Human Perception of the Measurement of a Network Attack Taxonomy in Near Real-Time

Renier van Heerden[1], Mercia M. Malan[2], Francois Mouton[1], and Barry Irwin[3]

[1] Defence Peace Safety & Security, Council for Industrial and Scientific Research,
Pretoria, South Africa
rvheerden@csir.co.za, moutonf@gmail.com

[2] University of Pretoria, Information and Computer Security Architecture Research Group,
Pretoria, South Africa
malan747@gmail.com

[3] University of Rhodes, Computer Science Department, Grahamstown, South Africa
b.irwin@ru.ac.za

Abstract. This paper investigates how the measurement of a network attack taxonomy can be related to human perception. Network attacks do not have a time limitation, but the earlier its detected, the more damage can be prevented and the more preventative actions can be taken. This paper evaluate how elements of network attacks can be measured in near real-time(60 seconds). The taxonomy we use was developed by van Heerden et al (2012) with over 100 classes. These classes present the attack and defenders point of view. The degree to which each class can be quantified or measured is determined by investigating the accuracy of various assessment methods. We classify each class as either defined, high, low or not quantifiable. For example, it may not be possible to determine the instigator of an attack (Aggressor), but only that the attack has been launched by a Hacker (Actor). Some classes can only be quantified with a low confidence or not at all in a sort (near real-time) time. The IP address of an attack can easily be faked thus reducing the confidence in the information obtained from it, and thus determining the origin of an attack with a low confidence. This determination itself is subjective. All the evaluations of the classes in this paper is subjective, but due to the very basic grouping (High, Low or Not Quantifiable) a subjective value can be used. The complexity of the taxonomy can be significantly reduced if classes with only a high perceptive accuracy is used.

Keywords: Network Attack, near real-time, Network Attack Taxonomy.

1 Introduction

Network attacks do not have a time limitation, but the earlier its detected, the more damage can be prevented and the more preventative actions can be taken.

This paper builds on a previous taxonomy, the taxonomy is used a base to measurement of a network attack taxonomy in near real-time. The taxonomy we use was

K.K. Kimppa et al. (Eds.): HCC11 2014, IFIP AICT 431, pp. 280–292, 2014.

developed by van Heerden et al (2012) with over 100 classes. These classes present the attack and defenders point of view. The degree to which each class can be quantified or measured is determined by investigating the accuracy of various assessment methods.

We classify each class as either defined, high, low or not quantifiable. For example, it may not be possible to determine the instigator of an attack (Aggressor), but only that the attack has been launched by a Hacker (Actor). Some classes can only be quantified with a low confidence or not at all in a sort (near real-time) time. The IP address of an attack can easily be faked thus reducing the confidence in the information obtained from it, and thus determining the origin of an attack with a low confidence. This determination itself is subjective.

All the evaluations of the classes in this paper is subjective, but due to the very basic grouping (High, Low or Not Quantifiable) a subjective value can be used. The complexity of the taxonomy can be significantly reduced if classes with only a high perceptive accuracy is used. The taxonomy refers to both the view of the attacker and the defender with relation of network attacks, whereas other taxonomies concentrate on either attack or defend.

Table 1 is a recap the network attack taxonomy developed in van Heerden (2012) [1, 2].

Table 1. Taxonomy classes

Class	Description
Actor	The group or individual that is performing the action of an attack.This class points to the entity that physically performs the attack, not to the person instigating the attack.
Actor Location	The physical location, such as a country or a state, from where an attack was launched.
Aggressor	The mastermind behind the attack. This person, group or organisation can be the actor or can instruct the actor to attack a network. For example, Brenner suggested that France, Russia, Japan, China, Germany, Israel and South Korea are actively engaged in economic espionage by means of the Internet and computer network attacks [3, 4].
Asset	The non-personalised item that is under attack. This class distinguishes between different assets that can be attacked. Examples of assets are information stored as data, the system that uses computers or the network infrastructure itself.
Attack Goal	The goal that the Aggressor wants to achieve. The first four goals correspond with the traditional CIA$+$+ information security principles (Confidentiality, Availability, Integrity Authentication).
Attack Mechanism	The approach used in the attack. This approach refers to the methodology that was used in the attack. This class has over thirty sub-classes which describes some of the methodologies available.

Table 1. (*continued*)

Attack Scenario	The broad categories which attacks can belong to. Further explained below.
Automation Level	The degree that the attack can be programmed automatically beforehand compared to the amount of manual effort required during the attack.
Effects	The impact of an attack and are classified as four levels: Null, Minor, Major and Catastrophic. "Null" refers to no effect on the target, "Minor" to recoverable damage and "Major" to non-recoverable damage. "Catastrophic" refers to damage of such a nature that the target ceases to operate as an entity, for example the declaration of bankruptcy.
Motivation	The incentive for the attack.
Phase	The temporal stages of an attack. These temporal phases are developed by evaluating the most commonly used phases during an attack [5, 6].
Sabotage	The form of damage or loss that has been achieved by the attack and is classified as either physical, financial, virtual or reputational. *Physical* sabotage refers to physical damage to a device, *Financial* sabotage refers to monetary loss. *Virtual* sabotage occurs when computer resources are lost (such as processing, bandwidth or memory). *Reputational* loss is not a measurable, tangible loss but may result in other related problems for a company at a later stage.
Scope	The network type that is attacked and are further classified into three types: Corporate network, government network and private network. *Corporate networks* are networks controlled by private companies, *Government networks* are networks controlled by the government and *Private networks* are networks that serves one person in his/her private capacity.
Scope Size	The size of the network under attack. If the attack affects a large portion of the internet or multiple countries, the scope size is referred to as a *Global network*. A *Large network* represent large corporates or significant government networks such as state departments. There are no solid definitions that separate small, medium and large networks and thus the separation is a subjective judgement. *Single* size is used to present attacks on a single person or a single computer.
Target	The *Target* The physical devices targeted in the attack, such as a *Server, Personal computer, Network infrastructure* or *SCADA*.
Vulnerability	The methodology of the attack and denotes the weakness exploited in the attack.

The *Attack Scenario* class was originally developed by van Heerden (2012) [2, 1]. These scenarios are:

- Denial of Service,
- Industrial Espionage,
- Web Defacement,
- Unauthorised Data Access,
- Financial Theft,
- Industrial Sabotage,
- System Compromise,
- Cyber Warfare and
- Runaway Malware.

2 Taxonomy Quantification

Each of the classes and sub-classes can either be directly or indirectly quantified. They can also be defined by the system under attack's configuration. Some classes cannot be quantified in a near real-time environment. There are three levels of quantification. This paper describes in which of the three levels of quantification each class is classified when measured. Some classes are not measured but defined by the nature of the attack and thus an accuracy of the quantification is assigned to it. Four levels of accuracy are assigned: High, medium, low or not quantifiable.

Fenz et al states [7]:

”Since the threat probability or in influencing factors cannot be determined quantitatively, a qualitative rating is used in this approach. In contrast to a quantitative rating with which it is hardly possible to determine the occurrence of a certain threat with a 67% and not with a 68% chance, a qualitative rating (e.g.high, medium, and low)”

Since the accuracy of quantifying the classes are also defined, the researcher effectively uses three levels of qualitative ratings.

Each of the classes defined in Section ? are investigated as to how they can be measured or quantified in near real-time. Some of the classes are quantified by definition, and do not require any sensors to determine their value and these classes are referred to as: Defined. For example, the target of an attack is not measured or quantified, but rather defined by the attack.

Some attacks are named after the target, such as the cases of the SCO and SpamHaus attacks. Some classes can not be measured in near real-time environment. These class's values only become apparent long after an attack and even then it is sometimes only speculation. For example, the *Aggressor* can not be determined in a near real-time environment and for some attacks the real power behind the attack is never determined or proven.

2.1 Actor Quantification

The *Group Actor* sub-class and its sub-classes, *Organised Criminal Group*, *Protest Group* and *Cyber Army*, can be quantified by their IP addresses. An IP address can be used to find the physical location of a *Group Actor*. Free and subscription geolocation databases exist which claim to be capable of identifying the physical location of any IP address worldwide and the lookup of IP addresses is thus considered a direct quantification. By utilising the gained IP location, the group that owns or rents the location can be determined. The *Group Actor* can be determined indirectly by using the IP address.

Shavitt studied the accuracy of geolocation databases and found that the results of most databases are similar, that the accuracy cannot be trusted [8]. Errors included wrongful estimation of distances and incorrect identification of the country. IP addresses can be spoofed, and intermediate computers located anywhere in the world can be used for attack.

For these reasons, using IP to locate the Group Actor is assigned a low accuracy. IP address location falls within the Network Layer.

The *Hacker* sub-class and its sub-classes *Script Kiddy* and *Skilled Hacker* can be quantified by looking at the pattern of an attack. Stoll [9] documented one of the first hacking attempts by a skilled hacker. It was determined that the hacker was extremely skilled by printing out all the keystrokes of the attack. Script Kiddies use standardised tools of which the characteristics (or fingerprint) are static and can be identified. For example, the pattern of standard Nmap scans can easily be identified [10]. By using an elaborate honeypot, the skill level of a *Hacker Actor* can also be determined [11]. Script Kiddies will attack the honeypot directly with standard tools such as Metasploit[1] with all the possible exploits, whereas skilled hackers will use more subtle techniques and only targeted exploits and also try to hide their origin [12].

The skill level of hackers can also be deduced by the consequences of their attacks. If the attack was successful in web defacement or a secure server was compromised, it can be assumed that a skilled hacker was involved. Tripwire[2] and other Host-based Intrusion Detection Systems (IDS's) can alert system administrators to compromises, although they can not prevent attacks. They notify administrators that some secure data has been accessed or modified.

For these reasons, the Hacker Actor can be measured indirectly, and the accuracy is low. Honeypot measurements fall within the Session Layer. Host-based IDS's work in the Application Layer.

Insider threats can be detected by internally-orientated honeypots or telescopes [13, 14]. These insider honeypots work according to the same principle as externally-orientated honeypots, but reside within a network and are not accessible from outside. Externally-orientated honeypots are connected to external networks and capture traffic from attackers from outside the scope of the defender's network. Insider honeypots can detect a *Normal User* but not an *Administrator*.

Administrators have access to most of the network. No network can be made safe against its own administrators, and thus administrators fall within the unmeasurable group whereas normal users can be measured directly. When such honeypots are triggered, the odds of it being an insider is low due to possible false positives or attackers

[1] `http://www.metasploit.com/`

[2] `http://www.tripwire.org/`

masquerading as insiders. As previously stated, Honeypot measurements fall within the Session Layer.

All the sub-classes for the*Actor* class have a low accuracy, thus is summary, the *Actor* class accuracy in defined as low.

2.2 Actor Location Quantification

The *Actor Location* class and its sub-classes can be measured similarly to the *Group Actor* sub-class by means of IP location. Only a single look-up in a geolocation database is required, thus it is considered to be directly measurable. The values from geolocation database are also considered unreliable, with Poese stating that these geolocation databases are accurate at a country level but not at a city level [15]. An alternative method to find the location of IP addresses is to use latency measurements [16]. Katz was able to achieve a medium error of 67 km in optimal circumstances. The same accuracy problems as stated for the *Group Actor* apply to the *Actor Location* sub-classes and thus the accuracy is considered to be low. As mentioned above, an IP address falls within the Network Layer.

2.3 Aggressor, Motivation, Effect and Sabotage Quantification

The *Aggressor* cannot be quantified in near real-time. In most cases the aggressor is only determined months after an attack. For example, it took a few months before the aggressor behind the Stuxnet attack was confirmed [17]. The aggressor and people behind most viruses is difficult if not impossible to obtain [18].

The *Aggressor* class and its sub-classes are considered as being not quantifiable. The same holds for the motivation of an aggressor, which can also not be determined in near real-time. The type of sabotage caused by an attack can only be calculated after the full impact of an attack is known, and thus can to be measured in real-time. The effects of an attack can only be quantified with a full investigation into the compromised systems and assessments of the damage done. This means that the *Aggressor*, *Effect*, *Motivation* and *Sabotage* class and its sub-classes cannot be measured in near real-time.

2.4 Asset Quantification

The *Access* and *System* sub-classes of the *Asset* class can be measured with automated testing scripts. These testing scripts simulate human requests at a very basic level, and can indicate when access to the system, or the system functionally have been altered. Stout stated that automated testing is critical to a quality website and his statement holds true for all servers [19]. The scripts directly measure access and the system's functionality, and the accuracy of these quantifications are regarded to be as high, on the Application layer.

The *Data* sub-class of *Asset* class can be quantified by host-based IDS. These sensors are capable of determining alterations to data. Typically, two main aspects of the data can be measured, namely unauthorised access or unauthorised manipulation of the data [20]. These quantification are direct and occur in the Application layer. Although there is a possibility for false alarms [21], these quantifications are considered to have a high accuracy.

The *Network* sub-class of the *Asset* class can be quantified indirectly by looking the networking performance of devices or testing whether systems in the network can communicate [22]. Communication errors, hardware breakdown or system misconfiguration can be possible reasons for disruption of communications. The accuracy of quantifying an attack on the network is regarded as high. Network communications are on the Network Layer.

All the sub-classes for *Asset* class have a high accuracy, and thus *Asset* class accuracy is defined as high.

2.5 Attack Goal Determination

The *Attack Goal* can be determined indirectly by finding the asset which is under attack. Similar to the *Data* sub-class of the *Asset* class, the *Destroy Data*, *Steal Data*, *Gain Control*, *Spread* and *Change Data* sub-classes can be determined by host-based IDS [20]. The *Disrupt* sub-class can be determined indirectly by looking at the type of attack that is launched on a honeypot or similarly to the *Network* sub-class of the *Asset* class, by monitoring network performance [20, 23].

The accuracy of determining the goal is considered to be high. These determinations occur in the Application layer, and on the Network layer if the disruption is with communication.

The *Gain Resources* sub-class of the *Attack Goal* class can be determined by intercepting communications that do not fit the normal profile. Strayer developed a system that identifies networks that support malicious traffic[3]. Thus malicious traffic bound for addresses listed in their system can be null-routed. The Finding Rogue Network project has since been discontinued, but similar work is done commercially by Lastline[4]. It can be determined if local systems are being used as a springboard for attacks on others. Since this determination depends on the accuracy of the identification of malicious networks, and the possibility of misconfigured networks looking like botnets, the determination of *Gain Control* are considered of low accuracy.

This determination uses IP addresses as references, thus it occurs in the Network Layer.

Since five out of the six sub-classes have a high accuracy, the *Attack Goal* class accuracy is defined as high.

2.6 Attack Mechanism Determination

The *Information Gathering* sub-class of the *Attack Mechanism* class can be indirectly measured by detecting scans. These scans can be detected by interpreting access logs or analysing network traffic [24]. Port scanning and vulnerability scan determination

[3] `http://maliciousnetworks.org/`

[4] `http://www.lastline.com/`

have a high accuracy rate. Port scanning is determined on the Network and Transport layers and have a high accuracy.

The *Brute Force*, *Escalation*, *Spoofing*, *Session Hijack* and *Buffer Overflow* sub-classes can be identified by network-based IDS and by looking at access logs. These are directly identified by matching known methods to observed events. These events are defined in the Application layer. The accuracy of identifying these attacks mechanisms are high.

The *Spear Phishing* and *Social Engineering* sub-classes can be identified by specially crafted traps that lure such attackers to a fake target [25, 26]. These traps operate in the Application layer. Due to the difficulty of detecting social engineering attacks, to determine that one of these attack mechanisms was used has a low confidence [27, 28, 29].

The *Network Based* sub-class can be identified indirectly by intercepting strange communications or by monitoring the amount of traffic on the system [30]. These strange communications can be occur in Data, Network or Session layers. Although it is difficult to distinguish between attacks and innocent network anomalies, it is simple to detect and thus the accuracy is high.

The *Malware* attack mechanism can be identified either on the OSI Application layer with Antivirus software, or in the OSI Network layer by IDS software [31]. Malware can be identified directly and the accuracy of the identification is high with a low false positive rate. False positive is when a classifier classifies some item as harmful incorrectly [32]. Malware that is not detectable is also a concern [31]. False negative refers to malware that is not detected. In Figure 1 the difference between False Negative and False Positive is shown. The detection of *Malware* has a high accuracy.

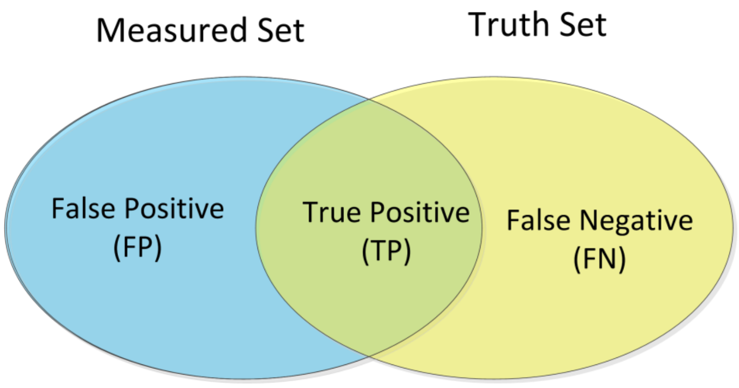

Fig. 1. The Difference between False Negative and False Positive

If a sub-system is abused, it can be measured simply by looking at at systems logs. The processing utilisation and disk usage can be measured on systems directly. Firewalls and some advanced routers can measure the network throughput, thus identifying bandwidth abuse. The detection of *System Abuse* has a high accuracy.

A *Web Application* such as a *SQL Injection* or a *Web Crawl* attack mechanism can be detected directly with specially crafted traps or logging of unusual web behaviour [33]. Error messages can also be used to detect *SQL Injection* attacks [34]. SQL queries can be used to identify safe request and then identity attacks by restricting the allowed queries [35]. Misuse of Web applications are detected in the Application layer and has a high accuracy level.

XSS Web Application attack mechanisms can be detected indirectly by comparing posted URL to black-listed sites [36], by identifying typical XSS coding patterns [37]. The detection and prevention of XSS attack are difficult because of incomplete implementations, inherent limitations, the complexity of development frameworks and the requirement for run-time compatibility [38]. The efficiency of this detection method is determined by the quality of the blacklist and accuracy level is low and is detected in the Presentation layer.

Denial-of-Service attack mechanisms can mostly be detected by filtering incoming network traffic [39]. Mirkovic presented a taxonomy in defences that can be used against DoS attacks, which includes: system security, protocol security, resource accounting, resource multiplication, pattern matching, anomaly detection, filtering, automated reconfiguring, rate limiting and agent identification [40]. The accuracy of detecting DoS attack mechanisms is high.

Since most of the sub-classes of the *Attack Mechanism* class have a high accuracy, the accuracy for the *Attack Mechanism* class is high.

2.7 Automation Level Quantification

The *Automatic* sub-class of the *Automation Level* class can be indirectly quantified by observing the scanning pattern and other features with honeypots and other scan detection sensors. Kuwatly was able to detect Nmap[5] scans by training their detection systems to recognise Nmap specific scan characteristics. Similarly it should be possible to detect automated tools by their specific behaviour. A lack of automation can point to the *Manual* or *Semi-automatic* automation level. The accuracy level for these quantifications are low, since the difference between automation and the other modes are difficult to determine and thus difficult to quantify and use data from the Network and Transport layers.

2.8 Phase Classification

The *Phase* sub-classed can only be measured indirectly. The *Target Identification* and *Reconnaissance* sub-classes can be identified by intercepting scans. These scans can be IP or Port based, or scans that crawl through web pages. The accuracy of determining that an attack is in the reconnaissance phase is high after a scan has been detected. These scans occur in the Network, Transport, or Session layers.

[5] `http://nmap.org`

The "Ramp-up" sub-class can be identified with honeypots or anomaly detectors that identify strange communications or attempts at obtaining unauthorised access. These measurements occur in the Application layer and have a medium accuracy.

The "Damage" sub-class is measured when the network, computers or systems stop working according to specifications. Host-based IDS can determine if the network is currently being damaged. It is difficult to determine if damage is being caused by an attack or by some other unrelated error, thus the accuracy is medium. Damage can be measured in the Network or Application Layers.

The "Residue" sub-class can be detected by communications similar to these in the Damage phase, but only on a limited scale. These measurements can also be influenced by other system errors and thus are considered to be of medium accuracy. Residue can be measured in the Network or Application Layers.

The "Post-Attack" sub-class can be measured by detecting unallocated communications after an attack. These are typically IP connections to and from unknown hosts at strange times. These measurements take place in the Network Layer. It is difficult to prove that the same attacker is snooping around again, thus these measurements are considered to be of low accuracy. Phase on average is of medium accuracy.

2.9 Scope and Scope Size Measurement

The target scope and the scope size are defined by the entity under attack. These classes represent physical attributes of the target, which can not be measured or quantified, but rather should be considered to be defined.

2.10 Target Monitoring

The *Target* class and its sub-classes can be monitored indirectly by observing which systems are not performing as expected.

The *Network Infrastructure* sub-class can be observed by monitoring network performance in the Network layer. Attacks that affect the *PC* sub-class can be observed with anti-virus software. Antivirus software monitor data in the Network, Session and Application layers. The *Server* sub-class can be monitored by heart-beat sensors or data integrity sensors [41]. These sensors monitor on the Application layer.

Industrial Equipment are monitored directly via their control software [42]. Industrial equipment can monitor communications in the Physical, Network and Application layers. Even though system problems or other errors can also lead to system failures, monitoring these classes are considered to be highly accurate.

2.11 Vulnerability Identification

The *Vulnerability* class and its sub-classes can be directly identified with a combination of IDS and honeypots [43]. Although IDS can have false positives (incorrectly identify attacks), their accuracy are considered to be high.

3 Conclusion

In Table 2, a summary of the required quantification is shown. This table lists all the classes with respect to quantification methodology and accuracy. Only five of the classes is considered quantified or measurable of high accuracy: *Asset, Target, Vulnerability, Attack Mechanism* and *Attack Goal*. Three classes are considered low: *Automation Level, Actor* and *Actor Location*. Four classed can not be measured or quantified in a near real-time environment: *Sabotage, Effect, Aggressor* and *Motivation*. The remaining two classes are defined: *Scope* and *Scope Size* and OSI layer within which the classes are quantified. The quantification are tabulated with respect to OSI layer and Accuracy level. By only considering which elements of the taxonomy can be measurement of a network attack taxonomy in near real-time. Thus by only this reduces elements are of use when looking at attacks in near-real time

Table 2. Summary of the Measurement Taxonomy

Class	Quantification	Accuracy
Actor	Indirect	Low
Actor Location	Direct	Low
Aggressor	Not Quantifiable	N/A
Asset	Direct	High
Attack Goal	Indirect	High
Attack Mechanism	Indirect	High
Automation Level	Indirect	Low
Effect	Not Quantifiable	N/A
Motivation	Not Quantifiable	N/A
Phase	Indirect	Medium
Sabotage	Not Quantifiable	N/A
Scope	Defined	N/A
Scope Size	Defined	N/A
Target	Indirect	High
Vulnerability	Direct	High

References

1. van Heerden, R., Pieterse, H., Irwin, B.: Mapping the most significant computer hacking events to a temporal computer attack model. In: Hercheui, M.D., Whitehouse, D., McIver Jr., W., Phahlamohlaka, J. (eds.) ICT Critical Infrastructures and Society. IFIP AICT, vol. 386, pp. 226–236. Springer, Heidelberg (2012)
2. van Heerden, R.P., Burke, I., Irwin, B.: Classifying network attack scenarios using an ontology. In: Proceedings of the 7th International Conference on Information-Warfare & Security (ICIW 2012), pp. 311–324. ACI (2012)
3. Joyal, P.: Industrial espionage today and information wars of tomorrow. In: 19th National Information Systems Security Conference, pp. 139–151 (1996)

4. Burstein, A.: Trade secrecy as an instrument of national security–rethinking the foundations of economic espionage. Arizona State Law Journal 41, 933–1167 (2009)
5. Grant, T., Venter, H., Eloff, J.: Simulating adversarial interactions between intruders and system administrators using ooda-rr. In: Proceedings of the 2007 Annual Research Conference of the South African Institute of Computer Scientists and Information Technologists on IT Research in Developing Countries, pp. 46–55. ACM (2007)
6. van Heerden, R., Leenen, L., Irwin, B., Burke, I.: A computer network attack taxonomy and ontology. International Journal of Cyber Warfare and Terrorism 3, 12–25 (2012)
7. Fenz, S., Neubauer, T.: How to determine threat probabilities using ontologies and bayesian networks. In: Proceedings of the 5th Annual Workshop on Cyber Security and Information Intelligence Research: Cyber Security and Information Intelligence Challenges and Strategies, p. 69. ACM (2009)
8. Shavitt, Y., Zilberman, N.: A geolocation databases study. IEEE Journal on Selected Areas in Communications 29(10), 2044–2056 (2011)
9. Stoll, C.: Tracking a spy through a maze of computer espionage, vol. 1. Doubleday (1989)
10. Ezzeldin, H.: Nmap detection and countermeasures. Online (March 2008) (accesed September 5, 2012)
11. Kibret, W.E.: Analyzing network security from a defense in depth perspective. Master's thesis, Department of Informatics University of Oslo (2011)
12. Yung, K.H.: Detecting long connection chains of interactive terminal sessions. In: Wespi, A., Vigna, G., Deri, L. (eds.) RAID 2002. LNCS, vol. 2516, pp. 1–16. Springer, Heidelberg (2002)
13. Spitzner, L.: Honeypots: Catching the insider threat. In: Proceedings of the 19th Annual Computer Security Applications Conference, pp. 170–179. IEEE (2003)
14. Myers, J., Grimaila, M., Mills, R.: Towards insider threat detection using web server logs. In: Proceedings of the 5th Annual Workshop on Cyber Security and Information Intelligence Research: Cyber Security and Information Intelligence Challenges and Strategies, pp. 54–58. ACM (2009)
15. Poese, I., Uhlig, S., Kaafar, M.A., Donnet, B., Gueye, B.: IP geolocation databases: unreliable? ACM SIGCOMM Computer Communication Review 41(2), 53–56 (2011)
16. Katz-Bassett, E., John, J.P., Krishnamurthy, A., Wetherall, D., Anderson, T., Chawathe, Y.: Towards ip geolocation using delay and topology measurements. In: Proceedings of the 6th ACM SIGCOMM Conference on Internet Measurement, pp. 71–84. ACM (2006)
17. Sanger, D.: Obama order sped up wave of cyberattacks against iran. Online (June 2012) (accessed August 24, 2012)
18. Shiffman, G., Gupta, R.: Crowdsourcing cyber security: a property rights view of exclusion and theft on the information commons. International Journal of the Commons 7(1), 93–112 (2013)
19. Stout, G.: Testing a website: Best practices. Technical report, Reveregroup (2001) (accessed January 2, 2013)
20. Lunt, T.F.: A survey of intrusion detection techniques. Computers & Security 12(4), 405–418 (1993)
21. Tjhai, G., Papadaki, M., Furnell, S., Clarke, N.: Investigating the problem of ids false alarms: An experimental study using snort. In: Jajodia, S., Samarati, P., Cimato, S. (eds.) Proceedings of the IFIP TC 11 23rd International Information Security Conference. IFIP, vol. 278, pp. 253–267. Springer, Boston (2008)
22. Hariri, S., Qu, G., Dharmagadda, T., Ramkishore, M., Raghavendra, C.S.: Impact analysis of faults and attacks in large-scale networks. IEEE Security & Privacy 1(5), 49–54 (2003)

23. Kuwatly, I., Sraj, M., Al Masri, Z., Artail, H.: A dynamic honeypot design for intrusion detection. In: IEEE International Conference on Pervasive Services (ICPS), pp. 95–104 (2004)
24. Bhuyan, M.H., Bhattacharyya, D., Kalita, J.: Surveying port scans and their detection methodologies. The Computer Journal 54(10), 1565–1581 (2011)
25. Merritt, D.: Spear phishing attack detection. Master's thesis, Air Force Institute of Technology (March 2011) (accessed January 1, 2013)
26. Mouton, F., Malan, M.M., Venter, H.S.: Social engineering from a normative ethics perspective. In: Information Security for South Africa, pp. 1–8 (2013)
27. Bezuidenhout, M., Mouton, F., Venter, H.: Social engineering attack detection model: Seadm. In: Information Security for South Africa, pp. 1–8 (2010)
28. Mouton, F., Malan, M., Venter, H.: Development of cognitive functioning psychological measures for the seadm. In: Human Aspects of Information Security & Assurance (2012)
29. Mouton, F., Leenen, L., Malan, M.M., Venter, H.S.: Towards an ontological model defining the social engineering domain. In: 11th Human Choice and Computers International Conference, Turku, Finland (July 2014)
30. Heberlein, L.T., Dias, G.V., Levitt, K.N., Mukherjee, B., Wood, J., Wolber, D.: A network security monitor. In: Proceedings of Computer Society Symposium on Research in Security and Privacy, pp. 296–304. IEEE (1990)
31. Christodorescu, M., Jha, S.: Testing malware detectors. ACM SIGSOFT Software Engineering Notes 29(4), 34–44 (2004)
32. Owen, D.: What is a false positive and why are false positives a problem? Online (May 2010) (accessed November 21, 2012)
33. Manmadhan, S., Manesh, T.: A method of detecting sql injection attack to secure web applications. International Journal of Distributed and Parallel Systems 3, 1–8 (2012)
34. Ciampa, A., Visaggio, C.A., Di Penta, M.: A heuristic-based approach for detecting sql-injection vulnerabilities in web applications. In: Proceedings of the 2010 ICSE Workshop on Software Engineering for Secure Systems, pp. 43–49. ACM (2010)
35. Win, W., Htun, H.H.: A simple and efficient framework for detection of sql injection attack. International Journal of Computer & Communication Engineering Research 1(2), 26–30 (2013)
36. Jim, T., Swamy, N., Hicks, M.: Defeating script injection attacks with browser-enforced embedded policies. In: Proceedings of the 16th International Conference on World Wide Web, pp. 601–610. ACM (2007)
37. Scholte, T., Robertson, W., Balzarotti, D., Kirda, E.: An empirical analysis of input validation mechanisms in web applications and languages. In: Proceedings of the 27th Annual ACM Symposium on Applied Computing, pp. 1419–1426. ACM (2012)
38. Rao, T.: Defending against web vulnerabilities and cross-site scripting. Journal of Global Research in Computer Science 3(5), 61–64 (2012)
39. Karig, D., Lee, R.: Remote denial of service attacks and countermeasures. Technical Report CE-L2001-002, Princeton University Department of Electrical Engineering (October 2001) (accessed January 1, 2013)
40. Mirkovic, J., Reiher, P.: A taxonomy of ddos attack and ddos defense mechanisms. ACM SIGCOMM Computer Communication Review 34(2), 39–53 (2004)
41. Bhide, A., Elnozahy, E.N., Morgan, S.P.: A highly available network file server. In: Proceedings of the 1991 USENIX Winter Conference, pp. 199–205. Citeseer (1991)
42. Yang, D., Usynin, A., Hines, J.W.: Anomaly-based intrusion detection for scada systems. In: 5th International Topical Meeting on Nuclear Plant Instrumentation, Control and Human Machine Interface Technologies (NPIC & HMIT 2005), pp. 12–16 (2006)
43. Gula, R.: Correlating ids alerts with vulnerability information. Technical Report Revision 4, Tenable Network Security (May 2011)