Dynamic Data Driven Security Framework for Industrial Control Networks using Programmable Switches

Reuben Samson Raj and Dong Jin

University of Arkansas, Fayetteville AR USA
{rs077, dongjin}@uark.edu

Abstract. Cyber-security for Industrial Control Systems (ICS) such as SCADA systems has been an important avenue of research over the last couple of decades. The need for secure ICS systems stems from several factors - legacy protocols, limited segmentation between operational technology (OT) and information technology (IT) networks, and increasing connectivity of ICS devices to corporate networks and the Internet, among others. Given the unique challenges to ICS security and the unique characteristics of ICS systems and network traffic, we propose a dynamic data-driven security framework utilizing P4 programmable switches. Our proposed framework consists of a switch-controller feedback loop mechanism that enables real-time cyber attack detection and mitigation. The P4 switch with its custom packet processing capabilities, generates statistics and metrics based on network traffic patterns. The controller employs these insights to further detect and mitigate attacks by updating forwarding rules on the switch in real time. Our prototype system of a Denial-of-Service defense on a Modbus network demonstrates the promising potential of P4-based DDDAS towards real-time cyber-defense for ICS networks.

Keywords: DDDAS · Dynamic Data Driven Applications Systems · InfoSymbiotic Systems · Industrial Control Systems · Programmable Switches

1 Introduction

Industrial Control Systems (ICS), such as Supervisory Control and Acquisition of Data (SCADA), are used by a significant majority (80%-90%) of utility operators [1,4]. These systems typically employ legacy protocols, such as Modbus, DNP3, or IEC 60870-5. These protocols were originally designed for operation in isolation without the need for interaction with external Wide Area Networks (WANs) or the Internet. However, the recent modernization of ICS systems into highly interconnected systems has rendered them vulnerable to evolving cyber-attacks [12,16]. Although upgrading the legacy devices to support advanced protocols involving security mechanisms, such as authentication and encryption, seems like a logical choice, there are constraints to this approach. The host-based security approach is thus problematic due to both operator hesitancy in upgrading and resource constraints of legacy devices.

Given the limitations of host-based security for ICS networks, the responsibility thus shifts to the network. Communication networks can be broadly separated across two levels of abstraction - the control plane (dealing with management of the network devices and forwarding rules) and the data plane (actual forwarding of data). While Software-Defined Networking (SDN) enabled a programmable control plane, P4 (Programming Protocol-Independent Packet Processors) is a recent network technology that enables a programmable data plane [9].

A P4 network switch comprises a programmable data plane based on unique application-specific integrated circuits (ASICs) and a programmable control plane based on a general-purpose CPU environment. This hybrid architecture enables a Dynamic Data-Driven Application Systems (DDDAS) approach to develop a security framework that enhances the attack resilience of industrial control networks in power systems. DDDAS is a paradigm that involves dynamically incorporating real-time data into computations to guide the measurement and control processes of application systems [10], and has been applied in power grid analysis, cyber security, and many other applications [8,11,14,7,20]. In our P4-based security framework, the data plane generates real-time and dynamic data-driven stats and metrics, which are then analyzed using advanced data analytics and learning algorithms on the control plane to detect attacks. The control plane then steers the measurement process in a feedback control loop by installing network updates on the match-action pipelines of the data plane to mitigate attacks.

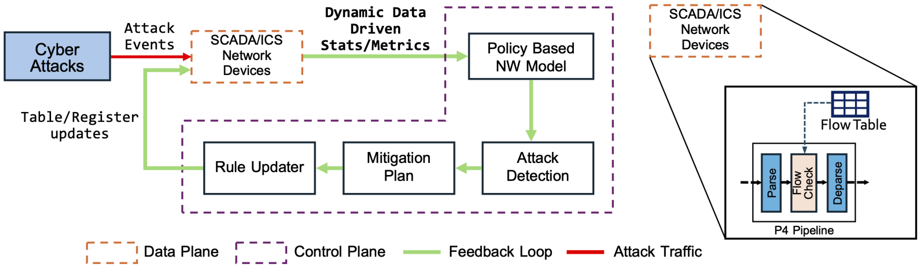

Fig. 1: ICS Security Architecture Using P4-based DDDAS

Studies on network traffic for Operational Technologies (OT) such as ICS [5,6] have shown that OT traffic differs from typical IT traffic. ICS traffic is more predictable, has a high level of periodicity, and often uses simple, proprietary protocols. These characteristics prove advantageous as they enable us to define security policies that can be centrally managed by a controller, and implemented on the P4 switch. Given the stability of network topology, known communication paths, and traffic patterns, the controller can maintain a constant, global security model of the network.

Given the benefits of the DDDAS paradigm, P4 switch capabilities, and ICS traffic characteristics, we thus explore a P4-based DDDAS system for ICS Security. Our proposed framework is a data-driven feedback loop system comprising of (i) high-speed programmable switches that can parse custom protocols for ICS networks, generate dynamic data, and (ii) a controller that can run applications based on these data, and accordingly update the forwarding or processing behavior of the switches (network data plane). Fig. 1 shows our proposed architecture. The switch data plane (orange dotted box) detects any cyber attack events in the form of anomalous traffic. P4's custom processing capabilities generate key statistics and metrics about the traffic, which are then sent to a controller application. The controller performs further analysis on attack detection and mitigation plans and arrives at a new forwarding behavior, which is then installed in the form of rules onto the switch.

Most network-based conventional ICS security approaches such as Intrusion Detection/Prevention Systems (ID/PS) involve a single point of deployment, usually at points of traffic aggregation [13,15,22]. On the other hand, programmable network approaches such as SDN and P4 are often distributed, providing a wider defense perimeter. The works in [21,19] are SDN-based, while the work in [18] is 2-level P4-based IDS for Modbus TCP, with one level handling flow, application level checks, and a controller that performs Deep Packet Inspection and updates flow entries in the switch.

The remainder of the paper is organized as follows: In Section II, we present the general framework of our proposed design. In Section III, we present a proof-of-concept demonstration of a P4-based DDDAS Denial-of-Service defense mechanism on a Modbus network, along with an evaluation of latency under varying combinations of network topology. Finally, in Section IV, we conclude and offer future directions for related research.

2 General Framework

Fig. 2 depicts the overall framework of our proposed P4-based DDDAS system for ICS security, with an expanded view of the architecture shown in Fig. 1. The framework consists of three main areas: the data plane, the control plane, and the interface between the two.

2.1 Data Plane

The data plane of our framework consists of four key components: the parser, match-action tables, custom processing, and dynamic data generation.

Parser: P4 provides a programmable packet parser that is flexible to parse most protocols. Given that most ICS protocols are unencrypted and follow a simple structure, P4 is an effective choice for parsing. Parsing enables the extraction of key fields of interest, which would then be used in later stages of the pipeline.

Match-Action Tables: These are a set of rules and corresponding actions. Usually, the extracted fields from the parsing stage are used to match against the table entries. For instance, an IP-forwarding table that matches a packet based on the destination IP field specifies what actions to be taken when there is a match("hit")/mismatch("miss") against a table entry. The actions could be a simple Drop or Forward, or perform some custom processing.

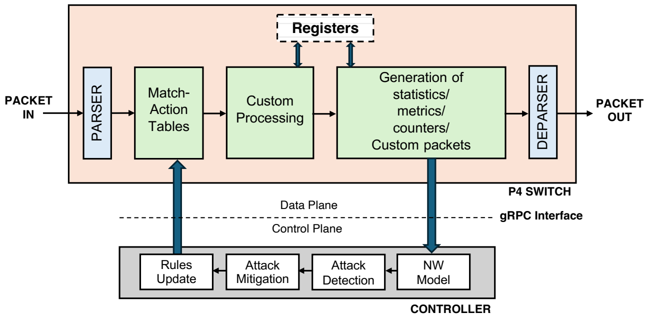

Fig. 2: General Framework for P4-based DDDAS

Custom Processing: In addition to match-action based validation, P4 can perform a wide range of custom processing involving registers, custom validations, packet modifications, etc. Index-based registers serve as quick storage and retrieval units.

Dynamic Data Generation: P4 supports the generation of custom statistics, metrics, counters, and custom packets that can be interfaced with an external controller for additional processing. In our proposed framework, these serve as dynamic data inputs to the controller for further attack detection and mitigation.

2.2 Control Plane

NW Model: The controller runs a policy-based network model of the entire ICS network under consideration. The policy encompasses a defined set of communication constraints within the network infrastructure, such as authorized host-to-host communication, allowed TCP port access, and allowable Function Code interactions. The network model can be initialized from device configurations, known traffic patterns or periodicity information, ports, or even custom logic.

Attack Detection and Mitigation: The dynamic data generated by the switch, coupled with the controller's computational capabilities, enables the detection of various ICS-specific attacks. Subsequently, the controller formulates rapid mitigation plans by deploying new forwarding rules to the switch.

2.3 Switch-Controller Interface

The controller communicates with the switch data plane over a gRPC interface. The controller can issue API calls to the switch to update table entries and read or write onto registers. The switch can transmit various dynamic data over this interface, including packet/byte counters, traffic flow statistics, and customizable application-level information. This information can provide insight into real-time network status, thus enabling attack detection and mitigation at the controller.

We thus have a dynamic data-driven system where the P4 switch provides a first line of cyber defense, and the controller can perform the next level of advanced detection and mitigation. In the next section, we demonstrate the implementation of such a system against a Denial-of-Service (DoS) attack on a Modbus TCP network, where a rogue host can spoof a legitimate Modbus client (master) and send Modbus request packets at a high rate, in an attempt to overwhelm Modbus server (slave) devices. DoS attacks tend to be high volume and adaptable to defenses, thus requiring real-time defenses at line rate. We would like to note that this work is ongoing, with future plans to explore a broader spectrum of attacks, including reconnaissance, response injections, and command injections. The primary objective of this paper is to present a proof-of-concept demonstration of a P4-based DDDAS for ICS networks.

3 P4-DDDAS for Modbus DoS Defense

Introduced in 1979 for utility device control in ICS networks, Modbus [2] is a simple protocol that employs a request-response model with function codes to indicate desired operations. Despite its enhancement to Modbus TCP for Ethernet and wireless networks, both protocols have significant vulnerabilities due to a lack of security mechanisms. Fig. 3 shows a Modbus Master-Slave network, where all Master (MTU, HMI) and Slave (RTU) hosts are connected via a P4 switch. The P4 switch is emulated using the BMv2 [3] software switch platform and Mininet [17]. It features local controller software capable of receiving dynamic data-driven inputs, detecting attack traffic, devising mitigation plans, and updating forwarding rules accordingly.

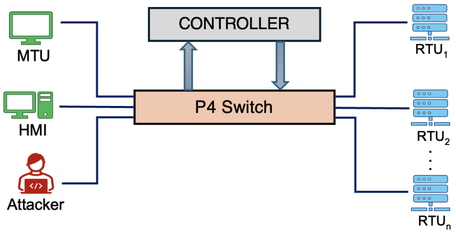

Fig. 3: P4-DDDAS Modbus Network
HMI - Human Machine Interface
MTU - Master Terminal Unit
RTU - Remote Terminal Unit

As mentioned earlier in Sec. 1, ICS traffic is mostly periodic. For instance, the work in [5] discusses the concept of "polling" in Modbus networks, where the Master periodically sends request messages to read the register contents on a Slave. In terms of Modbus semantics, this translates to - a certain Master sending a certain Function Code at a predefined periodicity to a certain Slave node. Based on this characteristic, in our case study we assume that each function code on a slave is defined by a specified periodicity. The utility operator configures each slave device with a set of supported Function Codes and corresponding periodicity values. These values represent the minimum inter-packet delay (in seconds or milliseconds) per function code per client-slave transaction.

The attack model in our demonstration involves a Modbus-level DoS, where a rogue host spoofs the Master node's IP address and continuously sends Modbus request packets to a Slave device at rates higher than the specified periodicity values. In our context, DoS attacks are primarily intended to overwhelm target slave devices, which typically have limited buffer sizes for processing incoming packets. The objective of our solution is to filter out any anomalous traffic arriving at the switch at a rate higher than the specified periodicity.

3.1 Switch and Controller Algorithm

Fig. 4 shows the sequence of operations at the switch and controller. For every Modbus request packet received, the switch performs the following steps:

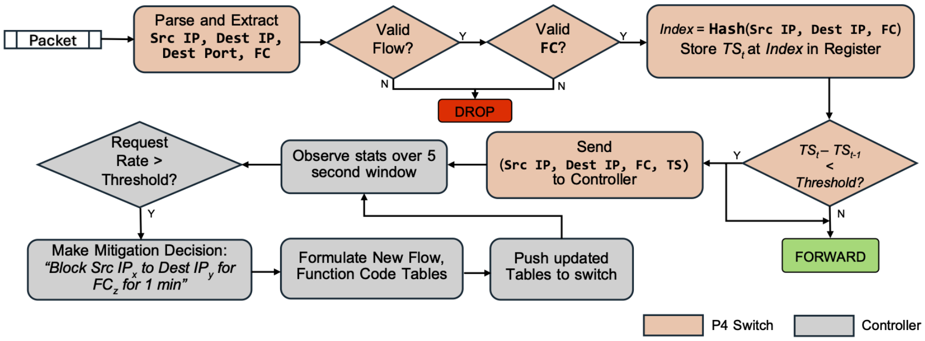
Packet
Parse and Extract
Src IP, Dest IP, Dest Port, FC
Valid Flow?
Y
N
Valid FC?
Y
N
DROP
Index = Hash(Src IP, Dest IP, FC)
Store TS_t at Index in Register
$TS_t - TS_{t-1}$ < Threshold?
Y
N
FORWARD
Send (Src IP, Dest IP, FC, TS) to Controller
Observe stats over 5 second window
Request Rate > Threshold?
Y
Make Mitigation Decision: "Block Src IP_x to Dest IP_y for FC_z for 1 min"
Formulate New Flow, Function Code Tables
Push updated Tables to switch
P4 Switch
Controller

Fig. 4: Switch-Controller Sequence for Modbus DoS Defense

- Parses and extracts key fields: Source IP address (Src IP), Destination IP address (Dest IP), Destination Port (Dest Port), and Function Code(FC).
- Checks the validity of these fields against pre-populated tables and drops the packet if there is any mismatch.
- Calculates the inter-arrival delay for packets matching the unique combination of Source-IP, Destination-IP, and Function Code.
- Formulates a 4-tuple (Src IP, Dest IP, FC, Timestamp) and sends this information to the controller if the arrival rate exceeds a preset threshold T.

At the controller:

- The controller continuously listens to incoming stats from the P4 switch over a pre-configured TCP port.
- The controller monitors stats over a 5-second moving window and compares against a threshold rate. Both the controller threshold and the switch-side threshold are configurable parameters.
- If the switch reports > 20 instances of high-rate traffic within a 5-second window for a particular 3-tuple (Src IP, Dest IP, FC), the controller identifies it as attack traffic.
- The controller formulates updated Flow and Function Code tables and pushes the table entries onto the switch.
- The switch continues to forward higher-than-threshold rate traffic until it receives the updated table rules. Upon receiving the updated rules, the switch blocks all subsequent packets that match the 3-tuple.

We evaluated the performance of our algorithm by measuring the round-trip latency between the switch and the controller. The round-trip latency encompasses the time from the switch sending 4-tuple stats to the controller, the controller performing attack detection, to the switch receiving updated blocking rules. Table 1 presents findings across various scenarios, varying the number of master nodes, slave nodes, and supported function codes per slave. The latency remains minimal (millisecond range) even with increasing complexity, ensuring scalability. In the context of Modbus systems, our focus is on preventing devices from being overwhelmed by abnormal traffic flows, swiftly identified by our P4-based in-network solution. Given the challenges in establishing accuracy metrics without a definitive ground truth, we prioritize latency evaluation as it aligns with the real-time constraints of ICS operations.

Table 1: Round-Trip Latency between Switch-Controller

# Master	# Slave	# Func. Codes	# Combinations	Avg. Latency (ms)	St. Dev (ms)
2	3	2	12	1.39	0.44
2	5	10	100	1.41	0.88
8	10	13	∼1000	2.6	1.64
21	21	25	∼11000	7.82	8.8

4 Conclusion

In this paper, we proposed a P4-based DDDAS design to secure ICS networks. Our proposed framework leverages the capabilities of both programmable switches and controller applications. Together, they establish a real-time feedback loop mechanism, enhancing the detection and mitigation of attacks. Our demonstration of a DDDAS-based DoS defense system in a Modbus ICS network highlights the effectiveness and potential of P4-based defenses against ICS-specific attacks. In the future, we aim to execute a comprehensive end-to-end attack-defense cycle on hardware P4 switches, incorporating a multi-protocol system such as Modbus, DNP3, and IEC 60870-5. In this work, we implemented rule-based attack detection and mitigation, and P4 switches can gather advanced network telemetry, enabling statistical-based methods in forthcoming research.

Acknowledgment. The authors are grateful for the support of the National Science Foundation under Grant CNS-2247721, EEC-2113903, and the Engineering Research and Innovation Seed Funding (ERISF) at the University of Arkansas. The authors also thank Dr. Yanfeng Qu for his valuable comments and feedback.

References

1. https://www.newton-evans.com/94-of-north-american-electric-utilities-surveyed-use-dnp3-for-scada/
2. https://modbus.org/docs/Modbus_Application_Protocol_V1_1b.pdf
3. https://github.com/p4lang/behavioral-model
4. Finding value in utility data (2022), `https://www.publicpower.org/periodical/article/finding-value-utility-data`
5. Barbosa, R.R.R., Sadre, R., Pras, A.: A first look into scada network traffic. In: IEEE Network Operations and Management Symposium. IEEE (2012)

6. Barbosa, R.R.R., Sadre, R., Pras, A.: Flow whitelisting in scada networks. International journal of critical infrastructure protection **6**(3-4), 150–158 (2013)
7. Blasch, E., Al-Nashif, Y., Hariri, S.: Static versus dynamic data information fusion analysis using DDDAS for cyber security trust. Procedia Computer Science (2014)
8. Blasch, E.P., Darema, F., Ravela, S., Aved, A.J.: Handbook of Dynamic Data Driven Applications Systems: Volume 1. Springer Nature (2022)
9. Bosshart, P., Daly, D., Gibb, G., Izzard, M., McKeown, N., Rexford, J., Schlesinger, C., Talayco, D., Vahdat, A., Varghese, G., et al.: P4: Programming protocol-independent packet processors. ACM SIGCOMM Computer Communication Review **44**(3), 87–95 (2014)
10. Darema, F.: Dynamic data driven applications systems: A new paradigm for application simulations and measurements. In: International Conference on Computational Science. pp. 662–669. Springer (2004)
11. Darema, F., Blasch, E.P., Ravela, S., Aved, A.J.: Handbook of Dynamic Data Driven Applications Systems: Volume 2. Springer Nature (2023)
12. Drias, Z., Serhrouchni, A., Vogel, O.: Taxonomy of attacks on industrial control protocols. In: International Conference on Protocol Engineering (ICPE) and International Conference on New Technologies of Distributed Systems (NTDS). IEEE (2015)
13. Fovino, I.N., Carcano, A., Murel, T.D.L., Trombetta, A., Masera, M.: Modbus/dnp3 state-based intrusion detection system. In: 24th IEEE International Conference on Advanced Information Networking and Applications. IEEE (2010)
14. Fujimoto, R., Barjis, J., Blasch, E., Cai, W., Jin, D., Lee, S., Son, Y.J.: Dynamic data driven application systems: research challenges and opportunities. In: Winter Simulation Conference (WSC). pp. 664–678. IEEE (2018)
15. Goldenberg, N., Wool, A.: Accurate modeling of modbus/tcp for intrusion detection in scada systems. International Journal of Critical Infrastructure Protection **6** (2013)
16. Huitsing, P., Chandia, R., Papa, M., Shenoi, S.: Attack taxonomies for the modbus protocols. International Journal of Critical Infrastructure Protection **1** (2008)
17. Lantz, B., Heller, B., McKeown, N.: A network in a laptop: rapid prototyping for software-defined networks. In: 9th ACM SIGCOMM Workshop on Hot Topics in Networks. pp. 1–6 (2010)
18. Ndonda, G.K., Sadre, R.: A two-level intrusion detection system for industrial control system networks using p4. In: 5th International Symposium for ICS & SCADA Cyber Security Research 2018 5. pp. 31–40 (2018)
19. Niazi, R.A., Faheem, Y.: A bayesian game-theoretic intrusion detection system for hypervisor-based software defined networks in smart grids. IEEE Access **7**, 88656–88672 (2019)
20. Qu, Y., Liu, X., Yan, J., Jin, D.: Dynamic data-driven self-healing application for phasor measurement unit networks. In: International Conference on Dynamic Data Driven Applications Systems. pp. 85–92. Springer (2020)
21. da Silva, E.G., Knob, L.A.D., Wickboldt, J.A., Gaspary, L.P., Granville, L.Z., Schaeffer-Filho, A.: Capitalizing on sdn-based scada systems: An anti-eavesdropping case-study. In: IFIP/IEEE International Symposium on Integrated Network Management. IEEE (2015)
22. Yusheng, W., Kefeng, F., Yingxu, L., Zenghui, L., Ruikang, Z., Xiangzhen, Y., Lin, L.: Intrusion detection of industrial control system based on modbus tcp protocol. In: IEEE 13th International Symposium on Autonomous Decentralized System. IEEE (2017)