Integration of Network Services in Tactical Coalition SDN Networks

dr. Anders Fongen, prof. Mass Soldal Lund, nov 2020
Norwegian Military University College, Cyber Defence Academy, Lillehammer
email: anders@fongen.no

SECURWARE 2020, Valencia, Spain

Presenter's bio

Anders Fongen

- Associate Professor, Norwegian Military University College
- Field of research: Software Defined Networking, Networking security
- PhD in Distributed Systems, Univ. of Sunderland, UK, 2004
- Career history
  - 4 years in military engineering education
  - 10 years research in military science (Chief Scientist)
  - 8 years in civilian college (Associate professor)
  - 11 years in oil industry
  - 6 years in electronics industry

Introduction

- Software Defined Networking (SDN) offers great opportunities to tactical networks, but calls for a different design than old fashioned networks.
- We will present our efforts to exploit those advantages

- Tactical Network = mobile and temporary network used in military operations, using wireless links to a large extent.
    - links are a scarce resource, should be employed well
    - coalition partners wish to keep their traffic separate
    - cyber-enemies are resourceful and perseverant

Current problems related to IPv4

- Frequent reconfiguration requires re-allocation of IP addresses
- Link layer measures, like VLAN, must be aligned with IP subnetting
- Traffic policing requires interoperable use of TOS labels
- Authentication of devices normally based on MAC addresses

SDN enables a different design:

- Layer 2 networking principles (network layer agnostic)
- Link cycles allowed
- COI separation (like VLAN) without port configuration
- End system authentication
- Whitelisted flows for attack protection

Chosen components

- SDN Switch (Network Element): OpenVSwitch
- SDN Controller: Ryu
- Platform: VirtualBox
- End nodes and switching nodes were all full Linux instances

Experimental network

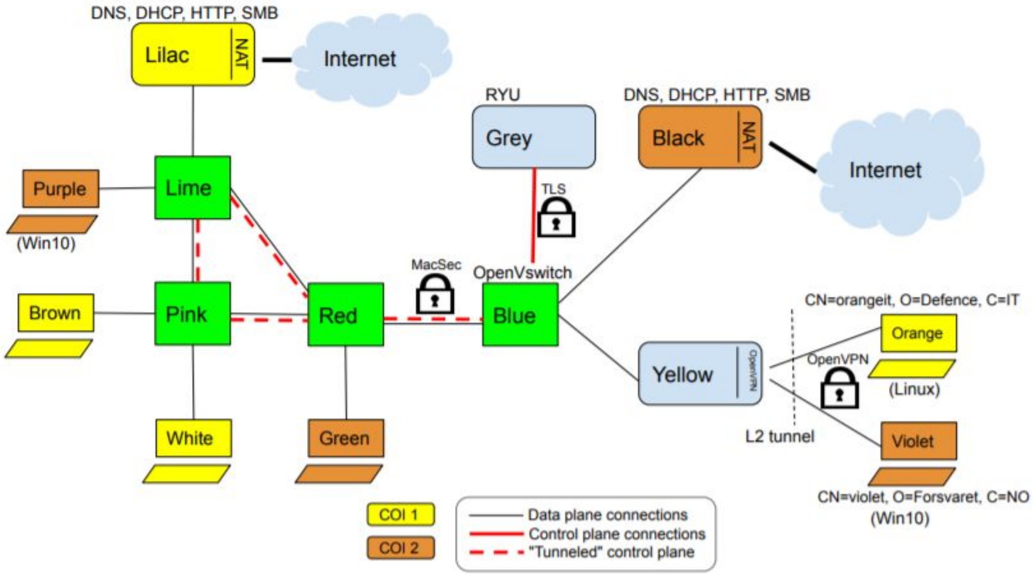


Fig. 1: Current SDN laboratory configuration

- Links are MACsec protected
- VPN connections are public key authenticated
- Control plane is protected by TLS
- Control plane is overlaid on Data plane (in-band)

Unicast forwarding method

- A link/neighbor discovery protocol establishes a link state topology map in the SDN controller
- Shortest paths between any two switches are calculated and established in switches as flows
- Frames are extended with a 802.1Q header containing the id of the destination switch
- Switches know the MAC address and port of locally connected end systems
- Switches know the connected switch id of other MAC addresses
    - obtained from controller on demand

Broadcast forwarding method

- Based on link/neighbor discovery, *spanning trees* are calculated with root in any switch
- Forwarding down spanning tree from any switch is installed as flows.
- Origin switch id is stored in 802.1Q header
- All switches will additionally forward frames to locally connected end systems

This arrangement allows links to form cycles without creating infinite loops, and allows redundant links to be employed for load balancing purposes (not only fail-over).

COI separation

Coalition partners need to separate their traffic, similar to VLAN.

A 802.1Q header is introduced and coded for a combination of forwarding decision and COI separation. Allowing 16 COIs and 127 switches (Network Elements)

COI “membership” is added by the ingress switch and checked before delivering to the destination port in the egress switch

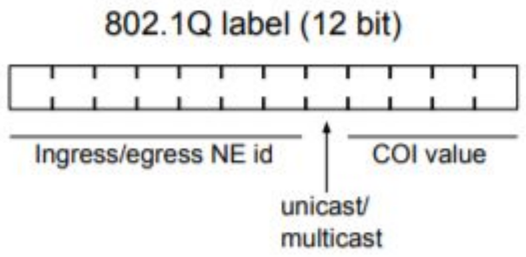


Fig. 2: Encoding of NE id and COI value in 802.1Q label

Whitelisted flows

The SDN switch lends itself well to simple inspection of traffic, where only approved ports and protocols are allowed to pass.

A typical whitelist may consist of: ARP, DHCP *(UDP/67,UDP/68)*, DNS *(UDP/53, TCP/53)*, HTTP *(TCP/80, TCP/443)*, SMB2 *(TCP/445)* and LLMNR *(UDP/5355)*

1. What is the performance penalty of the added number of flow rules?

2. What is the efficacy of the whitelist attack protection?

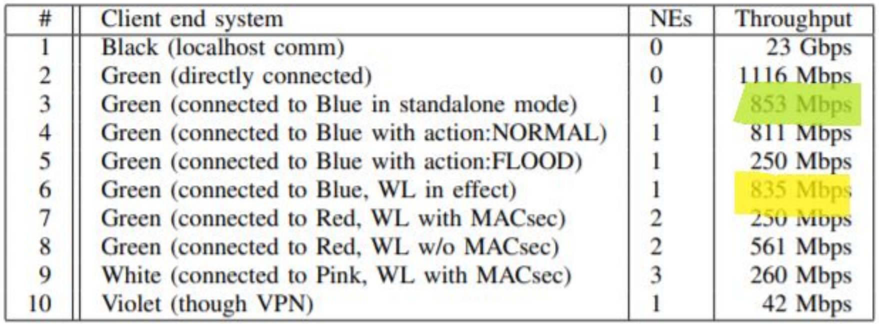
Performance penalty of SDN flow rules

To what extend will a larger set of flow rules degrade the performance of a switch?

A number of tests were conducted with the *iperf* utility. Full explanation in the paper

Conclusion: A realistic whitelist results in a performance (yellow marker) only marginally lower than a minimal configuration (green marker) of the switch.

TABLE I: Throughput evaluation of OpenVswitch



#	Client end system	NEs	Throughput
1	Black (localhost comm)	0	23 Gbps
2	Green (directly connected)	0	1116 Mbps
3	Green (connected to Blue in standalone mode)	1	853 Mbps
4	Green (connected to Blue with action:NORMAL)	1	811 Mbps
5	Green (connected to Blue with action:FLOOD)	1	250 Mbps
6	Green (connected to Blue, WL in effect)	1	835 Mbps
7	Green (connected to Red, WL with MACsec)	2	250 Mbps
8	Green (connected to Red, WL w/o MACsec)	2	561 Mbps
9	White (connected to Pink, WL with MACsec)	3	260 Mbps
10	Violet (though VPN)	1	42 Mbps

Efficacy of whitelist attack protection

A number of attempted attacks were targeted on unpatched Win7, WinXP and Metasploitable Linux. Weakly protected platform were intentionally chosen. Attacks were conducted using *Kali Linux* and the *Metasploit* framework

1. Exploits through non-listed ports did not succeed, nor did payloads which tried to make outbound connections.
2. Delivered payloads which listen to incoming port only succeeded if that port was not allocated by running services.
3. Exploits on software bugs through the normal service port (SQL injection, buffer overflow etc.) were not affected by the whitelist

Whitelist protection vs. Intrusion Detection Sys.

- SDN flows only inspect protocol headers, not payloads
- Whitelist protection can be offered on every port
- Both detection and protection is possible

- IDS will inspect deeply and statefully, covering a larger range of attacks
- ISD detects only , does not protect
- Typically one instance in the network (WLAN or Internet reachback)
  - Does not protect attacks between computers on the “inside”

The two techniques will work well in combination.

That’s all, thank you for your attention

Questions, comments: e-mail to

anders@fongen.no