ECAI 2008
M. Ghallab et al. (Eds.)
IOS Press, 2008

doi:10.3233/978-1-58603-891-5-873

Multi-Agent Reinforcement Learning for Intrusion Detection: A case study and evaluation

Arturo Servin and **Daniel Kudenko**[1]

Abstract. In this paper we propose a novel approach to train *Multi-Agent Reinforcement Learning* (MARL) agents to cooperate to detect intrusions in the form of normal and abnormal states in the network. We present an architecture of distributed sensor and decision agents that learn how to identify normal and abnormal states of the network using *Reinforcement Learning* (RL). Sensor agents extract network-state information using tile-coding as a function approximation technique and send communication signals in the form of actions to decision agents. By means of an on line process, sensor and decision agents learn the semantics of the communication actions. In this paper we detail the learning process and the operation of the agent architecture. We also present tests and results of our research work in an intrusion detection case study, using a realistic network simulation where sensor and decision agents learn to identify normal and abnormal states of the network.

1 Introduction

Intrusion Detection Systems (IDS) play an important role in the protection of computer networks and information systems from intruders and attacks. Despite previous research efforts there are still areas where IDS have not satisfied all requirements of modern computer systems. Specifically, *Denial of Service* (DoS) and *Distributed Denial of Service* (DDoS) attacks have received significant attention due to the increased security vulnerabilities in end-user software and bot-nets. A special case of DoS are the *Flooding-Base DoS* and *Flooding-Base DDoS* attacks. These are generally based on a flood of packets with the intention of overfilling the network resources of the victim. It is especially difficult to create a flexible hand-coded IDS for such attacks, and machine learning is a promising avenue to tackle the problem. Due to the distributed nature of this type of attacks and the complexities that involve its detection, we propose a distributed reinforcement learning (RL) approach.

In RL agents learn to act optimally via observations and feedback from the environment in the form of positive or negative rewards [7]. Multi-Agent RL has been successfully used to solve some challenging problems in various areas. Despite its apparent appeal, MARL needs to deal with problems such as the size of the action-state space which makes scalability an issue; the partial information that agents have of other agents' observations and actions; a non-stationary environment as result of the actions of other agents, and the credit assignment problem.

To overcome these problems we present an architecture of distributed sensor agents that get information from the environment and share it in the form of communication signals with other agents higher up the hierarchy. Without any previous semantic knowledge about the signals, higher-level hierarchical agents interpret them and consequently interact with the environment. This results in a learning process where agents with partial observability make decisions and coordinate their own actions to reach a common goal. In order to evaluate our proposal we explore its use of *Distributed Intrusion Detection Systems* (DIDS).

[1] University of York, United Kingdom, email: aservin,kudenko@cs.york.ac.uk

2 Agent Architecture

We propose an architecture of autonomus agents divided *sensor agents* (SA) and *decision agents* (DA). SA collect and analyse state information about the environment. Each SA receives only partial information about the global state of the environment and they map this local state to communications action-signals. These signals are received by the DA and without any previous knowledge it learns their semantics and how to interpret their meaning. In this way, the DA tries to model the local state of cell environment. Then it decides which final action to trigger (in our case study it triggers an alarm to the network operator). When the DA triggers the action and this is appropriate accordingly with the goal pursued, all the agents receive a positive reward. If the action is not correct, all the agents receive a negative reward. The goal is to coordinate the signals sent by the SA to the DA in order to represent the global state of the environment.

To detect the abnormal states that DoS and DDoS generate in a computer network we have designed an architecture composed by four agents. These agents are a Congestion Sensor Agent (CSA), a Delay Sensor Agent (DSA), a Flow Sensor Agent (FSA) and the Decision Agent (DA). We need this diversity of sensor information to develop more reliable IDS. The idea is that each sensor agent perceives different information depending on their capabilities, their operative task and where they are deployed in the network. Furthermore not all the features are available in a single point in the network. Flow and congestion information may be measured in a border router between the Internet and the Intranet whilst delay information may be only available from an internal router.

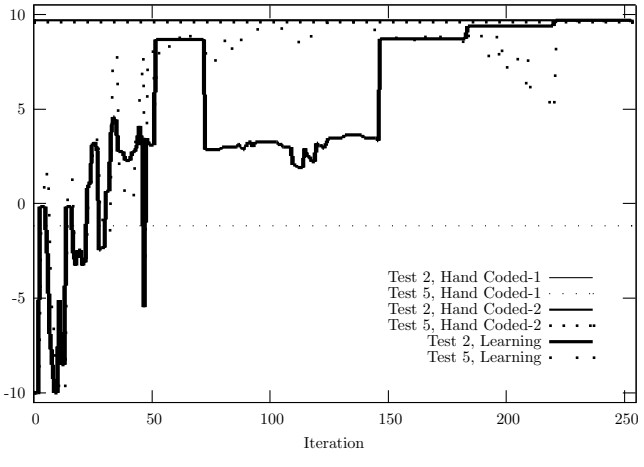
3 Results

We set up several tests to verify the learning capabilities of our agent architecture. We used a control test to train the agents to categorise basic normal and abnormal activity in the network. To simulate the normal traffic we randomly started and stopped connections from node 0 (TCP/FTP) and node 1 (UDP stream). Using another random pattern of connections we used node 4 to simulate the attacks to the network characterised by a flood of UDP traffic. To evaluate the adaptability of the agents we ran tests changing the normal and abnormal traffic patterns. We also ran tests designed to create more

complex scenarios where the attacker changes its attack to mimic authorised or normal traffic.

We compared our learning algorithm against two hard-coded approaches. The first hard-coded approach (Hard-Coded 1) emulated a misuse IDS. In this case the IDS is looking for the patterns that match an attack. The Hard-Coded 2 approach integrates the same variety of input information as our learning algorithm. We evaluated the learning and hard-coded approaches using test 2 and test 5. Test 2 only changes the traffic pattern of the attack and it must be very simple to detect. Attacks in test 5 we changed the packet size and the attack UDP port to be the same used by normal applications. This test is the hardest to detect because it emulates some of the signatures of normal traffic. The learning curves of the test are shown in Fig.1. Hard-Coded 1 had no problem to identify attacks and have low false negatives for test 2 but it completely failed to detect attacks test 5. This is the same problem that misuse IDS have when the signature of the attack changes or when they face unknown attacks. The results for Hard-Coded 2 and our learning approach confirm our argument that for more reliable intrusion detection we need a variety of information sources. Both solutions were capable of detecting the attacks even though one of the sensors was reporting incorrect information. This scenario also could be seen as the emulation of a broken sensor sending bogus information or a sensor compromised by the attacker and forced to send misleading signals. Either way it demonstrates that a system using more than one source to detect intrusions could be more reliable than single-source IDS.

Figure 1. Learning Curves



Both the Hand-coded 2 and learning approaches present very good results regarding the identification of normal and abnormal states in the network. While the learning algorithm requires some time to learn to recognise normal and abnormal activity, it does not require any previous knowledge about the behaviour of the measured variables. Hand-coded 2 reaches maximum performance since the beginning of the simulation but it requires in-deepth knowledge from the policy programmer about the the network traffic and the variables measured to detect intrusions.

4 Related Work

Problems such as the curse of dimensionality; partial observability and scalability in MARL have been analysed using a variety of methods and techniques and they represent the foundation of our research. An application of MARL to networking environments is presented in [2] where cooperative agents learn how to route packets using optimal paths. Using the same approach of flow control and feedback from the environment, other researchers have expanded the use of RL in routing algorithms [6], explore the use of MARL to control congestion in networks [4], routing using QoS [5] and more recently to control DDoS attacks [9].

The use of RL in the intrusion detection field has not been widely studied and even less in distributed intrusion detection. Some research works are [3] where the authors trained a neural network using RL and [1] where game theory is used to train agents to recognise DoS attacks against routing infrastructure. Other recent research work include the use of RL to detect host intrusion using sequence system calls [10] and the previously mentioned [9].

5 Conclusion and Future Work

We have shown how a group of agents can coordinate their actions to reach the common goal of network intrusion detection. During this process decision agents learn how to interpret the action-signals sent by sensor agents without any previously assigned semantics. These action-signals aggregate the partial information received by sensor agents and they are used by the decision agents to reconstruct the global state of the environment. In our case study, we evaluate our learning approach by identifying normal and abnormal states of a realistic network subjected to various DoS attacks. We have also successfully applied RL in a group of network agents under conditions of partial observability, restricted communication and global rewards in a realistic network simulation. Finally we can conclude that using a variety of network data has generated good results to identify the state of the network. In some cases the agents can generate good results even when some of this information is missing.

Future work include scaling up our learning approach to a large number of agents a hierarchical approach. This architecture will allow us to create more complex network topologies and eventually the emulation of real packet streams inside the network environment.

REFERENCES

[1] B. Awerbuch, D. Holmer, and H. Rubens, 'Provably Secure Competitive Routing against Proactive Byzantine Adversaries via Reinforcement Learning', *John Hopkins University, Tech. Rep., May*, (2003).

[2] J.A. Boyan and M.L. Littman, 'Packet routing in dynamically changing networks: A reinforcement learning approach', *Advances in Neural Information Processing Systems*, **6**(1994), 671–678, (1994).

[3] J. Cannady, 'Next Generation Intrusion Detection: Autonomous Reinforcement Learning of Network Attacks', *NISSC00: Proc. 23rd National Information Systems Security Conference*, (2000).

[4] J. Dowling, E. Curran, R. Cunningham, and V. Cahill, 'Using feedback in collaborative reinforcement learning to adaptively optimize MANET routing', *Systems, Man and Cybernetics, Part A, IEEE Transactions on*, **35**(3), 360–372, (2005).

[5] E.G. Gelenbe, M. Lent, and R.P.L.P. Su, 'Autonomous smart routing for network QoS', *Autonomic Computing, 2004. Proceedings. International Conference on*, 232–239, (2004).

[6] A. Nowe, K. Steenhaut, M. Fakir, and K. Verbeeck, 'Q-learning for adaptive load based routing', *Systems, Man, and Cybernetics, 1998. 1998 IEEE International Conference on*, **4**, (1998).

[7] R.S. Sutton and A.G. Barto, *Reinforcement Learning: An Introduction*, MIT Press, 1998.

[8] X. Xu, Y. Sun, and Z. Huang, 'Defending DDoS Attacks Using Hidden Markov Models and Cooperative Reinforcement Learning', *LECTURE NOTES IN COMPUTER SCIENCE*, **4430**, 196, (2007).

[9] X. Xu and T. Xie, 'A Reinforcement Learning Approach for Host-Based Intrusion Detection Using Sequences of System Calls', *Proceedings of the International Conference on Intelligent Computing*, (2005).