

WHITE PAPER

Enterprise Web Security Best Practices

How to Build a Successful Security Process

You cannot achieve complete web application security in a large organization using a simple vulnerability scanner. **You need to choose the right tools and build a comprehensive and scalable enterprise web security process.**

Table of contents

3 **Fundamentals of enterprise web security**

3 Thorough: web asset discovery

4 Trustworthy: minimum false alarms

5 Efficient: automatic issue management

6 Unavoidable: scheduling and continuous integration

7 Governed: enterprise-class visibility and reporting

9 **Building an enterprise web security process**

9 Step 1: Discover web assets

10 Step 2: Create logical groups

10 Step 3: Add users and permissions

11 Step 4: Build the inventory

12 Step 5: Integrate with the Issue Tracker

13 Step 6: Schedule scans

14 Step 7: Include in the SDLC

15 The complete workflow

Fundamentals of enterprise web security

The scope of challenges related to web security grows exponentially with the size of your business. The primary reason is not the number of assets that you must supervise but the complexity of the structure. In a big enterprise, it is easier for an issue to slip through security checks. And in a big enterprise, a small issue may lead to exponentially greater consequences.

Most security offerings on the market focus on the small print, not on the big picture. If you were to compare a typical web security offering with a physical security system, it is like hiring exclusively specialized guards. If you have one small office to cover, a few inquisitive guards will keep it very safe. If you have hundreds of large buildings worldwide and the guards have no managers, guidelines, plans, or schedules, criminals will easily circumvent them. For the same reasons, a web vulnerability scanner is not enough to protect a global enterprise from attackers.

Web security challenges have a lot in common with physical security challenges. When you design a security system, you want it to be thorough, trustworthy, efficient, unavoidable, and governed. The same qualities are necessary to create a robust web security process.

When you design a security system, you want it to be thorough, trustworthy, efficient, unavoidable, and governed.

Thorough: web asset discovery

Can you or anyone else in your company list all your websites and web applications? In most large enterprises, this is not possible. Even if you meticulously keep a manual catalog of assets, you may simply not be informed about all of them. An employee of your office across the globe may create a WordPress site for a one-time marketing campaign and never tell you about it.

Consequences of unprotected assets are similar to those for a physical security system. If you have a small warehouse that is not watched, someone will easily break into it. Even if you have no valuables in that warehouse, a criminal may use your warehouse to conduct illegal operations. Similarly, a black hat hacker may use your forgotten website to conduct attacks on other companies. In both cases, this will irreparably damage your company's reputation.

A robust enterprise web security process must include activities that let you find out exactly what you need to protect. You must automatically and continuously discover all your new enterprise web assets as soon as they appear. No manual process will be thorough enough – you must have a tool that can do it for you. Such a tool must also be impossible to circumvent.



You must automatically and continuously discover all your new enterprise web assets as soon as they appear.

The best way to do it is to use the same techniques that specialized agents use when discovering new sites for search engines. This is implemented in Netsparker Enterprise. Its crawlers scan IP address ranges, top-level domains, second-level domains, and search for the organization name or other characteristics that you can configure. In a large enterprise, the number of assets found may be overwhelming so the tool also provides functions that make it easy to prioritize the results.

Web asset discovery

Pros +	Cons -
Finds most web assets that belong to the business	Does not completely eliminate human resources: requires manual verification of results
Saves resources: replaces manual cataloging efforts	

Trustworthy: minimum false alarms

A website security scanner is like an investigator. You expect it to deliver results for every potential vulnerability and you expect it to deliver minimum results that turn out to be false alarms. One false alarm may cost you much more than one overlooked minor security issue.

Let’s again use the comparison to a physical security system. If you get an alarm call from an automated system from one of your warehouses, you must dispatch a security team. Once the security team gets to the site, they must figure out why the alarm was set off. This can cost them a lot of time and effort if it is a false alarm.

The same happens when a web vulnerability scanner sets off an alarm. Due to the initial lack of trust, that alarm is usually investigated by a security expert and/or a developer. Often, a security expert is asked to reproduce the reported vulnerability. If the vulnerability is a false alarm, the expert or developer may spend a lot of time and effort trying to prove or fix something that does not exist.

Now imagine what happens if you have a physical security system with an alarm that sounds several times every night. In many cases, you find out that there was no security breach at all. After a few days, you stop trusting the alarm or even mute it. Your security system becomes completely useless if it reports false alarms. Just as a web security system becomes useless if it keeps reporting false positives because you quickly stop trusting it.

A web security system becomes completely useless if it keeps reporting false positives because you quickly stop trusting it.

If false positives occur in a small business, they are usually rare enough not to cause a loss of trust. However, in a big enterprise with thousands of assets, even a minuscule false positive rate may mean dozens of false alarms on a regular basis. Therefore, the bigger the enterprise, the more important it is to have a system that only reports real issues.

To achieve this, you can employ proof-based scanning, which is a technology used by Netsparker Enterprise. The philosophy behind it is simple: the scanner attempts to exploit every vulnerability that it finds. If the scanner can exploit it, it proves beyond doubt that the attacker can exploit it as well. This eliminates the possibility of any false positives slipping through and makes it unnecessary for security experts to reproduce vulnerabilities.



Proof-based scanning

Pros +	Cons -
Proves beyond doubt that a vulnerability may be exploited	Slightly increases scan time: requires more time than scanning without proof
Greatly improves trust in the system: helps to completely avoid false positives	
Saves a lot of resources: no need to manually confirm vulnerabilities	

Efficient: automatic issue management

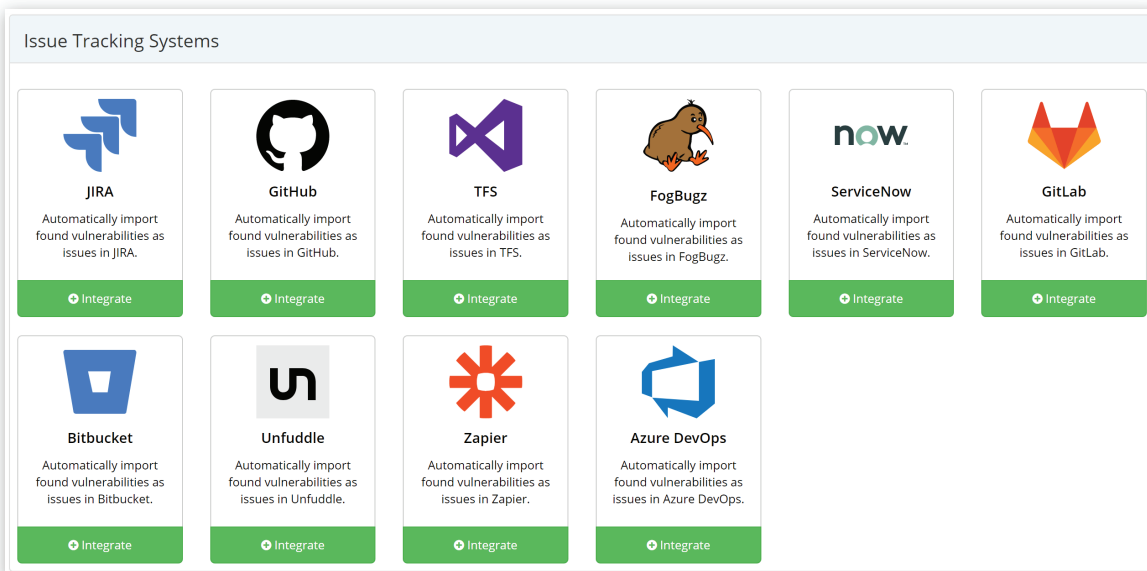
Imagine a situation, when a security guard at a warehouse discovers a break-in attempt and needs to call for an armed squad. That security guard must call the guard manager using a mobile phone. The guard manager, upon receiving the call, must email the squad captain. The squad captain, upon receiving the email, must use a web system to dispatch the squad. The whole process is so complex that it can fail at any stage, for example, if the guard manager's phone is turned off. And even if it is successful, it takes so long that when the squad arrives, the burglars are long gone.

The described situation seems absurd but it is a reality in many enterprises that do not integrate their software systems. A web security equivalent would be a penetration tester who asks the security manager to ask the service owner to ask the developer to fix the discovered vulnerability. Vulnerability scanning and reporting must be integrated with issue tracking systems that are already used by your business. To be efficient, the vulnerability scanner must be able to read and create issues without human intervention.

Vulnerability scanning and reporting must be integrated with issue tracking systems that are already used by your business.

An enterprise-class web security system must automate discovery, reporting, and remediation processes. When a vulnerability is discovered, the system should create an issue in the issue tracker and assign it to the preconfigured person or team. When an issue is reported as closed in the issue tracking system, the system should verify if the vulnerability is gone. If it is not gone, the system should reopen the issue.

This is exactly how Netsparker Enterprise works. You can integrate it with all the leading issue trackers and configure in a way that aligns with your development strategy. This way, you can automate the whole process and just have your service owners or project managers supervise it.



Integration with issue trackers

Pros +	Cons -
Greatly speeds up the remediation process	Initially, it may cause a large number of issues to appear in the issue tracking system
Greatly increases security: eliminates the possibility of vulnerabilities never being addressed	
Saves resources: may require supervision only	

Unavoidable: scheduling and continuous integration

If you hire security guards but they don’t watch your warehouse between 2.00 AM and 3.00 AM or they don’t watch the back door, their skills are useless. Similarly, if any website or web application is not regularly scanned, it is more prone to attacks. New vulnerabilities are discovered regularly and exploited shortly after discovery.

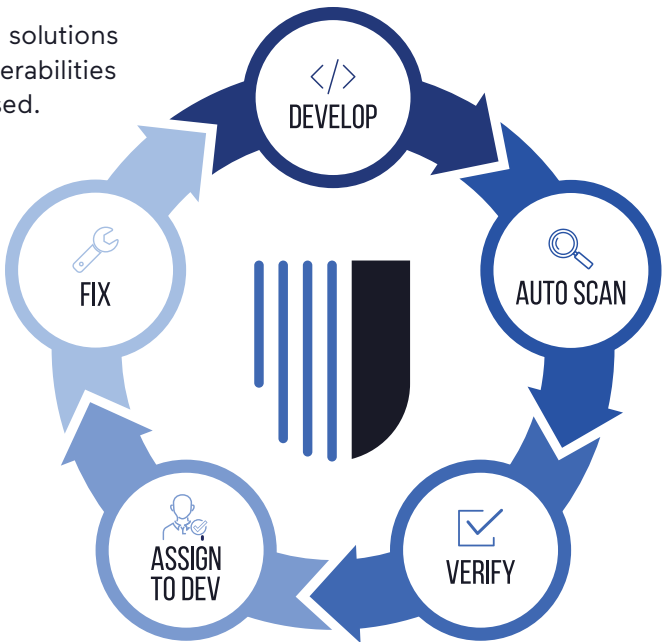
Every web security process should involve scheduled nightly scans. In a large enterprise, the number of websites and web applications may make this very resource-intensive. That is why an enterprise-class web security solution must let you classify websites and assign different schedules depending on different risk factors.

However, in the case of applications that are developed in-house, it is not the best idea to test websites and web applications when they make it to production. It introduces unnecessary risks and is a huge waste of time. Every new version of a web application or website should be scanned for vulnerabilities as soon as it is created.

Every new version of a web application or website should be scanned for vulnerabilities as soon as it is created.

To manage complex development processes, enterprises most often use continuous integration (CI) solutions. They help automate compilation, deployment, and many other tasks, and they can help automate vulnerability scanning as well. If you integrate your enterprise-class web vulnerability scanner with a CI system, every time a developer commits new code and that code is compiled, the scanner will check the code for vulnerabilities.

You can integrate Netsparker Enterprise with different CI solutions to achieve such automation. This way, any potential vulnerabilities introduced by the new code will be immediately addressed. You can even configure the CI solution to mark the process as failed if vulnerabilities are discovered. This way, there is no way for a security issue to make it past the development stage.



Integration with CI systems

Pros +	Cons -
Makes it impossible for in-house code security vulnerabilities to make it past the development stage	CI systems need a little more time to process commits because the vulnerability scanner makes an incremental scan of every commit
Greatly increases security: reduces the number of potential vulnerabilities in live systems	
Saves resources: issues are managed only by the person or team who caused them to appear	

Governed: enterprise-class visibility and reporting

In an enterprise, the number of assets and limited resources make it impossible to personally oversee every issue and to eliminate every vulnerability as soon as it appears. Large organizations also suffer from much more inertia. In some cases, enterprises must have access to very specific information to meet legal requirements.

That is why the bigger the enterprise, the bigger the need for visibility and prioritization of security-related issues. With the right information, you can group and prioritize discovered vulnerabilities. You can make sure that high-risk issues in high-risk applications are dealt with immediately. If required, you can also meet compliance requirements in your industry.

Even if you implement a robust web security process for your enterprise, without the right tools you have no idea whether it is working. You need feedback from the system that provides you with information such as how often are security issues introduced, how quickly are they fixed by every team or external entity, as well as which technologies, third-party applications, libraries, or tools cause the most vulnerabilities to appear. Such information gives you decision power: where to invest time and money for training or purchases. It also gives you risk visibility.

The solution to this is an executive-oriented interface that uses business intelligence technologies to provide you with just the right amount of information. In an enterprise, nobody would have the time to manually look through hundreds or possibly even thousands of issues one-by-one. Web security reports need to be concise but they need to address the right factors. Reports also may require to be in a specific format or have specific information to meet security compliance requirements.

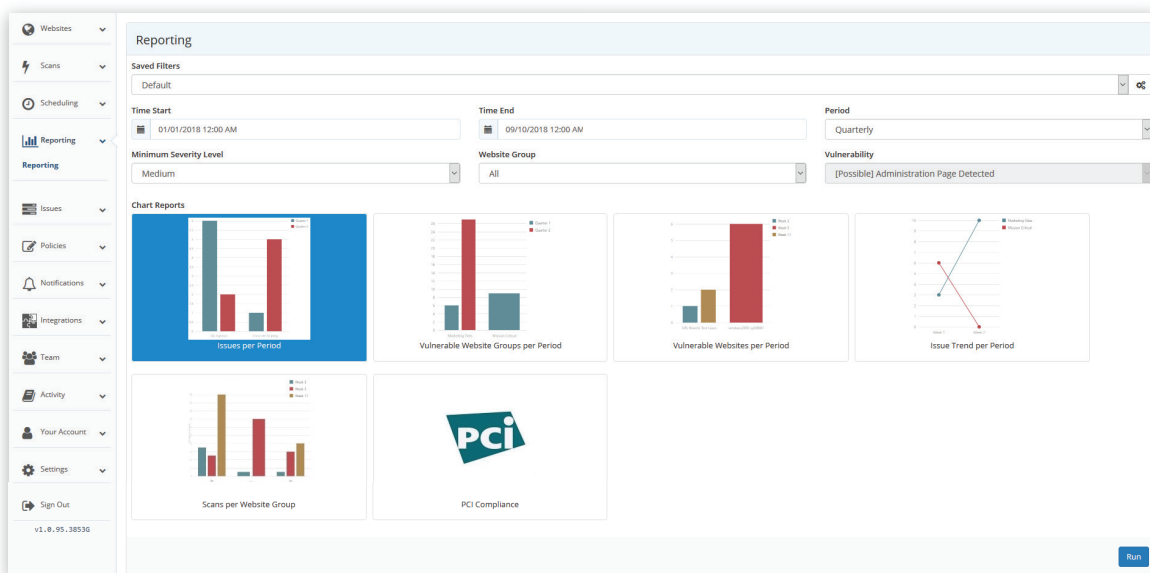


In an enterprise, web security reports need to be concise but they need to address the right factors.

Most web vulnerability scanners are meant to be used by penetration testers and are aimed at smaller businesses. The dashboard and reporting system of Netsparker Enterprise was designed especially with enterprise security managers and executives in mind. Netsparker Enterprise also includes specialized compliance reports for key security compliance standards.

Enterprise reporting

Pros +	Cons -
Provides visibility of how well the security process is performing (is it a success or does it require changes)	None
Provides information on risk areas: where you need to focus your resources and investments	



Building an enterprise web security process

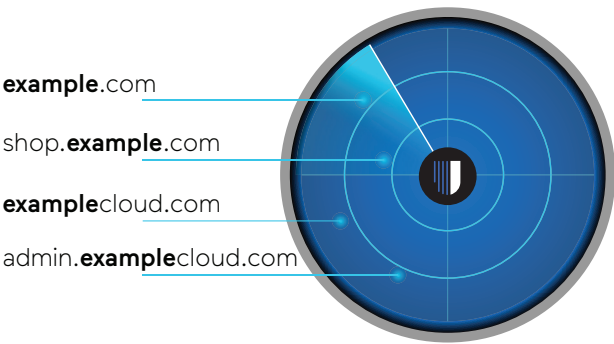
The following is an example of a tested best-practice enterprise web security process. This example is a result of several years of experience with creating enterprise web security processes. It uses Netsparker Enterprise as the central tool of choice.

This example may not align perfectly with your enterprise’s team organization or work practices. Please treat it as a baseline and customize it to your specific needs. If you need more information about our experiences with designing such processes or if you need help with building your specific process, please contact us.

Step 1: Discover web assets

You can use Netsparker Enterprise to discover your assets immediately with no other setup needed. All you need to do is to provide at least one of the following types of information: second-level domains, organization names, or IP address ranges.

Netsparker Enterprise finds matching entries in its databases, which are populated by agents that continuously crawl the Internet. Information is gathered from many different sources, including DNS and WHOIS records, SSL certificates, and more. As a result, you get a list of websites to treat as the starting point. At this point, you should inquire with relevant departments in your organization about any of the websites that you were previously not aware of. Once you start discovery, it continues in the background giving you new results (if any) every day.



Note that sites on staging servers, QA servers, OAT servers, development servers, etc. may be more difficult to discover automatically and may need to be added manually later. However, they are key to lowering risk because they help you catch vulnerabilities before you publicly expose them.

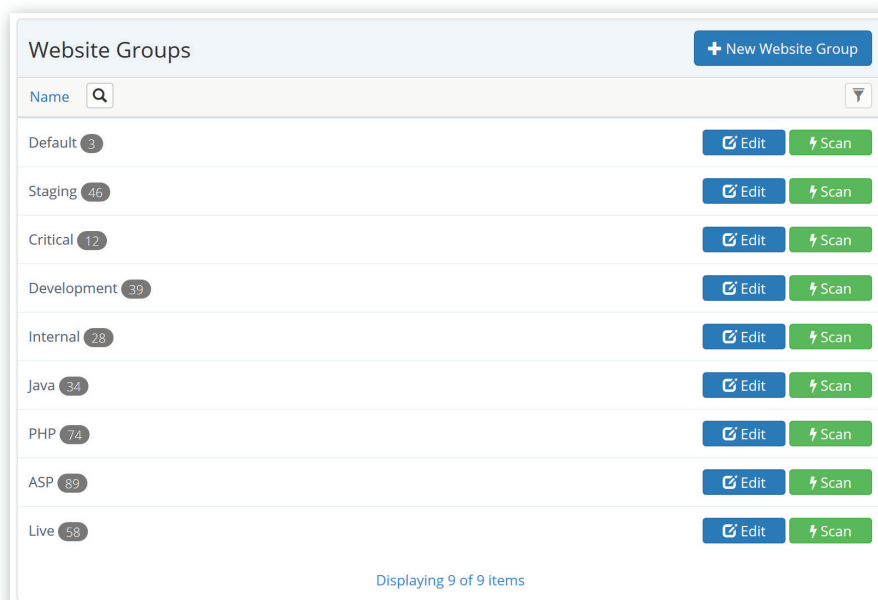
Discovered Websites							Export to CSV	+ Create	Ignore
Authority	IP Address	Second Level Domain	Top Level Domain	Organization Name	Status				
test.example.fm	66.111.4.54	example	fm		New		Create	Ignore	
testtestex.example.kz	94.247.135.17	example	kz		New		Create	Ignore	
www.example.cf	195.20.50.226	example	cf		New		Create	Ignore	
www.example.lk	162.144.80.21	example	lk		New		Create	Ignore	
mail.example.lk	162.144.80.21	example	lk		New		Create	Ignore	
example.lk	162.144.80.21	example	lk		New		Create	Ignore	
example.my	104.18.62.107	example	my	CloudFlare	New		Create	Ignore	
example.com.au	103.224.212.253	example	com.au		New		Create	Ignore	
www.example.pk	104.27.133.168	example	pk		New		Create	Ignore	
example.pk	104.27.133.168	example	pk	CloudFlare	New		Create	Ignore	
www.example12.com	23.20.239.12	example12	com		New		Create	Ignore	
www.example2.click	104.19.152.30	example2	click	CloudFlare	New		Create	Ignore	
example2.click	104.19.151.30	example2	click	CloudFlare	New		Create	Ignore	
www.example.pp.ua	195.149.114.22	example	pp.ua		New		Create	Ignore	
mail.example.pp.ua	195.149.114.22	example	pp.ua		New		Create	Ignore	
www.example.cc.ua	195.234.4.57	example	cc.ua		New		Create	Ignore	
example.cc.ua	195.234.4.57	example	cc.ua		New		Create	Ignore	

Step 2: Create logical groups

The key to managing a large number of assets is efficient grouping. Netsparker Enterprise uses the concept of logical groups that are similar to tags. Groups are used for responsibility assignment, prioritization, scheduling, reporting, and more. A single asset may be part of as many groups as you need it to be. After you create groups, immediately add discovered and imported assets to these groups as required.

As a best practice, you can create groups that represent the following aspects:

- ✓ **Criticality** (e.g. Mission-Critical, Private Information, Promotional)
- ✓ **Departments** (e.g. Marketing, Development, Third Party Contractor)
- ✓ **Offices** (e.g. NYC Office, SF Office, London Office)
- ✓ **Stages** (e.g. Staging, QA, OAT, Production)
- ✓ **Environment** (e.g. Internal, External)
- ✓ **Teams** (e.g. Team1, Team2)
- ✓ **Technology** (e.g. Java, PHP, ASP)
- ✓ **Maturity** (e.g. New, Live, Legacy)



Some examples of how you can apply logical groups:

- ✓ **Use logical groups that represent development technologies to schedule scans that only apply to a given technology, thus saving time.**
- ✓ **Differentiate the scan schedule on the basis of logical groups that represent maturity and criticality, for example, scan New+MissionCritical sites nightly and scan Legacy+Promotional sites on weekends only.**
- ✓ **View reports for logical groups that represent teams to see how well each team is performing in terms of fixing vulnerabilities.**

Step 3: Add users and permissions

In a large organization, many parties have varying levels of interest in web asset security. Different parties may also have their access limited to certain scopes of information. Netsparker Enterprise is designed to be used by all stakeholders: executives (e.g. CEO, CTO), security (e.g. CSO, security engineers, penetration testers), development (e.g. product owners, service owners, project managers, developers, QA), and more. It is best practice to involve all of them in the security process.

When adding users to Netsparker Enterprise, you define their permissions by the type of activity and by the scope of web assets. Type-of-activity permissions depend on whether the stakeholder requires only report information, whether they need to perform scans, and whether they need to manage issues found during scans. For example, executives will probably only require report access, developers will probably only need issue access, but penetration testers must have full access.

Web asset permissions are based on logical groups created in the previous step. You may permit the user to access only some of the groups. For example, personnel of your New York office will probably not need access to your London office assets.

New Team Member

Name

John Smith

Email

admin@netsparker.com

Phone Number ?

🇺🇸

(415) 877-4450

Account Permissions

☐ Administrator
 ☐ Manage Websites

Scan Permissions

☐ Start Scans ?
 ☒ View Scan Reports
 ☒ Manage Issues

Website Groups

Select All

☒ Staging
 ☒ Critical
 ☐ Development
 ☐ Internal

Step 4: Build the inventory

Once the preparatory steps are complete, you can now build an inventory by adding discovered web assets and assigning them to logical groups, thus giving selected users permissions. If needed, you may manually enter additional web assets in this step or import them.

To ensure that the vulnerability is addressed, the best practice is to identify one person who is responsible for a particular web asset. That person becomes the single point of contact and delegates activities associated with addressing the vulnerability.

In Netsparker Enterprise, this is implemented by assigning a technical contact to every web asset in the inventory. The technical contact is a selected Netsparker Enterprise user with sufficient permissions. If the scanner finds a new issue, it assigns that issue to this user by default (unless other conditions cause the scanner to assign the issue to another user).

Edit Website

✕ Delete

Name

PHP Testsparker

URL

http://php.testsparker.com/

Verified

Default Protocol

HTTP

HTTPS

Website Groups

Select All

☒ Stage
 ☒ Test

Technical Contact ?

Admin (admin@netsparker.com)

Agent Mode ?

Cloud

Internal

Include in the security policy

Your company's security policy surely already includes web security. You may need to modify that security policy to include new features added by your enterprise-class web security solution

Step 5: Integrate with the Issue Tracker

In a large organization, every new software solution that is to be accessed by a large number of users may introduce major resource costs. To streamline processes, you want to be able to use software that everyone already knows and uses. Therefore, a vulnerability scanner that works only within its own interface is not suitable for the purposes of an enterprise.

Every enterprise that deals with software development most probably already uses a standalone issue tracker or an issue tracker that is bundled with a project management system. Therefore, you want the vulnerability scanner to be able to work together with such tools. Issues found by the scanner should be added into the issue tracking system and two-way communication should also be possible, for example, to automatically retest closed issues.

Netsparker Enterprise provides such options via its integration endpoints and notifications. When your asset inventory is ready, first create all the required integration endpoints with selected users and/or teams that issues are to be assigned to. Then, create scan completion notifications for websites or website groups and align them with the integration endpoints. Now, each time a scan is finished, Netsparker Enterprise creates an issue in the issue tracker for every found vulnerability and assigns it to the right team and/or user. Your development teams can take it from there with no need to manage the issues in Netsparker Enterprise. With some issue trackers, you can also trigger a scan when a user of the issue tracker changes the status of an issue to closed. If such a scan fails, the scanner can also automatically reopen the issue.

Issue Tracking Systems



JIRA

Automatically import found vulnerabilities as issues in JIRA.

Integrate



GitHub

Automatically import found vulnerabilities as issues in GitHub.

Integrate



TFS

Automatically import found vulnerabilities as issues in TFS.

Integrate



FogBugz

Automatically import found vulnerabilities as issues in FogBugz.

Integrate



ServiceNow

Automatically import found vulnerabilities as issues in ServiceNow.

Integrate



GitLab

Automatically import found vulnerabilities as issues in GitLab.

Integrate



Bitbucket

Automatically import found vulnerabilities as issues in Bitbucket.

Integrate



Unfuddle

Automatically import found vulnerabilities as issues in Unfuddle.

Integrate



Zapier

Automatically import found vulnerabilities as issues in Zapier.

Integrate



Azure DevOps

Automatically import found vulnerabilities as issues in Azure DevOps.

Integrate

Continuous Integration Systems



TeamCity

Automatically trigger security scans with TeamCity

Integrate



Jenkins

Automatically trigger security scans with Jenkins

Integrate



Bamboo

Automatically trigger security scans with Bamboo

Integrate



GitLab

Automatically trigger security scans with GitLab

Integrate



Azure DevOps

Automatically trigger security scans with Azure DevOps

Integrate

New JIRA Integration

Name ⓘ

Jira Integration

Mandatory

URL ⓘ

https://netsparker.atlassian.net/

Username or Email ⓘ

user1@netsparker.com

Password ⓘ

.....

Project Key ⓘ

NET

Issue Type ⓘ

Bug

Title Format ⓘ

Vulnerability - {0}

Optional

Assigned To ⓘ

User 1

☐ Auto assign to person who started the scan

Reporter ⓘ

Netsparker

Priority ⓘ

Reopen Status ⓘ

To Do

Resolved Status ⓘ

Done

Labels ⓘ

Due Days ⓘ

Due Date is not activated for JIRA projects by default. Please see JIRA documentation to get help on [Configuring Fields and Screens](#) if you want to use this field.

Custom Fields

Name

Value

+ New Custom Field

Create Sample Issue

Save

Step 6: Schedule scans

An enterprise may have hundreds or even thousands of websites and web applications. If every single one of them was to be scanned every day, it would require a lot of computer resources. Therefore, many security managers schedule scans depending on various factors. For example, an e-commerce system that is used by thousands of customers and contains a lot of personal data may need to be scanned very often. However, a purely promotional website hosted in a separate instance may require only weekly scans just to make sure it is not defaced and/or taken over.

In Netsparker Enterprise, you can take all these factors into consideration and create a schedule that works best for you. You can vary scan scopes and scan types, pick the right dates and times, and to save more time, manage schedules on the basis of website groups and not individual websites.

New Scheduled Scan

Target URL

Target website URL, including the path.

Default

Scan Type

Full

Incremental

Scheduling Options

Name

Start Time

19/04/2019 06:14 PM

Schedule

Once

Scan Options

General

Scan Scope

Additional Websites

Imported Links

URL Rewrite

Form Authentication

Basic, NTLM/Kerberos

Client Certificate

Header Authentication

Scan Time Window

Notifications

PCI Scan

Scan Policy ⓘ

Default Security Checks (Default)

Report Policy ⓘ

Default Report Policy (Default)

Custom Cookies

Enter any required cookies in the format `cookieName=value`. The value must be URL encoded. Use semicolons (;) to separate multiple cookies.

Crawling

☒ Find & Follow New Links ⓘ

☒ Enable Crawl & Attack at the Same Time ⓘ

Maximum Scan Duration ⓘ

48 hours

Schedule

13

Scheduled Scans

+ Schedule Scan

+ Schedule Group Scan

Name	Website/Group	Type	Run Type	Last Scan	Next Execution Time	
Schedule	Php Testsparker http://php.testsparker.com/	Website	ONCE	No Scan Executed	Starting in 23h 57m	+ Scan Edit Delete
Weekly	Php Testsparker http://php.testsparker.com/	Website	WEEKLY	No Scan Executed	Starting in 1d 23h 56m	+ Scan Edit Delete
aspnet	Asp Testsparker http://aspnet.testsparker.com/	Website	ONCE	17/12/2018 09:10 AM	N/A	+ Scan Edit Delete
test	Php Testsparker http://php.testsparker.com/	Website	ONCE	No Scan Executed	N/A	+ Scan Edit Delete
Group	new	Group	ONCE	No Scan Executed	N/A	+ Scan Edit Delete
Daily	Php Testsparker http://php.testsparker.com/	Website	DAILY	22/02/2019 11:01 AM	Starting in 23h 47m	+ Scan Edit Delete

Displaying 6 of 6 items

Step 7: Include in the SDLC

The primary purpose of scheduled scans is to make sure that vulnerabilities do not appear due to, for example, issues found in third-party software. Scheduled scans also act as the last line of defense for your own applications. However, your first line of defense against vulnerabilities should start a long time before code makes it to production.

To eliminate security issues at the earliest possible time, you must make sure that every piece of code that makes it into the code repository is compiled and scanned for vulnerabilities. A commit should never make it out of a feature branch unless it is deemed secure.

The best way to achieve this is to integrate your security scanner with your continuous integration (automation) solution. Most probably, when your developer commits anything to the repository, a continuous integration process is triggered. This process involves compiling the application to make sure that there are no errors and usually automatic tests. This stage must also include a security vulnerability scan. And just as the feature branch cannot be merged if it does not compile, it should not be merged if the scanner can find critical vulnerabilities.

Netsparker Enterprise supports such integration and can also work together with the issue tracker in an SDLC setup. This means, that you may also automatically create issues and assign them to the relevant user, for example, the committer, the developer responsible for bug tracking, etc. In case of less critical vulnerabilities, this may be your method of choice to make sure that issues are addressed (instead of rejecting the commit).



The complete workflow

The complete, best practice workflow achieved after implementing the seven steps above is as follows:

1. [START] Netsparker Enterprise starts a scan that is triggered by one of the following sources:
 - * A new code commit (via the continuous integration system) or
 - * Completion of a ticket (via the issue tracker) or
 - * A scheduled event
2. Netsparker Enterprise completes the scan:
 - * If no vulnerabilities are found, the workflow is complete. [END]
 - * If vulnerabilities are found, the workflow continues.
3. For every vulnerability that Netsparker Enterprise found in the previous step:
 - * If the severity of the vulnerability is at or above the configured reaction threshold:
 - If the vulnerability was found as a result of a new code commit, the commit is rejected or a new ticket is created in the issue tracker and assigned to the party responsible for introducing the vulnerability.
 - If the vulnerability was found during a scan initiated by the completion of a ticket in the issue tracker, no new ticket is created (unless the scan discovers new vulnerabilities) and the original ticket is reopened.
 - If the vulnerability was found during a scheduled scan, a new ticket is created in the issue tracker and assigned to the default party responsible for the web asset group or individual web asset.
 - * If the severity of the vulnerability is critical, Netsparker Enterprise sends a notification (email, SMS, Slack) to the party responsible.
 - * The party responsible begins to fix the vulnerability.
 - * The party responsible completes the fix and marks the issue as fixed. The workflow goes back to [START].
 - * If all vulnerabilities are analyzed and all of them are below the configured reaction threshold, the workflow finishes. [END]



About Invicti Security

Invicti Security is changing the way web applications are secured. A global leader in web application security for more than 15 years, Invicti's dynamic and interactive application security products help organizations in every industry scale their overall security operations, make the best use of their security resources, and engage developers in helping to improve their overall security posture. Invicti's product [Netsparker](#) delivers industry-leading enterprise web application security, while [Acunetix](#) is designed for small and medium-sized companies.

Netsparker

About Netsparker

Scalable tech is fundamental for enterprises. Established by security expert Ferruh Mavituna in 2009, Netsparker can scan thousands of websites and web apps then prioritize and verify vulnerabilities with proprietary Proof-Based Scanning technology.

www.invicti.com

