Assessment Report

Date Range: All

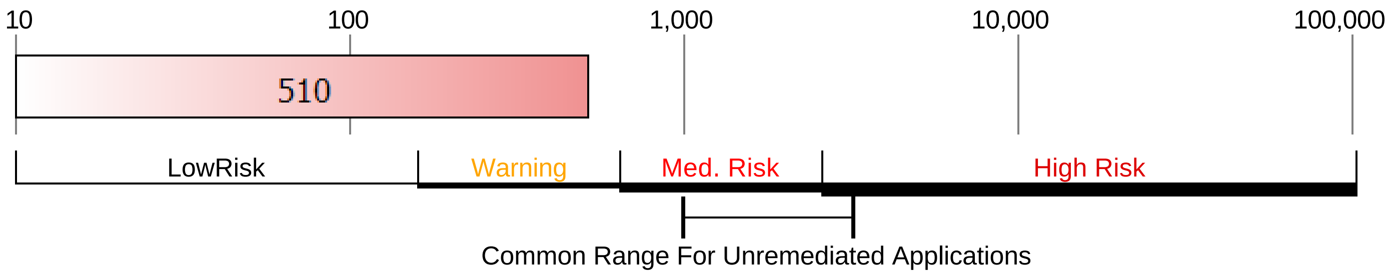
Application(s) Assessed: QA_MyPremiumTitle

Summary

An assessment was performed on the QA_MyPremiumTitle application, using 49 SmartAttacks ™ to identify vulnerabilities. As shown below, issues were found with the application that resulted in a HARM™ score of **510** . Typical HARM scores for complete assessments of individual non-remediated applications range from "1000 to 4500"(although large numbers of any one type of vulnerability can sometimes drive up scores to much higher levels).It is recommended that action be taken to remediate these vulnerabilities. The tests performed, and specific issues to be reviewed, and their contribution to the total HARM score can be viewed in the "SmartAttack Results (by HARM)" section of this report below.

Total Harm™ Score: 510 = 510 (Raw Scores) x 1.0 (App Risk Factor)

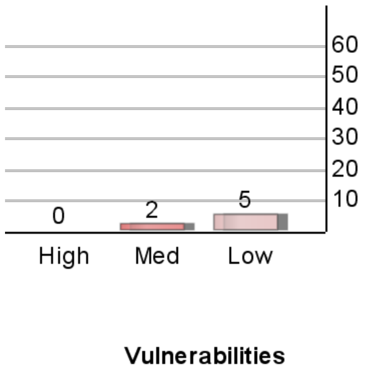
HARM Scoring

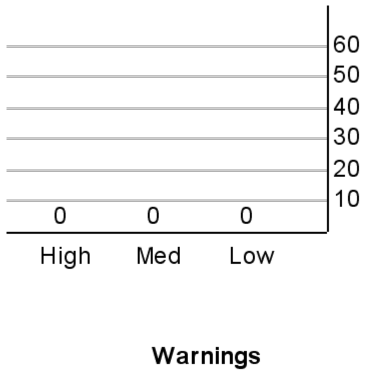


The **'Total HARM Score'**, above, is a sum of the HARM scores for all the SmartAttack assessments included in this report. SmartAttacks have different HARM scores based on the risks associated with each kind of vulnerability. The charts reflect the raw HARM scores without application specific risk adjustments.

Severity of Findings

SmartAttack Result Summary





Pages Tested	43
Attack Count	60138

Note: High, Med. and Low relate to the severity of the findings. Warnings are findings for which there is less confidence of being real vulnerabilities.

High Severity

SmartAttack	Vulnerability	Warning

Medium and Low Severity

SmartAttack	Vulnerability	Warning
🟧 Application Path Disclosure	1	0
🟨 Cookie Vulnerabilities	1	0
🟧 Cross-Frame Scripting	1	0
🟨 GET for POST	4	0

SmartAttack Results (by HARM)

SmartAttack Name	Status	Severity	HARM	Vulnerability	Warning	Info	Doc
Application Path Disclosure	Note	Medium	100	1	0	0	Doc
Cross-Frame Scripting	Note	Medium	120	1	0	0	Doc
GET for POST	Note	Low	200	4	0	0	Doc
Cookie Vulnerabilities	Note	Low	90	1	0	1	Doc
Cookie Vulnerabilities	Ok	Medium	0	0	0	1	Doc
Application Exception	Ok	Medium	0	0	0	0	Doc
Authentication Bypass - [OWASP 2013 A 2]	Ok	High	0	0	0	1	Doc
Blind SQL Injection	Ok	High	0	0	0	1	Doc
Browse HTTP from HTTPS List - [OWASP 2013 A 5]	Ok	Default	0	0	0	0	Doc
Buffer Overflow	Ok	High	0	0	0	0	Doc
Check HTTP Methods	Ok	Medium	0	0	0	1	Doc
Credit Card Disclosure [PCI 6.5.4]	Ok	Default	0	0	0	1	Doc
Cross Site Request Forgery [PCI 6.5.9]	Ok	Default	0	0	0	0	Doc
Cross-Site Scripting	Ok	High	0	0	0	0	Doc
Cross-Site Scripting Through Flash	Ok	High	0	0	0	0	Doc
Directory Browsing	Ok	Medium	0	0	0	0	Doc
DOM Based Cross-Site Scripting	Ok	High	0	0	0	0	Doc
File & Directory Discovery	Ok	Medium	0	0	0	2	Doc
Form Caching	Ok	Medium	0	0	0	0	Doc
Format String	Ok	High	0	0	0	0	Doc
Forms Submitted Without Using Post	Ok	Medium	0	0	0	0	Doc
Frame Injection	Ok	Medium	0	0	0	0	Doc
HTML & JavaScript Comments	Ok	Low	0	0	0	14	Doc
HTTP Parameter Pollution	Ok	Low	0	0	0	0	Doc
HTTP Response Splitting	Ok	High	0	0	0	0	Doc
Integer Overflow	Ok	Medium	0	0	0	0	Doc
LDAP Exception	Ok	Medium	0	0	0	0	Doc
LDAP Injection	Ok	High	0	0	0	1	Doc
Login Redirect - [OWASP 2013 A 2]	Ok	High	0	0	0	0	Doc
Non-SSL Form	Ok	High	0	0	0	0	Doc
Non-SSL Password	Ok	High	0	0	0	0	Doc
Open Redirect	Ok	Medium	0	0	0	0	Doc
Parameter Addition	Ok	Medium	0	0	0	0	Doc
Password Autocomplete	Ok	Medium	0	0	0	0	Doc
PHP & Perl Code Injection	Ok	Medium	0	0	0	0	Doc
Platform Path Disclosure - [OWASP 2013 A 5]	Ok	Default	0	0	0	0	Doc
Private IP Disclosure	Ok	Low	0	0	0	0	Doc
Redirection Through Flash	Ok	Medium	0	0	0	0	Doc
Remote File Inclusion	Ok	High	0	0	0	0	Doc
Social Security Disclosure [PCI 6.5.4]	Ok	Default	0	0	0	1	Doc
SQL Disclosure	Ok	High	0	0	0	1	Doc
SQL Error Message	Ok	Medium	0	0	0	0	Doc
SSI Injection	Ok	Medium	0	0	0	0	Doc
SSL Checks	Ok	Medium	0	0	0	2	Doc
Unix Command Injection	Ok	High	0	0	0	1	Doc
Unix Relative Path	Ok	High	0	0	0	0	Doc
Unrestricted File Upload	Ok	Medium	0	0	0	0	Doc
Web Server Vulnerabilities	Ok	Low	0	0	0	0	Doc

Windows Command Injection	Ok	High	0	0	0	1	Doc
Windows Relative Path	Ok	High	0	0	0	1	Doc

Detail (by SmartAttack / Report Item Type) SmartAttack Details

Note: *Detailed individual findings start on the next page.*

Detail (by HARM / SmartAttack)

Attack: Application Exception	Version: 1.7.4	Observations: 1
Description: Application Exceptions are vulnerabilities where unexpected inputs can trigger inappropriate exceptions, or error responses disclosing implementation information, such as a stack trace. The SmartAttack sends various unusual inputs and looks for text in responses evidencing poor error handling.		
Severity: Medium HARM: 96 Total HARM: 0	Online Documentation for: Application Exception	

Impact An attacker might gain administrative control of your web application by using vulnerabilities known to be in the servers or server-side technologies used. It is also be possible to gain remote access to restricted information via exploit of some vulnerability in your application.

An Application Exception vulnerability helps the attacker in formulating the correct attack depending on information disclosed in the error message. Such an error message may also disclose information about the deployment of the server, such as the database server used, server-side technology used, *etc*. Error messages often give an attacker useful information about how an application interacts with back-end components, and can reveal potential vulnerabilities within the application itself.

All Web applications use some server-side technology and back-end components which can be disclosed through error messages deliberately obtained by an attacker. For example, an error generated by a Web- based form for searching products in a shopping application may disclose the database or server-side technology it uses, their versions, *etc*. Or, errors generated by a Web-based banking application may expose further vulnerabilities such as Cross-Site Scripting present in the application.

Application Exception Pass Findings

Assessment: QA_MPT_Assessment **Run:** 10/17/2017 11:05:31AM **Traversal:** MPT_Trav

Application Exception
1 Pass (Medium, HARM: 0):
CVSS:
Message: `No vulnerabilities were found for: Application exception`

Detail (by HARM / SmartAttack)

Attack: Application Path Disclosure	Version: 1.1.1	CWE-200	Observations: 1
Description: This SmartAttack reports each page on which it finds path disclosure vulnerabilities, that is, where bad input can lead to an internal application error which reveals the path information. This SmartAttack exploits the application business logic to reveal path disclosures.			
Severity: Medium HARM: 100 Total HARM: 100	Online Documentation for: Application Path Disclosure		

Impact Error messages often give an attacker useful information about how an application interacts with back-end components, and can reveal potential vulnerabilities within the application itself.

Application Path Disclosure Vulnerable Findings

Assessment: QA_MPT_Assessment **Run:** 10/17/2017 11:05:31AM **Traversal:** MPT_Trav

Application Path Disclosure

1 Vulnerable (Medium, HARM: 100): https://www.qa-mypremiumtitle.com/Requester.aspx

CVSS: 5.0

Message:
```
Application Path Disclosure vulnerability found
Injectable request #: 4
Injected item: POST: viewpdfdata
Injection value: %00
Detection values:
  c:\inetpub\wwwroot\mptuat\requester.aspx.cs
  c:\inetpub\wwwroot\mptuat\requester.aspx.cs:
```

Detail (by HARM / SmartAttack)

Attack: Blind SQL Injection	Version: 2.7.0	CWE-89	Observations: 2
Description: Blind SQL Injection is a vulnerability caused by a Web application sending user input into a SQL query without validation. It is a type of SQL Injection vulnerability, where an attacker infers information from differences in responses observed for different injections. This SmartAttack injects pairs of SQL injections and looks for evidence of different responses.			
Severity: High HARM: 230 Total HARM: 0		Online Documentation for: Blind SQL Injection	

Impact A Blind SQL Injection vulnerability enables an attacker to gain access to information that he does not have the privileges for. This includes data stored in the tables and probably credentials for accessing sections of an application requiring higher privileges. Further, the attacker can elicit information about the database itself such as the tables therein, their structure, information about the users, etc.

Many Web applications store important information in SQL databases which may be leaked through SQL Injection even when all error messages are suppressed. For example, a cleverly crafted Blind SQL Injection attack on a form for finding customers of an online business may result into theft of credit card numbers and other sensitive information. Similarly, a Blind SQL Injection attack on an application storing financial information for an organization may reveal confidential salary structures.

Blind SQL Injection Information Findings

Assessment: QA_MPT_Assessment **Run:** 10/17/2017 11:05:31AM **Traversal:** MPT_Trav

Blind SQL Injection
1 Information (High, HARM: 0): https://www.qa-mypremiumtitle.com/Requester.aspx
CVSS:
Message:
```
No response from server.
Injectable request #: 4
Injected item: POST: viewpdfdata
Injection value:  or 13=3
```

Blind SQL Injection
2 Pass (High, HARM: 0):
CVSS:
Message:
```
No vulnerabilities were found for: Blind SQL Injection
```

Detail (by HARM / SmartAttack)

Attack: Browse HTTP from HTTPS List - [OWASP 2013 A 5]	Version: 1.2	CWE-200	Observations: 1
Description:	Browse HTTP from HTTPS is a vulnerability allowing HTTPS pages to be accessed via HTTP, thus disclosing potentially sensitive information. The SmartAttack attempts to access pages using HTTP, which it observes to be originally accessed via HTTPS. If allowed to access what appears to be the same content, it reports vulnerability.		
Severity: High HARM: 360 Total HARM: 0		Online Documentation for: Browse HTTP from HTTPS List - [OWASP 2013 A 5]	

Impact This type of vulnerability constitutes an access control weakness that can compromise the confidentiality of your data. Also, the availability of particular pages outside of a secured context can cause legitimate users to believe that the session is secure, and therefore submit private information in clear text.

For example, if credit card details are entered in a session which is accessed over https and if this session is accessible through http, then these details can be used by the attacker resulting in loss of confidentiality.

Browse HTTP from HTTPS List - [OWASP 2013 A 5] Pass Findings

Assessment: QA_MPT_Assessment **Run:** 10/17/2017 11:05:31AM **Traversal:** MPT_Trav

Browse HTTP from HTTPS List - [OWASP 2013 A 5]

1 Pass (High, HARM: 0):

CVSS:

Message: `No HTTPS pages were accessible via HTTP`

Detail (by HARM / SmartAttack)

Attack: Buffer Overflow	Version: 1.4.15	CWE-120	Observations: 1
Description: This SmartAttack reports each page on which it finds buffer overflow vulnerabilities.			
Severity: High HARM: 512 Total HARM: 0		Online Documentation for: Buffer Overflow	

Impact An attacker can crash the target application or the computer that is hosting it, potentially gaining administrative access to the host computer and all data and applications to which it has access.

Buffer Overflow Pass Findings

Assessment: QA_MPT_Assessment **Run:** 10/17/2017 11:05:31AM **Traversal:** MPT_Trav

Buffer Overflow
1 Pass (High, HARM: 0):
CVSS:
Message: `No vulnerabilities were found for: Buffer overflow`

Detail (by HARM / SmartAttack)

Attack: Cookie Vulnerabilities	Version: 1.1.2	CWE-539	Observations: 2
Description: A failure to specify proper attributes for cookies may result into stealing of cookie information through various attacks like Cross-Site Scripting (XSS) or a Man-In-The-Middle attack. Hence, this is a Vulnerability we call Cookie Vulnerabilities. This SmartAttack optionally reports each page where session cookies are set insecurely, persistently, without proper caching directives or without HTTPOnly attribute.			
Severity: Low HARM: 90 Total HARM: 90	Online Documentation for: Cookie Vulnerabilities		

Impact Insecure cookies: When a cookie is not set securely, then it is sent by the browser even with unencrypted requests, even if they are generated in an application using SSL encryption otherwise. If an attacker is able to intercept such requests, he can steal the cookie.

Persistent session handling cookies: When a session handling cookie is set persistently it allows the cookie to be valid even after a user terminates a session. Therefore an attacker can use a session cookie stored as a text file by the browser to access restricted information.

Cacheable Cookies: Such Cookies could be cached at a proxy or a gateway. It can result in serving cookie value that is out of date or stale. An attacker may also steal such cookies if he has compromised the said proxy or gateway.

Cookies with HTTPOnly attribute not set: If the HTTPOnly attribute is not set for a cookie, then it can be accessed and manipulated by JavaScript from the domain setting the cookie. The sensitive information contained in the cookie can be sent to a hacker's computer or Web site using a script-based attack such as Cross-Site Scripting.

Cookie Vulnerabilities Information Findings

Assessment: QA_MPT_Assessment **Run:** 10/17/2017 11:05:31AM **Traversal:** MPT_Trav

Cookie Vulnerabilities
1 Information (, HARM: 0):
CVSS:
Message:
```
Summary for Cookie Vulnerabilities:

Following cookies are missing  "Secure" attribute:
  _gid       qa-mypremiumtitle.com

Following cookies are missing "HTTPOnly" attribute
  _gid       qa-mypremiumtitle.com
```

Cookie Vulnerabilities
2 Vulnerable (Low, HARM: 90): https://www.qa-mypremiumtitle.com/ibbLogin.aspx
CVSS: 5.0
Message:
```
_gid Cookie has problem(s)

 _gid = GA1.2.2089595188.1508252776;
 Host = qa-mypremiumtitle.com;
 Path = /
   1. Cookie does not have secure attribute.
   2. Cookie does not have HTTPOnly attribute.
```

Detail (by HARM / SmartAttack)

Attack: Cross Site Request Forgery [PCI 6.5.9]	Version: 2.5	Observations: 1
Description: Cross-site Request Forgery vulnerabilities allow unauthorized requests from a victim's machine to improperly initiate transactions using an existing authenticated session. The SmartAttack, in combination with utilities used to identify session ids, identifies cases where the application is using only cookies to maintain and communicate session ids, and is therefore vulnerable.		
Severity: High HARM: 256 Total HARM: 0		Online Documentation for: Cross Site Request Forgery [PCI 6.5.9]

Impact If a Web application is vulnerable to CSRF, an attacker can make the victim's browser send unauthorized HTTP requests on behalf of the victim without her knowledge. This allows an attacker to perform all the legitimate actions which a legitimate user can perform after a log-in. All Web applications which use only HTTP cookies to store session information are vulnerable to CSRF.

Such applications expose their users to a typical CSRF attack every time they log in to the application. For example, a banking application which is vulnerable to CSRF may allow an attacker to transfer funds from a victim's account to his own account. If the victim is using an e-mail application vulnerable to CSRF, the attacker can send malicious e-mails using the victim's account without her knowledge.

Cross Site Request Forgery [PCI 6.5.9] Pass Findings

Assessment: QA_MPT_Assessment **Run:** 10/17/2017 11:05:31AM **Traversal:** MPT_Trav

Cross Site Request Forgery [PCI 6.5.9]
1 Pass (High, HARM: 0):
CVSS:
Message: `No Cross Site Request Forgery vulnerability found in an application`

Detail (by HARM / SmartAttack)

Attack: Cross-Frame Scripting	Version: 1.3.0	CWE-293	Observations: 1
Description: If a page is allowed to be embedded inside an HTML frame by another page, this may be used by a phishing site to replicate the look and feel of this page. Older browsers even allowed script from the outer page to access content from the page inside the frame. Hence, a page allowing it to be embedded in a frame is a vulnerability, which we call Cross-Frame Scripting. This SmartAttack identifies pages which can be embedded in frames under the control of an attacker.			
Severity: Medium HARM: 120 Total HARM: 120	Online Documentation for: Cross-Frame Scripting		

Impact An attacker may be able to steal sensitive information or make a victim perform certain actions without his knowledge through attacks such as Phishing or Clickjacking. In some cases, the attacker may also be able to take control of the user's machine.

Web pages that can be embedded in frames allow an attacker to perform such attacks. In older browsers, an attacker may be able to capture text written inside input boxes or from the page content of an embedded page.

Cross-frame scripting is a type of a phishing attack that involves instructions to an unsuspecting user to follow a specific link to update confidential information in an online application. The attackers can capture all the information that the user enters because the link leads to a legitimate page from the online application that is embedded in a frame hosted by the attackers' server.

Cross-Frame Scripting Vulnerable Findings

Assessment: QA_MPT_Assessment **Run:** 10/17/2017 11:05:31AM **Traversal:** MPT_Trav

Cross-Frame Scripting
1 Vulnerable (Medium, HARM: 120): https://www.qa-mypremiumtitle.com/SellerNetSheet.html

CVSS:	`4.3`
Message:	`Page found vulnerable to XFS`

Detail (by HARM / SmartAttack)

Attack: Cross-Site Scripting	Version: 3.4.6	Observations: 1
Description: Cross-site Scripting vulnerabilities allow malicious scripts to execute in the context of a trusted session with a web site. The SmartAttack alters the inputs to the web application to send benign versions of such malicious scripts, and detects the actual execution or unfiltered reflection of such scripts.		
Severity: High HARM: 320 Total HARM: 0		Online Documentation for: Cross-Site Scripting

Impact Cross-Site Scripting enables an attacker to run scripts inside a victim's browser. Using such a script, the attacker can modify the look-and-feel of a page, deface page contents and even steal user credentials and session information. If an application uses cookies for session management, then a Cross-Site Scripting vulnerability also assists the attacker in exploiting certain session-based attacks such as Session Fixation, if present.

Many Web applications display user input on their Web pages. Depending on whether the input is stored by the application for repeated use (*e.g.* user comments), a Cross-Site Scripting vulnerability may be *reflected* - *i.e.* usually one time - or *persistent*. A persistent Cross-Site Scripting vulnerability has greater impact than a reflected Cross-Site Scripting vulnerability, because a large number of users are affected without elaborate actions on the victims' part.

The effects of a Cross-Site Scripting vulnerability may range from simple defacement of Web pages to serious identity theft. For example, a Cross-Site Scripting vulnerability on a page that displays user-uploaded images could enable an attacker to show offensive images as if they were uploaded by a legitimate user, while a Cross-Site Scripting vulnerability on a banking Web site may expose the credentials of customers of the bank. While the offensive image may only affect a handful of users and the effect would be more annoyance than real harm, exposure of the credentials poses the threat of the attacker stealing money from them.

Cross-Site Scripting Pass Findings

Assessment: QA_MPT_Assessment **Run:** 10/17/2017 11:05:31AM **Traversal:** MPT_Trav

Cross-Site Scripting

1 Pass (High, HARM: 0):

CVSS:

Message:
```
No vulnerabilities were found for: Cross Site Scripting IG
```

Detail (by HARM / SmartAttack)

Attack: Cross-Site Scripting Through Flash	Version: 1.0.4	CWE-79	Observations: 1
Description: Cross-Site Scripting Through Flash allows malicious scripts to run in the context of a trusted session of the Web application. This SmartAttack alters the FlashVars of Flash content with benign version of such malicious scripts and detects actual execution of those scripts.			
Severity: High HARM: 280 Total HARM: 0	Online Documentation for: Cross-Site Scripting Through Flash		

Impact Cross-Site Scripting Through Flash enables an attacker to run scripts inside a victim’s browser. Using such a script, the attacker can modify the look-and-feel of a page, deface page contents and even steal user credentials and session information. If an application uses cookies for session management, then a Cross-Site Scripting Through Flash vulnerability also assists the attacker in exploiting certain session-based attacks such as Session Fixation, if present. Attacker can also steal session information stored in the cookies.

Like traditional Cross-Site Scripting, a Cross-Site Scripting Through Flash vulnerability may range from simple defacement of Web pages to serious identity theft. For example, a Cross-Site Scripting Through Flash vulnerability on a banking Web site may expose the credentials of customers of the bank. Exposure of the credentials poses the threat of the attacker stealing money from them.

Cross-Site Scripting Through Flash Pass Findings

Assessment: QA_MPT_Assessment **Run:** 10/17/2017 11:05:31AM **Traversal:** MPT_Trav

Cross-Site Scripting Through Flash
1 Pass (High, HARM: 0):
CVSS:
Message: `No Cross-Site Scripting Through Flash Vulnerabilities were found`

Detail (by HARM / SmartAttack)

Attack: Directory Browsing	Version: 1.4.0	CWE-548	Observations: 1
Description: Directory Browsing is a vulnerability caused by unintentionally disclosing directory listings to users. The SmartAttack attempts to retrieve and identify such listings and reports them as vulnerabilities based on the assumption that the listings are unintended.			
Severity: Medium HARM: 160 Total HARM: 0	Online Documentation for: Directory Browsing		

Impact If a Web application is vulnerable to directory browsing, an attacker can gain information about the web application by browsing directory listings that reveal files and folder hierarchy in the application. These resources may store sensitive information about web applications and operational systems, such as source code, credentials, internal network addressing, and so on which can be used to exploit vulnerabilities in the web application.

Information leakage occurs when a web site reveals sensitive data, such as authentication information, absolute or relative paths, which may aid an attacker in exploiting the system. While leakage through such directory listings does not necessarily represent a breach in security, it does give an attacker useful guidance for future exploitation. Leakage of sensitive information may carry various levels of risk and should be limited whenever possible.

For example, any attacker can guess file and directory names not intended for public viewing. The files/paths often have common naming convention and reside in standard locations. Hence by making educated guesses an attacker can attain absolute path.

Directory Browsing Pass Findings

Assessment: QA_MPT_Assessment **Run:** 10/17/2017 11:05:31AM **Traversal:** MPT_Trav

Directory Browsing
1 Pass (Medium, HARM: 0):
CVSS:
Message: `No directory listings found.`

Detail (by HARM / SmartAttack)

Attack: DOM Based Cross-Site Scripting	Version: 1.2.3	CWE-79	Observations: 1
Description: DOM based Cross-Site Scripting is an Cross-Site Scripting (XSS) variant where the malicious script is executed as a result of modifying DOM in a victim's browser. The malicious injected script is run by client side code, modifying the DOM environment. This attack is a client side code flaw while reflected and persistent Cross-Site Scripting are server side flaws.			
Severity: High HARM: 320 Total HARM: 0		Online Documentation for: DOM Based Cross-Site Scripting	

Impact Cross-Site Scripting enables an attacker to run scripts inside a victim's browser. Using such a script, the attacker can modify the look-and-feel of a page, deface page contents and even steal user credentials and session information. If an application uses cookies for session management, then a Cross-Site Scripting vulnerability also assists the attacker in exploiting certain session-based attacks such as Session Fixation, if present.

Many Web applications display user input on their Web pages. Depending on whether the input is stored by the application for repeated use (e.g. user comments), a Cross-Site Scripting vulnerability may be reflected - i.e. usually one time - or persistent. A persistent Cross-Site Scripting vulnerability has greater impact than a reflected Cross-Site Scripting vulnerability, because a large number of users are affected without elaborate actions on the victims' part. DOM based Cross-Site Scripting is special form of Cross-Site Scripting aseven if Web Application has server side defenses against XSS; the application can still be vulnerable if client side code is not validating input from certain places like URL.

The effects of a Cross-Site Scripting vulnerability may range from simple defacement of Web pages to serious identity theft. For example, a Cross-Site Scripting vulnerability on a page that displays user-uploaded images could enable an attacker to show offensive images as if they were uploaded by a legitimate user, while a Cross-Site Scripting vulnerability on a banking Web site may expose the credentials of customers of the bank. While the offensive image may only affect a handful of users and the effect would be more annoyance than real harm, exposure of the credentials poses the threat of the attacker stealing money from them.

DOM Based Cross-Site Scripting Pass Findings

Assessment: QA_MPT_Assessment **Run:** 10/17/2017 11:05:31AM **Traversal:** MPT_Trav

DOM Based Cross-Site Scripting

1 Pass (High, HARM: 0):

CVSS:

Message: `No vulnerabilities were found for: DOM based Cross-Site Scripting`

Detail (by HARM / SmartAttack)

Attack: Form Caching	Version: 1.1.35	CWE-525	Observations: 1
Description: Form Caching is a vulnerability caused by allowing the browser to cache sensitive form field values which could later be displayed to a different person using the same browser. The SmartAttack observes caching directives specified for pages, and reports each page that contains one or more forms while allowing such caching. The significance of such findings is dependent on the sensitivity of the cached data.			
Severity: Medium HARM: 640 Total HARM: 0		Online Documentation for: Form Caching	

Impact The impact of this vulnerability is entirely dependent on the sensitivtity of the data involved. Even common data associated with a person, such as address, email, and phone should commonly be considered sensitive.

Accessing a Web application having a Form Caching vulnerability will cause content in HTML forms to be stored on the machine from which the browsing happens. An attacker who has access to such a machine may be able to make the browser give away those cached forms by visiting its cache. If the previously filled form entries contain sensitive information like Credit Card numbers, usernames *etc*. then the attacker can easily steal and misuse such information.

An attacker who has access to a number of shared machines, such as at an Internet Cafe or by posing as a computer maintenance professional, he may be able to collect sensitive data of users of applications with this vulnerability. It may also be possible for an attacker to install malware on victims' computers which will compromise the browser, thus making the attacker's presence at the machine unnecessary.

Form Caching Pass Findings

Assessment: QA_MPT_Assessment **Run:** 10/17/2017 11:05:31AM **Traversal:** MPT_Trav

Form Caching
1 Pass (Medium, HARM: 0):
CVSS:
Message: `No forms were found that allow caching of their contents`

Detail (by HARM / SmartAttack)

Attack: Format String	Version: 1.4.11	CWE-134	Observations: 1
Description: This SmartAttack reports each page on which it finds format string vulnerabilities.			
Severity: High HARM: 640 Total HARM: 0		Online Documentation for: Format String	

Impact An attacker can crash the target application or the computer that is hosting it, potentially gaining administrative access to the host computer and all data and applications to which it has access.

Format String Pass Findings

Assessment: QA_MPT_Assessment **Run:** 10/17/2017 11:05:31AM **Traversal:** MPT_Trav

Format String
1 Pass (High, HARM: 0):
CVSS:
Message: `No vulnerabilities were found for: format string`

Detail (by HARM / SmartAttack)

Attack: Forms Submitted Without Using Post	Version: 1.2.1	Observations: 1
Description: Forms submitted without using POST is a vulnerability where a form with sensitive information is submitted via an HTTP GET request. This type of submission can result in disclosure of the submitted values. The SmartAttack reports all form submissions made using a GET request. The significance of such findings is dependent on the sensitivity of the submitted data.		
Severity: Medium HARM: 160 Total HARM: 0		Online Documentation for: Forms Submitted Without Using Post

Impact The impact of this vulnerability is entirely dependent on the sensitivity of the data involved. Even common data associated with a person, such as address, email, phone should commonly be considered sensitive.

A Web application that does not use POST for form submissions can leak this potentially private information. This submitted information will be retained in the browser history of the Web browser for later users of the same machine to access. It will also be collected in Web server logs, which are commonly not the most secure storage. And, for non-SSL communications, this data can be kept in logs of intermediary communication devices.

An attacker with access to a user’s browser, Web server logs, or communication infrastructure between the user and the website can easily collect such information. For example, an attacker who has access to machines and infrastructure at an internet cafe may be able to collect sensitive data of users of applications with this vulnerability, even when such attackers do not have the access needed to install malicious software.

The HTTP standard specifies that forms submitted that change the state of the server, such as performing some registration or transaction, should use POST.

Forms Submitted Without Using Post Pass Findings

Assessment: QA_MPT_Assessment **Run:** 10/17/2017 11:05:31AM **Traversal:** MPT_Trav

Forms Submitted Without Using Post

1 Pass (Medium, HARM: 0):

CVSS:

Message: `No forms were submitted without using the POST method`

Detail (by HARM / SmartAttack)

Attack: Frame Injection	Version: 1.0.13	CWE-20	Observations: 1
Description: This SmartAttack reports each page on which it finds Frame Injection vulnerabilities.			
Severity: Medium HARM: 100 Total HARM: 0		Online Documentation for: Frame Injection	

Impact The attacker can inject an arbitrary URL in a field that specifies the source of a frame or iframe, effectively enabling phishing attacks.

Frame Injection Pass Findings

Assessment: QA_MPT_Assessment **Run:** 10/17/2017 11:05:31AM **Traversal:** MPT_Trav

Frame Injection
1 Pass (Medium, HARM: 0):
CVSS:
Message: `No vulnerabilities were found for: Frame Injection`

Detail (by HARM / SmartAttack)

Attack: GET for POST	Version: 1.1.6	CWE-20	Observations: 4
Description: This Smart Attack identifies vulnerabilities that allow users to submit information of POST method by GET method.			
Severity: Low HARM: 50 Total HARM: 200		Online Documentation for: GET for POST	

Impact An attacker can gain access to same page by GET method which is expected to be accessed only by POST method.

GET for POST Vulnerable Findings

Assessment: QA_MPT_Assessment **Run:** 10/17/2017 11:05:31AM **Traversal:** MPT_Trav

GET for POST
1 Vulnerable (Low, HARM: 50): https://www.qa-mypremiumtitle.com/Create_an_account_with_Altisource_today.aspx
CVSS: 5.0
Message:
```
GET for POST vulnerability detected:

Comparison of pages:
  Original Response Status: 200
  Injected Response Status: 200
```

GET for POST
2 Vulnerable (Low, HARM: 50): https://www.qa-mypremiumtitle.com/ibbLogin.aspx
CVSS: 5.0
Message:
```
GET for POST vulnerability detected:

Comparison of pages:
  Original Response Status: 200
  Injected Response Status: 200
```

GET for POST
3 Vulnerable (Low, HARM: 50): https://www.qa-mypremiumtitle.com/ibbLogin.aspx
CVSS: 5.0
Message:
```
GET for POST vulnerability detected:

Comparison of pages:
  Original Response Status: 200
  Injected Response Status: 200
```

GET for POST
4 Vulnerable (Low, HARM: 50): https://www.qa-mypremiumtitle.com/YourPrivacyRights.aspx
CVSS: 5.0
Message:
```
GET for POST vulnerability detected:

Comparison of pages:
  Original Response Status: 200
  Injected Response Status: 200
```

Detail (by HARM / SmartAttack)

Attack: HTTP Parameter Pollution	Version: 1.2	CWE-235	Observations: 1
Description: This SmartAttack reports each page on which it finds HTTP Parameter Pollution vulnerabilities.			
Severity: Low HARM: 30 Total HARM: 0		Online Documentation for: HTTP Parameter Pollution	

Impact Multiple occurrences of same parameter are treated differently in various platforms. If such occurrences are not validated properly, the attacker can craft various attacks by using the same parameter multiple times and thus forcing the Web application to use the evil input. HTTP Parameter Pollution typically involves exploiting this loophole on the Web application.

HTTP Parameter Pollution Pass Findings

Assessment: QA_MPT_Assessment **Run:** 10/17/2017 11:05:31AM **Traversal:** MPT_Trav

HTTP Parameter Pollution
1 Pass (Low, HARM: 0):
CVSS:
Message: `No vulnerabilities were found for: HTTP Parameter Pollution`

Detail (by HARM / SmartAttack)

Attack: HTTP Response Splitting	Version: 1.4.18	CWE-113	Observations: 1
Description: HTTP Response Splitting is a vulnerability allowing an attacker to structure a request that results in two responses, the second of which is totally under the control of the attacker. The SmartAttack sends benign requests designed to fool the server into returning a response setting a custom Cenzic Cookie where normally it would not and reports vulnerabilities if successful.			
Severity: High HARM: 300 Total HARM: 0		Online Documentation for: HTTP Response Splitting	

Impact An attacker can inject arbitrary HTTP response information that appears to a browser as legitimate content originating from the server. As a result, it is possible to perform a range of attacks, such as site content spoofing, web cache poisoning, or other types of attacks.

HTTP Response Splitting vulnerability helps the attacker to inject malicious or unexpected characters in user input which is then used for a 302 Redirect, in the Location or Set-Cookie header. The attacker can structure the response any way he wants. For example if a user is using a banking application vulnerable to this attack then the user can be tricked in sending his details to the attacker without his knowledge.

HTTP Response Splitting Pass Findings

Assessment: QA_MPT_Assessment **Run:** 10/17/2017 11:05:31AM **Traversal:** MPT_Trav

HTTP Response Splitting
1 Pass (High, HARM: 0):
CVSS:
Message: `No vulnerabilities were found for: HTTP Response Splitting`

Detail (by HARM / SmartAttack)

Attack: Integer Overflow	Version: 1.0.13	CWE-190	Observations: 1
Description: This SmartAttack reports each page on which it finds integer overflow vulnerabilities.			
Severity: Medium HARM: 384 Total HARM: 0		Online Documentation for: Integer Overflow	

Impact An attacker can crash the target application or the computer that is hosting it, potentially gaining administrative access to the host computer and all data and applications to which it has access.

Integer Overflow Pass Findings

Assessment: QA_MPT_Assessment **Run:** 10/17/2017 11:05:31AM **Traversal:** MPT_Trav

Integer Overflow
1 Pass (Medium, HARM: 0):
CVSS:
Message: `No vulnerabilities were found for: Integer overflow`

Detail (by HARM / SmartAttack)

Attack: LDAP Exception	Version: 1.0.5	CWE-90	Observations: 1
Description: This SmartAttack reports each page on which it finds LDAP exception vulnerabilities, that is, where bad input leads to an internal LDAP processing error.			
Severity: Medium HARM: 120 Total HARM: 0	Online Documentation for: LDAP Exception		

Impact Error messages often give an attacker useful information about how an application interacts with back-end components, and can reveal potential vulnerabilities within the application itself. It is common for intruders to manipulate application input parameters in order to elicit application exceptions. An attacker can reverse engineer the structure of LDAP query structure with the help of these application exception messages. Once the query structure is determined, the attacker can generate more attacks to access sensitive information by injecting valid search filters.

LDAP Exception Pass Findings

Assessment: QA_MPT_Assessment **Run:** 10/17/2017 11:05:31AM **Traversal:** MPT_Trav

LDAP Exception

1 Pass (Medium, HARM: 0):

CVSS:

Message: `No vulnerabilities were found for: LDAP Error Message`

Detail (by HARM / SmartAttack)

Attack: LDAP Injection	Version: 1.2.5	CWE-90	Observations: 2
Description: This SmartAttack reports each page on which it finds LDAP Injection vulnerabilities.			
Severity: High HARM: 230 Total HARM: 0		Online Documentation for: LDAP Injection	

Impact An attacker can access information that should normally be inaccessible. The attacker can systematically find out LDAP query structure by injecting specially crafted LDAP search filter characters. Once the query structure is determined, the attacker can generate more attacks to access sensitive information by injecting valid search filters.

Similarly the LDAP directory access control flaw can be exploited by manipulating directory tree's user object attributes to gain unauthorized access to proprietary data.

LDAP Injection Information Findings

Assessment: QA_MPT_Assessment **Run:** 10/17/2017 11:05:31AM **Traversal:** MPT_Trav

LDAP Injection
1 Information (High, HARM: 0): https://www.qa-mypremiumtitle.com/Requester.aspx
CVSS:
Message:
```
Server response status: 500
Internal server error
Injectable request #: 4
Injected item: POST: viewpdfdata
True Injection value :undefined
False Injection value :undefined
```

LDAP Injection
2 Pass (High, HARM: 0):
CVSS:
Message:
```
No vulnerabilities were found for: LDAP Injection
```

Detail (by HARM / SmartAttack)

Attack: Login Redirect - [OWASP 2013 A 2]	Version: 1.2	CWE-525	Observations: 1
Description: This SmartAttack reports vulnerability if 302 redirect is not found on login page.			
Severity: Medium HARM: 160 Total HARM: 0		Online Documentation for: Login Redirect - [OWASP 2013 A 2]	

Impact Successful exploitation of this vulnerability will give an unauthorized access of the victim's credentials to the attacker.

Login Redirect - [OWASP 2013 A 2] Notify Findings

Assessment: QA_MPT_Assessment **Run:** 10/17/2017 11:05:31AM **Traversal:** MPT_Trav

Login Redirect - [OWASP 2013 A 2]

1 Notify (Medium, HARM: 0):

CVSS:

Message:
```
No logged-in status found. There should be atleast one login request with valid login
credentials.
```

Detail (by HARM / SmartAttack)

Attack: Non-SSL Form	Version: 1.2.20	CWE-201	Observations: 1
Description: Non-SSL Form is a vulnerability caused by allowing submission of sensitive form data without using SSL encryption. The SmartAttack observes and reports any form that is not submitted via SSL. The significance of such findings is dependent on the sensitivity of the submitted data.			
Severity: High HARM: 288 Total HARM: 0		Online Documentation for: Non-SSL Form	

Impact The impact of this vulnerability is entirely dependent on the sensitivity of the data involved. Even common data associated with a person, such as address, email, phone should commonly be considered sensitive.

Accessing a Web application having a Non-SSL Form vulnerability can leak this potentially private information. An attacker with access to communication infrastructure between the user and the website can easily and automatically collect such information. Here, which information is considered as private is entirely dependent on the context set by the Web application.

For example, an attacker who has access to a shared switch or router at an Internet Cafe may be able to collect sensitive data of users of applications with this vulnerability.

Non-SSL Form Pass Findings

Assessment: QA_MPT_Assessment **Run:** 10/17/2017 11:05:31AM **Traversal:** MPT_Trav

Non-SSL Form
1 Pass (High, HARM: 0):
CVSS:
Message: `All forms were submitted using SSL`

Detail (by HARM / SmartAttack)

Attack: Non-SSL Password	Version: 1.3.4	CWE-201	Observations: 1
Description: Non-SSL Password is a vulnerability caused by a failure to submit passwords via SSL. The SmartAttack inspects all requests where passwords are submitted and reports those where SSL is not used for the submission. It also reports those where the forms containing password pages are placed on pages not served via SSL, according to OWASP recommendations.			
Severity: High HARM: 640 Total HARM: 0		Online Documentation for: Non-SSL Password	

Impact An attacker sniffing traffic on a computer network may notice and consequently steal usernames and passwords if they are sent unencrypted to a Web application. The attacker may then use this information to use the accounts corresponding to it for various malicious purposes, ranging from disclosure of personal information to identity theft.

Moreover, if an attacker manages to run a non-transparent HTTP proxy (*i.e.* a proxy that can modify HTTP requests and responses) running between victims' browsers and the Internet, then he may even be able to modify actions of forms submitting passwords to re-route all login requests through a server under his control. In this case, no other packet sniffing is needed as the login form submissions will automatically send the passwords to him.

Although the use of a packet sniffer, network monitoring tool or a proxy is required to capture unencrypted traffic on a network, such tools are widely available and may be easy to install. This means that the complexity involved in capturing unencrypted traffic can be very less. Depending on which place the attacker is successfully able to install a sniffer, the scope of such an attack may be large, thereby increasing the chance of the attacker succeeding to get more passwords.

Non-SSL Password Pass Findings

Assessment: QA_MPT_Assessment **Run:** 10/17/2017 11:05:31AM **Traversal:** MPT_Trav

Non-SSL Password
1 Pass (High, HARM: 0):
CVSS:
Message: `All forms containing password fields have actions that appear to use SSL`

Detail (by HARM / SmartAttack)

Attack: Open Redirect	Version: 1.0.26	CWE-601	Observations: 1
Description: This SmartAttack reports each page on which it finds Open Redirect vulnerabilities.			
Severity: Medium HARM: 100 Total HARM: 0		Online Documentation for: Open Redirect	

Impact The attacker can inject arbitrary URL in the field that causes redirection, and can effectively cause phishing attacks.

Open Redirect Pass Findings

Assessment: QA_MPT_Assessment **Run:** 10/17/2017 11:05:31AM **Traversal:** MPT_Trav

Open Redirect

1 Pass (Medium, HARM: 0):

CVSS:

Message: `No vulnerabilities were found for: Open Redirect`

Detail (by HARM / SmartAttack)

Attack: Parameter Addition	Version: 1.0.9	CWE-472	Observations: 1
Description:	This SmartAttack reports each page on which it finds Parameter Addition vulnerabilities.		
Severity: Medium HARM: 128 Total HARM: 0		Online Documentation for: Parameter Addition	

Impact An attacker can access information that should normally be inaccessible. The attacker can systematically find out special parameters that can be added to URL or POST body or Cookie of request to gain unauthorized administrative access.

Parameter Addition Pass Findings

Assessment: QA_MPT_Assessment **Run:** 10/17/2017 11:05:31AM **Traversal:** MPT_Trav

Parameter Addition

1 Pass (Medium, HARM: 0):

CVSS:

Message: `No vulnerabilities were found for: Parameter Addition`

Detail (by HARM / SmartAttack)

Attack: Password Autocomplete	Version: 1.1.14	CWE-525	Observations: 1
Description: Password Autocomplete is a vulnerability caused by allowing caching of passwords by browsers. The SmartAttack inspects responses from the application to identify if autocomplete is explicitly set to “off”.			
Severity: Medium HARM: 210 Total HARM: 0		Online Documentation for: Password Autocomplete	

Impact Accessing a Web application with the Password Autocomplete vulnerability will cause users’ passwords to be stored on the machine from where the browsing happens. An attacker who has access to such a machine may be able to make the browser give away those passwords by visiting the pages with the login forms, or by means of compromising the browser. He may be able to use those passwords for identity theft and/or performing malicious actions on behalf of the users.

An attacker who has access to a number of shared machines, such as at an Internet Cafe or posing as a computer maintenance professional may be able to collect passwords of users of applications with this vulnerability. It may also be possible for an attacker to install malware on victims’ computers which will compromise the browser, thus making the attacker’s presence at the machine unnecessary.

Password Autocomplete Pass Findings

Assessment: QA_MPT_Assessment **Run:** 10/17/2017 11:05:31AM **Traversal:** MPT_Trav

Password Autocomplete

1 Pass (Medium, HARM: 0):

CVSS:

Message: `No password fields found that are autocompleted`

Detail (by HARM / SmartAttack)

Attack: PHP & Perl Code Injection	Version: 1.0.12	CWE-94	Observations: 1
Description: This SmartAttack reports each field on which it finds PHP/Perl Code-Injection vulnerabilities.			
Severity: Medium HARM: 100 Total HARM: 0		Online Documentation for: PHP & Perl Code Injection	

Impact The user is able to execute arbitrary commands on the server if he is able to inject perl/php code and manage to get it executed.

PHP & Perl Code Injection Pass Findings

Assessment: QA_MPT_Assessment **Run:** 10/17/2017 11:05:31AM **Traversal:** MPT_Trav

PHP & Perl Code Injection
1 Pass (Medium, HARM: 0):
CVSS:
Message: `No vulnerabilities were found for: PHP/Perl code injection`

Detail (by HARM / SmartAttack)

Attack: Platform Path Disclosure - [OWASP 2013 A 5]	Version: 1.2	CWE-209	Observations: 1
Description:	This SmartAttack reports each page on which it finds path disclosure vulnerabilities, that is, where it finds path information of any components in the original response or in the response for a modified URL. This SmartAttack examines each page and makes additional exploits to cause the platform (for example, ASP.NET, PHP) to reveal path disclosures.		
Severity: Medium HARM: 100 Total HARM: 0		Online Documentation for: Platform Path Disclosure - [OWASP 2013 A 5]	

Impact Error messages often give an attacker useful information about how an application interacts with back-end components, and can reveal potential vulnerabilities within the application itself. Often, application exceptions give out information about the web server directory structure, paths of application components, or clues about types of programming or scripting languages used within the application environment.

Platform Path Disclosure - [OWASP 2013 A 5] Pass Findings

Assessment: QA_MPT_Assessment **Run:** 10/17/2017 11:05:31AM **Traversal:** MPT_Trav

Platform Path Disclosure - [OWASP 2013 A 5]
1 Pass (Medium, HARM: 0):
CVSS:
Message: `Pages do not disclose any path information`

Detail (by HARM / SmartAttack)

Attack: Private IP Disclosure	Version: 1.5	CWE-200	Observations: 1
Description: Disclosing the internal IP address of a server enables an attacker to launch multiple advanced attacks. Hence, we call such a vulnerability as Private IP Disclosure. This SmartAttack looks for private IP addresses in the response of traversed pages.			
Severity: Low HARM: 16 Total HARM: 0		Online Documentation for: Private IP Disclosure	

Impact An attacker can use private IP addresses to perform various attacks on the hosts with those IP addresses, provided he can bypass the gateway. Attacker can know the types of the servers present on those hosts and perform various attacks depending on versions of the servers.

If an application discloses private IP addresses, then it can help attacker in planning advanced attacks based on the information disclosed through such a vulnerability. If he can get access to the internal network where the Web application is hosted, then such a disclosure may help him, for example, in planning an internal Denial of Service attack using the hosts.

Many Web applications are hosted in an environment where other machines exist on an internal network. If an attacker can access the network, details of these machines can help an attacker quickly capitalize such access. For example, if an attacker comes to know the IP address of the e- mail server on the internal network of a Banking Application, he may be able to send spurious e- mails authorizing malicious transactions.

Private IP Disclosure Pass Findings

Assessment: QA_MPT_Assessment **Run:** 10/17/2017 11:05:31AM **Traversal:** MPT_Trav

Private IP Disclosure
1 Pass (Low, HARM: 0):
CVSS:
Message: `No traces of private IP address found in the response of the traversal.`

Detail (by HARM / SmartAttack)

Attack: Redirection Through Flash	Version: 1.0.4	CWE-601	Observations: 1

Description: Redirection Through Flash is a vulnerability caused by embedded Flash content trying to load a Web page on another domain based on user's input without any validation. This SmartAttack injects a URL in the FlashVars of the Flash content and tries to detect whether it loads the injected URL without any validation.

Severity: Medium HARM: 100 Total HARM: 0 — Online Documentation for: Redirection Through Flash

Impact An attacker may be able to steal sensitive information from users without their knowledge using phishing attacks, which use pages looking similar to legitimate Web applications to trick them into giving away such information. Less technology savvy users can not readily distinguish such pages and easily fall victim to such scams

Flash applications which are vulnerable to Redirection Through Flash allow an attacker to perform such phishing attacks. An attacker can easily create URLs or pages which exploit this vulnerability by redirecting users to any Web page of the attacker's choice. This could either be a phishing site or an application designed to perform other attacks such as Cross-Site Scripting.

Many Flash applications use FlashVars to alter the behavior of the Flash files, or *movies*, used by them which could lead to unauthorized and unexpected redirects. For example, a vulnerable Flash advertisement for a new feature in a popular Web application may enable the attacker to redirect victims to a phished login page, where they may inadvertently enter credentials for that Web application.

Redirection Through Flash Pass Findings

Assessment: QA_MPT_Assessment **Run:** 10/17/2017 11:05:31AM **Traversal:** MPT_Trav

Redirection Through Flash
1 Pass (Medium, HARM: 0):
CVSS:
Message: `No Redirection Through Flash Vulnerabilities were found`

Detail (by HARM / SmartAttack)

Attack: Remote File Inclusion	Version: 1.0.13	CWE-98	Observations: 1
Description: Remote File Inclusion is a vulnerability where a submitted value is used directly, without sanitization, to reference any specified URL of a file for processing. The SmartAttack submits its own URL to a Cenzic hosted file, as parameter values, and reports each occurrence where it can detect evidence of the Cenzic file being processed.			
Severity: High HARM: 480 Total HARM: 0		Online Documentation for: Remote File Inclusion	

Impact Since code written in a remote file is included directly into the script and executed as server side code, execution of arbitrary commands/code in context of the web server is possible.

Remote File Inclusion Pass Findings

Assessment: QA_MPT_Assessment **Run:** 10/17/2017 11:05:31AM **Traversal:** MPT_Trav

Remote File Inclusion

1 Pass (High, HARM: 0):

CVSS:

Message: `No vulnerabilities were found for: Remote File Inclusion`

Detail (by HARM / SmartAttack)

Attack: SQL Disclosure	Version: 1.4.32	CWE-566	Observations: 2
Description: SQL Disclosure is a vulnerability caused by a Web application sending user input into a SQL query without validation. It is a type of SQL Injection vulnerability, where an attacker is able to extract much more information than he is authorized for, simply by using SQL statements that always evaluate to true in the input. This SmartAttack injects such statements and looks for evidence of extra information getting revealed in the response.			
Severity: High HARM: 768 Total HARM: 0	Online Documentation for: SQL Disclosure		

Impact A SQL Disclosure vulnerability enables an attacker to gain access to information that he does not have the privileges for. This includes data stored in the tables and probably credentials for accessing sections of an application requiring higher privileges.

Many Web applications store important information in SQL databases, such as financial transactions of users, confidential company records and credit card numbers. Querying such databases without proper access restrictions can cause leakage of this data to anyone who can send queries to them through a Web application. For example, a well-crafted injection on a banking application may reveal transactions done by all users to one malicious user. Similarly, an injection on an application used for taxation by companies may expose tax details of many companies to an attacker.

SQL Disclosure Information Findings

Assessment: QA_MPT_Assessment **Run:** 10/17/2017 11:05:31AM **Traversal:** MPT_Trav

SQL Disclosure
1 Information (High, HARM: 0): https://www.qa-mypremiumtitle.com/Requester.aspx
CVSS:
Message:
```
Server response status: 500
Internal server error
Injectable request #: 4
Injected item: POST: viewpdfdata
Injection value: -1 union select
99999.99/3,99999.99/3,99999.99/3,99999.99/3,99999.99/3,99999.99/3,99999.99/3 -- -
```

SQL Disclosure
2 Pass (High, HARM: 0):
CVSS:
Message:
```
No vulnerabilities were found for: SQL disclosure
```

Detail (by HARM / SmartAttack)

Attack: SQL Error Message	Version: 1.6.5	CWE-209	Observations: 1
Description: SQL Error Message, or SQL Exception, is a vulnerability caused by a Web application inserting user input in a SQL query without validation and failing to suppress error messages that may result from use of such input. This SmartAttack injects SQL characters in order to cause errors in SQL execution, and looks for evidence of such errors.			
Severity: Medium HARM: 120 Total HARM: 0		Online Documentation for: SQL Error Message	

Impact An attacker might gain administrative control of your web application or database by using specially crafted SQL queries. It is also be possible to gain remote access to restricted information via queries to your database.

A SQL Error Message vulnerability helps the attacker in formulating the correct query depending on information disclosed in the error message. Such an error message may also disclose information about the deployment of the database, such as the database server used, server-side technology used, *etc*.

Many Web applications use SQL databases to store important information which can be disclosed through error messages deliberately obtained by an attacker. For example, an error generated by a Web-based form for searching employee details of an organization may lead to disclosure of personal information stored along with. Likewise, errors generated by a Web-based inventory management app may divulge important information about products yet to be released.

SQL Error Message Pass Findings

Assessment: QA_MPT_Assessment **Run:** 10/17/2017 11:05:31AM **Traversal:** MPT_Trav

SQL Error Message
1 Pass (Medium, HARM: 0):
CVSS:
Message: `No vulnerabilities were found for: SQL Error Message`

Detail (by HARM / SmartAttack)

Attack: SSI Injection	Version: 1.0.12	CWE-97	Observations: 1
Description: SSI – Server side includes. This SmartAttack reports each field on which it finds SSI Injection vulnerabilities.			
Severity: Medium HARM: 192 Total HARM: 0		Online Documentation for: SSI Injection	

Impact Some SSI tags allow the developer to execute arbitrary commands while serving the page. If user input is interpreted as SSI tag, user might be able to execute arbitrary commands on the server.

SSI Injection Pass Findings

Assessment: QA_MPT_Assessment **Run:** 10/17/2017 11:05:31AM **Traversal:** MPT_Trav

SSI Injection
1 Pass (Medium, HARM: 0):
CVSS:
Message: `No vulnerabilities were found for: SSI injection`

Detail (by HARM / SmartAttack)

Attack: SSL Checks	Version: 1.8.1	Observations: 3

Description: HTTPS Web servers supporting weak ciphers are vulnerable to a breach of secure communication. Such servers, if they are using expired certificates, expose users of applications hosted on them to Phishing attacks. This SmartAttack reports HTTPS Web servers which support weak ciphers. The SmartAttack also checks if SSLv2 is supported. The SmartAttack reports hosts for which the certificate has expired or is not issued to the domain of Web server.

Severity: Medium HARM: 100 Total HARM: 0 | Online Documentation for: SSL Checks

Impact If an HTTPS server uses known weak ciphers in its communication to a Web browser, an attacker may be able to breach the secure communication and know the contents. A ‘Man In The Middle’ (MITM) attack may happen as the attacker is able to alter the communication after breaking a weak cipher. If the SSL certificate of an HTTPS server is not valid, it may happen that the victim is not communicating with the authentic Web server and is unknowingly falling victim to a Phishing attack.

After breaking the weak cipher communication, attacker may get important information, like financial or sensitive data or even credentials to user account. This information can be used by him to harm the user or the Web application. If SSLv2 is being used then it is possible for an attacker to change the algorithm or the key lengths chosen by the client resulting in a weak SSL connection being set up between the client and the server. If a certificate has expired, it may be insecure to use and the communication may be performed with an entirely different server, claiming to be the owner of the certificate.

Consider an example where a banking Web application supports a 60 bit cipher. When the cipher is used for communication, an attacker can use various hacking tools to break the cipher. Once the communication is breached, attacker can know sensitive information about financial data of the victim.

SSL Checks Information Findings

Assessment: QA_MPT_Assessment **Run:** 10/17/2017 11:05:31AM **Traversal:** MPT_Trav

SSL Checks
1 Information (Medium, HARM: 0):
CVSS:
Message:

```
No Heartbleed vulnerability found.
```

SSL Checks
2 Information (Medium, HARM: 0): www.qa-mypremiumtitle.com:443
CVSS:
Message:

```
Summary of ciphers supported by server
  (Other than SSLv2 and SSLv3):

  Weak Strength Ciphers   : 0
  High Strength Ciphers     : 6


High strength ciphers supported:

  SSL_RSA_WITH_3DES_EDE_CBC_SHA      :168
  TLS_RSA_WITH_AES_128_CBC_SHA      :128
  TLS_RSA_WITH_AES_256_CBC_SHA      :256
  TLS_ECDHE_RSA_WITH_3DES_EDE_CBC_SHA      :168
  TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA      :128
  TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA      :256
```

SSL Checks

3 Pass (Medium, HARM: 0):

CVSS:

Message: `No SSL discrepancy found.`

Detail (by HARM / SmartAttack)

Attack: Unix Command Injection	Version: 1.6.4	CWE-78	Observations: 2
Description: This SmartAttack reports each page on which it finds UNIX command injection vulnerabilities.			
Severity: High HARM: 960 Total HARM: 0		Online Documentation for: Unix Command Injection	

Impact An attacker can run arbitrary command on the computer hosting the target application, potentially gaining administrative access to the host computer and all data and applications to which it has access.

Unix Command Injection Information Findings

Assessment: QA_MPT_Assessment **Run:** 10/17/2017 11:05:31AM **Traversal:** MPT_Trav

Unix Command Injection
1 Information (High, HARM: 0): https://www.qa-mypremiumtitle.com/Requester.aspx
CVSS:
Message:
```
Server response status: 500
Internal server error
Injectable request #: 4
Injected item: POST: viewpdfdata
Injection value: %0aid
```

Unix Command Injection
2 Pass (High, HARM: 0):
CVSS:
Message:
```
No vulnerabilities were found for: Unix command injection
```

Detail (by HARM / SmartAttack)

Attack: Unix Relative Path	Version: 1.6.2	CWE-22	Observations: 1
Description: This SmartAttack reports each page on which it finds UNIX relative path vulnerabilities.			
Severity: High HARM: 960 Total HARM: 0		Online Documentation for: Unix Relative Path	

Impact An attacker can cause the target application to traverse to the root of the host machine's file system and return sensitive information, such as a password file.

Unix Relative Path Pass Findings

Assessment: QA_MPT_Assessment **Run:** 10/17/2017 11:05:31AM **Traversal:** MPT_Trav

Unix Relative Path
1 Pass (High, HARM: 0):
CVSS:
Message: `No vulnerabilities were found for: Unix relative path`

Detail (by HARM / SmartAttack)

Attack: Unrestricted File Upload	Version: 1.0.6	CWE-434	Observations: 1
Description: File upload is the first step taken by the attacker to upload his malicious code on the server. Attacker can breach the security of the Web server by finding a way to execute the uploaded code on the application platform or at the victim's side. This SmartAttack tries to upload malicious files and reports vulnerabilities if Web application is not restricting the file upload.			
Severity: Medium HARM: 128 Total HARM: 0		Online Documentation for: Unrestricted File Upload	

Impact If this Vulnerability is present, an attacker might gain sensitive information of the server. Attacker might be able to put a phishing page on the server which may lead to phishing attacks. This vulnerability can make the website vulnerable to attacks such as Cross-Site Scripting. The malicious file can exploit the server vulnerabilities if executed by the administrator of the server.

An attacker can easily upload malicious file on the server if there are no restrictions while uploading the file. Attacker then looks for ways to get the file executed. The file can breach the security of server by getting executed on the server. Alternatively various Web vulnerabilities can be exploited at the client side.

For example: an attacker can upload an HTML file containing a script. If the server is sending the HTML file directly to the user, the malicious script can get executed at client side. Thus this Vulnerability may lead to Cross-Site Scripting attack. Similarly a malicious file or a virus can be uploaded by an attacker on the Web server which can infect Web server as well as that Web application's users.

Unrestricted File Upload Notify Findings

Assessment: QA_MPT_Assessment **Run:** 10/17/2017 11:05:31AM **Traversal:** MPT_Trav

Unrestricted File Upload
1 Notify (Medium, HARM: 0):
CVSS:
Message: `No file upload requests were seen.`

Detail (by HARM / SmartAttack)

Attack: Web Server Vulnerabilities	Version: 1.1.35	Observations: 1
Description: Web Server Vulnerabilities are a variety of CVE style vulnerabilities regarding known security flaws in known versions of various software infrastructures, such as Apache, PHP, Oracle, etc. This SmartAttack does hundreds of tests looking for versions and resources with evidence pointing to known CVEs.		
Severity: Low HARM: 30 Total HARM: 0	Online Documentation for: Web Server Vulnerabilities	

Impact An attacker may gain administrative control of your application or launch other advanced attacks such as Command Execution, Buffer Overflows if he has an accurate knowledge of the vulnerabilities in your Web and application servers. He may also be able to influence configuration settings of your servers if certain common configuration files are kept accessible.

Web Server Vulnerabilities, if present, help an attacker plan advanced attacks based on the information disclosed through such vulnerabilities. These vulnerabilities not only disclose the Web and application servers being used, but may also disclose their versions, server-side technologies used, configuration settings and sensitive information about the application and its users.

Many Web servers advertise themselves by adding headers to each HTTP response that they send out. Many servers also come with typical configurations for ease of use. These details are publicly available most of the times, making an attacker’s job easier. For example, if he comes to know the version number and configuration of a server used by a banking application, he may try to exploit any of the vulnerabilities reported in the public domain for that version of that server for manipulating accounts of the customers of the bank.

Web Server Vulnerabilities Pass Findings

Assessment: QA_MPT_Assessment **Run:** 10/17/2017 11:05:31AM **Traversal:** MPT_Trav

Web Server Vulnerabilities

1 Pass (Low, HARM: 0):

CVSS:

Message: `App Scanner database contains no known vulnerability for current server configuration.`

Detail (by HARM / SmartAttack)

Attack: Windows Command Injection	Version: 1.6.4	CWE-78	Observations: 2
Description: This SmartAttack reports each page on which it finds Windows command injection vulnerabilities.			
Severity: High HARM: 960 Total HARM: 0		Online Documentation for: Windows Command Injection	

Impact An attacker can run arbitrary command on the computer hosting the target application, potentially gaining administrative access to the host computer and all data and applications to which it has access.

Windows Command Injection Information Findings

Assessment: QA_MPT_Assessment **Run:** 10/17/2017 11:05:31AM **Traversal:** MPT_Trav

Windows Command Injection
1 Information (High, HARM: 0): https://www.qa-mypremiumtitle.com/Requester.aspx
CVSS:
Message:
```
Server response status: 500
Internal server error
Injectable request #: 4
Injected item: POST: viewpdfdata
Injection value: ||type \boot.ini%00
```

Windows Command Injection
2 Pass (High, HARM: 0):
CVSS:
Message:
```
No vulnerabilities were found for: Windows command injection
```

Detail (by HARM / SmartAttack)

Attack: Windows Relative Path	Version: 1.6.2	CWE-22	Observations: 2
Description: This SmartAttack reports each page on which it finds Windows relative path vulnerabilities.			
Severity: High HARM: 960 Total HARM: 0		Online Documentation for: Windows Relative Path	

Impact An attacker can cause the target application to traverse to the root of the host machine's file system and return sensitive information, such as a password file.

Windows Relative Path Pass Findings

Assessment: QA_MPT_Assessment **Run:** 10/17/2017 11:05:31AM **Traversal:** MPT_Trav

Windows Relative Path
1 Pass (High, HARM: 0):
CVSS:
Message:
```
No vulnerabilities were found for: Windows relative path
```

Windows Relative Path
2 Information (High, HARM: 0): https://www.qa-mypremiumtitle.com/Feedback.aspx
CVSS:
Message:
```
Server response status: 500
Internal server error
Injectable request #: 4
Injected item: POST: ctl00%24ctl00%24ContentPlaceHolder1%24ContentPlaceHolder1%24txtname
Injection value: ../../../../../../../../windows/system32/drivers/etc/hosts
```