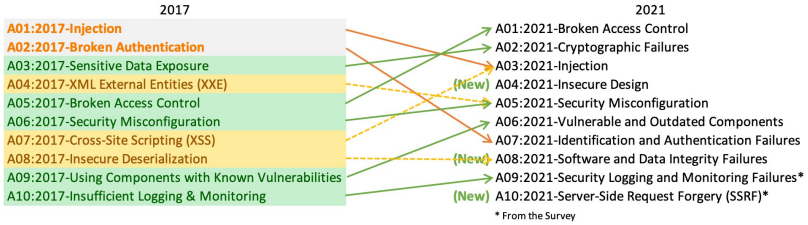
OWASP Top 10 2021 介紹

歡迎來到最新版本的 OWASP Top 10！！OWASP Top 10 2021 是一個全新的名單，包含了你可以列印下來的新圖示說明，若有需要的話，你可以從我們的網頁上面下載。

在此我們想對所有貢獻了他們時間和資料的人給予一個極大的感謝。沒有你們，這一個新版本是不會出現的。謝謝。

Top 10 for 2021 有什麼新的變化？

這次在 OWASP Top 10 for 2021 有三個全新的分類，有四個分類有做名稱和範圍的修正，並有將一些類別做合併。



A01:2021-權限控制失效 從第五名移上來; 94% 被測試的應用程式都有驗測到某種類別權限控制失效的問題。在權限控制失效這個類別中被對應到的 34 個 CWEs 在驗測資料中出現的次數都高於其他的弱點類別。

A02:2021-加密機制失效 提升一名到第二名，在之前為 敏感資料外曝，在此定義下比較類似於一個廣泛的問題而非根本原因。在此重新定義並將問題核心定義在加密機制的失敗，並因此造成敏感性資料外洩或是系統被破壞。

A03:2021-注入式攻擊 下滑到第三名。94% 被測試的應用程式都有驗測到某種類別注入式攻擊的問題。在注入式攻擊這個類別中被對應到的 33 個 CWEs 在驗測資料中出現的次數為弱點問題的第二高。跨站腳本攻擊現在在新版本屬於這個類別。

A04:2021-不安全設計 這是 2021 年版本的新類別，並特別針注在與設計相關的缺失。如果我們真的希望讓整個產業"向左移動"＊註一＊，那我們必須進一步的往威脅建模，安全設計模塊的觀念，和安全參考架構前進。

＊註一: Move Left 於英文原文中代表在軟體開發及交付過程中，在早期找出及處理相關問題，同 Shift Left Testing。＊

A05:2021-安全設定缺陷 從上一版本的第六名移動上來。90% 被測試的應用程式都有驗測到某種類別的安全設定缺陷。在更多的軟體往更高度和有彈性的設定移動，我們並不意外這個類別的問題往上移動。在前版本中的 XML 外部實體注入攻擊（XML External Entities）現在屬於這個類別。

A06:2021-危險或過舊的元件 在之前標題為 使用有已知弱點的元件。在本次版本中於業界問卷中排名第二，但也有足夠的統計資料讓它可以進入 Top 10。這個類別從 2017 版本的第九名爬升到第六，也是我們持續掙扎做測試和評估風險的類別。這也是唯一一個沒有任何 CVE 能被對應到 CWE 內的類別，所以預設的威脅及影響權重在這類別的分數上被預設為 5.0。

A07:2021-認證及驗證機制失效 在之前標題為 錯誤的認證機制。在本次版本中油第二名下滑至此，並同時包含了將認證相關缺失的 CWE 包含在內。這個類別仍是 Top 10 不可缺少的一環，但同時也有發現現在標準化的架構有協助降低次風險發生機率。

A08:2021-軟體及資料完整性失效 這是 2021 年版本全新的類別，並在軟體更新，機敏及重要資料，和 CI/CD 管道中並沒有做完整性的確認為前提做假設並進行評估。在評估中影響權重最高分的 CVE/CVSS 資料都與這類別中的 10 個 CWE 對應到。2017 年版本中不安全的反序列化現在被合併至此類別。

A09:2021-資安記錄及監控失效 在之前為不完整的紀錄及監控並納入在業界問卷中在本次列名為第三名並從之前的第十名上移。這個類別將擴充去納入更多相關的缺失，但這也是相當難去驗證，並沒有相當多的 CVE/CVSS 資料可以佐證。但是在這個類別中的缺失會直接影響到整體安全的可視性，事件告警及鑑識。

A10:2021-伺服端請求偽造 這個類別是在業界問卷排名第一名，並在此版本內納入。由資料顯示此問題有較低被驗測次數和範圍，但有高於平均的威脅及影響權重比率。這個類別的出現也是因為業界專家重複申明這類別的問題相當重要，即使在本次資料中並沒有足夠的資料去顯示這個問題。

分析方法

本次 Top 10 的選擇方式比以往更重視資料分析，但並不是完全以資料分析為主。我們從資料分析中挑選了八個風險類別，然

後由業界問卷中挑選兩個風險類別。我們從過往的分享資料中去瞭解，並有我們一個基本的理由。原因是所有的資安研究人員都不斷的在找新的弱點並找出方法去驗證弱點，但會需要時間才能將這些驗測方法納入到既有的工具和測試流程中。當我們能有效的大量測試這個弱點時，有可能已經過了多年的時間。為了要讓兩者之間有平衡，我們使用業界問卷請教在前線的資安研究專家們並瞭解他們覺得有哪些是他們覺得嚴重但尚未出現在測試資料中的漏洞及問題。

這是幾個我們為了要讓 OWASP Top 10 更加成熟的重要改變。

如何建構風險類別

有別於上一個版本，在這次的 OWASP Top 10 有一些風險類別的修改。我們在此以比較高的角度說明一下這次的類別修改。

在上一次的資料收集當中，我們將資料收集的重心放在預先定義好的約 30 個 CWEs 並納入一個欄位徵求其他的發現。從這裡我們看到決多數的組織都只會專注在這 30 個 CWEs 而不常加入其他他們可能發現的 CWEs。在這次的改版中，我們將所有的問題都以開放式的方法處理，並沒有限制在任何一個 CWEs。我們請教了從 2017 年開始所測試的網頁應用程式數量，然後在這些程式中至少有一個 CWE 被發現的數量。這個格式讓我們能夠追蹤每個 CWE 跟所有被驗測及統計的應用程式的數量跟關係。我們也忽略了 CWE 出現的頻率，雖然在某些狀況下這也許是必須的，但這卻隱藏了風險類別本身與應用程式數量整體的關係。所以一個應用程式有 4 個或是 4,000 個弱點並不是被計算在 Top 10 的基礎。但同時我們也從原本的 30 多個 CWEs 增長到快 400 多個CWEs 去進行分析。我們因此也計畫未來做更多的資料分析，並在對此版本進行補充說明。而這些增加的 CWEs 也同時影響了這次風險類別的規劃。

我們花了好幾個月將 CWEs 進行分組跟分類，而且其實可以一直花更多個月去做這件事情。但我們必須在某一個時間點停住。在 CWEs 當中，同時有 原因 以及 症狀 的問題，而像是 "加密機制失效" 和 "設定問題" 這類型的 原因 與 "機敏資料外洩" 和 "阻斷服務" 這類型的 症狀 是對立的。因此我們決定在可以的時候要更專注於底層的原因，因為這是可以有效指出問題的本體跟同時提供問題的解決方向。專注在問題核心而不將重心放在症狀並不是一個新的概念，Top Ten 有史以來一直是症狀跟問題核心的綜合體，只是這次我們更刻意的將他突顯出來。在這次的新版本中，每一個類別內的平均有 19.6 個 CWE，而最低的 *A10:2021*-伺服端請求偽造 有一個 CWE 到 *A04:2021*-不安全設計 有四十個 CWE。這個新的類別架構能提供企業更多的資安訓練的好處，因為在新的架構下可以更專注在某個語系或平台上的 CWE。

選擇類別時資料的使用方式

在 2017 年，我們用事件發生次數去判斷可能發生的機率去選擇類別，然後透過一群在業界擁有數十年經驗的專家團對討論並依照 可發生性，可發現性（同可能性），和 技術影響力 去做排名。在 2021 年，我們希望如果可以的話用資料證明可發生性和技術影響性。

我們下載了 OWASP Depndency Check 並取出了 CVSS 漏洞，並將相關的 CWE 用影響力分數分群。這花了一些時間和力氣去研究因為所有的 CVEs 都有 CVSSv2 分數，但是在其中因為 CVSSv2 跟 CVSSv3 之間有一些缺失是必須被修正的。經過了一段時間後，所有的 CVEs 都會有對應的 CVSSv3 的分數。再者，分數的範圍和計算的公式在 CVSSv2 和 CVSSv3 之間也做了更新。

在 CVSSv2 中，漏洞和影響力兩者都可達到 10.0 分，但是公式本身會將兩者調整為漏洞佔 60%，然後影響力佔 40%。在 CVSSv3 中，理論上的最高值將漏洞限制在 6.0 分而影響力在 4.0 分。當考慮到權重比率時，影響力的分數會偏高，在 CVSSv3 中幾乎平均會多出 1.5 分，而漏洞分數卻會平均少 0.5 分。

從 OWASP Dependcy Check 翠取出的 NVD 資料當中有將近 12.5 萬筆 CVE 資料有對應到 CWE，而有 241 筆獨特的 CWEs 有對應到 CVE。6.2萬筆 CWE 有對應到 CVSSv3 分數，所以大約是整體資料中一半的部分。

而在 Top Ten，我們計算漏洞和影響力的平均分數的方式如下。我們將所有有 CVSS 分數的 CVE 依照 CWE 分組，然後依照有 CVSSv3 的漏洞和影響力在所有資料中的百分比作權重，在加上資料中有 CVSSv2 的資料去做平均。我們將這些平均後的 CWEs 對應到資料中，然後將他的漏洞和引想力分數使用在另一半的風險公式中。

為什麼就不純粹做統計分析？

這些資料的結果最主要是被限制在能使用自動工具測試出來的結果。可是當你跟一位有經驗的應用程式安全專家聊的時候，他們會跟你說絕大多數他們找到的問題都不在這些資料裡面。原因是一個測試要被自動化的時候，需要花時間去開發這些弱點測試的方法論，當你需要將這個測試自動化並能對大量的應用程式去驗證時，又會花上更多的時間。當我們回頭看去年獲以前有可能沒出現的一些問題的趨勢，我們發現其實都沒有在這些資料當中。

因此，由於資料不完全的關係，我們只有從資料中選出 8 個類別，而並不是 10 個。剩下的兩個類別是從業界問卷中所選出的。這會允許在前線的參與者去選出他們認為的高風險，而不是純粹依據資料去判斷（甚至可能資料永遠都不會有出現的蹤跡）。

為什麼用事故率而不是用發生次數

There are three primary sources of data. We identify them as Human-assisted Tooling (HaT), Tool-assisted Human (TaH), and raw Tooling.

Tooling and HaT are high-frequency finding generators. Tools will look for specific vulnerabilities and tirelessly attempt to find every instance of that vulnerability and will generate high finding counts for some vulnerability types. Look at Cross-Site Scripting, which is typically one of two flavors: it's either a more minor, isolated mistake or a systemic issue. When it's a systemic issue, the finding counts can be in the thousands for an application. This high frequency drowns out most other vulnerabilities found in reports or data.

TaH, on the other hand, will find a broader range of vulnerability types but at a much lower frequency due to time constraints. When humans test an application and see something like Cross-Site Scripting, they will typically find three or four instances and stop. They can determine a systemic finding and write it up with a recommendation to fix on an application-wide scale. There is no need (or time) to find every instance.

Suppose we take these two distinct data sets and try to merge them on frequency. In that case, the Tooling and HaT data will drown the more accurate (but broad) TaH data and is a good part of why something like Cross-Site Scripting has been so highly ranked in many lists when the impact is generally low to moderate. It's because of the sheer volume of findings. (Cross-Site Scripting is also reasonably easy to test for, so there are many more tests for it as well).

In 2017, we introduced using incidence rate instead to take a fresh look at the data and cleanly merge Tooling and HaT data with TaH data. The incidence rate asks what percentage of the application population had at least one instance of a vulnerability type. We don't care if it was one-off or systemic. That's irrelevant for our purposes; we just need to know how many applications had at least one instance, which helps provide a clearer view of the testing is findings across multiple testing types without drowning the data in high-frequency results.

What is your data collection and analysis process?

We formalized the OWASP Top 10 data collection process at the Open Security Summit in 2017. OWASP Top 10 leaders and the community spent two days working out formalizing a transparent data collection process. The 2021 edition is the second time we have used this methodology.

We publish a call for data through social media channels available to us, both project and OWASP. On the OWASP Project page, we list the data elements and structure we are looking for and how to submit them. In the GitHub project, we have example files that serve as templates. We work with organizations as needed to help figure out the structure and mapping to CWEs.

We get data from organizations that are testing vendors by trade, bug bounty vendors, and organizations that contribute internal testing data. Once we have the data, we load it together and run a fundamental analysis of what CWEs map to risk categories. There is overlap between some CWEs, and others are very closely related (ex. Cryptographic vulnerabilities). Any decisions related to the raw data submitted are documented and published to be open and transparent with how we normalized the data.

We look at the eight categories with the highest incidence rates for inclusion in the Top 10. We also look at the industry survey results to see which ones may already be present in the data. The top two votes that aren't already present in the data will be selected for the other two places in the Top 10. Once all ten were selected, we applied generalized factors for exploitability and impact; to help rank the Top 10 in order.

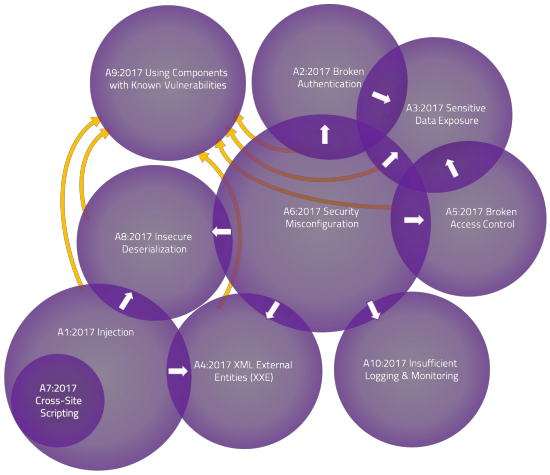
Data Factors

There are data factors that are listed for each of the Top 10 Categories, here is what they mean:

- *CWEs Mapped*: The number of CWEs mapped to a category by the Top 10 team.
- *Incidence Rate*: Incidence rate is the percentage of applications vulnerable to that CWE from the population tested by that org for that year.
- (Testing) *Coverage*: The percentage of applications tested by all organizations for a given CWE.
- *Weighted Exploit*: The Exploit sub-score from CVSSv2 and CVSSv3 scores assigned to CVEs mapped to CWEs, normalized, and placed on a 10pt scale.
- *Weighted Impact*: The Impact sub-score from CVSSv2 and CVSSv3 scores assigned to CVEs mapped to CWEs, normalized, and placed on a 10pt scale.
- *Total Occurrences*: Total number of applications found to have the CWEs mapped to a category.
- *Total CVEs*: Total number of CVEs in the NVD DB that were mapped to the CWEs mapped to a category.

Category Relationships from 2017

There has been a lot of talk about the overlap between the Top Ten risks. By the definition of each (list of CWEs included), there really isn't any overlap. However, conceptually, there can be overlap or interactions based on the higher-level naming. Venn diagrams are many times used to show overlap like this.



The Venn diagram above represents the interactions between the Top Ten 2017 risk categories. While doing so, a couple of essential points became obvious:

1. One could argue that Cross-Site Scripting ultimately belongs within Injection as it's essentially Content Injection. Looking at the 2021 data, it became even more evident that XSS needed to move into Injection.
2. The overlap is only in one direction. We will often classify a vulnerability by the end manifestation or "symptom," not the (potentially deep) root cause. For instance, "Sensitive Data Exposure" may have been the result of a "Security Misconfiguration"; however, you won't see it in the other direction. As a result, arrows are drawn in the interaction zones to indicate which direction it occurs.
3. Sometimes these diagrams are drawn with everything in *A06:2021 Using Components with Known Vulnerabilities*. While some of these risk categories may be the root cause of third-party vulnerabilities, they are generally managed differently and with different responsibilities. The other types are typically representing first-party risks.

Thank you to our data contributors

The following organizations (along with some anonymous donors) kindly donated data for over 500,000 applications to make this the largest and most comprehensive application security data set. Without you, this would not be possible.

AppSec Labs	GitLab	Micro Focus	Sqreen
Cobalt.io	HackerOne	PenTest-Tools	Veracode
Contrast Security	HCL Technologies	Probely	WhiteHat (NTT)

Thank you to our sponsors

The OWASP Top 10 2021 team gratefully acknowledge the financial support of Secure Code Warrior and Just Eat.

SECURE CODE WARRIOR

JUST EAT Takeaway.com