# An isolated virtual cluster for SCADA network security research

Antoine Lemay
École Polytechnique de Montréal
2500, Chemin de Polytechnique
Montreal, Qc, CA
H3T1J4
*antoine.lemay@polymtl.ca*

José Fernandez
École Polytechnique de Montréal
2500, Chemin de Polytechnique
Montreal, Qc, CA
H3T1J4
*jose.fernandez@polymtl.ca*

Scott Knight
Royal Military College of Canada
13 General Crerar Cres
Kingston, ON, CA
K7K 7B4
*knight-s@rmc.ca*

**Research aimed at securing the SCADA and ICS networks has taken off in the wake of Stuxnet. Unfortunately, it is difficult for researchers to fully capture the integration between cyber and physical components that is intrinsic to these systems. To enable researchers to perform network security experiments while taking into account the physical component of ICS networks, we propose the use of the ICS sandbox. The ICS sandbox uses the proven virtualized cluster approach to emulate SCADA networks with high fidelity. The virtualized cluster is interfaced with an electrical power flow simulator to integrate the physical component of an ICS network controlling electrical grid critical infrastructure without imposing scale constraints. Parts of the proposed sandbox were validated in a training session offered to industry professionals where a satisfaction survey indicated that hands-on session with the ICS sandbox provided significant training value to the participants that could not have been obtained in traditional training.**

*Experimental research, Security Training, SCADA security, power grid security.*

## 1. INTRODUCTION

Industrial control systems, or ICS, allow operators to remotely operate a number of industrial systems from nuclear reactors, to water treatment plants and the electrical power grid. This is largely achieved through the use of the Supervisory Control and Data Acquisition (SCADA) protocols. While the integration of SCADA systems with traditional IP networks has brought flexibility and economy scale to industrial control and automation, it also allows attackers to leverage their knowledge on vulnerabilities and offensive techniques on such networks to attack the ICS that use them. The discovery of Stuxnet in 2009 Falliere et al. (2011) underlined this point. Consequently, many researchers have taken an interest in the security of SCADA and ICS networks. Nonetheless, the interaction between cyber components and physical components makes the study of ICS security more complex than that of traditional IT networks.

One of the redeeming properties of SCADA networks, from a security point of view at least, is that they are well structured and have well defined and constrained protocols. Using this property, Hadeli et al. (2009) showed that it is possible to automatically construct firewall and IDS rules from configuration files of SCADA equipment by leveraging determinism in ICS traffic. Unfortunately, it is difficult to validate the usefulness of this kind of research for ICS networks because there is little public domain information about the way these networks truly operate. For example, how can we be certain that these IDS rules detect the attacks that really matter? Or, will such counter-measures interfere with the operation of critical networks? To answer these questions, we need an experimental setup that enables us to model the interaction between SCADA networks and the physical systems they control. To this aim, we present an ICS sandbox that allows us to perform large scale cyber security experiments integrating physical components. We achieve this goal by interfacing an isolated cluster designed for large scale security experimentation (i.e. a cyber range) with a computerized model of the physical system, in our case a simulator that models the power flows through the electric grid. This allows us to perform cyber security experiments on a system that fully replicates the behavior of a real ICS system controlling the power grid. For example, since ICS traffic is determinist in nature, anomaly-based intrusion detection should be particularly effective. We propose to test this kind of hypothesis on the ICS sandbox.

In this paper, we start by presenting a short primer on SCADA networks for electrical grids. We then

review various experimental approaches to test ICS security. We present the design and implementation of the ICS sandbox, and then the results of the use of the ICS sandbox in a training session with industry practitioners. Finally, we describe the limitations of the ICS Sandbox and discuss future research work employing it and potential improvements.

## 2. SCADA PRIMER

In order to present the ICS Sandbox, a brief overview of ICS networks is required. The focus will be SCADA networks employed in the control of electrical power infrastructure, as this is the first kind of infrastructure implemented in the ICS sandbox.

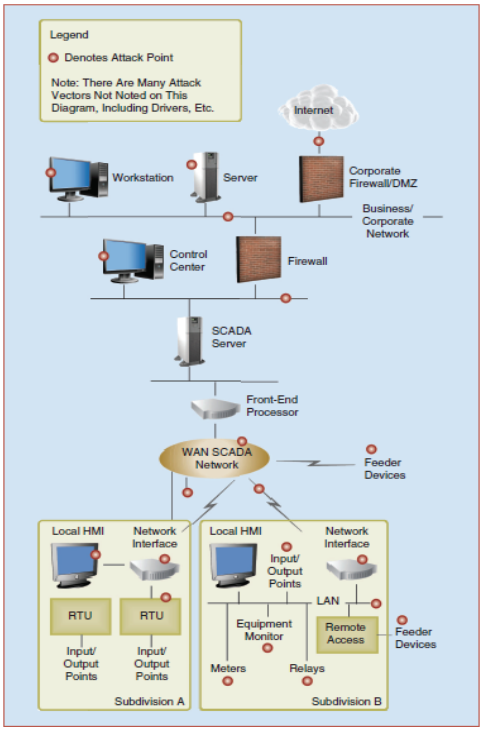


***Figure 1:*** *SCADA architecture for the power grid Hull et al. (2012)*

The purpose of the electrical transport network is to deliver electricity from the source (either the power generation plant or another transport network) to the distribution center. High voltage lines that minimize power loss are typically used. Substations that convert power from generation voltage to transport voltage and from transport voltage to distribution voltage are the primary points of control of the electric grid. Substations that perform path switching if power needs to be rerouted through another line are also control points. In terms of industrial control, the health of the power grid has to be monitored by placing programmable logic controllers (PLC) that can measure important indicators (e.g. voltage, current, power and so on), which are called measurement points, or PLCs that can specify the state of a piece of equipment (breaker on/off, transformer voltage setting, etc.), which are called control points.

These PLCs require physical interfaces with power equipment deployed in substations. It is customary to aggregate the data from all the PLCs in the same physical site (substation) to a remote terminal unit (RTU). The RTU allows local operators to read the values of measurement points and operate control points through a local human-machine interface (HMI) station. The RTU receives the communication from the PLC and may convert the communication to a protocol that is suitable for long-distance transmission, such as TCP/IP, towards a central controller.

The master terminal unit (MTU) is connected to all RTUs within a region and aggregates the data and provides control to all these sites. As such, the MTU is typically physically located in the control centre of the electrical grid operator. HMI consoles for human operators are also typically collocated on the same network as the MTU. In most cases, this operational network is separated from the operator's administrative network by a firewall. However, for cost saving reasons, some of the operator stations might reside on both office and production networks to allow operators to read email and access Internet on the same workstation. A historian application, i.e. a database that records all historical values of measurement points, might also reside on this same network. This historian will typically require some communication with the office network in order for office workers to perform data analytics or to support other business functions such as billing.

Overall, the SCADA network for the control of a power grid is a logical tree network with the MTU at the root of the tree. The MTU is connected to RTUs, who can be connected to PLCs or intermediate RTUs. Finally, the PLCs are connected to either control points or measurement points. Figure 1 illustrates a typical SCADA architecture. In this figure, each subdivision represents a physical location such as a power substation. Each subdivision hosts a local HMI and one RTU connected to the network. On the other side of the SCADA WAN, the MTU sits in the control centre to administer an entire region.

## 3. EXPERIMENTAL APPROACHES IN ICS SECURITY

ICS security has become a popular topic since the public discovery of Stuxnet. A number of approaches have been proposed to provide an experimental framework for this research. This section provides an overview of approaches used for SCADA and ICS security research.

### 3.1. Physical Deployment

One of the more realistic approaches to do research on SCADA and ICS systems is to actually deploy a system and perform experiments on that system. The National SCADA Test Bed U.S. Department of Energy (2009), with its seven substations and 61 miles of high voltage transmission lines, is an example of this kind of implementation. This approach allows researchers to create experiments that have a high resemblance to real world systems, because it is using a full implementation of both the physical component and the software component. However, this approach suffers from a number of drawbacks for security research.

The first drawback is that deploying a real system requires significant investment, both in terms of capital and in terms of manpower. In terms of capital investment, let us consider Hydro Quebec's annual report Hydro-Québec (2012). The cost of replacement of the software for the management and analysis of the transport network is budgeted at 32 million Canadian dollars. This does not include any physical components (such as power lines and substations). This would suggest that the cost of standing up an at-scale laboratory is likely to cost tens of millions of dollars. In addition, SCADA equipment usually needs to be manually configured, requiring specialized knowledge to configure. This increases the manpower cost to stand up this kind of laboratory. The bulkiness of physical equipment also creates a substantive lab space cost, especially in an academic setting.

A second problem is that of "decontamination" and reconfiguration of the experimental setup. Because real equipment is used, any modification to the original configuration, e.g. as a result of performing an attack on one of the machines, needs to be undone manually. This increases the costs of operating the test bed, increases the downtime of the test bed and may create unpredictable states if the decontamination is not thorough. This causes significant drawbacks for the repeatability of experiments.

### 3.2. SANS Cyber city

A possible compromise to reduce lab space use is to limit the scope of both the network studied and the physical equipment required. The SANS Institute, with their Cyber City project O'Harrow (2012), followed this path. The computer network of a small town was reproduced on virtual machines, including the user profiles and actions, in order to be able to train experts in attacking and defending networks. This training includes SCADA systems that might be operated in a small city, i.e. the water treatment and transport systems. The SCADA components are connected to a small scale model town in order for the students to be able to observe the physical consequences of cyber attacks. For example, an attacker might send a false command and switch a railroad track, causing two model trains to crash together.

By limiting the scale to a small town, it is possible to create an environment interesting enough for students, while keeping it manageable, both in terms of manpower and real estate. The use of virtual machines allows for fast resets to initial configurations making decontamination straightforward. The physical consequences of attacks on ICS networks can also be very plainly observed. This provides a good environment for education. Unfortunately, the Cyber City model is limited in terms of possible research. As with physical test beds, Cyber City requires physical components, making it harder to do testing on configurations other than the default configuration. Also, addressing the problem of scale by scoping it to a small city prohibits any research done on problems with a larger scale.

### 3.3. Software only

Another option for ICS security research is to use demo versions of SCADA software. A number of ICS vendors offer trial versions of their industrial control software. This software is usually a HMI application (software designed to allow human operators to interact remotely with industrial control equipment) with some missing functionalities such as a limited number of days the software can be used or a limited number of machines the software can interact with. This allows a researcher to observe communications that are properly formatted with minimal effort. As such, it is often used for research focused on protocol security (ex. Kobayashi et al. (2009)).

The major drawback of this approach is the lack of physical effects. While it is possible to hook a trial version to a couple of actual machines and turns lights on and off, it is not practical to use this setup to measure realistic physical effects. Unless great care is put into designing the physical network connected to the SCADA system, it is unlikely that the physical network will provide a realistic feedback to the SCADA system. For example, in a real system, turning off a breaker will shut down the power to the line making the sensor register a drop in voltage and possibly increase the load on power generators. In that sense, a network packet may very well have a scope of influence far greater than is possible to model with trial versions of HMI software.

### 3.4. Simulation

To solve the problems of scale with physical implementations, it is possible to use simulation. A

simulation approach uses a model that is an approximation of reality to approximate the results of whatever inputs a user provides the system. Davis et al. (2006) introduced a SCADA test bed that provides a user with an electrical power system HMI which is plugged into a computer network simulator. So, when a user sends a command to turn off a breaker for example, this network simulator reproduces the network packet and its delivery to the destination. Once it reaches its destination, the simulation software generates a real packet with a virtual IP address and sends it to the PowerWorld electrical simulator to see what effects the command had on the power flow. Power World can send packets back through the simulated network and ultimately be displayed on the HMI. In that sense, the physical effects of cyber attacks on the power grid can be observed on the HMI from the results of the power flow calculations. Unfortunately, the approach suffers from some drawbacks.

The first drawback is in terms of the validity of the model and the soundness of the measurements. Because the simulation is not using real equipment, but a mathematical model of the equipment, there may be a significant difference between results observed in a simulation and an identical real world deployment. It is possible to validate the simulation models for both the network side and the power flow side to make sure they behave in a way similar to real world networks. However, security research has a tendency to deal with extreme or edge cases for which a model, even if it has been validated under normal operating conditions may react differently than a real implementation.

Another inconvenience is that the configuration required and data produced are in formats that are not directly portable. For example, the RINSE network simulator used in Davis et al. (2006) is focused on coarse traffic. So, data on the packet level is not always available. In that sense, results are less portable than if a more conventional PCAP format was used.

### 3.5. Emulation

If a physical implementation is too expensive and a simulation does not quite allow us to represent a real network with the fidelity we would want to, we may consider emulation, i.e. a system that duplicates exactly rather than approximate the behaviour of a real system. For security research, the DETER test bed Braden et al. (2006) and the Emulab network test bed with large-scale virtualization Hibler et al. (2008) are two examples of medium to large scale network emulation environments that could be used for ICS security research and training. In both of these cases, an environment very similar to a real deployment can be programmatically deployed in the test bed. Malware and attacks can then be tested without impacting real systems. If dedicated virtual machines can be used, an approach similar to the isolated virtual clusters from École Polytechnique de Montréal's SecSI lab Calvet et al. (2010) can also be successful.

Past experience showed that an emulation approach can address a number of problems such as: containment of experiments, isolation for concurrent experiment interference, confidentiality and integrity of configuration and results, and the prevention of misuse of the test bed. Also, because deployment and experiments are run programmatically, it is easy to perform decontamination and reconfiguration efficiently. In the isolated virtual cluster architecture, decontamination is even more straightforward using VMWare snapshots. However, these approaches have a major drawback –the modelling of physical effects. Because all three test beds described above were designed to emulate cyber attacks, they only emulate electronic components. In that sense, it is even harder to model the physical effects than with the use of trial software. Usually, in the operating environment of emulation clusters, it is physically impossible to install the custom I/O cards that can create the analogue signals required by many PLCs or ICS machines.

## 4. THE ICS SANDBOX

Based on our study of existing approaches, we found that physical implementation-based approaches are too costly, but simulations cannot fully capture the interaction between the physical and computer system. Emulation approaches seem to provide the correct balance between realism and feasibility. However, they struggle to integrate the physical aspect. Our approach strives to find a way to integrate the ICS physical component to existing emulation infrastructure in order to create an ICS sandbox.

The goal of this ICS sandbox would be to study the effects of network attacks, such as denial of service, falsification or injection of data, malware infection and so on, on both the network infrastructure of SCADA networks and on the power grid. In other words, the goal is not to find and test vulnerabilities in specific equipments, but rather to perform impact assessments of known attacks or to evaluate the effectiveness of network defences to prevent these attacks. This distinguishes us from other works in emulation, such as Davis et al. (2006), which focuses on the behaviour of SCADA equipment and do not offer the granularity of network traffic necessary to perform network security research. In that sense, our approach is, as far as we know, the only methodology available for high risk network

security experiments for SCADA systems that take into account the physical side of the equation.

### 4.1. Scoping

The first design step is to scope the project in order to deduce requirements. The focus our ICS sandbox is on network security. In that sense, only the elements relevant to network security are required to be fully emulated. Our focus was to make sure the network traffic that can be observed resembles as closely as possible that of a real-world implementation. In addition, any system component directly interfacing with the network, i.e. clients and servers, need to be as close as possible to real-world implementations. Any other elements requirements in terms of fidelity are less severe.

In terms of SCADA systems, we require the actual network to have the highest degree of fidelity, MTU and RTU machines to have good level of fidelity and the HMI, PLCs and the actual physical system require less fidelity. In fact, for all intents and purposes, the physical system can be considered a black box where the inputs are values of control points (ON/OFF values for breakers and voltage or current values for set point controls). To achieve this, we chose an architecture such that in the core, where fidelity requirements are high, an emulation approach similar to DETER Braden et al. (2006) is used and in the edge, where less fidelity is required, a simulation approach similar to PowerWorld Davis et al. (2006) is employed.

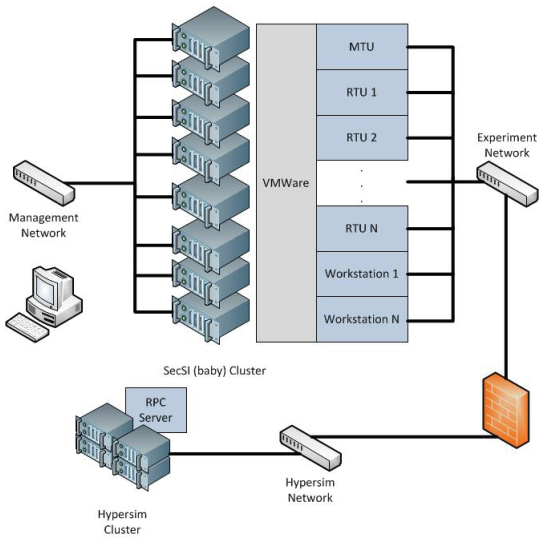


***Figure 2:*** *Architecture*

### 4.2. Implementation

For the core of the network, we want a suitable platform for emulation where we can run actual SCADA software and perform real-world attacks. We decided to adopt the test bed for high risk security experimentation and training proposed by Calvet et al. (2010).

Our infrastructure employs a number of IBM Blade servers running VMware software for virtualization. A management network allows the deployment of experimental configuration (deployment of machines, starting/stopping the VMs, setting IP addresses, etc.) through the xCAT scripting language as described in Calvet et al. (2010). Figure 2 shows the architecture of the ICS sandbox. The entire SCADA system is emulated on virtual machines running on the SecSI cluster. The SCADA system is then connected through RPC requests to the electrical power flow simulator. Each section of the infrastructure will be covered in detail in the following paragraphs.

The experiment network itself is a physical Ethernet network. Other network technologies could eventually be used to experiment with other types of interconnection technologies. Managed switches are currently used because the network topologies are simple, but virtual switches that could be configured programmatically by xCAT could be used. Port mirroring is used to capture the network traffic so great care has to be applied in making sure experimental traffic makes it on the wire. Concretely, this means that communication between multiple virtual machines on the same server should not be using the virtual networking provided by VMware unless it was specifically designed in the experiment plan.

The experiment network connects to the electrical power flow simulator hosted on a separate computing cluster. A firewall separates the experiment network from the computing cluster for a number of reasons. The first reason is to prevent any traffic from the computing cluster to interfere with the experiment. At the same time, we do not want malicious software used in our experiments to contaminate the computing cluster. Thus, the firewall prevents all traffic from getting in and only allows the RPC requests to the Hypersim servers to get through. Soundness of network captures is impacted by the communication with the electrical simulator because the out-of-band communication occurs through the experiment network. However, this can be addressed by filtering out this specific traffic either at the capture or post-hoc in the PCAP files.

Running on this physical infrastructure is a SCADA software designed to control an electrical grid. For this purposed, we used a commercial SCADA product obtained through special research funding. The MTU and historian were hosted on a Red-Hat Enterprise Linux machine running the DNP3 version of the GENe SCADA software from General Electric (2011). The DNP3 version was chosen because of the popularity of this communication protocol for electrical grid ICS. The physical server provided by the software vendor was backed up and restored on a virtual machine.

This severely impacted its performance, but it provided the ability to make snapshots of the machine for quick restoration. This trade-off proved critical for fast re-initialization of the experimental setup in a training setting and for saving development time.

The second piece of commercial software is the RTU emulator. The RTU Load Simulator (RLS) is a special-purpose RTU software designed to perform load simulation for acceptance testing of GENe software. The RLS software is run on a virtualized Windows XP machine. Because each RLS typically represents an electrical substation, the RLS VM is cloned multiple times to achieve the desired scale. With the RLS machines (playing the role of RTUs) and the MTU machine, we have a fully functioning commercial grade implementation of a SCADA network. We can also add additional machines, such as operator workstations to enrich the network model. This implementation generates perfectly formed traffic on the network. The implementation also responds exactly like a real system to cyber attacks. However, we have to address the physical component feedback. Because we consider the physical component to be a black box, we need to provide the inputs from the SCADA system (i.e. the values of the command points) and integrate the outputs (i.e. the value of the measurement points).

The RLS behaves exactly like a GE RTU in terms of network communications, but does not interface with actual PLCs. Instead, each RLS has a database residing in RAM with values for each PLC. A command-line interface (CLI) was built by Rosset (2012) to interact programmatically with the values. This was achieved by injecting a DLL into the RLS program memory to be able to read and modify values directly into RAM. A scheduled task on the RLS machine runs a script periodically to extract the values of the control points (through the CLI get method), and feed them to the power flow simulator and retrieve the results. The results are then fed back to the RTU emulator through the CLI set method. The frequency of execution of this script depends on the polling rate from the MTU and the convergence time of the electrical network simulator. The script needs to run faster than the polling to present accurate measurement point values, but must allow enough time for the simulator to converge.

Because of the bulkiness of physical equipment and because our scope does not require detailed granular fidelity of the electrical side, we chose not to emulate physical equipment. However, we still need a system that could provide us with the physical feedback a real system would present. We chose to use a power flow simulation to provide us with the physical feedback. The simulator we used is the Hypersim simulator which was developed by Hydro Quebec and is currently commercialized by Opal-RT. The simulator was the subject of peer-review validation by the scientific community Paré et al. (2003). This choice was further motivated by the ease of access to this software (was being used by other research labs in our institution), the ability to simulate networks at scale, the near real-time operating mode (convergence in the order of a couple of seconds), and the availability of an API accessible through RPC remote procedure calls.

The API allows a user to write C code to perform actions in the simulator, such as obtaining the instantaneous value for a number of sensors or changing the configuration of the electrical network. Once compiled, these C programs can be called to produce properly formed RPC requests to the RPC server running on the Hypersim machines.

All of these characteristics make Hypersim suitable as a black box equivalent to a physical electrical network in the context of our ICS sandbox. This black box design also makes it easier to change the simulator if another power flow simulator is used instead or if a different physical system is modelled. For example, if we wanted to model an oil and gas pipeline SCADA system instead of the electric grid, we could exchange the physical system black box.

## 5. EXPERIENCE WITH THE ICS SANDBOX

The ICS sandbox had an opportunity to prove itself in a training offered to industry practitioners. Due to logistical constraints, it was not possible to move or access remotely the Hypersim simulator. However, the acknowledgement of usefulness from operators of real SCADA networks can provide some validation to significant parts of our approach.

### 5.1. Description of the training

The training was organized by [a government agency] Natural Resources Canada (NRCan) which is the designated lead for energy infrastructure protection, including cyber threats. The training took place at the Nation Energy Infrastructure Test Centre (NEITC) located in Ottawa. The ICS sandbox was moved to that location for the duration of the training. A previous 1-day demonstration training on the ICS sandbox had been made to industry leaders in order to get their feedback on the type of training that would be most valuable to their staff. The topic of incident handling in an ICS environment was identified as being the most important topic to cover. Consequently, an introductory training on cyber incident handling in an ICS environment with a focus on hands-on interaction was prepared and delivered.

The training was followed by 28 industry practitioners. The level of skill varied. Trainees

included system administrators, SCADA system engineers, security experts, security policy practitioners, security managers, compliance consultants and penetration testers. All were working in industry, either for energy providers or for consulting firms working for them. The length of training was two and a half days. The ICS sandbox was used in four 90-min tracks and in a 3-hour training exercise on the last day. For the purpose of the training, additional machines representing corporate infrastructure were added to the ICS sandbox. The network infrastructure is presented in Figure 3.

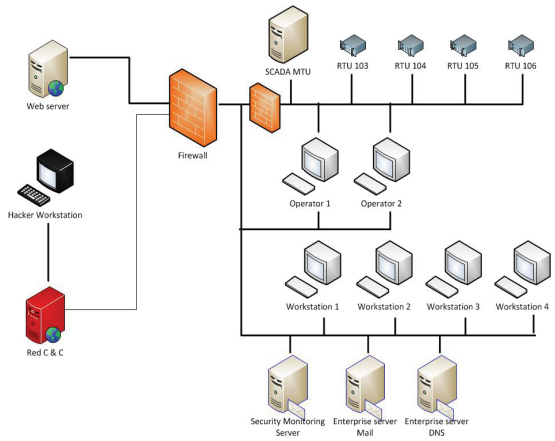


***Figure 3:*** *Training network infrastructure*

In effect, we had the ICS sandbox with one MTU and 4 RTUs connected to 4 PLCs each. We also had two "dual-homed" operator workstations (with one network card on the office network and another on the SCADA network) configured as HMI stations and four Windows XP user workstations. Three servers provided enterprise services including mail, Domain Name Service (DNS) and security monitoring (Snort IDS). A small representation of the Internet containing one web server, one hacker workstation (Backtrack 5 R3) and a Web server for malware command and control was also included. A single OpenBSD machine was doing the role of router and firewall. A managed switched with VLAN support provided the layer 2 connectivity. All the machines were virtualized for easy restoration. The MTU and IDS were each running on a dedicated server and everything else was run over 3 desktop PCs with multiple network cards. A dedicated control network to access VMware application on each machine is not shown in the figure. The power flow simulator could not be physically moved and was not integrated in the training.

The first track was a demonstration of how a persistent attacker would infect a machine in the office network then pivot and worm his way to the SCADA network through dual-homed machines. The second track was a demonstration of network components and counter-measures used in ICS, such as, looking at IDS and firewall logs and performing containment with the firewall. The third and fourth track were training on Wireshark and Sysinternal tools where a copy of one of the workstations was attacked by a drive-by download (automatically generated by the Social Engineering Toolkit (SET) Metasploit plug-in available on Backtrack) and numerous post exploitation actions were taken. The traffic from this attack had been recorded and provided on the virtual image distributed to the students. The last-day exercise required the students to perform the full PICERL (Preparation, Identification, Containment, Eradication, Recovery, and Lessons Learned) incident response steps on the network shown in Figure 3, where we had unleashed a custom-made program than emulated a worm. The initial infection was via USB key and the worm then connected to the red C&C server and propagated over the network by brute forcing weak Windows share passwords.

### 5.2. Evaluation and lessons learned

Trainees were asked by the NEITC to fill out a questionnaire to help guide future training. In particular, they were asked to rate the course and the various sessions. They could give a grade of "adequate", "good" or "very good". Overall, the training was highly rated with 45% "very good", 55% "good", and 0% "adequate". In addition, all participants unanimously responded that they would recommend this course to a colleague.

Of the four sessions using the ICS Sandbox, two of them were very highly rated by the trainees. The SysInternals training track received 56% "very good" ratings and 33% of "good" ratings, the APT demo track got 40% "very good" and 50% "good". Participants were also asked which session they enjoyed the most. The most popular session ICS-related session was the APT demo session (20%), followed by SysInternals tool workshop (16%) and the Wireshark workshop (13%). The sessions with the least amount of hands-on training finished last. This data seems to suggest that the ICS sandbox provided value to the trainees. It seems clear that the students preferred hands-on exercise to lectures. The use of the ICS sandbox to generate materials for the exercise helped frame the hands-on in the context of industry practitioners

Additional conclusions can be taken from the observation of the trainees. During the hands-on sessions, a small majority of the students seemed to have good working knowledge of the tools covered in the hands-on sessions, but were interested in the exercise nonetheless, because they did not know that their knowledge of the tool could be relevant in the context of incidence response in ICS. For example, they knew Wireshark can be used to examine network traffic, but they did not know what traffic related to an incident would look like. By being able to observe

artefacts of real attacks, something they cannot normally do on their production network, trainees managed to learn when to use the tools. This proved more relevant than how to use the tools for many students with prior knowledge. From that perspective, the ability of the ICS sandbox to perform and observe real attacks and provide before/after pictures of infected systems, probably proved to be a key factor in achieving the high satisfaction results we observed across a wide range of attendee skill level.

## 6. LIMITATIONS AND FUTURE WORK

While our approach has the advantage of being able to reproduce the physical effects without imposing a significant burden in terms of lab space and budget, a number of challenges still remain.

The first challenge is that of repeatability of experimentation. In most situations, we want to be able to repeat an experiment a number of times to prove the statistical validity of the results for the independent variable. We also want to study the impact of model simplifications by analyzing the sensitivity of the results to variation in control variables, such as was demonstrated in Calvet et al. (2010). In practice, most of the SCADA components still need to be configured manually. In particular, the MTU asset database, which is used to determine which equipment should be polled, requires extensive manual configuration. HMI visualizations screens also need to be crafted by hand if a human is expected to look at them. While using VMware snapshots for sterilizing the environment makes repeatability for independent variables easy, repeatability for control variables would require modifying the SCADA configuration. Using the xCAT tool, it is possible to craft a number of experimental configurations and run them sequentially for repeatability. However, the production of each of those experimental configurations is very time consuming if it cannot be done programmatically.

Another important challenge is the presence of a synchronization problem, caused by the choice to run scripts to update the power flow simulation values and measurement point values runs on the RLS machines at regular intervals. If a control point value changes between those intervals, for example as a result of a command sent by an operator to trip a breaker, there will be a delay between the change in the control point's value and the electrical network effects. For drastic changes in values, this can have an impact on the soundness of the DNP3 network traffic because the DNP3 protocol allows for traffic initiated outside of polling sequences by the slaves to report outages. This could also create inconsistencies if a command is sent within the convergence time of the power flow simulator from a polling request from the MTU. A full study of the impact of the choice of discrete time rather than discrete event simulation would be required to evaluate the impact of the design decision. The synchronization problems can also become more significant when a change affects the value of multiple points across a number of RTUs which may not all update at the same time.

A final challenge with our infrastructure is the availability of standard models to validate this emulation approach and eventually proposed security solutions. While there exist some toy models for electrical networks, computer networks and SCADA topologies (such as that of Figure 1), there are no models that integrate all three aspects. For example, while standard benchmark models exists for power grid simulation (such as those proposed by the IEEE), these models do not describe the corresponding SCADA infrastructure (i.e. the placement of measurement and control points). The physical SCADA test beds have yet to produce data sets (such as traffic captures on the network component) that could be used to validate our ICS sandbox model. Packet captures from live networks could also be used, but unfortunately critical infrastructure operators are typically reluctant to provide the information, due to confidentiality concerns. However, this problem is common to all ICS security research.

## 7. CONCLUSION

Even though more work is required to allow the ICS sandbox to fully reproduce a full SCADA network, the hybrid approach of emulation and simulation allows us to run experiments and training on SCADA network security. In addition, this hybrid approach enables the sandbox to achieve a high degree of fidelity in the network components, where our research resides, and trade off fidelity for scalability for the physical components. This distinguishes us from other works in this space because we can provide the granularity of network traffic necessary to perform network security research while taking into account the physical side of the equation and makes us the only methodology available for high risk network security experiments on SCADA systems and is a significant contribution to the SCADA security community.

The usefulness of the ICS sandbox approach was validated by a training session on ICS incident response to cyber incidents given to professionals from the energy sector. The response from trainees showed that hands-on modules involving the ICS sandbox were amongst the most highly rated, proving that the ICS sandbox was perceived as adding significant value to the training.

The use of the ICS sandbox in an experimental research context is the next logical step. The resulting data sets have the potential to be great assets to the research community that currently suffers from a lack of publicly available sources. Other future work includes addressing some of the current technical limitations of the ICS sandbox. In addition, the adaptation of the ICS sandbox to other SCADA environments such as oil and gas, water supply or manufacturing would enable researchers to study and address issue specific to those systems.

## 8. ACKNOWLEDGEMENTS

This research was funded by NSERC through the ISSNet network and through the Research tools and Instrument program. We would also like to thank NRCan and the NEITC for organizing the ICS incident response training and for providing the questionnaire

## 9. REFERENCES

Braden, R. et al. (2006) Experience with DETER: A testbed for security research. In: 2nd International Conference on Testbeds and Research Infrastructures for the Development of Networks and Communities, 2006. TRIDENTCOM 2006, Barcelona, Spain, 1–10.

Calvet, J. et al. (2010) Isolated virtualised clusters: Testbeds for high-risk security experimentation and training. In: Proceedings of the 3rd International Conference on Cyber Security Experimentation and Test. CSET 2010. Berkeley, CA, USA, 1–8.

Calvet, J. et al. (2010) The case for in-the-lab botnet experimentation: Creating and taking down a 3000-node botnet. In: ACSAC '10 Proceedings of the 26th Annual Computer Security Applications Conference. New York, NY, USA, 141–150.

Davis, C. M. et al. SCADA cyber security testbed development. In: 38th North American Power Symposium, 2006. NAPS 2006. Carbondale, IL, USA, 483–488.

Falliere, N., Murchu, L. O., and Chien, E. (2011) W32.Stuxnet Dossier Version 1.4. Symantec Security Response.

General Electric (2011, Jan.) GENe Software Suite. Available from http://www.gedigitalenergy.com/products/brochures/uos/GENeSoftwareSuite.pdf. (8 Apr. 2013).

Hadeli, H. et al. (2009) Leveraging determinism in industrial control systems for advanced anomaly detection and reliable security configuration. In: IEEE Conference on Emerging Technologies & Factory Automation, 2009. ETFA 2009, Mallorca, Spain, 1–8.

Hibler, M. et al. (2008) Large-scale virtualization in the Emulab network testbed. In: USENIX 2008 Annual Technical Conference. Boston, MA, USA, 113–128.

Hull, J. et al. (2012) Staying in control: Cybersecurity and the modern electric grid. IEEE Power Energy Mag., 10 (1/Jan.–Feb.). 41–48.

Hydro-Québec (2012) Rapport annuel 2012. Available from http://www.hydroquebec.com/publications/fr/rapport_annuel/pdf/rapport-annuel-2012.pdf. (2 Apr. 2013).

Kobayashi, T. H. et al. (2009) Analysis of malicious traffic in modbus/TCP communications. In: Lecture Notes in Computer Science. 5508. Berlin, Germany: Springer-Verlag. 200–210.

O'Harrow, R. J. (2012, Nov.) CyberCity allows government hacker to train for attacks. Available from http://www.washingtonpost.com/investigations/cybercity-allows-government-hackers-to-trainfor-attacks/2012/11/26/588f4dae-1244-11e2-be82-c3411b7680a9_story.html. (20 Mar. 2013).

Paré, D. et al. (2003) Validation tests of the hypersim digital real time simulator with a large ACDC network. In: International Conference on Power System Transients - IPST 2003. New Orleans, LA, USA, 1–6.

Rosset, B. (2012) Rapport de Stage - Étape 5. Polytech Paris Sud, Paris, France.

U.S. Department of Energy (2009, Sep.) National SCADA test bed enhancing control systems security in the energy sector. Available from http://energy.gov/sites/prod/files/oeprod/DocumentsandMedia/NSTB_Fact_Sheet_FINAL_09-16-09.pdf. (2 Apr. 2013).