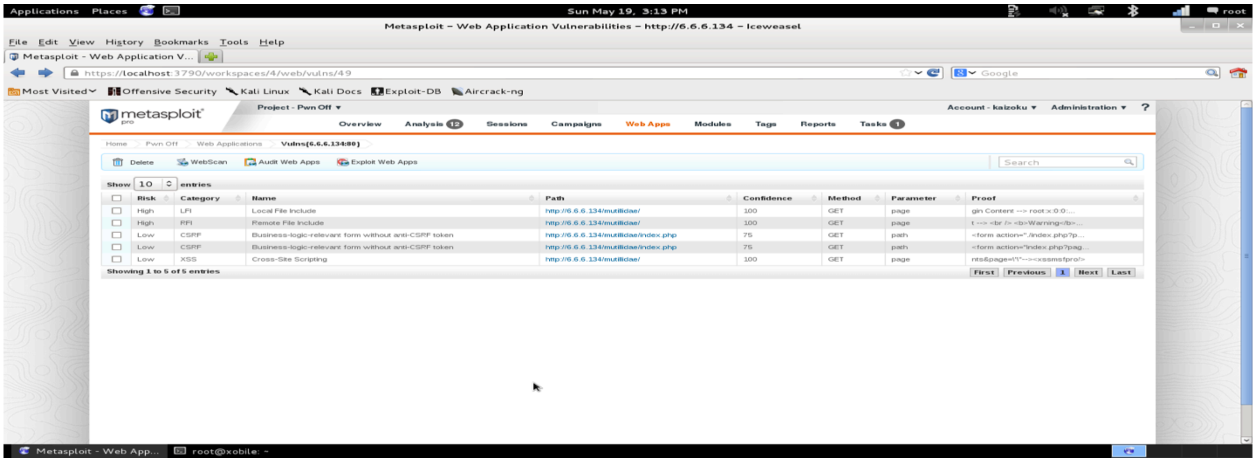
One interesting metric Nexpose calculates is the amount of vulnerabilities that will be remediated by applying the “Top 25” fixes. Nexpose will also save any text files, such as a comprised /etc/passwd file, into the report.  For its comprehensive reporting capabilities, Nexpose gets a score of **4.8/5**

### Post-Exploitation Features

In the HackMiami 2013 PwnOff, Metasploit Pro from Rapid 7 was also featured as an add-on to Nexpose. The integration of Metasploit Pro allows the user to engage in exploitation and post-exploitation against identified vulnerabilities.

Figure 4.11 shows an example of web exploitation performed by Metasploit Pro. Metasploit detects and successfully exploits an LFI vulnerability against the target.

**Fig 4.11 – Metasploit Pro exploiting LFI**



Metasploit – Web Application Vulnerabilities – http://6.6.6.134 – Iceweasel

https://localhost:3790/workspaces/4/web/vulns/49

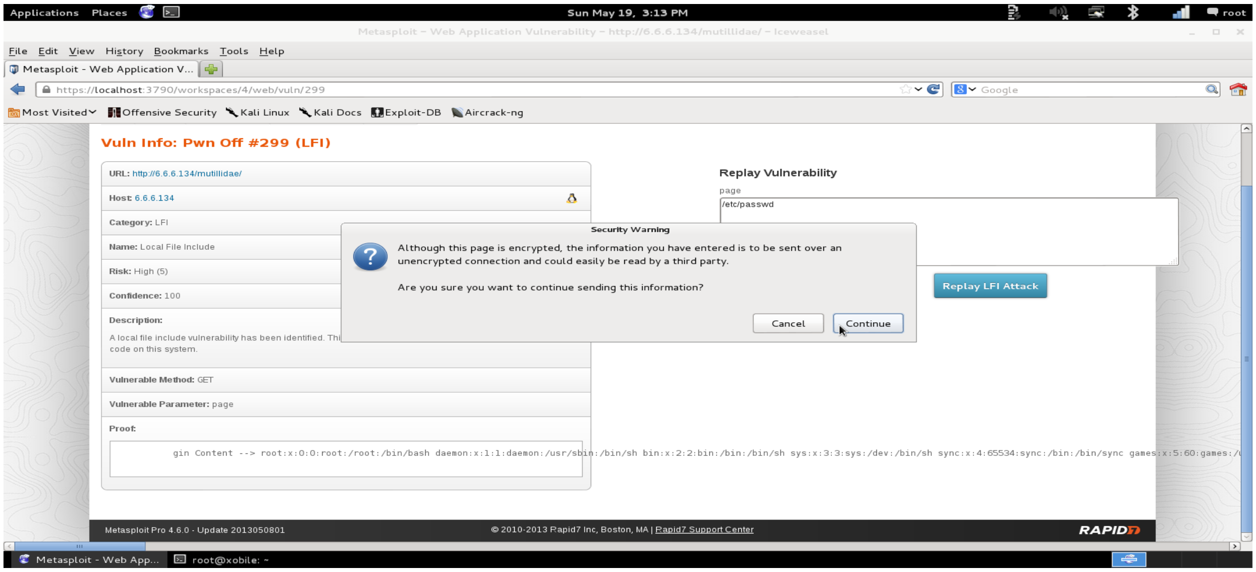
Project - Pwn Off

Home > Pwn Off > Web Applications > Vulns(6.6.6.134:80)

Show 10 entries

| | Risk | Category | Name | Path | Confidence | Method | Parameter | Proof |
|---|---|---|---|---|---|---|---|---|
| □ | High | LFI | Local File Include | http://6.6.6.134/mutillidae/ | 100 | GET | page | gin Content --> root:x:0:0:... |
| □ | High | RFI | Remote File Include | http://6.6.6.134/mutillidae/ | 100 | GET | page | t --> <br /> <b>Warning</b>... |
| □ | Low | CSRF | Business-logic-relevant form without anti-CSRF token | http://6.6.6.134/mutillidae/index.php | 75 | GET | path | <form action="./index.php?p... |
| □ | Low | CSRF | Business-logic-relevant form without anti-CSRF token | http://6.6.6.134/mutillidae/index.php | 75 | GET | path | <form action="index.php?pag... |
| □ | Low | XSS | Cross-Site Scripting | http://6.6.6.134/mutillidae/ | 100 | GET | page | nts&page=\"\'--><xssmsfpro/> |

Showing 1 to 5 of 5 entries



Metasploit – Web Application Vulnerability – http://6.6.6.134/mutillidae/ – Iceweasel

https://localhost:3790/workspaces/4/web/vuln/299

**Vuln Info: Pwn Off #299 (LFI)**

**URL:** http://6.6.6.134/mutillidae/

**Host:** 6.6.6.134

**Category:** LFI

**Name:** Local File Include

**Risk:** High (5)

**Confidence:** 100

**Description:**

A local file include vulnerability has been identified. Thi
code on this system.

**Vulnerable Method:** GET

**Vulnerable Parameter:** page

**Proof:**

```
        gin Content --> root:x:0:0:root:/root:/bin/bash daemon:x:1:1:daemon:/usr/sbin:/bin/sh bin:x:2:2:bin:/bin:/bin/sh sys:x:3:3:sys:/dev:/bin/sh sync:x:4:65534:sync:/bin:/bin/sync games:x:5:60:games:/
```

**Replay Vulnerability**

page: /etc/passwd

**Security Warning**

Although this page is encrypted, the information you have entered is to be sent over an unencrypted connection and could easily be read by a third party.

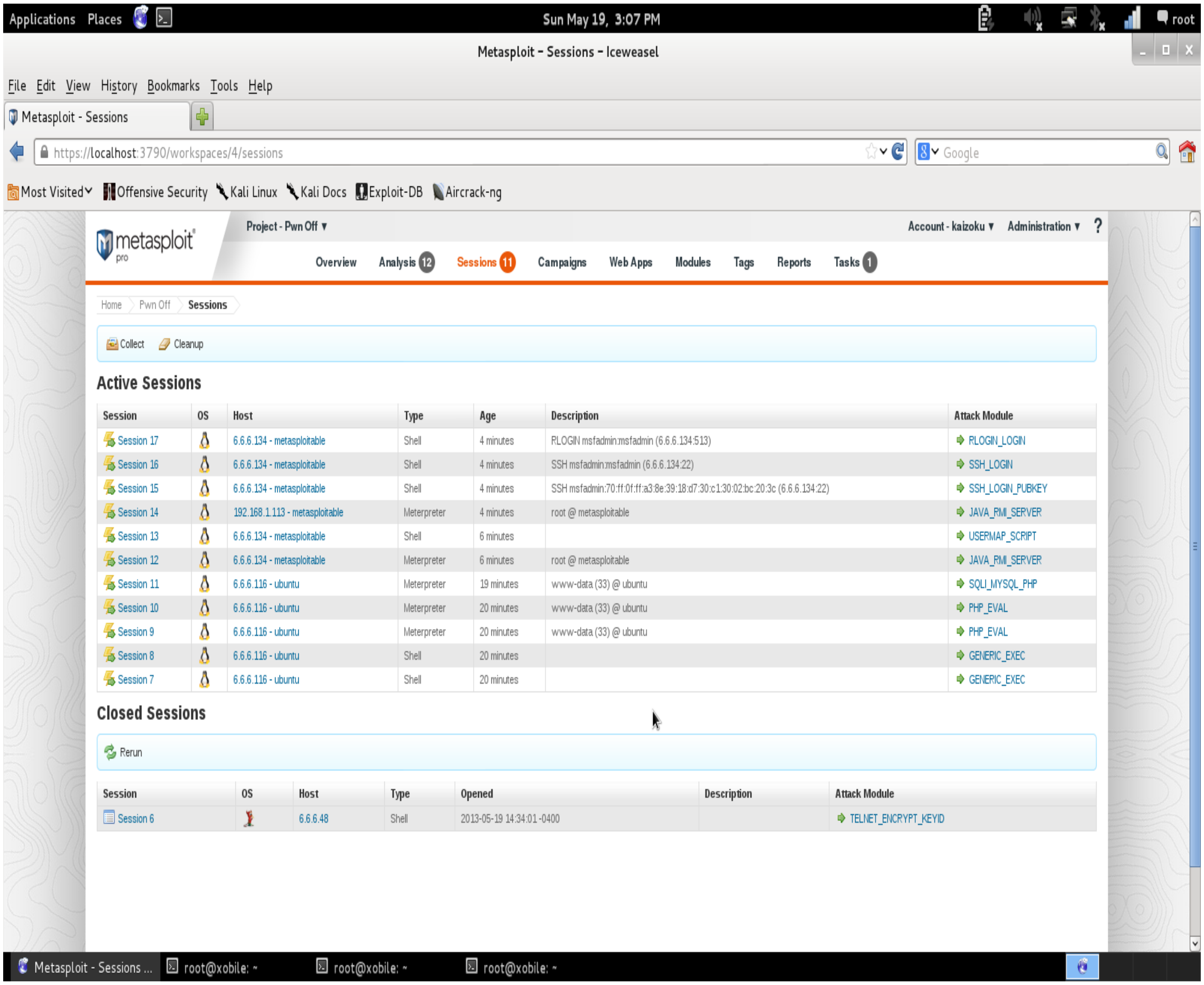
Are you sure you want to continue sending this information?

Metasploit Pro 4.6.0 - Update 2013050801

© 2010-2013 Rapid7 Inc, Boston, MA | Rapid7 Support Center

Having tools like Nexpose integrated with Metasploit Pro allows the vulnerability analyst the ability to streamline tasks and perform more assessments in a shorter amount of time.  Users are also able to manage multiple sessions during scenarios with multiple exploited targets.

Fig 4.13 – Active sessions with Metasploit Pro



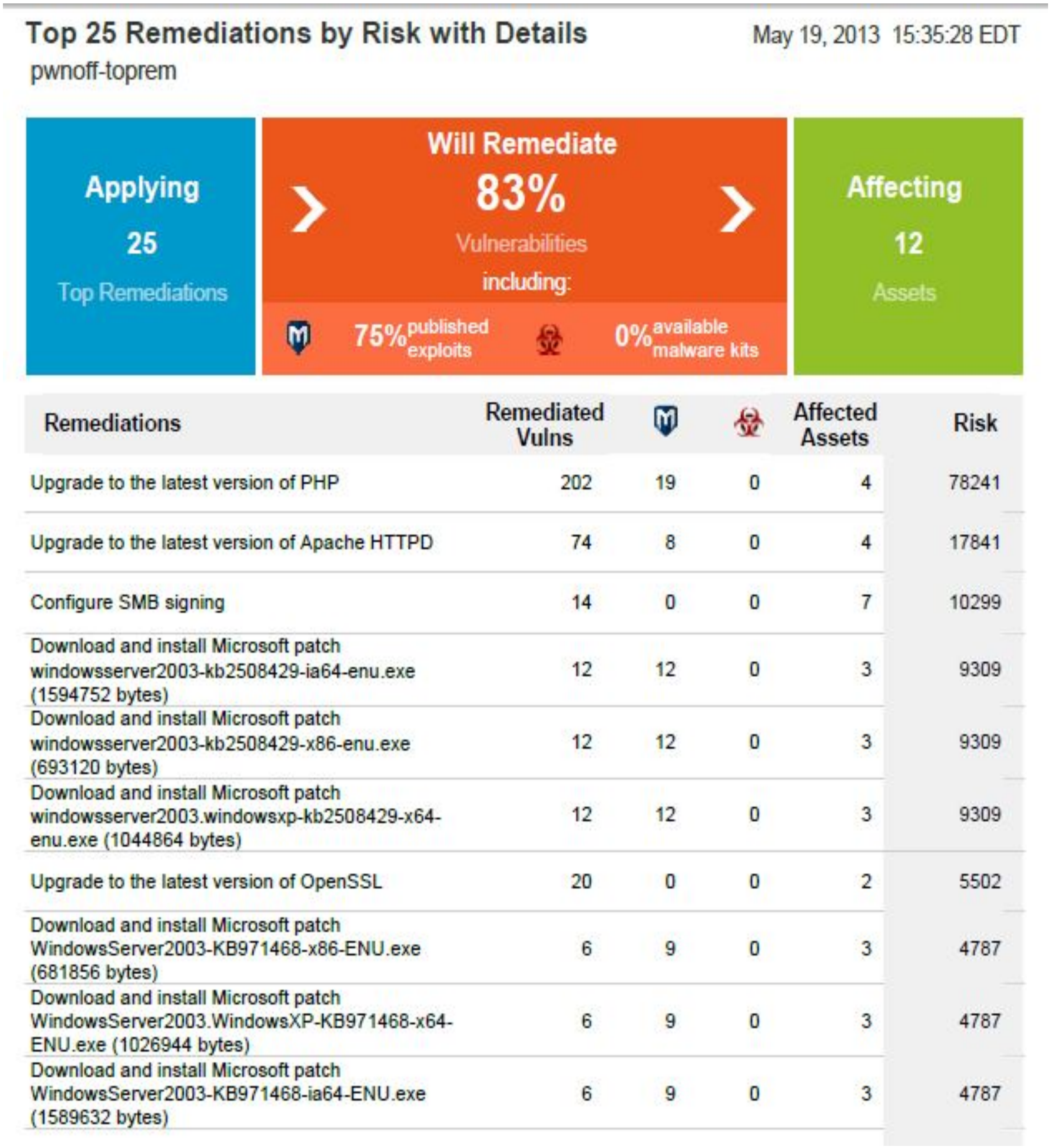
Active Sessions

| Session | OS | Host | Type | Age | Description | Attack Module |
|---|---|---|---|---|---|---|
| Session 17 | 🐧 | 6.6.6.134 - metasploitable | Shell | 4 minutes | RLOGIN msfadmin:msfadmin (6.6.6.134:513) | RLOGIN_LOGIN |
| Session 16 | 🐧 | 6.6.6.134 - metasploitable | Shell | 4 minutes | SSH msfadmin:msfadmin (6.6.6.134:22) | SSH_LOGIN |
| Session 15 | 🐧 | 6.6.6.134 - metasploitable | Shell | 4 minutes | SSH msfadmin:70:ff:0f:ff:a3:8e:39:18:d7:30:c1:30:02:bc:20:3c (6.6.6.134:22) | SSH_LOGIN_PUBKEY |
| Session 14 | 🐧 | 192.168.1.113 - metasploitable | Meterpreter | 4 minutes | root @ metasploitable | JAVA_RMI_SERVER |
| Session 13 | 🐧 | 6.6.6.134 - metasploitable | Shell | 6 minutes | | USERMAP_SCRIPT |
| Session 12 | 🐧 | 6.6.6.134 - metasploitable | Meterpreter | 6 minutes | root @ metasploitable | JAVA_RMI_SERVER |
| Session 11 | 🐧 | 6.6.6.116 - ubuntu | Meterpreter | 19 minutes | www-data (33) @ ubuntu | SQLI_MYSQL_PHP |
| Session 10 | 🐧 | 6.6.6.116 - ubuntu | Meterpreter | 20 minutes | www-data (33) @ ubuntu | PHP_EVAL |
| Session 9 | 🐧 | 6.6.6.116 - ubuntu | Meterpreter | 20 minutes | www-data (33) @ ubuntu | PHP_EVAL |
| Session 8 | 🐧 | 6.6.6.116 - ubuntu | Shell | 20 minutes | | GENERIC_EXEC |
| Session 7 | 🐧 | 6.6.6.116 - ubuntu | Shell | 20 minutes | | GENERIC_EXEC |

Closed Sessions

| Session | OS | Host | Type | Opened | Description | Attack Module |
|---|---|---|---|---|---|---|
| Session 6 | | 6.6.6.48 | Shell | 2013-05-19 14:34:01 -0400 | | TELNET_ENCRYPT_KEYID |

**Fig 4.14 – Nexpose identified vulnerabilities within the K&&K CTF**



## Top 25 Remediations by Risk with Details

pwnoff-toprem

May 19, 2013 15:35:28 EDT

Applying 25 Top Remediations

Will Remediate 83% Vulnerabilities including:

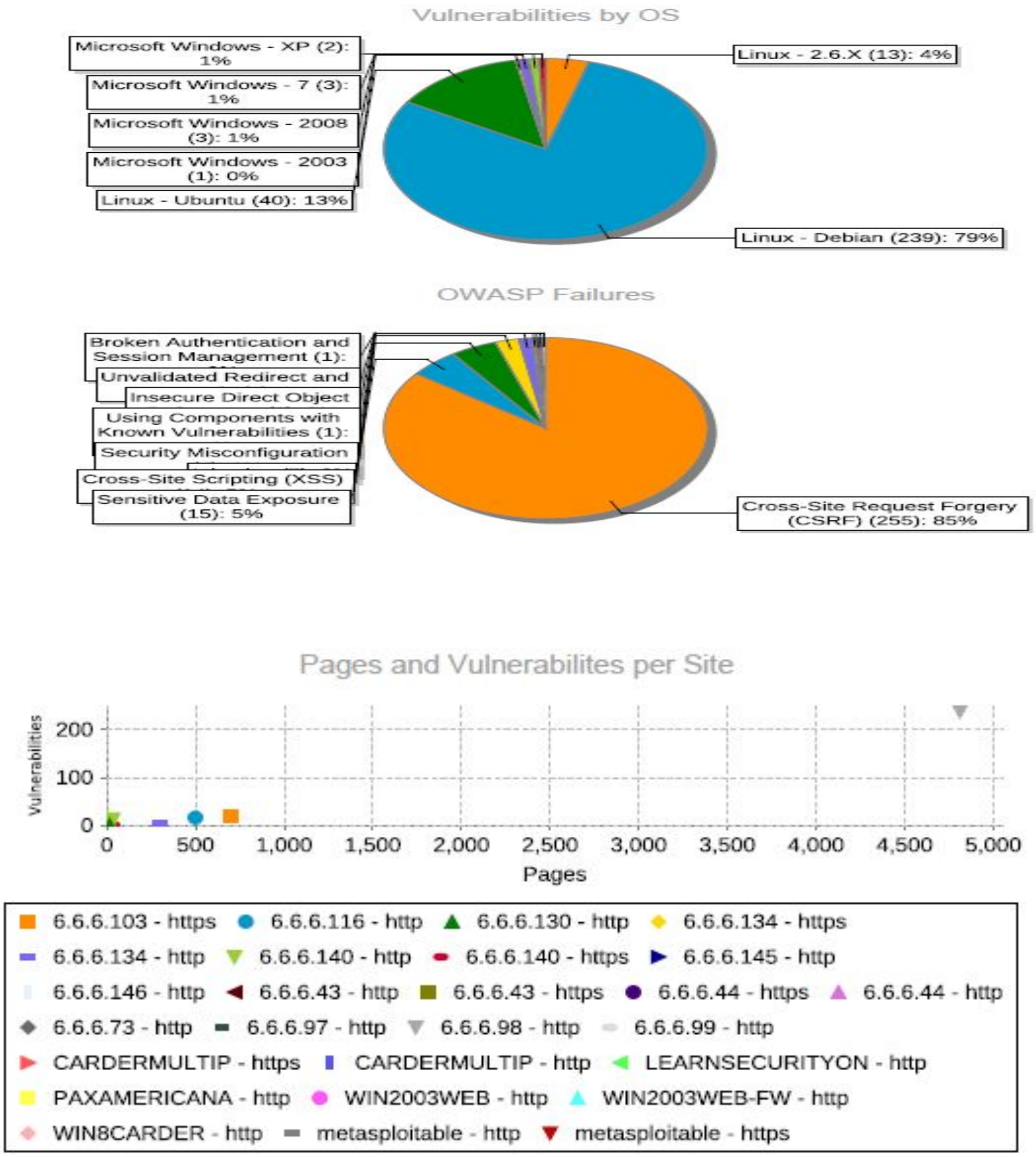
M 75% published exploits

☣ 0% available malware kits

Affecting 12 Assets

| Remediations | Remediated Vulns | M | ☣ | Affected Assets | Risk |
|---|---|---|---|---|---|
| Upgrade to the latest version of PHP | 202 | 19 | 0 | 4 | 78241 |
| Upgrade to the latest version of Apache HTTPD | 74 | 8 | 0 | 4 | 17841 |
| Configure SMB signing | 14 | 0 | 0 | 7 | 10299 |
| Download and install Microsoft patch windowsserver2003-kb2508429-ia64-enu.exe (1594752 bytes) | 12 | 12 | 0 | 3 | 9309 |
| Download and install Microsoft patch windowsserver2003-kb2508429-x86-enu.exe (693120 bytes) | 12 | 12 | 0 | 3 | 9309 |
| Download and install Microsoft patch windowsserver2003.windowsxp-kb2508429-x64-enu.exe (1044864 bytes) | 12 | 12 | 0 | 3 | 9309 |
| Upgrade to the latest version of OpenSSL | 20 | 0 | 0 | 2 | 5502 |
| Download and install Microsoft patch WindowsServer2003-KB971468-x86-ENU.exe (681856 bytes) | 6 | 9 | 0 | 3 | 4787 |
| Download and install Microsoft patch WindowsServer2003.WindowsXP-KB971468-x64-ENU.exe (1026944 bytes) | 6 | 9 | 0 | 3 | 4787 |
| Download and install Microsoft patch WindowsServer2003-KB971468-ia64-ENU.exe (1589632 bytes) | 6 | 9 | 0 | 3 | 4787 |

**Fig 4.15 – Visualizations of Identified vulnerabilities**



## Overall Value

The Nexpose Enterprise edition, which includes the web scanner, costs $20,000 for the license. Although it is more expensive than the other tools, the comprehensive exploitation framework of Metasploit Pro ensures that the tester will will be getting a lot more than just a web application test suite. **5/5**

# NTO OBJECTIVE NTOSPIDER

## PHP APPLICATION SCAN RESULTS

Figures 5.1 – 5.3 displays the results of the NTOSpider scan against the PHP web application.

### Fig 5.1 – PHP Pre-authentication scan

| Vulnerability | Detected | False Positives | False Negatives |
|---|---|---|---|
| XSS | **No** | **0** | **2** |
| SQLi | **No** | **0** | **0** |
| Blind SQLi | **No** | **0** | **0** |
| Traversal | **No** | **0** | **0** |
| CSRF | **Yes (3)** | **3*** | **0** |
| Command Injection | **No** | **0** | **0** |
| Application Error | **Yes (4)** | **0** | **0** |
| Bruce Force | **Yes (1)** | **0** | |
| Bad HTTP Methods | **No** | **0** | **0** |
| HTTP Response Splitting | **No** | **0** | **0** |
| Directory Listing | **Yes (8)** | **0** | **0** |
| Informational** | **Yes (6)** | **-** | **-** |

### Fig 5.2 - PHP Post-authentication (user)

| Vulnerability | Detected | False Positives | False Negatives |
|---|---|---|---|
| XSS | **Yes (3)** | **0** | **3** |
| SQLi | **Yes (1)** | **0** | **1** |
| Blind SQLi | **Yes (1)** | **0** | **0** |
| Traversal | **Yes(1)** | **0** | **0** |
| CSRF | **Yes (6)** | **3*** | **0** |
| Command Injection | **No** | **0** | **0** |
| Application Error | **Yes(8)** | **0** | **0** |
| Bruce Force | **No** | **0** | **0** |
| Bad HTTP Methods | **No** | **0** | **0** |
| Directory Listing | **YES(8)** | **0** | **0** |
| Session/ Cookie flags | **YES(2)** | **0** | **0** |
| Session fixation | **Yes** | **0** | **0** |
| Weak Cookie | **No** | **0** | **0** |
| Sensitive Info In URL | **Yes(8)** | **0** | **0** |

### Fig 5.3 - PHP Post-authentication scan (Administrator)

| Vulnerability | Detected | False Positives | False Negatives |
|---|---|---|---|
| XSS | Yes (3) | 0 | 3 |
| SQLi | Yes (2) | 0 | 1 |
| Blind SQLi | Yes (1) | 0 | 0 |
| Traversal | Yes (1) | 0 | 0 |
| CSRF | Yes (9) | 6* | 0 |
| Command Injection | No | 0 | 1 |
| Application Error | Yes (9) | 0 | 0 |
| Bruce Force | No | 0 | 1 |
| Bad HTTP Methods | Yes (1) | 0 | 0 |
| Directory Listing | Yes (9) | 0 | 0 |
| Session/ Cookie flags | Yes (2) | 0 | 0 |
| Session fixation | Yes | 0 | 0 |
| Weak Cookie | Yes | 0 | 0 |
| Sensitive Info In URL | Yes (9) | 0 | 0 |

## NTOSpider JSP APPLICATION SCAN RESULTS

Figures 5.4 – 5.6 displays the results of the NTOSpider scan against the JSP web application.

### Fig 5.4 - JSP Pre-Authentication

| Vulnerability | Detected | False Positives | False Negatives |
|---|---|---|---|
| XSS | No | 0 | 0 |
| SQLi | No | 0 | 0 |
| Blind SQLi | No | 0 | 0 |
| Traversal | No | 0 | 0 |
| CSRF | Yes (1) | 1* | 0 |
| Command Injection | No | 0 | 0 |
| Application Error | Yes(4) | 0 | 0 |
| Bruce Force | YES(1) | 0 | 0 |
| Bad HTTP Methods | No | 0 | 0 |

| | | | |
|---|---|---|---|
| HTTP Response Splitting | **No** | **0** | **0** |
| Directory Listing | **YES(2)** | **0** | **0** |
| Informational** | **YES(5)** | **-** | **-** |

## Fig 5.5 – JSP Post-Authentication (User)

| **Vulnerability** | **Detected** | **False Positives** | **False Negatives** |
|---|---|---|---|
| XSS | **Yes (3)** | **0** | **1** |
| SQLi | **Yes (2)** | **0** | **1** |
| Blind SQLi | **Yes (3)** | **2** | **0** |
| Traversal | **Yes (1)** | **0** | **0** |
| CSRF | **Yes (3)** | **1*** | **0** |
| Command Injection | **No** | **0** | **0** |
| Application Error | **Yes (8)** | **0** | **0** |
| Bruce Force | **Yes (1)** | **0** | **0** |
| Bad HTTP Methods | **No** | **0** | **0** |
| Directory Listing | **Yes (2)** | **0** | **0** |
| Session/ Cookie flags | **Yes (2)** | **0** | **0** |
| Session fixation | **No** | **0** | **0** |
| Weak Cookie | **No** | **0** | **0** |
| Sensitive Info In URL | **Yes (4)** | **2** | **0** |

## Fig 5.6 – JSP Post-Authentication (Administrator)

| **Vulnerability** | **Detected** | **False Positives** | **False Negatives** |
|---|---|---|---|
| XSS | **Yes (3)** | **0** | **1** |
| SQLi | **Yes (2)** | **0** | **1** |
| Blind SQLi | **Yes (3)** | **2** | **0** |
| Traversal | **Yes (1)** | **0** | **0** |
| CSRF | **Yes (4)** | **1*** | **0** |
| Command Injection | **No** | **0** | **1** |
| Application Error | **Yes (9)** | **0** | **0** |
| Bruce Force | **Yes (1)** | **0** | **0** |

| | | | |
|---|---|---|---|
| Bad HTTP Methods | **Yes (1)** | **0** | **0** |
| Directory Listing | **Yes (3)** | **0** | **0** |
| Session/ Cookie flags | **Yes (2)** | **0** | **0** |
| Session fixation | **No** | **0** | **0** |
| Weak Cookie | **Yes (1)** | **0** | **0** |
| Sensitive Info In URL | **Yes (6)** | **4** | **0** |

### NTOSpider .NET APPLICATION SCAN RESULTS

Figures 5.7 – 5.9 displays the results of the NTOSpider scan against the .NET web application.

**Fig 5.7 – .NET Pre-Authentication**

| Vulnerability | Detected | False Positives | False Negatives |
|---|---|---|---|
| XSS | **No** | **0** | **0** |
| SQLi | **No** | **0** | **0** |
| Blind SQLi | **No** | **0** | **0** |
| Traversal | **No** | **0** | **0** |
| CSRF | **No** | **0** | **0** |
| Command Injection | **No** | **0** | **0** |
| Application Error | **Yes (2)** | **0** | **0** |
| Bruce Force | **Yes (1)** | **0** | **0** |
| Bad HTTP Methods | **Yes (1)** | **0** | **0** |
| Unencrypted View State | **Yes (1)** | **0** | **0** |
| HTTP Response Splitting | **No** | **0** | **0** |
| Directory Listing | **Yes (2)** | **0** | **0** |
| Informational** | **Yes (12)** | **-** | **-** |

**Fig 5.8 – .NET Post-Authentication (User)**

| Vulnerability | Detected | False Positives | False Negatives |
|---|---|---|---|
| XSS | **Yes (3)** | **0** | **3** |
| SQLi | **Yes (2)** | **0** | **0** |
| Blind SQLi | **No** | **0** | **0** |
| Traversal | **No** | **0** | **0** |
| Command Injection | **No** | **0** | **0** |

| | | | |
|---|---|---|---|
| Application Error | **Yes(8)** | **0** | **0** |
| Bruce Force | **YES(1)** | **0** | **0** |
| Bad HTTP Methods | **Yes (1)** | **0** | **0** |
| Directory Listing | **YES(2)** | **0** | **0** |
| Session/ Cookie flags | **YES(2)** | **0** | **0** |
| Session fixation | **No** | **0** | **0** |
| Unencrypted View State | **Yes(1)** | **0** | **0** |
| Weak Cookie | **Yes(1)** | **0** | **0** |
| Sensitive Info In URL | **Yes(4)** | **3** | **0** |

### Fig 5.9 – .NET Post-Authentication (Administrator)

| **Vulnerability** | **Detected** | **False Positives** | **False Negatives** |
|---|---|---|---|
| XSS | **Yes (4)** | **0** | **2** |
| SQLi | **Yes (3)** | **0** | **0** |
| Blind SQLi | **Yes (2)** | **1** | **0** |
| Traversal | **No** | **0** | **0** |
| Command Injection | **No** | **0** | **0** |
| Application Error | **Yes (9)** | **0** | **0** |
| Bruce Force | **Yes (1)** | **0** | **0** |
| Bad HTTP Methods | **Yes (1)** | **0** | **0** |
| Directory Listing | **Yes (3)** | **0** | **0** |
| Session/ Cookie flags | **Yes (2)** | **0** | **0** |
| Session fixation | **No** | **0** | **0** |
| Unencrypted View State | **Yes (1)** | **0** | **0** |
| Weak Cookie | **Yes (1)** | **0** | **0** |
| Sensitive Info In URL | **Yes (8)** | **5** | **0** |

### Interface

The NTOSpider Interface is clean, but also resource intensive. Navigating around the tool and configuring the scans was a simple process. NTOSpider did have a slight issue in maintaining the Session, and testers had to do a bit more configuration that with other tools.

NTOSpider prides itself on testing the Web 3.0 Technologies such as JSON and REST queries. For these features to work, the tester would need to use another proxy tool like BURP to import any Web 3.0 technologies like GWT. Creating login and crawling macros is simple to do.

Overall, the tool is easy to use and easy to navigate with. NTOSpider receives an interface score of **4/5**

### Vulnerability Detection

During testing NTOSpider, missed some of the Cross Site Scripting (XSS) on the PHP site. On the official website, they do disclaim that NTOSpider does not try to check for known and easy vulnerabilities, instead it caters its scans towards web service testing.

The scanner detected a majority of the security issues, so NTOSpider gets a vulnerability detection score of **3/5**

### Reporting

NTOSpider has a reporting feature that allows you to order vulnerabilities by categories and types. The tool generates the reports initially as a web page and it allows the user to validate the finding through BURP by clicking on the "Validate Finding" button. The reports are very customizable, the "Validate Finding" feature is great to proof of concept vulnerabilities . NTOSpider gets a score of **4/5**

### Overall Value

Currently, NTOSpider costs $10,000 for a license. While the tool is promising, the tool is not within reach of an independent consultant. NTOSpider is gearing itself for the newer technologies such as mobile web application testing, but it may be missing out on some older and more common vulnerabilities. For Overall Value, NTOSpider receives a score of **3/5**

## Conclusion

Ongoing cyclical web application vulnerability assessments are a critical part of the software development lifecycle (SDLC) for any organization. The harried release cycles of web applications and scarce availability of skilled security engineers to conduct thorough manual assessments makes the market for automated web application vulnerability scanner suites one that will continue to grow. As more products come to market, and more exploitable vulnerabilities are identified, the choices will continue to grow. The end consumer will almost always be faced with picking a product that meets their strictest requirement, the budget.

In terms of overall value, it is the conclusion of the researchers conducting the HackMIami 2013 Hackers Conference PwnOff that Portswigger BURP and Rapid7 Nexpose/Metasploit Pro currently provide the most value to the independent security consultant in terms of discovered vulnerabilities, ease of use, licensing flexibility, and rage of functionality.

## About HackMiami

HackMiami is the premier resource in South Florida for the recruitment of highly skilled hackers that specialize in vulnerability analysis, penetration testing, digital forensics, and all manner of information technology and security

HackMiami is made up of experienced information security professionals that have years of experience working with large corporations, governments, and small businesses.

Members of HackMiami are on the cutting edge of vulnerability research and regularly present at local information security group meetings (ISSA, OWASP) and international hacking conferences around the world (Defcon, HOPE, OWASP AppSec, Hacker Halted).

HackMiami seeks to develop and harness the participation of the information security community through regular meetings, presentations, labs and competitions. These events allow the hacker community a forum to present their research, develop new techniques and methodologies, and at the same time provide valuable a networking resource for contracting opportunities.

Visit HackMiami on the web at http://hackmiami.org