future internet

MDPI

Review

Misconfiguration in Firewalls and Network Access Controls: Literature Review

Michael Alicea and Izzat Alsmadi *

Department of Computing and Cyber Security, Texas A&M University, San Antonio, TX 78224, USA; malic01@jaguar.tamu.edu
* Correspondence: ialsmadi@tamusa.edu

Abstract: Firewalls and network access controls play important roles in security control and protection. Those firewalls may create an incorrect sense or state of protection if they are improperly configured. One of the major configuration problems in firewalls is related to misconfiguration in the access control roles added to the firewall that will control network traffic. In this paper, we evaluated recent research trends and open challenges related to firewalls and access controls in general and misconfiguration problems in particular. With the recent advances in next-generation (NG) firewalls, firewall roles can be auto-generated based on networks and threats. Nonetheless, and due to the large number of roles in any medium to large networks, roles' misconfiguration may occur for several reasons and will impact the performance of the firewall and overall network and protection efficiency.

Keywords: network firewalls; network access controls; firewall roles misconfiguration

Citation: Alicea, M.; Alsmadi, I. Misconfiguration in Firewalls and Network Access Controls: Literature Review. *Future Internet* **2021**, *13*, 283. https://doi.org/10.3390/fi13110283

Academic Editor: Steven Furnell

Received: 9 October 2021
Accepted: 2 November 2021
Published: 8 November 2021

1. Introduction

Network access control is an important topic for virtually every organization today. It is difficult to find a business that does not in some way utilize computers and other internet-connected devices for day-to-day operations. Both organizations and households keep sensitive information stored on machines in their networks. The purpose of network access controls is to keep those machines and the data stored within them safe. Network access controls are implemented through devices on the network such as a firewall or an Intrusion Detection/Prevention System (IDS/IPS).

These devices are often the first line of defense against malicious actors trying to gain access to information or resources in a network. A potential issue with these devices is that attackers may find ways to exploit them and bypass their security. In this paper, we are not so much interested in flaws in the devices code that the manufacturer has to patch. Instead, we are interested in the human errors in the setup of these devices that can cause a security breach. What we focus on is the rule set to these network access control devices, the firewall and the IDS/IPS.

Oftentimes, when there is an error in a config file or the device is not plugged in correctly, it is easy to notice, and oftentimes, the device will give a warning and will not function until the error is fixed. Errors inside of the rule sets of these devices are a different story. We are not talking about incorrect syntax, as that will often throw an error, forcing the user to fix it for the device to function. Instead, we will focus on the rules themselves and how they interact with one another. As an example, consider a network admin who has a web server in their network that they want to be accessible from the web. They want to block ssh but allow the web ports, so the admin blocks port 22 and makes a rule to allow port 443. By mistake, the rule to allow port 443 allows the range of 0 to 443. As far as the firewall is concerned, this is a perfectly valid rule, and it is successfully added. Despite the rule having valid syntax, this firewall is misconfigured, as the rules it is enforcing are not what the user intended.

Here we will be discussing the topic of rule misconfigurations and detection. Just because a set of rules can be validated by the device, it does not mean that they work correctly. There has been research to detect these potential issues in the past decades. It is these methods and tools that we will take a closer look at, namely tools that correct the human error portion of a network access control device's rules.

2. Methods

This paper will serve as a structured literature review (SLR). The goal of this paper is to assess research papers on access control misconfigurations and the authors' proposed solutions to those problems. In this paper, we will list the different kinds of misconfigurations as defined by the authors, how they occur and the author's recommended solution to each of them.

2.1. Research Questions

There are several research questions that we will be addressing:

- What are the different types of access control misconfigurations, in both firewall and IPS/IDS?
- What causes these misconfigurations to occur in each network access control system?
- What are some of the proposed solutions in the literature for firewall access control misconfigurations?

2.2. Search Strategy

To find articles for this SLR, Google Scholar was used to find relevant articles. There were two searches done to produce articles. The first was for the terms "firewall", "misconfiguration" and "snort", which produced nearly 700 articles. A small number of the articles in this search were relevant, but of the few relevant articles a new search term was found. Most of the relevant articles also contained the phrase "rule anomaly", which was the next search that was made. A search for the phrase "rule anomaly" was made and produces about 300 results with a drastically higher proportion being relevant to the research topic. These two searches produce around 90 articles about the topic of access control misconfigurations and fixes or mitigations to them.

2.3. Inclusion and Exclusion Criteria

There were several criteria regarding which papers were included. The first was about whether the paper had sufficient mention of misconfigurations in either firewalls or IDS/IPS. A paper did not necessarily have to be about network access control misconfigurations, but it did have to discuss the topic in detail. Papers that fit in this category would have a section or two that talk about misconfiguration in multiple paragraphs. Papers that are entirely about this topic or towards some fix to the topic were accepted. Papers denied under this criteria were denied because of insufficient discussion, such as a passing comment or the misconfiguration mentioned not being about firewalls or IDS/IPS. The first example is when a paper makes a one-off comment such as "this error could be due to a misconfiguration". The second is when the misconfiguration was about something else: one paper discussed misconfigurations in Amazon S3 services. The last example of an irrelevant paper is when the only mention of misconfigurations is in the citations. This often takes place due to the search terms being in the paper but the term "misconfiguration" only occurrings in the title of a cited paper. Papers that are in line with these examples were all excluded as being off topic.

The third criteria is access. While Google Scholar lists papers from a large variety of sources, not all papers listed can be publicly accessed. For the purposes of this SLR, we only use papers that can be accessed publicly.

Finally, the paper must be written in English. Only a small number of papers were affected by this due to the way Google Scholar search works. All papers affected by this

rule had their abstract written twice, once in their native language and once in English, with the rest of the paper being written in only the authors' native language.

3. Misconfigurations

All of the gathered papers talk about at least one of the three most common misconfigurations, namely shadowing, correlation and redundancy. There are other types of misconfigurations, but these are either platform-specific or can only occur when there are multiple network access control solutions on the same network. Examples of these are inter-firewall anomalies or a flowbit misconfiguration inside of Snort. Most of the listed misconfigurations are a result of multiple rules which overlap in scope. Firewall rule sets generally do have some deliberately overlapping rules. Most of the automated tools choose to notify the user that there is a potential issue.

Table 1 below lists the misconfigurations covered in each paper.

Table 1. Types of Misconfigurations covered in literature.

Shadowing	[1–52]
Correlation	[1–6,8–22,24,27,28,31–35,37–50,52–55]
Redundancy	[1–22,24,25,27–35,37–62]
Generalization	[1–5,7–12,14–19,22,24,25,27–29,31,33,34,38,39,43–46,50,52]
Irrelevance	[2,7,9,11,16,18,19,21,22,24,30,31,40,43,46,50]
Flowbit	[39]

3.1. Shadowing

In a firewall rule list, Reference [3] states that "A rule is shadowed when a previous rule matches all the packets that match this rule, such that the shadowed rule will never be activated". A more generic rule with a wide scope is preventing the activation of a second rule with a more narrow scope. This definition, or one very similar to it, is present in every paper discussing the topic of rule shadowing. This definition is fairly generic, and we can see that some authors add one key condition. The author [13] gives the additional condition that the two rules must have different actions, if the shadowed rule is an allow, the shadowing rule must be a deny and vice versa. This distinction is important as it is what differentiates shadowing from other misconfigurations. With this definition, we can now observe exactly how this misconfiguration occurs. Traditionally in firewalls, rules are enforced in the order they occur in the list of rules. Shadowing will occur when a rule higher in the list completely overlaps a rule lower in the list. This makes rule ordering something to take into careful consideration [15]. When you have a rule with a scope that is entirely contained in another rule, shadowing may occur depending on which order they occur in. Having the correct rules makes no difference if they are in the wrong order. Some firewalls have an option to add a numbered priority onto rules to set the order in which they are enforced. An example of a firewall with this implementation would be the firewall in Google Cloud Platform. Instead of relying on the order in which the rules are written, the rule with a higher priority can potentially shadow a rule with lesser priority. In a firewall that enforces rules on a numbered priority system, rule order is completely irrelevant. In this case, take care to assign the correct priorities and know how the firewall handles rules with the same priority.

An IDS/IPS such as Snort can suffer from these issues as well, but the reason they occur is different. Reference [39] lays out that rule priority inside of Snort is determined by the action associated with the rule such as log, alert, pass, etc. Shadowing occurs in Snort when one rule has a scope that encompasses a second rule and has a higher-priority action. While action taken has no impact on the order of rule enforcement in a firewall, an action in Snort can only shadow an action of lower priority. For example, an alert rule

in Snort cannot shadow a pass rule, but a pass rule can shadow an alert rule. Due to the higher priority of the pass action, a pass rule will always be evaluated before an alert rule. In the event that a packet matches two or more rules with the same action, every matching rule will be triggered. For instance, if an incoming packet meets the criteria for multiple log rules, each of those rules' alerts will be logged. This trend will continue with the other misconfigurations listed in this SLR, Snort suffers from the same potential misconfigurations, but they occur for slightly different reasons.

Shadowing does not only occur when one rule completely encompasses another rule. The authors of [27] list a different kind of shadowing that they call total shadowing. The authors describe total shadowing as when the combined scope of multiple rules covers the entire scope of another rule. As an example, suppose we have a simple port blocking rule that blocks inbound traffic between ports 1 to 100. A shadowing set could be one allow rule from 1 to 50 and a second from 51 to 100. This causes the initial block rule to be shadowed by the other two rules combined. As we will see later in this SLR, total shadowing actually refers to multiple instances of another misconfiguration, but the end result is the same, and a rule is being prevented from being evaluated.

So far, the discussion around the concept of shadowing has revolved around the assumption that there is only one network access control mechanism present on the network. Larger networks can contain multiple firewalls or IDS/IPS that could conflict with one another. Reference [11] discusses the possibility of rules within two different firewalls on the same network conflicting with one another. When a rule in one firewall shadows a rule in a second firewall, we have what is called an inter-firewall shadowing anomaly. Individually, the two firewalls in question can have perfectly valid rule sets with no misconfigurations individually. The issue is when the rule sets put together cause shadowing. This will result in parts of the network suffering from the results of the misconfiguration. For example, if a firewall at the network edge blocks social media but a firewall for a specific network allows it, only the users behind the second firewall will suffer the effects of the misconfiguration. It is important to note that such an anomaly does not necessarily have to occur between two of the same type of device, i.e., two firewalls or two IDS. What this tells us is that you cannot look at every component individually. Having a well-made network access control policy on individual devices does not necessarily mean the overall network access control policy works as intended. Network access control policy needs to take place on a macro level. Whether or not the sum of all policies mesh together well is just as important as the individual components in the network.

Shadowing is a serious misconfiguration as its existence in a rule set completely nullifies other rules. While this can occur intentionally, such as leaving old rules, unintentional shadowing can lead to many problems. This could cause a denial in legitimate traffic, which could hurt an organization's productivity and cause issues for users. Even worse, this could cause malicious traffic to get through since the rule that would block them is shadowed [16]. Since a shadowed rule cannot be triggered, its removal has no effect on a firewall rule set. The issue is that simply removing a shadowed rule may not be the best course of action if it is supposed to catch some traffic. Alternatively, it may be more desirable to unshadow a rule by limiting the scope of the shadowing rule. When shadowing is detected within a rule set, it is important that the issue is identified as quickly as possible because an accidental inclusion of shadowing can be devastating [10]. Whether the rules are denying good traffic or allowing bad traffic, this kind of anomaly is at best annoying to users and at its worst a threat to the entire network. The high severity of shadowing stems from the fact that it causes the wrong action to be taken even though there is no problem with any individual rule that would throw an error by the firewall or IDS/IPS. Shadowing and total shadowing can be difficult to diagnose without the use of automated tools for this reason, and therefore it is important to evaluate the impact of the interactions of old rules with new rules.

3.2. Correlation

A correlation anomaly is similar to shadowing. Reference [14] defines correlation as when "one rule matches some of the packets that another rule may capture, and that other rule matches some packets, which the original rule captures, the rules may be in correlation. However, the actions of the two rules need to be different". In short, correlation can be thought of as partial shadowing. Two rules have a partially overlapping scope, causing the higher-priority rule to catch traffic that meets the criteria in that overlap. Only in the overlapping part of the two rule's respective scopes will there be an issue. This is what separates correlation from shadowing, i.e., the fact that a rule is unintentionally overridden some of the time rather than all of the time.

How this anomaly occurs is also very similar to that of shadowing. An earlier rule in a rule list is enforced first before the correlating rule later in the list. As [2] notes, switching the order of the two rules inverses which rule overrides the other in their shared scope. Regardless of which rule overrides the other, the fundamental cause of correlation is that two rules have a partially overlapping scope. The easiest way to think of this is as a Venn diagram, the portion where the two circles overlap is where the issue occurs. Assuming that this overlap is not deliberate, this means that the wrong action is being taken in this shared scope.

Earlier, we discussed how the concept [27] describes as total shadowing. We mentioned before that total shadowing represents multiple instances of a different misconfiguration. Total shadowing is the result of multiple instances of correlation that together entirely cover the scope of another rule. This results in that rule being effectively shadowed as the other rules will not permit this rule to be evaluated when a packet that meets its criteria comes through. Just like normal shadowing, removing this shadowed rule causes no change in the overall policy. Again, this may not be the desired solution if the shadowed rule is wanted. Whether the correlating rules are causing traffic to be denied or allowed when that action should not be the case, these rules should be carefully evaluated to make the correct change to achieve the desired result.

Just like shadowing, a network access policy can suffer from inter-firewall correlation. Reference [8] discusses how rules in multiple firewalls can correlate with one another and cause issues for portions of a network where the rules clash with one another. When multiple firewalls are involved, an upstream firewall can block or allow traffic in which a downstream firewall takes a different action but the two rules on each firewall do not completely overlap one another. This is equally true when using different types of components in the network, where an IPS may block or allow traffic it is not supposed to going to a downstream firewall. Different subnets inside a larger network will have their own policies in regards to what traffic is allowed in and out, and this may contradict the policy of upstream firewalls resulting in traffic not making it to the downstream firewalls in the first place. Fixing this issue will involve making sure that there are rules in place on upstream firewalls that are specific to certain subnets to avoid any such correlation between the firewalls.

Correlation is a serious misconfiguration for the same reasons as shadowing: it causes the wrong action to occur on traffic. In the policy visualization tool called PolicyVis, Reference [17] describes the issue of correlation as occuring when two rectangles when placed in 2D space have some, but not total, overlap. In a visualization tool like theirs, this is quite easy to notice when two such rectangles overlap. Without tools, however, correlation can be difficult to diagnose. In the event of shadowing, it is very clear that something is being allowed or denied every time when it is not supposed to. Correlation creates a situation where the policy will appear to sometimes work. This inconsistent appearance can be problematic due to the fact that the rule may work properly for some people or machines and not others. This could lead to the assumption that the problem lies with a specific computer's configuration or some users' actions. Once it is discovered that the problem lies within firewall rules, changes will need to be made so that the overlap is

either removed, a rule is made more specific to not block the intended traffic or the rule order is reversed so that the correct action is being taken.

3.3. Redundancy

Redundancy is the least severe of the misconfigurations. Redundancy is as simple as it sounds: it is when two different rules cover the same potential packets and take the same action [4]. Similar to shadowing, this can take place as two identical rules or as a general rule followed by a more specific one. Say, for instance, a firewall has two rules that block ports 20–30; these rules are redundant because both rules perform the same actions on the same packets. It is also a case of redundancy if a rule blocks that same range of 20 to 30 but a second rule only blocks port 22. Since port 22 is in the range of 20 to 30, there is no need to have another rule blocking port 22 since it is already blocked by the first rule.

Redundancy is not as straightforward when it comes to inter-firewall or inter-component misconfigurations. Reference [11] identifies an inter-firewall redundancy anomaly when a downstream firewall blocks traffic that an upstream firewall is already blocking. A rule-blocking specific traffic from outside the network is not needed if an upstream firewall already does so. If there is concern for insider threats, such traffic could occur if the attacker is on the internal network already making the need for some redundant rules between multiple firewalls necessary. This will still cause that rule to be evaluated twice on outside traffic but could nonetheless protect from traffic from an attacker within the same network. Whether or not inter-firewall redundancy is a misconfiguration depends entirely upon the threat level from within the user's own network.

Redundancy is one of the few misconfigurations that has no impact on the overall security of a network. Having multiple copies of the same rule, even across multiple devices does not affect policy enforcement. The only effect redundancy has on a network is to performance [6]. The issue with redundant rules is that it slows down enforcement. A firewall will try to match a packet to one of its existing rules and if it does not match any, it takes a default action. The problem with redundancy is that whenever a packet comes in, it gets checked by more rules than it should have to. Every time a packet is checked against a redundant rule, time and performance of the device is wasted. Redundant rules can be safely removed from a firewall as their removal will have no impact on security and will result in better performance from the firewall.

3.4. Irrelevance

Like redundant rules, irrelevant rules are of little to no severity to the security of a network. A rule is irrelevant if it is impossible for the firewall to encounter a packet that matches it [16]. Examples of an irrelevant rule could be blocking packets from a subnet that has no route to this particular firewall. If communication is outright impossible due to lack of a physical/wireless connection or otherwise, there is no need to make any rules regarding that subnet. Another example could be blocking traffic from the internet when hosts outside the user's network cannot reach their subnet. If no hosts behind a particular firewall have port forwarding or some way to be accessible from the internet, there would be no need to block traffic from the internet or from certain IP addresses out on the internet because outside hosts cannot send packets to the hosts behind the firewall.

Unlike redundancy, irrelevance cannot occur because of the configuration of other devices. If an upstream firewall blocks traffic that a downstream firewall also blocks, that situation is not irrelevance. That traffic cannot reach the downstream firewall, but it is considered to be inter-firewall redundancy because the traffic is blocked in two different firewalls. The distinction is that irrelevant rules are irrelevant because even in the absence of any rules, the traffic is still impossible [19].

Irrelevant rules are another one of the misconfigurations with zero impact on a network's security. Whether or not a network has irrelevant rules will not impact the security of the network. In addition, like redundant rules, the largest impact of irrelevant rules is on the performance of the firewall [18]. Irrelevant rules suffer from the same issues where

their very existence slows down the firewall with no benefit to the security of the network. If a rule is found to be irrelevant, it can be safely removed without any effect on the overall security on the network, and doing so can only be beneficial to network speeds.

3.5. Generalization

Rule generalization is a misconfiguration that is essentially the opposite of shadowing. Generalization is defined by [9] as when a rule with a more narrow scope has higher priority than a second rule with a scope that encompasses the first. If the rules were reversed, you would have shadowing. An important distinction in generalization is that the second rule must have a larger scope than the first. Using a similar example to that from redundancy, the first rule allows port 22, while the second rule blocks 20 to 30. The different action means that instead of redundancy, the second rule is a generalization of the first.

Generalization is quite similar to correlation. The authors of [9] define generalization as "Suppose the first rule matches the packets that also match the second rule while performing different actions, the rules are then considered generalized". Earlier we gave the example of a Venn diagram to describe correlation. Generalization is like having a smaller circle inside a bigger circle. It can be thought of as "poking holes" inside of another rule.

The impact that generalization has is similar to that of correlation. One rule is partially covering another rule. This can cause issues depending on what the the covered rule was blocking or allowing. Just like correlation, this could potentially lead to some good traffic getting blocked or some bad traffic being allowed. This misconfiguration is, however, the most likely to be intentional. As [12] puts it, "Generalization is considered only an anomaly warning because inserting a specific rule makes an exception of the general rule, and thus confirming this action by the administrator is important". In their tool that detects and corrects firewall misconfigurations, they do not automate fixes for generalization. Oftentimes, network administrators will create a broad rule then have exceptions for certain allowed traffic. In fact, an implicit deny all is a general rule to every single allow rule; thus any rule inside of a firewall that allows traffic is generalized by the implicit deny all. For these reasons, generalization is not always harmful to access control policy so long as its presence is deliberate.

3.6. FlowBit Misconfiguration

This final misconfiguration has the most variety in exactly what it is and how it occurs. In advanced firewalls and in IDS/IPS such as Snort, there is a feature that allows the user to create rules based on the state of the application protocol. One such example of this is Snort's flowbit feature, which is capable of keeping track of an application state. A flowbit can keep track of whether a user is logged in, what they are doing inside an application, etc. The exact implementation is different, depending on the platform and even the version of the platform. While different devices may refer to this feature by different names, we will refer to this feature as "flowbits", Snort's implementation.

In Snort, flowbits replaced the now-depreciated activate and dynamic rules. Flowbits allow Snort to keep track of what has happened in a session. Rules checking flowbits will only fire if the associated flowbit is set on that session. As [39] explains, there are multiple ways a flowbit rule can be misconfigured. A flowbit could be set to check for a connection state that is impossible, such as being logged in and not being logged at the same time. There is also the issue of not checking for enough connection states. It is possible that any given attack can be carried out in different ways that would have different connection states. While no single rule would be misconfigured, the policy as a whole is misconfigured as it did not take into account other means of that same attack occurring that one or more of its rules are attempting to block. There is also the issue of checking for flowbits that are too specific or broad. Checking for overly broad conditions, say being logged in, may result in many false positives as normal traffic triggers the alerts. Alternatively, too specific

conditions could allow an attack to sneak by because they did not meet everything that the flowbits were flagging.

Checking for a connection state can be fairly tricky and thus could be easy to misconfigure. The nature of flowbits often means that rules that check flowbits or their counterparts in other systems will often be looking for a few specific attacks. The result of a flowbit rule being misconfigured will leave a network either vulnerable or label good traffic as that type of attack. As stated above, the exact impact is determined by how the misconfiguration occurs. It can result in the attack slipping by entirely or causing benign traffic to trigger the rule.

4. Challenges and Research Trends

4.1. Stateless vs. Stateful

The author Al-Shaer was one of the first to go into this topic in depth back in the early 2000's when filtering firewalls were the norm. His work seems to be the catalyst for all of the tools that came later. Decades later, modern papers still reference his work and use the same definitions for misconfigurations, even if they are working with stateful firewalls or an IDS such as Snort. It is more difficult to find a paper that does not reference Al-Shaer than it is to find one that does. The main reason for this is the fact that the foundation of anomaly detection remains the same whether the tool is stateful or stateless. Adding connection states to rules does not impact the overall definitions of each misconfiguration, it only affects the complexity of checking for them by the program or algorithm. At a base level, the misconfiguration definitions for stateless and stateful are the same because of this fact.

Surprisingly, there is still work being done in the area of firewall misconfiguration and correction on stateless firewalls. Just to list a few, References [5,7,12,40,56,63] all discuss detection of misconfigurations on stateless firewalls. Despite firewalls advancing to be more complex than in the days of simple filtering, the research done into stateless firewalls has not stopped. The only real changes to each new paper is the approach taken to the problem. Most of the earlier papers took simple approaches. The project by [10] that eventually lead into "Firewall Policy Advisor" describes a technique used to detect misconfigurations. These early papers essentially replicated the manual process a human would take evaluating these rules one by one. The papers during this time took a similar approach to Al-Shaer's tool and built up from it. The more modern papers take different approaches. Papers such as [32,35] use algorithms that take entirely different approaches with efficiency in mind. While stateless firewall anomaly detection has waned in popularity, you can still find relatively new papers that still test on stateless firewalls. It would seem that stateless firewalls are still being tested on in order to keep the problem more simple so they can focus on creating faster algorithms.

Stateful firewall misconfiguration checking as stated before does not have any substantial differences, even using the same definitions without adding more types of misconfigurations. Some of the papers that check stateful firewalls include [26,27,29,41,59]. The first paper listed is an extension of "Firewall Policy Advisor", the tool that essentially started the work in this field. The paper's main focus is updating the existing tool to work on stateful firewalls. As for the rest of the papers, they have something in common. The majority of the stateless firewall papers are on openflow SDN firewalls. Several even focus on the challenges of dealing with inter-component misconfigurations inside of an SDN. Since SDNs are not physically connected, the authors work on the methods of finding routes through the SDN and the rules on each component inside the network. The biggest contribution of these papers is how they tackle this issue in a completely different environment. Some of these tools even tackle the issue of inter-component anomalies. Part of this problem included mapping out the network by finding all hosts and potential routes to those hosts. This is important because some papers discuss the potential of bypassing a firewall in the event that there are multiple routes to a host.

4.2. Algorithms vs. Model Based Approaches

As mentioned earlier, modern papers not using Openflow focus on particular algorithms. The reason for this is that openflow requires new implementation to these tools since the very structure of SDNs is different. Papers not discussing SDN on the other hand are focused on unique algorithms with the main focus being the efficiency of the algorithm. This trend is rather recent as many of the earlier papers do not even discuss the topic of efficiency, or if they do, it is a one-off statement of how long the program took to run.

Those early projects such as [12,17] gave very little if any attention into the efficiency of their approaches. These earlier approaches focused on completing the task of finding misconfigurations. The papers after Al-Shaer's "Firewall Policy Advisor" focused on adding extra features such as visualization to their implementation of misconfiguration checking. They modeled their algorithms after the process that humans would take when auditing rules. Given a list of rules, the algorithm looks at one rule and compares it against the rest of the rules. This process is repeated with every rule until a full list of misconfigurations between any two rules is created. During this time period, the test rule sets were fairly short with maybe a couple dozen rules. These rule sets for the most part were a proof of concept to show that the tool functions properly, rather than to be representative of a real rule set. This initial approach was replicated with little to no changes for years. Even Al-Shaer himself continued with his project in [26], which kept the same model while adding stateful firewall support to his existing "Firewall Policy Advisor".

In the mid 2010s, we can see that this approach became less normal. Over time, rule lists began getting longer with more complex features as smarter firewalls were capable of looking for more specific types of attacks on the network. Modern papers eventually stop adding new features. Instead, they focus on making the misconfiguration detection algorithms more efficient. We can see that recent papers such as [13,27,32,35,64,65] all take efficiency of their detection algorithm seriously, whereas before it was an afterthought. Since networks are much larger and have many more devices and services than in the past, a firewall rule list is much larger on average. There is also now the consideration that there are several firewalls on a network with anomaly detection needing to be checked between different components. The initial anomaly detection programs test with at most, a few dozen rules and maybe gave the number of seconds the program took to run. New research in the topic is more likely to heavily focus on this. Newer papers have large artificial rule sets that can reach tens of thousands. Many of them give a Big O notation alongside the time it takes for the algorithm to run for each of their test rule sets.

4.3. Traditional Networks vs. SDN

Around the mid 2010s, we start seeing a shift away from traditional networks, with most recent research being exclusively in SDNs. The majority of the efforts on this topic were on dedicated firewalls, single-entry-point-firewalls, or networks with multiple firewalls that interact with one another. Nearly all of the examples so far have fallen into this category. It was alluded to in the previous section that there are now two separate trends going on. The first, as discussed, was the efficiency of new detection algorithms. The paper discussed in that section used traditional networks or even took a static rule list and ran it through their algorithm. SDNs did not exist when this topic began, so it was not even discussed until the mid 2010s. Anomaly detection inside of an SDN is the other trend occurring.

Back in the early 2000s, firewalls were rather fractured and proprietary devices with their own special way of inserting and retrieving the rules on them. As such, the tools to detect misconfigurations were platform-agnostic. They typically worked by having some sort of interface in which the user could manually type their rules into the program's GUI, in the form of a table where each cell in a row will represent the fields of a single rule. No matter what firewall was being used, and regardless of the input format of that firewall, the anomaly detectors had a simple interface that allowed packet headers to be inserted into each cell. In fact, programs like [12] had an interface for inputting rules that resemble

how modern day firewall GUIs accept basic filtering rules. Once SDNs gained traction, things began to change a little.

SDN-focused anomaly detection occurs in very recent papers such as [1,29,41,59,60]. Unlike the early days of firewalls, the majority of the development into SDNs goes into an open framework that the various solutions are built upon. SDNs are built on the OpenFlow protocol, with most of the firewalls available for the various controllers being accessible via the REST API. These newer attempts at anomaly detection have an interesting advantage: as a tool that can read directly from the REST API, their tool would be cross-platform across different controllers. The open nature of SDNs is really a benefit as they do not need to make generalized tools like the original "Firewall Policy Advisor". When it comes to anomaly detection within an SDN, not much has changed. While the network organization is different, the actual anomalies that can occur within a rule set are still the same.

4.4. Traffic vs. Application

Over time, firewalls have become much smarter than they used to be. Filtering based on headers is the simplest kind of firewall. Stateful firewalls added the capability to remember details about sessions and taking that information into account. Today, it is more common for firewalls to have layer-7 capabilities. While the capability has existed for a while, it has recently become more practical to use layer-7 firewalls in large scale networks with a lot of traffic.

There are few papers that cover layer 7-firewalls such as [22,25,39]. The definitions of anomalies as defined in the early 2000s were general enough that they still hold up in the implementation of application layer firewalls. The same definitions find their way into the projects of anomaly detection in application firewalls. That being said, application-layer firewalls are much more complex than the early stateful firewalls. For this reason, there are very few papers that discuss application-layer firewalls, even among the SDN new papers. The projects that do discuss layer-7 firewalls have some limitations such as requiring every host behind the firewall being configured a certain way. There is definitely a severe lack of progress in bringing anomaly detection into layer-7 firewalls. With every step up in the OSI layers, it becomes more difficult as there are more checks required in order to classify a rule misconfiguration or lack thereof.

4.5. Autonomous/Next Generation Security Control

Over time, it became inevitable that the automation of access control devices would be considered an option. A firewall that is completely autonomous, i.e., a firewall that can be plugged in and protect a network without further interaction, does not currently exist. An autonomous firewall would have to generate rules dynamically, based on traffic that it deems normal and traffic that it deems malicious. When adding a new rule into the existing ruleset, the device would have to ensure that the new rule does not conflict with the existing ones. The author Al-Shaer in [11] discussed this possibility back in 2004. While the authors did not create a next-generation firewall, they discussed how adding new rules automatically could result in the need for automatically re-configuring. While there are not many projects making fully autonomous firewalls, there have been authors that have automated the process of reconfiguration rules. Most of the programs until very recently have only highlighted problematic rule pairs. These tools simply pointed out the existence of an anomaly so that a human could correct them. Recently developed algorithms have automated this task so that no human interaction is required to create a reconfigured ruleset without any rule anomalies. The algorithmic approaches to anomaly detection by [32,35] have solved the task of rewriting a sound rule list. These two projects from the section on efficient algorithms not only found misconfigurations but also rewrote the rule list to eliminate those issues. In order to create a firewall capable of dynamically changing its own rules, an algorithm such as the two listed could enable the capability as it would be able to resolve any problems by modifying or adding rules.

5. Summary

Over the years, the different types of misconfigurations have not changed since the inception of anomaly detectors. Despite firewalls gaining more and more functionality, the types of misconfigurations have remained the same since the early 2000s. There was only one anomaly added, the flowbit misconfiguration, which is very difficult to detect. The reason for this is that these are usually configured for a very specific attack, so these misconfigurations cannot be detected without the tool knowing exactly what those rules are intended to do. As for the rest of them, additional features only increased the complexity of checking for anomalies as there are more features that need to be checked. In short, these misconfigurations are as follows:

Shadowing is an anomaly that occurs when one rule has the same scope, or a larger encompassing scope, along with a higher priority than a second rule with a different action. This prevents the second rule from being evaluated because the first rule is triggered instead. Due to the first rule's higher priority, the second rule is never triggered under any circumstance.

Correlation is when two rules with different actions have an overlapping scope but not entirely. Correlation can best be described by a Venn diagram: in the overlapping area, the higher-priority rule is enforced instead of the lower-priority rule. If you the rule order were inverted, there would be exactly the same overlap, with just a change in which rule had priority.

Redundancy is when two rules with the same action have any amount of overlap. Whether the higher-priority rule entirely encompasses the scope of the second rule or if it only overlaps on one specific packet, it is redundant if the other rule also takes the same action on that traffic. Unlike other misconfigurations, redundancy has no impact on security, but extra rules slow down the performance of the firewall.

Irrelevance is an anomaly in which the firewall contains rules that affects packets that cannot possibly reach that firewall. Since the particular packets cannot possibly reach the firewall, irrelevant rules therefore have no impact on security, but their presence wastes performance on the firewall.

Generalization occurs when a rule with a scope that is entirely encompassed by a second rule with a different action but a lower priority. This means that every possible packet that the first rule catches would be caught by the second rule if the first did not exist. If the priority of the two rules were to be flipped, it would result in shadowing.

The term "Flowbit Misconfiguration" is Snort-specific, but most stateful access control programs have some form of similar method of keeping track of the session state. This type of misconfiguraiton refers to an error in a rule's detection of a session state when taking an action. When session state is being taken into account in a rule, the rule is oftentimes a response to a specific type of attack. A flowbit misconfiguration refers to when the flowbit in question is incorrect, whether the session state in question is impossible, too broad, or too specific. A misconfigured flowbit, or its equivalents, can result in the specific attack it attempts to stop sneaking through or be too broad and block legitimate traffic.

Very clear trends have been occurring over time. These trends follow the general development involving firewalls and networking. The growing prominence of SDN has been an influence to some extent for most of these trends such as efficiency or the type of firewall used such as an application firewall. In fact, many of these trends have influenced each other. The combination of SDN, stateful and application firewalls increased the need for efficiency, resulting in the trend of better algorithms. The major trends in anomaly detection are as follows: The transition between stateless and stateful occurred a long time ago for most projects, but stateless anomaly detection applications have not gone away entirely. Stateful firewalls do not bring new types of misconfigurations. They suffer from all the same anomalies with the only difference being that there need to be more checks for the extra features offered by stateful firewalls. Stateful anomaly detection brings with it more complexity, but in the end, they check for the same anomalies as the earliest projects.

In the early 2000s, networks were generally much simpler than they are today. As a result, access control policy was also much simpler. High efficiency was not much of a focus. The early anomaly detection tools had little if any focus on how quickly or efficiently the programs were executed. They were simple automation of how a human would check for rule conflicts. Over time, networks became more complex and access control policies became much more complex with many new services being run. This caused firewall rules lists to become much longer. We can see a trend around the early to mid 2010s that efficiency becomes more of a focus in projects. Papers began giving larger sections with more tests of of their implementation efficiency, some of which even focus the entire paper on making a faster algorithm.

This dichotomy brought one of the best changes. Older projects were using proprietary firewalls, all of which had slightly different syntax and stored rules differently enough so that it made such programs difficult to make. The response was to make a tool that had a generic way to input rules so that it could be compatible with any firewall. The tool could not read the firewall directly, and the user had to manually input their rules. Luckily, open-source standards seem to dominate early on in SDN. Almost every SDN test use OpenFlow SDN controllers, most of which come with open source firewall implementations that use the REST API. The switch to SDN appears to be leading to a future of interoperability between various platforms.

Just like the difference between stateful and stateless, application layer firewalls do not suffer from new anomalies, but do suffer from drastically higher complexity. This added complexity means that application-layer firewall misconfiguration detection projects are not as common. A very small number of anomaly detection projects use layer 7 firewalls.

Autonomous firewalls are still a ways off. One of the tasks required for an autonomous firewall has been worked on, even if was not the intent of the projects. One of the major features of autonomous firewalls is that they can add rules on their own. If a firewall were to do this, it would have to check the new rule it wishes to create against the existing rules to make sure there is no conflict, then re-arrange them if necessary. There are projects that can take a rule set, detect anomalies and then recommend a rearranged ruleset that removes said anomalies. While these need some further development, this task is an important feature for autonomous firewalls and is absolutely required to have properly functioning rule sets.

Author Contributions: Conceptualization, I.A.; methodology, M.A.; software, M.A.; validation, I.A.; formal analysis, M.A.; investigation, I.A.; resources, M.A.; data curation, M.A.; writing—original draft preparation, M.A.; writing—review and editing, I.A. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Ethical review and approval were waived for this study, Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The study did not report any data.

Conflicts of Interest: Authors declare no conflict of interest.

References

1. Aryan, R.; Yazidi, A.; Engelstad, P.E.; Kure, Ø. A general formalism for defining and detecting openflow rule anomalies. In Proceedings of the 2017 IEEE 42nd Conference on Local Computer Networks (LCN), Singapore, 9–12 October 2017; pp. 426–434.
2. Chao, C.S.; Liu, A.C. An internet firewall policy verification system. In Proceedings of the 9th Asia-Pacific Network Operations and Management Symposium, Busan, Korea, 27–29 September 2006; Volume 1, pp. 364–374.
3. Chandre, P.R.; Surve, R.R.; Badhan, S.R.; Surve, A.B.; Mane, V.T. Anomalies of Firewall Policy Detection and Resolution. *Int. J. Eng. Res. Appl.* **2014**, *I*, 696–701.
4. Chao, C.S.; Yang, S.J. A Novel Mechanism for Anomaly Removal of Firewall Filtering Rules. *J. Internet Technol.* **2020**, *21*, 949–957.

5. Penmatsa, R.K.V.; Vatsavayi, V.K.; Samayamantula, S.K. Ant colony optimization-based firewall anomaly mitigation engine. *SpringerPlus* **2016**, *5*, 1–32. [CrossRef] [PubMed]
6. Hu, H. Assurance Management Framework for Access Control Systems. Ph.D Thesis, Arizona State University, Tempe, AZ, USA, July 2012.
7. Kim, S.; Lee, H. Classifying Rules by In-out Traffic Direction to Avoid Security Policy Anomaly. *KSII Trans. Internet Inf. Syst.* **2010**, *4*, 671–690. [CrossRef]
8. Al-Shaer, E.; Hamed, H.; Boutaba, R.; Hasan, M. Conflict classification and analysis of distributed firewall policies. *IEEE J. Sel. Areas Commun.* **2005**, *23*, 2069–2084. [CrossRef]
9. Vanikalyani, G.; Avinash, P.; Pandarinath, P. Cross-domain search for policy anomalies in firewall. *Int. J. Comput. Appl.* **2014**, *104*, 20–24. [CrossRef]
10. Al-Shaer, E.; Hamed, H. Design and Implementation of Firewall Policy Advisor Tools. DePaul University, CTI. Tech. Rep. 2002. Available online: http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.134.3344&rep=rep1&type=pdf (accessed on 15 October 2021).
11. Taibah, M.; Al-Shaer, E.; Hamed, H. Dynamic Response in Distributed Firewall Systems. Technical Report, DePaul CTI Technical Report, CTI-TR-05-002. 2004. Available online: http://facweb.cs.depaul.edu/research/TechReports/TR05-002.pdf (accessed on 15 October 2021).
12. Al-Shaer, E.S.; Hamed, H.H. Firewall Policy Advisor for Anomaly Detection, Rules Editing and Translation. *Int. Symp. Integr. Netw. Manag.* **2003**, *118*, 17–30.
13. Clark, P.G. Firewall Policy Diagram: Novel Data Structures and Algorithms for Modeling, Analysis, and Comprehension of Network Firewalls. Ph.D. Thesis, University of Kansas, Lawrence, KS, USA, 2013.
14. Geeringh, C. Generic Firewall Rule Compiler And Modeller. Master's Thesis, Napier University, Edinburgh, UK, 2007.
15. Al-Shaer, E.S.; Hamed, H.H. Modeling and management of firewall policies. *IEEE Trans. Netw. Serv. Manag.* **2004**, *1*, 2–10. [CrossRef]
16. Zhang, Y.; Zhang, Y.; Wang, W. Optimization of Firewall Filtering Rules by a Thorough Rewriting. In Proceedings of the 4th Latin American Network Operations and Management Symposium, Porto Alegre, Brazil, 29–31 August 2005; pp. 77–88.
17. Tran, T.; Al-Shaer, E.S.; Boutaba, R. PolicyVis: Firewall Security Policy Visualization and Inspection. In Proceedings of the 21st conference on Large Installation System Administration Conference, Dallas, TX, USA, 11–16 November 2007; Volume 7, pp. 1–16.
18. Thwin, L.W.; Aye, Z.M. Classification and Discovery on Intra-Firewall Policy Anomalies. *Natl. J. Parallel. Soft Comput.* **2019**, *1*, 235–242.
19. Ahmed, Z.; S Askari, S.M. Firewall Rule Anomaly Detection: A Survey. *Int. J. Comput. Intell. IoT* **2018**, 6 pages.
20. Al-Shaer, E.; Al-Haj, S. FlowChecker: Configuration analysis and verification of federated OpenFlow infrastructures. In Proceedings of the 3rd ACM Workshop on Assurable and Usable Security Configuration, Chicago, IL, USA, 4 October 2010; pp. 37–44.
21. Kim, S.; Lee, H. Abnormal policy detection and correction using overlapping transition. *IEICE Trans. Inf. Syst.* **2010**, *93*, 1053–1061. [CrossRef]
22. Brand, M. A Comprehensive Firewall Testing Methodology. 12-04-2007. Edith Cowan University, Research Online. Available online: https://ro.ecu.edu.au/cgi/viewcontent.cgi?article=1021&context=ism (accessed on 15 October 2021).
23. Diekmann, C.; Naab, J.; Korsten, A.; Carle, G. Agile network access control in the container age. *IEEE Trans. Netw. Serv. Manag.* **2018**, *16*, 41–55. [CrossRef]
24. Karoui, K.; Ftima, F.B.; Ghezala, H.B. A multi-agent framework for anomalies detection on distributed firewalls using data mining techniques. In *Data Mining and Multi-Agent Integration*; Springer: Berlin/Heidelberg, Germany, 2009; pp. 267–278.
25. Basile, C.; Lioy, A. Analysis of application-layer filtering policies with application to HTTP. *IEEE/ACM Trans. Netw.* **2013**, *23*, 28–41. [CrossRef]
26. Abedin, M.; Nessa, S.; Khan, L.; Al-Shaer, E.; Awad, M. Analysis of firewall policy rules using traffic mining techniques. *Int. J. Internet Protoc. Technol.* **2010**, *5*, 3–22. [CrossRef]
27. Aryan, R.; Yazidi, A.; Engelstad, P.E. An incremental approach for swift openflow anomaly detection. In Proceedings of the 2018 IEEE 43rd Conference on Local Computer Networks (LCN), Chicago, IL, USA, 1–4 October 2018; pp. 502–510.
28. Akiki, M. An Integrated Framework for Firewall Testing and Validation. Ph.D. Thesis, Concordia University, Montreal, QC, Canada, 2009.
29. Aryan, R.; Yazidi, A.; Kure, Ø.; Einar Engelstad, P. A parallel approach for detecting OpenFlow rule anomalies based on a general formalism. *Concurr. Comput. Pract. Exp.* **2020**, *33*, e5907. [CrossRef]
30. Alfaro, J.G.; Boulahia-Cuppens, N.; Cuppens, F. Complete analysis of configuration rules to guarantee reliable network security policies. *Int. J. Inf. Secur.* **2008**, *7*, 103–122. [CrossRef]
31. Al-Shaer, E.S.; Hamed, H.H. Discovery of policy anomalies in distributed firewalls. In Proceedings of the IEEE Infocom 2004, Hong Kong, China, 7–11 March 2004; Volume 4, pp. 2605–2616.
32. Gobjuka, H.; Ahmat, K.A. Fast and scalable method for resolving anomalies in firewall policies. In Proceedings of the 2011 IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS), Shanghai, China, 10–15 April 2011; pp. 828–833.
33. Hanamsagar, A.; Jane, N.; Borate, B.; Wasvand, A.; Darade, S. Firewall anomaly management: A survey. *Int. J. Comput. Appl.* **2014**, *105*, 1–5.

34. Al-Shaer, E.S.; Hamed, H.H. Firewall policy advisor for anomaly discovery and rule editing. In *International Symposium on Integrated Network Management*; Springer: Berlin/Heidelberg, Germany, 2003; pp. 17–30.
35. Arthur, J.K.; Kwadwo, E.; Doh, R.F.; Mantey, E.A. Firewall Rule Anomaly Detection and Resolution using Particle Swarm Optimization Algorithm. *Int. J. Comput. Appl.* **2019**, *975*, 8887.
36. Trabelsi, Z.; Zeidan, S.; Shuaib, K.; Salah, K. Improved session table architecture for denial of stateful firewall attacks. *IEEE Access* **2018**, *6*, 35528–35543. [CrossRef]
37. Hwang, J.; *Improving the Quality of Security Policies*; ProQuest Dissertations Publishing: Discover, NC, USA, 2014; p. 3584006.
38. Khummanee, S. IP Packing Technique for High-speed Firewall Rule Verification. *J. Internet Technol.* **2019**, *20*, 1737–1751.
39. Tran, T. Misconfiguration Analysis of Network Access Control Policies. Master's Thesis, University of Waterloo, Waterloo, ON, Canada, 2009.
40. Ftima, F.B.; Karoui, K.; Ghezala, H.B. Misconfigurations discovery between distributed security components using the mobile agent approach. In Proceedings of the 11th International Conference on Information Integration and Web-Based Applications & Services, Kuala Lumpur, Malaysia, 14–16 December 2009; pp. 663–668.
41. Li, G.; Zhou, H.; Feng, B.; Li, G.; Zhang, H.; Hu, T. Rule anomaly-free mechanism of security function chaining in 5g. *IEEE Access* **2018**, *6*, 13653–13662. [CrossRef]
42. Yuan, L.; Chen, H.; Mai, J.; Chuah, C.N.; Su, Z.; Mohapatra, P. Fireman: A toolkit for firewall modeling and analysis. In Proceedings of the 2006 IEEE Symposium on Security and Privacy (S&P'06), Berkeley/Oakland, CA, USA, 21–24 May 2006; p. 15.
43. Khummanee, S. The semantics loss tracker of firewall rules. *International Conference on Computing and Information Technology*; Springer: Berlin/Heidelberg, Germany, 2018; pp. 220–231.
44. Bandara, A.K.; Kakas, A.; Lupu, E.C.; Russo, A. Using argumentation logic for firewall policy specification and analysis. In *International Workshop on Distributed Systems: Operations and Management*; Springer: Berlin/Heidelberg, Germany, 2006; pp. 185–196.
45. Golnabi, K.; Min, R.K.; Khan, L.; Al-Shaer, E. Analysis of firewall policy rules using data mining techniques. In Proceedings of the 2006 IEEE/IFIP Network Operations and Management Symposium NOMS, Vancouver, BC, Canada, 3–7 April 2006; pp. 305–315.
46. Khummanee, S.; Chomphuwiset, P.; Pruksasri, P. Decision Making System for Improving Firewall Rule Anomaly Based on Evidence and Behavior. *Adv. Sci. Technol. Eng. Syst. J.* **2020**, *5*, 505–515. [CrossRef]
47. Khatkar, P.K. Firewall Rule Set Analysis and Visualization. Ph.D. Thesis, Arizona State University, Tempe, AZ, USA, 2014.
48. Hu, H.; Ahn, G.J.; Kulkarni, K. Discovery and resolution of anomalies in web access control policies. *IEEE Trans. Dependable Secur. Comput.* **2013**, *10*, 341–354. [CrossRef]
49. Lord, A. An Optimization Tool For Firewall Rule Management. Master's Thesis, Valley View University, Oyibi, Ghana, 2020.
50. RADOMSKIY, S. Security Policy Rules Optimization and Its Application to the Iptables Firewall. Master's Thesis, University of Tampere, Tampere, Finland, 2011.
51. Chao, C.S. A flexible and feasible anomaly diagnosis system for internet firewall rules. In Proceedings of the 2011 13th Asia-Pacific Network Operations and Management Symposium, Taipei, Taiwan, 21–23 September 2011; pp. 1–8.
52. Khummanee, S.; Khumseela, A.; Puangpronpitag, S. Towards a new design of firewall: Anomaly elimination and fast verifying of firewall rules. In Proceedings of the 2013 10th International Joint Conference on Computer Science and Software Engineering (JCSSE), Khon Kaen, Thailand, 29–31 May 2013; pp. 93–98.
53. Hu, H.; Ahn, G.J.; Kulkarni, K. Anomaly discovery and resolution in Web access control policies. In Proceedings of the 16th ACM Symposium on Access Control Models and Technologies, Hong Kong, China, 22–24 March 2011; pp. 165–174. [CrossRef]
54. Ahn, G.J. *Assured Resource Sharing in Ad-hoc Collaboration*; Technical Report; Arizona State University: Tempe, AZ, USA, 2015.
55. Darade, R.; Kumbharkar, P. Firewall policy anomaly detection and resolution. *Int. J. Adv. Comput. Technol.* **2015**, *3*, 879–883.
56. Unde, M.M.; Khiani, S. A novel technique for effective optimization of cross domain network protocol for redundancy removal in firewall policies. *Int. J. Comput. Appl.* **2015**, *122*, 16–21.
57. Yan, L. Applying Model Checking to Pervasive Computing Systems. Ph.D. Theses, National University of Singapore, Singapore, 2014.
58. Mues, C.; Vanthienen, J. Efficient rule base verification using binary decision diagrams. In *International Conference on Database and Expert Systems Applications*; Springer: Berlin/Heidelberg, Germany, 2004; pp. 445–454.
59. Chowdhary, A.; Alshamrani, A.; Huang, D. SUPC: SDN enabled universal policy checking in cloud network. In Proceedings of the 2019 International Conference on Computing, Networking and Communications (ICNC), Honolulu, HI, USA, 18–21 February 2019; pp. 572–576.
60. Reaz, R. Theory and Practice of Firewall Outsourcing. Ph.D. Thesis, University of Texas, Austin, TX, USA, 2020.
61. Wong, E.G. *Validating Network Security Policies via Static Analysis of Router ACL Configuration*; Technical Report; Naval Postgraduate School: Monterey, CA, USA, 2006.
62. Baumeister, J.; Seipel, D. Verification and refactoring of ontologies with rules. *International Conference on Knowledge Engineering and Knowledge Management*; Springer: Berlin/Heidelberg, Germany, 2006; pp. 82–95.
63. Alicea, M.; Alsmadi, I. Mis-Configurations in Network Security Access Controls. 2021. Available online: https://ssrn.com/abstract=3808732 (accessed on 20 March 2021).

64. Chen, Y.; Feng, C.; Zhang, Q.; Tang, C. Negative selection algorithm with variable-sized r-contiguous matching rule. In Proceedings of the 2010 IEEE International Conference on Progress in Informatics and Computing, Shanghai, China, 10–12 December 2010; Volume 1, pp. 150–154.
65. Chao, C.S. A Feasible Anomaly Diagnosis Mechanism for Stateful Firewall Rules. In Proceedings of the 2018 27th International Conference on Computer Communication and Networks (ICCCN), Hangzhou, China, 30 July–2 August 2018; pp. 1–2.