If I Knew Then What I Know Now: On Reevaluating DNP3 Security using Power Substation Traffic

Celine Irvine, Tohid Shekari, David Formby, and Raheem Beyah

cirvene3@gatech.edu, t.shekari@gatech.edu, djformby@gatech.edu, rbeyah@ece.gatech.edu

Georgia Institute of Technology

Atlanta, Georgia

ABSTRACT

In the modern world, the reliable and continuous operation of cyber-physical systems (CPSs) have become increasingly crucial factors of our daily life. As a result, the networking protocols of CPSs have been developed to achieve availability without serious consideration for security. Security flaws in these protocols could lead to system misconfigurations or malicious network penetrations which would severely impact the operation of critical infrastructure and control devices on a CPS network. To combat this some researchers have made efforts to design effective intrusion detection and prevention systems (IDSs/IPs) for providing security in CPS networks. Most of the past and ongoing work in this space explores security from virtual testbeds or simulated systems, many of which make simplifying assumptions. These artificial platforms generally rely on the expectation that CPS networks are behaviorally very similar to traditional information technology (IT) networks and this does not always hold true in practice. In this paper, we investigate and discuss the feasibility and efficacy of previously proposed DNP3 application layer attacks and their mitigation techniques on network traffic captured from four real-world power grid substations. Based on this and a traffic characterization of the captured data we suggest a set of lightweight, but effective mechanisms to help enhance the security of power substations utilizing the DNP3 protocol. This work primarily focuses on DNP3 since it is the most widely used protocol in power substations which form the backbone of the electricity grid.

CCS CONCEPTS

• **Networks** → **Cyber-physical networks; Network measurement; • Security and privacy** → **Intrusion detection systems; Network security; • Computer systems organization** → **Embedded and cyber-physical systems.**

KEYWORDS

Cyber-physical systems (CPSs), DNP3, intrusion detection system (IDS), network security

ACM Reference Format:

Celine Irvine, Tohid Shekari, David Formby, and Raheem Beyah. 2019. If I Knew Then What I Know Now: On Reevaluating DNP3 Security using Power

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

ICSS '19, December 10, 2019, San Juan, PR

© 2019 Association for Computing Machinery.

ACM ISBN 978-1-4503-7719-5/19/12...\$15.00

<https://doi.org/10.1145/3372318.3372324>

Substation Traffic. In *Fifth Annual Industrial Control System Security (ICSS) Workshop (ICSS)*, December 10, 2019, San Juan, PR, USA. ACM, San Juan, PR, 12 pages. <https://doi.org/10.1145/3372318.3372324>

1 INTRODUCTION

In recent years, the development of highly complex control systems along with huge advancements in information technology (IT) have led to the introduction of a new class of systems called cyber-physical systems (CPSs) [24]. Such systems, in the form of industrial control systems (ICSs) and supervisory control and data acquisition (SCADA), are leveraged in manufacturing processes and critical infrastructures (power grid etc) where safe and reliable operation is of the utmost importance [22].

For the better part of the last 20 years, DNP3 has been a network protocol widely used for communication between SCADA devices. These devices are typically used for process control automation and extend across many industries. Two industries where DNP3 is particularly prevalent are the power grid and water utilities. DNP3 is very useful for data acquisition and is usually deployed for communication between a master and slave/secondary node or for master-master interactions. In power grid applications, the master station is typically the control center and the slave/secondary node is a remote terminal unit (RTU) or intelligent electronic device (IED) located in a power substation [7]. To facilitate these connections, DNP3 uses a layered approach similar to TCP. One of its main advantages is operating under the ‘only communicate when necessary’ doctrine, this has been crucial to DNP3’s adaptability and has made it especially effective over even the slowest communication channels.

Historically, the Internet and other IT networks have been a source of much research interest, and as a result, there have been a large number of modeling and measurement studies on the security and availability of IT protocols [17, 29]. This has not been the case with ICS networks and its protocols. An early paper in the ICS space, published by Cheung et al. [6], investigated anomaly detection in the network. The researchers found that network anomaly detection is more effective when communication patterns are stable and consistent. This specific behavior is more accurately describes ICS networks than traditional IT networks. That work was then followed by a few other research efforts which proposed applying intrusion and anomaly detection to smart grid networks [1, 3], water/waste management systems [19], and other ICSs [5, 16, 27].

With the DNP3 protocol reaching heights never before known, it is important to understand its implementation in practice to enhance the security of the systems which rely on it. Needless to say, there is a wealth of knowledge available about how it should operate in theory, but behavior in the wild may vary significantly. At present, there are few papers thoroughly studying the characterization of ICS

networks and the DNP3 protocol in particular. The main reasons for this are: i) the industries which DNP3 is used are very resistant to change and ii) security of CPSs has only recently become common.

This paper presents an analysis on the effectiveness of previously proposed DNP3 application layer attacks based on a large scale dataset collected from four power substations. We also evaluate the existing mitigation techniques proposed in the literature and discuss how these methods fare on our real world dataset. To gain an understanding of real ICS network semantics and to detect abnormalities and interesting characteristics we also conduct a traffic characterization of DNP3 application layer data captured from over two years of live power substation networks. The major contributions of this paper can be summarized as follows:

- An analysis on the application layer of a large scale DNP3 traffic is conducted to characterize its abnormalities and unique features;
- A broad assessment of previously proposed DNP3 application layer attacks and their mitigation techniques are carried out from the perspective of a real-world dataset to evaluate their effectiveness in the ICS context;
- A set of lightweight recommendations are proposed to enhance the security of power grid substations through the defense of DNP3 application layer attacks.

The remainder of the paper is organized as follows, Section 2 discusses the related work in the field. Section 3 provides background on the DNP3 attack space. Then, Section 4 describes our dataset and presents the traffic characterization experiments, results, and insights. Next, Section 5 discusses feasibility of previously proposed application layer attacks from the perspective of our datasets. Then Section 6 details some lightweight proposed network defense and mitigation strategies and finally, Section 7 concludes.

2 RELATED WORK

While there has been a decent amount of research in characterizing pure Internet traffic, there has been little research published related to the characterization of dedicated ICS network traffic. One of the first and most widely cited Internet characterization papers was on end-to-end behavior of bulk TCP transfers across nodes published in 1999 by Vern Paxson et. al. [29]. This paper revealed abnormal behavior seen on the network including details about packet loss, bottleneck bandwidth, and out of order deliveries which opened the floodgates for similar work.

In ICS networks however, there are different challenges faced than those in traditional IT. For example, in ICS network bandwidth consumption is typically low and it is uncommon for the network layout to change frequently (nodes are not constantly connecting and disconnecting). ICS networks more and more contain embedded devices/IEDs which, in terms of their network footprint, are not well studied or understood. Much of the limited research that has taken place in the SCADA domain was released in 2012 and found that the traffic of a water distribution facility was unlike that of a typical IT network [2]. In addition to this, research was published in 2014 that provided a high-level look at power substation traffic [21]. Then in 2017, an in-depth study was published that provided a more detailed characterization [15] of behavior across multiple substations and discussed methods to improve TCP for ICS networks.

On the DNP3 attack side, there has been a handful of work taxonomizing [14], analyzing [9, 35], and specifying [23] offensive techniques aimed at vulnerabilities in the various layers of the DNP3 protocol. In addition to attacks, many papers proposed mitigation and detection techniques for their attacks. Among these proposed techniques are statistical approaches [8, 10] and empirical/experimental methods [31, 32]. Some [20, 25] of which even adapt network security software from the traditional computing domain, like Snort [33] and Bro [30], for ICS networks. None of the previously mentioned works, however, test their attacks or detection techniques on real critical infrastructure in the wild. They all use theoretical analysis or virtual/physical testbeds with varying levels of design complexity.

Motivated by this, this paper takes a deep dive into datasets collected from four real power substations over the course of 2.5 years and characterizes the DNP3 application layer traffic in order to analyze attack and mitigation strategies previously proposed in the literature.

3 BACKGROUND ON DNP3 ATTACKS

In December 2015, the BlackEnergy malware was used to trigger a blackout which left over 200,000 Ukrainians without power for several hours [39]. While this attack did not exploit DNP3, it was considered to be the first known successful attack on the power grid and drew international attention to the practical repercussions of ICS and SCADA system attacks. Following BlackEnergy other ICS malware have been discovered including, TRITON and CRASHOVERRIDE. The TRITON/TRISIS [4] malware, exploits the proprietary TriStation protocol, targets Schneider Electric Triconex Safety Instrumented System controllers, and was only discovered after an accidental shutdown an industrial process on a Saudi Arabian oil and gas facility triggered an investigation. The CRASHOVERRIDE/Industroyer [18] malware is believed to specifically target the electric grid and was deployed in a December 2016 cyberattack on a Ukrainian transmission station causing an hour-long blackout. These attacks and others have led to an elevated concern about the security of DNP3 communications. Detailed background information on the structure of the DNP3 protocol can be found in Appendix D.

As the primary SCADA protocol used in the electric industry, the defense of many power grids relies, in part, on the assurance of DNP3. To maintain this assurance we must study and understand the attack and threat landscape. In the remainder of this section we will discuss some prominent attacks by layer, which have been presented in the literature [14].

3.1 Multi-layer Attacks

Most attacks on DNP3 are broken down into the protocol layers for which the attack is deployed. There are, however, three compromises which are indifferent to the protocol layer. The first is network reconnaissance or snooping, passively collecting messages for developing a deeper understanding of the overall network infrastructure and devices in use. Next is the replay response attack where the master station is sent simulated/replayed packets and the outstation is sent fake/spoofed packets. This attack requires precision in order to keep both ends of the communication link unaware of the breach. Lastly is the pure man-in-the middle (MITM) attack where an attacker uses

a device to ingest messages from both sides of a communication link and then injects the packets it desires in their stead.

3.2 Link Layer Attacks

The DNP3 link layer contains Start/Magic bytes, Length, Control, Source/Destination address, and a CRC field, shown in Appendix D.0.1. The most prominent attacks on the link layer involve altering the Length field, setting the DFC bit, spoofing the Function codes in the Control field, and altering the Source/Destination fields. To attack the length field an attacker could intelligently input a value that causes the entire packet to be improperly processed. This could lead to unexpected device behavior, data corruption, or even device crashes. The DFC bit is used to prevent the overflow of buffers in the outstation, if set the outstation will appear unavailable to the master. This attack could cause serious delays and interruptions to the outstation as the master will retry communication at a later time. The DNP3 Control field contains function codes which indicate what type of frame the message is, and Function Codes 1, 14, and 15 could be leveraged for major disruption. If Function Code 1 received by an outstation, it completely resets the user process, when Function Code 14 is received by the master, it signals the outstation is not functioning, and when Function Code 15 is received by the master, it means the requested service is not supported by the outstation. These three Function codes, if used by a clever attacker at the right time, could wreak a lot of havoc on a DNP3 network. Lastly, by changing the Source or Destination address of a packet an attacker could reroute message replies and requests potentially leading to an unstable network state.

3.3 Transport Layer Attacks

The DNP3 transport layer consists of three fields: FIR (first), FIN (final), and SEQ (sequence number), shown in Appendix D.0.2. There are two main attacks which can be used to disrupt the normal operation of the transport layer. The first entails improper message fragmentation. By properly timing the transmission of a message with the FIR or FIN bit set, an attacker can effectively cause denial of service (DoS) in the master or outstation device. This happens because packets received with the FIR bit set trigger the disposal of all previously collected incomplete fragments and when a packet is received with the FIN bit set the message reassembly process is halted at the current state. Both of these attacks can result in fragmented communications, and thus, prevent proper network flows. The final transport layer attack also aims to disrupt proper message fragmentation by injecting and modifying packet sequence numbers (SEQ field). Since DNP3 uses a simplistic method of incrementing each packet SEQ number by one, it would be straightforward to spoof the proper data.

3.4 Application Layer Attacks

The DNP3 application layer, Appendix D.0.3, is the most susceptible one to attack because it provides the majority of the functionality for the protocol. The application layer is made up of at least one field, the application header and a variable number of object header/object pairs. Though many attacks could be carried out against the application layer, the most catastrophic will exploit the Internal Indications and Function Code fields.

Function code 2 (WRITE) is the main method for storing objects in an outstation. An attacker could use this to overflow or corrupt the outstation's memory. Function codes 7, 8, 9, A, B, and C are FREEZE and CLEAR commands, used for holding or erasing the value of counters stored in outstations. If an attacker injects this packet to an outstation, it could lead to device malfunctions and crashes. Another attack could be perpetually sending messages with Function codes D and E, COLD and WARM RESTART, to DoS the outstation and never let it completely start up. Transmitting a packet with Function code F (INITIALIZE) accompanied with random data objects would cause the outstation to reinitialize itself and could lead to system state inconsistencies resulting in device failures. An attack which injects or modifies a packet to contain Function code 12 (STOP) could terminate applications running on the outstation and make it unresponsive to commands from the master. Function code 82 (UNSOLICITED RESPONSE) can be used to spontaneously send message from an outstation to the master. Assuming unsolicited response mode is enabled, attackers can use this to cause DoS or buffer overflow unsuspecting nodes by repeatedly transmitting data packets.

Internal indications (IINs) are sent with RESPONSE DNP3 messages in order to offer more insight into the status of an operation. One attack that could be carried out would be to send a message with IIN2.5, configuration corrupt, set from the outstation to the master. This triggers the master to send a new configuration file which could then be intercepted by a MITM and swapped with the attacker's desired configuration. Setting IIN2.3, event overflow buffer, in a RESPONSE message will trigger the master to request event data, if an attacker keeps this bit set she can dupe the master into an infinite loop of requesting data, impeding it from performing other tasks.

The emphasis of this work is on attacks against the DNP3 application layer, because it provides the majority of the functionality for DNP3 and has the largest attack space of the three layers. Consequently, it is the most vulnerable target for attackers.

4 PROCESS TRAFFIC CHARACTERIZATION

The first step towards securing ICS protocols is studying normal behavior in a real-world environment. Thus, in order to better understand the DNP3 attack surface as it exists in the wild, we performed an application layer traffic characterization. To do this we processed all of our raw power substation packet captures into a succinct human readable form via a python script which parsed selected packet attributes into JSON format. The selected attributes extracted from the network captures are shown in Appendix A. Next we organized the JSON files by vendor and created data visualizations to better observe trends in the application layer function codes, internal indications fields, and packet sizes. Due to the large size of the datasets, this analysis was performed on PACE [28] (a high performance computing cluster for Georgia Tech researchers).

4.1 Dataset Details

Six datasets of network traffic data were used to conduct this research. The datasets are labeled A1, A2, A3, B, C, and D and were taken from four medium-voltage distribution substations non-consecutively over the course of two and a half years. In that time, there was a minor network architecture reconfiguration, due to which, dataset

A1 has a slightly different network structure than the other datasets do.

Datasets A1, A2, and A3 were collected at the same substation (A) and the others were taken from separate substations on the same SCADA network. Dataset A1 is ~22GB in size and was collected from substation A from September 2013 to February 2014. Dataset A2 is ~147GB in size and was collected from substation A from January to August of 2015. The final dataset A3 gathered from substation A was ~147GB taken from August 2015 to April 2016. Dataset B was collected over the course of 1 day from August 10 to August 11, 2015 and was 0.34GB in size. Datasets C and D were both collected from April to May 2016 and were ~6GB and ~12GB respectively.

4.2 Dataset A1

The A1 dataset consists of devices from Beckwith, Cooper, Garrett, Versalog, Pronet, and Cisco. All of the function codes from the Beckwith and Cooper devices were solicited responses, this makes sense because these devices are usually IEDs or slave devices. In a traditional DNP3 network configuration, slave devices will typically only send data if they receive a request or an unexpected event occurs and they must alert the master. Of these requests over 90% were less than 50 bytes in size and approximately 3% were greater than 200 bytes. In total 46% of function codes captured were RESPONSEs, everything else was a REQUEST. Of the captured REQUESTs over 80% were READ requests and about 7% were CONFIRM messages.

Out of the five total vendor devices in this dataset only three vendor devices (Versalog, Cooper, Beckwith) had any internal indications fields set, this can be seen in Figure 1. The internal indications were grouped into six categories:

- Broadcast Received: used to inform the master that a broadcast message was received
- Unreported Events: used to alert the master that new events have occurred and it should request class 1/2/3 data from the device
- Need time: used to ask for time synchronization from the master
- Local Control: used to notify the master that some or all of the devices' control outputs are inaccessible through the DNP3 protocol
- Restarted: gets set when the user application running on the device is restarted
- Miscellaneous Error: used to tell the master station about various errors that may have occurred on the device from buffer overflows to bad configurations

From Figure 1 we see that approximately 74% of internal indications sent were unreported events. That is, the outstation alerting the master station that something new has happened and it should request the updated data. The next two highest transmitted internal indications were local control and errors.

4.3 Dataset A2

The A2 dataset consists of devices from Beckwith, Cooper, Versalog, Pronet, and Cisco. Similar to the A1 dataset all of the function codes from the Beckwith and Cooper devices were solicited responses. Again, this is a strong indication that all of these devices were IEDs or slave devices. The RESPONSE function code made up 48% of the

Figure 1: Dataset A1 Internal Indications Field Distributions

function codes transmitted in the A2 dataset, almost half of all communications. The other near half, 46%, of transmitted function codes were READs. READ is a request function code that prompts the outstation to send the requested data to the master station. The remaining 4% of function codes were CONFIRMs which are essentially acknowledgments from the master that a fragment was received. In the A2 dataset, more than 95% of packets contained application data lengths less than 50 bytes in size and only 3% were between 100 and 150 bytes.

Just like with dataset A1, in the A2 dataset, only three vendor devices (Versalog, Cooper, Beckwith) of the five captured had any internal indications fields set, which can be seen in Figure 2. The dominating IIN field, at approximately 73%, is local control based solely on traffic from the Cooper devices. Unreported events and miscellaneous errors follow at 16% and 8% respectively. From the number of unreported events generated by Cooper devices it can be inferred that the polling rate of the master may have been too slow during that period of time. This in combination with the number of local control packets transmitted indicate that someone may changed the device from remote to local to make changes.

Figure 2: Dataset A2 Internal Indications Field Distributions

4.4 Dataset A3

The A3 dataset consists of devices from Beckwith, Cooper, Versalog, Schweitzer Engineering, Pronet, Cisco, and an unknown vendor. Just as in dataset A1 and A2, the Beckwith and Cooper devices only transmitted packets with functions codes that were solicited RESPONSEs. In addition to Beckwith and Cooper, the Versalog devices in the capture also sent a large number of RESPONSE function codes. In total the RESPONSE function code made up 47% of the function codes

transmitted in the A3 dataset, followed by the READ function code at 48%. Versalog devices, which from their communication pattern may have been master devices, sent 98% of READ function codes. As in other datasets CONFIRM function codes were seen a lot less than the aforementioned, making up less than 5% of the traffic captured. About 98% of packets in the A3 dataset contained application data lengths less than 50 bytes in size.

As with dataset A1 and A2, only three vendor devices (Versalog, Cooper, Beckwith) of the five captured had any internal indications fields set, this can be seen in Figure 3. From the figure we see the dominating IIN field, at approximately 72%, is local control based primarily on traffic from the Cooper devices. Unreported events and miscellaneous errors followed at ~15% and ~6%, respectively.

Figure 3: Dataset A3 Internal Indications Field Distributions

4.5 Dataset B

Dataset B consists of devices from Beckwith, Cooper, Schweitzer Engineering, Versalog, Pronet, and Cisco. In this dataset, the overwhelming majority of the packets contained in the READ function code. Over 88% of all traffic captured was READ function codes and only 6% and 4% consisted of RESPONSEs and CONFIRM, respectively. All of the packets containing the RESPONSE function code came from three vendor devices, Versalog, Beckwith, and Cooper, as has been the trend in all of the other datasets. Another trend that remained consistent in dataset B was the packet sizes. Over 99% of DNP3 application data lengths were smaller than 50 bytes.

Figure 4 shows a plot of the IIN field data from dataset B. The only vendor devices that had IINs set were Versalog, Cooper, and Beckwith. This makes sense because only packets which have the RESPONSE function code set would also set internal indications expressing the status of the performed operation. From the figure, we see the dominating IIN field is unreported events, making up over 75% of the packets. We primarily see these events occurred on the Cooper devices. This likely means there was a lot of activity occurring during the time of capture at those devices and the master did not ask for updates frequently enough. The next greatest IIN sent were local control, making up 23% of the traffic.

4.6 Dataset C

Dataset C consists of devices from Cooper, Schweitzer Engineering, Versalog, Pronet, and Cisco. In terms of the function codes seen, it is almost evenly split between READs and RESPONSEs, which make up 51% and 44% of all codes seen, respectively. CONFIRM

Figure 4: Dataset B Internal Indications Field Distributions

function code makes up a mere 5% and as usual the RESPONSEs only came from Versalog, and Cooper devices. The Versalog and Schweitzer devices offer more variety in terms of function codes transmitted than the other vendor devices. The Schweitzer device(s) transmitted nearly 11,000 packets with the FREEZE function code set and about 1000 WRITE packets. FREEZE is used for freezing or holding the value of counters or input points. The data gets copied by an outstation to a freeze buffer where it is saved to be requested for at a later point in time. The WRITE function code indicates to the outstation to store the objects sent in the REQUEST. Regarding packet sizes, dataset C consisted of a majority of packets, 75%, with application data lengths below 50 bytes and about 22% between 150 and 200 bytes. Given the periodic and systematic nature of control networks, the large number of small packets is not unusual.

Figure 5 shows a plot of the Cooper and Versalog IIN field data from dataset C. We can see from the figure that the most popular internal indications transmitted were unreported events, just like in dataset B. Unreported events made up approximately 75% of the IINs sent and the bulk originate from Cooper devices. The next greatest IINs sent were local control, which means that some or all of the devices digital output points were in a local state. These points are therefore inaccessible by the DNP3 protocol.

Figure 5: Dataset C Internal Indications Field Distributions

4.7 Dataset D

Dataset D consists of devices from Cooper, Schweitzer Engineering, and Versalog. C and D are the only two datasets without any Beckwith devices. Another similarity with dataset C is that the function codes transmitted in D are almost evenly split between RESPONSEs

and READs, making up 49% and 46% of the codes captured. Versalog devices were the main contributor of packets with the READ function code set. These devices also sent the most CONFIRM function codes. As with many of the other dataset, approximately 94% of the DNP3 application data lengths in this dataset were smaller than 50 bytes. Something which deviated from the norm was that almost 5% were between 100 and 150 bytes, which could possibly be due to the transmission of large object ranges within packets.

Figure 6: Dataset D Internal Indications Field Distributions

Figure 6 shows a plot of the IIN field data from dataset D. The vendor devices that had IINs set were Versalog and Cooper. This makes sense because only packets which have the RESPONSE function code set would also set internal indications expressing the status of the performed operation and Cooper and Versalog were the only vendor devices that sent RESPONSE codes. From the figure we see the dominating IIN field, at approximately 64%, is unreported events. We primarily see these events occurred on the Cooper devices. This likely means there was a lot of activity occurring during the time of capture at those devices and the master did not ask for updates frequently enough. The next most common IINs sent were local control, making up nearly 35% of the traffic.

5 DNP3 ATTACK/DEFENSE ANALYSIS

From the traffic characterization above, it can be seen that real world DNP3 data can vary greatly from substation to substation and even within the same substation over time. Nevertheless, there are differences certain trends can be witnessed throughout. Figure 7 shows a breakdown of all the function codes seen at each substation over the course of the 2.5 year capture period. From the figure, we can see that the majority of function codes seen are READs and RESPONSEs. No STOP, RESTART, or INITIALIZE function codes are ever observed at any of the four substations. The SELECT/OPERATE, FREEZE/CLEAR, TIME, CONFIRM, SET and WRITE are the only other function codes and they make up less than 6% of the total application layer traffic of each substation.

To better understand the breadth of DNP3 attack landscape, in the remainder of this section we will discuss several DNP3 attacks proposed in the literature. Additionally, we will analyze their proposed mitigation techniques (if given) from the perspective of our real-world dataset to determine their practical feasibility.

Figure 7: Dataset Function Code Distributions

5.1 Attack 1

In [38] the authors describe two attacks, abnormal control/data transfer and traffic flooding attack. Abnormal transfer attacks occur when malware or irregular data packets traverse through a control network. The flows may be disguised as normal network traffic looking for specific target hosts to infect or may come in the form of a series of vulnerable commands. Traffic flooding attacks occur when a node receives more data than its buffer can handle and is overwhelmed by the load.

To counter these attacks, the authors propose a whitelist-based approach which categorizes all seen network traffic into bursts (data tuples containing the direction of traffic flow, function code, and data object). Once the whitelist model is created rules must be generated for an IDS, and then if any packet is identified that violates a rule it is flagged.

Figure 8: Bursts Required for White Model Approach Assuming 1 Function Code-Data Object for Each Master/Outstation Pair

This approach will require a lot of overhead in training (to create the whitelist model) and in real time (for burst matching and identification). In addition to this it will not scale well with the size of the network because it requires the storage of every burst (all unique function code and data object combinations) seen by every master/outstation pair. Figure 8 shows how the amount of bursts

stored would blow up once the simplifying assumption made in the paper, that every outstation talks to 1 master, is removed. From the figure we see 8 bars, 6 representing the number of bursts required for our real-world datasets, 1 representing the 43 outstations from [38], and 1 for a hypothetical ICS network with 500 outstations. Even just assuming 1 function code-data object pair per master-outstation in smallest case (Yun [38]), we can see how the numbers go from 43 bursts with 1 master to 172 bursts when there are 4 masters. In the largest case (hypothetical 500 outstation network), the number of bursts skyrocket from 500 to 2000 when there are 4 masters. These are still modest because they assume an outstation exclusively talks to 1 master with the exact same function code and data object every single time. Appendix B shows how the number of bursts scale assuming a network with 300 outstations and varying number of masters as well as varying number of function code-data object pairs. In the case where there are 300 outstations, each with 2 masters and 5 function code-data object pairs we can expect to see 3000 bursts. The authors evaluated their approach with 10 days of real substation traffic, totaling 887MB, and many of the simplifying assumptions made due to their small dataset will not hold in practice.

5.2 Attack 2

In [35], the goal was to launch attacks based on three threat categories, interception, fabrication, and modification in order to identify the function codes which pose the largest threats to DNP3. To do this, a multidrop DNP3 network was simulated which consisted of one Master Terminal Unit (MTU) and three Remote Terminal Units (RTUs), all configured with OpenDNP3 version 1.1.0. Traffic was then generated and Scapy scripts were used to fabricate, inject, and modify packets with assorted function codes. From their experiments the authors concluded that the function codes with the largest attack surface from RTUs were RESPONSE and UNSOLICITED RESPONSE and the function codes with the largest attack surface from MTUs were READ, SELECT, OPERATE, and ENABLE/DISABLE UNSOLICITED RESPONSE.

From our real world traffic captures, collected from networks with hundreds of IEDS/RTUs, we observed some similarities and some differences in our analyses. The differences, highlighted in Table 1 and Appendix C, are likely artifacts of the limited resolution of the OpenDNP3 framework used, namely the reduced function code set and restricted allowable MTU-RTU communications. The authors had to make simplifying assumptions in their experiments which repress the fullness of the protocol as it exists in real deployments and thus their subsequent assessments are incomplete.

5.3 Attack 3

In [31], four theoretical attacks from [14] were reproduced via a Scapy extension of DNP3 and assessed on a small testbed of critical infrastructure equipment. The attacks were eavesdropping, address alteration, length overflow, and configuration capture. We will only discuss the address alteration and configuration. Eavesdropping through ARP (Address Resolution Protocol) cache poisoning is assumed to be a prerequisite for an attacker to launch intelligent attacks and the length overflow attack is almost guaranteed to succeed (regardless of network setup) as it prevents the frame from being properly parsed by the victim.

Table 1: DNP3 Function Code Attack Surface Comparison

Function Code	Support		Attack Surface	
	Us	Singh[35]	Us	Singh[35]
READ	✓	✓		✓
WRITE	✓	✓	✓	
CONFIRM	✓	✓		
SELECT	✓	✓	✓	✓
OPERATE	✓	✓	✓	✓
FREEZE	✓		✓	—
CLEAR	✓		✓	—
RESTART	✓		✓	—
INIT	✓		✓	—
RESPONSE	✓	✓		
UNSOL RESPONSE	✓	✓		
START	✓			—
STOP	✓		✓	
EN/DIS UNSOL	✓	✓	✓	✓

The researcher's testbed consisted of one master, one slave, three switches, and one attacker node (connected to the same switch as the master). In the address alteration attack the DNP3 frame destination address was modified to that of a nonexistent device on the network and it was observed that the slave device did not receive those packets. We expect the same behavior from a real deployment and note that if a slave device has multiple masters, common in practice as shown by Table 2 from our dataset (not verifiable in [31] due to testbed constraints), this attack could be used to destabilize an unsuspecting node. For the configuration capture attack the goal is to alter packets from the slave to master by setting the corrupt configuration IIN bit. The trick is triggering the master to resend the configuration file over the network, which an attacker can then capture and swap for any malicious configuration file of their choice. When performing this attack on the testbed the authors found that even though the master received the configuration corrupt packet, it never retransmitted any configuration files. This was attributed to the fact reloading a configuration file requires critical infrastructure to be taken offline and in an effort to prevent system downtime many ICS networks do not allow master devices to store slave configurations.

Figure 9: Dataset Analysis of 3 Internal Indication Fields

From our research we note two additional points, the first is that according to the DNP3 specification support for the configuration corrupt IIN bit is optional and as a result unlikely to be implemented in practice. Keeping the machines up and running is so crucial in ICSs that even scheduled maintenance only occurs once or twice a year for an hour or so. Even then only a portion of the substation is deenergized at a time in an effort to keep providing services. In an industry where devices operate with a maximum 3000ms delay after start up [34], the seconds or minutes required for an outstation to reboot after configuration reload cannot be spared. Second is that in our data capture of more than two years of traffic from four substations we never observed a single configuration corrupt packet, we highlight this in Figure 9. From the figure we see no configuration corrupt packets, but very many miscellaneous errors (IIN2.1 object unknown, IIN2.2 parameter error, and IIN2.4 already executing) and device restarts.

Table 2: Number of Masters Observed by Dataset

Dataset	Number of Masters
A1	1
A2	3
A3	3
B	5
C	4
D	2

5.4 Attack 4

In [13] an event buffer flood attack is described. The premise of this attack is that DNP3 communications are hierarchical and rely on data aggregators at level 1 to gather sensor data from various devices and send them to the control center on level 2. To launch the attack a connection must be established to the data aggregator by either a compromised legitimate node or a spoofed node. Once the connection is initiated the aggregator's sequence of event buffer can be flooded by repeatedly transmitting unsolicited messages. At that time, data from other legitimate nodes will be dropped until the aggregator's queue is cleared.

To counter this attack, the researchers developed an evaluated a Discrete-Time Markov Chain (DTMC) analytical model and a stochastic activity network (SAN) for determining the conditions under which buffer flooding is successful. They discovered that buffer flooding is generally successful when the buffer space is shared among multiple sources and follows a first-come-first-serve rule. Ultimately, they recommended the use of fair queuing policies like, round robin, to ensure every input has reserved buffer space. In addition to this they advocate for the use of setting IIN bits as an indication to the master that an overflow has occurred and for better protocol authentication like in DNP3Sec [26].

While the proposed approaches will in fact curtail the effects of buffer flooding attack they do nothing to alert network administrators of the compromise and will require wide scale SCADA network device deployment to be effective. Implementing a fair queuing policy on an active ICS network would require taking several nodes offline for reprogramming and downtime is unacceptable in many

ICSs, especially in electric power systems. Figure 10 gives an estimate of the downtime required to reprogram data aggregators assuming reprogramming takes 1,2,5, or 10 minutes and is done in series, 5 in parallel, or 10 in parallel. As expected the, the downtime grows with the number of aggregators that must be reprogrammed. Even though downtime can be reduced by reprogramming in parallel, in the moderate case of 50 aggregators with reprogramming 5 at a time and taking 5 minutes from start to finish, the substation would still be down for nearly an hour. The modesty of this estimate is emphasized when we take into consideration that of the four substations from our real world datasets the minimum number of outstations/aggregators was 104.

Figure 10: Substation Downtime as a Function of Reprogramming Data Aggregators

Furthermore, wide scale new protocol adoption is a hard sell in large scale legacy networks, so switching to a more robust protocol like DNP3Sec is also not feasible in practice. Even in the event that a handful of ambitious power substation or ICS network operators decided to use a more secure protocol there would be time delay consequences from the various protocol translations and conversions that would be required. Protocol translators (also known as protocol converters) must be implemented as an intermediary between different protocols and are known to cause communication delay, processing latency, and an overall increase in end to end processing time [37]. This additional overhead is unacceptable in critical infrastructures, where even 20ms delays could affect the performance of protection schemes and cause instability in the entire system.

6 RECOMMENDATIONS

With the insights gathered from reviewing the literature in the DNP3 application layer attack space and performing a traffic characterization of large scale power substation data we discovered some behaviors that deviated from previous implicit assumptions. The assumption is on the content of data transmissions in real ICS networks, many former works have based their attacks and mitigations on the theory that DNP3 masters/outstations have greatly varying communication sequences. From our analysis of four power substations we see that, in fact, the exact opposite is true. The range of communications that actually take place in the wild, as seen in Figure 7, is actually fairly narrow. This is further illustrated in Figure 11, where the occurrences of function codes with the largest attack surface are represented for our four power substations and there are no STOPs, INITIALIZEs, or RESTARTs. We assert that the limited

number of function codes transmitted between nodes can be used to secure the network. In the following subsections we will describe our proposed lightweight technique for DNP3 data availability as well as recommend other previously proposed mitigations for data confidentiality and integrity.

Figure 11: Dataset Distribution of Function Codes with Largest Attack Surface

6.1 Baseline Distribution Countermeasure

The key idea behind our proposed mitigation is to leverage the limited set of application layer function codes which occur in practice to generate a baseline distribution of normal network behavior. A distribution of DNP3 application layer function code occurrences and average packet interarrival time (PIT) per function code, as well as an occurrence sensitivity tolerance are the main ingredients required to generate an IDS rule which will alert network administrators of an application layer anomaly. Computing the distribution of function code occurrences consists of collecting a real-world traffic capture then counting each instance of a particular function code. Getting the average PIT is also a straightforward calculation and only requires measuring the amount of time which between each packet of a particular function code then computing mean given the number of packets seen over the capture period. This is shown mathematically, assuming function code a , in Equation 1, where t is time of packet and j is the total number of packets for that function code. The last component is the occurrence sensitivity tolerance, this will determine how fine or coarse grained the alerts from the IDS will be. If the tolerance is too low then anytime a packet is received which deviates from the baseline distribution there will be an alert (many false positives), if it is very high then there will be many false negatives. Getting a proper balance for the tolerance setting will depend greatly on the variability of the ICS network and how much captured traffic was used to generate the baseline distribution.

$$AveragePIT_a = \frac{\sum_{i=1}^j t_{i+1} - t_i}{j} \quad (1)$$

Our lightweight baseline distribution technique makes detecting many of the application layer attacks proposed in [14, 31, 35] trivial, because they violate the natural function code structure. With a learned distribution of function codes, if substation A, for example, had never seen any RESTART packets over the baseline capture period then one day all of a sudden begins receiving a flood of

RESTART packets, it would be considered anomalous. The baseline distribution can feed the IDS crucial information, like how many and what type of function codes are expected. Then as an additional input the network administrators must set an allowable tolerance over/under the distribution which is still considered normal. For instance, if on average 1000 packets/hour are transmitted in substation A and 1 packet/hour (1/1000th of the traffic) is usually time synchronization requests it may not be particularly anomalous for 3 packets to be observed over a particular hour. It may, however, be anomalous if an order of magnitude difference (10 packets/hour) was noted in the amount of time synchronizations requests. This is where the occurrence sensitivity tolerance and average packet interarrival become key. Consider a network where receiving 10000 WRITE packets is expected and normal, on this same network if those 10000 WRITE packets arrive within a 2-minute window it may not be normal and the function code interarrival time helps provide a basis for the expected time between packets. It should be noted that our approach will vary depending on the traffic of the network being monitored. For this paper it is tailored to DNP3 traffic, because that is data we have, but the concept generalizes.

6.1.1 Limitations. While there are clear benefits in the simplicity of our proposed approach there are also some limitations. The baseline distribution technique can help to alert on anomalous network traffic (attacks compromising network availability), but it will not expose more attacks on data integrity or confidentiality. A data integrity attack could utilize functions codes such as READ or SELECT and include erroneous or corrupt data objects intended to destabilize the destination node. An attack on confidentiality could intercept messages and forward them along without being detected. We discuss possible defenses to address these in Section 6.2.

6.2 Additional Defense Strategies

In addition to our baseline distribution approach for detecting network availability compromises, there have been other proposed mitigations which we recommend for securing DNP3's data integrity and confidentiality.

In [10] a time based mitigation technique which could be effective at detecting MITM attacks is presented. The main idea is to use the round trip delay time (RTTD) between master/outstation nodes plus a safety margin to verify the origin of received communication. If a packet arrives outside of the expected window it is dropped giving the legitimate node the chance to retransmit. The authors also develop a Bayesian Estimate to model the communications between legitimate and illegitimate network nodes.

In [23] they propose a DNP3 Authenticated Encryption method, which utilizes authentication and encryption to guarantee data confidentiality. In their work a software proof of concept is presented in C++ which utilize Galois/CounterMode (GCM) mode of operation and the PRINCE cipher for high speed performance with modest resources. Although this technique adds a sizable 24 byte overhead to each packet it is a step in the right direction and could be feasible in practice with the right optimizations.

The last two recommended defenses are similar to ours with proposals of using IDSs to detect attacks on SCADA networks. In [16] they aim to detect chained commands which when viewed in isolation appear harmless, but combined wreak havoc. To do this they

create rules which describe critical ICS system states. Comparably in [36], they present two anomaly detection techniques: pattern-based anomaly detection for monitoring the patterns of hosts communication and flow-based anomaly detection for monitoring individual network flows.

7 CONCLUSION

In this paper we present an application layer characterization of real power substation network traffic, analyze the efficacy previously proposed DNP3 attacks, propose a lightweight application layer defense and make DNP3 data integrity and confidentiality recommendations based on previously proposed mitigations. Our proposed technique leverages insights we discovered from a traffic characterization of four medium voltage power substation network data and undermines the notion that intricate machine learning techniques and sophisticated mathematical algorithms are the only effective approaches for securing the DNP3 application layer.

REFERENCES

- [1] Muhammad Qasim Ali and Ehab Al-Shaer. 2013. Configuration-based IDS for advanced metering infrastructure. In *Proceedings of the 2013 ACM SIGSAC conference on Computer & communications security*. ACM, 451–462.
- [2] Rafael Ramos Regis Barbosa, Ramin Sadre, and Aiko Pras. 2012. A first look into SCADA network traffic. In *Network Operations and Management Symposium (NOMS), 2012 IEEE*. IEEE, 518–521.
- [3] Robin Berthier and William H Sanders. 2011. Specification-based intrusion detection for advanced metering infrastructures. In *2011 17th IEEE Pacific Rim International Symposium on Dependable Computing*. IEEE, 184–193.
- [4] Marina Krotofil Dan Scali Nathan Brubaker Christopher Glyer Blake Johnson, Dan Caban. 2017. Attackers Deploy New ICS Attack Framework “TRITON” and Cause Operational Disruption to Critical Infrastructure. <https://www.fireeye.com/blog/threat-research/2017/12/attackers-deploy-new-ics-attack-framework-triton.html>
- [5] Andrea Carcano, Alessio Coletta, Michele Guglielmi, Marcelo Masera, I Nai Fovino, and Alberto Trombetta. 2011. A multidimensional critical state analysis for detecting intrusions in SCADA systems. *IEEE Transactions on Industrial Informatics* 7, 2 (2011), 179–186.
- [6] Steven Cheung, Bruno Dutertre, Martin Fong, Ulf Lindqvist, Keith Skinner, and Alfonso Valdes. 2007. Using model-based intrusion detection for SCADA networks. In *Proceedings of the SCADA security scientific symposium*, Vol. 46. Citeseer, 1–12.
- [7] Gordon Clarke, Deon Reynders, and Edwin Wright. 2004. *Practical modern SCADA protocols: DNP3, 60870.5 and related systems*. Newnes.
- [8] I. Darwish, O. Igbe, O. Celebi, T. Saadawi, and J. Soryal. 2015. Smart Grid DNP3 Vulnerability Analysis and Experimentation. In *2015 IEEE 2nd International Conference on Cyber Security and Cloud Computing*. 141–147. <https://doi.org/10.1109/CSCloud.2015.86>
- [9] I. Darwish, O. Igbe, and T. Saadawi. 2015. Experimental and theoretical modeling of DNP3 attacks in smart grids. In *2015 36th IEEE Sarnoff Symposium*. 155–160. <https://doi.org/10.1109/SARNOF.2015.7324661>
- [10] I. Darwish and T. Saadawi. 2018. Attack Detection and Mitigation Techniques in Industrial Control System -Smart Grid DNP3. In *2018 1st International Conference on Data Intelligence and Security (ICDIS)*. 131–134. <https://doi.org/10.1109/ICDIS.2018.00028>
- [11] Marshall DenHartog. 2012. *DNP3 Tutorial: Learn the Industry-Standard SCADA Protocol*. Technical Report. DPS Telecom.
- [12] DNP. 2002. DNP3 Quick Reference. <http://read.pudn.com/downloads151/doc/comm/655523/DNP3QuickReference.pdf> [Online; accessed September 18, 2018].
- [13] Dong Jin, D. M. Nicol, and Guanhua Yan. 2011. An event buffer flooding attack in DNP3 controlled SCADA systems. In *Proceedings of the 2011 Winter Simulation Conference (WSC)*. 2614–2626. <https://doi.org/10.1109/WSC.2011.6147969>
- [14] Samuel East, Jonathan Butts, Mauricio Papa, and Sujeet Shenoi. 2009. A Taxonomy of Attacks on the DNP3 Protocol. In *Critical Infrastructure Protection III*, Charles Palmer and Sujeet Shenoi (Eds.). Springer Berlin Heidelberg, Berlin, Heidelberg, 67–81.
- [15] David Formby, Anwar Walid, and Raheem Beyah. 2017. A Case Study in Power Substation Network Dynamics. *Proceedings of the ACM on Measurement and Analysis of Computing Systems* 1, 1 (2017), 19.
- [16] Igor Nai Fovino, Andrea Carcano, Thibault De Lacheze Murel, Alberto Trombetta, and Marcelo Masera. 2010. Modbus/DNP3 state-based intrusion detection system. In *2010 24th IEEE International Conference on Advanced Information Networking and Applications*. IEEE, 729–736.
- [17] Chuck Fraleigh, Sue Moon, Bryan Lyles, Chase Cotton, Mujahid Khan, Deb Moll, Rob Rockell, Ted Seely, and S Christophe Diot. 2003. Packet-level traffic measurements from the Sprint IP backbone. *IEEE network* 17, 6 (2003), 6–16.
- [18] Andy Greenberg. 2017. ‘Crash Override’: The Malware That Took Down a Power Grid. <https://www.wired.com/story/crash-override-malware/>
- [19] Dina Hadziosmanovic, Robin Sommer, Emmanuele Zamboni, and Pieter Hartel. 2013. Through the eye of the PLC: towards semantic security monitoring for industrial control systems. *International Computer Science Institute, Berkeley* (2013).
- [20] Hao Li, Guangjie Liu, Weiwei Jiang, and Yuewei Dai. 2015. Designing snort rules to detect abnormal DNP3 network data. In *2015 International Conference on Control, Automation and Information Sciences (ICCAIS)*. 343–348. <https://doi.org/10.1109/ICCAIS.2015.7338690>
- [21] Sang Shin Jung, David Formby, Carson Day, and Raheem Beyah. 2014. A first look at machine-to-machine power grid network traffic. In *Smart Grid Communications (SmartGridComm), 2014 IEEE International Conference on*. IEEE, 884–889.
- [22] Eric D Knapp and Joel Thomas Langill. 2014. *Industrial Network Security: Securing critical infrastructure networks for smart grid, SCADA, and other Industrial Control Systems*. Syngress.
- [23] Dongsoo Lee, Hakju Kim, Kwangjo Kim, and Paul D. Yoo. 2014. Simulated Attack on DNP3 Protocol in SCADA System. In *2014 Symposium on Cryptography and Information Security (SCIS)*.
- [24] Edward A Lee. 2008. Cyber physical systems: Design challenges. In *2008 11th IEEE International Symposium on Object and Component-Oriented Real-Time Distributed Computing (ISORC)*. IEEE, 363–369.
- [25] Hui Lin, Adam Slagell, Catello Di Martino, Zbigniew Kalbarczyk, and Ravishankar K. Iyer. 2013. Adapting Bro into SCADA: Building a Specification-based Intrusion Detection System for the DNP3 Protocol. In *Proceedings of the Eighth Annual Cyber Security and Information Intelligence Research Workshop (CSIIRW '13)*. ACM, New York, NY, USA, Article 5, 4 pages. <https://doi.org/10.1145/2459976.2459982>
- [26] Munir Majdalawieh, Francesco Parisi-Presicce, and Duminda Wijesekera. 2006. DNPSec: Distributed Network Protocol Version 3 (DNP3) Security Framework. In *Advances in Computer, Information, and Systems Sciences, and Engineering*, Khaled Elleithy, Tarek Sobh, Ausif Mahmood, Maged Iskander, and Mohammad Karim (Eds.). Springer Netherlands, Dordrecht, 227–234.
- [27] Stephen McLaughlin. 2013. CPS: Stateful policy enforcement for control system device usage. In *Proceedings of the 29th Annual Computer Security Applications Conference*. ACM, 109–118.
- [28] PACE. 2017. *Partnership for an Advanced Computing Environment (PACE)*. <http://www.pace.gatech.edu>
- [29] Vern Paxson. 1997. End-to-end Internet packet dynamics. In *ACM SIGCOMM Computer Communication Review*, Vol. 27. ACM, 139–152.
- [30] Vern Paxson. 1999. Bro: A System for Detecting Network Intruders in Real-time. *Comput. Netw.* 31, 23-24 (Dec. 1999), 2435–2463. [https://doi.org/10.1016/S1389-1286\(99\)00112-7](https://doi.org/10.1016/S1389-1286(99)00112-7)
- [31] Nicholas R. Rodofile, Kenneth Radke, and Ernest Foo. 2015. Real-Time and Interactive Attacks on DNP3 Critical Infrastructure Using Scapy. In *2015 Australasian Information Security Conference (AISC)*.
- [32] Nicholas R. Rodofile, Kenneth Radke, and Ernest Foo. 2016. DNP3 Network Scanning and Reconnaissance for Critical Infrastructure. In *Proceedings of the Australasian Computer Science Week Multiconference (ACSW '16)*. ACM, New York, NY, USA, Article 39, 10 pages. <https://doi.org/10.1145/2843043.2843350>
- [33] Martin Roesch. 1999. Snort - Lightweight Intrusion Detection for Networks. In *Proceedings of the 13th USENIX Conference on System Administration (LISA '99)*. USENIX Association, Berkeley, CA, USA, 229–238. <http://dl.acm.org/citation.cfm?id=1039834.1039864>
- [34] Siemens. 2017. *SIMATIC RTU3030C Complete DNP3 Device Profile*. Siemens. Rev. 2.0.
- [35] C. Singh, A. Nivangune, and M. Patwardhan. 2016. Function code based vulnerability analysis of DNP3. In *2016 IEEE International Conference on Advanced Networks and Telecommunications Systems (ANTS)*. 1–6. <https://doi.org/10.1109/ANTS.2016.7947865>
- [36] A. Valdes and S. Cheung. 2009. Communication pattern anomaly detection in process control systems. In *2009 IEEE Conference on Technologies for Homeland Security*. 22–29. <https://doi.org/10.1109/THS.2009.5168010>
- [37] A. F. J. von Gordon and G. P. Hancke. 2004. Protocol conversion for real-time energy management systems. In *IEEE International Workshop on Factory Communication Systems, 2004. Proceedings.* 319–322. <https://doi.org/10.1109/WFCS.2004.1377736>
- [38] Jeong-Han Yun, Sung-Ho Jeon, Kyoung-Ho Kim, and Woo-Nyon Kim. 2013. Burst-based anomaly detection on the DNP3 protocol. *International Journal of Control and Automation* 6, 2 (2013), 313–324.
- [39] Kim Zetter. 2016. INSIDE THE CUNNING, UNPRECEDENTED HACK OF UKRAINE'S POWER GRID. <https://www.wired.com/2016/03/inside-cunning-unprecedented-hack-ukraines-power-grid/>

A SELECTED PACKET ATTRIBUTES

Protocol Layer	
Ethernet	Packet Length Packet Timestamp Src/Dst MAC Address
IP	Src/Dst IP address
DNP3 Link Layer	DNP3 length Link Layer Control
DNP3 Appl Layer	Function Code Internal Indications Field Sequence Number

B BURSTS REQUIRED FOR ATTACK 1 WHITE MODEL APPROACH ASSUMING 300 OUTSTATIONS

C DNP3 INTERNAL INDICATIONS FIELD ATTACK SURFACE COMPARISON

Internal Indications	Attack Surface
	Us Singh[35]
Broadcast	—
Unreported Events	—
Need Time	—
Local Control	✓ —
Device Restart	✓ —
Buffer Overflow	✓ —
Config Corrupt	—

D THE DNP3 PROTOCOL

In process automation DNP3 is one of the de facto protocols used for communication between components. The primary domains of use are utilities like water and electric companies. These companies spend hundreds of thousands of dollars to build and maintain their

Figure 12: DNP3 Protocol Stack

SCADA networks. In the power distribution network where our data was collected the master stations/CCs (control centers) talk to RTUs to gather summarized information from the IEDs in the field. This information is usually provided by the RTU after a request from the CC, but can also be event triggered or unsolicited.

The DNP3 protocol contains three layers from the OSI model for networks the link, transport, and application layer. While DNP3 can operate over a serial bus connection, many modern day SCADA systems run it over a TCP/IP network. When using TCP/IP, the DNP3 protocol messages are sent via a TCP stream and Figure 12 shows the network protocol stack.

D.0.1 DNP3 Link Layer. As shown in Figure 12 the DNP3 link layer can be compared with the Data Link layer from the OSI model. The link layer is broken down into 10 bytes, and the first two are magic bytes with the value 0x0564 simply indicating the frame is DNP3. Next is a one byte length field which gives the length of the remainder of the packet (not inclusive of the CRC lengths). Then there is a one byte control field and a two byte destination and source field which specifies the receiver and sender of the frame. Finally, the link layer concludes with a two byte CRC.

Figure 13: DNP3 Link Layer [12]

D.0.2 DNP3 Transport Layer. The transport layer is one byte long and is used primarily for DNP3 fragmentation purposes. The FIN and FIR bits shown in Figure 14 are used to express whether or not the frame is the first or final in a sequence of packets. The remaining six bits specify the sequence number and can be used by the receiver for fragment reassembly. If the message is only one fragment both the FIN and FIR bits will be set.

D.0.3 DNP3 Application Layer. The application layer combines an application service data unit (ASDU) and an application protocol control info (APCI) block to make an application protocol data unit (APDU). The transport layer breaks the APDU into segments with

Figure 14: DNP3 Transport Layer [12]

a maximum size of 16 bytes. It then packages them with an 8-bit transport control header and 16-bit segment CRC separators into a transport frame. The link layer adds a header to the control and addressing information to prepare the packet for delivery to a single destination [11].

Figure 15: DNP3 Application Layer [12]

DNP3 application layer packets consists of an application layer header and a series of zero or more object ranges (headers followed by the object data), an example of this is shown in Figure 15.

The application layer header contains a one byte application control field, one byte function code field, and a two byte internal indications field. The application control field is made up of an octet and is responsible for allowing fragmentation and signifying whether the message is a confirmation or if it is unsolicited. The first two bits of the octet are FIR and FIN, when set they indicate that the current message is either the first or final application layer fragment. The next bit is CON or confirm, when set it indicates that the receiver of the message should return a confirm message. The next bit is UNS or unsolicited, and when its set it means the current message was unsolicited. The final 4 bits are the SEQ field and they represent the sequence number of the message. The SEQ field is useful for verifying the order of fragments and/or to detect duplicates. Figure 16, shows a graphical representation of the application control octet.

Figure 16: Breakdown of DNP3 Application Control Octet [12]

The function code field is one byte in length and specifies the operation the target (or destination) DNP3 device should perform. Function codes are divided into response and request messages, where responses are further categorized based on whether the response was unsolicited or not. Requests messages have more variability than responses and can be FREEZE, CONFIRM, READ, WRITE, etc. The internal indications field is a two-byte field that is only included if the function code field is a response. The main purpose of internal indications is to offer further insight to the status of an operation

that was performed. Some examples of this are time synchronization requests (need time), events that have yet to be reported to the master (class 1/2/3 data available), broadcast received (all stations). Figure 17 shows the full list of DNP3 application layer function codes and internal indication values.

Function Codes		Internal Indications	
Requests (Hex)		LSB	
0	Confirm	10	Initialize application
1	Read	11	Start application
2	Write	12	Stop application
3	Select	13	Save configuration
4	Operate	14	Enable unsolicited
5	Dir operate	15	Disable unsolicited
6	Dir operate – No resp	16	Assign class
7	Freeze	17	Delay measurement
8	Freeze – No resp	18	Record current time
9	Freeze clear	19	Open file
A	Freeze clear – No resp	1A	Close file
B	Freeze at time	1B	Delete file
C	Freeze at time – No resp	1C	Get file information
D	Cold restart	1D	Authenticate file
E	Warm restart	1E	Abort file
F	Initialize data		
Responses (Hex)		MSB	
81	Response	IIN1.0	All stations
82	Unsolicited response	IIN1.1	Class 1 events
		IIN1.2	Class 2 events
		IIN1.3	Class 3 events
		IIN1.4	Need time
		IIN1.5	Local control
		IIN1.6	Device trouble
		IIN1.7	Device restart
		IIN2.0	Function code not supported
		IIN2.1	Object unknown
		IIN2.2	Parameter error
		IIN2.3	Event buffer overflow
		IIN2.4	Already executing
		IIN2.5	Configuration corrupt
		IIN2.6	Reserved 1
		IIN2.7	Reserved 2

Figure 17: DNP3 Application Layer Function Codes and Internal Indications [12]

DNP3 object ranges consist of as little as none and to as many objects as can be fit within the 255 byte DNP3 packet size limit. The object group and variation (known as the object type) make up the first 2 bytes of the object header shown in Figure 15 and a more detailed representation of the object header structure is shown in Figure 18. The supported object types vary from device to device, but the information can be found either in the manufacturer documentation or the device data object library. The qualifier field is made up of an object prefix code and a range specifier code, these are used to determine the indexing structure of the objects to follow. Specifically, the object prefix code tells whether each object will begin with an index or a size field and the range specifier code gives the range field type. Finally the last field of the object header is the range field which states the number of objects within the range.

Figure 18: Single DNP3 Object Header [12]