MACHINE LEARNING FOR PENTEST

GVHD: TS NGUYỄN TẤN CẦM

Introduction

Theoretical and related researches

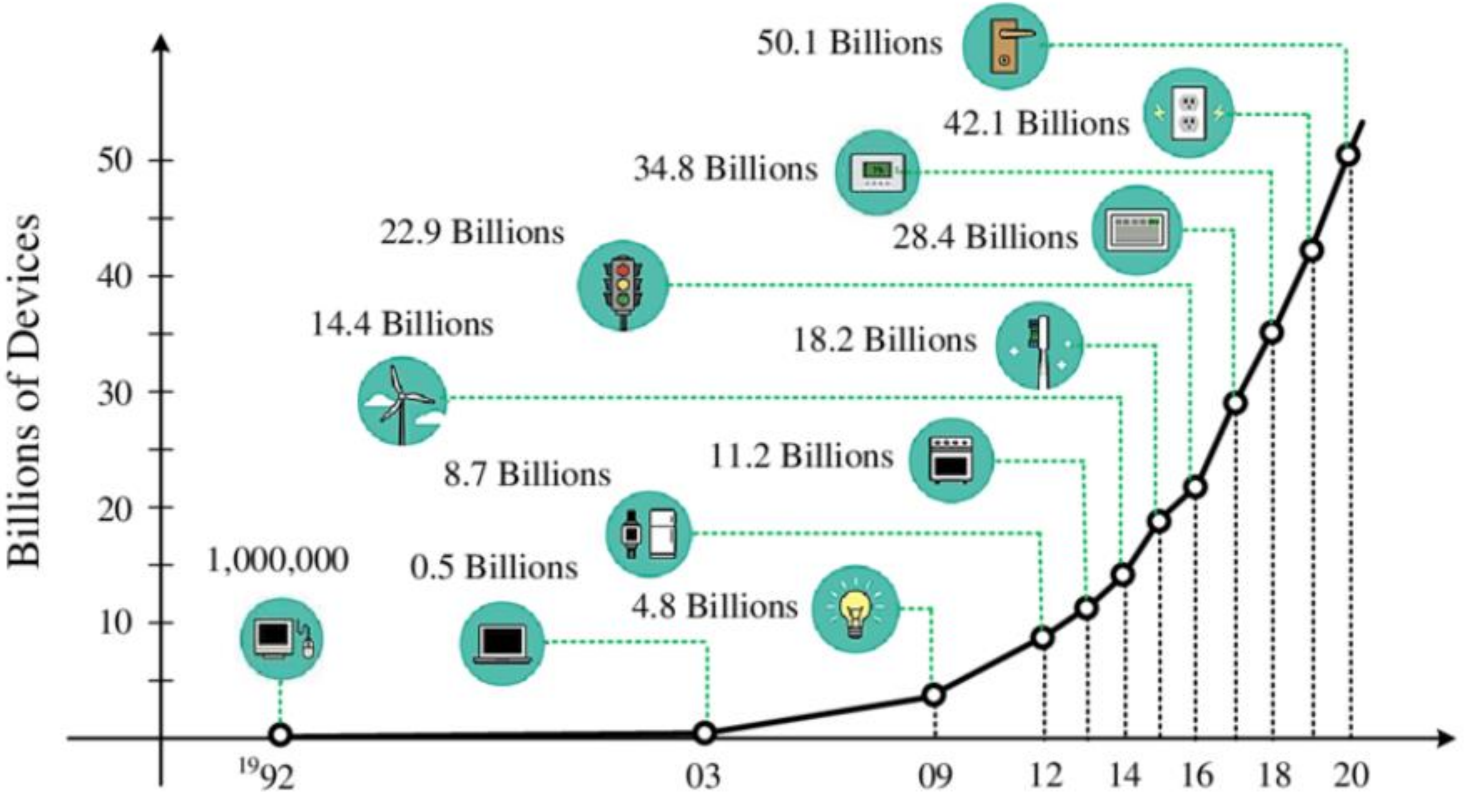
Proposed method and model

Conclusions

- Number of devices are growth up day by day.

- In the period 2010 – 2022, amount of losses of cyberattacks is ~ 24$ Billion

- Security by perform as a hacker is one of first and effective method.



1.2. Why need machine learning in pentest?

- Have many tools, services can use when pentest

- People need have experience to use tools, but too much software, system need to pentest

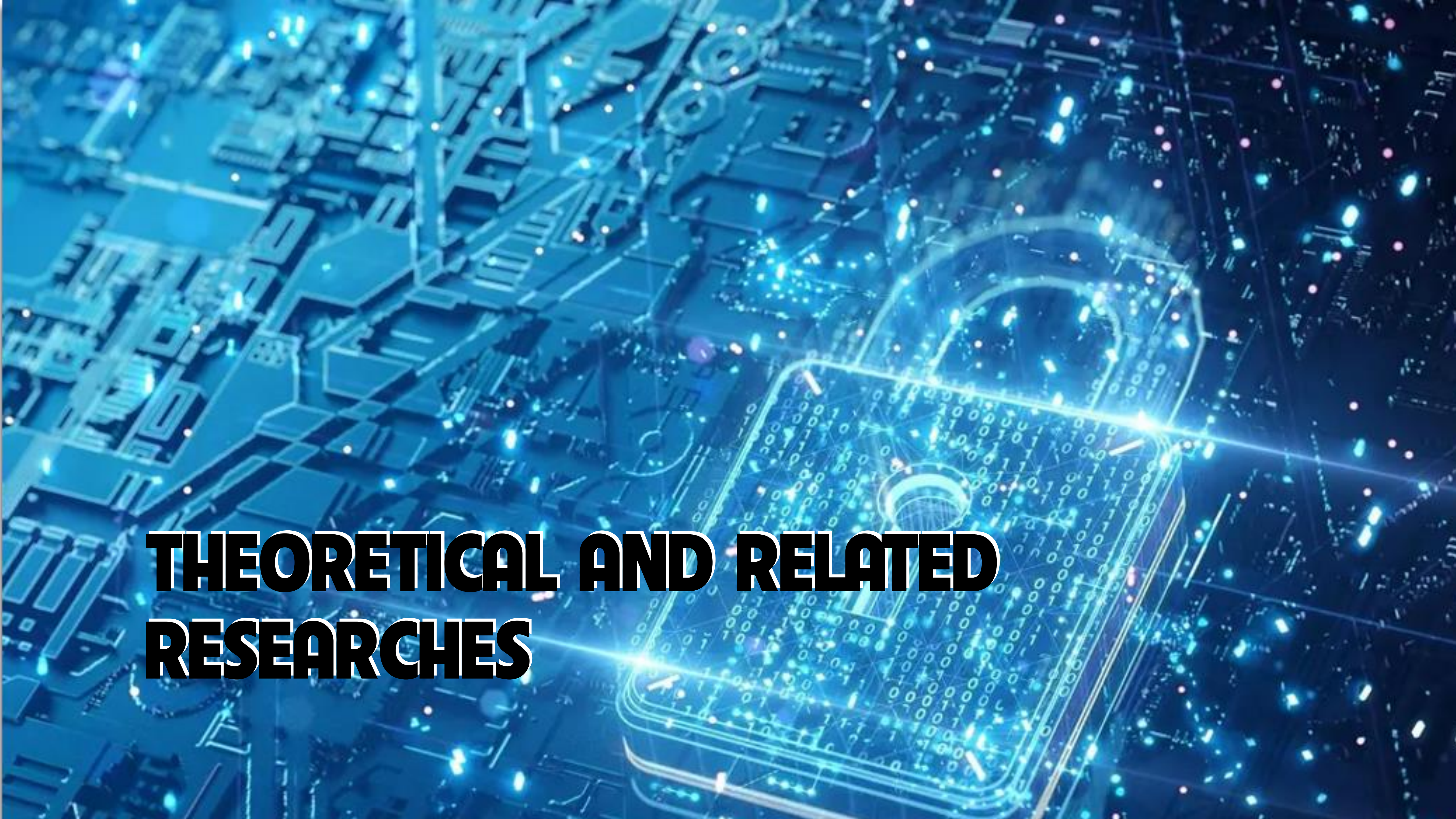
- People can happen objective error

- Need a framework can be automatic do something according to previous payload

- An application can learning and use this for choose correct payload

- A public paper for this research

THEORETICAL AND RELATED RESEARCHES



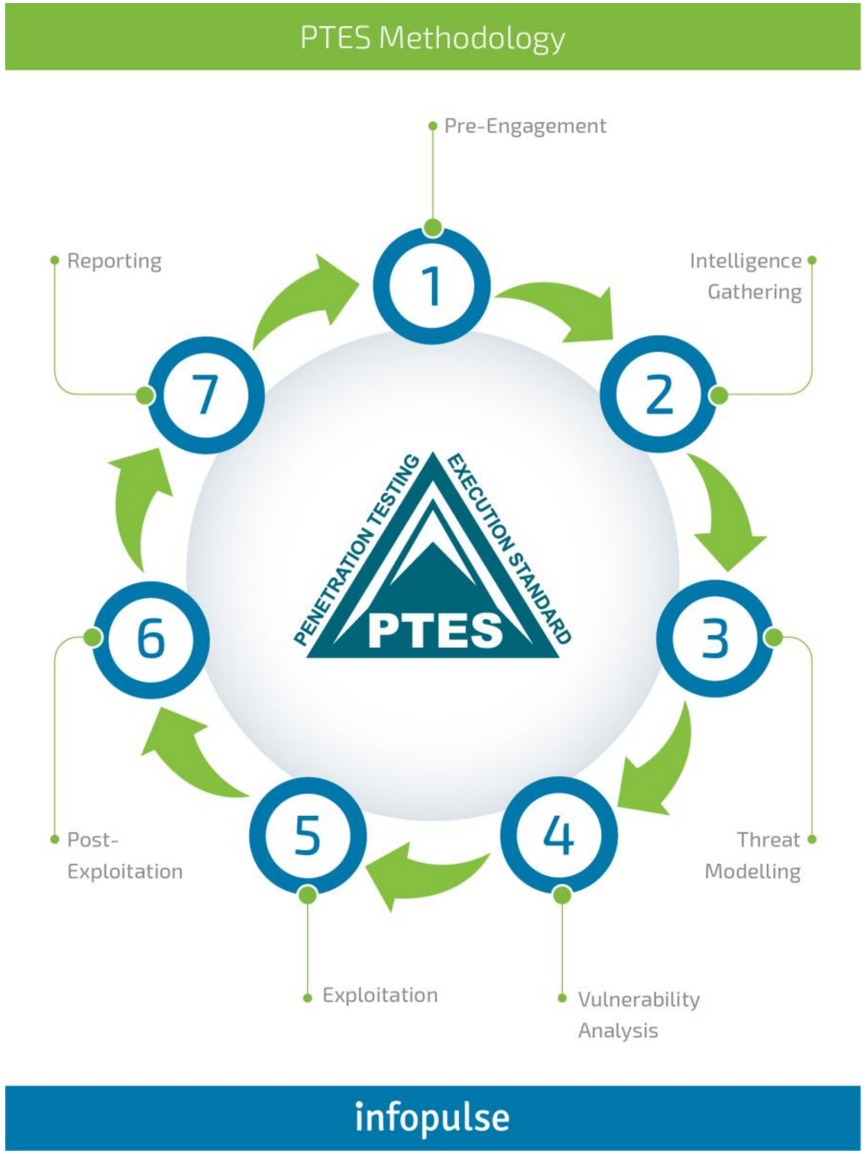
- Penetration testing first became a concept in the 1960s

- First team named "Tiger Teams", whose work for US government and military.

- Work as a hacker, find and report vulnerability

Penetration Testing Execution Standard

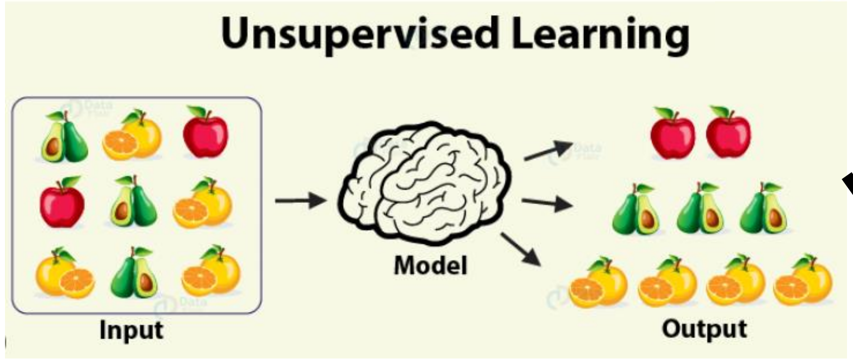
- Pre-engagement Interactions
- Intelligence Gathering
- Threat Modeling
- Vulnerability Analysis
- Exploitation
- Post Exploitation
- Reporting

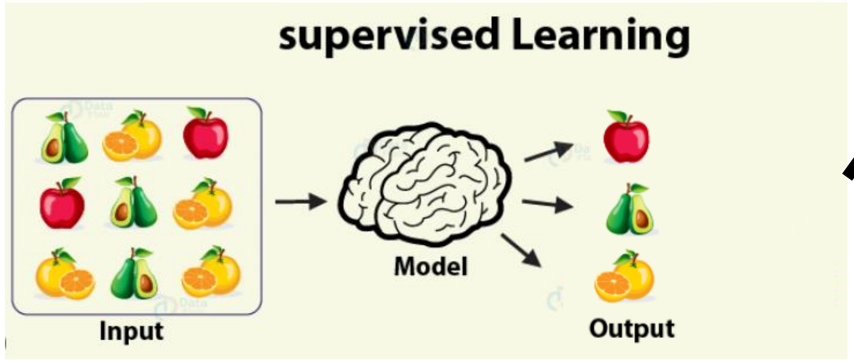


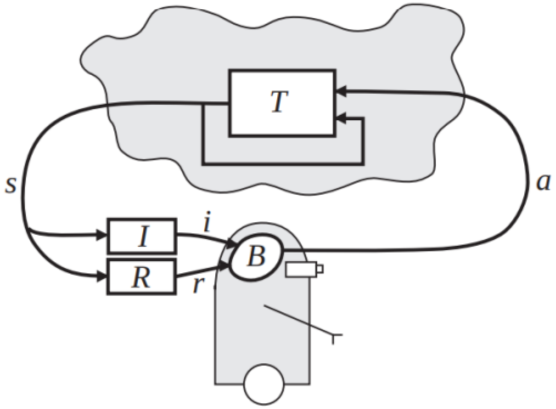

Contents [hide]

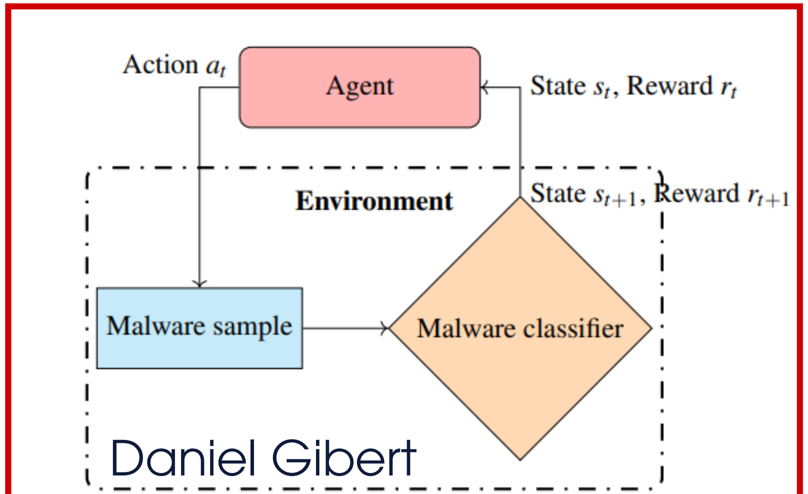
- Use supervised model was trained such as deep neural network is base.

- Use unsupervised model to get experience when pentest

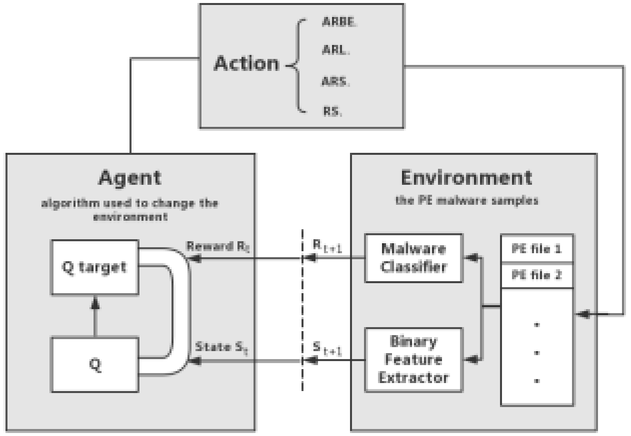




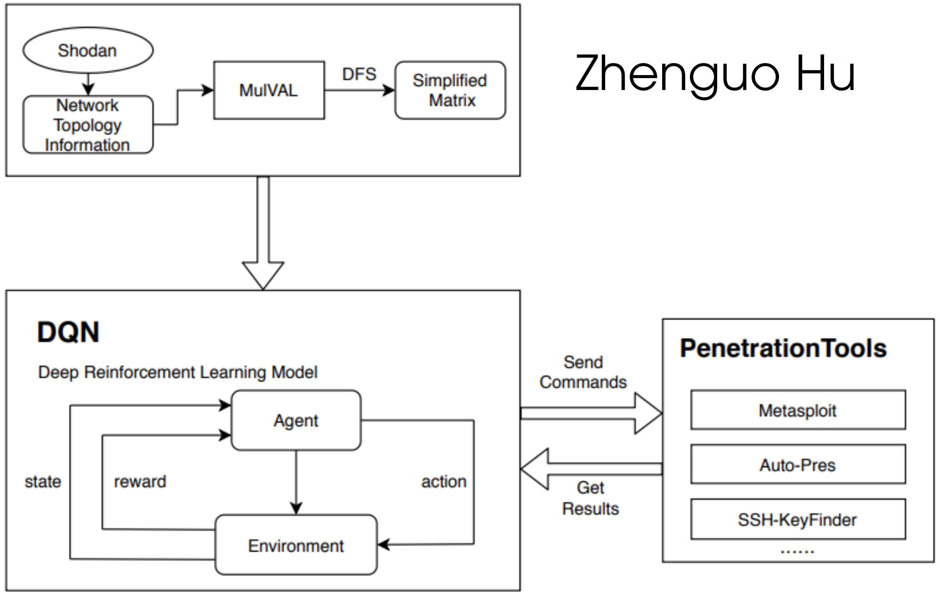




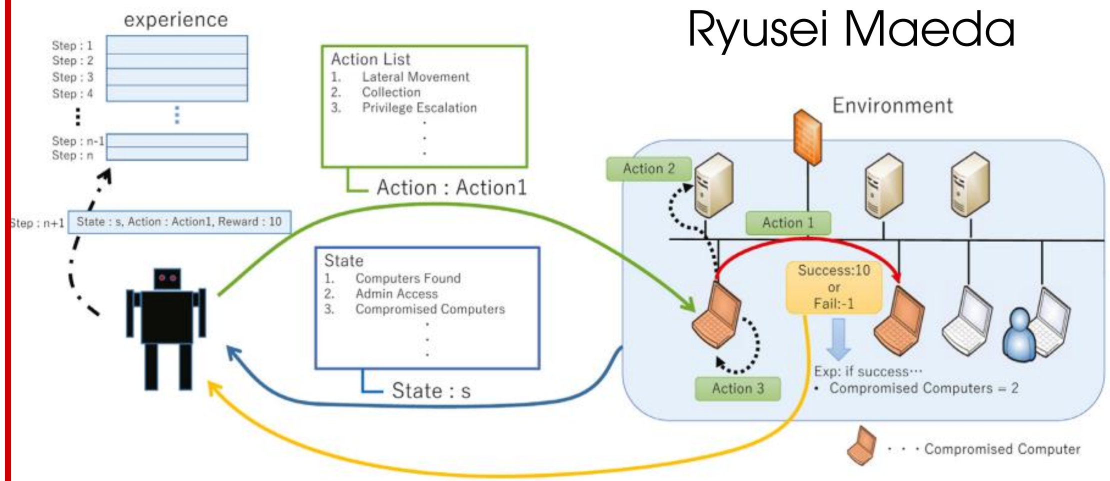
Daniel Gibert



Zhiyang fang

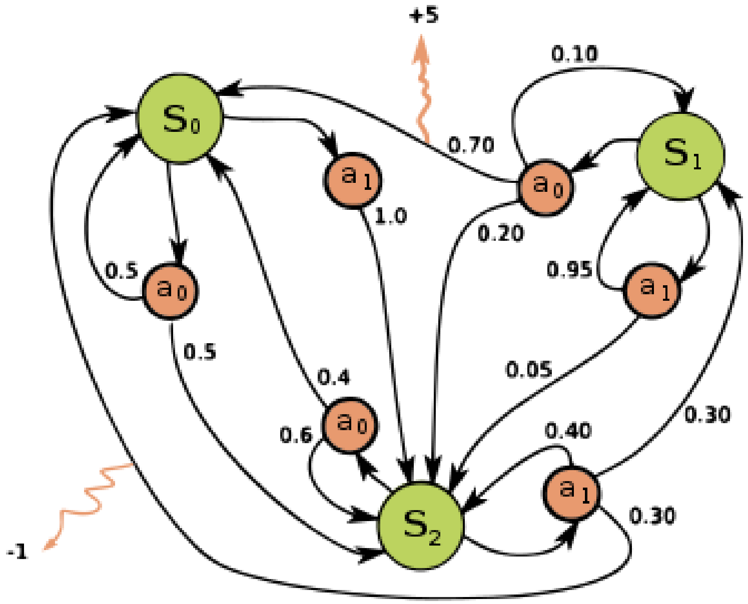


Zhenguo Hu



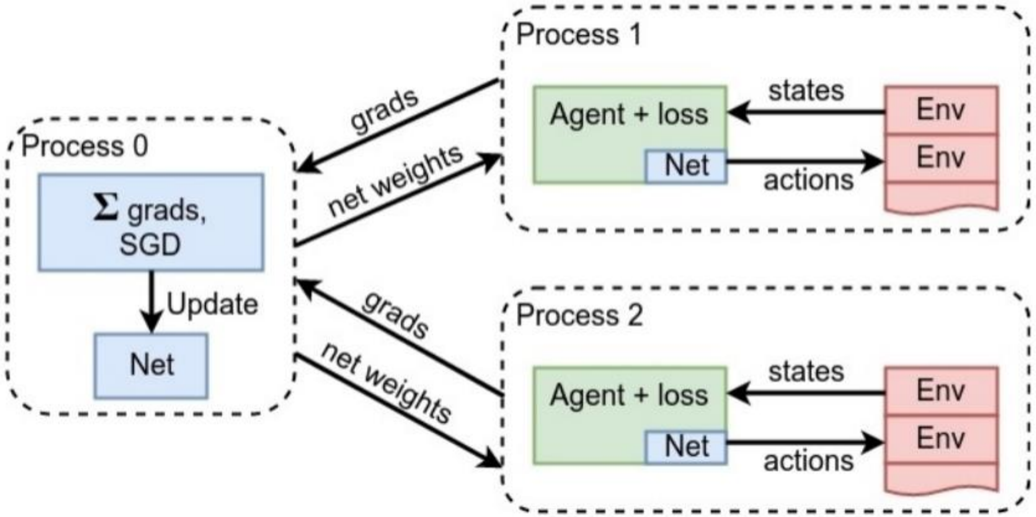
Ryusei Maeda

Proposed method and model

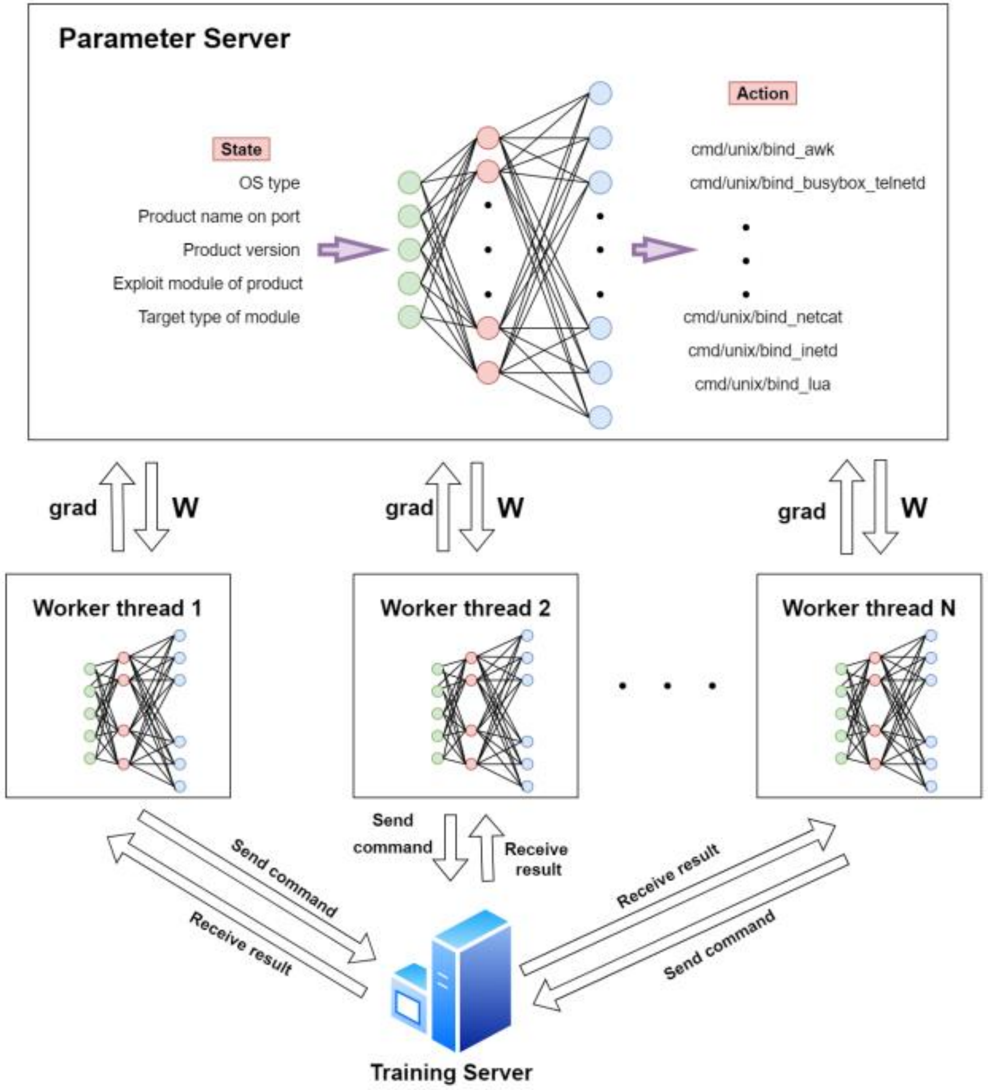


- Use to modeling decision-making problem
- Have some part: environment state, active space and reward function.
- The next state only depend on current state and decision, not depend on the past.

- **Advantage**: use advantage function
- **Actor Critic:** The policy is updated using the value function, and the value function is updated using the policy
- **Asynchronous:** allows multiple agents to learn simultaneously



- A training server control worker – which training the agent
- Agent will get environment state are 5 param, choose action in list (payload) and exploit target.
- Multi worker thread help decrease training time



Parameter	Value
Gamma	0.99
Epsilon greedy start	0.5
Epsilon greedy stop	0
RMSprop learning rate	0.005
RMSProp decay	0.99
Loss coefficient	0.5
Loss entropy coefficient	0.01
Number of test worker	1
Greedy rate	0.8

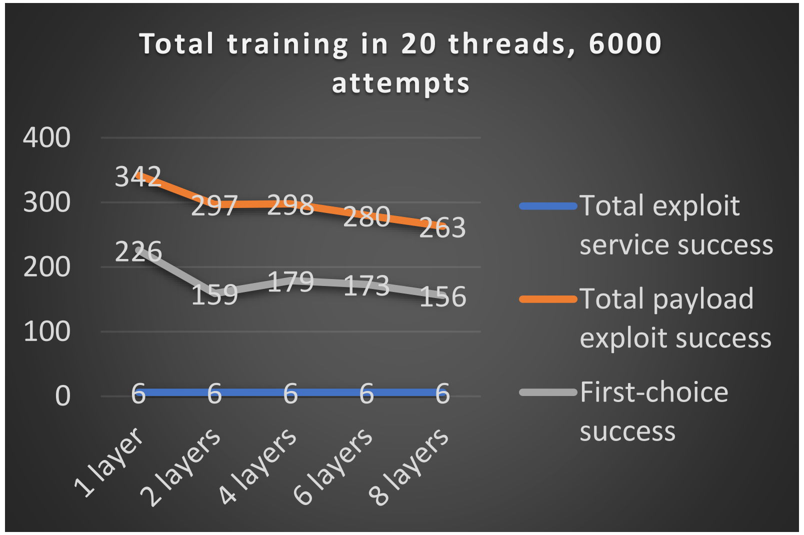
Name of VMs	Operating System	RAM	CPU
Penetration Testing	Kali Linux	12 GB	4 cores
Testing Server	Ubuntu 18.04	12 GB	4 cores

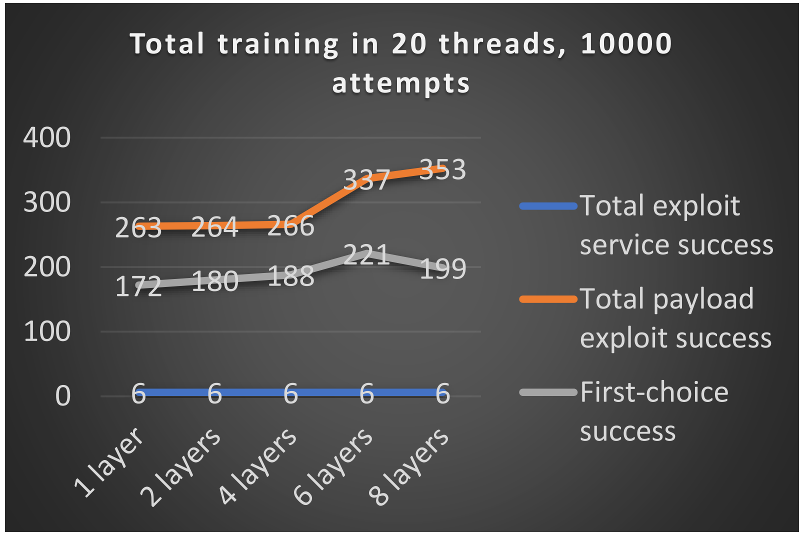
Service	Port	Operating System	CVE
Samba	445	Docker Ubuntu	CVE-2017-7494
WebLogic	7001	Docker Ubuntu	CVE-2017-10271
PostgreSQL	5432	Docker Ubuntu	CVE-2019-9193
Supervisor	9001	Docker Ubuntu	CVE-2017-11610

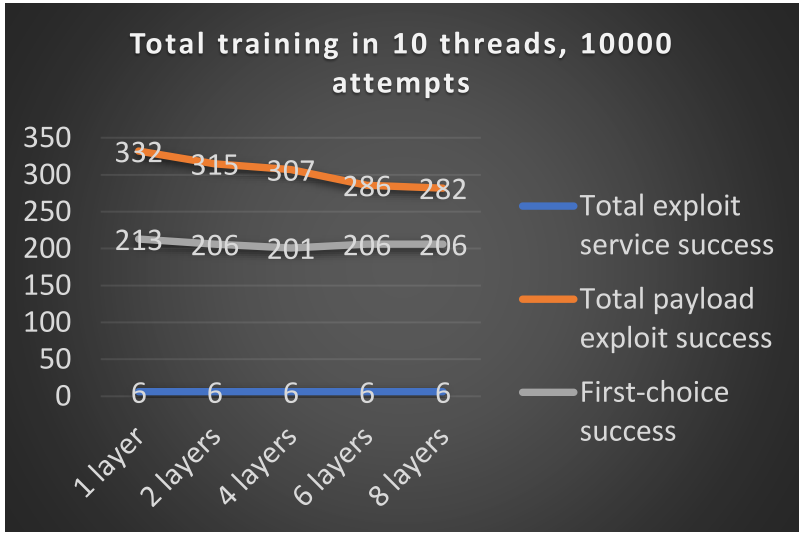
Case	Number of threads	Number of attempts
1	20 threads	6000 attempts
2	20 threads	10000 attempts
3	10 threads	10000 attempts

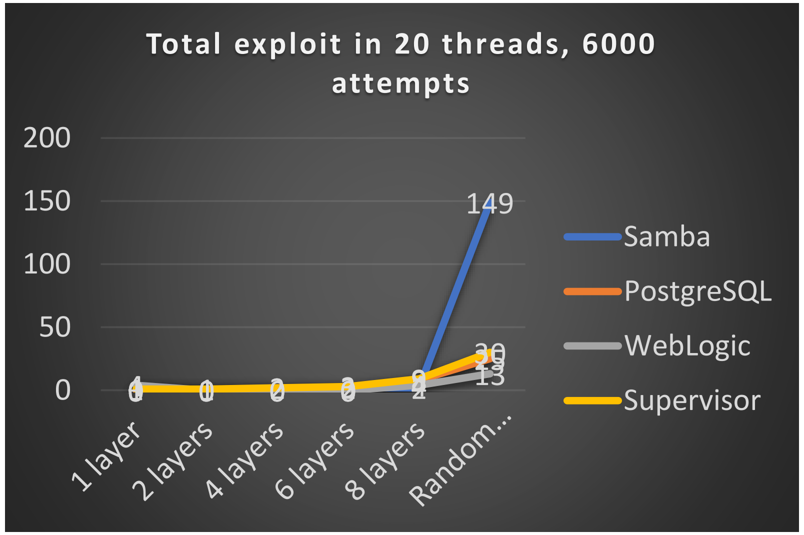
Number of hidden layer	The number of nodes in the hidden layer
1 hidden layer	300
2 hidden layers	100 – 300
4 hidden layers	50 – 100 – 200 – 400
6 hidden layers	50 – 100 – 200 – 300 – 400 – 500
8 hidden layers	50 – 100 – 150 – 200 – 250 – 300 – 350 – 400

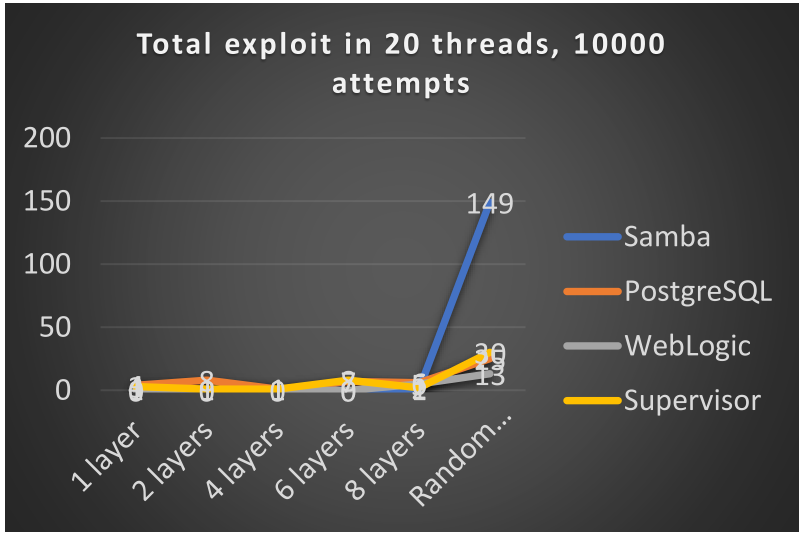
3.5. Results

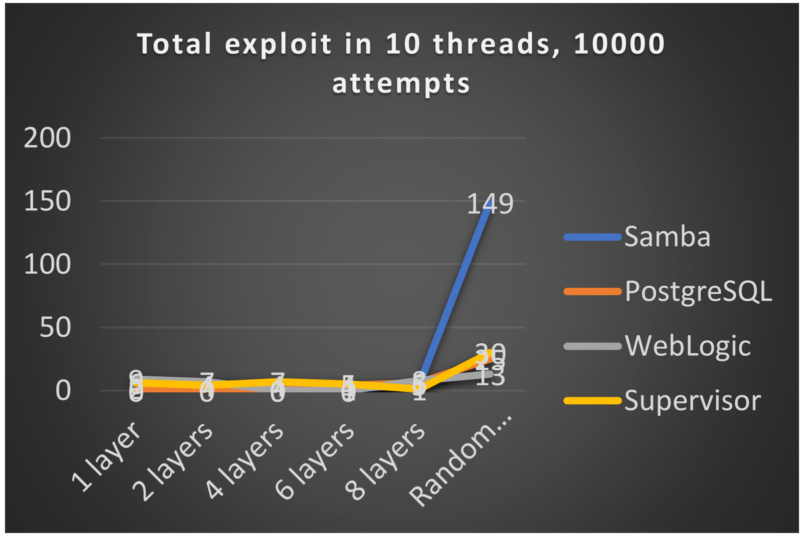

Total training in 20 threads, 6000 attempts
400
300
200
100
0
342
297
298
280
263
226
159
179
173
156
6
6
6
6
6
1 layer
2 layers
4 layers
6 layers
8 layers
Total exploit service success
Total payload exploit success
First-choice success


Total training in 20 threads, 10000 attempts
400
300
200
100
0
263
264
266
337
353
172
180
188
221
199
6
6
6
6
6
1 layer
2 layers
4 layers
6 layers
8 layers
Total exploit service success
Total payload exploit success
First-choice success


Total training in 10 threads, 10000 attempts
350
300
250
200
150
100
50
0
332
315
307
286
282
213
206
201
206
206
6
6
6
6
6
1 layer
2 layers
4 layers
6 layers
8 layers
Total exploit service success
Total payload exploit success
First-choice success


Total exploit in 20 threads, 6000 attempts
200
150
100
50
0
149
13
1 layer
2 layers
4 layers
6 layers
8 layers
Random...
Samba
PostgreSQL
WebLogic
Supervisor


Total exploit in 20 threads, 10000 attempts
200
150
100
50
0
149
13
1 layer
2 layers
4 layers
6 layers
8 layers
Random...
Samba
PostgreSQL
WebLogic
Supervisor


Total exploit in 10 threads, 10000 attempts
200
150
100
50
0
149
13
1 layer
2 layers
4 layers
6 layers
8 layers
Random...
Samba
PostgreSQL
WebLogic
Supervisor

Conclusions

- Kết quả đạt được

- Hướng phát triển

[1] Leslie Pack Kaelbling, Michael L Littman, and Andrew W Moore. "Reinforcement learning: A survey". In: *Journal of artificial intelligence research* 4 (1996), pp. 237–285.

[2] Esther Levin, Roberto Pieraccini, and Wieland Eckert. "Using Markov decision process for learning dialogue strategies". In: *Proceedings of the 1998 IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP'98 (Cat. No. 98CH36181)*. Vol. 1. IEEE. 1998, pp. 201–204.

[3] B Arkin, S Stender, and G McGraw. *Software penetration testing. IEEE Secur. Priv. 3 (1), 84–87 (2005)*. 2005.

[4] Ben H Thacker et al. "Probabilistic engineering analysis using the NESSUS software". In: *Structural safety* 28.1-2 (2006), pp. 83–107.

[5] Jeff Heaton. *Introduction to neural networks with Java*. Heaton Research, Inc., 2008.

[6] Gordon Fyodor Lyon. *Nmap network scanning: The official Nmap project guide to network discovery and security scanning*. Insecure, 2009.

[7] Lee A Bygrave. "Privacy and data protection in an international perspective". In: *Scandinavian studies in law* 56.8 (2010), pp. 165–200.

[8] Patrick Engebretson. *The basics of hacking and penetration testing: ethical hacking and penetration testing made easy*. Elsevier, 2013.

[9] Filip Holik et al. "Effective penetration testing with Metasploit framework and methodologies". In: *2014 IEEE 15th International Symposium on Computational Intelligence and Informatics (CINTI)*. IEEE. 2014, pp. 237–242.

[10] Abdelmohsen Ali, Walaa Hamouda, and Murat Uysal. "Next generation M2M cellular networks: Challenges and practical considerations". In: *IEEE Communications Magazine* 53.9 (2015), pp. 18–24.

[11] H Jabbar and Rafiqul Zaman Khan. "Methods to avoid over-fitting and under-fitting in supervised machine learning (comparative study)". In: *Computer Science, Communication and Instrumentation Devices* 70.10.3850 (2015), pp. 978–981.

[12] Thomas J Holt, Olga Smirnova, and Yi Ting Chua. "Exploring and estimating the revenues and profits of participants in stolen data markets". In: *Deviant Behavior* 37.4 (2016), pp. 353–367.

[13] Volodymyr Mnih et al. "Asynchronous methods for deep reinforcement learning". In: *International conference on machine learning*. PMLR. 2016, pp. 1928–1937.

[14] Maxim Lapan. *Deep Reinforcement Learning Hands-On: Apply modern RL methods, with deep Q-networks, value iteration, policy gradients, TRPO, AlphaGo Zero and more*. Packt Publishing Ltd, 2018.

[15] Mehdi Yousefi et al. "A reinforcement learning approach for attack graph analysis". In: *2018 17th IEEE International Conference On Trust, Security And Privacy In Computing And Communications/12th IEEE International Conference On Big Data Science And Engineering (TrustCom/BigDataSE)*. IEEE. 2018, pp. 212–217.

[16] M Ugur Aksu, Enes Altuncu, and Kemal Bicakci. "A first look at the usability of openvas vulnerability scanner". In: *Workshop on usable security (USEC)*. 2019.

Thank for watching

Ask anything and answer.

Leveraging Deep Reinforcement Learning for Automating Penetration Testing in Reconnaissance and Exploitation Phase

Publisher: IEEE

Le Van Hoang ; Nguyen Xuan Nhu ; To Trong Nghia ; Nguyen Huu Quyen ; Van-Hau Pham ; Phan The Duy **All Authors**

Abstract:

Penetration testing is one of the most common methods for assessing the security of a system, application, or network. Although there are different support tools with great efficiency in this field, penetration testing is done mostly manually and relies heavily on the experience of the ethical hackers who are doing it, known as pentesters. This paper presents an automated penetration testing approach that leverages deep reinforcement learning (RL) to automate the penetration testing process, including the reconnaissance and exploitation phases. More specifically, the RL agent is trained with the A3C model to gain experience choosing an exact payload to exploit available vulnerabilities. Ad-ditionally, our RL-based pentesting tool has three main functions: information gathering, vulnerability exploitation, and reporting. The performance of this approach is benchmarked against real-world vulnerabilities in our experimental environments. After training with environmental settings, the RL agent can assist pentesters in quickly identifying vulnerabilities in their own servers. The RL-based approach can mitigate the problems of labor costs and hunger data for automating penetration testing in the system by learning how to execute exploits on its own. The more pentesters who use this tool, the more accurate the pentesting results will be. With outstanding results, this method proves that it can accumulate learning results from previous environments to successfully exploit vulnerabilities for the next exploit in another environment on the first try.