Journal of Information Systems Education

Volume 34 | Issue 4 Article 3

12-15-2023

Teaching Tip: Hackalytics: Using Computer Hacking to Engage Students in Analytics

Andy Luse
Oklahoma State University, andyluse@okstate.edu

Forough Nasirpouri Shadbad
Oregon State University, forough.shadbad@oregonstate.edu

Journal of Information Systems Education

Volume 34
Issue 4
Fall 2023

Teaching Tip

Hackalytics: Using Computer Hacking to Engage Students in Analytics

Andy Luse and Forough Nasirpouri Shadbad

Recommended Citation: Luse, A., & Shadbad, F. N. (2023). Teaching Tip: Hackalytics: Using Computer Hacking to Engage Students in Analytics. *Journal of Information Systems Education*, 34(4), 370-386.

Article Link: https://jise.org/Volume34/n4/JISE2023v34n4pp370-386.html

Received: June 10, 2022
First Decision Made: July 17, 2022
Accepted: September 21, 2022
Published: December 15, 2023

ISSN: 2574-3872 (Online) 1055-3096 (Print)

Journal of Information Systems Education, 34(4), 370-386, Fall 2023

Teaching Tip
Hackalytics: Using Computer Hacking to Engage Students in Analytics

Andy Luse
Department of Management Science & Information Systems
Oklahoma State University
Stillwater, OK 74078, USA
andyluse@okstate.edu

Forough Nasirpouri Shadbad
Department of Business Information Systems
Oregon State University
Corvallis, OR 97331, USA
forough.shadbad@oregonstate.edu

ABSTRACT

The demand for qualified analytics professionals remains high with forecasts showing a continued need over the next few years. While this demand necessitates instruction in analytics in the classroom, many students find analytics concepts to be complicated and boring. This teaching brief describes a novel approach to teaching analytics through computer hacking. Students are exposed to the entire data lifecycle by first collecting intrusion detection data through the hacking of other student machines and then utilizing simple analytics procedures to analyze this data. Qualitative results show that the students enjoy the activity both in terms of the fun of hacking their fellow classmates as well as analyzing this data in an area less utilized in analytics instruction – security analytics. Three levels of the exercise are provided as well as how-to materials for students to run the exercise.

Keywords: Ethical hacking, Analytics, Online laboratory, Security information and event management

1. INTRODUCTION

Undeniably, the adoption of data analytics technologies has become a leading factor in enhancing organizational performance by helping make strategic decisions for businesses. Market Research Future and other industry reports forecast the global data analytics market size to reach almost 133 million U.S. dollars in 2026 and the worldwide revenue from big data and business analytics to increase to 274.3 billion U.S. dollars in 2022 (Market Research Future, 2021; Mlitz, 2021). The World Economic Forum and the U.S. Bureau of Labor Statistics also report that by 2022, the data analyst role will become one of the most demanded jobs, with a 25% increased demand in employment from 2019-2029 (Bureau of Labor Statistics, U.S. Department of Labor, 2021b; DuBois, 2020). Hence, educating individuals and preparing the future workforce for an in-demand career like data analytics are crucial.

Given the high demand for data analytics roles, universities and colleges offer degree programs and certifications to provide the required analytical skills for students (Parks et al., 2018). Pedagogical research on data analytics has examined different methods to design and teach analytics-related courses. Some exemplar approaches to teach business analytics (and big data analytics) include, but not limited to, CRISP-DM (CRoss Industry Standard Process for Data Mining) framework, artificial neural networks, project-based learning, and game-based team collaboration techniques facilitated by hands-on experiences via tools and applications such as MS Excel spreadsheets, Tableau, SAP-related applications (e.g., SAP ERP, SAP BusinessObjects Lumira, SAP Business Intelligence, SAP Analytics Cloud, SAP HANA), and Hadoop-based platforms like IBM InfoSphere BigInsights (Hoelscher & Mortimer, 2018; Jaggia et al., 2020; Johnson et al., 2004; Murray, 2022; Rienzo & Athappilly, 2012; Yang & Guo, 2020; Yazici, 2020; Zadeh et al., 2021). These studies provide noteworthy findings and successful evidence of learning outcomes and emphasize the role of active class participation and collaboration for given analytics problems.

Nonetheless, teaching analytics courses might sometimes be challenging for both learners and educators. Lack of knowledge, limited computer/programming skills, access to data and metadata, connections to remote networks, and availability of analytics tools are some factors that may cause difficulties (Attaran et al., 2018; Fischer et al., 2020; Yap & Drye, 2018). Delen (2021) also indicates significant challenges in teaching business analytics in terms of a lack of technical and behavioral skills (e.g., programming, data wrangling,

knowledge elicitation) and the broad scope of the topic due to the integration of multiple disciplines and knowledge areas like decision modeling, statistics, and databases. At the same time, students may feel bored and less motivated to engage in business analytic classes (Brookshire & Palocsay, 2005). Past research indicates that computing students show poor engagement (lower than average) than other majors (Sinclair et al., 2015). Findings of a survey on the state of business intelligence and analytics in academia indicated that students perceive the course as boring and uninteresting (Wixom et al., 2011). Thus, making the education of analytic courses fun and providing real-world examples would help students become more motivated (Presthus & Bygstad, 2012).

Recently teaching tips for both analytics (Jeyaraj, 2019; Wang & Wang, 2019; Zhang et al., 2020) and cybersecurity (Spears, 2018; Young, 2020) have appeared, but little research has investigated teaching tips in analytics in the context of cybersecurity. In order to meld the areas of analytics and cybersecurity and provide more fun and engagement when tackling associated challenges with teaching in these areas (e.g., data access and skill barriers), we take a novel approach to teach analytics to undergraduate students using security-related data. Network hacking is utilized to increase fun and engagement for students by allowing them to attack their classmates and exploit their machines. From this attacking, students are able to generate their own intrusion detection data and then analyze it. We applied this method in a security analytics module for several reasons. First, data analytics techniques have been highly utilized in cybersecurity problems such as malware and intrusion detection (Buczak & Guven, 2015). Second, students should be equipped with security analytical skills since there is a high shortage of cybersecurity analytics skills (ISC^2, 2019; Marquardson & Elnoshokaty, 2020). Third, the growth in cybersecurity analysts' employment (Bureau of Labor Statistics, U.S. Department of Labor, 2021a) calls for a need to prepare and motivate students for in-demand cybersecurity careers as well as prepare individuals for certificates such as the Cisco CyberOps, the CompTIA Cybersecurity Analyst, or the GIAC Certified Detection Analyst that test cybersecurity analytics skills. Fourth, the majority of educational information systems research explores the effectiveness and success scenarios of building cybersecurity programs and curricula that might include data analytics courses (Bicak et al., 2015; Marquardson & Gomillion, 2018; Stoker et al., 2021; Ward, 2021; Wymbs, 2016). Yet, to the best of our knowledge, few have applied data analytics techniques and pedagogical methods in cybersecurity courses. To this end, we propose a teaching method in analytics while students work with security data. The approach taken for teaching, step-by-step instructional process, analytics problems for a given module, and the course learning outcomes are described in later sections.

2. BACKGROUND

Data analytics has been expanded to various domains such as healthcare, banking and financial services, e-commerce and customer behaviors, and government (Krishnaprabha & Rahul, 2020). Given the high capability of data analytics in detection and analyzing cyberthreats, cyber analytics is another primary data analytics application, wherein big data analysis can play a key role in detecting cyber-attacks (Krishnaprabha & Rahul, 2020). A cyber analyst uses a set of algorithms and statistical analysis to surveil computers and networks, scrutinize security violations and breaches, and identify when a system becomes compromised. Cyber analytics techniques allow cyber analysts to use this information and provide security solutions to prevent future attacks. Job demand in cybersecurity analysts recently has become considerably higher and is predicted to grow 31% in the following years (Bureau of Labor Statistics, U.S. Department of Labor, 2021a). Many institutions offer data analytics programs and cybersecurity as independent degrees. Therefore, it is worth investigating research on recent pedagogical approaches to teaching analytics by using security modules that bridge these two areas.

Cyber defense education has utilized instruction in hacking techniques over the past two decades to both educate students in defensive techniques as well as provide engagement for students. Learning how to hack has been justified as the need to secure corporate systems necessitates an understand of the attacking methods that will be used by nefarious actors. As a corollary in sports, teams typically study tapes of other teams before playing in order to prepare for what they will bring, with many teams utilizing their backup squad to run the offense of the opposing team so they will be prepared to defend against it. So too does education in offensive techniques justify the means of offering such a course. Cyber defense competitions first became popular by providing a competitive learning environment for individuals to test their offensive and defensive skills (Luse & Triplet, 2009; Rursch et al., 2012; Rursch et al., 2009). These types of activities were also brought into the classroom environment to educate individuals in various aspects of cybersecurity. Both physical and virtual lab environments were constructed to enable education in network security, penetration testing, computer scanning, etc. (Anisetti et al., 2007; Son et al., 2015; Willems & Meinel, 2012; Wolf, 2009; Wu et al., 2014). Overall, this provided a fun and engaging atmosphere conducive to learning concepts integral to securing information systems.

While "hacking" education provides a much-needed tool, a key ingredient missing in many security educational environments is the area of log analysis. The ability to analyze and detect anomalies is extremely important to organizations, but the traditional method of perusing vast amounts of logs provides little engagement for students. Few studies examine different teaching methods using security data. Among them, project-based learning is common to apply in cybersecurity courses (e.g., Hamdan, 2017; Yates et al., 2019). In Hamdan's (2017) teaching case study, a group project was utilized to simulate the role of security analysts among students, and students were supposed to analyze Web server log data to detect potential intrusions. However, learning outcomes were not examined in this study. Yates et al. (2019) proposed machine learning techniques to educate students on how the method could be beneficial for malware identification and protection on mobile devices. Taking a different approach, Cornel et al. (2017) integrated gamification into cybersecurity concepts, including cryptography, incident response, and log analysis. They found that their method was successful in enhancing students' engagement and educational outcomes. Overall, education in penetration testing techniques has been shown to provide a highly engaging educational tool, but little research has looked at education on the defensive side through analysis of intrusion detection logs.

The effectiveness of such education can be viewed through the lens of the engagement theory of learning (Kearsley & Shneiderman, 1998). This theory provides a framework for technology-based teaching wherein students are meaningfully engaged in learning activities through interactions with tasks or collaborations with their group mates. This technique requires students to utilize their cognitive processes (i.e., problem-solving, reasoning, decision-making, and evaluation) to develop constructive learning. Drawing on this theory, we propose that our analytics exercise activities engage students in several meaningful activities to learn the concepts better.

3. HACKALYTICS EXERCISE

A hackalytics exercise was designed to introduce analytics techniques to computer and network security students. Several learning objectives were identified with regard to the course including 1) increasing student feelings of their capabilities with performing security analytics, 2) increasing student interest in security analytics, and 3) increasing student satisfaction with security analytics. Furthermore, areas of enjoyment and areas of improvement are of interest for future development of the project. The steps needed to implement the solution are given below including the environment setup and three exercise types depending on the needs of the instructor and students in a specific course. The example implementation for this research utilized an introductory course on computer security concepts. This course is a lab-based course with a substantial hands-on component. As part of their major, the students are able to add an analytics concentration, an information assurance concentration, or both, consisting of four classes in each. The analytics courses concentrate on the primary areas of descriptive, predictive, and prescriptive analytics. Given this, the full exercise was implemented to provide the most thorough example of the hackalytics exercise possible.

3.1 Environment Setup

Virtualization provides a useful technology when constructing multiple machines without the need for separate hardware for each machine. Virtualization provides the ability to install different operating systems on a single machine and has been demonstrated as useful for educational environments (Luse & Rursch, 2021). Several forms of virtualization exist from client programs all the way to enterprise software. While all the above platforms can run on most modern systems, the number of virtual machines capable of running on a single system is dependent on the hardware specifications of the system in accordance with those needed by each of the software products. For example, one of the biggest bottlenecks is the available random access memory (RAM) on a system, as each virtual machine needs its own RAM to run.

For this exercise, both individual client and server platforms have been utilized in the course over several semesters. Originally, each machine in the computer lab included an installation of a client-based virtualization product whereby students could install operating systems on each machine (The client software used at the time was first Microsoft Hyper-V and then VMWare Workstation). Each of these machines had suitable RAM to enable the installation of multiple operating systems at one time. While effective, a dedicated server was later installed where each student could install their operating systems (The server software used at the time was VMWare ESX managed by vCenter). This setup was deemed more advantageous for management (allowing each student to log in using their school credentials) as well as to enable the students to work remotely on their systems without needing physical access to the lab, which became even more needed given the increased use of virtual learning. One downside of this setup is that a server is needed with necessary RAM for all the students to install their machines on the same physical server. Some may not have access to this type of hardware and therefore the client machine method may be more advantageous given most machines in a typical lab today have sufficient RAM for one individual's virtual machines. Furthermore, if only one set of attacking machines will be utilized, as described in the scenarios below, then significantly less RAM is needed as each student would only need one machine to do attacking.

The students were each given access to the server machine and the ability to install multiple operating systems. They were also provided with a virtual switch to plug all their machines into a virtual network. This virtual switch worked just like a home router whereby each virtual machine is plugged into the switch to enable communication with other virtual machines. While each student was given their own virtual switch, this is not necessary if a smaller installation is desired. Promiscuous mode was also enabled on the switch, so that the intrusion detection system could easily gather all data that passed through the switch.

One potential issue with this setup is that you do not want your machines directly connected to an outside Internet. Given security issues with older operating systems as well as the hacking that will take place, separating these machines from the actual Internet is beneficial. While a network address translation (NAT) environment can suffice, we chose to implement the freely available ISEAGE environment. ISEAGE, or Internet-Scale Event and Attack Generation Environment, is a system that mimics the wider Internet by providing public IP addresses and routing between networks. ISEAGE (http://www.iserink.org/) is freely available for download and installation on your own hardware and has been used for education in several areas across several institutions (Luse et al., 2021; Luse et al., 2011; Luse & Rursch, 2021; Luse et al., 2014; Rursch et al., 2009). Students were then given an IP address range (e.g., 83.76.91.1-253), subnet mask (e.g., 255.255.255.0), and default gateway (e.g., 83.76.91.254) to utilize on their network as well as a common virtual switch to connect all their virtual machines. Students could assign these IP settings to each machine to enable them to communicate on the network. Given varying levels of expertise, hardware, and time constraints, three different exercises are described for use in the classroom based on preference. Figure 1 provides a logical architectural layout for the setup.

The lab used for this exercise was a general lab utilized by all students in the college. Given the use of one server for all students and the ISEAGE environment, the only needed connection for the students was the use of a web browser to connect to their respective virtual machine consoles on the server. The server was housed in a separate room, though in previous semesters the instructors for the course actually housed the servers in a corner of the general lab due to space constraints. The locked-down nature of the virtual server and ISEAGE did not necessitate any added security issues and

therefore no special permission was needed from the IT department. In previous semesters when the virtual machines were installed on the individual machines in the lab, a separate NIC was installed on each of the individual machines leading to the ISEAGE environment to lock down the traffic of the VMs to keep it from getting on the general network, though this type of strict enforcement would not necessarily be needed if the students just attack their own VMs as then the traffic would not need to leave their virtual environment.

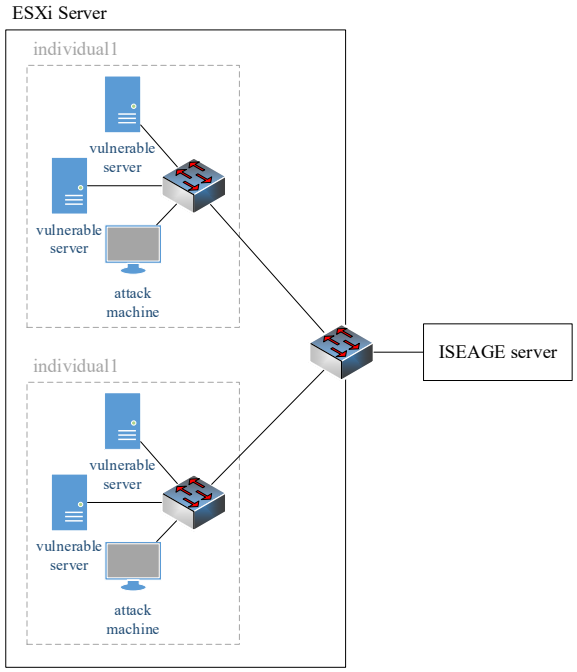


Figure 1. Logical Architectural Layout for Full Exercise

3.2 Full Exercise

The full exercise involves each student 1) installing their own operating systems, 2) installing a network capture device, 3) attacking the systems, and 4) running the analytics exercise. Students are first given step-by-step instruction for installing and configuring an intrusion detection (ID) virtual machine (see Appendix A for the how-to). Students utilized the Security Onion operating system given its built-in functionality for network traffic analysis (https://securityonionsolutions.com/). The instructions helped the student setup the virtual machine and configure it to capture data on their individual network.

Next, each student is provided with software to enable the installation of various operating systems. The students are encouraged to make their network as vulnerable as possible by installing old operating systems, turning off firewalls, removing passwords, etc. While not recommended for real-world settings, this setup has been found to considerably increase the fun of the experience for the students. Students enjoy installing antiquated software their parents used such as Windows 2000 or XP and have fun interacting with other students during the installation deriding and making fun of these old operating systems. Making these operating systems vulnerable by removing firewalls, deleting passwords, etc. is also fun for the students, who have been instructed in proper security practices their entire lives and now get to do the opposite. Furthermore, these old operating systems provide a nice playground for hacking by allowing anyone easy access to hack these old operating systems (Luse et al., 2018). Students are also encouraged to install other old operating systems such as Damn Vulnerable Linux (https://en.wikipedia.org/wiki/Damn_Vulnerable_Linux) or Hannah Montana Linux (http://hannahmontana.sourceforge.net/Site/Home.html) as these again increase the fun nature of the activity. The students have fun with these various niche distros as they each try to find the most unique OS to install on their network. After installation, the students then give IP settings to each virtual machine and connect each machine to the supplied virtual switch. Utilizing the ping utility can allow them to see if both their own machines are able to communicate on the network and also to see if they are able to communicate with the machines of other students (e.g., ping 83.76.91.5 to see if the machine with this IP will reply to the machine who pinged it).

After installation, the students are each provided with an attacking machine. Kali Linux was utilized as the operating system given its inclusion of attacking tools within the distribution (https://www.kali.org/). The students were also assigned a partner to attack. While individuals or corporations typically do not partner with unauthorized threat actors, this provided a collaborative educational method to enable the students to work with each other on attacking their networks. The first exercise involved students installing the Nessus vulnerability scanning tool (https://www.tenable.com/products/nessus) and scanning the network range of their partner. Nessus provides a delineation of vulnerabilities present on scanned machines as well as severity levels associated with each vulnerability. While limited in the number of IP addresses the user can scan, since each student was only attacking one other student network the free Nessus Essentials version was utilized for this exercise. Given the antiquated, vulnerable operating systems on the network, several critical vulnerabilities were present and students were encouraged to concentrate on these more easily hackable vulnerabilities.

Students were also provided an unpatched version of Windows 10 vulnerable to the EternalBlue (https://en.wikipedia.org/wiki/EternalBlue) exploit in order to give them a sample machine on which to practice. Students were then provided with several step-by-step exercises to demonstrate how to obtain remote access to a partner's Windows 10 machine and, once command and control was established, how to conduct other nefarious actions such as creating a folder mocking their opponent on the desktop, copying password hashes to further analyze, etc. They utilized their Kali machine and both the Metasploit command-line interface as well as the Armitage graphic interface to break into their Windows 10 machine using multiple methods. The students were also told to break into the other machines their partner installed using modules associated with the vulnerabilities found during their Nessus scans. This hacking exercise was perceived to provide enjoyment for the students as they broke into each other's machines. Further, this hacking provided traffic data to the Security Onion IDs for use later during the analysis phase. One advantage to this method is that the IDS captures traffic it sees as nefarious in nature, regardless of whether the attack succeeded. Therefore, even if the student is unsuccessful with attacking certain machines, just the act of

trying to attack will generate the necessary data for analysis for the activity.

After the attacking was completed, students were given a step-by-step module for gathering the log data for analysis (see Appendices B and C for the two-day exercises). Students exported the data from the Security Onion IDs in CSV format and downloaded this data to a Windows user machine. The modules for analysis utilized Python and involved three basic questions

1. What types of attacks are being used?
2. Which IP addresses (source and destination) appear more often than others?
3. When did these attacks occur?

Students completed the analysis using Python and then wrote out answers to these questions using their numerical findings as well as basic graphs. The questions were asked in such a way as to build upon one another in regards to both conceptual knowledge as well as analytic coding concepts. Figure 2 provides a flowchart of the overall analytics exercise. A fuller table listing of the steps is provided in Appendix E, including an indication whether the explicit commands were provided or if parts were strategically incomplete to encourage use of programming concepts from previous steps.

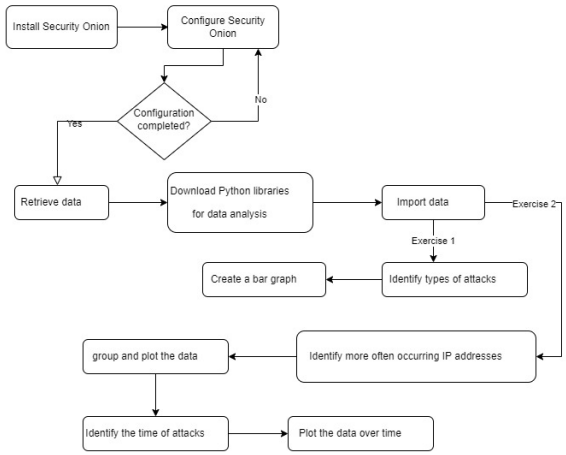
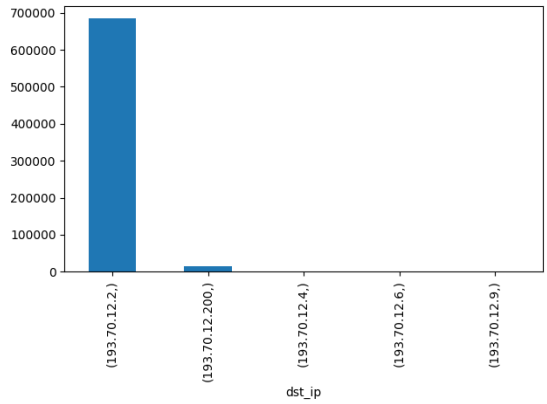


Figure 2. Flowchart of Analytics Exercise

The end product for the students included a written report. While the reports included additional prose, Figures 3 and 4 provide screencaps and associated answers by one student as answers to the questions in the module. Figure 3 shows the IP on the student's network that was getting attacked the most while Figure 4 shows the week period during which most of the attacking was taking place against their network.

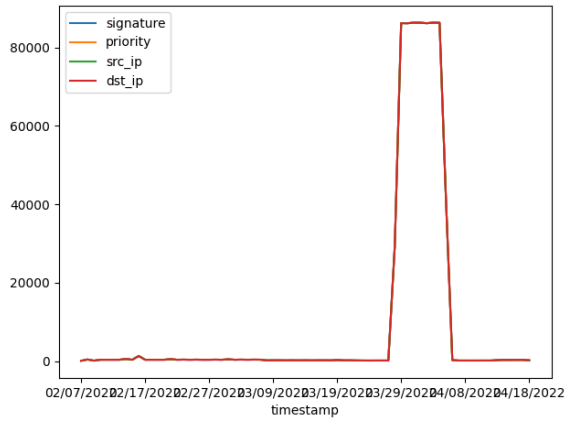
3.3 Partial Exercise

Some environments may not provide the necessary time or resources to complete an entire exercise. In these types of situations, a partial exercise is still feasible. The partial exercise would involve the instructor for the course setting up the vulnerable machines for the students to attack. Each student would then only be given a Kali VM for attacking the instructor's vulnerable network. The rest of the full exercise would then play out the same, with students scanning and attacking the instructor's network and then analyzing the network data created from these attacks. This solution can be desirable for those students who are novices at installing their own operating systems and also unfamiliar with network addressing. This partial exercise can also be extended to providing vulnerable operating systems to each student with network addressing already configured.



The second question my logs (screenshots above) were able to answer is which IP address was attacking the most and which of my IP addresses was getting attacked the most. The only source of the attacks came from 194.68.36.5 which is Matt's Kali machine. The two main targets for his attacks were 193.70.12.2 (my Windows 10 machine) and 193.70.12.200 (my Windows 2003 server hosting my DNS).

Figure 3. IP of the Machine Getting Attacked Most



The final questions this set of logs (screenshot above) was able to answer was when most of the attacks occurred. The chart is slightly hard to read but the attacks occurred between March 29th, 2022 and April 8th, 2022. This was around when we started doing the attacks in class.

Figure 4. Timeline of Attacks Against a Student's Network

3.4 Data Only Exercise

Finally, for those who wish to concentrate on the security analytics portion of the activity, the students can be provided

with already captured network intrusion detection data. This would involve the instructor setting up their own environment and attacking this environment prior to the student activity. Students are then provided with the data the instructor has already generated. While much less engaging, the students will still interact with a network security dataset that is a novel area for most analytics students. Table 1 shows a delineation of each of the modules as they are performed per participant by exercise type.

	Full exercise		Partial exercise		Data only exercise		min
	S	I	S	I	S	I	
Install OSes	x			x		x	45
Make vulnerable	x			x		x	10
Connect to network	x			x		x	10
Install IDS	x			x		x	30
Attack Systems	x		x			x	60
Export logs	x		x			x	10
Analyze logs	x		x		x		120

Table 1. Exercise Modules by Participant and Exercise with Minutes (min) to Complete (S=student, I=instructor)

As seen in Table 1, with the partial and data-only exercises, the onus on the instructor becomes greater. Nevertheless, the necessary expenditure of the instructor is not overly daunting. For example, in the data-only exercise, given the students will not be interacting with the machines themselves, the instructor can gather the data in a separate personal environment. The instructor can utilize a free personal virtualization product such as VMWare Player or VirtualBox to setup two or three vulnerable operating systems as well as a Security Onion IDS instance. Furthermore, Kali Linux provides a premade virtual machine to download for both of these environments to use to attack. From here, the instructor just needs to get attack traffic, so an "intense" nmap scan will do the trick. Once complete, the data can be exported from Security Onion using the method in the appendices. The advantage of this method is that, once completed, this data can be used in subsequent courses without the need to setup the environment again.

4. MEASURING LEARNING OUTCOMES

Learning outcomes were assessed for the security analytics exercise by collecting qualitative survey responses from a section of an applied information systems security course. Prerequisites for the course include introductory networking but no requirements for Python scripting. The full exercise was completed with each student having collected security alert data for their individual network while it was being attacked. The security analytics was broken up into two separate classes with the initial setup and introductory question asked the first day, and two more complex questions asked the second day (see Appendices B and C for the entire analytics exercise script). The class met two times per week for one hour 15 minutes each time. After completing the security analytics exercises for both days in class, a post survey was administered at the end of the second class. To obtain an in-depth understanding of how students perceive the given exercise – whether they found the module challenging or beneficial – they were given freeform open-ended questions pertaining to the exercise to share their thoughts (see Appendix D for the questions). The activity was graded as part of their work for the course and the survey was classified as pedagogical improvement. Overall, 14 students participated in the exercise. Demographic information of students is shown in Figure 5.

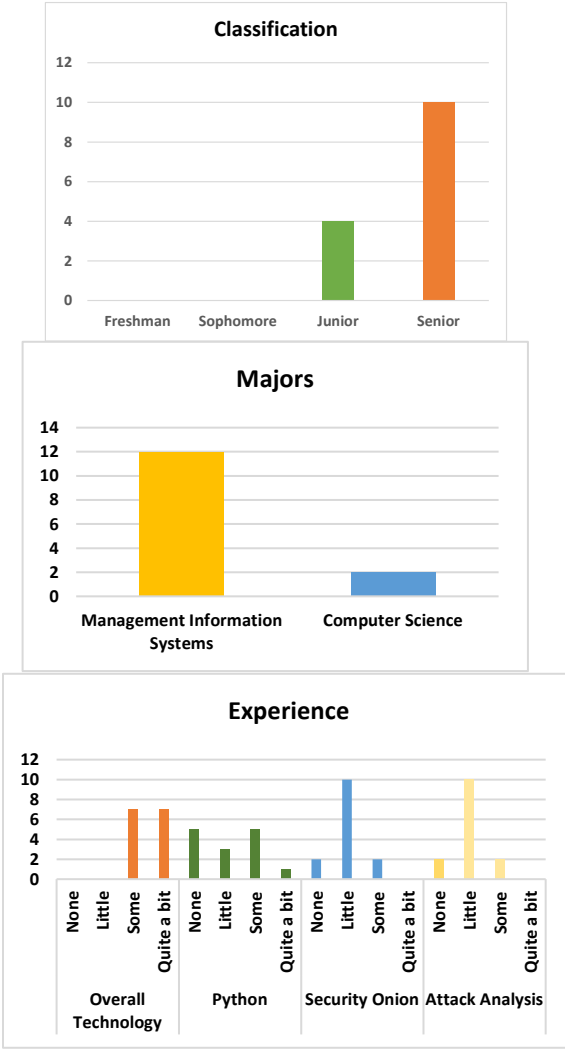


Figure 5. Demographic Information

4.1 Results of the Qualitative Analysis

We used qualitative responses to better understand the experience of students of performing the given assignments. This allowed us to obtain insights into the effectiveness of our proposed method. We analyzed a total of 39 text passages (13 students responded to three open-ended questions) using a coding approach to find main categories based on the perceptions from different parts of the module. We summarize

our findings in three sections: relevance and difficulty of the module activities, psychological and educational outcomes of performing the assignments, and recommendations for updating the module. We provided a sample of statements reflecting students feedback for each category.

4.1.1 Relevance and Difficulty of the Module Activities. Generally speaking, students find the module quite easy to understand. They found step-by-step instructions easy to follow that require less effort, as are indicated in the following statements:

"The module was straightforward in the instructions and wasn't too difficult to understand."
"Knowing that a script can do all of the hard work for you is great, as there would be a lot of manual labor otherwise."

We asked students what part of the assignment was more straightforward or challenging. Some students responded that the initial steps like setting up the environment and finding the IPs were the easiest steps in the experiment.

"Setting up the environment went very smoothly, and I didn't run into any problems retrieving the logs."
"I thought that the most straight forward step was finding the source IP from the data in order to see what machine was attacking my network the most."

However, some stated that IP filtering and data preparation for the analysis were the toughest steps:

"I think the most difficult part was formatting the data frame to get the data I needed for the timestamps."
"The more difficult part was making the correct data sets in order to graph the correct data."
"The most difficult part of this module was filtering out my IP addresses from the source IP addresses and using that in the plot."
"Once we got deeper into the filtering it became a little more complex and required some thinking."

Although IP filtering was the most difficult step for a few students, there was a student that reported this step as the most straightforward step. He stated that *"The step I found most straight forward was filtering the source IPs and finding the destination IPs."* This implies that background knowledge is important in the perception of the level of difficulty of the assignments.

Overall, students were able to perform all tasks and they were satisfied with what they had done and did not report any major difficulty with performing tasks:

"Overall, nothing was too difficult to where I couldn't proceed."
"The overall flow was pretty straightforward and easy."
"I thought this entire assignment was pretty easy but I also have a background in coding so it was nothing new for me."

4.1.2 Emotional Experience and Educational Outcomes. As an experiential emotionality of learning experience, students perceived the tasks as enjoyable. They found the module fun to participate in the activities and learn:

"I enjoyed doing this module."
"Overall it was fun to learn."

Particularly, for those who did not have previous experience in analyzing data, this module brought out enjoyment since one student stated that

"I thought the use of dataframes and the imported modules was helpful. I haven't worked with them much in my other classes, so I enjoyed the exposure."

Similarly, some steps of the module, like coding was perceived to be interesting. For example, two students indicated that:

"The coding aspect of this module was interesting since we learned how to find trends in large amounts of data."
"I found it interesting that I could monitor the time and date when the attacks happened."

Regarding educational outcomes, students described that their level of learning concepts enhanced after the module was completed. Some also stated that their technical skills in visualization and scripting in Python improved:

"I found it very beneficial learning how to make graphs out of all of the data that we were given. It helped me really understand what was happening when looking at all of the intrusion events."
"I have never really used Python before, so this was a good learning tool. I liked how it described why I was doing certain steps and then giving me the code I needed to write. I liked the steps where it did not give me the code word-for-word and let me figure out the code for myself."

Furthermore, the majority of participants emphasized the usefulness and benefits of the given module. They expressed that what they learned from this module would be beneficial to practice in the future and implement in real-world problems:

"I liked the assignment because it has real world applications with using Python code in order to view the attacks on a specific network."
"Being able to set up the environment to analyze logs is very useful, and the scripting skills I learned from this module will prove very useful in the future."
"I would continue this module in future semesters because I feel it is unique compared to other things we have done in the past."

4.1.3 Recommendations for the Future. The majority of the students liked the module as is. They found it easy to understand and its instructions straightforward with no main change recommendations. The following statements are some cases of what students thought about this module:

"I don't know what I would change about this module. It was nice to focus on coding for a little bit in this class because learning how to utilize code to help us defend a network will benefit me in a future career path."
"I would not change anything; module is great. The module is fine the way it is being taught"

However, some students suggested additional guidance on visualizations as one stated:

"If I were to use this module in the future, I would explain how to create more detailed visualizations of the attacks. The visualizations we created in the module were good for

the assignment, but some tips for creating more visualizations would be useful."

Also, two students recommend providing some detailed explanation of coding in Python. It seems a lack of knowledge in coding was challenging for some students. For example:

"I would say that it would be useful to have a slightly better understanding of coding Python for this module, but this is more on me as a student obviously."

"If you have someone in this class with little coding experience then it might be a little hard for them to understand what they are actually doing with each line of code they type."

5. DISCUSSION

Scholars have investigated the implementation of new tools and techniques to teach data science concepts such as Hadoop and Neural Network (Rienzo & Athappilly, 2012; Yang & Guo, 2020), yet in the context of security, little research has employed teaching analytics methods to examine students' learning performance. Data analytics courses are often perceived as boring and uninteresting, and educators face some obstacles to teach data analytics concepts to students (Brookshire & Palocsay, 2005). Therefore, this study proposes a teaching approach for analytics through incorporation of computer network hacking to increase student engagement. Students were given step-by-step instructions to attack each other's systems utilizing hacking techniques and older operating systems. After hacking was completed, log data were gathered and analysis performed to detect what types of attacks occurred and when.

The results of our qualitative analysis using open-ended questions showed that our approach was easy to follow and understand, even for those with little programming/technical skills. Students liked the uniqueness of this exercise as it was different from what they have done before. It encouraged them to participate in activities, address challenges, and figure out the problems. Students enjoyed the process and found the module interesting. Our findings based on students' feedback showed that they learned both concepts and analysis techniques. They found the method beneficial and instructive to implement in the future if needed.

Several areas for future research are available. First, while our proposed exercise showed promising results, its applications might be limited to using introduced tools and technologies. Given the evolving nature of technology, other software and technologies can be substituted to conduct the exercise. Second, only the full exercise was utilized to test this module. While all portions of the partial and data-only exercises are included in the full exercise, the participants (both students and instructors) may have different experiences with the other exercise types. Third, the technicality of this exercise can place some burden on the instructor. The instructor for this research also had a teaching assistant (TA) that had taken the course in a previous semester to aid the students. Furthermore, the larger the class, the greater the burden of the professor. The instructor for this course has had over 30 in the course in previous years and does note the added overhead with this many students.

Overall, our method of utilizing computer network hacking to increase student engagement in security analytics shows promise. By instructing students to install an intrusion detection system, hack systems to generate intrusion data, and analyze this data to solve security problems, students were able to gain understanding of analytics concepts in the novel area of security analytics. While teaching technical concepts for business students might sometimes be considered difficult or boring, our exercise engaged students to participate in tasks and enhance their educational levels through a fun exercise whereby they hacked other systems.

6. REFERENCES

Anisetti, M., Bellandi, V., Colombo, A., Cremonini, M., Damiani, E., Frati, F., Hounsou, J. T., & Rebeccani, D. (2007). Learning Computer Networking on Open Paravirtual Laboratories. *IEEE Transactions on Education*, 50(4), 302-311.

Attaran, M., Stark, J., & Stotler, D. (2018). Opportunities and Challenges for Big Data Analytics in US Higher Education: A Conceptual Model for Implementation. *Industry and Higher Education*, 32(3), 169-182.

Bicak, A., Liu, X. M., & Murphy, D. (2015). Cybersecurity Curriculum Development: Introducing Specialties in a Graduate Program. *Information Systems Education Journal*, 13(3), 99-110.

Brookshire, R. G., & Palocsay, S. W. (2005). Factors Contributing to the Success of Undergraduate Business Students in Management Science Courses. *Decision Sciences Journal of Innovative Education*, 3(1), 99-108.

Buczak, A. L., & Guven, E. (2015). A Survey of Data Mining and Machine Learning Methods for Cyber Security Intrusion Detection. *IEEE Communications Surveys & Tutorials*, 18(2), 1153-1176.

Bureau of Labor Statistics, U.S. Department of Labor (2021a). *Information Security Analysts*. https://www.bls.gov/ooh/computer-and-information-technology/information-security-analysts.htm#tab-1

Bureau of Labor Statistics, U.S. Department of Labor (2021b). *Operations Research Analysts*. https://www.bls.gov/ooh/math/operations-research-analysts.htm

Cornel, C. J., Rowe, D. C., & Cornel, C. M. (2017). Starships and Cybersecurity: Teaching Security Concepts through Immersive Gaming Experiences. *Proceedings of the 18th Annual Conference on Information Technology Education*.

Delen, D. (2021). Better Practices for Teaching Business Analytics. *Business Analytics*. https://doi.org/10.1287/orms.2021.04.27

DuBois, J. (2020). *The Role of Data Analysts in 2020 and Beyond*. quanthub. https://quanthub.com/data-analysts/

Fischer, C., Pardos, Z. A., Baker, R. S., Williams, J. J., Smyth, P., Yu, R., Slater, S., Baker, R., & Warschauer, M. (2020). Mining Big Data in Education: Affordances and Challenges. *Review of Research in Education*, 44(1), 130-160.

Hamdan, B. (2017). Teaching Case Study: Introducing Data Analytics in an Advanced Cybersecurty Course. *Journal of Computing Sciences in Colleges*, 33(2), 113-120.

Hoelscher, J., & Mortimer, A. (2018). Using Tableau to Visualize Data and Drive Decision-Making. *Journal of Accounting Education*, 44, 49-59.

ISC^2. (2019). *(ISC)² Finds the Cybersecurity Workforce Needs to Grow 145% to Close Skills Gap and Better Defend*

Organizations Worldwide. https://www.isc2.org/News-and-Events/Press-Room/Posts/2019/11/06/ISC2-Finds-the-Cybersecurity-Workforce-Needs-to-Grow--145

Jaggia, S., Kelly, A., Lertwachara, K., & Chen, L. (2020). Applying the CRISP-DM Framework for Teaching Business Analytics. *Decision Sciences Journal of Innovative Education*, 18(4), 612-634.

Jeyaraj, A. (2019). Teaching Tip: Pedagogy for Business Analytics Courses. *Journal of Information Systems Education*, 30(2), 67-83.

Johnson, T., Lorents, A. C., Morgan, J., & Ozmun, J. (2004). A Customized ERP/SAP Model for Business Curriculum Integration. *Journal of Information Systems Education*, 15(3), 245-254.

Kearsley, G., & Shneiderman, B. (1998). Engagement Theory: A Framework for Technology-Based Teaching and Learning. *Educational Technology*, 38(5), 20-23.

Krishnaprabha, S., & Rahul, R. (2020). A Study on Application of Data Analytics in Different Sectors. *International Journal of Research in Engineering, Science, and Management*, 3(4), 225-229.

Luse, A., Al Marzooq, A., & Burkman, J. (2018). Windows ME: Using Antiquated Software to Learn about Security. *IEEE Potentials*, 37(2), 10-12.

Luse, A., Brown, A., & Rursch, J. (2021). Instruction in 802.11 Technology in Online Virtual Labs. *IEEE Transactions on Education*, 64(1), 12-17.

Luse, A., Mennecke, B. E., Triplett, J. L., Karstens, N., & Jacobson, D. (2011). A Design Methodology and Implementation for Corporate Network Security Visualization: A Modular-Based Approach. *AIS Transactions on Human-Computer Interaction*, 3(2), 104-132.

Luse, A., & Rursch, J. (2021). Using a Virtual Lab Network Testbed to Facilitate Real-World Hands-on Learning in a Networking Course. *British Journal of Educational Technology*, 52(3), 1244-1261.

Luse, A., Rursch, J. A., & Jacobson, D. (2014). Utilizing Structural Equation Modeling and Social Cognitive Career Theory to Identify Factors in Choice of IT as a Major. *ACM Transactions on Computing Education*, 14(3), 1-19.

Luse, A., & Triplet, J. (2009). Utilizing Visualization Mechanisms to Improve User Performance During Cyber Defense Competitions. *MWAIS 2009 Proceedings*, 35.

Market Research Future. (2021). *Data Analytics Market to Hit USD 132.90 Billion by 2026 | North America Region to Spearhead the Global Data Analytics Industry with the Projected CAGR of 26.4%*. https://www.globenewswire.com/news-release/2021/02/08/2171129/0/en/Data-Analytics-Market-to-Hit-USD-132-90-Billion-by-2026-North-America-Region-to-Spearhead-the-Global-Data-Analytics-Industry-with-the-Projected-CAGR-of-26-4.html

Marquardson, J., & Elnoshokaty, A. (2020). Skills, Certifications, or Degrees: What Employers Demand for Entry-Level Cybersecurity Jobs. *Information Systems Education Journal*, 18(1), 22-28.

Marquardson, J., & Gomillion, D. (2018). Cyber Security Curriculum Development: Protecting Students and Institutions while Providing Hands-on Experience. *Information Systems Education Journal*, 16(5), 12-21.

Mlitz, K. (2021). *Big Data and Business Analytics Revenue Worldwide 2015-2022*. https://www.statista.com/statistics/551501/worldwide-big-data-business-analytics-revenue/

Murray, M. J. (2022). Teaching How Supply Chain Operations Impact Financial Results: A Case Study Using Cloud-Based Simulation. *Southwestern Business Administration Journal*, 20(1), article 2.

Parks, R., Ceccucci, W., & McCarthy, R. (2018). Harnessing Business Analytics: Analyzing Data Analytics Programs in US Business Schools. *Information Systems Education Journal*, 16(3), 15-25.

Presthus, W., & Bygstad, B. (2012). Business Intelligence in College: A Teaching Case with Real Life Puzzles. *Journal of Information Technol Education: Innovations in Practice*, 11, 121-137.

Rienzo, T. F., & Athappilly, K. K. (2012). Introducing Artificial Neural Networks through a Spreadsheet Model. *Decision Sciences Journal of Innovative Education*, 10(4), 515-520.

Rursch, J. A., Jacobson, D. W., & Luse, A. (2012). Using Content Analysis to Evaluate Student Inquiry-Based Learning: The Case of High School Students Preparing for a Cyber Defense Competition. Paper presented at the *2012 American Society for Engineering Education (ASEE) Annual Conference & Exposition*. San Antonio, Texas.

Rursch, J. A., Luse, A., & Jacobson, D. (2009). IT-Adventures: A Program to Spark IT Interest in High School Students Using Inquiry-Based Learning with Cyber Defense, Game Design, and Robotics. *IEEE Transactions on Education*, 53(1), 71-79.

Sinclair, J., Butler, M., Morgan, M., & Kalvala, S. (2015). Measures of Student Engagement in Computer Science. *Proceedings of the 2015 ACM Conference on Innovation and Technology in Computer Science Education*.

Son, J., Bhuse, V., Othmane, L. B., & Lilien, L. (2015). Incorporating Lab Experience into Computer Security Courses: Three Case Studies. *Global Journal of Enterprise Information System*, 7(2), 69-80.

Spears, J. L. (2018). Gaining Real-World Experience in Information Security: A Roadmap for a Service-Learning Course. *Journal of Information Systems Education*, 29(4), 183-201.

Stoker, G., Clark, U., Vanajakumari, M., & Wetherill, W. (2021). Building a Cybersecurity Apprenticeship Program: Early-Stage Success and Some Lessons Learned. *Information Systems Education Journal*, 19(2), 35-44.

Wang, S., & Wang, H. (2019). A Teaching Module of Database-Centric Online Analytical Process for MBA Business Analytics Programs. *Journal of Information Systems Education*, 30(1), 19-26.

Ward, P. (2021). Development of a Small Cybersecurity Program at a Community College. *Information Systems Education Journal*, 19(3), 4-10.

Willems, C., & Meinel, C. (2012). Online Assessment for Hands-on Cyber Security Training in a Virtual Lab. *Proceedings of the 2012 IEEE Global Engineering Education Conference (EDUCON)*.

Wixom, B., Ariyachandra, T., Goul, M., Gray, P., Kulkarni, U., & Phillips-Wren, G. (2011). The Current State of Business Intelligence in Academia. *Communications of the Association for information Systems*, 29(1), article 16, 299-312.

Wolf, T. (2009). Assessing Student Learning in a Virtual Laboratory Environment. *IEEE Transactions on Education*, 53(2), 216-222.

Wu, D., Fulmer, J., & Johnson, S. (2014). Teaching Information Security with Virtual Laboratories. In *Innovative Practices in Teaching Information Sciences and Technology* (pp. 179-192). Springer.

Wymbs, C. (2016). Managing the Innovation Process: Infusing Data Analytics into the Undergraduate Business Curriculum (Lessons Learned and Next Steps). *Journal of Information Systems Education*, 27(1), 61-74.

Yang, Z., & Guo, X. (2020). Teaching Hadoop Using Role Play Games. *Decision Sciences Journal of Innovative Education*, 18(1), 6-21.

Yap, A. Y., & Drye, S. (2018). The Challenges of Teaching Business Analytics: Finding Real Big Data for Business Students. *Information Systems Education Journal*, 16(1), 41.

Yates, D., Frydenberg, M., Waguespack, L., McDermott, I., OConnell, J., Chen, F., & Babb, J. S. (2019). Dotting i's and Crossing T's: Integrating Breadth and Depth in an Undergraduate Cybersecurity Course. *Information Systems Education Journal*, 17(6), 41-50.

Yazici, H. J. (2020). Project-Based Learning for Teaching Business Analytics in the Undergraduate Curriculum. *Decision Sciences Journal of Innovative Education*, 18(4), 589-611.

Young, J. A. (2020). The Development of a Red Teaming Service-Learning Course. *Journal of Information Systems Education*, 31(3), 157-178.

Zadeh, A. H., Zolbanin, H. M., & Sharda, R. (2021). Incorporating Big Data Tools for Social Media Analytics in a Business Analytics Course. *Journal of Information Systems Education*, 32(3), 176-198.

Zhang, L., Chen, F., & Wei, W. (2020). Teaching Tip: A Foundation Course in Business Analytics: Design and Implementation at Two Universities. *Journal of Information Systems Education*, 31(4), 244-259.

AUTHOR BIOGRAPHIES

Andy Luse is a William S. Spears Professor in Business and Associate Professor of Management Science and Information Systems in the Spears School of Business at Oklahoma State University. He received a B.A. degree in Computer Science from Simpson College, M.S. degrees in Information Assurance, Computer Engineering, Business Administration, and Psychology, and Ph.D. degrees in Human Computer Interaction, Computer Engineering, and Information Systems from Iowa State University. Andy has been published in the *Journal of Management Information Systems*, *Journal of the Association for Information Systems*, *Journal of Business Research, Communications of the Association for Information Systems*, *Decision Support Systems, Computers in Human Behavior*, and many other outlets.

Forough Nasirpouri Shadbad is an assistant professor at Oregon State University, College of Business. Her research interests include intentional/unintentional insider threats, technostress, and information privacy in social networking sites. She has published in the *European Journal of Information Systems, Communications of the Association for Information Systems*, *Information Technology & People, Journal of Computer Information Systems*, *the Americas Conference on Information Systems, the Hawaii International Conference on Systems Sciences*, and other proceedings.

APPENDICES

Appendix A. Security Onion Setup

Security Onion is a Linux distro built on top of Ubuntu that is made specifically for monitoring network traffic for intrusion detection.

Install Security Onion

- Make a new VM with the following settings…
  - Ubuntu 64-bit
  - 2 core CPU
  - 4GB RAM
  - 80 GB HD (**thin provision**)
  - NIC on your network
- Boot to the ISO file live boot
- Once it's booted up, double-click "Install SecurityOnion"
- Click Continue through all of the default settings and fill out any information requested that is appropriate
- Restart the machine when prompted

Configure Security Onion

- Run Setup on the desktop
- Configure the network interface with static IP settings on your network
- Reboot when prompted
- Once logged back in, make sure you can ping your default gateway.
- Run Setup again
- Skip the network configuration
- Run in **Production** Mode
- Make a New deployment
- Create a username and password
- Do a Custom setup
- Enter 90 for the number of days to keep alerts
- Keep the default repair days of 7
- Use the Emerging Threats Open ruleset
- Use the **Snort** IDS engine
- Enable sensor services
- Use default Ring size
- Enable the IDS
- Identify the name of your network using CIDR notation
- Enable Bro
- Enable file extraction
- Yes, full packet capture
- Use default pcap file size
- Enable mmap
- Accept default PCAP ring buffer
- Keep default purging
- Disable Salt
- Yes enable Elastic Stack
- Store logs locally
- Set the size to 60 GB
- Proceed with the changes and wait for the configuration to complete
- Read and click through the dialog boxes for more information on the tools that have recently been installed and user instructions on how Security Onion works

Make sure the timezone/date/time is correct, correct if needed

Appendix B. Day 1 Security Analytics Exercise

Objective: Combine simple data analytics tools and network data to provide insights related to cybersecurity.

Questions to answer:
1. What types of attacks are being used?

Retrieve data
The data we will be using will come from your SecurityOnion machine. These machines store what it sees as a potential attack in a MySQL database**.**

- Launch a terminal on your SecurityOnion machine. Run the following command:

```
sudo mysql --defaults-file=/etc/mysql/debian.cnf -Dsecurityonion_db -e 'select * from event' > Scan_Data.txt
```

This command queries the built-in MySQL database and outputs a file called "Scan_Data.txt" that is a tab-separated file containing all data SecurityOnion has collected.

- Send yourself the file via email

Setup environment for analytics
The actual analytic work will be done from a Windows 10 (NOT one of your Win10 VMs).

- Retrieve your data file by downloading the file you sent yourself, and saving it to the desktop.
- We must install a couple of third-party packages to assist in our work. Launch a regular Windows command prompt.
- To install packages, use the following command:

```
py -m pip install <package name> --user
```

- The packages you will need to install using the previous command are as follows:
  - pandas
  - matplotlib
  - datetime
- For future use, we need to know the version of Python that is installed (remember for later). To find out, use the following command:

```
py -V
```

- In the Windows search bar, type in "IDLE", run the version using the same version of Python that you found previously and the 64-bit version. For example, if **py -V** returned 3.7, then run the version of IDLE that uses Python 3.7 64-bit.
- Under "File," create a new file, which will be our Python script

Analytics analysis
It's finally time to start coding! To start, we must import the packages that we will be using. Some of these packages are included with a base installation of Python, which is why we did not install them earlier.

- To import a package, type **import <package name>** The packages that we will be importing are as follows:
  - **os** – allows for file/directory traversal
  - **pandas** – popular package for data manipulation and transformation
  - **ipaddress** – can convert IP addresses between formats
  - **warnings** – some code will throw warnings. This will let us ignore them.
- Datetime and matplotlib are a bit different. Import them as follows:

`import matplotlib.pyplot` – visualization framework
`from datetime import datetime` – working with dates

- To make things a bit easier and our code easier to read, we can rename packages to something shorter. For example, people commonly rename pandas to pd when working with it. To do that, we can edit our import statement to say "import pandas as pd" In this exercise, this is how the packages are named. (The packages that aren't listed here stay as they are).

```
import pandas as pd
import matplotlib.pyplot as plt
from datetime import datetime as dt
```

- In order to see all of the data that we need to, there is an option that needs to be set. After your import statements, set the maximum number of columns to display to 10. Otherwise, our data will be cut off when we try to look at it.

```
pd.set_option('display.max_columns', 10)
```

- Since we imported the warnings package to ignore warnings, we need to tell it to ignore them. Somewhere after you imported warnings, use this code:

```
warnings.filterwarnings("ignore")
```

- Let's import our data. The "os" package will let us access our data file wherever it is. Assuming that you have saved your data file to your desktop, use this statement:

```
os.chdir(r'C:\Users\<Your okey short name>\Desktop')
```

For example: it may read os.chdir(r'C:\Users\kecarmi\Desktop')

- We can read in our data using pandas into what is called a dataframe. Essentially, you can think of a dataframe as an Excel spreadsheet, with data organized into rows and columns. For simplicity, we can call our dataframe "df" This command looks for a file with the name provided in the current working directory. Since our raw data file is tab-separated, we need to tell pandas this.

```
df = pd.read_csv('scan_data.txt', sep='\t')
```

- Let's take a look at what columns are included in our data. To do that, we can print the columns of our data frame.

```
print(df.columns)
```

- From here, let's run our script so far to make sure there are no errors and to take a first look at our data. Shortcut: F5. If you have not saved yet, you will be prompted to do so (remember you'll be turning this in later).
- We can see that our data includes a ton of columns, and we will not need all of them. Let's filter out the columns we don't need by only including the ones we do need.

```
df = df[['signature', 'timestamp', 'priority', 'src_ip', 'dst_ip']]
```

This will only keep the listed columns. The 'signature' column provides a brief description of the attack, and 'priority' assigns a number denoting the importance of that event.

- Let's take a look at what our full dataset looks like now by examining the first few rows. Before you run the script again, comment out the previous print statement by inserting a "#" on the same line before the statement. Feel free to comment out any unused print statements in the future. After commenting out the previous print statement, run the script again.

```
print(df.head())
```

- You might have noticed that our IP addresses look a bit different than what you would expect. We can easily fix those by applying a lambda function to those columns. "Applying" just means applying a function to every value within a column. A lambda function is a quick way to define a custom function that we only need once or twice. To correct our IP addresses, we can do this using the following.

```
df['src_ip'] = df['src_ip'].apply(lambda x: str(ipaddress.IPv4Address(x)))
```

This line takes the existing integer IP addresses and changes them to the familiar XXX.XXX.XXX.XXX format as a string.

- Using this same technique, go ahead and fix your destination IP's as well. After doing this, check to see that it worked correctly by running another "print(df.head())". You will need this located under the previous commands to see them change.

<u>Question 1: What types of attacks are being used?</u>

- Let's start answering the questions that we asked at the beginning. To start, what kind of attacks are occurring more often than others? We can answer that by examining our "signature" column. In a SQL database, we would want to run a group by statement and get the count of the results. Luckily, Pandas has a built-in function that can give us exactly that. We can also save our results to a new Pandas object as not to overwrite our data.

```
df_q1 = df.value_counts(subset=['signature'], sort=True)
```

This will give us what we're looking for in an object called "df_q1" that has also been sorted from greatest to least.

- Go ahead and take a look at the first few rows of this using the head() function. Using your results, you can answer question 1.

- Data insights are commonly well communicated via visualization, and this is no different. There are many more complex functions and more complex libraries for making fancy visualizations and animations, but luckily, Pandas has built-in graphing functionality to use matplotlib to make quick and simple visualizations.

```
df_q1.head().plot(kind='bar')
plt.tight_layout()
plt.savefig('Q1.png')
```

Open the graph you just made on the Desktop. This creates a quick and simple bar graph showing the most commonly occurring event signatures. The "tight_layout" function tries to ensure that everything shows up on the picture. Unfortunately, since our signatures are very long, they get cut off otherwise. However, you would normally not have x-axis labels that long, so this is fine in our case. The "tight_layout" function will be of use later, however.

Points –provide the data requested including

- **What types of attacks are being used?**
- **Your script file**
- **Your graph image file**

Appendix C. Day 2 Security Analytics Exercise

Objective: Combine simple data analytics tools and network data to provide insights related to cybersecurity.

Questions to answer:

2. Which IP addresses (source and destination) appear more often than others?
3. When did these attacks occur?

Question 2: Which IP addresses (source and destination) appear more often than others?

- Next, let's take a look at which IP addresses appear more often than others. For the most part, this is a similar process to what we did previously. However, there's a couple of things to add. To start, when we are looking at common source IP's, we don't particularly care if the source comes from within our own network, so we need to exclude IP's from our own networks. To do this, we can use a for loop and an array to create a list of IP addresses that belong to us. For this example, my network is 96.172.97.0-255 (you should use your IP range).

```
ip_array = []
for i in range(0, 255, 1):
ip_array.append("96.172.97."+str(i))
```

- Using another new Pandas object called df_src_filter, we can filter out unwanted IP's like this:

```
df_src_filter = df[~df['src_ip'].isin(ip_array)]
```

- Once we have filtered out unwanted IP's, we can use the previous techniques to **group** and **plot** the data. Go ahead and do this for both **source IP's** and **destination IP's**. Note: do not filter out IP's from the destination IP's.

Question 3: When did these attacks occur?

- Lastly, let's take a look at attacks over time. Working with dates and times in data can sometimes be a bit of a pain, but luckily, the datetime package makes it much easier. To start, we have to convert our existing timestamps into datetime objects.

```
df['timestamp'] = pd.to_datetime(df['timestamp'])
```

- You've probably noticed that our timestamps are quite granular, going down to the specific second. For our purposes, this is a bit too specific. Rather than taking a substring, we can extract different date and time values and format it how we want. This command extracts the month, day, and year from our existing timestamps. Notice the capitalization.

```
df['timestamp'] = df['timestamp'].dt.strftime('%m/%d/%Y')
```

- Grouping this data is slightly different. First, create a new dataframe using only timestamps and one other existing column of your choice. After this, group the time data using this code:

```
df_q3 = df_q3.groupby([‘timestamp’]).count()
```

- From here, we can plot similar to what we have been doing. To plot the data over time as opposed to a bar graph, simply remove the "kind='bar'" argument from the plot. Feel free to open the PNG you've created and look at spikes in events that occurred on your network.

Points –provide the data requested including

- **Which IP is attacking you most?**
- **Which of your IPs is being attacked the most?**
- **On what day did most attacks occur?**
- **Your script file**
- **Your 3 graph image files**

Appendix D. Survey

If we were to use this module in the future, what things did you find useful/beneficial?

If we were to use this module in the future, what things would you change?

Which step did you find the most straightforward? The most difficult?

Appendix E. Analytics Exercise Steps

Activity	Commands provided	Minutes to complete
DAY 1 SECURITY ANALYTICS EXERCISE		
Retrieve data		
• Launch a terminal on your SecurityOnion machine. Run the provided command	Yes	1
• Send yourself the file via email	No	1
Setup environment for analytics		
• Retrieve your data file by downloading the file you sent yourself, and saving it to the desktop.	No	1
• Install the following packages using the provided command	Yes	3
• Identify the version of Python	Yes	1
• In the Windows search bar, type in "IDLE", run the version using the same version of Python that you found previously	Yes	1
• Under "File," create a new file, which will be our Python script	Yes	1
Analytics analysis		
• Import packages	Yes	3
• Import data	Yes	2
• Read data	Yes	1
• Filter data	Yes	1
• Correct IP addresses	Yes	1
Question 1: What types of attacks are being used?		
• Run a group by statement and get the count of the results (signature column)	Yes	2
• Plot a graph	Yes	2
DAY 2 SECURITY ANALYTICS EXERCISE		
Question 2: Which IP addresses (source and destination) appear more often than others?		
• Filter out unwanted IPs	Yes	2
• Group and plot the data	No	5
Question 3: When did these attacks occur?		
• Convert our existing timestamps into datetime objects	Yes	1
• Extracts the month, day, and year from our existing timestamps	Yes	1
• create a new dataframe and then group the time data	Yes	1
• Plot the data over time	No	10

Information Systems & Computing Academic Professionals

Education Special Interest Group

STATEMENT OF PEER REVIEW INTEGRITY

All papers published in the ***Journal of Information Systems Education*** have undergone rigorous peer review. This includes an initial editor screening and double-blind refereeing by three or more expert referees.

ISSN: 2574-3872 (Online) 1055-3096 (Print)