You Snooze, You Lose: Measuring PLC Cycle Times under Attacks

Matthias Niedermaier
University of Applied Sciences Augsburg
18 PUBLICATIONS **97** CITATIONS
SEE PROFILE

Jan-Ole Malchow
Freie Universität Berlin
7 PUBLICATIONS **92** CITATIONS
SEE PROFILE

Florian Fischer
University of Applied Sciences Augsburg
14 PUBLICATIONS **61** CITATIONS
SEE PROFILE

Daniel Marzin
Freie Universität Berlin
3 PUBLICATIONS **89** CITATIONS
SEE PROFILE

You Snooze, You Lose: Measuring PLC Cycle Times under Attacks

Matthias Niedermaier[1], Jan-Ole Malchow[2], Florian Fischer[1], Daniel Marzin[2],
Dominik Merli[1], Volker Roth[2] and Alexander von Bodisco[1]

[1]Hochschule Augsburg, Augsburg, Germany
[2]Freie Universität Berlin, Berlin, Germany

Abstract

In this work, we show that the electrical side of a Programmable Logic Controller (PLC), that is, the controlled process, can be influenced by packet flooding. This differs from already known Denial of Service (DoS) attacks as the target is the actual process and not network connectivity. We conducted our experiments with 16 devices from six vendors, giving a good overview of the current market. Except for one device, all are susceptible to network flooding attacks. In three cases, an attack even lead to a DoS on the electrical side, completely disrupting any controlled process. In addition, we show that well-known scanning tools have measurable impacts on PLCs. These findings should be taken into consideration by administrators and researchers planning scanning activities.

1 Introduction

Programmable Logic Controllers (PLCs) are pervasive in modern societies and form a vital part of modern infrastructure. Nearly all aspects of automation are controlled by them in one way or another. Controlled systems include air-conditioning, traffic lights, factories, and power plants. Security and safety violations in these systems can lead not only to inconveniences but also risks to human lives.

When PLCs were first introduced, it was uncommon for them to be interconnected on a larger network. Meanwhile, PLCs come with Ethernet interfaces and are increasingly connected to TCP/IP networks due to benefits related to cost and convenience. This makes PLCs vulnerable to network-based intrusion.

PLCs run control programs that can be thought of as the software implementation of a switching circuit. Control programs use sensor data as input and set outputs to activate actors. In the following, we refer to the sensor and actuator connections as the *electrical side* of the PLC.

We focus on the question whether network traffic can influence the electrical side of PLCs. If the electrical side can be influenced, then a controlled process may be disturbed or even stop altogether. Obviously, such a capability can be used in cyberattacks. This question is also relevant when scanning the internet for benign purposes, which is currently a trend in academic research. If scans potentially affect controlled processes, then enhanced precautions are required to assure the safety and security of (largely unknown) scan targets. We are not interested in flooding attacks that seek to saturate a network or a network interface in order to deny service to communicating devices.

To assess the risks that arise in controlled processes from network traffic, we conducted three types of experiments in a testbed with 16 PLCs from six different vendors. We explored the effects of: 1. SYN flooding, 2. fourteen high-level protocols and 3. three popular scanning tools on the electrical side of PLCs. To this end, we used a control program that switched the outputs of PLCs at the maximum supported rate (e.g. freewheeling task) and measured deviations from that rate. Various settings of lower switching frequencies can be used as a benchmark as well. We decided against this because the maximum rate is conservative and application-neutral.

We found that all except one PLCs are prone to being influenced by network traffic. Most PLCs were affected by SYN packet floods. The effects of high-level protocols varied for different PLCs. Several PLCs stopped working completely, resulting in a Denial of Service (DoS) condition on the electrical side. However, we also found that data rate-limiting features available on Wago PLCs can reduce the effects.

The rest of the paper is organized as follows. We begin with a description of related work in § 2. We provide background information on the functions of a PLC and PLC certification in § 3. In item 3.3, we summarize our experimental methods and materials. We report the results of our experiments in § 5 and provide conclusions in § 6.

2 Related Work

DoS attacks on SCADA/PLC/ICS systems have been a topic in academic research since at least 2005 [4, 11]. However, most studies only outline the potential of attacks and do not present evidence derived from experimentation or simulation. In what follows we limit our discussion to the literature that provides at least partial evidence for possible DoS attacks.

Teixeira et al. [20] describe a variety of attacks on control systems. They focus on the disruption of communication between sensors/actuators and a PLC but overlook the effects on the electrical side. The authors of [2] present a formalization of DoS attacks on control systems and derive an 'optimal' attack plan. However, they do not evaluate their attack plan against an actual PLCs. [9] conducted flooding experiments on an unspecified remote telemetry unit (RTU) based on IP, SYN, and 104APCI packets. In all cases, they measured an impact on the response time of the RTU. However, their report lacks clarity with respect to what exactly caused the effects they measured. The reasons for this may range from RTU resource depletion to the saturation of other components in their test network. The authors of [12] simulated User Datagram Protocol (UDP) flooding attacks in a Supervisory Control And Data Acquisition (SCADA) network model. They concluded that CPU utilization, packet drop, and traffic delays increased. In [11], the impact of DoSs attacks on network-based control is simulated and two countermeasures are proposed. The authors focus on the communication without analysing the behaviours of the devices. A method of testing the communication robustness of industrial devices is introduced in [21]. However, their article mentions no concrete results. [16] set up a testbed with an *Omron PLC CJ1M-CPU11-ETN* and demonstrated DoS attacks on the network interface of the device based on TCP/IP SYNs, UDP, and HTTP traffic. They did not measure effects on the electrical side, nor did they test different PLCs systematically as we did in our experiments.

3 Technical Background

Programmable Logic Controllers (PLCs) are industrial digital computers designed to control physical processes. A PLC is electrically connected to sensors and actuators. A user-specific program running on the PLC controls the actuators based on the inputs read from the sensors. Since the majority of PLCs operate in a cycle-oriented fashion (see next section for details), we focus on this type of device.

PLCs are usually part of a larger architecture that includes Enterprise Resource Planing (ERP) and Manufacturing Execution System (MES). The latter systems often have a data exchange time of several hours or days. For SCADA systems, a common requirement is that data transfer time must be in the order of seconds to minutes. This is in contrast to mostly hard real-time processes at the control and field levels, where transmissions must complete in milliseconds. These timings must be ensured in order for the processes to run correctly.

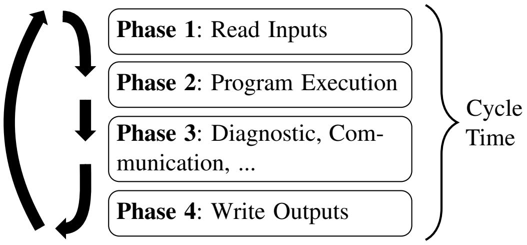
Phase 1: Read Inputs
Phase 2: Program Execution
Phase 3: Diagnostic, Communication, ...
Phase 4: Write Outputs
Cycle Time

Figure 1: Simplified sequence of a PLC cycle.

3.1 PLC Cycle Time

The *run mode* of a cycle-oriented PLC consists of a loop of four phases, as illustrated in Fig. 1. In the first phase, inputs such as sensors are read into the internal registers of the PLC. In the second stage, the program execution is performed. The third phase handles internal housekeeping, for example diagnostic functions and communication. At the end of the scan cycle, the outputs are written back from internal registers to the electrical circuits. Typical cycle times are between one and 10 milliseconds. In more powerful models, or small programs, cycle times may be in the order of microseconds. There are versions with either fixed or asynchronous cycles. The user program may include branches and conditional calls, resulting in varying execution times.

3.2 Controlled Process

Fig. 2 shows a simple example application where a PLC controls the filling of a container on a conveyor belt. The sensor reports to the PLC when a container passes it. The PLC then controls the valve that opens and fills the container. This process must have the right timing, or else the liquid would not end up in the container. If the cycle time is too high, the opening or closing of the valve gets delayed and occurs at false container positions [6].

3.3 Certification Programs

There are three certification programs for Industrial Internet of Things (IIoT) components. In the following, we mention three such programs which were previously discussed by Schierholz and McGareth [17], and Xie et

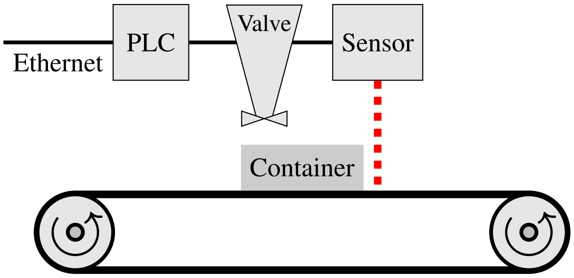

Figure 2: Example application where liquid/goods is filled in a container

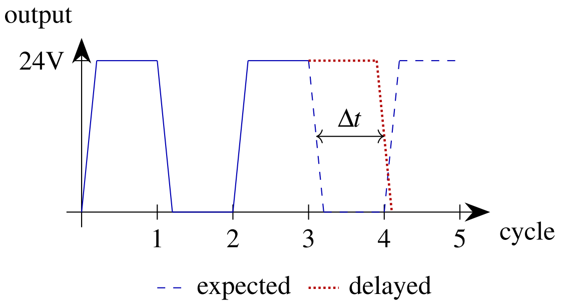

Figure 3: Electrical view of a PLC toggling an output.

al. [23]. Certificates of these three programs communicate an acceptable level of stack robustness. Schierholz and McGareth argue that security-related certificates may send incorrect signals regarding security. This is primarily because not all threat vectors may be covered by a certification program. Our reports support this argument as we found that nearly all PLCs we tested are vulnerable to network flooding attacks. We give a short overview of the mentioned programs with respect to network robustness.

(i) *Achilles Certification*[1] – Initially developed by Wurdtech Security Technologies, the Achilles Program was later bought by General Electric. The program relies on a proprietary test device called the 'Achilles Satellite'. Applied tests include protocol fuzzing and packet storms. We are especially interested in the packet storm sub-test. While the Satellite is proprietary, the requirements for a certification are publicly documented. For the level 2 certification of Achilles, the PLC is configured with a period cycle output of 1000ms (500ms high output and 500ms low output) with an acceptable tolerance of 4%.

(ii) *ISASecure EDSA Certification*[2] – The EDSA includes *CRT Test Requirements for Protocols* for Ethernet, ARP, IPv4, ICMPv4, UDP, and TCP. With the exception of Ethernet, the requirements state that the device under test maintains its essential services under high load but can reduce or cease network communication during periods of high load. In all cases, the high load period (maximum supported data rate) must be long enough to allow saturation effects to manifest.

(iii) *Mu Dynamics MUSIC Certification*[3] – Mu Dynamics Inc. was acquired by Spirent Communication Inc. in 2012. The current status of the certification program is unknown. According to Xie et al. [23], MUSIC operated similarly to Achilles.

The basic idea of a PLC cycle time attack is to influence the timing of a PLC by means of network traffic. In other words, the attacker aims to alter the timing of PLC outputs. Wedgbury and Jones [22], as well as Cárdens [5], already predicted that extra network traffic might affect the process controlled by an Industrial Control System (ICS). However, they did not present evidence for their prediction. Our experiments lend support to their assertion because we found that network traffic can affect user programs running on PLCs.

The attack surface is a combination of device design and software implementation; more precisely, it is the implementation of the network stack, PLC-specific protocols, and PLC runtime. For example, sharing resources between system tasks and the actual control program can be problematic. If an attacker is able to exhaust the resources available to system tasks, he also succeeds in preventing normal operation of the control program.

3.4 Attacker Model

We assume that the attacker is able to send network packets to the target PLC at the maximum rate supported by the device. This may be feasible because the device is connected to the internet, or another device on the same network is compromised by the adversary. The compromised device may well be another PLC [10]. With regard to the types of attacks we consider, we do not assume that the adversary has or needs specific knowledge about the actual process controlled by the PLC or the program running on the PLC.

4 Materials and Methods

The basic idea is to measure the changes to the signal captured on the electrical (digital) outputs of PLCs. We conducted three sets of experiments. In the first set (§ 5.1), we focused on the reaction of devices to different loads of SYN packets. In the second set (§ 5.2), we measured the reaction to different protocols including device-specific control protocols. In the final set of experiments (§ 5.3), we assessed the impact of scanning tools. In the remainder of this section, we give an overview of our methods and materials.

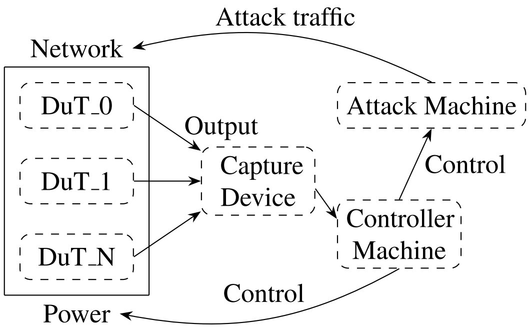

Figure 4: Test set-up for the measurement.

Regarding the electrical side, we configured the PLCs under test to run on their maximum performance (shortest possible cycle time). This means that an output is switched at the maximum rate. Depending on the actual device, this leads to a more or less periodic reference signal. If an attack is successful, the reference signal will be shifted. Fig. 3 depicts the reference signal (blue, solid) and the shifted signal under attack (red, dotted). For the attack scenario proposed in this paper, the attacker does not need to know the details of the ICS.

We expected the impact of attacks on the cycle time of a PLC to differ across devices. This is due to differences in the system design, quality of implementation, and possible safety mechanisms. For example, some manufacturers indirectly tie cycle time to the cost-efficiency of their devices, since the manufacturing process can be operated at a higher speed if the cycle time is shorter. An extreme example is provided by Schneider Electric [3], where a reduction in the cycle time from 30ms to 6ms resulted in the gain of two million dollars per year.

In our experiments, we flooded the device under test with packets for a specific protocol and measure the cycle time of the device. The used protocols are depicted in Tab. 1. We designed and implemented a test set-up for our experiments. The set-up comprises a capture device, an attack machine, and a controller machine (see Fig. 4). The capture device can digitize the outputs of the PLCs. The attacker machine generates traffic for the respective protocol under test. The controller device starts and stops the attack traffic, and stores the data sent by the capture device. It has the option to power on and off the Device under Tests (DuTs).

In the following, we detail our measurement set-up and test cases.

4.1 Capture Device

The influences of the different test cases are monitored with a capture device that measures one digital output pin of each PLC. We use a BeagleBone Black from Texas Instruments with a custom measurement Printed Circuit Board (PCB) to handle the 24V square wave signal. The PCB consists of a protection circuit and a level shifter Integrated Circuit (IC). The BeagleBone runs a Debian Linux with the BeagleLogic application [4], which makes use of the AM335x processor's two programmable real-time units. It is possible to analyse up to 14 channels in a continuous mode with a maximum of 100 Mega samples per second (Ms/s). The Ethernet interface (100 Mbps) was used to send the data to a computer for further analysis. We captured on a fixed rate of 1 MHz, which allowed us to calculate the timing without an additional timestamp. We were only interested in the state of the output of the device currently under test. Therefore, a byte per sample was needed to transfer data over the network, leading to a feasible data rate.

To ensure the validity of our tests, we used a function generator and a Picoscope 2208[5] oscilloscope to measure the capabilities of the BeagleLogic. We used 1 Ms/s on the BeagleLogic, which is also the sample rate in the test set-up. The deviation of the BeagleLogic adapter compared to the function-generator was always below 0.1

4.2 Controller Machine

The controller machine is a standard PC with two network interfaces. One interface is connected to the capture device and the other to the attacker machine. The controller machine runs a custom experimental server written in Go. The server reads the definition of an experiment defined as a JSON file. An experiment defines the tool to use specific parameters, the target to measure, the channel to capture, and the runtime of the experiment. Based on this definition, the controller configures the capture device and attack server. In addition, the control machine stores the data produced by the capture device.

4.3 Attacker Machine

The attacker machine is a default PC with two Gigabit Ethernet network interfaces. One interface is connected to the DuTs, while the other is connected to the controller machine. The attacker machine runs a custom experimental client that connects to the corresponding experimental server on the control machine. The purpose of the client is to start and stop the actual load generating program. The tools used for load generation are listed in Tab. 1.

4.4 Device under Tests (DuTs)

The DuTs are PLCs from different vendors. We selected a variety of devices in order to get a representative sample of the current market. A summary of the currently deployed PLCs in our testbed [15] is given in Tab. 2.

Table 1: Overview of programs used, corresponding protocols, and respective parameters

Program	Protocols	Parameters
ZGrab[1]	S7comm / HTTP(S) / Modbus/TCP / Ethernet/IP / DNP3 / Bacnet/IP	`-s7 --port 102 / --port 80 --http="" / --port 443 --tls --http="" / -modbus --port 502 / -dnp3 --port 20000 / -enip --port 44818`
Vegata[2]	HTTP	`attack`
hping3[3]	SYN / UDP	`-c 1 -1 -C 17 / -S -P -U --flood`
syn_spam[4]	SYN	`-worker 20`
arp_spam[4]	ARP	`-worker 20`
gre_spam[4]	GRE	`-worker 20`
snmp_spam[4]	SNMP	`-worker 20`

Table 2: Currently deployed devices in our test set-up

No.	**Vendor**	**Manufacturer number**	**Name**	**Firmware**
1	Wago	750-889	Controller KNX IP	01.07.13(10)
2	Wago	750-8100	Controller PFC100	02.05.23(08)
3	Wago	750-880	Controller ETH.	01.07.03(10)
4	Wago	750-831	Controller BACnet/IP	01.02.29(09)
5	Siemens	6ES7211-1AE40-0XB0	Simatic S7-1211*	V4.2.0
6	Siemens	6ES7212-1AE31-0XB0	Simatic S7-1212	V 3.0.2
7	Siemens	6ES7155-6AU00-0AB0	Simatic ET 200SP	V 3.3.0
8	Siemens	6ES7314-6EH04-0AB0	Simatic S7-314*	V 3.3.0
9	Siemens	6ES7516-3FN01-0AB0	Simatic S7-1516F*	V 2.0.5
10	Siemens	6ED1052-1CC01-0BA8	Logo! 8*	1.81.01
11	Phoenix	2700974	ILC 151 ETH	V.4.42.04
12	Phoenix	2985330	ILC 150 ETH	V.3.94.03
13	Phoenix	2700975	ILC 171 ETH 2TX	V.4.42.04
14	ABB	1SAP120600R0071	PM554-TP-ETH	2.5.4.15626
15	Crouzet	88981133	em4 Ethernet	1.2.75/1.0.27
16	Schneider	TM221CE16T	Modicon M221	1.5.1.0

* Achilles Level 2 Certified

We aimed to identify and measure a worst-case scenario. Hence, each PLC was configured to switch a digital output at the maximum rate. This was configured in a cyclic task and only changed if necessary (e.g. freewheeling task). This called for device-specific configurations, especially setting the cycle time to the device-specific minimum if applicable. We emphasize that we used the default settings for all controllers, wherever possible. Of special interest are parameters for communication overhead. For the used Siemens devices, we kept the default at 20 %. Wago allows setting a data rate limit; however, this setting was disabled by default (see § 5.4 for effects of this setting). The used control program was simple; it only switched the value of an output from 0 to 1, and vice versa.

4.5 Protocol Implementations

In Tab. 1, we summarize the used protocols. For most of the protocols, we used off-the-shelf tools. If no standard tool was available, we implemented our own tool. With the off-the-shelf tools, we did not have much control over the sent packets. As a result, we used custom implementations for some protocols. All custom tools were implemented in Go and were capable of saturating the outgoing Gigabit Ethernet link of the attacker machine.

syn_spam – This implementation uses hard-coded SYN packets with no additional TCP options set.

arp_spam – RFC 826 defines multiple variants for ARP requests. The standard uses the following abbreviations: sender protocol address (*SPA*), sender hardware address (*SHA*), target protocol address (TPA), and target hardware address (THA). We implemented the following four ARP request variants: 1. Who has 2. Probe 3. Gratuitous ARP Request $TPA = SPA$, $THA = 0$ 4. Gratuitous ARP Reply $TPA = SPA$, $THA = SHA$.

gre_spam – This implementation uses GRE-encapsulated SYN packets. The SYN packets do not have any additional TCP options. We tested with GRE packets as modern DoS attacks sometimes use such packets [18].

snmp_spam – Our implementation uses SNMPv1 with a hard-coded community string: `302902010004067075626c6963a01c0204036a5f430 20100020100300e300c06082b060102010101000500`.

4.6 Methods

Although the actual procedure differed across the three sets of experiments, the basic procedure remained the same. Prior to each experiment, the DuTs were powered off and on so as to start with a clean system state.

To make the experiments more convenient, the execution of individual experiments was automatized. To this end, the individual experiments were combined in a single large experiment definition for the experimental server. The gathered data was stored on the control server in a single file per phase and experiment. After all the experiments had been executed, the resulting files were downloaded for analysis.

5 Experiments, Results, and Discussion

In this sections, we present the three series of experiments we conducted. In the first series, we measured reaction of devices under different loads of SYN packets. In the second series, we measured the reaction to different pro-

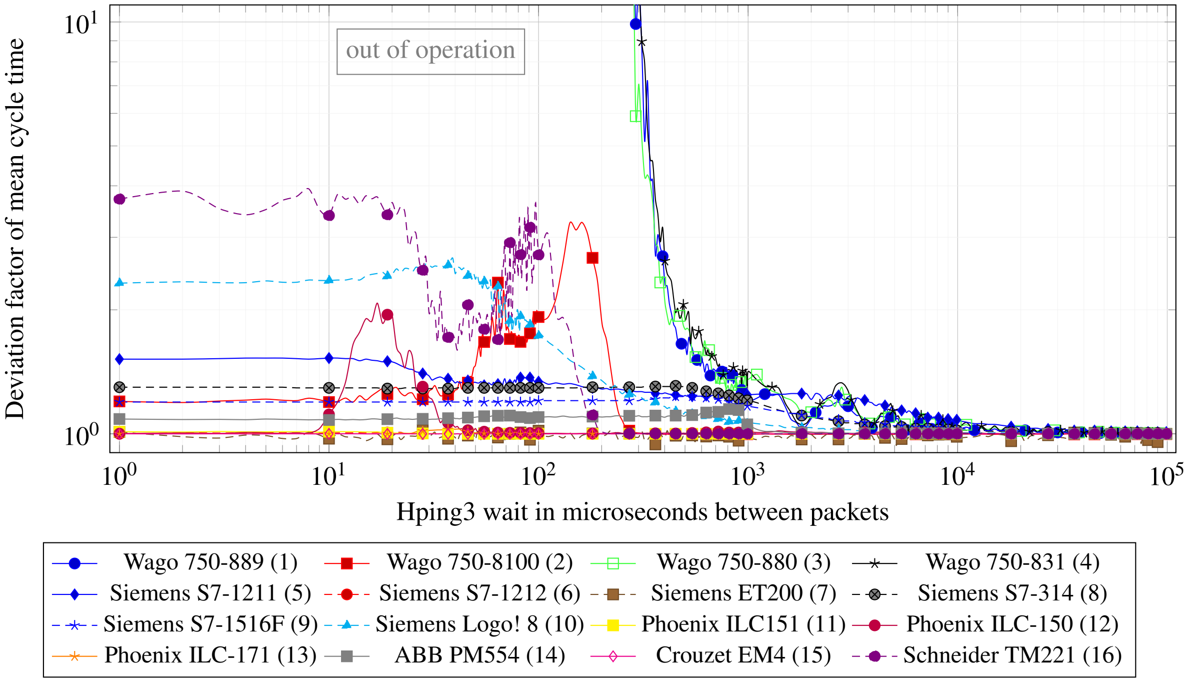

Figure 5: Controlled attack on PLCs with delays during packets, to achieve different network loads.

tocols. In the final series, we assessed the impact of scanning tools.

5.1 Increasing SYN Loads

As a baseline for the communication robustness of the tested devices, we performed a series of tests (hping3 SYN flood) with increasing inter-packet delay. Every hping3 attack lasted 60s followed by a 30s idle phase. The delays between the flooding was created by the wait parameter of hping3 (`hping3 -i u<wait for x microseconds> <IP>`). Through this, after each packet, hping3 waited x microseconds until the next packet was sent. We used the resulting mean cycle time for comparison. The mean cycle time of each segment was calculated as shown in Equation 1.

$$\bar{t} = \frac{1}{n} \cdot \sum_{i=1}^{n} t_i \tag{1}$$

For better comparability, we normalized the results by dividing them by the mean idle time.

$$\Delta t = \frac{\bar{t}_{mean}}{\bar{t}_{idle}} \tag{2}$$

An overview of the results is given in Fig. 5.

We found that for some PLCs (5, 8, 9, 10, 14, and 16) a higher network load led to higher cycle times.

For some controllers (1, 3, and 4), we even observed an ‘out-of-operation’ state under specific data rates. We defined a device as out of operation if its cycle time was increased by a factor 10 or more.

Some PLCs (2 and 12) were not influenced at the maximum packet flooding but at lower rates. This shows that it is not always useful to execute a DoS attack at the maximum available data rate.

During the hping3 measurement, the mean cycle time of the Siemens ET200 (7) somewhat decreased, meaning that the device runs fast at different packet rates.

Furthermore, four devices (6, 11, 13, and 15) in the testbed were not influenced by the hping3 flooding attacks. However, most of the PLCs were affected, and further analysis showed that only the Crouzet em4 (15) was not influenced at all by our tests.

Conducting all the experiments summarized in Fig. 5 took about a month. These experiments show that most devices can be influenced by sending SYN packets at a defined rate. Since SYN packets already have an influence on devices, it can be expected that higher-level protocols such as Hypertext Transfer Protocol (HTTP), Simple Network Management Protocol (SNMP), and ICS-specific protocols will be even more effective. This is due to additional resource consumption at higher levels of the network stack. In the following, we present a more detailed analysis of this phenomenon.

5.2 Detailed Analysis of Protocols

Each experiment in this series consisted of four phases. First, the device to test was powered off and on to guarantee a clean system state. The actual attack phase was flanked by two idle phases. The idle phase prior to the attack served as a reference to determine the impact of the attack. The post-attack idle phase was intended to observe any possible long-term effects of the attack. Each phase lasted for 600 seconds. There was a 60-second break between successive experiments.

Different impacts on the PLCs cycle time during the attacks could be observed. Due to space constraints, we categorized the impact into six different effect classes. For each class, we only present the worst case observed. The results are detailed in the Appendix A.

The results of the measurements are shown in a boxplot with calculated arithmetic mean (▼) and median (–). The quantiles are respectively 25 % and 75 %, with whiskers up to factor 1.5 of the box.

5.2.1 Class 1: PLC 'Stops'

An extreme behaviour is that the PLC 'stops' during the attack. This means that the outputs are not updated during packet injection. Fig. 6 shows this behaviour during an Address Resolution Protocol (ARP) flood attack.

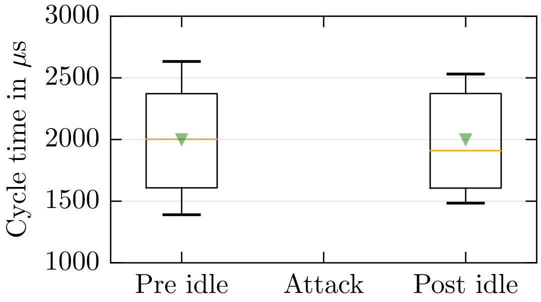

Figure 6: Boxplot of a Wago 750-831 (4), where the PLC stops during ARP 3 flooding.

It is worth noticing that an ARP flooding attack can be sent to the whole broadcast domain. Therefore, all devices that can be influenced in a broadcast domain can be affected by this type of attack. However, ARP requests do not cross subnet boundaries, and as such only local adversaries can apply ARP flooding attacks.

In the example given in § 3.2, the valve remains open if it is opened when the attack has started. Thus, the material will not be filled into the container but next to it. This can obviously lead to all sorts of trouble.

Devices in this class clearly exceed the requirements for a certification as described in § 3.3.

5.2.2 Class 2: High Deviation

During a flooding attack, the cycle time of some controllers increases by several seconds. In the measurement illustrated in Fig. 7, the cycle time increases up to 5 seconds. In the example, this influence is achieved through UDP flooding. During the pre- and post-idle phase, the PLC functions as expected and toggles with about 2 ms.

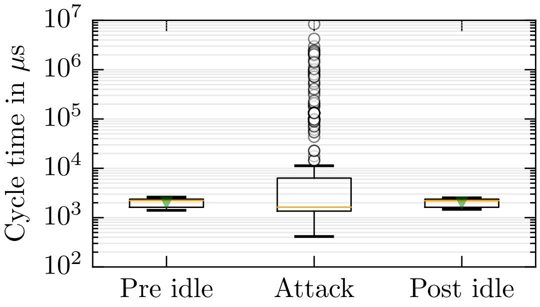

Figure 7: Boxplot of UDP flooding attack on a Wago 750-889 (1), resulting in a high deviation of the cycle time

Considering the example in § 3.2, the PLC will nearly stop reacting. More precisely, the outputs will remain at the current level (on or off), only being updated every few seconds. This means that, if the valve is opened at this moment, it will remain opened for several seconds, and the convey belt will still move forward, resulting in a similar effect to the one described before.

Devices in this class break the requirements for certification as described in § 3.3. Neither do the devices maintain essential services, nor is the deviation smaller than 4 %.

5.2.3 Class 3: Medium Deviation

Another effect that can be observed is a 'medium' deviation of the cycle times. Devices in this class show increased cycle times below one second. Fig. 8 shows an example. The device toggles in idle with about 2 ms. During UDP flooding, the cycle time is up by a factor of about 40.

The controller processes everything at a slower rate due to this factor. It is possible that a process is still running correctly, but at a much slower pace or imprecisely. Considering the example in § 3.2, the container may have already passed the valve when the sensor input is processed. Therefore, the loading could miss the container.

As for classes 1 and 2, the criteria for certification would not be met.

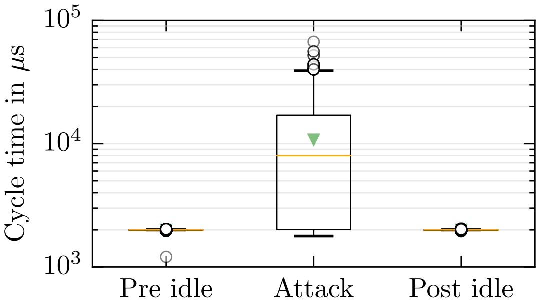

Figure 8: Boxplot with medium deviation during UDP flooding with hping3 of the Schneider TM221CE16T (16).

5.2.4 Class 4: Increased Variance of Cycle Times

With regard to the results in Fig. 9, the cycle time is only minimally affected by packet flooding attacks. The boxplot as well as the mean value shows a delay of about 25 %. However, the variance is still large under the attack. On some controllers, the boxplots and mean value representations are misleading. In fact, there may be effects which are only viewable in other representations.

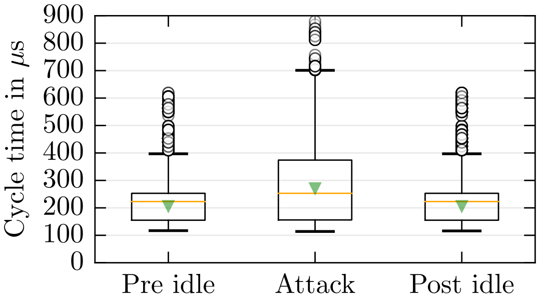

Figure 9: Boxplot, while an attack on a Siemens S7-314 (8) is generating a high network load with the S7Com implementation of zgrab.

Fig. 10 shows the kernel density estimation in a histogram plot. The number of bins is set to 1,000 in order to get a good resolution of the distribution. In this, the cycle time is plotted against their probability (density). With this representation, the influenced cycles are clearly visible.

The density plot of the cycle time shows two peeks in idle, for the electrical low and high signals. We noticed that the low and high signals do not have the same length. In fact, the high signal is longer than the low signal. During the attack, the cycle time increases and new peeks are formed. The two peeks are shifted by a factor of about 2, which is not obvious in the boxplot but is visible in the density representation. This in turn means that some cycle times are twice as slow. Regarding our example (§ 3.2), the result would be variable filling quantities.

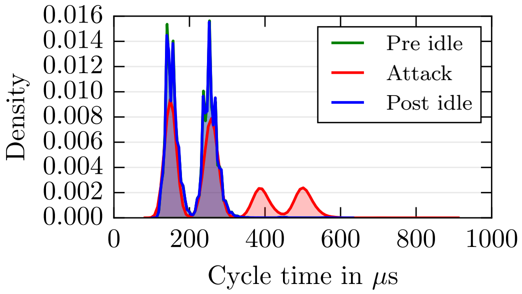

Figure 10: Probability Density Function, to view the distribution during the S7Com flooding of a Siemens S7-314 (8) with zgrab.

For devices in this class, it is not entirely clear if they would fulfil the requirements of the certifications. This is mainly due to the relatively broad definition of our classes. However, for the device we selected as example here, the answer is still clear. For the Siemens S7-314 (8) under test in our study, the maximum communication load was set to 20 %. As such the assurance of the device was exceeded. In addition, the Siemens S7-314 (8) is Achilles level 2-certified, but the findings indicate that the device is still susceptible to network-based attacks on the electrical side of the device.

5.2.5 Class 5: Faster Cycle Time

By considering only the mean cycle time of the PLC, no changes can be determined. However, on a closer look, the cycle time appears to be more spread and some cycles become even faster during an attack. An example under a UDP flooding attack is given in Fig. 11.

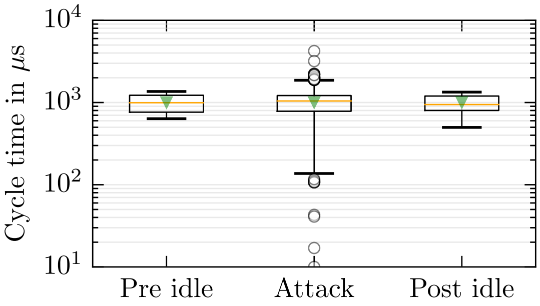

Figure 11: A boxplot representing a shorter cycle time of a Phoenix ILC151 (11) during Modbus/TCP flooding with zgrab.

We believe that this effect is caused by a kind of buffer overflow of the network stack and results in packet drop. Furthermore, maybe this is achieved by blocking or crashing the network stack, thereby allowing the Central Processing Unit (CPU) to process the control process faster. In a real-world example, this could make the process unpredictable if it gets faster than usual. In the context of our example (§ 3.2), the container could not be positioned correctly, or the valve could close earlier than expected, leading insufficient filling.

To the best of our knowledge, the certification programs listed in page 2 do not take into account that PLCs could work faster. As such, devices in this class would meet the requirements while still being prone to attacks.

5.2.6 Class 6: No Measurable Influence

Some tests indicated no measurable influence. Fig. 12 shows an example where the three phases are similar.

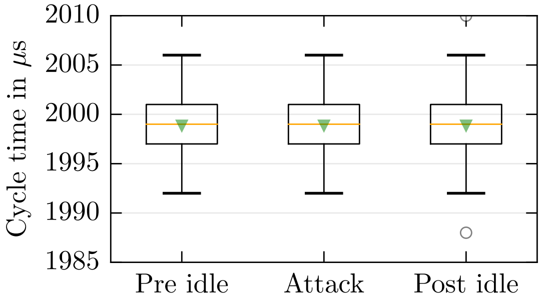

Figure 12: Example of a boxplot with no measurable influence on the Crouzet em4 (15).

5.2.7 CPU Load During Attacks

In our testbed, most devices are based on Real-Time Operating System (RTOS) and the CPU usage cannot be supervised. However, the Wago 750-8100 is based on Linux (with root access), which allows the measurement of CPU utilization during attacks. The device has a single-core 600MHz ARM processor with 256MB of RAM. The flooding attack started after 10 ticks and stopped after 20 ticks. Fig. 13 illustrates the CPU usage during the experiment.

During the attack, the software Interrupt Request (IRQ), which, for example, handles the network traffic, increases to nearly 100 %. In case of an interrupt, the regular software execution is halted and the interrupt is handled. A high interrupt load seems to affect the control software of the PLC, which influences the continuous execution, resulting in asynchronous cycle times.

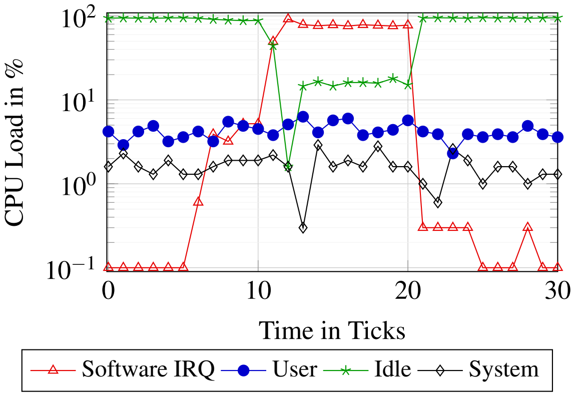

Figure 13: CPU load during SYN flooding attacks of a Wago 750-8100 (2) with hping3

5.3 Effects of Active Scanning

In the literature listed in § 2, it is stated that active scans should be avoided. However, this claim is not backed by empirical evidence. Using our testbed, we were able to precisely assess the influences of an active scan. For this comparison, we used a selection of active scanners (Nmap 7.60[6], PLCScan version 0.1[7], and RiskViz Search Engine[8], which uses ZGrab [7] for application scanning) to analyse the behaviour of ICS components under an active scan. For this measurement, the default configuration of the scanners was used. For this analysis, we selected a control system (Wago 750-880 (3)) which we already knew was influenceable. Fig. 14 summarizes the measured effects of these three scanners compared to the idle cycle time.

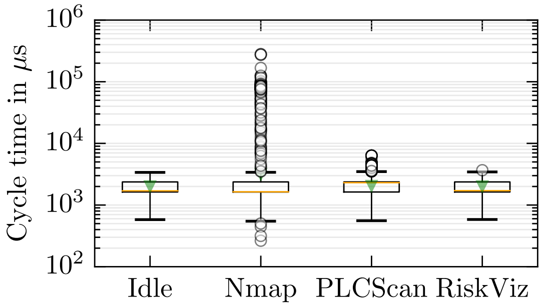

Figure 14: Influences of active scanners on a Wago 750-880 (3).

Fig. 15 illustrates the influences of the cycle time of the three network scanners over the 30-second scan time. The data used for the plots in Fig. 14 and Fig. 15 are from the same scan.

Our analysis of active scanning in ICS networks shows that there are measurable influences for some devices.

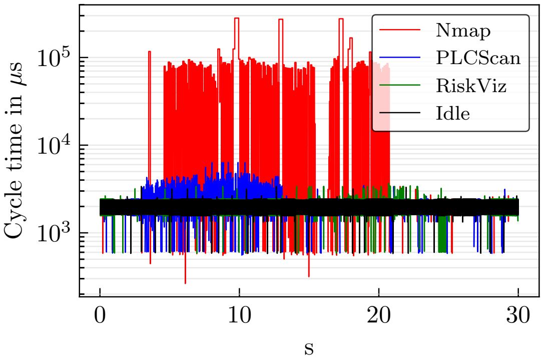

Figure 15: Influences of different network scanners on a Wago 750-880 (3) during network scanning

Therefore, scanning of ICS networks presents a chicken or egg problem. Specific devices should not be scanned. On the other hand, it is not known which devices are in a network prior to a scan. The only option, if a scan cannot be avoided, is to keep the data rate as low as possible.

5.4 Mitigation and Future Work

In order to secure assets, systems, machines, and networks against cyber threats, it is necessary to implement and maintain a state-of-the-art industrial security concept [19]. This includes validation of the communication robustness of single components, for example, with flooding tools and specialized ICS fuzzers [14]. Our results with these testing tools have shown that there is a lack of secure ICS component architectures. Furthermore, existing tests are not vendor-independent or transparent to the public.

Data rate limitations on the network provide a possible software solution. This feature is already implemented by controllers from Wago (1,2,3,4). Our measurements show that this option can be an efficient mitigation (see Appendix A). The Wago 750-8100 is not prone to flooding attacks for data rates of 16 MBit/s and below. The effect of flooding is drastically reduced for the remaining devices for data rates of 1 MBit/s and below. Only the longest measured cycle time is increased. There is no change in the mean cycle times. For data rates of 8 MBit/s and above, the effects measured without the feature are still evident. This possibility of rate-limiting indicates that there are other configuration options which could prevent cycle time influences.

Another software-based solution would be RTOSs with hard real-time scheduling like FreeRTOS [8]. Such schedulers guarantee a certain task tick time. If mapped to PLCs cycle times, the expected characteristics on the electrical side could be guaranteed.

Besides software solutions, specific hardware configurations provide another option [13]. A possible configuration could be a multi-controller set-up, for example, two dedicated controllers, or a System-on-a-Chip (SoC), where one controller process the real-time task and the other controls communication. A challenge in this scenario is to prevent feedback effects between the controllers. A hardware solution is obviously only possible for new products, but it would increase production and integration costs.

6 Conclusion

In this paper, we tested the communication robustness of PLCs under network flooding attacks. Our results show that the electrical side of PLCs is prone to network flooding attacks. Variances in the runtime of control programs can have disastrous effects. This differs from well-known DoS attacks, as in this case physical processes are involved.

Our analysis shows that most of the PLCs are affected, irrespective of manufacturers. With the exception of one device (Crouzet em4 (15)), all the devices in our testbed showed measurable changes during network flooding attacks. Some of the controllers even ‘stopped’ operating and did not update their outputs for the duration of the attack. Additionally, we have shown that active network scans have a detectable effect on the electrical side of PLCs. These results are relevant as active networks scans are a current trend in academic research. Network scans with high data rates may influence internet facing PLCs accidentally. We recommend taking this possibility into account for the risk assessment of a planned project.

Apart from casualties, network-based DDoS attacks are another current trend [18]. This is mainly because network flooding attacks are technically simple. In the presented scenario, an attacker can influence an actual physical process. This increases the thread imposed by DDoS attacks.

To summarize our research, it can be said that a secure system configuration is of great importance. We were glad to see that Wago offers at least a partial function mitigation feature. However, operators need to learn about and use configuration features to enable a secure operation.

We plan to extend our analysis with more devices to provide a broader overview. We informed all affected vendors about our findings using an adapted responsible disclosure.

Acknowledgments

The work on network traffic effects on PLC cycle time is part of the RiskViz [1] research project. It is funded by the Federal Ministry of Education and Research (BMBF), with the aim of creating a risk map of SCADA systems in Germany.

References

[1] RiskViz - Risk Map of the Industrial IT-Security in Germany. Online: `https://www.riskviz.de/`. Accessed: 2018-05-30.

[2] Amin, S., Cárdenas, A. A., and Sastry, S. S. Safe and Secure Networked Control Systems under Denial-of-Service Attacks. In *International Workshop on Hybrid Systems: Computation and Control* (2009), Springer, pp. 31–45.

[3] Boville, J. Productivity Secrets: Don't Underestimate the Power of PLC Scan Time. Online: `https://tinyurl.com/ybrmu3py`. Accessed: 2018-05-30.

[4] Bowen, T. C. L., and Thomas, R. W. A Plan for SCADA Security to Deter DOS Attacks. In *Proceedings of the Department of Homeland Security: R&D Partnering Conference* (2005), Citeseer.

[5] Cárdenas, A. A., Amin, S., and Sastry, S. Research Challenges for the Security of Control Systems. In *HotSec* (2008).

[6] Corporation, E. C. Scan Times. Online: `https://tinyurl.com/yc6jr5e4`. Accessed: 2018-05-30.

[7] Durumeric, Z., Wustrow, E., and Halderman, J. A. Zmap: Fast internet-wide scanning and its security applications. In *USENIX Security Symposium* (2013), vol. 8, pp. 47–53.

[8] Inam, R., Mäki-Turja, J., Sjödin, M., and Behnam, M. Hard real-time support for hierarchical scheduling in freertos. In *23rd Euromicro Conference on Real-Time Systems* (2011), pp. 51–60.

[9] Kalluri, R., Mahendra, L., Kumar, R. S., and Prasad, G. G. Simulation and Impact Analysis of Denial-of-Service Attacks on Power SCADA. In *Power Systems Conference (NPSC), 2016 National* (2016), IEEE, pp. 1–5.

[10] Klick, J., Lau, S., Marzin, D., Malchow, J.-O., and Roth, V. Internet-facing PLCs as a Network Backdoor. In *Communications and Network Security (CNS), 2015 IEEE Conference on* (2015), IEEE, pp. 524–532.

[11] Long, M., Wu, C.-H., and Hung, J. Y. Denial of Service Attacks on Network-based Control Systems: Impact and Mitigation. *IEEE Transactions on Industrial Informatics 1*, 2 (2005), 85–96.

[12] Markovic-Petrovic, J. D., and Stojanovic, M. D. Analysis of SCADA System Vulnerabilities to DDoS Attacks. In *Telecommunication in Modern Satellite, Cable and Broadcasting Services (TELSIKS), 2013 11th International Conference on* (2013), vol. 2, IEEE, pp. 591–594.

[13] Nao, L., Passaro, P., Gioia, E., and Petracca, M. Asymmetric multiprocessing techniques in smart devices: Application in a drone navigation system. In *Software, Telecommunications and Computer Networks (SoftCOM), 2017 25th International Conference on* (2017), IEEE, pp. 1–5.

[14] Niedermaier, M., Fischer, F., and von Bodisco, A. PropFuzz - An IT-Security Fuzzing Framework for Proprietary ICS Protocols. In *2017 International Conference on Applied Electronics (AE), Pilsen* (2017), pp. 1–4.

[15] Niedermaier, M., von Bodisco, A., and Merli, D. CoRT: A Communication Robustness Testbed for Industrial Control System Components. In *4th International Conference on Event-Based Control, Communication, and Signal Processing EBCCSP 2018* (2018).

[16] Sayegh, N., Chehab, A., Elhajj, I. H., and Kayssi, A. Internal Security Attacks on SCADA Systems. In *Communications and Information Technology (ICCIT), 2013 Third International Conference on* (2013), IEEE, pp. 22–27.

[17] Schierholz, R., and McGrath, K. Security Certification–A Critical Review. *ABB Corporate Research* (2010).

[18] Seaman, C. Threat Advisory: Mirai Botnet. Tech. rep., Akamai, 2016.

[19] Stouffer, K., Falco, J., and Scarfone, K. Guide to Industrial Control Systems (ICS) Security. *NIST special publication 800*, 82 (2011).

[20] Teixeira, A., Pérez, D., Sandberg, H., and Johansson, K. H. Attack Models and Scenarios for Networked Control Systems. In *Proceedings of the 1st international conference on High Confidence Networked Systems* (2012), ACM, pp. 55–64.

[21] Tilaro, F. Assessment and Testing of Industrial Devices Robustness against Cyber Security Attacks. In *Conf. Proc.* (2011).

[22] Wedgbury, A., and Jones, K. Automated Asset Discovery in Industrial Control Systems - Exploring the Problem. In *Proceedings of the 3rd International Symposium for ICS & SCADA Cyber Security Research* (2015), British Computer Society, pp. 73–83.

[23] Xie, F., Peng, Y., Zhao, W., Gao, Y., and Han, X. Evaluating Industrial Control Devices Security: Standards, Technologies and Challenges. In *IFIP International Conference on Computer Information Systems and Industrial Management* (2014), Springer, pp. 624–635.

Notes

[1] https://www.ge.com/digital/products/achilles-vulnerability-testing-platform

[2] http://www.isasecure.org/en-US/Certification/IEC-62443-EDSA-Certification

[3] https://www.spirent.com/

[4] https://github.com/abhishek-kakkar/BeagleLogic

[5] https://www.picotech.com/oscilloscope

[6] https://nmap.org/

[7] https://code.google.com/p/plcscan/

[8] https://riskviz.de

Availability

We plan to publish the scripts used for the penetration tests and for the evaluation under GPL licence. Furthermore, the collected data and the results derived from the testbed are available on request for further research.

A Overview

The Appendix lists additional measurement results for every PLC in the testbed. The attack with the most influence for each controller is marked with a grey background.

Table 3: Cycle time in μs during attacks against Wago devices

Device	Attack	Mean Pre	Mean Att	Mean Post	Median Pre	Median Att	Median Post	Min Pre	Min Att	Min Post	Max Pre	Max Att	Max Post
750-889 (1)	zgrab modbus	2000	2213	2000	1816	2356	2164	54	237	1467	2579	52024	2538
	zgrab http	2000	2000	2000	2167	1826	2227	1491	274	1447	2502	2539	2559
	syn	2000	2000	2000	1841	2010	2225	505	1579	1316	2562	2413	2697
	snmp	2000	2000	2000	2142	1800	2140	1393	973	1413	2620	2633	2594
	http	2000	2000	2000	2054	2168	2165	1430	1442	1483	2574	2568	2531
	hping udp flood	2000	140044	2000	2226	1633	2202	1406	414	1471	2603	8397008	2519
	hping S P U flood	2000	18282	2000	1806	2346	2166	279	1533	1447	2533	8067682	2628
	hping c1 1 C17 flood	2000	2000	2000	1784	1760	1772	425	775	463	2713	2620	2641
	arp 4	2000	199062	2000	2228	1789	1828	1445	604	243	2557	5526857	2559
	arp 3	2000	182637	2000	1758	1636	2235	684	597	1332	2530	5085928	2687
	arp 2	2000	306792	2000	1789	1639	1800	1408	584	386	2598	6540940	2511
	arp 1	2000	375307	2000	1831	1640	2167	1420	566	1487	2601	11540100	3365
750-8100 (2)	zgrab modbus	10101	19148	10093	10334	10398	10334	7160	8303	9553	30389	779529	30408
	zgrab http	10071	10073	10084	10327	10332	10334	9575	2486	7683	30407	30414	30428
	syn	10091	13545	10097	10330	10355	10334	7261	791	717	39742	90279	30434
	snmp	10063	10064	10057	10324	10330	10333	5876	9569	9569	30440	30432	30385
	http	10064	10057	10088	10334	10326	10329	9559	9575	8140	30403	30428	30407
	hping udp flood	10054	14724	10074	10334	10353	10334	9579	1431	9549	30432	89676	39737
	hping S P U flood	10066	14163	10066	10329	10348	10334	4387	8005	1819	30415	109617	30399
	hping c1 1 C17 flood	10091	10053	10078	10337	10333	10331	7087	8057	9569	30429	30394	30452
	arp 4	10113	11131	10085	10333	10348	10332	9451	9562	9571	30452	39632	40511
	arp 3	10054	11493	10054	10323	10350	10332	9580	4960	9521	30390	40462	30401
	arp 2	10081	11472	10060	10336	10341	10331	9549	9543	9562	49663	40400	30401
	arp 1	10081	11332	10074	10338	10340	10335	9557	7883	9563	30395	49598	30389
750-880 (3)	zgrab modbus	1999	2214	1999	1702	2337	1780	581	248	581	3388	61882	3556
	zgrab http	1999	1998	2000	1717	1701	2148	580	583	580	3384	3445	3472
	syn	1999	2019	1997	1712	2308	1733	581	1624	309	3426	2639	3392
	snmp	1997	1996	1996	1727	1723	1721	581	308	23	3475	3385	3453
	http	1999	1999	2000	1709	1716	2284	581	377	582	3430	3385	3451
	hping udp flood	1997	178903	1997	1731	1630	1727	580	332	580	3444	6283455	3491
	hping S P U flood	1998	87062	1996	1732	1634	1722	153	601	339	3384	42502647	3469
	hping c1 1 C17 flood	1998	1997	1996	1727	1728	1729	578	580	581	3410	3381	3484
	arp 4	1999	160511	1995	1774	1630	1724	577	575	580	3440	5400406	3392
	arp 3	1997	162632	1998	1717	1712	1740	581	557	580	3445	3761630	3446
	arp 2	1999	465689	1998	1734	1629	1734	574	612	580	3386	10973587	3487
	arp 1	1998	575663	1997	1732	2372	1725	581	617	580	3390	9987443	3392
750-831 (4)	zgrab modbus	2000	2233	2000	1890	2334	1903	320	337	1230	3448	96089	3226
	zgrab http	2000	2000	2000	2003	2192	2162	1439	1371	1385	3308	3367	3440
	syn	2000	1999	2000	1809	2006	2010	1326	1568	1447	2649	2448	2567
	snmp	2000	2000	2000	2232	1784	2233	547	578	1367	3342	3382	2646
	http	2000	2000	2000	1918	2002	1862	1407	1418	1336	3316	3208	3157
	hping udp flood	2000	75698	2000	1819	2336	2038	920	338	1457	2600	3869519	3043
	hping S P U flood	2000	18353	2000	1798	2326	2204	236	1521	1396	3424	7968669	3246
	hping c1 1 C17 flood	2000	2000	2000	2044	2180	2248	1085	1411	562	2678	3151	3399
	arp 4	2000	151997	2000	2200	1676	2020	1374	315	1074	2641	3321851	2650
	arp 3	2000	244520	2000	2002	2344	1910	1390	568	1484	2633	4065957	2531
	arp 2	2000	653732	2000	1777	2368	2133	1440	605	73	2606	7627166	2684
	arp 1	2000	467949	2000	1762	2366	2146	61	586	982	2646	6923920	2672

Table 4: Cycle time in μs during attacks against Siemens devices

Device	Attack	Mean Pre	Mean Att	Mean Post	Median Pre	Median Att	Median Post	Min Pre	Min Att	Min Post	Max Pre	Max Att	Max Post
S7-1211 (5)	zgrab s7	223	233	223	192	141	192	110	108	110	1336	1572	1360
	zgrab https	223	223	223	192	192	192	109	109	54	1377	1320	1365
	zgrab http	223	223	223	192	192	192	92	37	108	1304	1324	1344
	szl	223	223	223	192	192	192	109	108	108	1353	1395	1354
	syn	223	325	223	192	161	191	91	108	109	1413	1638	1389
	snmp	223	223	223	192	192	192	110	110	109	1237	1297	1283
	http	223	223	223	192	192	192	89	111	26	1371	1252	1346
	hping udp flood	223	310	223	191	167	191	16	108	109	1372	1945	1363
	hping S P U flood	223	343	223	191	166	191	109	109	107	1312	1736	1337
	hping c1 1 C17 flood	223	223	223	191	191	191	104	109	106	1414	1348	1414
	arp 4	223	306	223	192	168	191	109	108	12	1340	1756	1207
	arp 3	223	301	223	192	156	191	85	62	108	1299	1645	1361
	arp 2	223	317	223	192	160	191	110	108	20	1003	1612	1371
	arp 1	223	317	223	192	160	191	109	108	57	1339	1587	1354
S7-1212 (6)	zgrab s7	30497	38125	30500	30968	39129	29073	20327	4211	28939	32061	44032	32063
	zgrab https	30500	30501	30500	29075	30501	29068	28940	28810	28937	32062	32069	32068
	zgrab http	30501	30500	30489	30502	29073	30501	28938	28938	5919	32064	32063	32068
	szl	30500	30494	30500	29072	30498	29077	28939	15975	28941	32066	32065	32063
	syn	30486	30760	30501	29075	31974	30500	276	28938	28450	32062	41043	32061
	snmp	30494	30489	30494	30502	30502	30499	15425	4873	15925	32064	32064	32068
	http	30489	30500	30490	30501	29076	30504	5396	28939	6947	32063	32061	32067
	hping udp flood	30487	30669	30496	29078	31967	30502	2286	28780	18996	33002	38999	32063
	hping S P U flood	30490	30676	30500	30498	31974	29074	6812	28937	28938	32065	39011	32064
	hping c1 1 C17 flood	30501	30500	30496	30498	29074	30501	28939	28936	19497	32064	32066	32063
	arp 4	30490	30671	30500	30501	31968	29074	7606	28940	28939	32057	38026	32062
	arp 3	30501	30671	30500	30500	31970	29075	28243	28499	28937	32065	38434	32063
	arp 2	30501	30699	30500	30504	31973	29070	28936	28418	28943	32063	39578	32062
	arp 1	30486	30720	30500	29075	31971	29074	895	28933	28938	32067	40011	32064
ET200-SP (7)	zgrab s7	4575	4530	4422	3947	3947	3944	1943	949	1054	33946	28054	36054
	zgrab https	4610	4571	4512	3947	3947	3947	1943	1943	1943	36054	35949	31943
	zgrab http	4468	4523	4608	3947	3947	3947	1054	945	1943	41949	30052	36054
	szl	4517	4634	4470	3947	3947	3947	1054	1943	1943	29948	31947	40056
	syn	4647	4395	4583	3947	3944	3946	1055	944	1053	36058	34059	44054
	snmp	4440	4383	4310	3947	3944	3943	1198	1943	1054	32055	38056	32055
	http	4477	4437	4572	3944	3944	3947	1055	1943	548	27945	34055	56056
	hping udp flood	4425	4551	4645	3944	3947	3947	948	945	1943	31945	35948	32055
	hping S P U flood	4506	4574	4505	3947	3947	3947	1943	945	945	26054	33945	28054
	hping c1 1 C17 flood	4475	4275	4620	3944	3943	3947	1943	945	794	37947	33944	33948
	arp 4	4451	3869	4587	3947	2058	3947	405	945	944	32056	26055	30057
	arp 3	4576	3921	4459	3947	2058	3947	1943	944	1943	34054	29947	36054
	arp 2	4541	4492	4505	3946	3947	3947	945	1054	1943	30056	33948	29948
	arp 1	4450	4400	4408	3947	3947	3947	1055	105	1943	30056	30052	37946
S7-314 (8)	zgrab s7	206	271	206	223	253	223	117	114	116	619	880	619
	zgrab https	206	206	206	223	223	223	117	117	117	674	660	619
	zgrab http	206	206	206	223	223	223	117	117	117	674	619	619
	szl	206	258	206	223	252	223	75	114	117	633	880	674
	syn	206	266	206	223	253	223	116	114	116	619	853	661
	snmp	206	206	206	223	223	223	117	117	117	619	646	619
	http	206	206	206	223	223	223	116	117	117	660	661	620
	hping udp flood	206	265	206	223	252	223	117	3	117	619	922	619
	hping S P U flood	206	267	206	223	253	223	102	112	117	660	853	633
	hping c1 1 C17 flood	206	206	206	223	223	223	117	117	117	619	620	619
	arp 4	206	264	206	223	252	223	16	115	117	674	840	619
	arp 3	206	265	206	223	252	223	117	114	117	619	840	661
	arp 2	206	267	206	223	253	223	117	114	117	674	881	619
	arp 1	206	267	206	223	253	223	75	115	117	619	840	633
S7-1516F (9)	zgrab s7	108	130	108	131	135	131	18	18	15	576	509	584
	zgrab https	108	108	108	131	131	131	18	18	18	616	588	588
	zgrab http	108	108	108	131	131	131	18	16	18	580	592	601
	szl	108	108	108	131	131	131	18	17	18	589	592	580
	syn	108	129	108	131	135	131	19	18	18	581	584	608
	snmp	108	108	108	131	131	131	18	19	18	621	580	561
	http	108	108	108	131	131	131	18	22	18	565	592	576
	hping udp flood	108	128	108	131	135	131	18	18	15	588	593	588
	hping S P U flood	108	128	108	131	135	131	16	18	18	600	573	619
	hping c1 1 C17 flood	108	108	108	131	131	131	22	18	18	565	564	617
	arp 4	108	129	108	131	135	131	18	18	19	577	585	581
	arp 3	108	129	108	131	135	131	18	18	22	592	597	635
	arp 2	108	129	108	131	135	131	17	19	18	569	581	576
	arp 1	108	129	108	131	135	131	22	22	18	604	608	621
Siemens Logo! 8 (10)	zgrab modbus	171	354	171	260	282	260	32	23	30	1420	2456	1443
	zgrab http	171	171	171	259	259	260	32	32	32	1459	1445	1436
	syn	171	420	171	260	285	260	32	20	32	1474	2170	1430
	snmp	171	171	171	260	260	260	18	32	32	1494	1446	1477
	http	171	171	171	260	260	260	22	32	32	1437	1454	1435
	hping udp flood	171	373	171	260	282	259	32	21	18	1440	2141	1481
	hping S P U flood	171	365	171	259	281	260	32	22	19	1429	2181	1502
	hping c1 1 C17 flood	171	171	171	260	260	260	18	32	18	1445	1429	1453
	arp 4	171	289	171	259	289	260	29	32	32	1479	1716	1460
	arp 3	171	290	171	259	289	259	32	19	32	1433	1714	1453
	arp 2	171	756	171	259	277	259	32	20	32	1454	96776	1435
	arp 1	171	791	171	260	276	260	19	17	32	1434	96160	1453

Table 5: Cycle time in μs during attacks against Phoenix Contact devices

Device	Attack	Mean Pre	Mean Att	Mean Post	Median Pre	Median Att	Median Post	Min Pre	Min Att	Min Post	Max Pre	Max Att	Max Post
ILC 151 ETH (11)	zgrab modbus	1000	1001	1000	994	1044	945	636	10	498	1363	4227	1338
	zgrab http	1000	1000	1000	992	1097	1070	545	624	644	1329	1417	1351
	syn	1000	1003	1000	998	1029	1080	642	14	629	1330	4253	1385
	snmp	1000	1000	1000	916	1127	910	125	568	160	1404	1465	1337
	http	1000	1000	1000	899	1018	971	646	627	626	1368	1340	1336
	hping udp flood	1000	1001	1000	882	1012	980	470	31	329	1360	4228	1397
	hping S P U flood	1000	1002	1000	1142	1031	934	632	55	550	1423	3186	1385
	hping c1 1 C17 flood	1000	1000	1000	1068	926	1142	614	360	595	1387	1365	1409
	arp 4	1000	1002	1000	964	1072	854	84	13	519	1374	4231	1426
	arp 3	1000	1003	1000	1008	1072	946	702	27	630	1285	4226	1348
	arp 2	1000	1003	1000	1011	1063	925	133	18	84	1383	4260	1398
	arp 1	1000	1003	1000	873	1056	1056	253	7	568	1400	4238	1450
ILC 150 ETH (12)	zgrab modbus	1086	1086	1086	1111	1110	1111	953	269	952	4244	6313	4089
	zgrab https	1085	1085	1085	1111	1111	1111	434	6	341	4244	4240	4241
	zgrab http	1085	1085	1085	1111	1111	1111	929	953	954	4232	4244	4244
	szl	1084	1084	1084	1110	1111	1111	534	954	124	4242	4242	4240
	syn	1085	1088	1084	1111	1111	1111	953	952	287	4239	4260	4244
	snmp	1083	1084	1085	1111	1111	1111	917	954	952	4244	4239	4244
	http	1085	1085	1085	1111	1111	1111	954	520	93	4244	4244	4244
	hping udp flood	1086	1087	1088	1111	1111	1111	437	952	256	4246	4259	4244
	hping S P U flood	1086	1088	1085	1111	1111	1111	787	952	297	4244	4338	4246
	hping c1 1 C17 flood	1086	1086	1086	1111	1111	1111	809	953	232	4240	4240	4246
	arp 4	1086	1091	1086	1111	1111	1111	267	469	953	4241	4247	4245
	arp 3	1085	1090	1086	1111	1111	1111	952	455	953	4246	6160	4242
	arp 2	1084	1091	1085	1111	1111	1111	953	927	475	4242	6295	4243
	arp 1	1084	1090	1084	1111	1111	1111	841	953	540	4238	6156	4246
ILC 171 ETH (13)	zgrab modbus	1000	1000	1000	982	994	993	730	56	717	1275	4202	1279
	zgrab https	1000	1000	1000	949	988	992	719	718	727	1274	1275	1271
	zgrab http	1000	1000	1000	807	802	988	718	100	718	1274	1274	1278
	szl	1000	1000	1000	799	1190	799	625	701	157	1271	1297	1272
	syn	1000	1003	1000	994	1123	1001	701	54	774	1297	4219	1228
	snmp	1000	1000	1000	849	867	850	299	182	466	1232	1228	1227
	http	1000	1000	1000	799	994	799	261	727	272	1330	1271	1271
	hping udp flood	1000	1002	1000	1002	1070	964	755	40	722	1222	4221	1279
	hping S P U flood	1000	1001	1000	852	1013	856	254	22	88	1232	4239	1222
	hping c1 1 C17 flood	1000	1000	1000	856	1024	889	119	698	755	1222	1222	1222
	arp 4	1000	1003	1000	851	1085	852	540	22	147	1224	4217	1245
	arp 3	1000	1003	1000	806	1078	1004	261	9	769	1518	4235	1226
	arp 2	1000	1003	1000	855	1082	1191	206	25	715	1228	4244	1276
	arp 1	1000	1003	1000	850	1075	855	468	8	592	1228	4263	1229

Table 6: Cycle time in μs during attacks against ABB devices

Device	Attack	Mean Pre	Mean Att	Mean Post	Median Pre	Median Att	Median Post	Min Pre	Min Att	Min Post	Max Pre	Max Att	Max Post
PM554-TP-ETH (14)	zgrab modbus	1000	1143	1000	1074	1079	1076	908	97	903	1916	3100	1097
	zgrab http	1000	1000	1000	926	1072	926	510	903	626	1099	1097	1097
	syn	1000	1107	1000	1000	1079	1074	903	895	232	1095	5089	1911
	snmp	1000	1000	1000	1074	1074	1076	523	902	904	1916	1095	1097
	http	1000	1000	1000	1073	1076	925	905	904	732	1097	1097	1095
	hping udp flood	1000	1070	1000	1074	1079	1074	905	662	903	1096	3919	1910
	hping S P U flood	1000	1073	1000	925	1078	926	747	231	542	2087	3099	1098
	hping c1 1 C17 flood	1000	1000	1000	1073	1074	924	906	903	839	1094	1915	1096
	arp 4	1000	1000	1000	1074	1002	1073	902	901	902	1097	1101	1095
	arp 3	1000	1000	1000	926	1000	926	114	901	238	1096	1099	1097
	arp 2	1000	1010	1000	927	1074	926	72	897	240	2079	3909	1096
	arp 1	1000	1014	1000	1000	1075	1072	903	322	903	2086	3091	1096

Table 7: Cycle time in μs during attacks against Crouzet devices

Device	Attack	Mean Pre	Mean Att	Mean Post	Median Pre	Median Att	Median Post	Min Pre	Min Att	Min Post	Max Pre	Max Att	Max Post
em4 (15)	zgrab modbus	1999	1999	1999	1999	1999	1999	435	1841	1992	2006	2007	2006
	zgrab http	1999	1999	1999	1999	1999	1999	920	1992	1992	2006	2006	2006
	syn	1999	1999	1999	1999	1999	1999	565	1992	1991	2006	2006	2006
	snmp	1999	1999	1999	1999	1999	1999	615	793	1992	2006	2006	2006
	http	1999	1999	1999	1999	1999	1999	1634	861	1992	2006	2006	2006
	hping udp flood	1999	1999	1999	1999	1999	1999	1992	1992	170	2006	2006	2007
	hping S P U flood	1999	1999	1999	1999	1999	1999	768	984	1593	2010	2006	2006
	hping c1 1 C17 flood	1999	1999	1999	1999	1999	1999	1992	1992	1381	2006	2005	2006
	arp 4	1999	1999	1999	1999	1999	1999	1571	557	1992	2006	2005	2006
	arp 3	1999	1999	1999	1999	1999	1999	1992	1988	1992	2005	2010	2006
	arp 2	1999	1999	1999	1999	1999	1999	1992	1992	1991	2006	2006	2006
	arp 1	1999	1999	1999	1999	1999	1999	1069	1992	1992	2006	2006	2006

Table 8: Cycle time in μs during attacks against Schneider devices

Device	Attack	Mean Pre	Mean Att	Mean Post	Median Pre	Median Att	Median Post	Min Pre	Min Att	Min Post	Max Pre	Max Att	Max Post
Modicon M2 (16)	zgrab modbus	2000	2152	2000	2000	2001	2003	134	1944	188	2034	54041	2045
	syn	2000	2852	2000	1998	2002	2000	644	1943	742	2045	52012	3008
	snmp	2000	2000	2000	2000	2000	2000	1464	1703	1953	2051	2056	2048
	hping udp flood	2000	10774	2000	2000	8003	2000	1208	1782	1948	2035	67019	2054
	hping S P U flood	2000	7773	2000	2000	4002	2000	1266	48	686	2035	76015	2050
	hping c1 1 C17 flood	2000	2000	2000	2000	2000	2000	1964	1317	1951	2051	2051	2051
	arp 4	2000	9004	2000	2000	8000	2000	1943	1621	1967	2057	102011	2033
	arp 3	2000	9050	2000	2000	8002	2000	1953	1946	1134	2047	97013	2048
	arp 2	2000	2628	2000	2001	2002	2000	1968	1935	1798	2033	36006	2033
	arp 1	2000	2494	2000	2000	2002	2001	1953	1275	1967	2048	35005	2034

B Wago at different rates

Table 9: Cycle time in μs during attacks against Wago devices at 64 KBit/s

Device	Attack	Mean Pre	Mean Att	Mean Post	Median Pre	Median Att	Median Post	Min Pre	Min Att	Min Post	Max Pre	Max Att	Max Post
750-889 (1)	arp 4	2000	2001	2000	2132	2199	1755	1207	598	890	2783	3702	2665
	arp 3	2000	2001	2000	2149	2198	2178	1284	376	1292	2716	3691	2709
	arp 2	2000	2004	2000	2099	2152	1816	1342	598	1225	2684	4403	2782
	arp 1	2000	2004	2000	1786	2154	1754	481	244	1154	2775	4516	2629
750-8100 (2)	arp 4	9999	9999	9997	10320	9697	9715	9573	9565	214	10478	10456	10462
	arp 3	9999	9999	9998	10323	10008	9736	9568	9555	5315	10470	10462	10457
	arp 2	9999	9999	9998	10288	9699	9746	9570	9502	2781	10495	10448	10491
	arp 1	9997	10005	9999	9737	10016	10312	2408	6947	9566	10488	30394	10474
750-880 (3)	arp 4	2000	2009	1998	2276	2302	1715	580	352	580	3382	4387	3387
	arp 3	2000	2007	1998	1840	2298	1710	581	357	579	3417	4386	3419
	arp 2	1999	2010	1998	1757	2302	1704	580	518	581	3402	5377	3430
	arp 1	2000	2011	1998	1795	2300	1708	578	288	580	3384	4699	3391
750-831 (4)	arp 4	2000	2003	2000	2088	2163	2008	1377	322	1358	3042	4398	3368
	arp 3	2000	2003	2000	2226	2162	1981	1375	312	1411	3373	4385	3347
	arp 2	2000	2004	2000	2220	2112	2240	519	306	1353	3514	4433	3343
	arp 1	2000	2004	2000	2156	2112	2238	1383	58	638	2618	4413	3343

Table 10: Cycle time in μs during attacks against Wago devices at 1 MBit/s

Device	Attack	Mean Pre	Mean Att	Mean Post	Median Pre	Median Att	Median Post	Min Pre	Min Att	Min Post	Max Pre	Max Att	Max Post
750-889 (1)	arp 4	2000	2002	2000	2034	2224	2140	1403	588	1481	2983	3686	2516
	arp 3	2000	2001	2000	1859	2227	2001	1321	543	1481	2678	3661	2531
	arp 2	2000	2004	2000	2002	2179	1810	1302	174	896	3293	4617	2518
	arp 1	2000	2004	2000	2026	2199	1826	1384	588	1115	2608	4585	2669
750-8100 (2)	arp 4	9998	9999	9998	9823	10259	9808	7934	9510	6628	10539	10514	10538
	arp 3	9999	9999	9997	10270	9752	9798	9515	9445	2508	10557	10516	10565
	arp 2	10002	10116	9999	10266	10333	10258	9523	9522	9520	30375	30404	10583
	arp 1	10001	10135	9998	9806	10331	9756	370	9567	6016	30432	30418	10560
750-880 (3)	arp 4	1998	2037	1997	1701	2363	1695	544	145	576	3419	4515	3423
	arp 3	1998	2036	1997	1714	2363	1695	582	581	425	3442	4491	3437
	arp 2	1998	2038	1998	1704	2311	1695	580	579	290	3386	5568	3414
	arp 1	1998	2041	1999	1711	2313	1747	576	584	576	3388	5434	3411
750-831 (4)	arp 4	2000	2003	2000	2009	2205	1993	812	525	1130	3412	4410	3364
	arp 3	2000	2003	2000	2192	2215	1773	1400	430	748	2631	4447	3403
	arp 2	2000	2004	2000	1828	2170	2226	23	417	1425	3341	4719	2958
	arp 1	2000	2004	2000	2166	2140	2215	1162	454	589	3370	4641	3439

Table 11: Cycle time in μs during attacks against Wago devices at 8 MBit/s

Device	Attack	Mean Pre	Mean Att	Mean Post	Median Pre	Median Att	Median Post	Min Pre	Min Att	Min Post	Max Pre	Max Att	Max Post
750-889 (1)	arp 4	2000	2111	2000	1867	2362	1853	75	531	236	2612	8139	2533
	arp 3	2000	2111	2000	1916	2362	2066	1295	535	1489	2707	8923	2523
	arp 2	2000	2272	2000	2197	1690	2276	1360	343	1476	2608	107437	2538
	arp 1	2000	2278	2000	2195	1676	1756	1297	287	271	2716	92413	2590
750-8100 (2)	arp 4	9999	11163	9997	10305	10344	9764	9565	4078	85	10508	30475	10486
	arp 3	10002	11296	9999	10028	10344	10304	9546	9539	9562	30380	30493	10489
	arp 2	9999	11488	9999	10319	10351	9747	9550	2775	8522	10473	30476	10450
	arp 1	9999	11424	9998	10017	10347	9726	9534	5075	6532	10515	49692	10495
750-880 (3)	arp 4	1999	2116	2000	1709	2313	2299	576	410	580	3420	9521	3447
	arp 3	1997	2115	1998	1702	2343	1707	582	451	575	3380	9676	3414
	arp 2	1998	2267	1997	1702	1695	1730	580	226	579	3418	90386	3480
	arp 1	1998	2261	1996	1700	1693	1696	579	253	580	3385	142657	3449
750-831 (4)	arp 4	2000	2137	2000	2264	2334	2199	1436	329	1432	3346	9737	3403
	arp 3	2000	2139	2000	2092	2334	1994	1347	328	1348	2648	9769	3382
	arp 2	2000	2276	2000	2222	2107	2009	627	262	1441	3432	114800	3345
	arp 1	2000	2269	2000	2144	1719	1815	305	352	530	3398	100319	3335

Table 12: Cycle time in μs during attacks against Wago devices at 16 MBit/s

Device	Attack	Mean Pre	Mean Att	Mean Post	Median Pre	Median Att	Median Post	Min Pre	Min Att	Min Post	Max Pre	Max Att	Max Post
750-889 (1)	arp 4	2000	2826	2000	2238	2288	1782	1493	555	1152	2519	493534	2602
	arp 3	2000	2791	2000	2161	2330	2162	1451	346	1480	2548	358603	2533
	arp 2	2000	2821	2000	1819	2302	1802	869	560	756	2601	479991	2659
	arp 1	2000	2822	2000	2169	2322	2244	1349	606	1413	2652	497548	2601
750-8100 (2)	arp 4	9998	9999	9998	9823	10259	9808	7934	9510	6628	10539	10514	10538
	arp 3	9999	9999	9997	10270	9752	9798	9515	9445	2508	10557	10516	10565
	arp 2	10002	10116	9999	10266	10333	10258	9523	9522	9520	30375	30404	10583
	arp 1	10001	10135	9998	9806	10331	9756	370	9567	6016	30432	30418	10560
750-880 (3)	arp 4	1999	2775	1997	1735	1693	1698	61	453	580	3448	360654	3382
	arp 3	1997	2785	1996	1728	1695	1726	577	566	578	3386	453659	3488
	arp 2	1996	2806	1998	1728	1692	1725	579	561	580	3429	384598	3386
	arp 1	1998	2833	1998	1735	1694	1732	580	567	582	3424	437690	3383
750-831 (4)	arp 4	2000	2798	2000	2218	1740	2203	1343	541	1387	2681	389171	3093
	arp 3	2000	2811	2000	2216	1733	2170	1399	564	133	3369	424686	3385
	arp 2	2000	2818	2000	2195	1731	2200	442	569	275	2647	451619	3389
	arp 1	2000	2850	2000	2006	2264	2060	1359	394	723	3337	511724	3285