Design and Emulation of Physics-Centric Cyberattacks on an Electrical Power Transformer

JOHN OLIJNYK¹, BENJAMIN BOND², AND JULIAN RRUSHI³

¹Oakland University, Rochester, MI 48309 USA (e-mail: johnolijnyk@oakland.edu)
²Oakland University, Rochester, MI 48309 USA (e-mail: benjaminbond@oakland.edu)
³Oakland University, Rochester, MI 48309 USA (email: rrushi@oakland.edu)

ABSTRACT
Malware that attacks the electrical power grid consists of exploits and operations modules. The exploits are similar to those of traditional malware. These exploits hack into an industrial computer and subsequently deploy operational modules. Some operational modules penetrate the operating system of the compromised industrial computer to take over computing functions and hence facilitate further attacks. Examples include interception of cryptographic keys, and generation of deceptive status data that indicate normal operation, while in reality the power transformer is in distress due to the attacks. Other operational modules are designed to recognize and disrupt the physics of the physical equipment. We refer to these operations modules as physics-centric modules. The subject of this paper is research on how physics-centric modules of current and future malware can cause physical damage to power grid equipment. The physical equipment used in this research was a power transformer. We make several contributions in this paper, namely: i) we emulate in Python the protection algorithms that run on an industrial computer and monitor and protect a power transformer from a variety of faults; ii) we leverage these emulations to analyze the cyberattack surface of a power transformer; iii) with these insights at hand, we devise attack modus operandi that malware could use against a power transformer; and iv) we emulate these cyberattacks in Python to empirically observe and quantify their destructive effects on a power transformer. Our overall research findings in this paper serve the purpose of informing better defense against the physics-centric modules of malware that attack the electrical power grid.

INDEX TERMS electrical power grid, industrial control systems, cybersecurity, malware

I. INTRODUCTION
A cyber-physical system (CPS) commonly consists of industrial computers deeply integrated with physical equipment through I/O boards, which in turn reads measurements from sensors and applies commands to actuators. This is how industrial computers interact closely with physical equipment by sensing and changing their operational states. Computerized physical process control is usually performed with feedback loops, where physical processes affect computations and vice versa. In the physical world, the passage of time is constant and linear, and concurrency is inherent [1].

A CPS involves a high degree of complexity at numerous spatial and temporal scales. Furthermore, highly networked communications integrate computational and physical components [2]. In the electric power industry, computer-based systems have evolved to perform many complex tasks in energy control centers. Research efforts directed at the prospect of using digital computers to perform the tasks involved in power system protection date back to the mid-sixties, and were motivated by the emergence of industrial computers [3]. The availability of microprocessors used as a replacement for electro-mechanical and solid-state relays provides a number of advantages while meeting the basic protection philosophy requirement of decentralization [4]. The use of microprocessor-based protection methods provides benefits to the industry in terms of data for analysis, fine tuning of algorithms for performance, and cost improvements. These microprocessor-based systems have evolved into CPSs to enable improved monitoring and control capabilities over greater distances.
On the other hand, the use of microprocessor-based methods of physical process control also opens the door for threat actors to exploit vulnerabilities in industrial computers and gain unauthorized access to data or simply cause physical destruction. This is possible because an industrial computer is simply a computer equipped with I/O boards, special-purpose and real-time operating systems and software, algorithms, and network communication protocols.

Threat actors commonly implement their attacks on an electrical power grid as an automated attack code. Similar to malware that attacks general-purpose computing, these malware rely on exploits for the penetration of industrial computers. At times this occurs over the network. The exploits of malware that attack the electrical power grid are similar or identical to the exploits of malware that attack general-purpose computing. For example, exploitation of a memory vulnerability, supported by heap spraying to evade address space randomization, works similarly on both general-purpose computers and industrial computers.

Nevertheless, the two types of malware differ significantly in their operational modules. Malware which attack general-purpose computing contains modules specialized in keystroke interception, secondary storage interception, activation of the microphone and webcam for audio and video surveillance, network traffic interception, and interception of sensitive data such as passwords and cryptographic keys in physical main memory. The operational modules of malware that attack the electrical power grid are all related to physics; consequently, we refer to them as physics-centric modules.

Physics-centric modules locate and track physics data in memory, which they alter and/or suppress via their expert knowledge to cause physical destruction of power grid equipment. Physics-centric modules are mostly unexplored in the current open science literature. The methods of attack that those modules execute are inaccessible to researchers. This study aims to fill this void. We describe our research on the modus operandi of physics-centric modules that attack the electrical power grid.

Our work provides defenders with in-depth knowledge that is often required to devise a more effective defense against malware in the electrical power grid. We focus our work on specific power grid equipment, namely, a power transformer. Nevertheless, similar principles and findings are applicable to other types of power grid equipment.

**Contribution:** In this paper, we make several interconnected contributions. First, we designed and implemented a software system in Python that emulates the physics of a power transformer in operation, along with the physics-centric power transformer protection algorithms. These algorithms run on an industrial computer and monitor and protect a power transformer from a variety of faults. Second, we leverage these emulations to analyze the cyberattack surface of a power transformer, namely points of intervention on the physics of a power transformer where physics-centric modules of malware could act against the power transformer.

Third, and most importantly, guided by the findings on the attack surface of a power transformer, we devise a sample of attack modus operandi that a physics-centric module of malware could use against a power transformer. Finally, we emulate these cyberattack techniques in Python to empirically observe and quantify their destructive effects on a power transformer.

**Novelty:** This research is among the first works to provide a sample of physics-centric cyberattack methods on power grid equipment, with emphasis on attack principles that can be leveraged to inform better defense from present and future malware.

Our analysis of the cyberattack surface of a power transformer is based on the exploration of the inner workings of a power transformer and protection algorithms that work with its physics. While background on power transformers and protection algorithms is provided in numerous other works, textbooks, and manuals, we methodically identify and integrate pertinent elements of such a background, which results in the emergence of a cyberattack surface that can be analyzed.

The emulation software in this work includes attack modules that may be deployed without the added cost of damaged or destroyed physical equipment. This allows for an unlimited number of experiments, which can also be freely revised and extended to probe any factors of interest, cyber or physical, without any safety concerns.

Our study is related to several other important studies. For example, the Aurora generator test carried out by the Idaho National Laboratory (INL), a research lab of the United States Department of Energy (DoE), demonstrated how a cyberattack on an industrial computer could cause physical damage to a power generator. The target power generator was connected to a power grid using a circuit breaker controlled by an industrial computer. In this experiment, the attack code on the industrial computer operated the circuit breaker in a way that caused the power generator to connect while being out of synchronization with the power grid. Connection out of sync with the frequency of the power grid led to substantial damage and an eventual explosion, and hence practically proved the physical impact that a cyberattack could have on power grid equipment [5].

Our research aims to continue the line of work of Aurora, but does so through emulation to avoid destruction of actual power grid equipment, while maintaining high levels of fidelity for research purposes.

Real-world malware has physics-centric modules, and thus our research addresses a realistic problem. An example of a known malware is BlackEnergy [6]. BlackEnergy was originally a distributed denial of service (DDoS) malware, but was later deployed as a key part of a malware campaign that targeted power distribution substations in Eastern Europe in 2015. The malware, once installed on information technology workstations, allowed threat actors to gain privileged access to the industrial control system network.

From there, the malware gained direct control of supervisory control and data acquisition (SCADA) workstations and...
servers. Threat actors can then directly control the physical processes of the electrical substation and bring power distribution to a halt [7]. Our attack methods are different from BlackEnergy in that they seek to emulate how physics-centric modules could cause physical damage to power grid equipment, whereas BlackEnergy mostly disrupted power [8].

To the best of our knowledge, the Aurora Generator Test is the first to demonstrate physics-centric cyber attacks and their ability to cause physical damage. Physics-centric attacks on physical processes and equipment were also described in a later work by Garcia et al. in [9]. The authors developed a rootkit, dubbed HARVEY, which incorporates expert knowledge of physics, and hence knows how to manipulate physics such as to maximize the damage to power grid equipment [9].

Similar to the work of Garcia et al., our work explores physics-centric attack methods. The differences are i) the kind of physics leveraged by the two works and hence the type of faults caused by their attacks; and ii) how the two works interplay with legitimate algorithms on industrial computers, compromised or not.

To the best of our understanding of the work of Garcia et al., HARVEY simulates the power system to determine how to cause faults that are external to power grid equipment such as the power generator. The external faults caused by HARVEY aim to cause large-scale failure of the power grid. Because of their protection algorithms, industrial computers are able to sense these faults in a timely fashion and trip circuit breakers to protect physical equipment from physical destruction. Thus, failure appears to equate to the interruption of power.

In contrast to Harvey, our attack methods leverage the inner physics of power grid equipment to cause faults that are internal to those equipment. The consequences of such attack methods include physical damage and possible interruption of power.

HARVEY implements an attack variant of the control algorithm, such as the optimal power flow. HARVEY intercepts between the real control algorithm and the actuators of a programmable logic controller (PLC), discarding the commands from the real control algorithm and apply the commands coming from its malicious variant of the control algorithm. Garcia et al. did not explore attacks on protection algorithms, although we believe that HARVEY could be applicable to those algorithms as well by discarding their computation results. Our attack methods dive deeper into control algorithms, in addition to interposing between those algorithms and the I/O board.

Organization: The remainder of this paper is organized as follows. Section II provides the physics background of power transformers and electrical power grids. Section III provides a description of our cyber and physics-centric threat model. Section IV shows how the physics of the system expands the attack surface. Section V describes our attack methods emulated to mimic the physics-centric modules of malware. In Section VI we provide an overview of the attack emulation results. Section VII describes further lines of study that our research enables. Section VIII shares similar lines of research that supports our work. Finally, in Section IX we summarize our contributions and conclude the paper.

II. BACKGROUND

We now examine the use case of a power transformer, which is physical equipment commonly found in electrical substations of the electric power grid. Electric power grids and power transformers are governed by the laws of physics and principles of electrical and mechanical engineering. A deep understanding of their operations is required to understand potential vulnerabilities.

A. POWER TRANSFORMER

Electric energy is produced at electric power generating stations and transported over high-voltage transmission lines to the utilization points. The trend toward using higher voltages is motivated by the increased line capacity while reducing the line losses per unit of power transmitted. The reduction in losses is significant and is an important aspect of energy conservation. A power transformer is a key technology that allows for the efficient transport of power from its generation to the customer [4].

A power transformer is a device that is designed to input one current or voltage and output a transformed value. This transformed value is a current or voltage that is higher or lower than the input, and a transformer is identified as being either "step-up" or "step-down", respectively. This change in value is influenced by the ratio of the number of turns in two wire lengths that are wrapped around the core of a conductive material. The input wire is referred to as the primary side, and the output is the secondary side. These two coils are insulated from one another, so no current can be directly exchanged [10].

Because of Faraday’s law of induction, when an alternating current is passed through any one wire, electrical energy in the other is induced by the magnetic flux that is created in the shared core. This flux creates an electromotive force in the secondary wire that produces current \( I_s \), the magnitude of which is determined by the initial current \( I_p \) in the primary wire and the ratio of windings between the primary \( N_p \) and secondary \( N_s \) sides. This is given as \( I_p, N_p = I_s, N_s \). This phenomenon is known as electromagnetic induction and allows for the transfer of electrical power between two circuits that are not physically connected to one another and the ability to transform the power either up or down as needed for the situation [10].

Traditionally, power transformers have been installed as purely passive devices. They required no external control and operated solely using electromechanical or solid-state devices. With no microcontroller and no connection to any network, a power transformer was beyond the reach of cyber attackers. The rise in the popularity of CPSs, which allow real-time management of system states and processes via internet-connected sensors and controllers, has opened the
door for threat actors to remotely attack devices. In particular, power transformers are now commonly equipped with microcontrollers that monitor various processes of the transformer and report back to the central control stations via network connections.

B. TAP CHANGER

A tap changer is a transformer subsystem that is used to regulate and change the output voltage of a transformer. This is needed because of losses when the transformer is integrated into a less than perfect real-world physical system. This enables a transformer to dynamically change its windings, thereby changing its output voltage and current to respond to changes in electrical loads. This is done by altering the number of turns in one winding, thereby changing the turns ratio of the transformer. Tap changers are mechanical devices driven by actuators that control tap engagement or disengagement. Tap changers have evolved from purely mechanical no-load changers to fully computer-controlled under load tap changers. Under load tap changers actuate while the transformer is energized while no load requires the transformer to be de-energized [11].

C. POWER FAILURE MODES

Short circuits or faults can occur in electric power and distribution systems. When a fault occurs on the load side of a transformer, the fault current passes through the transformer. Fault currents flowing through transformers are significantly higher than the rated currents of the transformers. These currents produce both mechanical and thermal stress in transformers [12].

A short-circuit fault occurs when two or more conductors come in contact with each other when they operate normally with a potential difference between them. The contact may be a physically metallic one, or it may occur through an arc. In the metal-to-metal contact case, the voltage between the two parts is reduced to zero. However, the voltage through an arc will have a very small value [4].

A failure in a transformer is caused by insulation breakdown between turns in the same slot or between the winding and the steel structure of the machine. The breakdown is due to insulation deterioration combined with switching or lightning over voltages. The contamination of insulators and lightning over voltages generally results in short-circuit faults. This is mainly attributed to aging combined with overloading [4].

Historically, windings and tap changers have been among the top causes of transformer failures [13]. Faults in tap changers can be caused by dielectric failures due to oil quality, clearance-related thermal failures due to coking or crimp problems, mechanical failures due to contact wear and misalignment, limit switches, sheared pins on the linkage that operates the reversing switch, and lubrication problems [11].

Power transformer faults have several potential consequences. Abnormally large currents flow in parts of the system, causing associated overheating of components. The amount of current may be much greater than the designed thermal ability of the conductors in the power lines or machines feeding the fault. As a result, the temperature rise may cause damage by the annealing of conductors and insulation charring. In addition, the low voltage in the neighborhood of the fault causes equipment malfunction. System voltages will be off their normal acceptable levels, resulting in possible equipment damage. Parts of the system will be caused to operate as unbalanced three-phase systems, meaning that some or all of the equipment will operate improperly. This is a small sampling of the errors caused by a fault in a power transformer [4].

The cost of a power transformer varies considerably and is based on several factors. This includes megavolt ampere (MVA) ratings, core designs, and other features [14]. A small power transformer rated at 10 MVA or lower costs approximately $600,000, while a large specialty transformer may cost $4,000,000 or more. Thus, the failure or damage of a power transformer represents a substantial loss to a utility supplier, and it may take months to replace a damaged unit.

D. SYSTEM PROTECTION

Operators must take necessary action to prevent faults from occurring in the transformer to minimize possible damage or power disruption. A protection system continuously monitors the power system to ensure maximum continuity of the electrical supply with minimum damage. A protective system detects fault conditions by continuously monitoring variables such as current, voltage, power, frequency, and impedance. Currents and voltages are measured by instrument transformers of the potential type (PT) or current type (CT).

Instrument transformers feed the measured variables to the relay system, which in turn, upon detecting a fault, command a circuit-interrupting device known as the circuit breaker (CB) or relay to disconnect the faulted section of the system. An electric power system is divided into protective zones for each apparatus in the system. The division is such that zones are provided adequate protection while maintaining service interruption to a minimum [4].

A relay is a device that opens and closes electrical contacts to control the operation of other devices under electric control. The protection system detects intolerable or undesirable conditions within an assigned area. The protection system then actuates the relay to operate the appropriate circuit breakers to disconnect the affected area to prevent damage to personnel and property [4].

Relays are classified according to their function as measuring or on-off relays. The latter class is also known as all-or-nothing and includes relays such as time-lag relays, auxiliary relays, and tripping relays. Here the relay does not have a specified setting and is energized by a quantity that is either higher than that at which it operates or lower than that at which it resets. Relays are made up of one or more fault-detecting units along with the necessary auxiliary units. The basic units for relay systems can be classified as...
follows: electromechanical units, sequence networks, solid-state units, or computer-based units [4].

In this study, we focus on digital computer-based units. These units, by virtue of being computer-based, are susceptible to cyber attacks, unlike electromechanical and solid-state units. The main elements of a digital computer-based relay include [4] the analog input subsystem, digital input subsystem, digital output subsystem, relay logic and settings, and digital filters.

III. THREAT MODEL

The CPS that we use for our threat modeling is a digital microcontroller-based power transformer that is integrated with current sensors and output relays. This digital microcontroller runs algorithms that monitor the input sensors and runs protection algorithms to protect the physical power transformer by controlling the relays.

We look at the various ways in which threat actors can compromise such a CPS as a power transformer. A part of a threat model is understanding the motivation that threat actors are taking. The common motivation that we see with threat actors attacking personal computers and information technology systems is to gain access to sensitive information such as passwords or encryption keys for some kind of financial gain or trade secret. However most CPSs and operational technology do not store personal information, encryption keys or anything related to financial data or trade secrets of a company. Instead they control the physical ecosystems and equipment. This implies that threat actors attacking such systems are doing so primarily for disruptive purposes.

We make several assumptions for our threat actors. To carry out a successful attack on the power grid, an attacker must locate a CPS that is connected to a power transformer. Threat actors can run code on the compromised CPS. The threat actor can inspect and alter the input/output buffers, memory locations and registers. This includes the ability to alter intermediate calculations.

We also assume that a threat actor can compromise the system to run arbitrary code and malware. We assume that this malware can execute code within the operating system address space, including the input/output, communications, control system, and system protection address spaces. Therefore, the threat actors code can access and modify any buffers of data that are used by the system algorithms.

These input buffers and intermediate calculations are based on the physics laws governing the controlled physical system. In this use case of an electrical power transformer, the inputs include the principle, secondary, differential, and restraining currents, harmonics, and any intermediate data of protection system algorithms. Access to this physics data can be performed via injected code or by replacement of the original code with malware code.

Threat actors in this case are physics-aware and will potentially leverage this knowledge to discover physical system architectures, configurations, or processes. However, physics-aware threat actors can move from reconnaissance to physical attacks. Through their influence and control of the system, they can force the system to cause physical damage by violating the safety of the physics of the equipment.

They can also prevent a safety system from executing proper safety procedures. This changes the safety scenario from a graceful shutdown to a life-threatening physical disaster. Furthermore, threat actors can deceive monitoring systems and operators into believing that operations are still normal by transmitting data that is false but within normal system parameters while simultaneously blocking factual data from transmission. The monitoring system’s inability to detect unsafe system operations prevents operators from taking action to prevent disaster.

The threat model in this work is empirically seen in real-world attacks. There have been several examples of attacks that have compromised CPSs. In the Triton malware, threat actors were able to compromise the PLC-based safety system controller of a nuclear power plant, and were able to install a remote access trojan. The remote access trojan could then modify the safety system to operate improperly, leading to physical damage or a disaster [15].

Another example is the CRASHOVERIDE / INDUSTROYER [16], where the malware disrupts the industrial CPSs that control the electrical substations of the electric grid. This lead to large-scale electrical outages in the Ukraine. Another example is STUXNET, which targets SCADA systems and PLCs that control physical industrial processes. STUXNET is believed to have caused significant damage to the Iran nuclear program centrifuges [17].

In conclusion, the threats of attacks on power grid equipment are real and can be executed by physics-aware threat actors with physical impacts and concrete damage, as demonstrated by several examples over the past decade.

IV. ATTACK SURFACE

The construction of an attack surface of a CPS requires a deep understanding of the physics of the system, namely, the power transformer and its ecosystem. The following is an in-depth analysis of the operations of a power transformer and the associated protection algorithms. Namely, we focus on the algorithms for differential protection and harmonic restraint. We will also look at other ways threat actors can manipulate the physics of the system to cause mechanical damage.

These are of particular interest because they provide for safe operation of the system to avoid physical damage to the equipment and safety to the users of the equipment. Threat actors will seek to understand such operations in order to exert malicious operation on the equipment to either cause damage directly or to inhibit the protection algorithm safety features so that damage occurs that could have been prevented by the safety feature.

A. DIFFERENTIAL PROTECTION

Differential protection is the main protection algorithm used for power transformers. A differential relay operates accord-
ing to the scalar or vector difference between two quantities, such as the current and voltage. Here, the currents on each side of the protected apparatus for each phase are compared in a differential circuit. Any difference in the currents operates a relay. During normal operation, only the difference between the current transformer magnetizing currents passes through the relay.

This is due to the fact that with no faults within the protected transformer, the currents entering and leaving are equal. If a fault occurs between the two sets of current transformers, one on the left-hand side will suddenly increase, while that on the right-hand side may decrease or increase with a direction reversal [4].

Differential protection algorithms for power transformers are based on a comparison (differential) of the primary and secondary winding currents of the two constituent coils of the transformer. Any electrical power that enters the transformer must exit it. If there is an imbalance between the input and output of the system, an internal fault likely occurs, and a relay trips a signal to the circuit breakers. A circuit diagram is shown in Fig. 1.

Because the purpose of a power transformer is to change the value of the current entering the device, the ratio of current and number of turns in the primary and secondary coils is used to determine the differential rather than a direct comparison of the input and output current values. This is achieved by adding two small current transformers wrapped around the input and output wires of the transformer, referred to as CT1 and CT2.

These small current transformers each induce a current \( I_1 \) and \( I_2 \) in the circuit to which they are attached, valued at the ratios \( I_1 = \frac{I_p}{N_1} \) and \( I_2 = \frac{I_s}{N_2} \), where \( I_p \) and \( I_s \) are the primary and secondary currents, and \( N_1 \) and \( N_2 \) are the number of turns in the secondary-side coils of CT1 and CT2. These currents are opposite in direction to one another when they meet in the Differential Relay, which calculates their difference, or differential.

In an ideal situation, currents \( I_1 \) and \( I_2 \) are equal to one another and opposite in direction, and thus the differential current \( I_d = I_1 - I_2 = 0 \). If \( I_d \) is detected to differ, from 0 by greater than a pre-established threshold, referred to as the restraining current; then, a fault is assumed to be present in the transformer and a trip signal is sent to the circuit breakers to disconnect the transformer from the rest of the power grid [18] [19] [20] [21].

As we examine the differential protection algorithm, we see that it is designed to detect when the circuit on either side of the power transformer is pulling a current that does not match the expected value within tolerance. When this occurs, the algorithm will trip a circuit breaker to protect the power transformer by disconnecting it from the grid. It is possible for threat actors to alter the results of the differential protection algorithm to cause the opposite effect. See Fig. 2 for the truth table.

To compromise the differential protection algorithm, the threat actors will need to modify the memory locations of the input values from the analog to digital converter to the desired matching values. Threat actors can inspect the memory locations to determine typical values and then use those values to deceive the algorithm by modification to achieve the desired outcome, as shown in Fig. 2.

### B. HARMONIC PROTECTION

Before the main transformer reaches steady-state operation, transient issues must be considered before the differential protection circuit can operate reliably. All transformers must establish a flux in the transformer core. This flux causes a current known as the magnetizing current to flow. The magnetizing current appears as a differential current to the relay. The nonlinearity of the core results in a nonlinear magnetizing current waveform. When an abrupt change in the excitation voltage occurs, a large magnetizing current can flow [22].

When a transformer is switched into the circuit at any point of the supply voltage wave, the peak values of the core flux wave depend on the residual flux as well as on the time of switching. The peak value of the flux is higher than the corresponding steady-state value and is limited by core saturation. The magnetizing current necessary to produce the core flux can have a peak of eight to ten times the normal full-load peak and has no equivalent on the secondary side. This phenomenon is called the magnetizing inrush current and appears in the system as an internal fault to a differential protection system.

The maximum inrush occurs if the transformer is switched in when the supply voltage is zero. It is important for the

---

**FIGURE 1. Power Transformer with Differential Relay**

**FIGURE 2. Differential Protection Truth Table**
design of differential relays for transformer protection to take this into account so that the circuit breaker is not tripped owing to the magnetizing inrush current. A number of schemes based on the harmonic properties of the inrush current are used to prevent tripping owing to the large inrush currents. Overheating protection is provided for transformers by placing a thermal-sensing element in the transformer insulating oil-filled tank [4].

Current transformer (CT) and potential transformer (PT) signals are processed through data acquisition and these data are analyzed using an analog-to-digital converter (ADC). This signal is then decomposed using a Fourier transform and compared with a configurable threshold value. During the magnetizing inrush condition, the normally fundamental component, DC component, 2nd harmonic, 3rd harmonic, 4th harmonic, and 5th harmonic components are present as a percentage of 100%, 55%, 63%, 26.8%, 5.1%, and 4.1% respectively.

The magnetizing inrush current reaches the second harmonic current component. Thus, the magnetizing condition in the transformer is detected based on the 2nd harmonic and the fundamental current ratio. The fast Fourier transform (FFT) algorithm is used to extract the fundamental and all other harmonic components from no load to full load to faulty current signals. The full-cycle FFT algorithm can extract the exact fundamental frequency components from a given input signal.

According to the FFT algorithm, if the 2nd harmonic component increases by more than 20% of the fundamental component, then this situation is considered as a magnetizing inrush [22].

To implement the harmonic protection algorithm, we must first generate the required input signal. Electrical current is typically transmitted with an alternating current (AC) at an established frequency. In the United States, this frequency is 60 Hz. In our simulation implementation, we first need to create a class that has the ability to generate a set of sinusoidal waves.

We used our sinusoidal class to generate instances of a signal. We then assume the protected device has achieved steady state. This is typically when the current transformer is steady state. This is when the current transformer is initially turned on or turned off and the magnetic cores still achieve flux in the presence of an inrush current.

During this period, harmonic distortions are known to occur. By detecting the harmonic distortions, we note that the current values that will be inputs to the differential protection circuit will surely cause the circuit to return a “fail” condition. This “fail” is actually a false “fail”. If the harmonic restraint protection is detecting harmonics above a configurable threshold, then we know the differential protection is generating a false “fail”. But if the harmonic restraining protection is not detecting harmonics above a configurable threshold, we then assume the protected device has achieved steady state since the harmonic signals have dissipated and that the differential protection algorithms results are no longer “false”.

We will create a harmonic restraint protection class to implement the algorithm which is simply “If harmonic > fundamental * percent threshold then return TRUE (harmonic protection ON) else return false (harmonic protection OFF).”

As we examine the harmonic restraint protection algorithm, we see that it is designed to detect when the circuit is initially being energized such that an inrush current is present. The inrush current has harmonic components that distort the current waveform and its amplitude.

During this time, the differential protection algorithm does not function correctly. However, what if threat actors were to alter the results of the harmonic restraint algorithm to cause the opposite effect? Here, we assume that only harmonic restraint is modified. Presented in Fig. 4 is a truth table depicting the possible outcomes.

To compromise the harmonic protection algorithm, threat actors need to recreate a continuous signal using a sine waveform. Threat actors can sample the input waveform from the analog to digital converter to determine the magnitude and frequency of the waveform.

If the threat actors seek to cancel out the harmonic signals, they would need to employ their own FFT to determine the magnitude and frequency of those signals. They can then generate a negative signal to counter the input signal harmonics or the input signal itself.

Threat actors could potentially simplify their attacks further by making some key assumptions. The frequencies are standardized per country, but there are variances by application. Typically 60 Hz is used in the United States, and 50 Hz is used in Europe. Threat actors can assume the appropriate frequency for the target and can then quickly generate harmonic signals at high magnitudes and store them for playback in their code. This assumes that the threat actors seek to trip the harmonic restraint protection by adding harmonic signals. This would not work to cancel as one needs precise magnitudes to cancel.
If we assume that our threat actors seek to cause a maximum negative impact, threat actors will seek to manipulate the result of both the differential protection and harmonic restraint algorithms in tandem. The results of manipulating both are summarized in the table in Fig. 5.

Again, we assume that threat actors seek to cause a maximum negative impact. We can see in Fig. 6 a table depicting the maximum negative result if both protection algorithms are manipulated.

Looking at the table, it appears that threat actors would seek disable the harmonic restraint for simplicity while examining the differential protection to determine when they can cause a false trip or cause the algorithm to fail its protection.

C. MECHANICAL SYSTEMS

Another attack surface that physics-centric threat actors can target are the mechanical systems in a power transformer. These attacks constitute a mechanical attack surface. The following are sensitive data in a CPS that can be modified that will result in a change in behavior of the system.

1) Relay
A relay is controlled by a digital output. Threat actors can trigger or prevent the trigger of a digital relay with a proper output. Alternatively, an attacker can cycle a relay with the intent of damaging it through repeated actuation at a high frequency. This is simply done by writing values or by preventing the change in value to the memory-mapped output locations.

2) Distribution Tap Changers
Threat actors can control the tap changers to alter the voltage levels to create an over-current situation or an under-voltage to cause disruption. Threat actors can also cycle the tap changers mechanically at a high frequency to produce mechanical stress and damage on the taps.

Potential tap damage can result in internal shorts. This could be done with direct control of the outputs that control the tap changers or could be done indirectly by changing the monitored input value of the current such that the control system for the distribution attempts to change the tap settings.
to compensate. If this is done at a high frequency this will cause mechanical stress on the electromechanical systems. This is simply done by writing values or preventing the change in value to the memory-mapped output locations or changing the values of the intermediate calculations.

D. ATTACK SURFACE GATEWAYS

Malware can observe data by accessing memory-mapped locations as shown in the model in Fig. 7. Typically, external inputs are mapped from the hardware to memory locations. These locations are updated periodically based on the input’s sample rate, or those inputs can be interrupt-driven. An interrupt-driven input will raise an interrupt when a new value is loaded into the memory location. Threat actors can take advantage of the specifications of the microprocessor and controllers to determine where the memory-mapped input and output locations can be found and can then scan those memory locations.

In some CPS implementations, sensor data are delivered via a communication bus. Threat actors can compromise the communication bus before the data are transmitted by the sensor unit. Threat actors can then inject communications messages on the bus that provide false values but appear to originate from the sensor. Threat actors can also compromise the data on receipt of the message and alter it before it is passed to the consuming application from the communications layer.

Threat actors can alter the sensor data in various ways and create synthetic data. One way is to compromise smart sensors, so the data coming into the input port is already compromised. Another method is to alter the data at the output ports. Data can also be altered in temporary variables and registers for intermediate calculations or in cases where the value is copied from the I/O channel to preserve its fidelity through calculations to avoid write hazards caused by I/O channels overwriting a value.

The data can also be altered by modifying the calibration values. Calibration values are used by CPSs to specify system behavior by allowing varying thresholds that can be calibrated for system implementation in the field. Understanding their physics-centric impact on the system algorithms allows one to manipulate those values to deceive the system into erroneous behavior.

Constant writing to buffers is noisy to buffers where the sensor data are stored. This has a risk of detection. An alternative attack is to compromise the protection system by modifying the calibration settings that are used by protection algorithms to drive their logic decisions. This allows the protection systems to read and process the data normally and drive the outputs but the result is erroneous.

Another approach is to watch the events; once threat actors see a fault, they can present altered data. Threat actors will read data at selected points, buffers, and addresses using logic to recognize when the computation will be concluded. The data is changed before the final conclusion is reached, and the output signals are triggered.

One can allow the protection systems to read genuine data, perform harmonic analysis, reach conclusions and only look at the conclusion and alter it. The challenge here is that there may be a race condition at the output between threat actors’ writes, the system’s write and the activation of the relay based on the output write. This would be the case when setting the output buffers to the desired value that will trigger the relay or prevent the relay from operating.

Having a deep understanding of the physics of the system allows threat actors to take advantage of their operation and change these systems into an attack surface by which they manipulate and gain control of the physical system. Essentially the sensitive data associated with these systems are now gateways of system comprise through the attack surface. These physics-centric gateways can then be leveraged to build attack methods.

V. ATTACK METHODS

We now discuss our attack methods. The goal of these attack methods is to manipulate the physics data to cause physical damage. The physics data are the input to the CPS. These data are observed, stored, and acted upon by the CPS. These industrial devices then control the electrical grid, and any modification of this sensitive data at any point in the process would allow an attacker an entry gateway through the attack
sensitive data to the system and its output in a man-in-the-middle
concept, which leads to false conclusions regarding the state of the variables. These changes are made before they are used by the protection
system.

An interposition attack is an attack wherein threat actors
interpose between the input of the sensor and the output of the system. This attack will identify sensitive data that can be manipulated to deceive the CPS in such a way as to control the outputs of the CPS.

These sensitive data could be inputs into the system, could be temporary calculations that are manipulated internally in memory, or in the case where total control by threat actors is achieved, they are the outputs of the CPS. Namely the variable that could be changed to alter the operation of the controlled physical ecosystem. Sensitive data are physics-centric variables as they provide entrance analogous to gates through the attack surface to cause a change in system behavior. Hence these are sensitive data in this context.

In this study, we will look at differential protection, the harmonic restraint algorithms, and mechanical systems to better understand how they can be manipulated by threat actors for malicious intent. Protection algorithms are in place and are designed to reinforce the safety of equipment, ecosystems, and life. Threat actors looking to disrupt CPSs are doing so for these goals since a CPS is not a financial system or a repository of personal information.

Malware is targeting CPSs to compromise the safety and proper operation of the system. For example, tripping a circuit breaker might be a normal action in the correct conditions but doing so at the wrong time will cause disruptions. Furthermore, protection algorithms have deep knowledge of safety. Inhibiting their operation also results in improper operation, equipment damage, and life-threatening safety issues if equipment is not shut down in an emergency.

A. PHYSICS-AWARE ATTACK

A physics-aware attack is an attack in which threat actors utilize the knowledge of the physics of the ecosystem to launch an attack against a CPS. A CPS is designed to interact physically with the ecosystem for which it was designed to gather inputs and provide control and information outputs based on its control algorithms. This attack will identify sensitive data that can be manipulated to deceive the CPS in such a way as to control the outputs of the CPS.

These sensitive data could be inputs into the system, could be temporary calculations that are manipulated internally in memory, or in the case where total control by threat actors is achieved, they are the outputs of the CPS. Namely the variable that could be changed to alter the operation of the controlled physical ecosystem. Sensitive data are physics-centric variables as they provide entrance analogous to gates through the attack surface to cause a change in system behavior. Hence these are sensitive data in this context.

In this study, we will look at differential protection, the harmonic restraint algorithms, and mechanical systems to better understand how they can be manipulated by threat actors for malicious intent. Protection algorithms are in place and are designed to reinforce the safety of equipment, ecosystems, and life. Threat actors looking to disrupt CPSs are doing so for these goals since a CPS is not a financial system or a repository of personal information.

Malware is targeting CPSs to compromise the safety and proper operation of the system. For example, tripping a circuit breaker might be a normal action in the correct conditions but doing so at the wrong time will cause disruptions. Furthermore, protection algorithms have deep knowledge of safety. Inhibiting their operation also results in improper operation, equipment damage, and life-threatening safety issues if equipment is not shut down in an emergency.

B. INTERPOSITION ATTACKS

An interposition attack is an attack wherein threat actors interpose the flow of the system and alter a value at a critical moment. This is illustrated in Figure 8. For example, as the input signals are loaded into the main memory, and these values are changed before they are used by the protection algorithms. In this way, the protection algorithm is deceived and leads to false conclusions regarding the state of the system.

Threat actors will interpose between the input of the sensor data to the system and its output in a man-in-the-middle attack. In this way, it observes the true input supplied to the controller, modifies it in some way, and outputs a value that will use the physics-centric knowledge of the system to deceive the protection algorithms. When the attack restricts itself to this type of operation, we refer to it as an interposition attack, as described in [23], because the malware stands between a function’s input and its output.

The interposition attack can modify the input sensor and signal data, modify the output sensor and signal data, and execute a replay attack where it replays a sampled section of sensor or signal data in place of the current valid data. When an interposition attack occurs on the input side, the input signals are manipulated to deceive the protection algorithms. The attack can also occur on the output side where the control outputs of the protection algorithms are manipulated to the desired values of the threat actors.

The output-side scenario requires less sophistication by threat actors because it does not need to deceive the protection algorithm, but threat actors need to understand the output specifications in order to produce signals that downstream devices will react to and to prevent the system from overwriting the malware outputs with correct outputs.

It is important to understand how a CPS obtains its inputs and delivers its outputs. These are gates on the attack surface through which an attacker can manipulate the control algorithms and drive outputs to control the physics ecosystem in question. Here we look at our use case example of a power transformer.

The input signals to the relay are analog (continuous) and digital power system variables. The digital inputs are of the order of five to ten and include status changes (on-off) of contacts and changes in voltage levels in a circuit. The analog signals are the 60 Hz currents and voltages. The number of analog signals needed depends on the relay function, but is in the range of 3-30 in all cases. The analog signals are scaled down to acceptable computer input levels and then converted to digital form through analog-to-digital converters [4].

The digital output of the relay is available through the parallel output port of the computer. Five-to-ten digital outputs are sufficient for most applications. The analog signals are sampled at a rate between 240 Hz and 2000 Hz. The sampled signals are entered into the random access memory (RAM) and stored in a secondary data file for historical recording. A digital filter removes the noise effects from the sampled signals. The relay logic program determines the functional operations of the relay and uses the filtered sampled signals to arrive at a trip or no-trip decision, which is then communicated to the system.

The heart of the relay logic program is a relaying algorithm that is designed to perform the intended relay function such as overcurrent detection, differential protection, or distance protection. [4].

C. COORDINATED DISTRIBUTED ATTACKS

In a coordinated distributed attack, threat actors launch several attacks to compromise various subsystems in a planned
and coordinated manner such that a cascading effect occurs within each interdependent subsystem. For example, an attacker may create an overload condition on the input current while simultaneously altering the reading of those values by the protection algorithm to have the fault condition go undetected, thus resulting in damage to the system.

Although a coordinated attack is more complicated, threat actors seeking to cause immediate damage will seek a means to do so such that their efforts produce a result. Otherwise the threat actors must accept only generating a false positive fault or wait for a natural equipment failure event to occur, such as a short to ground caused by a downed power line, or a failed transformer winding in the same circuit as a compromised protection algorithm controller to create a more damaging false negative scenario. Given that threat actors seek to have the fruits of their labor mature, threat actors will most likely execute a coordinated distributed attack to get more significant results.

An example would be to execute an active attack against the tap changer system that will cause mechanical damage to the tap changes and the transformer core thereby creating an internal fault. Coordinated with this attack would be a passive deception attack on the differential and harmonic protection systems to mask the internal fault from the protection systems to keep the equipment in service and thereby causing further damage to the system and other systems connected to the malfunctioning power transformer.

**D. PHYSICS-AWARE MALWARE EMULATION**

We replicate a threat actors’ tactics and actions by devising an attack strategy and creating an emulation of the attack. We assume that the attack will be an interposition attack that alters physics data but does not alter the code. Threat actors will use the physics laws of a power transformer to present data that will deceive the protection algorithms. In this case, we will focus on adversarial computation data of the differential and harmonic protection algorithms. We will also emulate an active attack against the tap changer system to emulate a destructive attack that could be coordinated with a deceptive attack on the protection systems to mask the damaging result.

The following is an overview of the attack our emulated threat actors perform in order to allow a fault to occur without the differential protection algorithm detecting the fault and removing the transformer from the power grid. The basic attack works regardless of the origin of the physics data.

We first consider the actions that threat actors would take to launch an attack on a networked power transformer. Threat actors will access the digital controller that is connected to relays and other sensors on the transformer via remote shell. Threat actors then use a debugger application to dump memory content. Next, threat actors search the memory dump for locations of buffers that hold physics data of interest. Then, threat actors will place a watch point in these locations and monitor. Next, the threat actors find the location of the object representing a circuit breaker and note the Boolean variable for ”break” or ”don’t break”. Finally, threat actors will observe the system in action and see when the threshold is reached in the protection algorithm. If the circuit breaker object is accessed and its Boolean is switched, they will switch it back so no action is taken.

These actions are straightforward. The only difficulty is the search through the memory dump for the addresses of input and output currents and the circuit breaker object. The threat model may then place watch points and observe the system. It will learn what input and output current values lead to the circuit breaker being tripped.

Threat actor processes may find pertinent buffers via a classification task and then use watch points to monitor the data changes in these buffers of interest. Threat actors next develop an approach that can predict or conclude, in a timely fashion, that an internal fault has occurred that will lead to a circuit breaker being tripped. Finally, threat actors can take action to suppress the tripping of the circuit breaker, leaving the power transformer unprotected.

We present emulated attacks that threat actors could make against the power transformer or other similar CPS that employ the same physical protection methods. We will assume the use of a CPS that relies on microcontroller based algorithms. The inputs into the microcontroller are the current sampled on either side of the power transformer.

The current sensors on either side of the power transformer are calibrated to create an input current in the expected operating range for comparison. The current sensors could be direct inputs into a microcontroller. They could also provide their data to the microcontroller via a network connection.
and message. We assume the use of alternating current for our power distribution system as these are the most widely used. We note that the physics of the current in an alternating current power distribution system will be sinusoidal.

We noted in the harmonic restraint protection system that the algorithm uses Fourier analysis to detect the presence of harmonic signals on the orders of 2-5. When harmonics are present, this is typically an indication of inrush current when the system is first energized. For the Fourier analysis to operate, the system must sample the current values and track them in a vector of values for each time slice. The vector needs to be long enough to satisfy the Nyquist relationship that requires the sample rate to be at least twice that of the frequency of the measured signal. At 60 Hz for alternating current that could have a 5th order harmonic at 5*60Hz = 300 Hz, the minimum sample frequency needs to be 600 Hz. This implies that large vectors that need to be filled before the Fourier Analysis can be executed. Hence, these vectors hold a time history of samples of the input current.

Threat actors will leverage this knowledge of the power transformer CPS to devise algorithms that take advantage of the physics of the system, the known operation of the algorithms, the variables used, the vulnerabilities in any communication methods, and the vulnerabilities in the microcontroller operating system, application code, and hardware. We describe a set of algorithms that threat actors can develop to deceive a power transformer protection systems.

Algorithm 1: Physics Aware Malware Emulation on Power Transformer

1. Power-Transformer-Malware-Emulation()

   *Input*: Set of known physics data consisting of \( I \cup O \);
   *Output*: Compromised System

2. \( \Pi \leftarrow \) Set of known physics data consisting of \( I \cup O \); \n3. \( X \leftarrow \) Set of variable signatures for each \( \pi \in \Pi \);
4. \( \Psi \leftarrow \) Set of variable locations in memory;
5. \( M \leftarrow \) Memory space of microcontroller;
6. \( \Delta \leftarrow \) Register space of microcontroller;
7. for \( \forall \chi \in X, \psi \in \Psi \) do
   8. \( \psi \leftarrow \) Search-Variable(\( \Delta, \chi \));
9. Given \( \Psi \subseteq X \);
10. Harmonic-Restraint-Attack-Emulation(\( \Psi \));
11. Differential-Attack-Emulation(\( \Psi \));
12. Tap-Changer-Attack-Emulation(\( \Psi \));

The task of the main algorithm described in Algorithm 1 is to gather information on the input physics data to the power transformer system, classify these data into relevant subsets, find the memory locations of specific variables, and then to call individual attack algorithms conditioned on the data it has successfully located and classified within the controller memory space.

Algorithm 2: Search Memory For Specific Variables

1. Search-Variable(\( \Delta, \chi \))
   *Input*: Memory and Register space \( \Delta \), Variable signature \( \chi \)
   *Output*: Set of Variable Location in Memory \( \Lambda \)

2. Let \( \lambda \in \Lambda \) be the set of found variables in \( \Delta \)
3. for \( \forall \delta \in \Delta \cap \forall \chi \in X \) do
4. if \( \delta \) matches with signature \( \chi \) then
5. \( \lambda \leftarrow \delta \)
6. Classify as \( I \) or \( O \)

The algorithm first groups the input and output data into a single set. The input and output data consist of input channels to the system and the outputs. The set of outputs is generally just a signal indicating whether or not to trip the circuit breaker given a particular state of the system. Next, a set of variable signatures is instantiated.

These signatures will allow the malware to identify the physics data of interest as it searches through the microcontroller’s memory space. When a match is found between a variable signature and data held in memory, this memory location is recorded in a data set along with the type of physics data it represents. This search process is described in algorithm 6 and the flow is shown in Fig. 9.

Once the memory and register space has been searched and the locations of the data of interest are collected, these data are used to determine which of the possible attacks to implement. Each has a specific set of physics data must be used and manipulated. If each required physics data has been located by algorithm 6, then the attack algorithm is called.

E. HARMONIC RESTRAINT ATTACK EMULATION

The harmonic restraint algorithm described in in Algorithm 3 is designed to detect harmonics in the input signal and determine if those values are large enough to imply that the system is experiencing inrush current. Threat actors can exploit these physics to attack the algorithm. Threat actors can do so by manipulating the input signal, changing the calibrations, or directly impacting the output variable. The result is controls when to engage the harmonic restraint on the differential protection algorithm. This is discussed in the truth table in Fig. 4.

The following are different types of attacks that can be executed against the harmonic restraint protection algorithm.

1) Harmonic Restraint Output

If the malware can simply write to the output value, it can easily set the value to either communicate that harmonic restraint is ON or that harmonic restraint is OFF. This could be a manipulation of an internal variable that detects each harmonic or could be the final output variable.
The malware can then create a signal at any frequency of the signal from the Fourier analysis. The physics of the system indicates that the signal for the current is sinusoidal. The malware now knows the amplitude and frequencies and then use that information to attack the system. Fourier Analysis to determine if the signal contains harmonics. If the malware can access the vector that contains the current value or the smallest value of the variable to force the harmonic restraint system to trip on even small harmonic values. Note that harmonics must actually be present in the signal for this to occur. Many algorithms will check if harmonics are present before proceeding execution.

3) Harmonic Restraint Amplitude and Frequency Manipulation

By manipulating the amplitudes or frequencies found by the Fourier analysis of the harmonic algorithm, one can also cause the harmonic system to provide false information. If we look at Fig 10 as an example, if one were to delete those amplitudes and frequencies from the return vector by subtracting those signals from the input stream, then there would not be harmonics found by the system. Another example would be to change the amplitude of the fundamental signal or the harmonic amplitudes to create the desired logic condition per the harmonic algorithms known approach: $\text{FundamentalAmplitude} \times \text{HarmonicAmplitude} > \text{HarmonicRestraintCutoff}$

The amplitude values can be changed to either the largest value or the smallest value of the variable to force the condition to evaluate as either TRUE or FALSE to get obtain desired effect.

4) Harmonics Restraint Signal Manipulation

If the malware can access the vector that contains the current sample of the time history of the signal that will be used by the harmonic restraint algorithm, it can perform its own Fourier Analysis to determine if the signal contains harmonics and then use that information to attack the system.

In the first case, if there are no harmonics detected after Fourier analysis, the malware now knows the amplitude and frequency of the signal from the Fourier analysis. The physics of the system indicates that the signal for the current is sinusoidal. The malware can then create a signal at any given harmonic by multiplying the known frequency by an integer.

With the amplitude value and knowing that the harmonic restraint is looking for harmonic amplitudes over a certain threshold, one simply creates a sinusoidal signal with the known amplitude at a desired harmonic frequency and adds it to the existing signal. Now, the resulting composite signal is fed through the harmonic restraint, which finds the false harmonic hence deceiving it into triggering the harmonic restraint.

In the second case, if harmonics are detected after the Fourier analysis, the malware now knows the frequencies and amplitudes of each harmonic signal. The physics of the system indicates that the signal for the current is sinusoidal. The malware can create sinusoidal signals at the same magnitude amplitude and frequency for each harmonic found and then subtract those values from the signal, which then effectively cancels out those harmonics. The result is a clean signal with no harmonics, which deceives the harmonic restraint algorithm into disabling the harmonic restraint.

Another approach here is simply to either record and play back a clean signal or to create a sinusoidal signal at the fundamental frequency and amplitude, which then replaces the existing signal samples in the signal vector. Physics-centric threat actors can also simplify their malware with key assumptions such as a 60 Hz fundamental frequency and also common amplitudes based on the use case of the power transformer. This means smaller code and faster execution which aids malware in going undetected. In all cases the phase angle of the signal also needs to be determined, especially considering the impact of the signal going on through a differential protection algorithm as shown in Fig. 10.

F. DIFFERENTIAL ATTACK EMULATION

The differential protection algorithm is described in section IV. The microcontroller is programmed to receive input from the primary-side circuit transformer and secondary-side circuit transformer. Because of the particular ratio of turns in the windings of the transformers, these two input currents are assumed to sum to zero under normal circumstances. This calculation is performed by a microcontroller located in the circuit relay. As shown in Fig. 11, if this sum exceeds a threshold stored in the microcontroller’s memory, the digital...
Threat actors who intend to inject malicious code into this protection algorithm have a few options. To remain dormant and undetected for some time, it is important that the malware not manipulate more data than is necessary to carry out its task. In this case, the data that may be altered to cause issues for the differential protection algorithm range from the input and output currents supplied to the transformer ($I_p$ and $I_s$), the number of turns in the circuit transformers ($N_1$ and $N_2$), or the various calculations used in the comparison.

The attack on the differential protection algorithm is as follows. Its goal is to either mask the presence of a true fault in the system, or to make it appear as though as there is a fault when none exists in reality. The attack attempts to control when the circuit breaker is tripped, or suppresses a trip entirely as fits with the threat actor’s overall attack goal.

First, after the set of physics data $\Psi$ has been collected, the location of the circuit breaker Boolean variable (trip/ do not trip), the differential current values, and the restraining current are noted. If a false negative is desired, the attacker will utilize a watchpoint placed at the memory location of the variable representing the circuit breaker.

When a true fault is detected in the system, the circuit breaker Boolean value is written to. The malware will be alerted to this by the watchpoint and will immediately change the value back before the signal propagates to the physical circuit breaker object. Thus each time a true fault is detected, the circuit breaker will not receive the message to isolate the power transformer, and this puts the device at greater risk of damage as it continues to operate in an unsafe state.

The opposite tactic of tripping the circuit breaker when there is no fault is just as straightforward. Any watchpoint placed on the circuit breaker must be deactivated so that the previously described attack is not triggered. Next, depending on the data available, the malware will either overwrite the restraining current value with 0 or else rewrite the value held in the differential current buffer to be more than the restraining current.

All that is needed is for the digital relay to detect a differential current greater than the restraining current, at which point it will write to the circuit breaker variable and disconnect the power transformer from the power grid. Thus, threat actors have two options: suppress a legitimate fault from causing the circuit breaker to trip, or to trip the circuit breaker when no corresponding fault exists.

**G. TAP CHANGER ATTACK EMULATION**

Threat actors seek to cause damage to the tap changer system by oscillating the tap changer actuators repeatedly to cause mechanical stress and failure from overuse. We assume that the tap changers are actuated by computer-controlled outputs. These output buffers are either ON if a tap changer is actuated or OFF is the tap is disengaged.

We also assume that there is a real time control loop that maintains the output states of the tap changers. On each pass through the control loop, the algorithm described in Algorithm 5 tracks the state and alternates the outputs to OFF and then on the next ON. It will continue to do so at the
**Algorithm 3: Malware Attack Emulation on Harmonic Restraint System**

1 **Harmonic-Restraint-Attack \( (\Psi) \)**
   
   **Input** : \( F_{\text{Harmonic}}O \), Frequency\( K \), Set of tuples of \{\( H2, A2, A1, S2, Ict1 \)\}, \{\( H3, A3, A1, S3, Ict1 \)\}, \{\( H4, A4, A1, S4, Ict1 \)\}, \{\( H4, A5, A1, S5, Ict1 \)\}, \( Ict1 \) \( \in \) \( \psi \) where \( Hn \) is boolean to denote if harmonic condition satisfied, \( A\) \( n \) is the amplitude, \( S\) \( n \) is the cutoff ratio, and \( Ict1 \) is the current value.

   **Output** : Compromised System

2 Given that \( F_{\text{Harmonic}}O = H2 \cup H3 \cup H4 \cup H5 \)
3 Such that we know for a given \( Hn \) what values of \( Ict1 \), \( A\) \( n \), \( A1 \), and \( S\) \( n \) result in \( Hn \) to be True or False
4 Based on variable location found exploit the system.
5 **if** \( F_{\text{Harmonic}}O \in \psi \) **then**
6   To Disable Differential Protection: \( F_{\text{Harmonic}}O \leftarrow \text{TRUE} \)
7   To Enable Differential Protection: \( F_{\text{Harmonic}}O \leftarrow \text{FALSE} \)
8 **if** \( (H2 \vee H3 \vee H4 \vee H5) \in \psi \) **then**
9   **for** \( \forall Hn \in \Psi \) **do**
10   To Disable Differential Protection: \( Hn \in \psi \leftarrow \text{TRUE} \)
11   To Enable Differential Protection: \( Hn \in \psi \leftarrow \text{FALSE} \)
12 **if** \( (A1 \vee A2 \vee A3 \vee A4 \vee A5 \vee S2 \vee S3 \vee S4 \vee S5) \in \psi \) **then**
13   To Disable Differential Protection:
14     \( A1, A2, S2 \leftarrow \text{value to satisfy} \ A2 \geq A1 \ast S2 \text{ is TRUE} \)
15     \( A1, A3, S3 \leftarrow \text{value to satisfy} \ A3 \geq A1 \ast S3 \text{ is TRUE} \)
16     \( A1, A4, S4 \leftarrow \text{value to satisfy} \ A4 \geq A1 \ast S4 \text{ is TRUE} \)
17     \( A1, A5, S5 \leftarrow \text{value to satisfy} \ A5 \geq A1 \ast S5 \text{ is TRUE} \)
18   To Enable Differential Protection:
19     \( A1, A2, S2 \leftarrow \text{value to satisfy} \ A2 < A1 \ast S2 \text{ is TRUE} \)
20     \( A1, A3, S3 \leftarrow \text{value to satisfy} \ A3 < A1 \ast S3 \text{ is TRUE} \)
21     \( A1, A4, S4 \leftarrow \text{value to satisfy} \ A4 < A1 \ast S4 \text{ is TRUE} \)
22     \( A1, A5, S5 \leftarrow \text{value to satisfy} \ A5 < A1 \ast S5 \text{ is TRUE} \)
23 **if** \( Ict1 \in \psi \) **then**
24   To Disable Differential Protection:
25     **for** \( n \in 2 \) **to** \( 5 \) **do**
26     Add in false harmonic signals to the fundamental signal
27     \( Ict1 \leftarrow Ict1 + A1 \text{ at Frequency}\( K \) \ast n \)
28   To Enable Differential Protection:
29     Playback the Fundamental Signal Only
30     \( Ict1 \leftarrow A1 \text{ at Frequency}\( K \) \)

**Algorithm 4: Malware Attack Emulation on Differential Protection Algorithm**

1 **Differential-Attack \( (\Psi) \)**
   
   **Input** : The set \( \Psi \) consisting of all physics data of interest

2 **Output** : Fault status = True, False
3 \( K \leftarrow \Psi_{[\text{circuit breaker object boolean}]} \)
4 \( I \leftarrow \Psi_{[\text{restraining current}]} \)
5 \( P \leftarrow \Psi_{[\text{restraining current}]} \)
6 **if** value written at \( K \) **then**
7 \( K \leftarrow \text{False} \)
8 Create False Positive:
9 Disable watchpoint at \( K \)
10 \( I \leftarrow \text{value greater than} \ P \)
11 **or**
12 \( P \leftarrow 0 \)

**Algorithm 5: Malware Emulation on Tap Changer**

1 **Tap-Changer-Attack-Emulation \( (\Psi) \)**
   
   **Input** : \( \text{TapOutput} \in (\Psi) \)

2 **Output** : Compromised System
3 To Oscillate the Tap Changer System On and Off at High Frequency to Cause Damage:
4 **if** \( \text{TapOutput} \) is ON **then**
5 \( \text{TapOutput} \leftarrow \text{OFF} \)
6 **else**
7 \( \text{TapOutput} \leftarrow \text{ON} \)

**VI. ATTACK EMULATION**

We leveraged code written in Python to generate a sinusoidal input signal for the current. This signal can be manipulated to add harmonic orders to simulate harmonics and manipulated with a small random noise value. The intent was to simulate physical signals.

To test our simulation code, we performed various edge cases. We looked at cases within the tolerance of the differential protection algorithm, outside of tolerance of the differential protection algorithm. We used test cases with harmonics and without harmonics. We validated the operation of the transformer simulation and the protection algorithms by exercising the normal operation signals, abnormal signals, and edge cases to trigger the various algorithms to see if we achieved expected results for our test cases.

Next we created the attack emulations. We then tested each attack emulation against the model to determine the results. This allowed us to refine and improve our attack algorithms iteratively. Finally we analyzed the overall ecosystem to identify defensive techniques.
A. POWER TRANSFORMER SIMULATION

Our simulation begins with a simple implementation of a power transformer. As was described in Section II, a power transformer is simply a pair of coiled circuits placed in close proximity to one another. Current in one circuit will induce a current in the other, and this current can be multiplied or divided based on the ratio of the number of turns in the two windings of the coil.

Thus, a simulated power transformer object requires only an AC signal input to its primary side \( i_p \), variables to hold the number of turns in the primary side winding \( N_p \) and the secondary side winding \( N_s \), and a function that relates these terms to produce an output secondary current \( i_s \). The input signal \( i_p \) is generated by a function that takes as input the desired amplitude, frequency, phase shift, and time, and outputs an array of evenly spaced samples from the sinusoidal.

This array is input to the power transformer object as \( i_p \). A function takes this array one sample at a time and calculates \( j * N_p / N_s \) for each sample \( j \) in the input array. This is the ratio that determines whether the transformer is a step-up or step-down transformer. This new array is output as the secondary current \( i_s \). Our simulation, to allow a more realistic input to the differential relay, adds in a small amount of noise to the output signal.

In this case, simulated is a step-up transformer that multiplies its input signal by 2. As can be seen, the amplitude of the currents, as well as the time over which the wave is sampled, are both artificially low. Realistically, over the course of one second there would be 60 cycles of the sine wave, or in other words, the frequency of the sine wave would be 60 Hz. In the interest of presentation, this has been reduced to 1 Hz, and all subsequent plots reflect this choice.

In addition, the input sine wave was created as a smooth pure waveform, while the output can be seen to include some degree of noise. Again, demonstrates the functionality of the differential algorithm. Realistically, both currents would include a small amount of noise under normal circumstances.

B. DIFFERENTIAL RELAY IMPLEMENTATION

Separate from the power transformer, a software object for the differential relay and circuit was created to perform the work of the differential protective algorithm. The differential circuit takes \( i_p \) and \( i_s \) as inputs from the power transformer object and uses these in conjunction with variables \( N_1 \) and \( N_2 \) which represent the number of turns in the primary side and secondary side circuit transformers, or \( CT_1 \) and \( CT_2 \). The value of \( i_1 \) the current being induced by \( CT_1 \) and the primary side current, is equal to \( i_p / N_1 \).

The same holds true for \( i_2 \) with \( i_s \) and \( N_2 \) substituted in. Similar to the primary transformer, the CTs are able to adjust the current induced in their secondary sides. This is used in the differential circuit to undo the multiplication done on \( i_p \) to create \( i_s \). In our experiment shown in Figure 12, \( i_s \) is twice the value of \( i_p \).

Therefore, in order to make a comparison with \( i_1 \), \( i_2 \) must be stepped down at the same ratio that \( i_s \) is stepped-up. With \( N_2 \) set to equal 2, and \( N_1 \) set to equal 1, \( i_2 \) is halved and can be entered into the differential protection algorithm. This normalization of current values is done so that the only effect that alters the currents is an internal fault in the transformer.

To set a threshold for testing the differential, the relay dynamically determines a restraining current \( i_{rt} \) based on the values of the input differential circuit currents. To this end, the simulation calculates the Root Mean Square (RMS) value of the buffer of current values. The RMS is often used as a convenient measure of the “average” voltage or current supplied by an AC source.

Because AC is fluctuates between its maximum and minimum values, the RMS considers values across time and outputs a single value that is equivalent to a DC source supplying the same amount of power. The RMS is determined by taking the square root of the mean of the sum of each sampled value from the sinusoidal squared.

Once this is calculated, there are at least three possible choices to create \( i_{rt} \). The sum of the two RMS values for \( i_1 \) and \( i_2 \) can be used. An average of the absolute value of the difference between the two is an option. Finally, the maximum between the two can serve as \( i_{rt} \). These methods all produce very different values; however, for this simulation, the sum option was chosen. This is the largest of the three, and is therefore the most conservative in declaring a fault state.

Finally, the differential current \( i_d \) is calculated and compared to \( i_{rt} \). \( i_d \) is the amount of current moving in the relay circuit when \( i_1 \) and \( i_2 \) are present. As described above, these two currents will be very similar to one another when there is no fault drawing away current or causing an unexpected spike. They enter the circuit in opposite directions, so when they meet in the conjunction where the differential relay sits, there is often little to no current in the loop.

Thus, the simulation works by finding the absolute difference between the \( i_1 \) and \( i_2 \) at each associated value, and then subtracting this \( i_d \) from the calculated \( i_{rt} \). The normal situation is shown in Figure 12. Noisy \( i_2 \) and pure \( i_1 \) can be seen on top of one another. They are close because no fault detected. Staying along the x-axis is the \( i_d \) value calculated from the distance between them. The differential current is shown in Figure 12. The values increase as the amplitude of the input currents increase, but this never exceeds the value of .10, meaning that these currents are far from signaling the presence of a fault.

C. DIFFERENTIAL MALWARE EMULATION

With our power transformer and protective system simulation in place, we can now describe the results of implementing a physics-centric deceptive interposition attack emulation on the differential protection algorithm. As presented section V-D, the differential malware must first gather data and the locations in memory in which these data are stored. This is done in a real scenario via live memory analysis and machine learning classification algorithms.
In our simulated attack, the malware object obtains the calculated value of the restraining current $i_{rt}$ from the differential relay object. It then locates the $i_2$ buffer (array) and creates a copy of its contents. For each value in the $i_2$ buffer, the malware adds the value held in $i_{rt}$ so that the entire buffer is shifted upwards. This compromised buffer is then rewritten over the old $i_2$ before the protection algorithm has had a chance to read from it to calculate the differential current.

Once this is done, $i_d$ that results from the difference in values of $i_1$ and $i_2$ will be shifted towards the threshold value of $i_{rt}$ regardless of the value assigned by the algorithm. In this way, a fault will be tripped by the differential relay despite the malware doing nothing but modifying the values of one buffer input to the protection algorithm.

Figure 13 shows this scenario in visual form. The noisy blue and pure red sine waves represent the true values input by the circuit transformers, with their absolute difference appearing below them, near zero. The yellow sine wave, shifted up by a value equal to the restraining current, appears in yellow. The red sine wave $i_1$ is similarly subtracted from it, but this time the resulting differential current groups around the threshold value $i_{rt} - \text{sum}$. In this way, the malware steps in between calculations in the protection algorithm, modifies the value of one buffer, and allows the algorithm to finish its calculations.

VII. FUTURE WORK

The focus of this paper is to demonstrate the need for deep physics awareness of CPSs that threat actors could leverage to attack such a system. We intend to build upon this depth of physics knowledge and attack specification to identify and design countermeasures for defense. For example, in the case of the harmonic restraint, one potential best practice would be to have some type of watchdog timer or state machine on the harmonic restraint to disallow its operation past a given time period. In rush current occurs at the startup of equipment and not during steady-state run. If we are in a steady run state, then if the system sees the harmonic restraint activating, this might mean a potential threat actor attack. Other countermeasures can be identified using this work as a starting point.

VIII. RELATED WORK

The energy distribution grid predates the advent of networked computer systems. It largely depends on technology from that era for its communication and emergency response procedures. For the most part this involves operators in substations calling one another on telephones. A smart grid is an energy distribution grid in which substations can monitor the state of various devices and communicate instantaneously with other substations. This would allow automated rerouting of electrical power and enable operators to pinpoint the location of faults in the system.

This omniscience of the grid relies on processors and sensors installed on all devices that make up the grid and connect these to the internet. Of course, with the convenience of internet-connected sensors and controllers comes the risk of threat actors remotely connecting to these devices over the internet. They may then observe the state of the system or even modify the programs running on the processors. This has dire repercussions on the integrity and safety of a smart grid. Considerable work has been done to model these threats and provide a useful framework for how the grid may be hardened against such intrusions.

The Idaho National Laboratory ran the Aurora Generator Test in 2007 to demonstrate how a cyberattack could destroy the physical components of the electric grid. The experiment used a computer program to rapidly open and close a diesel generator’s circuit breakers out of phase from the rest of the grid and cause it to explode. This vulnerability is referred to as the Aurora Vulnerability [24]. The Aurora Vulnerability was also addressed by IEEE through the work of Zeller [5].
This vulnerability is especially a concern because many grid equipment supports legacy communications protocols that were designed without security in mind. As such, they do not support authentication, confidentiality, or replay protection, which means any attacker that can communicate with the device can control it and use the Aurora Vulnerability to destroy it [24].

In support of methods to quickly and economically explore similar attack methods, our approach utilizes emulation to achieve goals similar to that of the Idaho National Laboratory. With emulation, one can demonstrate an attack without destroying equipment and maintain a credible level of fidelity for a more effective defense.

In Garcia et al., a PLC rootkit that implements a physics-centric stealthy attack against power grid control systems is demonstrated that implements a physics-centric man-in-the-middle attack against CPSs. The root kit damages the underlying physical system, while providing the operators with an exact view of the system that they would expect to see following their issued control commands [9]. We further explore the implementation of physics-centric attacks on power grid systems with a focus on exploiting the protection systems through the use of the physics data without modification of the code itself.

In 2015, three power distribution substations in Eastern Europe were the target of a well-coordinated attack that left more than 225,000 people without power for three hours. These attacks seemed to be the work of nation-state actors due to the resources, time, and expertise required for the attack. The ICS malware known as BlackEnergy[cert] seems to have played a key role in extracting login credentials and other information from IT workstations at the substation offices.

BlackEnergy is delivered via macros hidden in Microsoft Office products, and the attackers used social engineering techniques to convince users to enable them on their machines [8]. Once the BlackEnergy malware was run, it connected to command and control (C2) IP addresses and shared data necessary for threat actors to access the IT network with elevated privileges.

From there they could determine the VPN needed to access the ICS network, and thereby take direct control of the SCADA dispatch workstations and servers [7]. Finally, the KillDisk malware was used to delete selected files that corrupt the master boot record, rendering systems inoperable. This greatly increased the response time after the damage was done.

A model of smart grid threat actor scenarios is given in [25]. The authors base their model on a novel use of Petri nets, which allows them to simulate simultaneous offensive actions in the network.

A smart grid will involve many industrial control system (ICS) components to regulate the processes of electrical equipment. In [26] the authors show through a survey of notable cyberattacks how threat actors may quickly shift from the cyber to the physical and inflict real-world damage. The temporary shutdown of the Ukrainian power grid, the Turkish Pipeline explosion, and Stuxnet are all highlighted to show the potential for sophisticated cyberattacks on otherwise highly secure targets.

A survey of cybersecurity for smart grid power systems was presented in [27]. The essential architecture of a smart grid and substations is described in [28]. Interposition data attacks were analyzed in [29]. Cybersecurity issues related to protection relays was reported in [30].

In [31], a method to systematically identify the critical substation/IED hypothesized attack scenarios by eliminating a large number of insignificant cases in an online environment.

Many techniques have been developed to protect electrical transformers from both external and internal faults. Tuning a detection system that is sensitive to both types has proven to be a challenge for electrical engineers. Internal faults tend to be very small in magnitude and external faults (such as with power surges) are often very large in comparison.

The use of differential protection is the traditional method for detecting internal faults in transformers, while harmonic restraint is a common external fault detector that prevents the differential relay from tripping during initial energizing of the transformer. The two are used in concert in many systems, although variations on this theme abound.

Many authors have designed systems of protection for power transformers that utilize differential and harmonic analysis. In [19] [18] [20] [32] similar schemes were designed and tested using harmonic analysis to supplement differential analysis of incoming and exiting electrical current. The goal of each is to reliably identify inrush currents, external faults, internal faults, and the presence of threat actors. The exact meaning of “harmonic restraint” and “harmonic blocking”, typically used interchangeably in the literature, is examined in [33] and shown to be distinct concepts.

In a marked departure from the other techniques mentioned here, the authors of [34] developed a method for protecting power transformers that utilizes a machine learning algorithm. The algorithm is trained on data gathered from an electrical distribution substation and a resident power transformer. The authors show that the machine learning algorithm can successfully categorize outlying currents as either a fault or a normal fluctuation.

In [35], a successful intruder can block the measurement transmitting an electronic instrument transformer and send fake measurements to a corresponding protection system, which usually leads to its maloperation. In order to provide a complementary mechanism against other measures, context information-based cybersecurity protection is proposed in this paper. When the protection system detects a fault based on the measurements of a single or coupled instrument transformers, the measurements of the whole substation are collected to carry out pattern classification to validate the fault. A probabilistic neural network (PNN) used as a pattern classifier was constructed. The test of real and fake fault exemplars generated with specific rules show that the proposed
PNN can identify fake faults from real faults with selected smoothing parameters.

**IX. CONCLUSION**

The physics-centric modules of malware that attack the electrical power grid are designed to execute attack methods that cause physical destruction of power grid equipment. While these attack methods can be very creative, and hence could change in unpredictable ways, they all share designs that are driven by the attack surface of power grid equipment. Our emulation in python of protection algorithms, which work with the physics of a power transformer, enabled an in-depth discovery and analysis of the cyberattack surface of a power transformer.

Findings on the cyberattack surface of a power transformer were leveraged in this paper to ethically design concrete attack methods, which provide a viable sample of attack modus operandi. With roots in the cyberattack surface of a power transformer, similarly to attack techniques used by real-world malware and possibly by unknown malware of the present and the future, this sample of attack methods provides insights and indicators that inform a more effective design of defensive approaches against malware that attack the electrical power grid.

We emulated these attack methods in practice alongside protection algorithms, and thus empirically demonstrated and measured their effects on the physics of a power transformer. In conclusion, our contribution offers a line of work that could be extended and/or redesigned to produce more samples of attack methods on power grid equipment, with the objective of better informing more effective defense.

**REFERENCES**

[1] E. A. Lee, “Cyber physical systems: Design challenges,” in 2008 11th IEEE international symposium on object and component-oriented real-time distributed computing (ISORC). IEEE, 2008, pp. 363–369.

[2] S. Ying and J. Sztpanovits, “Foundations for innovation in cyber-physical systems,” in Workshop Report, Energetics Incorporated, Columbia, Maryland, US, 2013.

[3] G. D. Rockefeller, “Fault protection with a digital computer,” IEEE Transactions on Power Apparatus and Systems, no. 4, pp. 438–464, 1969.

[4] M. E. El-Hawary, Introduction to electrical power systems. John Wiley & Sons, 2008, vol. 50.

[5] M. Zeller, “Myth or reality—does the aura vulnerability pose a risk to my generator?” in 2011 64th Annual Conference for Protective Relay Engineers. IEEE, 2011, pp. 130–136.

[6] ICS-CERT, “Cyber-attack against ukrainian critical infrastructure,” 2016, [Online; accessed 22-April-2021]. [Online]. Available: https://ics-cert.cisa.gov/ics/alerts/IR-ALERT-H-16-056-01

[7] T. C. Robert M. Lee, Michael J. Assante, “Analysis of the cyber attack on the ukrainian power grid,” Tech. Rep., 2016.

[8] K. L. Global Research & Analysis Team, “Blackenergy apt attacks in ukraine employ spearphishing with word documents,” 2016, [Online; accessed 22-April-2021]. [Online]. Available: https://securelist.com/blackenergy-apt-attacks-in-ukraine-employ-spearphishing-with-word-documents/73440/

[9] L. Garcia, F. Brasser, M. H. Cintuglu, A. Sadeghi, O. A. Mohammed, and S. A. Zonouz, “Hey, my malware knows physics! attacking plcs with physical model aware rootkit.” in NDS, 2017.

[10] W. Eccles, “Pragmatic electrical engineering: Fundamentals,” Synthesis Lectures on Digital Circuits and Systems, vol. 6, no. 1, pp. 1–199, 2011.

[11] J. Faiz and B. Siahkolah, Electronic tap-changer for distribution transformers. Springer Science & Business Media, 2011, vol. 2.

[12] “Transformer short circuit considerations,” Hammond Power Systems. [Online]. Available: https://www.hammondpowersolutions.com/en-US/news/general/transformer-short-circuit-considerations

[13] S. Tenbohlen, J. Jager, J. Gebauer, P. Muller, J. Lapworth, S. Yukiyasu, B. Desai, A. McIntosh, G. Bastos, and L. Bo, “Transformer reliability survey interim report,” Electra, no. 261, pp. 46–49, 2012.

[14] A. Mohammed, “Power transformer: Learn the purpose, cost, and lead time to procure,” pageuru.com/2019/08/power-transformer/, 2020, accessed: 2021-02-24.

[15] S. Miller, N. Brubaker, D. K. Zafra, and D. Caban. Triton actor tap profile, custom attack tools, detections, and attack mapping. [Online]. Available: https://www.fireeye.com/blog/threat-research/2019/04/triton-actor-tap-profile-custom-attack-tools-detections.html

[16] J. Guinn II, L. Luque, and J. Ray. Managing malware crashoverride/industroyer malware assessment. [Online]. Available: https://www.accenture.com/_acmmedia/PDF-69/Accenture-Managing-Malware-Crash-Override.pdf

[17] S. Karnouskos, “Suzexnt worm impact on industrial cyber-physical system protections,” pp. 4400–4407, 2015.

[18] A. H. Hamouda, F. Q. Al-Anzi, H. K. Gad, and A. Gastli, “Numerical differential protection algorithm for power transformers,” in 2013 7th IEEE GCC Conference and Exhibition (GCC). IEEE, 2013, pp. 440–445.

[19] O. Dris, F. M. Elmareemi, and R. Fouad, “Transformer differential protection scheme with internal faults detection algorithm using second harmonics restrain and fifth harmonics blocking logic,” Semantic Scholar, pp. 1–5, 2007.

[20] R. Hamilton, “Analysis of transformer inrush current and comparison of harmonic restraint methods in transformer protection,” IEEE Transactions on Industry Applications, vol. 49, no. 4, pp. 1890–1899, 2013.

[21] H. S. Kainth and G. Sharma, “A new method for differential protection in power transformer,” IOSR Journal of Electrical and Electronics Engineering (IOSR-JEEE), vol. 9, no. 2, 2014.

[22] D. Patel and N. Chothani, “Digital protective schemes for power transformer.”

[23] C. Parampalli, R. Sekar, and R. Johnson, “A practical mimicry attack against powerful system call monitors,” in Proceedings of the 2008 ACM Symposium on Information, Computer and Communications Security, ser. ASIACCS ’08. New York, NY, USA: Association for Computing Machinery, 2008, p. 156–167. [Online]. Available: https://doi.org/10.1145/1368310.1368334

[24] Wikipedia contributors, “Aurora generator test — Wikipedia, the free encyclopedia,” 2021, [Online; accessed 21-April-2021]. [Online]. Available: https://en.wikipedia.org/wiki/Aurora_Generator_Test

[25] T. M. Chen, J. C. Sanchez-Aarnoutse, and J. Buford, “Petri net modeling of hypothesized attack scenarios on bus differential relays,” in 2014 64th Annual Conference for Protective Relay Engineers. IEEE, 2014, no. 4, pp. 741–749, 2011.

[26] M. G. Angle, S. Madnick, J. L. Kirtley, and S. Khan, “Identifying and anticipating cyberattacks that could cause physical damage to industrial control systems,” IEEE Power and Energy Technology Systems Journal, vol. 6, no. 4, pp. 172–182, 2019.

[27] Y. Yan, Y. Qian, H. Sharif, and D. Tipper, “A survey on cyber security for smart grid communications,” IEEE Communications Surveys & Tutorials, vol. 14, no. 4, pp. 998–1010, 2012.

[28] G. N. Ericsson, “Cyber security and power system communication—essential parts of a smart grid infrastructure,” IEEE Transactions on Power Delivery, vol. 25, no. 3, pp. 1501–1507, 2010.

[29] K. C. Sou, H. Sandberg, and K. H. Johansson, “On the exact solution to a smart grid cyber-security analysis problem,” IEEE Transactions on Smart Grid, vol. 4, no. 2, pp. 856–865, 2013.

[30] S. Ward, J. O’Brien, B. Beresh, G. Bennouyal, D. Holstein, J. T. Tengdin, K. Federo, M. Simon, M. Carden, M. V. Yalla et al., “Cyber security issues for protective relays c1 working group members of power system relaying committee,” in 2007 IEEE Power Engineering Society General Meeting. IEEE, 2007, pp. 1–8.

[31] R. Bulbul, Y. Gong, C.-W. Ten, A. Ginter, and S. Mei, “Impact quantification of hypothesized attack scenarios on bus differential relays,” in 2014 Power Systems Computation Conference. IEEE, 2014, pp. 1–7.

[32] A. Guzman, H. Altuve, and D. Tziouvaras, “Power transformer protection improvements with numerical relays,” CIREF Study Committee B5—Protection and Automation, vol. 11, 2005.

[33] K. Behrendt, N. Fischer, and C. Labuschagne, “Considerations for using harmonic blocking and harmonic restraint techniques on transformer diffuse

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/AACCESS.2022.3148046, IEEE Access
Olijnyk et al.: Physics-centric attack methods on a power transformer.

ferential relays,” in proceedings of the 33rd Annual Western Protective Relay Conference, 2006.
[34] M. Z. Jahromi, A. A. Jahromi, S. Sanner, D. Kundur, and M. Kassouf, “Cybersecurity enhancement of transformer differential protection using machine learning,” in 2020 IEEE Power & Energy Society General Meeting (PESGM). IEEE, 2020, pp. 1–5.
[35] S. Sheng, W. L. Chan, K. Li, D. Xianzhong, and Z. Xiangjun, “Context information-based cyber security defense of protection system,” IEEE Transactions on Power Delivery, vol. 22, no. 3, pp. 1477–1481, 2007.

JOHN OLIJNYK (M’94) is currently a PhD student in computer science at Oakland University in Rochester, Michigan, USA. He has completed a Masters of business administration (2004) and a Masters of Science in electronics and computer control systems engineering (1998) from Wayne State University in Detroit, Michigan, USA. He also has a Bachelor of Science in electrical engineering (1994), computer science (1994), and mathematics (1994) with a minor in physics also from Wayne State University.

He has worked in industry for 25+ years including IBM, Detroit Diesel Corporation, Ford Motor Company, Idaho National Laboratory, and Cognizant Mobility in Troy, Michigan in various technical and leadership capacities in the automotive, product development, manufacturing, and information technology domains. His research interests are focused on cyber physical system and software security engineering. Mr. John Olijnyk was awarded a US Patent for a Vehicle Convertible Track System: US6530621, Howard Donnelly Award for the top graduate of Wayne State University in leadership, service, and academics, Wayne State Board of Governors’ Certificate of Appreciation, Stephen P. Hepler Computer Science Academic Award, Detroit Chapter of the National Society of Professional Engineers Student of the Year Award.

Benjamin Bond was born in Waverly, Iowa, USA in 1990. He received the B.S. in philosophy from Iowa State University in 2014, the M.S. degree in information from the University of Michigan, Ann Arbor, and the M.S. degree in computer science from Oakland University, Rochester, Michigan, in 2017 and 2021 respectively.

From 2019 to 2021, he was a Research Assistant with the Cybersecurity Lab at Oakland University, under the supervision of Dr. Julian Rrushi. Since 2021, he has been a Ph.D. student in computer science at Purdue University, West Lafayette, Indiana, USA, and a Microkernel Development Intern in the Critical Infrastructure Security group at the Idaho National Lab, Idaho Falls, Idaho, USA. His research interests are in the area of computer and network security, in particular secure microkernels, embedded systems security, and cryptography.

Julian Rrushi is on the faculty of engineering at Oakland University, Michigan. He carries out research at the intersection of operating systems, computer architectures, data science and artificial intelligence to break new ground in cybersecurity. His research projects are supported by the U.S. Department of Defense, the U.S. Department of Energy, and industry. Prior to commencing his academic career, Dr. Rrushi was a senior vulnerability researcher with industry for several years.

He received a B.S. in Computer Science, an M.S. in Information Technology, and a Ph.D. in Computer Science, all from the University of Milan. His Ph.D. work focused on detection of malware and exploits in cyber physical systems. Dr. Rrushi is a recipient of the Young Faculty Award from DARPA, class of 2020, for research on the cyber physics of computing machines and networks.