PURSUING COST-EFFECTIVE SECURE NETWORK MICRO-SEGMENTATION

Submitted in partial fulfilment
of the requirements for the degree of

MASTER OF SCIENCE

of Rhodes University

Mark Richard Fürst

Grahamstown, South Africa
March 2018

Abstract

Traditional network segmentation allows discrete trust levels to be defined for different network segments, using physical firewalls or routers that control north-south traffic flowing between different interfaces. This technique reduces the attack surface area should an attacker breach one of the perimeter defences. However, east-west traffic flowing between endpoints within the same network segment does not pass through a firewall, and an attacker may be able to move laterally between endpoints within that segment.

Network micro-segmentation was designed to address the challenge of controlling east-west traffic, and various solutions have been released with differing levels of capabilities and feature sets. These approaches range from simple network switch Access Control List based segmentation to complex hypervisor based software-defined security segments defined down to the individual workload, container or process level, and enforced via policy based security controls for each segment. Several commercial solutions for network micro-segmentation exist, but these are primarily focused on physical and cloud data centres, and are often accompanied by significant capital outlay and resource requirements.

Given these constraints, this research determines whether existing tools provided with operating systems can be re-purposed to implement micro-segmentation and restrict east-west traffic within one or more network segments for a small-to-medium sized corporate network. To this end, a proof-of-concept lab environment was built with a heterogeneous mix of Windows and Linux virtual servers and workstations deployed in an Active Directory domain. The use of Group Policy Objects to deploy IPsec Server and Domain Isolation for controlling traffic between endpoints is examined, in conjunction with IPsec Authenticated Header and Encapsulating Security Payload modes as an additional layer of security. The outcome of the research shows that revisiting existing tools can enable organisations to implement an additional, cost-effective secure layer of defence in their network.

ACM Computing Classification System Classification

Thesis classification under the ACM Computing Classification System[1] (2012 version, valid through 2018):

- **Security and privacy ~ Distributed systems security**
- **Networks ~ Logical / virtual topologies**
- **Networks ~ Network privacy and anonymity**
- **Networks ~ Network layer protocols**
- *Security and privacy ~ Access control*
- *Security and privacy ~ Virtualization and security*
- *Security and privacy ~ Firewalls*
- *Networks ~ Bridges and switches*
- *Networks ~ Firewalls*
- Software and its engineering ~ Virtual machines

Keywords: micro-segmentation, distributed firewalls, IPsec, Active Directory, Group Policy, access control lists, virtualization, server and domain isolation, zero trust, deperimeterization

[1] http://www.acm.org/about/class/2012/

Acknowledgements

The genesis and completion of this thesis would not have been possible through my efforts alone, without the supporting framework of the many varied and important individuals who have in one way or another provided me the wherewithal to finish the race and complete the task set out before me. I would especially like to thank:

- My Lord.
- My beautiful wife **Caren**, *my beloved and my friend*, for believing in me and standing alongside me,
- My children **Gem, Cami and Jonty**, for being my delightful inspiration every day,
- My parents **Marius and Glenda**, for their unwavering and unconditional love,
- Prof. **Karen Bradshaw**, for her patient supervision and encouraging guidance,
- and all Computer Science researchers, on whose collective shoulders my thesis stands.

Contents

List of Figures

List of Tables

Acronyms

ACI application centric infrastructure

ACL access control list

AD Active Directory

AH authentication header

ARP address resolution protocol

DC domain controller

DHCP Dynamic Host Control Protocol

DNS domain name system

DS domain services

ESP encapsulating security payload

ESXi Elastic Sky X Integrated

GPO group policy object

IEEE Institute of Electrical and Electronics Engineers

IKE Internet Key Exchange

IP Internet Protocol

IPsec Internet Protocol security

MAC media access control

MPLS Multi-Protocol Label Switching

NAC network access control

NFV network functions virtualization

NIC network interface card

OSI Open Systems Interconnection

OU organizational unit

SDDC software defined data centres

SDI server and domain isolation

SDN software defined networking

SVI switch virtual interface

TCP Transmission Control Protocol

VEN virtual enforcement node

VLAN virtual local area network

VNIC virtualized network interface card

VPN virtual private network

VXLAN virtual extensible LAN

Chapter 1

Introduction

1.1 Context of Research

The increased trend towards de-perimeterization of the network as envisaged in Forrester's zero trust model (Kindervag, 2010b), enabled by new identity-based access and authorisation frameworks, has manifested in various commercial and open source technology platforms designed to enable secure access to resources from any location. However, enterprises may face significant challenges in securing legacy networks and IT systems due to budget and resource constraints that preclude migration to a new network or infrastructure architecture design to protect against common network security challenges.

Perimeter firewalls are designed to address these challenges by evaluating and restricting traffic flow between firewall interfaces, also known as north-south traffic (Gartner Inc., 2016), but are incapable of controlling traffic between endpoints within a network segment as such east-west traffic tends to traverse a network switch without entering or exiting a firewall interface. As east-west traffic is opaque to perimeter firewalls, in the event of an endpoint within that network segment becoming compromised by an attacker, there is no barrier preventing the attacker from pivoting and moving laterally to other endpoints because of the typically large attack surface present in unrestricted network segments (Caldwell, 2015).

Traditional workarounds to address this limitation include security controls such as switch Access Control Lists (ACLs), Virtual Local Area Networks (VLANs), network access controls, host-based firewalls and intrusion detection / prevention systems.

These approaches may be encumbered with management of security rules that tend to be complex and decentralised, and typically do not scale well with multiple endpoints (Al-Shaer, 2014).

Network micro-segmentation is another approach intended to resolve the challenges in securing east-west traffic (Miller and Soto, 2015). Several commercial options exist for host-based micro-segmentation using native security controls, including VMware NSX (VMware, 2015a) for hypervisor-based network virtualization and distributed firewalls to implement micro-segmentation, Cisco Application Centric Infrastructure (ACI) (Cisco Systems, 2017) for software defined networking (SDN) based micro-segmentation and Illumio Adaptive Security Platform for overlay based micro-segmentation (Illumio, 2016). However, these are primarily focused on physical and cloud data centres, and are often accompanied by significant capital outlay and resource requirements.

1.2 Research Statement

Traditional security infrastructures were primarily focused on securing the network perimeter to control north-south traffic between different segments, without building in effective security gates within the internal network segment to control east-west traffic. Micro-segmentation protects these east-west traffic flows between endpoints by implementing security controls based on an organisational security policy, which is designed to prevent attackers that have already established a foothold in one segment from being able to move laterally to other endpoints within the same segment.

Given the constraints of commercial products for small-to-medium sized organisations, this research aims to investigate whether cost-effective network micro-segmentation may be achieved by use of various tools and technologies that are, by default, provided with existing operating system platforms. It is hypothesised that east-west traffic flow within a network segment containing a heterogeneous mix of endpoints can be controlled in a cost-effective manner.

1.3 Research Objectives

To prove the aforesaid hypothesis, the following sub-objectives were defined:

1. Identify existing open-source and commercial software, tools and techniques in the operating systems or networking domains, which can be re-purposed to implement micro-segmentation.

2. Set up a suitable experimental test-bed including a high-level security policy for testing options.
3. Determine criteria by which to evaluate the micro-segmentation implementations.

1.4 Experimental Approach

The approach used to perform the experiments and test the hypothesis is detailed below.

- Design experiments to simulate a typical IT infrastructure and networking environment.
- Define a simple, high-level security policy specifying the connectivity rules to be applied and enforced on endpoints within the same network segment.
- Implement different micro-segmentation approaches using the tools and techniques identified to test compliance with the security policy.
- Analyse the effectiveness of each micro-segmentation approach by simulating connection attempts between endpoints, and intercept traffic flows using a packet capture tool.
- Assess the feasibility of each micro-segmentation approach in controlling east-west traffic within the subnet.

1.5 Organisation of the Thesis

The remainder of the thesis is organised as follows:

Chapter 2 covers related work in the field and briefly discusses other alternative network security approaches.

Chapter 3 details the approach, system specifications, methodology and processes followed for evaluating each micro-segmentation approach.

Chapter 4 analyses the results of the experimental setup.

Chapter 5 provides a summary of the experiment outcomes, lessons learnt and challenges experienced.

Chapter 6 presents the conclusions on the analysis and discussion in the previous chapters, and concludes with some suggestions for future work.

Chapter 2

Literature Review

This chapter commences with a high level overview of network security, and follows with an examination of the various information security controls that delineate different approaches for securing the network.

2.1 Overview of Network Security

Network security requirements in the current era have become increasingly complex to the extent that traditional perimeter firewalls are not sufficiently capable to address the associated threats. Historically, perimeter firewalls were designed to govern communications between different network segments (north-south traffic, as shown in Figure 2.1), and relied on added security controls such as intrusion detection systems (Stepanek, 2001), anti-virus software and other specialised software to inspect and detect potentially malicious traffic flowing between the segments (Gartner Inc., 2016).

Traditional perimeter firewalls are not, however, able to govern intra-network traffic within a network segment, as host-to-host communication (east-west traffic, as shown in Figure 2.2) does not flow past the network switch to the firewall interface. This means that activity within east-west traffic, whether malicious or otherwise, cannot be inspected, detected and managed in the absence of dedicated tools such as host based intrusion detection and prevention system sensors (Miller and Soto, 2015). Addressing this gap would require the deployment of multiple network segments with physical firewalls along with fine-grained security policies and centralised access control, which is generally costly and impractical (VMware, 2014a).

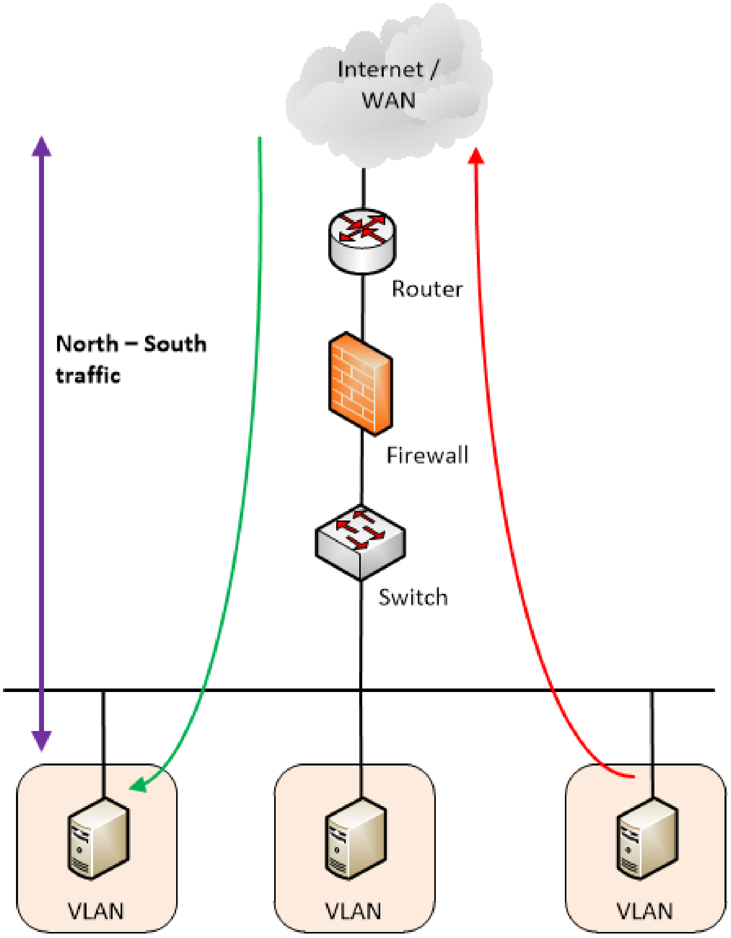


Figure 2.1: North-south traffic flow, adapted from Montemer (2016)

This has led to the concept of ’trust zones’ in which all traffic within a particular segment is assumed to be trustworthy (VMware, 2015a). However, in the current information security environment, such a concept is no longer viable as new iterations of network security threats may allow an attacker to gain a foothold on an infected host within a segment, and move laterally (east-west) to other hosts within that segment.

2.2 Network Security Constructs

Some basic networking constructs are explored in this section, followed by a brief discussion on the various approaches that have been developed towards enabling network security.

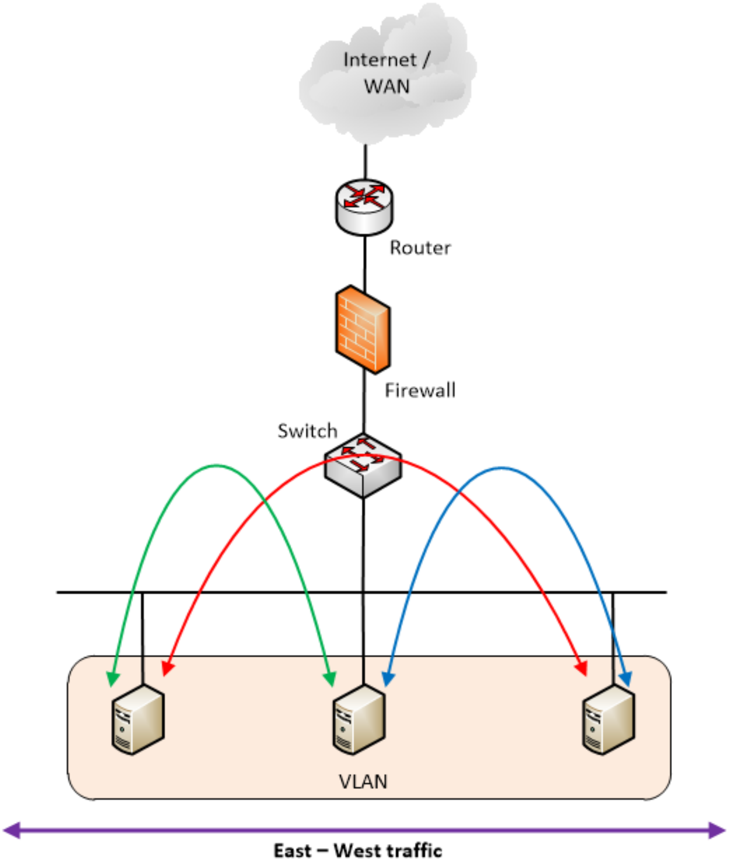


Figure 2.2: East-west traffic flow, adapted from Montemer (2016)

2.2.1 Switches

Traditionally, network traffic on a layer 2 switch is delivered to the respective switch port based on the packet's destination media access control (MAC) address (Cisco, 2015), which can be resolved through an address resolution protocol (ARP) query broadcast throughout the network segment (Figure 2.3). When an ARP reply is received, the switch will forward the packet to the correct port based on its updated MAC address table (Figure 2.4).

2.2.2 Access Control Lists

A local area network comprises one or more network segments that logically correspond with different departments or locations, and are normally physically connected through layer 2 / 3 network switches. As the default switch action is to allow all packets passing through the switch to be forwarded to all parts of the network, connectivity to internal and external network services that are accessible between these zones is typically governed by a high level organisational security

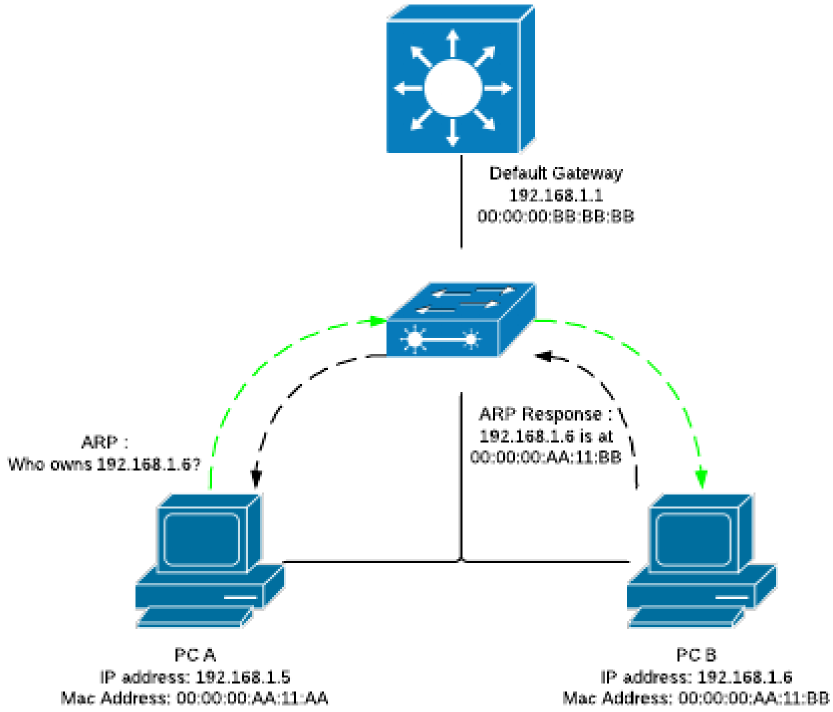


Figure 2.3: Address resolution protocol, taken from Cisco (2015)

policy. The policy is then translated into a sequential collection of permit or deny conditions called ACLs which are configured on the appropriate network interfaces (Rinehart, 2013). The ACLs are designed to filter traffic as it passes through a switch and permit or deny packets crossing specific interfaces.

When a packet is received on a specified interface, the switch verifies that the packet has the required permissions to be forwarded by comparing the fields in the packet against the conditions specified in a list of ACLs applied on that interface. After the first matching condition, the switch stops all further ACL evaluations, and either accepts or drops the packet as specified by the ACL rule. In the event that no conditions are matched, the default action is to drop the packet (Cisco, 2014).

Manual configuration of ACLs on a large scale introduces significant overhead, especially if the access control policies need to be changed dynamically in response to a changing network environment. Proposals for dynamic access control policies based on trust management systems have been mooted to address this challenge (Naldurg and Campbell, 2003), as well as automated generation of policy-based security implementations based on a given network topology and organisational security policy (Bera *et al.*, 2010).

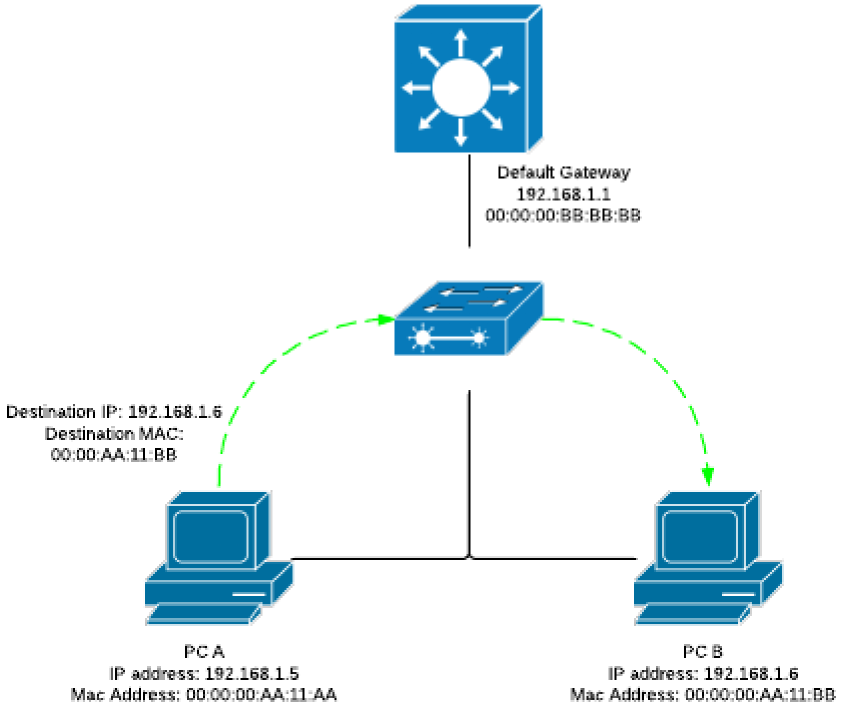


Figure 2.4: Layer 2 switching, taken from Cisco (2015)

2.2.3 Virtual Local Area Networks

VLANs, defined by the Institute of Electrical and Electronics Engineers (IEEE) 802.1Q standard[1], operate on layer 2 of the Open Systems Interconnection (OSI) reference model (Day and Zimmermann, 1983) and are used to segregate physical networks that may or may not be in proximity to each other, into one or more logical networks.

VLAN segregation is typically implemented through the use of VLAN tagging, which appends an additional 4 bytes of data inside the original Ethernet header, between the sender MAC address and the ethertype fields. IEEE 802.1ad[2] also provides for the appending of a second tag to create separate local and provider VLAN domains. The first two bytes of the VLAN tag are used to notify VLAN capable switches, while allowing older switches to process these frames transparently. The last two bytes of the VLAN tag are used to identify the origin of the packet's VLAN network (Kiravuo *et al.*, 2013). VLAN capable switches are then able to enforce the boundaries of these VLANs by being configured to permit an edge device to communicate with other devices within the same VLAN only, or on other trusted VLANs as shown in Figure 2.5. This addresses the challenges inherent within a traditional network topology that

[1] `http://www.ieee802.org/1/pages/802.1Q.html`
[2] `http://www.ieee802.org/1/pages/802.1ad.html`

would otherwise have necessitated a unique physical infrastructure for each bridged network as depicted in Figure 2.6.

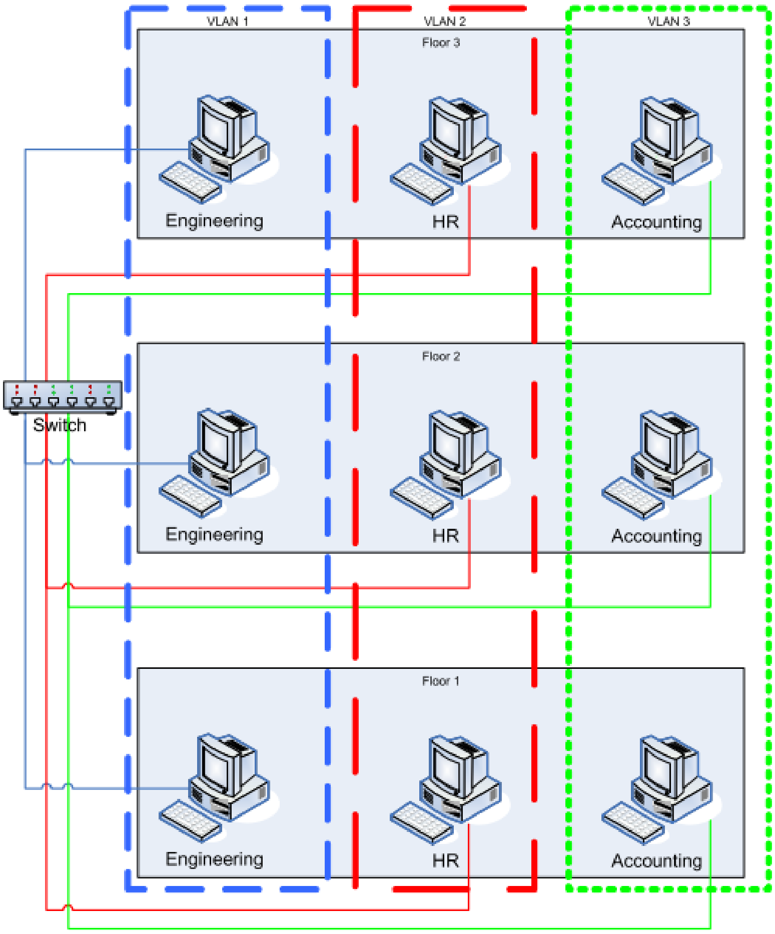


Figure 2.5: VLAN separated network, taken from Leischner and Tews (2007)

VLANs are generally used to increase networking efficiency by limiting the size of the broadcast domain and supporting address space reuse, as well as allowing for different classes of traffic to share the same physical infrastructure while maintaining traffic segregation similar to physical segregation (Leischner and Tews, 2007). However, VLAN tagging was not designed as a security measure, and if not properly configured, is vulnerable to a number of layer 2 attack techniques, including VLAN hopping, MAC attacks and spoofing attacks (Kiravuo *et al.*, 2013).

VLAN access lists, also called VLAN maps, can be used to control traffic within the boundaries of a configured VLAN, and also have the ability to perform packet matching based on MAC addresses as well as standard and extended access lists (Rinehart, 2013).

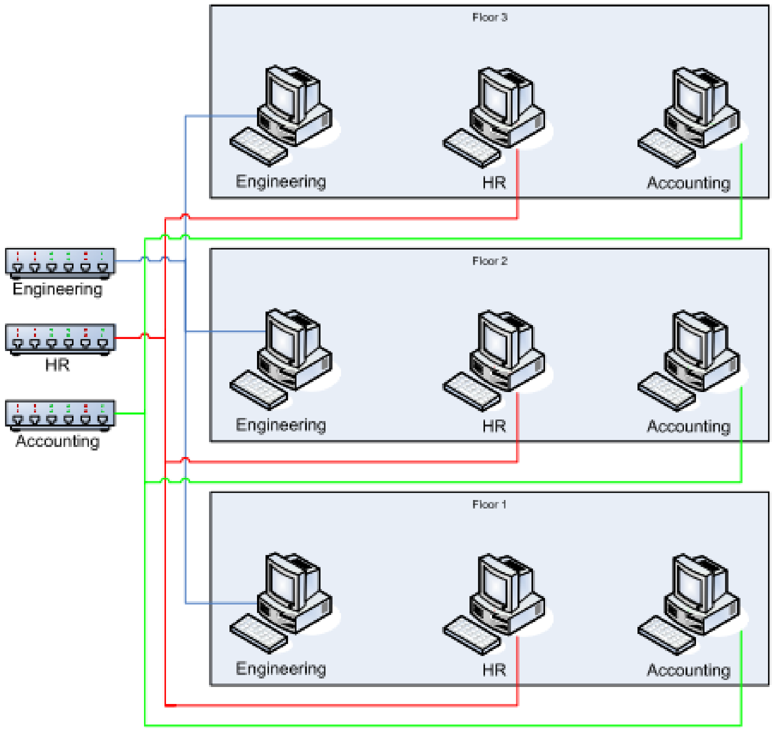


Figure 2.6: Traditional network topology, taken from Leischner and Tews (2007)

2.2.4 Firewalls

Firewalls are positioned as the frontier security defense for enterprise networks, and provide one of the most critical network security functions for filtering out unwanted or unauthorised network traffic entering or exiting the network (Al-Shaer, 2014). On this basis, firewalls are typically implemented on the network border to protect systems from external and internal attacks by isolating domains of varying security risk levels.

A firewall, which is normally located at the ingress point boundary between two discrete networks, is designed to inspect network traffic flowing through the interface, and compare firewall rules in sequential order against each packet until a matching rule is found that determines whether the packet is accepted and forwarded, or rejected and discarded (Gouda and Liu, 2004).

2.2.5 Distributed Firewalls

The concept of a distributed firewall architecture is not new, and was originally proposed to address shortcomings in performance and security challenges inherent with traditional firewalls (Ioannidis *et al.*, 2000). This approach embodies a central security policy server to define the functional equivalent of packet filtering rules, which is then securely distributed to individual endpoints using Internet Protocol security (IPsec, discussed in Section 2.5) to enable authentication and packet integrity, and is finally implemented by an enforcement mechanism to apply the security policy at each endpoint.

Centralised security policy management with decentralised enforcement of the security policy at each endpoint was proposed by Ioannidis *et al.* (2000). In these proposals KeyNote is used as a trust management system making use of public key cryptography for authentication, and which by design allows for delegation of authority as a key enabler of decentralised administration (Stepanek, 2001). Security policy rules can then be defined in a separate, higher level language. These rules are then consumed by the KeyNote PolicyMaker engine that interprets and converts assertions to functional low level firewall rules that are compatible with the respective endpoints.

2.2.6 Network Edge Security

A similar distributed firewall approach in the form of network edge security was proposed by Markham and Payne (2001) as an alternative solution to the constraints of perimeter firewalls that are unable to address insider threats. As network edge security is not intended to function as a personal firewall, policy management is decoupled from the network topology by deploying and enforcing centrally managed security policies (policy enforcement points) to the edge of the network, where endpoints are located. This complies with the Trusted Computer Systems Evaluation Criteria requirements for security mechanisms to be *correct*, *non-bypassable* and *tamper-resistant*, and resolves challenges inherent with software host-based firewalls that are able to be defeated through exploitation of weaknesses and attack vectors in other system components (Department of Defense, 1985).

2.2.7 Hair-pinning

Another approach to resolving the challenges inherent in the lack of visibility of network traffic within a network segment, is the concept of hair-pinning east-west network traffic, in which communications between servers are forced to route through a firewall, as illustrated in Figure 2.7. Hair-pinning traffic typically results in performance issues within the network, including being a bottleneck for traffic flow, presenting as a single point of failure, increasing congestion and latency on the network, and increasing the complexity and redundancy of firewall rules over time (Miller and Soto, 2015).

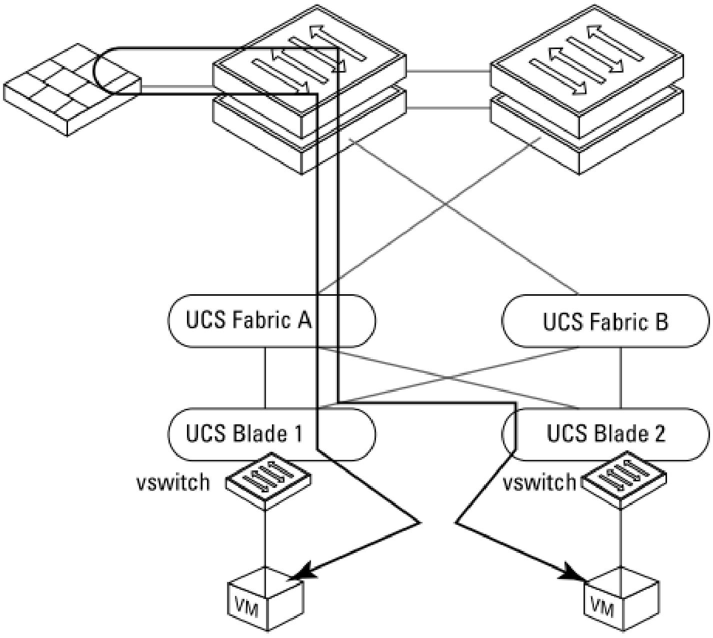


Figure 2.7: Host-to-host east-west firewalling or hair-pinning traffic, taken from Miller and Soto (2015)

2.2.8 Network Access Control

Network access control (NAC) is a network security technique to regulate the access of new devices to a network. This is typically accomplished by a NAC enforcer installed at the perimeter of the network that evaluates devices during the pre-connection phase to verify whether the device complies with a set of policies. Devices that fail authentication or compliance are automatically quarantined or rejected from connecting to the network (Frias-Martinez *et al.*, 2009). In the post-connection phase, devices that successfully connect to the network are continuously monitored for compliance. This approach prevents rogue or unauthorised devices from attacking network hosts or services.

2.2.9 Virtual Private Network

Virtual networks often consist of overlay networks, which are built on top of existing networks by means of tunnelling and encapsulating methods to accomplish virtualization of the network topology (Wang *et al.*, 2013). This technique forms the basis of a virtual private network (VPN) that connects multiple, geographically dispersed remote sites by creating private and secured tunnels over shared or public communications networks. These tunnels are encapsulated and overlaid on top of Internet Protocol (IP) / Multi-Protocol Label Switching (MPLS) and Border Gateway Protocol based public networks, but which are still functionally isolated from the underlying network.

VPNs may comprise one of the following categories suggested by Chowdhury and Boutaba (2009):

1. **Layer 1 VPN** - based on extending the traditional OSI layer 2 / layer 3 packet-switching concepts to advanced circuit-switching domains, to enable carrying of data payloads on any layer (e.g. asynchronous transfer mode or IP protocols. Layer 1 VPNs allow for the provisioning of independent address spaces, layer 1 resource views, separate policies and complete isolation from other VPNs.

2. **Layer 2 VPN** - provides end-to-end layer 2 connectivity between remote sites by transporting layer 2 frames (Ethernet, asynchronous transfer mode or frame relay) between participating sites, and as it is agnostic about higher level protocols, is more flexible than layer 3 VPNs.

3. **Layer 3 VPN** - uses layer 3 protocols (e.g. IP or MPLS) as the VPN backbone for transporting data between the distributed customer edge sites (Knight and Lewis, 2004).

4. **Layer 4+ VPN** - uses layer 4 and higher protocols, such as secure sockets layer (SSL) / transport layer security (TLS) VPNs, to provide connectivity from remote locations.

It has been suggested by Schulz *et al.* (2013) that existing VPN designs and deployments have not addressed known security issues related to traffic analysis and covert channels, in which information from VPN traffic can be inferred without decrypting it. While various proposals have been made to mitigate against network covert channels, the authors suggest that there are as yet no practical solutions that do not carry prohibitively high padding overhead and performance implications.

This view is echoed by DeCusatis *et al.* (2016) in a recent paper discussing the implementation of zero trust cloud networks, in which network segmentation using VLANs and similar techniques as a security mechanism were found to be insufficient.

The authors favour an architectural redesign of the data centre network based on the explicit zero trust model, noting that enhanced security can be accomplished by micro-segmentation approaches as covered in Section 2.3.2.

2.2.10 Virtualization

Virtualization is a hardware abstraction technique that improves resource sharing and utilisation and reduces hardware inventory, power and maintenance costs associated with bare metal hardware. Server and network virtualization can also contribute to an improvement in overall system security and reliability by isolating multiple software stacks in their own virtual environments (Collier *et al.*, 2007). This enables network intrusions to be confined to the virtual environment in which they occur, as well as enabling integrated intrusion detection, security forensics analysis and prevention.

There are generally two classes of hypervisors - Type 1 bare metal hypervisors that run directly on commodity hardware, and Type 2 hosted hypervisors that run on top of a conventional operating system environment (Fayyad-Kazan *et al.*, 2013). Type 1 hypervisors may be based on either monolithic or micro-kernelized designs. Since bare metal hypervisors have direct access to hardware resources, they tend to be more efficient, scalable and robust as well as offering better performance than a hosted architecture model.

2.2.11 Software Defined Networking

Historically, management of networking hardware was constrained by low-level vendor-specific configurations, as the rigidity of the underlying proprietary and closed source hardware typically renders implementation of high-level network security policies a complex process (Kim and Feamster, 2013). This also precludes opportunities for innovation or improvement of network management processes.

SDN is intended to address this challenge by decoupling the control and data planes, logically centralising the controller and view of the network, and abstracting the underlying network infrastructure from applications by providing an application programming interface (API) between devices in the control and data planes (Sezer *et al.*, 2013). This capability to configure, manage and optimise the network for specific flows of traffic provides opportunities for dynamically controlling and adapting the software defined network to meet specific requirements.

Implementations of SDN abound in both commercial and open source incarnations, including Open vSwitch, an open source virtual switch used in most hypervisor platforms that is based on the programmable OpenFlow SDN protocol and are typically deployed in network virtualization environments (Pfaff *et al.*, 2015). Virtual switches are logically abstracted from physical data centre networks and function as the network services provider for virtual machines.

OpenVirteX[3] is an OpenFlow hypervisor based network virtualization platform that provides virtual SDNs by creating multiple virtual and programmable networks on top of a single physical infrastructure (Al-Shabibi *et al.*, 2014). With this network slicing approach, tenants are able to use the full addressing space, specify their own network topology and deploy any network operating system as required.

2.2.12 Network Virtualization

Network virtualization is described by Wang *et al.* (2013) as any form of partitioning or combination of a set of network resources consisting of nodes, links or topologies that are abstracted to users or tenants, such that each tenant has a unique, separate view of the network. This allows for the simultaneous co-existence of multiple virtual networks within the same physical substrate, where each virtual network in a network virtualization environment is a collection of virtual nodes and virtual links that abstracts the virtual network as a subset of the underlying physical network resources (Chowdhury and Boutaba, 2009).

Network virtualization allows each tenant the ability to specify and configure a logical network of their own design that is opaque to, and independent of, other tenants within the network virtualization platform. This is achieved by decoupling the tenant control planes and the implementation of the network virtualization platform from the physical infrastructure through the use of software switching at the edge (Koponen *et al.*, 2014).

Microsoft Hyper-V network virtualization[4] is an example of a network overlay approach to separate virtual and physical networks using packet encapsulation functionality provided by the Hyper-V virtual switch, which is based on the Network Virtualization using Generic Routing Encapsulation protocol supported in Windows

[3] https://ovx.onlab.us/

[4] https://docs.microsoft.com/en-us/windows-server/networking/sdn/technologies/hyper-v-network-virtualization/hyperv-network-virtualization-overview-windows-server

Server 2008/2012, and the virtual extensible LAN (VXLAN) protocol in Windows Server 2016 platforms. These protocols provide the capability to implement isolation or segmentation by using identifiers to differentiate between logical network segments or virtual subnets.

In the same way that a hypervisor provides virtual machine capabilities to the host operating system, Hyper-V network virtualization provides virtual networking capabilities to virtual machines by decoupling virtual networks from the physical network infrastructure. This approach removes the constraints of VLAN and hierarchical IP address assignment from virtual machine provisioning, which enables multi-tenant isolation and enforces security requirements.

2.2.13 Network Functions Virtualization

Traditional networks are comprised of dedicated networking equipment, such as firewalls, switches and routers. As each device requires its own life cycle management supported by dedicated staff and resources, these devices generally lack the capability for capacity provisioning and resource sharing, which means that efficient management and scalability thereof present a challenge (Wang *et al.*, 2016). Network functions virtualization (NFV) is intended to provide an alternative approach by implementing network functions through software virtualization techniques that are executed on off-the-shelf commodity hardware, as depicted in Figure 2.8.

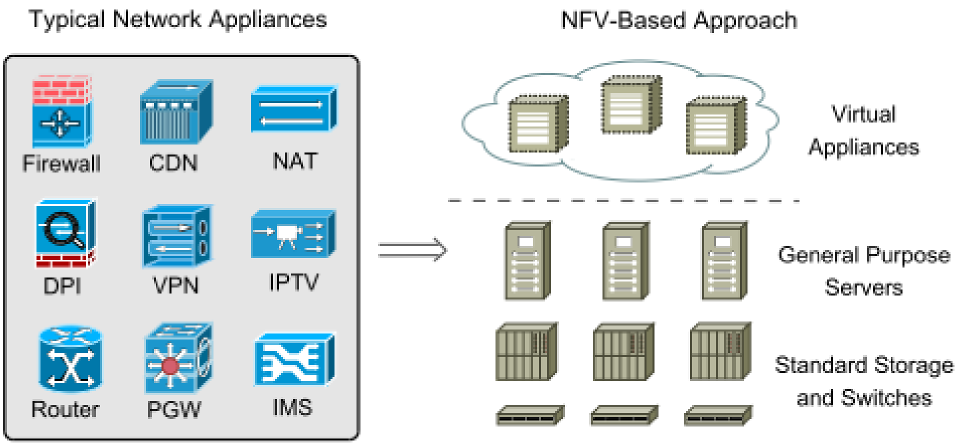


Figure 2.8: Network functions virtualization, taken from Han *et al.* (2015)

A properly designed NFV platform can be leveraged to improve the overall security posture of networking services through the centralised creation, management and modification of security zones, since virtualized firewalls can be created and deployed on demand to protect specific network domains and update the security rules of deployed firewalls remotely (Han *et al.*, 2015).

Hypervisor-based technologies, including Xen[5] and KVM[6] are well established solutions in the NFV space, and offer security and performance isolation out of the box. However, they have been criticised for only supporting a limited number of tenants and offering unsatisfactory networking performance, primarily due to the lack of optimisation for middlebox processing (Martins *et al.*, 2014).

2.2.14 Network Interface Virtualization

Network interface card (NIC) virtualization generally consists of software and hardware enabled virtualization, in which the networking hardware is emulated as a virtualized NIC (vNIC) for use by virtual machines (Wang *et al.*, 2013). This may extend to virtual switches being defined to emulate physical switches, which are connected to vNICs using virtual links, and from which traffic is routed using virtual routing tables maintained by one or more processes, known as virtual routing and forwarding. This concept is also generally known as "network-in-a-box".

From a security perspective, network interface virtualization is particularly prone to performance issues as a result of virtualization overhead. Shea and Liu (2012) found that when faced with a network based denial-of-service attack, virtualized network interfaces experience a tenfold increase in resource utilisation compared to a bare metal machine.

2.3 De-perimeterization Approach to Network Security

In recent years, there has been a gradual change in the information security approach towards the architectural strategy for protecting IT infrastructure and network perimeters, which are examined briefly below.

2.3.1 Jericho Forum

The Jericho Forum originated as an international think-tank focused on driving and influencing the development of information security standards designed to address increasing business demands for secure IT operations over an insecure network

[5] `https://www.xenproject.org`
[6] `https://www.linux-kvm.org`

environment (Open Group, 2007). The key challenge of existing security mechanisms not being sufficiently scalable to accommodate ever-increasing transaction and data volumes led to the concept of 'de-perimeterization', in which emphasis is placed on securing data by shifting the security model from the traditional network perimeter towards the data level itself, as opposed to securing the infrastructure that supports the data. The proposed four-stage road map towards reaching the goals of the Jericho Forum included the following (Stan, 2007):

1. Making services available across the perimeter
2. Removing the perimeter
3. Developing a standards-based approach to data access
4. Controlling access to the data, not the underlying infrastructure

The goal of the de-perimeterization approach was to remove the boundary between an organisation and the outside world by developing a new architecture design based on the Jericho Forum commandments, in which security becomes the core of any organisation's distributed technology architecture (Open Group, 2007). This necessarily requires that security is implemented in all end-user devices and application services by using a combination of encryption, secure computer protocols, secure computer systems and data-level authentication to effectively safeguard critical information assets. The de-perimeterization research culminated in the development and publication of the collaboration oriented architecture (COA) and framework, which leverages off the service oriented architecture (SOA) for delivery of the technical framework (Jericho Forum, 2008).

Within the COA model, there are four main components:

- **Principles** - which define the requirements and constraints of the architecture. Generally, these include knowing the identity of participating parties, establishing the transactional trust level and assurance thereof, assessing the associated risks and ensuring compliance with the applicable legal, contractual, regulatory and privacy requirements.
- **Processes** - these govern the life cycle management of key processes including users, information, devices and enterprise, and the risk management thereof.

- **Services** - these address the provision of supporting services including identity management and federation, policy management, information classification, information asset management and auditing.
- **Attributes** - which measure whether the objectives are being met in pursuance of usability, manageability, availability, efficiency / performance, effectiveness and agility.

The de-perimeterization approach proposed by the Jericho Forum was one of the early precursors of the zero trust model of information security proposed by Kindervag (2010b), and discussed in the next section.

2.3.2 Zero Trust

The zero trust model states that current and historical trust models and approaches that are based on the traditional philosophy of *"trust but verify"* are fundamentally flawed, due to the faulty assumption that a properly secured external perimeter obviates the requirement to implement additional internal security measures (Kindervag, 2010b). For example, the practice of allocating network interfaces into trusted and untrusted categories necessarily means assuming that all network traffic flowing across the trusted interface must be implicitly trusted, regardless of whether the source has been verified or not.

From an internal threat perspective, malicious insiders within an enterprise may compromise their position of trust and defeat or bypass the security controls that were designed against external threat actors. Moreover, the concept of trust cannot be properly applied at the network packet level, due to the inherent inability of the traditional network security design with security overlay to establish and verify identity beyond that which can be derived from packets (Kindervag, 2010a), as illustrated in Figure 2.9.

To address these limitations that preclude effective protection of organisations, the zero trust model as envisaged by Kindervag (2010b) negates the concept of trusted and untrusted networks or segments, and introduces the requirement for end-to-end pervasive and granular protection of data. From an implementation perspective, this would typically entail requiring the use of encrypted tunnels for accessing data on both internal and external networks, implementing strict access control based on the principle of least privilege, and actively inspecting and logging all network traffic to enable situational awareness of data access by all entities (Kindervag *et al.*, 2013).

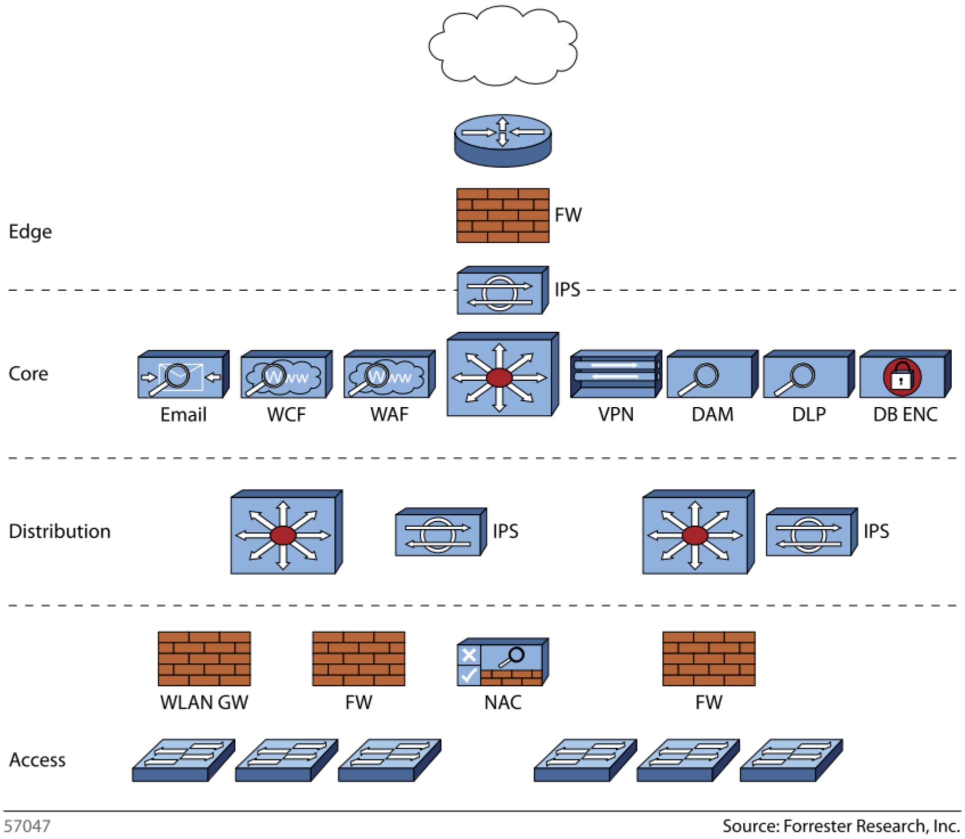


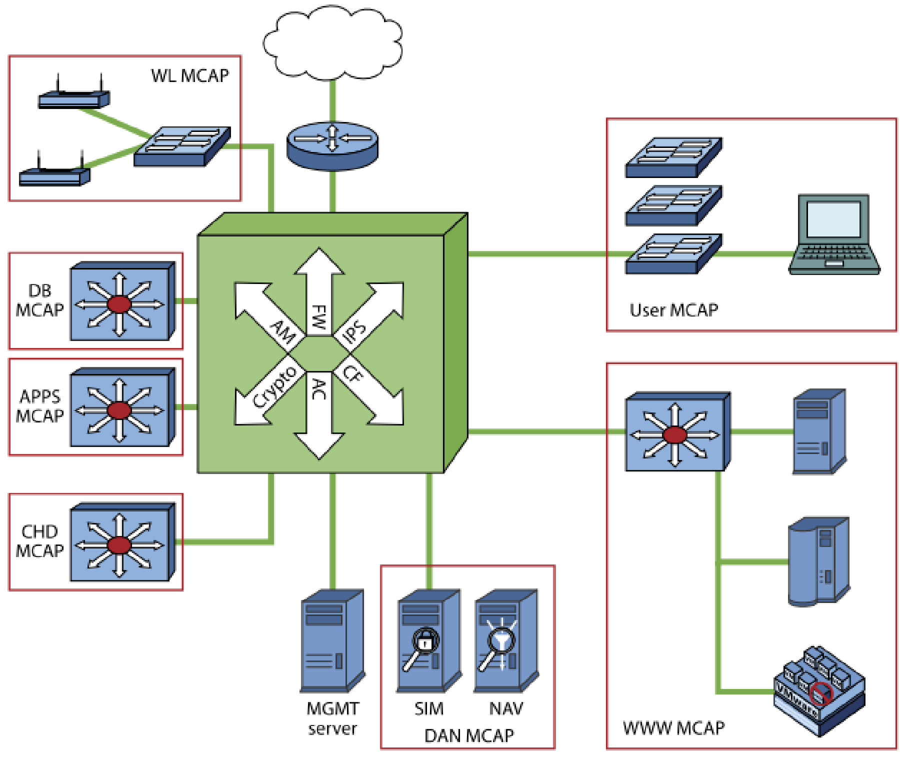
Figure 2.9: Traditional hierarchical network with security overlay, taken from Kindervag (2010b)

Implementing fine-grained security and enforcing compliance requires extensive segmentation of the network environment, which is difficult to achieve with traditional, hierarchical networks due to limitations with existing switch fabric and backplane designs prevalent in most enterprise networks (Kindervag, 2010a). In contrast, the key components of a zero trust network architecture are based on a hub-and-spoke design, in which a network segmentation gateway sits at the core of the network, and where global security policies can be defined and deployed to each parallel, secure network segment that runs its own, centrally managed virtualized switch, as depicted in Figure 2.10. Two implementations of commercial zero trust solutions are briefly assessed below.

2.3.3 BeyondCorp

The development of the BeyondCorp[7] initiative by Google was designed to address concerns around the traditional hierarchical network model discussed in Section 2.3.2,

[7] https://www.beyondcorp.com/



57047

Source: Forrester Research, Inc.

Figure 2.10: Segmented zero trust network architecture, taken from Kindervag (2010b)

and embraces the zero trust model by requiring device and user credentials for fully authenticated, authorised and encrypted access to enterprise resources regardless of the user's location (Ward and Beyer, 2014). The BeyondCorp approach includes the following components:

1. Secure identification of managed devices using a master device inventory database and device certificates.
2. Continuous monitoring, analysis and cross-referencing of devices throughout their lifecycle.
3. Unique identification, qualification and verification checks of managed devices.
4. Tracking, validation and management of users within user and group databases cross-referenced with HR processes, before permitting authorisation and access provisioning through short-lived single sign-on tokens.
5. Delegation of user access requests via a globally accessible identity access proxy after passing access control engine checks.

6. Ongoing and dynamically managed user/device access level via inferred risk as assessed by the access control engine.

The high-level interconnection between these components is illustrated in Figure 2.11.

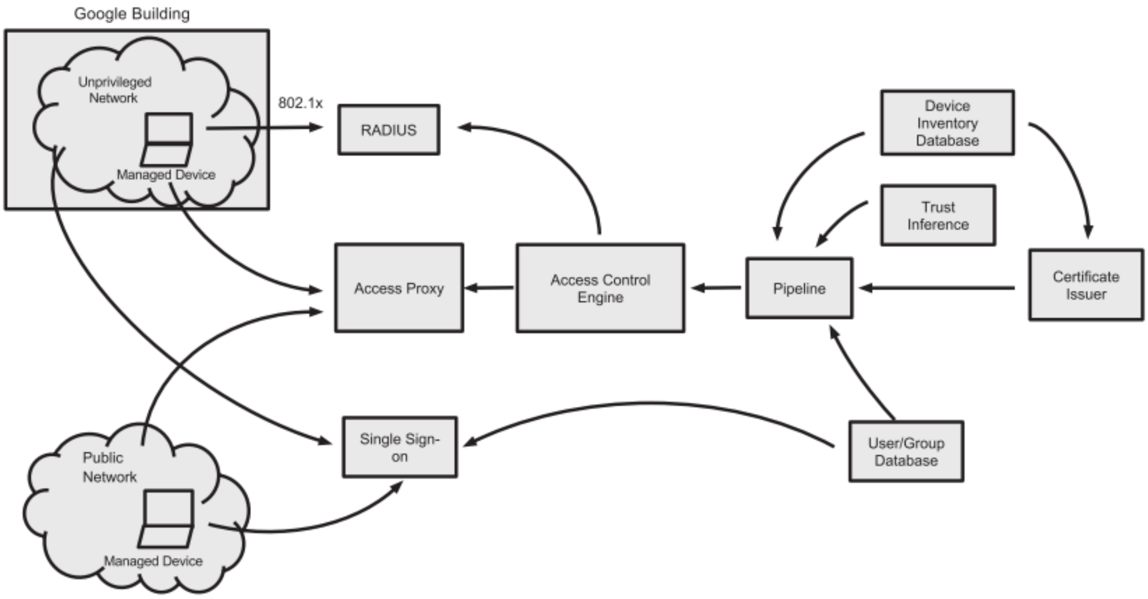


Figure 2.11: BeyondCorp components and access flow, taken from Ward and Beyer (2014)

Due to the size and complexity of the Google corporate and IT environments, a phased migration to the BeyondCorp architecture took several years to complete, and was prefaced by the deployment of an unprivileged private network with minimal infrastructure services that was also connected to the Internet (Gilman and Barth, 2017). After integrating all in-house applications with the access proxy and access control engine, corporate users that were previously limited to direct VPN connections from privileged networks were thereafter able to connect to applications via the access proxy from any network location, whether external, privileged or unprivileged. Google's BeyondCorp implementation serves as a practical validation of the zero trust model described by Kindervag (2010b).

2.3.4 PagerDuty

PagerDuty, a commercial incident response platform, experienced challenges in managing the traditional perimeter system for handling server-to-server communications within a multi-cloud based infrastructure. To overcome these, an alternative, provider-agnostic solution based on the zero trust model was designed that made use of Chef[8], a continuous automation platform, to manage and deploy pre-defined security policies to all hosts (Gilman and Barth, 2017). This was enforced by dynamically configuring local firewall rules based on each host's assigned role within the PagerDuty environment. An IPsec host-to-host mesh network was implemented as the network encryption and authentication layer, which was also designed for scalability to enable future growth. User management within this model was, however, decentralised by forgoing the use of a central LDAP server in favour of programmatically creating local users and groups on each host within the network, primarily to prevent a potential single point of failure for user access and authentication (Gilman, 2014).

2.4 Micro-segmentation Approach to Network Security

Micro-segmentation is a network security approach that partitions a network into sections or segments to restrict unauthorised lateral movement within one or more network segments by isolating applications and systems from each other (Wagner *et al.*, 2017b). This enables fine grained monitoring and control of traffic flows through isolation, segmentation and granular access control mechanisms based on security policies aligned to unique trust models defined by the organisation. Mämmelä *et al.* (2016) propose enhancing the security of 5G networks by implementing micro-segmentation to isolate parts of a single- or multi-domain network dedicated for particular applications and services, as shown in Figure 2.12.

Wagner *et al.* (2017a) point out that the evaluation and implementation of segmentation architectures is currently judgement-based as there is no clear guidance on how to appropriately implement segmentation. To address this, a continuous-time Markov chain is proposed as a low-cost method for evaluating the architecture and supporting security practitioners by enabling them to examine multiple candidate segmentation architectures to find the most efficient model for their network environment.

As data centres continue to move towards virtualization for computing, networking and storage resources, traditional perimeter-based security becomes even less effective. It has been proposed that the new model for data centre security will be software-based, implement micro-segmentation, and adopt the zero trust model (Miller and Soto, 2015).

There are at least four approaches to micro-segmentation (Young, 2017):

[8] `https://www.chef.io/chef/`

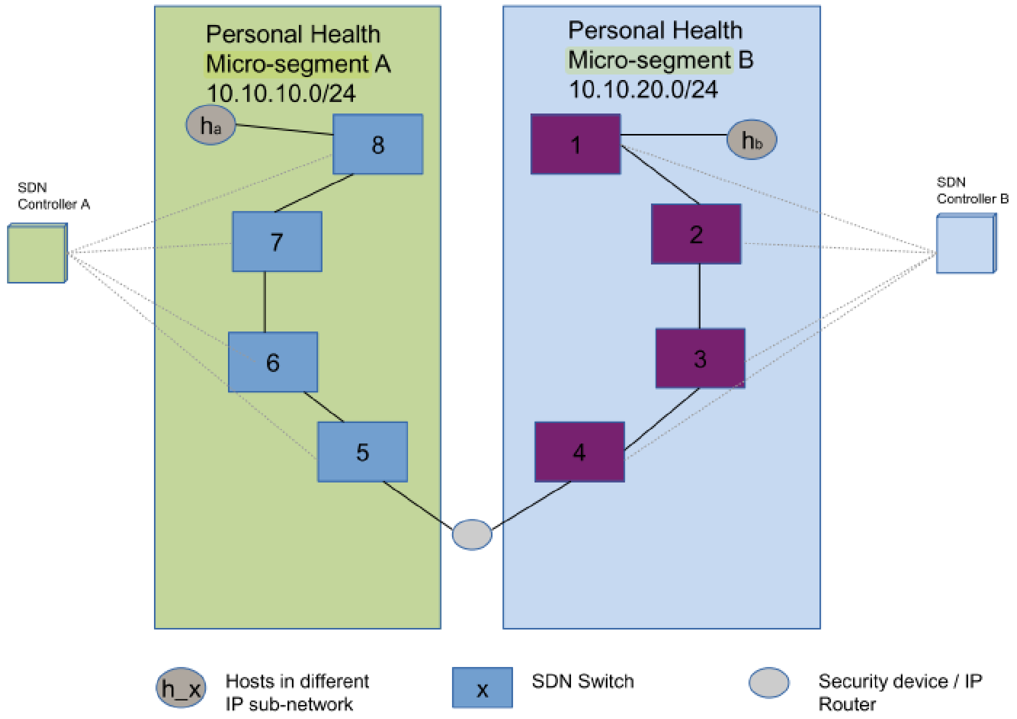


Figure 2.12: Micro-segmentation in a multidomain scenario, taken from Mämmelä *et al.* (2016)

- **Native** micro-segmentation is offered through a dedicated virtualization platform delivered by the operating system, hypervisor or infrastructure. Current native micro-segmentation solutions include VMWare NSX, Cisco ACI (Cisco Systems, 2017), Microsoft Server 2016[9] and Amazon Virtual Private Cloud[10].

- **Third-party** capabilities for micro-segmentation are offered as part of virtual firewall functionality integrated with 3rd party firewall platforms, including Cisco, CheckPoint[11] and Fortinet[12].

- **Overlay** micro-segmentation is typically implemented as a software-based agent installed on each host. Solutions in this space include Cisco ACI, CloudPassage[13],

[9] https://www.microsoft.com/en-in/cloud-platform/software-defined-networking

[10] https://docs.aws.amazon.com/AmazonVPC/latest/UserGuide/VPC_Security.html

[11] https://www.checkpoint.com/products/vsec-vmware-nsx/

[12] https://www.fortinet.com/solutions/enterprise-midsize-business/enterprise-firewall/internal-segmentation-firewall-isfw.html

[13] https://www.cloudpassage.com/wp-content/uploads/2016/08/microsegmentation-solution-brief.pdf

Illumio (Illumio, 2016), vArmour (VArmour, 2017) and Unisys[14].

- **Hybrid** method involves integrating native and third-party controls to achieve micro-segmentation.

2.4.1 VMware

VMware's virtualization platform covers a broad suite of products that are tailored for a diverse spectrum of virtualization use cases, of which some are described below.

- **VMware vCenter Server**[15], which provides a centralised platform for managing VMware vSphere environments that enable automation and provisioning of virtual infrastructures.
- **VMware NSX** (VMware, 2015b), which enables the creation of SDNs that are embedded in the hypervisor layer, decoupled from the underlying physical hardware. NSX is discussed in more detail in Section 2.4.2.
- **VMware vRealize Log Insight**[16], a syslog collector which also provides log management tools with informational dashboards, analytics and third party extensibility to provide operational visibility and troubleshooting across the physical and virtual environment. Traffic flows between the virtual machines are visualised to determine the effectiveness of the distributed firewall feature for enforcing micro-segmentation.
- **VMware vRealize Network Insight**[17], a netflow collector that delivers operations management capability for SDN and security that enables the building of optimised and secure network infrastructure spanning multiple environments. In particular, it can assist in micro-segmentation planning and deployment, enable visibility across virtual and physical networks and provide operational views to manage and scale VMware NSX deployments.

2.4.2 VMware NSX

VMware NSX is a software networking and security virtualization platform that provides stateful packet inspection and firewall capabilities to provide granular levels

[14] http://outreach.unisys.com/microsegmentation_UIS

[15] https://www.vmware.com/products/vcenter-server.html

[16] https://www.vmware.com/products/vrealize-log-insight.html

[17] https://www.vmware.com/products/vrealize-network-insight.html

of segmentation within virtual networks. This allows for rapid software-defined creation and deployment of complex network topologies and security profiles that can be distributed to and enforced by virtual ports, and which follow virtual machine migrations across the network (VMware, 2015b). The NSX platform includes a collection of logical networking services that are decoupled from hardware and can be programmatically provisioned, including switches, routers, firewalls, load balancers, VPN and distributed security.

Primarily targeted at software defined data centres (SDDCs), NSX provides the capability to implement fully isolated virtual networks, segmented virtual networks via fully automated native firewalls, and segmentation with third party services integration. These options are defined in Miller and Soto (2015) as follows:

- **Isolation:** Instead of relying on manually configured routing, ACLs or firewall rules, virtual networks are created in isolation and remain isolated until explicitly connected to each other. Additionally, virtual networks are isolated from the underlying physical infrastructure, and traffic between hypervisors is encapsulated.
- **Segmentation:** Replicates the functionality of physical firewalls or routers by allowing or denying traffic between network segments, except network services are provisioned along with programmatically created workloads and distributed to the hypervisor virtual switch (vSwitch), where segmentation and firewalling are enforced at the virtual interface. This means that traffic within the virtual network never leaves the virtualized environment.
- **Segmentation with third party integration:** Network services are distributed via a virtualized switch to create a logical pipeline of services applied to virtual network traffic, allowing for physical or virtual devices or third party services to be integrated and consumed by the pipeline. For example, security alerts by firewalls or anti-virus services can trigger security policies that execute automated processes, such as moving an infected host to a quarantined segment.

Within the context of the SDDC, implementing micro-segmentation is not readily achievable with existing tools and technologies, and it cannot be effectively implemented on an existing underlay network due to the absence of contextual grouping of workloads, against which dynamic network and security policies can be applied (VMware, 2015a). The NSX approach for implementing micro-segmentation within the SDDC is proposed as an alternative, superior solution to hair-pinning of east-west traffic described in Section 2.2.7, which tends to be computationally expensive and costly from a performance and resource point of view.

2.4.3 Cisco ACI

Cisco ACI is another commercial approach to data centre micro-segmentation based on a scalable spine and leaf network fabric (Cisco Systems, 2017). The ACI architecture enables the creation of application-aware endpoint groups, in which endpoints assigned to the group are normalised regardless of their type, origin or network location. This workload classification permits the enforcement of granular endpoint security policies through micro-segmentation for any application with physical or virtual workloads across any hypervisors, and thereby enables automated control over the flow of east-west traffic via the application policy infrastructure controller (APIC).

Some technical limitations of ACI, including a lack of visibility into the virtualized infrastructure, limited policy enforcement and a tendency for all traffic to be hair-pinned to the leaf switch may result in complex ACI configuration and performance constraints. Third party tools such as vArmour address this by introducing an additional security layer to complement the ACI micro-segmentation architecture (VArmour, 2017).

2.4.4 Illumio

Illumio Adaptive Security Platform is a commercial, overlay micro-segmentation solution that enables applications to be micro-segmented without relying on the network (Illumio, 2016). This capability is provided through a concept called virtual enforcement node (VEN), which resides within the workload operating system and leverages off built-in host operating system tools such as software firewalls and application programming interfaces to enforce relationships between workloads.

These relationships are governed by a centralised policy compute engine which receives contextual information on workloads from the VENs, and determines the correct security policy to be applied by the respective host firewall on each system (Illumio, 2016). Once the rules are deployed to individual VENs, only permitted traffic is allowed by the ruleset, which effectively creates a container around each application.

An advantage of this overlay approach is the removal of dependencies between the security policy and the physical or virtual network, as micro-segmentation based on application workloads is opaque to the underlying network infrastructure. If any of the workloads are migrated or decommissioned, these changes are detected by the policy compute engine, and the associated security policies are automatically and immediately amended. This means that Illumio's adaptive micro-segmentation approach does not require re-investment in a particular network or access fabric that would otherwise render it an infeasible option for networks and data centres. Additionally, the capability to encapsulate applications in a micro-segmentation container allows for the transparent bridging of on-premise data centres and hosted public cloud data centres on an application layer (Cummins and Sanabria, 2016).

2.5 IPsec Approach to Network Security

IPsec is a layer 3 security protocol for establishing secure network tunnels with authenticated and encrypted data flow to protect information against interception or tampering (Blaze *et al.*, 2002). This layer of security is provided by enabling a system to select one of two security protocols, IP Authentication Header (AH) and Encapsulating Security Payload (ESP) (Elkeelany *et al.*, 2002). IPsec has the capability to operate in transport mode, which provides end-to-end security by encrypting network traffic across the entire routing path between both endpoints, or tunnel mode by encrypting network traffic only for a subset of the routing path between the source and destination endpoints, typically across an untrusted network.

Data integrity checking validates that packet data sent between IPsec-enabled endpoints is not damaged or manipulated. This is done by calculating the cryptographic hashes of packets which are then encrypted and included in the packet. The receiving computer computes the same hash and compares it with the original hash. Matching hashes results in the packet being accepted and processed, however, if there is a hash mismatch, the packet is dropped. Ferguson and Schneier (1999) criticised IPsec for being too complex to be secure, due to the over-engineered design that was intended to support a range of different situations with different options. Subsequent revisions and improvements to the IPsec specifications have been made to address these concerns.

2.5.1 Active Directory

Microsoft Windows Active Directory[18] (AD) Domain Services (DS) is a logical, hierarchical directory services structure designed to store information about objects in a data store for retrieval by authorised users and administrators on the network.

Stored objects typically include shared resources such as servers, volumes, printers and network user and computer accounts. Access to these objects are secured through AD logon authentication and access control. With a single network logon, administrators can manage directory data and organisation throughout their network. Policy-based administration enables centralised management of all domain member computers joined to the AD domain through the application of Group Policy (Dias, 2002).

Singular or mutual authentication for client/server or server/server communications on the domain is provided by the Kerberos[19] protocol mechanism for secure authentication of user identity, using the Kerberos Key Distribution Centre (KDC) service integrated with AD.

2.5.2 Group Policy Object

Group Policy[20] is an AD based hierarchical infrastructure tool that provides centralised management and configuration of user and computer settings, including operating systems, applications, security and networking policies.

A Group Policy Object (GPO) is a collection of rules created by a domain administrator to enforce specific configuration settings for a particular group of computers or users within the domain. When a computer connects to the domain, these settings are automatically deployed and merged with the local GPO stored on the computer, and then applied to the computer's active configuration.

This enables a systems administrator to create and link a GPO with customised configuration settings to an Organizational Unit (OU) that contains specific domain users or computers. When applied, the GPO enforces the policy defined by the domain administrator, which overrides any local user or computer settings.

[18] https://docs.microsoft.com/en-us/windows-server/identity/ad-ds/get-started/virtual-dc/active-directory-domain-services-overview

[19] https://technet.microsoft.com/en-us/library/cc780469(v=ws.10).aspx

[20] https://technet.microsoft.com/en-us/library/hh147307(v=ws.10).aspx

2.5.3 Microsoft Windows Firewall and IPsec

Within the Microsoft Windows platform, IPsec functionality has long been available as part of the Windows operating system (Downin and LaFountain, 2001), and has been improved and extended over subsequent operating system release iterations to provide various centrally managed options for securing network traffic. IPsec is tightly integrated with the built-in Windows Firewall with Advanced Security component to allow for flexible configuration options for enabling fine grained connection security rules to identify, specify and enforce protection of endpoints (Bishop, 2009).

2.5.4 Server and Domain Isolation using IPsec and Group Policy

The objective of IPsec server and domain isolation (SDI) is to mitigate the threat posed by unauthorised access to a trusted computer by an untrusted computer. This is achieved by restricting inbound network access based on the ability to successfully authenticate as a domain member computer, using the IPsec Internet Key Exchange (IKE) security negotiation protocol. Following this, validated user authentication onto the domain is required. In the event of a successful connection between two endpoints, all upper layer protocol and application connections between the two computers are protected by IPsec security associations (Clark *et al.*, 2006).

Domain isolation (Microsoft Corp, 2016a) allows domain computers to receive unsolicited inbound traffic only from other members of the isolated domain, while being able to send traffic to any domain or non-domain endpoints. Computers in a boundary zone are part of the isolated domain, and can accept connections from untrusted endpoints. Trusted non-domain endpoints are allowed to communicate with isolated domain computers, while untrusted non-domain endpoints are isolated from the domain. A representative example of a domain isolation implementation is shown in Figure 2.13.

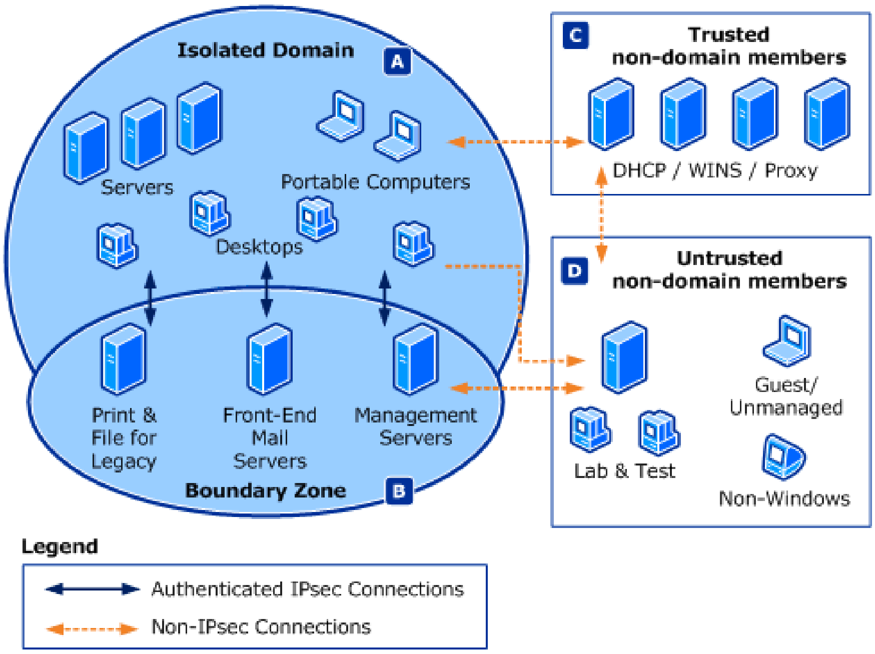


Figure 2.13: Domain isolation policy design, taken from Microsoft Corp (2016a)

Server isolation (Microsoft Corp, 2016b) can be layered on top of a domain isolation implementation or deployed only to the computers that must participate, and functions by specifying the GPO based connection, authentication and security group membership rules that must be met, for example a Network Access Group, before allowing connectivity between the two endpoints. An example of server isolation functionality is shown in Figure 2.14.

The two separate SDI concepts can be implemented separately or layered together to enable logical isolation and prevent untrusted and unauthenticated endpoints from connecting to a trusted endpoint on the IPsec-enabled domain. In a scenario where communications need to be restricted and secured between trusted endpoints only, IPsec SDI can be used to authenticate and encrypt traffic between two or more servers within the domain (Platts, 2008).

Limitations of IPsec SDI include potential decreased network performance, inadvertent crossing of security boundaries, as well as the operational impact of not being able to accommodate devices that cannot communicate using IPSec, such as different operating system platforms, routers, printers and other networking devices (Holtzman, 2005). This means that IPsec SDI is best suited to organisations

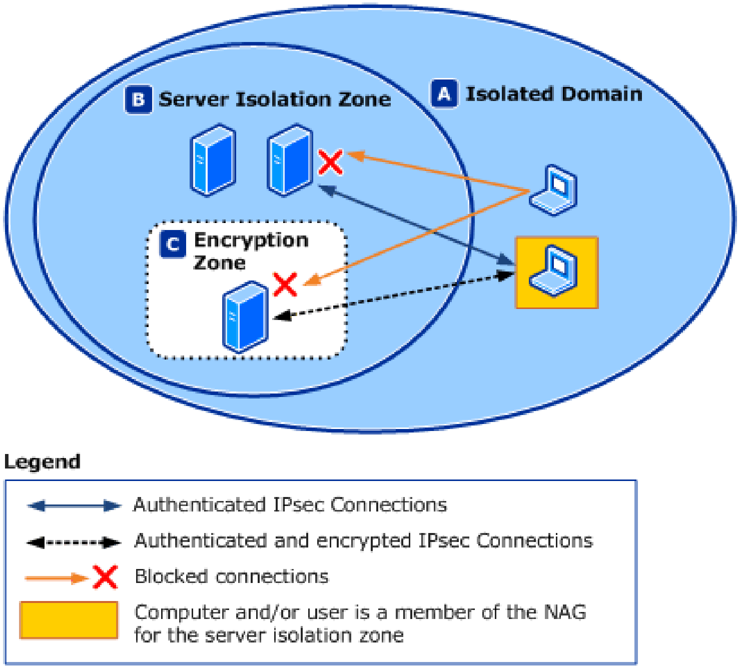

B Server Isolation Zone
A Isolated Domain
C Encryption Zone
Legend
Authenticated IPsec Connections
Authenticated and encrypted IPsec Connections
Blocked connections
Computer and/or user is a member of the NAG for the server isolation zone

Figure 2.14: IPsec server isolation policy design, taken from Microsoft Corp (2016b)

that have standardised on the Windows platform and which have other security controls in place to protect endpoints that cannot form part of SDI.

2.6 Summary

This chapter provided an overview of network security constructs and controls and highlighted the security challenges inherent in managing network security risks in the absence of proper preventive controls. The literature review examined the various approaches and security controls available that were designed or could be adapted to address these challenges, and framed the context for the micro-segmentation experiments discussed in Chapter 3.

Chapter 3

Experimental Setup

This chapter explains the experimental approach adopted, and details the methodology and technical specifications of each of the three experiments conducted.

3.1 Overview of Experiments

To evaluate various existing tools and technologies that could be utilised to implement cost-effective secure micro-segmentation to control east-west traffic between devices located within the same VLAN, three discrete experiments were undertaken.

As cost effectiveness is one of the main drivers for the research, the experimental scenario assumes an organisation with a limited IT budget, a basic, functional computing infrastructure and competent, non-specialised IT staff. These constraints generally apply to small-to-medium sized organisations, although this does not preclude adoption of micro-segmentation solutions by larger enterprises. Consequently, the scope for implementing micro-segmentation is restricted to physical network switches and existing operating system tools, specifically IPsec, due to their prevalence within organisations. These two experiments are contrasted with a commercial micro-segmentation solution to determine the associated effort, scalability and effectiveness.

1. The first experiment examines the feasibility of leveraging existing functionality within physical network switches to implement micro-segmentation by enabling simplified ACLs to control east-west traffic within a network segment or VLAN.

2. The second experiment implements VMware NSX, a commercially available Type 1 hypervisor with SDN capabilities that enables micro-segmentation as a function of kernel-based distributed firewalls. VMware's functionality and effectiveness are evaluated in this experiment as a baseline metric for comparison with the other micro-segmentation approaches.

3. The third experiment revisits Microsoft's implementation of IPsec Server and Domain Isolation (SDI) functionality to determine whether it can be effectively re-purposed to implement micro-segmentation.

3.2 Experiment 1: Switch ACLs

Based on the knowledge that network switches constitute one of the most numerous and readily available components of an IT infrastructure, the suitability of switch ACLs to control east-west traffic flow is explored in this section.

3.2.1 Approach

ACLs are a well understood mechanism for filtering ingress and egress traffic. Conceptually, a layer 2 switch that supports extended ACLs can be configured to enforce an explicit predefined security policy to restrict east-west traffic between two or more endpoints within the same VLAN. To evaluate the suitability of switch ACLs, a simple security policy is defined to indicate permitted and prohibited traffic flows, and then translated into a corresponding extended ACL applied on a physical layer 2 switch. The outcome is evaluated by recording traffic flows across the switch using a packet capture tool, and supported by visualisation of the traffic flows generated by a graph visualisation tool.

3.2.2 System Specifications

A Cisco Catalyst 2960-S[1] series layer 2 managed network switch with support for ACL functionality was procured. A private IP address space of 10.64.0.0/24 was split into tagged VLAN subnets as shown in Table 3.1.

To simplify the scope of the experiment, the switch gateway and physical test machines connected to the Cisco switch were assigned static IP addresses and hostnames as listed in Table 3.2.

[1] https://www.cisco.com/c/en/us/products/switches/catalyst-2960-s-series-switches/index.html

Table 3.1: VLAN allocation

VLAN	IP Subnet	Netmask	Description
VLAN 99	10.64.0.0	255.255.255.240	Management VLAN
VLAN 12	10.64.0.0	255.255.255.240	Experimental VLAN
VLAN 13	10.64.0.0	255.255.255.240	Reserved VLAN

Table 3.2: Host IP assignment

Host	IP Address	Netmask	VLAN
Gateway	10.64.0.33	255.255.255.240	VLAN 12
Host A	10.64.0.34	255.255.255.240	VLAN 12
Host B	10.64.0.35	255.255.255.240	VLAN 12
Host C	10.64.0.41	255.255.255.240	VLAN 12

The physical hosts connected to the switch consisted of three workstations running the Microsoft Windows operating system platforms. All software firewalls available on the workstations were disabled to ensure that traffic flows were not artificially restricted. In addition to this, the hosts were joined to a standard AD domain, and no restrictive GPOs which could potentially interfere with the experimental setup were applied to the server and workstation OUs.

3.2.3 Methodology

To verify that there were no artificial traffic restrictions that could potentially influence the outcome of the experiment, all enabled ports on the Cisco Catalyst switch had a default switch configuration applied that explicitly permitted traffic to be forwarded between all switch ports as required. The following nmap port scan parameters were initiated from hosts A and B against the entire VLAN /24 subnet:

nmap.exe -sT -vv 10.64.0.0/24

The resultant traffic flows were captured using Wireshark[2], a packet capture tool that was installed on hosts A and B, and visualised to verify that the hosts were able to initiate and respond to connection requests. This confirmed that there were no restrictive policies or settings in place that would otherwise inhibit traffic flow and generate misleading results.

[2] https://www.wireshark.org/

Security Policy

An organisational security policy specifies the rules to be applied to traffic flow from hosts connected to the switch. These rules determine whether certain hosts are permitted to communicate with other hosts, and the resultant policy is translated into corresponding ACLs. In this context, the simple security policy was defined as follows:

- **Rule 1**: Host *A* (10.64.0.34) may talk to any device except Host *B* (10.64.0.35)
- **Rule 2**: Host *B* (10.64.0.35) may talk to any device except Host *A* (10.64.0.34)
- **Rule 3**: Both Hosts *A* and *B* may talk to Host *C* (10.64.0.41)

Figure 3.1 illustrates the expected behaviour for Rule 1, in which Host *A*′s connectivity to Host *C* should be permitted. Conversely, Host *A*′s connectivity to Host *B* should be blocked at the switch and should never reach Host *B*. This demonstrates the concept of east-west traffic within the switch controlled by ACL based micro-segmentation, contrasted against north-south traffic flow controlled by the firewall between the switch and the wide area network uplink.

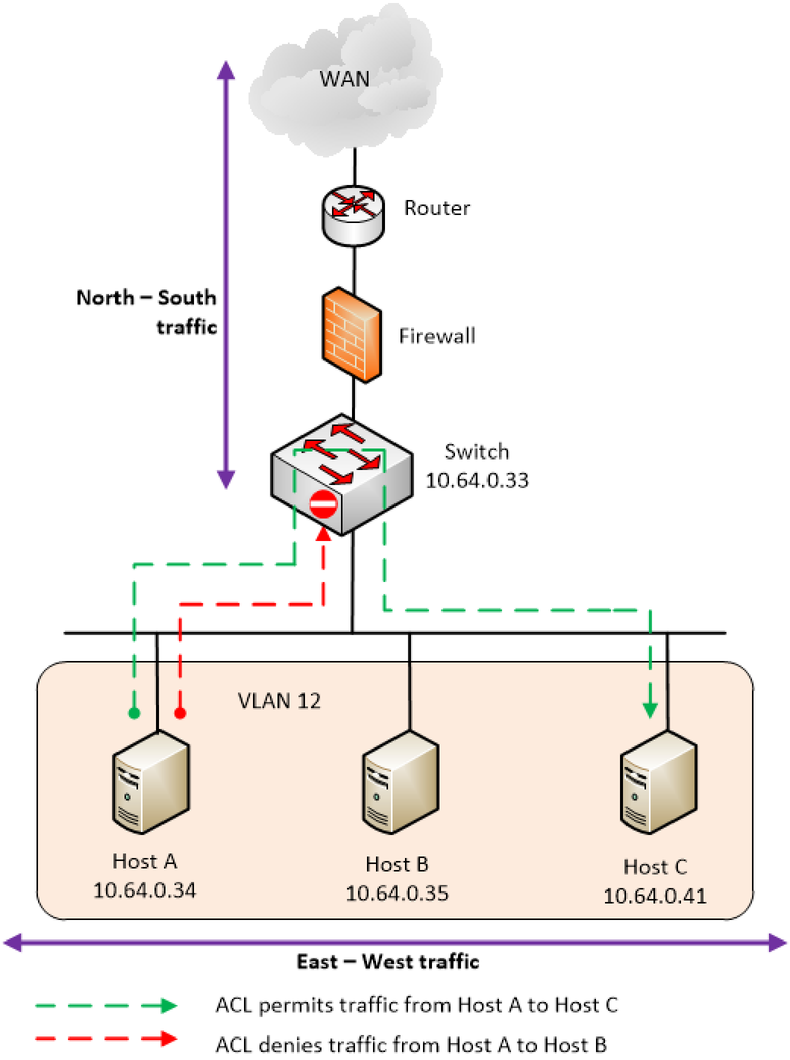


Figure 3.1: ACL connectivity rules for Host A, adapted from Cisco (2014)

Figure 3.2 shows the traffic flow restrictions for Rule 2 in which Host *B*'s connectivity to Host *C* is permitted, while connectivity to Host *A* is blocked.

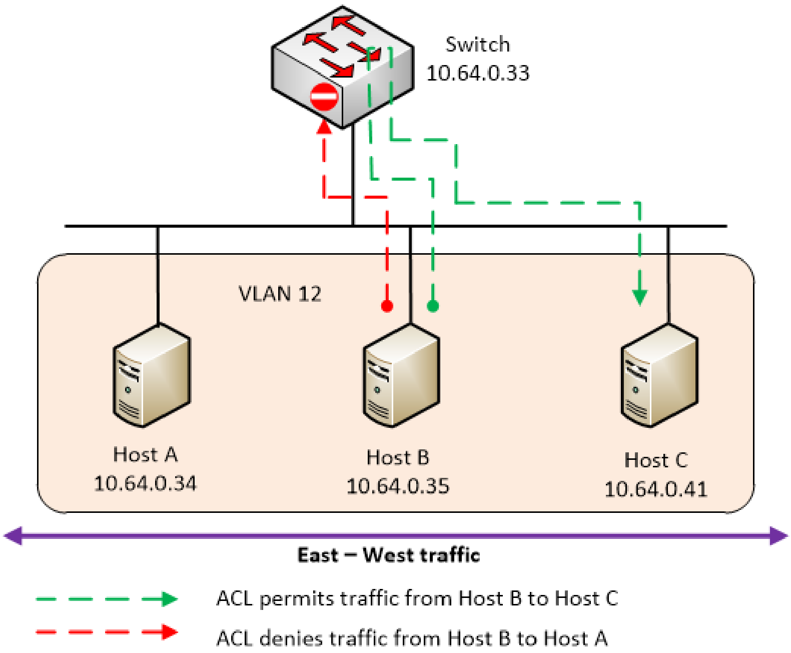


Figure 3.2: ACL connectivity rules for Host B

The traffic flow restrictions for Rule 3 are shown in Figure 3.3, where Hosts *A* and *B* can both connect to Host *C*. As the other ACL rules in Figures 3.1 and 3.2 are still in force, Hosts *A* and *B* are still unable to connect to each other.

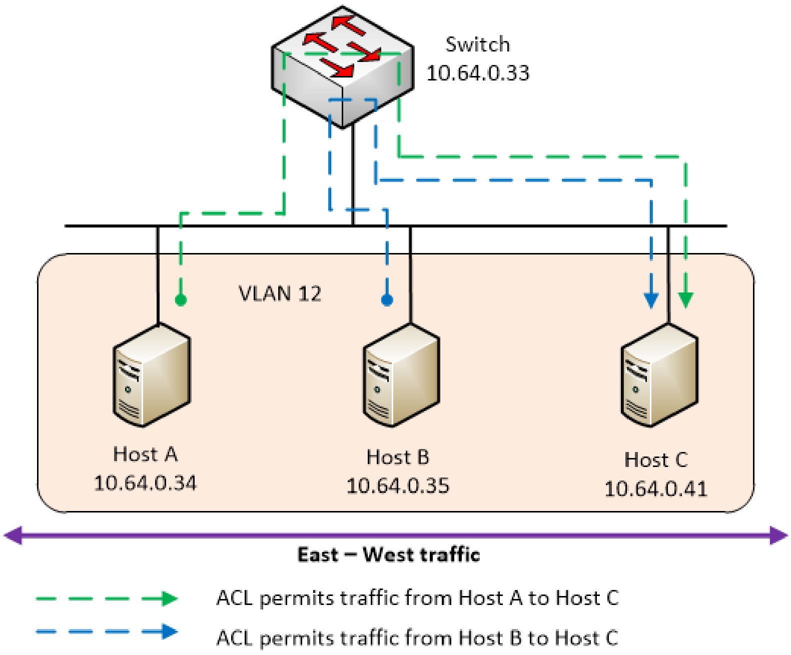


Figure 3.3: ACL connectivity rules for Host C

Switch ACL configuration

For the purposes of this experiment, a switch virtual interface (SVI) was configured on the Cisco Catalyst layer 2 switch to provide a layer 3 interface to the VLAN. By applying ACLs on this interface, extended access lists can be used, allowing both source and destination addresses and protocol information to be specified for ACL matching operations, as opposed to standard IP access lists that only allow source addresses for matching operations.

A set of ACL entries were then constructed based on the defined security policy described in Section 3.2.3, and applied to the SVI on the Cisco Catalyst switch to which the hosts were physically connected. An extract of these ACL entries and their functional descriptions are shown in Table 3.3.

Table 3.3: Experiment 1: Extract of ACL entries

#	ACL entry (Description of functional rule)
1	*extended IP access list VLAN-12* (Define the extended IP access list for VLAN 12)
2	*permit icmp 10.64.0.32 0.0.0.15 10.64.0.0 0.0.0.15 ace-priority 20 type any code any* (Allow any ICMP message types to all devices on VLAN 99)
3	*deny icmp 10.64.0.32 0.0.0.15 10.64.0.48 0.0.0.15 ace-priority 60 type any code any* (Block any ICMP message types to all devices on VLAN 13)
4	*deny ip host 10.64.0.34 host 10.64.0.35 ace-priority 140* (Block Host A connectivity to Host B within VLAN 12)
5	*deny ip host 10.64.0.35 host 10.64.0.34 ace-priority 160* (Block Host B connectivity to Host A within VLAN 12)
6	*permit ip host 10.64.0.41 host 10.64.0.34 ace-priority 180* (Permit Host C connectivity to Host A within VLAN 12)
7	*permit ip host 10.64.0.41 host 10.64.0.35 ace-priority 200* (Permit Host C connectivity to Host B within VLAN 12)
8	*deny ip 10.64.0.0 0.0.0.255 10.64.0.48 0.0.0.15 ace-priority 220* (Block all connectivity from VLAN 12 to VLAN 13)
9	*permit ip any any ace-priority 240* (Allow connectivity from all hosts within VLAN 12 to any destination)

This effectively means that, although Hosts A and B reside within the same VLAN and therefore do not cross a firewall interface, the intent behind the security policy and ACL rules is to micro-segment them from being able to communicate with each other, while permitting traffic to and from other hosts.

The ACLs were applied on the SVI, resulting in the security policy being enforced on packets traversing the interface by inspecting and comparing each packet against the

ACL rules. When the first matching ACL entry is found for each packet, the policy defined in the ACL rule determines whether the packet is forwarded to its destination or discarded.

3.2.4 Evaluation

Following the activation of the ACLs on the SVI, a packet capture was initiated on the wire to record traffic passing through the switch. This was followed by running a netcat[3] listener on Hosts A, B and C using the following parameters:

nc.exe -vv -p23 -L

Telnet connections were then initiated from Host *A* to *B*, Host *B* to *A*, and Host *A* to *C* to verify that traffic is either permitted or denied by the switch ACLs as appropriate. Additionally, an nmap[4] port scan was also initiated from Host *A* against Host *B*, Host *B* against Host *A*, and Host *A* against Host *C* using the following parameters:

nmap.exe -sT -vv 10.64.0.x where x = .34, .35, .41.

After completion of the port scan, the packet captures were saved to a file for analysis on the effectiveness of switch ACL based micro-segmentation, which is determined by comparing the traffic flows captured before and after enabling ACLs on the SVI. Analysis of the packet capture and the outcomes are discussed in the next chapter (see Section 4.2).

3.3 Experiment 2: VMware NSX

In this experiment, a commercial Type 1 hypervisor with SDN capabilities was implemented to evaluate the effectiveness of its networking and security module, which amongst other features is specifically designed to implement micro-segmentation transparently to the virtualized hosts.

3.3.1 Approach

To establish a baseline for assessing the resource requirements and effectiveness of a dedicated micro-segmentation implementation, several commercial micro-segmentation solutions were considered. Illumio Adaptive Security Platform[5] was

[3] http://sectools.org/tool/netcat/

[4] https://nmap.org/

[5] https://www.illumio.com/home

the original choice, as it allows for implementation of micro-segmentation on existing native bare-metal hardware without requiring virtualization of the existing environment, is infrastructure agnostic, and supports coarse-grained, micro, process-based and user segmentation. However, since the Illumio trial software was scheduled for release only in Q2 2018, it was unsuitable for use in this research.

The VMware product family was then selected as the platform of choice due to its level of maturity in the virtualization / SDN market (Ramel, 2017) and, in particular, its built-in support for distributed firewalls through the VMware NSX platform that can be deployed to achieve micro-segmentation.

This experimental setup required the deployment of a virtualized environment containing a proof of concept infrastructure environment hosting an AD domain controller (DC), member servers and workstations. Post-installation, a distributed firewall ruleset based on a simple security policy described in Section 3.3.3 was configured and published. The effectiveness of the VMware NSX platform's distributed firewall functionality in enforcing micro-segmentation within the VLAN containing the member servers and workstations was determined by analysing the traffic logs from the virtual machines that were forwarded to VMware vRealize Log Insight, a log visualization tool.

3.3.2 System Specifications

Initially VMware Workstation[6], a Type 2 hypervisor that allows multiple operating systems to be run as virtual machines on top of a Windows or Linux platform, was configured as the base experimental platform. However, the existing hardware constraints proved to be insufficient to service the combined resource requirements of VMware NSX, and necessitated a migration of the experimental platform to VMware vSphere ESXi, a Type 1 hypervisor for server virtualization. ESXi was deployed on a Dell PowerEdge T430 bare-metal server to provide the base platform for the installation of the VMware family suite of virtualization products previously described in Section 2.4.1, and which were installed on top of vSphere ESXi using temporary trial licences.

Lab Environment

Implementing a Type 1 VMware platform for the experiment was a fairly complex endeavour, as multiple dependencies had to be met through the installation and

[6] https://www.vmware.com/products/workstation-pro.html

configuration of several VMware appliances before a fully functional virtualization environment with micro-segmentation capabilities could be realised.

An architectural overview of the planned VMware platform installation is shown in Figure 3.4, which illustrates the physical bare metal server on which the ESXi hypervisor is installed, together with the SDN based virtual switches and distributed firewall layer between the switch and the virtual machines. A distinction is also made between the virtual appliances that provide enhanced functionality for the virtualization platform, and the virtual machines that comprise the experimental setup.

Although two bare metal servers and hypervisors are depicted in the architectural diagram, for the purposes of the experiment only one bare metal server instance was deployed as no clustering or fail-over capabilities were required to support the experiment.

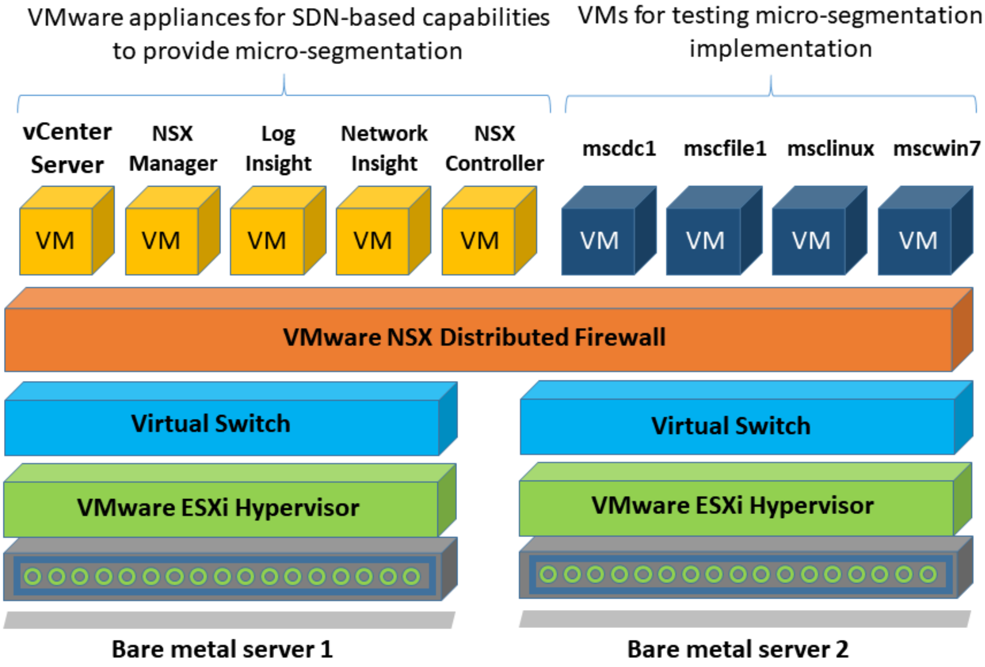


Figure 3.4: VMware infrastructure for micro-segmentation, adapted from Wilmington (2016)

The VMware appliances listed in Table 3.4 were built using the standard installation images and documentation provided by VMware.

After installation and configuration of VMware vSphere ESXi, VMware vCenter Server was installed as a virtual machine on the ESXi host based on the standard vCenter

Table 3.4: Experiment 2: VMware appliances

Appliance	Hostname	IP Address	Function
vSphere ESXi	esxi.compsci.local	192.168.1.8	hypervisor
vCenter Server	photon-machine.compsci.local	192.168.1.9	centralised virtualization platform
NSX Manager	nsxmanager.compsci.local	192.168.70	distributed firewall
Log Insight	loginsight.compsci.local	192.168.1.112	syslog
Network Insight	netinsight.compsci.local	192.168.1.12	netflow

Server installation and setup procedure (VMware, 2017). Within vCenter Server, a new virtual data centre was created as a container for all inventory objects, hosts and virtual machines. A datastore was also created to contain the virtual storage for the hosts and virtual machines. The ESXi host was then imported into vCenter Server as a new standalone host.

NSX Manager

Within vCenter Server, the VMware NSX Manager for vSphere appliance was deployed as an open virtual machine template to the VMware ESXi host, and configured based on the standard NSX Administration Guide (VMware, 2015b). The vCenter server was registered with NSX Manager, followed by configuration of the lookup service on NSX Manager to enable secure authentication through single sign-on functionality with vCenter server.

The next step was to install and assign the NSX for vSphere trial licence to activate the distributed firewall functionality. An optional, additional configuration step was the NSX Controller, which is

> *an advanced distributed state management system that provides control plane functions for NSX logical switching and routing functions. It serves as the central control point for all logical switches within a network and maintains information about all hosts, logical switches (VXLANs), and distributed logical routers.* (VMware, 2014b)

For the purposes of the experiment, it was not strictly necessary to deploy an NSX Controller Cluster, or Edge services, since the distributed logical routers or VXLAN networking capabilities were not used apart from the distributed firewall. To preempt inadvertent connectivity issues arising from accidental misconfiguration of the distributed firewall, the vCenter Server appliance was excluded from distributed firewall protection by adding it to the NSX Manager Exclusion List.

The last installation step was the process of preparing host clusters for NSX, in which NSX kernel modules were installed on ESXi hosts that were members of vCenter clusters, and the NSX control-plane and management-plane fabric was built. The kernel modules ran within the hypervisor kernel and provided the distributed firewall service transparently to the virtual machines.

Networking

A standard VMware ESXi VLAN was configured to represent a typical flat, unsegmented VLAN. The VMware vSphere ESXi host, vCenter Server, NSX Manager and vRealize Log Insight and Network Insight servers were placed in the same VLAN as the virtualised servers and workstations to test the distributed firewall capability.

Afterwards, a new distributed switch was created, containing a distributed port group to which the four virtual machines created above were added and assigned to an uplink port. The VMs were then migrated from the standard ESXi virtual switch to the new distributed switch. Within NSX Manager, a new VXLAN was created to enable layer 2 logical switching across hosts, and assigned to the newly created distributed switch. This was followed by the creation of a new transport zone, which controls the hosts that a universal logical switch can reach.

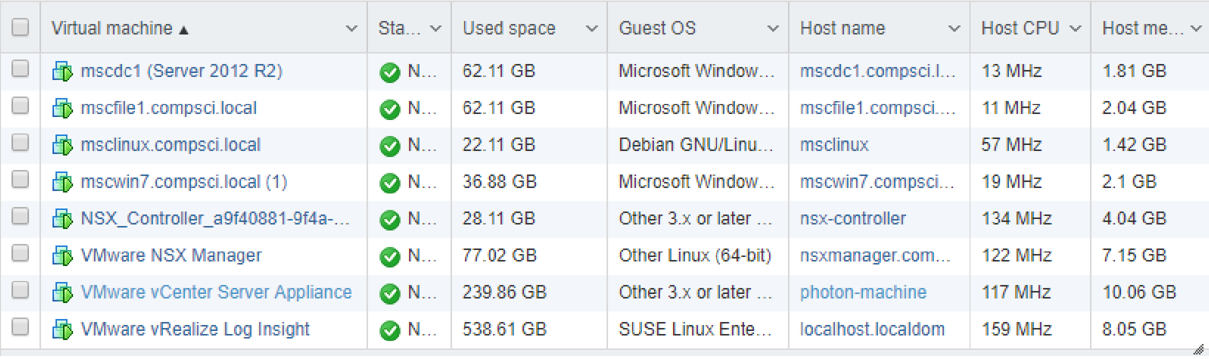
Server and Workstation Configuration

The nature of the experimental setup was designed to test the feasibility of implementing distributed firewalls as a micro-segmentation proof of concept to restrict east-west traffic within a VLAN. The parameters of the experiment were limited to a virtualised infrastructure and network environment comprising a heterogeneous mix of operating system platforms to reflect a standard corporate IT environment. The virtual machines defined in Table 3.5 were built based on clean operating system images obtained from the respective vendors.

The end result is shown in Figure 3.5, which depicts all the virtual machines installed and running on top of the vSphere ESXi hypervisor.

Table 3.5: Experiment 2: VMware virtual machines

Long hostname	**Short name**	**Operating System**	**IP Address**	**Description**
mscdc1.compsci.local	mscdc1	Windows Server 2012 R2	192.168.1.4	AD, DNS and DHCP
mscfile1.compsci.local	mscfile1	Windows Server 2016	192.168.1.5	File and web server
msclinux.compsci.local	msclinux	GNU/Linux Mint 18.2 Sonya	192.168.1.7	Application server
mscwin7.compsci.local	mscwin7	Windows 7 SP2	192.168.1.15	Workstation



□	Virtual machine ▲	Sta...	Used space	Guest OS	Host name	Host CPU	Host me...
□	mscdc1 (Server 2012 R2)	✓ N...	62.11 GB	Microsoft Window...	mscdc1.compsci.l...	13 MHz	1.81 GB
□	mscfile1.compsci.local	✓ N...	62.11 GB	Microsoft Window...	mscfile1.compsci....	11 MHz	2.04 GB
□	msclinux.compsci.local	✓ N...	22.11 GB	Debian GNU/Linu...	msclinux	57 MHz	1.42 GB
□	mscwin7.compsci.local (1)	✓ N...	36.88 GB	Microsoft Window...	mscwin7.compsci...	19 MHz	2.1 GB
□	NSX_Controller_a9f40881-9f4a-...	✓ N...	28.11 GB	Other 3.x or later ...	nsx-controller	134 MHz	4.04 GB
□	VMware NSX Manager	✓ N...	77.02 GB	Other Linux (64-bit)	nsxmanager.com...	122 MHz	7.15 GB
□	VMware vCenter Server Appliance	✓ N...	239.86 GB	Other 3.x or later ...	photon-machine	117 MHz	10.06 GB
□	VMware vRealize Log Insight	✓ N...	538.61 GB	SUSE Linux Ente...	localhost.localdom	159 MHz	8.05 GB

Figure 3.5: VMware vSphere ESXi hypervisor, appliances and virtual machines

3.3.3 Methodology

The VMware NSX logical firewall functionality offers two main components - the distributed firewall for controlling east-west traffic, and the edge firewall for controlling north-south traffic. For the purposes of the experiment, only the distributed firewall component was configured to enable micro-segmentation capabilities.

Security Policy

A security policy is a high level description of rules pertaining to devices connected to the virtual network. These rules govern whether traffic flowing to or from a particular device should be permitted or denied to comply with the security policy objectives, and are then translated into corresponding distributed firewall rules. In this context, a simple security policy was defined as follows:

- **Rule 1:** Host *mscdc1* (192.168.1.4) may talk to any device within the VLAN

- **Rule 2:** Host *mscfile1* (192.168.1.5) may talk to any device within the VLAN, except host *msclinux* (192.168.1.7)
- **Rule 3:** Host *msclinux* (192.168.1.7) may talk to hosts *mscdc1* (192.168.1.4) and *mscwin7* (192.168.1.15) only
- **Rule 4:** Host *mscwin7* (192.168.1.15) may talk to hosts *mscdc1* (192.168.1.4) and *mscfile1* (192.168.1.7) only

Distributed Firewall

The default installation of VMware NSX Distributed Firewall comes pre-configured with permit rules for layer 2 (Ethernet) traffic, which are converted to MAC addresses in the kernel; and layer 3 (IP) traffic, which are converted to IP addresses for the kernel to process. For the purposes of the experiment, only Layer 3 rules were created based on the security policy in Section 3.3.3 to enforce micro-segmentation.

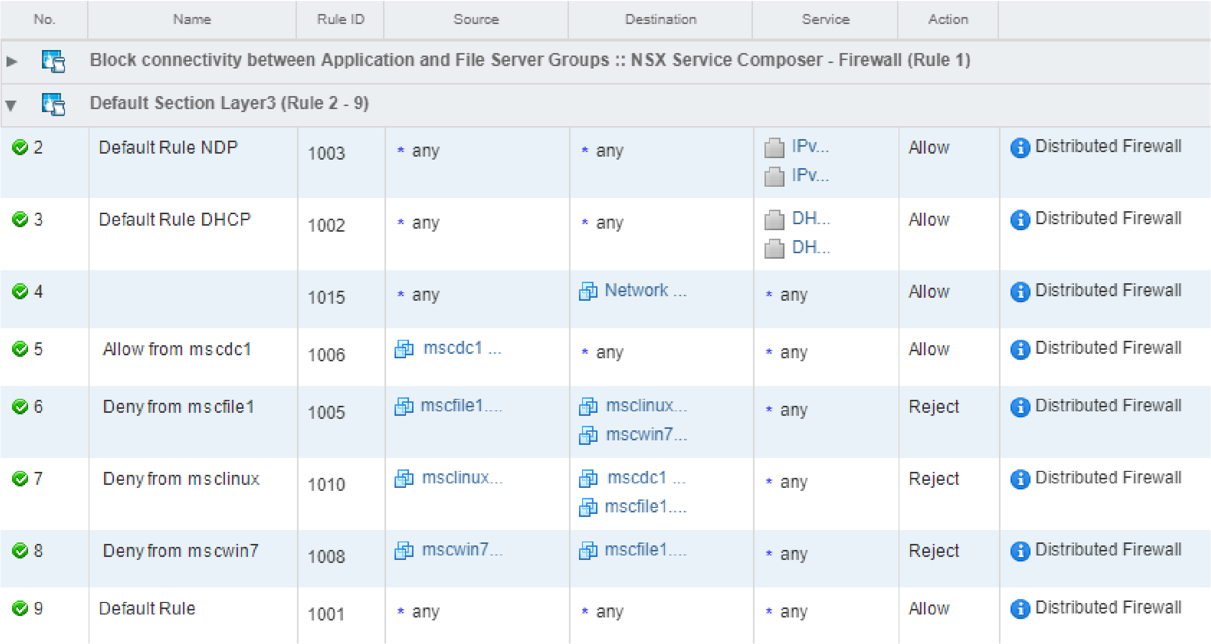
It is of particular interest that different object types can be defined in the distributed firewall rule's source or destination field (VMware, 2014b), namely:

- **Cluster:** a group of hosts
- **Data centre:** basic physical infrastructure including virtualization servers, storage networks and arrays, IP networks, servers and desktops
- **Distributed port group:** defines NIC teaming, failover, load balancing, VLAN, security, traffic shaping and other policies
- **IP sets:** a group of individual IP addresses
- **Legacy port group:** aggregates multiple ports under a common configuration
- **Logical switch:** creates logical broadcast domains or segments to which an application or virtual machine can be logically wired
- **Resource pool:** the aggregated physical compute hardware, including CPU and memory, allocated to virtual machines
- **Security group:** enables dynamic membership criteria based on security tags, VM name or logical switch name
- **Virtual app:** an agentless application virtualization solution
- **Virtual machine:** an emulation of a computer system

- **Virtual NIC:** emulates a full-fledged Ethernet network card

This means that distributed firewall rules can be logically applied to an object group instead of explicit IP addresses or hostnames. For example, if a virtual machine object group is defined, the distributed firewall policy still applies to virtual machines within that object group even if their IP addresses have changed. Another example of the flexibility of object groups is the placement of virtual machines in a security group. Once the rules for the security group are defined, the associated distributed firewall rules will follow that security group and its associated virtual machines, even if the virtual machines are migrated to a different port group, logical switch or data centre. The implication is that any changes to an infrastructure or network environment will not necessarily compromise the organisational policy expressed through the micro-segmentation firewall rules, as these automatically follow the virtual machine across any logical networking boundaries.

In configuring the distributed firewall rules, the virtual machine object group is used to define the firewall source and destination objects, instead of IP addresses. Based on the security policy set out in Section 3.3.3, the firewall sub-menu within the VMware NSX Networking & Security section is configured with customised distributed firewall rules as documented in Figure 3.6.



No.	Name	Rule ID	Source	Destination	Service	Action	
▶	Block connectivity between Application and File Server Groups :: NSX Service Composer - Firewall (Rule 1)						
▼	Default Section Layer3 (Rule 2 - 9)						
✓ 2	Default Rule NDP	1003	* any	* any	IPv... IPv...	Allow	Distributed Firewall
✓ 3	Default Rule DHCP	1002	* any	* any	DH... DH...	Allow	Distributed Firewall
✓ 4		1015	* any	Network ...	* any	Allow	Distributed Firewall
✓ 5	Allow from mscdc1	1006	mscdc1 ...	* any	* any	Allow	Distributed Firewall
✓ 6	Deny from mscfile1	1005	mscfile1....	msclinux... mscwin7...	* any	Reject	Distributed Firewall
✓ 7	Deny from msclinux	1010	msclinux...	mscdc1 ... mscfile1....	* any	Reject	Distributed Firewall
✓ 8	Deny from mscwin7	1008	mscwin7...	mscfile1....	* any	Reject	Distributed Firewall
✓ 9	Default Rule	1001	* any	* any	* any	Allow	Distributed Firewall

Figure 3.6: VMware NSX distributed firewall rules

3.3.4 Evaluation

To assess the effectiveness of distributed firewall based micro-segmentation, basic network traffic was generated including ping requests, telnet connections and nmap port scans between the virtual machines, before and after the distributed firewall rules were enabled, as documented in Section 3.2.4.

The VMware Log Insight and Network Insight tools were used to perform analytics on the distributed firewall traffic based on netflow logs. In addition, graphs were derived from these analytics to determine the effectiveness of micro-segmentation within the virtualized environment when measured against the simple security policy. These are discussed in the next chapter (Section 4.3).

3.4 Experiment 3: IPsec Server and Domain Isolation

In Section 3.2 the use of physical or virtual switch ACLs as a micro-segmentation option was discussed, followed by the implementation of a commercial hypervisor in Section 3.3 to provide micro-segmentation through SDN. This section explores the concept of implementing micro-segmentation using IPsec SDI.

3.4.1 Approach

Broadly speaking, the objective of the experimental setup in this section is to determine whether micro-segmentation can be implemented by reusing existing tools, technologies or approaches, including leveraging off currently available commercial or open-source solutions. While commercial products are available that render micro-segmentation in data centres a solved problem, from a cost perspective the challenge is to determine whether a functionally equivalent micro-segmentation implementation is possible without incurring substantial capital investment or re-engineering the existing network and infrastructure architecture.

Given the dominance of Microsoft Windows in most corporate IT environments, this experimental setup assesses whether IPsec SDI functionality as discussed in the previous chapter (see Section 2.5.4, combined with Advanced Firewall Security and AD GPO (Microsoft Corp, 2016c), can be re-purposed as a viable and functional alternative for implementing micro-segmentation.

To validate this hypothesis, a proof of concept virtualised network and infrastructure environment hosting an AD DC, member servers and member workstations was built.

The environment was intended to simulate a limited scale deployment of a typical enterprise network which would provide insight into the effectiveness and limitations of deploying IPsec SDI throughout the environment.

Following this, an IPsec GPO was constructed, based on the simple security policy described in Section 3.4.3, and linked to the respective AD OUs containing the member servers and workstations. The GPO centrally manages the IPsec configuration settings and supporting firewall rules deployed throughout the environment. The effectiveness of IPsec SDI was assessed through a packet capture analysis of network traffic flowing through the virtual network interface.

3.4.2 System Specifications

Installing a new AD DS environment is a well documented procedure that can be followed to replicate the experimental setup. In the subsections that follow, the lab setup is discussed, followed by an overview of the VMware virtual networking setup as well as configuration of member servers and workstations. This is followed by a more detailed description of the IPsec SDI settings configured as a GPO, including IPsec connection rules and its deployment.

Lab Environment

A Type 2 hypervisor, VMware Workstation 12.5.7, was used as the virtualisation platform to support the experiment, although any alternative Type 1 or 2 hypervisor such as Virtualbox[7], Hyper-V[8], KVM[9] and Xen[10] are also alternative options, as there is no dependency on any particular hypervisor to implement an IPsec based micro-segmentation proof of concept.

Networking

A single VMware host-only virtual network was created and tagged as *VMnet19* to represent a typical flat, unsegmented VLAN. All network traffic flows generated by virtual machines were confined to this VLAN, and were not able to exit the virtual switch interface via network address translation or network bridging. This configuration prevents inadvertent packet traversal to other physical or virtual networks.

[7] https://www.virtualbox.org

[8] https://technet.microsoft.com/en-us/library/mt169373(v=ws.11).aspx

[9] https://www.linux-kvm.org/page/Main_Page

[10] https://www.xenproject.org/

Server and Workstation Configuration

The experimental setup was designed to test the feasibility of implementing IPsec SDI as a micro-segmentation proof of concept to restrict east-west traffic within a VLAN. The parameters of the experiment were limited to a virtualized infrastructure and network environment comprising a heterogeneous mix of operating system platforms to reflect a standard corporate IT environment based on AD DS, which stores directory data and manages communication between users and domains, including user logon processes, authentication, and directory searches. The virtual machines defined in Table 3.6 were built based on clean operating system images obtained from the respective vendors:

- 1x Microsoft Windows Server 2012 R2 Standard virtual machine with the AD DS, Dynamic Host Configuration Protocol (DHCP) Server and Domain Name Service (DNS) Server roles installed. This server, which functioned as the DC, DHCP and DNS server for the network environment, was tagged as *mscdc1*, and left in the default OU named *Domain Controllers*.
- 1x Microsoft Windows Server 2016 virtual machine with the File Storage and Internet Information Server roles installed. This server, which functioned as a file and internal web server for the network environment, was tagged as *mscfile1* and placed in the Member Servers OU.
- 1x GNU/Linux Mint 18.2 Sonya virtual machine installed with the following packages:
  - *Systems Security Services Daemon* (SSSD)[11], which allows authentication on directory services, including AD, by providing a cross-domain compatible method for users to sign on with configurable parameters.
  - *realmd*[12], an on-demand system DBus service that permits a standardised method for callers to configure network authentication and domain membership.
  - *strongSwan*[13], an open-source IPsec-based VPN solution that implements IKEv1 and IKEv2 key exchange protocols for Linux and other Unix based operating systems.

[11] https://wiki.ubuntu.com/Enterprise/Authentication/sssd

[12] https://www.freedesktop.org/software/realmd/

[13] https://www.strongswan.org/

This server, which functioned as a Linux based platform joined to the AD domain to represent a heterogeneous operating system mix within the virtualized environment, was tagged as *msclinux* and placed in the Member Servers OU.

- 1x Microsoft Windows 7 virtual machine, which functioned as a typical workstation based endpoint for an end user interacting with the IT environment, was tagged as *mscwin*, and placed in the Workstations OU.

Table 3.6: Experiment 3: IPsec SDI virtual machines

Long hostname	**Short name**	**Operating System**	**IP Address**	**Description**
mscdc1.compsci.local	mscdc1	Windows Server 2012 R2	172.64.16.4	AD, DNS and DHCP
mscfile1.compsci.local	mscfile1	Windows Server 2016	172.64.16.5	File and web server
msclinux.compsci.local	msclinux	GNU/Linux Mint 18.2 Sonya	172.64.16.6	Application server
mscwin7.compsci.local	mscwin7	Windows 7 SP2	172.64.16.7	Workstation

3.4.3 Methodology

A standard AD forest labelled *compsci.local* was defined on the AD DS server *mscdc1*, and all systems within the virtualized network environment were successfully joined to the domain.

Security Policy

A security policy that describes the rules pertaining to devices within the network was defined as follows:

- **Rule 1:** Host *mscdc1* (172.16.64.4) may talk to any device within the VLAN
- **Rule 2:** Host *mscfile1* (172.16.64.5) may talk to any device within the VLAN, except hosts *msclinux* (172.16.64.6) and *mscwin7* (172.16.64.7)
- **Rule 3:** Host *msclinux* (172.16.64.6) may talk to hosts *mscdc1* (172.16.64.4) and *mscwin7* (172.16.64.7) only
- **Rule 4:** Host *mscwin7* (172.16.64.7) may talk to hosts *mscdc1* (172.16.64.4) and *msclinux* (172.16.64.6) only

An IPsec GPO was created containing the translated rules to enforce the security policy through the SDI capability to effect the micro-segmentation functionality.

Domain Users

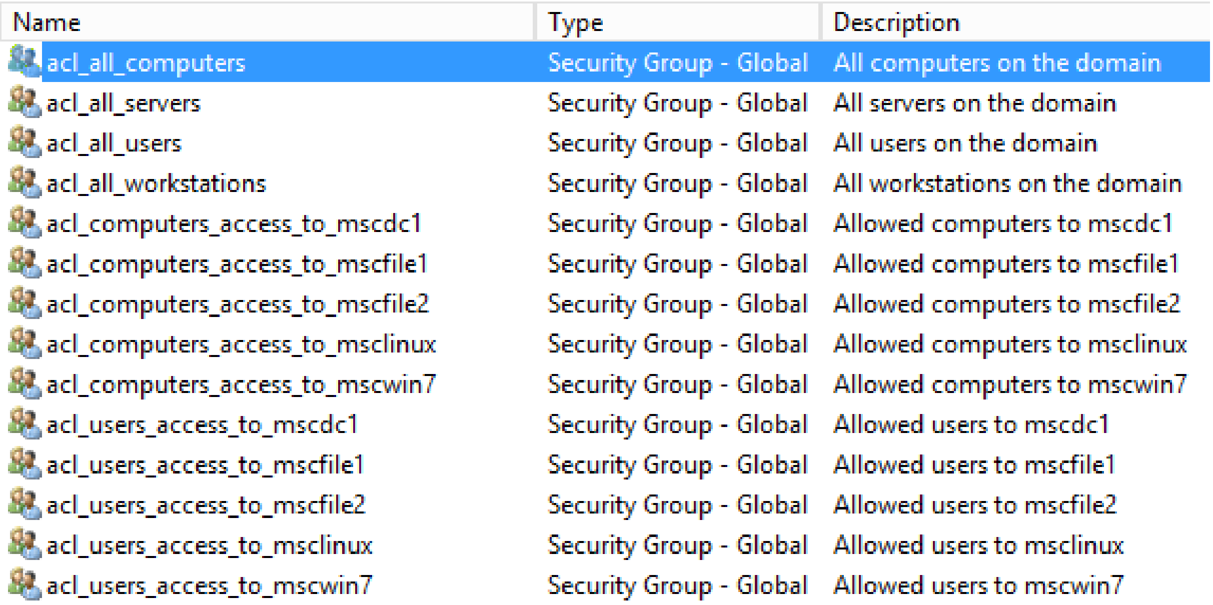
The user accounts defined in Table 3.7 were created on the domain and assigned to security groups as required. For example, user *tu1* may be added as a member of the security group *acl_users_access_to_mscwin7*, which indicates that the user is permitted to login and authenticate onto the workstation *mscwin7*, provided that the user is connecting from a domain workstation or server that is a member of the security group *acl_computers_access_to_mscwin7*.

Table 3.7: Experiment 3: User accounts

Account Name	**Name**	**Organizational Unit**
compsci\tu1	Test User 1	Users
compsci\tu2	Test User 2	Users
compsci\tu96	Test User 96	Users
compsci\mfurst	Test User mfurst	Users
compsci\adm_mfurst	Domain Administrator	Users

Security Groups

The security groups defined in Figure 3.7 were created on the domain, and named in accordance with their intended function. For example, a security group called *acl_computers_access_to_mscdc1* indicates that any member of that security group will be granted access to the destination computer *mscdc1*.



Name	Type	Description
acl_all_computers	Security Group - Global	All computers on the domain
acl_all_servers	Security Group - Global	All servers on the domain
acl_all_users	Security Group - Global	All users on the domain
acl_all_workstations	Security Group - Global	All workstations on the domain
acl_computers_access_to_mscdc1	Security Group - Global	Allowed computers to mscdc1
acl_computers_access_to_mscfile1	Security Group - Global	Allowed computers to mscfile1
acl_computers_access_to_mscfile2	Security Group - Global	Allowed computers to mscfile2
acl_computers_access_to_msclinux	Security Group - Global	Allowed computers to msclinux
acl_computers_access_to_mscwin7	Security Group - Global	Allowed computers to mscwin7
acl_users_access_to_mscdc1	Security Group - Global	Allowed users to mscdc1
acl_users_access_to_mscfile1	Security Group - Global	Allowed users to mscfile1
acl_users_access_to_mscfile2	Security Group - Global	Allowed users to mscfile2
acl_users_access_to_msclinux	Security Group - Global	Allowed users to msclinux
acl_users_access_to_mscwin7	Security Group - Global	Allowed users to mscwin7

Figure 3.7: AD security groups

For IPsec to function correctly in this experiment, all computers in the domain must be configured with the same IPsec connection settings and firewall settings. Due to the complex nature of IPsec and the possibility of inadvertently locking out a computer from the domain, a phased Group Policy implementation approach was adopted in which two base GPOs containing IPsec and firewall policy settings were created and deployed to computers on the domain. Additional configuration settings were then configured and applied in subsequent incremental GPO updates, and once finalised can then be merged into a single GPO.

The GPOs were created and assigned to the respective OUs defined in Table 3.8.

Table 3.8: Experiment 3: Group Policy Objects

Group Policy Object	Organizational Unit	Description
Firewall Settings	All computer OUs	Enable firewall rules
IPsec Settings	All computer OUs	Enable IPsec

Firewall Settings GPO

As IPsec is tightly integrated with Windows Firewall, a firewall settings GPO was configured to enable the firewall on all computers, with a base policy of blocking inbound connections, allowing outbound connections and disabling extraneous firewall and unicast notifications. Additionally, the merging of local firewall and connection security rules was disabled, meaning that GPO-defined firewall rules deployed to computers take precedence and discard any locally defined firewall rules.

In the firewall settings GPO, under *Computer Configuration - Windows Settings - Security Settings - Windows Firewall with Advanced Security - Properties*, the settings defined in Table 3.9 were configured.

Table 3.9: Experiment 3: Firewall settings GPO

Domain Profile	**Setting**
Firewall state	On (recommended)
Inbound connections	Block (default)
Outbound connections	Allow (default)
Firewall Settings - Display a notification	No
Unicast response - Allow unicast response	No
Rule merging - Apply local firewall rules	No
Rule merging - Apply local connection security rules	No

Normally, once all IPsec and firewall settings have been successfully tested, the *Outbound connections* setting should be changed from *Allow (default)* to *Block*, which prevents domain computers from establishing unauthorised connections to any endpoints that are not IPsec enabled or members of a permitted security group. However, the absence of network traffic as a result of this setting would make it impossible to demonstrate the efficacy of IPsec SDI via packet captures, and the default setting is left as is for the duration of the experiment.

Inbound Firewall Rules

Specific firewall rules needed to be defined and applied on all computers allowing them to listen for inbound IPsec authentication attempts, as well as outbound traffic for Kerberos ticket negotiation and DNS resolution queries. Within the same firewall settings GPO, under *Computer Configuration - Windows Settings - Security Settings - Windows Firewall with Advanced Security*, the predefined inbound firewall rules defined in Table 3.10 were added.

Included in the inbound firewall rules are IPsec specific rules defined in Table 3.11 that were configured to control connectivity to the specific destination machines by enforcing IPsec encryption and linking permitted remote users and remote computers to the associated security groups.

Table 3.10: Experiment 3: Inbound firewall rules

Rule Name	**Description**
Core networking	Allow basic networking protocols
ICMPv4	Allow ICMP
File and printer sharing	Allow file sharing
lsass.exe	Allow domain authentication
WMIPrvSE.exe	Allow Windows management instrumentation
NLA Service	Allow network location awareness
svchost.exe	Allow Group Policy client service
DNS TCP	Allow DNS
DNS UDP	Allow DNS
IKE protocol 50	Allow IPsec Encapsulating Security Payload
IKE protocol 51	Allow IPsec Authentication Header
IKE UDP port 4500	Allow Internet Key Exchange
IKE UDP port 500	Allow Internet Security Association & Key Management Protocol
Kerberos TCP port 88	Allow Kerberos authentication
Kerberos UDP port 88	Allow Kerberos authentication

Table 3.11: Experiment 3: Inbound IPsec rules

Rule Name	**Action**	**Local IP**	**Authorized Users**	**Authorized Computers**
Authorized to access mscdc1	Encrypt	172.16.64.4	acl_users_access _to_mscdc1	acl_computers_access _to_mscdc1
Authorized to access mscfile1	Encrypt	172.16.64.5	acl_users_access _to_mscfile1	acl_computers_access _to_ mscfile1
Authorized to access msclinux	Encrypt	172.16.64.6	acl_users_access _to_msclinux	acl_computers_access _to_msclinux
Authorized to access mscfile1	Encrypt	172.16.64.7	acl_users_access _to_mscwin7	acl_computers_access _to_mscwin7

Outbound Firewall Rules

Outbound firewall rules were also required to permit outbound IPsec, Kerberos and DNS traffic. Within the same firewall settings GPO, under *Computer Configuration - Windows Settings - Security Settings - Windows Firewall with Advanced Security*, predefined outbound firewall rules as defined in Table 3.12 were added.

The outbound firewall rules also included the outbound IPsec connectivity rules defined in Table 3.13 to control the IPsec parameters.

The final step was to create a GPO defining the IPsec connection security rules as defined in Table 3.14 to enable and enforce IPsec authentication on all domain member

Table 3.12: Experiment 3: Outbound firewall rules

Rule Name	Description
DNS TCP	Allow domain name system service
DNS UDP	Allow domain name system service
ICMP	Allow ping
IKE protocol 50	Allow IPsec Encapsulating Security Payload
IKE protocol 51	Allow IPsec Authentication Header
IKE UDP port 4500	Allow Internet Key Exchange (IKE)
IKE UDP port 500	Allow ISAKMP
Kerberos TCP port 88	Allow Kerberos authentication
Kerberos UDP port 88	Allow Kerberos authentication

Table 3.13: Experiment 3: Outbound IPsec rules

Rule Name	Action	Remote Address	Authorized Local Principals	Authorized Computers
Authorized to access mscdc1	Encrypt	172.16.64.4	acl_users_access _to_mscdc1	acl_computers_access _to_mscdc1
Authorized to access mscfile1	Encrypt	172.16.64.5	acl_users_access _to_mscfile1	acl_computers_access _to_ mscfile1
Authorized to access msclinux	Encrypt	172.16.64.6	acl_users_access _to_msclinux	acl_computers_access _to_msclinux
Authorized to access mscfile1	Encrypt	172.16.64.7	acl_users_access _to_mscwin7	acl_computers_access _to_mscwin7

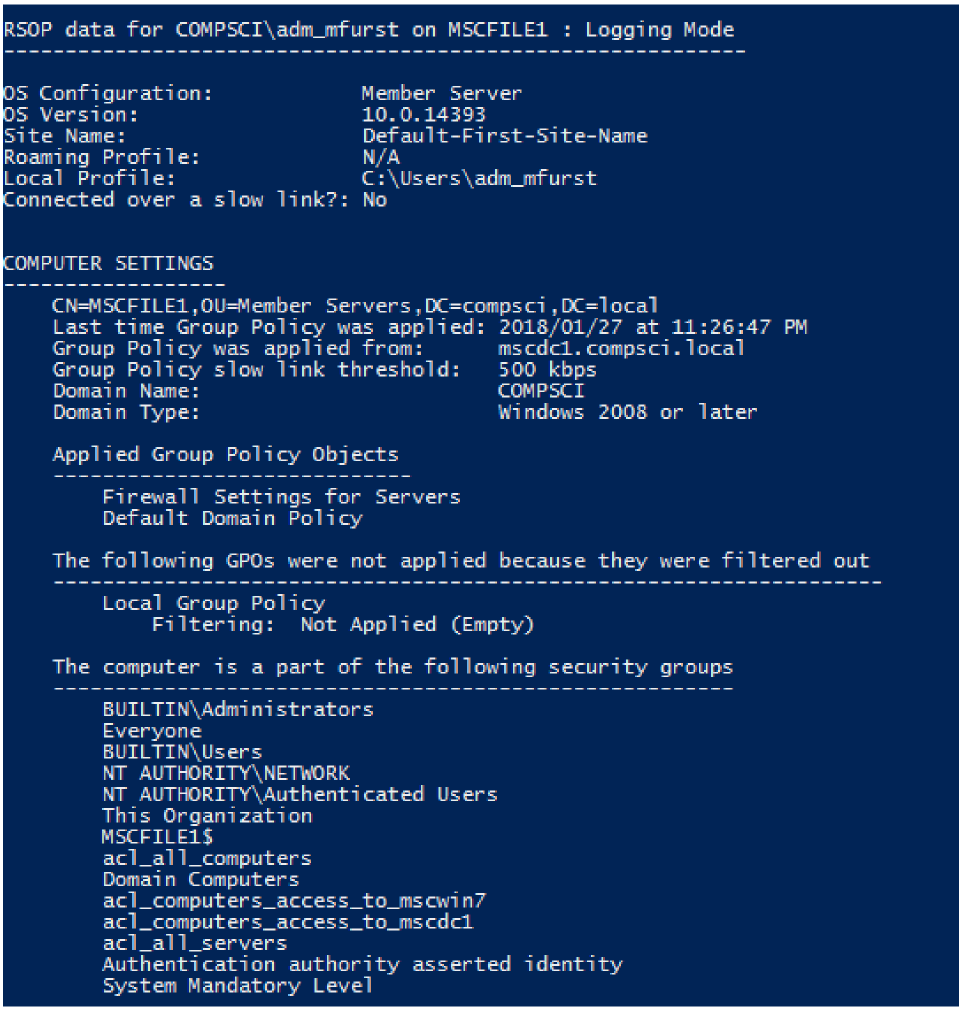
computers, which effectively isolates computers that are not members of the domain from being able to communicate with domain members. Initially, a connection rule requesting IPsec authentication was configured and deployed to all endpoints. Once all computers had responded and authenticated, the rule was changed to require mode. This prevents a situation where require mode is enforced before all computers are able to receive and apply the GPO, and effectively locks them out from being able to authenticate on IPsec.

Table 3.14: Experiment 3: IPsec connection security rules

Setting	Value
Rule Type	Isolation
Requirements	Request authentication for inbound and outbound connections
Authentication Method	Computer and user (Kerberos V5)
Profile	Domain

Enabling IPsec SDI

As the last stage of deployment, the firewall and IPsec GPOs were linked to all computer OUs, and a group policy update was executed to force replication of the new firewall GPOs to the endpoints. Running the command *gpresult /r* on any machine validated that the GPOs were successfully applied as shown in Figure 3.8.



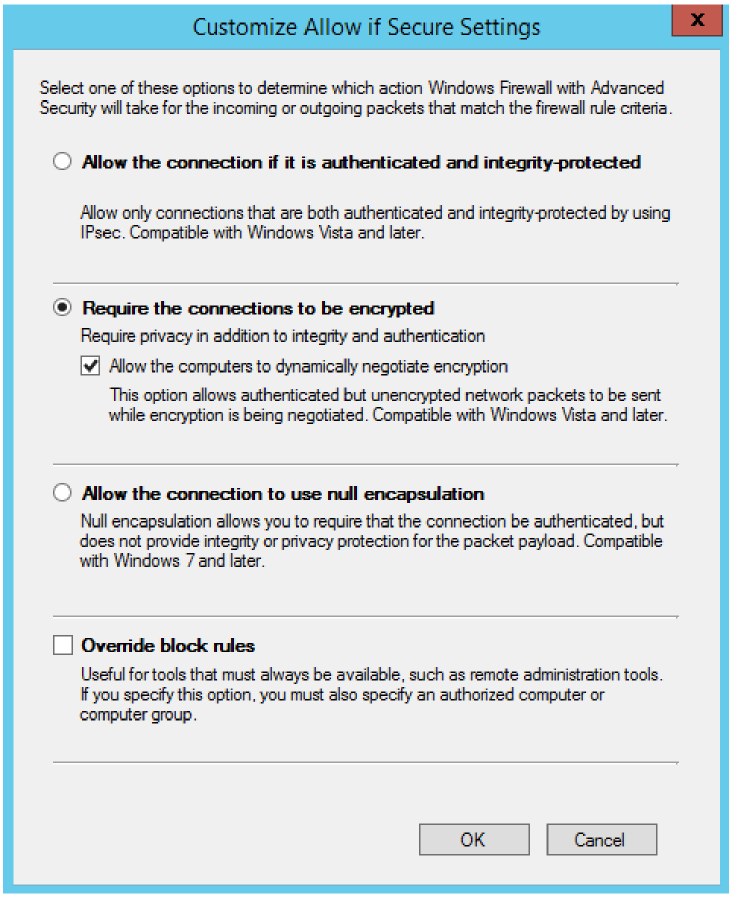
```
RSOP data for COMPSCI\adm_mfurst on MSCFILE1 : Logging Mode
------------------------------------------------------------

OS Configuration:            Member Server
OS Version:                  10.0.14393
Site Name:                   Default-First-Site-Name
Roaming Profile:             N/A
Local Profile:               C:\Users\adm_mfurst
Connected over a slow link?: No


COMPUTER SETTINGS
------------------
    CN=MSCFILE1,OU=Member Servers,DC=compsci,DC=local
    Last time Group Policy was applied: 2018/01/27 at 11:26:47 PM
    Group Policy was applied from:      mscdc1.compsci.local
    Group Policy slow link threshold:   500 kbps
    Domain Name:                        COMPSCI
    Domain Type:                        Windows 2008 or later

    Applied Group Policy Objects
    -----------------------------
        Firewall Settings for Servers
        Default Domain Policy

    The following GPOs were not applied because they were filtered out
    -------------------------------------------------------------------
        Local Group Policy
            Filtering:  Not Applied (Empty)

    The computer is a part of the following security groups
    -------------------------------------------------------
        BUILTIN\Administrators
        Everyone
        BUILTIN\Users
        NT AUTHORITY\NETWORK
        NT AUTHORITY\Authenticated Users
        This Organization
        MSCFILE1$
        acl_all_computers
        Domain Computers
        acl_computers_access_to_mscwin7
        acl_computers_access_to_mscdc1
        acl_all_servers
        Authentication authority asserted identity
        System Mandatory Level
```

Figure 3.8: Group Policy results

After the GPO has been deployed and enforced on all computers, the IPsec GPO was modified to change the isolation attribute from *Require authentication* to *Enforce authentication* for inbound connections, and *Request authentication* for outbound connections.

Combined with the other supporting IPsec firewall rules for each member server and workstation on the domain, these GPO policies form the basis of micro-segmentation based on IPsec SDI.

For example, the IPsec firewall rule *Authorized to access mscfile1* has the parameter *Require the connection to be encrypted* enabled as shown in Figure 3.9, and specifies that only connections to *mscfile1* are permitted from authorised users and computers that are members of the security group *acl_users_access_to_mscfile1* and *acl_computers_access_to_mscfile1*, respectively. This enables both SDI for *mscfile* by authenticating membership of these security groups. In this case, only domain administrators and the AD DC *mscdc1* are permitted to connect and login to the host *mscfile1*.


Customize Allow if Secure Settings

Select one of these options to determine which action Windows Firewall with Advanced Security will take for the incoming or outgoing packets that match the firewall rule criteria.

○ **Allow the connection if it is authenticated and integrity-protected**

Allow only connections that are both authenticated and integrity-protected by using IPsec. Compatible with Windows Vista and later.

● **Require the connections to be encrypted**

Require privacy in addition to integrity and authentication

☑ Allow the computers to dynamically negotiate encryption

This option allows authenticated but unencrypted network packets to be sent while encryption is being negotiated. Compatible with Windows Vista and later.

○ **Allow the connection to use null encapsulation**

Null encapsulation allows you to require that the connection be authenticated, but does not provide integrity or privacy protection for the packet payload. Compatible with Windows 7 and later.

□ **Override block rules**

Useful for tools that must always be available, such as remote administration tools. If you specify this option, you must also specify an authorized computer or computer group.

Figure 3.9: Encapsulating security payload settings

Full IPsec Implementation

When deploying IPsec SDI, current best practice guidelines from Microsoft (Clark *et al.*, 2006) specify that the AD server must be excluded so that it can continue to service non-IPsec enabled endpoints. For a trusted network environment, this scenario may not be compatible with the guidelines, and full end-to-end IPsec deployment may instead be implemented that includes AD in the SDI setup.

If an IPsec deployment is configured that includes AD as part of the SDI group, a specific sequence has to be followed whenever the IPsec or firewall GPO settings need to be modified, or a new computer needs to be joined to the domain. The sequence steps are listed below.

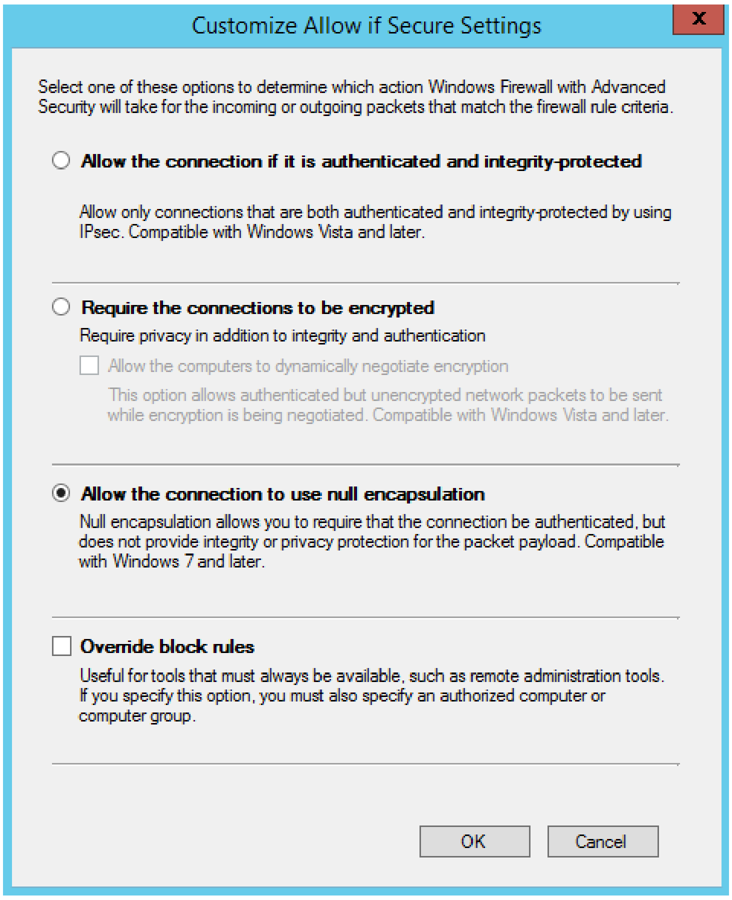
- The IPsec GPO must be delinked from the AD DC.
- The IPsec GPO can then be modified, or the new computer joined to the domain.
- If the modified GPO needs to be deployed to endpoints, three running Windows services *IKE and AuthIP IPsec Keying Modules*, *IPsec Policy Agent* and *Windows Firewall* must be stopped.
- A Group Policy update must be forced on all endpoints (*gpupdate /force*).
- The three services previously stopped must be restarted on all endpoints.
- Once the GPO has been successfully imported onto the endpoints, the IPsec GPO must be re-linked to the AD DC, and a Group Policy Update must be forced on the AD server.

3.4.4 Evaluation

As described in Section 3.2.4, a netcat listener and telnet client was used to simulate TCP traffic between endpoints.

Packet captures of network traffic between the various endpoints are performed to facilitate analysis of the IP traffic flows between the endpoints, both before and after IPsec SDI was applied to simulate micro-segmentation. Due to the IPsec GPO enabling authentication and encryption of traffic using AH and ESP modes, it is normally not possible to parse the traffic flows from the packet capture and extract clear-text information from the encrypted payload to demonstrate the effectiveness of IPsec based micro-segmentation.

As a subjective assertion on the effectiveness of IPsec expressed through a practical manifestation of the security policy being enforced at the endpoint is not sufficient, it was necessary to implement a temporary workaround for this issue by amending the IPsec GPO to change the ESP setting from *Require the connection to be encrypted* to *Allow the connection to use null encapsulation*, as shown in Figure 3.10. This setting effectively encapsulates packets without encrypting them, which allows for a packet capture of IPsec traffic flows between the endpoints to be decoded by a protocol dissector in a packet analysis tool, such as Wireshark as shown in Figure 3.11.



Customize Allow if Secure Settings

Select one of these options to determine which action Windows Firewall with Advanced Security will take for the incoming or outgoing packets that match the firewall rule criteria.

○ **Allow the connection if it is authenticated and integrity-protected**

Allow only connections that are both authenticated and integrity-protected by using IPsec. Compatible with Windows Vista and later.

○ **Require the connections to be encrypted**

Require privacy in addition to integrity and authentication

□ Allow the computers to dynamically negotiate encryption

This option allows authenticated but unencrypted network packets to be sent while encryption is being negotiated. Compatible with Windows Vista and later.

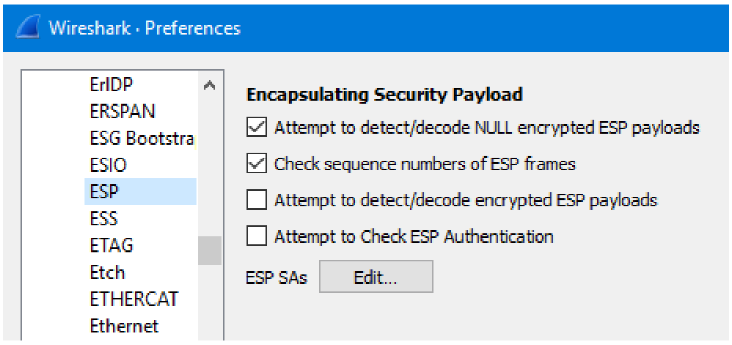
● **Allow the connection to use null encapsulation**

Null encapsulation allows you to require that the connection be authenticated, but does not provide integrity or privacy protection for the packet payload. Compatible with Windows 7 and later.

□ **Override block rules**

Useful for tools that must always be available, such as remote administration tools. If you specify this option, you must also specify an authorized computer or computer group.

Figure 3.10: Null encapsulation



Encapsulating Security Payload

☑ Attempt to detect/decode NULL encrypted ESP payloads

☑ Check sequence numbers of ESP frames

□ Attempt to detect/decode encrypted ESP payloads

□ Attempt to Check ESP Authentication

Figure 3.11: Decoding null ESP with Wireshark

The results of the IPsec SDI packet captures are analysed in more detail in Chapter 4 (see Section 4.4).

3.5 Summary

In this chapter, the particulars of each experimental setup were discussed, with the lab environment, security policy, hardware (whether physical or virtualized) and the micro-segmentation approach utilised covered in detail.

For each micro-segmentation approach, the selection, implementation and configuration of network switch ACLs, VMware NSX and IPsec SDI functionality were discussed and documented in sufficient detail to enable replication and validation of the experimental setup.

Chapter 4

Analysis of Results

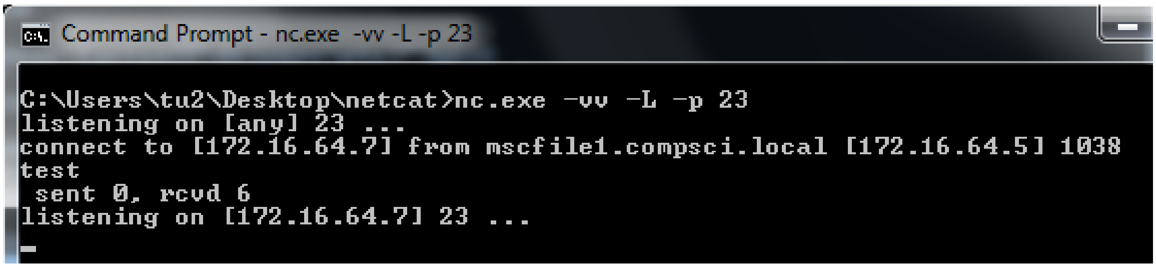
In this chapter, the output generated from the experiments documented in Chapter 3 is assessed to test the validity of the experimental setup, and offer an objective and meaningful interpretation of the data. In the next few sections, a recap of the test strategy is discussed, followed by an in-depth review of the packet captures obtained from each of the three experiments.

4.1 Test Strategy

As micro-segmentation focuses on the granular segregation of endpoints within a network segment, it stands to reason that the most efficient method to test the effectiveness of a particular micro-segmentation approach is by validating whether traffic flow between targeted endpoints is blocked.

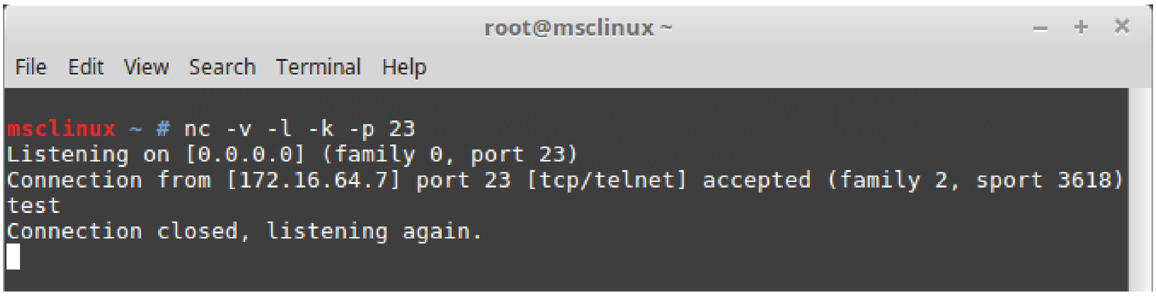
A reliable way of verifying the cessation of traffic flow is to perform a packet capture on the relevant host's active network interface. In the context of the switch ACLs and IPsec, the open-source Wireshark packet capture software was installed on all endpoints and simultaneously launched to capture traffic flow, both before and after enabling micro-segmentation. By inspecting the output of layer 3 traffic, it is straightforward to confirm whether data packets reached their intended destination or not, and whether such activity is aligned with the organisational security policy and corresponding rules. Multiple failed data packet delivery results indicate that the particular micro-segmentation strategy is indeed effective.

As mentioned in Section 3.2.4, a netcat utility that reads and writes data across a TCP or UDP network connection was configured on all endpoints to simulate a telnet server listening for incoming TCP connections on port 23. Figures 4.1 and 4.2 show typical netcat instances that respawned to listen for new inbound connections after a previous session was terminated.



```
C:\Users\tu2\Desktop\netcat>nc.exe -vv -L -p 23
listening on [any] 23 ...
connect to [172.16.64.7] from mscfile1.compsci.local [172.16.64.5] 1038
test
 sent 0, rcvd 6
listening on [172.16.64.7] 23 ...
```

Figure 4.1: Windows netcat tool



```
msclinux ~ # nc -v -l -k -p 23
Listening on [0.0.0.0] (family 0, port 23)
Connection from [172.16.64.7] port 23 [tcp/telnet] accepted (family 2, sport 3618)
test
Connection closed, listening again.
```

Figure 4.2: Linux netcat tool

Successful TCP handshakes are shown by the presence of data activity in the terminal window, while failed TCP handshakes show up as retransmission timeouts in the packet capture, thereby demonstrating the correct application of micro-segmentation in enforcing the security policy objectives.

It should be noted that for the purposes of this experiment, successfully blocked layer 3 traffic is considered sufficient, predicated on the assumption that adequate security controls have been implemented to mitigate against IPsec and VLAN attacks described in Section 2.2.3. Segmentation of layer 2 traffic as well as mitigating controls against layer 2 attack techniques is beyond the scope of this experiment.

4.2 Experiment 1: Switch ACLs

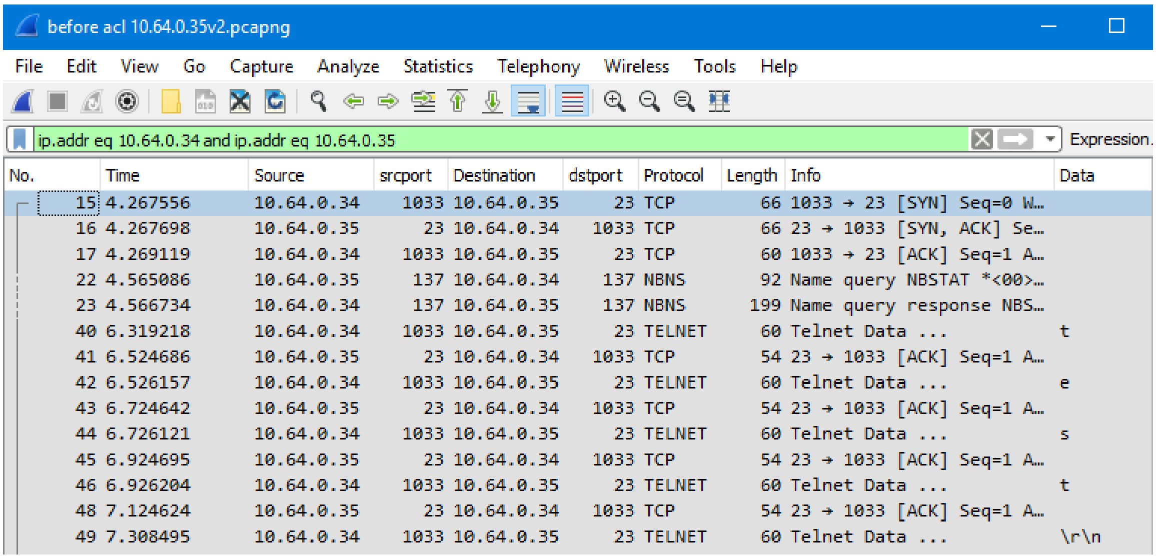
As referred to in Section 3.2, the intended outcome of this experiment was to determine the effectiveness of ACLs to control east-west traffic within a single network segment. To recap, the simple security policy defined in Section 3.2.3 is summarised as follows:

- **Host A** (10.64.0.34) may not connect to *Host B*
- **Host B** (10.64.0.35) may not connect to *Host A*
- **Hosts A** and **B** may connect to **Host C** (10.64.0.41)

4.2.1 Traffic Flows before ACLs

The following packet captures show the layer 3 traffic flows between the source and destination endpoints within the same VLAN before and after application of the ACL rules on the switch.

The first screenshot depicted in Figure 4.3 shows a simple telnet connection from Host *B* (10.64.0.35) to Host *A* (10.64.0.34), verifying that the two workstations are able to communicate with each other before the ACLs are applied. This can be seen by the 3-way TCP handshake SYN, SYN/ACK, ACK and the data flows following immediately afterwards.



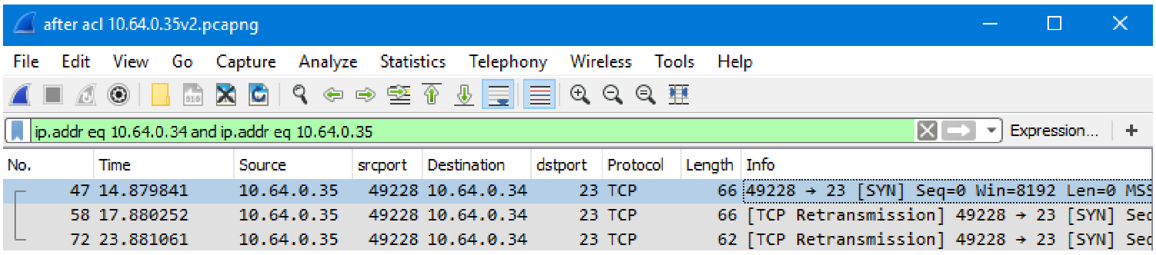
ip.addr eq 10.64.0.34 and ip.addr eq 10.64.0.35

No.	Time	Source	srcport	Destination	dstport	Protocol	Length	Info	Data
15	4.267556	10.64.0.34	1033	10.64.0.35	23	TCP	66	1033 → 23 [SYN] Seq=0 W…	
16	4.267698	10.64.0.35	23	10.64.0.34	1033	TCP	66	23 → 1033 [SYN, ACK] Se…	
17	4.269119	10.64.0.34	1033	10.64.0.35	23	TCP	60	1033 → 23 [ACK] Seq=1 A…	
22	4.565086	10.64.0.35	137	10.64.0.34	137	NBNS	92	Name query NBSTAT *<00>…	
23	4.566734	10.64.0.34	137	10.64.0.35	137	NBNS	199	Name query response NBS…	
40	6.319218	10.64.0.34	1033	10.64.0.35	23	TELNET	60	Telnet Data ...	t
41	6.524686	10.64.0.35	23	10.64.0.34	1033	TCP	54	23 → 1033 [ACK] Seq=1 A…	
42	6.526157	10.64.0.34	1033	10.64.0.35	23	TELNET	60	Telnet Data ...	e
43	6.724642	10.64.0.35	23	10.64.0.34	1033	TCP	54	23 → 1033 [ACK] Seq=1 A…	
44	6.726121	10.64.0.34	1033	10.64.0.35	23	TELNET	60	Telnet Data ...	s
45	6.924695	10.64.0.35	23	10.64.0.34	1033	TCP	54	23 → 1033 [ACK] Seq=1 A…	
46	6.926204	10.64.0.34	1033	10.64.0.35	23	TELNET	60	Telnet Data ...	t
48	7.124624	10.64.0.35	23	10.64.0.34	1033	TCP	54	23 → 1033 [ACK] Seq=1 A…	
49	7.308495	10.64.0.34	1033	10.64.0.35	23	TELNET	60	Telnet Data ...	\r\n

Figure 4.3: Experiment 1: Connection from 10.64.0.35 to 10.64.0.34 permitted

4.2.2 Traffic Flows after ACLs

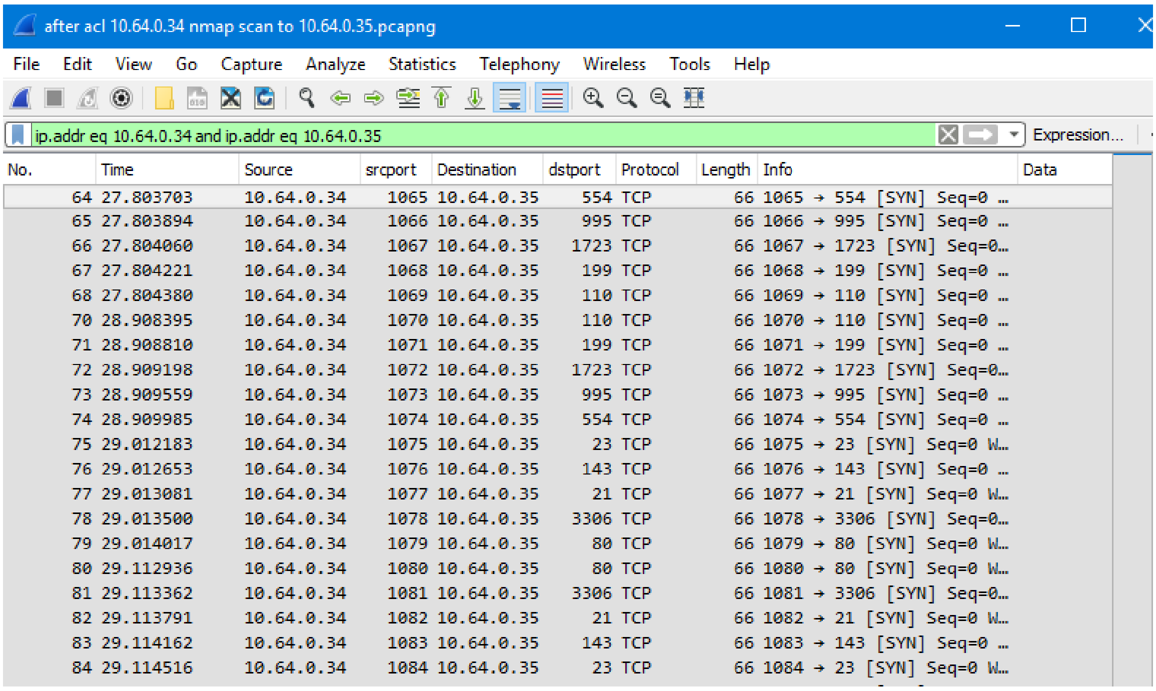
After applying the ACLs on the switch, a second telnet attempt was made, the results of which can be seen in Figure 4.4. Host *B* (10.64.0.35) can be seen retrying the TCP handshake request due to a lack of response from Host *A* (10.64.0.34), i.e. there was no ACK response.


after acl 10.64.0.35v2.pcapng
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
Expression… +

```
ip.addr eq 10.64.0.34 and ip.addr eq 10.64.0.35

No.  Time       Source      srcport Destination dstport Protocol Length Info
  47 14.879841  10.64.0.35  49228   10.64.0.34  23      TCP      66     49228 → 23 [SYN] Seq=0 Win=8192 Len=0 MSS
  58 17.880252  10.64.0.35  49228   10.64.0.34  23      TCP      66     [TCP Retransmission] 49228 → 23 [SYN] Sec
  72 23.881061  10.64.0.35  49228   10.64.0.34  23      TCP      62     [TCP Retransmission] 49228 → 23 [SYN] Sec
```

Figure 4.4: Experiment 1: Connection from 10.64.0.34 to 10.64.0.35 denied

Nmap is a free tool for performing network discovery and security auditing. To verify that the ACL was effective at blocking the entire layer 3 traffic between both source and destination hosts, an nmap port scan was executed from Host *B* (10.64.0.35) against Host *A* (10.64.0.34). As can be seen in Figure 4.5, no layer 3 traffic from Host *B* (10.64.0.35) reached Host *A* (10.64.0.34), as the switch ACLs were effective in blocking the traffic flow. This can be seen by the numerous SYN requests with no corresponding ACK responses.

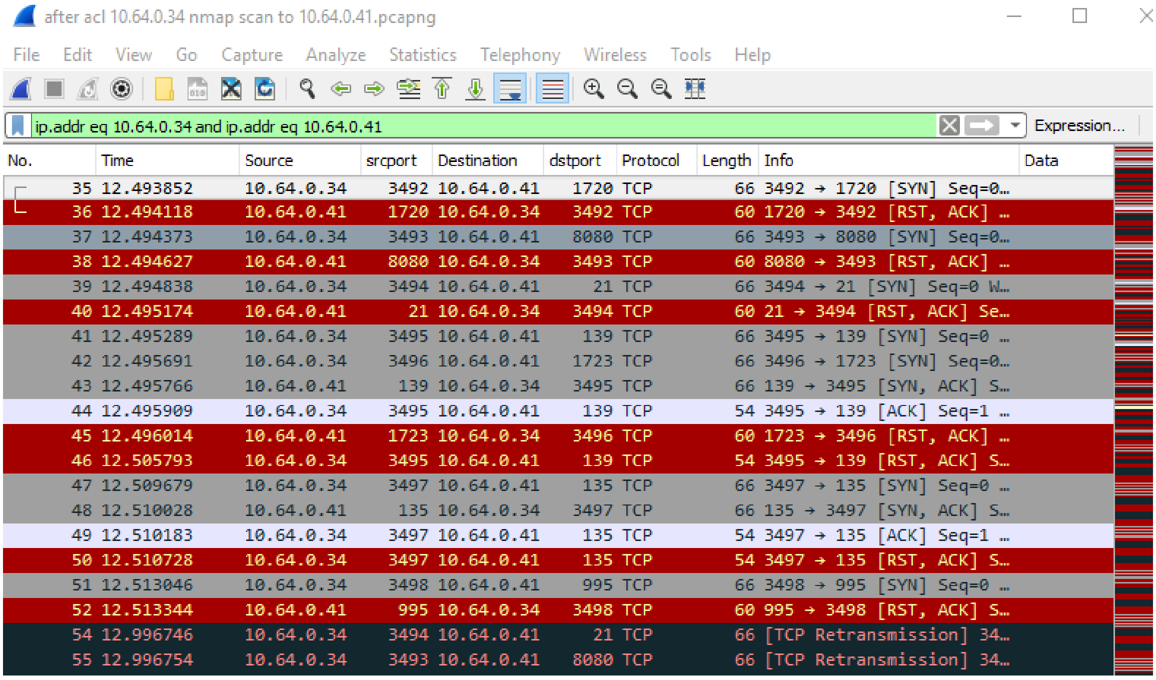

after acl 10.64.0.34 nmap scan to 10.64.0.35.pcapng
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
Expression…

```
ip.addr eq 10.64.0.34 and ip.addr eq 10.64.0.35

No.  Time       Source      srcport Destination dstport Protocol Length Info                         Data
  64 27.803703  10.64.0.34  1065    10.64.0.35   554    TCP      66     1065 → 554 [SYN] Seq=0 …
  65 27.803894  10.64.0.34  1066    10.64.0.35   995    TCP      66     1066 → 995 [SYN] Seq=0 …
  66 27.804060  10.64.0.34  1067    10.64.0.35  1723    TCP      66     1067 → 1723 [SYN] Seq=0…
  67 27.804221  10.64.0.34  1068    10.64.0.35   199    TCP      66     1068 → 199 [SYN] Seq=0 …
  68 27.804380  10.64.0.34  1069    10.64.0.35   110    TCP      66     1069 → 110 [SYN] Seq=0 …
  70 28.908395  10.64.0.34  1070    10.64.0.35   110    TCP      66     1070 → 110 [SYN] Seq=0 …
  71 28.908810  10.64.0.34  1071    10.64.0.35   199    TCP      66     1071 → 199 [SYN] Seq=0 …
  72 28.909198  10.64.0.34  1072    10.64.0.35  1723    TCP      66     1072 → 1723 [SYN] Seq=0…
  73 28.909559  10.64.0.34  1073    10.64.0.35   995    TCP      66     1073 → 995 [SYN] Seq=0 …
  74 28.909985  10.64.0.34  1074    10.64.0.35   554    TCP      66     1074 → 554 [SYN] Seq=0 …
  75 29.012183  10.64.0.34  1075    10.64.0.35    23    TCP      66     1075 → 23 [SYN] Seq=0 W…
  76 29.012653  10.64.0.34  1076    10.64.0.35   143    TCP      66     1076 → 143 [SYN] Seq=0 …
  77 29.013081  10.64.0.34  1077    10.64.0.35    21    TCP      66     1077 → 21 [SYN] Seq=0 W…
  78 29.013500  10.64.0.34  1078    10.64.0.35  3306    TCP      66     1078 → 3306 [SYN] Seq=0…
  79 29.014017  10.64.0.34  1079    10.64.0.35    80    TCP      66     1079 → 80 [SYN] Seq=0 W…
  80 29.112936  10.64.0.34  1080    10.64.0.35    80    TCP      66     1080 → 80 [SYN] Seq=0 W…
  81 29.113362  10.64.0.34  1081    10.64.0.35  3306    TCP      66     1081 → 3306 [SYN] Seq=0…
  82 29.113791  10.64.0.34  1082    10.64.0.35    21    TCP      66     1082 → 21 [SYN] Seq=0 W…
  83 29.114162  10.64.0.34  1083    10.64.0.35   143    TCP      66     1083 → 143 [SYN] Seq=0 …
  84 29.114516  10.64.0.34  1084    10.64.0.35    23    TCP      66     1084 → 23 [SYN] Seq=0 W…
```

Figure 4.5: Experiment 1: Port scan from 10.64.0.34 to 10.64.0.35 denied

As confirmation that traffic only between Host *A* (10.64.0.34) and Host *B* (10.64.0.35) was effectively blocked by the ACL rules, and not other hosts, another nmap port scan was executed against Host *C* (10.64.0.41). It is clear from Figure 4.6 that Host *C*

(10.64.0.41) received multiple TCP handshake requests, responding with either ACK if the requested port was open, or RST if the requested port was closed.



ip.addr eq 10.64.0.34 and ip.addr eq 10.64.0.41

No.	Time	Source	srcport	Destination	dstport	Protocol	Length	Info	Data
35	12.493852	10.64.0.34	3492	10.64.0.41	1720	TCP	66	3492 → 1720 [SYN] Seq=0…	
36	12.494118	10.64.0.41	1720	10.64.0.34	3492	TCP	60	1720 → 3492 [RST, ACK] …	
37	12.494373	10.64.0.34	3493	10.64.0.41	8080	TCP	66	3493 → 8080 [SYN] Seq=0…	
38	12.494627	10.64.0.41	8080	10.64.0.34	3493	TCP	60	8080 → 3493 [RST, ACK] …	
39	12.494838	10.64.0.34	3494	10.64.0.41	21	TCP	66	3494 → 21 [SYN] Seq=0 W…	
40	12.495174	10.64.0.41	21	10.64.0.34	3494	TCP	60	21 → 3494 [RST, ACK] Se…	
41	12.495289	10.64.0.34	3495	10.64.0.41	139	TCP	66	3495 → 139 [SYN] Seq=0 …	
42	12.495691	10.64.0.34	3496	10.64.0.41	1723	TCP	66	3496 → 1723 [SYN] Seq=0…	
43	12.495766	10.64.0.41	139	10.64.0.34	3495	TCP	66	139 → 3495 [SYN, ACK] S…	
44	12.495909	10.64.0.34	3495	10.64.0.41	139	TCP	54	3495 → 139 [ACK] Seq=1 …	
45	12.496014	10.64.0.41	1723	10.64.0.34	3496	TCP	60	1723 → 3496 [RST, ACK] …	
46	12.505793	10.64.0.34	3495	10.64.0.41	139	TCP	54	3495 → 139 [RST, ACK] S…	
47	12.509679	10.64.0.34	3497	10.64.0.41	135	TCP	66	3497 → 135 [SYN] Seq=0 …	
48	12.510028	10.64.0.41	135	10.64.0.34	3497	TCP	66	135 → 3497 [SYN, ACK] S…	
49	12.510183	10.64.0.34	3497	10.64.0.41	135	TCP	54	3497 → 135 [ACK] Seq=1 …	
50	12.510728	10.64.0.34	3497	10.64.0.41	135	TCP	54	3497 → 135 [RST, ACK] S…	
51	12.513046	10.64.0.34	3498	10.64.0.41	995	TCP	66	3498 → 995 [SYN] Seq=0 …	
52	12.513344	10.64.0.41	995	10.64.0.34	3498	TCP	60	995 → 3498 [RST, ACK] S…	
54	12.996746	10.64.0.34	3494	10.64.0.41	21	TCP	66	[TCP Retransmission] 34…	
55	12.996754	10.64.0.34	3493	10.64.0.41	8080	TCP	66	[TCP Retransmission] 34…	

Figure 4.6: Experiment 1: Port scan from 10.64.0.34 to 10.64.0.41 permitted

The above-mentioned sample network packet captures demonstrate that configuring and applying discrete ACL rules on physical network switches are effective in controlling east-west traffic flows in the absence of host-based firewalls or other equivalent security controls.

4.3 Experiment 2: VMware NSX

This section details the analysis of the packet capture results obtained from the VMware NSX experiment in Section 3.3, which was intended to evaluate the effectiveness of the distributed firewall capability in the NSX platform to enforce micro-segmentation.

Although the standard Wireshark software was installed on the virtual machines, the VMware Log Insight appliance also allow for detailed analytics of traffic flows. Graphs were generated to illustrate the distributed firewall capabilities in enforcing the security policy defined in Section 3.3.3 and are summarised below:

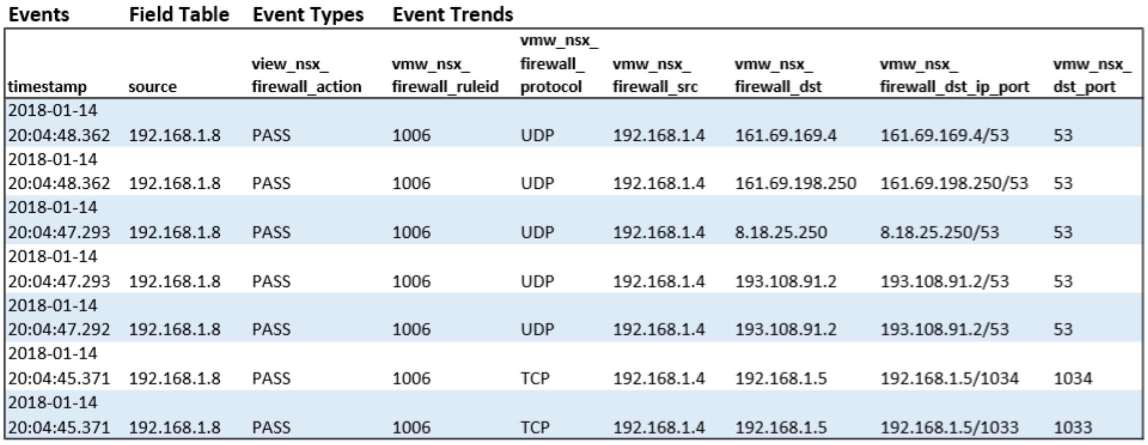
- **mscdc1** (192.168.1.4) may connect to any device

- **mscfile1** (192.168.1.5) may not connect to *msclinux* or *mscwin7*
- **msclinux** (192.168.1.7) may not connect to *mscdc1* or *mscfile1*
- **mscwin7** (192.168.1.15) may not connect to *mscfile1*

For comparative purposes, both Wireshark and Log Insight screenshots are presented for this experiment.

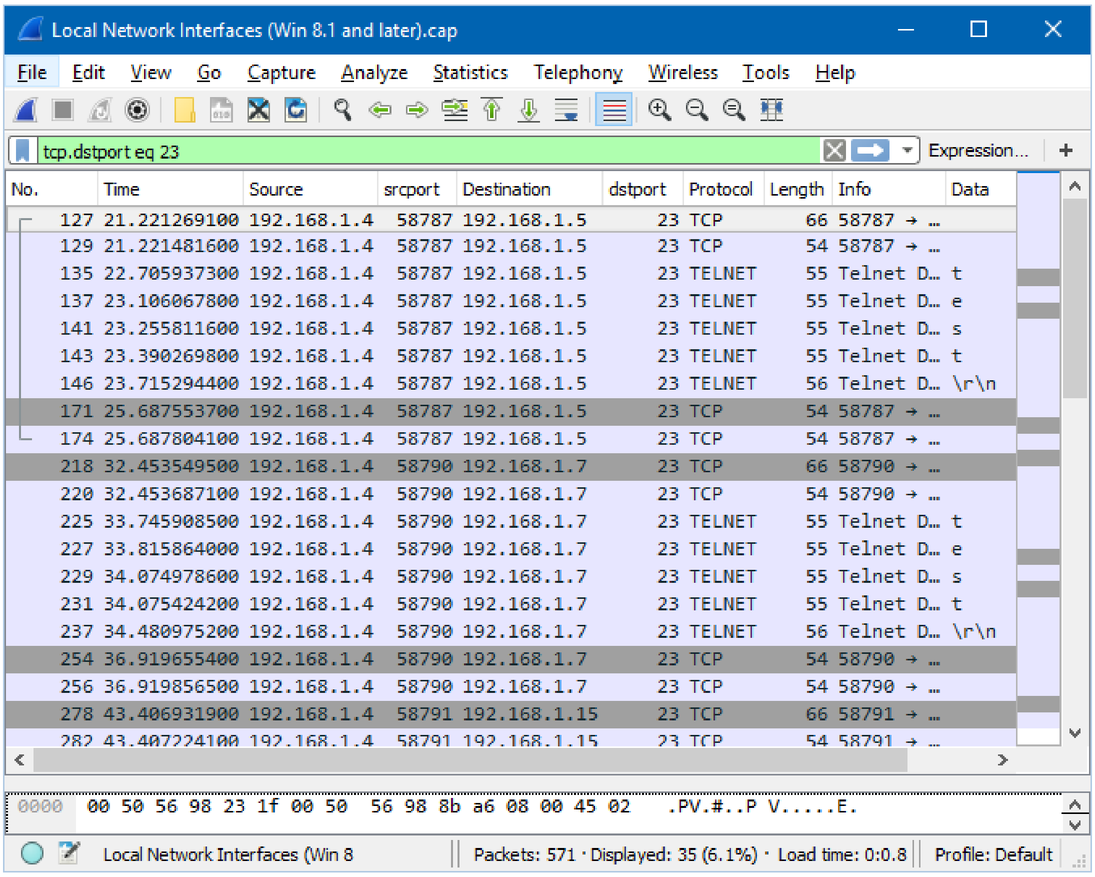
4.3.1 mscdc1.compsci.local (192.168.1.4)

As validation that there was no blanket deny rule configured on the NSX distributed firewall platform, Figure 4.7 shows an extract from the VMware Log Insight interactive dashboard, confirming that *mscdc1* (192.168.1.4) has full connectivity to all endpoints, as per the defined security policy. Figure 4.8 shows the Wireshark packet capture equivalent.



timestamp	source	view_nsx_ firewall_action	vmw_nsx_ firewall_ruleid	vmw_nsx_ firewall_ protocol	vmw_nsx_ firewall_src	vmw_nsx_ firewall_dst	vmw_nsx_ firewall_dst_ip_port	vmw_nsx_ dst_port
2018-01-14 20:04:48.362	192.168.1.8	PASS	1006	UDP	192.168.1.4	161.69.169.4	161.69.169.4/53	53
2018-01-14 20:04:48.362	192.168.1.8	PASS	1006	UDP	192.168.1.4	161.69.198.250	161.69.198.250/53	53
2018-01-14 20:04:47.293	192.168.1.8	PASS	1006	UDP	192.168.1.4	8.18.25.250	8.18.25.250/53	53
2018-01-14 20:04:47.293	192.168.1.8	PASS	1006	UDP	192.168.1.4	193.108.91.2	193.108.91.2/53	53
2018-01-14 20:04:47.292	192.168.1.8	PASS	1006	UDP	192.168.1.4	193.108.91.2	193.108.91.2/53	53
2018-01-14 20:04:45.371	192.168.1.8	PASS	1006	TCP	192.168.1.4	192.168.1.5	192.168.1.5/1034	1034
2018-01-14 20:04:45.371	192.168.1.8	PASS	1006	TCP	192.168.1.4	192.168.1.5	192.168.1.5/1033	1033

Figure 4.7: Experiment 2: Connection from 192.168.1.4 to all endpoints permitted (Log Insight)


Local Network Interfaces (Win 8.1 and later).cap
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
Expression…
+

tcp.dstport eq 23

No.	Time	Source	srcport	Destination	dstport	Protocol	Length	Info	Data
127	21.221269100	192.168.1.4	58787	192.168.1.5	23	TCP	66	58787 → …	
129	21.221481600	192.168.1.4	58787	192.168.1.5	23	TCP	54	58787 → …	
135	22.705937300	192.168.1.4	58787	192.168.1.5	23	TELNET	55	Telnet D…	t
137	23.106067800	192.168.1.4	58787	192.168.1.5	23	TELNET	55	Telnet D…	e
141	23.255811600	192.168.1.4	58787	192.168.1.5	23	TELNET	55	Telnet D…	s
143	23.390269800	192.168.1.4	58787	192.168.1.5	23	TELNET	55	Telnet D…	t
146	23.715294400	192.168.1.4	58787	192.168.1.5	23	TELNET	56	Telnet D…	\r\n
171	25.687553700	192.168.1.4	58787	192.168.1.5	23	TCP	54	58787 → …	
174	25.687804100	192.168.1.4	58787	192.168.1.5	23	TCP	54	58787 → …	
218	32.453549500	192.168.1.4	58790	192.168.1.7	23	TCP	66	58790 → …	
220	32.453687100	192.168.1.4	58790	192.168.1.7	23	TCP	54	58790 → …	
225	33.745908500	192.168.1.4	58790	192.168.1.7	23	TELNET	55	Telnet D…	t
227	33.815864000	192.168.1.4	58790	192.168.1.7	23	TELNET	55	Telnet D…	e
229	34.074978600	192.168.1.4	58790	192.168.1.7	23	TELNET	55	Telnet D…	s
231	34.075424200	192.168.1.4	58790	192.168.1.7	23	TELNET	55	Telnet D…	t
237	34.480975200	192.168.1.4	58790	192.168.1.7	23	TELNET	56	Telnet D…	\r\n
254	36.919655400	192.168.1.4	58790	192.168.1.7	23	TCP	54	58790 → …	
256	36.919856500	192.168.1.4	58790	192.168.1.7	23	TCP	54	58790 → …	
278	43.406931900	192.168.1.4	58791	192.168.1.15	23	TCP	66	58791 → …	
282	43.407224100	192.168.1.4	58791	192.168.1.15	23	TCP	54	58791 → …	

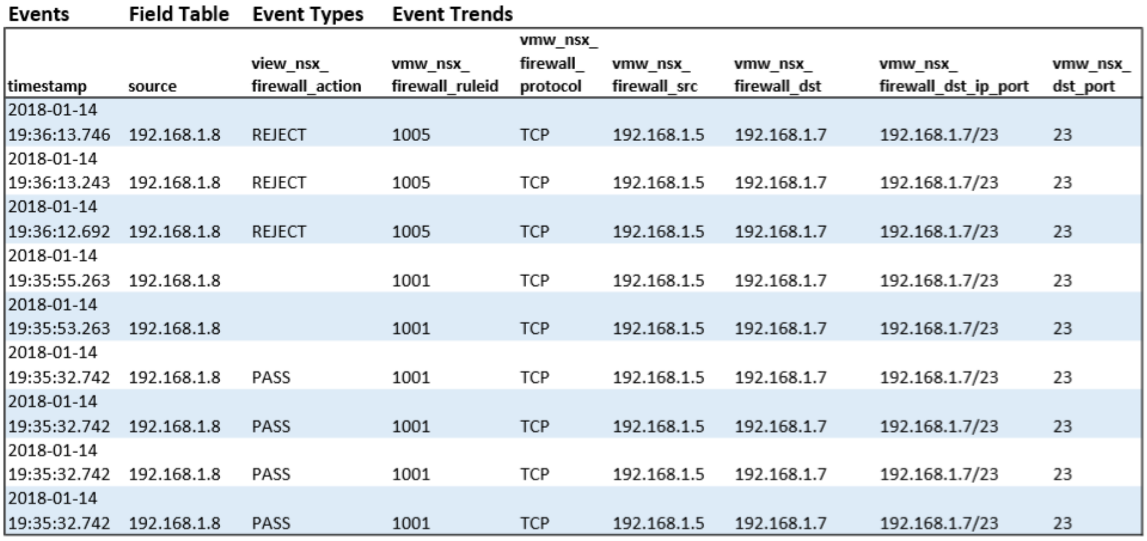
```
0000  00 50 56 98 23 1f 00 50  56 98 8b a6 08 00 45 02   .PV.#..P V.....E.
```

Local Network Interfaces (Win 8 | Packets: 571 · Displayed: 35 (6.1%) · Load time: 0:0.8 | Profile: Default

Figure 4.8: Experiment 2: Connection from 192.168.1.4 to all endpoints permitted (Wireshark)

4.3.2 mscfile1.compsci.local (192.168.1.5)

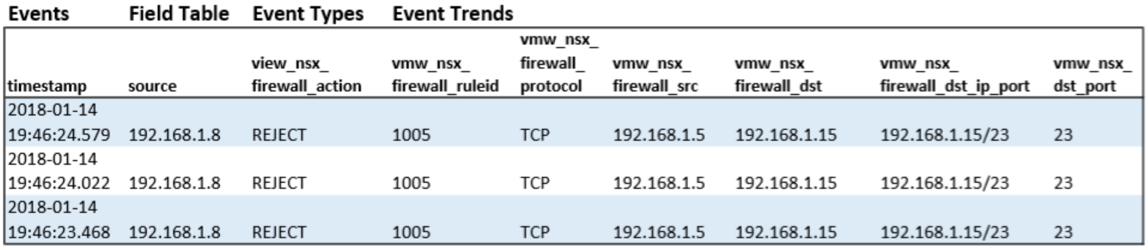
The screenshot in Figure 4.9 shows the initial successful telnet connection from *mscfile1* (192.168.1.5) to *msclinux* (192.168.1.7). When the distributed firewall rule in NSX was enabled to implement micro-segmentation between the two endpoints, the results were immediately apparent, as can be seen by the subsequent telnet connections being rejected. Figure 4.11 shows the Wireshark packet capture equivalent.


Events
Field Table
Event Types
Event Trends

timestamp	source	view_nsx_firewall_action	vmw_nsx_firewall_ruleid	vmw_nsx_firewall_protocol	vmw_nsx_firewall_src	vmw_nsx_firewall_dst	vmw_nsx_firewall_dst_ip_port	vmw_nsx_dst_port
2018-01-14 19:36:13.746	192.168.1.8	REJECT	1005	TCP	192.168.1.5	192.168.1.7	192.168.1.7/23	23
2018-01-14 19:36:13.243	192.168.1.8	REJECT	1005	TCP	192.168.1.5	192.168.1.7	192.168.1.7/23	23
2018-01-14 19:36:12.692	192.168.1.8	REJECT	1005	TCP	192.168.1.5	192.168.1.7	192.168.1.7/23	23
2018-01-14 19:35:55.263	192.168.1.8		1001	TCP	192.168.1.5	192.168.1.7	192.168.1.7/23	23
2018-01-14 19:35:53.263	192.168.1.8		1001	TCP	192.168.1.5	192.168.1.7	192.168.1.7/23	23
2018-01-14 19:35:32.742	192.168.1.8	PASS	1001	TCP	192.168.1.5	192.168.1.7	192.168.1.7/23	23
2018-01-14 19:35:32.742	192.168.1.8	PASS	1001	TCP	192.168.1.5	192.168.1.7	192.168.1.7/23	23
2018-01-14 19:35:32.742	192.168.1.8	PASS	1001	TCP	192.168.1.5	192.168.1.7	192.168.1.7/23	23
2018-01-14 19:35:32.742	192.168.1.8	PASS	1001	TCP	192.168.1.5	192.168.1.7	192.168.1.7/23	23

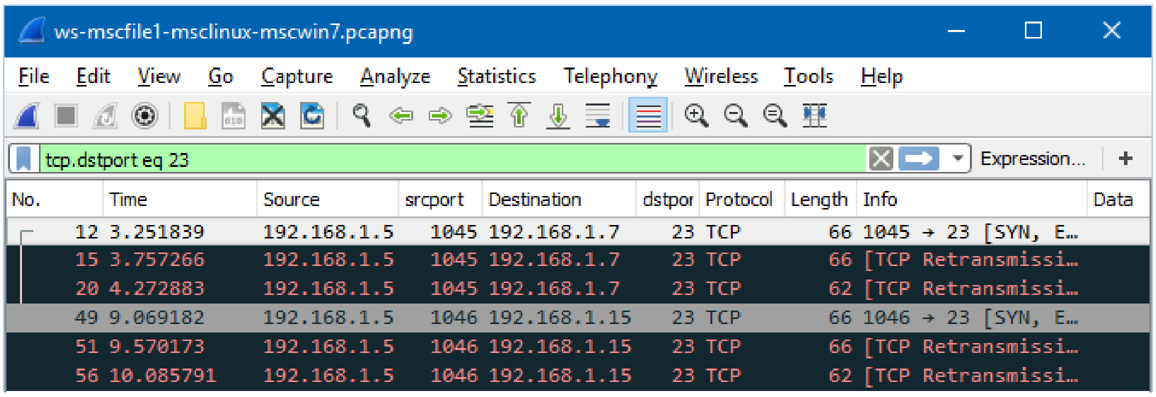
Figure 4.9: Experiment 2: Connection from 192.168.1.5 to 192.168.1.7 denied (Log Insight)

The same results are evident in Figure 4.10, which shows the connection attempt from *mscfile1* (192.168.1.5) to *mscwin7* (192.168.1.15) also being denied.


Events
Field Table
Event Types
Event Trends

timestamp	source	view_nsx_firewall_action	vmw_nsx_firewall_ruleid	vmw_nsx_firewall_protocol	vmw_nsx_firewall_src	vmw_nsx_firewall_dst	vmw_nsx_firewall_dst_ip_port	vmw_nsx_dst_port
2018-01-14 19:46:24.579	192.168.1.8	REJECT	1005	TCP	192.168.1.5	192.168.1.15	192.168.1.15/23	23
2018-01-14 19:46:24.022	192.168.1.8	REJECT	1005	TCP	192.168.1.5	192.168.1.15	192.168.1.15/23	23
2018-01-14 19:46:23.468	192.168.1.8	REJECT	1005	TCP	192.168.1.5	192.168.1.15	192.168.1.15/23	23

Figure 4.10: Experiment 2: Connection from 192.168.1.5 to 192.168.1.15 denied (Log Insight)



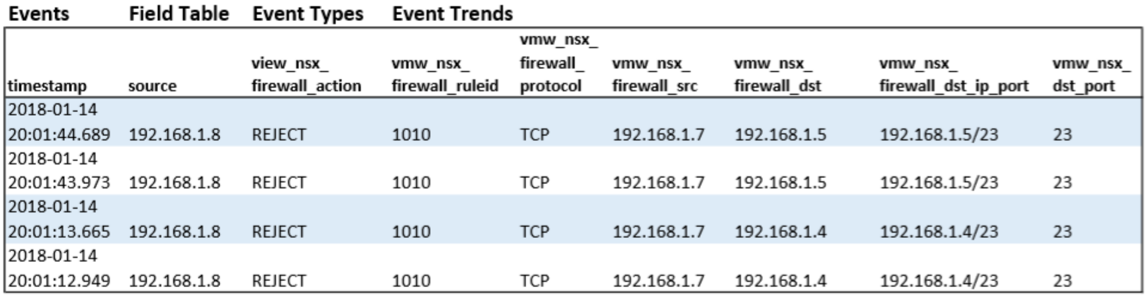
tcp.dstport eq 23

```
No.   Time        Source       srcport Destination   dstpor Protocol Length Info                 Data
   12 3.251839    192.168.1.5  1045 192.168.1.7      23 TCP     66 1045 → 23 [SYN, E…
   15 3.757266    192.168.1.5  1045 192.168.1.7      23 TCP     66 [TCP Retransmissi…
   20 4.272883    192.168.1.5  1045 192.168.1.7      23 TCP     62 [TCP Retransmissi…
   49 9.069182    192.168.1.5  1046 192.168.1.15     23 TCP     66 1046 → 23 [SYN, E…
   51 9.570173    192.168.1.5  1046 192.168.1.15     23 TCP     66 [TCP Retransmissi…
   56 10.085791   192.168.1.5  1046 192.168.1.15     23 TCP     62 [TCP Retransmissi…
```

Figure 4.11: Experiment 2: Connection from 192.168.1.5 to 192.168.1.7 and 192.168.1.15 denied (Wireshark)

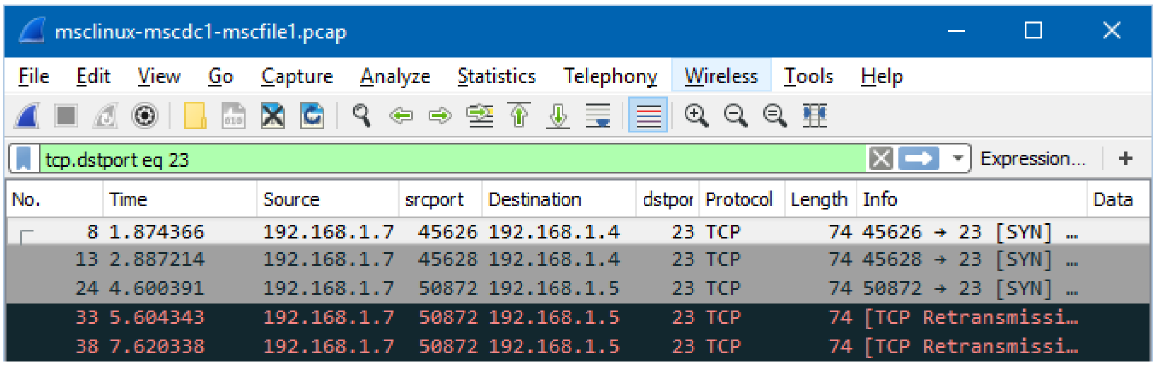
4.3.3 msclinux.compsci.local (192.168.1.7)

Figure 4.12 confirms that *msclinux* (192.168.1.7) is blocked from connecting to both *mscdc1* (192.168.1.4) and *mscfile* (192.168.1.5). Figure 4.13 shows the Wireshark packet capture equivalent.


Events
Field Table
Event Types
Event Trends

timestamp	source	view_nsx_firewall_action	vmw_nsx_firewall_ruleid	vmw_nsx_firewall_protocol	vmw_nsx_firewall_src	vmw_nsx_firewall_dst	vmw_nsx_firewall_dst_ip_port	vmw_nsx_dst_port
2018-01-14 20:01:44.689	192.168.1.8	REJECT	1010	TCP	192.168.1.7	192.168.1.5	192.168.1.5/23	23
2018-01-14 20:01:43.973	192.168.1.8	REJECT	1010	TCP	192.168.1.7	192.168.1.5	192.168.1.5/23	23
2018-01-14 20:01:13.665	192.168.1.8	REJECT	1010	TCP	192.168.1.7	192.168.1.4	192.168.1.4/23	23
2018-01-14 20:01:12.949	192.168.1.8	REJECT	1010	TCP	192.168.1.7	192.168.1.4	192.168.1.4/23	23

Figure 4.12: Experiment 2: Connection from 192.168.1.7 to 192.168.1.4 and 192.168.1.5 denied (Log Insight)



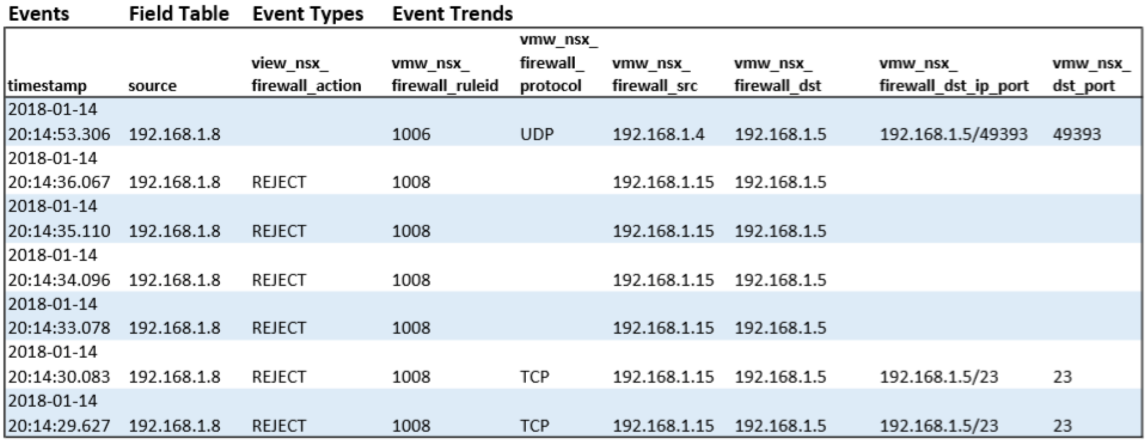
tcp.dstport eq 23

No.	Time	Source	srcport	Destination	dstpor	Protocol	Length	Info	Data
8	1.874366	192.168.1.7	45626	192.168.1.4	23	TCP	74	45626 → 23 [SYN] …	
13	2.887214	192.168.1.7	45628	192.168.1.4	23	TCP	74	45628 → 23 [SYN] …	
24	4.600391	192.168.1.7	50872	192.168.1.5	23	TCP	74	50872 → 23 [SYN] …	
33	5.604343	192.168.1.7	50872	192.168.1.5	23	TCP	74	[TCP Retransmissi…	
38	7.620338	192.168.1.7	50872	192.168.1.5	23	TCP	74	[TCP Retransmissi…	

Figure 4.13: Experiment 2: Connection from 192.168.1.7 to 192.168.1.4 and 192.168.1.5 denied (Wireshark)

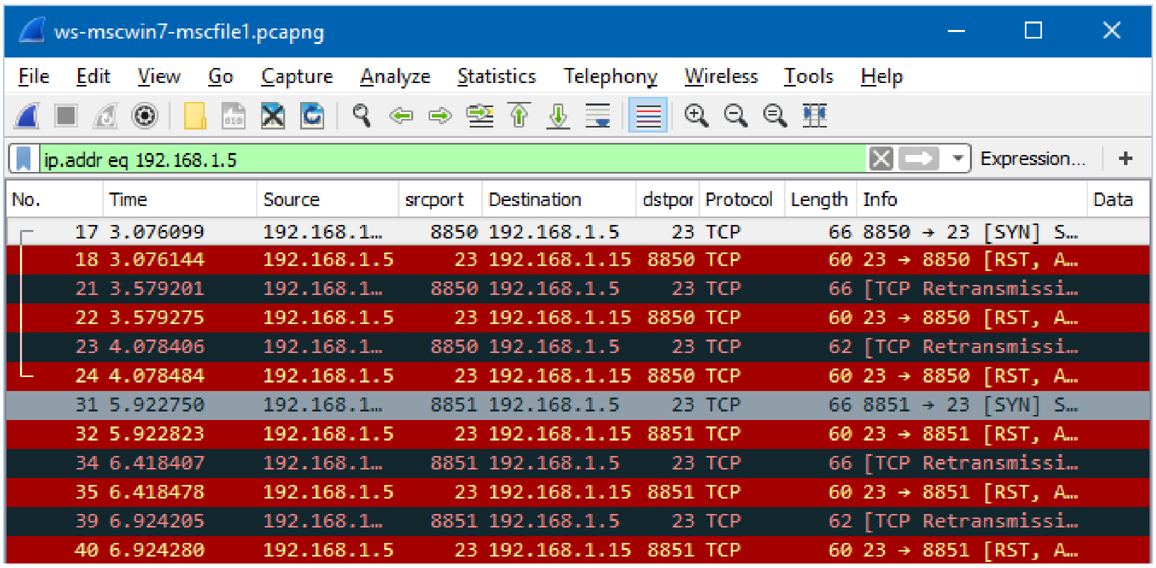
4.3.4 mscwin7.compsci.local (192.168.1.15)

Lastly, Figure 4.14 verifies that *mscwin7* (192.168.1.15) is blocked from connecting to *mscfile1* (192.168.1.5). Figure 4.15 shows the Wireshark packet capture equivalent.


Events
Field Table
Event Types
Event Trends

timestamp	source	view_nsx_firewall_action	vmw_nsx_firewall_ruleid	vmw_nsx_firewall_protocol	vmw_nsx_firewall_src	vmw_nsx_firewall_dst	vmw_nsx_firewall_dst_ip_port	vmw_nsx_dst_port
2018-01-14 20:14:53.306	192.168.1.8		1006	UDP	192.168.1.4	192.168.1.5	192.168.1.5/49393	49393
2018-01-14 20:14:36.067	192.168.1.8	REJECT	1008		192.168.1.15	192.168.1.5		
2018-01-14 20:14:35.110	192.168.1.8	REJECT	1008		192.168.1.15	192.168.1.5		
2018-01-14 20:14:34.096	192.168.1.8	REJECT	1008		192.168.1.15	192.168.1.5		
2018-01-14 20:14:33.078	192.168.1.8	REJECT	1008		192.168.1.15	192.168.1.5		
2018-01-14 20:14:30.083	192.168.1.8	REJECT	1008	TCP	192.168.1.15	192.168.1.5	192.168.1.5/23	23
2018-01-14 20:14:29.627	192.168.1.8	REJECT	1008	TCP	192.168.1.15	192.168.1.5	192.168.1.5/23	23

Figure 4.14: Experiment 2: Connection from 192.168.1.15 to 192.168.1.5 denied (Log Insight)



ip.addr eq 192.168.1.5

No.	Time	Source	srcport	Destination	dstpor	Protocol	Length	Info	Data
17	3.076099	192.168.1…	8850	192.168.1.5	23	TCP	66	8850 → 23 [SYN] S…	
18	3.076144	192.168.1.5	23	192.168.1.15	8850	TCP	60	23 → 8850 [RST, A…	
21	3.579201	192.168.1…	8850	192.168.1.5	23	TCP	66	[TCP Retransmissi…	
22	3.579275	192.168.1.5	23	192.168.1.15	8850	TCP	60	23 → 8850 [RST, A…	
23	4.078406	192.168.1…	8850	192.168.1.5	23	TCP	62	[TCP Retransmissi…	
24	4.078484	192.168.1.5	23	192.168.1.15	8850	TCP	60	23 → 8850 [RST, A…	
31	5.922750	192.168.1…	8851	192.168.1.5	23	TCP	66	8851 → 23 [SYN] S…	
32	5.922823	192.168.1.5	23	192.168.1.15	8851	TCP	60	23 → 8851 [RST, A…	
34	6.418407	192.168.1…	8851	192.168.1.5	23	TCP	66	[TCP Retransmissi…	
35	6.418478	192.168.1.5	23	192.168.1.15	8851	TCP	60	23 → 8851 [RST, A…	
39	6.924205	192.168.1…	8851	192.168.1.5	23	TCP	62	[TCP Retransmissi…	
40	6.924280	192.168.1.5	23	192.168.1.15	8851	TCP	60	23 → 8851 [RST, A…	

Figure 4.15: Experiment 2: Connection from 192.168.1.15 to 192.168.1.5 denied (Wireshark)

4.4 Experiment 3: IPsec Server and Domain Isolation

In Section 3.4, the concept of SDI as a cost effective candidate for micro-segmentation was discussed. Analysis of the SDI traffic flow packet captures are shown below to illustrate evidence of their compliance with the simple security policy defined in Section 3.4.3, and summarised below:

- **mscdc1** (172.16.64.4) may connect to any device

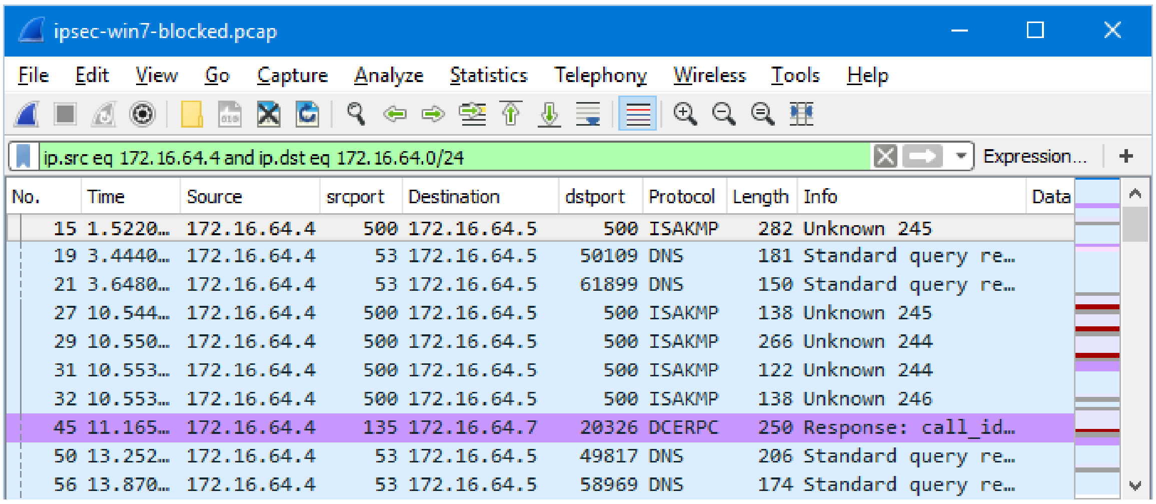
- **mscfile1** (172.16.64.5) may not connect to *msclinux* or *mscwin7*
- **msclinux** (172.16.64.6) may not connect to *mscfile1* or *mscwin7*
- **mscwin7** (172.16.64.7) may connect to *mscdc1* or *msclinux*

The default IPsec settings had the standard options configured to enable AH and ESP protocols for securing traffic. As discussed in Section 3.4.4, for the majority of packet captures it was necessary to enable null encapsulation in the IPsec GPO to facilitate decoding of IPsec data payloads using a Wireshark protocol dissector, which would otherwise be impossible with the standard IPsec ESP encryption algorithm.

4.4.1 mscdc1.compsci.local (172.16.64.4)

Two micro-segmentation scenarios were tested pertaining to AD DS. The first scenario included AD as part of the IPsec SDI environment, which means all communications between AD and endpoints were IPsec-enabled. While fully functional, this approach introduced significant issues that might constrain adoption of this particular implementation. The second scenario excluded AD from the IPsec SDI environment. This was the formally recommended approach, and implementing SDI based on this scenario was accompanied by a significant reduction in the complexity of deployment, especially for a dynamic IT environment. The advantages and disadvantages of both these deployment scenarios are discussed in Section 5.3.1.

The traffic flows of the particular packet capture shown in Figure 4.16 had the default IPsec encryption settings configured, so the data payloads were unable to be decoded. This demonstrates that *mscdc1* (172.16.64.4) is able to communicate with permitted endpoints within the lab environment.

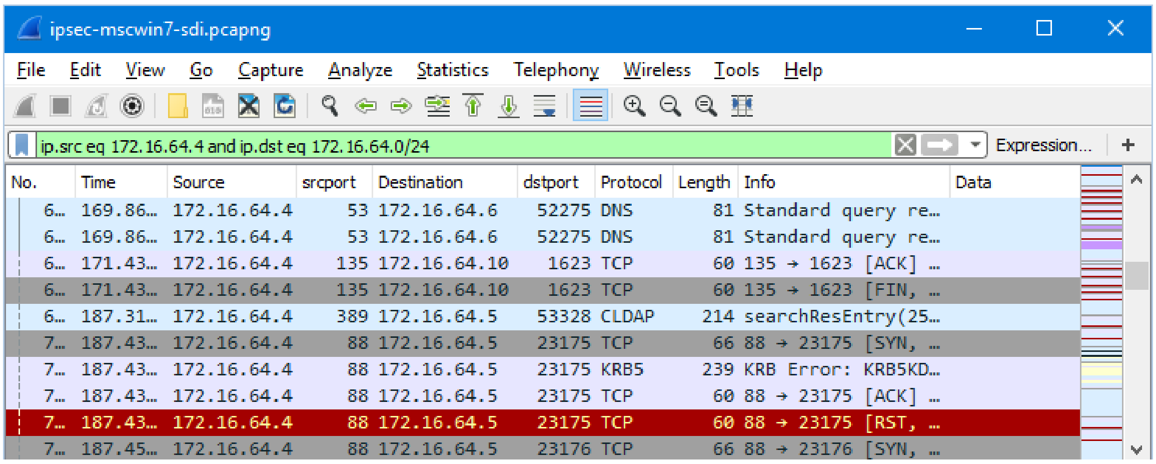

ipsec-win7-blocked.pcap
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
Expression…
+

ip.src eq 172.16.64.4 and ip.dst eq 172.16.64.0/24

```
No.  Time      Source       srcport Destination  dstport Protocol Length Info                     Data
  15 1.5220…   172.16.64.4      500 172.16.64.5      500 ISAKMP      282 Unknown 245
  19 3.4440…   172.16.64.4       53 172.16.64.5    50109 DNS         181 Standard query re…
  21 3.6480…   172.16.64.4       53 172.16.64.5    61899 DNS         150 Standard query re…
  27 10.544…   172.16.64.4      500 172.16.64.5      500 ISAKMP      138 Unknown 245
  29 10.550…   172.16.64.4      500 172.16.64.5      500 ISAKMP      266 Unknown 244
  31 10.553…   172.16.64.4      500 172.16.64.5      500 ISAKMP      122 Unknown 244
  32 10.553…   172.16.64.4      500 172.16.64.5      500 ISAKMP      138 Unknown 246
  45 11.165…   172.16.64.4      135 172.16.64.7    20326 DCERPC      250 Response: call_id…
  50 13.252…   172.16.64.4       53 172.16.64.5    49817 DNS         206 Standard query re…
  56 13.870…   172.16.64.4       53 172.16.64.5    58969 DNS         174 Standard query re…
```

Figure 4.16: Experiment 3: Connection from 172.16.64.4 to authorised endpoints permitted

Where AD is excluded from the IPsec SDI environment, there is no restriction on traffic flows between the DC (172.16.64.4) and all endpoints, as shown in Figure 4.17.


ipsec-mscwin7-sdi.pcapng
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
Expression…
+

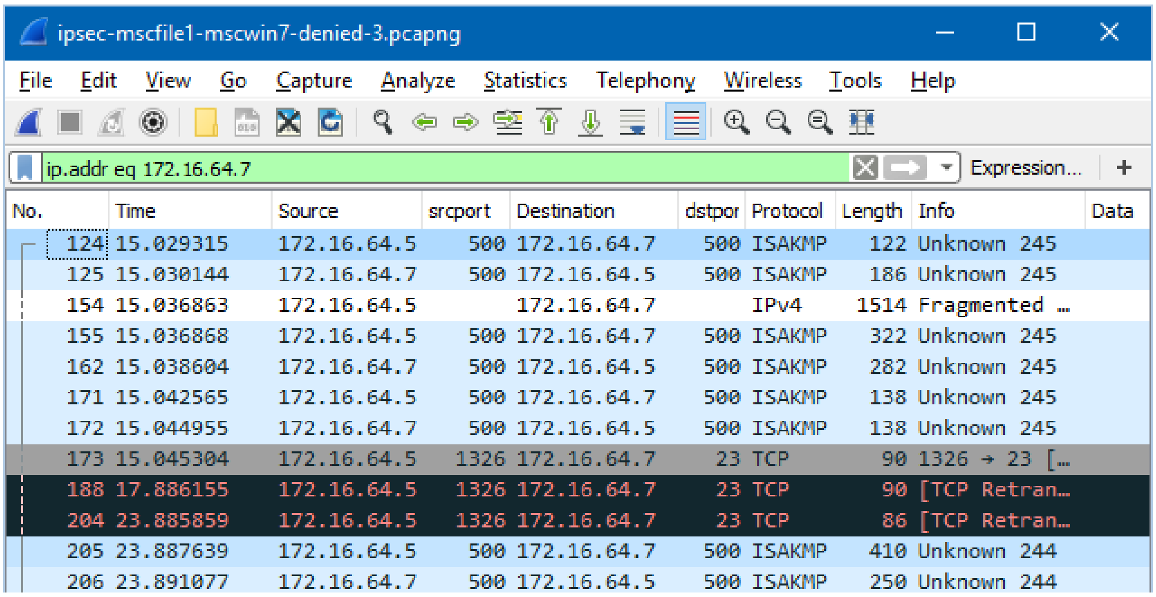
ip.src eq 172.16.64.4 and ip.dst eq 172.16.64.0/24

```
No.  Time     Source       srcport Destination   dstport Protocol Length Info                      Data
 6…  169.86…  172.16.64.4       53 172.16.64.6     52275 DNS          81 Standard query re…
 6…  169.86…  172.16.64.4       53 172.16.64.6     52275 DNS          81 Standard query re…
 6…  171.43…  172.16.64.4      135 172.16.64.10     1623 TCP          60 135 → 1623 [ACK] …
 6…  171.43…  172.16.64.4      135 172.16.64.10     1623 TCP          60 135 → 1623 [FIN, …
 6…  187.31…  172.16.64.4      389 172.16.64.5     53328 CLDAP       214 searchResEntry(25…
 7…  187.43…  172.16.64.4       88 172.16.64.5     23175 TCP          66 88 → 23175 [SYN, …
 7…  187.43…  172.16.64.4       88 172.16.64.5     23175 KRB5        239 KRB Error: KRB5KD…
 7…  187.43…  172.16.64.4       88 172.16.64.5     23175 TCP          60 88 → 23175 [ACK] …
 7…  187.43…  172.16.64.4       88 172.16.64.5     23175 TCP          60 88 → 23175 [RST, …
 7…  187.45…  172.16.64.4       88 172.16.64.5     23176 TCP          66 88 → 23176 [SYN, …
```

Figure 4.17: Experiment 3: Connection from 172.16.64.4 to all endpoints permitted

4.4.2 mscfile1.compsci.local (172.16.64.5)

Figure 4.18 confirms that a telnet connection attempt was initiated from *mscfile* (172.16.64.5) to *mscwin7* (172.16.64.7), showing TCP retransmission timeouts due to IPsec SDI rules blocking connectivity because *mscfile1* is not authorised to connect to the destination endpoint.



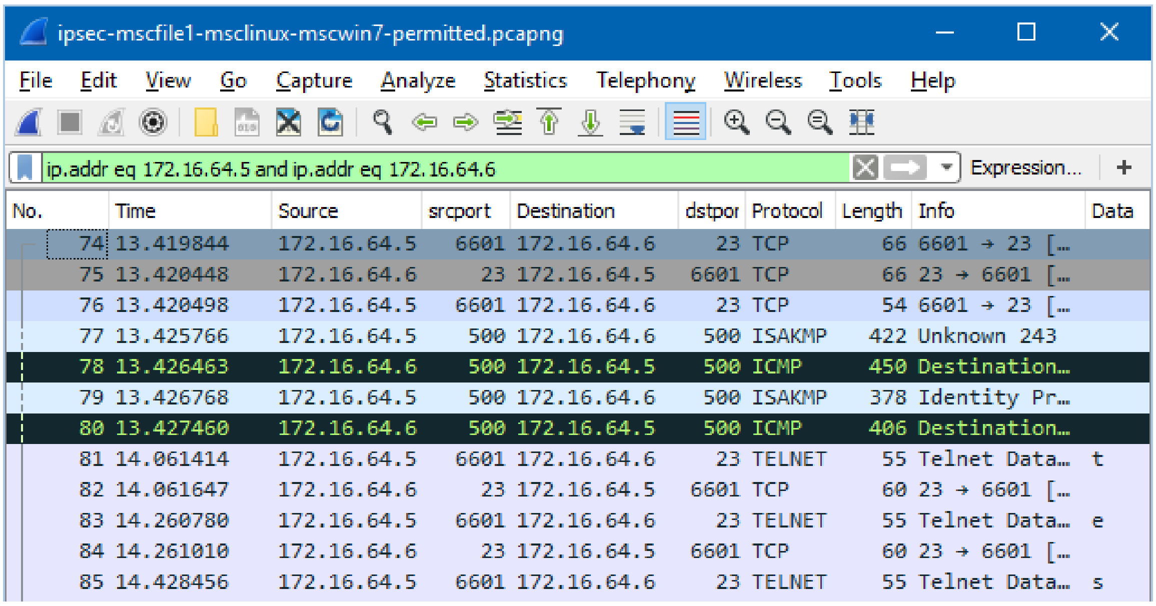
ipsec-mscfile1-mscwin7-denied-3.pcapng

ip.addr eq 172.16.64.7

No.	Time	Source	srcport	Destination	dstpor	Protocol	Length	Info	Data
124	15.029315	172.16.64.5	500	172.16.64.7	500	ISAKMP	122	Unknown 245	
125	15.030144	172.16.64.7	500	172.16.64.5	500	ISAKMP	186	Unknown 245	
154	15.036863	172.16.64.5		172.16.64.7		IPv4	1514	Fragmented …	
155	15.036868	172.16.64.5	500	172.16.64.7	500	ISAKMP	322	Unknown 245	
162	15.038604	172.16.64.7	500	172.16.64.5	500	ISAKMP	282	Unknown 245	
171	15.042565	172.16.64.5	500	172.16.64.7	500	ISAKMP	138	Unknown 245	
172	15.044955	172.16.64.7	500	172.16.64.5	500	ISAKMP	138	Unknown 245	
173	15.045304	172.16.64.5	1326	172.16.64.7	23	TCP	90	1326 → 23 […	
188	17.886155	172.16.64.5	1326	172.16.64.7	23	TCP	90	[TCP Retran…	
204	23.885859	172.16.64.5	1326	172.16.64.7	23	TCP	86	[TCP Retran…	
205	23.887639	172.16.64.5	500	172.16.64.7	500	ISAKMP	410	Unknown 244	
206	23.891077	172.16.64.7	500	172.16.64.5	500	ISAKMP	250	Unknown 244	

Figure 4.18: Experiment 3: Connection from 172.16.64.5 to 172.16.64.7 denied

However, the security policy objective of preventing *mscfile1* (172.16.64.5) from connecting to *msclinux* (172.16.64.6) was not enforced. Figure 4.19 shows that a telnet connection attempt initiated from *mscfile* to *msclinux* was successful, despite *mscfile1* not being a member of the *acl_computers_access_to_msclinux* security group authorised to connect to the destination endpoint.



ipsec-mscfile1-msclinux-mscwin7-permitted.pcapng

ip.addr eq 172.16.64.5 and ip.addr eq 172.16.64.6

No.	Time	Source	srcport	Destination	dstpor	Protocol	Length	Info	Data
74	13.419844	172.16.64.5	6601	172.16.64.6	23	TCP	66	6601 → 23 […	
75	13.420448	172.16.64.6	23	172.16.64.5	6601	TCP	66	23 → 6601 […	
76	13.420498	172.16.64.5	6601	172.16.64.6	23	TCP	54	6601 → 23 […	
77	13.425766	172.16.64.5	500	172.16.64.6	500	ISAKMP	422	Unknown 243	
78	13.426463	172.16.64.6	500	172.16.64.5	500	ICMP	450	Destination…	
79	13.426768	172.16.64.5	500	172.16.64.6	500	ISAKMP	378	Identity Pr…	
80	13.427460	172.16.64.6	500	172.16.64.5	500	ICMP	406	Destination…	
81	14.061414	172.16.64.5	6601	172.16.64.6	23	TELNET	55	Telnet Data…	t
82	14.061647	172.16.64.6	23	172.16.64.5	6601	TCP	60	23 → 6601 […	
83	14.260780	172.16.64.5	6601	172.16.64.6	23	TELNET	55	Telnet Data…	e
84	14.261010	172.16.64.6	23	172.16.64.5	6601	TCP	60	23 → 6601 […	
85	14.428456	172.16.64.5	6601	172.16.64.6	23	TELNET	55	Telnet Data…	s

Figure 4.19: Experiment 3: Connection from 172.16.64.5 to 172.16.64.7 permitted

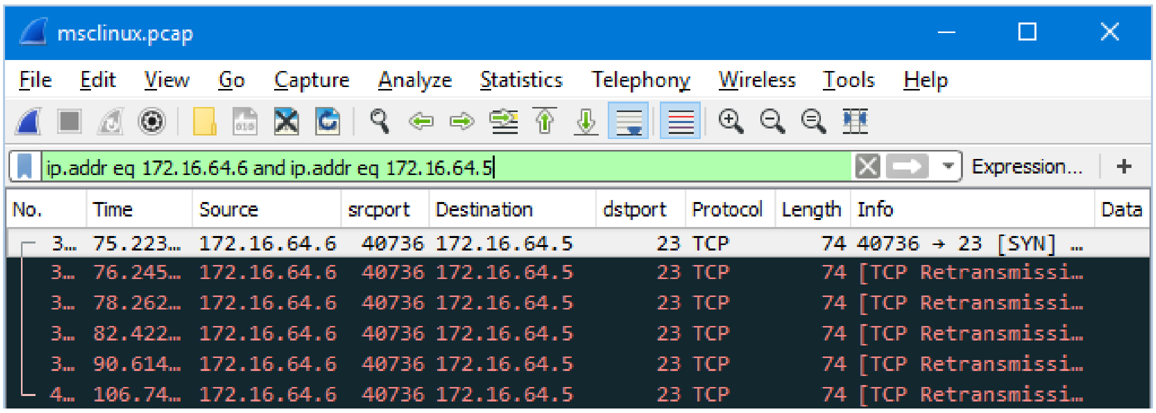
An identified workaround was to reconfigure the IPsec GPO firewall settings to block outbound connections, which effectively restricts communications to IPsec-enabled endpoints. This has the unwanted consequence of preventing communications with endpoints that are not configured with IPsec. An analysis of the failure of IPsec SDI to properly enforce this particular security policy objective is discussed in the next chapter (see Section 5.3.2).

4.4.3 msclinux.compsci.local (172.16.64.6)

One of the key objectives of the experiment was to implement IPsec SDI in a heterogeneous IT environment. However, critical issues were experienced in attempting a fully functional integration of the Linux server *msclinux* (172.16.64.6) with the AD domain.

While *msclinux* was successfully joined to the domain, serious challenges were experienced in configuring strongSwan, a Linux based IPsec implementation, to properly authenticate with AD. This means that IPsec was not enabled or configured on this server. To demonstrate the behaviour of IPsec SDI where machines are unable to communicate via IPsec or receive the AD GPO, multiple connection attempts were made from *msclinux* to *mscfile1* (172.16.64.5) and *mscwin7* (172.16.64.7). As expected, connectivity failed because these two endpoints, which were joined to the AD domain, had applied the IPsec connectivity rules prohibiting connections from unauthorised endpoints as shown by the TCP retransmission timeouts in Figures 4.20 and 4.21.

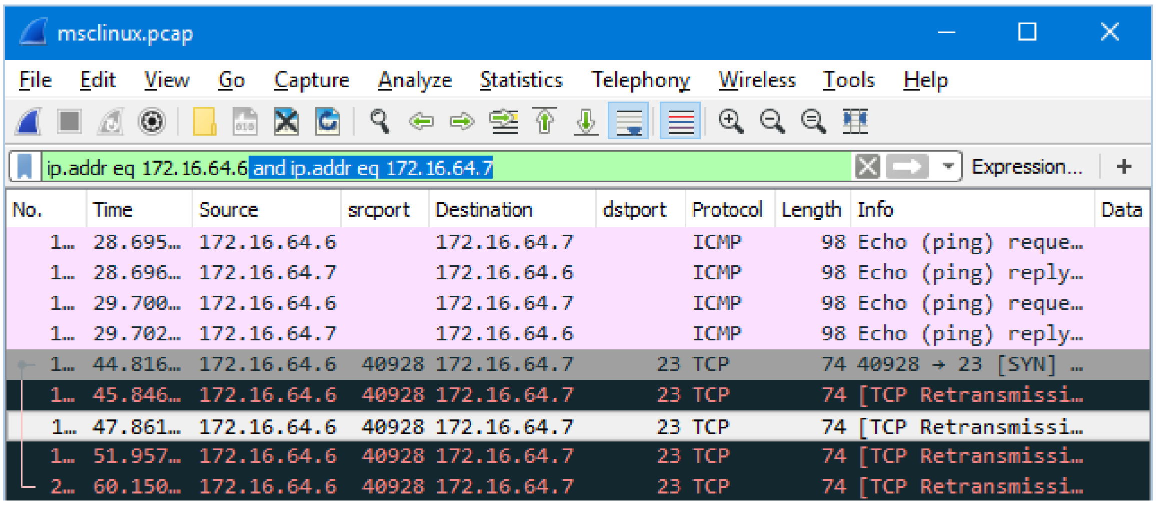
It is also evident from Figure 4.21 that Internet Message Control Protocol (ICMP) echo request and replies between *msclinux* and *mscwin7* are explicitly permitted due to the IPsec firewall rules in Section 3.4.3 being configured to allow ICMP to assist in network troubleshooting. This rule can, however, be disabled to achieve complete machine isolation if necessary.



ip.addr eq 172.16.64.6 and ip.addr eq 172.16.64.5

No.	Time	Source	srcport	Destination	dstport	Protocol	Length	Info	Data
3…	75.223…	172.16.64.6	40736	172.16.64.5	23	TCP	74	40736 → 23 [SYN] …	
3…	76.245…	172.16.64.6	40736	172.16.64.5	23	TCP	74	[TCP Retransmissi…	
3…	78.262…	172.16.64.6	40736	172.16.64.5	23	TCP	74	[TCP Retransmissi…	
3…	82.422…	172.16.64.6	40736	172.16.64.5	23	TCP	74	[TCP Retransmissi…	
3…	90.614…	172.16.64.6	40736	172.16.64.5	23	TCP	74	[TCP Retransmissi…	
4…	106.74…	172.16.64.6	40736	172.16.64.5	23	TCP	74	[TCP Retransmissi…	

Figure 4.20: Experiment 3: Connection from 172.16.64.6 to 172.16.64.5 denied


msclinux.pcap
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
Expression…
+

ip.addr eq 172.16.64.6 and ip.addr eq 172.16.64.7

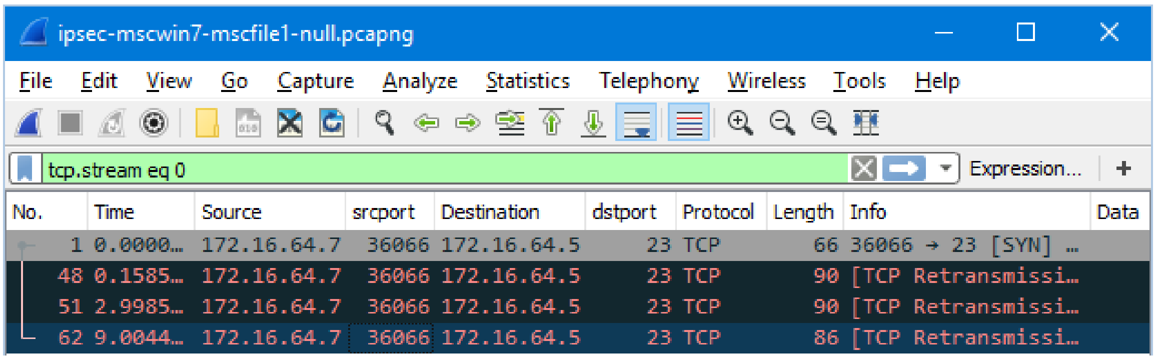
No.	Time	Source	srcport	Destination	dstport	Protocol	Length	Info	Data
1…	28.695…	172.16.64.6		172.16.64.7		ICMP	98	Echo (ping) reque…	
1…	28.696…	172.16.64.7		172.16.64.6		ICMP	98	Echo (ping) reply…	
1…	29.700…	172.16.64.6		172.16.64.7		ICMP	98	Echo (ping) reque…	
1…	29.702…	172.16.64.7		172.16.64.6		ICMP	98	Echo (ping) reply…	
1…	44.816…	172.16.64.6	40928	172.16.64.7	23	TCP	74	40928 → 23 [SYN] …	
1…	45.846…	172.16.64.6	40928	172.16.64.7	23	TCP	74	[TCP Retransmissi…	
1…	47.861…	172.16.64.6	40928	172.16.64.7	23	TCP	74	[TCP Retransmissi…	
1…	51.957…	172.16.64.6	40928	172.16.64.7	23	TCP	74	[TCP Retransmissi…	
2…	60.150…	172.16.64.6	40928	172.16.64.7	23	TCP	74	[TCP Retransmissi…	

Figure 4.21: Experiment 3: Connection from 172.16.64.6 to 172.16.64.7 denied

4.4.4 mscwin7.compsci.local (172.16.64.7)

The use of diverse operating system versions in the experimental setup was intended to validate that IPsec SDI was not dependent on a particular platform or version. Being required to standardise on one operating system version would significantly reduce the flexibility and likelihood of an IPsec SDI deployment in small-to-medium sized organisations.

Figure 4.22 shows that IPsec SDI is functional, as a connection attempt from *mscwin7* (172.16.64.7) to *mscfile1* (172.16.64.5) is blocked, as seen by the TCP retransmission timeouts.

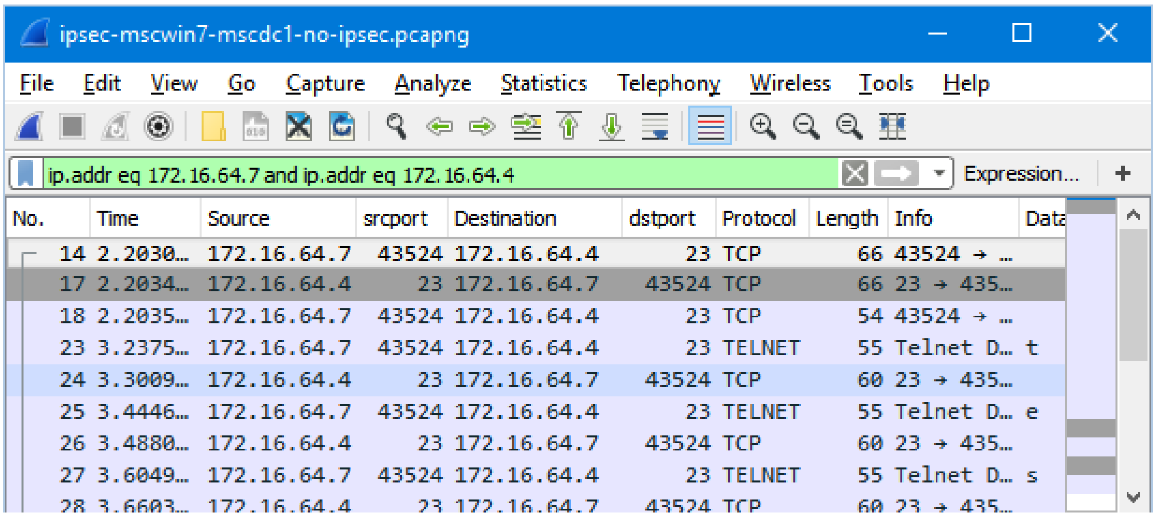

ipsec-mscwin7-mscfile1-null.pcapng
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
Expression…
+

tcp.stream eq 0

No.	Time	Source	srcport	Destination	dstport	Protocol	Length	Info	Data
1	0.0000…	172.16.64.7	36066	172.16.64.5	23	TCP	66	36066 → 23 [SYN] …	
48	0.1585…	172.16.64.7	36066	172.16.64.5	23	TCP	90	[TCP Retransmissi…	
51	2.9985…	172.16.64.7	36066	172.16.64.5	23	TCP	90	[TCP Retransmissi…	
62	9.0044…	172.16.64.7	36066	172.16.64.5	23	TCP	86	[TCP Retransmissi…	

Figure 4.22: Experiment 3: Connection from 172.16.64.7 to 172.16.64.5 denied

In a scenario where AD is excluded from IPsec SDI, a successful connection from *mscwin7* to *mscdc1* (172.16.64.4) can be seen in Figure 4.23. Another successful connection from *mscwin7* to *msclinux* (172.16.64.6) is also shown in Figure 4.24. However, this behaviour can most likely be attributed to the same technical misconfiguration of IPsec SDI that manifested in the same policy enforcement failure experienced in Section 4.4.2, and not as a result of the IPsec SDI policy permitting *mscwin7* connectivity to *msclinux*.

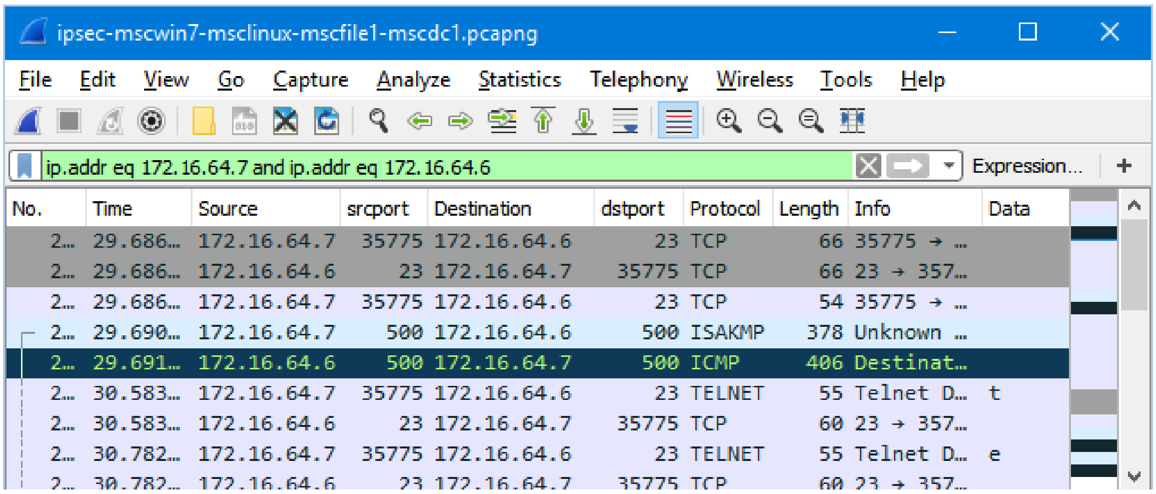


ipsec-mscwin7-mscdc1-no-ipsec.pcapng

ip.addr eq 172.16.64.7 and ip.addr eq 172.16.64.4

No.	Time	Source	srcport	Destination	dstport	Protocol	Length	Info	Data
14	2.2030…	172.16.64.7	43524	172.16.64.4	23	TCP	66	43524 → …	
17	2.2034…	172.16.64.4	23	172.16.64.7	43524	TCP	66	23 → 435…	
18	2.2035…	172.16.64.7	43524	172.16.64.4	23	TCP	54	43524 → …	
23	3.2375…	172.16.64.7	43524	172.16.64.4	23	TELNET	55	Telnet D…	t
24	3.3009…	172.16.64.4	23	172.16.64.7	43524	TCP	60	23 → 435…	
25	3.4446…	172.16.64.7	43524	172.16.64.4	23	TELNET	55	Telnet D…	e
26	3.4880…	172.16.64.4	23	172.16.64.7	43524	TCP	60	23 → 435…	
27	3.6049…	172.16.64.7	43524	172.16.64.4	23	TELNET	55	Telnet D…	s
28	3.6603…	172.16.64.4	23	172.16.64.7	43524	TCP	60	23 → 435…	

Figure 4.23: Experiment 3: Connection from 172.16.64.7 to 172.16.64.4 permitted



ipsec-mscwin7-msclinux-mscfile1-mscdc1.pcapng

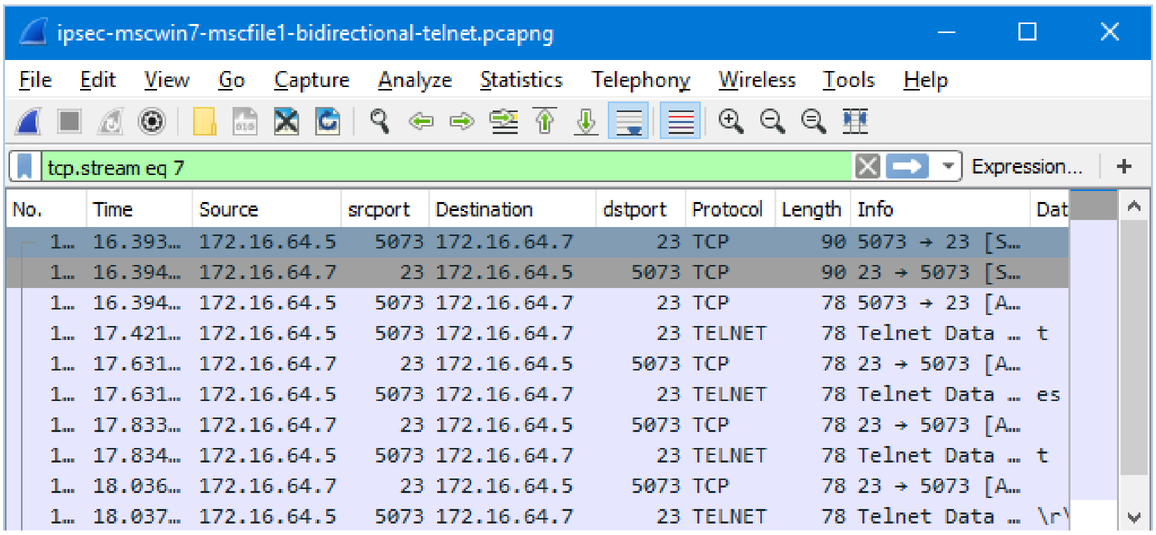
ip.addr eq 172.16.64.7 and ip.addr eq 172.16.64.6

No.	Time	Source	srcport	Destination	dstport	Protocol	Length	Info	Data
2…	29.686…	172.16.64.7	35775	172.16.64.6	23	TCP	66	35775 → …	
2…	29.686…	172.16.64.6	23	172.16.64.7	35775	TCP	66	23 → 357…	
2…	29.686…	172.16.64.7	35775	172.16.64.6	23	TCP	54	35775 → …	
2…	29.690…	172.16.64.7	500	172.16.64.6	500	ISAKMP	378	Unknown …	
2…	29.691…	172.16.64.6	500	172.16.64.7	500	ICMP	406	Destinat…	
2…	30.583…	172.16.64.7	35775	172.16.64.6	23	TELNET	55	Telnet D…	t
2…	30.583…	172.16.64.6	23	172.16.64.7	35775	TCP	60	23 → 357…	
2…	30.782…	172.16.64.7	35775	172.16.64.6	23	TELNET	55	Telnet D…	e
2…	30.782…	172.16.64.6	23	172.16.64.7	35775	TCP	60	23 → 357…	

Figure 4.24: Experiment 3: Connection from 172.16.64.7 to 172.16.64.6 permitted

4.4.5 Fine-grained Micro-segmentation

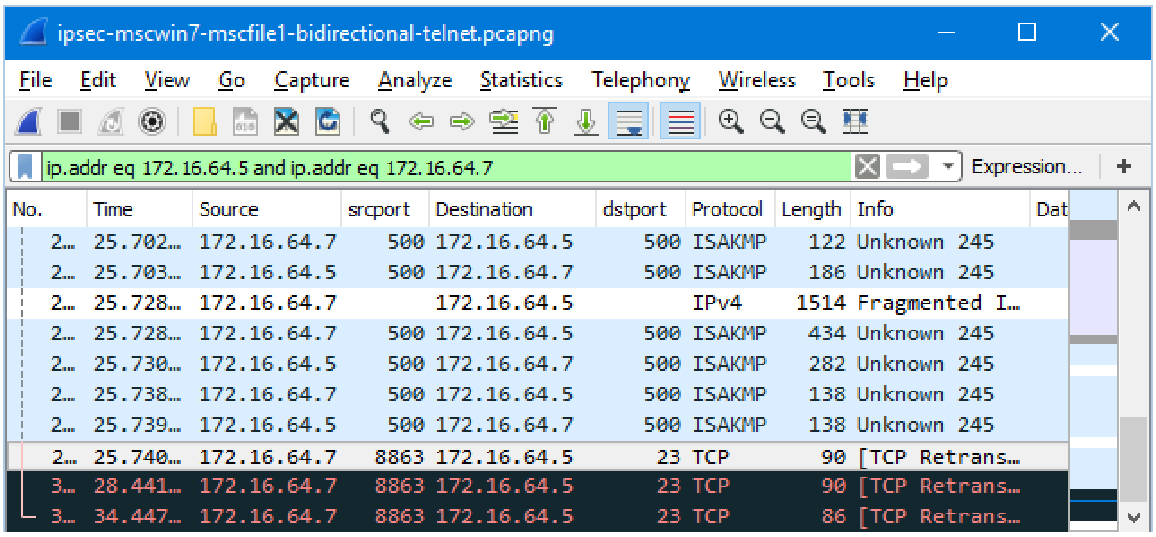
To demonstrate the potential for fine-grained micro-segmentation, the AD security groups were amended by adding the computer object *mscfile1* (172.16.64.5) to the security group *acl_computers_access_to_mscwin7*. This had the intended effect of allowing *mscfile1* connectivity to *mscwin7* (172.16.64.7), but not vice versa. Figure 4.25 shows that a telnet connection initiated from *mscfile1* to *mscwin7* is successful as shown by the numerous telnet data packets, while a telnet connection initiated from *mscwin7* to *mscfile1* was denied as seen by the TCP retransmission timeouts in Figure 4.26.



tcp.stream eq 7

No.	Time	Source	srcport	Destination	dstport	Protocol	Length	Info	Dat
1…	16.393…	172.16.64.5	5073	172.16.64.7	23	TCP	90	5073 → 23 [S…	
1…	16.394…	172.16.64.7	23	172.16.64.5	5073	TCP	90	23 → 5073 [S…	
1…	16.394…	172.16.64.5	5073	172.16.64.7	23	TCP	78	5073 → 23 [A…	
1…	17.421…	172.16.64.5	5073	172.16.64.7	23	TELNET	78	Telnet Data …	t
1…	17.631…	172.16.64.7	23	172.16.64.5	5073	TCP	78	23 → 5073 [A…	
1…	17.631…	172.16.64.5	5073	172.16.64.7	23	TELNET	78	Telnet Data …	es
1…	17.833…	172.16.64.7	23	172.16.64.5	5073	TCP	78	23 → 5073 [A…	
1…	17.834…	172.16.64.5	5073	172.16.64.7	23	TELNET	78	Telnet Data …	t
1…	18.036…	172.16.64.7	23	172.16.64.5	5073	TCP	78	23 → 5073 [A…	
1…	18.037…	172.16.64.5	5073	172.16.64.7	23	TELNET	78	Telnet Data …	\r\

Figure 4.25: Experiment 3: Connection from 172.16.64.5 to 172.16.64.7 permitted



ip.addr eq 172.16.64.5 and ip.addr eq 172.16.64.7

No.	Time	Source	srcport	Destination	dstport	Protocol	Length	Info	Dat
2…	25.702…	172.16.64.7	500	172.16.64.5	500	ISAKMP	122	Unknown 245	
2…	25.703…	172.16.64.5	500	172.16.64.7	500	ISAKMP	186	Unknown 245	
2…	25.728…	172.16.64.7		172.16.64.5		IPv4	1514	Fragmented I…	
2…	25.728…	172.16.64.7	500	172.16.64.5	500	ISAKMP	434	Unknown 245	
2…	25.730…	172.16.64.5	500	172.16.64.7	500	ISAKMP	282	Unknown 245	
2…	25.738…	172.16.64.7	500	172.16.64.5	500	ISAKMP	138	Unknown 245	
2…	25.739…	172.16.64.5	500	172.16.64.7	500	ISAKMP	138	Unknown 245	
2…	25.740…	172.16.64.7	8863	172.16.64.5	23	TCP	90	[TCP Retrans…	
3…	28.441…	172.16.64.7	8863	172.16.64.5	23	TCP	90	[TCP Retrans…	
3…	34.447…	172.16.64.7	8863	172.16.64.5	23	TCP	86	[TCP Retrans…	

Figure 4.26: Experiment 3: Connection from 172.16.64.7 to 172.16.64.5 denied

4.5 Summary

In this chapter, the criteria for a functional micro-segmentation approach were revealed. Supported by the use of a packet capture utility to inspect traffic flow between endpoints, it was verified whether endpoints were either permitted or blocked from communicating with each other depending on the associated security policy.

The packet analysis outcomes of each experimental approach were documented and analysed with sufficient rigour to confirm that all micro-segmentation approaches met the test objectives as expected, albeit with varying levels of complexity and effort. An exposition of the benefits and limitations of each approach forms the backbone of Chapter 5 that follows.

Chapter 5

Discussion of Micro-segmentation Approaches

After analysis of the experimental results in the preceding chapter, the resultant outcomes are dissected to examine the perceived or actual benefits and negative aspects of each micro-segmentation approach. The aim of each experimental approach is to assess whether it is functional, efficient and effective by assessing the implementation effort required, as well as the suitability of the approach for use in small-to-medium sized organisations. The criteria used for implementation effort encapsulates the relative availability and technical complexity of using existing hardware or software to enable micro-segmentation, compared to outright procurement of a dedicated solution. Constraints and limitations identified during the experimental processes are also mentioned.

5.1 Experiment 1: Switch ACLs

For static networks that make use of mainly static IP addresses or permanent DHCP leases, and provided that the network switch infrastructure supports ACL functionality, it was determined that the use of ACLs to control east-west VLAN traffic flows is an effective albeit inefficient method for implementing micro-segmentation to enforce a basic security policy.

5.1.1 Functional Micro-segmentation

ACLs are a low level method for implementing micro-segmentation that can be defined to enforce a particular security policy through simple permit and deny rules based on inspecting data packets, matching them against ACL rules and forwarding or dropping packets as appropriate. As extended ACLs operate on layer 3 of the OSI stack, this micro-segmentation approach is transparent to the endpoints regardless of whether they are physical, bare metal machines or virtual machines.

Where the network environment may include multiple remote sites, ACLs can also be configured to implement a hub-and-spoke network topology to block connectivity between remote sites while permitting restricted connectivity to the primary site.

5.1.2 Hardware Requirements

There are several constraints associated with using switch based ACLs to implement micro-segmentation. One of these is that the network switch must support VLAN and ACL functionality, which is usually only available in more expensive hardware and not in unmanaged switches. If ACL functionality is absent in an organisation's existing network infrastructure, it may require additional capital outlay to upgrade the switch hardware to an implementation that supports ACLs. For medium-to-large sized organisations, other micro-segmentation approaches that do not require replacement of existing hardware may be a preferred option.

5.1.3 Static Network Topology

Another constraint is a dependency on the network topology being fairly static, with a predictable IP addressing scheme being deployed to endpoints. Devices that are assigned DHCP leases via IP helpers that fall outside the defined VLAN and ACL range could potentially bypass the segmentation intent behind the security policy. This means that if a device is plugged into the switch and configured with a routable IP address not explicitly addressed by the ACL rules, that device may be able to establish unauthorised communications with other endpoints.

Future changes in the network topology driven by operational or business requirements will necessitate manual updates to each individual network switch ACL configuration. Large scale configuration updates that must be deployed to multiple network switches will impose significant operational load on network administrators and may necessitate additional investment in automated switch configuration management software that can keep track of dynamic security policy and networking changes within the environment.

5.1.4 ACL Maintenance

Similarly, within a dynamic environment, if an endpoint is moved to a different VLAN, the switch ACL is unable to follow the endpoint to enforce the security policy unless the ACL is manually updated in the switch configuration to reflect the endpoint's new network address. This necessarily implies that the security policy must be updated every time a device is added, changed or removed from the network environment, which introduces a significant device management burden.

5.1.5 Scalability of ACLs

In small networking environments, switch ACLs are a simple and effective means of implementing micro-segmentation. The challenge is that this approach does not scale up well for medium-to-large sized enterprises, and will likely severely limit the flexibility of the networking infrastructure to support evolving and changing business requirements in the absence of third party solutions that can centrally manage network switches and their associated ACLs.

5.2 Experiment 2: VMware NSX

Commercial virtualization solutions abound, most of which have the capability to support various network security approaches out of the box, including micro-segmentation. For the VMware virtualization platform that is primarily focused on data centres, VMware NSX is the SDN component that provides stateful packet inspection and firewall capabilities to provide granular levels of segmentation and isolation within virtual networks. In particular, the micro-segmentation capability implemented through distributed firewalls is an effective approach for controlling east-west traffic flows.

5.2.1 Platform Migration

For organisations with existing bare-metal server farms, adoption of the VMware virtualization solution would typically require a carefully planned strategy usually under the auspices of project management to migrate IT systems to the VMware platform as a starting point. This would necessitate substantial capital expenditure and commitment from the organisation to facilitate a successful migration exercise, which may prove difficult to motivate if the sole objective is to implement micro-segmentation. However, organisations that are already invested in the VMware platform will be able to rapidly implement the NSX networking and security stack, as NSX is a modular appliance that can integrate with existing installations.

5.2.2 Complexity

As the scope of the experiment was to evaluate the effectiveness of VMware's NSX-based implementation of micro-segmentation for comparison with other approaches, it was necessary to deploy a fully functional implementation of the VMware platform, as detailed in Section 3.3.

Implementation of the VMware platform is a complex process that requires a solid understanding of IT infrastructure, networking and virtualization concepts. Although there is a substantial amount of documentation available online with detailed installation procedures to guide implementation efforts, the environment is occasionally not very forgiving of errors made during deployment and this can lead to significant time and effort being expended towards resolution.

5.2.3 Functional Micro-segmentation

After installation and configuration of the NSX software appliance, the process for compiling distributed firewall rules is fairly trivial and requires only a basic understanding of firewall terminology and traffic flows. Given an understanding of these concepts, defining a security policy and translating the policy into functional firewall rules is a short, GUI-driven procedure in which the rules take effect immediately after being published to the environment. When used in conjunction with the Log Insight appliance for data analytics, the capability exists to correlate the firewall rules with inspection of traffic flows to acquire full visibility of the outcomes of micro-segmentation.

Leveraging off this toolset, it was determined that the NSX distributed firewall component (which installs kernel modules on virtual machines that sit on top of the NSX control-plane and management-plane fabric) is an effective approach for controlling east-west traffic flows. As was evident in the network flow graphs detailed in Chapter 4, NSX micro-segmentation proved to be highly effective in blocking traffic flows between endpoints that were not explicitly permitted by the security policy, as well as in isolating virtual machines within the VLAN.

5.2.4 Advanced Micro-segmentation

VMware's implementation of NSX SDN also allows for micro-segmentation to be enforced at multiple levels of networking abstraction. For example, hosts within a VLAN may be micro-segmented based on a particular virtual machine identifier, membership of a specific data centre, cluster, logical switch or distributed virtual switch to which the hosts are connected. This means that the security policy defined for a virtual machine is able to follow a particular endpoint to maintain enforcement of that policy, regardless of whether the endpoint is moved to a different virtual location or has its networking details changed.

When compared to the limitations of physical network switches in detecting and adapting to changes in endpoints, VMware NSX is considerably better equipped to manage dynamic changes in the virtualized infrastructure and network environment, and also has the benefit of supporting implementation of micro-segmentation within a diverse range of heterogeneous operating system platforms.

5.2.5 Cost Effectiveness

As mentioned previously in Section 5.2.1, the implicit capital and resource investment expenditure required to take advantage of a commercial micro-segmentation solution such as VMware NSX may prove to be too onerous and cost prohibitive for organisations that have not standardised on a particular virtualization platform, or have a substantial hardware footprint for which a cost-effective micro-segmentation solution is required.

Alternative commercial or open source hypervisors and micro-segmentation products may offer different segmentation approaches based on native, third party, overlay or hybrid models, along with varying tool capabilities and price points, but ultimately still remain out of reach for organisations with limited budget and resource constraints.

5.3 Experiment 3: IPsec Server and Domain Isolation

The IPsec SDI capability originally introduced in earlier editions of the Microsoft Windows operating system, and refined in subsequent iterations, was demonstrated in this experiment to be a workable approach for implementing micro-segmentation.

5.3.1 Functional Micro-segmentation

As Microsoft's implementation of IPsec is tightly integrated with AD DS, domain membership, GPOs, Windows Firewall with Advanced Security, OUs and security groups, these components can be combined to implement a holistic and cost-effective means of defining and enforcing connectivity rules for controlling traffic flow between discrete users and computers within or across VLANs. The concept of IPsec SDI is, for all intents and purposes, indistinguishable from micro-segmentation.

An organisational security policy can be enforced by configuring logical user and computer membership of explicitly defined security groups mapped to IPsec connection security rules and corresponding endpoint firewall rules. When an IPsec connection is established between two Windows based endpoints that have applied the IPsec GPO, part of the authentication and authorisation processes include validation of group membership such that an unauthorised user account on an authorised computer will be unable to connect to the destination endpoint, unless the user is explicitly added to the appropriate security group. Conversely, an authorised user on an unauthorised computer will be denied connectivity to the destination endpoint.

5.3.2 Operating System Compatibility

One of the original objectives of the micro-segmentation experiment was to accommodate a heterogeneous mix of operating system platforms. However, configuring IPsec on different operating system distributions turned out to be a complex issue relating to cross-platform incompatibility of IPsec implementations. For example, Windows platforms typically make use of Kerberos v5 to negotiate IPsec connections, but this method is not generally supported in Unix or Linux distributions. IPsec pre-shared keys were not considered an option due to the proclivity of keys being stored in clear text. The use of public key infrastructure to generate digital certificates for IPsec authentication is a more feasible option, but considerable effort will need to be expended to standardise an IPsec configuration setup that functions properly across different platforms. Even with certificate or pre-shared keys, various idiosyncracies in the IPsec packages that are available with various Linux based distributions mean that the IPsec configuration has to be specifically tailored for that particular distribution in order to make IPsec function as intended.

Inconsistencies within different versions of an operating system platform may also present IPsec compatibility challenges for a unified IPsec configuration deployment, as newer versions of operating systems may have feature sets that are unsupported in earlier versions.

5.3.3 Discrete Firewall Rules

A simplified IPsec implementation was configured that either permitted an endpoint full connectivity to another endpoint, or not at all. This coarse-grained level of connectivity may not be compatible with organisational security policies that have requirements for precise, fine-grained segmentation between endpoints. While it is possible to configure IPsec SDI with discrete firewall rules to cater for highly restricted connectivity rules, this may lead to firewall rule sprawl that becomes extremely challenging for systems administrators to accommodate as being outside the brief of their daily operational responsibilities.

5.3.4 End-to-End IPsec on Active Directory

A standard Microsoft recommendation for deploying IPsec is to exclude the AD DS environment from IPsec SDI, so that it can continue to service endpoints that have a dependency on common AD services, including Kerberos tickets, DNS, DHCP, authentication and core networking functionality. Exposing the AD servers to unauthenticated endpoints within a micro-segmented environment may introduce potential attack vectors, as an improperly secured AD server may allow an attacker to connect to the AD server and pivot from there to other IPsec-enabled endpoints and defeat micro-segmentation.

To address this, another experimental scenario was implemented that included AD in IPsec SDI while still allowing it to service unauthenticated (non-IPsec enabled) endpoints. However, this approach required careful and deliberate synchronisation across the environment that involved disabling GPO policies and services across servers and workstations before an amended GPO or updated firewall rules could be successfully deployed. This approach also has the potential to introduce a significant administrative burden when adding new devices to the domain, as they cannot be joined to the domain until IPsec is disabled on the AD server. As the margin for error is very narrow, with a misconfigured deployment having the potential to lock out all endpoints, enabling IPsec on AD DS environment may be impractical in a complex corporate environment with a significant number of endpoints, and especially where the majority of endpoints are geographically dispersed.

5.3.5 Excluding Active Directory from IPsec

Conversely, if the recommendation to exclude enabling IPsec on AD DS is adhered to, deployment of IPsec SDI to implement effective micro-segmentation between endpoints within a network segment is straightforward and predictable. Alternative security measures to manage the threat to AD can be implemented, or AD servers can be placed in a domain isolated zone where untrusted endpoints that have not been joined to the domain are blocked from connecting to the AD servers.

5.3.6 IPsec Limitations

IPsec is generally complex to set up correctly, and from an AD perspective, unexpected behaviour may result if an incorrect GPO is deployed to endpoints, including the potential to inadvertently lock out endpoints from being able to communicate across the network. Additionally, a basic configuration of IPsec SDI is dependent on computers being assigned static IP addresses to ensure that associated firewall rules apply only to these specific computers, although there are advanced configuration options available to work around this limitation. In large organisations that have a dynamic and rapidly changing environment, implementing IPsec SDI may not justify the administration overhead required to maintain the connectivity and segmentation requirements.

In some cases, as was seen in Section 4.4.3, a combination of IPsec SDI configuration complexity and limitations associated with non-Windows operating systems may result in a scenario where the desired security policy objective is not successfully enforced. In this particular case, the most probable root cause is user configuration error and not the underlying IPsec SDI design. This emphasises the requirement for skilled resources with the appropriate subject matter expertise to be retained for configuration and implementation of IPsec SDI.

As IPsec SDI has a dependency on the Windows platform, micro-segmentation using this approach is not a feasible option for organisations if isolation and authentication of different operating system platforms or devices is required. While it is possible to integrate non-Windows systems with a SDI deployment, this would likely require custom integration or a carefully designed SDI architecture and supporting security controls to protect and segment these non-IPsec aware devices outside of the Windows ecosystem.

5.3.7 Cost Effectiveness

For micro-segmentation implementations within a predominantly Microsoft Windows based environment, the default inclusion of IPsec SDI functionality as part of the operating system feature set means the barrier to entry is very low for organisations seeking to add another layer of security to their network environment. Deploying IPsec based micro-segmentation is a function of the organisation's risk appetite and resource availability with the appropriate knowledge and technical skill set. As with any technology implementation, testing of IPsec SDI should be carried out in a test or staging environment before proceeding with a full roll-out across the production environment, after all existing traffic data flows have been mapped and defining a proposed micro-segmentation configuration that is aligned with an organisational security policy based on business and security requirements.

5.4 Guidelines for Micro-segmentation

In testing the hypothesis from Section 1.2 stating that cost-effective micro-segmentation can be implemented using existing tools to control east-west traffic, a critical examination of the experimental outcomes shows that of the three approaches considered, the use of IPsec SDI is the most viable option for small-to-medium sized organisations, albeit with certain constraints.

Table 5.1 summarises the advantages, disadvantages and possible applications of each micro-segmentation approach in this context.

Table 5.1: Micro-segmentation Guidelines

Approach	**Advantages**	**Disadvantages**	**Possible uses**
Switch ACL	- Generally available - Simple to configure rules - Can segment layers 2 and 3 traffic - Effective and low resource requirement - Platform and OS agnostic	- Does not scale well - Maintenance overhead - Requires static network - Inefficient	- Suitable for small-to-medium sized networks - Interim solution - Specific use cases e.g. micro-segment de-militarized zones
VMware NSX	- Mature solution - Fine-grained segmentation capability - High scalability - Dynamic policy enforcement - Cater for north-south and east-west traffic - Platform and OS agnostic	- Costly and complex - Significant dependency on skilled resources - Requires adoption of VMware platform - Significant migration effort	- Suitable for medium-to-large organisations - Long term solution - Assumes software defined data centre strategy
IPsec SDI	- Capability exists in Windows OS platforms - IPsec protocol is proven - Caters for non-Windows platforms - Fine-grained micro-segmentation - Deployed using AD and GPO - Scalable	- Dependency on Windows platforms - Requires moderately skilled resources for implementation - Recommended configuration exposes AD, which needs mitigation - Including AD in SDI increases complexity - Cannot isolate non-IPsec aware devices	- Suitable for drop-in deployment within existing infrastructure - Viable for organisations without virtualization solutions

5.5 Summary

In this chapter it was determined on the basis of the experimental outcomes simulating a very limited scale Windows IT environment that the use of switch ACLs is a possible option for organisations with small scale, static network environments. More complex infrastructure and networking environments would require the procurement of a commercial third party tool with the capability to abstract and enhance switch management functionality to cater for automated, dynamic ACL and security policy management.

The VMware platform, integrated with the separate NSX appliance, offers a significantly superior micro-segmentation capability for providing rich security controls across the entire networking and infrastructure environment. Investing in this micro-segmentation would require adoption of and migration to the VMware SDDC platform, which may carry significant capital expenditure implications for which the target audience of small-to-medium sized organisations may lack the requisite investment appetite.

With this in mind, the IPsec SDI approach is a practical and cost effective option for organisations that have a business or security requirement to implement and enforce micro-segmentation, provided that the constraints of doing so are taken into account.

Chapter 6

Conclusion

This chapter presents a brief summary of the work done including the main conclusions drawn. This is followed by contributions made to the field and a suggestion for future work.

6.1 Summary of the Research

Complementing perimeter firewalls that control north-south traffic flows between firewall interfaces and external network perimeters, network micro-segmentation is a fine-grained mechanism to control east-west traffic flows within a network segment by restricting connectivity between endpoints to comply with a defined security policy. In the event of a foothold being gained on an endpoint within the network segment, micro-segmentation prevents unfettered lateral movement to other endpoints and network segments by containing the exposure to a significantly reduced attack surface area.

While commercial and open-source solutions for implementing micro-segmentation are available, small-to-medium sized organisations tend to lack the means or incentive for large-scale adoption of micro-segmentation tools that would require replacement of existing infrastructure along with substantial resource and capital expenditure.

In assessing the wide variety of available options to secure east-west traffic, two of the three segmentation approaches selected were focused on capitalising on existing IT infrastructure capability readily available within most organisations, and were contrasted with a commercial micro-segmentation approach to evaluate the effectiveness, efficiency, resource requirements and costs of the three approaches relative to each other. These segmentation approaches were tested in a simulated IT infrastructure and networking environment against a high level security policy describing the connectivity rules to be enforced for each endpoint.

Switch ACLs, being generally available in managed network switches, were found to be effective at enforcing micro-segmentation policies for static networks. ACLs do not scale up well from a maintenance perspective, however, as an increase in the number of ACL entries required to accommodate a large number of endpoints or policy rules can rapidly lead to a configuration schema with hundreds or thousands of lines. Additionally, manual tracking and amendment of rules in complex networks with dynamic IT assets and frequent security policy changes will be an increasingly onerous task in the absence of an automated switch configuration management tool.

Type 1 hypervisors, including VMware NSX with built-in SDN and hardware virtualization capabilities, provide visibility and insight into traffic flows and have the capability to implement micro-segmentation by enforcing security policies locally through a distributed firewall on each host. NSX is a mature solution that demonstrates effective enforcement of micro-segmentation transparently to the virtualized hosts, and has the ability to manage dynamic policy and asset changes to the extent that a security policy can be configured to follow an endpoint throughout its life cycle. However, adopting a strategy to migrate existing bare-metal or semi-virtualized infrastructure to Type 1 hypervisors as a software defined data centre platform requires significant capital and resource investments, which may be an undertaking beyond the means of small-to-medium organisations.

IPsec SDI is a hybrid of the Microsoft Windows AD DS, Group Policy and Advanced Firewall features that extends IPsec's capabilities to provide authenticated and encrypted connectivity between endpoints within and across network segments based on user and computer security groups. The functional capabilities of SDI in controlling east-west traffic is nearly indistinguishable from common micro-segmentation approaches available today, and as such can be readily and quickly deployed in any organisational IT infrastructure based on AD.

Apart from the relative complexity of configuring IPsec SDI, another limitation is that due to its reliance on the Windows platform, granular levels of authorisation and authentication will not necessarily be available on endpoints with different operating system platforms. Additionally, there may be technical issues with compatibility between IPsec SDI and other operating system IPsec implementations that preclude full adoption as a micro-segmentation solution. The trust level of an organisation's network environment will also determine the practicality and associated complexity of including or excluding AD in the IPsec SDI scope.

6.2 Contributions of the Research

In this study it was established that IPsec SDI is a conventional, under-used and readily available network security control available to most AD environments, and which can be re-purposed to implement micro-segmentation as a quick or interim solution for organisations seeking to add another layer of defence in their network environment. Limitations of IPsec SDI deployments were identified that suggest its implementation is more practical for homogeneous IT environments and not heterogeneous operating system platforms as was originally envisaged.

6.3 Future Work

The functional elements of a zero trust architecture have considerable overlap with other network security constructs including switch ACLs, distributed firewalls, network virtualisation and software defined networking, and for which alternative approaches to micro-segmentation were developed based on a native, third party, overlay or hybrid model.

Instead of a "big bang" micro-segmentation approach, it is likely that there will be a hybrid implementation based on the adoption of a zero trust model designed to integrate with the existing perimeter environment as part of a phased approach. With this in mind, there are ample opportunities for implementation of new micro-segmentation approaches within existing networks towards supporting a full blown zero trust implementation across the organisation.

When considering the scope and limitations for securing typical small-to-medium sized organisations, it is believed that an overlay micro-segmentation based model may be the most practical approach as it is decoupled from the underlying infrastructure and which takes advantage of the operating system's built-in packet filtering mechanisms, such as Windows Filtering Platform, Linux iptables, layer 4 firewalls and ACLs in network switches.

As overlay based micro-segmentation could provide complete application visibility and insight into processes that execute locally or across the network, since it integrates with heterogeneous environments, this strategy could provide another cost-effective option for organisations that seek to quickly implement micro-segmentation without needing to re-architect the existing environment, and with none of the restrictions or limitations of IPsec SDI based micro-segmentation. This approach remains to be explored in future work.

References

Al-Shabibi, A., Leenheer, M. D., Gerola, M., Koshibe, A., Snow, W., and Parilkar, G. OpenVirteX: A Network Hypervisor. In *Proceedings of Open Networking Summit*, pages 1–2. 2014.

Al-Shaer, E. Automated Firewall Analytics. Springer, 2014.

Bera, P., Maity, S., and Ghosh, S. Generating policy based security implementation in enterprise network: a formal framework. In *Proceedings of the 3rd ACM Workshop on Assurable and Usable Security Configuration*, pages 1–8. 2010. doi:10.1145/1866898.1866900.

Bishop, D. Step-by-Step Guide to Deploying Windows Firewall and IPsec Policies. Technical report, Microsoft Corp, 2009.

Blaze, M., Ioannidis, J., and Keromytis, A. Trust Management for IPSec. *Transactions on Information and System Security (TISSEC)*, 5(2):95–118, 2002.

Caldwell, T. The eagle has landed: part one. *Computer Fraud and Security Security*, 2015(12):5–9, 2015. doi:10.1016/S1361-3723(15)30110-X.

Chowdhury, N. M. K. and Boutaba, R. A survey of network virtualization. *Computer Networks*, 54(5):862–876, apr 2009. doi:10.1016/j.comnet.2009.10.017.

Cisco. Catalyst 2960 and 2960-S Switches Software Configuration Guide, Release 15.0(2)SE. Technical report, Cisco Systems, Inc., 2014.

Cisco. Layer 3 versus Layer 2 Switch for VLANs - Cisco Meraki. 2015. Last accessed: 2017-12-29.
URL `https://documentation.meraki.com/MS/Layer_3_Switching/Layer_3_versus_Layer_2_Switch_for_VLANs`

Cisco Systems. Cisco ACI Virtualization Guide, Release 3.0(1). Technical report, Cisco Systems, Inc., 2017.

Clark, S., Coombes, D., Denny, C., Dixon, W., Harrison, R., and Ryan, S. Server and Domain Isolation Using IPsec and Group Policy. Technical report, Microsoft Corp, 2006.

Collier, G., Plassman, D., and Pegah, M. Virtualization's Next Frontier: Security. In *Proceedings of the 35th Annual ACM SIGUCCS Fall Conference*, pages 34–36. 2007.

Cummins, D. and Sanabria, A. Illumio aiming high at scalable security policy portability, all workloads. Technical report, 451 Research, LLC, 2016.

Day, J. and Zimmermann, H. The OSI reference model. In *Proceedings of the IEEE*, volume 71, pages 1334–1340. IEEE, 1983. doi:10.1109/PROC.1983.12775.

DeCusatis, C., Liengtiraphan, P., Sager, A., and Pinelli, M. Implementing Zero Trust Cloud Networks with Transport Access Control and First Packet Authentication. In *2016 IEEE International Conference on Smart Cloud*, pages 5–10. IEEE, 2016. doi:10.1109/SmartCloud.2016.22.

Department of Defense. Department of Defense Trusted Computer System Evaluation Criteria. Technical report, Department of Defense of the United States of America, 1985.

Dias, J. A Guide to Microsoft Active Directory (AD) Design. Technical report, Lawrence Livermore National Laboratory, 2002.

Downin, K. and LaFountain, S. Microsoft Windows 2000 IPsec Guide. Technical report, National Security Agency, United States of America, 2001.

Elkeelany, O., Matalgah, M., Sheikh, K., Thaker, M., Chaudhry, G., Medhi, D., and Qaddour, J. Performance analysis of IPSec protocol: encryption and authentication. In *2002 IEEE International Conference on Communications. Conference Proceedings. ICC 2002 (Cat. No.02CH37333)*, volume 2, pages 1164–1168. IEEE, 2002. doi:10.1109/ICC.2002.997033.

Fayyad-Kazan, H., Perneel, L., and Timmerman, M. Benchmarking the Performance of Microsoft Hyper-V server, VMware ESXi and Xen Hypervisors. *Journal of Emerging Trends in Computing and Information Sciences*, 4(12):922–933, 2013.

Ferguson, N. and Schneier, B. A Cryptographic Evaluation of IPsec. Technical report, Counterpane Internet Security, Inc., 1999.

Frias-Martinez, V., Sherrick, J., Stolfo, S. J., and Keromytis, A. D. A Network Access Control Mechanism Based on Behavior Profiles. In *2009 Annual Computer Security Applications Conference*, pages 1–10. IEEE, 2009.

Gartner Inc. Defense-In-Depth with Internal Segmentation Firewalls. Technical report, Gartner Inc., 2016.

Gilman, E. How We Ensure PagerDuty is Secure for our Customers. 2014. Last accessed: 2018-01-31.
URL `https://www.pagerduty.com/blog/pagerduty-security/`

Gilman, E. and Barth, D. Zero Trust Networks. O'Reilly Media, Inc., first edition, 2017.

Gouda, M. and Liu, X.-Y. Firewall design: consistency, completeness, and compactness. In *Proceedings of the 24th International Conference on Distributed Computing Systems (ICDCS'04)*, pages 1–8. IEEE, 2004.

Han, B., Gopalakrishnan, V., Ji, L., and Lee, S. Network Functions Virtualization: Challenges and Opportunities for Innovations. *IEEE Communications Magazine*, 53(2):90–97, 2015.

Holtzman, J. A Guide to Domain Isolation for Security Architects. Technical report, Microsoft Corp, 2005.

Illumio. Micro-segmentation. White Paper. Illumio, Inc. 2016.

Ioannidis, S., Keromytis, A. D., Bellovin, S. M., and Smith, J. M. Implementing a distributed firewall. In *Proceedings of the 7th ACM Conference on Computer and Communications Security - CCS '00*, pages 190–199. 2000. doi:10.1145/352600.353052.

Jericho Forum. Collaboration Oriented Architecture. 2008. Last accessed: 2017-07-28.
URL `https://collaboration.opengroup.org/jericho/COA_v1.0.pdf`

Kim, H. and Feamster, N. Improving network management with software defined networking. *IEEE Communications Magazine*, 51(2):114–119, 2013. doi:10.1109/MCOM.2013.6461195.

Kindervag, J. Build Security Into Your Network's DNA: The Zero Trust Network Architecture. Technical report, Forrester, 2010a.

Kindervag, J. No More Chewy Centers: Introducing The Zero Trust Model Of Information Security. Technical report, Forrester, 2010b.

Kindervag, J., Ferrara, E., Holland, R., and Shey, H. Developing a Framework to Improve Critical Infrastructure. Technical report, The National Institute of Science and Technology, 2013.

Kiravuo, T., Sarela, M., and Manner, J. A Survey of Ethernet LAN Security. *IEEE Communications Surveys and Tutorials*, 15(3):1477–1491, 2013. doi:10.1109/SURV.2012.121112.00190.

Knight, P. and Lewis, C. Layer 2 and 3 virtual private networks: taxonomy, technology, and standardization efforts. *IEEE Communications Magazine*, 42(6):124–131, 2004. doi:10.1109/MCOM.2004.1304248.

Koponen, T., Amidon, K., Balland, P., Casado, M., Chanda, A., Fulton, B., Ganichev, I., Gross, J., Ingram, P., Jackson, E., Lambeth, A., Lenglet, R., Li, S.-H., Padmanabhan, A., Pettit, J., Pfaff, B., Ramanathan, R., Shenker, S., Shieh, A., Stribling, J., Thakkar, P., Wendlandt, D., Yip, A., and Zhang, R. Network Virtualization in Multi-tenant Datacenters. In *Proceedings of the 11th USENIX Symposium on Networked Systems Design and Implementation (NSDI 14)*, pages 203–216. International Computer Science Institute, 2014.

Leischner, G. and Tews, C. Security through VLAN segmentation: Isolating and securing critical assets without loss of usability. Technical report, Schweitzer Engineering Laboratories, Inc., 2007.

Mämmelä, O., Suomalainen, J., Ahola, K., and Vehkaperä, J. Towards Micro-Segmentation in 5G Network Security. In *2016 European Conference on Networks and Communications*, June. 2016.
URL `https://www.researchgate.net/publication/310447736_Towards_Micro-Segmentation_in_5G_Network_Security`

Markham, T. and Payne, C. Security at the network edge: A distributed firewall architecture. In *Proceedings of the DARPA Information Survivability Conference and Exposition II, DISCEX 2001*, volume 1, pages 279–286. 2001. doi:10.1109/DISCEX.2001.932222.

Martins, J., Ahmed, M., Raiciu, C., Olteanu, V., Honda, M., Huici, R., and Felipe, B. ClickOS and the Art of Network Function Virtualization. In *Proceedings of the 11th USENIX Conference on Networked Systems Design and Implementation (NSDI 14)*, pages 459–473. 2014.

Microsoft Corp. Domain Isolation Policy Design. 2016a. Last accessed: 2017-01-31. URL `https://docs.microsoft.com/en-us/previous-versions/windows/it-pro/windows-server-2012-R2-and-2012/jj721511(v%3Dws.11)`

Microsoft Corp. Server Isolation Policy Design | Microsoft Docs. 2016b. Last accessed: 2017-08-31. URL `https://docs.microsoft.com/en-us/previous-versions/windows/it-pro/windows-server-2012-R2-and-2012/jj721528(v%3Dws.11)`

Microsoft Corp. Server Isolation Policy Design Example. 2016c. Last accessed: 2017-08-31. URL `https://docs.microsoft.com/en-us/previous-versions/windows/it-pro/windows-server-2012-R2-and-2012/jj721557%28v%3Dws.11%29`

Miller, L. and Soto, J. Micro-Segmentation for Dummies. John Wiley and Sons, 2015, 68 pages.

Montemer, E. Network Security and Design. Technical report, Fast Lane Consulting and Education Services Inc., 2016.

Naldurg, P. and Campbell, R. H. Dynamic access control: preserving safety and trust for network defense operations. In *SACMAT '03: Proceedings of the eighth ACM Symposium on Access Control Models and Technologies*, pages 231–237. 2003.

Open Group. Jericho Forum Commandments. 2007. Last accessed: 2017-08-23. URL `https://collaboration.opengroup.org/jericho/commandments_v1.2.pdf`

Pfaff, B., Pettit, J., Koponen, T., Jackson, E. J., Zhou, A., Rajahalme, J., Gross, J., Wang, A., Stringer, J., Shelar, P., Amidon, K., and Casado, M. The Design and Implementation of Open vSwitch. In *Proceedings of the 10th USENIX Symposium on Networked Systems Design and Implementation (NSD I3)*, pages 117–130. 2015.

Platts, M. IPsec Domain Isolation: A Test Study. 2008. Last accessed: 2017-09-19. URL `https://blogs.technet.microsoft.com/networking/2008/05/30/ipsec-domain-isolation-a-test-study/`

Ramel, D. Datacenter Networking Report Shows Emergence of SDN. 2017. Last accessed: 2018-01-31.
URL `https://virtualizationreview.com/articles/2017/07/10/gartner-networking.aspx`

Rinehart, J. Demystifying Switch-based ACLs. 2013. Last accessed: 2017-09-29.
URL `https://www.globalknowledge.com`

Schulz, S., Sadeghi, A.-R., and Varadharajan, V. The Silence of the LANs: Efficient Leakage Resilience for IPsec VPNs. *IEEE Transactions on Information Forensics and Security*, 9(2):221–232, 2013.

Sezer, S., Scott-Hayward, S., Chouhan, P., Fraser, B., Lake, D., Finnegan, J., Viljoen, N., Miller, M., and Rao, N. Are we ready for SDN? Implementation challenges for software-defined networks. *IEEE Communications Magazine*, 51(7):36–43, 2013. doi: 10.1109/MCOM.2013.6553676.

Shea, R. and Liu, J. Network interface virtualization: Challenges and solutions. *IEEE Network*, 26(5):28–34, 2012. doi:10.1109/MNET.2012.6308072.

Stan, A. End-to-end encryption in Jericho networks. Technical report, Capgemini Nederland B.V., 2007.

Stepanek, R. Distributed Firewalls. *T-110.501 Seminar on Network Security*, 2001.

VArmour. Application-Aware Micro-Segmentation on Cisco ACI. Technical report, vArmour, Inc., 2017.

VMware. Data Center Micro-Segmentation. White Paper. VMware. 2014a.

VMware. NSX Installation Guide. Technical report, VMware, Inc., 2014b.

VMware. Beyond Firewalls and Virtual Appliances: Making Micro-Segmentation Work. Technical report, VMware, Inc., 2015a.

VMware. NSX Administration Guide. Technical report, VMware, Inc., 2015b.

VMware. vSphere 6.5 Installation and Setup. Technical report, VMware, Inc., 2017.

Wagner, N., Sahin, C. S., Pena, J., Riordan, J., and Neumayer, S. Capturing the Security Effects of Network Segmentation via a Continuous-time Markov Chain Model. In *Proceedings of the 50th Annual Simulation Symposium*, pages 17:1—-17:12. 2017a.

Wagner, N., Sahin, C. S., Winterrose, M., Riordan, J., Pena, J., Hanson, D., and Streilein, W. W. Towards Automated Cyber Decision Support: A Case Study on Network Segmentation for Security. In *2016 IEEE Symposium Series on Computational Intelligence, SSCI 2016*, March, pages 1–10. 2017b. doi:10.1109/SSCI.2016.7849908.

Wang, A., Iyer, M., Dutta, R., Rouskas, G. N., and Baldine, I. Network virtualization: Technologies, perspectives, and frontiers. *Journal of Lightwave Technology*, 31(4):523–537, 2013. doi:10.1109/JLT.2012.2213796.

Wang, C., Spatscheck, O., Gopalakrishnan, V., Xu, Y., and Applegate, D. Toward High-Performance and Scalable Network Functions Virtualization. *IEEE Internet Computing*, 20(6):10–20, 2016. doi:10.1109/MIC.2016.111.

Ward, R. and Beyer, B. BeyondCorp: A new approach to enterprise security. *Usenix Login: The Advanced Computing Systems Association*, 39(6):6–11, 2014.
URL `https://research.google.com/pubs/pub43231.html`

Wilmington, G. The VMware NSX Platform - Healthcare Series - Part 4.1: Microsegmentation Practical. 2016. Last accessed: 2018-01-27.
URL `https://vwilmo.wordpress.com/2016/11/`

Young, G. Technology Insight for Microsegmentation. 2017. Last accessed: 2017-08-19.
URL `https://www.gartner.com/doc/3640817/technology-insight-microsegmentation`