A Symbolic Honeynet Framework for SCADA System Threat Intelligence

Owen Redwood, Joshua Lawrence, Mike Burmester

Chapter 7

A SYMBOLIC HONEYNET FRAMEWORK FOR SCADA SYSTEM THREAT INTELLIGENCE

Owen Redwood, Joshua Lawrence and Mike Burmester

Abstract Current SCADA honeypot technologies present attackers with static or pseudo-random data, and are unlikely to entice attackers to use high value or zero-day attacks. This chapter presents a symbolic cyber-physical honeynet framework that addresses the problem, enhances the screening and coalescence of attack events for analysis, provides attack introspection down to the physics level of a SCADA system and enables forensic replays of attacks. The work extends honeynet methodologies with integrated physics simulation and anomaly detection utilizing a symbolic data flow model of system physics. Attacks that trigger anomalies in the physics of a system are captured and organized via a coalescing algorithm for efficient analysis. Experimental results are presented to demonstrate the effectiveness of the approach.

Keywords: SCADA systems, honeypots, threat intelligence, visualization

1. Introduction

Cyber-physical systems are computational systems that monitor and control physical systems; they encompass control systems, sensor-based systems, autonomous systems, robotic systems and more. Cyber-physical systems that control critical infrastructures span across many industries and, depending on their application, may be called supervisory control and data acquisition (SCADA) systems, process control systems, industrial control systems or distributed control systems. These control systems are usually composed of sensors, actuators, communications devices and control processing units such as networked remote telemetry/terminal units (RTUs), programmable logic controllers (PLCs) and intelligent electronic devices (IEDs). While most older control systems were designed to be air gapped, studies show that many current systems are directly or indirectly connected to the Internet [1].

Historically, proprietary SCADA protocols and the complexity of SCADA systems provided some degree of security (albeit through obscurity). However, as SCADA protocols have become increasingly standardized and open, researchers have found increasing numbers of vulnerabilities [20]. Meanwhile, tools such as the Sentient Hyper-Optimized Data Access Network (SHODAN) (`www.shodanhq.com`), Every Routable IP Project (ERIPP) (`eripp.com`) and Industrial Risk Assessment Map (IRAM) (`www.scadacs.org/iram.html`) make Internet-facing SCADA systems easy to track and, sometimes, trivial to exploit. In fact, using these and other tools, researchers have estimated that between 2,000 to 8,000 new SCADA devices are connected directly to the Internet each day [2].

Unfortunately, the anatomy of cyber attacks and the range of potential physical impacts against SCADA systems are poorly understood [7]. Moreover, few solutions are available that simultaneously address both issues.

Honeypots are a promising approach for detecting, analyzing, deflecting and possibly counteracting the unauthorized use of information systems. This is achieved by replicating the behavior of the targeted systems and devices. The degree to which system behavior is replicated by a honeypot leads it to be characterized as a low-interaction or high-interaction device. Honeypots generally have no production value and interactions with these devices are likely to constitute reconnaissance or attack activity; this renders them effective for deployment in networks for the purpose of intrusion detection. Two or more honeypots that are networked together can simulate a network and can be configured such that all activity is monitored and recorded. Interested readers are referred to [6] for a detailed discussion of honeypots.

This chapter describes the Symbolic Cyber-Physical Honeynet (SCyPH) framework, which provides SCADA administrators with a solution that bridges the understanding of cyber attack anatomy and potential physical impacts. The design addresses the three main challenges of honeypot operation: maintenance, attack analysis and attack screening and coalescence [6]. For operators, the end result is greater situational awareness, deep attack introspection and threat intelligence about various types of SCADA system attacks. However, the ultimate goal is to push honeypot development into a new direction and provide a common ground for research in security vulnerabilities and attack impact for critical infrastructures. Only the ROS honeypot [14] presents the ability to detect deeper problems in SCADA systems, enabling the discovery of the objectives and techniques of sophisticated attackers. Modeling the underlying cyber-physical system helps capture a wide range of post-exploitation activities by attackers. Performing anomaly detection on the cyber-physical system model helps filter common automated activities by worms and botnets and focus the analysis on sophisticated attacks. Sophisticated attackers are enticed and kept engaged by using interactive human-machine interfaces that react to attacker changes in a realistic manner.

2. Related Work

Despite significant advances in general honeynet technology, few honeypots are available for cyber-physical systems in general and SCADA systems in particular. In 2004, CISCO's Critical Infrastructure Assurance Group launched the first SCADA honeynet project [16], which extended Honeyd to simulate services for a popular programmable logic controller. While the CISCO project is no longer active, its contributions to fingerprinting banners and honeypot services are still very useful. Two other open-source honeynet projects are currently active: one is the SCADA Honeynet Project launched by Digital Bond [4] in 2010 and the other is Conpot launched by the Honeynet Project [3] in 2013.

The SCADA Honeynet from Digital Bond [4] utilizes two virtual machines, a target virtual machine that simulates a Modicon Quantum programmable logic controller and the Honeywall virtual machine that monitors traffic and activity on the target. The Honeywall virtual machine can be placed in front of an actual programmable logic controller or other control device. Honeywall captures attacks on the target, creates reports, generates alerts and provides other management functions. The provided target simulates a limited-interaction HTTP HMI service, Modbus TCP, FTP, SNMP, VxWorks Debugger and Telnet.

The Conpot honeynet [3] provides virtual programmable logic controller slaves controlled by a master server that simulates a Siemens S7-200 CPU with a CP 443-1 communications processor to handle a large set of industrial control system protocols. By default, Conpot provides low-interaction and supports Modbus TCP and SNMP, but it can be configured to support the BACnet protocol and the Intelligent Platform Management Interface (IPMI). The default virtual slaves in Conpot can publish captured data to the `hpfeeds` system in order to share honeypot data.

Researchers at Harvard and Los Alamos National Laboratory have developed the RobotOS (ROS) honeypot [14], the first true cyber-physical honeypot that utilizes real robotic hardware as the target. The ROS honeypot provides a high-interaction, vulnerable human-machine interface that connects with robotic hardware running ROS. Other closed-source honeypot projects are almost certainly active. For example, in 2013, Trend Micro [21] released data from its three closed-source projects dealing with a water pressure station honeypot, a production programmable logic controller honeypot and a factory temperature control programmable logic controller.

3. SCyPH Framework

This section describes the Symbolic Cyber-Physical Honeynet (SCyPH) framework, including its principal design features and layers.

3.1 Overview

The SCyPH framework can capture exploitation and post-exploitation activities against SCADA human-machine interfaces and programmable logic controllers, respectively, as well as model the resulting physics impacts. The novel honeypot feature is the symbolic simulation of a cyber-physical system to implement anomaly detection and the screening and coalescence of real attacks that succeed in altering the physics of the system. The simulation of the physics of the cyber-physical system allows for the presentation of a high-interaction human-machine interface and emulated programmable logic controllers that interact with the human-machine interface, which ultimately allow attackers to influence the system physics (voltage, current, pressure, etc.). The SCyPH framework incorporates the following design features:

- **Modular Components:** All the components, namely, the anomaly detection engine, interactive human-machine interface and sampling services, are modular. This facilitates crowdsourcing as in the ROS honeynet [14].

- **Human-Machine Interface and Physical Model Coupling:** Human-machine interface interactions are coupled with the simulated physical model. This enables adversarial interactions with the human-machine interface and the sampling services to be logged, simulated and organized depending on how they influence the simulated physical model.

- **Partitioned Layers:** All the layers are strictly partitioned. This provides a clear divide of component responsibilities and facilitates community development of the framework.

3.2 Honeynet Layer

The purpose of the honeynet layer (HL) is to abstract common honeynet technologies that are independent of cyber-physical systems so that the framework can isolate activity in a cyber-physical-centric manner. The composition of the honeynet is flexible – many existing honeynet technologies can coexist with the SCyPH framework inside the honeynet layer. The primary target is the human-machine interface, which ultimately has to be integrated to communicate with the devices in the interaction layer (described below).

Figure 1 presents a general architecture of the SCyPH framework. The framework utilizes anomaly detection on the simulated physics to perform attack screening. Other attack screening and detection mechanisms can be integrated into the framework.

As shown in Figure 1, attacker activity originates from the Internet-exposed human-machine interface (or honeynet), which provides a web-based interface on port 80 (or 8080). For instance, by default, the human-machine interface could present standard web application attack vectors for an adversary to exploit. These include injection attacks (i.e., SQL, operating system commands,

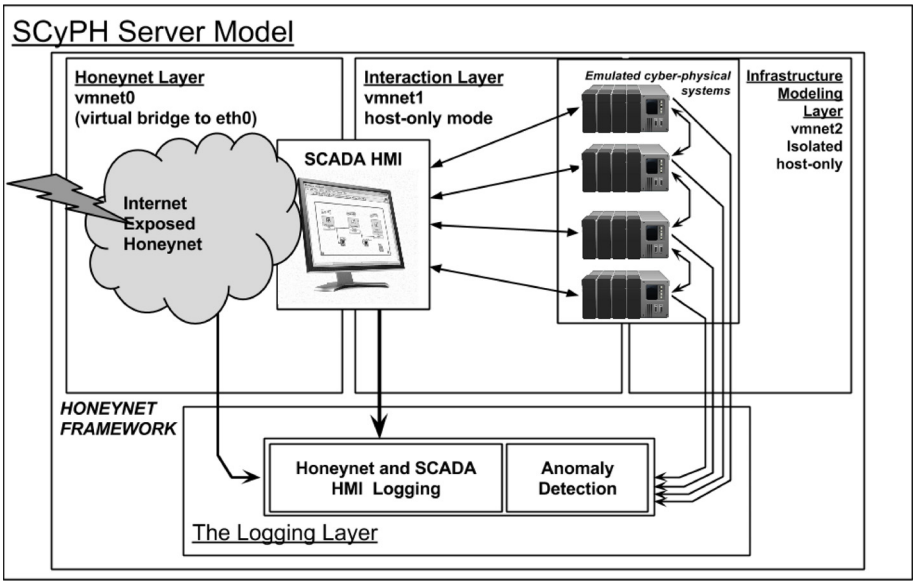

Figure 1. SCyPH architecture.

LDAP and Xpath), broken authentication, session management attacks, cross-site scripting attacks and others, depending on the attack surface of the human-machine interface [15]. Furthermore, services provided by the operating system potentially expose additional attack vectors.

The key requirement for other tools is that data collection and coalescence should be compatible with the logging layer; as described below, the `hpfeeds` protocol is used for logging. Furthermore, according to the honeypot tool survey presented in [6], it is beneficial to use server-side honeypot tools that have equal or better than good ratings for accuracy of emulation, good ratings for reliability; fair for support, and useful or essential ratings for utility. Specifically Dionea, Glastopf, Kippo and Conpot are recommended for their usefulness, support, reliability and support of `hpfeeds`. Readers should consult [6] for the ethical and legal issues involved in setting up SCADA or cyber-physical system honeypots.

3.3 Interaction Layer

The interaction layer (IL) facilitates the integration of the human-machine interface and the emulated cyber-physical system. This helps present a believable and highly-interactive human-machine interface for monitoring exploitation and post-exploitation activities. The features and inputs of the human-machine interface directly influence the firing rules of the actors of the process(es) in the interaction layer. The approach is agnostic to the specific

human-machine interface implementation. Interface alarms, events and buttons are integrated into the interaction layer.

Open-source, web-based human-machine interfaces are employed because they are easier to configure and modify. They capture realistic exploitation activity and, more importantly, post exploitation activity from the human-machine interface against the programmable logic controllers. Common human-machine interface components include a web server, database and web-based scripting language engine (PHP, Perl, Python or Java).

After an attacker has compromised the human-machine interface honeypot, he/she can observe and directly interact with `vmnet1`. Capturing attack data targeted at the remote terminal units or programmable logic controllers at the interaction layer is an important feature of the proposed approach. However, if an attacker's further exploitation exposes the operation(s) of `vmnet2`, then the attacker can compromise the data capture capabilities of the honeynet, which is a general operational concern with regard to honeypots [6].

3.4 Infrastructure Modeling Layer

The infrastructure modeling layer (IML) is primarily a symbolic data flow model that simulates cyber-physical system components to facilitate the interaction of programmable logic controllers with their human-machine interfaces. The programmable logic controllers each need to be integrated with the infrastructure modeling layer simulation.

Similar to a Kahn process network [8, 10], the data flow model in the infrastructure modeling layer defines a process as a set of signals, actors, firing rules and (optionally) one or more sampling service(s) to provide interaction with the human-machine interface.

- A signal φ is an output channel from one actor to the input of another actor. A signal φ comprises a sequence of tokens $x_m, x_{m-1}, \ldots, x_n$.
- An actor α maps input tokens to output tokens. Actors process incoming tokens from input signals in a first-in first-out (FIFO) manner.
- A set of one or more firing rules dictates when an actor α can fire. The token(s) that result from the most recent firing provide the state of the actor. When actor α fires, it consumes input tokens and produces output tokens.

Most tokens are numerical (e.g., voltage, current, pressure); however, exceptions require that tokens be symbolic to support the application of the approach to general cyber-physical systems (e.g., chemical and/or biological mixing).

Formally, an actor α is defined as $\{\varphi_{in}, F, \varphi_{out}, S\}$ where F is a set of firing rules and S is a set of sampling services. An actor for which φ_{in} is null can be seen as a source of tokens if φ_{out} is not null. Likewise, an actor for which φ_{out} is null, but φ_{in} is not null, can be seen as a sink. A set of concurrent actors $\{\alpha_0, \alpha_1, \ldots, \alpha_n\}$ compose a process, which can be extrapolated to mimic a variety of cyber-physical systems.

Theoretically, this approach allows the parallel simulation of continuous, discrete, multirate discrete or even totally ordered discrete time processes through the flexibility provided by the firing rules [8], without having to account for the timing costs of many measurements, conversions, industrial Ethernet latency and signals processing that would be necessary in a formally-modeled Kahn process network. However, this prevents the capture of attacks that target industrial Ethernet latency or signal processing. For ease of implementation, the proposed model does not attempt to meet the execution requirements of a formal Kahn process network [5, 8]. Thus, the model is, at best, a weak Kahn process network.

The operation of actors and their firing rules over signal channels simulate a process over `vmnet2`. The sampling service S supports the interactions of a remote terminal unit or programmable logic controller with a human-machine interface across `vmnet1` (Figure 1). The sampling service is instantiated by the interaction layer and handles the protocol(s) and interval(s) when a human-machine interface samples and/or interacts with the actor. For example, the Modbus, DNP3, MMS and BACnet protocols can be emulated to provide believable interactions with a human-machine interface.

The interaction layer provides the required names, numbers and information (i.e., system names, module identifiers, firmware identifiers, serial numbers, etc.) at initialization time. Optionally, specifications and variables utilized for anomaly detection may be provided by the logging layer at initialization time as well.

Each sampling service is defined as a control protocol between the human-machine interface and programmable logic controllers (or between programmable logic controllers) in the interaction layer. Protocol emulation can have significant engineering and reverse engineering costs because it is necessary to extract the protocol from actual programmable logic controller firmware/kernel images or software and emulate it in a compatible virtualization environment. This facilitates the monitoring of real post-exploitation activity against programmable logic controllers. However, sensors, merging units and other devices have to be simulated in order to integrate an emulated programmable logic controller with the infrastructure modeling layer simulation.

This research uses Conpot as a proxy to capture interactions and traffic with emulated programmable logic controllers. Conpot can be configured to integrate real SCADA hardware; however, the functionality of the hardware would still have to be integrated with the infrastructure modeling layer simulation.

3.5 Logging Layer

The logging layer utilizes anomaly detection in the infrastructure modeling layer physics simulation to filter uninteresting attacks. Data is collected using common honeynet solutions.

Data Collection. Honeynet deployment, administration and capture of network traffic are technical problems that have been addressed [6]. How-

ever, the main candidates considered for this task are again tools that use `hpfeeds`, although this is a choice left to the implementation. The collection of non-network-traffic data (i.e., files and/or processes created or manipulated by attackers) are usually specific to an individual honeypot; however, robust solutions for coalescing this data from a network of honeypots are not yet available.

Anomaly Detection. Given the state machine in the infrastructure modeling layer and a method for sampling the state of an actor, it is possible to execute a variety of anomaly detection algorithms. The following anomaly detection schemes are currently supported:

- Median absolute deviation
- Grubb's score
- First hour average
- Standard deviation from average
- Standard deviation from moving average
- Mean subtraction cumulation
- Least squares
- Histogram bins
- Kolmogorov-Smirnov test

Data Coalescing. Anomaly detection is performed on the physics in the infrastructure modeling layer solely for attack screening and detection. Attacks that do not cause anomalies are screened as uninteresting attacks. Attacks that have relevant information from each layer are coalesced into an analysis report (Figure 2). Note that the detection of a physics anomaly serves as a temporal point to coalesce attack data within a time or event window.

The coalescing algorithm is as follows:

- If a new anomaly is detected within a time window δ, then the additional data collected to this point is grouped with the anomaly.
- Else, if there are one or more active connections from the Internet to the honeypot over the `eth0` gateway, then all the data is highlighted as being related to the anomaly.
- Else, if no active Internet connections exist at the time of the anomaly, then a search is conducted for the connections spawned by files created by attackers (these could be due to an automated worm or delayed payload).

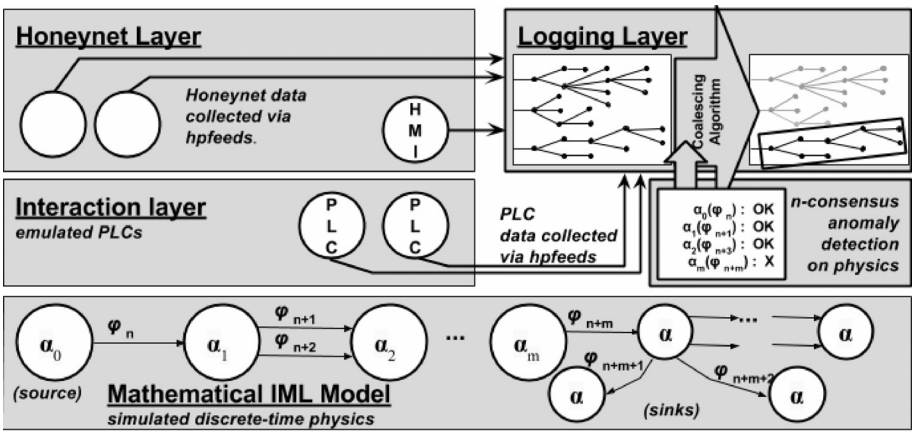

Figure 2. Anomaly detection in the infrastructure modeling layer.

4. GridPot

An electric grid was selected to demonstrate the SCyPH framework due to its mission-critical nature, widespread adoption of insecure automation protocols [9, 13, 17, 19] and the potentially severe consequences of attacks. Cyber attack effects and malware can persist in an electric grid long after the initial attack, exacerbating the fact that the "replacement of large transformers essential to the reliable operation of the grid may require twenty months or longer" [18].

The electric industry is moving to the IEC 61850 standard for automating the control of remote substations in the power grid. IEC 61850 incorporates numerous protocols for substation automation that run over a substation LAN or even a regional WAN. Also, IEC 61850 has communications protocols designed for intelligent electronic devices to communicate with other intelligent electronic devices as well as human-machine interfaces.

This section illustrates the applicability of the SCyPH framework using a sample implementation that simulates the IEC 61850 communications protocols as described in Figure 3. The section discusses the experiment design, infrastructure modeling, IEC 61850 protocol emulation, anomaly detection and logging mechanisms. However, details about the honeynet layer and human-machine interface used in the experiments are not provided for reasons of sensitivity.

4.1 IEC 61850

Most utility communications standards, such as IEC 61850, have largely restricted access to source code or documentation and usually assume domain-specific knowledge, making the subject largely inaccessible to outsiders (including security researchers and software developers). Intelligent electronic

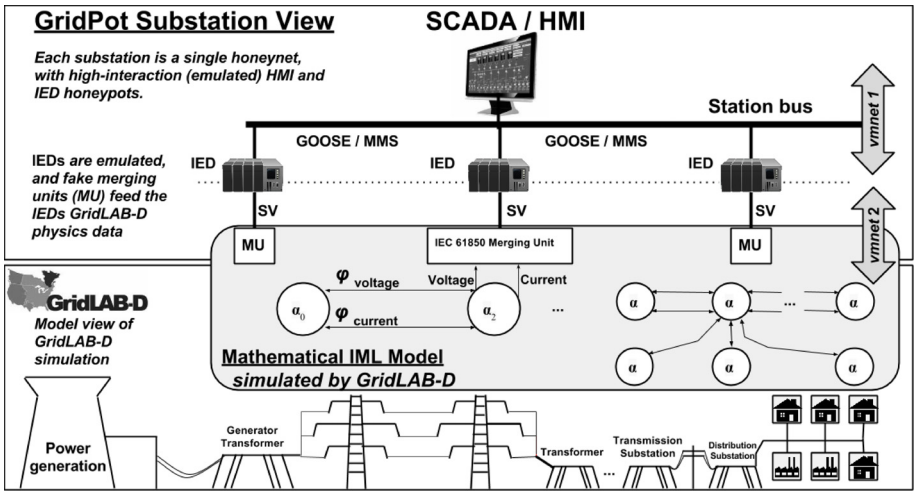

Figure 3. GridPot implementation.

devices are important components in electric power grids; they are embedded microprocessor-based controllers that send and receive data and control commands to/from external sources. These devices are similar to desktop computers (and may even run Linux-based operating systems), but they often contain special digital logic chips for performing domain-specific tasks such as voltage measurement and power regulation. Example intelligent electronic devices are circuit breaker controllers, smart meters, voltage regulators and relay devices [11].

Many intelligent electronic devices are legacy components that were not developed under the IEC 61850 standard; thus, they are often used in conjunction with an "IEC 61850 wrapper" device [11]. Regardless, intelligent electronic devices directly or indirectly interface with a substation (or simply station) bus (as shown in Figure 3) and are controlled by a human-machine interface. The human-machine interface may be remotely accessed by an operator via a secure gateway or a virtual private network. Electric grids comprise numerous distributed substations and the role of IEC 61850 is to implement substation automation. Interested readers are referred to [11] for details about the IEC 61850 standard and to [17] for a discussion of IEC-61850-specific attacks.

4.2 Experiment Design

The experiment simulated the power flow of an electric grid using GridLAB-D from Pacific Northwest National Laboratory, which comprises multiple substations and emulates the IEC 61850 digital communications protocol by extending Conpot as shown in Figure 3. The voltages and currents at major components across the electric grid are simulated in the infrastructure modeling layer. A number of substation honeynets were constructed, each tied in

with the overall infrastructure modeling layer for the entire electric grid. Each substation honeynet comprised emulated intelligent electronic devices that interacted with each other and the human-machine interface over a substation bus.

The goal of the experiment was to capture cyber attack activities against the emulated IEC 61850 protocols and understand how they affect the underlying simulated physical model. The honeynets presented to attackers represented transmission and distributional substations; however, every actor in the grid was simulated all the way upstream to the power generators.

4.3 Infrastructure Modeling

The voltage and current of the power flow between each actor was modeled. The Newton-Raphson power flow solver algorithm was used to solve the power flow during each iteration of the simulation. Figure 3 shows the GridPot implementation, including the architectures of the interaction and infrastructure modeling layers. As shown the figure, the channels are bidirectional according to the physics of the electric grid. A change in the voltage of a middle node can cause an influx in the electric grid and send voltage drops and currents in all directions before stabilizing. Thus, for each actor, a bidirectional voltage channel $\varphi_{voltage}$ and a bidirectional current channel $\varphi_{current}$ were modeled.

Generally, the firing rules for each actor regulate the voltage and current of the power flow and the opening and closing of switches and relays, and also adjust the power flow to react to system feedback from other actors. The firing rule specifications differ from actor to actor.

To support interactions between the infrastructure modeling layer actors and the human-machine interface, the MMS, GOOSE and Modbus protocols in IEC 61850 were implemented as sampling services. The emulation of these protocols requires the engineering and reverse engineering of intelligent electronic device binaries so that the real software logic handles the protocol implementations. MMS was used to provide real-time measurements from the intelligent electronic devices to the human-machine interface. GOOSE communications were emulated for time synchronization, intelligent electronic device configuration and status changes. The emulated GOOSE communications directly influenced the firing rules of the individual actors. This facilitated the simulation and analysis of denial, disruption, destruction and degradation of service attacks against the electric grid by sophisticated adversaries. A sample intelligent electronic device switch based on `rapid61850` may be downloaded from `gridpot.org`. The experimental system will implement additional IEC 61850 protocols in the future.

4.4 Logging and Anomaly Detection

The SCyPH framework employs a library of anomaly detection algorithms supported by the ETSY Skyline Project (`github.com/etsy/skyline`). The algorithms do not require manual fine tuning or static assignment of thresholds.

Each data point in a data series is assessed by all the selected anomaly detectors; if any anomaly is detected, a consensus vote of the detectors determines whether or not an anomaly is indicated. The consensus is configurable and can range from requiring just one detector to requiring all the detectors to agree that a data point is, in fact, anomalous. The convenient simulation data export features of GridLAB-D reduce the training time of the anomaly detectors. Indeed, the SCyPH framework is launched by feeding in months' or years' worth of baseline data into the anomaly detection engine, which takes only seconds to minutes to be processed. However, the training lead time ultimately depends on the grid architecture and configuration of GridLAB-D.

Experiments revealed that the minimum window of baseline data for reliable anomaly detection is in the order of hours, not months or years. Furthermore, incorporating additional factors in the simulation (e.g., weather) are also facilitated by GridLAB-D. However, weather, loads and other simulation factors can affect the false-positive and false-negative alarm rates.

Upon detecting an anomaly, the coalescing algorithm processes the recent data (within the $t - \delta$ time period) from the honeynet, interaction and infrastructure modeling layers. It then attempts to determine the point of origin in the network traffic and/or honeypot files to provide a rough event chain that shows how the cyber-physical anomaly was triggered.

Typically honeypots utilize intrusion detection systems to detect attacks. Attacks that trigger anomalies in the GridLAB-D simulation, but do not raise any intrusion flags, are potentially zero-day attacks that can be highlighted for prioritized analysis.

4.5 Experimental Results

A switching attack [12] against the GOOSE/MMS protocols was executed to demonstrate the analysis capabilities of GridPot. An electric grid based on the IEEE 13-node test feeder model was represented in GridLAB-D. Malware was written to exploit the GOOSE/MMS vulnerabilities presented in [19]; the malware caused the status of an intelligent electronic device switch in a substation to be flipped. Figure 4 demonstrates the attack. Note that the malware is not shared on `gridpot.org` as it can potentially impact any IEC-61850-based intelligent electronic device switch [17].

Details about the human-machine interface software used in the experiment are not presented so that the honeypot is not made trivially fingerprintable by attackers. However, a vulnerability in the default configuration of the human-machine interface was exploited to gain code execution rights. The human-machine interface vendor has been notified about this vulnerability. In any case, the second stage of the experimental attack downloaded the GOOSE/MMS malware and executed it.

The malware was designed to send GOOSE/MMS messages over the appropriate interface to the control network in order to affect the target switch. Figure 5 illustrates the real-time physics impact displayed by the system analytics during the attack. Cyber and control events are represented as vertical lines;

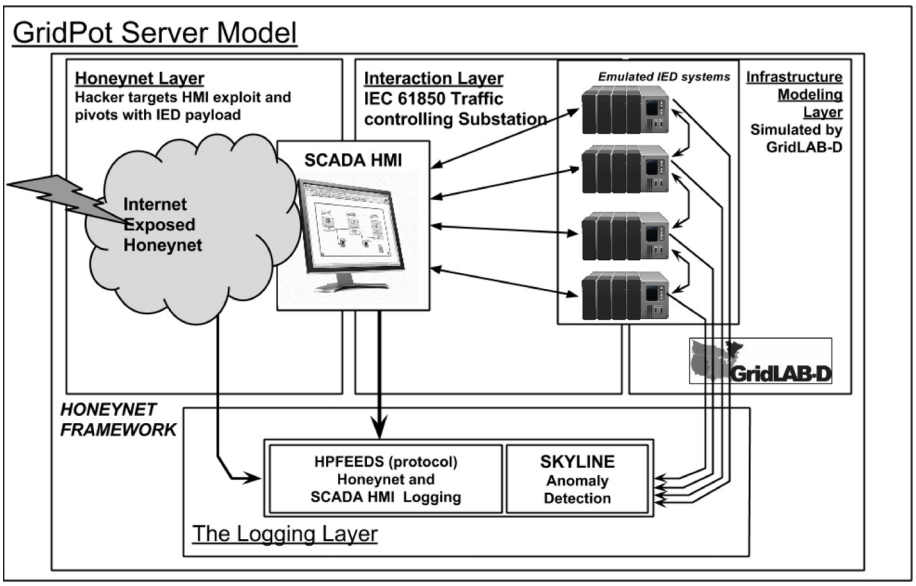

Figure 4. GridPot architecture and experimental attack.

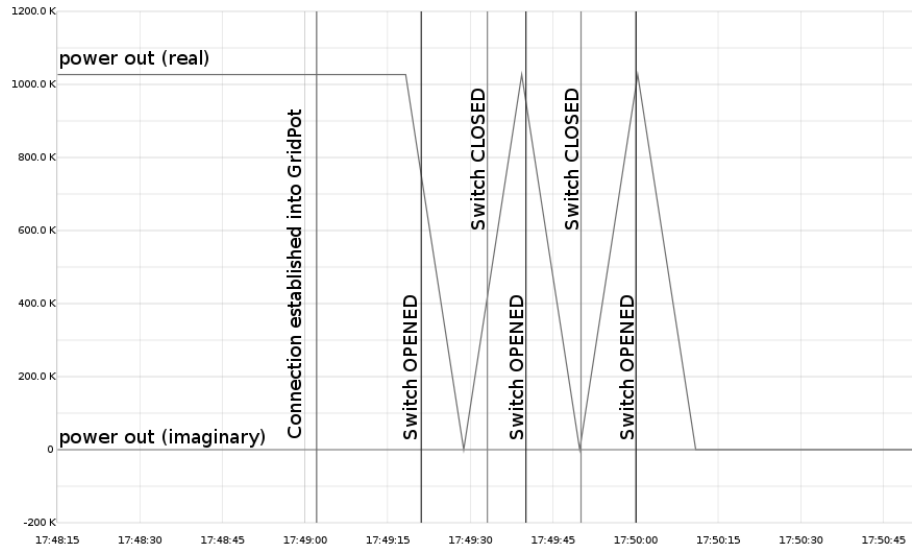

Figure 5. Attack analysis presented by GridPot.

the original image has only been modified by adding text labels for explanatory purposes. The power input and power output parameters were captured for anomaly detection at the switch; however, the physics analysis can be triv-

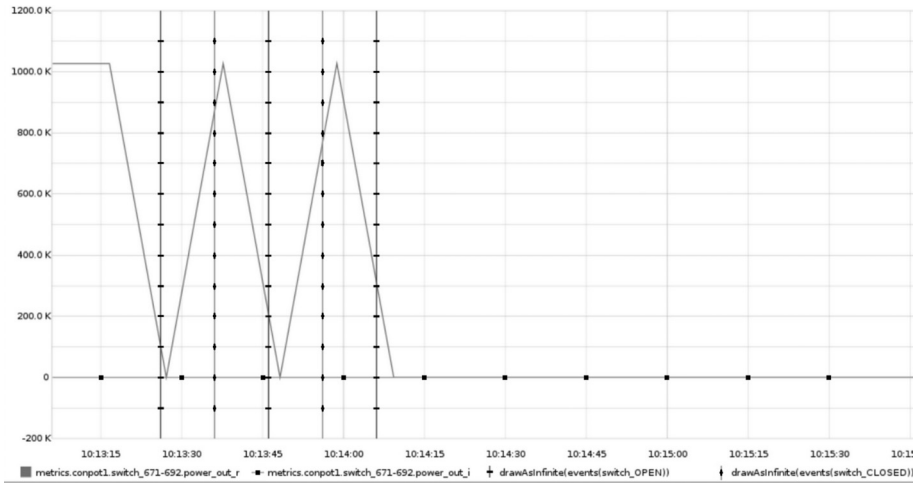

Figure 6. Dynamic analysis capability.

ially extended to any of the numerical parameters of the objects supported by GridLAB-D.

Samples and screenshots of the packet captures of the attack are also not presented because this is equivalent to releasing the malware source code itself. However, this information and other important information are presented to users to support analysis.

New files added to the honeypot are presented to users to enable them to dynamically analyze the malware to confirm the physics impact that was originally recorded. Figure 6 illustrates this feature. Note that malware may be dangerous or impossible to test on real hardware. Interested readers are referred to [12] for details about the physics impact of switching attacks.

5. Conclusions

The SCyPH framework is designed to entice attackers to use high value or zero-day attacks, thereby helping collect novel threat intelligence for cyber-physical systems. In particular, the framework extends honeynet methodologies with integrated physics simulation and anomaly detection utilizing a symbolic data flow model of system physics. This facilitates the screening and coalescence of attack events for analysis, provides attack introspection down to the physics level of a SCADA system and enables forensic replays of attacks. Experimental results demonstrate the effectiveness of the approach.

Future research will augment GridPot to be deployable by MHN and to support forensic replays of honeypot events. Additional intelligent electronic devices and human-machine interfaces will also be emulated to extend the SCyPH framework. Finally, efforts will be undertaken to extend the framework with a sandbox to support malware testing and analysis.

Acknowledgement

This research was partially supported by NSF Grant Nos. DUE 1241525, CNS 1347113 and DGE 1538850.

References

[1] R. Bodenheim, J. Butts, S. Dunlap and B. Mullins, Evaluation of the ability of the Shodan search engine to identify Internet-facing industrial control devices, *International Journal of Critical Infrastructure Protection*, vol. 7(2), pp. 114–123, 2014.

[2] E. Byres, Project SHINE: 1,000,000 Internet-Connected SCADA and ICS Systems and Counting, Tofino Security, Lantzville, Canada, 2013.

[3] Conpot Development Team, Conpot: ICS/SCADA Honeypot (`conpot.org`), 2013.

[4] Digital Bond, SCADA Honeynet, Sunrise, Florida (`www.digitalbond.com/tools/scada-honeynet`), 2015.

[5] M. Geilen and T. Basten, Requirements on the execution of Kahn process networks, *Proceedings of the Twelfth European Conference on Programming*, pp. 319–334, 2003.

[6] K. Gorzelak, T. Grudziecki, P. Jacewicz, P. Jaroszewski, L. Juszczyk and P. Kijewski, Proactive Detection of Network Security Incidents, European Union Agency for Network and Information Security, Heraklion, Greece, 2011.

[7] N. Hadjsaid, C. Tranchita, B. Rozel, M. Viziteu and R. Caire, Modeling cyber and physical interdependencies – Application in ICT and power grids, *Proceedings of the IEEE Power Engineering Society Power Systems Conference and Exposition*, 2009.

[8] G. Kahn, The semantics of a simple language for parallel programming, *Proceedings of the IFIP Congress*, pp. 471–475, 1974.

[9] E. Knapp and J. Langill, *Industrial Network Security: Securing Critical Infrastructure Networks for Smart Grid, SCADA and Other Industrial Control Systems*, Syngress, Waltham, Massachusetts, 2015.

[10] E. Lee and T. Parks, Dataflow process networks, *Proceedings of the IEEE*, vol. 83(5), pp. 773–801, 1995.

[11] Y. Liang and R. Campbell, Understanding and Simulating the IEC 61850 Standard, Technical Report UIUCDCS-R-2008-2967, Department of Computer Science, University of Illinois at Urbana-Champaign, Urbana, Illinois, 2008.

[12] S. Liu, S. Mashayekh, D. Kundur, T. Zourntos and K. Butler-Purry, A framework for modeling cyber-physical switching attacks in the smart grid, *IEEE Transactions on Emerging Topics in Computing*, vol. 1(2), pp. 273–285, 2014.

[13] P. Maynard, K. McLaughlin and B. Haberler, Towards understanding man-in-the-middle attacks on IEC 60870-5-104 SCADA networks, *Proceedings of the Second International Symposium on ICS and SCADA Cyber Security Research*, pp. 30–42, 2014.

[14] J. McClean, C. Stull, C. Farrar and D. Mascarenas, A preliminary cyber-physical security assessment of the Robot Operating System (ROS), *Proceedings of the SPIE, Unmanned Systems Technology XV*, vol. 8741, 2013.

[15] Open Web Application Security Project (OWASP), OWASP Top 10 – 2013: The Ten Most Critical Web Application Security Risks (`www.owasp.org/index.php/Top_10_2013-Top_10`), 2013.

[16] V. Pothamsetty and M. Franz, SCADA HoneyNet Project: Building Honeypots for Industrial Networks, Critical Infrastructure Assurance Group, Cisco Systems, San Jose, California (`scadahoneynet.sourceforge.net`), 2005.

[17] M. Rashid, S. Yussof, Y. Yusoff and R. Ismail, A review of security attacks on IEC 61850 substation automation system networks, *Proceedings of the International Conference on Information Technology and Multimedia*, pp. 5–10, 2014.

[18] Staff of E. Markey and H. Waxman, Electric Grid Vulnerability: Industry Responses Reveal Security Gaps, U.S. House of Representatives, Washington, DC, 2013.

[19] A. Timorin, SCADA deep inside, presented at the *Balkan Computer Congress*, 2014.

[20] S. Wade, SCADA Honeynets: The Attractiveness of Honeypots as Critical Infrastructure Security Tools for the Detection and Analysis of Advanced Threats, M.S. Thesis, Department of Electrical and Computer Engineering, Iowa State University, Ames, Iowa, 2011.

[21] K. Wilhoit, Who's Really Attacking Your ICS Equipment? Research Paper, Trend Micro, Irving, Texas, 2013.