SDN-Driven Authentication and Access Control System

Vainius Dangovas and Feliksas Kuliesius
Faculty of Physics
Vilnius University
Sauletekio av. 9, bld. 3, Vilnius, Lithuania
feliksas.kuliesius@ff.vu.lt

ABSTRACT

Attempts to secure the enterprise network even when using strong AAA (authentication, authorization and accounting) schemes meet the user box spoofing and security middle boxes (firewalls and other filtering tools) bypassing problems. To strengthen the network security level, the names (users, addresses) and user machines must be bound tightly to the unambiguously defined network appliances and its ports. Using traditional network architecture these solutions are difficult to realize. The SDN framework allows to solve these problems more sharply and securely.
The AAA design based on the Software Defined Networks (SDN) structure is presented in this paper. It is shown that it is possible to reuse the pre-existing authentication and account infrastructure as well as part of network hardware and warrant the tight binding of user/device to the topology.

KEYWORDS

SDN, OpenFlow, AAA, security, networking

1 INTRODUCTION

Well known attempts to secure the enterprise network [1,2], especially in multivendor and complicated environments, even when using strong AAA (authentication, authorization and accounting) schemes meet the user box spoofing and security middle boxes (firewalls and other filtering tools) bypassing problems [3]. To strengthen the network security level, the names (users, addresses) and user machines must be bound tightly to the unambiguously defined network appliances and its ports. Using traditional network architecture, aiming to support constantly changing network state, these solutions need careful and continuous maintenance and do not exclude possibility of human configuration errors. Otherwise, the SDN framework allows to solve these problems more sharply and securely.
The AAA design based on the Software Defined Networks (SDN) structure is presented in this paper. It is shown that it is possible to reuse the pre-existing authentication and account infrastructure as well as some of the network hardware and warrant the tight binding of user/device to the topology.

2 SECURE NETWORK DESIGN

The new paradigm of network set-up, that intensively explored during last decade is Software Defined Networks (SDN) [4]. The principal idea of SDN architecture is complete decoupling of the control plane from the data forwarding plane. The control unit – controller, i. e. control elements existing in every distributed network device – is decoupled from the data forwarding devices and centralized. This means that decision-making is concentrated in a single (or redundant) place – controller. The data forwarding is based on flow rules defined by controller instructions. Decisions of the controller can be made on much more various different criteria (as compared with conventional networks) as well as on the pre-programmed rules. The secure control channel is used to settle flow tables of the data plane switches (generally based on OpenFlow [5] protocol). Such a feature set

allows the centralized, smart control and inspection of data packets which can be forwarded by any SDN switch attached to network.

We suggest to combine SDN features with existing network architecture. When analyzing the data traffic and users behavior it is possible to state that the place where the introducing of SDN switches should decrease the possible vulnerability of the traditional network in the most effective way is the access layer. If the access layer switches are unambiguously authenticated, mapped to the network structure and faithfully tracked, it is possible to bind users and their boxes to the appropriate switch/port definitely and effectively control their traffic policy. Additionally, such architecture allows existing user authentication schemes to be used, without modifying the intrinsic SDN elements structure. So, pursuing to secure the network, it is enough to deploy SDN capable switches into the access layer. In the upper layers, SDN supported appliances (switches and routers) can be introduced into existing network gradually. Appending the network with SDN appliances up to the full scale should augment the security empowering the system to construct fully-managed topology with distinguished and predictive flows for all users and services.

The proposed topology of enterprise network is presented in Fig. 1.

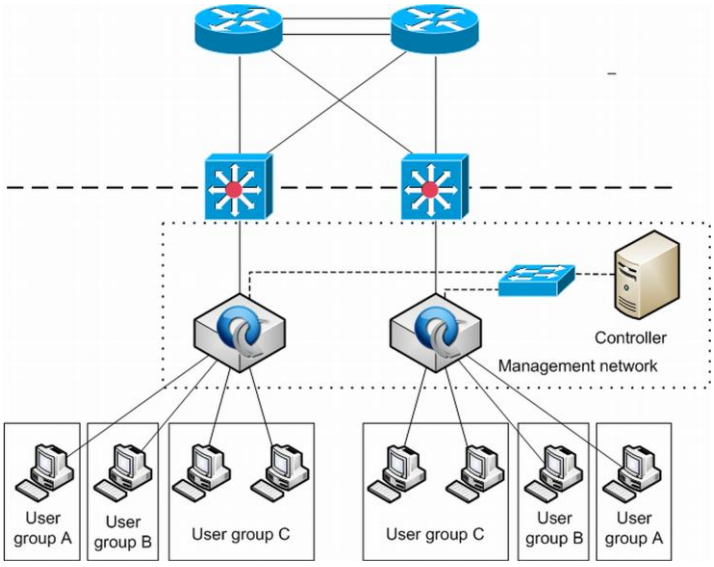

Figure 1. The simplified network topology

3 HARDWARE AND SOFTWARE

As stated earlier, the access layer is based on SDN capable forwarding-only switches, the upper layers – on the traditional appliances. Since SDN switches do not need to carry out most of the tasks which ordinary Ethernet switches must to solve, i.e. they do not hold MAC, ARP, routing tables, do not need processing capacities to learn MAC addresses and routing entries etc., it is possible to replace specialized switches with simple commodity ones. In this case Linksys WRT54GL wireless routers with Pantou OpenWRT software [6] were used. Pantou is based on the BackFire 10.03 OpenWRT release (Linux 2.6.32). This OpenWRT package includes the OpenSwitch (virtual switch) and OpenFlow modules which run as applications transforming the commodity router into an SDN enabled switch. To secure network performance these switches can be replaced by any other OpenFlow switches without changes in other components.

The clients were running as virtual machines in a VM ESXi hypervisor. Each of the clients had dedicated physical interfaces, which were connected to the access switches directly.

The open source platform controller Floodlight [7] was used in our deployment. Floodlight is a modular controller written in Java programming language. Its core modules are responsible for connections to OpenFlow capable switches, information gathering to form a full view of the active topology, management of local storage structures, computing threads and interpreting OpenFlow protocol packets, which are sent from the switches. The Floodlight controller (version 0.90) was installed as a virtual machine and has been running on a HP ProLiant DL140 server with Ubuntu 12.04 desktop OS on VMWare ESXi v5.5 hypervisor platform.

4 ACCESS CONTROL

Due to the fact that the controller modules we needed were not available, we coded the controller modules which would provide the logics for packet forwarding decisions (including the authentication and binding of the switches themselves) and could implement a sophisticated user access control based on the users and their boxes identity. We modified the controller to have the ability to control and manage all nodes in the testbed regarding our design: 1) to register and authenticate the switches to the controller; 2) to authenticate the hosts and to bind them to the switches (and ports); 3) to provide the authentication of users; 4) to manage data flows and user/host mobility.

Most of the reasonable network access control methods have limitations in one of these three areas: user mobility, determination of client rights, the level of solution complexity. Static definitions of end users in access devices, for example, access lists or MAC address lists are cumbersome and resource consuming if the user must be defined in all network access devices and limit user mobility if they are not distributed across the network. Furthermore, introduction of these rules brings serious difficulties in the client addressing scheme. More sophisticated network admission control solutions require client software (such as CNAC, NAP, TNC) to be capable of communicating in a way that is defined by specific tunneling protocols [3,8]. Such methods are even more difficult to manage since it includes end device software installation and support.

The network access control method, which is described in this paper uses a common RADIUS server based authentication and communication between an authentication engine and the controller (including an authentication interface, as described below).

Any flow from a newly connected device is redirected to the controller and is provided with default forwarding rules allowing ARP, DHCP, DNS and authentication traffic only. DHCP forwarding should be always directed through the controller to collect the user's IP address.

Installation of default traffic flows provides means for a client only to authenticate itself, and any other traffic is discarded.

The process of authentication should be accomplished as easily as possible in order to be attractive for everyman end user. Therefore, the web-based authentication interface was implemented, which should form an authentication request to the RADIUS authentication server and forward the response to the controller following dot1x procedures. The answer may or may not contain the client's IP address, but the essential information is the client's membership in a group. The RADIUS user attributes used for the VLAN ID assignment are: 1) IETF 64 (Tunnel Type)_Set this to VLAN, 2) IETF 65 (Tunnel Medium Type)_Set this to 802, 3) IETF 81 (Tunnel Private Group ID)_Set this to VLAN ID [9]. The group is declared as a VLAN ID, which should be applied to the port, but it can be used for another purpose in the next steps as well. Lastly, the controller updates its active user database and sends OpenFlow control messages to settle traffic flow entries to the switch. The controller remains separated from the authentication server and the authentication interface for security reasons, and therefore should not be possible for a network client to reach the controller.

The most important tasks for the controller are to manage the associations of client network items and its membership to a policy group and to apply these policies by making forwarding decisions.

To store required information hashed key databases of a dynamic capacity were implemented on the Floodlight controller: the active users, the user rights policy and the data flow counter databases. The most important of them was the database of active users with the MAC addresses used as a primary key value. The entries of this database contain the IP address, attachment switch ID and the interface that receives packets from the client. This set of

values uniquely and completely describes an end user device and its location in the network topology. By iterating over the values of this database the controller is able to find any attached client and form a forwarding decision to reach it.

After the client authenticates his identity the normal forwarding process starts. It includes the creation of a traffic flow counter (which holds a number of non-default flows installed for that client) and checking in previously mentioned databases: 1) firstly, the user group policy database is checked if the client belongs to a group that is allowed to reach the destination; 2) if the answer from the first database is positive, the destination MAC address is queried in the active user database. A found entry defines the forwarding decision. If the destination address is local to the switch, the controller provides a rule to forward traffic through the interface of the same switch, otherwise the traffic is destined for the default gateway or simply must be switched by a multiprotocol switch in the upper layer of the topology. An entry in the active users database and all related traffic flow entries in a switch are deleted if one of these events occurs: 1) the controller is informed that a port, which was bound to that client, changed its state or 2) a counter that is responsible for counting the installed traffic flows for a client decrements to a zero value.

A separate set of logics must be applied to uplink interfaces. Since data packets may not leave the layer 2 domain, when passing the aggregation layer, the uplink interfaces must not collect information from packets that are not from the default gateway of layer 3 interfaces. For this reason a predefined set of IP addresses that are used by the default gateways must be given to the controller. If one of these IP addresses is seen on an OpenFlow switch interface, its MAC address should be stored in the same active user database, but with a mark, which would allow deleting the entry only if that interface is shutdown or a connection to the uplink device is lost

5 CONCLUSIONS

The secure SDN-controlled network prototype for medium size enterprise, based on commodity access switches, has been developed and tested. The proposed system possesses strong authentication, authorization and accounting as well as policy management characteristics, compatibility with the traditional network structure and the possibility to introduce the SDN segment gradually.

The controller modules capable of controlling and managing all nodes in the testbed, regarding our design: 1) to register and authenticate the switches to the controller, 2) to authenticate the hosts and to bind them to the switches (and ports), 3) to provide the authentication of users, 4) to manage data flows and users/hosts mobility, were developed.

6 REFERENCES

[1] H. Okhravi and D. Nicol, “Applying trusted network technology to process control systems”, in IFIP Int. Federation for Information Proceessing, Vol. 290, Critical Infrastructure Protection II, M, Papa and S. Shenoi, eds., Springer, pp. 57-70, 2008.

[2] 802.1X: Port-Based Authentication Standard for Network Access Control (NAC), Juniper Networks, 2010.

[3] S. Parhi, “Attacks Due to Flaw of Protocols Used In Network Access Control (NAC), Their Solutions and Issues: A Survey”, Int. J. of Computer Network and Information Security, Vol. 4, No. 3, pp.31-46, 2012.

[4] W. Stallings, “Software-Defined Networks and OpenFlow”, The Internet Protocol Journal, Vol. 16 No. 1, pp. 2 – 14, 2013.

[5] N. McKeown, T. Anderson, H. Balakrishnan, G. Parulkar, L. Peterson, J. Rexford, S. Shenker and J. Turner, ”OpenFlow: Enabling Innovation in Campus Networks“, ACM SIGCOMM Computer Communication Review, Vol. 38, No 2, pp. 69-74, 2008.

[6] Y. Yiakoumis, “Pantou: OpenFlow 1.0 for OpenWRT”, www.openflow.org/wk/index.php/Pantou_:_OpenFlow_1.0_for_OpenWRT.

[7] Floodlight. A Java-based OpenFlow Controller. http://floodlight.openflowhub.org, 2012.

[8] B. Czarny, “Network Access Control Technologies”, OPSWAT White Paper, October, 2008, http://www.opswat.com/sites/default/files/Network_Access_Control_Technologies.pdf.

[9] G. Zorn, B. Aboba and D. Mitton, “RADIUS Accounting Modifications for Tunnel Protocol Support”, RFC 2867, 2000, http://tools.ietf.org/rfc/rfc2867.txt.