Security Analysis of Software Defined Networking Architectures — PCE, 4D and SANE

David Klingel
Department of Computer Science
TU Darmstadt, Germany
david.klingel@stud.tu-darmstadt.de

Rahamatullah Khondoker,
Ronald Marx, Kpatcha Bayarou
Fraunhofer SIT, Darmstadt, Germany
{rahamatullah.khondoker, ronald.marx,
kpatcha.bayarou}@sit.fraunhofer.de

ABSTRACT

Today's data networks are steadily growing in size and complexity. Especially in enterprise networks, these development lead to the requirement of a central network administration. With Software Defined Networking (SDN), this requirement can be fulfilled. However, new security considerations such as the protection of the central component must be taken into account. Motivated by the new security requirements that security should play a key role in SDN architectures. This paper chooses some architectures, namely, the Path Computation Element (PCE), 4D, and the Secure Architecture for the Networked Enterprise (SANE). These architectures are analyzed with respect to its security capabilities using Microsoft's threat modeling technique, STRIDE. The analysis shows that architectures such as PCE and 4D are vulnerable to tampering and information disclosure as well as Denial of Service attacks. The detected threats can be mitigated by using standard technologies such as TLS and IPsec for securing the communication between interactors.

Categories and Subject Descriptors

C.2.1 [**Computer-Communication Networks**]: Network Architecture and Design

General Terms

Security, Design

Keywords

Software Defined Networking (SDN), STRIDE, OpenFlow, SDN Security, SDN Architectures, SANE, PCE, 4D

1. INTRODUCTION

One design principle in the beginning of the Internet, was the decentralization of different components and cooperation of the systems in a distributed way. In a network with many autonomous systems under different administrative domains like the Internet, this is still the best approach. But for enterprise networks within a single administrative domain this may not be the best solution. The trend in enterprise networks is towards a central administration. Due to increasing size and complexity of networks, this requirement is becoming increasingly important. Further requirements such as flexible network structures, mobility and data center automation enhance the needs for a centralized administration. Hence, a new paradigm in networking has been evolved: *Software Defined Networking (SDN)*.

SDN separates the control plane from the data plane. The control plane is responsible for configuring the switch and for route computation. The data plane is responsible for packet forwarding based on the information from the control plane. SDN decouples the control plane from the switch/router[1] and puts it into a separate system, the so called SDN controller. This centralized instance takes care of the route computation, access control mechanisms as well as the configuration of the network devices.

With the introduction of this new technology, new security considerations have to be taken into account. An adversary who compromises a SDN controller can control the whole network, e.g., by maliciously reconfigure network devices. Therefore, security should play a key role in SDN architectures. This paper analyzes the security capabilities of some chosen SDN architectures to determine which role security plays in current architectures. Several network vendors including Cisco and HP offer their own solutions and architectures for SDN. In research, there exist several SDN architectures, such as 4D [5], the Secure Architecture for the Networked Enterprise (SANE) [2], Ethane [1], Tesseract [9], and

[1]In this paper the terms switch and router are interchangeable.

RCP [4]. But there are also techniques which are in use for several years such as the Path Computation Element (PCE) [3]. This paper focuses on PCE, the 4D design approach and SANE. PCE was chosen because it is matured in terms of standardization and deployment. The 4D design approach was analyzed because it provides well-structured design principles for SDN. The SANE architecture was chosen because it follows the 4D design principles and introduces additional security mechanisms.

This paper is outlined as follows: section 2 describes several SDN architectures such as the PCE, 4D and SANE. After discussing the fundamental architectures, section 3 introduces the threat modeling with STRIDE and analyses the previously discussed architectures with respect to its security. Section 4 concludes the paper.

2. SDN ARCHITECTURES

A requirement in today's enterprise networks is the possibility to centralize administrate and (re)configure the network without introducing additional complexity. These requirements can be fulfilled with SDN. Following, some chosen SDN architectures are discussed. These architectures are PCE, 4D, and SANE.

2.1 Path Computation Element

The Path Computation Element (PCE), as specified in RFC4655 [3], is not a new technology. A PCE is "an entity (component, application, or network node) that is capable of computing a network path or route based on a network graph and applying computational constraints" [3, p.3]. Path computation depending on special constraints is an essential component in traffic engineering, especially in multiprotocol label switching (MPLS) networks. Therefore, PCE decouples the control plane from the "head-end router" (router on the provider edge) and puts the control plane into the Path Computation Element. The Path Computation Element is responsible for, as the name implies, computation of path through the network. Therefore, it uses information provided by routers within the network. These routers send domain information collected by routing protocols such as OSPF and its traffic engineering extensions [8] to the PCE's Traffic Engineering Database (TED) (Figure 1). When a head-end router receives a service request from a router at the customer's edge, it requests a path computation to the PCE. The PCE calculates a route and responds it to the requester.

2.2 4D

Greenberg et al. [5] shows, that the difficulties in management of today's networks lay in the complexity of the control and management plane. Therefore, the 4D architecture follows a clean slate design approach. For the new architecture three key principles were identified on which 4D relies on. These principles respective design goals are: network-level objectives, network-wide views and direct control.

Network-level objectives: The design of robust data networks should be expressed as a set of requirements for its performance and reliability. Today, these requirements are expressed as a low-level configuration command [5, p.4]. Instead of using such low-level commands, the requirements should be expressed in a more intuitive way such as "keep all links low 70% utilization, even under single link-failures" [5, p.4, l.64] or "do not allow hosts in subnet B to access the accounting servers in subnet A" [5, p.4, l.62].

Network-wide views: A network-wide view provides a snapshot of the current network state, the network topology, traffic flows and events.

Direct control: A control and management system is solely responsible to set the states of the data plane regarding e.g packet forwarding. While today's protocols run several distributed algorithms to compute a network-wide view, this logic should be decoupled from the routers.

The name 4D is derived by the four planes of the architecture: discovery, dissemination, decision and data plane (Figure 2).

Discovery plane: The discovery plane is responsible for discovering a device's capabilities and a network's physical components. This includes, e.g., the discovery of the physical interfaces of a switch, its capabilities, the amount of possible FIB entries, but also the neighbors the switch is connected to. These information are used by the decision plane to construct a network-wide view.

Dissemination plane: The dissemination plane provides a logical channel between the controlled network devices and the decision element. This channel is used to carry control information. By establishing a separate channel, control information and data packets are separated, so that control information can be carried

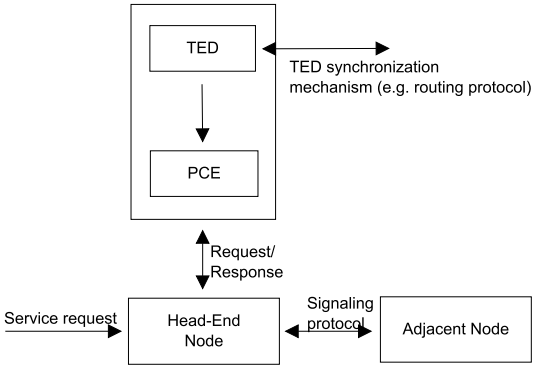

Figure 1: PCE architecture with external PCE node [3, p.12]

without previous path construction in the data plane.

Decision plane: The decision plane is the core component of the 4D architecture. It is a logically centralized element which makes all decisions for controlling the network. The decision element uses the information collected by the discovery plane for its decisions. These decisions are sent via the dissemination plane to the controlled network devices. Because this is a critical component and a potential single point-of-failure, multiple redundant decision elements are used in practice.

Data plane: The data plane processes packets based on the state established by the decision plane, such as packet forwarding based on information in the Forwarding Information Base (FIB), packet filtering, Network Address Translation (NAT) mappings, etc.

At this point in time, there are several other SDN architectures in research which build on the 4D design principles. Such architectures are Tesseract [9], Ethane [1] and SANE [2], as discussed in the following section.

2.3 Secure Architecture for the Networked Enterprise (SANE)

SANE [2] is an architecture which builds on the 4D design principles. SANE provides network-level objectives such as policies expressed in a natural and intuitive way like "allow everyone in group sales to connect to the http server hosting documentation" [2, p.1, l.60]. Furthermore, SANE uses a central decision element for direct control, which also has a network-wide view of the network topology. This central decision element is called *Domain Controller (DC)*. The DC itself performs three main functions: Authentication Service (AS), Network Service Directory (NSD) and Protection Layer Controller.

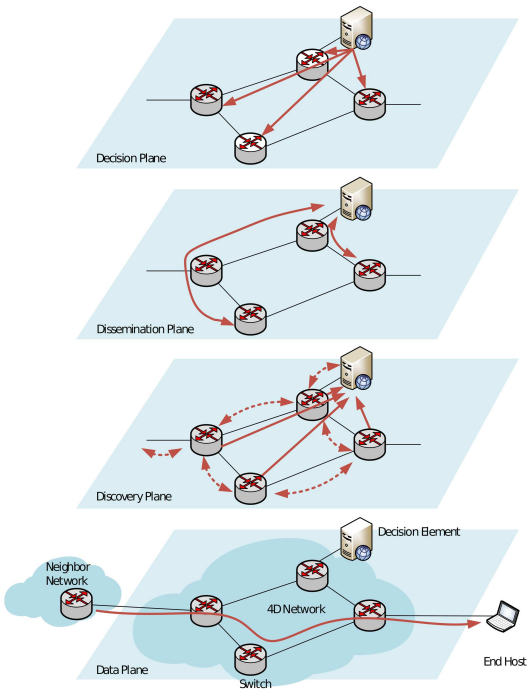

Figure 2: 4D Architecture plane overview [9]

Authentication Service (AS): This service authenticates users, hosts and switches. The service also provides symmetric keys for a secure communication.

Network Service Directory (NSD): This service replaces the traditional Domain Name System (DNS). New services are published in the NSD. When a user or host wants to use a service, it first looks up the service in the NSD by its unique name. The NSD checks the permissions and returns a so called capability – an encrypted source route from client to server.

Protection Layer Controller: The Protection Layer Controller is responsible for controlling the connectivity in a SANE network by generating and revoking capabilities.

When bootstrapping a SANE network, first a default connectivity to the DC will be established by building a minimum spanning tree (MST) with the DC as root. The minimum spanning tree is only used to establish a default route for communication between DC and switch. This connection will be used to establish a symmetric key for authentication and for generating and decoding capabilities. The DC establishes this connection first with its direct neighbor. After establishing connection, the DC receives link-state updates from them. Now the DC proceeds with an increased distance by one hop, and so on. After that the DC has an overview of the network topology.

When a host wants to communicate to a server (Figure 3), the host can only communicate to the DC by default. First of all, the host must authenticate itself to the DC. After successful authentication, a secure channel for further communication will be established. When the user wants, e.g., to communicate to a webserver, this server must previously publish in the NSD under a unique name. To gain access to the webserver, the client requests a capability for the service. When the client has the necessary permissions to communicate to the webserver, the DC returns the capability. The client can now access the server by prepending the capability to each packet.

SANE introduces its own header for neighbor discovery, communication to the DC, host-to-host data transmission and for revoking capabilities. The SANE header is located between the Ethernet and the IP header. Because of this own SANE header, end-hosts must be SANE-aware. To provide interoperability for host which are not SANE-aware, so called translation proxies [2, p.6] can be used. Translation proxies are the first hop for all unmodified clients and provide a mapping from DNS queries to NSD lookups.

3. SECURITY ANALYSIS

This section first discusses the basics of threat modeling with STRIDE. After that, STRIDE will be applied to analyze the SDN architectures discussed in Section 2.

3.1 Threat Modeling with STRIDE

Threat modeling is a technique to identify threats and vulnerabilities in software and system design. There are several techniques for modeling threats such as OCTAVE (Operationally Critical Threat, Asset, and Vulnerability Evaluation), P.A.S.T.A (Process for Attack Simulation and Threat Analysis), TRIKE, STRIDE, etc. While OCTAVE is a heavyweight and complex threat modeling technique which focuses on organizational instead of technical risks, STRIDE focuses on technical aspects. Because of our technical focus and the lightweight analysis process, the STRIDE methodology is used.

STRIDE is an acronym which consists of the threats **S**poofing, **T**ampering, **R**epudiation, **I**nformation disclosure, **D**enial of service and **E**levation of privilege.

- **Spoofing** means that the own identity is masqueraded to illegally access resources. The corresponding security goal is authentication.
- **Tampering** means that data is maliciously manipulated (changed or modified), e.g., during a transmission. This is related to the security property of integrity.
- **Repudiation** means that an action can be denied although it was performed. The target security property is non-repudiation.
- **Information Disclosure** is the exposure of information to unauthorized entities. The related security property is confidentiality.
- **Denial of Service** is the unavailability of a service caused by an attack. The corresponding security property is avalability.
- **Elevation of Privilege** is the expansions of an user's origin permissions, e.g., a user has "read-only" access and can somehow expand this permission to a "read-write" permission. This relates to the security property of authorization.

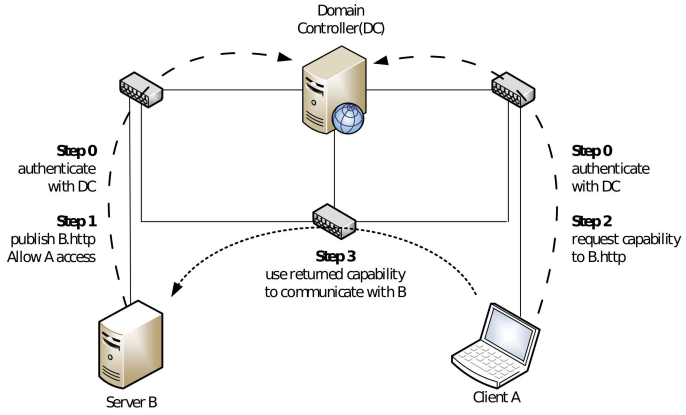

Figure 3: Communication in SANE Architecture [2, p.4]

STRIDE, as part of Microsoft's Secure Development Lifecycle (SDL) process, decomposes the system in its relevant components and analyses each component with respect to its vulnerability. To visualize the system, its components and interactions, so called Data Flow Diagrams (DFD) will be used. DFDs use a set of standardized symbols to model the system. Table 1 shows the DFD elements and the used symbol. In a DFD, each component is analyzed whether it is vulnerable or not. Each analyzed component is only susceptible to a subset of threats. *Data Flows* and *Data Stores* are only susceptible to Tampering, Information Disclosure and Denial of Service attacks, *Processes* are susceptible to any kind of threats (Spoofing, Tampering, Repudiation, Information Disclosure, DoS and Elevation of Privileges) and Interactors are susceptible to Spoofing and Repudiation.

The identified threats can be mitigated by using standard techniques. As described by Howard and LeBlanc [7], this is a two-step process. First, determine which techniques can help. Second, choose the appropriate technologies [7, p.107]. Table 2 shows the threat types with its corresponding mitigation techniques. The appropriate mitigation technologies are depicted in brackets.

3.2 Security Analysis of the Path Computation Element

The first step is to decompose the PCE architecture in its relevant components. Beginning with the interactors, the architecture consists of the Path Computation Client (PCC), the Path Computation Element (PCE) itself and an adjacent node. To request a path computation there must be a bidirectional communication between PCC and PCE. This communication is represented by a data flow. Moreover, the PCE holds the Traffic Engineering Database (TED) which contains the topology and resource information from the domain such as link neighbors and metrics. The TED is represented by a data store. The information for the TED are collected by an Interior Gateway Protocol (IGP) running on adjacent nodes. The communication between PCE and TED as well as TED and the adjacent node is also represented by a data flow. Last, the trust boundaries must be considered. Due to the assumption that the traffic engineering database is located on the same device and not on a remote server, the PCE trusts the TED. Thereby, these components are located

Table 1: DFD symbols and threat categories [6]

Item	Symbol	Description
Data flow		Represents data transmitted over a network connection, named pipes, RPC channels, etc. Is Susceptible to Tampering, Information Disclosure and DoS.
Data store		Data stores are databases, files, registry keys. A data store is susceptible to Tampering, Information Disclosure and DoS.
Process		Processes are programs or applications. A process is susceptible to Spoofing, Tampering, Repudiation, Information Disclosure, DoS and Elevation of Privileges.
Multi-process		Multi-processes are a collection of subprocesses. A multi-process is susceptible to Spoofing, Tampering, Repudiation, Information Disclosure, DoS and Elevation of Privileges.
Interactors		Interactors are system endpoints, e.g., users, server, web-services, etc. It is a entity which provides and uses data. It is susceptible to Spoofing and Repudiation.
Trust boundary		Trust boundaries are boundaries between trusted and untrusted elements.

Table 2: Chosen mitigation techniques and technologies [7]

Threat Type	Mitigation Techniques (Mitigation Technologies)
Spoofing	Appropriate authentication (digest authentication, X.509 certificate authentication, Kerberos, IPsec), Protect secret data, Do not store secrets
Tampering	Appropriate authorization, Hashes, Message authentication codes, Digital signatures, Tamper-resistant protocols (SSL/TLS, IPsec, encrypted filesystem, DCOM and RPC)
Repudiation	Digital signatures, Timestamps, Audit trails
Information Disclosure	Authorization (access control lists, privileges, IP restrictions, server-specific permissions), Privacy-enhanced protocols (SSL/TLS, IPsec, encrypted filesystem, DCOM and RPC)), Encryption, Protect secrets, Do not store secrets
DoS	Appropriate authentication (digest authentication, X.509 certificate authentication, Kerberos, IPsec), Appropriate authorization (access control lists, privileges, IP restrictions, server-specific permissions), Filtering (packet-filtering firewalls), Throttling, Quality of Service
Elevation of privilege	Run with least privilege

within the same trust area. Furthermore, the PCC and the adjacent nodes build their own trust zone.

The identified components are visualized in the DFD shown in Figure 4.

3.2.1 Data Flows

As described in Table 1, data flows are susceptible to tampering, information disclosure and DoS. Assuming that the TED is part of the PCE, the two remaining data flows are between PCC and PCE as well as between PCE and the adjacent node. Based on the assumption that these communications are not integrity protected in a cryptographic manner, the data sent can be tampered. This can be critical if an attacker responds with a manipulated route computation response to the requester. In this way the traffic can be re-routed over a router controlled by an attacker. This is also possible when manipulating the information sent by a routing protocol to the TED. This is because routing protocols operate without any encryption or integrity mechanisms. This can be mitigated by using a tunneled connection from PCC to PCE as well as from the adjacent node to the TED by, e.g., using IPsec. To prevent denial of service attacks the number of requests to the system can be limited.

3.2.2 Data Stores

The data store used by the PCE is the Traffic Engineering Database (TED). According to Table 1, data stores are susceptible to tampering, information disclosure and DoS. Assuming the TED is part of the PCE, it is susceptible to Denial of Service attacks. This is because holding the domain information may require a large amount of memory and CPU power [3, p.6]. Hence, an adversary can cause a DoS attack by flooding the TED with domain information. The susceptibility to DoS attacks can be mitigated by separating the TED from the PCE, as recommended in RFC4655 [3, p.6]. Because a single PCE represents a single point-of-failure, multiple PCEs should be used in practice. When multiple PCEs are used it is also possible to share the load between the PCEs.

3.2.3 Process

RFC4655 does not describe processes which can be analyzed.

3.2.4 Interactor

Interactors, in general, are susceptible to spoofing and repudiation. This affects also the stated interactors in PCE. This is caused by the underlying protocols such as IP, which does not contain protection against spoofing and repudiation. This can also be solved by using IPsec. IPsec provides a mutual authentication which verifies the identity of the communication partners.

Table 3 shows the results of the security analysis of PCE.

Table 3: PCE Security Analysis

Element	Data Flows	Data Stores	Interactors
Spoofing			X[1]
Tampering	X[1]	X[1]	
Repudiation			X[1]
Information Disclosure	X[1]	X[1]	
DoS	X[3]	X[2]	

X susceptible, [1]mitigated with IPsec, [2]mitigated when separating TED from PCE, [3]mitigated by using throttling

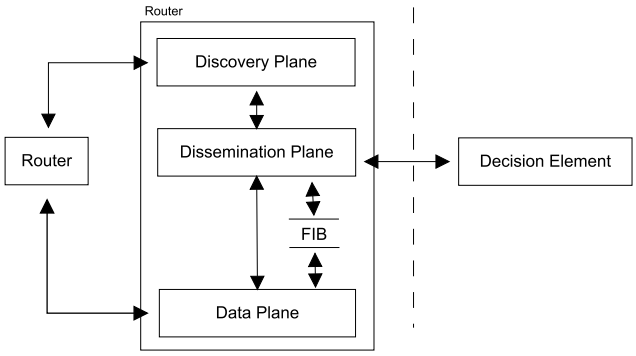

Figure 5: DFD 4D Architecture

3.3 Security Analysis of 4D

The 4D architecture is also decomposed in its fundamental components. The interactors are the router and the decision element. Further interactors are the discovery, the dissemination and the data plane as part of the router. Between the particular planes, there are data flows for data exchange. Moreover, there are data flows between the discovery and the data planes of the routers. Additionally, there is a trust boundary between decision element and router.

The 4D architecture is visualized in the DFD shown in Figure 5.

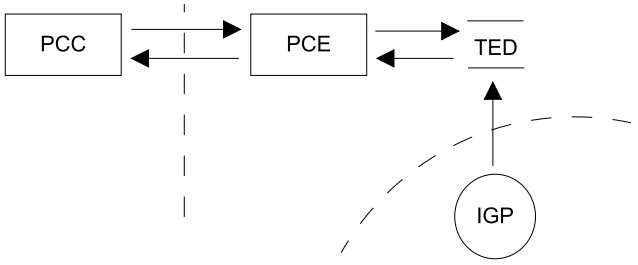

Figure 4: DFD PCE Architecture

3.3.1 Data Flows

According to Table 1, data flows are susceptible to tampering, information disclosure and Denial of Service attacks. The 4D architecture does not provide any security mechanisms to prevent these threats. To protect the flows against tampering and information disclosure it is also possible to use IPsec for securing the communication. Using IPsec does not protect against DoS attacks. To prevent denial of service attacks, other mechanisms must be used. As shown in Table 2, DoS attacks can be prevented by using, e.g., throttling. When using throttling, the PCE limits the number of requests and prevents from overloading the system.

3.3.2 Data Stores

Data stores in the 4D architecture are databases for, e.g., storing forwarding information and network address translations. A data store is the forwarding information base which holds the forwarding information for packets. The FIB is updated from the decision element via the dissemination plane and accessed from the data

Table 4: 4D Security Analysis

Element	Data Flows	Data Stores	Interactors
Spoofing			X[1]
Tampering	X[1]	X[1]	
Repudiation			X[1]
Information Disclosure	X[1]		
DoS	X[2]	X[2]	

X susceptible, [1]mitigated with IPsec, [2]mitigated by using throttling

plane. Data stores are vulnerable against tampering, information disclosure and denial of service attacks. Information disclosure can be neglected, because the FIB does not contain information which should be kept secret such as passwords. Denial of service attacks can be mitigated by limiting the number of requests to the system.

3.3.3 Process

Because 4D describes a more general design approach, this security analysis focuses on data flows, stores and interactors and not on processes. The analysis of processes of a 4D-based architecture is described in the security analysis of SANE.

3.3.4 Interactor

The interactors, router and decision element, are prone to spoofing and repudiation. As stated in the security analysis of the PCE architecture, these threats can be mitigated by, e.g., using IPsec.

Table 4 shows the security analysis of the 4D architecture by using the STRIDE approach.

3.4 Security Analysis of the Secure Architecture for the Networked Enterprise

First of all, the SANE architecture is decomposed into its relevant components. Like the analysis of the previously analyzed architectures, the interactors are identified. The identified components are the end-host, the switch and the DC as central component. Next, the data stores are identified. As part of the DC these are a database for storing the published network services (NSD) with its corresponding unique name and a database for user authentication (AS). For visualizing the communication between DC, switch, NSD and AS, flows are added. Additionally, a data flow between switch and host is added. The data stores are accessed via processes. The DC is a critical component in the architecture. For this reason the DC is placed in a data center. The data center builds its own security domain with special security requirements. A trust boundary separates the DC from the untrusted domain which contains the host and the switch. The DFD, representing the SANE architecture, is shown in Figure 6. The DFD shows a SANE architecture with SANE-aware end-hosts (without translation proxy).

3.4.1 Data Flows

The SANE architecture is an architecture with focus on security. Nevertheless, the symmetric key is only used to encrypt the source routes used for data flows between host and switch, as well as switch and DC. The data transmitted is still in plaintext. That means, tampering and information disclosure is still possible. Assuming that the data stores are on the DC and not on a remote machine, the flows between DC and NSD as well as DC and AS are treated as trusted.

3.4.2 Data Stores

SANE defines two data stores, the NSD which provides a database with published services and the AS a database for the user authentication. According to Table 1 data stores are susceptible to tampering information disclosure and DoS. SANE provides two mechanisms to protect against Denial of Service attacks. First possibility to prevent DoS attacks is to revoke the capability of an attacker's device to send data. The second possibility is to setup multiple data bases on distributed DCs. To prevent tampering, only authorized people can access the databases. This can be regulated by using access control lists (ACL). To prevent information disclosure no secrets should be stored in plaintext. Additionally, the database should only be accessible for authorized people, controlled by ACLs.

3.4.3 Process

Processes are susceptible to spoofing, tampering, repudiation, information disclosure, DoS and elevation of privilege (Table 1). Assuming that the data stores,

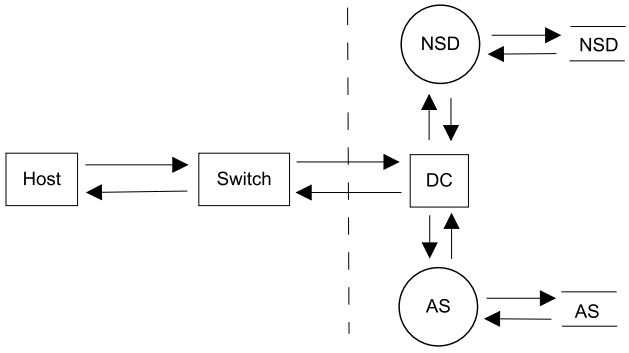

Figure 6: DFD SANE Architecture

Table 5: SANE Security Analysis

Threat Type	Data Flows	Data Stores	Processes	Interactors
Spoofing				-
Tampering	X[1]	X[2]		
Repudiation			X[4]	X[4]
Information Disclosure	X[1]	X[2,3]		
DoS	-	-	-	
Elevation of privilege			X[5]	

X susceptible, - not susceptible, [1]mitigated with IPsec, [2]mitigated with ACLs, [3]mitigated by not storing secrets, [4]auditing trails in logfile, [5]run with least privileges

which are accessed by the process, are located on the same machine, and so within a trusted environment, spoofing, tampering and information disclosure can be neglected. A possibility to prevent repudiation is, to log all access from processes to objects in a logfile. This logfile can be stored on a persistent storage for later analysis. To avoid a process from elevating its privileges, the process should run with least privileges. This means, that the process runs with just enough privileges to get the job done, and no more [7, p.118].

3.4.4 Interactor

The interactors in a SANE network are the DC, switch and host. Interactors, in general, are susceptible to spoofing and repudiation (Table 1). Because the DC negotiates a symmetric key between DC and the communication parties, spoofing might not be possible. Nevertheless, repudiation is still possible. To prevent repudiation, the DC can send control information in a logfile.

The aggregated results of the security analysis of the SANE architecture are shown in Table 5.

4. CONCLUSION

In this paper, security of three SDN architectures have been analyzed namely, PCE, 4D, and SANE. While PCE represents an architecture which is deployed in MPLS networks especially in carrier backbones, 4D represents a more general design approach which can also be deployed in enterprise networks. SANE, an architecture which builds on 4D, incorporates security in 4D design approach. These SDN architectures are vulnerable to threats including tampering and information disclosure. These vulnerabilities can be mitigated by using techniques and technologies which are deployed in current networks, such as TLS for securing the communication between the components. Future SDN architectures should incorporate these technologies for managing (enterprise) networks flexibly and in a secure way.

5. REFERENCES

[1] M. Casado, M. J. Freedman, J. Pettit, J. Luo, N. McKeown, and S. Shenker. Ethane: taking control of the enterprise. *ACM SIGCOMM Computer Communication Review*, 37(4):1–12, October 2007.

[2] M. Casado, T. Garfinkel, A. Akella, M. J. Freedman, D. Boneh, N. McKeown, and S. Shenker. Sane: A protection architecture for enterprise networks. *Proceedings of the 15th conference on USENIX Security Symposium*, 15(10), August 2006.

[3] A. Farrel, J.-P. Vasseur, and J. Ash. *A Path Computation Element (PCE)-Based Architecture (IETF RFC 4655)*, August 2006.

[4] N. Feamster, H. Balakrishnan, J. Rexford, A. Shaikh, and J. van der Merwe. The case for separating routing from routers. *Proceedings ACM SIGCOMM Workshop on Future Directions in Network Architecture*, August 2004.

[5] A. Greenberg, G. Hjalmtysson, D. A. Maltz, A. Myers, J. Rexford, G. Xie, H. Yan, J. Zhan, and H. Zhang. A clean slate 4d approach to network control and management. *ACM SIGCOMM Computer Communication Review*, 35(5):41–54, October 2005.

[6] S. Hernan, S. Lambert, T. Ostwald, and A. Shostack. Threat modeling. uncover security design flaws using the stride approach. http://msdn.microsoft.com/en-us/magazine/cc163519.aspx. Online, accessed on: 06.06.2014.

[7] M. Howard and D. LeBlanc. *Writing Secure Code*. Microsoft Press, 2nd edition, December 2002.

[8] D. Katz, K. Kompella, and D. Yeung. *Traffic Engineering (TE) Extensions to OSPF Version 2 (IETF RFC 3630)*, September 2003.

[9] H. Yan, D. A. Maltz, T. S. E. Ng, H. Gogineni, H. Zhang, and Z. Cai. Tesseract: a 4d network control plane. *Proceedings of the 4th USENIX conference on Networked systems design & implementation*, pages 27–27, April 2007.