Received: 3 September 2019 | Revised: 8 October 2019 | Accepted: 9 October 2019 | Published on: 24 November 2019

DOI: 10.1002/itl2.132

LETTER

WILEY

Teaching the process of building an Intrusion Detection System using data from a small-scale SCADA testbed

Leandros Maglaras[1] | **Tiago Cruz**[2] | **Mohamed A. Ferrag**[3] | **Helge Janicke**[1]

[1]School of Computer Science and Informatics, De Montfort University, Leicester, UK

[2]Department of Informatics Engineering, University of Coimbra, Coimbra, Portugal

[3]Department of Computer Science, University of Guelma, Guelma, Algeria

Correspondence
Leandros Maglaras, School of Computer Science and Informatics, De Montfort University, Leicester, UK.
Email: leandros.maglaras@dmu.ac.uk

Present Address
School of Computer Science and Informatics, De Montfort University

Security of Critical National Infrastructures (CNI) is one of the major concerns to countries both in a European and in a worldwide level. Training on scenarios that involve such systems is important to the effective handling of incidents. Experiential learning is based on the importance of involvement and active engagement of attendees to develop personal experiences and increase the understanding of the investigated topic. This article describes a final year module for teaching traffic anomaly detection, an important part of of network security methods, to Computer Science and Informatics students. The course consists of 12 week labs and lectures run at De Montfort University. Students learn how to combine programming, data mining and security skills in order to build their own Intrusion Detection System using data from a small-scale SCADA testbed.

KEYWORDS

datasets, intrusion detection systems, machine learning, pedagogy, SCADA

1 | INTRODUCTION

Over recent years network and information security has evolved into a primary field in information and communication technology research. Computer departments of Universities around the globe are incorporating modules or classes that are offering cyber-security related material. At a national level countries published their National Cyber Security Strategies, many containing clear statements in support of research and development programs and academic educational programs in the field of cyber security. Among the objectives is the collaboration to enhance cyber security across all levels, from threat information sharing, to awareness raising.[1] To achieve this at a European level, the European Security and Defense College (ESDC) has launched earlier this year a cyber platform to coordinate education, training, evaluation and exercises (ETEE)[2] in the field of cyber security/defense across Europe.

Security of the Critical National Infrastructures (CNI) is one of the major concerns at a European and worldwide level.[3] Europe has issued the Directive on security of network and information systems (NIS Directive) which is the first piece of EU-wide legislation on cybersecurity of critical infrastructures (CIs). Supervisory control and data acquisition systems (SCADA) systems are essential for the safe and reliable operation of CIs and during the recent years they have become the target of advanced cyber-attacks due to their convergence with public and corporate networks mainly for easier monitoring and control. Moreover, there is an increased industry demand in cyber security training for Industrial Control Systems (ICS) and SCADA.[4] These factors make the training on scenarios that involve such systems very important. Experiential learning is based on the importance of involvement and active engagement in different scenarios that help the attendees develop personal experiences and increase the understanding of the investigated topic.[5]

This article describes a final year course for teaching an important class of network security methods: traffic anomaly detection. The course is targeted at Computer Science and Informatics students and uses data from a small-scale SCADA testbed from University of Coimbra, which contains traffic from a variety of different network attacks targeting ICS systems.

2 | RELATED WORK

In order to minimize the risk of inappropriate student behavior, Trabelsi and Saleous[6] presented the fundamental concepts students must understand: network keylogging and eavesdropping attacks. Specifically, the students are put into groups of three and given three machines, which have CommView, a network analyzer, and packet sniffing software tool, installed. The Local Area Network (LAN) is used to expose the network eavesdropping, or sniffing, attacks.

For understanding security auditing methods, Zseby et al[7] proposed a project for teaching network traffic anomaly detection methods using a large IP darkspace monitor operated at the University of California, San Diego (UCSD). The exercises are based on four software tools, including, tcpdump, corsaro, MATLAB, and RapidMiner.

Teaching basic game theory has been used by Hamman et al[8] to improve adversarial thinking in Cybersecurity students. This work has proven that under 2 hours of basic instruction in game theory leads to a statistically significant improvement in students' ability to anticipate the strategic actions of others.[9]

Thompson et al[10] proposed a new RFID reference model for teaching RFID systems under the RFID INFOSEC project. The RFID INFOSEC project focuses on RFID-related security and privacy threats, risks and mitigation technologies. There are six modules that each include many lessons on INFOSEC RFID, such as RFID Background and RFID Standard. Zseby et al[7] introduced a network security laboratory project to educate students in electrical engineering on methods for the detection of network traffic anomalies. Lee et al[11] proposed a small competition-based network security lab, named NetSecLab, for teaching system and network security. Through the NetSecLab project, students learn how to install the specified Linux distribution, configure the required services, and investigate attack techniques.

Xu et al[12] proposed a cloud-based virtual laboratory education platform, named V-Lab, for network security education. The V-Lab platform uses OpenFlow switches and virtualization technologies. The V-Lab platform uses a teaching model based on the following three phases: (a) Transfer of fundamental knowledge in networking and cryptography; (b) Apply previous knowledge on more realistic and complex experiments, and (c) Investigate existing network security systems and build their own systems.

Numerous Universities around the globe are incorporating in their programs specialized modules that are related to cyber security, network security, and cyber threat intelligence among others. In some of those modules cyber security training and simulation platforms, so-called cyber-ranges, are used in order to enhance the involvement of students in realistic situations in operational or strategic level.[13] Most of these cyber-ranges have pre-built scenarios that students try to solve or include realistic red blue exercises where some students take the role of attackers while others try to defend the systems (Blue team). De Montfort University in addition to providing those modules, decided some years ago to teach students how to built from scratch an Intrusion Detection System (IDS). The IDS is based on the principles of One Class Support Vector Machine classifier, combining both basic programming in JAVA, Python, use of Machine Learning methods and experimentation with realistic datasets from a small scale SCADA testbed that were provided from University of Coimbra under the CoCkpitCi project.[14]

3 | APPROACH & EDUCATIONAL OBJECTIVES

DMU's Cyber Technology Institute, is recognized as an Airbus Centre of Excellence in SCADA cyber security and forensics, offers lectures about security concepts along with labs that complement the acquired knowledge with hands-on experiments. Specific modules about cyber threat Intelligence, incident response, forensics, industrial control systems security and privacy have been developed in order to give to students, both undergraduate and postgraduate, a holistic training in cyber security.

The educational objectives of the ICS Intrusion Detection System module are the following:

- Teach students basic network data analysis methods. During the module, students get familiarized with network data traces analysis, data features, statistical methods and ensemble learning.
- Provide students with insights on intrusion vector strategic handling and exploitation by means of hands-on exercises
- Enhance students' network security knowledge. Students gain knowledge about attacks that can target ICS systems. Datasets containing layer 2/3/4 attacks like FIN/SYN Scan and SYN Flooding along with SCADA protocol and process-level attacks are used and analyzed.
- Help students learn how an IDS is built. During the lab student built several components of an IDS that include, capturing of raw data, data preprocessing, creation of alerts, ensemble of outcomes from different classifiers, aggregation

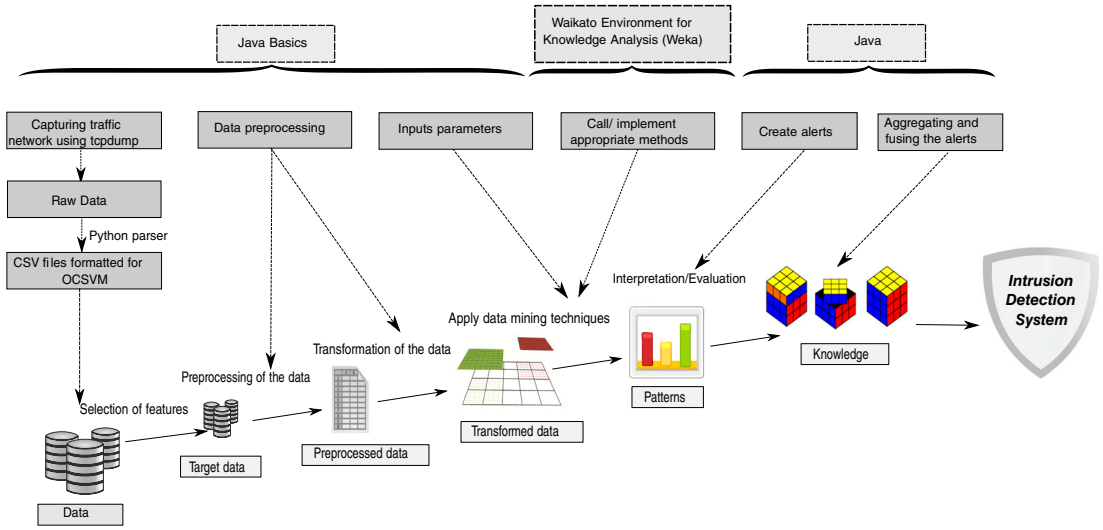

FIGURE 1 Buidling an IDS Step by step

and fusion techniques. Also students get familiarized with the procedure of creating and sending dedicated messages to a central entity that collects warnings from distributed IDSs.

- Enable student's general scientific skills. Students learn how to solve simple to complex problems, take right or wrong decisions and deal with difficult circumstances.

4 | KEY DESIGN DECISIONS

The module was set up in a way that students would learn how to implement their own IDS using TCPdump, JAVA, WEKA and Python capabilities. Following the steps as shown in Figure 1 students learn how to capture live traffic using TCPdump. The file that TCPdump create were transformed using Java to a CSV file that could be later used as an initial pool for feature selection.

When dealing with a classification problem, feature extraction is an essential step that once implement correctly helps the creation of an efficient method. Attributes inside datasets can come in all forms: continuous, discrete, symbolic, having significant variations both in resolution and ranges. Most classification methods cannot process data in such a format and pre-processing is required before classification models can be built and trained. Pre-processing in general consists of two steps. The first step involves mapping symbolic valued attributes to numeric valued attributes while the second step is scaling. Scaling is mainly used in order to avoid attributes in greater numeric ranges dominate those in smaller numeric ranges. During the data pre-processing phase, student build their own classes that can select features or create new ones.

The following labs involve familiarization of the students with using WEKA by calling specific procedures from JAVA, running several classification methods and setting up parameters such as kernel type, gamma, number of outliers, number of clusters, etc depending on the classification method that is used.

During the following labs, student learn how to create different classifiers that can work in parallel using the same or similar data sets and how to combine their outcomes by ensemble based mechanisms that use mean majority voting. Also additional fusion mechanisms that are based on clustering, for example, K-means clustering are introduced to the class and students can choose the one that induces low overhead in the communication among nodes while on the same time keeps all important information regarding attacks.

In order to be able to build an IDS that can be integrated in a general defense mechanism, students learn how to create IDMEF files. IDMEF is the standard for exchanging intrusion detection related events. A typical IDMEF file produced by our system is shown in Figure 2. Each IDMEF message contains information regarding the source of the attack (lines 11-17), the time that the attack was detected (line 10), the method (line 7) that detected it (When multiple IDS exist inside the defense mechanism), an initial classification of the detected attack (line 25) along with any other information that we want to encapsulate (line 6 & lines 18-24). Knowledge of the source node where the intrusion originates from or the attack is first detected is a very important feature of any IDS systems. Once the infected node is detected the infection can be potentially isolated from the rest of the network. Fast and accurate detection of the source node of a contamination and the type of the attack or infection is crucial for the correct function of an IDS.

```
<?xml version="1.0" encoding="UTF-8"?>
<idmef:IDMEF-Message version="1.0" xmlns:idmef="http://iana.org/idmef">
  <idmef:Alert>
    <idmef:Analyzer analyzerid="test">
      <idmef:Node category="unknown">
        <idmef:location>IT Network</idmef:location>
        <idmef:name>OCSVM</idmef:name>
      </idmef:Node>
    </idmef:Analyzer>
    <idmef:CreateTime ntpstamp="0x113dd481.0xf0000000">2015-01-22T10:50:28+01:00</idmef:CreateTime>
    <idmef:Source>
      <idmef:Node>
        <idmef:Address category="mac">
          <idmef:address>00:50:56:bf:41:d5</idmef:address>
        </idmef:Address>
      </idmef:Node>
    </idmef:Source>
    <idmef:Target>
      <idmef:Node>
        <idmef:Address category="mac">
          <idmef:address>00:0d:22:04:d1:cd</idmef:address>
        </idmef:Address>
      </idmef:Node>
    </idmef:Target>
    <idmef:Classification text="POSSIBLE ALARM"/>
  </idmef:Alert>
</idmef:IDMEF-Message>
```

FIGURE 2 Typical IDMEF message produced by the IDS for incident reporting

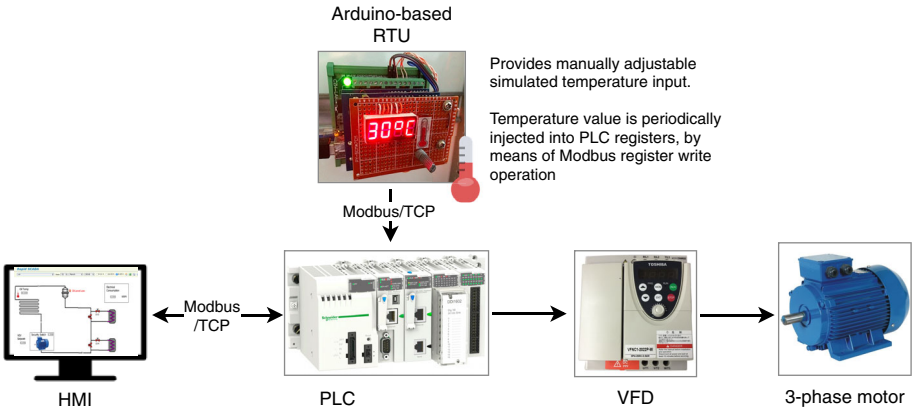

FIGURE 3 Cyber-physical process testbed

The reference testbed provided to the students (see Figure 3) emulates a Cyber-Physical process (comprising operations and field networks) controlled by a SCADA system using the Modbus/TCP protocol, being integrated within a layered IEC 62443-like structure.[15] It consists of a liquid pump simulated by an electric motor controlled by a Variable Frequency Drive (VFD), allowing multiple rotor speeds. The VFD is controlled by a Modicon M340 Programmable Logic Controller (PLC). The motor speed is determined by a set of predefined liquid temperature thresholds, whose measurement is provided by a MODBUS Remote Terminal Unit (RTU) device providing a temperature gauge. This is simulated by a potentiometer connected to an Arduino - the microcontroller is connected to a wired ethernet network by means of a expansion board with a Wiznet 5100 ASIC. The PLC communicates horizontally with the RTU, providing insightful knowledge of how this type of communications may have an effect on the overall system. The PLC also communicates with the Human-Machine Interface (HMI) which provides the supervisory interface for the system.

4.1 | Attacks as a security measure

As a preliminary step, and after learning the necessary basics of traffic capture and analysis procedures, students are encouraged to plan and execute a series of attacks against the testbed infrastructure, from scouting or classic Denial of Service (DoS) flooding procedures to process-level manipulation. The main rationale for this step-by-step approach is to provide students with a good insight about intrusion vectors and their strategic handling and exploitation by means of hands-on exercises. This is considered a valuable prior step before students proceed to the implementation of the IDS components.

Regarding DoS attacks, several categories are implemented, such as ping, TCP SYN or Modbus query flooding, all targeting the PLC. While the first two attacks operate mostly at OSI layers 2 to 4, attempting to overwhelm the capacity of the network or the networking subsystem in the target device with requests (TCP SYN flood attacks try to induce resource

exhaustion in the TCP/IP stack due to a series of incomplete connection attempts), the third attack works at the SCADA protocol layer, flooding the device with Modbus read request operations for a series of PLC register addresses, which may lead to side effects such as device resource exhaustion, scan cycle latency deviations or loss of connectivity. The first two attacks are usually implemented using the the hping3[16] and nping[17] tools, which are capable of generating conventional DoS attacks and simulated DDoS by spoofing the packet's IP address (specifically, by setting the source with a random valid IP address). The third attack was generated using the SMOD tool.[18] Moreover, a public dataset[19] was generated for these attacks, which was used for research purposes.[20]

When moving towards the execution of more sophisticated attacks, students are taught how to execute man-in-the-middle interception procedures (using ARP spoofing or DNS poisoning) using the ettercap tool,[21] being asked to perform a two-step man-in-the-middle interception procedure between the PLC and the HMI. In the first stage ARP poisoning is used to steer the communications through a third-party opponent, for scouting purposes - students are encouraged to capture and analyze SCADA traffic, in order to better understand the nature of the process under control, as well as the variables/registers involved in the device interaction workflow. Once students reach a viable conclusion about the relevant registers and their functions within the process model, the second stage takes place: students write ettercap scripts to disrupt the process behaviour by means of inflight manipulation of Modbus registers, while hiding the intrusion by feeding false data to the operator HMI (therefore blinding it). Advanced students are also encouraged to pursue the usage of the sophisticated (albeit also more complex) Scapy[22] fuzzing framework to implement elaborate intrusion procedures.

4.2 | IDS evaluation

In order to be able to evaluate the efficiency of the IDS in terms of accuracy and communication each student is given with labeled datasets that contain several attacks extracted from the testbed. Also during the last week, student try their trained IDSs during real time attacks. The lab is isolated and tutors of the lab are creating DoS and DDoS attacks against a random machine inside the network. The attack unfolds in a progressive way, starting with a few packets per minute to scale up to thousands of packets per second. By using their IDS, students try to identify the attack while also competing with each other in terms of time, accuracy and quality of alerting.

In order to evaluate the performance of their IDS, students learn to use several metrics, that represent the quality of the detection mechanism. On the one hand they learn how to compute important performance indicators, including, Detection Rate (DR), False Alarm Rate (FAR), Recall, Precision and Accuracy (ACC). These metrics can be used in order to evaluate the capability of the IDS to correctly classify each event inside the network. On the other hand other performance metrics, such as computational cost and communication overhead are calculated in order to measure the complexity of an IDS. Computational cost is measured as the total time required to perform classification of the dataset and output the final alarms, while communication overhead is measured by the number of IDMEF files that the IDS creates and sends to the HMI. Communication overhead and accuracy must be balanced in a way that the IDS identifies an anomaly correctly and fast, while on the same time it does not overwhelm the HMI with false or repeated alarms that could seriously affect the efficiency and usability of the detection mechanism.

5 | DISCUSSION

The module consists of 12 weeks labs and lectures and is run in DMU, following a structure co-designed and developed together with a team from the University of Coimbra (UC), where some lectures and exercises are also used as part of a cyber security course. During this period, students learn how to combine programming, data mining and security skills in order to build their own Intrusion Detection System. Basic knowledge about IP networks, as well as basic experience with programming (preferably Java or Python) and shell scripting, are expected of students enrolled in the module. The IDS that students produce can run in a distributed way and can be combined with other security mechanisms. The basic model that the module is based was one of the basic outcomes of the FP7 “Cockpit CI project” and it was already tested in realistic circumstances. Students that have basic knowledge on programming can cope easily with the module and can get their hands on implementation and experimentation in realistic environments. When the students finish the module they have acquired the knowledge needed in order to deploy an ML based IDS including all the necessary steps that include data capturing and preprocessing, creation, ensemble, aggregation and fusion of alarms.

In the near future we are planning to further improve the module by incorporating the IDS inside a hybrid testbed, entitled CYRAN,[13] that members of our lab have already implemented and used both for awareness and teaching purposes, mostly for red blue team exercises. CYRAN is a realistic environment used for cyber warfare training, cyber resiliency testing, cyber technology development and for the collection of labeled datasets for further research.

6 | CONCLUSIONS

The CTI lab teaches network traffic anomaly detection security implementation for ICS systems, to computer science and informatics students. The lab follows a research-oriented teaching approach and uses realistic network traffic from a small-scale SCADA testbed along with real time attack scenarios in order to help students react in difficult situations and solve problems by combining different techniques. The implementation of the module gives the students the ability to gain technical and problem solving skills that are required when dealing with real problems.

ORCID

Leandros Maglaras https://orcid.org/0000-0001-5360-9782
Tiago Cruz https://orcid.org/0000-0001-9278-6503

REFERENCES

1. ENISA. National Cyber Security Strategies. https://www.enisa.europa.eu/topics/national-cyber-security-strategies; 2018. Online; accessed November 7, 2018.
2. European External Action Service (EEAS). New EU cyber platform to boost cyber security capabilities across Europe, 2019. [ONLINE] Available at: https://eeas.europa.eu/headquarters/headquarters-homepage/39852/node/39852_el. accessed November 5, 2019.
3. Maglaras LA, Kim KH, Janicke H, et al. Cyber security of critical infrastructures. *Ict Express*. 2018;4(1):42-45.
4. Sitnikova E, Foo E, Vaughn RB. The power of hands-on exercises in SCADA cyber security education. In: IFIP World Conference on Information Security Education. Springer; 2009: 83–94.
5. Cook A, Smith RG, Maglaras L, Janicke H. SCIPS: using experiential learning to raise cyber situational awareness in industrial control system. *International Journal of Cyber Warfare and Terrorism (IJCWT)*. 2017;7(2):1-15.
6. Trabelsi Z, Saleous H. Teaching keylogging and network eavesdropping attacks: student threat and school liability concerns. In: Global Engineering Education Conference. IEEE; 2018: 437–444.
7. Zseby T, Vázquez FI, King A, Claffy KC. Teaching network security with IP darkspace data. *IEEE Transactions on Education*. 2015;59(1):1-7.
8. Hamman ST, Hopkinson KM, Markham RL, Chaplik AM, Metzler GE. Teaching game theory to improve adversarial thinking in cybersecurity students. *IEEE Transactions on Education*. 2017;60(3):205-211.
9. Zoto E, Kowalski S, Frantz C, Lopez-Rojas E, Katt B. A pilot study in cyber security education using CyberAIMs: a simulation-based experiment. In: IFIP World Conference on Information Security Education. Springer; 2018: 40–54.
10. Thompson DR, Di J, Daugherty MK. Teaching RFID information systems security. *IEEE Transactions on Education*. 2013;57(1):42-47.
11. Lee CP, Uluagac AS, Fairbanks KD, Copeland JA. The design of NetSecLab: a small competition-based network security lab. *IEEE Transactions on Education*. 2010;54(1):149-155.
12. Xu L, Huang D, Tsai WT. Cloud-based virtual laboratory for network security education. *IEEE Transactions on Education*. 2013;57(3):145-150.
13. Hallaq B, Nicholson A, Smith R, Maglaras L, Janicke H, Jones K. CYRAN: a hybrid cyber range for testing security on ICS/SCADA systems. In: Cyber Security and Threats: Concepts, Methodologies, Tools, and Applications. IGI Global. 2018 (pp. 622–637).
14. Cruz T, Rosa L, Proença J, et al. A cybersecurity detection framework for supervisory control and data acquisition systems. *IEEE Transactions on Industrial Informatics*. 2016;12(6):2236-2246.
15. Maglaras LA, Jiang J, Cruz TJ. Combining ensemble methods and social network metrics for improving accuracy of OCSVM on intrusion detection in SCADA systems. *Journal of Information Security and Applications*. 2016;30:15-26.
16. Hping - Active Network Security Tool. http://www.hping.org. accessed October 4, 2019.
17. nping - Network packet generation tool/ping utiliy. https://nmap.org/nping. accessed October 4, 2019.
18. SMOD MODBUS Penetration Testing Framework. https://github.com/Exploit-install/smod. accessed October 4, 2019.
19. Frazão I, Abreu PH, Cruz T, Araújo H, Simões P. Cyber-security Modbus ICS dataset, IEEE Dataport, https://doi.org/10.21227/pjff-1a03. 2019
20. Frazão I, Abreu PH, Cruz T, Araújo H, Simões P. Denial of service attacks: detecting the frailties of machine learning algorithms in the classification process. *International Conference on Critical Information Infrastructures Security*. Cham, Switzerland: Springer; 2018:230-235.
21. Ettercap project home page. https://www.ettercap-project.org. accessed October 4,2019.
22. Scapy Project home page. Available at: https://scapy.net. accessed October 4, 2019.

How to cite this article: Maglaras L, Cruz T, Ferrag MA, Janicke H. Teaching the process of building an Intrusion Detection System using data from a small-scale SCADA testbed. *Internet Technology Letters*. 2020;3:e132. https://doi.org/10.1002/itl2.132