Web Application Scanning

Testing Methodology

October 2018 | v2.6

Introduction

Qualys Web Application Scanning (WAS) is an automated, dynamic web application security testing tool. Its capabilities are provided in the form of software as a service (SaaS). WAS performs testing at the application layer and is designed to identify web application vulnerabilities and weaknesses that can potentially be exploited by remote attackers or other malicious users. Testing is performed from the same perspective as a user with a web browser. The tool provides recommendations on how to correct all identified issues so that the security posture of the web application can be improved and to reduce the likelihood of a successful application-layer attack.

More information is available at https://www.qualys.com/was.

Requirements

As a dynamic, black-box web application vulnerability scanning tool, Qualys WAS requires a running instance of the target web application.

If the application is internal (not exposed to the Internet), a Qualys scanner appliance is required to be deployed in the internal network so that scanning may take place. This can be in the form of a physical or virtual appliance.

Unauthenticated scans can be run simply by providing the target URL of the application. If the application includes authentication functionality and it is desired to scan the authenticated surface area, user credentials must be provided so that the scanner can authenticate during the scan to reach those areas of the application (see "Authenticated Scanning" below).

Testing Methodology

OVERVIEW

The Qualys WAS testing methodology is consistent with the de facto industry standard OWASP Top 10. However, the OWASP Top 10 is not as granular as the Qualys WAS detections. Each detection in WAS is assigned a unique QID (Qualys Identifier). Most but not all of the QIDs used by WAS are 6 digit numbers beginning with "150".

The majority of the WAS QIDs map to at least one of the OWASP Top 10 categories. Furthermore, where applicable, each QID maps to a specific Common Weakness Enumeration (CWE) ID and a category in the Web Application Security Consortium (WASC) Threat Classification system.

Note that WAS includes tests for vulnerabilities and weaknesses that are not explicitly covered by the OWASP Top 10, but nevertheless pose a risk to web applications.

The overall objective of the WAS testing methodology is to minimize false negatives and false positives and be non-intrusive. The tool's vulnerability detection capabilities focus on issues that can be reliably automated, identified accurately, and lead to actionable results.

The scanner does not perform denial-of-service (DoS) attacks of any kind. Nor does it perform SQL injection tests that attempt to insert data into a database. Login brute force testing is an option that can be enabled during a scan.

The scanning engine is typically updated 10 to 12 times per year, but its payloads are updated more frequently. This approach enables the scanner to respond and adapt quickly when new vulnerabilities emerge, current detections are refined, and feedback from customer scanning activity is received.

AUTHENTICATED SCANNING

WAS is designed to be as safe as possible, even in a production environment. However, it is not recommended to run authenticated scans against a production web application. The scanner will invoke any and all functionality it finds within an application. It would have access to the same functionality as the user whose credentials are given. For example, if the application allows a user to delete records from the database via normal application functionality, then the scanner may also delete records during a scan.

Performing authenticated scans with a high-privilege user account, such as an administrator, is especially risky and not recommended. Doing so could result in the application experiencing a loss of data and/or data integrity. This is because the scanner would have access to the same functionality as an administrator and would invoke any functionality it finds as part of its normal testing process.

TESTING PHASES

Testing with Qualys WAS consists of two main phases:

1. Discovery (crawling) phase
2. Vulnerability testing phase

Discovery Phase

Phase 1 of testing is the discovery phase, also known as crawling. The objective of the discovery phase is to find all areas of the web application. This includes all pages, URLs, HTML forms, JavaScript, Ajax (XHR) requests, etc. WAS renders web pages in the WebKit browser engine to assist in meeting this objective. No vulnerability tests are performed during the discovery phase.

1. The target URL of the web application is requested.
2. "404" error pages are identified and fingerprinted by sending requests to non-existent pages.
3. The response HTML is parsed and links and forms are identified.
4. The response is rendered in WebKit and associated resources such as JS files, CSS files, and images are requested.
5. Additional links are located on the rendered page and are crawled.
6. Any HTML forms on the page are submitted.
7. Login credentials supplied (if any) are used to authenticate to the application.
8. AJAX (XHR) requests are crawled.
9. Additional forms and links are crawled as needed in an iterative process.

Vulnerability Testing Phase

Phase 2 is the vulnerability testing phase. The objective of this phase is to detect security vulnerabilities and weaknesses within the target application. All crawled links and HTTP requests identified during the discovery phase, if not black-listed, are tested for a wide range of issues as noted below.

WAS includes tests for the following vulnerability types. Individual scans can be configured to test for all of these issues or a subset of the issues.

- Reflected cross-site scripting (XSS)
- Persistent XSS

- DOM-based XSS
- SQL injection
- Blind SQL injection
- PHP command injection
- PHP remote file inclusion
- Local file inclusion
- Directory listing enabled
- Cookies missing "secure" and/or "httponly" attributes
- Login brute force (easily-guessable credentials)
- Open redirect
- Login not submitted over HTTPS
- HTTP response splitting
- ASP.NET ViewState HMAC disabled
- Session ID in URL
- Static session ID (insufficient session expiration)
- Cross-site request forgery (CSRF)
- Clickjacking
- X-Frame-Options header not set
- Secure cookie set by insecure connection
- Overly permissive crossdomain.xml
- Sensitive form field without autocomplete disabled
- Arbitrary file uploads
- Insufficient session protection/regeneration
- Form on HTTPS submits over HTTP
- Active and passive mixed content
- Path traversal
- Basic authentication over HTTP
- Backup, old, or zipped files on the server
- Shellshock
- Httpoxy
- Apache Struts CVEs
- Apache Axis2 default admin account
- Server-side template injection (SSTI)
- Use of known-vulnerable JavaScript libraries
- Use of a known-vulnerable content management system (CMS)
- Use of known-vulnerable CMS plugins
- XML External Entity (XXE) vulnerabilities
- Edge Side Include (ESI) injection

Source IPs for External Scans

Qualys WAS external scans originate from an IP address within the following ranges. This IP range is subject to change. The most current information can be found by selecting *Help--About* within the Qualys Cloud Platform.

```
64.39.96.0/20 (64.39.96.1 - 64.39.111.254)
2600:0C02:1020:2111::/64
2600:0c02:1020:2881::/64
```