Information

Preparing Your System for Audit

Using Fortify WebInspect

MICRO FOCUS

Contents

Fortify WebInspect is an aggressive web application analyzer that rigorously inspects your entire website for real and potential security vulnerabilities. This procedure is intrusive to varying degrees. Depending on which Fortify WebInspect policy you apply and the options you select, it can affect server and application throughput and efficiency. When using the most aggressive policies, Fortify recommends that you perform this analysis in a controlled environment while monitoring your servers.

Sensitive Data

Fortify WebInspect captures and displays all application data sent between the application and server. It might even discover sensitive data in your application that you are not aware of. Fortify recommends that you follow one of these best practices regarding sensitive data:

- Do not use potentially sensitive data, such as real usernames and passwords, while testing with Fortify WebInspect.
- Do not allow Fortify WebInspect scans, related artifacts, and data stores to be accessed by anyone unauthorized to access potentially sensitive data.

Network authentication credentials are not displayed in WebInspect and are encrypted when stored in settings.

Firewalls, Anti-virus Software, and Intrusion Detection Systems

WebInspect sends attacks to servers, and then analyzes and stores the results. Web application firewalls (WAF), anti-virus software, firewalls, and intrusion detection/prevention systems (IDS/IPS) are in place to prevent these activities. Therefore, these tools can be problematic when conducting a scan for vulnerabilities.

First, these tools can interfere with WebInspect's scanning of a server. An attack that WebInspect sends to the server can be intercepted, resulting in a failed request to the server. If the server is vulnerable to that attack, then a false negative is possible.

Second, results or attacks that are in the WebInspect product, cached on disk locally, or in the database can be identified and quarantined by these tools. When working files used by WebInspect or data in the database are quarantined, WebInspect can produce inconsistent results. Such quarantined files and data can also cause unexpected behavior.

These types of issues are environmentally specific, though McAfee IPS is known to cause both types of problems, and any WAF will cause the first problem. Fortify has seen other issues related to these tools as well.

If such issues arise while conducting a scan, Fortify recommends that you disable WAF, anti-virus software, firewall, and IDS/IPS tools for the duration of the scan. Doing so is the only way to be sure you are getting reliable scan results. If it is not practical to disable these tools, you should allow exceptions within these tools for every issue that they detect related to WebInspect or a WebInspect scan.

Effects to Consider

During an audit of any type, Fortify WebInspect submits a large number of HTTP requests, many of which have "invalid" parameters. On slower systems, the volume of requests may degrade or deny access to the system by other users. Additionally, if you are using an intrusion detection system, it will identify numerous illegal access attempts.

To conduct a thorough scan, Fortify WebInspect attempts to identify every page, form, file, and folder in your application. If you select the option to submit forms during a crawl of your site, Fortify WebInspect will complete and submit all forms it encounters. Although this enables Fortify WebInspect to navigate seamlessly through your application, it may also produce the following consequences:

- If, when a user normally submits a form, the application creates and sends e-mails or bulletin board postings (to a product support or sales group, for example), Fortify WebInspect will also generate these messages as part of its probe.
- If normal form submission causes records to be added to a database, then the forms that Fortify WebInspect submits will create spurious records.

During the audit phase of a scan, Fortify WebInspect resubmits forms many times, manipulating every possible parameter to reveal problems in the applications. This greatly increases the number of messages and database records created.

Helpful Hints

- For systems that write records to a back-end server (database, LDAP, and so on) based on forms submitted by clients, some Fortify WebInspect users, before auditing their production system, backup their database, and then reinstall it after the audit is complete. If this is not feasible, you can query your servers after the audit to search for and delete records that contain one or more of the form values submitted by Fortify WebInspect. You can determine these values by opening the Web Form Editor.
- If your system generates e-mail messages in response to user-submitted forms, consider disabling your mail server. Alternatively, you could redirect all e-mails to a queue and then, following the audit, manually review and delete those e-mails that were generated in response to forms submitted by Fortify WebInspect.
- Fortify WebInspect can be configured to send up to 75 concurrent HTTP requests before it waits for an HTTP response to the first request. The default thread count setting is 5 for a crawl and 10 for an audit (if using separate requestors). In some environments, you may need to specify a lower number to avoid application or server failure. For more information, see Scan Settings: Requestor.
- If, for any reason, you do not want Fortify WebInspect to crawl and attack certain directories, you must specify those directories using the Excluded URLs feature of Fortify WebInspect settings (see Scan Settings: Session Exclusions). You can also exclude specific file types and MIME types.
- By default, Fortify WebInspect is configured to ignore many binary files (images, documents, and so on) that are commonly found in web applications. These documents cannot be crawled or attacked, so there is no value in auditing them. Bypassing these documents greatly increases the audit speed. If proprietary documents are in use, determine the file extensions of the documents and exclude them within Fortify WebInspect's default settings. If, during a crawl, Fortify WebInspect becomes extremely slow or stops, it may be because it attempted to download a binary document.
- For form submission, Fortify WebInspect submits data extracted from a prepackaged file. If you require specific values (such as usernames and passwords), you must create a file with Micro Focus's Web Form Editor and identify that file to Fortify WebInspect.
- Finally, Fortify WebInspect tests for certain vulnerabilities by attempting to upload files to your server. If your server allows this, Fortify WebInspect will record this susceptibility in its scan report and attempt to delete the file. Sometimes, however, the server prevents file deletion. For this reason, search for and delete files with names that start with "CreatedByHP" as a routine part of your post-scan maintenance.