12-4-2013

Acquisition Of Evidence From Network Intrusion Detection Systems

Brian Cusack
Edith Cowan University, brian.cusack@aut.ac.nz

Muteb Alqahtani
Auckland University of Technology, muteb.alqahtani@aut.ac.nz

DOI: 10.4225/75/57b3c1fefb86a
11th Australian Digital Forensics Conference. Held on the 2nd-4th December, 2013 at Edith Cowan University, Perth, Western Australia

ACQUISITION OF EVIDENCE FROM NETWORK INTRUSION DETECTION SYSTEMS

Brian Cusack[1,2], Muteb Alqahtani[1]
[1]Auckland University of Technology, Auckland, New Zealand
[2]Security Research Institute, Edith Cowan University, Perth, Australia
brian.cusack@aut.ac.nz, muteb.alqahtani@aut.ac.nz

Abstract
The literature reviewed suggests that Network Intrusion Systems (NIDS) are valuable tools for the detection of malicious behaviour in network environments. NIDS provide alerts and the trigger for rapid responses to attacks. Our previous research had shown that NIDS performance in wireless networks had a wide variation under different workloads. In this research we chose wired networks and asked the question: What is the evidential value of NIDS? Three different NIDS were tested under two different attacks and with six different packet rates. The results were alarming. As the work loading increased the NIDS detection capability fell rapidly and as the complexity of attack increased the NIDS detection capability fell more quickly. We conclude that NIDS have weak evidential value for either system improvement or legal admissibility.

Keywords
Detection, Forensic, Evidence, Performance, Admissibility

INTRODUCTION

Intrusion Detection systems are the tools that process, identify and respond to malicious activity targeted at information and networking resources (Lokhande, Bhaskarwar, Bhaskarwar, & Chidrawar, 2012). In general IDSs are categorized into the two types; network intrusion detection systems (NIDS) and host-based IDS. A NIDS monitors packets on the network and attempts to discover if a hacker is trying to break into a system (Maier, Sommer, Dreger, Feldmann, Paxson, & Schneider, 2008). A typical example is a system that watches a large number of TCP connection requests (SYN) for many different ports on a target machine, thus discovering if someone is attempting a TCP port scan. This technology has been made to compare users' action against known attack scenarios and be able to predict and indicate suspicious behaviour. There are two types of NIDS and each one has different mechanisms from the others. These types are: Anomaly-based detection also known as behaviour-based detection that works to detect the behaviour that is not normal or is not compatible with normal behaviour. Theoretically this type of IDSs requires a whole list of normal behaviour actions. However, in some environments this can work because the normal behaviour actions are limited (Ren, 2006). The model is considered to be dangerous because it might have unacceptable behaviour listed within the training data and it will be accepted later as normal behaviour. The second one is Signature-based detection also known as misuse-based detection. This type uses the signature of known malicious activities. It also works by adding a whole list of known malicious actions or misuse signatures (Saari & Jantan, 2011). Although there are a huge number of malicious actions, it is impossible to put all these activities on a list and keep it manageable. Therefore, there are only a limited number of action signatures added to the list and the limitation depends on three activities: unauthorized modification, unauthorized access and denial of service (Pomeroy & Tan, 2011).

NIDS AS SOURCES OF EVIDENCE

There is a wide array of intrusion detection systems that are specifically designed for wired networks (Sommer, 1999; Nikkel, 2005). These systems and/or products are available in two types; either open source software or commercial ones and all of them are addressing a range of organizational security goals and considerations. Snort is one of the most popular deployed systems in most network security environments and it is considered to have a huge dataset of signatures for malicious activities. It searches for very specific content in the network stream and reports each instance of a particular signature. Snort's modular design allows developers to create and add extra features into the core detection engine. However, snort may have some disadvantages such as

dropping packets when the process network rate is at 100-200 megabytes per second before reaching the processing limit of a single CPU because the Snort engine is designed to work with single-threaded multi-stage units. Research reports have compared Snort with other IDS such as Suricata and Bro-IDS that offer different features and capabilities. They concluded that Suricata's multi-threaded architecture requires more memory and CPU resources than Snort but it can accommodate many network instances with no need to use multiple instances. On the other hand, snort is efficient but with limited ability to measure more than 200-300 Mbps network bandwidth per instance (Saari & Jantan, 2011). Both systems were observed to miss several common malicious packets. Bro-IDS is slightly different in that it uses more sophisticated signatures by means of its policy language. It can analyse network traffic at a much higher-level of abstraction, and has powerful facilities for storing information about past activity and incorporating it into analysis of new activity which means it is better for forensics. IDS can be selected for context, used together and optimised for purpose.

Numerous papers have been published suggesting that NIDS can be used as sources of evidence; however, each reference acknowledges that there is a gap between the purposes of NIDSs and the needs of the legal system (Casey, 2004). This gap doesn't only lack at a functional and a purposive level, but also the preservation and completeness of evidence may be insufficient or incomplete. Hence consideration of embedding NIDS in support systems and improving NIDS capability are potential solutions. Some recent intrusion detection systems are capable of recording malicious commands including source and distinct network addresses, protocol used, event characteristics such as time and date and further related information such as username and filename applied within the application. These innovations are a move in the required direction for better evidence collection.

The published works on the use of using intrusion detection systems for network forensics purposes outline the analytic potential for attacks, duration accounting of the exploit and recording the methods applied during the attack (Saari & Jantan, 2011; Kaur & Kaur, 2012). These descriptions include log system information gathering, adaptive capture of network traffic, storing the historical network misuse patterns and active or automated responses for investigational forensics. A suggested investigatory system contains four elements which are a forensics server, network forensics-agents, network monitors, and network investigator software (Reith, Carr & Gunsch, 2002). Another simple framework for distributed forensics has an integrated platform for automatic evidence collection and efficient data storage, and matching known attribution methods against attacks. This model has proxy and agent architectures that collect, stores, processes and analyses forensic information. One advantage is automatic evidence collection and quick responses to network attacks. Other systems provide a combination of agent theories and artificial intelligence to dynamically assure evidence retention and extraction capabilities. The systems have different elements to store data in the forensics server and collect the data from forensic-agents, detector-agents and response-agents. The detector-agent, forensics-agent and response-agents manage to capture real-time network data, match them with intrusion behaviour and then send them to the forensics –agent. The forensics- agent helps to collect the digital evidence, create a digital signature and then transmit the evidence to the forensic server. The forensic server plays a role in analysing the evidence and replaying the attack procedure in order to create a quick response to attacks (Pilli, Joshi,& Niyogi, 2010).

THE RESEARCH QUESTIONS

Testing of NIDS in wireless networks showed that performances were related to work loadings and that generally inadequate evidence was stored for forensic purposes (Laurenson, 2010). These findings and the literature reviewed suggest that there are issues and problems associated with using NIDS for evidential purposes. Consequently the main research question was defined as: What is the evidential value of a NIDS?

Laurenson's (2010) research was conducted in a wireless network so we resolved the variation of testing his findings in a wired network. Also he chose to test NIDS under increasing workloads. We resolved to do the same but in addition to consider the complexity variable as a potential influence on performance. The motivation for researching tool complexity came from the literature on the architectures of different NIDS and their capabilities. Hence we asked set of sub-questions to tease out the main question in the problem context:

(1) What is the number of packets lost?
(2) What is the retention of event data?
(3) What complexity can a NIDS cope with?
(4) What are the system costs?
(5) What is the viability of the evidence for compliance with legal expectations?

Testing sub-questions (1) to (4) relies on stress testing and count data. Sub-question (5) however has legal complexity and requires the specification of admissibility criteria. Consequently we chose the Daubert criteria (http://en.wikipedia.org/wiki/Daubert_standard) to be a fair representation and generalisation of the grounds on which a court may accept evidence. The criteria are:

1. Empirical testing: whether the theory or technique is falsifiable, refutable, and/or testable.
2. Whether it has been subjected to peer review and publication.
3. The known or potential error rate.
4. The existence and maintenance of standards and controls concerning its operation.
5. The degree to which the theory and technique is generally accepted by a relevant scientific community.

THE RESEARCH DESIGN

NIDS are principally software assets that manage adverse events and maintain the system status quo. In our analysis of literature none addressed the evidential issue of prosecuting offenders and hence lifting the system defence to a level of human accountability. This would not be the case if a retail shop was broken into and damages done. The police would be called and evidence collected to identify the offenders. The retail shop owners would also review the security measures and make any suitable adjustments. To us it seemed unusual that NIDS with evidential capabilities could not be lifted to the status of prosecution evidence. To this end we designed a number of tests to see how good NIDS were at collecting prosecution evidence and what elements required improvement. Our testbed is illustrated in Figure 1 below. The testbed had facilities to generate packets at different controlled rates, a simulated attacker who could launch Cross-site scripting attacks and SQL injection attacks, a packet sniffer at the router, a firewall, a location to place the different NIDS for testing and forensic and web servers to respectively collect all packets and to provide target services.

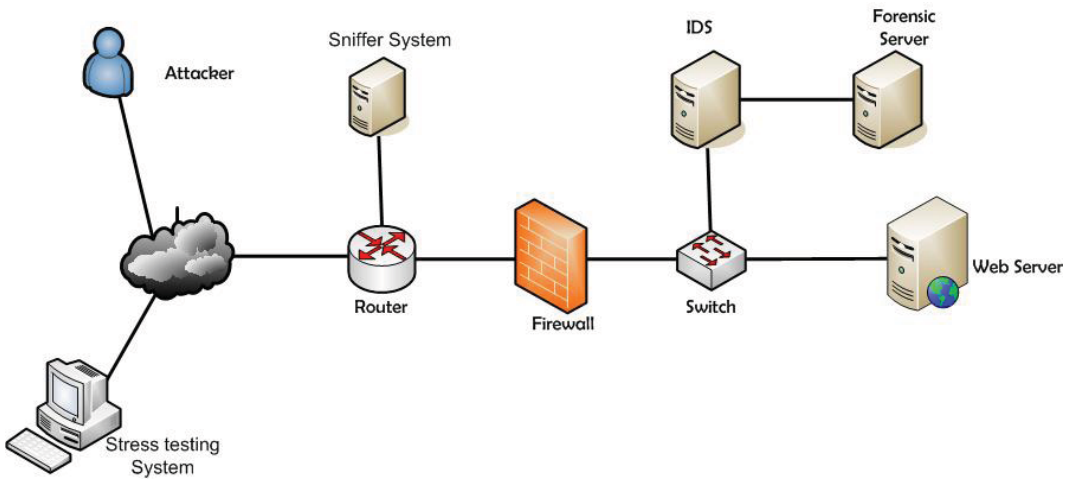


Figure 1. The Testbed Architecture

The research proceeded by benchmarking the system at status zero where everything was set to zero capability. The system was then powered up and the functionality of each component checked. The testing started by setting the packet rate value and by launching one attack. Once the data was collected then the next packet rate was set and run until the maximum value of 100Mbs. The rates started at an arbitrary .4Mbs and then were stepped at 20Mbs across 5 intervals. The procedure was repeated for the second attack type. Once completed the system was returned to the original zero state, a different NIDS mounted and the procedures repeated again until all the NIDS had been tested.

THE RESULTS

The results of our testing of three NIDS provided answers to each of the sub-questions. In the first instance (1) the number of packets dropped by the NIDS increased with the workload (see figure 2). Snort showed greatest increase in dropped packets from 30% at 30Mbs to 80% at 100Mbs. Bro-IDS and Suricata dropped some packets but remained under 10% for the whole experiment. Each NIDS retained no data unless the options were configured to do this (2). In the testbed architecture the forensic server retained all evidence generated by the configured NIDS plus all traffic passing through the network. Bro-IDS and Suricata delivered more evidence than Snort as both had more options and dropped fewer packets. The tests for complexity (3) showed variable performance between the NIDS (see figure 3). Figure 3 shows the results of the cross site scripting attack – the most complex attack. Four attacks were made at each workload. At 10 Mbs only Bro-IDS detected all attacks. At 100Mbs none of the NIDS detected any attacks. Snort had the lowest performance and Suricata performed better. The SQL injection attack had less complexity than the cross site scripting attacks but the results showed a similar pattern. Each NIDS maintained a higher detection rate but all fell to zero by 100Mbs. The system costs (4) in terms of CPU usage and memory requirement were as expected (see figure 4). As a percentage of CPU usage Bro-IDS and Suricata were both more costly. These costs were expected because the NIDS with higher level abstractions also require greater computational power and storage services. Snort is a long established product that was designed for minimal requirements and as a consequence is ineffective under complex attacks and heavy workloads. However it is economical in terms of its cost to the system and presents a low overhead. Sub-question (5) required higher level analysis to resolve the relationship between the data and the Daubert criteria.

The scenario is that a NIDS is being used to collect evidence of a system attack. Under the first Daubert criteria the evidence is not repeatable because the data sets are incomplete and each trial for empirical testing is a unique event. Under criteria 2., the use of NIDS for collecting evidence have been subject to peer review and publications but these publications (as referenced above) report problems with the integrity and consistency of the evidence in different conditions. Under criteria 3., the potential error rates are not tabulated but can be established in pre-tests. Under criteria 4., NIDS are used in network systems that can be certified and held in control by standards and requirements for the operational use. Under criteria 5., the theory, the use and the acceptance of NIDS is generally accepted in the IT security community.

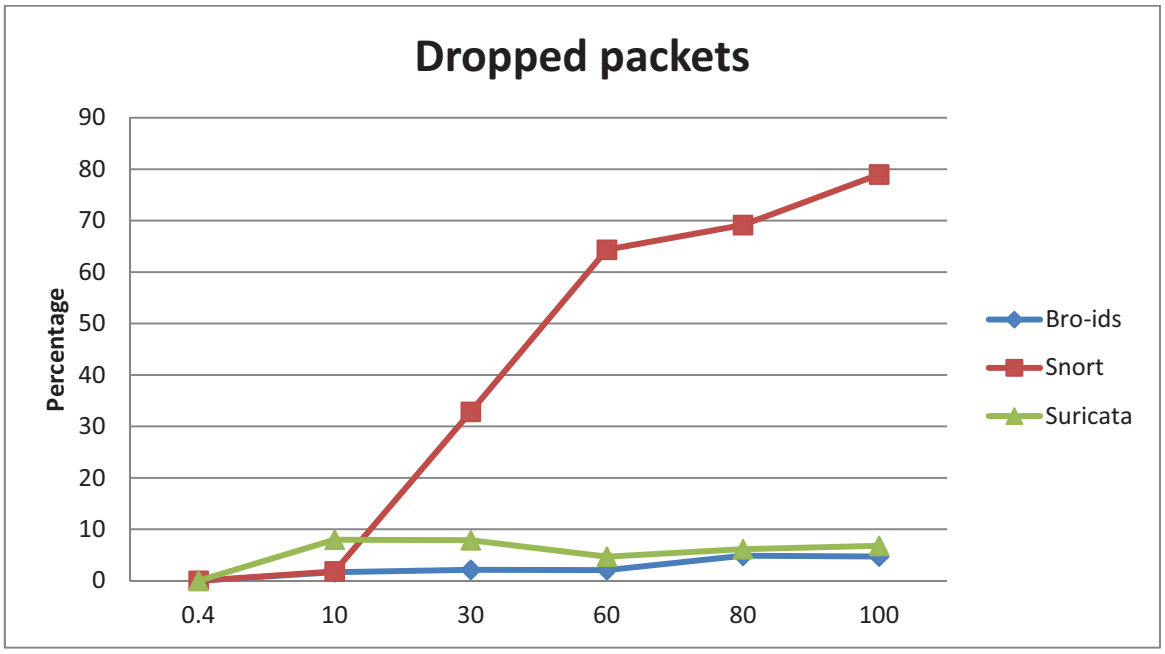
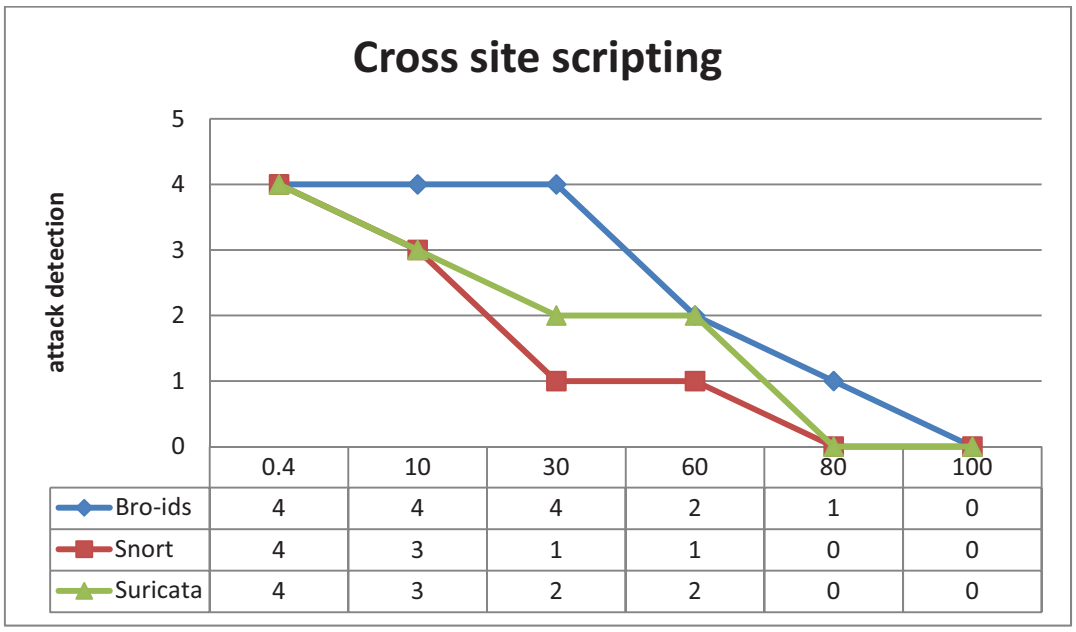


Figure 2. Dropped packet percentage



	0.4	10	30	60	80	100
Bro-ids	4	4	4	2	1	0
Snort	4	3	1	1	0	0
Suricata	4	3	2	2	0	0

Figure 3. Cross Site Scripting Attack Detection (4 in total for each NIDS)

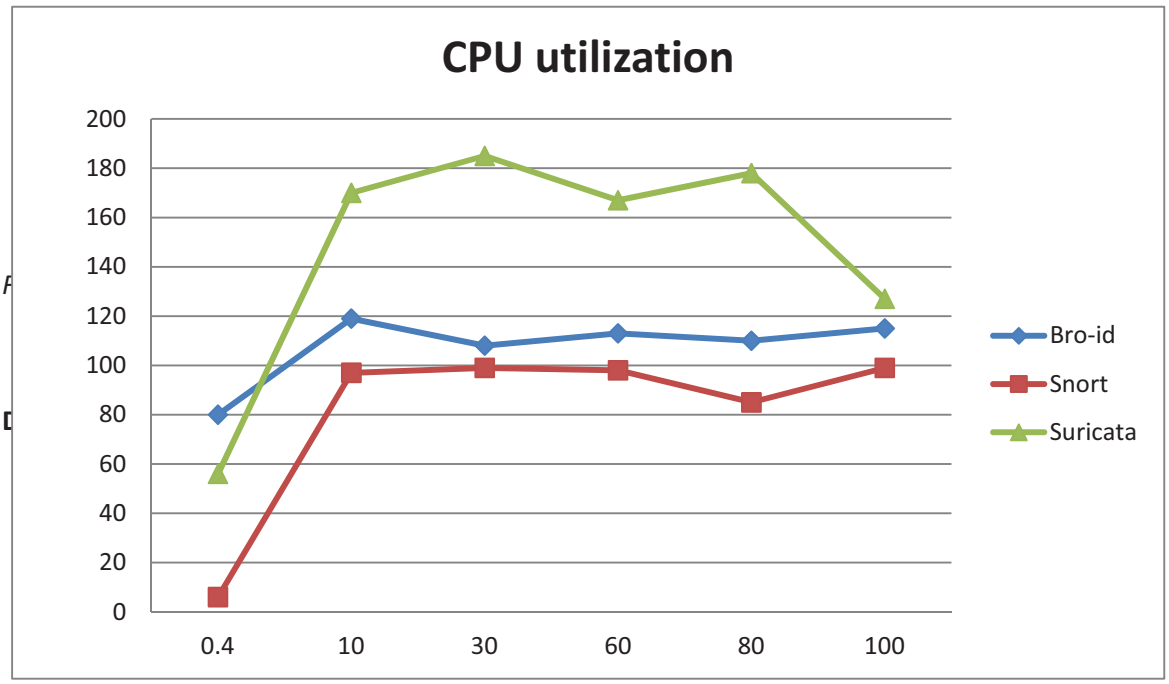


Figure 4. CPU utilization (Maximum 4 x 100%)

DISCUSSION

The results of this research are disturbing. The literature reviewed suggested that NIDS were valuable assets for network security and forensic investigation. The literature also indicated weaknesses in NIDS that can affect effectiveness and our results confirm Laurenson's (2010) general performance concerns found in wireless networks. NIDS act as detection systems and trigger response systems to protect assets. In addition log files are kept (when configured) so that records of events are retained. However, our research shows that a NIDS only performs its required functionality in particular conditions. When complex attacks happen during heavy workload periods most NIDS fail to perform. The implications of this impact the trust users may have and the degree of assurance that may be given to any network system protected by NIDS.

The main research question asked: What is the evidential value of NIDS? There are two parts in the answer to the question. Evidence from NIDS has from the literature reviewed principally concerned evidence of an attack that can be made available from the system log files. The evidence is consequently used in a forensic post event fashion to better prepare and to prevent a similar occurrence. However, contrary to the strong assertions made in the literature the popular NIDS we tested and under the conditions failed to perform as required. The implications of our findings are both for alerts and responses and for post mortem actions. The data showed that under some conditions 100% of the attacks passed unreported. A normal system is expected to perform at 100Mbs and against widely available and used attacks and yet the NIDS tested failed. System post mortem improvement based on such incomplete data sets can only be inadequate for defence. SQL injection attacks have been used on businesses to alter stock prices and to make legitimate sales at incorrect transaction points; effectively taking $1000s from stores. Similarly Cross Site scripting attacks have been used against banks to fuzz online banking forms and to steal $1000s. The cost to businesses of missing one of these attacks is high; and yet our study shows that 100% of the attacks can be missed under normal network working conditions. The results raise serious concerns for network security and the assurance of asset protection.

The second part of the question concerned the availability and the quality of evidence for prosecuting offenders. As mentioned above this is the usual expectation where damages or theft has occurred to a business or the systems. We used the Daubert criteria to assess the potential admissibility of the evidence to court. Assessment of the first criteria suggested that repeatability and the incomplete data sets are a problem. Criteria 2 has been satisfied but the test reports cite limitations and exceptions when using NIDS for collecting evidence. Criteria 3 has again been satisfied but the error rates require scrutiny against and benchmarking against acceptable assurance scales. Criteria 4 and 5 are satisfied. NIDS evidence may be admissible to a court of law but under cross examination many questions can be asked. An expert witness may be justified in presenting such findings in the report based on the strength of the investigation methodology. However, the gaps and exceptions existent in the primary evidence collection will require collaborative evidences from other sources to explain the NIDS contribution. A further and more challenging question is: Why is criteria 5 satisfied in the IT security community? From the literature reviewed it is easy to see that the strong case is that NIDS are effective and efficient for detection and reliant on for rapid response defence. The weaker case made in the literature is for conditions on performance. To simply explain the discrepancies as a cost benefit trade-off overlooks the weighted problem of reputation and credibility. A brand may financially accommodate many successful attacks through spreading financial assurance but poor protective performance weakens customer appreciation. Similarly successful criminals who roam unchallenged will continue to offend and harm systems.

CONCLUSION

Our findings suggest that NIDS performance for generating attack alerts varies greatly with regard to the number of packets passing through the network and also the complexity of attacks. The variability in performance raises serious issues regarding the value of evidence collected by NIDS. A simple DDOS attack when not much is happening on a network can be readily alerted but in normal busy day and with a complex attack the NIDS is of little use. Disturbingly the results indicate that an engineer hoping to protect information assets and/or a computing asset requires more than fully functional NIDS. Similarly, evidence collected by NIDS may fall short of admissibility or face a raft of challenges in a court room if offenders are to be prosecuted. We take issue with the assumption that NIDS can provide trustworthy evidence. Without consideration of the many variables impacting the performance of a NIDS unqualified statements of fact cannot be made. Our investigation shows that NIDS cannot act alone in protecting a system and that other evidence would also be required before admissibility.

REFERENCES

Casey, E. (2004). Network traffic as a source of evidence: tool strengths, weaknesses, and future needs. *Digital Investigation*, 1(1), 28–43.

Kaur, A., & Kaur, R. (2012).Digital Forensics.*International Journal of Computer Applications*, 50(5), 5-9.

Laurenson, T. (2011).Systems Architecture for the Acquisition and Preservation of Wireless Network Traffic.*Proceedings of the 11th Australian Conference on Digital Forensics*, Perth, WA.

Lokhande, S., Bhaskarwar, A., Bhaskarwar, S., & Chidrawar, S. (2012). Intrusion Dectection System to Detect Bandwidth Attacks. *Proceedings of the National Conference on Advancement in Electronics & Telecommunication Engineering*, NCAETE, IJCA, 18-22.

Maier, G., Sommer, R., Dreger, H., Feldmann, A., Paxson, V., & Schneider, F. (2008, August).Enriching network security analysis with time travel.*ACM SIGCOMM Computer Communication Review* (Vol. 38, No. 4, pp. 183-194). ACM.

Nikkel, B. J. (2005). Generalizing sources of live network evidence.*Digital Investigation*, 2(3), 193-200.

Pilli, E. S., Joshi, R. C., & Niyogi, R. (2010). Network forensic frameworks: Survey and research challenges. *Digital Investigation*, 7(1), 14-27.

Pomeroy, A., & Tan, Q. (2011, August).Effective SQL Injection Attack Reconstruction Using Network Recording.*Proceedings of 2011 IEEE 11th International Conference on Computer and Information Technology (CIT)*, (pp. 552-556).

Reith, M., Carr, C., & Gunsch, G. (2002).An examination of digital forensic models.*International Journal of Digital Evidence*, 1(3), 1-12.

Ren, W. (2006).Modeling Network Forensics Behavior.*Journal of Digital Forensic Practice*, 1(1), 57-65.

Saari, E., & Jantan, A. (2011). F-IDS: A Technique for Simplifying Evidence Collection in Network Forensics. In *Software Engineering and Computer Systems* (pp. 693-701). Springer Heidelberg.

Sommer, P. (1999). Intrusion Detection Systems as Evidence. *Computer Networks*, 31(23), 2477-2487.