Communications of the CCISA
Vol. 26 No. 1 Feb. 2020

霧運算環境中使用人工智慧網路分析中間人攻擊機制

張世豪[1]、蕭亦筑[2]
淡江大學資訊工程學系[1,2]
shhchang@mail.tku.edu.tw[1]，cris63@gmail.com[2]

摘要

本論文主要探討軟體定義霧聯網下的網路攻擊的研究，使用人工智慧分析，我們必需要清楚明白霧聯網與軟體定義網路的運作原理，才能洞察如何可以改善霧聯網與軟體定義網路的安全通訊機制，進而才可以再作頻寬效率度的提昇。雖然軟體定義霧聯網可以比目前網路大幅減少頻寬的浪費，但若沒有達到穩定與安全性的通訊目標，仍會有疑慮是否可以適用的問題，則使用霧聯網與軟體定義網路架構的意願會降低。此外，對於系統業者而言，若霧聯網與軟體定義網路的傳輸安全性與網路安全性無法確保。在本論文中，我們使用了人工智慧分析來解決了安全性的疑慮，並且提出有效的網路路徑最佳化演算法、軟體定義網路的封包安全檢測機制，人工智慧的分析機制。

關鍵詞：軟體定義網路，霧聯網，惡意封包分析，Ada Boost，有限狀態自動機

Regular Paper

Communications of the CCISA
Vol. 26 No. 1 Feb. 2020

Using Artificial Intelligence Network to Analyze Man-in-the-Middle Attack Mechanism in Fog Networking Environment

Shih-Hao Chang [1], Yi-Ju Shiau[2]
Department of Computer Science and Information Engineering, Tamkang University, New Taipei City 25137, Taiwan (R.O.C.)
shhchang@mail.tku.edu.tw[1]，cris63@gmail.com[2]

Abstract

This paper discusses the network attacks under software-defined fog networking. With artificial intelligence analysis, we need to clearly understand how fog networking and software-defined networking work. By this way, we can gain insights into how to improve the secure communication mechanism and bandwidth efficiency between the fog network and the software-defined network. Although software-defined fog networking can significantly reduce the bandwidth compared to current networks. If the goal of stable and secure communication is not met, there are still doubts about whether it is applicable. In addition, for the system operator, if the transmission security and network security of the fog networking and the software-defined network cannot be guaranteed. It will reduce the willingness to use the fog networking and software-defined network architecture. In this paper, we use artificial intelligence analysis to address security concerns. We also propose the effective network path optimization algorithm, packet inspection mechanism of software-defined network, and artificial intelligence analysis mechanism.

Keywords: SDN, Fog Networking, Malware Analysis, Ada Boost, FSM

壹、前言

隨著物聯網，雲端服務與人工智慧服務的普 及化，使我們的生活變得更加輕鬆。特別這些服務 也同時啟動了霧聯網 （Fog Networking）或稱之為霧運算（Fog Computing）。霧聯網與霧聯網是使用最終 端用戶裝置的邊緣裝置，以分散式協作架構進行資 料儲存或進行分散式網路封包傳輸通訊。或相關分 散式控制或管理。霧聯網是由思科在 2014 年所提 出的概念[1]，為雲端運算的延伸，這個架構可以將 運算需求分層次、分割區域處理，以化解可能出現 的網路塞車現象。霧聯網的應用和物聯網（IOT）及智慧型聯網（M2M）有密不可分的關係。

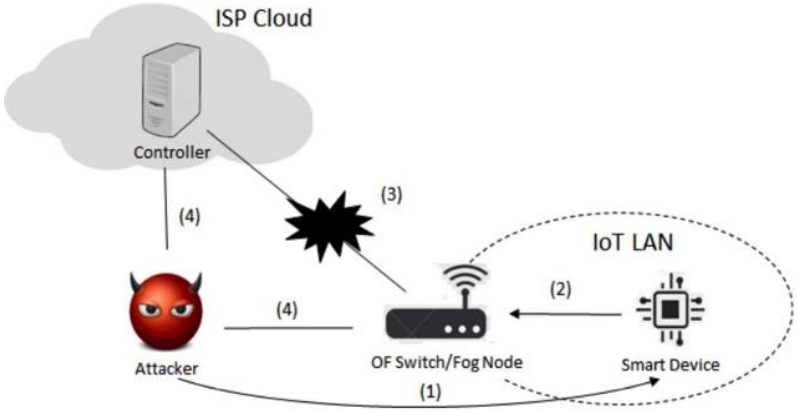
然而，霧聯網需要一種全新的網路架構，然而，過去傳統的網路架構不是針對霧運算所產成的資料量，種類和速度而設計的。比如說：若以目前數十億個尚連接的物聯網設備則每天產生超過 2 EB 的資料量。若到 2020 年，估計將有 500 億物聯網連接到網際網路，若將這些資料傳輸到雲端上進行儲存需要非常大量的頻寬。除此之外，在處理物聯網資料的種類和速度也都需要新的運算模型 [2, 3]。因此，霧運算所需要的下一代網路的主要訴求是：

- 最少的網路延遲時間：例如當嘗試阻止生產線停機或恢復電氣服務時，極短的延遲時間（如毫秒級）變的很重要，透過即時分析設備所產資料可以避免設備因停機所產生的災難和系統故障。
- 解決網路安全問題：物聯網資料需要在傳輸和儲存時都需得到保護。這保護的需求是在整個傳輸與儲存的過程中，需要有監控和防禦攻擊的機制。
- 資料完整性和可用性：物聯網資料越來越多用於影響民眾安全和關鍵基礎設施的決策，因此在關鍵基礎設施的資料完整性和可用性不容置疑。
- 惡劣環境與地理條件情況下資料的收集與保護：物聯網設備可分佈在數百公里或更廣的區域，並且經常部署在惡劣環境中，因此資料的收集與保護機制益顯的重要。

近年來，軟體定義網路 (Software Defined Wireless Networks, SDN) 一直備受關注，而且它是目前與下一代中最受歡迎的通訊網路架構之一[4]。由於它使用了控制層和資料傳輸層的分離方式，可以達到動態的網路配置，靈活的網路部署與敏捷的的網路測試，這些都讓軟體定義網路成為解決以往傳統網路所無法處理與面臨的挑戰，並且提供了實際問題的解決方法。同時，軟體定義網路的相關技術也是在研究領域仍非常熱門，並且在資通訊技術（ICT）相關的工業界也有多個成功的應用案例。例如，Google 運用了軟體定義網路來部署了 B4 網路，並且達到了前所未有的 95％網路使用率[5]。

在本文中，我們專注於霧運算與軟體定義網路的整合環境架構下 (Fog Computing-SDN) 所面臨的挑戰，我們稱之為軟體定義霧聯網–主要原因是這項技術目前在研究上著墨很少，但也是最關鍵，我們特別研究在此軟體定義霧聯網的整合架構下的中間人攻擊（Man-in-the-middle attack，MITM）[6-8] 所產生的威脅，如圖一所示，在

物聯網與軟體定義網路整合環境遭受中間人攻擊。因此，我們將此問題正式化並識別潛在的攻擊場景，同時突出可能存在的問題與漏洞，並在未來會模擬與演示以顯示此類攻擊所造成的嚴重後果。因此，我們考量目前的人工智慧技術用來保護軟體定義霧聯網[9]，以免受網路安全威脅，而人工智慧技術使用在惡意封包分析的主要目的在辨識和量化各種惡意的網路行為，以分析該網路的異常行為，以及安全機制的建立。



圖一: 在物聯網與軟體定義網路整合環境遭受中間人攻擊示意圖

貳、背景

刊載於 2018 年 Journal of Security and Communication Networks 由 Heng Zhang 等四位研究人員發表: A Survey on Security-aware Measurement in SDN [10]的軟體定義網路的安全感知量測技術調查。作者透過網路性能測量技術，即鏈路延遲、可用帶寬及網路拓撲測量，進行適當的預測。網路測量利用某些方法來理解和量化網路行為，這對於提前檢測匿名行為非常有幫助。網路量測指標在制定預防措施和後續反應時非常有用，而且網路測量技術有助於實時了解網路狀態，可應用於廣泛的領域，例如網路優化，故障檢測和故障排除等。而在鏈路延遲測量方面，作者應用 SLAM 延遲監控架構，可動態發送特定探測封包，以觸發從路徑的第一個和最後一個交換器至控制器的控制封包。SLAM 基於控制器處的控制封包的到達時間戳來估計沿路徑的延遲分佈。另外，作者也應用 DPTH 的靈活而統一的封包標記方法用來表示封包進入網路的時間。

刊載於 2013 年 IEEE Transactions on Dependable and Secure Computing, 由 Chung, Chun-Jen 等四位研究人員所發表的 NICE: Network intrusion detection and countermeasure selection in virtual network systems [11]。該篇論文說明了目前攻擊雲端實體主機或虛擬主機所使用的大型分散式阻絕服務（DDoS）通常涉及較早期的攻擊行動階段，例如: 使用多重的步驟試探，低頻率的漏洞掃描，並攻擊較脆弱的虛擬機器。而這種攻擊手法尤如殭屍病毒一般，分散式阻絕服務最後透過殭屍病毒攻擊雲端主機。在雲端運算系統中，尤其像是基礎設施即服務雲（IaaS），要檢測殭屍病毒攻擊是非常困難

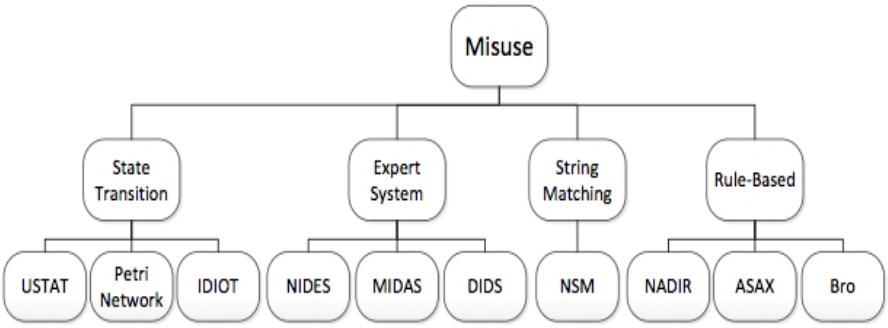
的。因此，為了要避免在雲端有更多受害虛擬主機，本文作者提出了一個多階段的分佈式漏洞檢測，測量和選擇對策，即所謂的 NICE 機制，這是建立在攻擊基礎的分析模型圖和可重構虛擬網路為基礎的對策。NICE 機制充分利用 OpenFlow 的網路的 API 界面來構建分佈式可程式化的虛擬交換機監視和控制平台，以顯著提高攻擊檢測的能力和緩解攻擊所造成的後果。 NICE 機制上主要元件被分佈和輕量化 NICE-A 在每個實體雲的服務器，這包括網路控制器，一個虛擬機分析服務器，以及攻擊分析器。最後的三個元件都位於一個集中控制中心，並且連接到每個雲服務器上的軟件的交換機（即建立在一個或多個 Linux 軟體橋接器）。NICE-A 是一個軟體代理程式將在每個雲端伺服器上執行，而與控制中心使用 OpenFLow 一條安全的專線連結。其中，網路控制是基於攻擊分析器來部署攻擊偵測機制。

刊載於 2018 年 IEEE Communications Magazine 的 Leveraging LSTM networks for attack detection in fog-to-things communications [12]的論文由由 Abebe Diro 與 Naveen Chilamkurti 兩位研究人員合著，該篇論文利用長短期記憶網路演算法進行霧對物通信中的攻擊檢測, 使用經典型的深度學習算法且已被廣泛用於網路入侵偵測上, 由於使用深度學習方法可以減輕網路入侵偵測的複雜度，除了不需要手動調整之外，深度學習還具有高偵測準確度能夠偵測與抵抗的變形攻擊。因此，在此篇論文中，作者提出應用於霧聯網之間通訊的分散式網路攻擊偵測的長短期記憶網路(Long Short-Term Memory, LSTM) 網路偵測機制。作者使用識別和分析針對物聯網設備的關鍵攻擊和威脅，特別是利用無線通訊的漏洞攻擊。兩個測試實驗證明了機器深度學習模型相對於傳統機器學習模型更有效和效率。由於霧運算是一種新的分散式平台，它為物聯網設備提供分散式和行動性支持，可以有效地解決物聯網設備的體系結構問題。因為雲計算也可為大數據分析提供彈性資源，每個霧節點都能提供基本計算和網路偵測機制，若加強物聯網所需要的安全通訊能力，就可以有效地減低竊聽和中間人攻擊的機會。

刊載於 2018 年 ACM Conference on Data and Application Security and Privacy 的論文：”Malware Analysis of Imaged Binary Samples by Convolutional Neural Network with Attention Mechanism” [13]。本篇的作者利用捲積神經網路（CNN）進行惡意軟體的行為分析，為了減少分析人員(調查樣本)的工作量，通過將捲積神經網路（CNN）與稱為 Attention 機制做技術整合，從二進制數據轉換的圖像，圖像中的分類可區分被分類出來的惡意軟體家族特有的特徵字串序列，並且可以在沒有專家知識的情況下，為人類分析者提供有用的訊息。此方法與傳統方法相比，提供了更高的分類精度。此外，基於運算的分析證實對惡意軟體的樣本在即使樣本被壓縮的情況下，過濾出來的序列也為了人工分析提供了有用的訊息。通過將二進制資料數據轉換為圖像來描述惡意軟體分類方法。並用深度學習的惡意軟體分類方法與稱為 Attention 機制，以及用於惡意軟體比較的類似序列檢測方法，將資料樣本分類為惡意軟體樣本與一般資料樣本。

參、研究方法

惡意封包與異常封包檢測隨著大數據與人工智慧取得了重要進展; 先進的封包分類算法已成為一個熱門話題在人工智慧與資訊安全的綜合研究領域之中。主要的惡意與異常封包檢測技術是基於深層的封包檢測的系統和狀態資料封包檢測為基礎的系統，再利用字串匹配來達成的檢測技術。通常使用深層封包檢測技術，諸如特徵匹配模式來分析單個的資料分組。雖然它可以有效地防止某個程度上惡意與誤用的行為。但是近期的雲端攻擊事件多屬於是組合零碎封包式的攻擊，如果只檢查一個封包攻擊時，是無法被找到。因此很多有效的深度封包檢查的算法已被提出，到目前為止，只有部份的組合式封包檢查演算法是有效的，例如重新排列來提高檢測系統的準確性，或使用防火牆系統，適用狀態檢測，網路管理員可以設置參數，以滿足特定的需求。如圖二，使用封包檢測來分類惡意封包檢測。



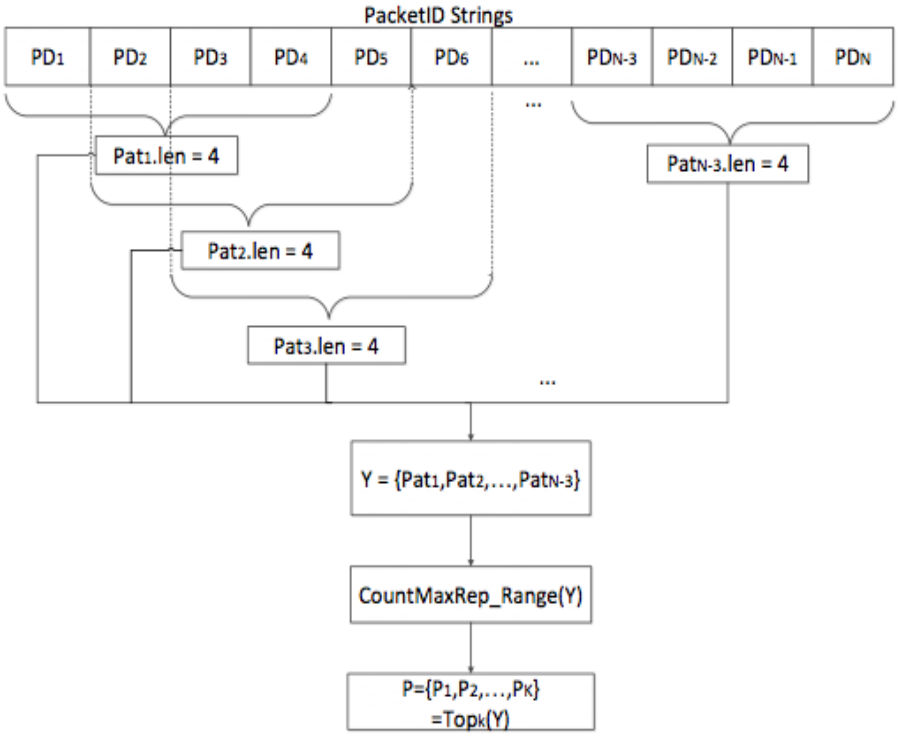
圖二：先進封包檢測演算法的分類圖

首先，我們應用確定性有限自動機(deterministic finite automaton, DFA)，接收每一個特徵模式的字串，通過映射函數跳到狀態，並做出相應的狀態計數器值加一，當在狀態計數器得到一個組值，則可以判定檢測到的資料流的相應的典型應用程式，這包含當檢測到在客戶端上運行的典型應用程序的資料流。因為特徵模式字串是從典型應用程式的資料串流中提取，可以保存會話狀態訊息，所以我們稱之為狀態確定性有限自動機。確定性有限狀態自動機從起始狀態開始，一個字元接一個字元地讀入一個字串 （這裡的指示 Kleene 星號運算子），並根據給定的轉移函數一步一步地轉移至下一個狀態。在讀完該字串後，如果該自動機停在一個屬於 F 的接受狀態，那麼它就接受該字串，反之則拒絕該字串。然而，為訓練與培養使用字串系統包含長型圖案的字符， 反而造成很大的局限性。例如，很可能會截斷完整的入侵資料串流成為幾個模式的字串，這樣做的後果是這些模式的字串不能包含一個真實有效的惡意封包進入網路的狀態。

因此，我們將原來的演算法修改成入侵封包偵測的演算法，利用通過統計的每一個可能的封包數量，我們可以發現哪些字串出現最頻繁，最高數量的字串，當然，最可能是有包含入侵程式碼，因此，我們選擇那些出現最頻繁，最有代表性的字串來訓練

與培養入侵的檢測系統。如圖三所示，Notation PD 指的是封包的 ID, Pati, Len 表示圖案的字串長度，而 CountMaxRep_Range(Y) 代表著計算每個不同的 Pati 在 Y 群組裡的重複數量。最後，使用 P 的值是用來訓練與培養入侵偵測系統，所以 P 是建立 AC Tree 的輸入資料，即:

$$Stateful\_Auto_T^i = AC\_Establish(P)$$



圖三：使用確定有限狀態自動機來執惡意封包行分類與異常檢測

如同上述，我們選擇那些出現最頻繁，最有代表性的字串轉換成二進制再轉換的圖像，圖像中的分類可區分惡意軟體家族特有的特徵字串序列。此方法與傳統方法相比應可以提供了更高的分類精度。此外，基於運算的分析證實對惡意封包的樣本在即使樣本被壓縮的情況下，過濾出來的序列也會為了人工分析提供了有用的訊息，透過將二進制資料數據轉換為圖像來描述惡意封包分類方法。

我們提出的深度學習的惡意封包分類方法結合 Ada Boost 機制，將資料樣本分類為惡意封包樣本與一般資料樣本，如下所示。Ada Boost 結合 Rectangle 特性成為辨識惡意封包的演算法。首先會有一堆從確定性有限自動機訓練並轉換出來的圖像，並分別標示著惡意封包 m 個以及正常封包 n 個。對於每一張圖像我們分別給他們 1/m 或是 1/n 的權重，端看它們是惡意封包或是正常封包。接著我們要從一大堆 Rectangle 裡面取出 T 個來。因此對下面的步驟，我們會重複 T 次。

1. 首先將所有的權重正規化加起來為 1。
2. 根據 hypotheses function，選出一個錯誤值 error 最小的特性 。
3. 紀錄可以使 error 值最小的參數。
4. 將權重值根據演算法的公式 update。
5. 最後我們求出來的使用區分器 classifier 做的結果就是由 T 個弱特性（weak feature）所組成的。

因此對於從確定性有限自動機丟入任一張圖像，就會由這 T 個特性 feature 投票，每一個特性 feature 的投票的權重都太一樣。但只有當加權值大於一半以上的所有分數時，才會認可這一張圖像為惡意封包，而其虛擬程式碼如圖四。

- Given example images $(x_1, y_1), \ldots, (x_n, y_n)$ where $y_i = 0, 1$ for negative and positive examples respectively.
- Initialize weights $w_{1,i} = \frac{1}{2m}, \frac{1}{2l}$ for $y_i = 0, 1$ respectively, where m and l are the number of negatives and positives respectively.
- For $t = 1, \ldots, T$:
  1. Normalize the weights, $w_{t,i} \leftarrow \frac{w_{t,i}}{\sum_{j=1}^{n} w_{t,j}}$
  2. Select the best weak classifier with respect to the weighted error
     $$\epsilon_t = \min_{f,p,\theta} \sum_i w_i \left| h(x_i, f, p, \theta) - y_i \right|.$$
     See Section 3.1 for a discussion of an efficient implementation.
  3. Define $h_t(x) = h(x, f_t, p_t, \theta_t)$ where f_t, p_t, and θ_t are the minimizers of ϵ_t.
  4. Update the weights:
     $$w_{t+1,i} = w_{t,i} \beta_t^{1-e_i}$$
     where $e_i = 0$ if example x_i is classified correctly, $e_i = 1$ otherwise, and $\beta_t = \frac{\epsilon_t}{1-\epsilon_t}$.
- The final strong classifier is:

$$C(x) = \begin{cases} 1 & \sum_{t=1}^{T} \alpha_t h_t(x) \geq \frac{1}{2} \sum_{t=1}^{T} \alpha_t \\ 0 & \text{otherwise} \end{cases}$$

where $\alpha_t = \log \frac{1}{\beta_t}$

圖四: Ada Boost 機制的虛擬程式碼

肆、實驗環境與結果

本節描述我們的評估測試，並就模擬實驗來討論了獲得的結果。我們使用了 Mininet 在本次的測試模擬的工作中，用於模擬軟體定義霧聯網。該網路使用 Mininet Python API 進行部署。為此，我們撰寫了一個 Python 腳本。在該腳本文件中，有一些用於在拓撲中部署 NAT 設備的指令行。此 NAT 設備充當我們的模擬霧聯網網路和遠端連線服務之間的網路閘道器。霧聯網裝置的網路 IP 地址為 10.0.0.0/8。Eth1 連接到 SDN-enabled Switch 的實體 port 3，用以複製流經 Switch 封包，我們使用兩台 PC 分別連接到 SDN-enabled Switch 的實體 port 1 及 port 2。除了會賦予 Switch 基本的封包傳輸功能，也會將流經 Switch 的封包導向至我們的 Honeypot 誘捕系統。而我們以自行修改 Mininet [15]使其可以驅動深度學習的惡意封包分類方法結合 Ada Boost 機制來自動阻擋網路攻擊。本實驗使用 Ada Boost 機制與 Mininet 協同作業來完成入侵偵測與防禦的工作，Ada Boost 機制會將封包與 Snort 定義的 rule 做比對，當發現可能的網路攻擊時，Snort 會發出 alert 至 Unix domain socket，再透過 network socket 傳送給 Ryu。並根據 alert 上的資訊寫入 database 中並下達 flow entry 至被入侵的 Switch 並加以阻擋網路攻擊。

```
priority=1,in_port=2,dl_dst=b2:2a:30:3a:e7:f2 actions=output:1
 cookie=0x0, duration=76.889s, table=0, n_packets=2, n_bytes=140, idle_age=71,
priority=1,in_port=1,dl_dst=00:00:00:00:00:01 actions=output:2
 cookie=0x0, duration=76.877s, table=0, n_packets=3, n_bytes=238, idle_age=71,
priority=1,in_port=3,dl_dst=b2:2a:30:3a:e7:f2 actions=output:1
 cookie=0x0, duration=76.875s, table=0, n_packets=2, n_bytes=140, idle_age=71,
priority=1,in_port=1,dl_dst=00:00:00:00:00:02 actions=output:3
 cookie=0x0, duration=76.870s, table=0, n_packets=3, n_bytes=238, idle_age=71,
priority=1,in_port=4,dl_dst=b2:2a:30:3a:e7:f2 actions=output:1
 cookie=0x0, duration=76.867s, table=0, n_packets=2, n_bytes=140, idle_age=71,
priority=1,in_port=1,dl_dst=00:00:00:00:00:03 actions=output:4
 cookie=0x0, duration=76.858s, table=0, n_packets=3, n_bytes=238, idle_age=71,
priority=1,in_port=3,dl_dst=00:00:00:00:00:01 actions=output:2
 cookie=0x0, duration=76.856s, table=0, n_packets=2, n_bytes=140, idle_age=71,
priority=1,in_port=2,dl_dst=00:00:00:00:00:02 actions=output:3
 cookie=0x0, duration=76.849s, table=0, n_packets=3, n_bytes=238, idle_age=71,
priority=1,in_port=4,dl_dst=00:00:00:00:00:01 actions=output:2
 cookie=0x0, duration=76.847s, table=0, n_packets=2, n_bytes=140, idle_age=71,
priority=1,in_port=2,dl_dst=00:00:00:00:00:03 actions=output:4
 cookie=0x0, duration=76.835s, table=0, n_packets=3, n_bytes=238, idle_age=71,
priority=1,in_port=4,dl_dst=00:00:00:00:00:02 actions=output:3
```

圖五：中間人攻擊前的 OpenFlow 交換機流規則表

伍、結論

本論文主要探討軟體定義霧聯網下的網路攻擊的研究，使用人工智慧分析，我們必需要清楚明白霧聯網與軟體定義網路的運作原理，才能洞察如何可以改善霧聯網與軟體定義網路的安全通訊機制，進而才可以再作頻寬效率的提昇。雖然軟體定義霧聯網可以比目前網路大幅減少頻寬的浪費，但若沒有達到穩定與安全性的通訊目標，仍會疑慮有適用的問題，特別是對於系統業者而言，若霧聯網與軟體定義網路的傳輸安全性與網路安全性無法確保，則因此導致使用者對於霧聯網與軟體定義網路的信心與意願度會降低。在本論文中，我們提出的深度學習的惡意封包分類方法結合 Ada Boost 機制達成封包安全檢測機制。未來，我們會將實驗的部份做更進一步的分析中間人攻擊的行為並且考慮透過靜態分析應用程序的程序集源代碼來檢索更多功能。我們認為基於權限的分類可以成為檢測惡意應用程序的良好輔助手段。

參考文獻

[1] A. Akhunzada, A. Gani, N. B. Anuar, et al., „Secure and dependable software defined networks“,in Proceedings of the Journal of Network and Computer Applications, vol. 61, pp. 199–221, 2016

[2] A. O. Oluwasogo and O. Oluwaseyitan, “A security architecture for software defined networks (SDN)“, in Proceedings of the International Journal of Computer Science and Information Security, vol. 13, no. 7, 2015

[3] Christian Banse and Sathyanarayanan Rangarajan “A Secure Northbound Interface for SDN Applications “in Proceedings of the 14th IEEE International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom), pp. 413–424, 2015.

[4] K. Bakshi, “Considerations for Sofware Defned Networking (SDN): Approaches and use cases,” in Proceedings of the 2013 IEEE Aerospace Conference, AERO 2013, USA, March 2013.

[5] L. Popa, P. Yalagandula, S. Banerjee, J. C. Mogul, Y. Turner, and J. R. Santos, “ElasticSwitch: Practical work-conserving bandwidth guarantees for cloud computing,” in Proceedings of the Annual Conference of the ACM Special Interest Group on Data Communication on the Applications, Technologies, Architectures, and Protocols for Computer Communication, ACM SIGCOMM 2013, pp. 351–362, August 2013.

[6] M. Bari et al. , “Data Center Network Virtualization: A Survey, “in Proceedings of the IEEE Commun. Surveys & Tutorials , vol. PP, no. 99.

[7] M. Jammal, T. Singh, A. Shami, R. Asal, and Y. Li, “Software defined networking: State of the art and research challenges“,in Proceedings of the Computer Networks, vol. 72, pp. 74–98, 2014

[8] M. R. Nascimento et al. , “Virtual Routers as A Service: the RouteFlow Approach Leveraging Software-Defined Networks,” Proc. CFI , Seoul, Korea. ACM, 2011.

[9] Mininet Team Mininet: An Instant Virtual Network on your Laptop. Available online: http://mininet.org/ (accessed on 6 April 2019).

[10] N. McKeown et al. , “OpenFlow: Enabling Innovation in Campus Networks,” in Proceedings of the SIGCOMM CCR , vol. 38, no. 2. ACM, 2008, pp. 69–74.

[11] O. N. Foundation, “Openflow switch specification“, 2011.

[12] R. A. Alhanani and J. Abouchabaka, “An overview of diferent techniques and algorithms for network topology discovery,” in Proceedings of the 2nd World Conference on Complex Systems, WCCS 2014, pp. 530–535, Morocco, November 2014.

[13] Sushant Jain, Alok Kumar, Subhasree Mandal, et al.,.“B4: Experience with a globally-deployed software defined WAN “, in Proceedings of the ACM SIGCOMM Computer Communication Review, vol 43(4), pp. 3-14, August 2013

[14] W. Braun and M. Menth, “Software-defined networking using openflow: Protocols, applications and architectural design choices“, in Proceedings of the Future Internet, vol. 6, no. 2, pp. 302– 336, 2014

[15] W. Li, W. Meng, and L. F. Kwok, “A survey on openflow-based software defined networks: Security challenges and countermeasures,” in Proceedings of the Journal of Network and Computer Applications, vol. 68, pp. 126–139, 2016