

公務機密

資訊安全維護

釣魚簡訊要小心 守護錢財防詐騙



資安時事案例

修但幾勒！你下單的購物網站安全嗎？小心落入「一頁式網購詐騙」陷阱！

實聯制要小心！掃QR碼APP藏病毒

個人資料保護法

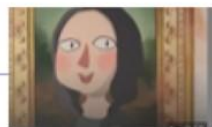
特種個人資料

生活中的資安

祕密已經不再是祕密？

數位學習

105年資安動畫金像獎 第一名
David's Secret



修但幾勒！你下單的購物網站安全嗎？小心落入 「一頁式網購詐騙」陷阱！



民眾日報 2022年2月10日

隨著科技的日新月異，消費者在社群平台上購物的頻率逐漸提升，許多代購、團購也紛紛轉戰至社群平台，消費者只要動動手指頭，或是在留言區喊聲「加1」，就可以買到各式各樣的物品。

然而詐騙也看上了如此便利的購物方式，加上社群平台的交易規範並不像大型購物平台一樣完善，因此詐騙更容易使用社群平台作為詐騙工具！越習慣在社群平台購物，就越要小心詐騙陷阱！快來看看社群網購詐騙常見的手法有哪些吧👉

篇名：早知道是這樣 如夢一場

天天在家防疫的小蓉快被悶壞了，無精打采的滑著社群網站。突然，一則廣告映入眼簾：

Fan Cold
5月16日上午 10:29

【專櫃豪華彈潤保養精華組合】限時出清大特價！

~知名女星代言限量專櫃保養品精華禮盒~
!!原價NT\$10588【限時下殺】只要NT\$2588!!

🌟內容物誠意滿滿🌟
玫瑰經典化妝水300ml x1
玫瑰萃取淨白乳液200ml x1
毛孔緊緻精華液100ml x1
水光保濕面膜1盒(6入)
美白透亮洗面乳200ml x1
潤澤卸妝凝露150ml x1

你還在等什麼？限量300組要搶要快！
購買連結：fancold2.shop.com

fancold2.shop.com
【女星代言強力推薦】限量要搶要快！

193 219則留言

讚 留言 分享

當小蓉準備下單時，她不禁懷疑：「等等，這麼豪華的專櫃保養組合，價格未免太便宜了吧！該不會是詐騙……？」就在此時，小蓉看見賣家大力宣傳「非詐騙！貨到付款最有保障！」的標語。

「對耶！選擇貨到付款的話，就不會有匯款後沒收到貨的問題了！」於是，小蓉放心的按下「立即購買」按鈕，開始期盼著豪華專櫃保養禮盒的到來……一個禮拜後，小蓉期待地到超商付款取貨，一打開包裹，卻只看見一堆莫名其妙的塑膠泡綿、五金零件和保麗龍！氣得她立刻找賣家理論，卻發現賣家帳號早已消失，求助無門的小蓉看著一箱子的垃圾，欲哭無淚……

你發現了嗎？4 大疑點揭穿社群網購詐騙！

社群網購詐騙篇

4大疑點揭穿網購詐騙！

單一商品
網站為單一網頁，而且只販賣一種商品？
小心可能是「一頁式詐騙網站」！

名人推薦
盜取名人照片並 P 圖偽冒的詐騙層出不窮，務必去本人的社群帳號查證！

限時特價
以「高價品、低價賣」的手法吸引目光，
以聳動標題、活動倒數計時等引誘上鉤！

強調保障
強調「貨到付款有保障」或有「七天鑑賞期」
，民眾發現包裹內容物不符後卻求償無門！

whoscall

若收到與訂購商品不符的包裹，想要向賣家退貨時，卻發現自己早已被詐騙賣家封鎖，即使有「七天鑑賞期」仍然會求助無門。而如果選擇「貨到付款」，也需要先付款後才能確認包裹內容物是什麼。因此，若賣家在購物網站上強調商品有七天鑑賞期，或只提供貨到付款的結帳選項，很可能高機率是詐騙集團所經營的網站！請務必要小心提防！小心！簡訊竟也是網購詐騙戰場？！

除了社群平台較常有網購詐騙的蹤跡外，「廣告簡訊」也時常夾帶一頁式詐騙網站的連結！根據 Whoscall 用戶主動回報的內容顯示，曾收到以下類似簡訊：

「【知名潮牌特賣會】
經典款運動鞋限量買一送一！
機能性強！舒適感十足！
購買要快 <http://run.shoes.com>」

常見的網購詐騙簡訊特徵包含：

1. 來源不明的廣告內容
2. 夾帶冗長、亂碼陌生網址

小心！此類簡訊所夾帶的連結可能會帶你掉入一頁式網站的詐騙圈套！詐騙先是用引人注目的優惠廣告台詞，再附上連結網址，讓民眾好奇點進至網站後下單！稍不注意，詐騙就會悄悄進入生活中，務必隨時提高警覺！

下單後才發現是詐騙有解嗎？教你黃金自救 3 步驟！

步驟一：蒐集證據

收到與訂購商品不符的貨物時，務必完整保存收到的「產品」及「貼在包裝盒上的貨運單」，因為上面通常記錄著重要的寄件人姓名、電話、地址。另外，訂購成功時的截圖，以及任何與賣家聯繫的對話記錄都是非常好的證據來源！

步驟二：申請「止付貨款」及「退貨退款」

商品於七天鑑賞期內，可立即打電話要求物流公司不要付款給寄件人，並向物流公司申請「退貨退款」。然而若是於超商取貨、或已經超過七天鑑賞期，就只能索取寄件人聯絡方式並自行要求退貨退款！

如果是用信用卡支付，可以打電話給銀行請求協助止付，並調出「請款人」資料。

步驟三：報案

如果還是無法取得退款，可以向消保會/消基會申訴物流公司和寄件人，並備齊所有相關證據資料後，使用內政部警政署165反詐騙線上報案！

社群/簡訊網購詐騙怎麼防？養成 5 個好習慣，遠離詐騙陷阱！

一、選擇有信用的購物平台

專業的購物平台擁有完善的退貨及退款制度，即使商品有問題，也能找到負責人員處理相關事宜。另外，也可以選擇具有第三方支付功能的網購平台，透過資金暫留的機制，可以在消費者確認商品沒有問題後，才撥款給賣家，讓消費者的權利多一層保障！

二、避免和賣家透過通訊軟體私下聯繫

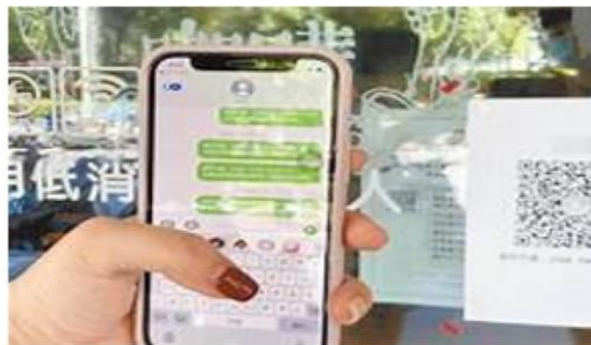
與交易平台不同，社群平台及通訊軟體缺乏完善的交易規範及審核機制，私下聯絡不僅無法留下公開的交易記錄，還很有可能在被詐騙後被賣家已讀不回、甚至封鎖！

三、查核商家的真實性

若發現從沒聽過商家名稱、賣家的粉絲專頁成立時間短、粉絲或追蹤人數很少等可疑之處，可立即至經濟部網站查詢公司名稱、地址等基本資料。如果發現網站沒有附上任何公司資料，或是公司資料不在台灣，建議不要購買此商家的商品！

四、點擊簡訊連結前先檢查

使用 Whoscall Android 檢查簡訊連結功能，多一個動作、多一層保護！前往購物網站前，先點擊檢查連結，若系統掃描後顯示該連結可能有風險，強烈建議不要點擊網址，以免進入惡意詐騙的購物網站！



實聯制要小心！掃QR碼APP藏病毒

三立新聞 2022-03-04 科技中心／李艾庭報導

疫情下，進出公共場所都得掃實聯制QR Code，不過國外資安公司Cleafy近日發現，木馬程式病毒「TeaBot」，隱身在一款QR Code掃描器APP中，而這個APP下載量目前超過1萬次，恐怕已經不少用戶的個資遭竊取。

資安公司Cleafy近日指出，病毒「TeaBot」2021年5月現身，假冒成QR Code掃描器APP，隱藏在Android的Google Play裡面，也因為它功能完整、方便，評價良好，目前下載量已經超過1萬次。

Google Play上的「QR Code & Barcode Scanner」原本是個良性程式，但安裝後，它會要求更新，以傳送病毒，而這個更新不是透過Google Play，而是另一種程式，一旦執行更新，TeaBot便會啟動，要求無障礙服務（Accessibility Services）權限，進而監控螢幕、竊取憑證，或是其他駭客惡意行為。

Cleafy表示，因為TeaBot是透過官方的Google Play散布，授權要求少，因此難以引發使用者的懷疑，也能經常躲過防毒軟體。所以用戶要掃描QR Code或實聯制時，最好還是透過手機相機、內建程式或是Google官方掃描器，別亂下載APP。

(圖片載取網路)



特種個人資料

個資法施行細則對於病歷、醫療、基因、性生活、健康檢查及犯罪前科的個人資料，有進一步的定義，例如性生活的個人資料是指性取向或性慣行，犯罪前科的個人資料是指經緩起訴、職權不起訴或法院判決有罪確定、執行的紀錄。而病歷的個人資料，是指醫療法第67條第2項所列的各款資料，醫療的個人資料則是指病歷及醫師或醫事人員在診察、治療、處方、用藥、施術或處置所產生的個人資料。由於醫療個資的定義包含了病歷，然而醫療個資依第6條規定屬特種個資，但病歷卻未屬於特種個資，因此行政院將提修法，把病歷也納入特種個人資料的範疇。



特種個人資料

(12~16項屬於特種個人資料，個資法第6條針對特種資料有特別規定)



◆ 社團法人台灣 E 化資安分析管理協會理事長、中央警察大學資訊密碼暨建構實驗室 (ICCL) — 王旭正教授

科技越發達，隱私越具價值。現代資安科技重視「個人隱私」、「家庭隱私」、「國家隱私」、「國際隱私」，恰對照古云「修身」、「齊家」、「治國」、「平天下」，也自然畫點出「資安就是國安」的精髓。

Security 和 Forensics 是搭檔 成為「SecForensics」趨勢

傳統鑑識 (Forensics) 不需與數位或電腦系統直接聯想，也不常發生，例如槍枝走火的判定、火災的火源在哪裡；然而，如今在數位洪流裡，卻在身邊、在把玩手機時，不知不覺就會有鑑識問題跑出來諸如大家的生活好朋友 LINE、FB 遭盜用，莫名地成為入侵攻擊事件的主角；民眾也會習慣性地藉 google 網路，期能獲得科技法律的諮詢與了解，進而尋求保障個人資料的安全等。所以 Security 和 Forensics 現今已是搭檔，經常綁在一起，兩者合體即為「SecForensics」，這詞是趨勢，也是資安生活時勢裡最重要核心觀念



在臺灣，LINE 已成為生活上不可或缺的手機軟體，諸如通話、視訊、看新聞、購物等各種應用皆會透過 LINE，其普及率已讓其容易成為資安事件的嫌疑者或受害者。

與共通認知，未來在新科技字典裡看到這個字可是預見的必然。

科技越是發達，隱私越具有價值，「個人隱私」、「家庭隱私」、「國家隱私」、「國際隱私」是現代資安科技呼應古云「修身」、「齊家」、「治國」、「平天下」的最佳寫照，似乎也自然地畫點出「資安就是國安」的重要精髓，decode it in total！

與手機的愛恨情仇

資安鑑識有分電腦鑑識（Computer Forensics）、網路鑑識（Network Forensics），是現在正夯的議題。鑑識無時無刻

不在，當我們想著手機，那可是與手機的愛與恨，即你非常喜歡手機，然手機也可以做很多的壞事而讓你恨之入骨呢。

無所不在的無線網路，便捷又免費，但是不用錢的最貴，不要為了省錢，資料全部都被截走了還不自知！情報都是有價值的，要防止訊息被截走，不要輕易去用免費的無線網路。在臺灣，LINE 已取代了很多我們生活上原本的習慣，例如打電話、看新聞、購物等……由於現代人使用手機非常普遍（甚可說是「氾濫」），在使用手機的過程當中，會很容易跟別人連上線、互相傳送訊息，手機在不知不覺當中就會成為資安事件的入侵嫌疑者或受害者。

在 LINE 裡，雖然是大部分是好朋友，但在好友列表裡面也有可能是不是經意加入的 LINE 好友。例如加商家為好友才能下載免費貼圖，逛街買東西要加店家的 LINE 才有打折等等。商家總會提供一些好處、優惠的方案給你。但在 10 個好處裡面，總有那麼一、兩個是準備要在你的手機裡面植入木馬。「最便宜的最貴！」就是抓住人們貪小便宜的心態，才會無時無刻有駭客入侵事件發生。

這些商家，偶爾會送些訊息到你的手機，可能會再提供一些網址請你按連結，不知不覺中，你的手機就會被植入木馬，不經意地被入侵了。這時就需要「數位鑑識」來解救了。



手機可以是一個資料庫、通訊器，但同時亦可能成為秘密暴露的出口。

驚人的現世報— 秘密已經不再是秘密

反過來，有時候我們也可能流程操作錯誤，而「不小心」入侵了別人的手機、電腦，雖非是故意的，但行為上就是已經被對方覺得你是在干擾他、入侵他了。無心之間擾亂了別人的系統，對方卻覺得就是你在作怪，入侵對方系統，無緣無故就被對方告了！「數位鑑識」裡證據會說話，可還你清白，卻也顯露了驚人的現世報—「秘密已經不再是秘密」。

為何「秘密已經不再是秘密」？手機可以是一個資料庫、資訊的來源，另一方面也是貼心的工具，能幫你存有許多不欲人知的秘密。而且如果真能成功地將木馬植入別人手機裡，就可以完全知道別人手機裡的狀況。手機還可以透過一些 APP 小程序去定位別人。這些軟體看似好用，可以直接呼朋引伴，也能清楚知道你在哪裡。啊，如此一來，秘密就已經不再是秘密了呀。

軟體「鑑識」與硬體「證據」 相輔相成

數位鑑識乃是使用科學技術進行搜集、鑑定、找出關聯性、運用各種技術將數位證據文件化，並找出與案件所需且相關的數位證據。數位證據有如電腦結構中之硬體，這些硬體散落在犯罪現場，需要

靠鑑識人員細心的將所有的證據一一找出，電腦若僅有硬體而沒有軟體的輔助，電腦硬體就像是英雄無用武之地，也由於電腦硬體與軟體的天作之合，才得以開啟電腦世代的新紀元。

反觀「鑑識」與「證據」的組合，互依互存有如天作之合的軟體「鑑識」與硬體「證據」，少了其中一種就無法發揮其作用。因此如果沒有「證據」的殘屑佇留，何來「鑑識」之推敲、溯衍，另一方面，沒有「鑑識」的抽絲剝繭，碎屑依然散落，就算有再多的證據也無「證明力」來證明犯罪事實。



「鑑識」就是將犯罪現場、證據及被害者與嫌疑人之間的關係與來龍去脈描述清楚。

如何從眾多證據中， 找到證明犯罪事實

了解數位鑑識與證據之關係後，最重要的是如何從眾多的證據中找到足以證明犯罪的事實；另一方面是利用數位鑑識工具及方法所萃取出來的證據，好好保存以免失去其證明犯罪之證據力及證明力。其中證據力，是一個水平的概念；證明力，則是垂直的概念。掌握愈多證據，愈多元化，就有愈強的證據力。而什麼是垂直的證明力？你可以從一根頭髮，判斷他是男生或女生（第一層）、年齡層（第二層），或是分析出這個人有哪些疾病（第三層），掌握愈多層次，代表證據證明力愈強！

「鑑識」是將事情弄得清清楚楚，是一個流程、一個說法，而整個過程當中，要有一些東西是實質、眼睛看得到的，就是所謂的「證據」。所以，要構成犯罪，需要具備的四個元素，第一個：真實的犯罪現場（網路上的虛擬、想像的，沒有留下痕跡的，是不成立；當有留下痕跡、紀錄、文字等，即可為成為證據的基本依據）；第二個：被害者；第三個：嫌疑人；第四個：證據。我們將整個過程、這四個元素的來龍去脈、兩兩的互動關係上，描述得非常清楚，這樣的一套過程模式，就是「鑑識」。

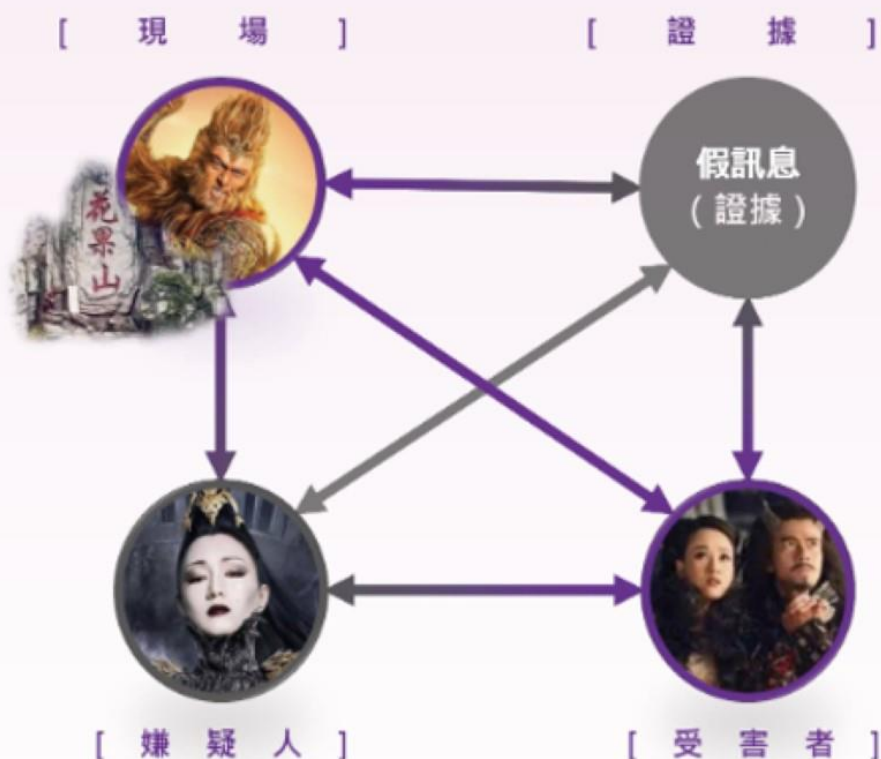


圖 1 數位鑑識的 K_4 系統面相

所以當手機犯罪，就需要將手機扣押，撈出各種可能的證據，再去定位一下犯罪現場在哪裡？誰是受害者？誰是可能的嫌疑人？到底做了什麼事的證據痕跡？證據和現場的關係？證據和受害者的關係？證據和嫌疑人的關係？還有嫌疑人和受害者的關係？受害者、嫌疑人為什麼會在現場？當把全部證據找得完完整整，整個犯罪過程弄得清清楚楚，才能真正地說服所有人。

圖 1 中，我們以數位鑑識的 K_4 完全圖說明如下，其中孫悟空的花果山為事件現場，白骨精為發布訊息的嫌疑人，發布的訊

息經資安的鑑識檢驗為假訊息證據，牛魔王與芭蕉公主為假訊息事件的受害者，此四元素的關係得呈現完全的相互關連性。

由 RootKit 談「反鑑識」概念

數位鑑識也有「反鑑識」的概念，RootKit（隱藏程式）就是比較趨向反鑑識（Anti-forensics）的概念，它不能讓別人知道它，但是它又嘗試在裡面扮演重要的角色。為什麼 RootKit 有點像反鑑識？「反鑑識主要是隱藏自己的身分不讓別人知道」。是啊，RootKit 是「隱藏程式」的概念，那 RootKit 的這種隱藏程式是為了做壞事？

還是在必要的時候讓自己發揮作用？其實 RootKit 以正面的角度來看，是系統管理者的重要助手，管理者是正面的機制，入侵者是負面、破壞者。

RootKit 本用於隱藏行程的功能，可對系統進行存取或將系統核心中所使用的行程隱藏起來，避免使用者在操作時不小心而影響到系統運作，出發點並非惡意。我們這個社會是一個正面的正義模式，當我們社會機制遭到破壞的時候，就可以透過這個正義模式的機制，讓破壞攻擊降到最低。但如果有人有心將這個正面的機制拿來破壞社會次序，那它還是會變成負面的。例如警察向來代表正義力量，但若他被人收買，變成負面的去做壞事，反而造成更

大的社會危機。引用這些譬喻是為了說明，當駭客利用此手法變向操作 RootKit 時，反會將木馬程式等隱藏到作業系統中，從而造成意料之外的危險，那麼 Rootkit 將被視為是非常危險的惡意軟體。

反鑑識的正向價值觀

數位鑑識與反鑑識，並不是狹隘的相反定位而是相互為用。反鑑識主要用意是保護商業利益，隱藏一些機密的資訊，反鑑識裡的證據不能被找到。然反鑑識不是「把犯罪的證據藏起來」，因為如果說「反鑑識是把犯罪的證據藏起來」，大帽子一扣，大家都會怕，聽了會心驚膽跳。例如一間商店寫「殺人放火店」，裡面賣凶器，

```
static void __init rootkit_init(void)
{
    // Load kernel module with rootkit
    if (kernel_init == 0) {
        if (kernel_init == 0) {
            return 0;
        }
        if (kernel_init == 1) {
            return 0;
        }
        if (kernel_init == 2) {
            return 0;
        }
        if (kernel_init == 3) {
            return 0;
        }
        if (kernel_init == 4) {
            return 0;
        }
        if (kernel_init == 5) {
            return 0;
        }
        if (kernel_init == 6) {
            return 0;
        }
        if (kernel_init == 7) {
            return 0;
        }
        if (kernel_init == 8) {
            return 0;
        }
        if (kernel_init == 9) {
            return 0;
        }
        if (kernel_init == 10) {
            return 0;
        }
        if (kernel_init == 11) {
            return 0;
        }
        if (kernel_init == 12) {
            return 0;
        }
        if (kernel_init == 13) {
            return 0;
        }
        if (kernel_init == 14) {
            return 0;
        }
        if (kernel_init == 15) {
            return 0;
        }
        if (kernel_init == 16) {
            return 0;
        }
        if (kernel_init == 17) {
            return 0;
        }
        if (kernel_init == 18) {
            return 0;
        }
        if (kernel_init == 19) {
            return 0;
        }
        if (kernel_init == 20) {
            return 0;
        }
        if (kernel_init == 21) {
            return 0;
        }
        if (kernel_init == 22) {
            return 0;
        }
        if (kernel_init == 23) {
            return 0;
        }
        if (kernel_init == 24) {
            return 0;
        }
        if (kernel_init == 25) {
            return 0;
        }
        if (kernel_init == 26) {
            return 0;
        }
        if (kernel_init == 27) {
            return 0;
        }
        if (kernel_init == 28) {
            return 0;
        }
        if (kernel_init == 29) {
            return 0;
        }
        if (kernel_init == 30) {
            return 0;
        }
        if (kernel_init == 31) {
            return 0;
        }
        if (kernel_init == 32) {
            return 0;
        }
        if (kernel_init == 33) {
            return 0;
        }
        if (kernel_init == 34) {
            return 0;
        }
        if (kernel_init == 35) {
            return 0;
        }
        if (kernel_init == 36) {
            return 0;
        }
        if (kernel_init == 37) {
            return 0;
        }
        if (kernel_init == 38) {
            return 0;
        }
        if (kernel_init == 39) {
            return 0;
        }
        if (kernel_init == 40) {
            return 0;
        }
        if (kernel_init == 41) {
            return 0;
        }
        if (kernel_init == 42) {
            return 0;
        }
        if (kernel_init == 43) {
            return 0;
        }
        if (kernel_init == 44) {
            return 0;
        }
        if (kernel_init == 45) {
            return 0;
        }
        if (kernel_init == 46) {
            return 0;
        }
        if (kernel_init == 47) {
            return 0;
        }
        if (kernel_init == 48) {
            return 0;
        }
        if (kernel_init == 49) {
            return 0;
        }
        if (kernel_init == 50) {
            return 0;
        }
        if (kernel_init == 51) {
            return 0;
        }
        if (kernel_init == 52) {
            return 0;
        }
        if (kernel_init == 53) {
            return 0;
        }
        if (kernel_init == 54) {
            return 0;
        }
        if (kernel_init == 55) {
            return 0;
        }
        if (kernel_init == 56) {
            return 0;
        }
        if (kernel_init == 57) {
            return 0;
        }
        if (kernel_init == 58) {
            return 0;
        }
        if (kernel_init == 59) {
            return 0;
        }
        if (kernel_init == 60) {
            return 0;
        }
        if (kernel_init == 61) {
            return 0;
        }
        if (kernel_init == 62) {
            return 0;
        }
        if (kernel_init == 63) {
            return 0;
        }
        if (kernel_init == 64) {
            return 0;
        }
        if (kernel_init == 65) {
            return 0;
        }
        if (kernel_init == 66) {
            return 0;
        }
        if (kernel_init == 67) {
            return 0;
        }
        if (kernel_init == 68) {
            return 0;
        }
        if (kernel_init == 69) {
            return 0;
        }
        if (kernel_init == 70) {
            return 0;
        }
        if (kernel_init == 71) {
            return 0;
        }
        if (kernel_init == 72) {
            return 0;
        }
        if (kernel_init == 73) {
            return 0;
        }
        if (kernel_init == 74) {
            return 0;
        }
        if (kernel_init == 75) {
            return 0;
        }
        if (kernel_init == 76) {
            return 0;
        }
        if (kernel_init == 77) {
            return 0;
        }
        if (kernel_init == 78) {
            return 0;
        }
        if (kernel_init == 79) {
            return 0;
        }
        if (kernel_init == 80) {
            return 0;
        }
        if (kernel_init == 81) {
            return 0;
        }
        if (kernel_init == 82) {
            return 0;
        }
        if (kernel_init == 83) {
            return 0;
        }
        if (kernel_init == 84) {
            return 0;
        }
        if (kernel_init == 85) {
            return 0;
        }
        if (kernel_init == 86) {
            return 0;
        }
        if (kernel_init == 87) {
            return 0;
        }
        if (kernel_init == 88) {
            return 0;
        }
        if (kernel_init == 89) {
            return 0;
        }
        if (kernel_init == 90) {
            return 0;
        }
        if (kernel_init == 91) {
            return 0;
        }
        if (kernel_init == 92) {
            return 0;
        }
        if (kernel_init == 93) {
            return 0;
        }
        if (kernel_init == 94) {
            return 0;
        }
        if (kernel_init == 95) {
            return 0;
        }
        if (kernel_init == 96) {
            return 0;
        }
        if (kernel_init == 97) {
            return 0;
        }
        if (kernel_init == 98) {
            return 0;
        }
        if (kernel_init == 99) {
            return 0;
        }
        if (kernel_init == 100) {
            return 0;
        }
    }
}

static void __exit rootkit_exit(void)
{
    procfs_clean();
    fs_clean();
}

module_init(rootkit_init);
module_exit(rootkit_exit);
```

RootKit 是系統管理者的重要助手，但若遭入侵者利用，就會變成危險的破壞者。(Photo Credit: Christiaan Colen, <https://www.flickr.com/photos/christiaancolen/21133308006>)

一定會倒，因為非社會正義當然立即會被取締關門。又若你貼文在 FB 上標明「殺人放火店」，按讚的人也都有可能被調查是否有犯罪動機。

現代巡邏有所謂的「網路巡邏」，在「科技—資安—鑑識」已是國際化的趨勢下，傳統巡邏也在資訊時代洪流裡演化成網路巡邏。就如剛才所說，若在 FB 上張貼具有擾亂社會秩序嫌疑內容時，也會很快被社會公權力單位之網路巡邏者發現，並迅速地被抑制。

藉此我們想說明「反鑑識」，並不是狹隘地將證據湮滅掉來鼓勵犯罪，而是在保護商業利益方面的智慧財產權（例如網站新聞內容是有財產權的），資訊隱藏、資料偽裝亦或是軍事間諜、線民臥底等，也有反鑑識的正面價值觀，能協助情報偵蒐，裡應外合地破案，透過迂迴方式以打擊危害社會秩序的各式非法行為，這也是「反鑑識」的最原始正面的價值觀。

運籌帷幄，決勝於網路之內

生活中一直都存在傳統犯罪，有了電腦犯罪之後，還是會有西瓜刀、棒球棍，總不能拿手機對砍吧！但科技犯罪遠比傳統犯罪誇張、無遠弗屆，戴著鴨舌帽搶銀行的行為已落伍，以「高科技方式搶銀行」正時興，如同「運籌帷幄，決勝於千里之外」，在看不到的地方算計才是勝敗關鍵。而網路犯罪產生的經濟災損，幾乎都是上億元起跳，超乎想像，也不是以棒球棍回擊就可了結的。

祕密或證據從「天知、地知、你知、我知」之時空背景，在數位鑑識時代裡成了「錯、錯、錯、錯」之大家都知道的祕密，「祕密已經不再是祕密」；證據全留存在系統、手機、電腦、網路，近在我們身邊！資安科技與個資已是密不可分的重要貼身好朋友，而懂資安、找證據（鑑識）、保障個資祕密（反鑑識）成為在現代生活中保護自己的必備武器，藉此，也才能自在地享受科技帶來的便利。



社團法人台灣E化資安
分析管理協會 (ESAM)



中央警察大學資訊密碼
暨建構實驗室 (ICCL)