Zero-Trust-Based Access Control in a Multi-Layered and Micro-Segmented Network Infrastructure

University of Turku
Department of Computing
Master of Science (Tech) Thesis
Cyber Security
May 2026
Jesse Eskelinen

Supervisors:
Ismayil Hasanov
Petri Sainio

UNIVERSITY OF TURKU
Department of Computing

Jesse Eskelinen: Zero-Trust-Based Access Control in a Multi-Layered and Micro-Segmented Network Infrastructure

Master of Science (Tech) Thesis, 63 p.
Cyber Security
May 2026

The cybersecurity field is constantly evolving alongside different technological advances in products designed for purposes such as asset protection. Although advances in technology are present and different frameworks and standards are created, organizations might not be confident in implementing concepts in more complex environments. Zero-trust is one of these concepts and, more specifically, zero-trust-based access control.

This thesis investigates how an organization could approach zero-trust-based access control solutions in complex environments that topologically consist of multiple layers and micro-segmented networks. The main focus is on how zero-trust-based access control can be designed for different access scenarios in this type of environment and how the designs can be implemented.

To get insight into how this type of implementation can be done, a case study was carried out in which a generalized design was first introduced for zero-trust-based access control and then the said design was tailored to different access scenarios with the existing firewall solution in mind. Due to existing firewall solutions, a good base was already present in the environment for this type implementation, but the designs are still adaptable to other environments.

After the designs were implemented in the case study, validation was performed through which it was deemed that the implementation was effective in securing traffic. As no major issues were observed during the case study, it can be deemed that it is not difficult to implement this type of solution. Overall, since the case study was a success and was carried out in a complex environment, it shows that organizations can move towards zero-trust-based access control even in complex environments.

In conclusion, the thesis explores a well-known concept through a case study and proves through practical tests and a literature review that organizations can implement such a concept. The thesis additionally acknowledges that further research is possible through different scaling methods, which can lead to further validation.

Keywords: identity, networking, micro-segmentation, OSI, infrastructure, firewalls, layering, zero-trust

Contents

List of Figures

1 Introduction

For a long time, the cybersecurity field and the technologies associated with it have evolved to meet the newer requirements defined by various standards and guidelines, and firewall technology has advanced as a byproduct of this evolution. Originally, access to resources was restricted through packet filtering, which required bidirectional rules to be effective. In the early 2000s, stateful firewalls were introduced, enabling the use of unidirectional rules and state tables to allow connections in both directions for the required traffic. Both types of firewalls operated by allowing traffic if the Internet Protocol (IP) address and port matched the configured firewall rules, which was sufficient for a period of time. Later, Next-Generation Firewalls (NGFW) were developed, introducing the capability to filter traffic at the application layer, further improving security when accessing resources.

However, this approach introduced an inherent limitation, namely the implicit trust granted to users accessing resources from within specific networks. This limitation motivated the shift toward a zero-trust architecture.

1.1 Background

Zero-trust architecture in cybersecurity is not a new concept; it has been discussed since 2010 by Kindervag [1] and has since been further developed by various organizations. The fundamental principle of this architecture is to "never trust, always verify", which can be implemented through multiple technologies depending on the

type of resource being accessed.

For example, when logging into an email service, users are now commonly expected to use Multi-Factor Authentication (MFA), which provides additional assurance that the individual accessing the service is the legitimate user. It could be argued that although the zero-trust architectural model has existed for a long time, organizations have only recently begun to take it seriously. This shift is likely influenced by advocates such as John Kindervag, the creator of the zero-trust cybersecurity model, and Sami Laiho, who advocates for zero-trust principles.

Due to the emergence of different standards and the increasing push towards zero-trust architecture, this transition provided an opportunity for firewall manufacturers to develop new solutions or build upon existing ones, particularly in the context of network access control.

The relationship between firewalls and zero-trust architecture lies in the emergence of new implementations that enable identity-based access control rather than traditional address-based or NGFW application-based filtering. This development introduces new opportunities for network and security administrators to implement more precise security measures for resources that are already protected by firewalls.

1.2 Research Focus

As zero-trust architecture is a prominent topic in the computational field, there is a significant amount of discussion surrounding it. However, administrators may not always know how to act on this information or may encounter challenges when attempting to implement it in practice.

There are multiple approaches to creating a zero-trust architecture and applying its underlying concepts. While many solutions exist and there are various ways to begin implementing zero-trust within an organization, it may be unclear whether these types of solutions work across all network environments.

Currently there is a sector with lack of certainty when it comes to zero-trust-based access control, as there are multiple studies that researched the topic and came to some conclusions in different environments, but there is a limited amount of studies which look into zero-trust-based access control in an on-premises infrastructure. More specifically, on-premises infrastructure that is complex by nature due to existing layering and segmentation that have already been implemented with some zero-trust concepts in mind.

In order to address the lack of certainty around the topic with this study, two activities will be carried out. First, the theoretical foundations of the relevant underlying concepts will be examined through a literature review. Second, a case study will be conducted in which the solution is evaluated from a practical perspective and verified within a complex network infrastructure. This approach aims to demonstrate that the solution can be generalized and implemented across different environments.

The methodology used in this thesis is new to the topic, as a generalized design has been created for the purpose of the case study, which can be adapted to different scenarios in other environments, as proven in the research.

1.3 Research aim and Questions

The general aim of the research conducted in this study is to further investigate how zero-trust-based access control can be implemented to provide additional protection within an organizational network against malicious adversaries, demonstrating that this type of protection can be achieved even in complex infrastructures.

This thesis addresses the following research questions:

1. How can the effectiveness of identity-based access control be evaluated in a multi-layered, micro-segmented on-premises network, where a working imple-

mentation allows necessary access only for specified identities?

2. What implementation challenges arise when deploying zero-trust-based access control across a multi-vendor, multi-cluster firewall environment with heterogeneous user populations (internal, external/RDS, VPN), and if they do arise, how can they be categorized and mitigated?

3. To what extent does a generalized identity-based access control design transfer across access scenarios (internal endpoints, multi-user hosts, remote VPN), and can a generalized design be modularized through customization for different access scenarios to make the generalized design applicable in complex network infrastructures?

1.4 Outline of research methods

This research relies on two types of data; a literature review and a case study conducted within an organization during the writing of this thesis. In the case study, a generalized design of the proposed implementation will first be developed. Based on this generalized design, modified versions will then be created to accommodate different types of scenarios within the organization. During the implementation of the design, it will be possible to derive answers to the proposed research questions as the study is purely technical. The research aims to address the research questions, supported by the extended explanations justifying the conclusions.

1.5 Value of this research

As this thesis examines how zero-trust-based access control can be implemented within a complex network infrastructure to support the advancement of the zero-trust architectural model, it presents results and relevant discussion regarding what

administrators should expect from this type of implementation and whether they should consider adopting similar solutions within their own environments.

Due to the nature of the research, the findings can be generalized to other organizations using different NGFW solutions. This generalization is enabled by the proposed generalized design, which is implemented within a complex environment, demonstrating that the same design can be applied to organizations with less complex infrastructures as well. As discussed earlier, these generalization possibilities allow administrators to use this research as a foundation for initiating their transition toward zero-trust architecture, even within complex environments.

As this research addresses the implementation of zero-trust architecture for access control in a complex network infrastructure, it also provides a basis for future research. Other researchers can propose alternative solutions or build on the work presented in this thesis. One potential direction for extending this research would be to further scale the implementation and introduce additional complexity by incorporating multiple sites.

2 Literature Review

In the computing domain, there are some concepts that are decades old, dating back to the 1980s and some newer developments from the 2010s that are important for this research paper, among other topics discussed over the years. In order to fully understand them, it is necessary to explore various types of media created by other authors and organizations so that a baseline for this study is established. The baseline and relevant topics are essential to understand the research presented in this paper.

2.1 TCP/IP and OSI -Models

Currently, there are two types of models created for network stacks, the Transmission Control Protocol/Internet Protocol (TCP/IP) and Open Systems Interconnection (OSI) model. The TCP/IP model defines four layers, while the OSI reference model defines seven layers. Although the OSI model has influenced the design of current network devices and its layers are often referenced, it remained purely theoretical.

The difference between these two models is straightforward. The fourth layer of the TCP/IP model, which is the application layer, is divided into three separate layers in the OSI model: the session, presentation and application layers. Additionally, the network access layer, also referred to as the link layer, is divided into data link and physical layers.

The TCP/IP stack was first introduced in 1974 through Request for Comments

(RFC) 675 [2] by Cerf et al., but was not fully functional. As a result, the authors revised their work and published working specifications for IPv4 in 1981. These publications included RFC 791 [3], which defines the Internet Protocol and RFC 793 [4], which defines the Transmission Control Protocol. Through these papers, the Defense Advanced Research Projects Agency (DARPA) was able to create an early version of the Internet, known as the Advanced Research Projects Agency Network (ARPANET).

As defined in RFC 1122 [5] and visualized in Figure 2.1, the TCP/IP stack is composed of four different layers: the link layer, the Internet layer, the transport layer and the application layer. Each of these layers serves a specific purpose. In the network flow of a sending host, the application layer prepares data using different protocols, such as HTTPS and FTP. The transport layer breaks the data into segments. The Internet layer applies the relevant IP addresses to the packet headers and determines the most appropriate route for the packet. Finally, the link layer converts the packets into frames and physically distributes them to other devices.

The flow on the receiving end begins at the link layer, where the received data is converted from frames and forwarded to the Internet layer. The Internet layer validates and removes the IP headers. The data then proceeds to the transport layer for checksum verification and other validation checks, after which it is sent to the application layer and delivered to the appropriate application.

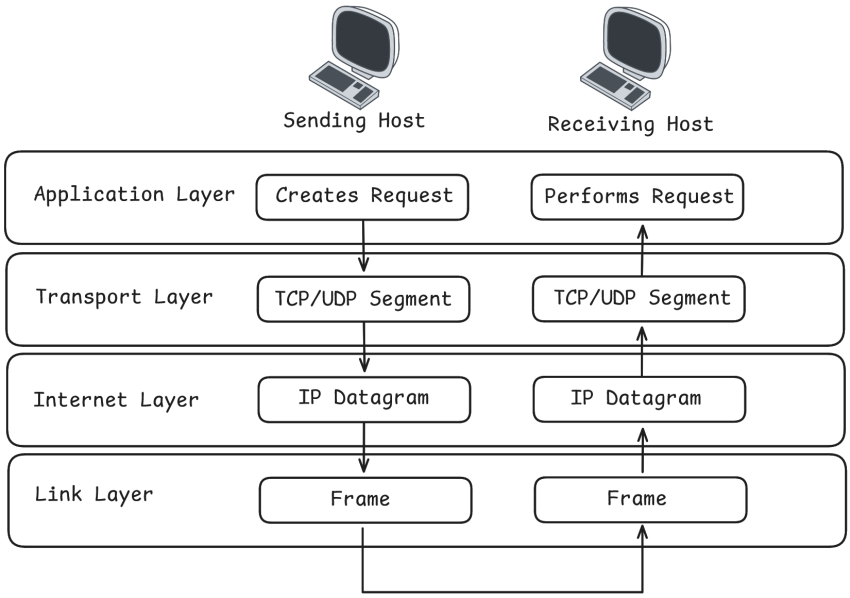

Figure 2.1: TCP/IP Protocol stack

As described in the study by Day and Zimmermann [6], the International Organization for Standardization (ISO) Technical Committee 97 formed a new subcommittee, SC16, to develop the OSI reference model. This subcommittee was tasked with creating a system in which devices would be able to communicate with any other device. This effort resulted in the creation of the OSI reference model, which consists of seven layers instead of the four defined in TCP/IP: application, presentation, session, transport, network, data link and physical layers.

From this structure, it is already possible to identify similarities with the TCP/IP model. The application, presentation and session layers can be mapped to the application layer of TCP/IP, while the data link and physical layers correspond to the link layer. As described in the study by Day and Zimmermann [6], the OSI reference model exhibited architectural issues that were categorized into general issues, lower-layer issues, middle-layer issues, upper-layer issues and OSI management issues.

Both OSI and TCP/IP aimed to fulfill the same fundamental goal. However,

as examined by Meyer and Zobrist [7], although the OSI reference model was well structured, TCP/IP was nourished in computing communities for a longer period of time. This resulted in greater maturity, ultimately causing it to "win" [8] the protocol race in the 1980s. As a consequence of TCP/IP "winning" over OSI reference model back in the 1980s, TCP/IP became standardized into networking devices and remains the framework currently used for communication.

Due to the OSI theoretical model defining more layers, it is often used as a reference within networking contexts. This can be observed, for example, in firewalls that are marketed as layer seven capable. Based on the author's observations in the field, the choice of model when referring to network layers appears to depend on context. For example, when discussing networking devices such as routers and switches, the TCP/IP model is commonly used. In contrast, when addressing issues related to computers and NGFWs, the OSI reference model is applied more frequently. This is likely because the OSI reference model provides more granular layers, allowing analysis of specific aspects of a problem, such as determining whether an issue occurs at the session or presentation layer.

2.2 Macro- vs. Micro-Segmented Networks

Network segmentation is a fundamental networking concept [9] and a security strategy [10], which every network administrator should implement to some extent, such as separating office networks from guest networks. Segmentation is not difficult to implement and significantly increases security, as devices can no longer communicate freely with each other and must instead pass through a firewall or another control mechanism.

To establish the foundation for how and why network segmentation is implemented, the layering of the TCP/IP model must first be examined, specifically layer two. As shown in the previous chapter, the TCP/IP model includes the Internet

layer, which is responsible for adding IP headers to the packets and deciding what routes the packets should take. This layer plays a crucial role in networking, as it enables packets to travel across different devices , and based on header information, directs packet movement within the network.

There is an underlying security issue related to this behavior, as seen in a study by Tyagi and Murugesan [11], where a "flat" network has a Lateral-Movement Susceptibility (LMS) score of near one. If a "flat" network is configured using only a single subnet, all devices are able to communicate with each other without any traffic filtering, aside from possible host-based solutions. Because devices are within the same network, route lookup is performed solely using device Medium Access Control (MAC) addresses, and frames are delivered directly to those addresses. For this reason, it is important to examine an example and consider why network segmentation should be implemented.

Al-Ofeishat and Alshorman authored a study [12] on building secure networks using segmentation and micro-segmentation techniques. As they state, network segmentation can be considered a defensive technique. This is a reasonable assessment, as the attack surface for threats can be significantly reduced by preventing devices from communicating freely within a network. They identified several important segmentation approaches, of which Virtual Local Area Network (VLAN) and firewall-based segmentation are examined in this work.

Assuming a network infrastructure is created using a subnet such as 192.0.2.0/24, and all machines are simply assigned IP addresses and some switches are deployed, every device will be able to communicate with one another. In Figure 2.2, an example is shown in which a building consists of three floors, each populated with different devices. All devices are assigned an IP address from the 192.0.2.0/24 subnet, with the switches only relaying traffic.

Because the devices have addresses within the same subnet, the switches between

the floors can use the devices' MAC addresses to forward traffic through the chain of switches to the relevant devices based on the MAC tables they maintain. From an ease-of-configuration and ease-of-access perspective, this scenario functions as intended. However, from a security perspective, it fails, as devices are able to access resources for which they are not intended to have access.

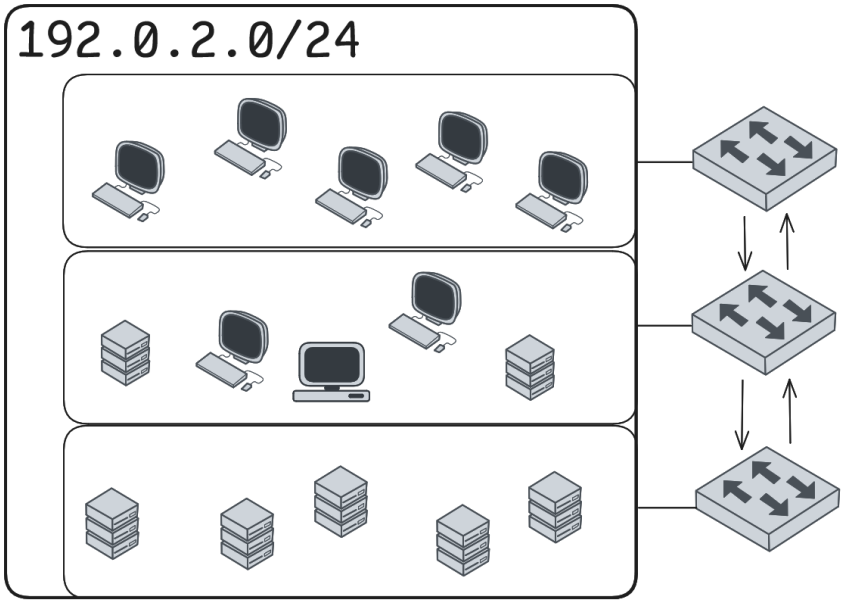

Figure 2.2: Full network without segmentation

To understand how this network could be segmented, it is important to first discuss VLANs and how they operate. VLANs were first introduced in the Institute of Electrical and Electronics Engineers (IEEE) 802.1Q standard in 1998 [13], which vendors and network architects commonly refer to as dot1q. This standard defines VLAN tagging, in which 8021.Q inserts a VLAN tag into the Ethernet frame header. This tag then contains a VLAN identifier (VLAN ID).

By introducing this type of tagging mechanism, it became possible for hosts to exist in the same logical network while physically located behind different switches. Due to the presence of this header, VLAN routing between different switches requires an additional step known as dot1q encapsulation, in which the VLAN tag is added

to the packet headers so that other network devices can determine the correct destination. Additionally, inter-VLAN routing is possible when devices, such as Layer 3 (L3) switches or routers, have interfaces for each VLAN, omitting the requirement for encapsulation. It is important to note that devices in different VLANs cannot communicate with one another unless a layer three capable device, such as a router, layer three capable switch, or a firewall, performs routing between them.

Instead of a single large subnet, consider a hypothetical network in which VLANs are used and the original subnet 192.0.2.0/24 remains in place. In addition, two new networks are introduced: 198.51.100.0/25 and 198.51.100.128/25, as illustrated in Figure 2.3. Following best practice methodologies, these logical networks are also physically distributed to different floors within an office building. In this logical segmentation, the first network is designated for office workers, the second represents a mixed network of different machines, and the third is allocated as the server network.

However, an issue arises when VLAN segmentation is implemented using switches alone. In such cases, machines may still be able to communicate with one another if the destination IP addresses are known and VLANs have been configured on switches with routing capabilities. To prevent this, a firewall, a layer three capable switch, or a router must be configured to block or allow traffic based on Access Control List (ACL) rules. This type of implementation further enhances the security of network segmentation, as also noted by Al-Ofeishat and Alshorman [12].

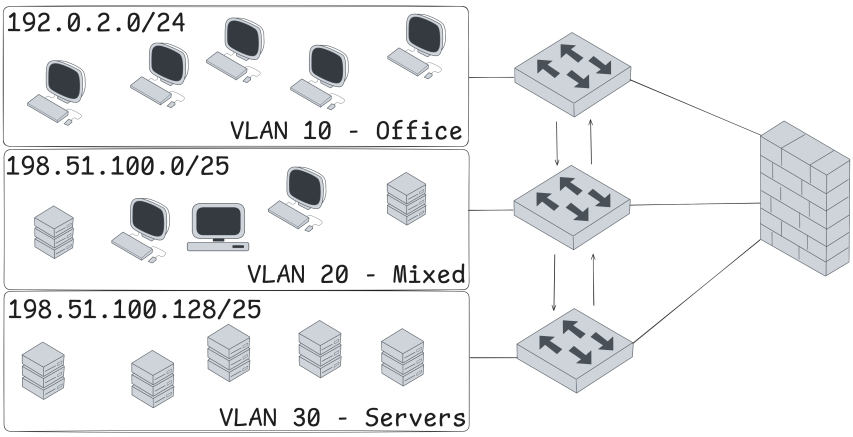

Figure 2.3: Logical network split with switches and a firewall

While network macro-segmentation is a relatively simple concept and is easy to implement in a typical environment, another approach should be considered when a network infrastructure can be classified as critical. This more granular approach to segmentation is micro-segmentation [14]. Rather than creating broad network segments, micro-segmentation divides networks based on individual resources and roles. This does not imply that larger networks, such as office networks, cannot exist. Instead, applications, services and domain-related resources can be micro-segmented into smaller networks.

Micro-segmentation does have a notable drawback. When an organization has an already established infrastructure, migrating to micro-segmentation can be complex and resource intensive [14]. For this reason, it is generally preferable to begin implementing micro-segmentation early in the development of a network infrastructure, which can save time and reduce operational difficulties in the future.

Macro-segmentation can be considered a basic security measure that topologically isolates different parts of an infrastructure. However, this type of approach does not fully mitigate security risks, as threats can still spread widely within the network. Micro-segmentation serves as an effective alternative that can address these remaining security gaps.

2.3 Benefits of Micro- vs. Macro-Segmented Networks

As macro-segmentation and micro-segmentation are both viable approaches within a network, it is reasonable to ask why one should be preferred over the other. As mentioned in the previous chapter, macro-segmentation can be viewed as an improvement over a flat network, but it still presents security concerns related to potential lateral movement. When comparing a micro-segmented network to a macro-segmented network, the most significant difference is already illustrated in Figure 2.4. In a micro-segmented network, assets are isolated into smaller sections, providing a higher level of control and security.

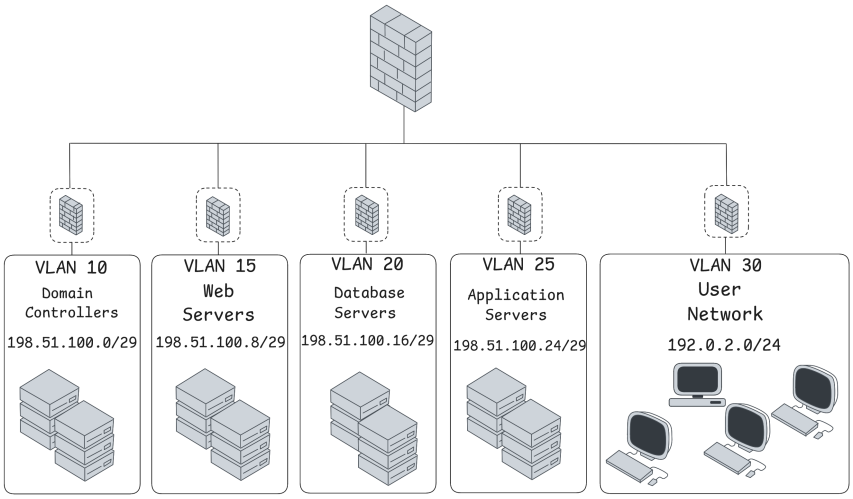

Figure 2.4: Micro-Segmentation in a network

While micro-segmentation can be considered a more complex approach that requires additional time to create granular access rules and deploy necessary VLANs across switches, it offers a significant underlying benefit. This benefit is the containment of possible threats and infections. To illustrate this, two hypothetical situations are examined to compare macro-segmentation and micro-segmentation in terms of threat isolation and to explain why this concept is closely linked to Zero

Trust Architecture (ZTA).

In the first scenario, two large subnets exist within a macro-segmented network: 192.0.2.0/24 and 198.51.100.0/24. The former is reserved for user devices, while the latter is allocated to infrastructure such as web servers, domain controllers and other essential services. If a public-facing web server is compromised due to a misconfiguration, it must be assumed that the attacker can move laterally throughout the entire network. This may result in infection of all other systems within the same network segment, as illustrated in Figure 2.5. In such a scenario, most, if not all, business operations would likely be disrupted.

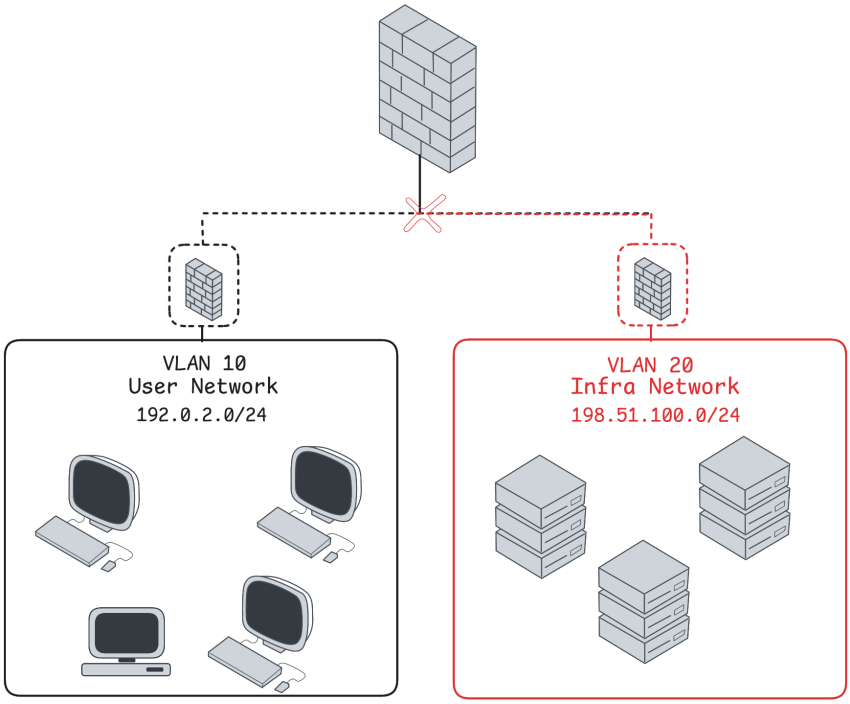

Figure 2.5: Infection in a Macro-Segmented Network

Now, consider a situation similar to the first scenario, but instead within a properly implemented micro-segmented network, where access between resources is restricted based on a need-to-access principle. Assume that a threat is detected within a micro-segmented network dedicated to web applications, as shown in Figure 2.6. In this scenario, it should again be assumed that the entire micro-segment has

been compromised by the threat actor. However, because the network is isolated, the threat cannot easily spread to other parts of the infrastructure.

Lateral movement to other networks is difficult for the attacker [15] because the addresses of segmented networks should not be predictable, meaning that the attacker does not know which resources to target. If there is no visibility to other networks and no configuration exposes connections, the attacker will be unable to access additional resources, thereby effectively isolating the threat. This concept represents a preliminary step toward ZTA, which is discussed later in this work.

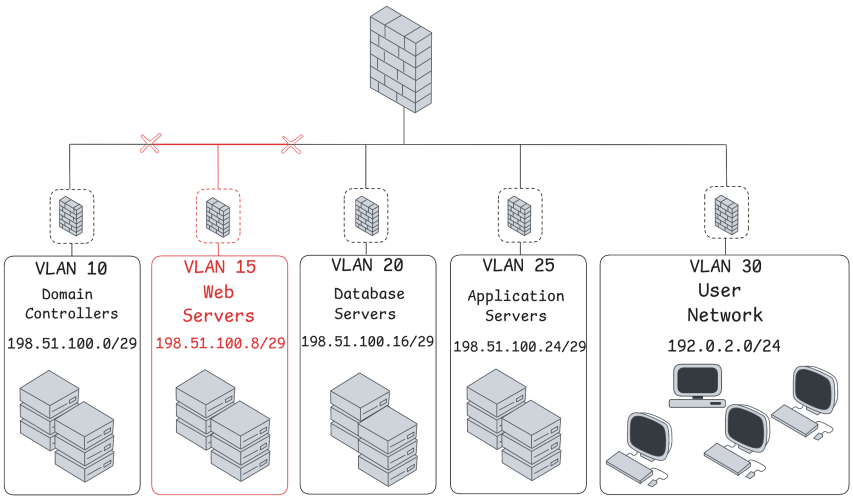

Figure 2.6: Infection in a Micro-Segmented Network

From these two hypothetical scenarios, it becomes clear that the complexity involved in implementing micro-segmentation is justified, as it effectively restricts threat actors from performing lateral movement [14].

2.4 Multi-Layered Network Architecture

Now that network segmentation in its various forms is a clearly defined concept, a related issue can be examined, namely the Internet and, more specifically, incoming traffic from external sources. If an organization offers services that are accessed

from outside the organization, it is important to consider zoning, also referred to as multi-layering, within the network. The number of zones depends on the size of the organization and the types of assets it manages. In larger and more complex organizations, there may be multiple zones, each protected by a firewall cluster.

A De-Militarized Zone (DMZ) typically handles incoming traffic from external sources, followed by an internal zone, with both located behind a firewall cluster. A third zone may be created for critical assets, protected by a separate firewall cluster. A controlled link between these clusters allows necessary traffic to flow between network assets and services.

There are multiple methods to implement the desired zoning architecture, but the objective remains the same: to separate assets within an infrastructure according to their purpose. One common approach is to create topological zones within the infrastructure and then use a firewall to grant the necessary access. For example, in Cisco Adaptive Security Appliance (ASA), security levels can be used to define zones, while Check Point relies on topology and zone definitions when configuring interfaces.

To understand how a multi-layered network architecture benefits an organization, it is important to first examine how it can be implemented. Network layers can be physically separated, configured with firewalls or software-defined, with firewall-based separation being the most common approach.

In their study, Kwon et al. [16] note through a case study that most networks use some form of security classification and perform logical separation based on it, creating zones for the intranet, DMZ, and the Internet. In many organizations, these zones are separated by a single firewall and its internal rule logic.

Assuming a relatively basic organizational network infrastructure with only one firewall or a firewall cluster, the resulting zoning would likely resemble the logical structure illustrated in Figure 2.7. This approach introduces additional security

measures, as traffic rules can be applied between zones. This architectural design aligns with the observations made by Kwon et al. [16]. Although it provides overall security improvements compared to network infrastructures without zoning, it may not be optimal for more complex environments.

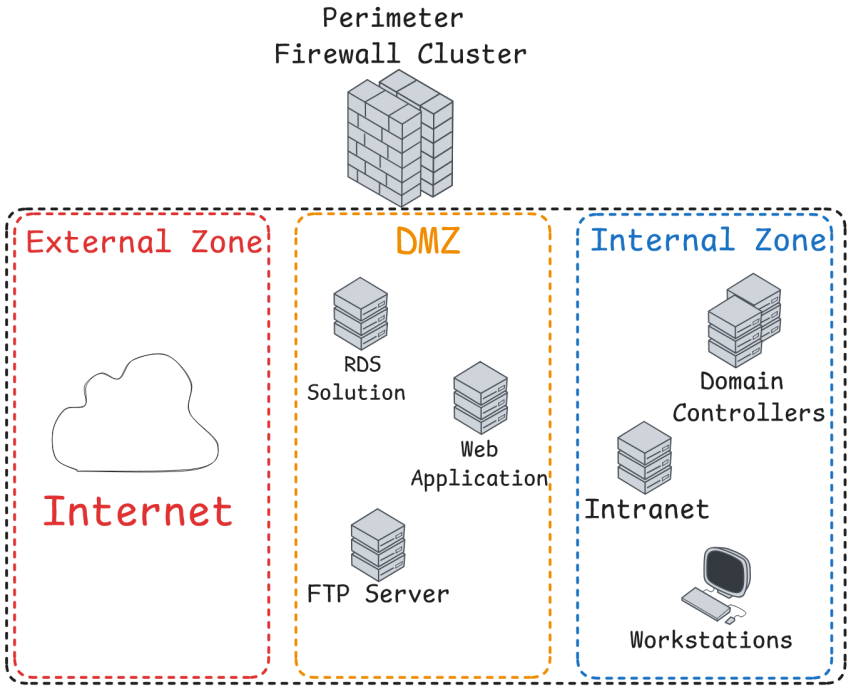

Figure 2.7: Logical zoning in a network infrastructure

In complex environments, it is ideal to design a network architecture in which resources are placed behind specific firewalls, creating an onion-layered structure. Mhaskar et al. [17], in their study, proposed that this design would be feasible and achievable in organizations with smaller networks. As shown in their study, resources have been very granularly divided per firewall.

In modern environments, it may be advisable to physically separate different sections of the network infrastructure using a front-end and a back-end firewall cluster, an elegant solution proposed by Lamdakkar et al. [18]. As noted in their article, NGFW solutions are even more effective, as they can filter threats at layer seven, further enhancing the security posture of the environment.

A possible network architectural diagram, based on articles by Mhaskar et al. [17] and Lamdakkar et al. [18] is shown in Figure 2.8. In this design, zoning is achieved by physically separating infrastructure sections with firewall clusters, and micro-segmentation is also applied within the zones.

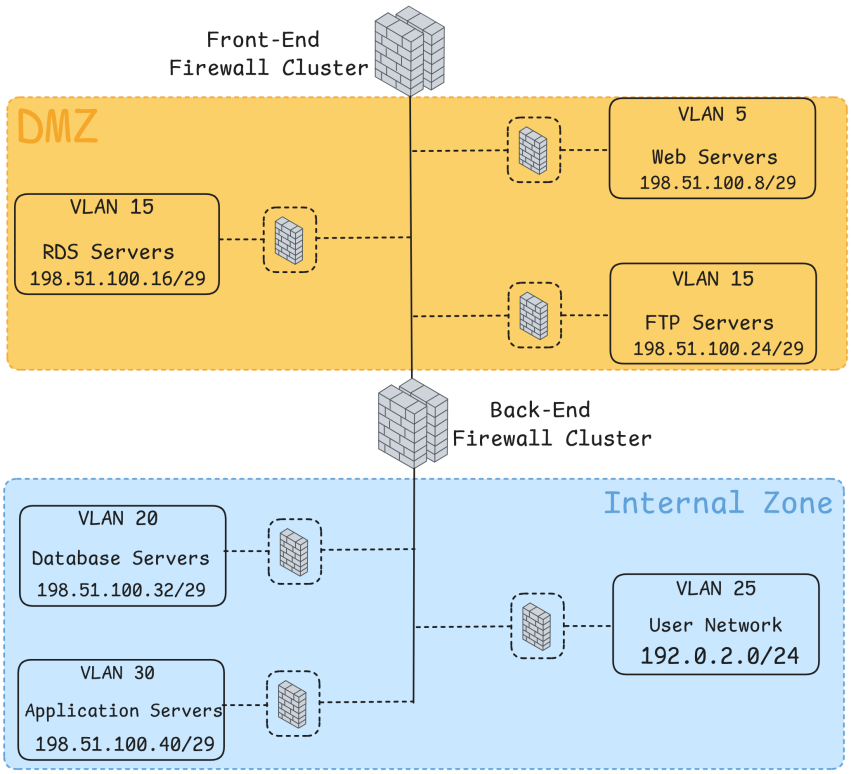

Figure 2.8: Zoning in a Multi-Layered Network Architecture

A relevant question arises: if micro-segmentation, as discussed in a previous chapter, is already implemented, does zoning provide any additional benefits to the environment? One could argue that implementing zoning on top of micro-segmentation allows for broader policies or access rules that control inter-zone traffic, while more granular rules can still be applied within each zone. For example, it is feasible to block traffic from the DMZ to an internal zone by default while permitting specific connections from certain resources through granular rules if they require access to internal assets.

2.5 Definition of an Identity

Defining an identity is a complex task, as it depends on the context in which it is considered. As Anderson suggests [19], the contextual nature of identity makes its meaning controversial. An example is given where an identity may usually be tied to the name of someone, which is not entirely accurate. Anderson proposes that an identity can be understood as a symbolic link between different instances of the same identity that corresponds to a specific principal.

In computing, the concept of identity raises further questions: is an identity a principal to which logon credentials, workstations, emails and other resources are associated, or could a machine itself serve the identity, with the user correlating to the machine? To illustrate this, consider two users, Alice and Bob, each who have a personal workstation in a network environment and also share an additional workstation. It is reasonable to assume that the personal workstations correspond to the individual users, but the shared workstation introduces a more complex scenario.

In this scenario, context appears to determine identity. For example, within an Active Directory (AD) environment, each workstation has its own machine account, creating machine identities, while users maintain separate accounts. Consequently, it can be understood that machine and user accounts represent distinct identities that may correlate with each other.

Based on Anderson's proposal [19] and the hypothetical scenario previously discussed, it can be assumed that when defining a user's identity in computational space, the user acts as the main principal to which accounts are symbolically linked. In the context of identity-based authentication, the identity is most likely represented by the user's credentials. For example, in cloud-based solutions, when a user logs in, the identity they use corresponds to the email or user account associated with that login.

It can be argued that in cyberspace, the identity is effectively the account. Con-

sider a user who has accounts within multiple tenants. When this user switches between accounts, they assume the identity associated with the account in the tenant where they log in. Similarly, on a local workstation where user and administrator accounts are separate, switching between these accounts results in the user assuming the identity of the account currently in use.

From a sociological perspective, a user's identity could be considered as the collection of different accounts that link to them. However, in cyberspace, identity can be defined as described above, further supporting the argument that identity is highly context-dependent. Given this context-based definition, it is essential to understand the relevant context when establishing systems that rely on identities, such as privileged identity management, identity-based access control and identity-based verification systems.

2.6 Zero Trust and Zero Trust for Cybersecurity

Now that identity has been defined to some degree, it is possible to examine a more complex subject, namely zero trust. Individuals working in the IT field have most likely encountered the phrase "Never trust, always verify", which captures the core principle of zero trust. To understand the origins of this phrase and why it has gained popularity, it is important to first provide background information on the formalization of trust. To ensure that the topic is properly understood, it is useful to examine the dissertation "Formalizing trust as a computational concept" by Dr. Marsh [20], in which Dr. Marsh explores the concept of trust and provides clarification on its meaning.

In his dissertation, Dr. Marsh notes that, based on an implication in a definition of trust by Deutsch [21], trust is not objective but rather subjective, and dependent on individual worldviews, or that it can be considered agent-centered. The author agrees with Dr. Marsh as trust is individual-based and can be undermined

or reinforced by the surrounding environment. An example of an environmental effect can be illustrated by considering a Security Operations Center (SOC) analyst performing their duties. In this hypothetical scenario, if the analyst has slept well and consumed their morning coffee, they may have full trust in their capabilities and perform their daily tasks effectively. However, if the analyst has slept for a few hours less and has not had their morning coffee, they may feel tired, which can negatively affect their trust in their own actions. This may cause them to second-guess decisions that would otherwise be made confidently.

Dr. Marsh notes that there are three types of trust: basic, general and situational. In the context of general trust between agents, this relationship can be expressed mathematically using the following notation: Given

$$x, y \in A, Tx(y) \tag{2.1}$$

it is denoted that agent x trusts agent y.

If this value is equal to zero, it indicates that there is no trust from x towards y. Dr. Marsh explains that although zero trust and no trust might appear to be the same, they represent distinct concepts due to differing contextual scenarios. For example, when general trust is mathematically defined as zero, it is interpreted as zero trust rather than an absolute absence of trust.

An example scenario illustrating zero trust occurs when an unknown device joins a neutral network and no trust relationship exists between the device and the network, or vice versa. From a sociological perspective, this is comparable to two individuals meeting for the first time, where no initial trust relationship has yet been established, resulting in a zero trust relationship. Dr. Marsh also discusses distrust and explains how it differs from zero trust, since distrust is based on prior judgment or experience.

In 2010, Kindervag et al. [1] built upon earlier research and introduced the zero

trust model for information security, which continues to evolve and is widely used today. The paper highlights several pitfalls that reveal important scenarios and help explain why zero trust is a strong and effective model in information security.

Starting with the first pitfall, Kindervag et al. note that there is an assumption in the field that security professionals can recognize and determine which interfaces are trusted and which are untrusted. This is illustrated by the question of where one should connect to the Internet. This design has been embedded into networking hardware for a long time, and the author agrees that it represents a significant issue that must be addressed, not necessarily at a physical level, but rather through logical controls.

It is proposed that instead of assuming that something is trusted or untrusted, the network should always be treated as untrusted, thereby requiring verification for all resources. Consider a scenario in which a standard internal employee network is used in an office on a daily basis. In the traditional model, this network might be considered trusted because it contains trusted personnel. However, this assumption does not account for the possibility that a malicious actor could breach the network and gain implicit trust.

The second pitfall is related to an older, now revised mantra of "trust but verify". It is often joked that not many people verify, even when they should, leading to implicit trust of various parties. This earlier mantra has since evolved into "never trust, always verify", which is more closely aligned with the zero-trust principle and results in improved protection within organizational environments.

The third pitfall is related to the previously discussed scenario in which malicious actors gain trust after infiltrating a network that is assumed to be trusted. It is also associated with trusted personnel performing malicious activity by exploiting the previously mentioned "trust but verify" mantra. As noted, corporate users are often trusted by default, which can lead to scenarios in which malicious activity occurs

without being detected.

The final pitfall relates to network packets and the fact that trust is not inherently associated with them. As discussed in a previous subchapter, identity was defined to be context-based. In this context, it is emphasized that network identity can only be inferred by examining packets moving through the network. If packets do not contain information that identifies the user, it becomes impossible to determine who is performing specific activities. This limitation further highlights the importance of the second pitfall, namely the inability to reliably identify who is present within the network.

In the newly proposed zero trust model, it is recommended to perform network-level analysis on both internal and external traffic and to treat all networks as untrusted. This approach provides greater visibility into network activity and allows different systems to verify users within the networks, thereby establishing a stronger security posture. As a concluding note it was mentioned that "zero trust is not a one-time project", a claim that can be supported, as effective implementation requires time, resources and continuous management to ensure that security professionals remain ahead of malicious activity.

Essentially, the zero trust model can be viewed as a framework in which every device, user, and resource within a network is verified and no entity is trusted by default. This approach helps security professionals evaluate networks from a different perspective and compels them to secure each network as if it were already compromised, including the devices within it. As a result, stronger internal security practices can be achieved overall.

2.7 Zero Trust Architecture

Rose et al. state in their study [22] that zero trust, by definition, assumes that no entity is trusted based on its physical or logical location within an environment.

This aligns with the observations of Dr. Marsh [20], where in a hypothetical scenario an unknown device joins a network and there is no trust relationship between the device and the network, or vice versa. These statements appear to be consistent with sociological behavior, in which an individual would not trust an unknown person before a trust relationship has been established.

This concept of a trust relationship can also be observed in computing environments through various authentication mechanisms. For example, a firewall may connect to a central management system if they both use the same assigned key for a secure handshake, thereby establishing a mutual trust relationship.

Rose et al. [22] further noted that, in order to achieve ZTA, several assumptions must be made from a zero-trust perspective. These assumptions then serve as a foundation upon which ZTA can be built. According to their study, there are seven major assumptions that must be established:

1. All data sources and computing services must be classified as resources.

2. Communication in any direction must always be secured, even within networks that are otherwise assumed to be trusted.

3. Individual access to any organizational resource must be granted only on a per-session basis.

4. Access to resources is dynamic. To gain access, different authentication-related information is evaluated, including the asset requesting the asset, the state of the current identity, and the application or service making the request. Additional environmental checks may also be performed, such as the geolocation of the user or detection of simultaneous attempts to access multiple files within a short period of time.

5. All assets, whether directly owned or indirectly related to the organization, must have their security posture and integrity continuously monitored.

6. Access is never granted unless dynamic authentication and authorization methods are successfully completed.

7. The organization must collect data regarding the assets, network infrastructure, and relevant communications within the network, whether directly or indirectly related to the organization. This data is then used to enhance the overall security posture of the organization.

In addition to the foundational concepts of zero trust discussed above, it is important to introduce certain fundamental principles of zero trust from a networking perspective. This ensures that all the necessary basics are covered for a comprehensive understanding of ZTA.

1. An organizational network cannot be assumed to be a trusted environment, as an adversary could already be present within the network performing malicious activities.

2. There may be instances where assets within the network are not managed by the organization. Even if security policies prohibit the connection of personal devices, employees may still connect their own devices for convenience, creating potential security risks.

3. Assets cannot be inherently trusted. Devices and their installed software may contain vulnerabilities unknown to the organization, highlighting the importance of conducting regular vulnerability assessments and security posture checks before granting access to organizational resources.

4. Organizational assets may be partially owned or controlled by third parties. For example, in Domain Name Resolution (DNS), organizational devices may rely on both the internal DNS server and public DNS infrastructure.

5. Personnel working remotely cannot assume that their local network is secure. It should always be assumed that the local network may be monitored or compromised by a hostile adversary. Consequently, all communication involving organizational resources should be implemented at the highest security level possible.

6. Any data or communication moving between organizational and non-organizational infrastructure must adhere to a consistent and robust security policy. All assets should maintain their security posture at all times to protect organizational data.

One of the fundamental principles of zero trust states that an organizational network cannot be assumed to be a trusted environment, a factor that must be considered when developing network infrastructure. ZTA can be established through micro-segmentation, as discussed in a previous chapter.

As noted in a research article by Phiayura and Teerakanok [23], a challenge in migrating to ZTA is the presence of legacy systems, since the migration process to ZTA could create defects or disruptions in these systems. To mitigate this risk, a comprehensive planning phase should be conducted, in which the potential quirks of older systems and the access between these resources are carefully analyzed and appropriately integrated into the ZTA design.

Although zero trust as a concept offers many benefits, He et al. created a survey-based study [24] to investigate the potential challenges that organizations may encounter. They provide an example related to identity authentication, noting that Single-Factor Authentication (SFA) fails if a password is stolen. MFA improves security, but a risk remains if continuous authentication is not enforced, as users may retain access for extended periods. This challenge is particularly relevant in scenarios where tokens are used, as stolen tokens grant an attacker access to critical resources until they are invalidated. For example, in Microsoft 365, if an initial

token and a refresh token are compromised, it is possible to repeatedly refresh the token for up to 90 days by default until it is revoked.

Continuous authentication introduces another concern that is closely related to both privacy and security. If users' actions are continuously monitored by a system to enforce access controls, questions arise regarding the boundary between personal privacy and organizational security requirements. Baig and Eskeland [25] highlight potential privacy issues associated with context-aware continuous authentication, which relies on personalized data that can be linked to individual users and monitored over time. If an unauthorized party gains access to this data, it becomes possible to identify users. Su et al. [26] demonstrate this risk, showing that more than 70% of Twitter users could be identified using only their browsing history.

2.8 Zero Trust Network Access

When discussing zero trust in the network, one key issue concerns access to resources within the network, specifically, any device on the network may be able to access a given resource if the firewall, router or other intermediate device allows the traffic to pass. This creates a security risk, as malicious actors could exploit this implicit trust to move laterally within the network or gain unauthorized access to sensitive resources, as discussed in previous chapters.

Consider a hypothetical scenario where Alice and Bob are located in an office network and a firewall permits access from the office network into the internal network. Although this setup functions correctly for Alice and Bob, as illustrated in Figure 2.9, if Charlie also connects to the network, they may similarly gain access to internal resources, highlighting the potential security risk.

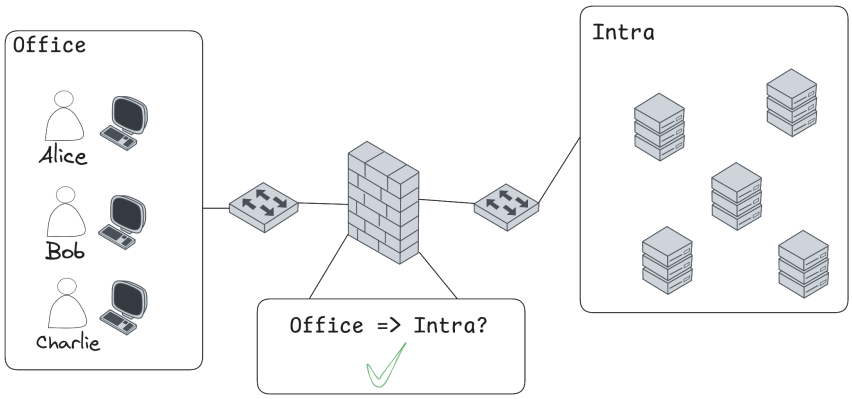

Figure 2.9: Access from an office network to internal network

He et al. state [24] that the "current zero trust access control model should not be limited to a certain access control model", which is a well-articulated observation on current technologies. Zero Trust Network Access (ZTNA) can be understood as a concept in which identity plays a central role, as noted by Fernandez and Brazhuk [27]. While identity verification is a core component of ZTNA, other factors such as location, user behavior and device posture can also be incorporated to enhance access decisions.

Mavroudis [28] highlights that although ZTNA represents a positive step forward in improving security by mitigating well-known risks, organizations may face several challenges when implementing it. Four notable issues have been identified, beginning with possible performance concerns.

While performance issues may arise due to the continuous authentication of users, it could be argued that the extent of the impact depends on the approach used for this authentication. In a scenario where a small number of endpoints handle all authentication and assessment, performance bottlenecks are likely. However, if a more distributed system is implemented where authentication occurs on multiple servers or directly on access points, these issues can be significantly minimized.

Another challenge is complexity. As policies become more complex, administra-

tive and operational burdens on different teams can increase. This challenge can be mitigated by establishing a well-defined framework for policy creation and consistently applying it to identity-based access control. Scalability concerns are closely related, as without a proper framework and consistent implementation, organizations may fail to enforce controls across all relevant endpoints, resulting in security gaps.

Finally, an issue concerning legacy systems is highlighted. Some firmware and systems may not meet the compatibility requirements for integration with modern solutions, requiring infrastructure teams to redesign or re-implement these systems. Without such updates, gaps in the ZTNA logic could emerge, potentially creating significant security vulnerabilities.

To address the underlying issue of malicious actors accessing resources without verification, an identity-based check can be introduced at the firewall level within the network. By performing identity verification, the firewall can restrict traffic to designated users or groups, as illustrated in Figure 2.10. This approach mitigates the potential security risk posed by unauthorized actors within the network infrastructure, at least to some extent.

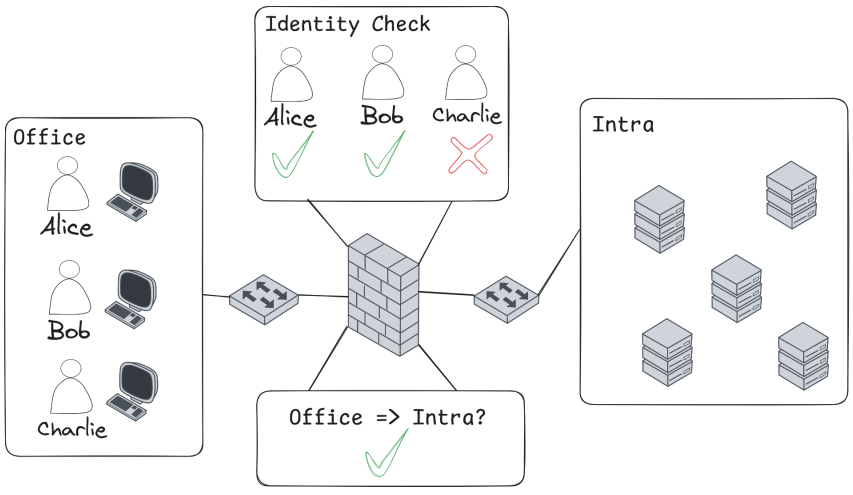

Figure 2.10: Identity Check when accessing internal network from office network

2.9 Identity Management

To ensure that the previously discussed identity-based access control functions effectively, identity management is a critical component. There must be a source-of-truth service that maintains an up-to-date record of all identities. For example, in an on-premises environment, this could be implemented using Active Directory or a Lightweight Directory Access Protocol (LDAP) system, such as OpenLDAP, providing a directory that can be queried by firewall solutions. Alternatively, a cloud-based identity provider, such as Entra ID, could serve this purpose.

Dabrowski and Pacyna [29] define Identity Management (IdM) as a system responsible for managing identities within computational systems. They note that multiple IdM approaches exist, beginning with isolated identity management, in which according to Cao and Yang [30], the Service Provider (SP) also functions as the identity provider, storing all identities and performing the necessary operations in the same system. A key distinction related to other approaches is that all credentials are completely separate and cannot be used across different SPs.

The isolated IdM previously mentioned can also be compared to personal authentication management, as illustrated by Dabrowski and Pacyna [29]. In this model, the SPs continue to act as identifier providers, but the requester, or user, is responsible for storing their identifier and credentials.

The second type of IdM as described by Cao and Yang [30], is the centralized model. In this model, multiple service providers exist, but identity and credential information is stored on a centralized server. An example of a centralized IdM system is AD, where all identities are maintained in domain controllers, which are also used for identity verification. Different systems can then interact with the centralized IdM using protocols such as Kerberos, in which the SP forwards the authentication request to the central server.

Finally, there is the federated IdM, which is a more complex approach. Federa-

tion can be implemented, for example, using AD through Active Directory Federation Services (AD FS), which provides Single Sign-On (SSO) capabilities. This can be applied to various web applications or other systems that support protocols such as OpenID Connect (OIDC) to authenticate against the AD FS endpoint. Federated IdM is considered the most flexible approach, as a user maintains an identity that can be used globally across domains or organizations. Another example of federation is when a user with an Entra ID account accesses resources in a different Entra tenant through a Business-to-Business (B2B) scenario.

A new type of identity management model, called Self-Sovereign Identity (SSI), is currently emerging in cyberspace. As described by Čučko et al. [31], in this approach, users or other entities have full control over their digital identities, including credentials, privacy settings, and other related data. Essentially, the entity acts as its own identity manager and can present verifiable proof of identity when requested.

Given the variety of IdM approaches, organizations must choose the most suitable based on their specific needs. For example, An organization with an on-premises environment and multiple applications could adopt a federated approach to enable SSO logins across different services, thus improving both security and user experience [32].

3 General and Specific Designs

In order to address the lack of certainty identified earlier with zero-trust-based access control in complex on-premises infrastructures and provide a solution, a case study will be performed for an organization. This case study examines an organization with an existing complex network infrastructure, consisting of multiple layers separated by physical firewall clusters. Zero-trust architecture is already partially implemented in the form of micro-segmentation, where networks are divided based on resources and firewall rules are granularly defined. Only the necessary ports are open between resources, while all other traffic is dropped.

In addition, employees and third parties access resources within the environment through the local network or through various remote access solutions. While all connections function correctly and users can access the resources permitted by the firewall, an underlying issue was identified: anyone within certain networks can access defined resources if they either authenticate through the remote services or gain physical access to a network with granted permissions.

Given the rapidly evolving cybersecurity landscape and the constant update of standards and directives, the company aims to resolve this issue by restricting access to only the necessary end-users or groups defined by the organization. Implementing this solution extends the existing zero-trust architecture, adding a zero-trust-based access control layer on top of the already established micro-segmented and multi-layered network design.

3.1 Methodology

This research was concerned with three distinct research questions. The main aims of the research were to examine whether access control based on identities is effective by using identities as a testing method to show that it is effective, if challenges appear during implementation, which could indicate that the implementation is difficult, and whether the implementation is plausible in a network infrastructure consisting of multiple different layers and micro-segments, which could be classified as complex.

The research approach was technical in nature, as the research questions required a technical perspective to be answered through a customized design. A case study method was chosen because it closely reflects a real-world organizational infrastructure, including potential complications such as legacy devices, a diverse user base, different types of network devices, and the presence of third-party users.

By conducting the case study through the implementation of a technical design, it was possible to obtain realistic results and address the research questions. Due to the technical design, the research focuses on functional results rather than on data analysis.

The organization in which the case study was implemented had an existing, fairly complex network infrastructure with multiple layers and micro-segmentation within these layers, achieved using NGFWs from Check Point. Existing virtualization platforms hosted different servers and systems that were used by individuals within the organization and by users outside the organization. However, from a research perspective, the specific vendors were not relevant. In addition, there were local networks to which individuals from different organizations connected when they were on-premise.

In this case study, the issue that the design aimed to address was related to how users access environments across different scenarios and through different methods, making it a suitable candidate for technical design implementation.

To enable initial testing within the environment, selected networks were used and proprietary software was installed on some servers within these networks. After the design was implemented through different firewall and system configurations, it was possible to use accounts created for testing purposes to validate that the design was functioning as intended. Once the initial tests were validated, the testing was scaled to other live systems for specific users, and the design was confirmed to function as expected.

The expected outcome for the tests was that zero-trust-based access control would function correctly even when traffic moved multiple layers to and from different micro-segments. A failure case would have been observed if the solution prevented users from accessing specified resources or allowed unrestricted access to resources despite the configured limitations. A successful implementation would permit access only for specified users to defined resources, while blocking all others.

There were two ways to validate that the design was working. The first approach was to monitor the firewall logs using different query filters. The second approach was to attempt to connect to a resource using an identity with specified access and, after a successful connection, to repeat the attempt using a different identity to verify that access was blocked. If both validation methods were successful, the implementation could be considered effective.

3.2 General Design

The general design to solve this underlying issue is to use firewall products that are capable of identity tracing technologies. As the company in the case study uses Check Point, the design in the case study will be implemented with Check Point products in mind. However, the same design concept also applies to other vendors.

The main idea to solve the issue is theoretically simple, but may require some medium-complexity configuration across different systems. First, it is possible to

examine a general design that addresses this issue in a product- or vendor-neutral manner by using terminology instead of direct product names. From this, the design for the case study is derived in order to allow a proper understanding of the architectural design and to provide the foundation for the solution used in this thesis.

The following logic is based on assumptions about the existing environment. It assumes that there is a firewall solution with role-based access control capabilities, a method for delivering identity information to the firewall solution, and finally, an identity management system or service that stores account and related group information.

In Figure 3.1, it can be seen that the user first authenticates on their computer using credentials managed by a service, cloud-based or on-premise. After this authentication, depending on the client solution, the firewall retrieves the identity of the user through a client, the user's computer runs a client that informs the firewall of its identity, or a terminal server in a remote access solution informs the firewall of the user's identity through a client configuration.

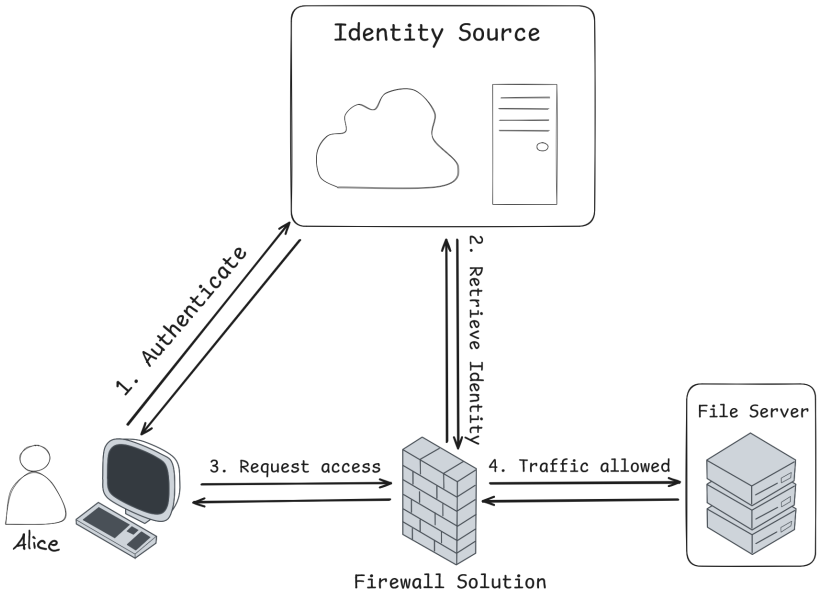

Figure 3.1: Identity-based access-control flow

Depending on whether the firewall retrieves the information or a client continuously sends the information, the firewall either binds the identity to the IP address of the computer, or a client attaches identity information to the packets going to the firewall, where the information is then stripped and verified.

After the authentication has been completed successfully and the identity of the user has either been bound to the IP address or being constantly sent, the firewall solution contacts an identity management solution to retrieve information about the groups to which the user may belong. The group information is then stored alongside the identity.

Consider a hypothetical scenario where a user attempts to access, for example, a server located in the development network from an office network. When the designated packets arrive at the firewall solution, the firewall will match the user's IP address to its table holding identities, retrieve any defined groups and match the identity alongside its group to the rule that allows traffic towards that development service. If the identity matches a configured role-based access control object or rule, access is granted and the session is maintained for that identity through stateful tables.

This design is one of the core pillars in ZTA. If the journey towards zero trust has not yet been started in an organization, this approach is theoretically a fairly easy solution to implement, and it adds an additional layer of security on top of the OSI-model defined layers in a firewall solution.

3.3 Design for External Users

Now that the generalized version has been discussed and established, it is time to examine how this model can be implemented in the case study presented in this thesis. As the company is currently utilizing products and hardware from Check Point, the generalized design must be adapted and mapped to the solutions

offered by Check Point, while also taking into account the current firewall solution implementation. As the network in the case study is moderately complex, there are two Check Point security gateway clusters operating on separate network layers, creating a physically multi-layered environment. The modified core design remains the same, but the scenarios can be viewed as modular components that can be integrated into the core solution to achieve the desired identity tracing functionality within the network infrastructure.

The first scenario, illustrated in Figure 3.2, involves remote services that are accessed by third parties or internal users over an external network. These remote services then direct the necessary traffic internally. For example, a user may access a remote service and, through the remote service, access a browser-based solution located within the inner layer of the network clusters. To apply role-based access control, it is necessary that the first firewall cluster is able to identify the user and then share that identity with the other firewall cluster.

In this environment, remote services host sessions from multiple users at the same time, which makes it necessary to associate identities with each of these users individually. This represents a special-case scenario, as compared to the general design, since there is a conflict caused by multiple users appearing to originate from the same IP address when connecting through remote services.

To resolve this issue in this environment, Check Point's Multi-User Host (MUH) client is used. First, the client creates a random ID range for the users and informs the firewall cluster of this new range upon each logon. This client then intercepts network traffic locally before sending it to the network and tags the traffic with the randomly selected ID range. The firewall cluster can then interpret this information, identify which user is attempting to access resources, and allow traffic accordingly.

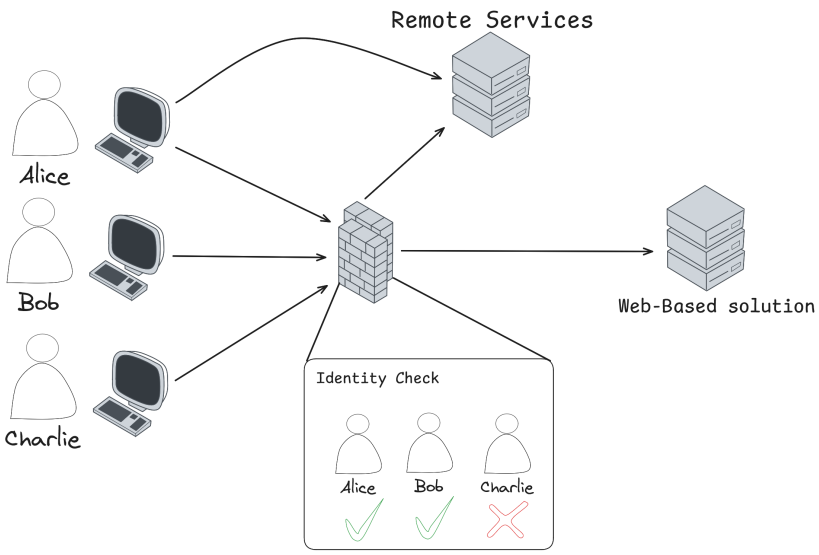

Figure 3.2: Flow of identity verification in remote services scenario

3.4 Design for Internal Users

The second scenario involves users within the internal network, where two possible solutions can be used and both will be implemented. An option is to install a client on the users' devices that connects to the firewall and continuously informs it of the logged-on user, which the firewall then associates with an IP address.

The second solution, as shown in Figure 3.3, involves an identity collector service that runs on its own server. In this configuration, whenever a user connects to the environment, whether to a server or a workstation, the identity collector reads the security logs from a domain controller and detects new logon events. These logons are then associated with an IP address as recorded in the event logs. After the data have been collected, the information is relayed through a secure channel encrypted with a Pre-Shared Key (PSK) to the firewall solution. The firewall then begins monitoring the identity and allows access according to configured policies based on the access roles associated with that identity.

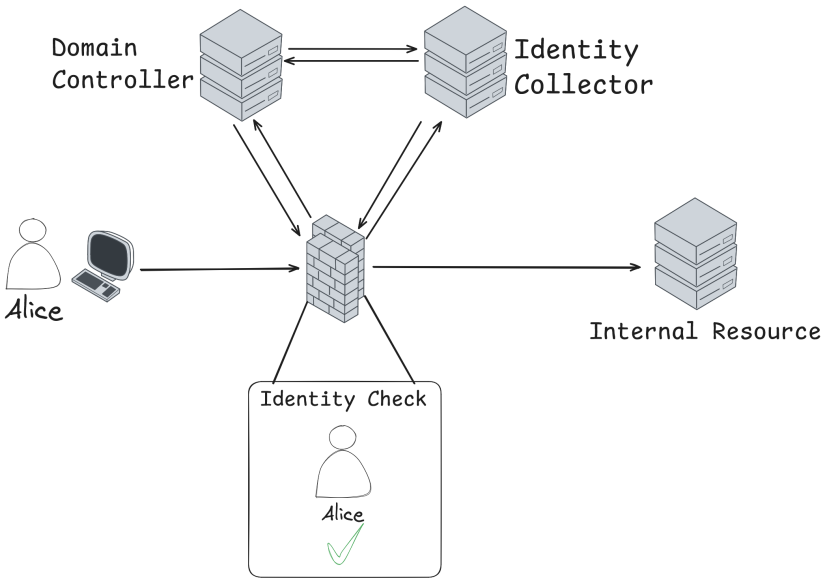

Figure 3.3: Identity tracing with identity collector

3.5 Design for VPN Users

The third scenario is shown in Figure 3.4, which currently presents certain challenges. As Check Point's remote access Virtual Private Network (VPN) does not support VPN profiles in the same way as solutions such as Cisco ASA, creating a secure method for different organizations to connect and access only necessary resources is not possible in the traditional sense.

However, the remote access client functions as an identity awareness client that transfers the logged-on user's identity to the firewall cluster. This makes it possible for third-party users who are part of the domain to be assigned access roles based on their group memberships, according to the defined design. As a result, both third-party identities and internal identities can be configured to access only specific resources when connecting through the VPN. By implementing this approach, the risk that users gain access to unauthorized resources can be significantly reduced.

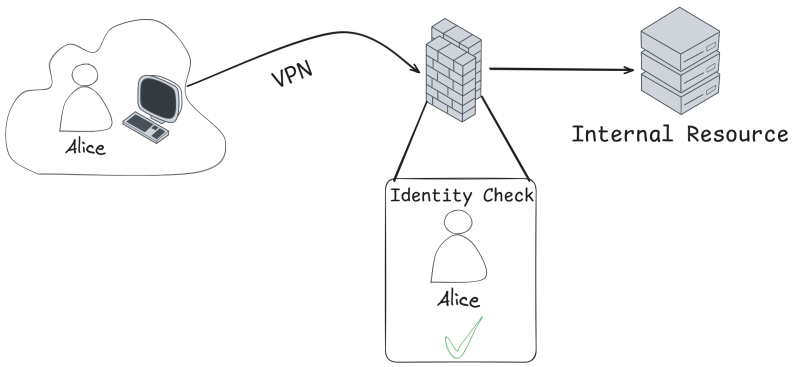

Figure 3.4: Identity tracing for VPN users

3.6 Outcome and privacy concerns

After the designs have been created for the specific scenarios and tailored to the use cases of the company, it is possible to construct a broader design based on all three scenarios. This is achieved by implementing all three designs simultaneously, creating a solid and effective solution to address identity-based access challenges within the network.

As discussed in the literature review, several studies have raised privacy concerns related to continuous authentication. However, this approach mitigates such concerns to some extent, as user behavior is not actively monitored. Instead, access is granted solely on the basis of predefined access roles associated with each user. Some may argue that linking a user to an IP address and granting access accordingly constitutes a breach of privacy. However, if this were the case, even basic mechanisms such as leasing a Dynamic Host Configuration Protocol (DHCP) address to a device would also be considered a privacy violation, since the address becomes associated with a specific device and can be traced through domain controller security logs to identify the user who obtained it. This design can therefore be viewed in a similar manner, as identities are already linked to devices through IP addresses.

4 Implementation

After the different scenarios were discussed with the customer, it was possible to start examining the pre-requisites for the implementation and configuring anything necessary. This process started by creating the required user account in the on-premise AD domain so that an LDAP account unit could be created in the Check Point's management environment for user and group retrieval from the AD. This LDAP account unit acts as a relay for the source of truth, which in this case is the AD. After obtaining the necessary linkage, it was possible to configure the required settings for different client endpoints. Following the initial configuration, the clients were installed and connected to the front firewall cluster. To ensure that identities could work seamlessly between the different network layers, identity sharing was enabled in firewall clusters. Finally, access roles were created for the required groups and users and rules were defined using those access roles.

4.1 Active Directory Linkage with LDAP Account Unit

To get started with the project, an LDAP account unit has to be created in SmartConsole, from which the Security Management Server (SMS) and firewalls will query user group information. For secure communication with the LDAP server, it is possible to perform certificate-based authentication. For this to work, a trusted

Certification Authority (CA) is required, which can be added through the object explorer.

There is a specific requirement for the user account that is used for the LDAP queries, and this requirement depends on whether AD query is used in the environment or only the Identity Collector. In the latter case, it is sufficient to configure a basic domain user that is used to read LDAP attributes of users logging in to the systems.

If there is a scenario where, for example, users are required to change their passwords from the remote access VPN client, a user with significantly higher permission levels in the AD environment is required. This account is then configured in the LDAP Account Unit.

The LDAP AU is relatively simple to configure. First, the required server object is defined, after which the credentials are entered, with the username being the full-length domain object name. Once these parameters have been configured, it is possible to use SSL binding. For SSL binding to function, the target server must have a certificate created by a trusted CA, where the Common Name (CN) corresponds to the Fully Qualified Domain Name (FQDN) of the target server. This certificate must then be stored in the personal certificate store on the local machine.

4.2 Identity Awareness Settings Configuration

Due to multiple different scenarios with different authentication options in this environment, the identity awareness settings on the firewall clusters must be configured appropriately. The configuration is always performed per identity source that needs to be defined. For example, when using the MUH client, a PSK must be configured in the settings and later used in the client configuration when it is installed on MUH servers. Remote access VPN, as an identity source, is configured by simply checking the corresponding checkbox.

When it comes to identity agents for endpoints, the configuration has to include where the agents can connect from, which type of authentication method they use, and specific session-related settings such as the re-authentication time period and whether the agent should be upgraded.

With the Identity Collector, the process is similar to that of the identity agent. It must be defined where the collector is allowed to connect from, but additionally, authorized clients have to be configured. For these clients, a PSK has to be generated, which will be used when setting up the Identity Collector itself. In addition, it must be configured where the gateway retrieves user information from when the Identity Collector notifies it of a logged-in user. In this case, the source is the previously created LDAP Account Unit.

4.3 Identity Awareness – Multi-User Host Client

In this environment, there are servers that host multiple users simultaneously, as they are terminal servers running technologies such as Citrix and Remote Desktop Services. It is important to protect traffic from these servers by implementing identity-based access rules. If a breach were to happen and an attacker gained access to the server level, they would have access to all resources published to different user groups. Identity-based rules restrict a malicious actor to only the resources that the compromised identity would normally be permitted to access.

There are two MUH client versions available, version one (v1) and version two (v2), and there is a major difference in how these clients operate. Version one uses a Transport Driver Interface (TDI) driver to intercept requests from processes that are attempting to establish new connections and then sends a query to the terminal server to determine which user is initiating the connection. After this, a port is selected from the generated ranges for allocation, after which the MUH v1 client informs the firewall of how it is currently controlling the connection. The firewall

then uses the source port to identify which user is behind the connection and allows it accordingly. This is an older client and has several limitations. First, it supports a maximum of 20 clients per session and does not support windows secure boot. The lack of secure boot support alone is sufficient for this version to be deprecated in favor of version 2, but the characteristics of that version are also examined below.

The MUH version 2 client instead uses a Windows Filtering Platform (WFP) driver and intercepts all traffic originating from any user. Whenever a user connects to a terminal server with the client installed, an integer range is assigned to that user. This information is then relayed to the firewall defined in the client settings. When a user initiates traffic, the WFP driver adds its own tag to the packet. When the packet reaches the firewall, it is stripped and the firewall searches for the identifier in the packet and matches it against known identifier ranges. This process maps the user to an identity and grants access based on the configured identity-based access rules. The MUH v2 client supports a maximum of 256 simultaneous users and additionally supports Windows secure boot. The only downside is that it does not add identifier ranges to local accounts unless Kerberos SSO authentication is configured.

This environment also includes a special scenario in which users log in with email addresses that do not belong to the domain. In this case, the client is able to correctly identify the user, but the firewall is not able to interpret this information. Due to this limitation, a registry change is required to use the User Principal Name (UPN) of the user and not the SAMAccountName attribute. To address this issue, a registry key named “ResolveFQDN” was created and configured.

The client is easy to configure. After the installer has ran, the necessary drivers are installed on the system, and the client becomes almost ready for use. To finalize the installation, the client must be connected to the firewalls using a PSK created in the earlier chapter, Identity Awareness Settings Configuration, and the required

registry key must be modified as described earlier to enable UPN recognition. There is also an additional parameter that hides the User Interface (UI) of the tool from the toolbar, although the configuration tool can still be opened on the Operating System (OS) level from the Windows menu.

4.4 Identity Awareness – Identity Agent for Endpoints

An identity agent for endpoints is also fairly easy to configure. After following the installer instructions, it is possible to leave the agent in auto-discovery mode, where it attempts to identify an existing identity awareness gateway in the environment and connect to it. Alternatively, the gateway can be defined directly in the settings.

When a user logs on to a workstation where this agent is installed, they are able to authenticate against AD. Alternatively, if SSO has been configured with Kerberos, the agent authenticates directly with the gateway. The firewall cluster then maps the IP address to the identity provided by the agent. The agent then acts as an identity source and the firewall cluster tracks that identity, mapping relevant access roles based on group information obtained from the identity provider.

4.5 Identity Awareness – Remote Access VPN

There are multiple different ways to connect to Check Point's remote access VPN services, including the remote access client, capsule connect, capsule workspace, Secure Sockets Layer (SSL) VPN portal, and IPsec VPN. In this environment, only the remote access client for workstations and capsule connect are used.

When a user authenticates using one of these clients, they first connect to the firewall cluster to obtain the current configuration profile and the available authenti-

cation methods. Depending on the selected authentication method, the authentication is performed against, for example, Entra ID as an identity provider service or a local active directory domain through Remote Authentication Dial In User Service (RADIUS). After successful authentication, the authentication endpoint forwards the authentication details to the firewall cluster for further evaluation.

After the authentication flow has been completed, the firewall effectively logs the user into the system and begins tracking the user identity. At the same time, it performs a check against the identity provider and retrieves the user's current group memberships based on which the user is assigned the relevant access roles.

4.6 Identity Collector

The identity collector is a type of software that can be installed on any server. It is configured to monitor logon activities in the AD by connecting to the Domain Controllers (DCs) and reading the security logs on those controllers. To achieve the required permission levels for the identity collector, a user account can be created with the event log readers role in the domain. This approach makes it possible to avoid granting higher permission levels.

In this implementation, the identity collector was deployed on a dedicated server as per the micro-segmentation requirements, providing proper isolation. Within the identity collector, a domain with the necessary filtering was configured using a specifically created service account, enabling it to connect to the domain and read relevant logs.

When users log on to systems that are connected to the Active Directory, the identity collector reads event logs and, based on these, identifies which users have connected to the environment and from which IP addresses. This information is then relayed to the firewall cluster. The downside of this method is that if multiple users log on from the same IP address, the identity collector solution is not able to

track all of them, as there is no unique identifier for the users. For this reason, an alternative solution for multi-user hosts is used in the environment to address this limitation as described in an earlier chapter.

4.7 Identity Sharing

As the environment is multi-layered and consists of different infrastructure security levels, there are resources across all network layers that different users and servers need to access. This raises the question of how identity-based access rules can be enforced on firewall clusters at multiple layers if all clients connect to only one of the clusters. Check Point provides identity sharing functionality through which both clusters exchange information about discovered identities in the network based on their identity awareness configuration and relay this information to one another.

For example, if a user uses remote access VPN to reach resources behind another firewall cluster, the remote access VPN client informs the connected cluster of the user's identity. That firewall cluster then forwards the identity information to the second firewall cluster, which can use it to perform access filtering. As a result, if the user needs to access resources behind the second network layer, this is possible because the second cluster has identity information and can associate it with a known remote access IP address.

4.8 Access Roles - Designing

As the LDAP Account Unit (AU) has been configured, it is possible to query users and groups from the domain. Creating access roles itself is fairly straightforward, as the only requirement is an active LDAP or LDAP over SSL (LDAPS) connection to the domain so that the necessary users and groups can be retrieved and assigned to the access role being created.

In this environment, and most likely in most other environments, the challenging part is to actually create valid designs for different user groups. When designing these groups, it is necessary to consider potential third parties, group purposes, and account types. For example, in a Remote Desktop Services (RDS) deployment, multiple third parties may use applications published through these services. As explained in earlier chapters, if an attacker were able to breach the environment due to a misconfiguration or a vulnerability, they could gain access to all resources available to each third party through the session host. To mitigate this risk, access role groups were designed in a way that accounts for multiple factors, resulting in a more streamlined and secure process. Assuming a session host is used solely to publish a browser that points to different Uniform Resource Locators (URLs) based on the third party accessing it, it becomes possible to create groups based on the published resources.

Hypothetically, consider a session host that publishes Microsoft Edge to multiple third parties. These third parties may have employees with different access requirements, where some may require access to, for example, webapp_erp, others to webapp_siem and some to both. To implement proper access control, AD groups must be created based on the destination resources rather than the source organization that accesses them. By creating groups in this manner, access can be granted on a need-to-access basis instead of defining broader rules for organizations or all their members. This approach results in a more granular access control and simplifies management, as users can be added to the appropriate group through a formal request process and thereby gain access to the required RDS resources. These groups also enable the creation of access roles that rely solely on AD group membership, on the firewalls, granting access strictly based on identity.

4.9 Access Roles - Configuration

As the access roles have been designed to match the specified scenarios in this environment, they can now be created based on the previously configured AD groups. The access role configuration is relatively straightforward. Although multiple options are available in the configuration panel, in this environment, only one setting per object was required: the designated user or group.

The configuration process works as follows: when creating a new access role object in the environment, the user or group selection section in the SmartConsole client connects to the identity provider through a configured object, such as the previously mentioned LDAP Account Unit. When adding a user or group to the access role, the client queries the entire domain for all users and groups. From this list, it is possible to select the pre-designed AD groups or, if individual access rights are needed, to select specific user accounts.

4.10 Access Rules based on Access Roles

There are multiple ways in which identities can be used in access rules, depending on how the rules are intended to be configured. For example, identities can be used in a "main" rule, where the identity is set as the source, and the destination, service, and action (such as accept or drop) are defined. Alternatively, identities can be used within inline layers. Inline layers function by first creating a standard access rule from a source to a destination and defining the service, after which the action is set to a new inline layer. This inline layer can then be configured with an implicit accept or drop rule to handle traffic that does not match other rules within the layer. Within this inline layer, access rules can be created similarly to the main rules, with the identity set as the source.

An illustrative example is a terminal server used in RDS or Citrix. Suppose that

this terminal server hosts an application, such as Chrome, published for different user groups with access to different sites. An access rule can be created with the terminal server as the source and a network object group that contains all possible target resources for the destination. The action is then set to an inline layer, within which new rules are created. Each rule corresponds to an access role that contains defined groups, and each rule allows access only from that access role to the specific target resources required by that group. This process is repeated for all groups. The result is simplified rule management while ensuring that users have access only to the resources they are intended to reach.

This type of access rule is also highly secure because it relies on identities rather than source and destination IP addresses. In a hypothetical scenario where a Citrix server is misconfigured and an attacker gains access through the applications, the attacker would not be able to access resources assigned to other groups. The inline layer system ensures that identities can access only resources explicitly defined for them, constraining the user within the machine. Additionally, if the terminal servers are micro-segmented, the attacker cannot move laterally to other machines.

5 Results

By implementing a solution based on the specific design choices presented in the case study, it became clear that the chosen direction was correct and that the results could be gradually observed as access control modifications were applied within the environment based on the design choices discussed previously. This demonstrated that the design was effective, easy to implement and functions well even in a more complex network architecture.

5.1 Effectiveness of access control based on identities

Identity-based access control has been noted to be exceptionally effective in securing the environment and provides network administrators with granular control over access rights. As firewall rules were created with micro-segmentation as a primary consideration and further enhanced through the use of identity-based access roles, managing access rights for users from external organizations accessing the resources becomes straightforward. This is because there is a single source of truth that needs to be modified to enforce changes, namely AD, which acts as the identity source for users through user and group management.

After the initial configuration and testing of necessary services, it was observed that access control schemes became effective in blocking unnecessary access to re-

sources. Specifically, access from identities that were not authorized or were not assigned to the correct groups in AD was successfully denied.

During the testing phase, different user accounts were assigned to different user groups and access roles were created based on those groups on the firewall solutions. When firewall rules were configured using these access roles, all unnecessary traffic between networks was blocked based on associated identities, demonstrating that this is an effective solution.

The effectiveness was further validated when similar access rules were created to replace existing rules through inline layers. In this configuration, the main rule allows traffic from defined sources to defined destinations with relevant ports, while sub-rules were used to provide greater granularity by restricting specific identities to access only certain destinations. Through log monitoring, it was possible to observe near real-time data and interpret it to confirm that the rules were functioning as intended.

5.2 Challenges in Implementation

As mentioned during the implementation phase, when MUH clients were deployed on terminal servers and other servers where multiple users may be logged on simultaneously, a minor issue emerged. This issue was related to the fact that third parties were using published resources. When a user logged onto the terminal server using Citrix or RDS, a situation arose in which the client identified the user as belonging to an organization based on the user's UPN. This meant that if users from contoso.local were accessing resources while the AD domain was fabrico.local, the identities were resolved to contoso.local. As a result, the authentication process on the firewall side failed to retrieve the correct group memberships for the user, leading to access being blocked for legitimate users.

To address this issue, a registry key named "ResolveFQDN" was configured. This

setting enabled recognition of the user's domain and mapped it to the customer's domain instead. As a result, authentication on the firewall side succeeded, the necessary group memberships were retrieved for the user, and access was granted based on the defined access roles.

Despite the issue described above, there were minimal implementation challenges when configuring the zero-trust-based access control, as different client installations were properly streamlined and configuration settings were easily manageable. The most challenging aspect of the case study was designing access roles and the necessary groups for different scenarios. Once this was completed, creating rules based on identities became relatively straightforward.

Overall, the project progressed smoothly and no major challenges were encountered. Therefore, it can be argued that this type of solution is easy to implement, provided that prerequisites such as modern firewall solutions or equivalent technologies are already in place within the environment.

5.3 Zero-Trust-based Access Control in a Complex Network

As the case study takes place in a network environment that consists of multiple layers and micro-segmentation, creating a complex environment, it acts as an ideal candidate for evaluating whether the designed solution functions effectively in a complex environment. This provides insight for applying the solution in environments of varying complexity.

Because identity sharing functions properly, even when identity information is relayed to one of two or more firewall clusters in environment, firewall solutions can utilize this information. This enables the implementation of zero-trust-based access control, where identities or objects derived from those identities are used as source

parameters in access rules.

Based on the results above, it can be concluded that the solution performed well, making a zero-trust-based access control solution feasible even in highly complex environments. In this case study, it was observed that identity information can be shared across layers and micro-segments with firewall clusters. This type of sharing capability, which is generally supported in modern firewall solutions, also simplifies administration.

6 Discussion

In this research, three research questions had to addressed: whether it can be shown that access control based on identities is effective by using identities as a testing method, whether challenges appear during implementation, which could possibly lead to the implementation being difficult, and whether the implementation is plausible in a more complex network infrastructure that consists of multiple different layers and micro-segments. The purpose of the research was to use a technical implementation in a case study environment to address these questions, providing a clear guideline for other researchers in this area.

One expectation from this research was that the zero-trust-based access control solution would be a valid and effective way to control traffic within a network infrastructure, which could be proved by using identities as a testing method. However, this still needed to be proven through the implementation of a possible design. This expectation was confirmed through multiple testing phases. The second expectation was that there would not be major challenges that would lead to the implementation being difficult, which was confirmed during the implementation phase, where only a minor challenge was encountered and it was relatively easy to resolve, demonstrating that the implementation was not too difficult. Finally, as the research was successful in demonstrating that zero-trust-based access control functions effectively and it is based on identities, it showed that such a solution can be implemented even in a more complex environment, as the environment in the case study had multiple layers

and micro-segmentation within said layers.

As expressed earlier, one of the expectations of this research was that the zero-trust-based access control solution would be a valid and effective way to control traffic within a network infrastructure. However, this still needed to be demonstrated by observing it in operation through the implementation of a possible design.

In the literature review chapter, one of the key aspects related to this objective was determining what the definition of an identity actually is. This is particularly important because, as identified in the literature review, the definition of an identity is highly context-based and can refer to multiple concepts. For this reason, clarity from the perspective of the computational space is essential, so that the identities can be appropriately referenced and used in the implementation. If the definition of an identity was unclear or did not exist in a meaningful way, it would undermine the results and the research question related to effectiveness, as there would be no clear guideline on what forms the basis of access control.

As noted in the chapter regarding zero trust network access, the existing literature already describes how access can be granted based solely on identities. This guided the research in the appropriate direction and confirmed that others have reached similar conclusions through conceptual work or technical testing.

In the case study, following the implementation of a design created for zero-trust-based access control within this environment and for specific scenarios, it became evident that this research question could be answered. Testing within the environment demonstrated that the implementation was successful. Achieving this result was important as it also supports the other research questions and assists other researchers in validating their assumptions regarding the use of zero-trust-based access control. Furthermore, such solutions can be implemented within their own organizations, as this research demonstrated their effectiveness.

One of the goals was to determine whether the implementation of this type of

solution would come with challenges that could make it difficult. In the literature review, several valid concerns regarding legacy systems and scalability issues were identified. However, these aspects cannot be fully discussed on the basis of the results of this implementation, as they may externally influence the difficulty of the implementation.

As noted earlier, the expected result of this research was that implementation would not come with challenges that would make the implementation difficult, and this proved to be the case. Most of the required installations and configurations were straightforward to perform. The most challenging aspect of the implementation was the creation of a design for specific use case scenarios. However, since this was achieved successfully within the given environment on the basis of the proposed generalized design, it demonstrates that such a process is feasible and not overly complex. It should be noted that this environment does not include site-to-site connections around the world, which means that potential implementation difficulties in such scenarios remain unexamined.

Overall, it can be concluded that the implementation was not difficult. Even the minor issue related to the MUH clients was easily resolved and required minimal effort to address. Together, the design phase, the implementation process, and the resolution of this issue indicate that such a solution can be implemented with relative ease, supporting the hypothesis that implementation is not difficult.

As implementation can be considered not difficult, it provides an opportunity for network and security administrators within their organizations to explore implementing a similar solution to enhance their security posture. Given the limitation regarding site-to-site connections, this factor should be taken into account when planning the implementation, particularly if the organization intends to extend zero-trust-based access control to all or some of its sites.

Regarding the complexity of the environment in this research, with its multi-

ple layers and micro-segments, it served as a suitable candidate for testing whether this implementation is feasible in complex environments. Investigating this question also provides insight into whether the implementation is plausible in simpler environments or potentially in even more complex environments.

This is a somewhat challenging subject, as in this environment the implementation was successful after confirming that access was correctly allowed and blocked for relevant identities according to the rules between layers and micro-segments. However, as previously noted, site-to-site connections were not included, which would introduce an additional layer of complexity. It can therefore be argued that the implementation has been shown to work in complex environments to the extent allowed by the scope of this research.

It could be hypothesized that the design developed for this case study could be applied to even larger network infrastructures with multiple sites, given the identity-sharing capability of the firewall solutions. This hypothesis requires the assumption that the firewall solutions are under the same management or otherwise be able to share identities. With this type of configuration, the implementation could be logically scaled upward, although some preliminary work and testing would likely be necessary.

Overall, it is reasonable to assume that the implementation can be applied on the basis of the proposed generalized design across different types of environment. For network and security administrators planning such implementation, it is important to develop scenario-specific designs tailored to the existing environment, as demonstrated in this research.

In summary, access control based on identities was found to be effective through testing in both test and live environments, as well as through monitoring the rules. This indicates that such a solution can enhance the security posture of an organization and support current or emerging zero-trust architectures. The implementation

is not overly difficult, and although designing the identity group logic and creating implementation plans for different use case scenarios may require some time, it is worthwhile and can be accomplished without major challenges. Since the case study focused on a complex environment, excluding site-to-site connections, which should be examined in future research, the implementation was successful, with all components functioning as intended. This suggests that the approach can be applied to environments of varying complexity.

7 Conclusion

The research conducted in the case study was successful and answered the different questions of this study. After validating that the proposed design functioned as intended, it was possible to conclude that the solution effectively resolved the issue faced by the organization in the case study. Throughout the implementation and validation phases, a limitation of the study was identified and noted for further research.

7.1 Research Summary

The research focused on three different questions through the implementation of a modified design of the originally suggested general design for zero-trust-based access control in a more complex network infrastructure.

Through the implementation of the modified design across different scenarios, it was possible to observe that the design is effective. The design was implemented using modern firewall solutions from Check Point, along with proprietary software provided by the vendor. After utilizing access rules based on the design in both test and live environments, the implementation proved to be effective in blocking unwanted traffic in relevant directions or to specific resources, as well as allowing desired traffic based on the configured identities. This verified that, in this aspect, the implementation functions as intended.

As the design was being implemented in the environment, the research also

aimed to determine whether implementing this type of design is difficult, considering complexity or other challenges. It was observed that, despite some minor difficulties, the design itself is not particularly difficult to implement.

One of the important questions of this research was to determine whether implementing this type of solution would remain plausible given a complex network infrastructure with multiple layers and segmentation solutions. As identities were shared between firewall solutions and could be utilized in rule sets on both firewalls, this approach mitigates the complexity of the environment and demonstrates that the implementation of zero-trust-based access control is plausible even in more complex environments.

7.2 Practical Implications

Given that the research was successful and that the defined research questions were answered appropriately, this research provides an opportunity for network and security administrators to work together and implement the proposed design in this paper within their own organizational network architectures. As demonstrated in the research, the design was applied to three different types of scenarios, indicating that it can also adapt to varying assumptions.

Although the generalized design proposed in this research depends on modern networking solutions, which may be a limiting factor in older environments where systems have not yet been upgraded, it is still possible to incorporate the design into the network architecture planning process. This allows organizations to design with future system upgrades in mind.

7.3 Limitations of the study

Although the network environment in this case study is complex due to different layering and segmentation practices, there is one subject that was not addressed, as it was not necessary in this scenario. This subject concerns the possibility of additional site-to-site connections within the organization. As the network did not contain this type of configuration, it was not possible to test or verify the research questions in the context of site transferability.

7.4 Suggestions for further research

Future researchers could examine the proposed design choices presented in this research to determine whether scalability issues may arise. For example, if an existing environment contains only a few firewall clusters implementing this type of configuration and additional clusters must be added for various reasons, it would be relevant to assess whether issues occur in identity sharing or whether access rules fail to function as expected based on the defined access roles.

Another additional subject of interest, as discussed in the limitations of the study, would be the following: assuming an organization operates globally and has implemented site-to-site connections between different locations with varying levels of network infrastructure complexity, would it be possible for these identities to still be shared between the gateways if they exist within the same tenant? If so, identities could be utilized across different sites, minimizing administrative overhead, as administrators would no longer need to configure access rules based on network attributes beyond the initial site-to-site connection rules.

References

[1] J. Kindervag, S. Balaouras, and L. Coit, "No More Chewy Centers: Introducing The Zero Trust Model Of Information Security", *Forrester*, Sep. 2010, Accessed on 18.11.2025. [Online]. Available: https://media.paloaltonetworks.com/documents/Forrester-No-More-Chewy-Centers.pdf.

[2] V. Cerf, Y. Dalal, and C. Sunshine, *Specification of Internet Transmission Control Program*, RFC 675, Dec. 1974. DOI: 10.17487/RFC0675.

[3] J. Postel, *Internet Protocol*, RFC 791, Sep. 1981. DOI: 10.17487/RFC0791.

[4] J. Postel, *Transmission Control Protocol*, RFC 793, Sep. 1981. DOI: 10.17487/RFC0793.

[5] R. Braden, *Requirements for Internet Hosts – Communication Layers*, RFC 1122, Oct. 1989. DOI: 10.17487/RFC1122.

[6] J. D. Day and H. Zimmermann, "The OSI reference model", *Proceedings of the IEEE*, vol. 71, no. 12, pp. 1334–1340, Dec. 1983. DOI: 10.1109/PROC.1983.12775.

[7] D. Meyer and G. Zobrist, "TCP/IP versus OSI", *IEEE Potentials*, vol. 9, no. 1, pp. 16–19, Feb. 1990. DOI: 10.1109/45.46812.

[8] I. Maathuis and W. Smit, "The battle between standards: TCP/IP Vs OSI victory through path dependency or by quality?", in *Proc. of the 33rd European*

Solid-State Device Research, Delft, Netherlands: IEEE, Dec. 2003, pp. 161–176. DOI: 10.1109/SIIT.2003.1251205.

[9] R. Dube, "A taxonomy of segmentation in network security", *IEEE Access*, vol. 14, no. 1, pp. 16 921–16 935, Jan. 2026. DOI: 10.1109/ACCESS.2026.3658250.

[10] N. R. Kotha, "Network segmentation as a defense mechanism for securing enterprise networks", *Turkish Journal of Computer and Mathematics Education*, vol. 11, no. 3, pp. 3023–3030, Dec. 2020. DOI: 10.61841/turcomat.v11i3.14942.

[11] S. Tyagi and G. Murugesan, *"Measuring Ransomware Lateral Movement Susceptibility via Privilege-Weighted Adjacency Matrix Exponentiation"*, 2025. arXiv: 2508.21005.

[12] H. A. Al-Ofeishat and R. Alshornam, "Build a Secure Network Using Segmentation and Micro-segmentation Techniques", *International Journal of Computing and Digital Systems*, vol. 16, no. 1, pp. 1499–1508, Sep. 2024. DOI: 10.12785/ijcds/1601111.

[13] "IEEE Standard for Local and Metropolitan Area Networks — Bridges and Bridged Networks Amendment 40: YANG for the Multiple Spanning Tree Protocol", *IEEE Sandard 802.1Qdy-2025*, 2025.

[14] G. Shaji, "The Critical Role of Micro-segmentation in Modern Cybersecurity Architectures: A Comprehensive Review", *Partners Universal Multidisciplinary Research Journal*, vol. 2, no. 2, pp. 24–34, Mar. 2025. DOI: 10.5281/zenodo.15063176.

[15] M. Arifeen, A. Petrovski, and S. Petrovski, "Automated Microsegmentation for Lateral Movement Prevention in Industrial Internet of Things (IIoT)", in *14th Int. Conf. on Security of Information and Networks (SIN)*, vol. 1, no. 1,

Edinburgh, United Kingdom, Dec. 2021, pp. 1–6. DOI: 10.1109/SIN54109.2021.9699232.

[16] J. Kwon, C. Hähni, P. Bamert, and A. Perrig, "Mondrian: Comprehensive Inter-domain Network Zoning Architecture", in *Proc. Network and Distributed System Security Symposium*, Reston, VA: Internet Society, Feb. 2021, pp. 1–16. DOI: 10.14722/ndss.2021.24378.

[17] N. Mhaskar, M. Alabbad, and R. Khedri, "A Formal Approach to Network Segmentation", *Computers and Security*, vol. 103, no. 1, pp. 1–32, Apr. 2021. DOI: 10.1016/j.cose.2020.102162.

[18] O. Lamdakkar, I. Ameur, M. Eleyatt, F. Carlier, and L. Aitibourek, "Toward a modern secure network based on next-generation firewalls: recommendations and best practices", *Procedia Computer Science*, vol. 238, no. 1, pp. 1029–1035, Jan. 2024. DOI: 10.1016/j.procs.2024.06.130.

[19] R. Anderson, *Security Engineering: A Guide to Building Dependable Distributed Systems*, 3rd ed. Hoboken, NJ, USA: John Wiley & Sons, Ltd., 2020.

[20] S. P. Marsh, "Formalising Trust as a Computational Concept", Ph.D. dissertation, Dept. Comp. Sci. and Math., Stirling Univ., Scotland, UK, 1994.

[21] M. Deutsch, "Cooperation and Trust: Some Theoretical Notes", *Nebraska Symposium on Motivation*, vol. 10, pp. 275–319, Lincoln, NE, USA: University of Nebraska Press, 1962.

[22] S. Rose, O. Borchert, S. Mitchell, and S. Connelly, "Zero trust architecture", U.S. Department of Commerce, Washington, D.C., Tech. Rep. NIST Special Publication 800-207, 2020. DOI: 10.6028/NIST.SP.800-207.

[23] P. Phiayura and S. Teerakanok, "A Comprehensive Framework for Migrating to Zero Trust Architecture", *IEEE Access*, vol. 11, no. 1, pp. 19 487–19 511, Feb. 2023. DOI: 10.1109/ACCESS.2023.3248622.

[24] Y. He, D. Huang, L. Chen, Y. Ni, and X. Ma, "A Survey on Zero Trust Architecture: Challenges and Future Trends", *Wireless Communications and Mobile Computing*, vol. 2022, no. 1, pp. 1–13, Jan. 2022. DOI: 10.1155/2022/6476274.

[25] A. F. Baig and S. Eskeland, "Security, Privacy, and Usability in Continuous Authentication: A Survey", *Sensor*, vol. 21, no. 17, pp. 1–26, Sep. 2021. DOI: 10.3390/s21175967.

[26] J. Su, A. Shukla, S. Goel, and A. Narayanan, "De-anonymizing Web Browsing Data with Social Networks", in *Proc. of the 26th Int. Conf. on World Wide Web*, Perth, Australia, Apr. 2017, pp. 1261–1269. DOI: 10.1145/3038912.3052714.

[27] E. B. Fernandez and A. Brazhuk, "A critical analysis of Zero Trust Architecture (ZTA)", *Computer Standards and Interfaces*, vol. 89, no. 1, pp. 1–12, Apr. 2024. DOI: 10.1016/j.csi.2024.103832.

[28] V. Mavroudis, *Zero-Trust Network Access (ZTNA)*, 2024, arXiv: 2410.20611.

[29] M. Dabrowski and P. Pacyna, "Generic and Complete Three-Level Identity Management Model", in *2nd Int. Conf. on Emerging Security Information, Systems and Technologies*, Cap Esterel, France, Aug. 2008, pp. 232–237. DOI: 10.1109/SECURWARE.2008.18.

[30] Y. Cao and L. Yang, "A survey of Identity Management technology", in *IEEE Int. Conf. on Information Theory and Information Security*, Beijing, China, Dec. 2010, pp. 287–293. DOI: 10.1109/ICITIS.2010.5689468.

[31] Š. Čučko, Š. Bećirović, A. Kamišalić, S. Mrdović, and M. Turkanović, "Towards the Classification of Self-Sovereign Identity Properties", *IEEE Access*, vol. 10, no. 1, pp. 88 306–88 329, Aug. 2022. DOI: 10.1109/ACCESS.2022.3199414.

[32] D. Fett, R. Küsters, and G. Schmitz, "A Comprehensive Formal Security Analysis of OAuth 2.0", in *Proc. of the 2016 ACM SIGSAC Conf. on Computer and Communications Security*, ser. CCS '16, Vienna, Austria: Association for Computing Machinery, Oct. 2016, pp. 1204–1215. DOI: 10.1145/2976749.2978385.