An Intelligent Agent for Computer Security and Forensic Training

Davi Teles*, André dos Santos† Marcial P Fernandez‡
Universidade Estadual do Ceará
Fortaleza, Ceará, Brazil
Email: *davi@insert.uece.br, †andre@insert.uece.br, ‡marcial.fernandez@uece.br

Abstract—**The information security and digital forensic training practice is a challenging task because it requires a controlled environment, operating system files and network elements, and is prone to corrupted files. The forensic professional needs to collect and analyze many system logs and historical data to provide a correct identification to unknown attacks. Thus, Shellter, a social network dedicated to teaching and practice of information security and digital forensic training is aimed to help educators, tutors, students and enthusiasts in this area. Shellter offers challenges, educational tracks, materials for studies, information sharing forums and simulations to provide a full arena for hands-on learning. In the simulated environment, one difficulty is motivating users to exploit their knowledge, to learn new things, and to reach the goals presented by the system. This work presents a multi-agent system to provide a realistic training environment, motivating students to learn information security and forensics. The proposed agent was evaluated in the Shellter environment and shows an improvement on creating new tasks to motivate the student.**

Keywords*–*Information Security and Forensic; Intelligence Agent; Security and Forensic Training; Simulation

I. Introduction

Human resources training and development in information security and digital forensic requires hands-on training, not just theoretical learning [1]. To practice and perform exercises without broken production systems, a simulated environment is necessary. The training system should be able to create realistic situations, but it should not impact the real world. By attacking your own network or a production network, you may damage the infrastructure. Therefore, it is more appropriate to learn and practice cyber security in an isolated computing environment specially created for this purpose.

Digital Forensics can be defined as the use of scientific methods for the collection, validation, identification, analysis and interpretation of digital evidence for reconstruction events found to be criminal or not, and the identification of unauthorized actions. In the traditional digital forensic training, the instructor needs to present evidence and request the student to identify the attack and discover the authors. This methodology permits discovery of only well-known attacks, deeming it unsuitable for the real world.

Simulation-based games are widely used in training professionals. The simulations start with a well-defined scenario developed during a match. Teams representing different countries begin to attack and react to the situations proposed. Gamification is a motivational technique derived from games. This helps to maintain students interest, thereby increasing their learning. It uses the game dynamic mechanics, as the reward and rank actions taken by a particular participant (computer or not) in a game [2]. It also reinforces the need to use the correct level of difficulty techniques for a specific student knowledge level. Motivating students to learn is an old challenge of educational professionals. Students feel motivated to participate in learning activities if they believe that, with their knowledge, talents, and skills, they can acquire new knowledge, master content, and improve their skills, etc [3].

Learning motivation is always under discussion within the school, pushing students to go further or driving them to go back, even withdrawal in more complex cases. It has a very important role in both, the instructors and the students results. In virtual-learning environments, i.e., non-classroom learning environments, motivating students to study becomes a great challenge because the instructor cannot identify the feelings expressed by the students, for example, their facial expressions or personal conversations [4].

The Shellter system, presented in Section III-C, is a social network for security information training. Shellter offers complete computer systems to solve various challenges in information security, as well as the simulation environment to reproduce raining in actual situations. One of the great challenges to improve the Shellter tool is to create brand new unknown attacks to improve student practice in information security and digital forensic. Another challenge is to motivate students to learn, to test their limits, seek new knowledge and overcome unknown challenges. So, mechanisms to increase learning and motivation in the Shellter system inspired this work.

This paper proposes an intelligent agent system to monitor learning in virtual environments and motivates students using gamification techniques. This system consists of five different agents that work together. This system will analyze student profile, student interests in social networks, success or failure in past challenges and attitudes towards difficulties, to define the techniques applied by the system. Thus, a definition and its requirements of the system will be made, and also, the architecture and interactions between agents. The system validation will be done by the implementation and testing of a prototype of one agent part of the system due to its similarity with the other agents. The choice of the prototype took into account the time available for this work.

ISBN: 978-1-61208-546-3

This work is organized as follows: in Section II, we present some related work, while the basis of agents and gamification is shown in Section III. In Section IV, we present the proposed agent architecture. Sections V and VI show the prototype, experimental evaluation and the results. Section VIII concludes the paper and presents some intended future work.

II. Related Work

The Tele-Lab project is a hands-on system to practice and train for information security [5]. It offers a virtual environment based on Web accessibility for any place. The system consists of text and video tutorials and practice exercises in a virtual environment in a pool of virtual machines. Students practice the information security exercises by accessing Secure Shell (SSH). For motivation, students are invited to assume the attacker's perspective.

SOFTICE is a proposal which focuses on teaching operating system with hands-on exercises [6]. In particular, its goal is aimed at learning Linux's kernel vulnerabilities focusing on its functions, definitions and implementations. In SOFTICE, students can test their knowledge and implement new Linux's kernel modules in a controlled environment. SOFTICE works by the hypothesis which Linux's kernel code is huge, complex and can make students lose motivation.

Insight is a simulation framework to create and imitate cyber attacks [7]. The attacks are part of scenarios available within the framework. Each scenario has different actors, e.g., network devices, software, network protocols and user. The goal is to simulate attacks from the attacker. With a customized interface, the students can create their own attacks. The attacks are executed inside the Insight framework to guarantee isolation and transparency of the simulated environment. A probabilistic model gives support to decide whether an attack has been successful, based on a combination of virtual machine configurations and attack techniques.

CTF365 is a security training platform for the IT industry with a focus on security professionals, system administrators and Web developers [8]. It provides a real life cyber range where users build their own servers and defend them while attacking other servers [8]. The platform implements Capture The Flag (CTF) concepts and leverages gamification mechanics to improve retention rate and speed up the learning/training curve [8].

III. Background

Given the literature review, in this section, the concepts related to, and influencing the design of the proposed architecture are presented.

A. *Inteligent Agent*

An Intelligent Agent (IA) is a piece of software that exists in an environment, is not controlled externally, responds (in a timely manner) to changes in its environment, persistently pursues goals, has multiple ways of achieving goals, recovers from failure and interacts with other agents [9].

B. *Gamification*

Gamification is the capacity to derive in a thoughtful way, the mechanism, fun and addiction of games for other contexts with no relationship with games to motivate people to accomplish results. This brings focus to humans, considering that they don't always feel motivated to accomplish their tasks and a lot of time they need something to become motivated. Gamification is a technique to apply game-design elements and game principles in learning contexts to improve student engagement. It explores the human instinctive natural behaviors to accomplish their goals [3].

C. *Shellter*

Shellter [10] is a social network dedicated to information security learning. Shellter is idealized, developed and maintained by the Information Security Research Team - Insert, a researching group from Universidade Estadual do Ceará - Brazil. When using Shellter, users can interact in the same way online game users do: building teams or playing solo in challenges so that, in time, they could evolve from novice hackers to pro hackers. In a cloud computing security environment, lab and virtual simulations happen with different types of challenges for distinct types of abilities. With gamification techniques, Shellter builds a space for security information continuing education. Users will test their abilities and can learn new techniques in Shellter's Cyber Warfare environment, providing a real experience in a computer network.

Through virtualization techniques, it will be possible to simulate different scenarios of attacks, defense or attack/defense. The goal is to create an actual hands-on environment for learning. In this simulation environment, users can play alone, against other users or against the intelligent agent system, proposed in this work.

IV. Intelligent Agent for Network Security and Forensic Training

To maintain users' motivation for learning, it is necessary to overcome student limits. Motivating users to seek new acknowledgments and experiences is the goals of Intelligent Agent for Network Security (IANS). The system will monitor users' performance and evolution and will interact with them to encourage continuous study and practice of information security. In addition, it will push them to overcome limits and knowledge. In particular, IANS will classify users according to their knowledge level and their experience with information security. With this classification, it will be possible the choose the most appropriate IA for users profiles.

These two types of IAs discourage the user because it makes the game more difficult rather than easier. The aim is to help users evolve gradually and, in time, they can face more difficult challenges and scenarios. After choosing the correct IAs to play, a second phase begins, namely, the evolution of IAs in the game time. The IAs need to follow users evaluation, because users will be constantly challenged to extend their knowledge. In this second phase, the IAs will use artificial intelligence techniques to learn, in real time, and can accomplish two main goals: (1) evolve according to the user, and (2) evolve according to the environment.

To accomplish these goals, in this work we use the following methodology: (1) Define the IANS architecture, specifying each one of its components and interactions in the system; and, (2) Define, implement, test and validate one of the IAs of IANS. We select the Environment Change Agent (ECA), due to its similar architecture to the others IAs.

A. Proposed Architecture

Figure 1 shows the IANS's architecture of the model based reflex agent that synthesizes the ideas of Russell & Norvig's, related to a reactive agent program [11], as well as the abstract architecture point of view proposed by Wooldridge in [12].

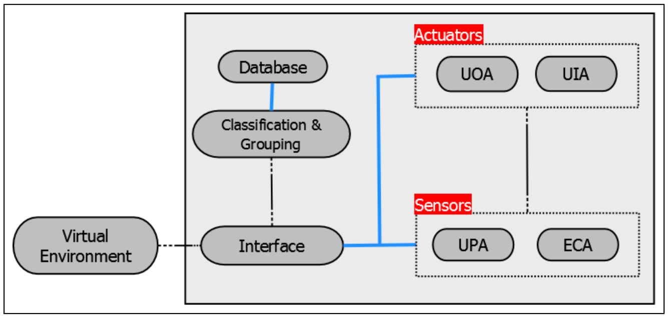

Fig. 1. Intelligent Agent for Network Security Architecture

1) Classification and Grouping - C&G: C&G engine is responsible for classifying users according to their knowledge and experience levels. This classification will be used to choose the appropriate IAs to play and interact with. Before accessing the simulated environment, the users will be asked to allow C&G analyze their social networks data set. With this, we will be able to capture users' interests, experiences and knowledge Some social networks considered are:

- **Facebook:** Users' activities (sharing posts, liked pages, community debates, etc.) will be analyzed to get interests about information security;
- **Github:** Punctuate the users for creation and participation in open source projects;
- **Stackoverflow:** It offers a space to ask about computer science and it sources relevance of issues and answers.
- **Tocoder:** Offers different type of programming challenges;
- **Shellter:** Shellter's profile offers a data set about learning and performance in Shelters environment and challenges, showing the users' strongest and weakest abilitys in information security.

2) User Progress Agent (UPA): In the group of sensor IAs, UPA monitors users' activities. The UPA's actions aim to identify affection and emotional experience that influence the learning process [4]. This IA will be used to compute affective techniques. Lester et. al [4] show some techniques to model emotions to measure the level of engagement and motivation. To capture these emotions, we will use the following plugins:

- **Keyboard plugin**: Will capture all user keyboard entries. These entries represent the user interaction frequency with the environment, access to challenges, number of answers, response time, etc. Keyloggers will be used to capture this information;
- **Video plugin**: It captures facial expressions, gestures, posture and any other body expressions made by the user. Then, it will be possible to identify the user's feelings and emotions [4].
- **Audio plugin**: This plugin will monitor users' sounds during the simulation: sounds emitted and heard. Music favors reasoning, evokes feelings and can change moods, reaching the cognitive and affective dimensions of the human being.
- **Content plugin**: It will be responsible for analyzing what the user is accessing during the simulation: web pages, open study material, etc. The goal is to identify whether users are focusing on solving challenges in a simulated environment.

3) Environment Change Agent (ECA): The ECA is responsible for monitoring and identifying changes in the computer environment for users and IAs which interact with users. ECA will be implemented in this work. Changes in computer environment are necessary to accomplish the most unique information security techniques that need to change files, open/close ports, change configurations, etc. ECA will connect these changes with information security techniques.

To identify these techniques, ECA will use a classification taxonomy, defined in Section V-A1, and a technique catalog of known information security techniques, defined in Section V-A2. To monitor and capture data in computers' environments, ECA will use keyloggers, for users' inputs, and plugins, for use and modifications in computer elements: virtual machines, services, applications, directories, files, virtual networks, switches, routers, firewalls, configurations, etc. It will also consider the access to resources like open and read files.

4) User Opponent Agent (UOA): Starting actuators IAs, UOA will play through attacks and defense actions based on information collected and processed for sensors IAs. This information allows UOA to formulate strategies to play against users and teams:

- What techniques are used to attack/defend in terms of user's knowledge?
- What techniques are used to attack/defend in terms of user's evolution?
- What techniques are used to attack/defend in terms of the modifications in computer environment?

Executing its actions, UOA will use plugins to act in virtual environment and verify the consequences of these actions on the environment. UOA will be analogous to a user, in the sense that it can execute any actions the user can in environment. For example, it can execute shell commands, create and execute scripts, click on icons, etc.

5) User Interaction Agent (UIA): The UIA is also an actuator IA which interacts with users based on the collected information by sensor IAs. However, its purpose is different from UOA. The UIA's goal is for interacting with users to motivate and encourage them to expand their knowledge. It acts like a user's tutor, following their activities to pursue their tasks' accomplishment, always keeping them motivated, even with difficult to complete tasks.

B. Agents Interaction

The use of IAs to improve users' motivation by monitoring simulated environment is a good approach because it can detect changes in environment, acting proactively to execute tasks to reduce negative effects. To achieve this, the IAs have the function to perceive their environment and interact with users and computer environment to maintain the users' motivation. Figure 2 shows the relationship among IAs, users and computational environment. The information exchange among sensors IAs, actuators IAs and monitored environment is a constant activity to achieve the goals of IANS. Figure 3 shows the interaction among IANS's IAs.

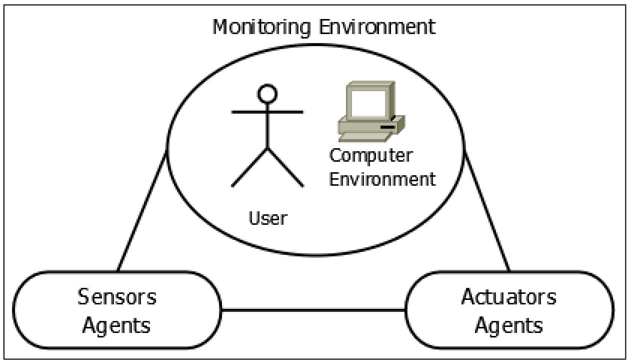

Fig. 2. Relationship among agents vs computer environment vs user

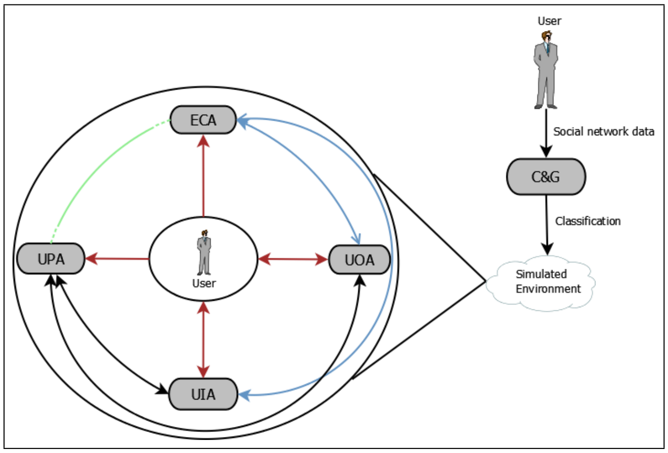

Fig. 3. Interaction among IANS's IAs

Before users gain access to the virtual environment, they need to provide their social network data set to the Classification and Grouping (C&A) engine in order to rank users. The C&A classifies the user and informs IANS's IAs to permit them to gain access to the virtual environment. In the environment, the users will face the most unique security information challenges while IA's sensors monitor their activities. The UPA monitors user's emotion and facial expression to measure motivation and engagement. Meanwhile, ECA monitors the environment modifications made by users to allow IAs to identify appropriate information security techniques. After collecting, processing and analyzing the information, the IAs sensor gives a command to IAs actuators. The UOA plays against the user, applying information security techniques based on data set taken from IAs sensors. At the same time, the UIA uses these data sets to seek the best way to motivate users and apply it, providing study materials, teaching techniques, etc.

V. Prototype Evaluation

In order to evaluate the proposed IA, it was developed as a prototype of Environment Change Agent (ECA). The ECA is the most important agent in the architecture. This choice is based on the agent importance for the system and for its influence on other agents. Moreover, ECA's architecture is similar to the others three agents's architecture. C&G use ECA's data to classify users, because the ECA classifies user's abilities and experiences, according to information security techniques applied in virtual environment. The UOA uses this data to analyze, plan and execute attack and defense techniques. The UIA tutoring users with ECA and UOA's data. Finally, UPA is indirectly influenced by ECA, because it depends on user classification and challenge levels, suggested in virtual in environment.

A. Environment Change Agent (ECA) Implementation

To implement ECA, first, it will be necessary to define an information security technique taxonomy and cataloging. Thereafter, we will define the ECA's agent program, the logic formalization, the test and the validations.

1) Security Technique Taxonomy: The proposed taxonomy is shown in Figure 4. Initially, the technique is classified in ***attack*** or ***defense***. Then, the technique is classified according to the information security area: ***Networking***, ***Operating System - OS***, ***Programming*** and ***Database- DB***. Finally, the technique is classified according the difficult level: ***easy***, ***medium*** and ***hard***.

2) Security Technique Cataloging: For cataloging techniques, it is necessary to set an unique identifier for each one. This identifier is formed by a combination of classification criteria and a counter. The first technique, cataloged ***Attack - Network - Hard***, will have the identifier ***ANH1***. Moreover, each identifier will be associated with two information security databases: *CVE* and *Exploit-DB*. The Common Vulnerabilities and Exposures (CVE) is a public dictionary about information security vulnerabilities and exposures, since 2000. Exploit-DB is an exploit repository, i.e., a piece of code which tries to compromise a computer system, created and maintained by Offensive Security [13].

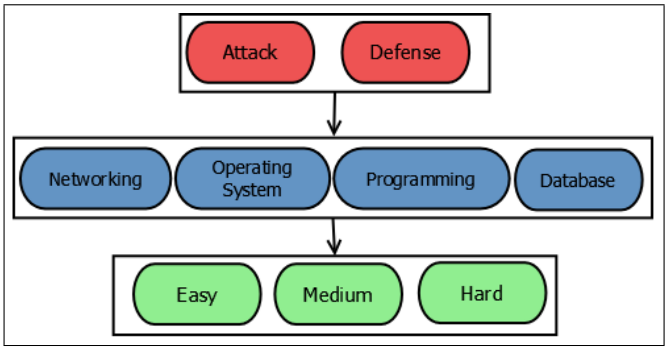

Fig. 4. Taxonomy of information security techniques

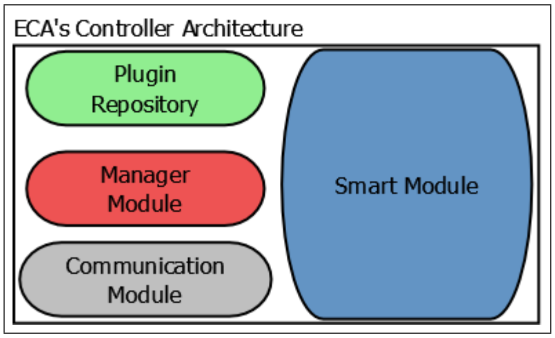

Fig. 5. Intelligent Agent for Network Security Architecture

3) Agent Implementation: The ECA has centralized software architecture for the best plugin's management, in order to provide scalability and resiliency. Another goal is to optimize the number of resources in virtual machines. In the ECA's architecture, it is possible to identify two modules: (1) the controller; and, (2) the plugin. The Controller will implement the ECA's rationality. It will receive the data from plugins, and the decision will be made based on its goals and it will be applied to an actuation (classification). Moreover, it will manage and distribute plugins to the computer environment according to demand. The control of plugins will happen through a table that will be handled with an IP address, operating system and version of computer environment and depending on if the plugins are active in this environment or not. For each unknown environment, a new line will be added to this table and, if an environment is deleted, the line will be marked as deactivated.

The controller, shown in Figure 5, has three modules responsible for agent operation:

- **Plugin Repository**: It is a repository for different types of plugins for distinct types of computer environments. Each environment, depending on platform, architecture and operating system version will have the appropriate plugin for it. Each plugin can be used for one or more environments;
- **Manager Module**: It has the responsibility of managing all plugins in all environments. It controls the entry, exclusion, activation, deactivation and sensor configuration. This module will have information about classification of information security techniques.
- **Communication Module**: It controls the communication between plugins and controller. Receive collected data, send new sensors, send commands and monitor the activity of them;
- **Smart Module**: It is responsible for interpreting and processing data sent from plugins and making a decision about it. The agent modeling will be in this module.

The controller sends a suitable plugin for the target environment to monitor the used information security techniques from the users on it. For this, initially, the Manager Module adds an entry in the control table containing IP address, operating system and OS version from the environment and chooses the appropriate plugin from Plugin Repository. The chosen plugin is sent into the computer environment by Communication Module through File Transfer Protocol (FTP) or Secure Copy Protocol (SCP) and starts the environment monitoring. The plugin has a local database with a set of information security techniques that will be monitored. This database is managed by the Manager Plugin.

The plugin sends the collected data to the Communication Module with their perceptions through REST requests. The received perceptions are passed to the Smart Module where they will be analyzed by ECA's formalization program (Section V-C). If the user received a positive classification for a technique, the plugin do not need to continuous monitoring if the this technique will be applied again. So, after the user is classified based on the information security techniques applied in the environment, the Manager Module checks what techniques were identified. It sends a command to plugin, through the Communication Module, to plugin stop monitoring the identified techniques. This action aims to minimize resources used in the computer environment.

B. Security Treads

The ECA's program will capture the user's commands in simulated environment and will identify and classify information security techniques based on its catalog. The technique shown is based on an Exploit-DB and CVE vulnerability.

1) Security Technique - SSH Root Access: This vulnerability is the capacity to access Linux environment as a root user, based on Debian, through SSH protocol. It is considered a vulnerability because with SSH root access allowed, an attacker can focus on a broken root password with social engineering or brute force techniques. Since root is a default super user in Linux distributions, it is highly recommended to disable remote access to avoid this type of attack. So, an attacker already has the information about a valid user with super powers in the system.

To classify it, ECAs will monitor, through its plugins, two types of user movements:

- **Attack**: ECA will verify if the user applies the following command: ***ssh root@TARGET_IP*** or ***ssh root@TARGET_IP « $password_dictionary***;

- **Defense**: in archive ***/etc/ssh/sshd_config***, ECA will check if parameter ***PermitRootLogin*** is set for ***YES*** or ***NO***. Parameter set to YES allows SSH root access and set to NO disables SSH root access.

This technique will be classified and cataloged as ***Attack, OS, Easy - AOE1*** or ***Defense, OS, Easy - DOE1***, depending on the context.

C. Agent Formalization

Figure 6 shows ECA's formalization algorithm.

```
Algorithm 1: ECA Formalization Algorithm
Data: Perception list - Data collected by plugins
      Expected states list - Expected perception for user
      classification
Begin;
for Perception list do
    for Expected states list do
        if Perception Code X == Expected Perception Code X then
            if Perception != Expected State then
                classified user like FALSE for analyzed
                  technique;
            else
                classified user like TRUE for analyzed technique;
            end
        end
    end
end
return User technique classification list;
End;
```

Fig. 6. ECA Formalization Algorithm

ECA receives the following inputs: (1) Perception list, a list of user's applied techniques received from plugins; and, (2) Expected state list, a list with expected state for user classification. The algorithm returns the user classification technique list, i.e, the user's classification related to the evaluated technique.

D. Prototype Validation

In this section, we will present the tests performed to validate the ECA's program. In other words, we will test the capability of the ECA to classify information security techniques cataloged in its database. For this, we will use the technique defined in Sub-subsection V-B1. ECA's validation tests done using this formalization. Therefore, we created lists to simulate the perceptions of ECA. The chosen perceptions were related to the technique defined in V-B.

In the tests, we tried to create a dataset of different combinations of inputs for the ECA because it would act in a different information security training environment and will find various circumstances, like configurations and users.

Our test scenario was based on *Linux Mint 17.3 64 Bits Debian* and for codification we chose *Python 3.4* and *Prolog*

VI. Results

Table I shows the IA evaluation results considering the expected classification and perception obtained by IA. The first column shows the simulated perceptions; the second column shows expected state for classification, and the third column indicates the results from agent's action. The table structure was made to compare the perception that was used for input of ECA prototype with the expected result of the classification, as this is a simulation, and the result of the ECA's program for classifying the input perception.

The results of the tests show that ECA classified all perceptions correctly, as shown in the *Result* column. In the input list perception, the first information is considered a technique identification. In this identification, the agent's program can find what is the correct state to be compared to the expected state list. Next, the data considers what is in brackets, *[ ]*. These refer to plugins perceptions in environment (simulated in this case).

To detect if the classification needs the combination of one or more parameters and commands, ECA's program searches for *&&* and || separators. They refer to logical operators considered by ECA. The operator *&&* indicates the logical ***AND*** combination, and so, the classification needs combination of one ***AND*** more parameters. The operator || indicates logical ***OR*** combination, and the classification needs one parameter ***OR*** more parameters.

Therefore, different input combinations were tested. For example, AOE1 was tested with distinct methods. Initially, we tested the simple brute force with a single password, then, we executed a more complex brute force test using a combination of multiple password dictionaries.

VII. Forensic Training

For computer forensics, it is important to understand about different areas. In addition to computer science, a forensic examiner needs to know about the local law, best practices, crime scene rules, police procedures, court rules, question from lawyers, etc. The responsibility of a computer forensic examiner goes beyond the limits of computer science.

Therefore, a student can be surprised and unmotivated by all these rules, because he/she is expecting to study and learn forensic computer techniques by learning about the file system, files structures, cryptography, algorithms, operating system, etc.

IANS can be used in all phases of a computer forensic training: preparation of the environment, data collecting, data duplication, data processing, data analysis, data carving, technical explanation, reporting, etc. The system can help students to stay motivated to learn all phases and procedures of a forensic investigation, face new challenges and learn new skills. Furthermore, it can be applied in many other areas, not only in information security knowledge, but also to meet the requirements to form a computer forensic examiner. In addition, ECA will work to identify and classify cataloged forensic techniques applied in a simulated environment.

For example, an advanced student of computer forensics will solve a simulated real-life scenario, where he must follow a specific procedure that will not contaminate evidence , understand the legal aspects and will not let lawyers contest his/her's report. So, IANS will monitor all his/her's performance, looking to monitor, classify, motivate and tutor him/her

TABLE I. INTELLIGENT AGENT EVALUATION RESULTS.

Perception	*Expected Classification*	*Result*
AOE1 - [ssh root@192.168.3.78 && ssh root@192.168.3.78 « $password_dictionary]	AOE1 - TRUE	AOE1 - TRUE
AOE1 - [ssh root@192.168.3.78 « $password_dictionary]	AOE1 - TRUE	AOE1 - TRUE
AOE1 - [ssh root@192.168.3.78]	AOE1 - TRUE	AOE1 - TRUE
AOE1 - [ ]	AOE1 - FALSE	AOE1 - FALSE
DOE1 - [PermitRootLogin YES]	DOE1 - FALSE	DOE1 - FALSE
DOE1 - [PermitRootLogin NO]	DOE1 - TRUE	DOE1 - TRUE
AOE1 - [ssh root@192.168.3.78 && ssh root@192.168.3.78 « $password_dictionary]	AOE1 - TRUE	AOE1 - TRUE
AOE1 - [ssh root@192.168.3.78 « $password_dictionary]	AOE1 - TRUE	AOE1 - TRUE
AOE1 - [ssh root@192.168.3.78]	AOE1 - TRUE	AOE1 - TRUE
AOE1 - [ssh admin@192.168.3.78 && ssh admin@192.168.3.78 « $password_dictionary]	AOE1 - FALSE	AOE1 - FALSE
AOE1 - [ssh admin@192.168.3.78 « $password_dictionary]	AOE1 - FALSE	AOE1 - FALSE
AOE1 - [ssh admin@192.168.3.78]	AOE1 - FALSE	AOE1 - FALSE
AOE1 - [ ]	AOE1 - FALSE	AOE1 - FALSE
DOE1 - [PermitRootLogin YES]	DOE1 - FALSE	DOE1 - FALSE
DOE1 - [PermitRootLogin NO]	DOE1 - TRUE	DOE1 - TRUE
DOE1 - [ ]	DOE1 - FALSE	DOE1 - FALSE

to apply his/her repertoire of forensic techniques and learn new ones, and he/she will do the same for the legal and procedure requirements defined in the scenario.

VIII. Conclusion and Future Work

This work presents a smart logical agent system in the Shellter security training system which was implemented in the synthesis of the architecture proposed in [12] and [11], that works in a rational way. The intelligent agent system monitors the students progress in the virtual environment to motivate them by using gamification techniques. The system analyzes the student profile in social networks searching for student interests, success or failure in past challenges to choose the techniques to be applied, emotional expressions, exceptional information security techniques and intervene in a student's activity to tutor him at the right moment.

A prototype agent was developed in order to show the system's functionality. This prototype was chosen based on the similarity of architecture of the agents that compose the system and its level of importance in the system. The Prolog implementation for the first predicate logic model-based reflex agent was evaluated in the test scenario by condition-action rules. The agent was subjected to a battery of tests, validating its operation using a simulated user case. So, it is possible to use the results from this work in a real-life scenario. The agent notation abstraction was implemented to facilitate the adoption of other security threats in the production system.

In the future, we will develop other agents not evaluated in this work, namely: (1) the UOA, (2) the UIA and (3) the UPA. Furthermore, there is room for considerable improvement in system performance.

Acknowledgments

The authors wish to thank the CAPES (Coordenação de Aperfeiçoamento de Pessoal de Nível Superior) and CNPq (Conselho Nacional de Desenvolvimento Científico e Tecnológico) for supporting this research work through the "Projeto FORTE - Forense digital tempestiva e eficiente - Processo 23038.007604/2014-69" project and Produtividade em Desenvolvimento Tecnológico e Extensão Inovadora - DT – Processo 305905/2016-3 funded by CNPq/Brazil.

References

[1] A. Nagarajan, J. Allbeck, A. Sood, and T. Janssen, "Exploring game design for cybersecurity training," in *Cyber Technology in Automation, Control, and Intelligent Systems (CYBER), 2012 IEEE International Conference on*, May 2012, pp. 256–262.

[2] B. Könings, F. Groh, N. Asaj, M. Poguntke, F. Schaub, B. Wiedersheim, and M. Weber, "Gamification: State of the art definition and utilization," in *Proceedings of the 4th Seminar on Research Trends in Media Informatics*, 2012, pp. 39–46.

[3] C. I. Muntean, "Raising engagement in e-learning through gamification," University of Cluj-Napoca, 1 Mihail Kogalniceanu Street, 400084 Cluj-Napoca, România, 2011, pp. 323–329.

[4] J. C. Lester, E. Y. Ha, S. Y. Lee, B. W. Mott, J. P. Rowe, and J. L. Sabourin, "Serious games get smart: Intelligent game-based learning environments," *AI Magazine*, vol. 34, no. 4, pp. 31–45, 2013.

[5] C. Willems and C. Meinel, "Practical network security teaching in an online virtual laboratory," in *Proceedings of the 2011 International Conference on Security & Management*, 2011.

[6] A. Gaspar, S. Langevin, J. Stanaback, and C. Godwin, "SOFTICE: facilitating both adoption of linux undergraduate operating systems laboratories and students' immersion in kernel code," *Systemics, Cybernetics And Informatics*, vol. 5, no. 3, pp. 30–35, 2007.

[7] A. Futoransky, F. Miranda, J. Orlicki, and C. Sarraute, "Simulating cyber-attacks for fun and profit," in *Proceedings of the 2Nd International Conference on Simulation Tools and Techniques*, ser. Simutools '09. ICST, Brussels, Belgium, Belgium: ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering), 2009, pp. 4:1–4:9, available in: <http://dx.doi.org/10.4108/ICST.SIMUTOOLS2009.5773>. [retrieved: december, 2016].

[8] M. Corici, "Ctf 365," 2017, available in: <https://ctf365.com/>. [retrieved: february, 2017].

[9] L. Padgham and M. Winikoff, *Developing Intelligent Agent Systems*. Wiley, 2005.

[10] S. Rangel and F. Hachem, "SHELLTER: um ambiente de aprendizado em segurança da informação com abordagem prática," 2015.

[11] S. J. Russell and P. Norvig, *Artificial Intelligence: A Modern Approach*. Prentice Hall, 2010.

[12] M. Wooldridge, *An Introduction to MultiAgent Systems*. Wiley, 2002.

[13] O. Security, "Exploit database," 2015, available in: <https://www.offensive-security.com/>. [retrieved: december, 2016].