5-2023

Realtime In-Network Cyberattack Detection in Power Grid Systems using a Programmable Network

Luke Waind

Realtime In-Network Cyberattack Detection in Power Grid Systems using a Programmable Network

An Undergraduate Honors College Thesis

in the

Department of Computer Science and Computer Engineering
College of Engineering
University of Arkansas
Fayetteville, AR
May, 2023

by

Luke Waind

Abstract

Power grid communication networks are important systems to detect intrusions from an attacker due to them being necessary to maintain critical infrastructure. This thesis applies recent advancements in P4 technology to detect cyberattacks in SCADA systems. In previous work, a list has been compiled of potential attacks that exploit one of the most common protocols in SCADA systems, DNP3. Solutions for detecting these attacks can be categorized by the broad methods that they use. The two methods that are focused on are single-packet inspection and multiple-packet inspection. For each of these, a specific attack is chosen and a detection algorithm is developed. These attacks are the length overflow attack and the outstation write attack. The detection algorithm for these attacks can act as an example of the methods that they were chosen for. For one of these attacks, the outstation write attack, the effectiveness of the algorithm is evaluated. This is done in a simulated network using a network simulation tool called Mininet, and a virtual attack scenario is created. When the detection algorithm detects a malicious packet, it is simply dropped. This algorithm is compared to a simple forwarding program to determine its effectiveness in preventing the attacker's desired effect on the network. The results show that the attack is effective at dropping malicious traffic in the network, making the attack unsuccessful.

THESIS DUPLICATION RELEASE

I hereby authorize the University of Arkansas Libraries to duplicate this thesis when needed for research and/or scholarship.

Agreed_____

Luke Waind

Refused_____

Luke Waind

TABLE OF CONTENTS

LIST OF FIGURES

1 Introduction

Power grid systems play a vital role in our society's infrastructure, powering other critical infrastructure systems such as hospitals and water supply that society uses daily. The effects of a power outage can be catastrophic, causing large geographic areas to lose power or causing power loss for long periods. Unfortunately, the importance placed on power grid systems can make them a target for attackers. One example of a targeted attack on a nation's power grid occurred in 2015 when a Russian group attacked the Ukrainian power grid and caused a power outage [1]. One of the potential avenues for an attacker to cause an outage is through its communication network. An important measure to stop this type of threat is using an intrusion detection system (IDS) to detect intrusions from attackers.

This thesis aims to apply advances in P4 technology to create an IDS for SCADA systems that benefit from the advantages of using P4. Some of the benefits that P4 provides that other solutions do not are line-rate speeds, protocol independence, and the use of in-network solutions. Work has been done to identify possible attacks that exploit one of the most commonly used protocols in SCADA systems, DNP3. Each of these attacks requires a different method to detect them, and these methods can be categorized into broad categories including single-packet inspection, multiple-packet inspection, communication with the SDN controller, and timing-based detection. From these categories, this thesis focuses on single-packet inspection and multiple-packet inspection. To provide a practical application of these approaches to detection, a specific attack is chosen for each of the two detection methods and an algorithm for P4 switches is created to detect each of them. The selection of which attacks to develop detection algorithms for is made by choosing attacks whose detection algorithms are fitting examples of the methods that the attacks were selected for. For single-packet inspection,

the length overflow attack has been selected. For multiple-packet inspection, the outstation write attack has been selected. The algorithm that is developed for detecting the outstation write attack is evaluated for its effectiveness in preventing the attacker's desired effect. This is done by simulating a network in which the attack is ongoing and measuring if the attacker's desired effect is present using a simulator called Mininet. During the experiment, packets that are detected as malicious are simply dropped. Other possibilities for countermeasures that can be used to prevent attacks after their detection are discussed but are not the focus of the work.

2 Background

Power grid networks use a Supervisory Control and Data Acquisition (SCADA) system. This is a system designed to collect data gathered by sensor data and aggregate them to a centralized location known as the control center. The data is collected by field devices located in stations called outstations. The data is then aggregated in intermediate devices called data aggregators before finally being collected in the control center. The control center is also responsible for sending commands to the outstations that control their operation.

SCADA systems have some requirements that are more strict than most networks. They rely on real-time delivery of data to allow the control center to make decisions quickly. Packet delay should be kept as low as possible. They are also sensitive to modified data. It is dangerous to allow an attacker to modify data that is used by the control center to make decisions. It is important to detect attacks quickly so that attackers have less of an opportunity to send modified data to the control center.

SCADA systems use special protocols to handle communications such as data transfer and commands. This thesis focuses on vulnerabilities regarding the Distributed Network Protocol 3 (DNP3), which is one of the most abundant protocols found in a SCADA network. DNP3 is used in TCP packets starting at layer 4. The first layer of DNP3 is the transport layer which states the purpose of the packet. If the packet contains data, the DNP3 application layer specifies the purpose and format of the data. DNP3 can send multiple values in a single packet in the form of objects. Each object has an object header that specifies the object type.

P4 is a new development in network hardware technology that allows network switches to be programmed and updated dynamically to dictate the flow of packets in the data plane [2]. P4 switches are designed to work in the context of a

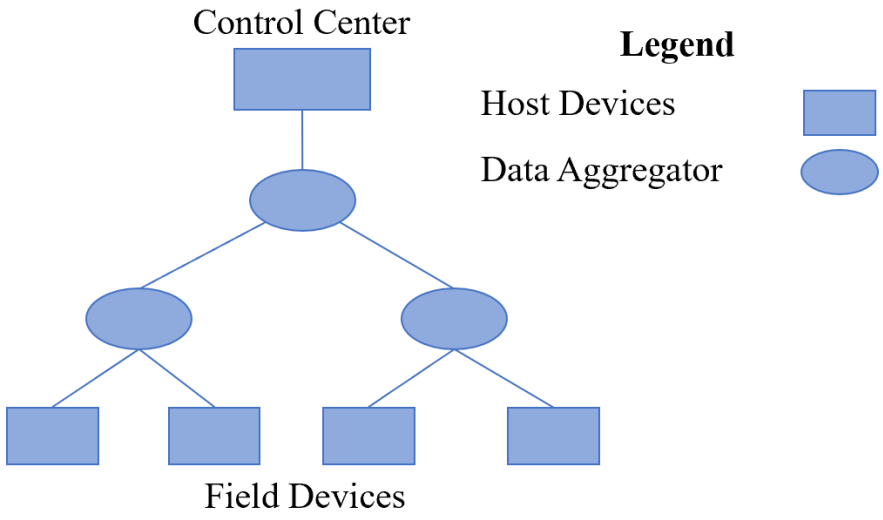

Figure 2.1: Simplified SCADA topology

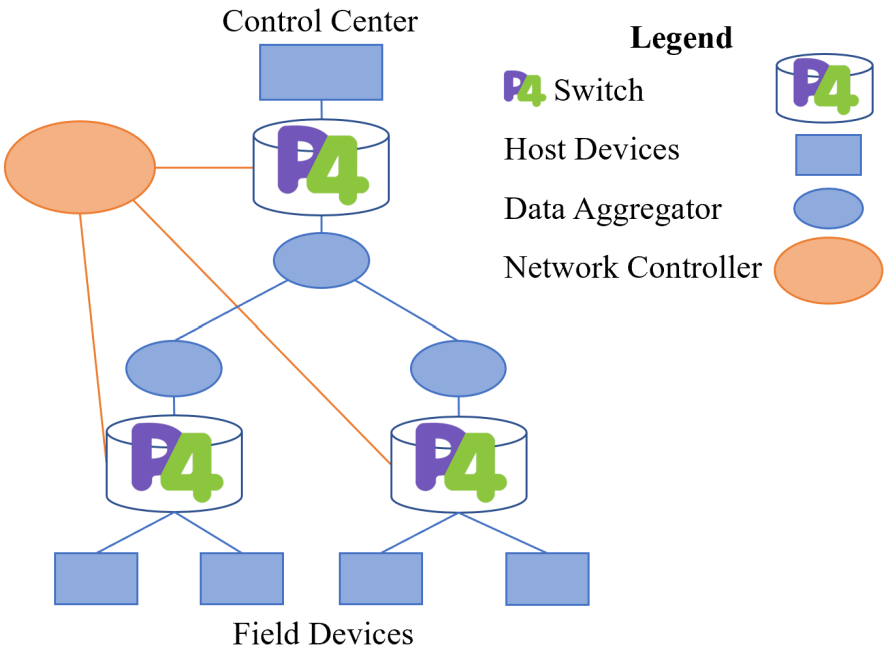

Figure 2.2: P4 switches inserted into SCADA topology

software-defined network (SDN) with a network controller that provides programs for the switches to install. With the introduction of P4, much work has been done using it to provide solutions to cybersecurity problems including IDSs.

Programming a switch to scan packets means that packets do not have to be scanned on a host machine, which typically runs on a CPU or GPU. When a high amount of traffic is being received by a host, it is possible that the host is not able to scan every packet with such hardware and must use sampling techniques. Switches can process packets at line rate speeds, which means that packet scanning can be done in the network instead.

P4 is protocol independent, which means that it can parse and manipulate packet data arbitrarily. Traditionally, packet fields could only be manipulated once vendors create hardware that can parse them. This is advantageous because lesser-used protocols often have little support from chip vendors, so without P4, there would be no way to scan fields from lesser-used protocols. This also means that P4 has the capability of scanning packet payloads rather than just packet headers. This is important because SDN protocols such as OpenFlow [3] are not able to access header fields from the DNP3. It is not a common enough protocol to implement.

The logic of a P4 switch can be dynamically changed during runtime without causing a delay in packet processing. This could allow network operators to design a system that can change dynamically in the presence of an attack. P4 switches can send information to the SDN controller to allow it to process that information and update the network with new rules for the switches.

2.1 Related Works

There are existing solutions to IDSs for SCADA networks such as Snort [4] or Zeek [5]; however, these solutions do not work within the network and create an extra step for the packets to pass through before their destination. These solutions can affect the packet delay of packets as each one needs to be scanned by the IDS.

Another consideration is that software-based IDSs such as these run on general-purpose hardware. General-purpose hardware is not fast enough to achieve packet scanning at line-rate speeds.

Work has been done on other detection methods using P4, such create a 2-layer approach in which P4 switches can send information to the SDN controller for further analysis [6]. This approach uses P4 switches as a first pass for packets and uses the SDN controller for more complex and time-consuming operations. P4 technology has also been used to implement intrusion detection systems in other contexts such as Internet of Things networks [7]. For each type of network, there are different considerations to be made due to the difference in topology, protocols used, and timing required for the network to operate.

3 Approach

Using P4, various techniques can be used to detect cyberattacks. Examples of such methods are single-packet inspection, multiple-packet inspection, communication with the SDN controller, and timing-based detection. This thesis highlights two of these methods of detection. To prove that these methods are possible using P4 in the context of a smart power grid system, a specific attack will be chosen for each method and the chosen attack for each method will be detected using that method. The two methods that are focused on are:

- Single-packet inspection – This is accomplished by scanning a packet and deciding if that individual packet is an attack. This is done with no context collected from other packets in the network and requires a high level of granularity to scan each packet individually.

- Multiple-packet inspection – This is accomplished by scanning individual packets in the same way that single-packet inspection does, but it also allows context gathered from previous packets that have passed through the switch. This requires that the switch stores information between processing packets and allows for a lower level of granularity.

Previous work on vulnerabilities in DNP3 has resulted in the compilation of a list of possible attacks that exploit DNP3. Some attacks include the false data injection attack [8], outstation application termination [9], fake outstation event buffer overflow [10], length overflow attack [9], and outstation write attack [9]. The attacks that are selected come from this list.

The two attacks that are selected are the length overflow attack [9] and the outstation write attack [9]. The length overflow attack is detected using single-packet inspection. An outstation write attack is detected using multiple-packet inspection. The details of these attacks follow.

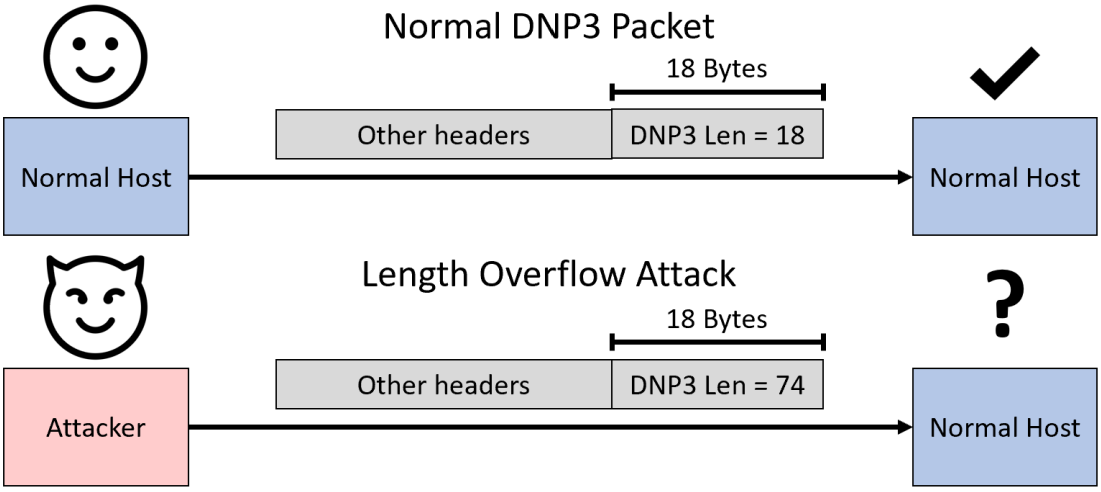

Figure 3.1: Length Overflow Attack Visualization

The length overflow attack is achieved by modifying the length field in the DNP3 length field to be much larger than its true length. This can cause an overflow in devices that attempt to parse the DNP3 payload and potentially cause a system crash. This attack is detectable using single-packet inspection. A P4 switch can compare the DNP3 length field to the true length of the packet and raise an alert if the values are not equal.

An outstation write attack is a form of denial-of-service attack in which a compromised outstation sends a high volume of write packets to the control center. The packets can fill the buffer of the control center that is storing the data, which can cause legitimate write traffic to be dropped. This attack is detectable using multiple-packet inspection. To detect this attack, a P4 switch must use context from previous packets, in this case, the arrival times of previous packets, to determine if the current packet is part of an ongoing attack. If many write packets pass through the switch in a short time, an alert is raised for this attack.

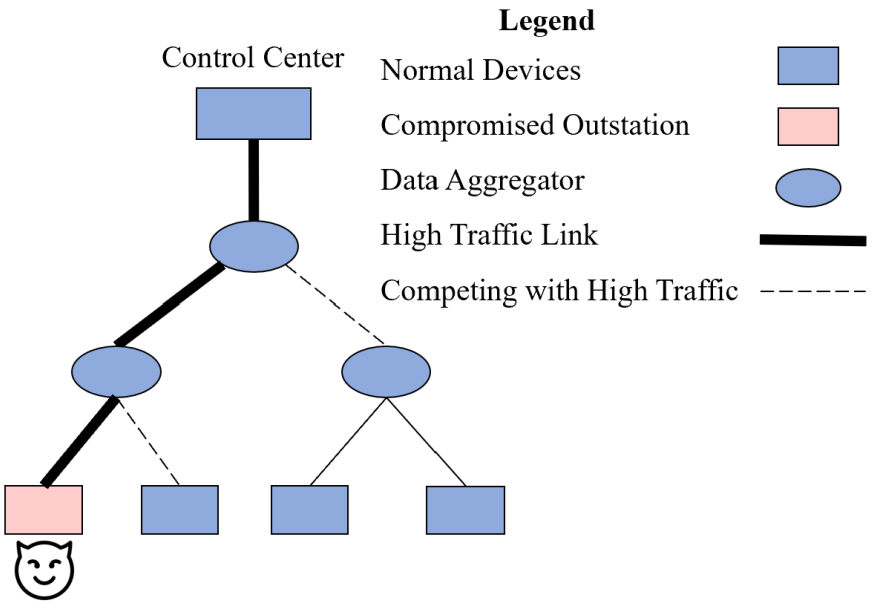

Figure 3.2: Outstation Write Attack Visualization

3.1 Length Overflow Attack

To detect a length overflow attack two values, are compared: the DNP3 length field and the true length of the DNP3 payload. Obtaining the DNP3 length field is done by parsing the DNP3 header using P4 and extracting the third byte that represents the length field. Two discrepancies prevent this extracted value from matching the true length of the DNP3 payload in bytes. The first discrepancy is that the length field only reports the length of the data following the length field. Because the length field is the third byte, adding three to the value from the length field corrects this discrepancy. The second discrepancy is that the length field excludes CRC checksum data. To account for this discrepancy, the CRC bytes must be added to the header length. In every DNP3 packet, there are two CRC bytes for the 8 bytes in the data link layer, then there are two more for every 16 bytes in the application layer rounding up to account for the last chunk which may not have a full 16 bytes. Accounting for both of these discrepancies will result in a value representing the length of the DNP3 payload in bytes as reported by the

DNP3 length field.

Obtaining the true length of the DNP3 payload cannot be obtained using P4 by measuring the length of the packet on the wire. Instead, the DNP3 payload length can be calculated by taking the total length of the packet and subtracting the length of the other layers until the DNP3 header length is left. The total length of the packet is stored in the total length field of the IP header, so the packet starts at layer 2. To subtract the IP header, the internet header length can be used. This field represents the length of the IP header in 32-bit words, so the field must be multiplied by four to obtain the number of bytes to subtract. The TCP header contains a data offset field that represents the length of the packet, also in 32-bit words. The TCP header length is subtracted by subtracting four times the value of this field. The remaining value is the true length of the DNP3 payload. To detect a length overflow attack, compare this value to the value of the DNP3 payload as reported by the DNP3 length field. If the values do not match, the packet should be flagged.

There is one consideration that must be made about calculating the true DNP3 payload length in this way. If an attacker has the ability to manipulate the DNP3 length field, then the attacker most likely has the ability to modify the other fields used to calculate the packet length. If the attacker modifies them the correct way, a malformed packet could pass this detection check. With a few assumptions it is possible to circumvent this issue. This work simulates P4 switches in software and uses the bmv2 framework to run that simulation. Bmv2 gives P4 access to metadata from packets, which includes the total length of the packet. This can be used in place of the total length field from the IP header. It would also be necessary to assume that the length of the IP header and TCP header are both 20 bytes. Using these values, the DNP3 payload length can be calculated in much the same way. Implementing a P4 program using hardware would use a different set of metadata, but the length of the packet is likely to be a part of any model's metadata.

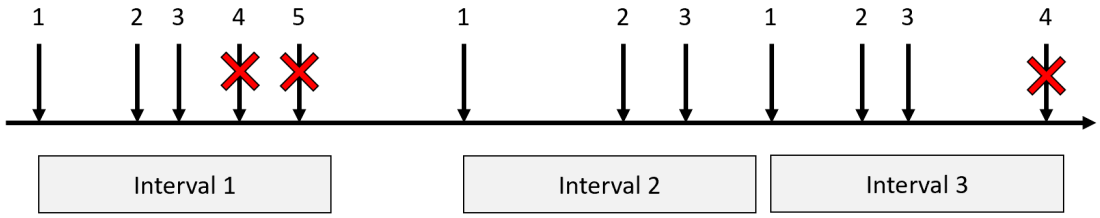

Figure 3.3: Detection Algorithm Visualization for N = 3

3.2 Outstation Write Attack

Under normal circumstances, write packets are sent in regular intervals by an outstation. The interval varies considerably depending on the configuration of the network ranging from minutes to hours. The amount of time between each packet is consistent as power grid networks are predictable by nature. In contrast, during an outstation write attack, there is a high volume of write traffic flooding the network to deny legitimate write packets from passing through the network. This large difference in traffic establishes a clear boundary between an attack flow and a normal data flow.

Because the packets that belong to an outstation write attack are indistinguishable from legitimate traffic, context collected from previously seen packets is required to detect this attack. To determine if the rate of write packets is too high, the arrival times of write packets are saved in registers to contextualize future write packets. The program follows an algorithm to determine if too many write packets have arrived in a small amount of time.

3.2.1 Detection Algorithm

The algorithm uses two parameters that can be tuned based on the needs of the network operator. These parameters are a time interval T and a threshold N. The algorithm works by counting the number of packets that have been seen within a time interval T. If the number of write packets exceeds N, that write packet and subsequent write packets will be dropped until the end of the interval.

When an interval expires, packets will be able to pass through again. When the first packet after an interval expires arrives, the packet's arrival time becomes the start of a new interval, and the count is set to one. This algorithm ensures that traffic is limited to at most N packets per T time.

To store these values, P4 can hold values in registers that retain their stored value between different packets. The values that I store are the start of the current time interval and the packet count within that interval for a total of two registers used. Every write packet causes a write to the count register. The interval start register is only updated when the first packet arrives after an interval expires.

3.2.2 Multiple Outstations

The structure of a SCADA network makes it very likely that a P4 switch will be receiving data from several outstations. This means that the algorithm described must have instances running concurrently for each outstation sending write packets to it. Instead of two registers being used, two register arrays are used with each index of the arrays corresponding to the source addresses from write packets. To map the source address onto a pair of registers from the two register arrays, CRC16 is used as a hash and the result is truncated to map to the number of registers that are being used. Running instances of the algorithm concurrently in this way allows the P4 switch to block the traffic coming from an attacker while allowing normal traffic coming from the other outstations.

4 Evaluation

Mininet network simulation tool that allows networks to be simulated with various topologies specified by the user. The links in these topologies can further be specified with more properties such as their link limit rates. The P4 solutions to length overflow attack detection and outstation write attack detection can both be evaluated in Mininet.

4.1 Attack Scenario

A simple topology where a P4 switch is connected to many outstations is used to evaluate the outstation write attack detection algorithm. Five hosts are acting as outstations in this topology that are linked to a P4 switch. A host acting as the control center is also linked to the same switch. In the attack scenario, one outstation is considered compromised by an attacker and the other four are normal. The compromised outstation sends a high volume of traffic filled with write packets. The other four outstations send a write packet every second. The value of one second was used to evaluate an extreme case of the detection algorithm. The time between write packets is normally on the order of minutes or hours, but

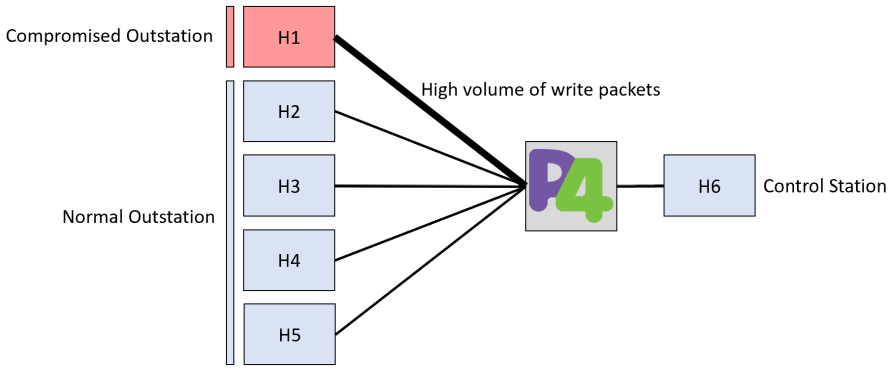

Figure 4.1: Attack Scenario Topology Diagram

detecting the attack with write packets sent every second should be more difficult than having write packets sent with any amount of time more than one second.

A problem with simulating an outstation write attack is the attack's reliance on real-time data processing. A P4 switch is a piece of specialized hardware that can perform faster than a CPU or GPU doing the same task. When a P4 switch is simulated on a CPU or GPU, its speed is greatly limited by the device it is simulated on. This means that during simulation, the switch would be the bottleneck in the network when in reality, it would never be. To combat this, the simulation is run with an amount of attack traffic that can be processed by the simulated P4 switch, which is up to 10 kbps. At this speed, the host acting as the control center can process all of the incoming data as well, so the link limit rate of the link between the control center and the P4 switch is artificially lowered to 5 kbps to simulate the control center processing data at a slower rate than the P4 switch.

For this experiment, the parameters N and T were selected and remained constant throughout the experiment. In the simulation, the normal outstations were set to send one packet per second. Because the pattern has a fixed interval and is predictable, the threshold parameter N is fixed to 1 packet. The time interval is one second, so the parameter T is set to 0.9 seconds. It is slightly lowered from the time between packets sent to account for some potential variability in the travel time of each packet. These parameters can be set by the network operator depending on the specific details for each use case.

5 Results

The data from the figure was collected by sending normal packets from normal outstations once per second, varying the higher attack rate from the compromised outstation from 0 kbps to 10 kbps. The data was only collected on normal packets using IP filtering and the value measured was the average time between when the packet was sent and when it was received. In each test, 240 normal write packets were received collectively from all four outstations, and the average delay was calculated from those packets. The test was run on two P4 programs: one that simply forwards all packets to their destination address and one that runs the outstation write attack detection algorithm.

The data shows that the average packet delay when using the detection algorithm does not change when the compromised outstation's attack rate varies. This is the expected behavior of the simulation. The detection algorithm is dropping packets from the attacker when the rate is too high, which means that normal packets do not have to compete with the attacker for resources in the control center. The data for the packet delay with basic forwarding shows that there is no delay until the attack rate is 5 kbps and then quickly increases. The reason the increase begins at 5 kbps is that the link limit rate of the link between the P4 switch and the control center is artificially set to 5 kbps to simulate processing time for the control center. The delay times plateau because packets timeout after 40 seconds, so the ones that are received arrive in less than 40 seconds. In a real scenario, one would expect to see more variation in the de due to the control center's available resources fluctuating. In this simulation, the only packets that were sent were write packets, but in a real power grid network, many more packets of different types would be flowing through the network.

The data also shows that when using the detection algorithm, the number of normal packets that the control center received in a fixed time span remained

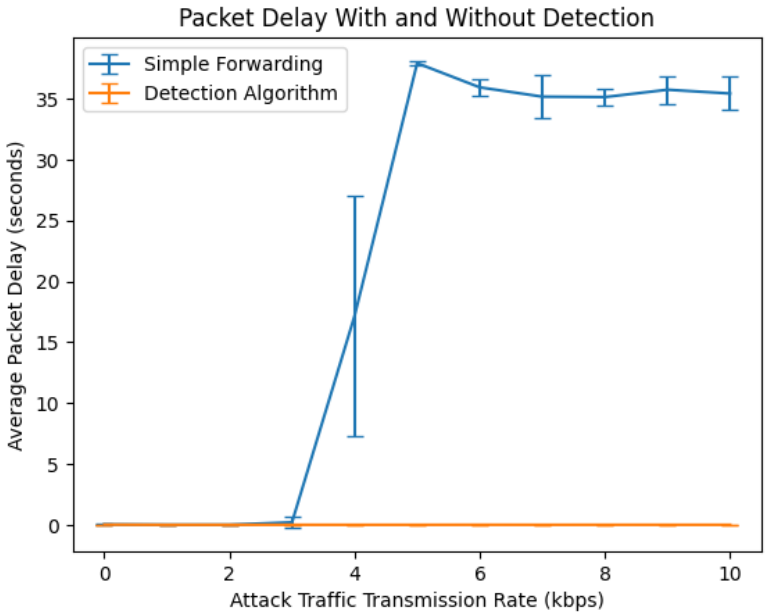

Figure 5.1: Plot of attack strength versus packet delay

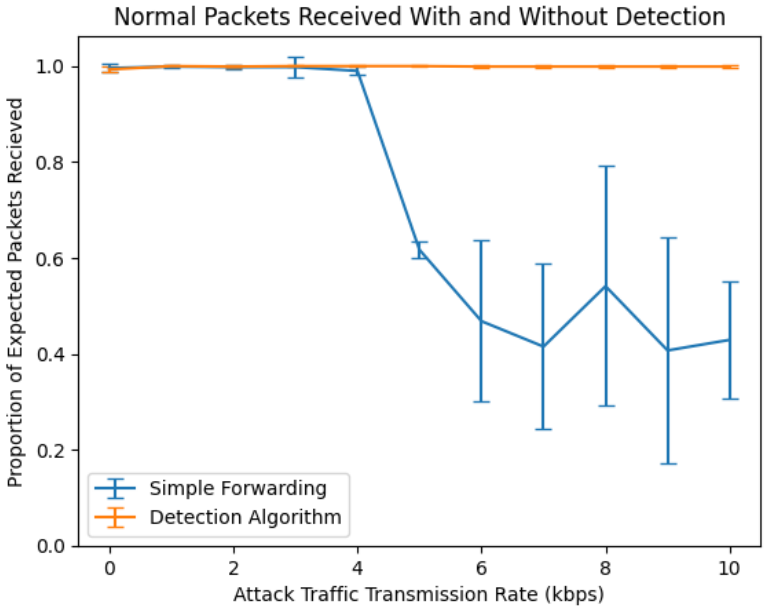

Figure 5.2: Plot of attack strength versus packets received

unchanged as the attack traffic increased. This is the expected behavior of the simulation. The detection algorithm is dropping attack traffic which prevents a denial of service that affects the ability of other outstations to send traffic to the control center. The number of normal packets received by the control center when using the basic forwarding algorithm diminishes as the attack traffic increases. This is caused by normal packets competing with the high traffic. The reason that there is high variance after the drop is because the result is highly dependent on the speed of the CPU while the test is running. This shows that the effects of the attack would have been apparent while using the detection algorithm if the algorithm was not successful. The values collected are standardized to a proportion based on the number of packets that are expected during the trial. In this case, the tests lasted 60 seconds, there are 4 outstations, and the outstations send a packet 1 time per second, so the expected number of packets to arrive is 240 packets.

6 Discussion

The success of detecting the length overflow attack and the outstation write attack highlights P4's ability to check for attacks in the network and in real-time. This is something that was not previously possible before P4's release. Denial of Service attacks such as the outstation write attack is an especially difficult type of attack to detect due to malicious traffic being indistinguishable from legitimate traffic and the very high volume of traffic. The detection of the two specific attacks can be applied more broadly to other attacks that are detectable using the single-packet inspection or multiple-packet inspection approach.

The two methods of detection shown in this thesis, single-packet inspection and multiple-packet inspection, are not the only two ways that P4 can be used to detect attacks. Because P4 builds on top of the context of an SDN, it is possible to detect attacks by sending packets to the network controller. This requires a 2-level approach [6]. The P4 switches act as a first pass that can flag packets that may be deemed as potentially malicious, then the flagged packets are sent by the switch to the SDN controller to be processed more thoroughly. This approach is beneficial because, while P4 is fast, it is only capable of simple operations such as addition, subtraction, bit manipulation, and match-action table lookups. On the SDN controller, more complex operations could be done such as division, floating point operations, and even machine learning categorization. The SDN controller processes packets much more slowly than a P4 switch, so the P4 switch must only send a small fraction of the traffic that it processes.

The data collected in this thesis is collected from a computer simulation of a network. Because of this, it is difficult to know if the P4 program that is used for the tests would be quick enough to run in a real system. The relative speed difference when running the basic forwarding P4 program and detection algorithm program are similar. The basic forwarding program is simple and would run fast if

replicated on real P4 hardware, so the detection algorithm likely would as well. The detection algorithm, however, uses register writes which may be more expensive operations on P4 hardware.

7 Conclusion

This thesis shows the potential benefits of using P4 technology in a power grid communication network's intrusion detection system as well as practical applications for detecting specific threats that can be extended to cover a variety of different threats based on the needs of the network. Not only does P4 provide a much faster solution to detecting attacks that could already be detected, but P4 also opens the door to detecting new types of attacks that other systems could not such as denial of service attacks. The data shows that after using a simple mitigation strategy, dropping traffic from the attacker, a P4 program running an algorithm that detects an outstation write attack, a kind of DoS attack, can eliminate packet delay caused by the attack.

Installing P4 switches in a smart power grid system would allow it to detect a much broader range of attacks and to detect attacks more efficiently. It would also aid in the development of new cyberattack detection algorithms. The P4 language allows for a high amount of versatility when it comes to parsing and manipulating packet data.

Bibliography

[1] CISA, "Cyber-attack against ukrainian critical infrastructure," https://www.cisa.gov/uscert/ics/alerts/IR-ALERT-H-16-056-01, 2021.

[2] P. Bosshart, D. Daly, G. Gibb, M. Izzard, N. McKeown, J. Rexford, C. Schlesinger, D. Talayco, A. Vahdat, G. Varghese, and D. Walker, "P4: Programming protocol-independent packet processors," *SIGCOMM Comput. Commun. Rev.*, vol. 44, no. 3, p. 87–95, jul 2014. [Online]. Available: https://doi.org/10.1145/2656877.2656890

[3] N. McKeown, T. Anderson, H. Balakrishnan, G. Parulkar, L. Peterson, J. Rexford, S. Shenker, and J. Turner, "Openflow: Enabling innovation in campus networks," *SIGCOMM Comput. Commun. Rev.*, vol. 38, no. 2, p. 69–74, mar 2008. [Online]. Available: https://doi.org/10.1145/1355734.1355746

[4] M. Roesch, "Snort," https://www.snort.org/.

[5] V. Paxson, "Bro: A system for detecting network intruders in real-time," *Comput. Netw.*, vol. 31, no. 23–24, p. 2435–2463, dec 1999. [Online]. Available: https://doi.org/10.1016/S1389-1286(99)00112-7

[6] B. Jiang, Y. Liu, H. Liu, Z. Ren, Y. Wang, Y. Bao, and W. Wang, "An enhanced ewma for alert reduction and situation awareness in industrial control networks¡sup¿*¡/sup¿," in *2022 IEEE 18th International Conference on Automation Science and Engineering (CASE)*. IEEE Press, 2022, p. 888–894. [Online]. Available: https://doi.org/10.1109/CASE49997.2022.9926545

[7] E. M. Campos, P. F. Saura, A. González-Vidal, J. L. Hernández-Ramos, J. B. Bernabé, G. Baldini, and A. Skarmeta, "Evaluating federated learning for intrusion detection in internet of things: Review and challenges," *Computer Networks*, vol. 203, p. 108661, 2022. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1389128621005405

[8] Y. Liu, P. Ning, and M. K. Reiter, "False data injection attacks against state estimation in electric power grids," *ACM Trans. Inf. Syst. Secur.*, vol. 14, no. 1, jun 2011. [Online]. Available: https://doi.org/10.1145/1952982.1952995

[9] S. East, J. Butts, M. Papa, and S. Shenoi, "A taxonomy of attacks on the dnp3 protocol," in *Critical Infrastructure Protection III*, C. Palmer and S. Shenoi, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2009, pp. 67–81.

[10] C. Irvene, T. Shekari, D. Formby, and R. Beyah, "If i knew then what i know now: On reevaluating dnp3 security using power substation traffic," in *Proceedings of the Fifth Annual Industrial Control System Security (ICSS) Workshop*, ser. ICSS. New York, NY, USA: Association for Computing Machinery, 2019, p. 48–59. [Online]. Available: https://doi.org/10.1145/3372318.3372324

[11] J. Hypolite, J. Sonchack, S. Hershkop, N. Dautenhahn, A. DeHon, and J. M. Smith, "DeepMatch: practical deep packet inspection in the data plane using network processors," in *Proceedings of the 16th International Conference on emerging Networking EXperiments and Technologies*. New York, NY, USA: ACM, Nov. 2020.

[12] B. Lewis, M. Broadbent, and N. Race, "P4ID: P4 enhanced intrusion detection," in *2019 IEEE Conference on Network Function Virtualization and Software Defined Networks (NFV-SDN)*. IEEE, Nov. 2019.