KEY TAKEAWAYS FROM THE

# 2020 WEB APPLICATION VULNERABILITY REPORT

## Analysis

- 5,000 random and anonymous targets (websites, web applications, servers, network devices) scanned by Acunetix web application security solution
- 156,291 web scans and 134,361 network scans performed from March 19 to February 20
- 222,000 vulnerability alerts triggered per month on average
- High Severity and Medium Severity vulnerabilities were further analyzed
- Based on 15 years of market experience, the report is being generated for the 15th consecutive year

**5,000** TARGETS | **290k** SCANS | **222k** ALERTS/MONTH | OVER **15** YEARS EXPERIENCE

## Summary of Findings

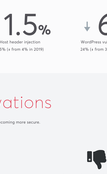
Whilst we're making progress, we are still far from being secure on the web.

- 26% of websites have high severity vulnerabilities
- 63% of websites have medium severity vulnerabilities
- 21% of web applications are vulnerable to XSS.
- 24% of websites have WordPress vulnerabilities

## Year-Over-Year Trend

The total number of web and network perimeter vulnerabilities reported year over year is decreasing.

↓ 9%




## 2020 vs. 2019

### TRENDS FOR DETECTED VULNERABILITIES

↑ 1% Remote code execution (RCE) 3% (↑ from 2% in 2019)

↓ 6% SQL injection (SQLi) 8% (↓ from 14% in 2019)

↑ 2% Directory traversal 4% (↑ from 2% in 2019)

↓ 8% Cross-site scripting (XSS) 25% (↓ from 33% in 2019)

↓ 9% Vulnerable JavaScript libraries 24% (↓ from 33% in 2019)

~ Server-side request forgery (SSRF) 1% (0% in 2019)

↓ 15% Cross-site request forgery (CSRF) 36% (↓ from 51% in 2019)

↓ 1.5% Host header injection 2.5% (↓ from 4% in 2019)

↓ 6% WordPress vulnerabilities 24% (↓ from 30% in 2019)

## Key Observations

Web applications in general are slowly becoming more secure.

Web applications that are protected by web vulnerability scanning are becoming more secure.

Websites and web applications scanned for the first time in 2019 have more vulnerabilities.

New developers lack the knowledge required to avoid vulnerabilities. These developers are also working within a development structure that does not promote web security.

## Trends in Programming

- JavaScript libraries are increasingly being used in web application development due to a growing demand for interactive web applications. Many have known vulnerabilities.
- The PHP programming language remains as popular as before.
- The second most popular language is ASP.NET, but developers more and more often use other, less popular server-side languages.
- The percentage of PHP vulnerabilities has declined a lot. The percentage of ASP or ASP.NET vulnerabilities is growing.
- The percentage of vulnerabilities in Apache/nginx has declined a lot. The percentage of IIS vulnerabilities is growing.

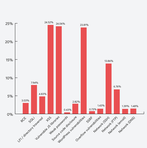

DATA OBTAINED FROM:
**https://w3techs.com/technologies/history_overview/programming_language/ms/all/2020**

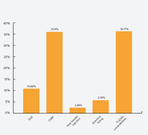

### OUR CONCLUSIONS

- The PHP+Apache/nginx platform is becoming more secure, mature, and robust. The market also keeps favoring this platform.
- The ASP/ASP.NET+IIS platform is slowly losing popularity. At the same time, it is still not as robust and mature as we would hope.
- ASP/ASP.NET web applications are more actively developed. The high percentage of vulnerabilities may be caused by active development.

## High Severity Vulnerabilities | Medium Severity Vulnerabilities

Vulnerabilities in the below graphs are sorted according to their severity:




## Vulnerability Analysis

### WHAT TO LOOK OUT FOR

### REMOTE CODE EXECUTION (RCE)

RCE is at the top of the high severity list of vulnerabilities. An attacker can use this vulnerability to run arbitrary code in the web application.

- The percentage of web applications vulnerable to RCE is low but it was much lower last year (2%).
- This is worrying because this vulnerability can cause serious damage and must be fixed as first priority.
- Such vulnerability appears because of poor design and programming, even when the best-of-class software and components are selected.


### SQL INJECTION (SQLI)

An SQL injection (SQLi) attack is possible if the developer does not examine or validate user input.

- Close to 8% of analyzed targets had at least one SQLi vulnerability.
- This was very unexpected. SQL injections first appeared in 1998. All major development environments and frameworks include tools to eliminate them.
- SQL injections should not be so common and are likely to appear because of poor design and programming.


### LOCAL FILE INCLUSION AND DIRECTORY TRAVERSAL

Local file inclusion (LFI) and directory traversal (path traversal) vulnerabilities let the attacker access the host system.

- We found 4% of sampled targets vulnerable to directory traversal.
- A further 1% were vulnerable to local file inclusion.
- Last year, the figure for directory traversal was only 2%. This is worrying because this is a very old and well-known vulnerability.


### CROSS-SITE SCRIPTING (XSS)

Cross-site scripting (XSS) occurs when the attacker injects malicious scripts into a web page, usually JavaScript.

- An alarming 25% of sampled targets were vulnerable to some type of XSS.
- Thankfully, this is less than last year, but developers still have a lot of work to do to defend users.
- New JavaScript templates and frameworks keep appearing on the market and gain popularity. Unfortunately, versions of these templates and frameworks with known vulnerabilities are also in use.


### VULNERABLE JAVASCRIPT LIBRARIES

JavaScript libraries help to make development faster and easier, but some library versions can be vulnerable.

- 24% of targets use JavaScript libraries with known XSS vulnerabilities.
- Libraries like JQuery are gaining more traction due to use in interactive websites.


### WORDPRESS (AND OTHER CMS) VULNERABILITIES

WordPress is so popular that it is no surprise that attackers focus on it. When it comes to WordPress security, there are three components: WordPress core, UI themes, and functionality plugins.

- 35% of sampled targets had one or more vulnerabilities linked to CMS platforms.
- The impact of these vulnerabilities can vary depending on the type of vulnerability (XSS, SQLi, RCE etc).
- This means that web applications are still quite vulnerable, but even so, this number is much less than for the last year.


### TLS/SSL VULNERABILITIES

Transport Layer Security (TLS) and its predecessor, Secure Socket Layer (SSL), are protocols used to authenticate and encrypt connections and verify the integrity of data exchanged between clients and servers.

- Nearly 47% of the targets had issues related to TLS / SSL.
- The majority of these (more than 38%) had broken ciphers (TLS 1.0, RC4) in the allowed cipher list.
- This data is quite worrying, as the integrity and privacy of the data transmitted between the user client and the server is at risk.


*Read the full report for more information on every type of vulnerability mentioned above, as well as more information on suggested ways to fix such issues.*

## HOW TO ENSURE YOUR WEB APPLICATIONS ARE SECURE?

To ensure your web applications are secure, **request a demo** of the Acunetix web application security testing solution today!