



City Research Online

City St George's, University of London

Citation: Ghanem, M. & Chen, T. (2020). Reinforcement learning for efficient network penetration testing. *Information*, 11(1), 6. doi: 10.3390/info11010006

This is the published version of the paper.

This version of the publication may differ from the final published version. To cite this item please consult the publisher's version.

Permanent repository link: <https://openaccess.city.ac.uk/id/eprint/23860/>

Link to published version: <https://doi.org/10.3390/info11010006>

Copyright and Reuse: Copyright and Moral Rights remain with the author(s) and/or copyright holders. Copies of full items can be used for personal research or study, educational, or not-for-profit purposes without prior permission or charge, unless otherwise indicated, provided that the authors, title and full bibliographic details are credited, a hyperlink and/or URL is given for the original metadata page and the content is not changed in any way. For full details of reuse please refer to [City Research Online policy](#).

Article

Reinforcement Learning for Efficient Network Penetration Testing

Mohamed C. GHANEM and Thomas M. CHEN

¹ School of Mathematics Computer Science and Engineering; City, University of London; mohamed.ghanem@city.ac.uk

² School of Mathematics Computer Science and Engineering; City, University of London; tom.chen.1@city.ac.uk

Version November 21, 2019 submitted to Information

Abstract: Penetration testing (also known as pentesting or PT) is common practice for actively assessing the defences of a computer network by planning and executing all possible attacks to discover and exploit existing vulnerabilities. Current penetration testing methods are increasingly becoming non-standard, composite and resource consuming despite the use of evolving tools. In this paper, we propose and evaluate an AI-based pentesting system which makes use of machine learning techniques, namely reinforcement learning (RL) to learn and reproduce average and complex pentesting activities. The proposed system is named Intelligent Automated Penetration Testing System (IAPTS) and will be a module that integrates with industrial PT systems and frameworks to enable them to capture information, learn from experience and reproduce the test in future nearly similar testing cases. IAPTS aims to save human resources while producing much enhanced results in term of time consumption, reliability and frequency of testing. IAPTS takes the approach of modelling PT environments and tasks as a partially observed Markov decision process (POMDP) problem which is solved by POMDP-solver. Although this paper scope is limited to network infrastructures PT planning and not the entire practice, the obtained results support the hypothesis that RL can enhance PT beyond the capabilities of any human PT expert in terms of time consumed, covered attacking vectors, accuracy and reliability of the outputs. In addition, this work tackled the complex problem of expertise capturing and re-use by allowing the IAPTS learning module to store and re-use PT policies in the same way that a human PT expert would learn but in a more efficient way.

Keywords: penetration testing, artificial intelligence, machine learning, reinforcement learning, network security auditing, offensive cyber-security, vulnerability assessment.

1. Introduction

Computer networks are more than ever exposed to cyber threats of increasing frequency, complexity and sophistication [1]. Penetration Testing (shortly known as pentesting or PT) is a well-established proactive method to evaluate the security of digital assets, varying from a single computer to websites and networks, by actively searching for and exploiting the existing vulnerabilities. The practice is an emulation of the operational mode that hackers follow in real-world cyber attacks. In the current constantly evolving digital environment, PT is becoming a crucial and often mandatory component of cyber security auditing particularly after the introduction of the European GDPR (General Data Protection Regulation) for organizations and businesses. In addition to legal requirements, PT is considered by the cyber security community as the most effective method to assess the strength of security defences against skilled adversaries as well as the adherence to security policies [2]. In practical terms, PT is a multi-stage process that often requires a high degree of competence and expertise due to the complexity of digital assets such as medium and large networks. Naturally,

34 research has investigated the possibility of automating tools for the different PT stages (reconnaissance,
35 identification, and exploitation) to relieve the human expert from the burden of repetitive tasks.
36 However, automation by itself does not achieve much benefits in terms of time, resources and outputs
37 because PT is a dynamic and interactive process of exploring and decision making, which requires
38 advanced and critical cognitive skills that are hard to duplicate through automation.

39 A natural question arises in regard to the capability of AI to provide a potential solution that
40 goes beyond simple automation to achieve expert-like output. In other research fields, AI proved very
41 helpful to not only offload work from humans but also possibly handle depths and details that humans
42 can not tackle fast enough or accurately enough. Rapid progress in the AI and notably machine
43 learning (ML) sub-field led us to believe that an AI-based PT system utilizing well-grounded models
44 and algorithms for making sequential decisions in uncertain environments can bridge the gap between
45 automation and expertise that PT community experience. In this perspective, the existing PT systems
46 and framework started shifting from executing experts' tasks to become more autonomous, intelligent
47 and optimized aiming that all existing threats are checked systematically and efficiently without or
48 with little human expert intervention. Furthermore, these systems should optimise the use of resources
49 by eliminating time-consuming and irrelevant directions and ensure that no threat is overlooked.

50 In addition to the regular use of PT, the testing results (output) should be processed and stored to
51 serve for further use. In fact, the main difference between human PT expert and automated systems is
52 that humans learn alongside performing the tests and enrich their expertise throughout, while systems
53 omit the re-usability of the data which is sometimes crucial especially when the testing is repeated
54 such as for regular compliance tests. In practical terms, the vast majority of the assessed network
55 configurations will not change considerably over a short period and therefore the output of previous
56 tests could remain entirely or partly applicable for an eventual re-testing required after one or more of
57 these following points occur:

- 58 • Network hardware, software, segments or applications were added, changed or removed
- 59 • Significant systems' upgrades or modifications are applied to infrastructure or applications
- 60 • Infrastructure changes including moving locations
- 61 • Security solutions or patches were installed or modified
- 62 • Security or users' policies were modified

63 Automation was and remains the best solution to save time and resources in any domain and PT
64 is not an exception to this rule. Therefore, the offensive cyber security community accorded during the
65 last decade a particular attention to the automation of the used systems. Such improvement permitted
66 to save significant time, efforts and resources in performing the task. Given the particularity of PT
67 practice, the increasing size and complexity of the tested assets along with the significant number of
68 vulnerabilities, exploit and attacks' vectors which should cover by the tester, the blind automated
69 system becomes powerless and often perform worse than manual practice pushing the researcher to
70 focus on improving such systems by adopting a variety of solutions. This paper explores in deep the
71 design and development of an ML-based PT system that allows intelligent, optimized and efficient
72 testing by perceiving its environment and decide autonomously how to act in order to perform PT
73 tasks as better as human experts and save time and resources along with improving accuracy and
74 testing coverage and frequency.

75 *1.1. Research Context*

76 Performing a periodic offensive security testing and auditing is an essential process to ensure
77 the resilience and the compliance of the assessed asset notably the confidentiality, availability and
78 integrity. PT is reputed to be the best approach to assess the security of digital assets by identifying
79 and exploiting its vulnerabilities. Currently, dozens of commercial and freeware systems, platforms
80 and frameworks are being used by PT experts with some offering some automation features which
81 nevertheless remain either local (specific to very limited context or tasks) or not optimized (blind

82 automation) and therefore creating significant accuracy and performance issues notably in case testing
 83 medium and large networks.

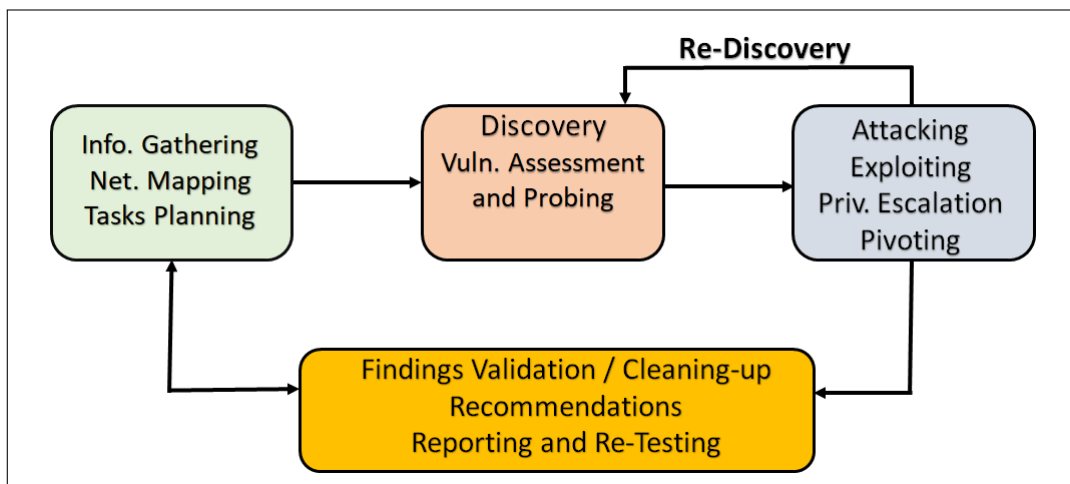


Figure 1. PT is a non-standard active method for assessing network defence by following a sequential and interactive multi-phase procedure starting by gathering information and ending by reporting the obtained results [1].

83

84

85

86

87

88

89

90

91

92

93

94

95

96

97

98

99

100

101

102

103

104

105

106

107

108

109

110

111

112

113

furthermore, others issues are usually related to the existing automated systems, notably the the
 congestion created in the assessed network triggering both security and performances issues along with
 the associated volume of data generated from the testing outputs that are often unexploited. Finally,
 PT environment is characterised as fast-changing and complex and the human experts are suffering
 from the complexity, repeatability and resemblance which in large and complex networks context such
 as large organisations using standard system and subsequently security protection. Performing PT
 in alike scenarios will create high degree of obfuscation and make it almost impossible to cover the
 whole asset properly [3-4].

During the last decade, the use of machine learning has intensified in the cyber security domain
 and especially in defensive applications such as intrusion detection and prevention systems (IDPS),
 Malware analysis and Anti-viruses solutions. Recently, MIT researchers developed a big data security
 framework named AI2 which combined security analyst expertise with machine learning to build an
 IDPS with active learning [2]. In the offensive cyber security domain, there have been few attempts to
 equip existing PT systems and frameworks with learning capabilities and thus for the obvious reason
 of complexity associated with PT practice notably into the modeling and design of an ML-led offensive
 security systems. In fact, it is natural to imagine that one or more machine learning techniques can be
 applied to different PT phases enabling systems to perform tests by learning and reproducing tests
 and thus improving efficiency and accuracy over the time [3] to reach systems capable of imitating
 human PT experts in performing intelligent and automated penetration [20].

In practical terms, incorporating ML in any PT system will at least reduce recurrent human errors
 due to tiredness, omission, and pressure. It will also boost system performance when performing
 different tests. ML-based automation will also relieve network congestion and downtime by reducing
 the number of tests by performing only relevant tests and doing that outside the regular business or
 office hours and thus avoid any type of assets' availability issues. Three core issues are expected to
 arise in a ML-based PT system. First, acquiring and generalising experience-use knowledge gained
 during the learning process for an optimal future use in similar situations. The second issue is
 adapting to the very particular context of learning that fulfil sequential decisions making with the
 rewarding process and approach (both automated and human-expert rewarding contexts). Finally, the
 exploration-exploitation trade-off aims to guarantee the best possible results within a reasonable use
 of resources. Furthermore, the training of such system will requires that the learning module be open

114 and able to interact directly with the expert to deal with complex situations by offering indications and
115 suggestions which can be accepted or rejected by the PT expert.

116 1.2. Paper Outline

117 In this paper, we are mainly concerned with the network perspective of the PT practice and we
118 will focus solely on the application of ML and specifically RL technique to the PT practice to make
119 it intelligent and efficient. The proposed solution can be extended to other types of PT such as Web
120 and application testing by introducing some changes in the core program. This paper will start with a
121 brief background on PT practice and highlight the fact that ML is so crucial to today's PT frameworks
122 and systems. The second section reviews relevant literature and surveys related works especially ones
123 tackling the uses of AI and ML in the PT practice and the limits and drawbacks of current PT. The third
124 section will briefly introduce the RL approach and justify this choice for the PT context along with
125 presenting the first version of the proposed model and its different components. Section 4 describes
126 the proposed system called Intelligent Automated Penetration Testing System (IAPTS), the adopted
127 learning approach and the modeling of PT as RL problem. Section 5 will describe IAPTS in more detail
128 as well as the performed tests and the obtained results within a specific context and test-bed network.
129 Finally we analyze and discuss the obtained results and make the relevant conclusions along with
130 highlighting future research works.

131 2. Literature Review

132 This work is rooted in a long line of applied research works on automating and optimising
133 offensive cyber-security auditing processes and systems especially vulnerability assessment (VA) and
134 PT [2, 4, 10]. Among the most significant contributions in this regard, we present here a summary of the
135 previously completed research with a special focus on the adopted approaches and the contributions.

136 Initially, researchers were interested in the planning phase. Some works were implemented within
137 the industrial PT systems and frameworks while others remained stuck at research ideas level [7-9]. As
138 PT automation and enhancement domain is situated between both cyber-security and AI research fields,
139 several axes of research were dressed started with the consideration of attack graphs and progressed
140 throughout different research fields and methodologies of Automated Planning consequently sub-area
141 of AI. Early research focused on modelling penetration as attack graphs and decision trees reflecting
142 the view of PT practice as sequential decision making [4]. Practically, most of the works were more
143 relevant to vulnerabilities assessment than to PT and among the most significant contributions in this
144 regard, we present in this literature review section a summary of the previously completed research
145 with a special focus on the adopted approaches and the contributions. For the purpose of clarity, we
146 start by dressing the full picture of the research in this field and we proceed later into dividing the
147 research axes by type, methodology, and approach [6].

148 2.1. Previous works on PT automation

149 Automation is an obvious approach to adopt for PT tasks when the objective is to produce
150 highly-efficient PT systems. Nonetheless, automating all the whole process of testing including the
151 versatile tasks and sub-tasks for each phase is challenging and often fails to reach the objective if done
152 in inappropriate way notably the use of automated tools and systems which blindly perform all the
153 possible and available tests without any optimisation or pre-processing [6-7]. The automated systems
154 require the permanent control of a human PT expert and often fail to produce acceptable results in
155 medium and large assets context because of the significant number of operations required to cover
156 the entire network [8-11]. In addition to the required time which surpass realistic duration of tests,
157 more others issues are created by automation such as the generated traffic (network congestion) and
158 the high number of false positives alerts triggered on the asset defence solution such as IDPSs and
159 FWs. giving what has been said, PT blind automation approach use was limited to a small network

160 and some medium size network with the use of customized scripts which are inconvenient requiring
161 substantial effort as well [3-4].

162 Early research focused on improving PT system by optimising the planning phase which was
163 modeled as attack graphs or decision trees problem which reflect the nature of PT practice as sequential
164 decision making. Most of the works were nonetheless relevant to vulnerabilities assessment (VA)
165 rather than PT because of the static nature of the proposed approach and its limitation to planning
166 phase [6-8]. Amongst the most significant contributions, we find the modeling of VA as attack graphs
167 in form of atomic components (actions), pre-condition and post-condition to narrow the targeted
168 vulnerability [11] but this approach was more an application of classical planning methods in order to
169 find the best attack graph. Further similar works were carried out on automating planning of PT tasks
170 but alike blind automation did not address the problem of enhancing performances and only covered
171 the planning phase of PT practice [3,12,16].

172 Nevertheless, a remarkable work on optimisation was introduced in [4] by modelling PT as
173 Planning Domain Definition Language (PDDL) which for the first time accounted for attacking
174 and post-attacking phases of PT in addition to the flexibility offered by the solution which enabled
175 integration with some PT systems [4]. The proposed solution generates different type of attack plans
176 (single and multi-paths) for real world PT scenarios which is then directly implemented within the
177 attacking and exploiting system and executed in the due course along with interacting with information
178 gathering tools for transforming the information acquired during that phase into input to a planning
179 problem to be solved separately and then used by the attacking system for the purpose of optimisation.
180 the only drawback of this approach was the scalability which was fatal as it was only limited to small
181 and medium size networks [6].

182 AI was also considered to improve PT practice in some research [5-9] but most of the proposed
183 modelling approach failed to deal with the persisting uncertainty in PT practice and especially the lack
184 of accurate and complete knowledge about the assessed systems. An exception was the use of ML
185 algorithms within a professional PT and VA system called Core-Impact in which researcher modelled
186 PT planning phase as a partially observable Markov decision process (POMDP) which was then solved
187 using external POMDP solver to determine the best testing plan in form for attack vectors. However,
188 the proposed model itself is questionable as it obviously fails to model the full PT practice and thus
189 can not cover the remaining testing phases and tasks especially the vulnerability assessment, testing
190 and pivoting phases reputed to be highly interactive, sequential and non-standard compared with the
191 planning and information gathering phases.

192 2.2. Drawbacks and limits of the current PT practice

193 In this subsection, we will present an overview of the domain of PT and the automation of
194 the practice along with highlighting the limitations of the current (existing) automated frameworks,
195 systems, and tools in dealing with the real-world situation. Penetration testing often involves routine
196 and repetitive tasks which make it particularly slow on large networks. These tasks are unfortunately
197 crucial for the practice and cannot just be dismissed although much of this routine can be automated.
198 Although the proposed solutions were in theory very relevant and seemed to solve the problem, the PT
199 practice demonstrated that the brought improvements were not enough to solve the core issue in the
200 practice which time and resources. Some solutions were on the other hand, fundamentally unfit and
201 inadequate for PT context. It is obvious that human capabilities and performance are limited when it
202 comes to large and complex tasks compared with a machine especially with nowadays computing
203 power.

204 The average penetration tester can spend days or weeks in testing a medium-size LAN (we are
205 concerned here by comprehensive testing when the entire network is covered). In addition to the
206 time and effort allocated, a considerable amount of systems downtime will be accounted as result of
207 the performed tasks. The first two points will be added to the poor performances in term of results
208 quality and accuracy including error and omission which could be crucial resulting from the fact that

209 human makes mistakes, change opinion and get bored. The penetration testing automation (automated
210 systems and tools) were therefore presented as the magic solution to the named issues. A fully or
211 even semi-automated solution was thus developed aiming to reduce human labor engaged in the
212 testing, save time, increase testing coverage and testing frequency and allowing Broader tests by
213 attempting more possibilities. The proposed solution was very diverse in term of adopted approach
214 when some relied on automated planning (phase 1) by generating automatically attack plan (named
215 attack graph) and then executing the attack in an automated manner. Others solution were more
216 creative and attempted to mimic the whole process and make it automated so the system can carry out
217 complex (chained) penetration testing tasks following different attack vector and use more exploits.

218 Cyber security research community start questioning the limits of the existing PT systems,
219 frameworks and tools which are expected to become more automated and perform most of PT tasks
220 with little or without human intervention and especially during the first 2 phases of PT; information
221 gathering and vulnerabilities' discovering. Organisations with constant need to internal security
222 auditing are, on the other hand, interested in more efficient PT systems which are fully automated
223 and optimised to perform basics and repetitive PT tasks without human intervention and therefore
224 alighting PT experts from that burden and dedicate them to more advanced tasks such testing advanced,
225 complex and non-common attacks [4-6]. Nonetheless, researchers were struggling with the automation
226 as PT practice is a complicate process which human barely master and therefore designing a machine
227 that replace PT experts in conducting tests is a challenging work giving the multi-phases nature of PT
228 practice with high-dependency between the different phases and tasks. alongside to the complexity of
229 the PT practice, the information handled is another major issue as PT reconnaissance and information
230 gathering phase usually produce incomplete profile of the assessed system and fail to yield a complete
231 knowledge and leave a certain amount of residual uncertainty, this issue is often dealt with by expert
232 by repeating some tasks, changing approach or simply making assumptions and continuing the tests.

233 On the top of the classic complexity associated to the PT, modern attacks upgraded their
234 capabilities by adopting evasive technique and complex attacking path that allow them to evade
235 network and systems defences. skilled attackers would usually seek to achieve their goals through the
236 exploitation of a series of vulnerabilities as part of a chain of sub-attack which enable them to can take
237 advantage of hidden (non-obvious) and composite vulnerabilities (composed of a chain of harmless
238 flaws when together become an exploitable vulnerability) in networks. Each part of the infrastructure
239 or systems may be approved to be secured when considered alone, but it/their combination and
240 interaction can often provide a pathway for an opportunistic attacker. The ability to detect and analyze
241 the interaction of multiple potential vulnerabilities on a system or network of systems leads to a better
242 understanding of the overall vulnerability of the assessed system [6]. Finally, PT output data is a
243 crucial issue because it is currently not used properly during retesting or future tasks and simply
244 discarded after the PT report generation. In cyber security context, only few security configurations
245 and systems' architecture change over short and medium term and therefore most of the previous tests
246 output remain applicable when a re-testing is required and this particular problem constitute one of
247 the key motivation of our research.

248 *2.3. Motivation and Contribution*

249 As a matter of fact, complexity is the worst enemy of control and thus security, computers
250 networks do not constitute an exception to this unanimous rule. During the last decade, protecting
251 and defending networks and critical digital assets from cyber threats required the security professional
252 to consider less classic approach (avoiding the trap of bolting on more and more security layers and
253 policies) and they turned their attention toward the offensive security. As with the real advances in
254 technology and thus cyber-criminality, cyber-security researchers were confronted with the need of an
255 intelligent PT system and framework to support human expert into dealing with high-demand on PT
256 and the associated complexity and risks by allowing systems to take over human and conduct some
257 or all of the PT tasks notably reconnaissances, information gathering, vulnerabilities assessment and

258 exploiting and therefore leave experts focusing on more complex issues such as post-exploiting and
259 testing complex attacks.

260 Giving what the aforementioned facts about PT practice, no other technique or approach rather
261 than ML seem to be fit to answer to our problem. In fact, several AI techniques were initially considered
262 and following a comprehensive suitability research only Machine Learning through Reinforcement
263 Learning was selected as the most prosperous option to allow an automated PT system to behave
264 like real tester in terms of operation mode and gain gradually the skills along with practice and thus
265 gathering information, assessing and exploiting in an intelligent and optimised manner allowing the
266 the discovery of all relevant and unlikely to be detected vulnerabilities and attacks to be tested along
267 with pivoting between different assets to mimic the work of the real hackers. This research comes to
268 bridge the gap in the current PT practice and will aim to resolve the following issues:

- 269 • Reducing the cost of systematic testing and regular re-testing due to human labor cost,
- 270 • Reduce the impact on the assessed Network notably the security exposure, performances and
271 downtime during the testing ,
- 272 • Alight human experts from the boring tasks repeatability during test and assign them to more
273 challenging tasks,
- 274 • Dealing more effectively with cyber threats' high emergence and fast changing rate (Short Lived
275 Patterns) by allowing flexibility and adaptability,
- 276 • Perform more broad tests by covering a wide variety of attack vectors and also consider complex
277 and evasive attacking paths which are hard to identify and investigate for human testers,

278 To sum up, cyber hackers seek to achieve specific goals through the exploitation of a series
279 of vulnerabilities as part of a chain of sub-attacks. Skilled attackers can take advantage of hidden
280 (non-obvious) and complex vulnerabilities (composed of a chain of harmless flaws) in a network
281 infrastructure or segment. Each part of the infrastructure or systems may be approved to be secured
282 when considered alone, but the combination or the interaction can result in opening a pathway for
283 an attacker. For this reason, the ability to assess and analyse the interaction of multiple potential
284 vulnerabilities on a system or network is becoming crucial in PT practice

285 3. Reinforcement Learning Approach

286 Cyber security system often categorised under two types; expert-driven or automated system
287 utilising unsupervised machine learning [5]. Expert-driven systems such as AVs, FWs, IDPSs and SIEMs
288 rely on security experts' input and usually lead to high rates of errors until Reinforcement Learning
289 (RL) techniques were used to give existence to more goal-directed learning systems that provide
290 autonomous or semi-autonomous decision making which accurately reflect real-world context of
291 cyber-security and especially offensive security domain such as vulnerabilities assessment and PT
292 context [12]. The main reason behind our choice of RL are:

- 293 • Effective autonomous learning and improving by allowing constant interaction with the
294 environment.
- 295 • Rewarding based learning and existing flexible rewarding schemes which might be delayed to
296 enable RL agent to maximize a long-term goal.
- 297 • Richness of the RL environment which help in capturing all major characteristics of PT including
298 the uncertainty and complexity.

299 As shown in Fig. 2, RL allows an agent to learn from its own behaviour within the RL environment
300 by exploring it and learning how to act basing on rewards received from performing actions undertaken.
301 This decision policy can be learned once and for all, or be improved or adapted if better results is
302 encountered in the future. If the problem is appropriately modelled, some RL algorithms can converge
303 to the global optimum which is the ideal behaviour that maximises the overall reward.

304 RL learning scheme exclude the need for a significant intervention from human who is expert
305 in the domain of application. In addition, RL implication will mean that less time is allocated for the

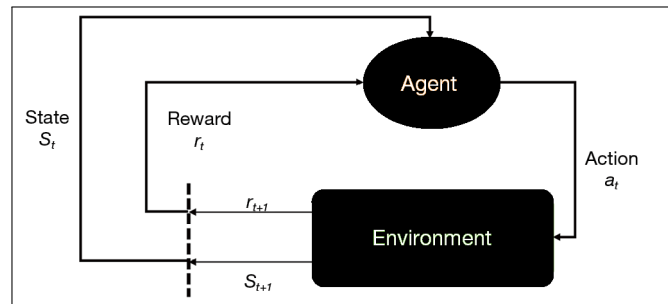


Figure 2. RL agent observes the state of the environment $x(t)$ at time t , selects an action $a(t)$ based on its action selection policy, and transitions to state $x(t + 1)$ at time $t + 1$ and receives a reward $r(t + 1)$. Over time, the agent learns to pursue actions that lead to the greatest cumulative reward [5].

306 learning and customisation as it is the case with ML and expert systems (ES) respectively. In addition
 307 to what has been said about the suitability of RL for enhancing the automation of PT solutions, RL
 308 branch is a very active domain of research and several new algorithms have been introduced recently
 309 along with some very efficient toolboxes and implementations with the ability of solving complex RL
 310 problem under constrained resources and producing great results [19].

3.0.1. Towards a POMDP modelling of PT

312 In PT, an attack is a set of tasks which are launched and executed, manually by a human tester or
 313 automatically by a PT platform, following a certain order in order to fulfill a goal or reach an objective.
 314 Depending on the context the goal can be predefined or unknown and also can vary throughout the
 315 attack. The goal (or also known as the objective) of the attack is known within the PT community as the
 316 target which can be either logical or physical entity. Often, the target is a computer (physical or virtual
 317 machine with an OS and running applications) or a computer network or some information hosted on
 318 a computer such as files, DBs or web-servers. The attack target can also switch during an attack if a
 319 more valuable or easily exploitable target is identified to serve as a pivot later on. Furthermore, it is
 320 also common that an attack has no specific target with the example of script kiddie hacker running a set
 321 of exploits against all reachable machines regardless of relevance in order to find one or more vulnerable
 322 to that specific attack [20].

323 The starting point for this research is an automated PT system which lacks efficiency and
 324 optimization in terms of the number of covered tests and the consumed resources and time as any PT
 325 test should not last forever and consume an excessive amount of time in performing or exploring irrelevant
 326 tests along with ensuring that no threat is ignored or underestimated. Therefore, the aim is developing
 327 RL-led autonomous PT systems which utilize RL and other techniques at different levels of the practice
 328 to improve performance, efficiency, testing coverage and reliability [20].

3.1. POMDP Solving Algorithm

330 RL algorithms are methods for solving real-world problems modeled in form of MDPs or POMDPs
 331 which usually involve complex and long sequences of decisions in which each decision affects what
 332 opportunities are available later and running for long-term goals. In this work, we are not
 333 concerned with the development of improvement of a new RL solving algorithm or methods, but only
 334 with finding the appropriate algorithm relevant to our problem and which produce acceptable results.

335 When it comes to solving a large and complex RL problem, it is often complicated and therefore
 336 an adequate choice of the solving algorithms and approach should be made. Therefore, for solving
 337 the PT POMDP complex environment, IAPTS should rely on different solving algorithms rather
 338 than simply one, in fact, depending on the context IAPTS will adapt to utilize the most adequate solving
 339 approach. Furthermore, the choice of different algorithms is justified by the constraints IAPTS may face
 340 in terms of the available resources (time, memory and computational) which make the use of one solving

341 algorithm challenging and thus adopting a flexible approach where the accuracy is often sacrificed
342 to acceptance. Finally, it is important to remind that large environment can also cause challenges to
343 solving algorithm especially when dealing with a large number of transitions and observations or
344 opting for a static rewarding schemes [20-24].

345 most of RL solving algorithms fall under to major categories; the reward (value) oriented solving
346 and policy search solving. The reward approach allows an RL agent evolving within the environment
347 to select the sequences of actions that lead to maximising the overall received reward or minimise
348 received sanctions in the long term run and not only in the immediate future, this approach aims to
349 dress an optimised and comprehensive rewarding function which rely on the atomic reward values
350 associated with the RL environment to determine and an optimal (best possible) rewarding scheme
351 (function) for each transition and observation. In term of efficiency, this solving approach is often
352 complex and time consuming with several cases of an infinite horizon if the problem representation is
353 not enough consistent and optimised. The second approach, namely policy search seeks to construct
354 a decision policy graph which is in practice done by learning the internal state/action mapping of
355 the environment and uses direct search method for identifying policies that maximizes long term
356 reward, optimal policy is reached when all the states and all the actions are tried and allocated a
357 sufficient amount of time to find the best possible associated policies. in this research we opted for
358 the use of both reward-optimisation and policy-search approaches. Nonetheless, for the purpose
359 of implementing policy-search algorithms we found that it is useful to include both On-policy and
360 Off-policy implementation to allow a better evaluation in term of policies quality. The IAPTS POMDP
361 solving module will use a powerful off-the-shelf POMDP-solver allowing the use of different solving
362 approaches and state-of-the-art algorithm [19] to allow the exclusion of all external factor when it
363 comes to evaluating different solving algorithms performances. Initially the following algorithm were
364 shortlisted:

365 3.2. *PERSEUS algorithm*

366 PERSEUS is a randomized point-based Value Iteration for POMDPs proposed by [5] performs
367 approximate value backup stages to ensure that, in each stage, the value of each point in the belief set
368 is improved. The strength of this algorithm is its capacity of searching through the space of stochastic
369 finite-state by performing policy-iteration alongside to the single backup which improve the value of
370 the belief points. Perseus backs up also a random basis by selecting a subset of points within the belief
371 set which are enough to improve the value of each belief point in the global set. In practice, PERSEUS
372 is reputed to be very efficient because of the approximate solving nature and is the best candidate for
373 solving large size POMDP problems as it operates on a large belief set sampled by simulating decisions
374 sequences from the belief space leading to significant acceleration in the solving process.

375 3.3. *GIP algorithm*

376 GIP (generalized incremental pruning) is a variant of POMDP exact solving algorithm family
377 relying on incremental pruning. GIP algorithm replaces the LPs that were used in several exact POMDP
378 solution methods to check for dominating vectors. GIP is mainly based on a Benders decomposition
379 and uses only a small fraction of the constraints in the original LP. GIP was proven in [19] that it
380 outperforms commonly used vector pruning algorithms for POMDPs and it reduces significantly the
381 overall running time and memory usage especially in large POMDP environment context. The latest
382 version of GIP is, to the best of our knowledge, the fastest optimal pruning-based POMDP [21].

383 3.4. *PEGASUS algorithm*

384 PEGASUS is policy-search algorithm dedicated to solving large MDPs and POMDPs and was
385 initially proposed by [13] and adopts a different approach to the problem of searching a space of policies
386 given a predefined model as any MDP or POMDP is first, transformed into an equivalent POMDP
387 in which all state transitions (given the current state and action) are deterministic and thus reducing

388 the general problem of policy search to one in which only POMDPs with deterministic transitions
389 are considered. Later, an estimation value of all policies is calculated making the Policy-search
390 simply performed by searching for a policy with high estimated value. This algorithm has already
391 demonstrated huge potential as it produces a polynomial rather than exponential dependence on the
392 horizon time making it an ideal candidate to the penetration testing POMDP solving.

393 3.5. Other candidates

394 In addition to the candidates, other RL algorithms will be considered such as Backwards Induction
395 and Finite Grid, this last is instance of point-based value iteration (PBVI) and will be mainly utilized
396 in determining the shortest attack-path when more than one policy is found. Some of the proposed
397 algorithms are already part of the POMDP-solver software and an optimized implementation is
398 provided by the contributor and constantly improved over the versions. Nonetheless, some algorithm
399 was implemented and integrated for the sake of bench-marking. Initially, and as the research focus was
400 to dress a high-quality POMDP model representation for the PT practice bridging the gap between the
401 theoretical research and real-world situation facing the industry professional, the use of such “ready
402 solution” was highly recommended and was hopeful in advancing the research and also for the impact
403 of the results obtained.

404 3.6. POMDP solving choices

405 PT is a complex practice in which the targets can be known or unknown, global or local, simple or
406 composite and each phase is a sequence of non-standard tasks in which the order is a crucial factor.
407 Therefore, the IAPTS should reflect to the best the real-world domain of PT and RL approach here
408 is meant to address the kind of learning and decision-making problems that allow the PT system to
409 capture, reproduce and store expertise in the whole PT tasks and sub-tasks relying on well established
410 RL solving algorithms elected to be the fit to PT context and produces acceptable results [23-24]. The
411 PT practice is thus represented as POMDP environment and serve as an input to the off-the-shield
412 solver in which a decision-making agent will be exploring its environment to aiming to maximize the
413 cumulative reward it receives or finding the optimal policies graph (PGs) through the RL agent which
414 perceives the environment and solve the problem by estimating the value function to to dress the best
415 decision policies or rewarding function [20].

416 4. IAPTS Design and Functioning

417 The proposed Intelligent Automated Penetration Testing System (IAPTS) functioning diagram is
418 detailed in Figure 3. Python scripts were developed to perform the pre-processing from the raw data
419 and then the produced results is used into optimising the representation of the PT domain in form
420 of POMDP problem. The IAPTS knowledge base (memory) will be initially handled manually and a
421 human PT expert will decide on the storage of the obtained results (policies extracted after applying the
422 generalization) along with the management of tasks related to expertise extracting and storing. In other
423 words, the extracted expertise will be performed manually until the IAPTS reach a pseudo-maturity
424 state in which it will be in charge of capturing, assessing and storing the expertise will be implemented
425 and embedded within the IAPTS expertise memory. The projected IAPTS will be an independent
426 module that can be embedded with the industrial PT framework. the current version of IAPTS is
427 associated with Metasploit Framework (MSF) as external module communicating via customised
428 python scripts with MSFRPC API. The purpose of such configuration is to avoid modifying the core
429 component of the MSF and allowing us, for research purpose, to measure the IAPTS performances
430 away from the PT framework.

431 4.1. IAPTS operative modes

432 IAPTS will evolve through different levels of automation and intelligence to reach the
433 pseudo-maturity level in which it should be able to perform an entire PT on networks. Overall,

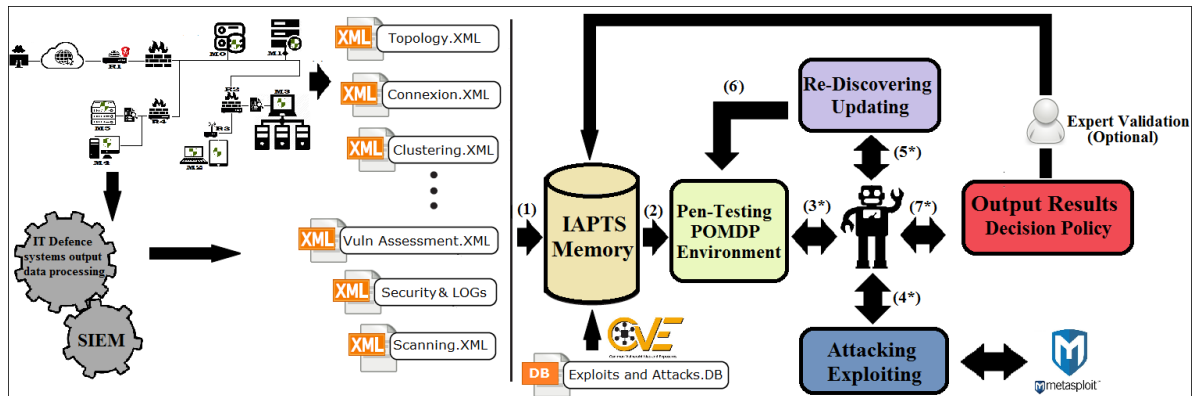


Figure 3. IAPTS functional diagram.

434 IAPTS can operate in four different levels which are dictated by the development of the system
 435 knowledge base in term of captured and generalised expertise as follow:

- 436 1. Fully autonomous; IAPTS entirely in control of testing after achieving maturity so it can perform
 437 PT tasks in the same way that human expert will do with some minor issues that will be reported
 438 for expert review.
- 439 2. Partially autonomous; the most common mode of IAPTS and reflect first weeks or months of
 440 professional use when IAPTS will be performing tests under constant and continuous supervision
 441 of a high-calibre PT expert.
- 442 3. Decision-making assistant; IAPTS will shadow human expert and assist him/her by providing
 443 pinpoint decision on scenarios identical to those saved into the expertise base and thus alight
 444 tester from repetitive tasks.
- 445 4. Expertise building; IAPTS running in the background while human tester perform tests and
 446 capture the decisions made in form of expertise and proceed to the generalisation and of the
 447 extracted experience and build the expertise base for future use.

448 4.2. From PT to a Reinforcement Learning Problem

449 We present here an improved version of the modelling of PT practice as a POMDP problem which
 450 constitute the core module of IAPTS. for simplicity purpose, we use an illustrative example to introduce
 451 the different steps towards the representation of a PT domain in form of POMDP problems. In the
 452 context of PT, we believe that there is no need to represent the entire network topology and security
 453 configurations in the RL environment but only representing specific data judged relevant from the
 454 PT point of view and thus alighting the RL environment [20]. The RL representation will capture the
 455 following information about the assessed network: machines and networking equipment architecture,
 456 connectivity and reachability, network defence and security configurations. The aforementioned
 457 information will be used to dress a PT-style view of the network without encumbering the RL
 458 environment and impact the performances. In addition, we used pre-processing output relevant
 459 data to be included within the environment or to serve in enhancing RL learning algorithms to acquire
 460 such as proxy server logs, web-server logs, database logs, routing device logs, apps and other security
 461 logs.

462 4.3. representing Network PT as RL Environment

463 We describe here the process of elaborating an RL environment starting from a given PT example.
 464 The overall extraction and elaboration process is explained, in mirror with the PT diagram, in Figure 6
 465 in which we build upon the IAPTS logic into converting PT domain into POMDP representation along
 466 with respecting real world PT and adopting the same approach into the elaboration of the POMDP
 467 environment sections. noting that all the sections are dynamic and allow high frequency changes as

468 the PT progress and information are updated or upgraded. The following are the different components
469 of the RL POMDP environment:

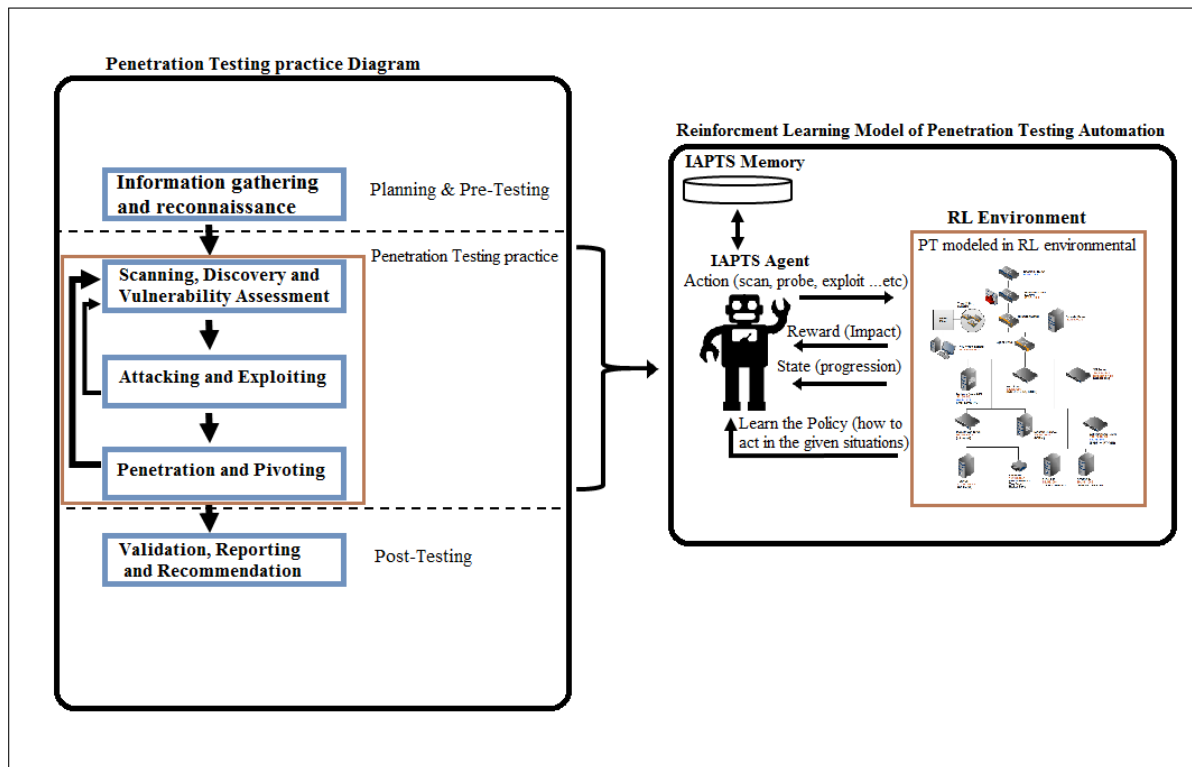


Figure 4. IAPTS modelling of PT as RL problem diagram.

470 *State space*: contain all relevant information, from PT expert view, about the assessed network. It
471 will include information about any software or hardware machine including virtual and networking
472 equipment that run an OS. the information are OS parameters, port, services and applications, patches
473 in addition to relevant security and connectivity information. These information are represented in
474 POMDP language using a special notation that aims to minimise the size of the file but remain concise,
475 clear and precise. In practice most of the Action space is dressed at early level as modern PT rely
476 on initial information gathered during the first phases. nonetheless, some information will remain
477 missing or not accurate enough and thus represented in a probabilistic way after being enhanced by
478 information coming from the pre-processed output to avoid redundant or useless representations [20].
479 Any machine or device within the network will be assigned a number "i" and will be represented as M_i
480 or R_i and the remaining associated information are represented in, but not limited to, the following way
481 M_i -OS1-Port80-ServiceABC or R_i -OS2-Port443-SerciceXXX. These information will be continuously
482 updated as the discovering and scanning tasks progress to confirm previous probabilistic information
483 or to add a new one. Furthermore, modern network Routers are more than simple transmission
484 equipment, in fact they can run Operating Systems and embed one or more security isolation and
485 protection mechanisms notably FWs, AVs, IDPSs, VLANs and others. Following this logic, network and
486 firewalls can be considered as machines (running OS and thus having vulnerabilities) or just security
487 isolation boarder for clustering purposes detailed later.

488 In addition to the machine and devices information, state space will include information about
489 the networking and security configuration of the assessed network such as connectivity, security
490 isolation (sub-net, virtual LAN) and defence restrictions. the purpose of such representation is to
491 enhance and optimise the input for the POMDP solving algorithms so a better RL environment is
492 represented. The following example summarise the information captured about two machines M_i
493 and M_j as M_i - M_j -TCP-SSH-0". Only relevant security and networking configurations information are

494 considered and machines that belong to the same segment and have the same protection should be
495 represented together then we represent other segments' machines.

496 *Action space:* POMDP model action are an exact reflection of the PT actions performed by testers
497 and thus en-globe all PT tasks and sub-tasks following a certain notation. as with any RL problem the
498 number of action is known, static and limited and PT does not fall out of this logic and we include in
499 this space as variety of Pt related actions such as Probe, Detect, Connect, Scan, Fingerprint, VulnAssess,
500 Exploit, PrivEscal, Pivot in addition to some generic action that will be used for control purpose by RL
501 agent.) that the expert can perform is huge and cannot be totally represented within the RL action
502 space such as Terminate, Repeat and Give-Up and others as detailed previously in [20]. Furthermore,
503 as in PT domain successful or failed action might require further or repeating actions we defined some
504 additional actions in order to differentiate between the original action and the others action. in practice,
505 the purpose of such re[presentation is to deal with the special and complex scenarios notably:

- 506 • a failed action to fully (root) control a machine that lead to further action attempting user-session
507 or escalate privileges or switching to other attack paths;
- 508 • dealing with action relying on uncertain information and fail because of the assumption made and
509 require further actions when additional information become available and might be successful;
- 510 • actions prevented or stopped by security defence (Ws and IDPSs) which may be re-attempted
511 following different circumstances.

512 4.4. POMDP Transitions and Observations probabilities

513 in the first phase of this work, transitions and observations were uniformly sampled. nonetheless,
514 after multiple attempts we found-out that in the particular context of PT, it is far more efficient and
515 reasonable to use real-world data built from IAPTS past tests and enhanced by the human-expert
516 initially meant to passively supervising the IAPTS. the data used to artificially simulate testing
517 environment is captured and stored by IAPTS during the regular testing but is carefully inspected by
518 the authors who will rely on their expertise to only include the adequate data and discard the rest
519 of the data. in addition to the regular output of the past experiences, failed or incomplete testing
520 scenarios will be of a crucial use during the retesting process. in fact, as IAPTS aims to gradually
521 replace human expert in PT, the system should act as human in dealing with failure into performing
522 some PT tasks or successfully carrying-out tests. similar to human IAPTS will uses an evaluation
523 procedure to recognise that what have been done could be useful in another context or with minor
524 amendments for the similar context. in IAPTS, we rely initially on human expert interaction to provide
525 a feedback on the failed and incomplete testing to select and store the highly prominent ones for future
526 use even if they ultimately failed. in term of data, IAPTS will be mainly dealing with the Policies
527 stored into the PG file which constitute the outcome of the POMDP problem solver [18-20].

528 In this research, the probabilistic output of PT action (scanning, fingerprinting, exploiting)
529 was a crucial factor we considered doing allocation the adequate probabilities for Transitions and
530 Observations in order to mirror the real-world PT practice. therefore, we opted for a cross validated
531 method using two well-established and standard sources respectively NIST National Vulnerability
532 Database (CVSS) [18] and Common Vulnerability and Exploits (CVE) which constitute a reliable online
533 catalog for all known proven vulnerabilities associated with different type of Operating systems,
534 software and Applications. The use of such sources is motivated by the rich content, easy accessibility,
535 regular update and the available scoring function and mechanism such as CVSSv3 and the calculation
536 of the Probabilities associated to each transition or observation is detailed in [20].

537 4.5. Rewarding schema

538 On the other hand, IAPTS Rewarding will be twofold depending on the system maturity. In early
539 stage, IAPTS will rely solely on the rewards allocated by the PT expert supervising it along with some
540 default rewarding values. rewarding the performed actions will be predefined by human expert who
541 will have to decide on the adequate reward for each action performed depending on his/her overall

542 sight he got on the practice, experience and testing achievements. Afterward, IAPTS will alight the
543 human expert from the rewarding task and only request human decision on the global PG (attack
544 policies). IAPTS reward function will be utilised and thus the reward for the performed actions will be
545 calculated following a well-established criteria such as: reaching a terminal state; achieving a final
546 (global) target or local goal (controlling an intermediates machines); or failing to reach any goal. The
547 criteria for the choice of rewards will mainly be: the estimated value of the achievement, the time
548 consumed; and the associated risk of detection as detailed in [20].

549 *4.6. IAPTS memory, expertise management and pre-processing*

550 This research is all about applying RL learning into medium and large LANs which subsequently
551 mean that the projected system IAPTS will need to deal with big amount of intimation described as
552 complex and redundant amongst the cyber-security community. modeling and representing the PT as
553 POMDP environment is particular complex and will result in producing a huge POMDP environment
554 and thus make it impossible to solve giving the restriction in time and computing power (memory).
555 Therefore, an optimising and smart use of resources is required and the problem modeling is where
556 all start. The system memory as shown in Fig. 5 is used for dynamically storing the data handled by
557 the system such as the environment's attributes (States, Action, Observation, transition, Reward) and
558 agent's memory (data regarding the Policy and Acquired knowledge and experiences that an agent
559 gains by acting within the environment). In fact, the first part of this research will focus exclusively
560 on searching the policy as an agent acts within the environment in a particular state and receives
561 a reward from the environment. Initially, for a purpose of research facilitation, the reward value
562 will be pre-defined by a human expert so no reward function will be used. Moreover, generalize
563 experience output for further use – use knowledge gained in similar situations to Equip penetration
564 testing system with “expert knowledge” will be completely done by this module in the future. in
565 practical term, IAPTS will solve the RL problem, extract PGs and instruct MSFRPC API which will
566 execute the testing plan and keep updating the IAPTS of the outcome on real-time base especially at
567 vulnerability assessment and exploiting phases. this will enable IAPTS to adjust and adapt the tests as
568 well as the the post-exploitation tasks such as pivoting or privilege escalation

569 In addition to the RL framework on which the system will operate, a parallel knowledge-based
570 expert system will be implemented and constantly (with every practice) enhanced and fed. This
571 pseudo-system will serve as RL initial belief. This system will capture details of the performed
572 (manually) action by the human tester and also extract knowledge from the output of the information
573 gathering phase and Security system data (Firewalls, AVs, IDPSs, SIEMs) and structure the relevant
574 details. One can say that such system will be useless alongside with the RL system which is a legitimate
575 interrogation. The answer will be that giving the known limits of RL in multi-dimensional state context
576 along with the important size of the RL components, including initial belief detail will only slow down
577 the system performance. Furthermore, the crucial information extracted from the security data will
578 otherwise be omitted.

579 The only remaining issue is the human intuition (the ability to acquire knowledge without
580 inference and/or the use of reason) which a system will not be able to substitute. Intuition provides
581 penetration tester expert with beliefs that cannot be justified in every case and human can sometimes
582 solve some brilliant problem without the use of any reasoning. Artificial decision making is the
583 ultimate aims of the use of AI but still cannot model the intuition. As results, this issue will be sorted
584 out by allowing the controlling human to interact with the system. In other words, a mechanism to
585 obtain feedback from the expert tester (security analysts) should be utilized to overcome this issue. The
586 feedback (along with the surrounding context) will be stored in the system memory for future use. the
587 system memory will incorporate policies assessment and generalization features and experience-replay
588 form previous test where the human expertise is extracted and defined as a policy automatically by the
589 system (direct learning) along with the management of the input and output data such as the initial
590 belief and reporting.

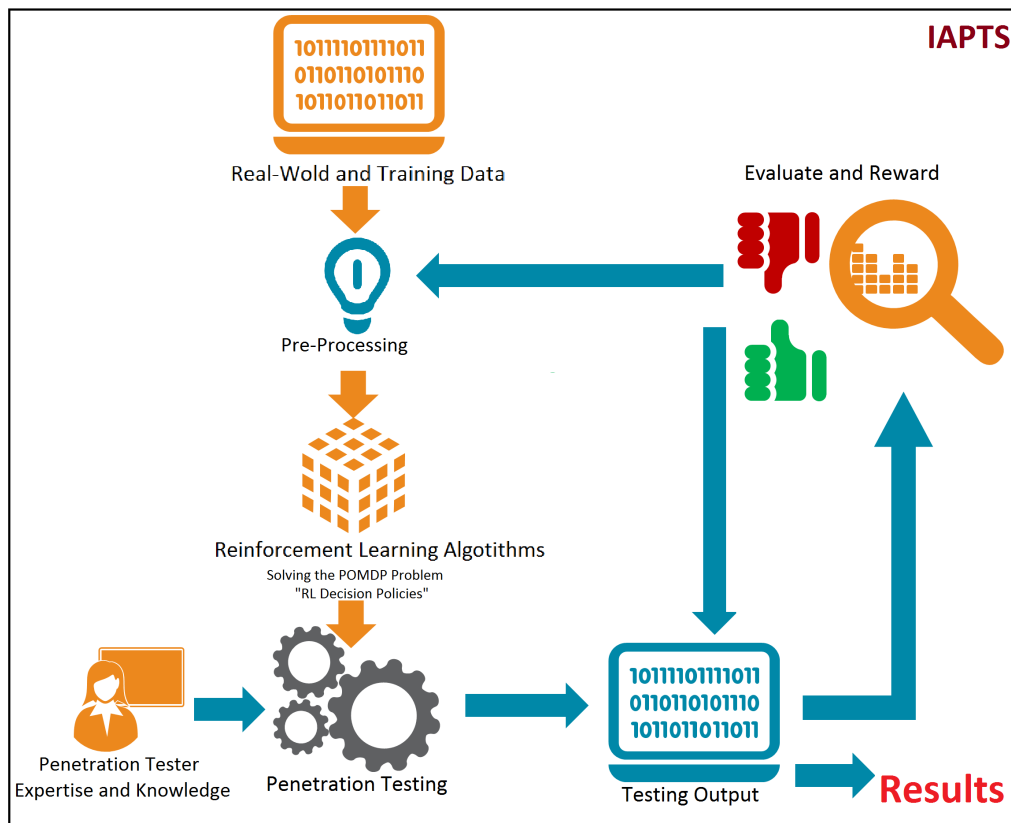


Figure 5. IAPTS learning, expertise extraction and validation procedure

591 Prioritised experiences' replay is an effective approach to improve the learning and thus efficiency
 592 in RL algorithms. In this work, we adopted this approach, but introduced some modifications for
 593 technical reasons, in order to enable RL algorithms to prioritise the use of certain sequences of
 594 transitions over others in order to enhance the learning of the IAPTS RL agent. In addition to selecting
 595 the most plausible and relevant policies (state-action decision sequences), we injected some other
 596 artificially construct transition sequences using information gathered from previous tests which were
 597 validated by a human PT expert. These sequences, when replayed, allow value function information to
 598 trickle down to smaller selection of POMDP transitions and observations, thereby improving solving
 599 algorithm efficiency in term of consumed time and memory. all the proposed customization were
 600 implemented within a modified version of the standard POMDP solving GIP LPSolve algorithm we
 601 called "with Initial belief".

602 Finally, it is important to introduce our modified GIP LPSolve algorithm which was meant to
 603 improve the performance of IAPTS and also allow the IAPTS to capture the appropriate expertise in
 604 form of decision policy) process it to make it general decision rule and store it within IAPTS memory
 605 for future use. the simplest way to illustrate the importance of the learning on the long-term PT practice
 606 by adopting a test scenario inspired from the real-world situation of re-testing the same network
 607 after some updates or upgrades. In the retesting process, one or more machine configuration will be
 608 changed but not all of the machines and therefore IAPTS will re-use already acquired PG when it
 609 comes to repeat PT on the partially modified network with the use as initial belief the output of the
 610 previous tests.

611 5. Testing IAPTS and Results

612 5.1. Simulation platform

613 To test IAPTS we designed and implemented several test-bed networks of different sizes. In the
 614 first phases of this research, we aimed to assess the effectiveness of the proposed POMDP modelling of
 615 PT and evaluating our choices in terms of learning approaches, used algorithms, and capturing and
 616 managing the expertise as we discussed in details in [20]. The adopted test-beds are, to the best of our
 617 knowledge, an illustration of the real-world networks used widely by different type of organisations
 618 which include; Internet connected side, DMZ, Intranet and internal sensitive segments where crucial
 619 data is kept securely. In this work, we tested IAPTS performance on different size experiment networks
 620 composed of a number of machine (computer) and networking routers varying from 2 to 100 machines.
 621 networking equipment are considered as machines as well as any network equipment that runs an OS
 622 and applications. The only excluded machine is the hacker(s) computer(s) which will be represented
 623 as one entity along with the Internet. Figure 6 shows a sample test-bed network of 10 machines (seven
 624 computers and three routers).

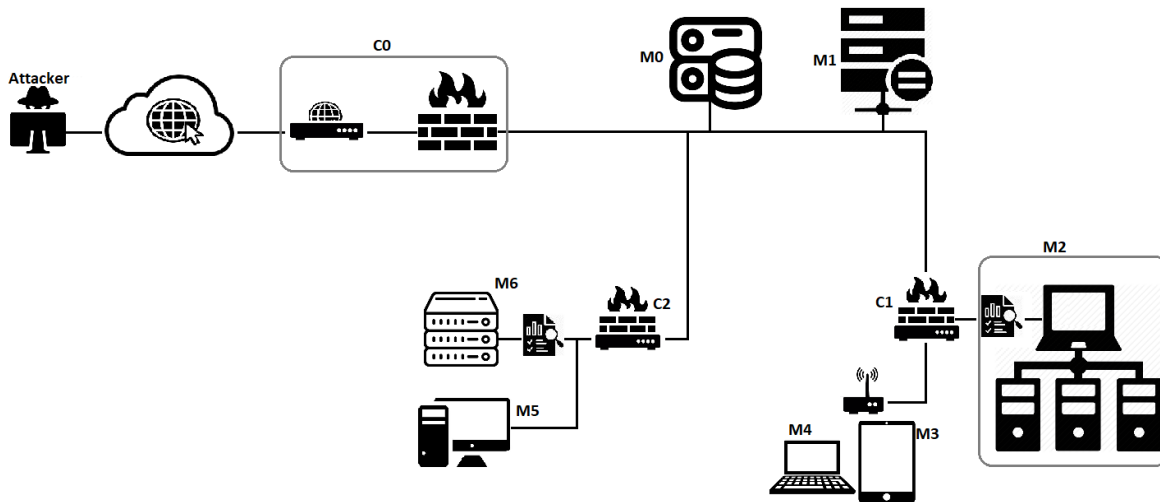


Figure 6. 10 Machines test-bed network.

625 5.2. IAPTS results

626 evaluating IAPTS is multi-folds operation starting by validating the RL approach, then examining
 627 the obtained results in solving real-world PT associated POMDP problems, and finally analysing
 628 the relevance and accuracy of the obtained results from PT point of view. In practice, the output of
 629 the RL solving is acting Policies Graphs (PGs) which undergo additional processing to convert the
 630 results into a more understandable format. In addition to the consumed time for solving the POMDP
 631 problem, other factors will be considered notably the time required to perform different PT tasks by
 632 the Metasploit MSF and other variables which are either calculated or approximated in order to define
 633 the overall consumed time that IAPTS will take to perform a full testing on the test-bed networks. The
 634 obtained results shown in the Figures 7, 8 and 9 illustrate an initial comparison of different RL solving
 635 algorithms performances on different LANs which were also compared with manual PT consumed
 636 time time basing on author experience as PT consultant and also the overall time required to perform
 637 an automated comprehensive PT with no optimisation (refer to results obtained by authors in [20]) and
 638 it is clearly obvious that IAPTS outperform by far both manual and automated PT. In addition, different
 639 discount rates were considered in the optic of finding the suitable balance between performances
 640 enhancements and preserving the realistic nature of our IAPTS. therefore, the discount rate of "0.5"
 641 was selected following multiple testing and simulations.

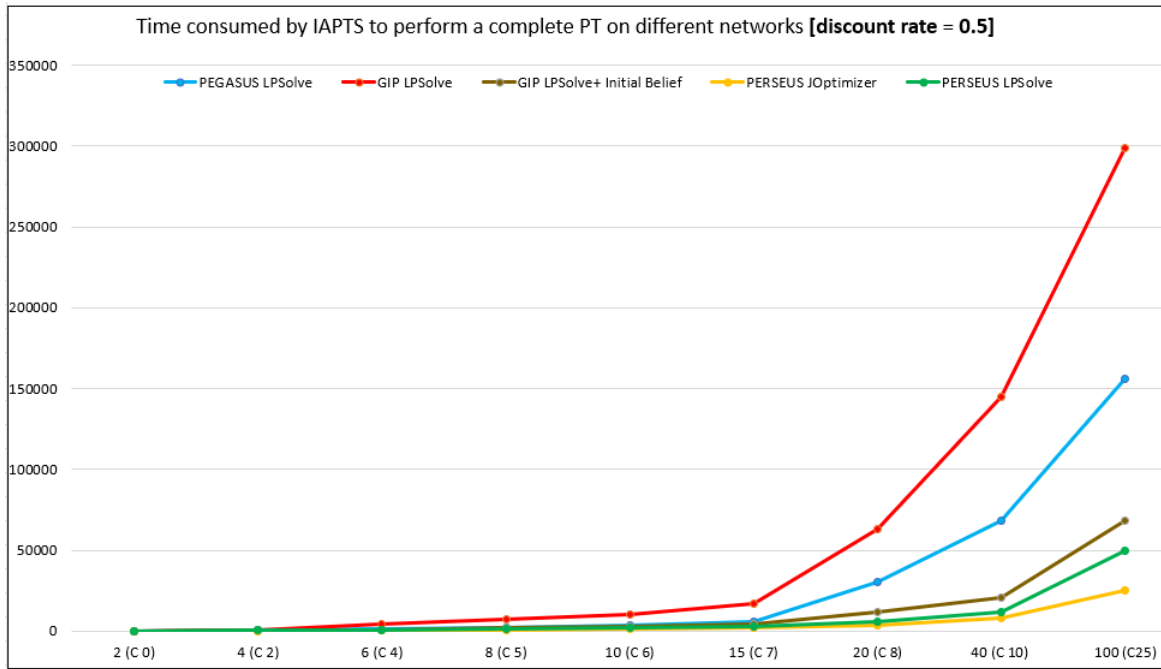


Figure 7. Time in seconds required by IAPTS to complete PT tasks on different LANs' sizes with a Discount rate of 0.5

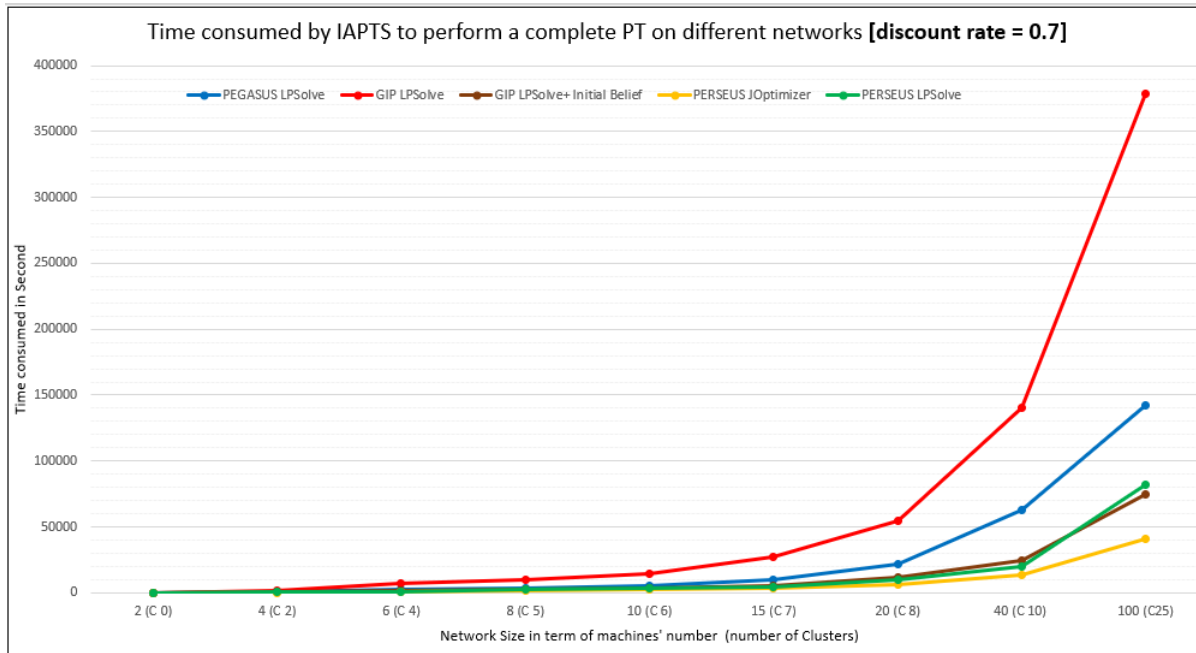


Figure 8. Time in seconds required by IAPTS to complete PT tasks on different LANs' sizes with a Discount rate of 0.7

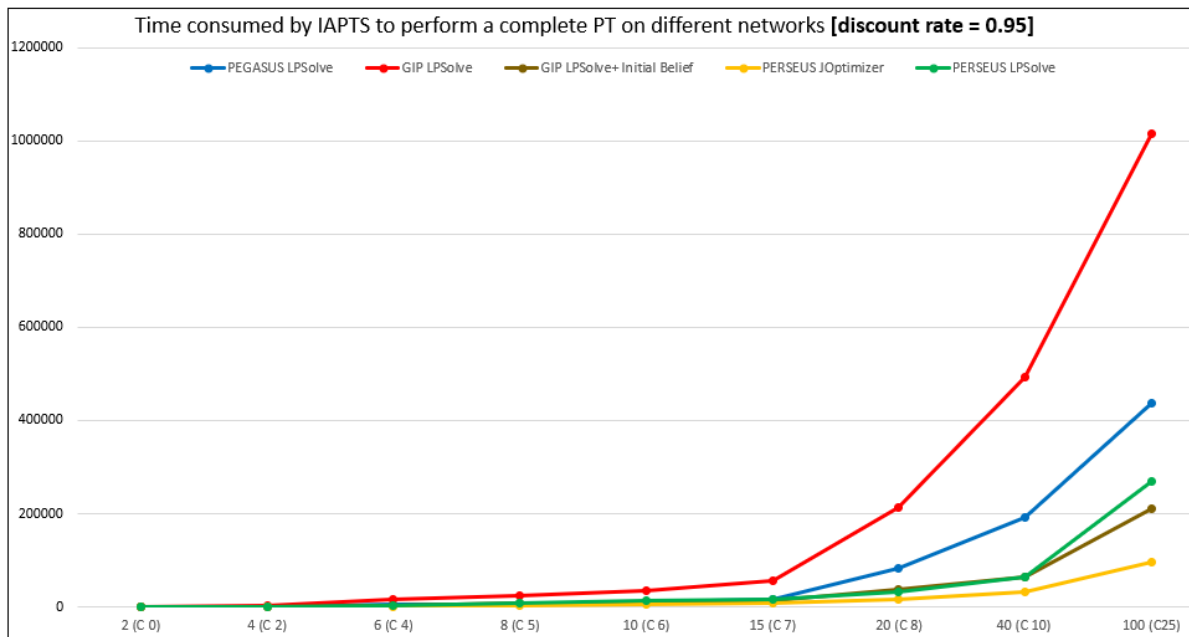


Figure 9. Time in seconds required by IAPTS to complete PT tasks on different LANs' sizes with a Discount rate of 0.95

642 Following the obtained results, we decided to introduce some changes within the solving
 643 algorithm and notably GIP aiming a better performance from IAPTS on short term basis. we opted
 644 for prioritized Transitions and Observations through the manipulation of the associate probabilities
 645 along with introducing some customisation into the initial beliefs sampling. the obtained results were
 646 surprisingly excellent and the new variant of GIP which we named GIP-LPSolve with Initial Belief
 647 performed much more better than the classic GIP in both time consumed and PG accuracy as shown
 648 in Figures 7, 8 and 9. Furthermore, in order to assess IAPTS performance expertise extraction and
 649 storing capabilities and the impact of performance enhancement we proceeded to the re-testing the
 650 same network with or without introducing minor or major changes to different number of machine
 651 configuration. the obtained results in the context of a 10 machines LAN were near to perfect as the
 652 performance enhancement was huge especially when re-testing the very same network as shown in
 653 Figure 10.

654 Finally, on the top of the overall performance enhancement and notably when using GIP LPSolve
 655 with initial belief algorithm, the quality of the produced decision policies was beyond human expertise
 656 especially in the case of 10 machines network when IAPTS highlighted two additional attack vectors
 657 which an average human PT expert would easily omit and illustrated in Figure 11.

658 5.3. Discussion and future works

659 the obtained results consolidate prior thoughts on the role of ML and specifically RL in the
 660 performance enhancement and resources-use optimization in PT. Commercial and open-source PT
 661 systems and frameworks were deigned initially to work either under human instructions or in a
 662 blindly automated manner, but both approaches fail to address the current environment in which PT
 663 practice is evolving notably the increasing size and complexity of the networks, the high number of
 664 vulnerabilities and the composite testing scenarios which mimic modern hackers operating approaches.
 665 RL revealed very efficient when used properly and IAPTS results are an additional evidence as in
 666 addition to the drastic performances enhancement comparing to an average human testers, several
 667 other positive points were noticed notably the pertinence of the produces result (acting policies) in term
 668 of relevance, coverage and accuracy. In practical term, using the adequate RL algorithm and adopting
 669 a new learning schemes enabled IAPTS to produce a very optimised attacking policies when targeting

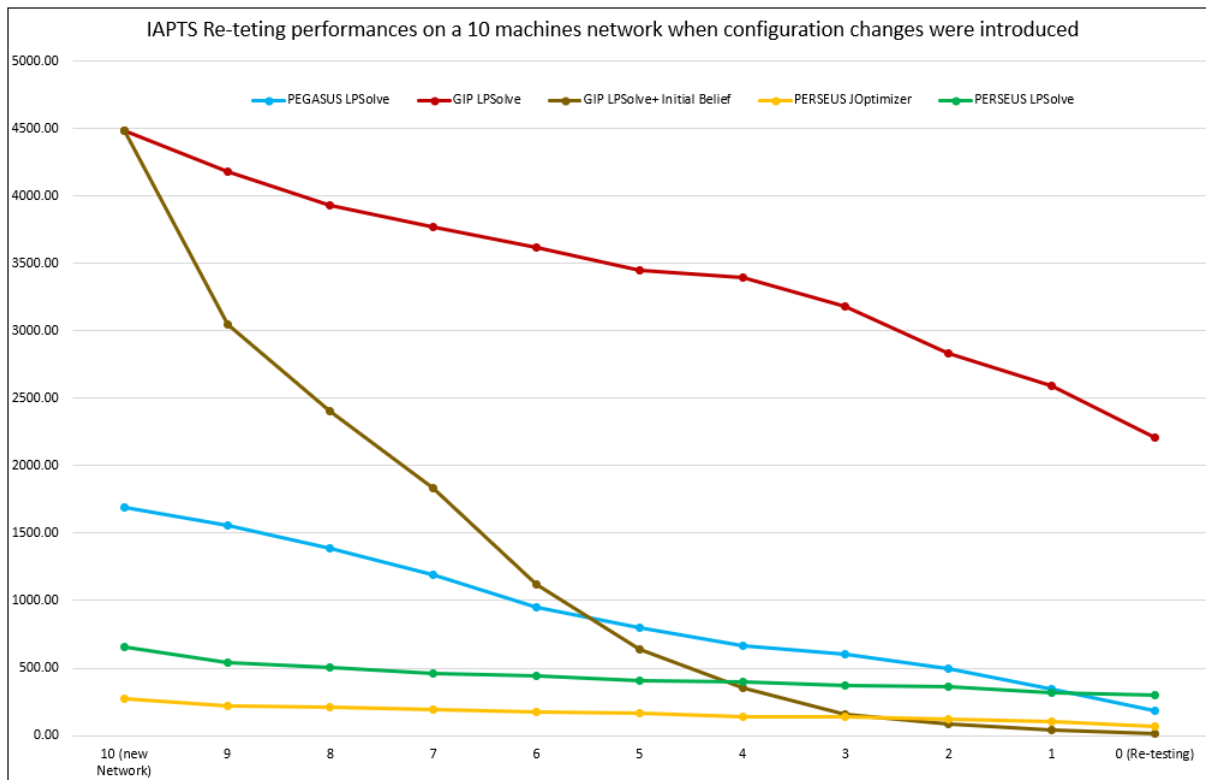


Figure 10. IAPTS re-testing performances' enhancement by algorithm on 10 Machines LAN

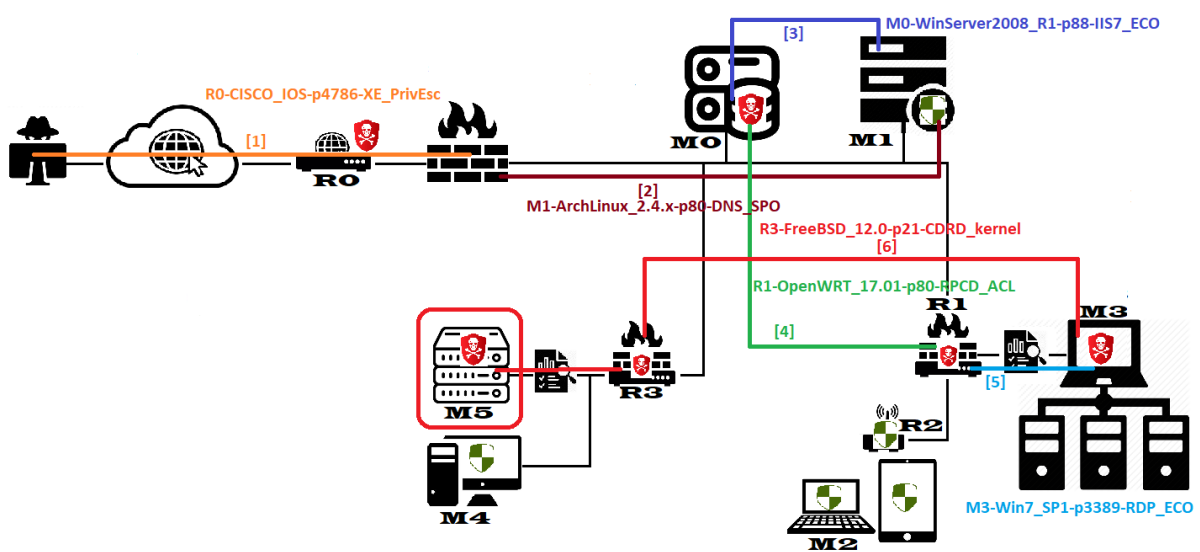


Figure 11. Example of IAPTS output PT policy translated into attacking vectors

670 the Machine M5 suspected to contain sensitive information and defined as the most secured machine
671 within the test-bed network as illustrated in Fig. 11. Indeed, the produced policy is from an attacker
672 point of view obvious but getting an automated system to opt for such attacking vectors despite being
673 not minimal in term of cost of the exploits and consumed time is the novelty in IAPTS which is able to
674 sacrifice simplicity for a higher objective. IAPTS exploring and large coverage capabilities was able
675 to find a very complex and non-obvious attacking path in medium size networks where, relying on
676 authors experience, no human tester will be tempted to adopt and possibly neglect in spite of being
677 very relevant and constitute a possible attack path which a real hacker can chance it.

678 Furthermore, the proposed enhanced GIP-LPSolve which utilise a new mechanism in creating
679 and managing POMDP initial belief was proved very efficient especially in small and medium size
680 LANs. in fact GIP LPSolve is a variant of an exact solving RL algorithm which are often labeled as
681 good in results quality but bad in performances, but the introduced changes in initial belief sampling
682 and managing along with prioritising some decision sequence over others enabled the new variant
683 to perform much more better and even outperform other RL approximate solving algorithms. On
684 the other, the re-testing of the same network after the introduction of minor changes in few machine
685 permitted to appreciate the full contribution of RL to PT practice by cutting drastically the consumed
686 time and thus allowing a fast and reliable re-testing which is often the case in PT when periodic
687 re-testing is compulsory despite the lack of any significant configuration changes within the networks
688 systems.

689 Finally, we noticed that IAPTS performances on large size LANs decreases sharply and this is
690 mainly due to the complexity which impact the size of the POMDP environments along with usage of
691 memory during the solving of the problem. This major issue is currently being dealt with by proposing
692 a hierarchical PT POMDP model relying on grouping several machines under the same cluster which
693 will be detailed in future works along with improving IAPTS pre-processing.

694 6. Conclusions

695 This paper explores a novel application of reinforcement learning techniques into the offensive
696 cyber-security domain which allows penetration testing systems and frameworks to become intelligent
697 and autonomous and thus perform most of testing and re-testing tasks with no or little human
698 intervention. the proposed system named IAPTS can act as a module and integrate with most of
699 the industrial PT frameworks to improve significantly the efficiency and accuracy on medium and
700 large networks context. The proposed modelling of PT in form of RL problem allowed the coverage
701 of the entire PT practice and thus producing a system fit for the real-world context, the current
702 implementation of IAPTS is integrated to the most commonly used PT frameworks called Metasploit
703 and permitted highly efficient testing in term of consumed time, allocated resources, covered tests and
704 accuracy of the produced results. The main drawback of IAPTS is the need of high-calibre human
705 expert supervision during early learning phases where a human trainer will perform PT along with
706 IAPTS and adjust the learning and veto the output of the system to ensure a good quality training by
707 acting as rewarding provider for the RL agent actions.

708 The major contribution of this approach is to apply RL techniques to a real-world problem
709 of automating and optimising PT practice and resulted into a net improvements of PT framework
710 performances notably in terms of consumed time and covered attack-vectors as well as enhancing the
711 produced results reliability and persistence which will lead optimistically to a PT system free from
712 human-error. The second major contribution of the system will be the capturing the expertise of human
713 experts without instructing it as IAPTS will rely initially the expert feedback in form of rewarding
714 values until it reach a certain maturity. Thirdly, IAPTS will on the top of saving time and reduce human
715 labour, increase testing coverage by attempting tests that a human expert won't be able to explore
716 because of the frequent lack of time. Finally, IAPTS permit the re-usability of the testing output by
717 either learning and/or capturing the expertise during test and storing it with the system memory for

718 future use and was proved to be very efficient on re-testing scenario (very common in PT) and nearly
719 similar cases when the testing time and accuracy of the produced results were exceptional.

720 References

- 721 1. J. Creasey, and I. Glover, A guide for running an effective Penetration Testing program,
722 <http://www.crest-approved.org>. CREST Publication , 2017.
- 723 2. N. Almubairik, G. Wills, Automated penetration testing based on a threat model. 11th International
724 Conference for Internet Technologies and Secured Transactions, ICITST, 2016.
- 725 3. A. Applebaum, D. Miller, B. Strom, C. Korban, and R. Wol, Intelligent, automated red team emulation. 32nd
726 Annual Conference on Computer Security Applications (ACSAC '16), 2016, pp. 363-373.
- 727 4. J. Obes, G. Richarte, and C. Sarraute, Attack planning in the real world. Journal CoRR Article, 2013,
728 [abs/1306.4044](https://arxiv.org/abs/1306.4044).
- 729 5. M. Spaan, Partially Observable Markov Decision Processes, Reinforcement Learning: State of the Art,
730 Springer Verlag, 2012.
- 731 6. J. Hoffmann, Simulated penetration testing: From Dijkstra to aaTuring Test++. 25th Int. Conf. on Automated
732 Planning and Scheduling, 2015, AAAI Press.
- 733 7. C. Sarraute, Automated attack planning. Instituto Tecnologico de Buenos-aires, Ph.D. Thesis, 2012, Argentina.
- 734 8. X. Qiu, Q. Jia, S. Wang, C.Xia and L. Shuang, Automatic generation algorithm of penetration graph in
735 penetration testing, 19th Int. Conf. on P2P, Parallel, Grid, Cloud and Internet Computing, 2014.
- 736 9. C. Sarraute, O. Buffet and J. Hoffmann, POMDPs make better hackers: Accounting for uncertainty in
737 penetration testing. 26th AAAI Conf. on Artificial Intel. (AAAI'12), pp. 1816–1824, July 2012.
- 738 10. C. Heintl, Artificial (intelligent) agents and active cyber defence: policy implications. 6th Int. Confe. on
739 Cyber Conflict. NATO CCD COE Publications, 2016, Tallinn.
- 740 11. M. Backes, J. Hoffmann, R. Kunemann, P. Speicher and M. Steinmetz, Simulated penetration testing and
741 mitigation analysis. <http://arxiv.org/abs/1705.05088>, 2017.
- 742 12. S. Jimenez, T. De-la-rosa, S. Fernandez, F. Fernandez and D. Borrajo, A review of machine learning for
743 automated planning. The Knowledge Engineering Review, Vol. 00:0, pp. 1–24. 2009.
- 744 13. Y. Andrew and M. Jordan, PEGASUS: A policy search method for large MDPs and POMDPs. 16th Conf. on
745 Uncertainty in Artificial Intel., 2013.
- 746 14. T. Schaul, J. Quan, I. Antonoglou and D. Silver, Prioritized experience replay, Google DeepMind. ICLR 2016.
- 747 15. K. Veeramachaneni, I. Arnaldo, A. Cuesta-Infante, V. Korrapati, C. Bassias and K. Li, AI2: Training a big data
748 machine to defend. CSAIL, MIT Cambridge, 2016.
- 749 16. K. Durkota, V. Lisy, B. Bosansk and C. Kiekintveld, Optimal network security hardening using attack graph
750 games. 24th Int. Joint Conf. on Artificial Intelligence (IJCAI-2015), 2015.
- 751 17. N. Meuleau, K. Kim, L. Kaelbling and A. Cassandra, Solving POMDPs by searching the space of finite
752 policies. 15th Conf. on Uncertainty in Artificial Intel., 2013.
- 753 18. NIST, Computer Security Resource Center - National Vulnerability Database, <https://nvd.nist.gov>, 2018.
- 754 19. E. Walraven, and M. Spaan. Accelerated Vector Pruning for Optimal POMDP Solvers, Proceedings of the
755 Thirty-First AAAI Conference on Artificial Intelligence, 2017.
- 756 20. M. Ghanem, and T. Chen. Reinforcement Learning for Intelligent Penetration Testing. World Conference on
757 Smart Trends in Systems, Security and Sustainability. 2018.
- 758 21. C.Dimitrakakis, and R. Ortner. Decision Making Under Uncertainty and Reinforcement Learning. Book
759 chapter. 2019.
- 760 22. I. Osband, D.Russo, and B. Van Roy. efficient reinforcement learning via posterior sampling. In NIPS, 2013.
- 761 23. R. Grande, T. Walsh, and J. How. Sample efficient reinforcement learning with gaussian processes. In
762 International Conference on Machine Learning, pages 1332–1340, 2014.
- 763 24. S. Agrawal, and R. Jia. Optimistic posterior sampling for reinforcement learning: worst-case regret bounds.
764 Annual Conference on Neural Information Processing Systems, 2017, pages 1184–1194, 2017.

765 © 2019 by the authors. Submitted to *Information* for possible open access publication
766 under the terms and conditions of the Creative Commons Attribution (CC BY) license
767 (<http://creativecommons.org/licenses/by/4.0/>).