**SDN CAaaS Survey, online version**

**May-2015**

*Specific* *human* *factors* of our research include 1) our security confidenceframework providing human reliability in data path security and 2) a survey of network security professionals to help establish metric weighting criteria, and 3) cognitive ergonomics via the metric analysis to facilitate user comprehension, reasoning and the decision making process of their network security.

**Research:**  We intend to design and implement a framework for SDN network security confidence analysis. In order to enhance traditional analysis, we utilize SDN in two key areas; route and destination verification and switch metrics analysis. Referring to Figure 2, this framework will allow SDN authentication applications to validate and verify the routing and destination of data as well as assess the network devices for unexpected behavior (i.e. data compromise, man-in the middle attacks, etc.)

**Purpose of Survey:** In order to better assess the value and qualities of these various networking metrics, we would like your evaluation.

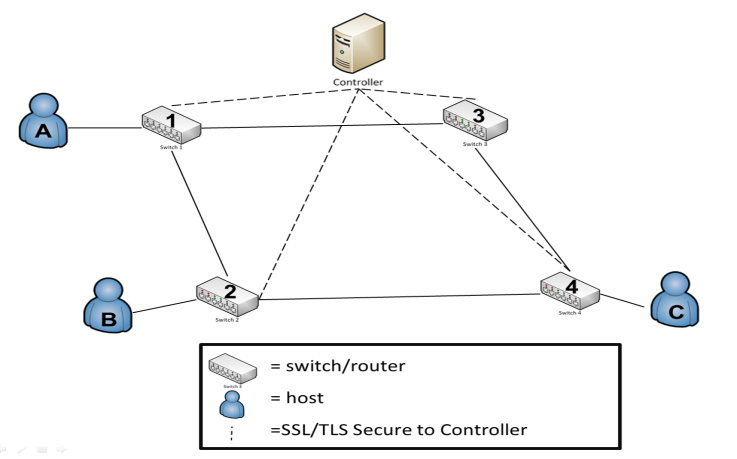
**General Criteria:** For assessing the meters/metrics in general, we propose the following criteria:

* S: Spoofability – this measures the ability of the metric in general to be falsified in some manner
* M: Measurability – the degree of exactness that SDN allows for measure (whether subjective or objective metric)
* V: Variability – measures the range of acceptable values that would be considered within bounds for a given metric.

|  |  |  |  |
| --- | --- | --- | --- |
| Scoring / Criteria | Spoofability | Measurability | Variability |
| 5 | Very difficult to spoof and/or easily recognized by most | Easily assessed metric with limited network overhead | Very narrow band of acceptable values |
| 4 | Hard to spoof and/or easily recognized by some | Easily assessed with moderate network overhead | Small band of acceptable values |
| 3 | Alterable and/or recognized with some training/effort | Assessable metric | Moderate band of acceptable values |
| 2 | Alterable by many and/or hard to recognize | Hard to assess and may reduce performance due to overhead | Large array of acceptable values within a single band |
| 1 | Easily altered and/or very difficult to recognize | Very difficult to assess metric with considerable overhead | Wide array of acceptable values, potentially in different bands |

**Metrics of Evaluation:**

### To better highlight the metrics collected and how they may be analyzed, reference the example network topology in the figure below and assume the host A is sending a sensitive message to host C.



**Example SDN Topology**

## Survey Questions

Example SDN Topology: Variables: P = Packet Cr = SDN Controller F = Flow T = Time x,y,z = Transmission Number n = Size/Bytes

1. *Route Verification*

Utilizing SDN to provide and compare the expected route (from the controller's flow table) to the actual route that data flows from the switch forwarding rules. For example a packet (Px) traveling from Host A to Host C (Reference the diagram above as needed). Path: PRx, -> A, 1, Cr, 1, 2, 4, C Verification Track: Px -> A,1,Cr,1,2,Cr,2,4,Cr,4,C

535

1. *Packet/Flow Size*

We could potentially measure the size of the first and last packets, then multiply it by the total number within a flow. Packet size can then be compared as the data flows from switch to switch and from each of the three elements of message traffic (pre/post and actual).

U(Px1 + Px2 + Px4)U , U(Py1 + Py2 + Py4)U , U(Px1 + Px2 + Px4)

(Py1 + Py2 + Py4) (Pz1 + Pz2 + Pz4) (Pz1 + Pz2 + Pz4)

1. *Packet Arrival Time to Controller*

One of the most basic elements of SDN metrics, at a minimum could be used to compare a sampling of packets from within a given flow to determine similarities or discrepancies. If traffic were intended for multiple recipients, then the arrival time of similar segments of the routing could be compared as well. Assessing the median arrival time from pre/post and actual traffic will provide a metric for any future transmission along the same route.

1. *Packet/Flow Lapse Time*

The time from when packet arrives a switch until it arrives at the next switch. Measuring the packets arrival at two switches provides metrics that should be validated with performance traceroute to assess speed and detect man in the middle attacks. Currently the OpenFlow 1.3 protocol does not support this metric collection.

TPx = Time Packet Arrives at Switch x

Calculating lapse time will help to identify issues within the network, calculating the lapse time between the switch 1 and switch 4.

(TP2 - TP1) + (TP4 – TP2)

1. *Packet/Flow Duration*

The time from the first packet until the flow exits the switch. This is the time it takes a flow from arrival to departure from an individual switch. Knowing flow sizes and types/protocol, we could compare the traffic against performance tests and standardized metrics to assess and create expectations for transport time and routing. Currently the OpenFlow 1.3 protocol does not support this metric collection. Comparing the duration of timea flow spends in the Flow Table may help determine if a high volume of malicious data is traveling utilizing the same flow table entry (i.e. many flows, but few packets).

TF1-TP1 = total flow time per a given switch

1. *Hop Count*

This is a simple metric and is available conventionally; however, the SDN controller could have a much more accurate estimation of the hop count per recipient. This estimation helps validate the path, eliminate routing to devices well outside of the network/system control, and would have limit overhead. It is important to note that hop count can be changed by the controller/installed Flow Entry Action. Therefore this would likely be a minor contributor to the overall metric analysis.

1. *Switch/Device Location*

This refers to the geospatial or at least time zone location of a device. This could be used with varying degrees of trust to assess the strength of a partial route. Some larger level (ISP) or company internal switches would have a higher level of trust versus the open internet. Further, in a close classified/sensitive network the location could have a higher degree of trust. Location data should be largely static, so the overhead of calculating the data and assessing a level of trust from proposed path versus the actual path should be minimal. Location data such as an authorized IP range could identify a switch/network owner, combined with an external entity like IANA or a business IT department.

1. *Switch/Device Characteristics*

Knowing the type of switch, level within the LAN/WAN hierarchy, and switch owner could all be utilized to develop an algorithm for trust with the network. Physical vs. virtual switches – typically a physical switch would have a higher degree of trust as it is harder to spoof. OpenFlow does not support switch configuration information like owner/manufacturer. Although this information can be gathered using SNMP. The level within the LAN/WAN hierarchy could be inferred by the controller’s view of the network topology. A sample of device characteristics questions includes:

* Is trusted network? Trusted SDN?
* Company owned device?
* Is the traffic wholly within company controlled LAN/WAN?
* Level of the switch/router within transmission at large (e.g. ISP/backbone devices vs. local WiFi Router)
* Grade/Quality of Device: Compare consumer grade vs. commercial grade
* Physical vs. virtual switch

1. *Sender/Receiver Role Based-Access*

Linking Active Directory or RADIUS Server to Controller App/Northbound App, traffic could be validated by message type, access of the users and/or user group, pull in location data of the group and compare routing data. This data could also be held or queued if a user was identified as logged off, so it would not flow to the device until the user was logged in and available to receive it. Allowing for integration with outside data sources into the overall SDN confidence analysis provides diversity to the set of metrics and demonstrates a capability that could be expanded networks for large amounts of outside data (biometrics, two-factor authentication, etc.).

1. *Average Number of Packets per Flow*

Data transmission is a two-way street, so it is equally as important to ensure the safety of the receiving node from malicious attack. Many attacks feature source IP spoofing, which makes the task of tracing the attack’s original source very difficult. A side effect is the generation of flows with a small number of packets, i.e. about 3 packets per flow. Given that normal traffic usually involves a higher number of packets. If we can determine a median value for this, then we can assess confidence.

1. *Median Bytes per Flow*

Attack payload size is often very small (for example TCP flooding attacks typically contain packets of ~100 bytes). If we can determine a median value for this, then we can assess confidence.

md(n) = Un(F/2) + n((F+1)/2)

2

1. *Growth of Single Flows*

Verify how many pair-flows occur in the flow stream during a certain interval. Malicious activity often increases the number of single-flows into the network because they send packets with a fake IP. [17]

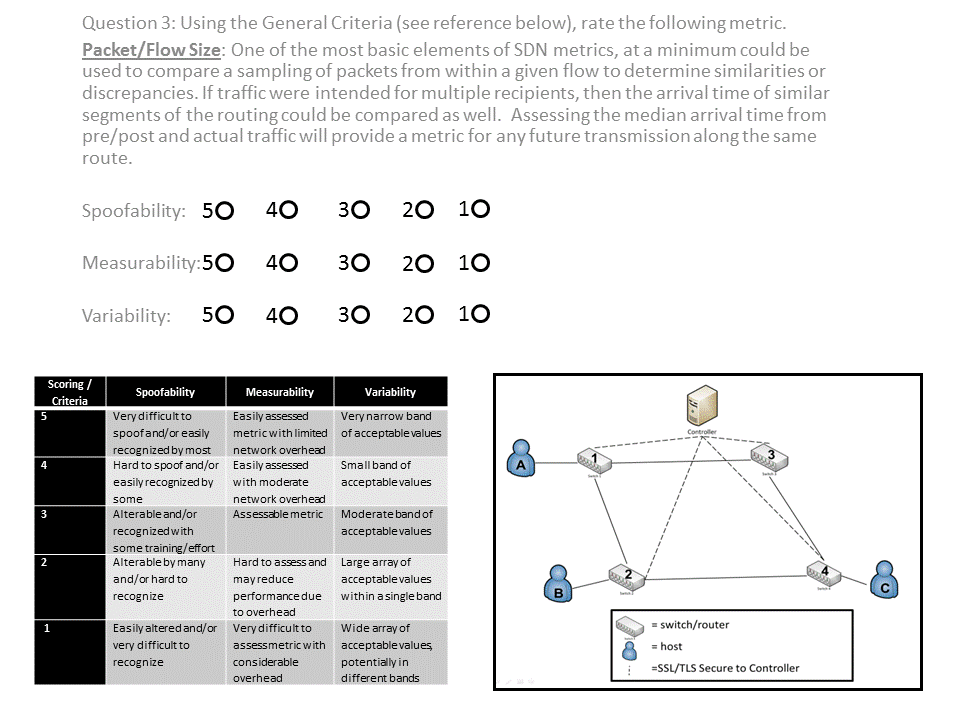
1. *Packet Timestamp Comparison*

By employing timestamping on the first packet of a given flow, we can assess exactly when traffic enters and exits SDN hybrid-network. Based upon experience and/or the pre-/post-traffic packets, we can identify if traffic is flowing at a different rate. Although latency may be the cause of this, any deviation would at a minimum degrade our confidence in the data path. Adan additionally, could potentially hash the packet header and timestamp. Passing this to hash to an authentication server (which knows the header) would to validate the packet based upon the returned timestamp.

1. *Percentage of Correlative Flows*

The Destination has the capability/requirement to reply to packet request (whether legitimate or not). Under normal condition, the rate of traffic from the Destination back to the source is typically half, whereas illegitimate requests would see a near zero.

**Concept of Survey Appearance:**

****