SDN Data Path Confidence

1st Author

1st author's affiliation
1st line of address
2nd line of address
1st author's email address

2nd Author

2nd author's affiliation
1st line of address
2nd line of address
2nd E-mail

3rd Author

3rd author's affiliation
1st line of address
2nd line of address
3rd E-mail

ABSTRACT

Data spillage whether the result of human error or malicious activity drives the need for added security measures to prevent unauthorized access of data once it has left the user's control. Human error can be the result of (1) misaddressing an email, (2) viewing sensitive data in an unsafe environment (i.e. in-line at Starbucks) and transferring data across unknown or unverified networks. Malicious activities include packet alteration, route-modification, and man-in-the-middle attacks. The techniques proposed in our research enable a data protection service based on a user specified delivery path and its verification. Software Defined Networks (SDN) allow the statistical network data analysis as well as the operational configuration examination of the network, which is not available in traditional network. By accumulating time series data in a repository, we can assess the environment to establish known behavior and security patterns. Our approach intends to ensure sensitive data transmissions, it is not designed for general network monitoring. Specific human factors examined in our research include 1) enhancing reliability in data path security interpretation with our security confidence framework, 2) establishing metric weighting via a survey of network security professionals, and 3) improving cognitive ergonomics via the metric analysis to facilitate user comprehension, reasoning and the decision making process of their network security.

Keywords

Software-Defined Networking (SDN), OpenFlow, Security Metrics, Traceroute.

1. INTRODUCTION

As we continue to transmit greater amounts of sensitive information via the internet (particularly via email), the risk of the data being viewed by incorrect or malicious actors increases. There are countless stories of someone misaddressing an email with sensitive company information to a competitor. Disabling the 'email address suggestion' feature might seem like a solution, but human error is more complex than a single feature and it results in millions of dollars remediation costs.

Working in conjunction with the Proximity Based Secure Access (PBSA) project, we explore an approach to limit human error / data spillage by providing a confidence analysis of the physical data transport using SDN metrics. SDN provides the means to create data resilience outside of the user's control and reduce the chances of accidental data spillage. Our approach will also help remove some of the security burden from the user, as the network is providing enhanced validation and authentication.

Today if you want to send a secure message from one person to another, the sender, Brad, first encrypts the message. Then Brad binds the message in an authenticated wrapper. Next Brad sends the message through/over the network to the receiver, Dave. Dave first verifies the authenticity of the message (is it really from Brad or is it tampered malware?). Once verified, Dave then decrypts the message (this may also include Dave verifying his identity to an authentication server – prior to receiving the decryption key).

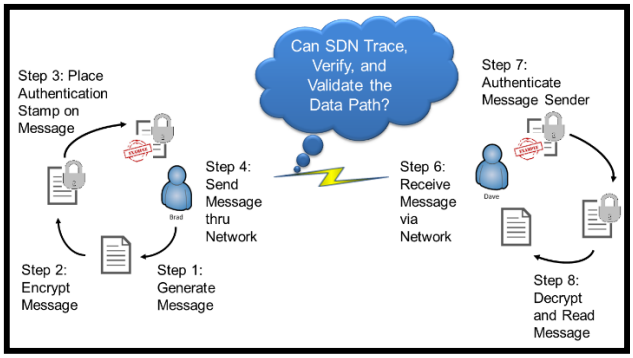

Figure 1. Traditional Network Secure Access Approach

At no point in the scenario above, see Figure 1, has any analysis of the transport medium been assessed. We propose utilizing the power of SDN to do exactly this – verify, validate, and assess the data path.

We designed and implemented a framework for SDN data path 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.) This framework provides a confidence analysis of network security elements.

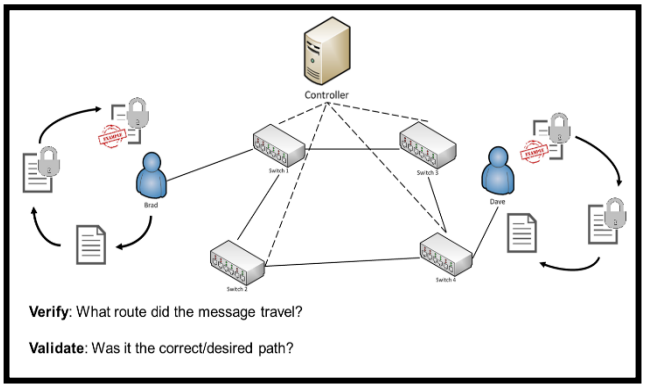

Figure 2. Framework for SDN Confidence Assessment

We established a framework within an SDN and in conjunction with traditional network analysis for the development of SDN

confidence applications that authenticate, validate and verify different aspects of security in the network.

Section 2 provides a review of existing related work. Section 3 covers a brief overview of SDN, highlighting the uniqueness of SDN compared to the traditional networking and defines the need for enhanced traffic security. Section 4 reviews our approach to prevent data spillage via a SDN data path confidence analysis. Section 5 is an explanation of the confidence analysis framework that we propose for this service. Section 6 is our SDN confidence evaluation methodology, experiments, and identified use cases, and Section 7 concludes the paper.

2. RELATED WORK

2.1 Pathlet Routing

A Berkeley team researched a new protocol for tracing the data path. They offer a new scheme, pathlet routing, in which networks advertise fragments of end-to-end paths from which a source can assemble an end-to-end route [1]. They propose this technique as a means to emulate network policy such as BGP, source routing and multipath routing.

2.2 Pathlet Tracer

NEC Labs developed Pathlet Tracer, a layer 2 (data plane) tracing utility that provides user-defined route verification [2]. Pathlet Tracer was designed to detect mistranslations between high level policy and the layer 2 forwarding plane behavior.

2.3 SDN Traceroute

IBM researchers propose a tool, SDN Traceroute, which can map the correct path taken by any packet through the switch plane of an SDN-enabled network [3]. The unique aspect of this approach is that the path is traced by using the actual forwarding mechanisms at each SDN-enabled device without changing the forwarding rules. Like Pathlet Tracer, this tool detects differences between the high level policy and low level, layer 2, forwarding behavior.

2.4 VeriFlow: Verifying Network-Wide Invariants in Real Time

VeriFlow is a low latency mechanism that resides in a layer between the SDN controller and network device [4]. This mechanism provides high level policy enforcement as each OpenFlow rule is inserted, modified or deleted.

2.5 Common Vulnerability Scoring System

The Common Vulnerability Scoring System (CVSS) provides an open framework for communicating the characteristics and impacts of IT vulnerabilities [5]. CVSS focuses on exploits, but its methodology is applicable to our data path metric analysis.

3. OVERVIEW OF SDN & BENEFITS

3.1 Understanding SDN

Gaining an understanding of the benefits and challenges of SDN/OpenFlow is essential to enhance the confidence that users have when sending sensitive traffic across the network. Regardless of the network topology or scale, the network hardware consists of three key components: application, control plane, and data plane. The control plane maintains information that can be used to change data used by the data plane. Maintaining this information requires handling complex signaling protocols. The data plane is a subsystem of a network node that receives and sends packets from an interface, processes them as required by the applicable protocol, and delivers, drops, or forwards them as appropriate, see Figure 3 below [6, 7]. The control plane is responsible for providing the routing logic to the data plane.

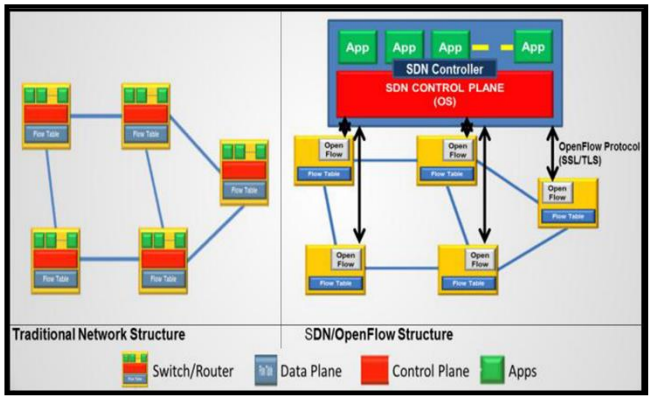

Figure 3. Traditional vs SDN Structure

With SDN, the data and control planes can be separated enabling more programmable and flexible networks, see Figure 3 above. One of the primary technologies behind SDN is the open source protocol - OpenFlow [8]. In additional to abstraction of the control plane, an additional TLS/SSL secure channel is established to support out of band communication directly between the controller and the switches.

3.2 Traditional Traffic Traceability & Switch Metric Limitations

The ability to independently verify and validate a network path is limited by the very architecture of traditional networks. This architecture puts the flow control decisions (how a packet is routed) into the switch fabric (forwarding plane). This allows routing protocols such as OSPF to dynamically change a data path in the middle of data transfer. To discover such changes, route tracing mechanisms must be present in every piece of hardware being monitored. Otherwise the route may change while the route is being verified. It can also change the path to include outside networks that would be previously unverifiable. Without complete monitoring coverage the volatility of network paths renders traceability and some switch metrics to a best guess effort [9].

3.3 Need for Traffic Traceability & Switch Metrics

Public and private networks alike depend on some of the same shared network infrastructure. Traffic isolation is key to the correct behavior of this infrastructure. For example, a business could be required by law to keep sensitive customer data isolated from other traffic. This data could also be restricted to what countries it resides in or travels through. Intelligence agencies restrict different levels of classified information from traveling across certain devices or network segments. Even at the host-level, datacenters must ensure that traffic does not flow across the devices of other customers. From a threat perspective, network route and destination verification is essential to providing security against unauthorized access and compromised infrastructure. One compromised switch could route data across forbidden boundaries without the policy enforcement mechanisms knowledge. For these reasons, route and

destination verification and validation are essential to network security.

4. SDN CONFIDENCE CONCEPT

4.1 SDN Data Path Confidence Analysis

To enhance the multiple factors that PBSA employs to verify authorized access, we examine the new networking paradigm - SDN. By separating the control plane from the data plane, SDN allows greater control and visibility into network state. We intend to capitalize on these benefits and provide, via a northbound API, a PSBA authentication service with a combined confidence analysis of not only the network medium (switches, routers, timing, etc.) but also a comparison with the physical data transport path.

Combining these two elements is not possible with traditional networking for several reasons: (1) traceroute is not a validation of the actual data path and is not in real-time using actual flow routes (2) proprietary switch/router hardware and software limit access and information sharing across the network fabric. The advent of SDN provides us with a framework for developing protocols to access network hardware and gather metrics like the actual network topology, latency between devices, lapse time, flow metrics and the physical device data. Assessing these data points alone can provide some increased confidence that sensitive data is reaching an authorized recipient; however, a near-real time trace of the actual data routing will add considerable validation of the data transport. We propose a method for providing multiple traceroute assessments with limited network overhead, which can then be compared with network metrics for a greater confidence assessment.

The overall goal of the SDN Data Path Confidence (and PBSA) is to ensure that data is accessed by the right person, in the right place, at the right time and delivered to a verifiable destination via the correct and verifiable network route. Our proposal of combining an analysis of network metrics and the actual data flow can provide an increased level of confidence in proving this goal. Providing a service to access this analysis will allow not only PBSA implementations, but any verification tool to access this critical information. We propose a SDN controller agnostic confidence analysis that can be used by the PBSA and other projects.

5. SDN CONFIDENCE FRAMEWORK

Our data path confidence analysis approach has two parts. The first validates and verifies the network configuration for the data being transferred using different methods. Second, we employ standards security algorithms against an array of switch metrics (i.e. duration, elapsed time and network metrics like latency, total transmission time, etc.) to establish a confidence level. This section identifies a Route Verification Method from the first part and several metrics and experimental values from the second.

5.1 Route Verification Approach

We propose a route verification technique that will validate and verify the data path using network metrics and analysis that represent the actual data flow. This is accomplished using the techniques and mechanisms in existing tools listed in the Related Work section as well as in the evaluation of traditional network metrics and metrics that can only be collected in a SDN. One of these methods is to compare data flow metrics from three flows: pre-flow, actual-flow, and post-flow. Pre and Post segments will be small in nature, to limit overhead, but will provide necessary route and flow metrics for comparison with the actual message traffic. It is important to note that this method is not intended for constant network monitoring, rather for establishing a baseline security confidence level that the end user can leverage in comprehending, reasoning and the decision-making process of their network security. One example process for route verification is as follows:

5.1.1 Pre-Traffic

Once the source identifies a destination for traffic, the SDN Confidence service will begin action. First, a pre-traffic flow will be sent from the source to the destination in order to baseline actual route and device metric. A single pre-traffic packet will have very small overhead on the network and require limited time to execute.

5.1.2 Actual Traffic

Next, the actual traffic will flow. During the flow, we will employ a process similar to the method describe by IBM in the SDN Traceroute research [3]. The overhead of this approach should also be minimal considering the communications are between the switches and the controller which is a separate channel and the actual message traffic.

5.1.3 Post-Traffic

Lastly, a post-traffic flow will be sent from the source to the destination in order to provide a post transmission measurement of the network path and device metrics. Similar to the pre-traffic packet, this should have minimum impact on the networks overhead.

5.2 Current SDN/OpenFlow Device Metrics

The current OpenFlow protocol supports some metrics collection capabilities. Durations are the amount of time that the Flow Entry has been installed on the switch. A brief sampling of what is available under OpenFlow 1.0 includes per flow duration and bytes received. OpenFlow 1.3 adds counters, packets received, and allows for the establishment of grouping [10].

5.3 Metrics for SDN Data Path Confidence Analysis

To better highlight the metrics we collect and how they may be analyzed, reference the example network topology in Figure 4 and assume the host A is sending a sensitive message to host C. We realize Figure 4 is a simplistic model, but it is sufficient for explaining metric collection and utilization.

5.3.1 Route Verification

This is described above in greater details; however, utilizing the example network and scenario in Figure 4 and the common variables:

P = *packet,* F = *flow,* x, y, z = *transmission number,* Cr = *Controller,* T = *time,* n = *size/bytes*.

Using the above scenario, we would accept the following:

$$Path: P_x \rightarrow A, 1, Cr, 1, 2, 4, C$$

$$Verification\ Track: P_x \rightarrow A, 1, Cr, 1, 2, Cr, 2, 4, Cr, 4, C$$

5.3.2 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 baseline metric for subsequent transmissions along the same route.

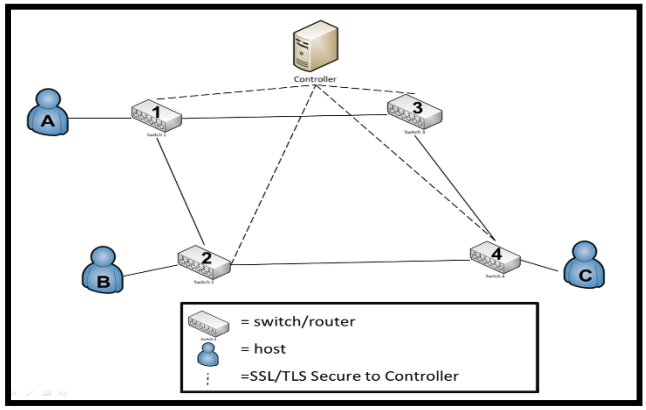

Figure 4: Example SDN Topology

5.3.3 Packet/Flow Size

We could potentially measure the size of the first and last packets, then multiple 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).

$$\frac{(P_{x1}+P_{x2}+P_{x4})}{(P_{y1}+P_{y2}+P_{y4})}, \frac{(P_{y1}+P_{y2}+P_{y4})}{(P_{z1}+P_{z2}+P_{z4})}, \frac{(P_{x1}+P_{x2}+P_{x4})}{(P_{z1}+P_{z2}+P_{z4})} \quad (1)$$

5.3.4 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.

$$T_{P_x} = Time\ Packet\ Arrives\ at\ Switch\ x$$

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

$$(T_{P_2} - T_{P_1}) + (T_{P_4} - T_2) \quad (2)$$

5.3.5 Packet/Flow Duration

This analysis uses metrics about a flow from when the first packet enters the switch plane at the first switch until the last packet has exited the switch plane at the last switch. The time from the first packet until the flow exits the switch. Using these metrics along with flow sizes and types/protocols, we compare traffic flows with performance tests and standardized metrics to assess and create expectations for transport time and routing. The duration of time a flow spends in the switch plane, specifically from switch to switch can help determine if a high volume of malicious data is utilizing the same flow table entry (i.e. many flows, but few packets).

$$Total\ Flow\ Time\ per\ Switch: (T_{F_1} - T_{P_1}) \quad (3)$$

5.3.6 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.

5.3.7 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.

5.3.8 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

5.3.9 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.).

5.3.10 Average 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. 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 [11].

5.3.11 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 [12].

$$md(X) = \begin{cases} \dfrac{X(n+1)}{2}, & if\ n\ is\ odd \\ \dfrac{X\left(\frac{n}{2}\right) + X\left(\frac{(n+1)}{2}\right)}{2}, & otherwise \end{cases} \quad (4)$$

5.3.12 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 [11].

5.3.13 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.

6. SDN METRIC EVALUATION, EXPERIMENTS, AND USE CASES

6.1 SDN Metric Evaluation

Gathering network device data and being able to validate the data path is not the end point for SDN Data Path Confidence Analysis. Rather it is the beginning of a greater statistical review and analysis of the data collected. As mentioned above in Related Work, there exists considerable research and standardization for measuring individual IT vulnerabilities with in CVSS framework. Scores are calculated based on a formula that depends on several metrics that approximate confidence of a secure data path. Scores range from 0 to 10, with 10 being the least secure.

To measure the above metrics, we employ the following characteristic double weighted analysis: Base and Environmental. In the Section 7 – Conclusion and Future Work, we will discuss the potential for a temporal grouping. First is a Base group, similar to CVSS [5] that represents the intrinsic qualities of security metric and user-defined acceptable impact scoring, and the Environmental group which represents the characteristics of user's network/data flow environment.

6.1.1 Base Group Methodology

For assessing the meters/metrics in general, we propose the following criteria (aka SMV):

- 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.

Table 1. SMV Rating Scale

Scoring / Criteria	**Spoofability**	**Measurability**	**Variability**
0.35	Very difficult to spoof and/or easily recognized by most	Easily assessed metric with limited network overhead	Very narrow band of acceptable values
0.48	Hard to spoof and/or easily recognized by some	Easily assessed with moderate network overhead	Small band of acceptable values
0.61	Alterable and/or recognized with some training/effort	Assessable metric	Moderate band of acceptable values
0.66	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
0.71	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

A baseline measurement for each of the previously identified metrics (ref. Section 5.3) and a weighting standard is based upon three groups of input:

(1) A general survey of industry/academic professional for their assessment of SMV for each metric – the current respondents are comprised of 40% IT Industry, 24% Academia, 18% Government, 18% Other Professional. Reference our NetworkSecurityConfidenceAssessment GitHub [13] for a copy of the survey and complete information from the survey respondents. We continue to pursue additional expert input for this survey to further broad the human factor of this analysis.

(2) The authors' review of existing security and metering research. This area is highly influenced by the CVSS standard for exploitability scoring; however, it will be counter-balance (to avoid bias) with item 1.

The SMV rating is based on a 0-1 scale. The lower the score, the greater the metrics ability to predictably and objectively provide higher security confidence.

The Base Score is a combination of the SMV-weighted metric quality and the user/admin-defined impact of a compromised data path. We utilize modified-CVSS exploitability equations for calculating the value of the SMV weight and retain the impact metrics intact. In the context of data path confidence it is important to consider impact from the perspective of the user and the sensitivity requirements for a specific transmission-type or data-type.

$$BaseScore = (0.6 * Impact + 0.4 * MetricQuality - 1.5) * f(Impact) \quad (5)$$

$$MetricQuality = 20 * Spoofability * Measurability * Variability \quad (6)$$

$$Impact = 10.41 * \left(1 - (1 - ConfImpact) * (1 - IntegImpact) * (1 - AvailImpact)\right) \quad (7)$$

$$f(Impact) = 0 \; if \; Impact = 0, else \; 1.176 \quad (8)$$

Impact values are scored as high, medium, low and are scored 0.660, 0.275, and 0, respectively. All three components of the Impact score have identical definitions to their CVSS counterparts [5]. For the purpose of example, an analysis of the 3 of the 13 criteria is provided. The Table 2 illustrates the Base Group assessment of the metrics using a SMV analysis.

Table 2. Example of Base Group

#	Metric Name	MetricQuality	Impact	BaseScore
1	Packet Duration	2.60	6.36	3.95
2	Flow Size	2.21	6.36	3.76
3	Device Characteristics	2.22	6.36	3.77

6.1.2 Environmental Group Methodology

The BaseScore is weighted against the quality of the actual measurement and their relation to a user's network. Drawing again on standardized CVSS equations, the Environmental group is reflected as a new value, EnvirScore. Many of the 'user-defined' factors of the EnvirScore are not necessarily end-user set, rather a combination of end-user and network administrator. More human testing is necessary to determine the exact amount of end-user customization versus central administration. Calculated by reassessing the weighting of the metrics based upon the user's actual network (see examples below); this new value is the Modified.MetricQuality:

$$Modified.BaseScore = (0.6 * Impact + 0.4 * Modified.MetricQuality - 1.5) * f(Impact) \quad (9)$$

$$Modified.MetricQuality = 20 * Modified.Spoofability * Modified.Measurability * Modified.Variability \quad (10)$$

$$EnvirScore = (Modified.BaseScore + (10 - Modified.BaseScore) * CollateralDamagePotential) * NetworkComplexity \quad (11)$$

The lower the value of Modified.SMV, the better the quality of the metric in that environment for predicting a secure data path. The following are example assessments of environmental scoring factors:

- Device Characteristic: Although this particular metric will be a compilation of several factors (vendor, trust-level, etc.), each element must be assessed for quality. Simply making an assessment that devices from Cisco are great and device from D-Link are poor is of dubious quality. In general due to the subjective nature of this metric, the quality of the data is expected to be lower than more objective measurements.
- Route Verification: Using the "sandwich" method described above in Section 5.1, we are able to verify and validate the route utilizing three distinct steps: using the pre-traffic flow, the actual traffic flow, and the post-traffic flow. Therefore the quality of "verified" route by three independent measurements using several different mechanisms is high.

CollateralDamagePotential retains the same meaning and measurements as described in the CVSS documentation. The variable TargetDistribution has been replaced with NetworkComplexity. This metric measures the user's specific network, its complexity, and the level of control. It is scored as follows:

Table 3. NetworkComplexity Measure

Measure	Score	Example
Low	0.25	Small-scale, fully SDN controlled, LAN
Medium	0.75	Multi-site, single admin
High	1.00	Large-scale; diverse HW, SW, & Management
Undefined	1.00	Internet-at-Large

Table 4. Example of Environment Group

#	Metric Name	Mod.MQ	Mod.BS	CDP*NC	EnvirScore
1	Packet Duration	3.86	4.54	0.075	4.54
2	Flow Size	2.81	4.05	0.075	4.49
3	Device Characteristics	4.91	5.03	0.075	5.03

6.1.3 Final SDN Data Path Confidence Analysis

The BaseScore and the EnvirScore are measured independently and can provide user's and network administrators with an industry established norm and a network specific analysis of the overall security of a given data path. The confidence analysis can then be assessed to determine whether or not data decryption keys are provided. Each SDN metric has range of 0-10, with 0 offering the highest level of confidence in data path security.

The metrics will not be weighted directly (i.e. packet size * weighting), but rather scored as to whether or not the metrics are what is expected. Metric value expectation are determine by measuring the standard deviation of the pre/post/actual-traffic scoring (ref. Section 5.1).

In order to assess what scores within an acceptable range two factors must be assessed: (1) is this actual production network data and (2) what is the sensitivity of the traffic itself.

The results of the SDN Data Path Confidence Analysis can then be accessed as a service via a northbound controller API by any network tool or system for route verification and overall security of the data path.

The analysis tool described above is for example purposes only and requires experimentation to refine assumptions prior to being deployed as a fully function northbound SDN service.

Table 5. Example of Final SDN Confidence Assessment

#	Metric Name	Raw Score	Base Score	Envir Score	Final Weight
1	Packet Duration	1-10	3.95	4.95	9.90-18.90
2	Flow Size	1-10	3.28	4.49	8.77-17.77
3	Device Characteristics	1-10	3.77	5.41	10.17-19.17

6.2 Experiments

Returning to our original hypothesis that a SDN data path confidence analysis can identify security concerns within a given transmission, the following two experiments illustrate its effectiveness in a (1) multi-factor analysis and (2) single metric review.

6.2.1 Multi-Factor Confidence Analysis

To better illustrate the confidence analysis process, we developed an application to collect three data points (flow duration, flow size, and switch characteristics) for analysis. The purpose of this collection is to provide an example of the confidence analysis concept which utilizes multiple metrics. This experiments directly addresses the need to provide humans with a reliability assessment. All experiments were run under VMware vSphere 5.1 as virtual machines with 4 x 2.66GHz CPUs, 4GB RAM, and 300GB of storage assigned. The network model for this data collection resembles that in Figure 4. Table 6 below shows a sample of the data collect for each metric, see our GitHub repository for the complete data set [13].

Table 6: Sample of Collected Metrics

Duration of Event (100ns)	Src. IP	Dest. IP	Trans. MAC	Rec. MAC	Packet ID	Packet Length
857230	<A	B>	<1	2>	122359	1070
899044	<A	B>	<2	4>	122359	1070
936830	<A	B>	<4	B>	122359	1070

After raw collection the data was analyzed for the standard deviation, s, between each point of collection. The ratio of deviation, n, was then assessed as to whether or not it meets expected values. If the tool does not have historical data to baseline expected values, then an average rating will be assessed.

Table 7: Flow Duration Confidence Matrix

	Host A-C [A,1,2,4, C] (ms)	Std. Dev., s:	Total , N	Sum (ms):	Ratio of Dev. n,	Median (ms)
Pre-Traffic	5.4171	0.087486	1	14.202	162.335	4.3940
ActualTraffic	4.3940					
Post-Traffic	4.3914					

Presently the deviation metric, stored the last three values, transmissions, with a confidence rating of 'High' or greater. These stored values are factored into future deviation scores. Long-term, we intended to apply machine learning to the deviation metric. In Table 7 we depicts this analysis for a single transmission; similar matrices are generated for each analyzed metric.

Table 8: Combined Confidence Analysis

Metric / Run	Base Score	Envir Score	Final Weighted Measure
Flow Duration	7.01	6.73	6.87
Flow Transfer Rate	4.76	4.48	4.48
Device Characteristics	6.10	7.74	6.92
Combined Confidence Analysis	6.96	7.32	**7.14**

The raw data assessments are then weighted based upon the criteria listed in Sections 6.1.1 and 6.1.2. Similarly the weighted values for this example are those listed in Tables 2 and 4.

Highest	0.1 to 3.9
High	4.0 to 6.9
Moderate	7.0 to 8.9
Low	9.0 to 10.0

Figure 5: Confidence Analysis Range [5]

To derive a summary SDN data path confidence analysis, see Table 8, the average of the final weighted measures for each point of collection are assessed against an acceptable confidence matrix. The confidence analysis ranges, which correspond with that of CVSS, are provided below.

6.2.2 Single Metric Analysis

To further enhance the assessment, individual points of data were compared against prior simulations (a baseline for the data flow) in order to provide a confidence analysis for each switch in each simulation, see GitHub repository [13]. The purpose of these experiments is to address user comprehension and reasoning based upon the results of the confidence analysis. This paper merely provides the results of the experiments, but future work will explore development of data visualization tools.

The graphs below illustrate the impact of switch delay when introduced into the confidence analysis tool. From these graphs we can determine the degree of impact s given delay (0ms, 50ms, 100ms, 200ms), will have on the overall Confidence Score. Experiments were conducted by introducing increasing levels of switch latency to simulate malicious activity on a switch. These graphs not only displays the impact of prolonged switch delay, they also illustrates the linear impact on both increasing the file size and as latency rises. These experiments validate our assumption that as network anomalies, in this case malicious latency, increase there is a linear impact to our confidence assessment regardless of the flow/packet size.

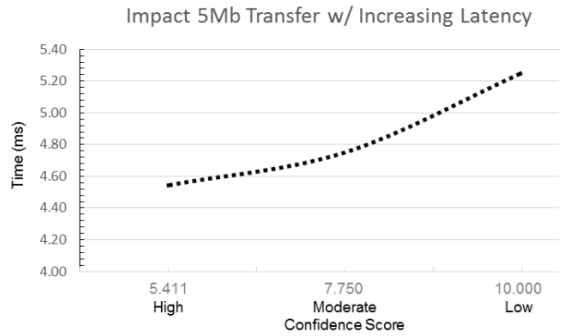

Figure 6: Impact for 5 Mb Transfer with Increasing Latency

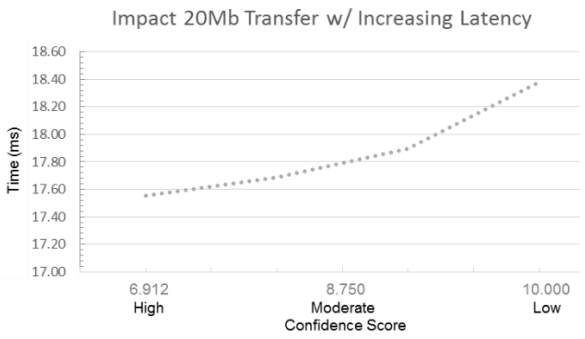

Figure 7: Impact for 20Mb Transfer with Increasing Latency

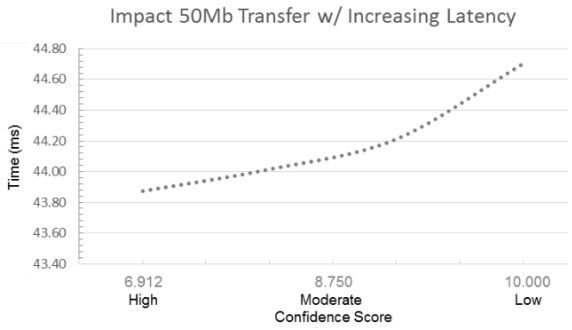

Figure 8: Impact for 50Mb Transfer with Increasing Latency

6.3 Use Cases

6.3.1 Verify Route Using Duration Metric

Confidence Analysis as a Service (CAaaS) is a SDN north bound application our team is developing on top of the SDN OpenDaylight controller framework. The goal of the service is to provide the end user an easy to understand assessment of the security of a network path. CAaaS uses many techniques, some are outlined in this paper, to evaluate the risk of exposure to sensitive data transmission. CAaaS returns an evaluation of this path and a confidence "score" using an expert system based on how professionals in the IT industry rate the security risk and value of these techniques.

The following use cases represent a small subset of techniques used to create the network security confidence assessment. They are included to help the reader understand the overall concept of confidence analysis using detailed steps that are concrete and easily understood.

6.3.2 Generalized Process

Preface – The Actor in this use case is the person who wishes to send an email using an unknown access point or is attempting transfer data across an unknown or unverified network. For the Actor, most of the use cases are the same and follow the steps below.

Assume the CAaaS server is configured and running.

1. The Actor starts by sending an email or making a file transfer across the network. Prior to starting this transfer a confidence analysis request is made to the CAaaS server.
2. CAaaS establishes a route using the SDN controller for that network path and performs one or many of the verification tests partially outlined in this paper.
3. Using these test results the CAaaS server makes a confidence assessment of that network path. Next CAaaS prompts the user with the confidence analysis and confidence score. The user then applies this information in their decision making process.

6.3.3 Verify and Validate Network Route

This method is designed to verify and validate a route using the actual flow rules on the switch plane. This method discovers discrepancies between the high level topology model at the controller and the actual model running on the switch plane. For transferring sensitive data, it can be extremely valuable to know that the data does not traverse a switch that is unknown or unverified for that transfer. Traditional networks cannot guarantee (they cannot even easily determine) the actual path taken across a switch fabric. SDN can guarantee that each switch the data traverses is indeed the switch that was approved for this transfer. CAaaS does this using a technique similar to one published from IBM called SDNTraceRoute.[3]

1. The Actor makes a request to send an email. (generalized process)
2. CAaaS queries the controller data store to see if a route exists to the destination. If it exists, then a higher priority flow is established ensuring it will be used for the transfer.
3. In the network topology, each switches are assigned a color with no switch linked to a switch with the same color.
4. Next CAaaS sends a packet with a different color than the first switch to the first switch. If the packet does not match the color of the switch it is sent to the controller then to CAaaS where the packet's actual route is recorded.
5. The controller then changes the color of the packet to the color of the originating switch and sends it back to that switch. The switch, seeing that the color of the packet now matches its own color, sends it to the next switch using the actual flow rules in the flow table.
6. This process is repeated until the last switch data is recorded.
7. CAaaS now has a complete and accurate view of the actual route the sensitive data will flow over. The results are then compiled with the other results and a confidence score is assigned based on the expert system's understanding of the risks involved with transferring data across this network path.

6.3.4 Validate Route Using Duration Metric

The duration metric for each flow is the elapsed time (in milliseconds) since the flow rule was created or modified on that switch. The duration metric is used to validate that each flow rule in a network path has not been changed since it was created. If this metric has changed from what is known to the controller then the network path is either compromised or there is some kind of normal or abnormal contention for this network path. Most techniques used by CAaaS have some amount of ambiguity like this which is taken into account when the analysis renders a confidence "score".

Assume the CAaaS server is configured to proxy email traffic from the user to the network.

1. The Actor makes a request to send an email. (generalized process)
2. CAaaS queries the controller data store to see if a route exists to the email destination.
3. If the route does not exist, CAaaS sends a test flow to the first SDN switch. The switch, not recognizing the destination of the email, punts the packet to the controller where a path is configured in the controller's data store and the flow rules are pushed to the switches in the network.
4. CAaaS then gets the route to the email destination from the controller. For each switch in the route, CAaaS requests the duration time of this flow rule.
5. The server then compares the switch's duration time for this flow rule to the duration time in the controller data store.

6. If the duration times for each switch matches what is known to the controller, then we can assume that this network route is what the controller knows it to be and has not changed.
7. If a switch's flow rule has a different duration, then something has changed in that flow.
8. Next CAaaS adds this result to the results of the other tests.
9. These results are analyzed and weighted using the expert rules established from the information gathered in the surveys or deduced from best practices.
10. The network security assessment and confidence assessment score is returned to the end user. The end user then decides if the security risks of sending this email are acceptable or not.

7. CONCLUSION AND FUTURE WORK

A challenging, well-desired requirement when delivering sensitive messages or data is getting it to the right person, place, time, location, and over the right route. Our SDN Data Path Confidence analysis provides the framework to verify a path meets a set of standards, validates the actual route that the message travels and assesses network metrics to reassure the user confidence in the overall transport process. We have laid the groundwork to provide a service that can be polled by an outside authentication/key exchange service, e.g. PSBA, to help ensure the quality of the data transmission. Future work will include a finer grained look into the weighting structure to take into account a larger survey audience as well as including temporal-based weighting similar to CVSS. Additionally, we hope to explore machine learning related to the metric deviation determination and address human cognitive ergonomics via enhanced visualization tools for the confidence analysis.

8. ACKNOWLEDGMENTS

We want to thank our families for their encouragement and support; without them none of this would be possible.

9. REFERENCES

[1] Godfrey, P. B., Ganichev, I., Shenker, S., and Stoica, I. Pathlet Routing. In *Proceedings of the ACM SIGCOMM*, 2009.
[2] Zhang, H., et al. Enabling Layer 2 Pathlet Tracing through Context Encoding in Software-Defined Networking. In *Proceedings of HotSDN '14*, 2014.
[3] Agarwal, K., Rozner, E., Dixon, C., and Carter, J. SDN Traceroute: Tracing SDN Forwarding Without Changing Network Behavior. In *Proceedings of the Third Workshop on Hot Topics in Software Defined Networking*, 2014.
[4] Khurshid A., Zou, X., Zhou, W., Caesar, W., and Godfrey, P.B. Veriflow: Verifying network-wide invariants in real time. In *Proceedings of the 10th USENIX Symposium on Networked Systems Design and Implementation (NSDI 13)*, 2013.
[5] Common vulnerability scoring system – SIG. http://www.first.org/cvss/.
[6] Way, A. Alan Talks Tech. http://alantestwiki.pbworks.com/.
[7] T. Sheldon. Packets and Packet Processing Networks. http://en.wikipedia.org/wiki/Packet_processing#Data_plane/.
[8] Enterprise Networking Planet. The OpenFlow Revolution is a Big Switch. http://www.enterprisenetworkingplanet.com/netsp/the-OpenFlow-revolution-is-a-big-switch.html/.
[9] Clayton, R. The Limits of Traceability. 2001.
[10] OpenFlow. OpenFlow Specification. http://www.opennetworking.org/sdn-resources/.
[11] Braga, R., Mota, E., and Passito, A. Lightweight DDoS flooding attack detection using NOX/OpenFlow. In *IEEE 35th Conference on Local Computer Networks*, 2010.
[12] Guo, R., Yin, H., Wang, D., and Zhang, B. Research on the Active DDoS Filtering Algorithm Based on IP Flow. In *International Journal of Communications, Network and System Sciences*, 2009.
[13] Network Security Confidence Assessment. https://github.com/NetworkSecurityConfidenceAnalysis.