CSSS: Cyber Security Simulation Service for Software Defined Tactical Networks

Ferdinando Battiati[a], Girolamo Catania[b], Lucio Ganga[b], Giacomo Morabito[c], Agatino Mursia[b], Andrea Viola[c]

[a]Italian Army Comando Trasmissioni – ITB NMS, Italy

[b]Leonardo SPA, Italy

[c]DIEEI - University of Catania, Italy

***Abstract*— The Cyber Security Simulation Service (CSSS) is a platform that provides a simulation environment modeling the impact of cyber attacks and related countermeasure in tactical networks exploiting the software defined networking (SDN) paradigm. CSSS integrates a generator of tactical scenarios, a network simulator, a graphical user interface, and a real SDN controller. The scenario generator, network simulator and user interface interact with each others exploiting a NATO standard called High Level Architecture (HLA); the real SDN controller instead interacts with the simulator by exploiting the *system in the loop* approach and can interact with the graphical user interface through a proprietary protocol. The CSSS can be used as training tool and/or Decision Support System by cyber and IT operators in tactical contexts.**

Keywords—Software Defined Networks, Tactical networks, Simulation

I. INTRODUCTION

By clearly separating control and data planes *software defined networking* (SDN) is radically changing how networks are built and designed [1].

SDN was initially thought to work in infrastructured networks. Recently, however, several solutions have been proposed that extend the SDN approach to networks applying the ad hoc networking paradigm such as most tactical networks.

We call the resulting networks, Software Defined Ad Hoc Networks (SDAHN). Examples of such networks include SDWN [2], FlowSensor [3], and SDN-WISE [4].

A Software Defined Tactical Network (SDTN) is a SDAHN utilized in tactical scenarios.

Main focus of such solutions was the support of typical functions of such networks such as routing, QoS support, and energy management.

In this paper we focus on very important matter that although important in several scenarios and crucial in tactical ones, has received little attention so far in SDAHNs: *security*.

To this respect, observe that the SDN paradigm shift implies a radical change in security and, therefore, new tools are required that assist cyber and IT operators in acquiring the needed competences and taking the correct decisions during operations.

In the military domain, simulators have been traditionally used to this purpose.

As a consequence, in this paper we present the Cyber Security Simulation Service (CSSS), which is an evolution of the Cyber Security Simulation Environment (CSSE) platform developed within the context of the Italian *National Program for Military Research* (PNRM). It can be used by cyber and IT operators for training purposes as well as a decision support system to face security treats in SDTNs.

The CSSS integrates a scenario simulator, a network/cyber simulator, a graphical user interface, and a real SDN Controller. Interactions between different CSSS elements exploit standard protocols mostly.

In this paper we will show the functionality of the CSSS in a specific use case, i.e., a black hole attack is performed and the BRAVO approach, proposed in [5], is utilized as a countermeasure.

Accordingly, the rest of this paper is organized as follows.

In Section II we will provide background information about Software Defined Networking (SDN) for ad hoc networks as well as the BRAVO approach.

In Section III we will present the CSSS platform and we will show its functionality in Section IV. Finally, in Section V we draw our conclusions.

II. BACKGROUND

In this section we provide the background information required for the understanding of the following of the paper. More specifically, in Section II-A, we discuss how the SDN paradigm can be applied in ad hoc networks; whereas we discuss BRAVO in Section II.B.

A. Software Defined Ad Hoc Networks (SDAHN)

As we have already mentioned, Software Defined Tactical Networks (SDTN) are a subset of Software Defined Ad Hoc Networks (SDAHN).

A SDAHN is an ad hoc network that implements the SDN paradigm.

Therefore, nodes of a SDAHN are forwarding elements and the totality of control operations are demanded to a (logically) centralized element running a software program called Controller. To perform efficient and effective control, the Controller exploits information about the current status of the network which requires the SDAHN forwarding elements to collect local information and send it to the Controller through an appropriate, secure communication channel.

Note that the connection between SDAHN nodes and the Controller can be achieved in two different ways:

1. There is a long range, low data rate wireless link connecting nodes and Controller directly or connecting both nodes and the Controller to an infrastructured network. Examples of such links include satellite links.
2. Nodes and Controller are connected by means of multi-hop wireless links. In this case an appropriate protocol is needed which allows nodes to send packets to the Controller even if they have not received the relevant information by the Controller. In our work we will consider the protocol introduced in [4].

The way in which packets are forwarded by nodes is determined by the content of a table named "Flow Table". Like in OpenFlow, each entry in the Flow Table is divided into three sections: rules, action, and statistics.

The *rules* section specifies the conditions that must be satisfied by the packets to be classified as belonging to a certain flow. Examples of such flows are "all the packets that must be delivered to a given node", "all the packets generated by a certain node", "all the packets generated by a given application", etc.

The *action* section specifies how the node should behave upon reception of a packet belonging to the corresponding flow. Examples of actions are "forward the packet to a certain node", "drop the packet with a certain probability", "modify the packet", etc. Finally, the statistics section specifies how many times a given Flow Table entry has been used.

Upon receiving a packet a SDAHN node browse its Flow Table to verify whether such packet satisfies the rules of a given Flow Table entry. If this is the case, then the node behaves as given in the action section and updates the statistics information. Otherwise, the node encapsulates the packet into a new packet, which is sent to the Controller.

The Controller will take care of delivering such a packet and will send the node a new rule that can be used in the future to deal with packets belonging to the same flow.

B. BRAVO

BRAVO has been proposed in [5] as a solution to black hole attacks. BRAVO is based on the assumption that most applications generate bidirectional traffic flows, i.e., the amount of data traffic sent by node *A* to node *B* is of the same order of the traffic sent by node *B* to node *A*.

If such condition does not hold it is likely that a black hole attack is ongoing and therefore, appropriate countermeasures must be taken. In BRAVO a given node *A* which is not receiving enough traffic from node *B*, puts node *B* in a black list and does not forward packet to it anymore. Using such an approach BRAVO achieves very good performance.

In SDTN the same reasoning is implemented by the Controller. In fact, the Controller receives statistical information about the usage of the rules in the network nodes. Therefore, it has a clear view of the way traffic is moving in the network and can detect easily whether there are nodes that might be executing black hole attacks. If this is the case, it can exclude suspicious nodes from the network or give a warning to the IT operator who is then in charge of adopting the countermeasures.

III. CYBER SECURITY SIMULATION SERVICE (CSSS) PLATFORM

The CSSS platforms integrates the following components as shown in Figure 1:

- The *Computer Generated Forces* (CGF) component generates and manages the simulated scenario(i.e. the movements of troops, etc) in a given operation .
- The *Network/Cyber Simulator* (NS) is responsible of simulating the communication elements along with the behavior of the elements performing the cyber attack.
- A Graphical User Interface (GUI) that is used by the operator to interact with the CSSS according to the "live" paradigm, which allows the operator to modify the simulation settings while it is running.
- A real Controller for SDTN that executes security countermeasures besides the usual network control operation. In our experiments we have implemented a Controller implementing the BRAVO logic into the SDTN.

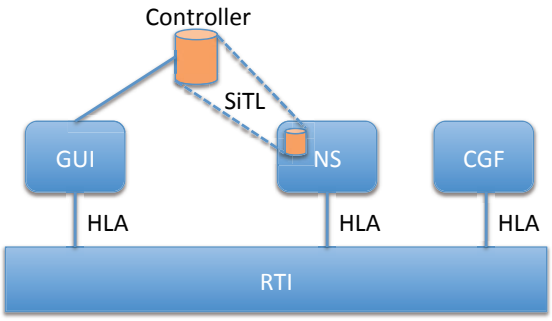

Figure 1: CSSS Architecture.

In this way, the CSSS provides a mixed Constructive-Live simulation environment [6], that is, an environment in which real and simulated people can operate real and simulated systems.

Furthermore, the CSSS can be used by the developers of network controllers, as a testing environment for their products. In other terms, similarly to mininet [7], it can be used as a rapid prototyping tool for the SDAHNs.

For what concerns the interactions between the different CSSS components we have decided to exploit available standards as much as possible. Accordingly, we have selected the NATO High Level Architecture (HLA) [8] for the interactions between the CGF, NS, and GUI. From a realization point of view the usage of HLA requires a component that manages a common timing for all the above elements. This is executed by the so called Run Time Infrastructure (RTI). The Controller and NS interact according to the System in the Loop (SiTL) paradigm which allows real system to interact with simulated ones.

One crucial task in the development phase was the choice of the simulation software to utilize for the Computer Generated Forces and the Network Simulator.

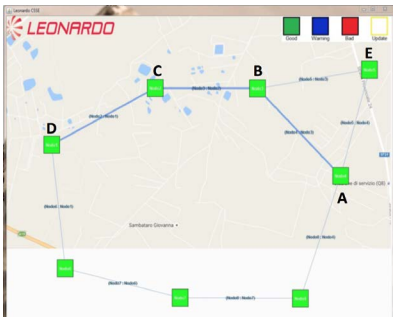

Figure 2: Initial situation.

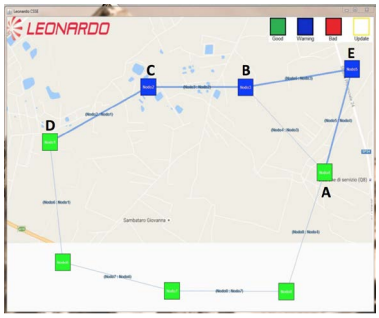

Figure 3: The BRAVO Controller identifies anomalies.

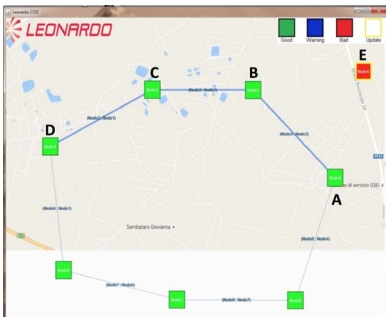

Figure 4: The operator excludes node E from the network.

We have analyzed a large number of software solutions and compared them in terms of fidelity of the results, hardware requirements, execution time, availability of models, level of acceptance in the industry, availability of documentation, support, costs, etc.

The result of our analysis shows that there is not something like the "*best"* solution. In fact, there was a balance in terms of strengths and weaknesses for all the solutions we have analyzed. Accordingly, given that the CSSS was meant as a tool utilized by the personnel at the ITB-NMS of the Italian Army Comando Trasmissioni, the major feature we have considered in our choice was the availability of knowhow.

Based on that, we have selected to use simulators that are built on the Presagis Stage for the CGF and Riverbed Modeler for the NS.

Finally, for what concerns the RTI we have chosen the MAK High Performance RTI.

Instead, the GUI and the Controller have been implemented as proprietary solutions.

IV. CSSS IN ACTION

In this section we will show an exemplary use of the CSSS platform. We have developed a model of a node performing black hole attacks in Riverbed Modeler, as well as a SDTN Controller implementing the BRAVO logic.

We consider the network scenario depicted in Figure 2 in which nodes A and D have bidirectional communications and the path from A to D passes through nodes B and C.

We assume that at a certain point node E begins a black hole attack and injects fake information in the control plane, misleading the Controller in such a way that the path from A to D will pass through E, B, and C. Node E, however, discards the traffic sent by A. After a few seconds the BRAVO Controller detects that there is an anomaly and triggers a warning for the operator. Note in Figure 3 that the SDN Controller highlights that there are anomalies in the traffic patterns traversing nodes E, B, and C.

The operator can easily spot that the anomaly is due to node E and therefore, commands the Controller to exclude it from the network.

The Controller excludes E from the network (in Figure 4 node E is red colored) and calculates the new routes accordingly.

Therefore, the anomaly disappears and the warning is terminated (all nodes are now green with the exception of node E which is red). The operator can re-include node E at any time.

The use case we have just described is an example of the use of CSSS as a training tool. In fact, by exploiting CSSS the operator can learn and experiment the effects of a black hole attack and can exercise in identifying the malicious node(s).

V. CONCLUSIONS

In this paper we have presented CSSS that is a platform that can be used to simulate cyber attacks and the related countermeasure in Software Defined Tactical Networks. CSSS integrates a scenario generator, a network/cyber simulator, a graphical user interface, and a real SDN Controller. Therefore, CSSS can be used both as a training and decision support system tool and as a rapid prototyping environment.

VI. ACKNOWLEDGEMTN

This work was partially supported by Italian MoD under contract "Cyber Security Simulation Environment (CSSE)".

REFERENCES

[1] N. McKeown, T. Anderson, H. Balakrishnan, G. Parulkar, L. Peterson, J. Rexford, S. Shenker, and J. Turner. OpenFlow: Enabling Innovation in Campus Networks. White paper. 2008.

[2] S. Costanzo, L. Galluccio, G. Morabito, S. Palazzo, Software Defined Wireless Networks: Unbridling SDNs. In Proceedings of the European Workshop on Software Defined Networking, October 2012.

[3] T. Luo, H. P. Tan, T. Q. S. Quek, Sensor OpenFlow: Enabling Software-Defined Wireless Sensor Networks. IEEE Communication Letters. October 2012.

[4] L. Galluccio, S. Milardo, G. Morabito, S. Palazzo. SDN-WISE: Design, prototyping and experimentation of a stateful SDN solution for WIreless SEnsor networks. In Proceedings of IEEE Infocom 2015. May 2015.

[5] E. Guardo, G. Morabito, G. Catania, A. Mursia, F. Battiati. BRAVO: A Black-hole Resilient Ad-hoc on demand distance Vector Routing for tactical communications. In Proceedings of BlackSeaComm 2014. May 2014.

[6] W. J. Bezdek, J. Maleport, and R. Z. Olshan. Live, Virtual & Constructive Simulation for Real Time Rapid Prototyping, Experimentation and Testing using Network Centric Operations. American Institute of Aeronautics and Astronautics. 2008.

[7] B. Lantz, B. Heller, and N. McKeown. A network in a laptop: rapid prototyping for software-defined networks. ACM HotNets 2014. October 2014.

[8] IEEE 1516–2000 – Standard for Modeling and Simulation High Level Architecture