Reinforcement Learning Agents for Simulating Normal and Malicious Actions in Cyber Range Scenarios

Alessandro Santorsola[1,*], Aldo Migliau[1] and Sabino Caporusso[1]

[1]BV-Tech s.p.a., 20123 Milan, Italy

Abstract

Cyber-attacks and their consequences have become one of the primary sources of risk in recent years. Cyber-attacks have the potential to cause physical damage both to infrastructures and to people. To prevent such risks, several methods have been proposed. Cyber security knowledge required for cyber defense can be developed by active learning in a cyber range. Although this type of cyber learning is popular and used worldwide by numerous organizations and companies, typically such simulations lack the presence of users and their relative effects on the systems. In particular, in a cyber environment where the only activities on the systems are those carried out by the Red Team, the assessment of malicious actions on the systems will be a trivial activity for the Blue Team. Hence, the reality of the resulting simulation does not reflect a real working condition. Users simulation is needed for providing more realistic scenarios for training sessions. Additionally, a cyber range that relies on the actions of simulated users introduces the possibility to simulate a *Zero Trust* (ZT) condition. In such scenarios, the simulated users act also as virtual attackers or use social engineering attacks (i.e., phishing) within the company network.

This work presents the development of a model whose purpose is to generate human-addressable actions in the cyber range. Moreover, the agent leverages a *Reinforcement Learning* (RL) algorithm to simulate the user-system interactions. Finally, the agent simulates both normal and malicious actions on the systems.

Keywords

Cyber Range, Reinforcement Learning, Simulation & Modeling, Cyber Attacks

1. Introduction

The rapid technological advancements (e.g., *Internet of Things* (IoT), 5G) have become the main transformation source for several IT/OT domains (e.g., energy, health care, public transport) by increasing their productivity, value creation, and the social welfare [1]. Despite these flourishing perspectives, the insufficient knowledge jointly with the lack of security awareness provides a fertile ground for several threat actors [2]. Threat actors may carry out different types of attacks that can produce tangible damages. In fact, there are several organizations or companies that

ITASEC'22: Italian Conference on Cybersecurity, June 20–23, 2022, Rome, Italy

[*] Corresponding author.

a.santorsola@bv-tech.it (A. Santorsola); a.migliau@bv-tech.it (A. Migliau); s.caporusso@bv-tech.it (S. Caporusso)

https://www.linkedin.com/in/alessandro-santorsola-680aaa145/ (A. Santorsola)

0000-0003-1094-8199 (A. Santorsola)

own or access to different cyber systems that can be exposed to several known and/or unknown attack vectors.

The majority of the cyber attacks have involved the categories of Transportation and Storage, *Industrial Control System* (ICS), Government, Healthcare and Entertainment [3]. Furthermore, the proliferation of the IoT devices in industrial plants (e.g., power grids, gas, and water distribution systems) led to an increasing transformation of the traditional ICS. Not only, due to the migration of the control components from the electronic world to the software one, the resulting ICS components are exponentially increased in complexity. Consequently, this led to the sudden increase of the attack surface. Moreover, the increasing trends include remote users, personal devices, and cloud-based assets that are not physically located within an enterprise-owned network boundary but are always reachable. According to the ZT paradigm, the aforementioned operations can generate normal alerts, but they can be also part of a more sophisticated cyber attack [4]. The resulting expanded attack surface jointly with the well-known capabilities and motivations of advanced cyber adversaries, has made modern-day critical infrastructure more likely to be compromised.

To provide the right tools to the workforce so that they are able to face such risks, several methods have been proposed. One of the possible solutions takes into account training platforms (i.e., cyber range). Cyber Range is a virtual environment that enables organizations to simulate cyber training, system/network development, testing, and benchmarking. Usually, such training follows the Red vs. Blue Teams format aiming to improve responsive capacity in case of a cyber crisis.

In a cyber range, the workforce can practice themselves in detection and response strategies using real-world tools and techniques. Despite these flourishing outcomes, such simulations do not consider the presence of users and their relative effects on systems. It is a matter of fact that users within a corporate network can be the additional source of security alerts linked with unauthorized operations or sporadic human errors. As a consequence, the lack of users' presence and their relative effects on the systems will result in a trivial assessment of malicious actions performed for the Blue Team. Hence, in order to simulate a real working condition, it is mandatory to include such behaviors in the cyber environment. Moreover, network traffic is generated in order to create external and/or internal requests (i.e., traffic that came from the Internet or Enterprise Intranet). In this fashion, the Blue Team activities will be much closer to a real situation. These activities will require the capability to discriminate the Red Team actions from the other kind of traffic.

In this paper, we take into account both the statistical characterization of a series of requests performed by a user and the possibility to vertically embed RL in such processes. Hence, we propose the adoption of an agent-based model to generate independent users network traffic according to specific network topology and security mechanisms by exploiting the potentiality of RL. The reported results show that the embedded RL agents generate human-addressable independent requests that are consistent with realistic traffic patterns for the specific network.

The paper is organized as follows: in Section 2 the contributions regarding cyber range and the application of reinforcement learning in such scenarios will be analyzed. This is followed by the description of the proposed contribution in Section 3. In Section 4 the experimental settings and results will be reported and commented. Finally, Section 5 concludes the work and draws the future perspective.

2. Background and Motivations

Reinforcement Learning is a machine learning technique in which a computer (i.e., agent) learns to perform an activity through repeated "trial-and-error" interactions with a static or dynamic environment. This approach to learning allows the agent to make a series of decisions that maximize a reward metric for the activity, without being explicitly programmed for such an operation and without human intervention [5]. Hence, the agent does not receive the information of what action to take as in other forms of machine learning, instead, it must find out which actions will produce the best reward in each state from interactions with the environment. Q-learning [5] is a model-free RL technique in which the action selection is based on the rewards (i.e., feedback). The agent always chooses the optimal action. The future reward function in the S_t state when performing an A_t action, denoted as $Q(S_t, A_t)$, is assimilated by interactions with the environment. The equation for updating the value function of the state-action pairs $Q(S_t, A_t)$ is based on the value-action function expressed as [5]:

$$Q(S_t, A_t) = Q(S_t, A_t) + \alpha \Big[R_{t+1} + \gamma \max_a Q(S_{t+1}, a_{t+1}) - Q(S_t, A_t) \Big] \tag{1}$$

where α represents the learning rate, γ is the discount factor and t represents the time instant. In general, $0 \leq \alpha \leq 1$ and $0 \leq \gamma \leq 1$. R_t represents the reward extracted from the feedback. Finally, $Q(S_t, A_t)$ represents the value of the Q-function when the agent is in the state S_t for the action A_t. A relevant aspect of the Q-learning approach is the choice of actions to be performed during the process of estimating the $Q(S_t, A_t)$ value function. This can be performed by using any method of exploration/exploitation or even randomly. The most used policies are: (i) random, (ii) epsilon-greedy, and, (iii) softmax. In the random strategy, the choice of the best action is modeled as a uniformly distributed random variable. In the epsilon-greedy strategy, the agent uses both exploitation to take advantage of prior knowledge and exploration to look for new options [6]. The epsilon-greedy approach selects the action with the highest estimated reward most of the time. In particular, let ϵ be a small probability and let P_M be the discrete uniform distribution probability function. For each t_{th} round, the model compares the $P_M(t)$ value with respect to ϵ in order to choose the exploration, i.e., not to exploit what the model has learned so far. Hence, if the exploration is chosen, the model adopts the random action selection. Whereas, if the exploitation is chosen, the model selects the upcoming action with the higher reward. It is crucial to underline that exploration is more important when the agent does not have enough information about the environment it is interacting with. Hence, the agent needs to interact optimally with the environment, allowing it to exploit its knowledge. As a consequence, ϵ should decay across the life of an agent to have it learn and act optimally. Finally, the softmax strategy gives every action in the set of possible actions a chance to be chosen, based on the estimated reward value of the action. Actions with higher values will have a higher probability to be chosen. A Boltzmann distribution is used to evaluate the action-selection probabilities. The distribution can be defined as follows:

$$P_t(a) = \frac{e^{\frac{Q_t(s,a)}{\tau}}}{\sum_i^K e^{\frac{Q_t(s,i)}{\tau}}} \tag{2}$$

where $P_t(a)$ is the probability that an action will be chosen, K is the total number of possible actions, and, τ is the temperature parameter, which indicates the amount of exploration.

Several approaches for cyber range development are reported in the literature. These approaches can be classified according to the specific simulation environment (e.g., IT, Industrial) or according to the nature of the cyber range itself (e.g., hardware and software infrastructure, simulation, virtualization, and hybrid approaches). Each approach is characterized by different advantages and disadvantages related to the following aspects: (i) complexity, (ii) learning efficiency, and (iii) cost.

For instance, in [7] a physical replication of an electric grid ICS hardware and software is used to recreate an environment that is truly representative of real-life processes. This approach is certainly closer to reality, but it is characterized by a higher cost and complexity. In addition, no machine learning techniques have been included.

In [8], a simulation-based cyber range is presented. The authors propose a sandbox environment that provides similar functions to a real system. Like other simulation-based cyber ranges, this contribution presented the following advantages: (i) reconfigurability, (ii) maintainability, and (iii) scalability. However, it does not provide high fidelity, especially when software exploits and cyberattacks need to be considered simply because network and/or physical interactions are not present. In [9], the authors aim to explore existing studies of AI-based cyber attacks and to map them into a dedicated framework, providing insight for new threats. The framework includes the classification of several aspects of malicious uses of *Artificial Intelligence* (AI) during the cyber attack life cycle and provides a basis for the detection to predict future threats. The authors report different types of application of their proposal to analyze AI-based cyber attacks in a hypothetical scenario of critical smart grid infrastructure. In [10], the authors start their research from the following assumption: adversaries show no restraint in adopting tools and techniques that can help them attain their goals. In particular, the authors used AI and machine learning approach to solve security challenges. Autonomous agents interactions have been investigated within a simulated enterprise network. Moreover, RL techniques have been considered to improve security. The simulations take into account the enterprise environment by using the high-level abstraction of computer networks and cybersecurity concepts.

In [11], the authors describe the design of cyber security virtualized cyber range designed to provide a flexible environment to evaluate cyber security tools. In particular, those tools involve AI/ML to provide realistic environments. The strengths and weaknesses of the tools have been also investigated. In this contribution, the challenges are related to the evaluation of the performances and operating costs for AI/ML-based cyber security tools for application into large, government-sized environments. In [12], a novel-modular framework is proposed to replicate complex SCADA Systems in a virtual simulation. The authors analyze the process of virtualization of each major component and they present a real world critical infrastructures as case studies. The authors demonstrate the use of the framework for cybersecurity research by including different cyber attacks.

Recently, cyber range development has investigated the possibility to combine simulation, virtualization, and physical device replication approaches in a single hybrid cyber range [13] [14]. The hybrid approach offers the possibility to overcome the disadvantages associated with the other types of cyber ranges. In [15], the authors present the development of a hybrid cyber range for ICS that is based on a real-time attacker-defender gameplay model in conjunction with dynamic simulation models of typical industrial systems. Moreover, the authors present an industrial gas turbine as one use case of an archetypal industrial system. Finally, the authors provide a demonstration of a sample training exercise. Finally, in [16], the authors propose an approach to minimize response time and the impact of cyber-attacks on the organizations. The authors propose a formative evaluation in the context of a digital twin implementation in the EU electrical power sector.

As shown in the literature, the majority of the applications of ML/AI-based models and algorithms in the cyber range are addressable to anomaly detection and malware behavior purposes. Moreover, the aforementioned contributions do not take into account the presence of users during the simulations. The contribution proposed in this paper wants to address this problem by developing two cooperative Reinforcement Learning Agents whose purpose is to simulate the presence of users within the virtual environment of the cyber range. In addition, the proposed contribution takes into account the possibility that such simulated users may introduce security events in such a way that can be human-addressable. Finally, the model can simulate the presence of additional attackers within the network that acts as a smoke screen for the Red Team. To the best of our knowledge, our contribution is the first one that addresses this kind of problem.

3. The Proposed Model

3.1. Statistical Characterization of Users Networking Actions

The statistical characterization of user actions takes into account the following observations: (i) the model has to generate network traffic in such a way that it can be addressed to independent users, (ii) the model has to take into account both external and internal network traffic (i.e., network traffic that came from Internet and Enterprise Intranet), and, (iii) the inter-arrival times between those requests have to be exponential. Hence, the external/internal traffic generation can be modeled as a series of Independent and Identically Distributed (i.i.d.) requests. Finally, the traffic generation process can be classified as a memory-less process. According to these considerations, the network transaction generation process can be modeled as a non-homogeneous Poisson Process. A non-homogeneous Poisson process is a Poisson process with a time-varying rate. It can be used to model the arrival times of customers at a store, users requests to services, events of traffic, and positions of damage along a road [17]. In particular, the model inter-request time is characterized by the following probability density function:

$$f(x, \lambda(t)) = \lambda(t)e^{-x\lambda(t)} \tag{3}$$

where $x \geq 0$ and $\lambda(t)$ is the rate function (i.e., requests per second) [18]. To guarantee the property of i.i.d. requests concerning external and internal traffic two solutions have been investigated: (i) selection via uniform binary probability distribution function, and, (ii) run two different and separated instances for simulating external and internal requests. Both the aforementioned solutions are valid.

3.2. Characterization of User - Service Interactions

In general, an high-level structure of an enterprise network can be composed as follows:

- External Firewall that exposes some corporate services to the Internet;
- Delimitarized Zone (DMZ) that contains and exposes some corporate resources (e.g., Web Server, FTP Server, Mail Server);
- Security Network that hosts Blue Team machines (e.g., SOC);
- Servers Network that hosts different IT systems (e.g., Active Directory Server);
- Hosts Network that hosts users workstations that have no interactions with other external networks.

The network traffic usually observes a set of rules that are defined at the firewall level. Considering as an example the incoming traffic from the external networks, this will affect only those services that are exposed by the company firewall and that are present in the DMZ. Furthermore, the same considerations can be done concerning the intranet-traffic. The definition of a set of firewall rules to permit or to block inter-subnet traffic is a common security best practice. Theoretically, by evaluating each source-destination combination it is possible to establish if a connection is permitted or not. Moreover, this quantity does not take into account the number of possible high-level actions that can be performed to the specific service, e.g., login procedures, HTTP GET methods, and files upload. However, this approach will be inefficient if the network is larger and highly segmented. Despite the computational problem, a possible solution is given by modeling the network as a cyber environment in which an agent has to perform a set of actions. Hence, a Reinforcement Learning approach can be adopted. In this work, we have trained two Q-Learning RL Agents that operate at the transport and application level of the TCP/IP stack respectively. In particular, the agents model the interaction between a user and the network in the specific cyber environment. In a nutshell, the cyber environment is automatically defined by the network topology jointly with the services and the firewall rules. At the end of the training, the Networking RL Agent will be able to generate a series of transactions compliant with respect to the environment. In the same fashion, the Application-Level RL Agent models the higher-level interactions to the specific service (e.g., login procedures, uploads procedures). Moreover, the second RL Agent can generate normal, malicious, or idle interactions according to a specific behavior. Hence, our model supports a learning and knowledge system that allow it to perform network operations correctly.

3.3. Network Traffic Characterization

Network traffic model has to generate and manage different connections with different sources. To address this problem, a series of networking plug-ins have been developed. These modules are implemented in order to manage the packets exchanged between the machine on which our model is installed and the target servers. In this perspective, the plug-ins manage TCP and UDP connections. Moreover, it is possible to define custom-made payloads. The plug-ins also implement messages such as echo request and reply (i.e., ping), DNS and ARP Request, and, finally, the handling of connections, login, and commands exchange with the servers. Additionally, it is possible to simulate a *Distributed Denial of Service* (DDoS) attack. Moreover, other types of malicious traffic are modeled (i.e., Brute Force, Host Discovery and Port Scan). As a final remark, phishing emails delivering and malicious file uploading can be performed by the model.

3.4. Model Architecture

Figure 1 shows the general architecture of our model. The cyber environment is automatically defined by the network topology, the services, and the firewall rules. The network plug-in module realizes the interface between the Q-Learning Agents (i.e., Networking and Application-Level RL Agents) and it provides all the networking functions. The configurations module provides the setup parameters both for the RL algorithm and for the services to emulate the desired traffic pattern (i.e., normal or malicious actions). The current implementation requires proper setup with topology information, firewall configurations, and network rules, which limits the applicability to pre-defined and well-known scenarios.

The RL Engine is composed by the following elements:

1. RL Dispatcher, acts as an interface between the RL Learners, it delivers initial configurations and the network primitives to the RL Learners and, finally, provides the action to be performed to the transaction generator;
2. RL Memory, stores the RL Learners Q-Tables;
3. RL Policy, implements the strategy used by the agent in pursuit of goals;
4. RL Learners, implement the Q-Learning algorithm.

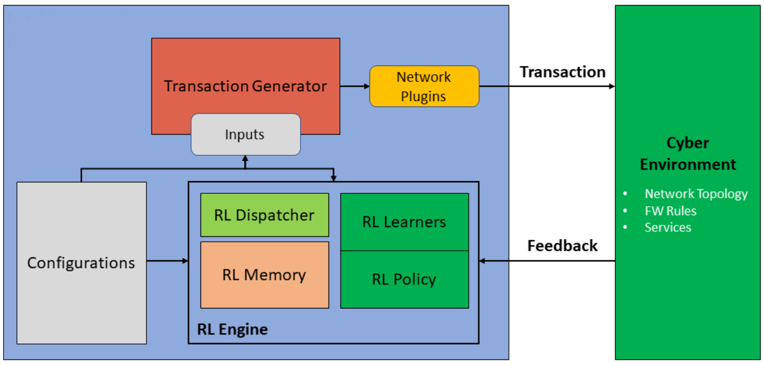


Figure 1: General Model Architecture.

Hence, we have chosen to develop two Q-Learning RL Agents that operate at networking and application levels respectively. In the learning phase, the RL learners update Q-Table values by evaluating the Q-function expressed in Equation 1. The RL policy rules the action selection and it can be defined in terms of Markov Decision Process [5]. The following policies are available within the model: (i) random strategy, (ii) epsilon-greedy strategy, and, (iii) softmax strategy.

Finally, the Q-tables are subdivided according to the traffic type (i.e., external or internal traffic) and with respect to the specific TCP/IP working level. Moreover, they are built dynamically according to number of services, sources within the network, agent behavior, and application-level functions. Hence, the transaction generator module builds a transaction request, and the model invokes the appropriate network plugin to correctly manage the connection.

4. Experimental Evaluation

4.1. Experimental Settings

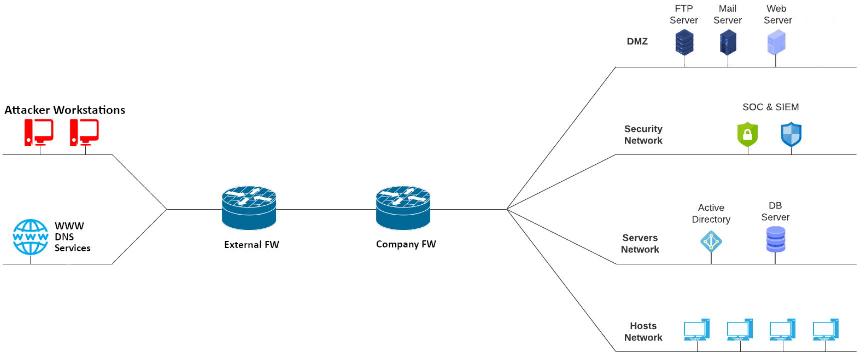


Figure 2: Reference Cyber Range Environment.

In Figure 2, the reference cyber range structure is shown. In particular, the company network is composed as follows:

- DMZ: FTP, Mail and Web Servers. The FTP and Mail servers are the only exposed services;
- Security Network: SOC and SIEM;
- Servers Network: Active Directory and DB server;
- Hosts Network: workstations.

The overall network is monitored by the Security Operations Center (SOC) and the SIEM and the network traffic is filtered by the company firewall. The firewall configurations are reported in Tables 1 and 2. The external firewall, the attacker workstations, and the other services are introduced to simulate Internet within the cyber range scenario. The rules regarding the redirected traffic to the SOC are not reported. In addition, the firewall default outcome for no-matching-rule traffic is the "block all" policy.

Table 1
Firewall Configuration for Exposed Services.

Allowed/Blocked	Source	Destination
ALLOW	*	FTP:21
ALLOW	*	Mail:25/143/993

Table 2
Firewall Configuration for Intranet Traffic.

Allowed/Blocked	Source	Destination
ALLOW	Servers Network	FTP:21
ALLOW	Hosts Network	FTP:21
ALLOW	Servers Network	Mail:25/143/993
ALLOW	Hosts Network	Mail:25/143/993
ALLOW	Hosts Network	WWWs:80
ALLOW	Hosts Network	Servers Network
BLOCK	DMZ	Servers Network

4.1.1. Model Deployment

The model deployment should take into account both the computational resources and the network capabilities of the agent machine. Three different approaches have been identified:

1. Distributed Approach: the model is deployed on each workstation and a common configuration file is shared between them;
2. Centralized Approach: the model is deployed on a routing-node;
3. Centralized "ad-hoc" Approach: the model is deployed on a single network node that has full view of the network traffic. This approach represents ad hybrid version of the previous ones.

Each of the aforementioned approaches are characterized by several advantages and disadvantages regarding: (i) configurations, (ii) network deployment, and (iii) model transparency with respect to the users. For instance, the first approach is the simplest one. It uses real workstations in order to generate traffic. However, the deployment procedures and the model transparency are not guaranteed. In particular, the model could be affected by unpredictable actions from a Red Team attacker on the workstation itself. The centralized approach guarantees complete model transparency with respect to the cyber range users. Moreover, the routing capability of the node provides complete network visibility. As a final remark, the model can virtually simulate a higher number of sources with respect to the real number of workstations within the cyber range. In this fashion, it is possible to introduce a time-variant number of users. Hence, in this work, the centralized approach is presented.

4.2. Experimental Results

The simulation results regarding both learning and traffic generation will be exposed in the following sections. In Table 3, the general model parameters both for the learning and traffic generator procedures are reported. The reference cyber environment is depicted in Figure 2.

4.2.1. Learning Performance

We first consider the Cumulative Permitted Transaction Rate (CPTR) generated by the Networking RL Agent. With reference to Figure 3, the CPTR is evaluated for each available policy

Table 3
General Model Parameters.

Parameter	Value/Set
Default FW Rule	Block All
Number of Simulations	10
Number of Episodes	50
Maximum Number of Transactions	600
Learning Rate α	0.9
Epsilon ϵ	0.2
Epsilon Decay	True
Temperature Parameter τ	5
Discount Rate η	0.1
Policies	Random, Epsilon-Greedy, Softmax
Normal/Attack Profile	50 %
Traffic Generation Sim Time	$\sim$ 70 min

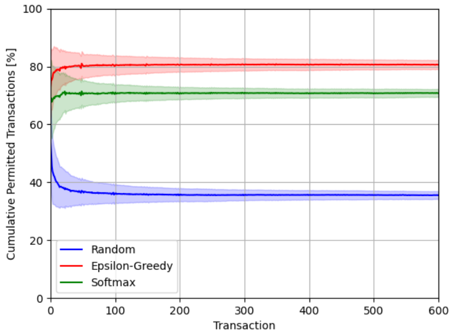
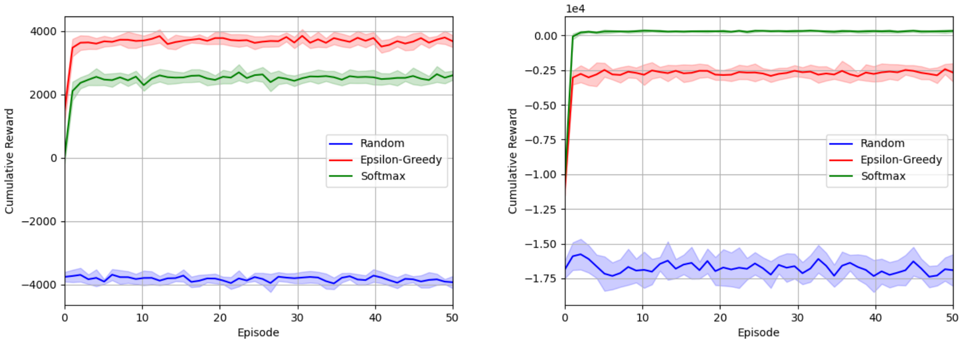


Figure 3: Cumulative Permitted Transaction Rate (CPTR) as a function of transactions and RL policies for the Networking RL Agent.

within the model and the average value over the total simulations and episodes number is reported as a function of the overall transactions. As depicted in the figure, the CPTR achieves an asymptotic value of $\sim$ 38 % in the case of a random policy. On the other hand, the CPTR scored with Epsilon-Greedy and Softmax policies is 70 % and 80 % respectively. For the sake of clarity, each curve depicted in Figure 3 is characterized by a confidence interval evaluated as the standard deviation of the simulation data. In Figure 4a and 4b the cumulative rewards of the Application-Level RL Agent in case of internal requests (a) and external requests (b) are reported. The action space of the Application-level agent is dynamically defined from the available network plugins within the model, as aforementioned. Moreover, the reward function presents a higher degree of granularity in order to take into account both the advantages and the cost of a specific operation (i.e., a DDoS attack is characterized by a higher cost with respect to a port scan). The cumulative rewards depicted in the figures are reported for each available policy and it is evaluated over the total simulations and transactions number. With reference to Figure 4a, the cumulative reward achieved by using the random policy is extremely negative



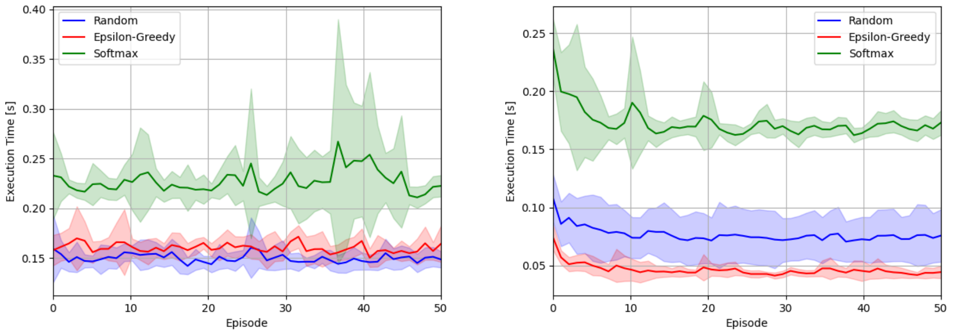
(a)

(b)

Figure 4: Cumulative Reward as a function of episodes and RL policies for the Application-Level RL Agent in case of internal (a) and external (b) requests.

(~ -4000 points). On the other hand, the cumulative reward achieved by using Epsilon-Greedy and Softmax policies are ~ 2100 and ~ 3800 points respectively. Those considerations change for the cumulative rewards reported in Figure 4b. In this case, the policy that achieved a positive cumulative reward is Softmax. On the other hand, random and epsilon-greedy policies do not achieve positive values. Regarding the epsilon-greedy scores, the amount of exploration jointly with the network constraints (i.e., only the FTP and Mail servers are reachable from external requests) tends to worsen the learning outcomes.

4.2.2. Simulation Statistics and Outcomes

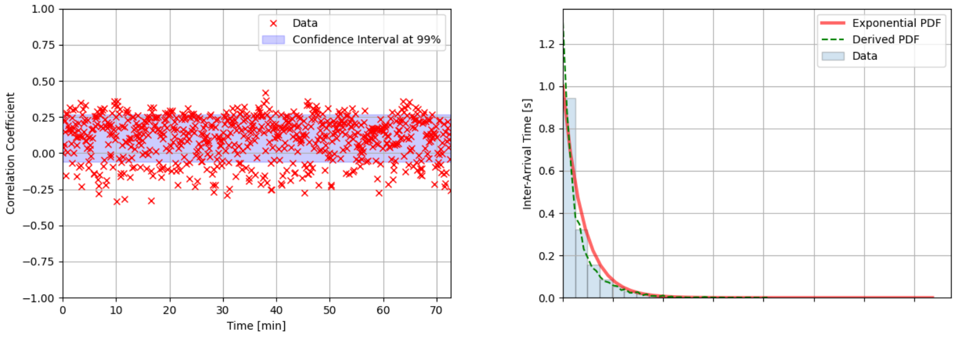


(a)

(b)

Figure 5: Execution Time as a function of episodes and RL policies for the Networking (a) and for the Application-Level (b) RL Agent.

Figures 5a and 5b report the total learning execution time for each available policy as a function of episodes number both for the Network and Application-Level RL Agents. The reported data are evaluated over the total number of simulations. As depicted in the figures, the total learning time for random policy is quite similar with respect to one achieved by epsilon-greedy. Whereas, the total learning time achieved by the softmax policy is higher with respect to the other policies for both the RL agents.



(a)

(b)

Figure 6: Pearson correlation coefficient between the generated transactions as a function of simulation time (a) and transactions inter-arrival time distribution (b).

Figures 6a and 6b show the Pearson correlation coefficient between the transactions and the transactions inter-arrival time distribution during a long-run simulation.

With reference to Figure 6a, the correlation coefficient is repeated as a function of the total simulation time. The results show that the 99 % of the correlation data are located within the confidence interval of -0.1 and 0.25 for an average correlation of 17.5 %. Hence, the generated transactions have a relatively low correlation.

Finally, Figure 6b shows the inter-arrival times distribution. In particular, the simulation data are reported with a histogram representation. The red curve represents an ideal exponential decay and the green curve represents the data-derived distribution. From these results, it is possible to see that the transactions generated by the model are characterized by a lower correlation and by an exponential inter-arrival times distribution. As a consequence, the model follows a Poisson Process.

5. Conclusions

In this paper, we analyzed the problem of users actions simulation in cyber range scenarios exploring the potentiality of Reinforcement Learning. The proposal is strengthened by a statistical characterization of user actions and by adopting RL Agents to learn and to perform actions at different protocol stack layers. Moreover, the designed RL Agents cooperate and perform networking operations by following specific behaviors. In detail, we focused our investigation on the impact of different RL policies for both the RL Agents among the learning capabilities and execution time. In addition, we carried out a long-run simulation to evaluate the correlation between the generated transactions, and, finally, the inter-arrival times distribution. Future works will focus on overcoming the limitation described in 3.4 by involving autonomous setup configurations, the implementation of more sophisticated RL algorithms (i.e., Deep Reinforcement Learning), and, cause-effect relationships between user actions.

Acknowledgments

This work was supported in part by the Fondo Europeo di Sviluppo Regionale Puglia Programma Operativo Regionale (POR) Puglia 2014–2020–Axis I–Specific Objective 1a–Action 1.1 (Research and Development)–Project Title: CyberSecurity and Security Operation Center (SOC) Product Suite by BV TECH S.p.A., under Grant CUP/CIG B93G18000040007.

References

[1] C. Ebert, C. H. C. Duarte, Digital transformation, IEEE Softw. 35 (2018) 16–21.

[2] S. Mendhurwar, R. Mishra, Integration of social and iot technologies: architectural framework for digital transformation and cyber security challenges, Enterprise Information Systems 15 (2021) 565–584.

[3] A. I. per la Sicurezza Informatica, Clusit report 2021, https://clusit.it/rapporto-clusit/, 2021.

[4] E. Gilman, D. Barth, Zero Trust Networks, O'Reilly Media, Incorporated, 2017.

[5] D. Bertsekas, Reinforcement learning and optimal control, Athena Scientific, 2019.

[6] M. Tokic, G. Palm, Value-difference based exploration: adaptive control between epsilon-greedy and softmax, in: Annual conference on artificial intelligence, Springer, 2011, pp. 335–346.

[7] I. N. Labs, "securing the electrical grid from cyber and physical threats, https://inl.gov/research-programs/grid-resilience/, 2021.

[8] C. Queiroz, A. Mahmood, Z. Tari, Scadasim—a framework for building scada simulations, IEEE Transactions on Smart Grid 2 (2011) 589–597.

[9] N. Kaloudi, J. Li, The ai-based cyber threat landscape: A survey, ACM Computing Surveys (CSUR) 53 (2020) 1–34.

[10] Microsoft, Cyberbattlesim, https://www.microsoft.com/en-us/research/project/cyberbattlesim/, 2020.

[11] J. A. Nichols, K. Spakes, C. Watson, R. A. Bridges, Assembling a cyber range to evaluate artificial intelligence / machine learning (ai/ml) security tools (????). URL: https://www.osti.gov/biblio/1772629.

[12] T. Alves, R. Das, A. Werth, T. Morris, Virtualization of scada testbeds for cybersecurity research: A modular approach, Computers & Security 77 (2018) 531–546.

[13] Q. Qassim, N. Jamil, I. Z. Abidin, M. E. Rusli, S. Yussof, R. Ismail, F. Abdullah, N. Ja'afar, H. C. Hasan, M. Daud, A survey of scada testbed implementation approaches, Indian Journal of Science and Technology 10 (2017) 1–8.

[14] T. Morris, R. Vaughn, Y. S. Dandass, A testbed for scada control system cybersecurity research and pedagogy, in: Proceedings of the Seventh Annual Workshop on Cyber Security and Information Intelligence Research, 2011, pp. 1–1.

[15] S. Khan, A. Volpatto, G. Kalra, J. Esteban, T. Pescanoce, S. Caporusso, M. Siegel, Cyber range for industrial control systems (cr-ics) for simulating attack scenarios, Proceedings of the Italian Conference on Cybersecurity (ITASEC 2021) 2940 (2021) 246–259.

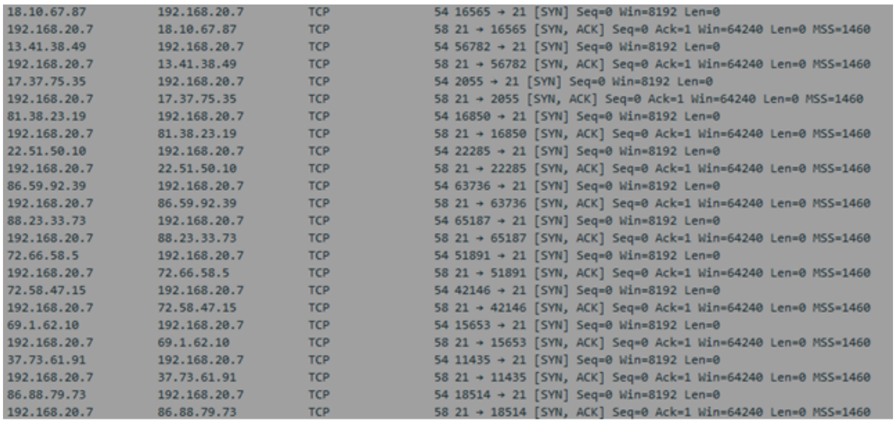
[16] A. Salvi, P. Spagnoletti, N. S. Noori, Cyber-resilience of critical cyber infrastructures: Integrating digital twins in the electric power ecosystem, Computers & Security 112 (2022) 102507.

[17] V. Sundarapandian, Probability, statistics and queuing theory, PHI Learning Pvt. Ltd., 2009.

[18] P. Z. Peebles Jr, Probability, random variables, and random signal principles, McGraw-Hill, 2001.

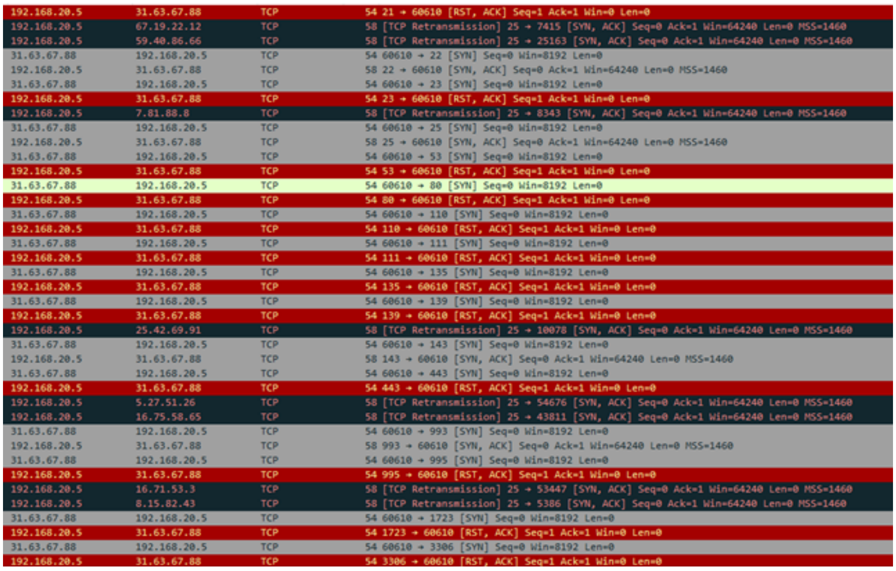
Appendix

Examples of the simulated network traffic.



```
18.10.67.87     192.168.20.7    TCP    54 16565 → 21 [SYN] Seq=0 Win=8192 Len=0
192.168.20.7    18.10.67.87     TCP    58 21 → 16565 [SYN, ACK] Seq=0 Ack=1 Win=64240 Len=0 MSS=1460
13.41.38.49     192.168.20.7    TCP    54 56782 → 21 [SYN] Seq=0 Win=8192 Len=0
192.168.20.7    13.41.38.49     TCP    58 21 → 56782 [SYN, ACK] Seq=0 Ack=1 Win=64240 Len=0 MSS=1460
17.37.75.35     192.168.20.7    TCP    54 2055 → 21 [SYN] Seq=0 Win=8192 Len=0
192.168.20.7    17.37.75.35     TCP    58 21 → 2055 [SYN, ACK] Seq=0 Ack=1 Win=64240 Len=0 MSS=1460
81.38.23.19     192.168.20.7    TCP    54 16850 → 21 [SYN] Seq=0 Win=8192 Len=0
192.168.20.7    81.38.23.19     TCP    58 21 → 16850 [SYN, ACK] Seq=0 Ack=1 Win=64240 Len=0 MSS=1460
22.51.50.10     192.168.20.7    TCP    54 22285 → 21 [SYN] Seq=0 Win=8192 Len=0
192.168.20.7    22.51.50.10     TCP    58 21 → 22285 [SYN, ACK] Seq=0 Ack=1 Win=64240 Len=0 MSS=1460
86.59.92.39     192.168.20.7    TCP    54 63736 → 21 [SYN] Seq=0 Win=8192 Len=0
192.168.20.7    86.59.92.39     TCP    58 21 → 63736 [SYN, ACK] Seq=0 Ack=1 Win=64240 Len=0 MSS=1460
88.23.33.73     192.168.20.7    TCP    54 65187 → 21 [SYN] Seq=0 Win=8192 Len=0
192.168.20.7    88.23.33.73     TCP    58 21 → 65187 [SYN, ACK] Seq=0 Ack=1 Win=64240 Len=0 MSS=1460
72.66.58.5      192.168.20.7    TCP    54 51891 → 21 [SYN] Seq=0 Win=8192 Len=0
192.168.20.7    72.66.58.5      TCP    58 21 → 51891 [SYN, ACK] Seq=0 Ack=1 Win=64240 Len=0 MSS=1460
72.58.47.15     192.168.20.7    TCP    54 42146 → 21 [SYN] Seq=0 Win=8192 Len=0
192.168.20.7    72.58.47.15     TCP    58 21 → 42146 [SYN, ACK] Seq=0 Ack=1 Win=64240 Len=0 MSS=1460
69.1.62.10      192.168.20.7    TCP    54 15653 → 21 [SYN] Seq=0 Win=8192 Len=0
192.168.20.7    69.1.62.10      TCP    58 21 → 15653 [SYN, ACK] Seq=0 Ack=1 Win=64240 Len=0 MSS=1460
37.73.61.91     192.168.20.7    TCP    54 11435 → 21 [SYN] Seq=0 Win=8192 Len=0
192.168.20.7    37.73.61.91     TCP    58 21 → 11435 [SYN, ACK] Seq=0 Ack=1 Win=64240 Len=0 MSS=1460
86.88.79.73     192.168.20.7    TCP    54 18514 → 21 [SYN] Seq=0 Win=8192 Len=0
192.168.20.7    86.88.79.73     TCP    58 21 → 18514 [SYN, ACK] Seq=0 Ack=1 Win=64240 Len=0 MSS=1460
```

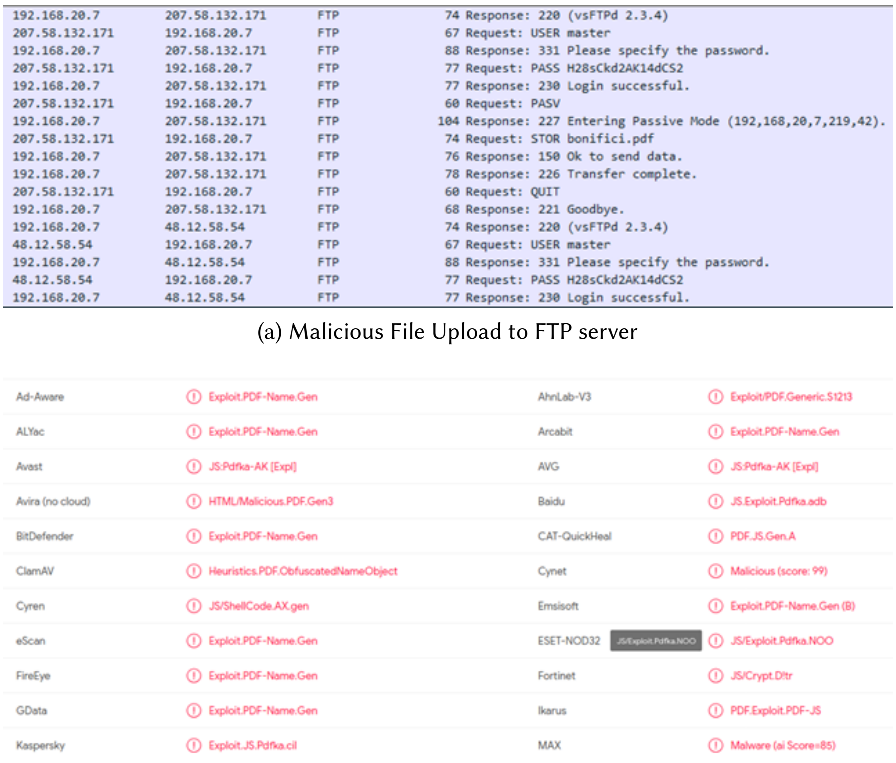
(a) DDoS Simulation



```
192.168.20.5    31.63.67.88     TCP    54 21 → 60610 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
192.168.20.5    67.19.22.12     TCP    58 [TCP Retransmission] 25 → 7415 [SYN, ACK] Seq=0 Ack=1 Win=64240 Len=0 MSS=1460
192.168.20.5    59.40.86.66     TCP    58 [TCP Retransmission] 25 → 25163 [SYN, ACK] Seq=0 Ack=1 Win=64240 Len=0 MSS=1460
31.63.67.88     192.168.20.5    TCP    54 60610 → 22 [SYN] Seq=0 Win=8192 Len=0
192.168.20.5    31.63.67.88     TCP    58 22 → 60610 [SYN, ACK] Seq=0 Ack=1 Win=64240 Len=0 MSS=1460
31.63.67.88     192.168.20.5    TCP    54 60610 → 23 [SYN] Seq=0 Win=8192 Len=0
192.168.20.5    31.63.67.88     TCP    54 23 → 60610 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
192.168.20.5    7.81.88.8       TCP    58 [TCP Retransmission] 25 → 8343 [SYN, ACK] Seq=0 Ack=1 Win=64240 Len=0 MSS=1460
31.63.67.88     192.168.20.5    TCP    54 60610 → 25 [SYN] Seq=0 Win=8192 Len=0
192.168.20.5    31.63.67.88     TCP    58 25 → 60610 [SYN, ACK] Seq=0 Ack=1 Win=64240 Len=0 MSS=1460
31.63.67.88     192.168.20.5    TCP    54 60610 → 53 [SYN] Seq=0 Win=8192 Len=0
192.168.20.5    31.63.67.88     TCP    54 53 → 60610 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
31.63.67.88     192.168.20.5    TCP    54 60610 → 80 [SYN] Seq=0 Win=8192 Len=0
192.168.20.5    31.63.67.88     TCP    54 80 → 60610 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
31.63.67.88     192.168.20.5    TCP    54 60610 → 110 [SYN] Seq=0 Win=8192 Len=0
192.168.20.5    31.63.67.88     TCP    54 110 → 60610 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
31.63.67.88     192.168.20.5    TCP    54 60610 → 111 [SYN] Seq=0 Win=8192 Len=0
192.168.20.5    31.63.67.88     TCP    54 111 → 60610 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
31.63.67.88     192.168.20.5    TCP    54 60610 → 135 [SYN] Seq=0 Win=8192 Len=0
192.168.20.5    31.63.67.88     TCP    54 135 → 60610 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
31.63.67.88     192.168.20.5    TCP    54 60610 → 139 [SYN] Seq=0 Win=8192 Len=0
192.168.20.5    31.63.67.88     TCP    54 139 → 60610 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
192.168.20.5    25.42.69.91     TCP    58 [TCP Retransmission] 25 → 10078 [SYN, ACK] Seq=0 Ack=1 Win=64240 Len=0 MSS=1460
31.63.67.88     192.168.20.5    TCP    54 60610 → 143 [SYN] Seq=0 Win=8192 Len=0
192.168.20.5    31.63.67.88     TCP    58 143 → 60610 [SYN, ACK] Seq=0 Ack=1 Win=64240 Len=0 MSS=1460
31.63.67.88     192.168.20.5    TCP    54 60610 → 443 [SYN] Seq=0 Win=8192 Len=0
192.168.20.5    31.63.67.88     TCP    54 443 → 60610 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
192.168.20.5    5.27.51.26      TCP    58 [TCP Retransmission] 25 → 54676 [SYN, ACK] Seq=0 Ack=1 Win=64240 Len=0 MSS=1460
192.168.20.5    16.75.58.65     TCP    58 [TCP Retransmission] 25 → 43811 [SYN, ACK] Seq=0 Ack=1 Win=64240 Len=0 MSS=1460
31.63.67.88     192.168.20.5    TCP    54 60610 → 993 [SYN] Seq=0 Win=8192 Len=0
192.168.20.5    31.63.67.88     TCP    58 993 → 60610 [SYN, ACK] Seq=0 Ack=1 Win=64240 Len=0 MSS=1460
31.63.67.88     192.168.20.5    TCP    54 60610 → 995 [SYN] Seq=0 Win=8192 Len=0
192.168.20.5    31.63.67.88     TCP    54 995 → 60610 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
192.168.20.5    16.71.53.3      TCP    58 [TCP Retransmission] 25 → 53447 [SYN, ACK] Seq=0 Ack=1 Win=64240 Len=0 MSS=1460
192.168.20.5    8.15.82.43      TCP    58 [TCP Retransmission] 25 → 5386 [SYN, ACK] Seq=0 Ack=1 Win=64240 Len=0 MSS=1460
31.63.67.88     192.168.20.5    TCP    54 60610 → 1723 [SYN] Seq=0 Win=8192 Len=0
192.168.20.5    31.63.67.88     TCP    54 1723 → 60610 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
31.63.67.88     192.168.20.5    TCP    54 60610 → 3306 [SYN] Seq=0 Win=8192 Len=0
192.168.20.5    31.63.67.88     TCP    54 3306 → 60610 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
```

(b) Port Scan Traffic Simulation

Figure 7: Appendix Figures I



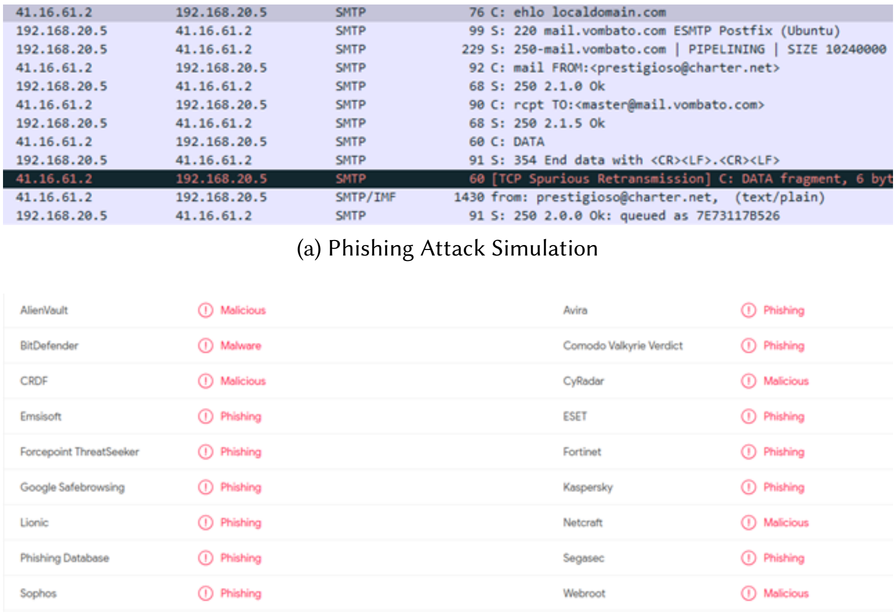
```
192.168.20.7        207.58.132.171      FTP              74 Response: 220 (vsFTPd 2.3.4)
207.58.132.171      192.168.20.7        FTP              67 Request: USER master
192.168.20.7        207.58.132.171      FTP              88 Response: 331 Please specify the password.
207.58.132.171      192.168.20.7        FTP              77 Request: PASS H28sCkd2AK14dCS2
192.168.20.7        207.58.132.171      FTP              77 Response: 230 Login successful.
207.58.132.171      192.168.20.7        FTP              60 Request: PASV
192.168.20.7        207.58.132.171      FTP             104 Response: 227 Entering Passive Mode (192,168,20,7,219,42).
207.58.132.171      192.168.20.7        FTP              74 Request: STOR bonifici.pdf
192.168.20.7        207.58.132.171      FTP              76 Response: 150 Ok to send data.
192.168.20.7        207.58.132.171      FTP              78 Response: 226 Transfer complete.
207.58.132.171      192.168.20.7        FTP              60 Request: QUIT
192.168.20.7        207.58.132.171      FTP              68 Response: 221 Goodbye.
192.168.20.7        48.12.58.54         FTP              74 Response: 220 (vsFTPd 2.3.4)
48.12.58.54         192.168.20.7        FTP              67 Request: USER master
192.168.20.7        48.12.58.54         FTP              88 Response: 331 Please specify the password.
48.12.58.54         192.168.20.7        FTP              77 Request: PASS H28sCkd2AK14dCS2
192.168.20.7        48.12.58.54         FTP              77 Response: 230 Login successful.
```

(a) Malicious File Upload to FTP server

Ad-Aware	ⓘ Exploit.PDF-Name.Gen	AhnLab-V3	ⓘ Exploit/PDF.Generic.S1213
ALYac	ⓘ Exploit.PDF-Name.Gen	Arcabit	ⓘ Exploit.PDF-Name.Gen
Avast	ⓘ JS:Pdfka-AK [Expl]	AVG	ⓘ JS:Pdfka-AK [Expl]
Avira (no cloud)	ⓘ HTML/Malicious.PDF.Gen3	Baidu	ⓘ JS.Exploit.Pdfka.adb
BitDefender	ⓘ Exploit.PDF-Name.Gen	CAT-QuickHeal	ⓘ PDF.JS.Gen.A
ClamAV	ⓘ Heuristics.PDF.ObfuscatedNameObject	Cynet	ⓘ Malicious (score: 99)
Cyren	ⓘ JS/ShellCode.AX.gen	Emsisoft	ⓘ Exploit.PDF-Name.Gen (B)
eScan	ⓘ Exploit.PDF-Name.Gen	ESET-NOD32 JS/Exploit.Pdfka.NOO	ⓘ JS/Exploit.Pdfka.NOO
FireEye	ⓘ Exploit.PDF-Name.Gen	Fortinet	ⓘ JS/Crypt.D!tr
GData	ⓘ Exploit.PDF-Name.Gen	Ikarus	ⓘ PDF.Exploit.PDF-JS
Kaspersky	ⓘ Exploit.JS.Pdfka.cil	MAX	ⓘ Malware (ai Score=85)

(b) Malicious File Specs

Figure 8: Appendix Figures II



```
41.16.61.2          192.168.20.5        SMTP             76 C: ehlo localdomain.com
192.168.20.5        41.16.61.2          SMTP             99 S: 220 mail.vombato.com ESMTP Postfix (Ubuntu)
192.168.20.5        41.16.61.2          SMTP            229 S: 250-mail.vombato.com | PIPELINING | SIZE 10240000
41.16.61.2          192.168.20.5        SMTP             92 C: mail FROM:<prestigioso@charter.net>
192.168.20.5        41.16.61.2          SMTP             68 S: 250 2.1.0 Ok
41.16.61.2          192.168.20.5        SMTP             90 C: rcpt TO:<master@mail.vombato.com>
192.168.20.5        41.16.61.2          SMTP             68 S: 250 2.1.5 Ok
41.16.61.2          192.168.20.5        SMTP             60 C: DATA
192.168.20.5        41.16.61.2          SMTP             91 S: 354 End data with <CR><LF>.<CR><LF>
41.16.61.2          192.168.20.5        SMTP             60 [TCP Spurious Retransmission] C: DATA fragment, 6 byt
41.16.61.2          192.168.20.5        SMTP/IMF       1430 from: prestigioso@charter.net,  (text/plain)
192.168.20.5        41.16.61.2          SMTP             91 S: 250 2.0.0 Ok: queued as 7E73117B526
```

(a) Phishing Attack Simulation

AlienVault	ⓘ Malicious	Avira	ⓘ Phishing
BitDefender	ⓘ Malware	Comodo Valkyrie Verdict	ⓘ Phishing
CRDF	ⓘ Malicious	CyRadar	ⓘ Malicious
Emsisoft	ⓘ Phishing	ESET	ⓘ Phishing
Forcepoint ThreatSeeker	ⓘ Phishing	Fortinet	ⓘ Phishing
Google Safebrowsing	ⓘ Phishing	Kaspersky	ⓘ Phishing
Lionic	ⓘ Phishing	Netcraft	ⓘ Malicious
Phishing Database	ⓘ Phishing	Segasec	ⓘ Phishing
Sophos	ⓘ Phishing	Webroot	ⓘ Malicious

(b) Phishing Email Specs

Figure 9: Appendix Figures III