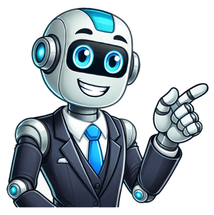
Continue



The OWASP Top Ten is a list of the top ten most critical web application security risks. The 2017 edition of the OWASP Top Ten includes vulnerabilities such as injection, broken authentication, sensitive data exposure, XML external entities (XXE), broken access control, security misconfiguration, cross-site scripting (XSS), insecure deserialization, using components with known vulnerabilities, and insufficient logging and monitoring. These risks pose significant threats to web applications and can lead to unauthorized access, data breaches, and other security incidents. It is important for organizations to be aware of these risks and implement appropriate security measures to protect their applications.The OWASP Top Ten 2017 is a list of web application security risks that organizations need to be aware of. These risks include injection, broken authentication, sensitive data exposure, XML external entities (XXE), broken access control, security misconfiguration, cross-site scripting (XSS), insecure deserialization, using components with known vulnerabilities, and insufficient logging and monitoring. Each of these risks poses a significant threat to the security of web applications and can lead to unauthorized access, data breaches, and other security incidents. It is important for organizations to understand these risks and implement appropriate security measures to protect their applications.The 2017's Top 10 risks identified by OWASP highlight the most critical vulnerabilities in web applications. These risks have the potential to cause significant damage to organizations, including unauthorized access to sensitive data, financial loss, reputational damage, and legal consequences. For example, injection flaws can allow attackers to execute malicious commands or access data without proper authorization. Broken authentication can lead to compromised passwords and unauthorized access to user accounts. Sensitive data exposure can result in financial fraud, identity theft, and other crimes. It is crucial for organizations to address these risks and implement robust security measures to protect their applications and users' data.The identification of the OWASP Top Ten 2017 risks has significant implications for regulatory compliance. Many industry-specific data protection regulations require organizations to implement appropriate security measures to protect sensitive data. Failure to address the top ten risks can lead to non-compliance with these regulations and potential legal consequences. Organizations need to ensure they have adequate controls in place to mitigate these risks and meet regulatory requirements. This includes implementing secure coding practices, strong authentication mechanisms, encryption of sensitive data, access control mechanisms, and robust logging and monitoring systems.As technology evolves, new vulnerabilities and risks will continue to emerge. Organizations need to stay updated with the latest security trends and adapt their security measures accordingly. The future outlook for web application security involves a stronger focus on proactive security measures, such as secure coding practices, threat modeling, continuous security testing, and security awareness training for developers. Additionally, there will likely be an increased emphasis on regulatory compliance and data protection, as more stringent regulations are introduced to address the growing threat landscape. Organizations must prioritize security and invest in effective security strategies to safeguard their web applications and protect sensitive data.eWEEK content and product recommendations are editorially independent. We may make money when you click on links to our partners. Learn More.Clearly, both of these solutions, IBM QRadar and Splunk, address a growing market demand for cybersecurity. There is no shortage of challenges facing cybersecurity teams: an increase in the volume and sophistication of cyberattacks, an explosion of data, an expanding attack surface, disjointed security tools and a shortage of skilled security staff.Both QRadar and Splunk are leaders in the Security Information and Event Management (SIEM) space. Both offer broad monitoring and analytics of security incidents, potential threats, and analysis of logs.Buyers looking for a general SIEM platform are likely to find both on their list of strong candidates. Overall, though, there are plenty of differences that will matter greatly to buyers with different goals in mind. Heres a look at both SIEM tools, and how they compare.Also see: Secure Access Service Edge: Big Benefits, Big ChallengesQRadar vs. Splunk: Key Feature Comparison The Splunk platform encompasses searching, monitoring, and analyzing of a vast amount of IT data to identify data patterns, provide metrics, diagnose problems and aid in business and IT decision making.To understand the scope of Splunk: SIEM can be considered just one small part of its feature arsenal. Beyond security, it takes in Application Performance Monitoring (APM), compliance, automation, orchestration, forensics, as well as plenty of features related to IT service management (ITSM) and IT operations management (ITOM).Splunks wide range of products and features are aggregated within the Splunk Observability Suite. The platform can be used to analyze, ingest, and store data for later use, as well as detect issues impacting customers. Overall, it offers a breadth of management. Those wishing to manage SIEM, ITOM and ITSM in an integrated fashion will find Splunk to be a fine tool to do the job. It offers a wealth of real-time visualization and analysis features, as well as management and monitoring.QRadar is a SIEM solution that defends against threats while scaling security operations through integrated visibility, detection, investigation, and response. It provides security teams with centralized visibility into enterprise-wide security data and actionable insights into the highest priority threats.Security analysts can work from one pane of glass in QRadar to quickly understand their security posture, identify the most critical threats, and drill down to get more details, helping to streamline workflows and eliminate the need to pivot between tools. Its anomaly detection capability helps to reduce events to a prioritized list of the most important alerts. It leverages automated, advanced analytics and threat intelligence to speed investigation time.Splunk represents itself as a complete platform to handle everything related to SIEM, security and ITOM. It ventures far beyond SIEM. QRadar is more tightly focused on SIEM and overall security. Your existing stack of security and management tools, therefore, should be considered before deciding between Splunk and IBM. Those with outdated tools that are in need of a complete overhaul should probably gravitate to Splunk due to its much wider feature set. Why buy five different management tools when you can buy one from Splunk and have them all integrated?But where Splunk goes wider, IBM goes deeper on the security side. As it is built on IBM Cloud Pak for Security, the open architecture of QRadar provides a great many additional and fully integrated security capabilities that save time enriching, correlating, and investigating threats. Artificial intelligence, pre-built playbooks, automatic root-cause analysis, and MITRE ATT&CK mapping are all part of the package. This can help to greatly improve the speed of investigation.On security features, IBM wins.QRadar vs. Splunk: Comparing Implementation and Ease of Use One potential challenge with QRadar is the size and scope of IBM. There are so many tools and capabilities available within the vast scope of IBM that sometimes products get lost. That said, IBM is investing a lot in QRadar so it appears it wont suffer the same fate of other lesser IBM tools.On implementation, a large collection of templates make the job of implementing the platform straightforward, relative to the typical SIEM deployment. Thus, users tend to report a shorter learning curve on QRadar than Splunk.As for ease of use, Splunk gets the nod. Some users consider the UI of QRadar a little clunky and dated. Splunk, being a newer platform, looks more modern.Splunk wins on ease of use; IBM on ease of implementation.QRadar vs. Splunk: Comparing Cloud and On-PremisesSplunk was born and raised in the cloud. It does not offer on-premises appliances but provides software for on-site deployment if desired. But most use it in the cloud.IBM has gone to great lengths over the past decade to shed its old school on-premises reputation. Its CloudPak initiative has QRadar available either in the cloud or on-premises. That said, Splunk still wins in the cloud and QRadar wins for on-premises. Splunk can be installed directly through the cloud onto a public, private, or hybrid cloud setting. IBM, too, can provide cloud-based SIEM.QRadar vs. Splunk: Integration ComparisonA big strength of Splunk and a key differentiator is its ability to integrate data streams from a huge number of sources. Some users ingest several PB per day. It supports a wide range of data formats like.xml, .csv and .json file. Those with needs that require such data stream integration from multiple data formats should opt for Splunk, as it offers over 1,000 add-on applications in its app store. It also heads a coalition of 30 partners on security collaboration.QRadar integrates very well with a great many IBM products and especially with the many security tools that fall under the QRadar umbrella. A large, open ecosystem integrates EDR, SIEM, NDR, security orchestration and response (SOAR) and threat intelligence solutions. But integrations beyond the IBM world are limited.Splunk wins on integration.Also see: Best Website ScannersQRadar vs. Splunk: Comparing Analytics and SearchSplunk is all about monitoring and analyzing data generated from various machines. It is great for analyzing the huge number of log files generated by enterprise systems. Splunk eliminates the need for IT to spend hours trawling through all the logs looking for that performance needle in the IT haystack. It makes use of the search processing language to find terms present in log files. For example, Splunk offers a wealth of real-time visualization and analysis features. If real-time management and monitoring are vital, then this one is a no contest. But it does come at a price.QRadar, however, benefits from IBMs long-term leadership in Artificial Intelligence this is a major advantage. It can tap into IBM Watson and other IBM analytic capabilities for threat identification and analysis. This also adds a greater level of automation to SIEM.IBM wins on analytics.How Do QRadar and Splunk Prices Compare?Neither Splunk nor QRadar come cheap. The various modules within Splunk have a reputation for being expensive. Further, upselling can send the budget much higher. If you need performance monitoring that adds in an APM module, and slowly other modules creep in and the price tag rises. This is normal enough in IT. But when you are already dealing with a pricey platform, it is important to determine what you really need and what you can dispense with.QRadar is also expensive. Perpetual licenses are available with general licensing done based on the number of events and flows received in the event collector. Those who are already partners or significant users of IBM products and services benefit from considerable package discounts.Splunk prefers to price based on the maximum daily data volume. Thus, the most economic platform will vary from enterprise to enterprise based on how the workloads run and performance/data patterns.Also see: Secure Access Service Edge: Big Benefits, Big ChallengesShould You Choose QRadar or Splunk?Splunk and QRadar are both excellent tools designed to solve a great many challenges related to security and performance monitoring. You cant go wrong too far wrong with either one. Both are strong in SIEM. User ratings overall from a variety of IT review sites show little difference in rating between Splunk and QRadar. Both are regarded as leaders in the latest Gartner SIEM Magic Quadrant.Splunk is a much broader platform and toolset that proves invaluable in rapidly analyzing log files and making sense of mountains of data so IT knows what is going on, and it encompasses a far wider range than just security. Whether its a performance slowdown or a security incursion, Splunk is a good way to stay one step ahead of trouble. QRadar can rival Splunk on many features directly related to SIEM, but it provides a much deeper set of integrated security tools.In the end it comes down to needs. Those wanting an all-encompassing security and IT management platform will find Splunk closer to their needs. Additionally, those with aging applications that are ready for a major management makeover will find Splunk a good fit. It covers a large amount of ground.But if it is only SIEM that is needed, the equation shifts. QRadar wins on many fronts, and offers a great many other security bells and whistles, too. And those invested in the IBM universe should likely not look beyond QRadar. Visit website Log360 is a SIEM solution that helps combat threats on premises, in the cloud, or in a hybrid environment. It also helps organizations adhere to several compliance mandates. You can customize the solution to cater to your unique use cases.It offers real-time log collection, analysis, correlation, alerting and archiving abilities. You can monitor activities that occur in your Active Directory, network devices, employee workstations, file servers, Microsoft 365 and more. Try free for 30 days! Learn more about ManageEngine Log360 OWASP Top Ten 2017 (de) OWASP Top 10 Risiken fr die Anwendungssicherheit 2017 A1:2017-Injection: Injection-Schwachstellen, wie beispielsweise SQL-, OS- oder LDAP-Injection, treten auf, wenn nicht vertrauenswrdige Daten von einem Interpreter als Teil eines Kommandos oder einer Abfrage verarbeitet werden. Ein Angreifer kann Eingabedaten dann so manipulieren, dass er nicht vorgesehene Kommandos ausfhren oder unautorisiert auf Daten zugreifen kann. A2:2017-Fehler in der Authentifizierung: Anwendungsfunktionen, die im Zusammenhang mit Authentifizierung und Session-Management stehen, werden hufig fehlerhaft implementiert. Dies erlaubt es Angreifern, Passwrter oder Session-Token zu kompromittieren oder die entsprechenden Schwachstellen so auszunutzen, dass sie die Identitt anderer Benutzer vorbergehend oder dauerhaft annehmen knnen. A3:2017-Verlust der Vertraulichkeit sensibler Daten: Viele Anwendungen schtzen sensible Daten, wie personenbezogene Informationen und Finanz- oder Gesundheitsdaten, nicht ausreichend. Angreifer knnen diese Daten auslesen oder modifizieren und mit ihnen weitere Straftaten begehen (Kreditkartenbetrug, Identittsdiebstahl etc.). Vertrauliche Daten knnen kompromittiert werden, wenn sie nicht durch Manahmen, wie Verschlsselung gespeicherter Daten und verschlsselte Datenbertragung, zustzlich geschtzt werden. Besondere Vorsicht ist beim Datenaustausch mit Browsern angeraten. A4:2017-XML External Entities (XXE): Viele veraltete oder schlecht konfigurierte XML Prozessoren bercksichtigen Referenzen auf externe Entitten innerhalb von XML-Dokumenten. Dadurch knnen solche externen Entitten dazu eingesetzt werden, um mittels URI Datei-Handlern interne Dateien oder File-Shares offen-zulegen oder interne Port-Scans, Remote-Code-Executions oder Denial-of-Service Angriffe auszufhren. A5:2017-Fehler in der Zugriffskontrolle: Hufig werden die Zugriffsrechte fr authentifizierte Nutzer nicht korrekt um- bzw. durchgesetzt. Angreifer knnen entsprechende Schwachstellen ausnutzen, um auf Funktionen oder Daten zuzugreifen, fr die sie keine Zugriffsberechtigung haben. Dies kann Zugriffe auf Accounts anderer Nutzer sowie auf vertrauliche Daten oder aber die Manipulation von Nutzerdaten, Zugriffsrechten etc. zur Folge haben. A6:2017-Sicherheitsrelevante Fehlkonfiguration: Fehlkonfigurationen von Sicherheitseinstellungen sind das am hufigsten auftretende Problem. Ursachen sind unsichere Standardkonfigurationen, unvollstndige oder ad-hoc durchgefhrte Konfigurationen, ungeschtzte Cloud-Speicher, fehlkonfigurierte HTTP-Header und Fehleraus-gaben, die vertrauliche Daten enthalten. Betriebssysteme, Frameworks, Bibliotheken und Anwen-dungen mssen sicher konfiguriert werden und zeitnah Patches und Updates erhalten. A7:2017-Cross-Site Scripting (XSS): XSS tritt auf, wenn Anwendungen nicht vertrauenswrdige Daten entgegennehmen und ohne Validierung oder Umkodierung an einen Webbrowser senden. XSS tritt auch auf, wenn eine Anwendung HTML- oder JavaScript-Code auf Basis von Nutzereingaben erzeugt. XSS erlaubt es einem Angreifer, Scriptcode im Browser eines Opfers auszufhren und so Benutzersitzungen zu bernehmen, Seiteninhalte verndert anzuzeigen oder den Benutzer auf bsartige Seiten umzuleiten. A8:2017-Unsichere Deserialisierung: Unsichere, weil unzureichend geprfte Deserialisierungen knnen zu Remote-Code-Execution-Schwachstellen fhren. Aber auch wenn das nicht der Fall ist, knnen Deserialisierungsfehler Angriffsmuster wie Replay-Angriffe, Injections und Erschleichung erweiterter Zugriffsrechte ermglichen. A9:2017-Nutzung von Komponenten mit bekannten Schwachstellen: Komponenten wie Bibliotheken, Frameworks etc. werden mit den Berechtigungen der zugehri-gen Anwendung ausgefhrt. Wird eine verwundbare Komponente ausgenutzt, kann ein solcher Angriff von Datenverlusten bis hin zu einer bernahme des Systems fhren. Applikationen und APIs, die Komponenten mit bekannten Schwachstellen einsetzen, knnen Schutzmanahmen unterlaufen und so Angriffe mit schwerwiegenden Auswirkungen verursachen. A10:2017-Unzureichendes Logging & Monitoring: Unzureichendes Logging und Monitoring fhrt zusammen mit fehlender oder uneffektiver Reaktion auf Vorflle zu andauernden oder wiederholten Angriffen. Auch knnen Angreifer dadurch in Netzwerken weiter vordringen und Daten entwenden, verndern oder zerstren. Viele Studien zeigen, dass die Zeit bis zur Aufdeckung eines Angriffs bei ca. 200 Tagen liegt sowie typischer-weise durch Dritte entdeckt wird und nicht durch interne berwachungs- und Kontrollmanahmen. OWASP Top Ten 2017 OWASP Top 10 Application Security Risks - 2017 Injection flaws, such as SQL, NoSQL, OS, and LDAP injection, occur when untrusted data is sent to an interpreter as part of a command or query. The attackers hostile data can trick the interpreter into executing unintended commands or accessing data without proper authorization. A2:2017-Broken Authentication Application functions related to authentication and session management are often implemented incorrectly, allowing attackers to compromise passwords, keys, or session tokens, or to exploit other implementation flaws to assume other users identities temporarily or permanently. A3:2017-Sensitive Data Exposure Many web applications and APIs do not properly protect sensitive data, such as financial, healthcare, and PII. Attackers may steal or modify such weakly protected data to conduct credit card fraud, identity theft, or other crimes. Sensitive data may be compromised without extra protection, such as encryption at rest or in transit, and requires special precautions when exchanged with the browser. A4:2017-XML External Entities (XXE) Many older or poorly configured XML processors evaluate external entity references within XML documents. External entities can be used to disclose internal files using the file URI handler, internal file shares, internal port scanning, remote code execution, and denial of service attacks. A5:2017-Broken Access Control Restrictions on what authenticated users are allowed to do are often not properly enforced. Attackers can exploit these flaws to access unauthorized functionality and/or data, such as access other users accounts, view sensitive files, modify other users data, change access rights, etc. A6:2017-Security Misconfiguration Security misconfiguration is the most commonly seen issue. This is commonly a result of insecure default configurations, incomplete or ad hoc configurations, open cloud storage, misconfigured HTTP headers, and verbose error messages containing sensitive information. Not only must all operating systems, frameworks, libraries, and applications be securely configured, but they must be patched/upgraded in a timely fashion. A7:2017-Cross-Site Scripting (XSS) XSS flaws occur whenever an application includes untrusted data in a new web page without proper validation or escaping, or updates an existing web page with user-supplied data using a browser API that can create HTML or JavaScript. XSS allows attackers to execute scripts in the victims browser which can hijack user sessions, deface web sites, or redirect the user to malicious sites. A8:2017-Insecure Deserialization Insecure deserialization often leads to remote code execution. Even if deserialization flaws do not result in remote code execution, they can be used to perform attacks, including replay attacks, injection attacks, and privilege escalation attacks. A9:2017-Using Components with Known Vulnerabilities Components, such as libraries, frameworks, and other software modules, run with the same privileges as the application. If a vulnerable component is exploited, such an attack can facilitate serious data loss or server takeover. Applications and APIs using components with known vulnerabilities may undermine application defenses and enable various attacks and impacts. A10:2017-Insufficient Logging & Monitoring Insufficient logging and monitoring, coupled with missing or ineffective integration with incident response, allows attackers to further attack systems, maintain persistence, pivot to more systems, and tamper, extract, or destroy data. Most breach studies show time to detect a breach is over 200 days, typically detected by external parties rather than internal processes or monitoring. After a difficult gestation, the OWASP Top 10 Final is out.You can get it from here: As many of you know, there was a lot of passion within the application security community about the OWASP Top 10 2017 RC1, so it was critical that we worked with the community to firm up the data and obtain a consensus view of how to proceed. After the change of leadership from Dave Wichers and Jeff Williams to Andrew van der Stock in late May 2017, we added diversity to the leadership team, by adding Neil Smithline, Torsten Gigler, and Brian Glas. Each of the leaders brings their own experience and point of view to the OWASP Top 10, making it far stronger. I couldn't have done this by myself, and it would have been a far weaker document if it was just little old me. I thank my co-leaders from the bottom of my heart. I also thank the founding leadership of Dave Wichers and Jeff Williams for creating the OWASP Top 10, and trusting in us to get this done. In June, Dave Wichers and Brian Glas attended the OWASP Project Summit in London, and I participated remotely. During the summit, as a community, we agreed to governance, methodology, data analysis and transparency improvements. The highlights of this are:A diversity of leadership at all times (at least two unrelated leaders). This has been an incredible win for the OWASP Top 10, and I hope more OWASP Flagship projects consider doing it.The methodology was improved by confirming that we will be using risks, rather than any other metric, and agreeing to up to two items will be selected by the community for up and coming risksData analysis performed by Brian Glas, in particular how to improve the balance from largely automated findings that swamp manual findings, as well as re-opening the data call to obtain 2016 data and survey the community for the two forward looking itemsTransparency is now aligned with OWASP's values - we work in the open at GitHub, and folks can see who suggested an improvement or issue, and how this was resolved in the text. For the first time, there is a strong traceability between the data submitted by participating data contributors and the OWASP Top 10. This means that if you want, you can fork the OWASP Top 10, re-analyze the data to suit your needs and create your own version. (Just don't call it the OWASP Top 10 :-)) The data call was very successful. We obtained a great deal of new data covering previous years, including 2016, from a wide variety of consultancies and vendors. We have data from over 40 data contributors, 23 of which were used in the final data analysis. From those 23 data sets, the data covered over 114,000 applications, which is one of the biggest data sets on application security anywhere. And you can download it from our GitHub repo. At the last minute, we also received data from BugCrowd. The interesting thing about bug bounty programs is that kudos and payouts only occur when fully validated, and it also shows what is on the top of the list from the point of view of bug bounty programs. The bug bounty data backed up our analysis in terms of prevalence data, so we were definitely on the right track. The survey was wildly successful. We received over 500 survey responses, so I think we can safely claim consensus on the two new items - Insecure Deserialization and Insufficient Logging and Monitoring. These two items were obviously top of mind for many this year considering the era of the mega breach is not slowing down. We discuss our methodology in more detail within the OWASP Top 10 - 2017 itself, as many will wonder why we didn't use the two top items directly. The short answer - and this should be no surprise - some of these other issues were already in the OWASP Top 10 due to prevalence data, such as XXE and access control.I will address some of the frequently asked questions - why have CSRF and unvalidated redirects and forwards been removed? It's time to move on. The data for these is no longer strong enough to warrant inclusion, especially when we only have 8 data supported spots with our new methodology, and these two items didn't rank in the community survey. This is actually a sign of success; the fact that CSRF is finally going away is a sign that the OWASP Top 10 has been successful at its mission. Back when I included CSRF in 2007 as a forward looking item, there was no data for it. At all. But ~ 100% of applications had CSRF at that time. Now it's less than 5% of all applications. If you use a modern framework, you're pretty much covered without doing anything. That's a huge success. This then leads into the discussion about renumbering. We risk rated the resulting list over about a 5 hour meeting, and this is the result. I asked the Twitter community if they wanted a risk based order, a likelihood order, an impact order, or the order from previous OWASP Top 10's. Overwhelmingly risk based order won. Interestingly, the previous OWASP Top 10's kept the previous order, but this was wanted by less than 10% of respondents, compared to over 55% for risk based ordering. So that's what happened. What surprised me is that after re-risk rating many of the existing items didn't move. I was actually surprised by this, particularly in relation to SQL injection, but because we include all forms of injection (which theoretically can cover XSS), it remained at the A1:2017 position. This is because we couple three forms of likelihood (prevalence, detectability, and exploitability) and impact. We have strong prevalence data, but the others were our best judgement. You can look at what we decided upon and review our work. I encourage everyone to do so. The last common discussion we've had is why we didn't roll up XSS into injections, because it's either HTTP, HTML, or JavaScript injection. The reality is that it would have swamped the important discussion on other injections, and the solutions for XSS are significantly different to preventing OS command injection or SQL injection. I will defend this decision until the day we see XSS gone the way of CSRF. And I can't see that day ... yet. There is hope in the form of CSP and XSS-resistant frameworks such as Ruby on Rails 3 and React, but there's a lot of code out there that is still vulnerable. The new or heavily updated risks need little explanation:We cover API as well as web apps throughout the entire Top 10. This covers mobile, single page apps, RESTful API and traditional web apps.A3:2017 Sensitive Data Exposure is now firmly about privacy and PII breaches, and not stack traces or headers.A4:2017 XXE is a new data supported item, and so tools and testers need to learn how to find and test for XXE, and developers and devops need to understand how to fix it.A6:2017 Misconfiguration now encompasses cloud security issues, such as open buckets.A8:2017 Deserialization is a critical issue, asked for by the community. It's time to learn how to find this in tools, and for testers to understand what Java and PHP (and other serialization) looks like so it can be fixed.A10:2017 Insufficient Logging and Monitoring. Many folks think this is a missing control, rather than a weakness, but as it was selected by the community, and whilst organizations still take over half a year to detect a breach - usually from external notification - we have to fix this. The way to go forward here for testers is to ask the organization if they detected whatever activity was undertaken, and if they would have responded to it without being prompted. Obviously, we are looking for testing to be undertaken through security devices, but whitelisted, so that logging, escalation and incident response can also be assessed. These new items are modern era issues, and I hope that in the next three years, the industry can make headway on them. So after more than 370 closed issues and 650 commits, we are finally finished. We received a lot of feedback from the community, and we thank those who reviewed and QA'd the document extremely closely, such as Osama Elnaggar, Dirk Wetter and Jim Manico, as well as over 40 others. For a full list of reviewers, please see the acknowledgement page. What is the future of the OWASP Top 10? I think if anything, the community's passion during this time around shows how important the OWASP Top 10 is. It is widely adopted and a lot of folks care about it very deeply. It was a time for us to listen and learn from the process, and that will result in improvements for the OWASP Top 10 - 2020. We will be starting the data collection process much earlier, and we will improve our methodology particularly in relation the survey to provide more choices (we only had 25 CWEs). On top of that, we need to work with NIST / MITRE to keep CWE up to date, because some of the biggest up and coming (and to be fair, some of the existing) weaknesses do not have a CWE entry. But first, we need a break. Thank you to everyone who participated to make the OWASP Top 10 a much stronger and more evidence based standard. The OWASP Top 10 - 2017 is by far the best sourced, most reviewed, application security standard out there. I encourage everyone to download it and start cracking on the new and updated items. We need translations as well, so if you want to do that, please contact us at @owasptop10 on Twitter or via GitHub.

Owasp top 10 web application security. Owasp top 10 web application security risks. Owasp top 10 web app security risks. Owasp top ten web application security risks. Owasp top 10 application security risk.