什麼是 OpenJDK? 是否合法與安全呢？

翻譯及整理｜叡揚資訊 資安事業處

1995年昇陽電腦發表 Java，那是整個軟體發展被簡單的用二分法分成私有或開源，大多數人都對開源軟體還保持懷疑態度的年代。Java 抱持著美好願景—「編寫一次，到處使用」（Write once, run anywhere），試圖征服這個混亂的年代。不過這個美好願景勢必要集眾人的力量才有可能達到目標，因此昇陽一開始便打算讓 Java 是個開放的環境，實際上當時也有許多廠商與社群投入 Java 的發展。

不過眾人總打著自己的如意算盤，漸漸的，Java 發展逐漸顯得混亂，不同廠商之間的 Java 環境並不能完美相容，反而成為「編寫一次，到處除錯」（write once, debug everywhere）的天大笑話。

昇陽為統一規格與凝聚開發能力，建立了 JCP（Java Community Process）社群，讓任何人都可以對 Java 規格提出建議，以 JSR（Java Specification Requests）統一所有 Java 的功能與特性、及 Java 版本的 Roadmap，並且使用 JCK (Java Compatibility Kit，也常常使用 TCK，Technology Compatibility Kit)，檢測各廠商所發展的 Java 環境是否符合規格。

之後昇陽也逐步開放自身所持有的原始碼，並建立 OpenJDK 專案，因為不需要重複製造輪子了，這讓 Java 規格更進一步走向統一。

> 昇陽逐步開放自身所持有的原始碼，並建立 OpenJDK 專案，因為不需要重複製造輪子了，這讓 Java 規格更進一步走向統一。

Duke，Java 的吉祥物，也是 OpenJDK 專案中的一個項目。
（圖片來源：https://openjdk.java.net/）

OpenJDK 專案曾經沒落

然而，好景不常，到了 2010 年甲骨文收購昇陽，同時也接收 OpenJDK 專案，但是卻又極度冷落這個專案。甚至一直到 2018 年四月才推出該專案的 Windows 與 MacOS 建構，因此若是在這段期間接觸 OpenJDK 的使用者，便會有 OpenJDK 什麼都沒有，支援度非常低的錯誤印象。但實際上，這只是 Oracle 冷落自己的 OpenJDK 社群，事實上有許多公司與社群使用這些公開的原始碼建立自己的組件，例如曾是最多使用者的 AdoptOpenJDK，或是各雲端廠商發布自己的 JDK。同時也因為昇陽已開放 Java 原始碼，因此事實上現在的 Oracle Java SE 與 Oracle OpenJDK 都是相同原始碼建構而成的實作參考 (Reference Implementation)，只是採用不同的授權而已。

不過在 2006 年原始碼剛開放之初，Java 7 的時期，有些 Java 程式碼授權是與 GPL 不相容的，例如，Java Web Start，導致這些不相容的套件不能包含在 OpenJDK 專案，這可能是江湖謠傳 OpenJDK 不能取代 Java SE 的主因。然而山不轉路轉，強大的社群仍實作出這些遺失的額外套件。同時，這些未包含在 OpenJDK 專案的部分，也逐步從 JSR 中去除，而不再是 Java 特性的一部分。如剛才提到的 Java Web Start，他有社群製作的 IcedTea-Web，能夠完全替代 Java Web Start，而 Java Web Start 也已在 Java 11 中移除。

那 OpenJDK 專案缺少了哪些套件呢？分別是 Applet Browser Plugin、Java Web Start、JavaFX 以及打包於 Java SE 內的商用授權字型。除了字型之外，這些缺少的特性都已經是 Deprecated 或 Deleted 的套件，因此未來再也不會看到這些「不法份子」了。

GPLv2 含 Classpath Exception 與 Oracle No-Fee Terms and Conditions

因為 OpenJDK 專案一開始便是昇陽公開 Java SE 的原始碼而來，因此與 Oracle Java SE 之間不存在仿製、侵權、反組譯之類的法律問題，甚至可說是使用相同原始碼來源，以不同授權釋出的建構而已，然而 Java 使用者要注意不同的授權所相對應的責任問題。Oracle Java SE 便採 Oracle No-Fee Terms and Conditions 商業授權，只允許個人免費使用，若要商業應用便要收費。

也因此許多企業紛紛棄用 Oracle Java SE，轉向 OpenJDK。至於 OpenJDK 採 GPLv2 含 Classpath Exception，與一般的 GPLv2 相比，GPLv2 含 Classpath Exception 允許開發者的程式碼靜態與動態連接 Java 函式庫，而不需要強制遵守 GPLv2，這讓許多社群願意使用 Java 進

行開發，例如 Apache 得以開發許多 Java 殺手應用，同時以 Apache 授權釋出。

然而，OpenJDK 畢竟是 GPLv2 為主的授權，就算允許例外，也難保會不會有「誤用」導致所有程式碼全部被 GPL 污染的狀況，進而被要求公開原始碼。這時小心選用保證不會有汙染，且有賠償的 OpenJDK 建構便顯得重要。

OpenJDK 安不安全？看看 CVE-2019-2699

2019 年 4 月 23 日，NVD 公布編號 CVE-2019-2699 的 Java 漏洞，這是一個高達 9 分的高風險漏洞，達成手法略微複雜，不過一旦成功，便有可能對 Java SE 與其所有相關套件造成破壞，因此必須要盡快修補。然而這個漏洞的重要性不單只於它本身造成的攻擊，更重要的是時間點。這是 Oracle 宣布任何 Java SE 的商業應用必須收費之後所出現的第一個重大 CVE，同時該修補也在第一個收費版本 8u212 中釋出。因此不願意付費的 Java SE 使用者企業，只能停留在 8u202 之前的版本。這對維護資訊安全是極差的決定，最好還是大刀闊斧，轉移到安心可靠，持續維護的 OpenJDK 還比較安全。

目前 Java 有兩種版本號碼推進模式，Java 8 之前每一個大版本號碼都是長期支援，相同版號特性會相同，如 Java7, Java8，安全更新採整包替換的方式，每季固定推出新的關鍵更新 (CPU, Critical Patch Updates)，如 2022 年一月推出的 8u321。

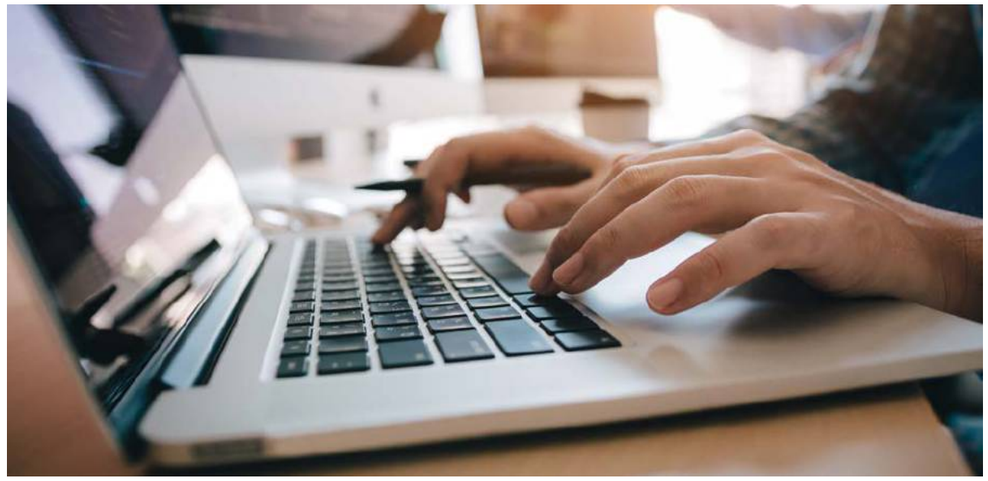
同時，JSR 規定不同供應商所提供相同版本號碼 (含 update 小版號) 的 Java 建構特性都必須要一致，基於大家的來源都是 OpenJDK 專案，甚至不可諱言的原始碼可能有相當程度的重疊，因此出現在 Oracle Java SE 的漏洞，有可能其他 OpenJDK 建構也會有，不同供應商所建構的 OpenJDK 偶爾也有自身特有的漏洞，因此供應商是否持續維護是很重要的一個要點。

免費 Java 是否適合您

如果問到使用什麼 Java 環境，得到的回答是 OpenJDK，就好像是問使用什麼作業系統，回答 Linux 一樣。事實上，Java 目前至少有超過 8 家廠商提供免費的建構，包括 Oracle 與 Azul。既然有這些選擇，為什麼要付費？為什麼不直接部署一個免費 OpenJDK 就好？使用免費的 OpenJDK 會有多大風險？

評估一個 OpenJDK 建構以及其供應商是否可用並不單只做安裝與回歸測試，請多考慮以下幾點：

- 這些供應商的支援能力和維護記錄
- 季安全更新記錄是否依慣例正常釋出 (Java 的安全更新慣例，一般每季釋出一次)，且是否有確保回歸性能而不需要使用者每次部屬都需要完整回歸測試



- OpenJDK 發行版是否經過 TCK 測試，亦即是它是否真正相容於所有 Java
- 您的智慧財產權是否受到保護

供應商是否能持續且穩定的提供支援是最重要的首要考量，尤其是 GPLv2 授權本身即說明了不提供保固 (見 GPLv2 , 章節 11, NO WARRANTY)。因此誤用了爹不疼娘不愛的 OpenJDK 建構，也許部屬階段暫時沒有任何問題，碰到任何異常可是求救無門，而且也可能遇到斷炊，社群不再維護 OpenJDK。

其次是 Oracle 對於 Oracle Java SE 會每季提供季更新，稱之為“關鍵更新”(CPU, Critical Patch Updates)，同時也讓 NVD 公開該 CPU 所修補的 CVE 漏洞們，因此各廠商維護能力是否有辦法跟上腳步也是一個重要的評鑑要素，例如 Oracle 自己的 OpenJDK 就不一定跟上 Oracle Java SE 的腳步釋出更新。

而提供的 OpenJDK 是否通過 TCK 相容測試是技術分析時就應該先檢查的項目，選擇有 TCK 的建構之後再進行啟動測試與回歸測試。最後，由於目前對於 GPL 的解釋仍有模糊地帶，因此是否對於授權疑慮有明確解釋或保障，也是評估的要點。