一種SDN中基於使用者的驗證機制實作簡介

Introduction of an Implementation of an User-Based Authentication Mechanism Using SDN

陳厚任　王定山　李光益　楊慧卿
Hou-Ren Chen, Ting-Shan Wong, Kuang-Yi Li, Hui-Ching Yang

中文摘要

隨著SDN發展及企業級部署之需求日益增加，SDN解決方案的安全控管能力成為了不可或缺的要件，而可攜式裝置、筆記型電腦與BYOD(Bring your own device)風潮也使得以IP位址或者網路卡MAC資訊等靜態資訊做權限控管的方式漸趨不敷使用。本研究實作了一個使用者驗證模組，透過802.1X協定，實現SDN網路之使用者驗證功能，提供未來網管系統使用者規劃更多策略、差異化需求的可能性，並提出實際上可用的情境做為範例。

Abstract

Due to the fast developing of SDN technology and the increasing demand of enterprise-level deployment, the security management is an indispensable component in a SDN enterprise solution. Also, using static information such as IP or MAC address of network card as authentication target is also no longer sufficient according to the current portable devices, notebooks in BYOD trend. In this research, we implemented a core module of SDN Controller to cope with user authentication, which based on 802.1X protocol. With our module, network manage system user can acquire more possibility of strategic planning and demands separation. Last but not least, we made up several scenarios to demonstrate the functionality and feasibility of our system.

關鍵詞(Key Words)

軟體定義網路(Software Defined Networking；SDN)

企業網路(Enterprise Network)

使用者驗證(User Authentication)

802.1X

1．前言

SDN (Software Defined Networking)將網路設備的控制層(control plane)與資料層(data plane)功能分離，將傳統的嵌入式控制軟體移出到交換器設備外的Controller，亦即資料層的所

有網路硬體設備皆聽命於集中控制層的軟體SDN Controller，透過Controller協同運作，搭配智能化SDN Application，進而實現相關應用領域的解決方案。為了提供創新多樣化的應用服務，SDN允許管理者及開發者自行依控制平台所提供的北向介面來存取網路狀態、資源與服務，建立符合自己需求與差異化的SDN應用服務。

但隨著網路軟體化趨勢的發展，SDN網路日益增大，可攜式裝置、筆記型電腦與BYOD(Bring your own device)風潮也使得以IP位址等靜態資訊做權限控管的方式漸趨不敷使用。本研究目的為提供User-based網路管理機制，在SDN網路中，發展基於802.1X的使用者驗證模組。

擁有使用者驗證能力之SDN網路，便能有效利用SDN網路可完全控管與可程式化網路的特色，相較於傳統網路，更能夠在企業複雜的實體網路上，給予不同的網路資源與相關服務策略，如頻寬保證、限制等、網路安全管理政策(如存取允入、允出、對外連線規則等)，以滿足企業對不同使用者族群(如內部人員Staff、訪客Guest等)使用網路資源及網路存取控制的差異化管理需求，同時也提供SDN網路更好的自我保護能力。

在本文中，我們首先在第二章簡介ESMES(Enterprise SDN Migration for Edge Switch)[1]之Controller中使用者驗證相關執掌的模組；第三章就802.1X及我們實作的過程做進一步的說明，並於第四章概述我們目前實驗的成果；第五章則介紹兩個搭配Policy Manager模組可能的應用情境，最後在第六章為本文進行總結。

2 · ESMES Controller介紹

2.1 ESMES Controller簡介

OpenDaylight[2](ODL)為一個由Linux foundation主持的開放原始碼SDN controller專案，並有諸多大廠(如Cisco)推進其開發，目前已是工業界解決方案中市占率相當大的一個標準。

ESMES Controller即本團隊奠基於此平台上再做發展之產品，新增許多企業級網路佈建時所需之特色(包含此篇所述之使用者驗證模組)，增加其產業上的競爭力，以實際產品化為目標來推進專案。

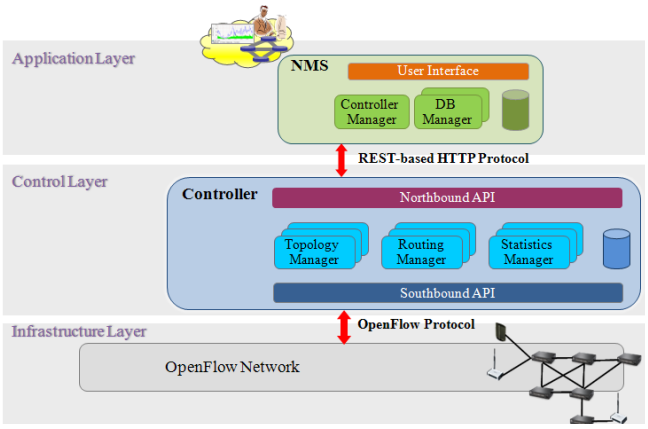

圖 1 ESMES Controller架構圖

ESMES Controller如同其他SDN Controller，主要以集中式的方法管理Switch之間的溝通與行為，藉以控制封包運行、達成實質管理網路的目標。架構如圖1所示。主要分為Controller及NMS兩個組件，為各自獨立的可執行實體(entity)，分別屬於SDN整體架構的控制層(Control Layer)及應用服務層(Application Layer)。

系統內各組件包括：

(1) Controller組件：為一模組化的SDN控制器，除了負責網路拓樸、繞送路徑處理等多個核心控制模組、服務等智能模組之外，北向(northbound)介面提供RESTful API供應用層開發相關服務，南向(southbound)介面則以plugin方式支援OpenFlow協議[3]。

(2) NMS組件：為一Web-based的網路管理介面，其核心模組透過RESTful API與Controller組件溝通，並支援圖形化的操作界面，供使用者管理與監控整體網路運作。

詳細的功能介紹可參考[1]之說明。

2.2 Controller組件規格

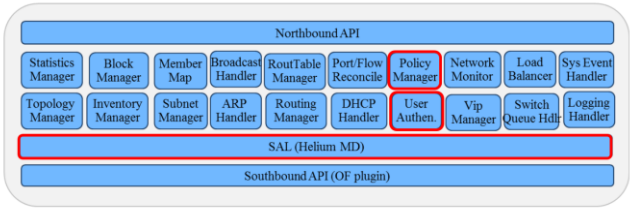

圖 2　Controller 組件架構

Controller組件為SDN架構中的控制中樞，由數個核心模組組成，如圖2所示。組件內與本研究相關的模組包括：

(1) User Authentication Handler：即本研究實作之部分。擔任Authenticator角色(稍後介紹)，負責處理host (supplicant)與後端Authentication Server透過802.1X[4]協定進行使用者身分驗證，以確認使用者的合法性與權限，同時儲存相關資料以供其他模組查詢使用者身分。
後端Authentication Server目前支援RADIUS (Remote Authentication Dial In User Service) Server。

(2) Policy Manager：根據網管人員預先設定的規則(例如IP與port組合或使用者代碼等)分群，規劃每個群體的網路存取規則。規則可調變的屬性可包含packet允出允入、時間段、規則失效時間等等。

(3) SAL (Service Abstraction Layer)：負責Controller組件中樞運作，將Controller內的架構分成Network Function/ Northbound以及OF Plugin/ Southbound兩個layer。SAL提供Model-Driven運作模式，為一個類似OSGi (Open Service Gateway Initiative)[5]的實作架構，讓模組開發者透過撰寫標準化模組語言(Modelling Language) YANG files來定義並產生API，再依照API的形式提供implementation，以支援使用者(模組開發者)自行定義、實作API，不需受限於原先API-Driven架構下SAL既有提供的介面，使其擁有較高的自由度與彈性。

2.3 組件的模組介面

Controller使用YANG model以建構及定義模組之間相互叫用或實作的介面，以及可供共用的資料。包括：

(1) RPC (Remote Procedure Calls)：提供給其他模組直接叫用的介面，由實作此介面的模組提供實際的功能內容。

(2) Notification：支援的事件種類，提供事件通知的模組將在適當時機進行事件發布，由實作此介面的模組接收事件通知。

3．驗證過程及實作

3.1 802.1X 介紹

802.1X為一IEEE所制定之標準，為一Port-based network access control (PNAC)，屬於IEEE 802.1網路協議群之一份子。其機制提供裝置欲接入Ethernet或WLAN時之驗證機制。

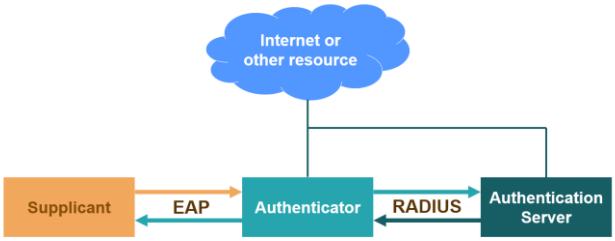

圖 3　802.1X架構圖

在802.1X標準中，有三個角色在互動，如圖3所示：

(1) 申請者 (Supplicant)：欲連接至網路的host(device)。

(2) 驗證者 (Authenticator)：負責接收申請者之request，並轉送至驗證伺服器。為兩者之間

的橋梁。通常都由Switch擔任，申請者在驗證成功之後即可連接到Authenticator背後的網路。實際認證的最小單位為網路介面，故若一個switch的某port同時有多個host界接或一host擁有多個VM(Virtual Machine) instance模擬實際的host界接至同一個port時，每個host或VM仍可擁有不同的驗證成果。

(3) 驗證伺服器(Authentication server)：根據Supplicant所提供的資訊並驗證其是否為真正具有該身分的人。

在802.1X中，運用到了兩個協定：

(1) EAP (Extensible Authentication Protocol)[6]：在無線網路或點對點傳輸中普遍被使用的驗證框架，僅定義訊息要有的格式，而訊息的內容則由實作它的協定決定。常用的EAP method包括EAP-MD5, EAP-TLS, PEAP等。此篇研究中使用EAP-MD5來做實作。

(2) RADIUS (Remote Authentication Dial In User Service)[7][8]：設計來集中處理Authentication, Authorization及Accounting之protocol。在應用層中，使用UDP或TCP來傳輸RADIUS之Client/ Server的訊息。

3.2 使用者驗證過程與其實作

在過去，802.1X的Authenticator都會由擁有處理此protocol行為能力的switch擔任，但在SDN的架構中，switch已被去智慧化，故此工作便須由Controller之使用者驗證模組承擔。透過802.1X驗證方式協助連進的host進行身分驗證，並且在host成功驗證後記錄此host的相關資訊，提供其他模組查詢。以下介紹此模組協助802.1X驗證的程序。

3.2.1 Phase1: Host連接上Switch

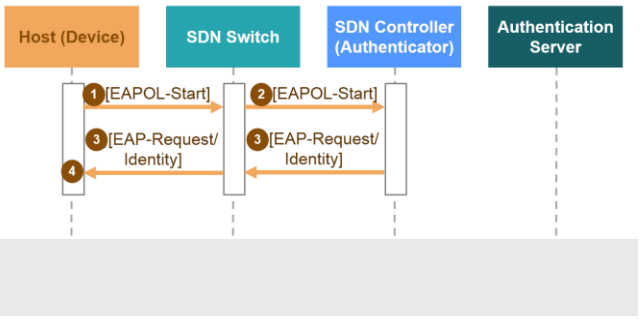

圖 4 Phase 1 程序圖

參考圖4：

(1) 當有一台host連接上OF switch並嘗試對網際網路進行存取時，選用802.1X驗證方式的host會發送EAP封包(EAPOL-Start)進switch來要求接入網路。一般封包也會觸動整個驗證流程，成功與否則端看host是否能回應第三步的EAP-Request/ Identity封包。

(2) Switch收到請求並上送(packet-in) Controller。

(3) Controller會要求Switch發送EAPOL訊息(EAP-Request/Identity)給該host，嘗試與該host進行EAP驗證。

(4) 若host不支援802.1X驗證方式，或一段時間內沒有收到host送出之EAP-Response/Identity封包，則此波驗證流程即在此結束，待由host端再次觸發。

3.2.2 Phase2: Host透過Controller向Authentication server進行驗證

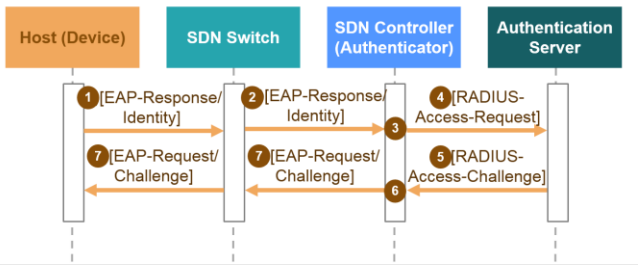

圖 5 Phase 2 程序圖

參考圖5：

(1) 若此host支援802.1X驗證方式，則會送出EAP-Response/Identity封包來告知自己的身分。

(2) Switch此時將封包上送(packet-in)給Controller之使用者驗證模組。

(3) 模組收到host回覆之封包後，透過我們撰寫的函式庫將其轉換為RADIUS訊息(RADIUS- access-request)

(4) 訊息透過UDP將RADIUS訊息發送至後端Authentication Server，進行802.1X規範之驗證行為。

(5) 後端的Authentication Server收到Controller送出的RADIUS-Request封包後，會根據此封包中夾帶的驗證User Name來發送RADIUS-access-challenge驗證訊息回Controller，此訊息是authentication Server用來驗證host是否與其允許的身分條件相同。在本系統所使用的情境下(EAP-MD5)，即希望某host提供與他User Name相對應的密碼。

(6) Controller收到訊息後，於使用者驗證模組將此RADIUS格式之訊息拆解並轉換成為EAP格式的訊息(EAP-Request /Challenge)封包

(7) 封包透過Switch將發送給待驗證的host。

3.2.3 Phase 3: 比對驗證標的與告知結果

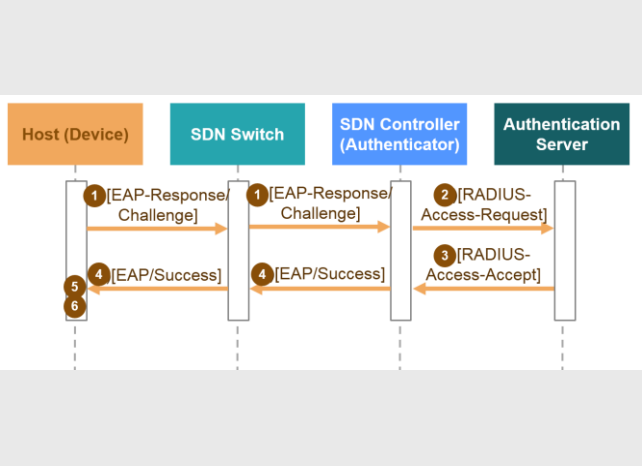

圖 6 Phase 3 程序圖

參考圖6：

(1) 待驗證的host收到此EAP-Request/Challenge封包後，根據其中夾帶的訊息回傳EAP-Response/Challenge身分回應EAP-Response/Challenge訊息。由於不同的作業系統於此處的行為會有些微的差距，我們採取了最精準的實作方式，並實際驗證其回覆正確性。

(2) Controller之使用者驗證模組透過Switch收到此訊息後，將此EAP訊息做封裝以產生出所需的RADIUS格式訊息(RADIUS-access-request)，再發給後端Authentication Server。

(3) 後端Authentication Server收到Controller送出的RADIUS-access-request且判斷該host為合法用戶時，Authentication Server會送出RADIUS-access-accept訊息給Controller之使用者驗證模組。

(4) Controller之使用者驗證模組收到此訊息後將其轉換成EAP-success訊息並發送給目標host，通知其通過身分驗證。

(5) 在host接到自Switch傳送的EAP-Success同時，即為驗證完成。

(6) 同時，使用者驗證模組會發送(publish) host驗證成功的notification通知其他同在Controller中的模組，並將此host的身分資訊寫入一模組共享的資料空間(稱為DataStore，存放在SAL的中央資料庫)，以便維護、管理，並且提供其他模組查詢。

至此，雙邊的驗證即告一段落，驗證成功。

3.2.4 額外行為

除了接入網路中的的host所送進的packet-in外，其他模組也可透過RPC或是RESTful API的方式來啟動802.1X驗證程序、或是清除DataStore的資料。如其他模組偵測到網路線被拔除，便可請使用者驗證模組將記錄刪除。

每隔一段時間，模組也會自動要求已經驗證且連接的使用者重新進行驗證，若使用者皆無法在過期前成供驗證，模組便會通知其他模組將此使用者之連結取消並清除DataStore之紀錄。

4 · 實驗

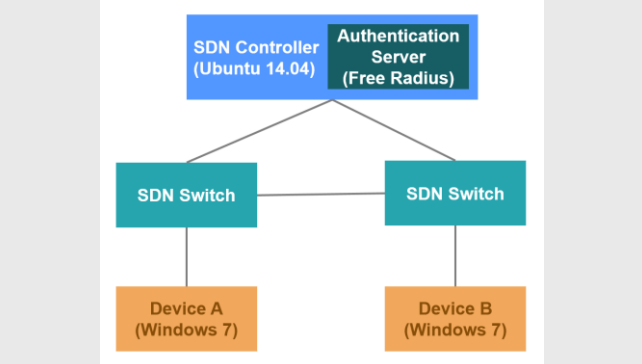

圖 7　硬體架構圖

參考圖7。實驗旨在驗證Device A/B兩者之間的通訊行為是否能正確依照驗證的程序而改變。實驗的SDN controller(即ESMES)架於Ubuntu14.04，上方再架設一Authentication Server (採用FreeRADIUS [7]這個open source的RADIUS server解決方案)，底下連接兩台SDN Switch (Edge-corE-4600)，Switch再各自連接一台運行Windows7的host (Device A/B)。

在實驗一開始，Device A, B會進行互ping，因無法尚未通過驗證，ICMP封包在Packet-in到Controller時就會被捨棄。此時，我們將Device A/B完成驗證，兩者之間便可打通。

接著，我們將停用Device A的網路卡，Device B便再度ping不通。在Device A輸入錯的使用者資料後，ping仍然無法打通，直到正確的使用者帳號密碼使整個驗證流程成功結束後，連線才能再度建立。

5 · 應用

除了安全性的保障外，使用者驗證的模組也可搭配Policy Manager模組以達成更靈活的權限控管。目前的ESMES中，Policy Manager模組已經完成實作，以下簡述兩個情境：

5.1 不同使用者權限不同

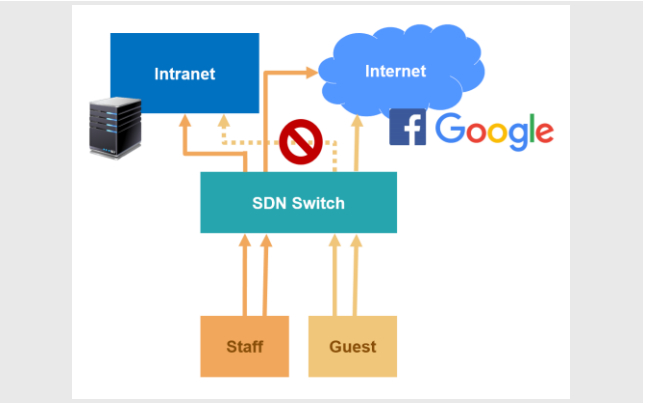

圖 8　情境5.1示意圖

參考圖8，例如在企業中，使用者的裝置連上網後，會根據其是否為企業內部人士區分為兩群：員工(Staff)與訪客(Guest)。員工可瀏覽企業內部網路與外部網路，但與此同時，訪客只能瀏覽外部網路，瀏覽內部網路的要求會被阻擋。傳統網路的switch登入只能區分是否能夠連線，但若配合SDN及我們的系統，便有可達成這樣細膩的管控。

5.2 同一使用者，不同時段權限不同

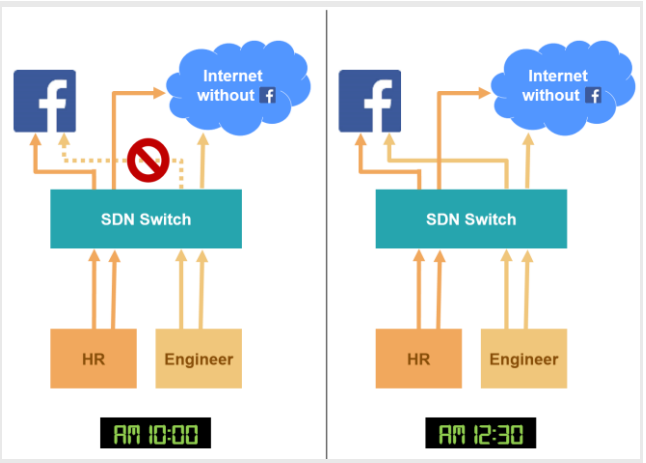

圖 9　情境5.2示意圖

參考圖9，例如對於公司中的員工經由驗證登入後，在上班時間封鎖社群網路，讓員工能夠更專心致志在工作上，中午休息或下班之後開啟等。Policy Manager模組可以根據上下班的時間段規劃不同的網路服務策略(policy)，使用者驗證模組則可針對不同使用者採用不同網路服務策略。例如，職掌招募的HR為了尋找履

歷，其身分不會受封鎖的影響。

這邊簡述兩個可能的使用例，當然系統本身的彈性可讓使用者有更多的發揮，端看使用者的需求為何。

6 · 結論

SDN之發展及企業級部署之需求中，針對使用者的安全控管能力必不可少。擁有使用者驗證能力之SDN網路，除了安全性外，更能有效利用SDN網路可完全控管與可程式化網路的特色，能夠在企業複雜的實體網路上，給予不同的網路資源與相關服務策略，如頻寬保證、限制等，以滿足不同使用者族群給予網路資源的差異性及網路存取控制的管理解耦合需求。

本研究完整實作了使用者驗證模組，強化了我們的ESMES Controller，使其朝完善的SDN企業級解決方案更進一步。同時，本文也簡介了實作使用者驗證於SDN的方法，並針對驗證過程中的每一步予以說明，提供欲實作此protocol者約略的藍圖。最後搭配Policy Manager，提出了兩種可能的情境來演示此模組的可能性。在未來，此研究的成果也會持續與controller的其他模組與操作介面做整合，使其功能得到最大的發揮。

參考文獻

[1] 何哲勳等人。Mar. 2015。基於SDN架構之企業網路容和技術與試驗場域布建。電腦與通訊 No.161：5-12

[2] OpenDaylight, https://www.opendaylight.org/

[3] OpenFlow: Enabling innovation in campus networks, Nick McKeown等人。Apr.2008。ACM Communications Review.

[4] IEEE 802.1X, http://standards.ieee.org/getieee802/download/802.1X-2004.pdf

[5] OSGI, https://www.osgi.org/

[6] Extensible Authentication Protocol (EAP) [RFC 3748], https://tools.ietf.org/html/rfc3748

[7] RADIUS (Remote Authentication Dial In User Service) Support For Extensible Authentication Protocol (EAP) [RFC3579], https://tools.ietf.org/html/rfc3579

[8] Remote Authentication Dial In User Service (RADIUS) [RFC2865], https://tools.ietf.org/html/rfc2865

[9] FreeRADIUS, http://freeradius.org/

作者簡介

陳厚任

工研院資通所網路通訊服務技術部副工程師。台灣大學資訊工程研究所碩士，專長為網際網路通訊技術、SDN技術、人機介面等。

王定山

現就讀國立中正大學通訊工程研究所博士班，專長為網際網路通訊技術、SDN技術等。

李光益

國立中正大學通訊工程研究所博士候選人，專長為軟體定義網路、節能通訊網路、雲端中心網路技術和高存活網路設計等。

楊慧卿

工研院資通所網路通訊服務技術部資深工程師。專長為網際網路通訊技術、P2P串流技術、SDN技術等。