netsparker

Proof-Based Scanning™ with Netsparker

The Key to Confident Automation

What Is Proof-Based Scanning?

Proof-Based Scanning is a proprietary vulnerability scanning technology that Netsparker uses to safely exploit web application vulnerabilities and extract internal sensitive data as proof. Without proof, issues identified by automated scanners still have to be manually verified by security experts. Proof-Based Scanning changes the rules of the game by delivering solid proof that a vulnerability is real, enabling confident automation and bringing a host of other benefits.

Proof-Based Scanning Saves Time and Resources

With Proof-Based Scanning, security staff don't have to manually verify proven vulnerabilities and developers can get straight to work fixing the issue.

Without Proof-Based Scanning

Hey Lucy, you have an SQL injection vulnerability in admin.php

Hey Frank, I can't find anything, are you sure?

Yes, I'm sure, please fix it.

But where exactly?

The user ID

OK, I've fixed it.

No, it's still there, check again.

Oh, I missed one case, should be good now.

Issue resolved

With Proof-Based Scanning

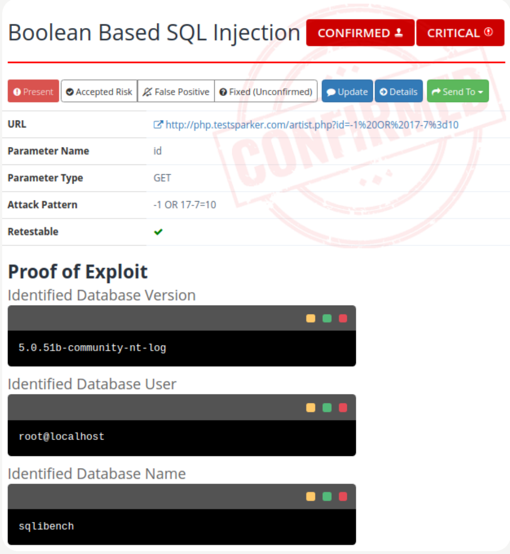
SQL injection vulnerability confirmed in admin.php

Detected by injecting code into userID (POST parameter)

Proof: Database name is sqlibench, user is root@localhost

Developer fixes vulnerability and resolves the issue

Netsparker automatically retests vulnerability and closes the issue


Boolean Based SQL Injection CONFIRMED CRITICAL

Present | Accepted Risk | False Positive | Fixed (Unconfirmed) | Update | Details | Send To

URL	http://php.testsparker.com/artist.php?id=-1%20OR%2017-7%3d10
Parameter Name	id
Parameter Type	GET
Attack Pattern	-1 OR 17-7=10
Retestable	✓

CONFIRMED

Proof of Exploit

Identified Database Version

```
5.0.51b-community-nt-log
```

Identified Database User

```
root@localhost
```

Identified Database Name

```
sqlibench
```

Proof-Based Scanning Improves Working Relations

Without Proof-Based Scanning:

- All vulnerabilities reported by the scanner need to be verified manually, creating an information overload and extra work.
- Until confirmed by the security team or developers, each scan result could be a false positive.
- Developers often waste a lot of time and effort looking for non-existent issues. They start treating security as an annoying chore and distrusting the security team and their tools.
- Security bug reports are treated with suspicion and often need additional communication to agree on the issue.

Developer

They keep flooding me with security bugs that take ages to fix or turn out to be complete nonsense! I sometimes waste hours checking code for non-existent issues... Even if the issue is real, it takes so much time to get all the information I need that releases can get delayed.

Security Analyst

As soon as they see a security issue, the developers start arguing instead of getting to work fixing it. I spend a lot of time double-checking results from the scanner and then I waste even more time convincing the devs that the issue is real and telling them how to fix it!

Developer

When I see a Netsparker-confirmed vulnerability report in my inbox, I can get straight to work on it. I get detailed information about the bug, see how the vulnerability can be exploited, and know what the impact is. The security guys always have time to educate us about the latest threats and secure coding practices.

Security Analyst

Automatic verification takes a lot of work off my shoulders. For the first time, I can really trust the scanner because it provides solid proof that a vulnerability is real and exploitable. The devs get confirmed bug reports directly in their issue tracker, so I can focus on vulnerability management and education.

With Proof-Based Scanning:

- The scanner automatically verifies many direct-impact vulnerabilities, which means less work for the security team.
- Every vulnerability that is automatically confirmed comes with solid proof that the issue is real.
- Developers get all the information they need to completely fix the issue, including a proof of exploit and accurate remediation guidance.
- Security bug reports are trusted by the developers and handled without the need for additional communication.

Problem:

Traditional web vulnerability scanners can return false positives, so each result needs to be verified manually.

Solution:

If the scanner can safely exploit a suspected vulnerability and extract proof, the issue is definitely not a false positive. Netsparker's Proof-Based Scanning technology eliminates uncertainty from automated vulnerability testing, so you always know your true security status. Confirmed results can immediately go to developers with no manual processing, leaving the security team with more time to focus on complex vulnerabilities that require human expertise.

Problem:

Security teams can have problems convincing developers that a vulnerability really exists.

Solution:

Each result confirmed with Proof-Based Scanning includes sample data that was safely extracted from the target application to prove that the issue is real and exploitable. The developer gets a vulnerability report that includes the proof, the attack payload, vulnerability details, and recommendations for fixing the issue. This greatly improves communication and collaboration between security teams and developers.

netsparker

Proof-Based Scanning™ with Netsparker

The Key to Confident Automation

Automation at Scale Requires 100% Accuracy

Automation is the key to agile software development and DevOps workflows, allowing small teams to develop and maintain large applications with frequent deployments.

To add automated security testing into these processes to build DevSecOps, your results must be 100% accurate. Otherwise you risk burdening your developers with false positives, losing their trust and delaying the whole development pipeline.

Vulnerability reports confirmed with Proof-Based Scanning can go straight into issue trackers without the bottleneck of manual verification by the security team – and without the risk of false positives.

Restoring Confidence in Vulnerability Scanning

Typical DAST Tools

- ✗ **Limited coverage:** Legacy tools can have problems scanning pages that require authentication or heavily rely on JavaScript to generate content. This can leave large sections of application environments without any automated testing.
- ✗ **False positives:** Simple automated scanners are notorious for sounding false alarms. This means that any reported item could potentially be a false positive, requiring security staff to check issues manually or risk sending developers to hunt for non-existent bugs.
- ✗ **Limited information:** Developers don't get enough information from vulnerability reports generated by legacy DAST tools and need to follow up with security personnel to understand the issue.
- ✗ **Bottleneck to automation:** Because suspected issues need to be verified and described manually, true automation is not possible. It also means that security teams and developers don't fully trust the DAST results.

Netsparker with Proof-Based Scanning

- ✓ **Industry-leading coverage:** Netsparker supports modern authentication schemes and adds asset discovery and advanced crawling to maximize web app testing coverage, even with JavaScript-heavy pages generated using popular web frameworks.
- ✓ **Confirmed results can't be false positives:** Vulnerabilities confirmed with Proof-Based Scanning come with actual proof that they are exploitable and not false positives.
- ✓ **Detailed information:** Developers get vulnerability reports that they can act on immediately. For automatically confirmed issues, each report includes proof of the vulnerability, impact information, and recommended remedies.
- ✓ **Confident automation:** Results from Proof-Based Scanning can go directly into automated workflows without the risk of false positives. The DAST solution becomes a trusted and essential partner in the application development lifecycle, not just another helper utility.

What's in It for Me?

Security Analyst

- Many vulnerabilities are confirmed automatically, saving me a lot of manual work on separating false positives from real issues.
- I don't have to convince developers that an issue is real – they get proof that a vulnerability is exploitable plus all the information they need to fix the bug.
- Combined with Netsparker's severity ratings, I can clearly see which vulnerabilities are directly exploitable and need to be addressed immediately.
- Confirmed issues are 100% real, so they can go straight into the issue tracker without extra manual steps.

Developer

- I don't waste my time on asking for confirmation or chasing false positives.
- When a security issue confirmed by Netsparker arrives, I can get straight to work because I know it is real.
- The reports I get from Netsparker include the proof of exploit along with detailed information to help me locate the issue, understand its impact, and fully fix the bug so it doesn't come back later.
- The security guys provide me with expert support and security education instead of arguing about issues.

Testimonials

> Netsparker provides very thorough scans that are easy to understand, and include well-made proofs of the vulnerabilities found during a scan. This results in saving time and money instead of going around on a wild-goose chase. We now have a lower cost of remediation due to having almost zero false positives.
>
> **PAOLO DA ROS** *CryptoNet*

> Netsparker continually executes scans in a more optimized way and delivers actionable results every time. The false-positive free scanning means that Netsparker has already attempted to validate the finding for itself before it provides the results, thus eliminating the need for our consultants to spend time chasing down false positives.
>
> **JIM BROOME** *DirectDefense*

> I have a hard time finding any negative aspects to Netsparker Enterprise. It is hands down a great tool – all you could wish for from an automated web security scanner. Easy to use and detailed with a low false positive rate.
>
> **KLEMEN STIRN** *HESK*

netsparker

CONTACT US
Email: support@netsparker.com
Website: www.netsparker.com

FIND US
twitter.com/netsparker
facebook.com/netsparker
linkedin.com/company/netsparker