2013

Testing a distributed denial of service defence mechanism using red teaming

Samaneh Rastegari
Edith Cowan University

Philip Hingston
Edith Cowan University

Chiou-Peng Lam
Edith Cowan University

Murray Brand
Edith Cowan University

10.1109/CISDA.2013.6595423

Rastegari, S. , Hingston, P., Lam, C., & Brand, M. (2013). Testing a distributed denial of service defence mechanism using red teaming. Proceedings of the 2013 IEEE Symposium on Computational Intelligence for Security and Defense Applications (CISDA). (pp. 23-29). Singapore.

Testing A Distributed Denial of Service Defence Mechanism Using Red Teaming

Samaneh Rastegari, Philip Hingston, Chiou-Peng Lam, and Murray Brand
School of Computer and Security Science
Edith Cowan University
WA 6050, Australia
Email: {s.rastegari, p.hingston, c.lam, m.brand}@ecu.edu.au

***Abstract*—The increased number of security threats against the Internet has made communications more vulnerable to attacks. Despite much research and improvement in network security, the number of denial of service (DoS) attacks has rapidly grown in frequency, severity, and sophistication in recent years. Thus, serious attention needs to be paid to network security. However, to create a secure network that can stay ahead of all threats, detection and response features are real challenges. In this paper, we look at the the interaction between the attacker and the defender in a Red Team/Blue Team exercise. We also propose a quantitative decision framework which is able to provide optimal solutions to defend against well-organized and sophisticated attacks. A large number of possible scenarios for testing of DoS defences will be examined through this framework in order to help experts to improve decisions regarding optimal solutions to defend against DoS threats.**

I. Introduction

Recently, denial of service (DoS) attacks have proved to be the most serious threats to the Internet. As a result, many defence mechanisms have been developed to find them and fight against them. Based on a report from InformationWeek magazine, DoS attacks are getting more sophisticated and increasingly are growing since 2005 [1]. Since attack mechanisms are changing over time, the set of defences should also be dynamic in their behaviour. Most of the proposed defence approaches will perform different functions in different environments. Implementing and applying every possible defensive strategy is not possible and it is difficult to compare different defensive strategies.

In order to gain a deep understanding of the performance of a defence mechanism, testing a system against complex attack scenarios is of paramount importance. By investigating the interaction between various DoS attacks and defence systems, we will be able to create a reliable cyber-defence system that can protect the network against DoS attacks. However, choosing an effective set of defences and adjusting their behavioural functions are not simple and consequently, humans are not individually capable of selecting good choices when complex trade-offs are involved. Thus, we need a suitable framework which can provide a reasonable trade-off between the functional and non-functional properties of defence mechanisms.

In order to achieve the best observation of the mechanisms' behaviour, the testing environment has to possess realistic characteristics of the Internet. This observation leads to control the functioning of the network continuously and analyse the possible risks, which can help the network administrators to collect more knowledge about counteraction and use it for enhancing the defence system.

The aim of this paper is to provide a framework for investigation of strong DoS attack strategies and best defensive solutions against these attacks. In order to achieve the aim, we first need to integrate different attack and defence strategies into a simulation environment to easily evaluate the interactions. The interaction between attacks and defence strategies is studied using Red Teaming (RT). RT is a concept that derives its name from military planning and decision-making. The Blue Team objectives are to evaluate all security layers for possible vulnerabilities and provide the system security in an effective and efficient manner. On the other hand, the enemies are represented by the Red Team. They try to find security gaps, which have not been considered during system development. By having a Red Team, the Blue Team can test and evaluate its reaction against enemies and then it can measure how the defence mechanisms deal with different attacks. These exercises provide knowledge about interaction between participants, which lead to rapid improvement of the defence quality. Based on the interactions between attacks and defence mechanisms, different defensive strategies that would be applicable against attacks can be realized. In this stage, the computational side of RT will help to find the optimal solutions by investigating a space of possibilities. In this paper, we focus on RT exercises to show how the evaluation process was simulated. In particular, we do not present our defence strategies as optimal solutions comparing to other systems.

This paper is structured as follows: Section II presents the related work. Section III outlines the simulation environment used in this research. The experimental setup is described in section IV. The whole system is evaluated and the results are explained in Section V. Finally, section VI outlines conclusions and indicates areas for future work.

II. Related Work

DoS attack has been the most devastating Internet security problem for almost a decade. A distributed denial of service (DDoS) is a special case of DoS in which the attacker tries

to deploy multiple machines to achieve the goal. In this case, the service is denied by sending a huge amount of traffic to the victim in two phases: a deployment phase and an attack phase [2]. DoS attacks exploit different strategies in order to deny the service of the victim to its legitimate clients. They are generally classified into two broad categories: logic attacks and flooding attacks. Logic attacks (or protocol attacks) exploit vulnerabilities in some installed protocols at the victim. Most of the Internet infrastructure DoS attacks are in this category, such as DNS cache poisoning attack, Teardrop attack based on overlapping IP fragments, and injection of false routing information [3], [4]. On the other hand, flooding attacks are known as brute force attacks and they perform the attack by sending a vast amount of traffic to the victim. This traffic is much higher than what will be seen in logic attacks. Both flooding and logic attacks exploit direct attack mechanisms to transmit the attack packets to the victim. TCP flooding, ICMP Echo flooding, and UDP flooding are the most common direct flooding attacks [5]. In this paper, our focus is on flooding attacks.

Defence mechanisms against DoS attacks are generally classified into three broad categories: attack prevention, attack detection, and attack reaction [6]. A good defence system should have a proper attack detection phase before any reaction. The goal of every attack detection mechanism is to detect intrusions before any serious damage. A good system can detect attacks in a short period of time with a low proportion of false positives. Based on the analysis methods used by detection mechanisms, there are two broad groups of detection systems: signature-based detection and anomaly-based detection. Anomaly-based detection systems explore intrusions based on deviations from normal behaviours. On the other hand, signature-based detection systems can identify an attack if the monitored traffic matches the attack patterns (signatures). Therefore, anomaly-based detection systems are able to detect new or modified attacks. There are different methods and techniques involved in anomaly-based detection systems such as threshold detection, statistical measurement, rule-based methods, and evolutionary computation methods [7].

As the Internet is a resource-sharing architecture, a reaction mechanism should be employed when an attack is underway. The advantages of having a good reaction technique are saving the bandwidth and separating the packets belonging to attack traffic and normal traffic. However, it needs to make sure that in the filtration phase, only attack traffic is filtered, and it has no impact on the legitimate traffic [8]. There are several reaction mechanisms proposed in the literature such as killing of active network connections, filtering, rate limiting, reconfiguration, and source traceback mechanism. In [9], the effectiveness of rate limiting for mitigating TCP-based flooding DoS attacks is evaluated. According to their results, this mechanism is a suitable reaction technique against flooding DoS attacks.

Many Intrusion Detection Systems (IDSs) have been proposed against DoS attacks in recent years. While the focus of research in this area was on designing the most effective solutions, little attention has been paid to evaluate these solutions. The reason for the lack of study in the evaluation of the solutions is the complexity of the DoS phenomenon [10]. However, measuring the capability of an IDS is necessary since it will enable us to refine the IDS configurations and compare different IDSs. It is not a simple process since there is a need to compare different defence strategies and decide about the best.

RT has long been a valuable tool for military planning and decision-making. Nowadays, this technique is used in different expanded areas for varying purposes such as financial markets, air traffic, and politics [11]. Identifying vulnerabilities, decreasing risks, diagnosing unforeseen consequences, and overall, a better understanding of operational environment are some of the benefits of RT, which lead to improved decision making and effective actions. In order to evaluate the security of a network from the attackers' point of view, RT techniques can be implemented. RT is an ethical hacking process to test the security of networks by modelling the actions of an attacker. It is worth noting that RT was introduced as one of SANSs 20 critical security controls for organizations [12]. There are two participating teams in this process: Red Team and Blue Team. The interaction of these two teams is similar to the concepts of methods of understanding competitions such as game theory or drama theory [11]. Furthermore, in order to overcome the problem of computational expense in RT exercises, evolutionary computation techniques can be used. The most well-known algorithms in this category include genetic algorithms, genetic programming, and evolutionary programming [13]. From this category, co-evolutionary algorithm is a powerful tool that involves the simultaneous evolution of competing species with coupled fitness [14]. Computations in Computational Red Teaming (CRT) play the decision-making role in a RT exercise, while this process is done by a human in a traditional RT.

In [15], a cooperative co-evolutionary genetic algorithm is proposed for network security in a grid computing environment. Each detector collaborates with other detectors from different species and finally the optimal detectors for the target set are selected. Each species contains a collection of a selected detector and represents only a partial solution. The experiments are implemented on KDD cup 1999 data. Finally, the proposed co-evolutionary algorithm showed positive cooperation for converging the results to an optimum solution for detection of four different DoS attacks.

In a Red Team/Blue Team competition, in which, Red Team strategies are evolved against Blue Team strategies in order to find optimal solutions for each side, we need a competitive co-evolutionary algorithm. This method has been used by a number of research groups for different applications. In [16], an Automated Co-Evolution (ACE) framework was used to discover the dynamics of RT in a military context through simulations. Several evolutionary algorithms such as Elite Pareto Genetic Algorithm (EPGA) and Strength Pareto Evolutionary Algorithm Version 2 (SPEA 2) were compared

in this paper. The results from ACE can be used by Blue force to improve the effectiveness of their tactics.

In [17], a combination of agent-based simulation and evolutionary algorithm was used to design an Automated Red Teaming (ART) framework to study RT scenarios. The framework was examined in an urban operations scenario [18], and the results provided strong strategies for each side. For example, the Blue Team requires effective sensors to track down and destroy the hidden Red Team forces.

To the best of our knowledge, the research by Mirkovic et al. [19], is the first RT exercise, which included both Red Team and Blue Team in the scope of DoS defence systems. There are two security systems developed by the Information Science Institute (COSSACK [20]) and the UCLA (D-WARD [21]), which have been evaluated in this experiment. The researchers of Sandia National Laboratories formed the Red Team and the Blue Team consisting of researchers of UCLA and Information Science Institute. The experiments developed in a real environment during a period of eight months, which was sponsored by DARPAs Fault Tolerant Program. The Red Team is trying to change the attack behaviour ranging from simple to sophisticated in order to stress test the two defined defence mechanisms. There is a limitation on the Blue Team since they cannot change the defensive strategy once the game is started. If both teams were allowed to change their strategies and modify their initial settings, the results would have been much more worthwhile [19]. Implementing a complete interaction of Red Team and Blue Team in a real environment requires large amounts of time, money, and resources. However there is a possibility of considering other testing approaches such as simulation to easily achieve the repetition of the exercises.

For this purpose, we would need a network simulator, which can simulate the Internet environment as realistically as possible. It should be able to simulate a detailed implementation of different protocols that are employed in DoS attacks. In order to simulate different strategies, simulation parameters need to be changed during simulation. To be able to compare defensive strategies, different attack detection and response mechanisms should be easily integrated into the simulator. Finally, it is important for researchers to have access to the simulator for educational and research purposes [22], [23]. Considering these requirements, we identified the network simulator OMNeT++ as the best choice for our testing approach. To address the challenging problem of integration of detection mechanisms into the simulator, there are two approaches in the literature: DDoSSim [24] and Gamer et al.'s simulation toolchain for distributed attack detection [22]. DDoSSim is an extension of discrete event simulator OMNeT++, which has been developed for the purpose of investigating attack scenarios and protection mechanisms. This software simulation tool, however, is not publicly available. The Gamer et al.'s simulation toolchain is also based on OMNeT++. To simulate the attack and defence mechanisms, two extra components are developed. All of the tools used in the simulation toolchain (INET framework, ReaSE [25], Distack [26]) are publicly available as open source software. Usability, simplicity, close to reality, flexibility, and scalability are the features addressed by this toolchain [22].

III. Simulation Environment

As mentioned before, the experiments in this research are based on the RT concept. This concept has long been used for military planning and decision-making. There are basically two teams involved in RT exercise, which leads to more realistic test scenarios. In our experiments, the Red Team generates different attack scenarios by varying the characteristics of the DoS attack such as packet size, attack type, and address spoofing status. On the other hand, the Blue Team applies some countermeasures such as filtering or rate limiting. The complex process of interactions between Red Team and Blue Team in these experiments is achieved through a simulation environment.

The first required package for Internet simulations which is based on OMNeT++, is INET framework [27]. It provides models for Internet protocols including UDP, TCP, IP, OSPF, etc. Furthermore, it provides functionality of Internet end and intermediate systems through Internet-like entities such as StandardHost and Router, respectively.

In the next step, attack generation and implementation of the attack detection method are carried out using ReaSE and Distack modules. ReaSE, which was first introduced by Gamer and Scharf [25], provides realistic and repeatable Internet simulation scenarios. Their objective was facilitating the evaluation of Internet-like systems and protocols, especially with regards to attack detection techniques. ReaSE is able to generate traffic types such as web, streaming, mail, and ping traffic. Furthermore, it provides different attack traffic types including distributed DoS attacks and UDP-based and TCP-based worm propagations.

For development of the anomaly attack detection mechanism, we use the Distack framework, which was first introduced by Gamer et al. [26]. For integrating Distack into OMNeT++, the whole framework can be loaded as a shared library in OMNeT++. The process of attack detection by Distack is performed on routers. Various light weight modules are used in Distack to implement the detection and analysis methods. In order to provide different defensive strategies for the Blue Team in our experiments, there is a need to coordinate the detection and remediation mechanisms. Remediation mechanisms are usually activated once a challenge is detected by the detection component. While the Distack framework is responsible for detection anomalies in the simulated network, the Ponder2 framework provides remediation of those anomalies through some policy configurations [28]. This policy-based framework facilitates management of defence strategies by controlling the detection and response mechanisms. Ponder2 was first evaluated based on a SSFNet implementation [28] and then, it was ported into OMNeT++ by Yu et al. [29].

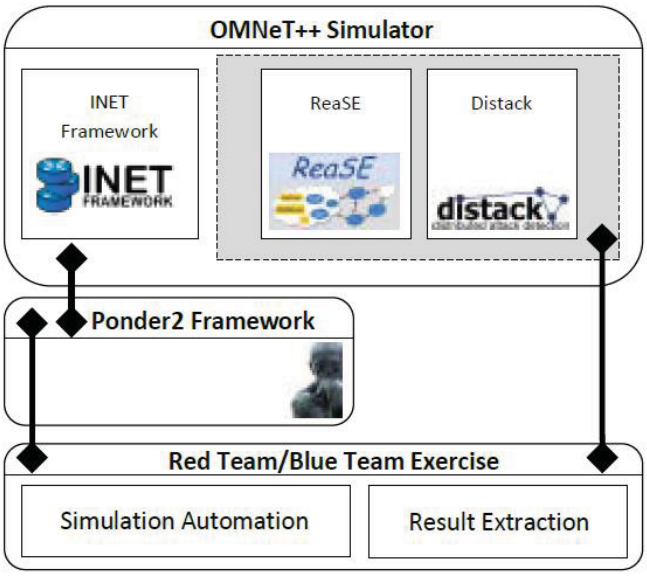


Fig. 1. Architecture of simulation environment for Red Team/Blue Team exercises

To speed up the process of the Red Team/Blue Team exercise, we developed a Simulation Automation component. In addition to that, a Result Extraction component is also developed to evaluate the effectiveness of strategies. A combination of Java programming and shell scripting is used to implement the automation and extraction processes. As shown in Figure 1, the Simulation Automation and Result Extraction components are interacting with Ponder2, ReaSE, and Distack framework to perform the Red Team/Blue Team exercise. All the attack and defence strategies generations are done in this stage. Furthermore, the interaction of Red Team and Blue Team is analysed through some effectiveness parameters, which are introduced in the system evaluation section.

IV. Experiment Setup

In this section we define network configurations in order to clarify the search spaces for the Red Team and Blue Team.

A. Network Topology

We consider a generic network topology for analysing DoS/DDoS attacks and their countermeasures as shown in Figure 2. In the network, we have a set of source hosts including attackers and legitimate users. Attackers are flooding a victim host that can be a server, which is providing some essential services for the legitimate users. In our simulation, we are using the policy-driven approach proposed by Yu et al. [29] for attack detection and reaction phases. An anomaly-based detection mechanism is used with a rate limiter for the reaction phase in this approach. The enhanced router in Figure 2 is responsible for data collection and rate limiting behaviour. The studied simulation consists of 20 subnetworks which are interacting through the enhanced router. Each subnetwrok consists of several hosts and routers.

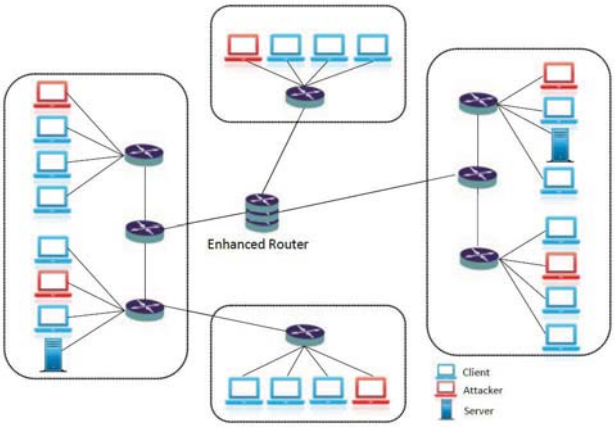


Fig. 2. Topology specification for the case-study

During the simulation time, 34 hosts are randomly selected to launch the DDoS attacks on the victim server.

B. Attack and Defence Strategies

We need to generate a comprehensive set of attacks and defence strategies to analyse the behaviour of the system. Several parameters are involved in each side of the system (attacker vs. defender). For the attacker side, we mainly consider the attack packet size; attack type; and address spoofing status. The attacking nodes will be randomly selected. The attack types considered in our experiments are TCP flood, Ping flood, and UDP flood. These three types of attacks are the most popular packet flooding attacks used by DoS attack tools [30].

The attackers usually use IP spoofing techniques to hide their identities on the Internet. They simply forge packets source addresses with a non-existent computer IP address [31]. This parameter in our experiments is a boolean parameter which can be activated in the case of IP spoofing flooding attacks.

On the other hand, we define our defence strategies based on these three parameters: flow limit rate, link limit rate, and IP limit rate. The detection and response phases are controlled by these three parameters. In [32], the authors integerated several reusable modules into OMNeT++ for detection of anomalies in the simulated network and as a complementary work, Schaeffer-Filho et al. [28] implemented the response phase for remediation of those anomalies. These two frameworks provide a good protection of a network by fast detection and efficient reaction. A summary of the defence process is explained below.

Step 1: The information about the link utilization is extracted by the link monitor module.
Step 2: If the link utilization is higher than a predefined threshold, the anomaly detector raises an alarm.
Step 3: The response mechanism (rate limiter) starts limiting the link capacity with a given link limit rate.

In this stage, the overall impact of the attack is reduced without knowing the cause of attack.
Step 4: The detector starts investigating the cause of anomaly and finding the victim IP address.
Step 5: After a period of time, rate limiter starts limiting the flows toward the victim with a given IP limit rate.
Step 6: A classifier module identifies the specific attack flows and,
Step 7: Finally, the rate limiter starts limiting the detected attack flows with a given flow limit rate.

In Step 3 and Step 4, the rate limiter will affect both normal and attack traffic.

V. System Evaluation

In this section, we show selected initial results from a total of 81 interactions. We have three types of DoS attacks in our experiments and for each type, the defence strategies are varying by changing the values of flow limit rate, link limit rate, and IP limit rate from three levels: Low: 30, Moderate: 60, and High: 90.

Each simulation run was for 300 seconds. All the users including normal clients and attackers start generating traffic using a uniform random generator function. Thus, different sets of traffic are used in the simulations to produce more realistic results. Figure 3 shows the amount of packets per second on the network link to the victim. It illustrates how a defence strategy affects the behaviour of overall (benign/malicious) traffic. A TCP flood attack is shown in Figure 3a while Ping flood and UDP flood attacks are depicted in Figure 3b, 3c respectively. The attack is started at approximately 40 seconds, and then the detector will start generating the alarms to the rate limiter in order to mitigate the DoS attacks. The results show that the defence mechanism used in this experiment is more effective in detecting and filtering TCP flood attacks than Ping flood and UDP flood attacks. As shown in Figure 3a, the defence mechanism was able to mitigate the TCP flood attacks by filtering the malicious traffic. After a period of time, all the remaining traffic is related to the normal background traffic.

For comparing the effectiveness of attack and defence strategies, there are two parameters measured for each simulation run. In our Red Team/Blue Team exercise, the Red Team is trying to maximize the false alarm rate while minimizing the detection rate. On the other hand, a high detection rate with an acceptable false alarm rate shows the effectiveness of the Blue Team. Due to the lack of space, 32 interactions are listed in Table I. For example in the third row, the defence strategy selected by the Blue Team (60%, 60%, 90%) against a TCP flood attack (64, TCP, True) selected by the Red Team, was able to detect 94.60% of attacks destined for the victim with 1.93% false alarm rate. This can be compared with the seventh row of the table in which the Red Team changed its strategy (64, TCP, False) and the previous Blue strategy is not the optimal solution. Therefore, the Blue strategy should be modified with a better setting to mitigate the Red Team. As another example to show how the defence response selected by the Blue Team depends on the attack strategy picked by the Red Team, a 30%, 60%, 90% Blue strategy was able to mitigate a 64, Ping, True Red strategy, while it was not very effective on a Red strategy with a bigger attack packet size (1000, Ping, True).

All the tested simulations in this section are in the category of traditional RT exercises. In this category a human is responsible for checking the testing process, statistics collection, and result evaluation [19]. As the space of Red Team and Blue Team strategies grow in size, the searching process for the optimal solutions becomes impractical. In order to evolve both Red and Blue strategies in the RT exercises, co-evolution is a step forward to find optimal solutions [11]. In co-evolutionary algorithms, the fitness of an individual is measured based on the interaction of individuals with other individuals [33]. Therefore, the process of interaction between individuals leads toward finding increasingly innovative attack and defence strategies.

VI. Conclusion

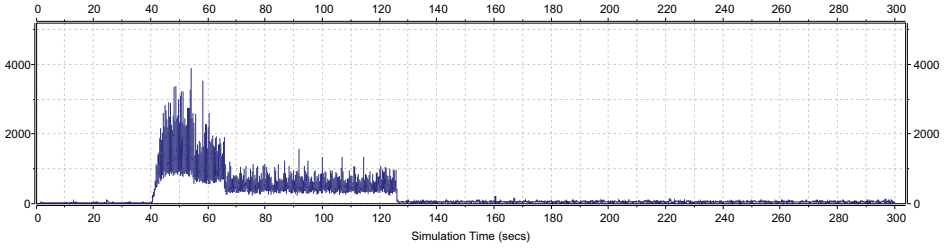
This paper has presented the interaction between the attacker and the defender in a Red Team/Blue Team exercise. The whole process of testing different strategies and statistic collection is done using a simulation platform. The initial results from the simulation showed that one fixed defence mechanism is not always the optimal solution for different DoS attack strategies. Based on the presented results, we conclude that the defence mechanism used in an environment should be dynamically enhanced based on the new sophisticated attack strategies and therefore we need a smarter way of selecting the optimal defensive solutions for our network. For further research into CRT, a tighter interaction of Red Team and Blue Team will be considered using co-evolutionary algorithms. Furthermore, additional topologies, attacks, and defence mechanisms will be considered in the future work.

Acknowledgment

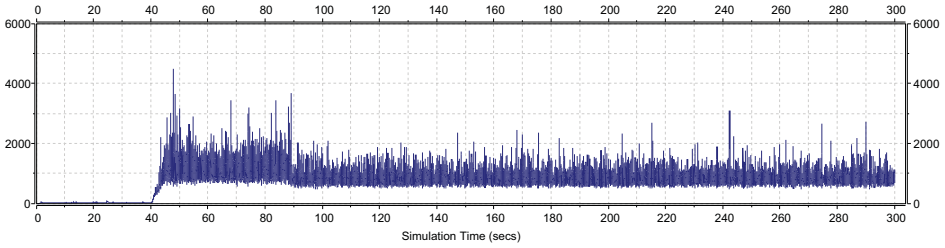
The authors would like to thank Alberto Schaeffer-Filho, Thomas Gamer, and Christoph P. Mayer for their valuable assistance in installing the simulation packages.

References

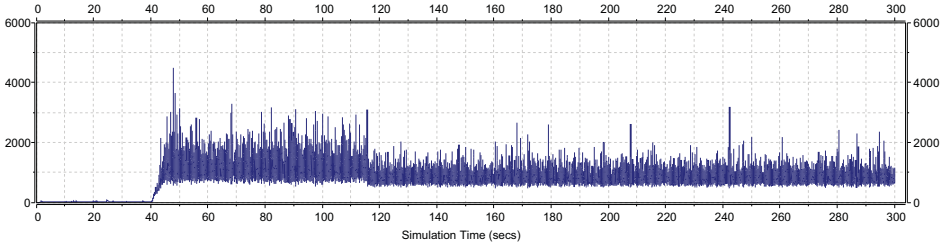
[1] M. J. Schwartz, "Denial of service attacks increased sharply in 2010," 2011. [Online]. Available: http://www.informationweek.com/news/security/attacks/229301118?subSection=News
[2] J. Mölsä, "Mitigating denial of service attacks: A tutorial," *Journal of computer security*, vol. 13, no. 6, pp. 807–837, 2005.
[3] S. Northcutt, "Network intrusion detection: An analyst's hand-book," 2000.
[4] P. Papadimitratos and Z. Haas, "Securing the internet routing infrastructure," *Communications Magazine, IEEE*, vol. 40, no. 10, pp. 60–68, 2002.
[5] I. Cisco Systems, "Characterizing and tracing packet floods using cisco routers," 2003.



(a)



(b)



(c)

Fig. 3. Initial results from simulations ((a) TCP flood attacks (b) Ping flood attacks (c) UDP flood attacks)

[6] A. Householder, A. Manion, L. Pesante, G. Weaver, and R. Thomas, "Managing the threat of denial-of-service attacks," Technical report, CMU Software Engineering Institute CERT Coordination Center, Tech. Rep., 2001.

[7] C. Haag, G. Lamont, P. Williams, and G. Peterson, "An artificial immune system-inspired multiobjective evolutionary algorithm with application to the detection of distributed computer network intrusions," *Artificial Immune Systems*, pp. 420–435, 2007.

[8] T. Peng, C. Leckie, and K. Ramamohanarao, "Survey of network-based defense mechanisms countering the dos and ddos problems," *ACM Computing Surveys (CSUR)*, vol. 39, no. 1, p. 3, 2007.

[9] J. Mölsä, "Effectiveness of rate-limiting in mitigating flooding dos attacks," in *Proceedings of the Third IASTED International Conference on Communications, Internet, and Information Technology*. Citeseer, 2004, pp. 155–160.

[10] J. Mirkovic, S. Fahmy, P. Reiher, and R. Thomas, "How to test dos defenses," in *Conference For Homeland Security, 2009. CATCH'09. Cybersecurity Applications & Technology*. IEEE, 2009, pp. 103–117.

[11] H. Abbass, A. Bender, S. Gaidow, and P. Whitbread, "Computational red teaming: past, present and future," *Computational Intelligence Magazine, IEEE*, vol. 6, no. 1, pp. 30–42, 2011.

[12] C. Peake, "Red teaming: The art of ethical hacking," *SANS Institute*, 2003.

[13] T. Back, U. Hammel, and H. Schwefel, "Evolutionary computation: Comments on the history and current state," *Evolutionary computation, IEEE Transactions on*, vol. 1, no. 1, pp. 3–17, 1997.

[14] S. Chong, P. Tiňo, and X. Yao, "Measuring generalization performance in coevolutionary learning," *Evolutionary Computation, IEEE Transactions on*, vol. 12, no. 4, pp. 479–505, 2008.

[15] M. Ahmadi and D. Maleki, "A co-evolutionary immune system framework in a grid environment for enterprise network security," *networks*, vol. 11, p. 12, 2006.

[16] C. C. Seng, C. C. Lian, L. K. M. Spencer, and O. W. S. Darren, "A co-evolutionary approach for military operational analysis," in *Proceedings of the first ACM/SIGEVO Summit on Genetic and Evolutionary Computation*, ser. GEC '09. New York, NY, USA: ACM, 2009, pp. 67–74. [Online]. Available: http://doi.acm.org/10.1145/1543834.1543845

[17] C. Choo, C. Chua, and S. Tay, "Automated red teaming: a proposed framework for military application," in *Proceedings of the 9th annual conference on Genetic and evolutionary computation*. ACM, 2007, pp. 1936–1942.

[18] M. Lee, D. Ang, and L. Hung, "Team 7: Applying automated red teaming in an urban ops scenario," *Scythe 1: Proceedings and Bulletin of the International Data Farming Community*, pp. 24–30, 2005.

[19] J. Mirkovic, P. Reiher, C. Papadopoulos, A. Hussain, M. Shepard, M. Berg, and R. Jung, "Testing a collaborative ddos defense in a red team/blue team exercise," *Computers, IEEE Transactions on*, vol. 57, no. 8, pp. 1098–1112, 2008.

[20] C. Papadopoulos, R. Lindell, J. Mehringer, A. Hussain, and R. Govindan, "Cossack: Coordinated suppression of simultaneous attacks," in *DARPA Information Survivability Conference and Exposition, 2003. Proceedings*, vol. 1. IEEE, 2003, pp. 2–13.

[21] J. Mirkovic and P. Reiher, "D-ward: a source-end defense against flooding denial-of-service attacks," *Dependable and Secure Computing, IEEE Transactions on*, vol. 2, no. 3, pp. 216–232, 2005.

[22] T. Gamer and C. Mayer, "Simulative evaluation of distributed attack detection in large-scale realistic environments," *Simulation*, vol. 87, no. 7, pp. 630–647, 2011.

[23] I. Kotenko and A. Ulanov, "Agent-based simulation of ddos attacks and defense mechanisms," *Journal of Computing*, vol. 4, no. 2, pp. 16–37, 2005.

[24] I. Kotenko and E. Ulanov, "Simulation of internet ddos attacks and defense," in *In Proc. of ISC*, 2006, pp. 327–342.

[25] T. Gamer and M. Scharf, "Realistic simulation environments for ip-based networks," in *Proceedings of the 1st international conference on Simulation tools and techniques for communications, networks and*

TABLE I
INTERACTION OF RED TEAM AND BLUE TEAM

Red Team (Attack strategy)	Blue Team (Defence strategy)	Detection Rate	False Alarm Rate
Attack packet size, attack type, address spoofing status	flow limit rate, link limit rate, IP limit rate		
64, TCP, True	30%, 30%, 60%	93.97%	33.32%
64, TCP, True	30%, 60%, 90%	93.76%	2.20%
64, TCP, True	60%, 60%, 90%	94.60%	1.93%
64, TCP, True	90%, 90%, 60%	94.09%	31.18%
64, TCP, False	30%, 30%, 60%	93.81%	4.63%
64, TCP, False	30%, 60%, 90%	93.60%	4.20%
64, TCP, False	60%, 60%, 90%	93.27%	25.56%
64, TCP, False	90%, 90%, 60%	79.94%	20.05%
1000, TCP, True	30%, 30%, 60%	92.56%	0%
1000, TCP, True	30%, 60%, 90%	93.70%	0%
1000, TCP, True	60%, 60%, 90%	92.57%	0%
1000, TCP, True	90%, 90%, 60%	93.98%	21.04%
1000, TCP, False	30%, 30%, 60%	92.33%	0%
1000, TCP, False	30%, 60%, 90%	93.44%	0%
1000, TCP, False	60%, 60%, 90%	92.60%	0%
1000, TCP, False	90%, 90%, 60%	93.81%	0%
64, Ping, True	30%, 30%, 60%	4.34%	79.14%
64, Ping, True	30%, 60%, 90%	21.47%	6.06%
64, Ping, True	60%, 60%, 90%	13.07%	9.47%
64, Ping, True	90%, 90%, 60%	24.67%	62.39%
1000, Ping, True	30%, 30%, 60%	6.42%	68.45%
1000, Ping, True	30%, 60%, 90%	19.21%	39.83%
1000, Ping, True	60%, 60%, 90%	19.54%	6.76%
1000, Ping, True	90%, 90%, 60%	19.62%	6.29%
64, UDP, True	30%, 30%, 60%	6.36%	21.34%
64, UDP, True	30%, 60%, 90%	46.08%	53.91%
64, UDP, True	60%, 60%, 90%	16.40%	63.97%
64, UDP, True	90%, 90%, 60%	20.90%	67.96%
1000, UDP, True	30%, 30%, 60%	5.30%	57.31%
1000, UDP, True	30%, 60%, 90%	12.41%	10.06%
1000, UDP, True	60%, 60%, 90%	20.31%	15.36%
1000, UDP, True	90%, 90%, 60%	12.77%	46.45%

systems & workshops. ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering), 2008, p. 83.

[26] T. Gamer, C. Mayer, and M. Zitterbart, "Distack–a framework for anomaly-based large-scale attack detection," in *Emerging Security Information, Systems and Technologies, 2008. SECURWARE'08. Second International Conference on*. IEEE, 2008, pp. 34–40.

[27] A. Varga, "Inet framework, 2007."

[28] A. Schaeffer-Filho, P. Smith, and A. Mauthe, "Policy-driven network simulation: a resilience case study," in *Proceedings of the 2011 ACM Symposium on Applied Computing*. ACM, 2011, pp. 492–497.

[29] Y. Yu, M. Fry, A. Schaeffer-Filho, P. Smith, and D. Hutchison, "An adaptive approach to network resilience: Evolving challenge detection and mitigation," in *Design of Reliable Communication Networks (DRCN), 2011 8th International Workshop on the*. IEEE, 2011, pp. 172–179.

[30] C. Center, "Trends in denial of service attack technology," *World Wide Web, http://www. cert. org/archive/pdf/DoS_trends. pdf*, 2001.

[31] S. Lee and C. Shields, "Challenges to automated attack traceback," *IT professional*, vol. 4, no. 3, pp. 12–18, 2002.

[32] T. Gamer and C. Mayer, "Large-scale evaluation of distributed attack detection," in *OMNeT++ 2009: Proceedings of the 2nd International Workshop on OMNeT++(hosted by SIMUTools 2009)*, 2009.

[33] R. Dreżewski and L. Siwik, "Agent-based co-evolutionary techniques for solving multi-objective optimization problems," 2008.