Project Tink: A Methodology in Simulating Network Attacks

Mark Simone Cabie[1], Justine Clarence Co [2], Pauline Joy Salgado[3], Samantha Isabel Segovia [2] and Jocelynn Cu[4]

Center for Network and Information Security, College of Computer Studies

[1]mark_simone_cabie@dlsu.edu.ph
[2]justine_co@dlsu.edu.ph
[3]pauline_salgado@dlsu.edu.ph
[4]samantha_segovia@dlsu.edu.ph
[5]jocelynn_cu@dlsu.edu.ph

Abstract: Simulation of network attacks could be useful in making a research regarding network security. Since it is best to set-up a network in a real environment to get more accurate results, we would want to test the setup first in a controlled environment. This will enable us to prepare for a real environment setup. In our work, we would be simulating network attacks and get the logs from the honeypot that is included in the setup for further use. To accomplish this, 3 physical machines will be used with 4 virtual machines each, representing the attackers. A honeypot will also be installed in another physical machine. The attackers would then attack the honeypot; concurrently, the attacks will then be logged from the honeypot. These logs will be PCAP files which will then be fed to Snort for processing.

Key Words: Machine learning; network attacks; network attack simulation.

1. INTRODUCTION

One of the tasks of a network security analyst is to study incoming network attacks and create reports. System administrators use the results and analysis from these reports to harden the system in order to protect it from future network attacks. Studies have been made to try and be a few steps ahead of hackers. These studies apply predictive analytics. Predictive analytics gives a competitive edge, and when applied in network security, could give administrators an advantage against attackers.

The work of J. Lei and Z. Li (2007) focuses on predicting multi-stage attacks. This study centers on dividing a whole network attack operation into several stages and when one stage is detected, how probable would it be that the next event or stage will occur after. Studies use machine learning and data mining techniques to get desired results. *Hybrid framework for behavioral prediction of network attack using honeypot and dynamic rule creation with different context for dynamic blacklisting* (R. Prasad & A. Abraham, 2010), *Characterizing attackers and attacks: an empirical study* (G. Salles-Loustau, R. Berthier, E. Collange, B. Sobesto, and M. Cukie, 2011), and *Fuzzy clustering and iterative relational classification for terrorist profiling* (J. Chen, J. Xu, P. Chen, G. Ding, R. F. Lax, and B. Marx, 2008) are studies that use machine learning techniques to create profiles for terrorists, and even network attackers. G. Xuang, X Wang, and L. Yin (2012) used different attack scenarios and an attack graph to visualize the sequence. They used Markov's chain to predict the next stage of attack and also based it on the probability score per stage. Their data set came from different attack sources and not just one main source. Time was not a major factor in this research because the main focus was not when the next attack is, but what attack will come next.

Internet bad neighborhoods temporal behavior (G. C. M., Moura, R. Sadre, & A. Pras, 2014) focused on the temporal behavior of network attacks, specifically the IP addresses. It also used DShield data. Focusing on the topic of blacklists, the study was about how often a blacklisted IP address appears, and if it does appear on the second day, how many days will it take to come back. The study used a 13 to 14-day period for observation and used different sources as well. The research did not only use DShield's data but also data gathered from their own university and other sources as well. However, their main focus was blacklists so they only focused on the IP addresses and the instances they appeared on the blacklist.

These studies, in order to be done, needed a good data set. Machine learning involves having a large enough data set to create models. These models will be used to classify and even predict outcomes based from data of the past. Researchers often use entries from DShield[1], which contains a large repository of known blacklisted IP addresses. These blacklisted IPs are submitted by different corporations and organizations, the IPs were detected by their sensors and results or logs were submitted to DShield. A major advantage in using DShield data is that it is free. Researchers may pull from their repository using different means. And it's large enough for analytical studies. But one disadvantage is that the data provided by DShield can only be used for certain areas of studies. The data may be large, but is not flexible enough to cater to different areas of research.

The study *New attack scenario prediction methodology* (S. Fayyad, 2013) has its own authentic data on hand. Existing (honeypots) and Intrusion Detection Systems (IDS) of universities or companies can be used. (Honeypots) are used to create a virtual network wherein its purpose is to trap, monitor and identify malicious requests within a network. Honeypots are usually used for gathering information about attackers and gaining insight into their attack methodology. Honeypots that are set up by corporations can have a large data set that can be used for different types of analysis. A good example of a honeypot is Honeyd which is a small daemon that creates virtual hosts on a network. An IDS is a means to detect suspicious or irregular activity on a system or network. The system is also responsible for gathering and storing important information regarding user and system activity to be analyzed when the need arises. (O. Hashmi, S. Sheikh, 2012)

Honeypot logs can give limited information (Figure 1) but if a packet sniffer like Wireshark[2] or TCPDump[3] is present in the network, it can capture the network activity in a packet capture file (PCAP) for further analysis or give a more detailed view of the packets. Packet sniffing software not only capture the network activity (either in verbose or host-only mode), but they are also used to analyze the captured network activity. These PCAP files can be sent through to IDS so the attacks done on the honeypot can be translated into actual attack names, severity, the source and the target. If a company or organization already has a honeypot and IDS set up long before the research has begun, this would be an ideal source for data. The two devices would have a large amount of data for analysis. The advantage of using company collected information is that the data is authentic and, if the company is well-known, the volume of the dataset would be large. Disadvantages would be that not every company or organization is willing to provide their data for privacy issues. Paperworks and contracts should be done before researchers can be given free reign over the data.

Creating simulated data is difficult since there is little to no written methodology on how to do it. S*imulating cyber-attacks for fun and profit* (A. Futoransky, et. al), *Noobz Guide for Setting Up a Vulnerable Lab for Pentesting* (INFOSEC Institute), and *Simulation of computer network attacks* (CoreLabs Core Security Technologies) discuss this topic, although these do not explicitly describe significant details such as the network set-up, and the method of collecting data. Another challenge is that it is difficult to generate attacks against a network set-up given a limited amount of time.

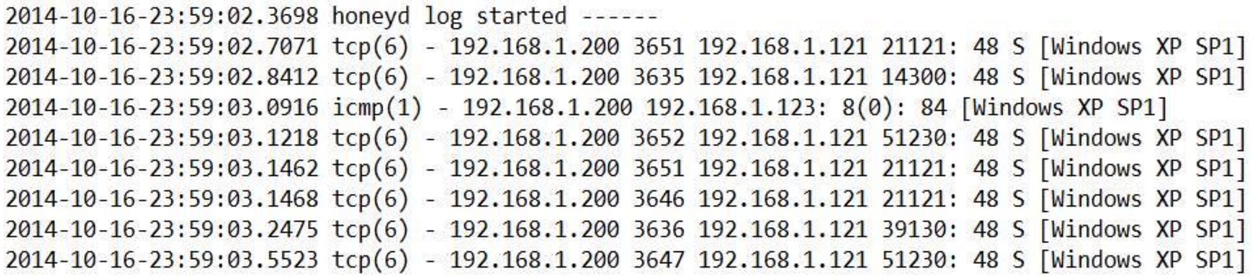
Running scripts instead of manually doing the attacks on the “attacker virtual machines (VMs)” is a good approach since it will speed up the process and save a lot of time, but another challenge is that it is hard to look for attack scripts that could be executed in the “attacker” VMs.

In this paper, Section 2 will discuss the methodology on how the simulation was set up. This includes the physical setup, the logical setup, and the simulation of attacks. Section 3 will be the discussion on the flow of the simulation and how data will be collected. And finally, as concluding remarks, Section 4 will discuss the on-going work.

[1] http://www.dshield.org

[2] http://www.wireshark.org

[3] http://www.tcpdump.org



```
2014-10-16-23:59:02.3698 honeyd log started ------
2014-10-16-23:59:02.7071 tcp(6) - 192.168.1.200 3651 192.168.1.121 21121: 48 S [Windows XP SP1]
2014-10-16-23:59:02.8412 tcp(6) - 192.168.1.200 3635 192.168.1.121 14300: 48 S [Windows XP SP1]
2014-10-16-23:59:03.0916 icmp(1) - 192.168.1.200 192.168.1.123: 8(0): 84 [Windows XP SP1]
2014-10-16-23:59:03.1218 tcp(6) - 192.168.1.200 3652 192.168.1.121 51230: 48 S [Windows XP SP1]
2014-10-16-23:59:03.1462 tcp(6) - 192.168.1.200 3651 192.168.1.121 21121: 48 S [Windows XP SP1]
2014-10-16-23:59:03.1468 tcp(6) - 192.168.1.200 3646 192.168.1.121 21121: 48 S [Windows XP SP1]
2014-10-16-23:59:03.2475 tcp(6) - 192.168.1.200 3636 192.168.1.121 39130: 48 S [Windows XP SP1]
2014-10-16-23:59:03.5523 tcp(6) - 192.168.1.200 3647 192.168.1.121 51230: 48 S [Windows XP SP1]
```

Figure 1. Honeyd honeypot logs as viewed from Notepad

2. METHODOLOGY

This study shows how to build a controlled environment for a simulation of network attacks against a corporate system. To achieve this, the setup must be able to represent attackers and specific targets within a network. Virtual machines are helpful in creating a network set-up as it removes the need of having physical machines to represent each attacker and target. Honeyd is a tool that allows the creation of multiple virtual hosts in a single machine.

2.1. Physical Setup

The physical set-up as shown in Figure 2 is comprised of four computer machines: 3 attacker machines and 1 machine for the honeypot. Each attacker machine runs 4 virtual machines that represent 2 script kiddies (SK) and 2 determined attackers (DA) as shown in Figure 3.

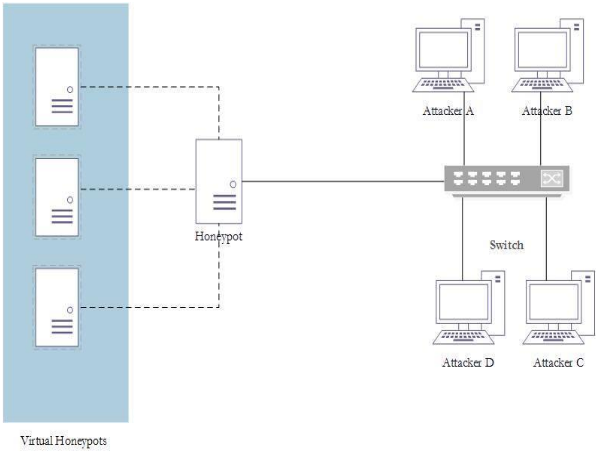


Figure 2. Physical Topology Setup

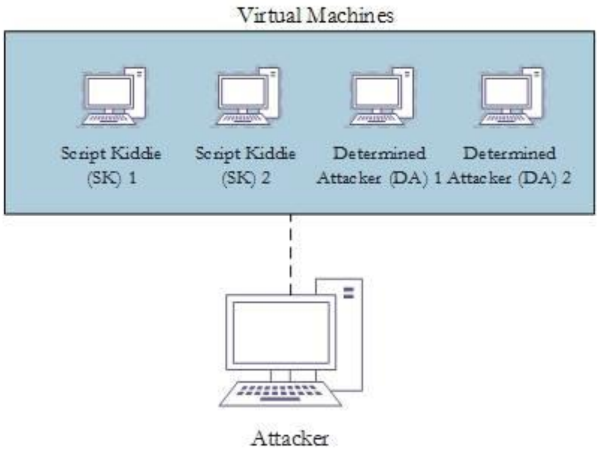


Figure 3. Attacker Machine Setup

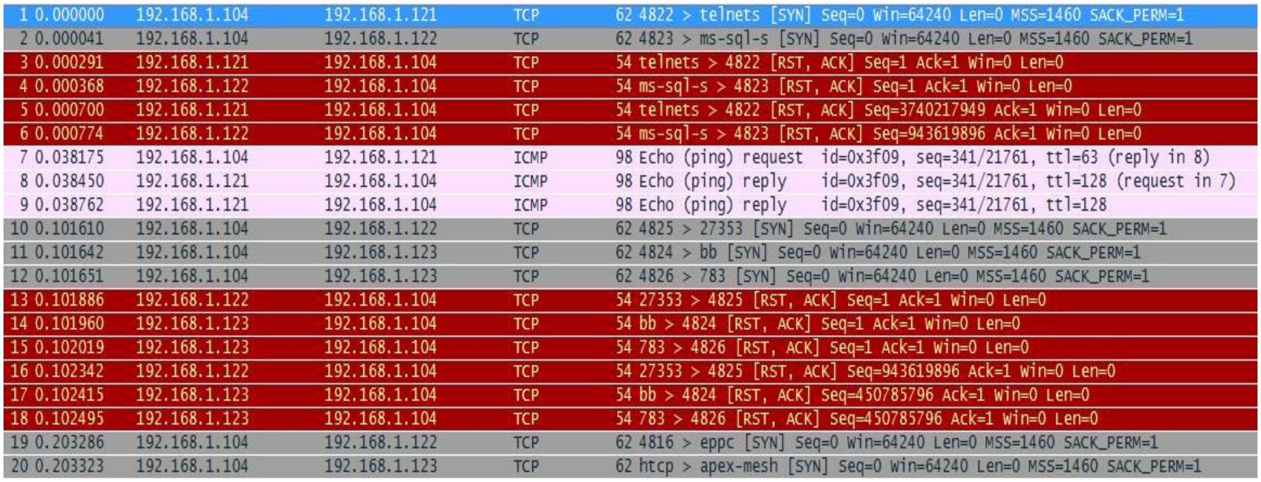
2.2. Logical Setup

SK virtual machines use the Windows XP operating system, while DA virtual machines use the Backtrack[4] operating system. The honeypot machine uses an Ubuntu 12 server as its operating system. Table 1 discusses the VMWare specifications of each virtual machine.

Table 1. VM Hardware Settings

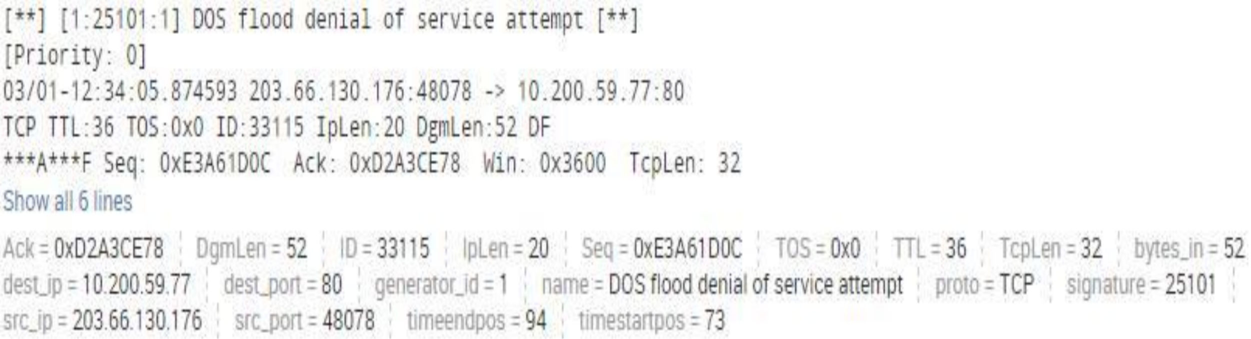
Hardware	Setting
Memory	128 MB
Processor	1
Hard Disk	40 GB

[4] http://www.backtrack-linux.org



```
 1 0.000000  192.168.1.104  192.168.1.121  TCP   62 4822 > telnets [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM=1
 2 0.000041  192.168.1.104  192.168.1.122  TCP   62 4823 > ms-sql-s [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM=1
 3 0.000291  192.168.1.121  192.168.1.104  TCP   54 telnets > 4822 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
 4 0.000368  192.168.1.122  192.168.1.104  TCP   54 ms-sql-s > 4823 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
 5 0.000700  192.168.1.121  192.168.1.104  TCP   54 telnets > 4822 [RST, ACK] Seq=3740217949 Ack=1 Win=0 Len=0
 6 0.000774  192.168.1.122  192.168.1.104  TCP   54 ms-sql-s > 4823 [RST, ACK] Seq=943619896 Ack=1 Win=0 Len=0
 7 0.038175  192.168.1.104  192.168.1.121  ICMP  98 Echo (ping) request  id=0x3f09, seq=341/21761, ttl=63 (reply in 8)
 8 0.038450  192.168.1.121  192.168.1.104  ICMP  98 Echo (ping) reply    id=0x3f09, seq=341/21761, ttl=128 (request in 7)
 9 0.038762  192.168.1.121  192.168.1.104  ICMP  98 Echo (ping) reply    id=0x3f09, seq=341/21761, ttl=128
10 0.101610  192.168.1.104  192.168.1.122  TCP   62 4825 > 27353 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM=1
11 0.101642  192.168.1.104  192.168.1.123  TCP   62 4824 > bb [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM=1
12 0.101651  192.168.1.104  192.168.1.123  TCP   62 4826 > 783 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM=1
13 0.101886  192.168.1.122  192.168.1.104  TCP   54 27353 > 4825 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
14 0.101960  192.168.1.123  192.168.1.104  TCP   54 bb > 4824 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
15 0.102019  192.168.1.123  192.168.1.104  TCP   54 783 > 4826 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
16 0.102342  192.168.1.122  192.168.1.104  TCP   54 27353 > 4825 [RST, ACK] Seq=943619896 Ack=1 Win=0 Len=0
17 0.102415  192.168.1.123  192.168.1.104  TCP   54 bb > 4824 [RST, ACK] Seq=450785796 Ack=1 Win=0 Len=0
18 0.102495  192.168.1.123  192.168.1.104  TCP   54 783 > 4826 [RST, ACK] Seq=450785796 Ack=1 Win=0 Len=0
19 0.203286  192.168.1.104  192.168.1.122  TCP   62 4816 > eppc [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM=1
20 0.203323  192.168.1.104  192.168.1.123  TCP   62 htcp > apex-mesh [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM=1
```

Figure 4. PCAP file opened in Wireshark



```
[**] [1:25101:1] DOS flood denial of service attempt [**]
[Priority: 0]
03/01-12:34:05.874593 203.66.130.176:48078 -> 10.200.59.77:80
TCP TTL:36 TOS:0x0 ID:33115 IpLen:20 DgmLen:52 DF
***A***F Seq: 0xE3A61D0C  Ack: 0xD2A3CE78  Win: 0x3600  TcpLen: 32
Show all 6 lines

Ack = 0xD2A3CE78 | DgmLen = 52 | ID = 33115 | IpLen = 20 | Seq = 0xE3A61D0C | TOS = 0x0 | TTL = 36 | TcpLen = 32 | bytes_in = 52
dest_ip = 10.200.59.77 | dest_port = 80 | generator_id = 1 | name = DOS flood denial of service attempt | proto = TCP | signature = 25101 |
src_ip = 203.66.130.176 | src_port = 48078 | timeendpos = 94 | timestartpos = 73
```

Figure 5. Snort alert as displayed in Splunk

Network Adapter	Bridged

Honeyd is an open-source virtual honeypot used in this research. An arbitrary number of hosts was created inside the machine to serve as targets for the attacks. This is to provide a variety of IP addresses as targets. The hosts emulate different operating systems and each may claim a specific IP address. There were three hosts configured in this set-up, emulating the operating systems of Microsoft Windows XP SP1, Linux 2.4.20, and Microsoft Windows 2000 Server SP2.

2.3. Attack Simulation

Three physical machines each containing 4 virtual machines (2 of which are the SKs while the remaining are DAs) are used to attack the honeypot. First, a ping test was done to check if the honeypot was already up and running. In the Windows VMs, a continuous ping was used. The command was:

```
ping -t [target IP address]
```

For the Backtrack VMs, a pingsweep command was used to check which IP addresses were up. The command was:

```
fping -a -g [first target IP address] [last target IP address]
```

The "-a" is so the output will only include the live IP address to make the report clean and much easier to read. The "-g" is used to specify which range of IP addresses will be scanned.

To generate a Denial of Service (DoS) via the command prompt (CMD) in the Windows VMs, the command used is:

```
ping [target IP address] -t -l [buffer size]
```

The "-t" will make the ping continuous until the command is halted. The "-l" is to indicate the size of the buffer which is from 0 to 65500.

A port scan attack was done using the tool Zenmap in Backtrack. Zenmap is a UI-based tool in Backtrack that can execute ping sweeps and port scans, which can range from regular scans to noisy, 'intense scans'. These attacks were detected by the honeypot Honeyd and were saved as log files. TCPdump and honeyd were run simultaneously to collect log and pcap files during on-going attacks.

3. RESULTS AND DISCUSSION

Tcpdump produces a pcap file containing the attacks made against the virtual honeypots. The file contains the whole network activity of the setup and can be read through packet sniffer software such as Wireshark and IDSs such as Snort[5]. Figure 4 shows the pcap file opened in Wireshark.

The file will be fed to Snort, an IDS, to produce a csv file. The csv file will then be fed to Splunk in order to separate the date and time into more columns, so that the model will be able to process the data set. The data will be annotated to see which attack levels were detected. Attack levels include Scanning, Footprinting, Enumeration, Exploitation, and Maintaining Access. Once attack levels are identified per attack, attacker type may now be determined. Weka and RapidMiner cannot process IP addresses so the source and destination IP addresses will be converted to their corresponding countries. Figure 5 shows a Snort log when viewed from Splunk.

The physical setup can be validated through the use of ping. If the ping was successful then the machines are able to communicate with each other which means that the setup is working properly. Since the setup is now working properly then the attackers, SK and DA, can now attack the honeypot.

We chose to use VMs in representing the attackers and the honeypot so that there will be no need to have a physical machine for every SK and DA. In doing so, we will only be managing 4 physical machines. VMs allow the use of different operating without needing different physical machines for the different operating systems. Also, an advantage of using VMs is that it is cheaper than having physical machines for each attacker and for the honeypot.

4. CONCLUSIONS

The simulated data created will be used as training and testing data in creating the model. The model will go through the process of 10-fold cross validation. 10-fold cross validation is the process of getting 9/10 of the data set to use as training data while, the remaining will be used as testing data (P. Refaeilzadeh, L.Tang, H. Lu). This process will be repeated until all parts of the data set have been used as training and testing data.

According to *Pattern of life and temporal signatures of hacker organizations* (2013), there is a temporal signature of activities that could differentiate attackers. In knowing the temporal signature, the work schedule and even the location of the attackers can be learned. Script kiddies are mostly teenagers, and it is possible that they only attack during their free time or during the weekends. In creating the simulated data for the attacker type script kiddies, we will be running the attacks only during our respective free times. Also, in order to make the simulated data more authentic there will be a case where both script kiddies and determined attackers are going to attack at the same time.

Simulation of attacks will continue in order to gather more data to use for the training and testing of models.

5. ACKNOWLEDGMENTS

[5] http://www.snort.org

We would like to first thank the Center for Network and Information Security for providing us with the resources for our study. We would also like to thank Mr. Ravi Kumar for being our initial adviser and for guiding us in creating our proposal, along with our panelists, Mr. Miguel Alberto N. Gomez, Ms. Rhia S. Trogo, Mr. Alexis V. Pantola and Ms. Kaye Solomon for their continuous support and guidance, comments and insights on the study. Finally, we would like to thank Mr. Isaac Herculano S. Sabas for allowing us to consult with him with regards to network security, Dr. Remedios D. Bulos for giving us her time for consultation regarding data mining and machine learning, and Mr. Rafael Carlo C. Sabaulan for consultations regarding Time Series Analysis.

6. REFERENCES

Abraham, A., & Prasad, R. (2010) Hybrid framework for behavioral prediction of net-work attack using honeypot and dynamic rule creation with different context for dynamic blacklisting. Paper presented at 2010 Second International Conference on Communication Software and Networks, Singapore. doi: 10.1109/ICCSN.2010.82

Berthier, R., Collange, E., Cukier, M., Salles-Loustau, G., & Sobesto, B. (2011). Characterizing attackers and attacks: an empirical study. Paper presented at the 2011 17th IEEE Pacific Rim International Symposium on Dependable Computing, Pasadena CA. doi: 10.1109/PRDC.2011.29

Chen, J., Xu, J., Chen, P., Ding, G., Lax, R. F., & Marx, B. (2008). Using network attack graph to predict future attacks. Paper presented at the IEEE International Conference on Granular Computing, Hangzhou. doi: 10.1109/GRC.2008.4664739

Fayyad, S. (2013). New attack scenario prediction methodology. Paper presented at 2013 10th International Conference on Information Technology: New Generation. doi:10.1109/ITNG.2013.16

Futoransky, A., Miranda, F., Orlicki, J., Sarraute, C. (2009). Simulating cyber-attacks for fun and profit. Paper presented at 2nd International Conference on Simulation Tools and Techniques, Belgium. doi:10.4108/ICST.SIMUTOOLS2009.5773

Hashmi, O., Sheikh, S. (2012). Impact of social attributes on predictive analytics in telecommunication industry. Paper presented at 2012 15th International Multitopic Conference. doi: 1109/INMIC.2012.6511470

Kuang, G., Wang, X., Yin, L. (2012). A fuzzy forecast method for network security situation based on Markov. Paper presented at 2012 International Conference on Computer Science and information Processing. doi: 10.1109/CSIP.2012.6308971

Lei, J., & Li, Z. (2007). Using network attack graph to predict future attacks. Paper presented at the Second International Conference on Communications and Networking, China. doi: 10.1109/CHINACOM.2007.4469413

Lu, H., Refaeilzadeh, P., Tang, L. (2008). Cross-validation. Retrieved October 15, 2014 from http://leitang.net/papers/ency-cross-validation.pdf

Miranda, F., Orlicki, J., Sarraute, C. (2010). Simulation of Computer Network Attacks. arXiv:1006.2407 [cs.CR]

Moura, G. C. M., Pras, A., & Sadre, R (2014). Internet bad neighborhoods temporal behavior. Paper presented at IEEE/IFIP NOMS 2014: Network Operations and Management Symposium Management in a Software Defined World, Poland. doi: 10.1109/NOMS.2014.6838306

Turla, J. (2012, September 19). Noobz Guide for Setting up a Vulnerable Lab for Pentesting. Retrieved October 13, 2014, from http://resources.infosecinstitute.com/hacking-lab/

(N.A.) (2013). Pattern of Life and Temporal Signatures of Hacker Organizations. Retrieved October 13, 2014 from https://www.recordedfuture.com/temporal-signatures-of-hacker-organizations/