Rate-Limiting Evaluation using an Intelligent DDoS Attacker

Jeremy Pattison

Department of Computer Science and Software Engineering, The University of Melbourne, Australia

Abstract

DDoS defence evaluation is an important task in Network Security to measure the effectiveness of the defender and to identify vulnerabilities prior to deployment. Commonly defence proposals are evaluated through attack replications mimicking typical attack traffic behaviour. A major drawback of the approach is that it does not consider an intelligent attacker thus does not inform researchers of defence effectiveness for when the attacker seeks to exploit structural vulnerabilities.

In this paper, we focus on how to address this drawback by way of an automated penetration tool that adapts to the availability of resources and a deployed defender. We devise a novel approach of automating DDoS penetration testing by training an Intelligent DDoS Attacker (IDA) through Deep Reinforcement Learning. We demonstrate the advantageous of our approach by measuring defence effectiveness in terms of legitimate packet loss when the network is exposed to attacks patterns generated by IDA compared to commonly tested attacks. IDA is seen to either match or outperform the traffic loss obtained when replicating attack techniques found in prior papers. Analysis of the attack patterns generated by IDA exposes an undocumented vulnerability against the MARL rate-limiting defender.

In this paper, we focus on how to address this drawback via an automated penetration tool that adapts to the availability of resources and a deployed defender. We devise a novel approach of automating DDoS penetration testing by training an Intelligent DDoS Attacker (IDA) through Deep Reinforcement Learning. To demonstrate the advantages of our approach, we compare the defence effectiveness when the network is exposed to attacks patterns generated by an IDA with commonly tested attacks, in terms of legitimate packet loss.

Keywords: Denial of Service Attacks, DDoS Evaluation, Penetration Testing, Reinforcement Learning, Deep Learning

1. Introduction

Denial of Service (DoS) attacks remain an ongoing challenge despite significant research and defence design. They pose significant risks to businesses and infrastructure; a survey estimated a successful Distributed DoS (DDoS) attack to average a cost of $500,000 on the targeted business [1]. It is vital that a proposed defender is evaluated against potential attacks due to the significant cost that a successful attack would incur. Commonly a DDoS defender is tested against algorithmic traffic patterns such as a constant stream or periodic on/off attacks however an attacker may create a tailored attack that seeks to exploit [2] or bypass [3] the network defender. It is difficult to model an intelligent attacker due to the large variance of potential attack behaviour; attackers often adopt a distributed structure similar to Fig 1, the attack may differ based on strength, packet contents and variance of emitted traffic [4]. Ideally, proper evaluation would identify such exploits before deployment, current limitations for investigating robustness of proposed defences risk insufficient testing for network defence.

Evaluation seeks to investigate the suitability of a proposed defence, primarily we are interested in expected network performance incorporating scenarios when the attacker seeks to exploit the system. As a case-study this paper investigates the evaluation of the previously proposed attack reaction defenders AIMD [5] and MARL [6]; the defenders, which respond similarly to congestion by requesting upstream routers to throttle incoming traffic, were compared against DDoS traffic following five algorithmic attack patterns [7]. A sophisticated attack may seek to game rate-limiting defenders

Email address: jpattison@student.unimelb.edu.au (Jeremy Pattison)

so that the network inadvertently drops legitimate traffic thus contributing to the denial of service. Previous evaluations have not investigated the effectiveness against intelligent attackers leaving an open question how to consider the robustness of the defender during DDoS evaluation. In this paper, we present a self learning agent responsible for the attack synthesis used to measure defence effectiveness against intelligent attacks.

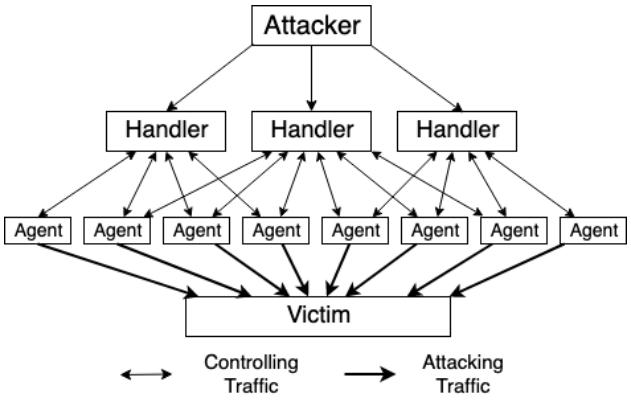


Figure 1: General DDoS Structure [8].

Often proposed mechanisms are evaluated through measuring network performance when the defender is subject to a series of DDoS attacks [9]. Replicating all possible attacks is computationally infeasible, therefore, prior papers generate attacks using commonly seen attack patterns. Mathematical theory can also be used to investigate the robustness of a defence however the approach is costly and of limited utility due to a lack of theoretical tools to model traffic flow [9].

Traditional approaches to evaluation are often inadequate to assess the robustness of a defence. The restricted number of attack scenarios provides insight only for attack patterns tested against and is non-exhaustive. In particular there is limited capacity to analyse how the defender may respond to an intelligent attacker seeking to exploit the system. As attackers and defenders continue to become further sophisticated, investigating only common DDoS attacks risks leaving proposed defensive solutions vulnerable to atypical attacks. Our goal is to identify the most damaging traffic sequence against the defender given a network topology in a cost computationally efficient way. Determining the most effective attack would establish a lower-bound of defence effectiveness when attacked by an intelligent attacker.

Our evaluation utilises an intelligent attacker modelled through Reinforcement Learning to generate attack traffic. The intelligent attacker controls the traffic output of the attack and learns an attacking policy that seeks to minimise the legitimate packets served by the targeted defender. Initially the agent is trained over successive network episodes where the distribution of attacker and legitimate resources is randomly generated. Once trained, the percentage of legitimate traffic obstructed over a defined series of episodes provides a metric of defence effectiveness. Evaluating separate network defenders over the same set of episodes allows comparisons of effectiveness of the defenders against intelligently crafted attack patterns.

A key advantage of our approach is the identification of vulnerabilities that are not apparent through typical evaluation approaches. The agent learns through interaction with the network and is able to adapt its attack based on resources and network constraints. Consequently, the approach can be considered an automated penetration tool whose effectiveness can be used for comparative analysis for defender robustness. Deep Learning is used to address dimensionality challenges caused by the large number of potential states of a network. Our implementation, which we refer to as IDA, extends the prior comparison [6] between the MARL and AIMD rate-limiting systems. IDA is demonstrated to adapt to each defender and either matches or exceeds the most damaging common attack for each given defender. Importantly, the black-box approach allows IDA to be compatible with current attack replication approaches reducing the workload for researchers to incorporate our approach into current methodology. The key contributions for this paper are:

- We introduce a novel automated penetration testing tool to identify traffic flow vulnerabilities against a DDoS attack reaction defence.
- We demonstrate how such tools can be used to create a metric for effectiveness against intelligent attacks allowing comparative analysis.
- We contribute to previous literature by identifying an unexplored vulnerability of MARL which is not apparent through standard evaluation practices.

The remainder of this paper is structured as follows. Section 2 investigates the limitations of current evaluation methodology and motivates the use of Reinforcement Learning to provide an adaptive attacker. Section 3 formally defines the problem statement. Section 4 details the design of our implemented IDA attacker. Section 5 describes the experimental set-up followed by the experimental results in Section 7. Section 8 provides further discussion of the results and Section 9 concludes the paper.

2. Related Work

The purpose of evaluation is to show that the defence is effective [9]. Considering robustness is important for evaluations of attack reaction systems as they can filter incomming traffic thus inadvertently contribute to the DDoS outage.

2.1. Common approaches

We provide a brief overview of common evaluation techniques observed through publications of proposed attack reaction defenders and to what degree robustness is investigated.

2.1.1. Attack Replications

Replicating network attacks allows researchers to observe how the defender performs against an attack. Attacks are often replicated through network emulations or network simulation, emulation provides greater realism however is subject to extensive computational challenges [9, 10]. For the experiment to assess the robustness of the defence, it must include attacks where the attacker seeks to exploit the proposed defender.

Researchers commonly devise attack scenarios specific to the defender being proposed. While reviewed papers differed in the number of network topologies evaluted, the fidelity of the replication and the measured metrics; only a limited number of attacking traffic patterns were observed in each experiment. For example the attacker in StopIT used a constant stream of traffic in both Destination Flooding and Link Flooding scenarios [11], the Fair Throttle [5] and ACC [12] were demonstrated against both pulse attacks of on/off traffic as well as a constant stream whilst MARL was demonstrated against five traffic patterns [6].

Similarly, Recorded DDoS attacks are often incorporated in evaluations to accurately replicate the attack strategies of attackers. For example the evaluation of PSP used DDoS logs to replicate realistic attacks [13]; similarly the AT&T database was used in the evaluation of the AIMD system by providing realistic attack pathways [5]. Databases like NGIDS-DS database [14], which capture stealthy attacks commonly used in penetration attacks, could be used to investigate robustness by mimicking common attacks used to bypass defenders. However whilst further incorporation of databases would increase the realism and breadth of attack replications, such an approach is an expansion of pre-written attacks that fail to explore the robustness of the defender.

The commonality of the discussed approaches is that they do not model an intelligent attacker and instead rely on the vulnerability to be exposed in a pre-written attack scenario. Using common attack scenarios is informative for the effectiveness of the defender against expected attacks. The limited number of investigated attack strategies leaves the performance of the defender untested against atypical attacks, due to the large cost of a successful DDoS attack it is important to establish that the defender isn't vulnerable to attacks outside the commonly seen network attacks. As attackers become increasingly sophisticated, the omission of modelling an intelligent attacker represents a vulnerability in the evaluation of network defenders. The reliance of prewritten attack scenarios of limited attacker behaviour is due to lack of an automated approach of identifying vulnerabilities, therefore current approaches fail to verify the robustness of the defender.

2.1.2. Theory

Theory has been used to create a mathematical model of the network to investigate questions on stability, scalability and robustness for proposed defenders through a mathematical representation of the network.

The network defenders SOS [15], Fair Throttle [5] and Speak-up [16] all used mathematical theory to demonstrate the robustness or stability of the defence in conjunction with attack replications to demonstrate effectiveness. Limitations in modelling network traffic restrict such proofs to critical scenarios as opposed to demonstrating overall robustness [17].

Investigations of strategy may provide insights how an attacker would seek to exploit a proposed defender. *Jakóbik et al.* presented a game theory model for analysing the optimal strategies for both attacker and defender in a DDoS environment [18]. Restricting the focus solely to the attacker would determine the optimal strategy of the attacker against any defender. Using a Markov Decision Process, *Hoffman* proposed a 'simulated penetration model' to automate the discovery of exploits [19]. Such an approach could be used to plan adversarial traffic patterns that maximise network disruption.

Current limitations for replicating the network interactions presents a barrier for using theory to establish the robustness of a defender against traffic attacks. This is due to a lack of theoretical tools for accurate modelling of a network [9] and the considerable cost to the researcher in creating the mathematical representation resulting in simplifications and guesswork. These limitations present a barrier to designing network attacks, so far prior strategy investigations have operated on a high level abstraction where the focus is types of attacks as

opposed to the design of an attack [20, 21, 22]. A sophisticated DDoS attack, which may rely on network interactions or uncertainty that is not captured in the mathematical model would not be replicated through such approaches. Therefore we establish that although a mathematical approach such as a MDP can calculate an optimal strategy, to do so would require a modelling of a network that is yet to be solved.

2.2. Efficient Calculation of Attacking Strategies

The challenge we address is identifying network patterns used by the attacker to minimise the effectiveness of the defender. An intelligent attacker has access to a large number of actions at any period, would interpret a complex network environment and consider the likely payoffs of successive actions rather than focus on the immediate reward. The breadth of these complexities have led to prior evaluation approaches to only consider a restricted subset of attack scenarios.

Previously we identified Markov Decision Processes as a tool to identify how an attacker would game a defender however current limitations in modelling a realistic network and defender prevent researchers from using MDPs to investigate traffic attacks. By contrast, attack replication approaches can model a realistic network however does not calculate the optimal attacking strategy. We identify Reinforcement Learning as an approach that combines the advantages of MDP attack calculation with the realistic network environments provided by live networks or emulations.

2.2.1. Markov Decision Processes

A MDP is a mathematical modelling of the relationship of an agent and the environment [23]. It provides a framework for an agent to determine the optimal action for a given state that maximises a given reward function.

Formally a MDP consists of a set of states S, actions A, rewards R and a probability function $P_a(s, s')$ which represents the probability of action a in state s resulting in state s'. The degree future rewards are valued is set with the discount value γ. Provided a perfect model, the MDP can be solved using dynamic programming maximising the value function through the Bellman Equation:

$$v_*(s) = \sum_{s',r} p(s', r|s, a)[r + \gamma \, max_{a'} \; q_*(s', a')] \quad (1)$$

Previously MDPs have been proposed to simulate how an attacker would compromise a service [19]. To design a DDoS attack, the agent must determine the optimal traffic flow at a given point of time that will result in the greatest expected network disruption requiring an accurate network representation. Until an accurate mathematical representation of the network and defender can be easily created for researchers it is unlikely that MDPs will be used to demonstrate the robustness of proposed defenders.

2.2.2. Simulation Through Reinforcement Learning

Reinforcement Learning is a Machine Learning approach used to simulate the learning of a Markov Decision Process. Like MDPs, the agent must learn a policy that maximises a given reward function for a provided environment. The learning of the agent's policy is loosely coupled to an environment where it learns through interaction. The policy is learnt iteratively through choosing an action and then observing the reward and the consequent state as seen in Fig 2. Critically the environment is a blackbox where the agent does not require complete knowledge of the inner working of the environment [23]. Therefore the agent is able to learn directly from a live environment as opposed to a mathematical representation. Consequently, Reinforcement Learning provides approach to learn and test the optimal policy of the attacker in a live environment compatible with current attack replication techniques.

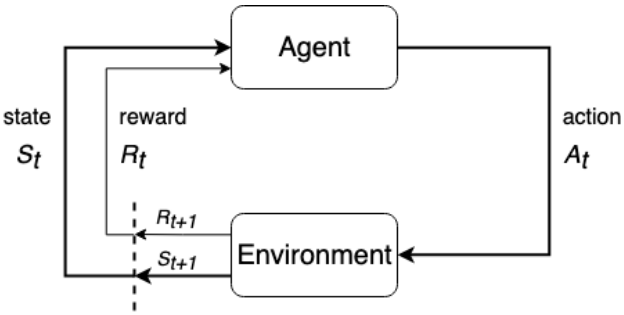


Figure 2: Representation of a Reinforcement Learning Agent [23].

Reinforcement Learning has been previously used to identify which penetrative tool is best suited to compromise an online service [24]. In this paper we shift the focus from coordinating a set of penetration tools to attack synthesis where the agent learns and generates attack patterns.

At any point of time t an agent chooses an action a_t for state s_t receiving reward R_t. The environment object is responsible for determining the consequence and reward of actions. The policy determining the action to take for a given state is stored in a Q-table. The policy is

updated iteratively through interaction with the environment with learning rate α. After each move the Q-table is updated by the rule:

$$Q(S_t, A_t) = Q(S_t, A_t) + \alpha[R_t + \gamma Q_{next} - Q(S_t, A_t)] \quad (2)$$

When there are a large number of states, or the states are continuous, it is necessary to approximate the Q-table. Deep Reinforcement Learning utilises a multi-layered neural network to interpret large states and identify non-linear relationships. The Deep Q-Network (DQN) algorithm is a Q-learning implementation of Deep Reinforcement Learning [25]. Q_{next} represents the expected future expected reward for a given state allowing long term strategy to be created, Q-learning uses a single estimate often leading to maximisation bias by overestimating future rewards during training [23]. Double Learning DQN (DDQN) is a Deep Reinforcement approach that uses two Q_{next} estimations to reduce bias. Identifying the optimal traffic attack would involve interpreting a complex and non-linear network suitable for Deep Reinforcement Learning.

The above discussion highlights the limitations of current DDoS evaluation practices whilst identifying the potential for Reinforcement Learning to craft intelligent attacks. Reinforcement learning provides a mechanism to learn optimal traffic patterns through interaction of a network environment.

2.3. Motivating example

MARL [6] and the Fair Throttle [5] are both rate-limiting defences that limit congestion by filtering upstream packets during an attack. This is achieved by upstream routers setting a 'throttling rate' that discards a selection of incoming traffic, such mechanisms may be gamed by an intelligent attacker resulting in legitimate traffic discarded. The MARL defender is interesting through its use of Reinforcement Learning to learn a throttling policy. A potential danger of Machine Learning defenders is when trained against expected traffic, that they may be susceptible to overfitting resulting in performance drops when exposed to atypical attacks, therefore there is an increased need to establish the robustness of such systems.

A previous comparison of the defenders against five different attack strategies found both defenders to demonstrate a significant decease in effectiveness against pulse attacks aligned to the monitoring window of the defender [6]. The pulse attack demonstrated increased disruption compared to a constant stream of the maximum DDoS rate due to a combination of server failure and inadvertent filtering of legitimate traffic. The limited number of tested attack patterns leaves the possibility of further attack patterns more damaging than currently tested attacks. The cost of investigating all possible traffic patterns increases when considering different scenarios regarding network size and attacking strength. Our goal is to provide a scalable approach to identifying the most effective attack given the defender, network and attacking resources for purposes of network evaluation.

3. Problem Statement

The problem we address is relevant to the evaluation of all network defenders. Network defenders are often measured through performance during attack replications, identifying the optimal traffic attack patterns allows comparisons of defence effectiveness against intelligent attackers. However, in this paper we limit our investigation to the AIMD and MARL network defenders as to investigate the contribution of modelling an intelligent attacker to network evaluation.

The experiment consists of a network defender that we wish to investigate over a set of network environments. A network environment consists of a defender, legitimate users and attacking resources which are coordinated by an attacker, the goal of the defender is maximise the number of legitimate packets served. During the experiment are provided current and past states of the network and defender.

Our goal is to identify traffic flow vulnerabilities of a network defender by determining how an intelligent attacker would minimise the defender's effectiveness. There exists an exponential number of potential attack scenarios, an experiment of T seconds consisting of N attacking Hosts each with A actions would contain A^{T*N} attack scenarios.

Traditionally, evaluations are composed of a mathematical analysis and attack replications. Questions of robustness are investigated through mathematical analysis however this approach incurs a significant cost to the researcher with limited utility due to constraints of modelling network patterns [26]. Therefore, instead of testing against optimal attack patterns researchers often replicate attacks using a predefined set of attack patterns providing insights of performance against standard attacks. However, the approach does not consider consider robustness where an attacker may seek to exploit a deficiency or how such an attacker would minimise the effectiveness of the defender.

Given this background, our aim is to provide an automated approach to identifying exploits through selective attack replications. In creating an attacking strat-

egy we aim to maximise the value function previously solved via the Bellman Equation, due to the lack of a mathematical representation the value function must be approximated through Reinforcement Learning. Not all network defenders can be gamed thus the optimal attacking policy may be one of the common attack patterns observed in prior investigations. Therefore, for the intelligent attacker to be of benefit to defence evaluation, it should either match or outperform all standard attacks when provided with the same attacking resources. We replicate past evaluation experiments by measuring the percentage of legitimate traffic served by the network, through maximising the value function the attacker seeks to minimise the number of legitimate packets served during the attack.

Let $\pi_{1...5}$ represent five attack policies commonly used in attack replications which we detail in Sec 7, let π_* be the learnt policy of intelligent attacker trained over network m.

Let $f(\pi, m)$ be value function measuring the legitimate traffic served during an attack of network m with the attack policy π.

The trained attacker must be at least as effective as the most damaging common attack, therefore given topology M over N randomly generated network distributions, π_* must satisfy:

$$\forall n \in 1..5, \sum_{i=1}^{L} (f(\pi_n, M_i) - f(\pi_*, M_i)) \leq 0 \qquad (3)$$

Key complexities of calculating an optimal attack include a large and continuous state-space, the number of possible actions and complexities of linking successive actions to create a long term strategy. The attacker must interpret a network representation to determine the optimal attack, the state consists of the distribution and strength of each attacking Host as well as prior attacking actions and a measure of their success over a recent time period, values of the state space may be continuous and share non-linear relationships over time.

3.1. Network Model

An example of our network model is presented in Fig 3 and is similar to the one used by *Yau et al.* [5]. The network is a connected graph G=(V,E) where V is the set of nodes and E is a set of bidirectional edges. Nodes may either represent the Server (S), Internal Routers (R) or Hosts (H). The goal of the network is to service all legitimate traffic. Hosts generate traffic and represent either legitimate users or attackers who generate traffic at rates r_l and r_a. Our simulations consider attacks where $r_a >> r_l$ allowing defenders to differentiate attacking

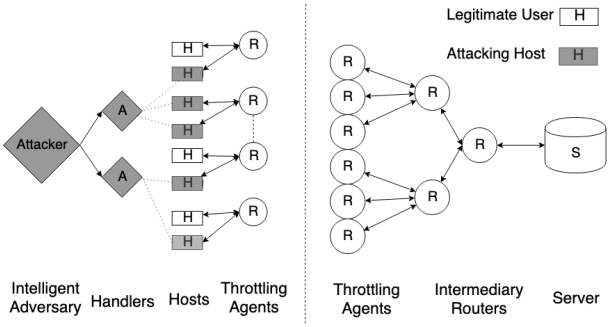


Figure 3: Network Topology demonstrating the learning process of the attacker.

streams by the rate of traffic. Attacks where $r_a \approx r_l$ require more attacking hosts incurring significant cost to the attacker, by limiting the size of the botnet the attacker is challenged to use the available DDoS resources effectively. Network routers can either drop or forward traffic to connecting nodes. The network can service up to U_S packets per second, we adopt a bottleneck located at the server of capacity U_S which was previously modelled by *Malialis and Kudenko* [6]. Throttling Agents are capable of dropping incoming packets to regulate incoming traffic.

We extend the model by including an intelligent attacker modelled through a set of Handlers that sends commands to attacking hosts. Handlers are detached from the network and direct attacking hosts via broadcasts. The attacker can observe the response by the defender to update its attacking policy. After initiating the attack, each Handler directs the volume of traffic emitted by their respective Hosts. Handlers direct Hosts every $T_{attacker}$ seconds with the shared goal of disrupting legal traffic. The network's agents set throttling rates every $T_{defender}$ seconds to achieve the contrary goal of maximising the number of legitimate packets serviced.

4. Intelligent DDoS Attacker (IDA)

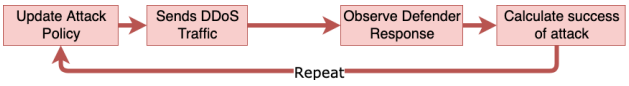


Figure 4: Overview of IDA learning process.

Fig 4 provides an overview of the learning process of the Intelligent DDoS Attacker which we refer to as IDA, a more detailed diagram showing the interactions between the agents and the network is shown in Fig 5. IDA follows an iterative process of observing the current state, choosing an attacking action, observing the

success of the attack, then updating attacking policy based on the success of attack. Each attack is divided into separate time intervals, at the start of each interval IDA must interpret the current state and choose an attacking action. Therefore each attack is outcome of many actions. Over time, given a defence policy, the attacker learns an attack policy that exploits vulnerabilities in the defender.

The design IDA considers four research challenges relevant for Reinforcement Learning agents controlling network traffic. The first challenge is the modelling of the attacker regarding the structural representation and availability of actions. The second challenge is the domain representation of the network presented to the Reinforcement Learning agent. The third challenge is the appropriate learning mechanism and why we chose Deep Reinforcement Learning. Lastly we address the reward given to the attacker and how it is to be distributed to individual groups of attackers.

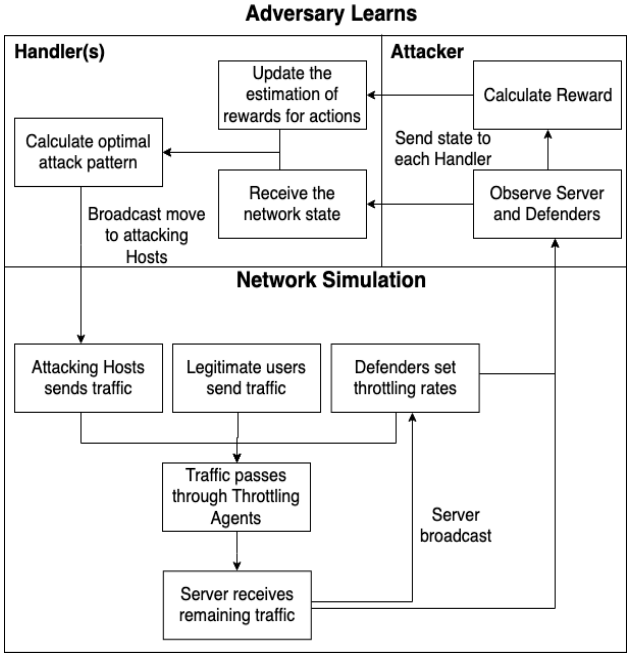


Figure 5: Detailed illustration of the interactions of all actors in the learning of optimal attack patterns.

4.1. Modelling The Attacker

The role of the attacker is, provided attacking resources, to learn the optimal attacking strategy that minimises defence effectiveness. It is difficult to quantify a strategy as each attacking Host operates on a continuous scale for both emitted load and time. In creating an attacker the trade-off we address is reducing the possible number of actions so that an ideal strategy can be found through Reinforcement Learning without restricting the attacker to an extent that the ideal attack strategy is no longer effective. Key challenges in the modelling of the attacker are (1) The availability of actions for each decision making agent and (2) The architectural structure of the attacker.

Representation of each action: The purpose of this work is to identify the sequence of actions that are effective against the specified defender. Each attacking Host can emit a variable proportion of its total load at any point of time during the attack. It is necessary to model the attacker onto a discrete action and time space to formulate a strategy involving multiple continuous Hosts. The attack is separated into a series of decision points where the attacker may change the emitted load of traffic every $T_{attacker}$ seconds. Similarly the availability of actions provided to the attacker limits each attacking bot to N possible actions from $\frac{0}{N-1}...\frac{N-1}{N-1}$ of the potential load. We note that the MARL defender which IDA evaluates similarly encounters the problem of multiple learning agents on a continuous action space and that both investigated defenders used an interval of $T_{defender} = 2$. In evaluating a defender it is necessary to ensure that the attacker's model does not restrict the discovery of the an ideal strategy, however in this paper we adopt a value of $N = 11$ and $T_{attacker} = 2$ to mimic the capabilities of MARL to demonstrate that IDA is capable of exceeding standard practice methodologies even when the action-space is as restricted as the defender. Thus an action of 3 would result in the agent's bots emitting $\frac{3}{10}$ of its potential load for the following 2 seconds.

Structure of decision making body:

We now consider allowing the attacker to exist of multiple groups of attacking Hosts with groups following separate actions. An increase in the number of groups allows more complicated attacks however results in an exponential growth of available actions per step. For an experiment with M groups of Hosts, where each Host has N actions, there exists N^M possible actions at any point of time.

To mitigate the action-space for the learning agent we implement a decentralised decision model where each group of Hosts is controlled by a Handler which contains a separate decision making bot. This structure mimics general structure of a DDoS attack that was shown in Fig 1 and allows flexibility to the researcher to increase the complexity of attacks based on available resources. The use of separate learners to control the attacker reduces the available actions for each agent down to N possible actions. We note Reinforcement Learning faces challenges learning an optimal policy for multiple

simultaneous learners in a non-stationary environment [27]. Investigating the advantageous and challenges of introducing multiple learners is investigated in Sec 7.3.

4.2. Domain Selection

It is necessary to create a state representation of the network to be provided to the learning agent. In doing so we considered 1) the information is the attacker allowed access to 2) ensuring that the provided state is informative but concise to limit what is referred to as the 'curse of dimensionality'.

The first problem we address is the availability of information to the attacker. Unless the network has been compromised, the attacker would normally have limited visibility of the state of the network, therefore an accurate representation of an attacker might consider limiting available information. Such an approach may increase realism but would conflict with our aim to identify the most damaging attack an attacker may choose to establish a minimum expected performance for the attacking resources. Therefore we choose not to restrict information of the current or past state of the network.

It is now necessary to select the features provided in the state that is given to each learning agent. We aim to ensure the state is informative to allow a strategy to be identified, encourages cooperation between Handlers whilst avoiding dimensionality challenges. To encourage collaboration between Handlers our attacker utilises a global state where each Handler receives the same state representation, therefore the policy learnt by each agent considers the actions of all Handlers. Our state incorporates past actions of all Handlers, the defender and server load to allow the attacker to link successive actions for long term rewards. We keep track of the prior three actions providing the necessary context to the attacker whilst limiting the size of the domain. The traffic potential of all Hosts is also provided where the attacker can learn to the emitted load associated with each action. In summary the domain provided to all Handlers consists of the traffic potential of all Hosts as well as the actions of all Handlers and Defenders over the prior three $T_{defender}$ intervals including the traffic load at the server.

4.3. Learning Mechanism

The role of the learning agent is to identify the action most likely cause network disruption for the given state. As detailed in the background reading of Sec 2, Reinforcement Learning was chosen due to its ability to learn a policy in a blackbox where we do not have a theoretical model of the network or defender. The reinforcement learning network defender MARL which was previously examined utilised a Function Approximation learning mechanism using SARSA learning [6].

Reasons for using Deep Reinforcement Learning: The proposed domain representation grows linearly as the network environment expands and is likely to exhibit non-linear relationships. These considerations led us to utilise the Deep Reinforcement Learning DDQN approach where the multi-layered neural network interprets the large and non-linear statespace. Deep Reinforcement Learning is often used for challenges of large state spaces over Function Approximation allowing an expanded state more informative than what could be provided to a Function Approximation solution.

4.4. Reward Structure

A Reinforcement Learning agent learns a policy that seeks to maximise its reward function, therefore the design of the reward must consider how the reward leads to a policy to achieve our desired aim. In designing our reward we considered first the reward that is provided to the attacker and secondly how the reward is distributed to each of the learning agents of the attacker.

The provided reward must be designed so to learn a policy that minimises the effectiveness of the defender. Commonly, and in this paper, network effectiveness is calculated by the percentage of legitimate traffic served throughout the experiment given by variable L resulting in the success of the attacker is calculated by $1 - L$. The attack is made over successive actions however rewarding all actions equally during an episode is inefficient as an attack may include intervals of ineffective attack sequences that should not be rewarded. Therefore, we reward the attacker of the traffic disruptive after each action by $r_{attacker}(l) = 1 - l$, where l is the percentage of legitimate traffic served in the preceding interval. A discount value is introduced to allow successive actions to be rewarded. Notably the provided reward does not negatively reward agents for ineffective attacks, by contrast MARL punished server failure. However unlike MARL, suboptimal short-term actions do not incur additional cost to the attacker. An advantage of lack of reward shaping is it removes potential human influence in identifying the most damaging attack.

The next challenge is how to distribute the reward as there exists multiple learning agents. Rewarding all agents equally can result in reward misattribution where agents are rewarded for ineffective or disadvantageous actions masked by effective actions by other agents. Recall that each agent seeks to maximise its personal reward, not the reward given to the attacker as a whole. A local reward mechanism, where agents are rewarded

based on their personal contribution to the overall result, may result in competitive behaviour where agents fail to coordinate seeking to maximise their personal contribution over long term strategy. Therefore we use a global reward, where agents are rewarded equally to avoid potential selfish behaviour that can arise from local rewards [28].

5. Experimental Objectives

It is difficult to quantify whether implementations IDA are more informative than traditional approaches as an understanding how the defender responds to common attacks would always be necessary. Rather our aim is demonstrate that IDA provides further information by exceeding the effectiveness of traditional replication attacks establishing a new lower bound of defence effectiveness which can then be examined (a) the effectiveness of attack strategies created by IDA relative to commonly used attack strategies against a defender (b) the effect of changes to resource distributions for the formulated attack by IDA (c) the effect of a self learning attacker to identify attack patterns that exceed commonly used attack strategies (d) the effect of increasing the availability of actions for the attacker given the proposed self learning attacker.

As the basis of our evaluation we require a network environment where we can replicate attacks and compare different defenders. Our network as discussed in Sec: 3.1 can drop incoming traffic through upstream Throttling Agents, traffic is produced by Hosts which generate either legal or attacking traffic, the attacker can change the generated traffic every $T_{attacker} = 2$ seconds. The experiment is replicated in two network topologies of 3 Throttling Agent and 9 Throttling Agents with two Hosts for every Throttling Agent. All traffic rates and server capacities are measured in Mbit/s. The 3 defender topology has an upper and lower server boundaries of L_S=6 and U_S=8 whilst the 9 defender topology has boundaries of L_S=17 and U_S=20. We model a $10ms$ delay between adjacent nodes, edges have infinite capacity outside the bottleneck S-R_1 capped at U_S. Our network simulator was written in Python on the SPARTAN HPC machine [31].

6. Experimental Methodology

We study the effectiveness of attack strategies generated by IDA against four network defenders. A set of commonly replicated attacks provide a baseline effectiveness and are described in Tab: Tab. In the attack we label traffic coming from legitimate and attacking Hosts. A successful attack minimises the effectiveness (E) of the defender through disruption of legal traffic which is measured as the legal traffic that is produced (P) over the legal traffic received (R) by the server.

$$E = \frac{R}{P} \tag{4}$$

Defence effectiveness is measured for 500 episodes of differing resource allocation. Using the same set of episodes enables evaluations of defence effectiveness between defenders against an intelligent attacker. As effectiveness is measured per episode we are given a range of effectiveness for each experiment. Comparisons of effectiveness of two attacks (A and B) assess difference of the observed means ($\bar{d}$) against the defender and whether the intervals are statistically significant. We use the paired t-test which is converted to a p-value at significance 0.05 to test statistical significance factoring ($\bar{d}$), the sample variance (s) and number of used episodes n providing the t-value:

Each result represents the mean difference over 500 episodes averaged over 10 repetition; each repetition were run on the same 500 episodes with independently trained learning agents.

$$t = \frac{\bar{d}}{\sqrt{s^2/n}} \tag{5}$$

The experiment is repeated in in two different network topologies each using a separate set of examined episodes. In the creation of each episode a Host had a 0.6 chance of being part of the attack, the traffic potential of legal Hosts and attacking Hosts was uniformly sampled from $r_l \in \{0.05, 1\}$, $r_a \in \{2.5, 6\}$ respectively. This restricts the independent variables to the employed attack strategy and used defender. At least one Host producing legal traffic and one Host producing attacking traffic exist in the episode. During evaluation, episodes last for 120 seconds with the attack beginning at t=10 and finishing at t=110. Our training and evaluation of IDA involves only effective attacks where the total packet capacity exceeds $1.2U_S$.

Against each defender our DDQN attacker was trained during 350,000 randomly generated episodes of 60 seconds length initiating the attack at $T = 10$. There were 50,000 episodes of pre-training followed by 150,000 episodes of linearly decreasing exploration. Our learning rate and discount value was set at $\alpha = 0.005$ and $\delta = 0.8$.

6.1. Defenders

We briefly summarise the four rate limiting defenders IDA is demonstrated on. The chosen defenders examine the performance of IDA against two previously published network defenders, the performance of IDA when the frequency of moves is limited and variation of the Fair Throttle where we observe IDA's capacity to identify a structural defect.

To maintain consistency with prior comparisons [6, 29], unless specified otherwise, each defender makes a move every 2 seconds having observed the prior 2 seconds for their decision. The evaluated defenders are similar in that they filter incoming upstream packets through throttling agents to limit server congestion. For conciseness we define two filtering approaches:

Drop Rate: A throttling agent receives a percentage of incoming traffic to discard .

Forwarding Rate: A throttling agent receives a maximum rate of traffic it can allow through.

6.1.1. MARL Agents

We replicate the MARL Reinforcement Learning to provide a decentralised throttling response [6]. MARL uses a drop rate where each throttling agent is provided a set percentage of traffic to discard until the following instruction, at each throttling agent a Reinforcement Learning Function Approximation learner is installed.

To improve the effectiveness of MARL, the learning rate was reduced to $\alpha = 0.01$ while the length of training was extended to 120,000 episodes of attacks. Training included 20,000 episodes of pretraining followed by 60,000 episodes of linearly decreasing exploration.

A variant of MARL named MARL-DS was created, it is designed to challenge IDA by providing an opponent able to respond quicker than IDA. MARL-DS utilises the same DDQN learner as IDA, it is rewarded based on the evaluation performance rather than being punished for congestion and produces a move every 0.5 seconds as opposed to every two seconds.

6.1.2. Fair Throttle Agents

The Fair Throttle by contrast sets a global forwarding rate determined by the AIMD algorithm limiting the volume of traffic that can pass through each router. Therefore if the incoming load was to drop below this threshold, the router would allow all traffic to pass [12]. The functionality is similar to the Traffic Policing mechanism supported by Cisco routers that support 'Cisco Express Forwarding' [30].

We introduce a variant, the Fixed Throttle, whose differentiation is the use of a drop rate that discards a specified percentage of incoming traffic. The approach is an approximation of the Fair Throttle and examines IDA's ability to identify exploits in suboptimal network conditions.

7. Evaluation Results

7.1. Evaluation of attacks

The effect on network disruption comparing IDA and the standard attacks is displayed in Fig: 6, Fig: 6a and Fig: 6b show the experiment repeated on a 3 and 9 defender topology respectively. The results show the mean difference of the percentage of disrupted legal traffic between IDA and each standard attack, statistical significance is demonstrated through error bars which represent the 95% confidence interval.

The results show that IDA has increased the mean effectiveness of the attack compared to all investigated attacks, these results are statistically significant for all results outside the Fixed Throttle in Fig: 6b which recorded $p = 0.3$ against the Constant Attack. We can see that despite defenders demonstrating vulnerabilities against either Constant Traffic or Pulse Attacks, IDA designed an attack policy that matched or exceeded the effectiveness of all attacks against each defender. Thus, using our intelligent attacker, we create tailored attacks that matched or exceed all standard attacks satisfying the first two experimental objectives of this paper.

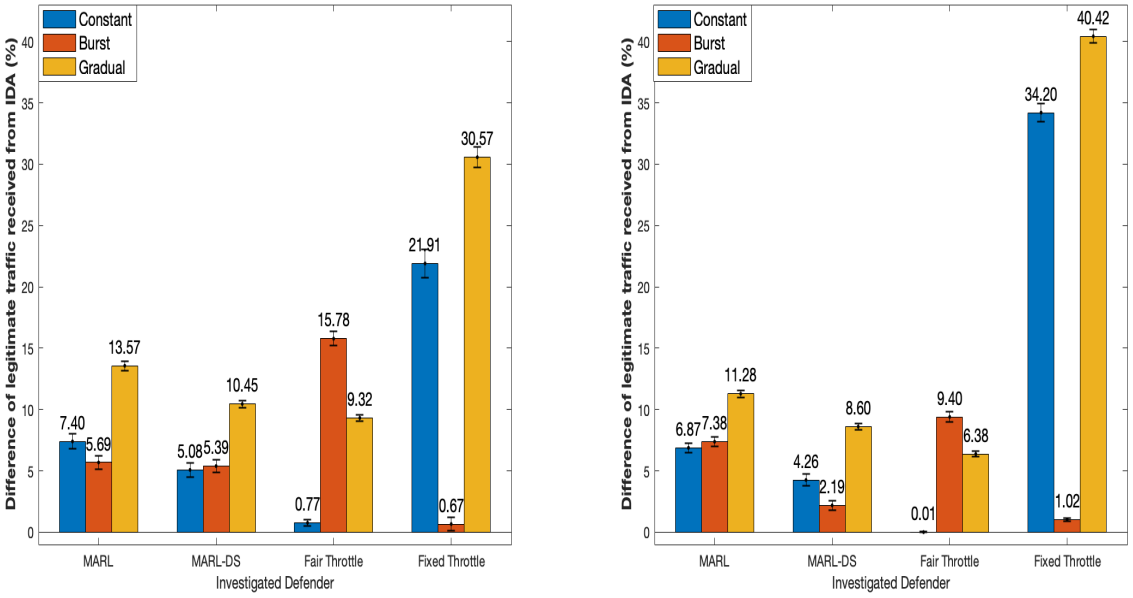
7.2. Analysing IDA attacks

While our previous results demonstrate that IDA is able to generate attacks that exceed the baseline performance, to understand exposed vulnerabilities we want to examine when and how did IDA outperform all standard attacks.

7.2.1. Comparison of attack distributions

In Fig: 6b we observed IDA to demonstrate a greater mean disruption compared to the Constant Attack against the Fair Throttle in the Nine Defender Topology, therefore exists a subset of episodes where an attack outside a Constant Attack would be advantageous.

Fig 7 displays the effectiveness of IDA with the Constant Attack for the Fair Throttle displaying the percentage of legal traffic served in each episode by the attackers, episodes have been ordered by the effectiveness of the Constant Attack of that episode. We see that the attack patters generated are similarly effective across the majority of episodes. The figure clearly shows IDA to outperform the Constant Attack when such an attack would result in less than approximately 5% of legitimate traffic disrupted.



(a) Three Defender Topology.

(b) Nine Defender Topology.

Figure 6: Comparison of legitimate packets served by both IDA and standard attacks. IDA is seen to reduce the traffic served compared to all standard attacks.

Attack Name	Description
Constant Attack	Attacker maintains constant traffic rate
Pulse Attack	Attacker sends traffic for two seconds followed by two seconds of no traffic sent
Gradual	The attack is increased linearly from 0 to the maximum rate over the first half of the attack where the maximum rate is maintained
IDA	Each Handler directs the attacking traffic rate of their Hosts every two seconds, the Handler learns a policy through Reinforcement Learning

It is clear that IDA has developed a policy that matches the Constant Attack unless the resource distribution would ensure that such an attack would be ineffective. In other words, IDA considers the provided resource distribution to design an attack. Interestingly there are several episodes where Constant Attack drops a greater amount of traffic. This result highlights that our training mechanism is an approximation of the ideal attack and can be outperformed.

7.2.2. Investigating generated attack patterns

We have identified that IDA diverges from the Constant Attack when such an attack would result in less than 5% traffic dropped against the Fair Throttle. Similarly IDA generated an attack that exceeded all attacks agsint the MARL defender. In this section we analyse the attack patterns to provide insights of potential vulnerabilities.

Fig 8a shows a snapshot of attack behaviour from an episode that outperformed the Constant Attack from Fig 7. We observe that IDA performs a Constant Attack of 6 second intervals of no traffic aligning with the deactivation period of the provided AIMD algorithm. Here it is clear that IDA has learnt a policy that exploits the functionality of the AIMD algorithm and chose to sacrifice 6 seconds with no disruption to reset the Fair Throttle.

The snapshot against MARL in Fig 8b shows a unique variant of the Short-Pulse converging on a lower bound of 10% during the 'off-period' as opposed to 0% of attacking traffic. In this episode IDA reduced the legitimate traffic served by 5.73% relative to the next most damaging standard attack. An emission of 10% of the adversarial potential would see the attacking Hosts produce traffic at similar rates as seen by legitimate users; without explicit training the attacker found mimicking user traffic loads to be advantageous against MARL.

Our investigation saw IDA match or exceed the effectiveness of all other attacks, it is evident that IDA achieved this by adapting its attack considering the pro-

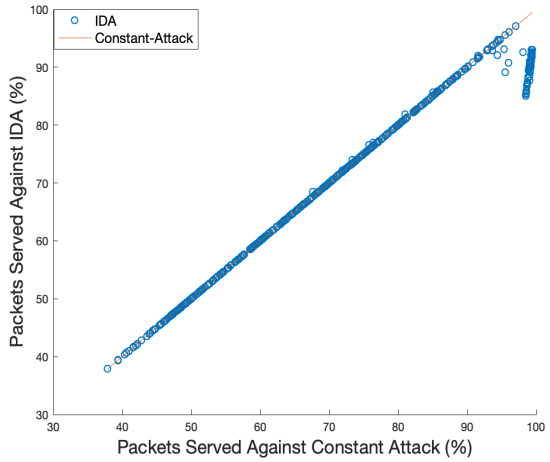


Figure 7: Divergence of network disruption between IDA and Constant Attack against the Fair Throttle in the Nine Defender topology. IDA is seen to deviate when the Constant Attack would disrupt less than 5% of legitimate traffic.

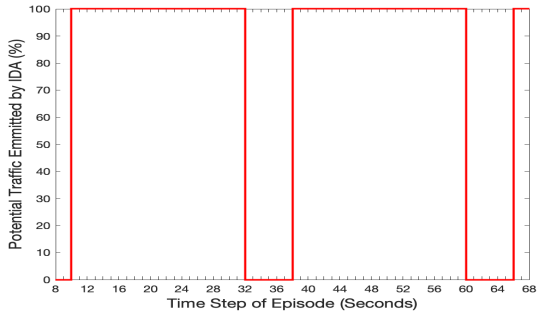
No. Handlers	1	2	3	4	5
FairThrottle	79.85	79.96	80	80.28	80.41
FixedThrottle	45.27	45.93	45.93	45.58	46.19
MARL	68.61	69.85	70.34	70.66	72.42
MARL-DS	77.4	76.58	75.24	74.37	74.13

Table 1: Percentage of legitimate traffic served as the number of Handlers is increased in the Nine Defender topology. Underlined cells mark the existence of a standard attack that exceeded IDA at statistically significant levels.

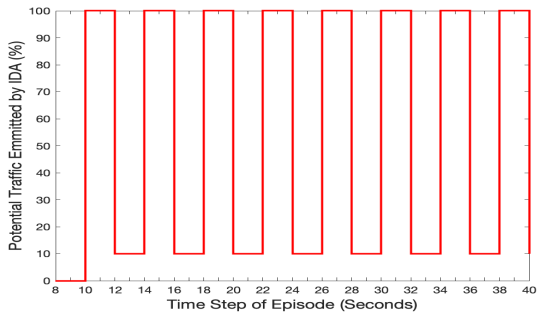
posed defender and available resources. By providing a new baseline performance of the attacker we aid network evaluation using a mechanism that can detect edge cases based on provided resources. The analysis of attack patterns is useful for defence design by alerting the researcher of invalid assumptions or unseen vulnerabilities not evident through standard evaluation.

7.3. Multiple Handlers

In Sec 4 we provided a learning mechanism capable of distributed attacks with individualised attack streams. Our previous experiments have considered a centrally controlled attack like the attack patterns generated from the standard attacks used. In this experiment we examine the effect of increasing the number Handlers where each Handler has an individual Reinforcement Learning agent that must coordinate with other Handlers. Multiple learning agents are often needed to address the 'curse of dimensionality' in Reinforcement Learning as the state space and action space grows [28].



(a) Observed Long Burst Variant generated by IDA against the Fair Throttle.



(b) IDA mimics legitimate traffic loads to bypass MARL.

Figure 8: Observed attack patterns by IDA against the Fair Throttle and MARL in the Nine Defender Topology.

In our design additional Handlers allowed for individualised attack streams however the existence simultaneous learning agents can obstruct the formation of an optimal policy.

Table 1 examines attack effectiveness for the nine defender topology for IDA variants where the number of Handlers is increased from 1 to 5. We display the percentage of legal traffic that is served across all episodes where a lower percentage indicates a more effective attack. In the case of MARL-DS, an increase of the provided Handlers resulted in an increase of attack effectiveness by 3.3%. Other defenders observed decreased effectiveness of the attack where the Fair Throttle performed below the Constant Attack at statistically significant levels. The decrease of attack effectiveness demonstrate difficulties in learning a coordinated attack policy between Handlers.

8. Discussion

Against each defender in each topology the centralised IDA matched or outperformed the most damaging standard attack given the same resources. By

converging on an attack policy that matched or outperformed all standards attacks we demonstrate that a Reinforcement Learning attacker provides an adaptive evaluator. It demonstrated a clear capacity to model an attack considering the the defender and the provided resources which can be used as a metric of defence effectiveness against intelligent attackers. Analysis of the attack can aid defence design by identifying new vulnerabilities. For example, our investigation discovered a new attack pattern effective against MARL, conversely it was clear that IDA only outperformed standard attacks for ineffective attacks.

The semi-decentralised structure of the attacker enabled expansions of IDA that employed individualised attack streams. Simultaneous learning of Reinforcement Learning agents was used to mitigate dimensionality challenges of an increased number of attacks. The multiple learning agents struggled to coordinate against other defenders leaving an open research challenge discussed in Sec 8.0.1.

Past research in DDoS design has used Reinforcement Learning to develop adaptive defenders, a complexity for this approach is that suboptimal policies will result in financial costs for the victim due to insufficient protection against attacks. The long training times associated with Reinforcement Learning and the challenges in calculating accurate rewards, given the difficulties in distinguishing DDoS traffic, elongate the risk of a successful attack whilst the defender learns or adapts to a live environment. Reinforcement Learning may be more easily used in practice for evaluating defences for attack synthesis where evaluation can occur in a closed environment as DDoS traffic can be tracked and training times are affordable.

8.0.1. Scalability Concerns

Our approach used multiple learning agents to address the exponential growth of potential actions caused by the addition of Handlers. The computational expense from multiple Deep Learning agents as well as the extensive network simulation needed for training would require a distributed framework for larger simulations. In Sec 7.3 we observed that additional Handlers can increase performance of the attacker however often the Handlers struggle to coordinate as evident by the decreasing effectiveness of the attack against all defenders outside MARL-DS. Deep Reinforcement Learning often samples past experiences through Experience Replay to avoid unreliable and slow learning [32]. Experience Replay assumes a constant environment which is challenged by the existence of multiple agents simultaneously learning [33]. The sampling of past rewards would see the agent rewarded for scenarios not reflective of the current environment. *Foerester et al.* has provided two possible approaches to mitigate disadvantageous sampling that may be on benefit to larger simulations [33]. A further challenge stems from the use of a global reward; the use of a global reward for a small number of agents encourages cooperation however larger systems may experience the credit-assignment problem where learning agents fail to determine which individual actions contributed to the reward leading to suboptimal policy creation. Alternative mechanisms that balance individual rewards and cooperative behaviour are well documented in *Panait et al.* [28].

We introduced multiple learning agents to mitigate what is commonly referred to as the 'curse of dimensionality' due to an exponentially increasing action space. An alternative approach would be to centralise the decision making, similar to the experiments in Sec 7.1, where each learning agent is responsible for a larger number of actions. This will incur a new challenge due to the significantly larger amount of actions that each Handler would be responsible for. There has been limited research in Reinforcement Learning for large discrete action spaces which may pose a challenge for more sophisticated environments, *Dulac-Arnold et al.* demonstrated an Actor-Critic policy architecture where an agent with up to 1 million actions was able to formulate an effective policy [34].

8.0.2. Further Application to Network Defence

We chose to emulate a DDoS attacker in a rate-limiting environment, which did not consider the contents of generated packets so that the action space could be restricted, reducing the size of the Q-table. As Reinforcement Learning becomes more sophisticated the capacity of simulating an expanded attacker, not limited to traffic levels, becomes more feasible. The model free learning system can adapt for similar tasks requiring only modifications to the reward mechanism to reflect the new goal. Therefore there exists the potential to use Reinforcement Learning as a blackbox penetration testing tool for network defence.

9. Conclusion

In this paper we proposed the use of Reinforcement Learning as an evaluation tool for rate-limiting systems. Our results show that IDA learnt an attack policy that equalled or surpassed the algorithmic attacks typically used and can be used to identify vulnerabilities not apparent through standard methodology. Our model free

learning mechanism provides an attacker that adapts to proposed defenders and networks. Incorporation of similar methodology into standardised test-beds would extend evaluation practices by providing a realistic intelligent attacker.

10. Conflict of interests

The authors declare that there is no conflict of interest in the subject matter of this paper.

11. Acknowledgements

This research was undertaken using the LIEF HPC-GPGPU Facility hosted at the University of Melbourne. This Facility was established with the assistance of LIEF Grant LE170100200.

References

[1] T. Matthews, What ddos attacks really cost businesses?, Tech. rep., Incapsula (2014).

[2] S. S. Kanhere, A. Naveed, A novel tuneable low-intensity adversarial attack, in: The IEEE Conference on Local Computer Networks 30th Anniversary (LCN'05) l, IEEE, 2005, pp. 8–pp.

[3] T. Vissers, T. Van Goethem, W. Joosen, N. Nikiforakis, Maneuvering around clouds: Bypassing cloud-based security providers, in: Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security, ACM, 2015, pp. 1530–1541.

[4] T. Peng, C. Leckie, K. Ramamohanarao, Survey of network-based defense mechanisms countering the dos and ddos problems, ACM Computing Surveys (CSUR) 39 (2007) 1–42.

[5] D. K. Yau, J. C. Lui, F. Liang, Y. Yam, Defending against distributed denial-of-service attacks with max-min fair server-centric router throttles, IEEE/ACM Transactions on Networking 13 (1) (2005) 29–42.

[6] K. Malialis, D. Kudenko, Multiagent router throttling: Decentralized coordinated response against ddos attacks, in: Twenty-Fifth IAAI Conference, Vol. 25, Innovative Applications of Artificial Intelligence, 2013, pp. 1551–1556.

[7] K. Malialis, S. Devlin, D. Kudenko, Distributed reinforcement learning for adaptive and robust network intrusion response, Connection Science 27 (2015) 234–252.

[8] S.-C. Lin, S.-S. Tseng, Constructing detection knowledge for ddos intrusion tolerance, Expert Systems with applications 27 (2004) 379–390.

[9] J. Mirkovic, S. Fahmy, P. Reiher, R. K. Thomas, How to test dos defenses, in: 2009 Cybersecurity Applications & Technology Conference for Homeland Security, IEEE, 2009, pp. 103–117.

[10] D. M. Nicol, Scalability of network simulators revisited, in: Proceedings of the Communication Networks and Distributed Systems Modeling and Simulation Conference, Vol. 28, 2003.

[11] X. Liu, X. Yang, Y. Lu, To filter or to authorize: Network-layer dos defense against multimillion-node botnets, in: ACM SIGCOMM Computer Communication Review, Vol. 38, ACM, 2008, pp. 195–206.

[12] R. Mahajan, S. M. Bellovin, S. Floyd, J. Ioannidis, V. Paxson, S. Shenker, Controlling high bandwidth aggregates in the network, ACM SIGCOMM Computer Communication Review 32 (2002) 62–73.

[13] J. C.-Y. Chou, B. Lin, S. Sen, O. Spatscheck, Proactive surge protection: a defense mechanism for bandwidth-based attacks, IEEE/ACM Transactions on Networking (TON) 17 (6) (2009) 1711–1723.

[14] D. Hadfield-Menell, S. Milli, P. Abbeel, S. J. Russell, A. Dragan, Inverse reward design, in: Advances in neural information processing systems, Vol. 31, 2017, pp. 765–774.

[15] A. D. Keromytis, V. Misra, D. Rubenstein, Sos: An architecture for mitigating ddos attacks, IEEE Journal on selected areas in communications 22 (1) (2004) 176–188.

[16] M. Walfish, M. Vutukuru, H. Balakrishnan, D. Karger, D. Karger, S. Shenker, Ddos defense by offense, in: ACM SIGCOMM Computer Communication Review, Vol. 36, ACM, 2006, pp. 303–314.

[17] J. Mirkovic, P. Reiher, A taxonomy of ddos attack and ddos defense mechanisms, ACM SIGCOMM Computer Communication Review 34 (2004) 39–53.

[18] A. Jakóbik, F. Palmieri, J. Kołodziej, Stackelberg games for modeling defense scenarios against cloud security threats, Journal of network and computer applications 110 (2018) 99–107.

[19] J. Hoffmann, Simulated penetration testing: From" dijkstra" to" turing test++", in: Twenty-Fifth International Conference on Automated Planning and Scheduling, 2015, pp. 364–372.

[20] Q. Wu, S. Shiva, S. Roy, C. Ellis, V. Datla, On modeling and simulation of game theory-based defense mechanisms against dos and ddos attacks, in: Proceedings of the 2010 spring simulation multiconference, Society for Computer Simulation International, 2010, pp. 159–167.

[21] G. Yan, R. Lee, A. Kent, D. Wolpert, Towards a bayesian network game framework for evaluating ddos attacks and defense, in: Proceedings of the 2012 ACM conference on Computer and communications security, ACM, 2012, pp. 553–566.

[22] S. Shiva, S. Roy, D. Dasgupta, Game theory for cyber security, in: Proceedings of the Sixth Annual Workshop on Cyber Security and Information Intelligence Research, Vol. 6, ACM, 2010, pp. 34–38.

[23] R. S. Sutton, A. G. Barto, Introduction to reinforcement learning, MIT press, 2018.

[24] J. Schwartz, H. Kurniawati, Autonomous penetration testing using reinforcement learning, arXiv preprint arXiv:1905.05965 (2019).

[25] V. Mnih, K. Kavukcuoglu, D. Silver, A. Graves, I. Antonoglou, D. Wierstra, M. Riedmiller, Playing atari with deep reinforcement learning, arXiv preprint arXiv:1312.5602 (2013).

[26] J. Mirkovic, S. Wei, A. Hussain, B. Wilson, R. Thomas, S. Schwab, S. Fahmy, R. Chertov, P. Reiher, Ddos benchmarks and experimenter's workbench for the deter testbed, in: 2007 3rd International Conference on Testbeds and Research Infrastructure for the Development of Networks and Communities, IEEE, 2007, pp. 1–7.

[27] L. Buşoniu, R. Babuška, B. De Schutter, Multi-agent reinforcement learning: An overview, in: Innovations in multi-agent systems and applications, Springer, 2010, pp. 183–221.

[28] L. Panait, S. Luke, Cooperative multi-agent learning: The state of the art, Autonomous agents and multi-agent systems 11 (2005) 387–434.

[29] K. Malialis, D. Kudenko, Large-scale ddos response using cooperative reinforcement learning, in: 11th European Workshop on Multi-Agent Systems (EUMAS), Vol. 11, European Workshop on Multi-Agent Systems, 2013, pp. 1–14.

[30] Cisco, Comparing traffic policing and traffic shaping for

bandwidth limiting, https://www.cisco.com/c/en/us/support/docs/quality-of-service-qos/qos-policing/19645-policevsshape.html, accessed: 2019-05-04 (May 2017).

[31] B. Meade, L. Lafayette, G. Sauter, D. Tosello, Spartan hpc-cloud hybrid— delivering performance and flexibility, https://doi.org/10.4225/49/58ead90dceaaa (Apr 2017). doi:10.4225/49/58ead90dceaaa.

[32] M. Riedmiller, Neural fitted q iteration–first experiences with a data efficient neural reinforcement learning method, in: European Conference on Machine Learning, Springer, 2005, pp. 317–328.

[33] J. Foerster, N. Nardelli, G. Farquhar, T. Afouras, P. H. Torr, P. Kohli, S. Whiteson, Stabilising experience replay for deep multi-agent reinforcement learning, in: Proceedings of the 34th International Conference on Machine Learning, Vol. 70, JMLR. org, 2017, pp. 1146–1155.

[34] G. Dulac-Arnold, R. Evans, H. van Hasselt, P. Sunehag, T. Lillicrap, J. Hunt, T. Mann, T. Weber, T. Degris, B. Coppin, Deep reinforcement learning in large discrete action spaces, https://arxiv.org/pdf/1512.07679 (June 2015).