Computest Security

State of Application Security 2024

Management Summary

State of Application Security 2024 >>

Background

Computest Security performs several hundred security tests every year on all types of applications. Our specialists have in-depth information and unique insights into the security of applications in the Netherlands and abroad. By analyzing this information and sharing it in our report 'The State of Application Security 2024' we want to create awareness and demonstrate the urgency to actively contribute to strengthening societal security by means of enhancing applications security.

Research rationale

Applications fulfil a crucial economic and social function. For many organizations, applications support business-critical processes and therefore also generate revenue. This includes web applications that are an important source of information and the starting point for new customer relationships, but also e-commerce platforms that directly generate revenue. These applications are supported by (often privacy-sensitive) data stored in linked databases.

In addition, there are also a large number of web applications from organizations that fulfil a public role. These are aimed at communication, direct interaction, and provide services to citizens. Furthermore, there is a large and growing diversity of SaaS and other (customized) applications such as CRM or ERP platforms that support business processes and methods.

Applications are interesting targets for online criminals for various reasons and are therefore part of the attack surface. For example, malicious parties try to steal money or data via applications. Applications often contain a lot of privacy-sensitive data, which is why they are also misused for extortion and identity fraud. Furthermore, applications can be used to gain access to other systems of the organization. With all the consequences and (reputational) damage this may entail.

Recent research by the British National Cyber Security Centre (NCSC UK) makes it clear that web applications remain interesting targets, especially as more attention is paid to the security of the cloud and office environments. According to the NCSC, attackers are increasingly focusing on finding unknown vulnerabilities in applications, but they also make use of known vulnerabilities. In short, application insecurity poses a major risk, deserves attention, and must be addressed accordingly.

Our research methodology in brief

The basis for this research is the slightly more than 300 application security tests that we carried out during 2023 and in the first quarter of 2024. All these tests have been carried out for various organizations that have requested this for various reasons. For example, as part of their test cycle, because stakeholders or customers required this, or based on (security) standards or regulations. Although these tests took place on different types of applications, they were tested making use of the same method, based on comparable test elements. This means that the different tests can be compared with each other. The findings that were included for purposes of data analytics, were used on anonymized basis. Below we provide insight into the most important results.

Please note the following:

- The data included in this analysis stems from applications of organizations that are already investing in improving the security of their applications. The results therefore do not necessarily reflect the average level of security. We expect that the actual level of application security in 2024 is actually lower.
- For a technical substantiation of these findings, we would like to refer to our technical write-up.
- Furthermore, in the appendix, we have added a comparison between the top 10 vulnerabilities we identified and the most recent top 10 of the 'Open Web Application Security Project' (OWASP), the widely used global vulnerabilities index.

The State of Application Security 2024: Key Findings

Based on the 300 application security tests that were included in our research several key findings stand out. These are presented below.

> "More than 30% of applications contain vulnerabilities that need to be resolved immediately"

1. One-third of the tested applications are insecure

When our specialists perform a security test on an application, they discover an average of twelve vulnerabilities. Divided according to the severity of the vulnerability, using the CVSS scoring methodology, it appears that important ('High') or even critical ('Critical') vulnerabilities are found in almost one-third (31%) of the tests carried out. More specifically, these are vulnerabilities with an industry acclaimed ('CVSS') score of 7.0 or higher. In practice, critical vulnerabilities should always be resolved immediately because of the possible harmful consequences. Major vulnerabilities must be resolved as quickly as possible.

2. Many applications are vulnerable to unauthorized access or even full takeover

As many as 29% of the applications tested contain vulnerabilities in the authorization mechanism. This means that it is not properly checked whether the logged-in user has the right to call the requested functionality. In almost 11% of the tests, it turned out to be possible to perform administrator tasks from a normal user. A possible consequence of this is that the attacker can take over the entire (web) application. If we look further at the authentication of the applications, relevant vulnerabilities were found in 34% of the tests. If we zoom in on this, we find that 19% of applications do not use measures such as multi-factor authentication (MFA) or have not implemented this correctly. This makes it easier for an attacker to gain access via stolen credentials.

3. Malicious code injection poses a risk not to be underestimated

Although it is one of the most notorious web application vulnerabilities, we found cross-site scripting (XSS) in 32% of the performed security tests. This allows the attacker to inject malicious pieces of code into the application. This code is executed when someone

visits the (web) application. The attacker can steal sensitive information, influence the application, or unknowingly redirect users to an untrustworthy website. In 59% of these cases, it is even possible to exploit the vulnerability without a registered account for the application. This is surprising because modern web frameworks provide the building blocks to mitigate cross-site scripting (XSS) vulnerability. We see that this vulnerability arises because available measures are not applied correctly, when standard frameworks are not used or when custom code is developed as an extension of the framework.

> "69% of applications running on outdated and therefore vulnerable software, of which 39% is software that is no longer supported"

4. Outdated software and lack of updates are the biggest cause for insecurity

Our research also shows that the use of third-party components still poses a risk for many organizations. For example, our security researchers found vulnerable components in 69% of the tests. Also, in many cases, security updates for such components are not installed. In 39% of the tests, third-party software was even found that is no longer supported by the developer at all. It is safe to say that no new security updates should be expected in that case.

The State of Application Security 2024: Key recommendations

Based on our analysis and the most important vulnerabilities identified, we present our recommendations below. On the one hand, aimed at limiting vulnerabilities in the (further) development of applications based on prevention, and on the other hand, aimed at limiting (new) vulnerabilities by means of raising awareness of the security of applications.

1. From insecure by design to fundamentally safe(r) applications and code

To lay a secure foundation for a new application, it is important to work according to the Secure by Design principle. This sounds simple, but in practice, we often see that principles of secure design are unknown or not implemented. Precisely by embracing these principles, the organization ensures that developers have the knowledge to develop an application safely and that they work based on agreements made for the safe development of the application. To achieve the safest possible starting situation,

it is essential to make the code base of the application in particular as safe as possible. This means that the code is analyzed line by line for possible vulnerabilities. This can help prevent by directly averting flaws or by installing measures that will help circumvent insecurities, for example through errors in the logic of the processes.

2. Testing to verify security and eliminate new vulnerabilities

Another way to prevent and find vulnerabilities is through automated and non-automated penetration testing. These tests can be performed before an application goes live. While performing these tests, ethical hackers can detect vulnerabilities that can be abused to, for example, obtain sensitive data, disrupt the application, or redirect users to an untrustworthy website. But these tests can also be performed when the application is already live. Periodic 'manual' and continuous automated testing ensures that recently published vulnerabilities in software are detected so that the impact is minimized.

> "Retesting results in 100% of the cases that all critical vulnerabilities are resolved"

3. Retesting is an essential step towards a safe application

Our figures show that when a retest is performed on a previously tested application, on average more than 47% of the vulnerabilities have been resolved and (so far) even all critical vulnerabilities have been resolved. Following recommendations and objectively determining resolved vulnerabilities through a targeted retest therefore contributes to strengthening security.

Conclusion

Our research makes insightful that even when on average more attention is paid to securing applications, by means of testing, vulnerabilities still persist. Varying from quite obvious flaws to more complex code or authorization related issues, in quite a lot of cases of ‘high’ or even ‘critical’ nature. Apart from underlining the obvious function of security testing to unravel these flaws and raise the overall application security, our findings also make clear that there is still a world to be gained for the security of applications that have never been tested and for which it is not clear whether they have been developed securely. What our recommendations illustrate is that there are measures and controls at hand to limit vulnerabilities beforehand. Overall applications are, and should be considered as, part of our attack surface. Bolstering application security should therefore also be deemed as an integral part of raising our security posture. Both on the level of individual organizations and society as a whole.

To keep track of how application security is developing in general, we will repeat our research in 2025 and provide insight into the state of application security. This is to help to strengthen the resilience of organizations against security threats.

Appendix: Top 10 Vulnerabilities Computest Security vs. Top 10 OWASP

As you may know, there is a global initiative called the 'Open Web Application Security Project' (OWASP). This is a network of specialists involved in promoting application security. The 'Top 10' of vulnerabilities that OWASP periodically publishes is a household name. The list is created by researching incidents that have occurred, examining the results of pen tests carried out, and sending a questionnaire to the partners and specialists involved. For that reason, we also use this measuring stick, which is well-known among specialists, for the tests we perform for our clients. Based on these research results, we can make a comparison between 'The State of Application Security' and the 'Top 10' published by OWASP (this is published every four years).

The ranking of the vulnerabilities in the OWASP list is based, among other things, on the frequency, impact, and how easily the vulnerability can be exploited. It is good to realize that in our research we looked at frequency and impact (the latter based on the CVSS scoring). It is interesting to determine how the results of the Computest Security specialists differ from those of OWASP.

For example, the 'Top 10' of Computest Security has a different order than the OWASP top 10, except for vulnerability number 10. The top two vulnerabilities based on the research make it clear that missing or sub-optimal configurations and failing identification and authentication mechanisms constitute the most important risks.

#	OWASP TOP 10 2021	COMPUTEST SECURITY TOP 10 2024	Difference
1	A01: Broken Access Control	A05: Security Misconfiguration	+ 4
2	A02: Cryptographic Failures	A07: Identification and Authentication Failures	+ 5
3	A03: Injection	A02: Cryptographic Failures	- 1
4	A04: Insecure Design	A06: Vulnerable and Outdated Components	+ 2
5	A05: Security Misconfiguration	A08: Software and Data Integrity Failures	+ 3
6	A06: Vulnerable and Outdated Components	A04: Insecure Design	- 2
7	A07: Identification and Authentication Failures	A01: Broken Access Control	- 6
8	A08: Software and Data Integrity Failures	A09: Security Logging and Monitoring Failures	+ 1
9	A09: Security Logging and Monitoring Failures	A03: Injection	- 6
10	A10: Server-Side Request Forgery	A10: Server-Side Request Forgery	=

Computest Security

Any questions about the State of Application Security 2024?

Contact us:

info@computest.nl

+31(0)88 733 1337

Independent. Security. Partner. >>