CERIAS Tech Report 2002-25

HACKER: AN INTELLIGENT LEARNING AGENT

by Pranathi Venkatayogi, Bharat Bhargava

Center for Education and Research in
Information Assurance and Security,
Purdue University, West Lafayette, IN 47907

HACKER: AN INTELLIGENT LEARNING AGENT

CERIAS TR 2002-25

Pranathi Venkatayogi, Bharat Bhargava
Center for Education and Research in Information Assurance,
Computer Sciences, Purdue University
West Lafayette, IN, 47907, USA
[pv, bb]@cs.purdue.edu

ABSTRACT

The security threats involved in any software system are due to unanticipated attacks by hackers or terrorists. Research in security concentrates on providing technical solutions to these security threats [1, 2].These solutions might not work well once the assumed attacker behavior changes. Attackers quickly understand the current security structure of the system and come up with innovative ways to achieve their objectives. In order to estimate the objectives and possible attacks, one needs to know the behavior of a hacker. This report proposes the design for the simulation of a hacker as an intelligent learning agent, which can be used to observe the behavior change patterns and enhance the existing solutions to security threats.

The design supports the following:

1) The hacker learns from his experience and also from the information provided by the other hackers.

2) The mistrust component is used to decide on the extent to which the information provided by other hackers can be relied upon.

1. INTRODUCTION

The World Wide Web has become a vast resource of open information, which causes every piece of information to be accessible to any person. This has its pros and cons. Pros being easy availability of highly valued information at a mouse click, and Cons being the risk involved in the unconcealed nature of information. The major hit is to businesses which employ e-commerce. These risks are either due to low quality components both software and hardware involved in a system or due to hackers, or organized crime, business competitors , who get to know of the vulnerabilities of a system and then act accordingly. The major problem in overcoming the risks due to hackers is the unpredictable behavior of the hackers. The hacker quickly learns about the security strategy employed by a system and attacks again, now with more potential. Goal is to simulate this learning mechanism of the hacker and see how the firms could be affected by such attacks. Design begins with a simple model in which the firms are not adaptive; while the hacker tries to acquire resources which a target firm has and attacks it successfully. Hacker has the ability to learn not only from its own experience, but also from the information provided by other hackers about their attacks.

2. APPROACH

Hacker is designed as an intelligent learning agent. As mentioned in [3], the process of building the knowledge base would be greatly simplified if the agent could learn. Once the learning agent is provided with some initial incomplete and incorrect knowledge base the agent will be able to extend and correct it through learning. The remainder of this report is structured as follows: Section 3 talks about the related work. Section 4 illustrates the architecture of the system's agents. Section 5 provides the conclusions. Finally section 6 comments on related work and outlines future work directions.

3. RELATED WORK

The Related work can be categorized into following broad categories:

Simulation of Attacks.
Multi agent systems.
Application of Machine Learning Techniques.
Handling of Incomplete Information.
Studies of Hacker Psychology.
Simulation of Behavior.

Simulation of Attacks

[13] develops a model to evaluate the tradeoffs between the cost of defense mechanisms for networked systems and the resulting expected survivability after a network attack. The model consists of three sub models. The first sub model simulates the occurrence of attacks or incidents. The second sub model simulates the impact of an attack on the system. This depends on the type of attack and the defense mechanism installed in the system. The third sub model assesses the survivability of the system which depends on the degree of its degradation after the attack. Simulation of attacks or incidents is done by using a marked, stochastic point process, where the incidents are the events that occur at random points in time.

Multi agent systems

The agent based artificial market system in which customers and merchants delegate variety of tasks to personal intelligent agents that act as their artificial employees, and communicate using underlying interaction protocols is given in [4].The development of purchaser and seller agents in this system is based on a generic and reusable architechture. Each of the agents has communication, coordination and decision making modules.

Building of reconnaissance agents which are learning agents that infer user preferences and interests by tracking interactions between the machine and users in a long term is given in [6].The examples of reconnaissance agents Letizia and Powerscout are presented in [6].Letizia uses local reconniassance- searching the neighborhood of the current page, which Powerscout uses global reconniassance - making use of a traditional search engine to search the web in general.

Application of Machine Learning Techniques

A neuro-genetic approach to developing a multi-agent system which meta-searches for multi-media information in online information sources on web is given in [5].[5] uses neural networks for local searching and learning. Genetic algorithms are used to facilitate the evolution of agents on a global scale.

Machine learning techniques are used to develop an IDS to identify the specific behavior of the users in the company and raise alarm when any deviation to normal user behavior is observed[8].The goal is to detect insiders who are inappropriately intruding on the computers of others, with as many few false alarms as possible.

Anomaly detection also uses various learning mechanisms, such as neural networks [16], machine learning classification techniques [17], [18] and even mimicking of the biological immune systems [19]. Crosbie [20] also proposed genetic programming approach to detect anomalous behavior in a system.

Handling of Incomplete Information

Work related to coping with incomplete information provided by traders in middleware is cited in [7].Instead of relying 100 % on a trader, [7] assumes that traders provide only rough matches and trust evolves with the client's experience.

Studies of Hacker Psychology

Hacker behavior has been studied through analysis of security incidents on the Internet. Attackers have been classified [13] into various groups such as hackers, spies, terrorists, corporate raiders, professional criminals and vandals depending on what their intention is for breaking into a computer or a computer network. The results of the attacks could be corruption of information due to any unauthorized alteration of files stored on a host computer or data in transit across a network, disclosure of information - the dissemination of information to anyone who is not authorized to access that information, theft of service - the unauthorized use of computer or network service without degrading the service to other users or the most common denial-of-service - the intentional degradation or blocking of computer or network. Hackers could try to get access to the computer by using a variety of tools.

From the analysis of the criminal activities [15] gives the following insight about the psychology of the hacker. People who commit computer crimes vary widely in skills, knowledge, resources, authority and motives. Computer criminals may have different levels of skill in formal education, social interactions and use of computer systems. There are three classes of computer criminal: tool makers, tool users and script followers.

Motives for a hacker include greed, need (to solve personal problems such as paying gambling debts), inability to recognize the harm done to other, personification of computers (seeing computers as adversaries in a game), the Robin Hood syndrome (seeing corporations as so rich that stealing from them is morally justified).

Simulation of Behavior

The Physical Conditions, Emotional State, Cognitive Capabilities and Social Status (PECS) architecture proposed by Schmidt [14] is intended to support the design process of agent-based simulation models. Individual human behavior and decision making, interaction between individuals as well as interactions of individuals with their environment form the crux of this approach. This reference model provides a concept for the construction of agents, a communication infrastructure, an environment component and domain independent model architecture.

4. REPRESENTATION OF FIRM

The four important components involved in the firm profile generation task are

IT Profile
Vulnerability Profile
Security Profile
Risk Profile

The IT profile of a firm includes the resources that form the organization’s IT infrastructure: (1) the type of network (LAN, WAN, MAN), (2) the database (ASES, BKD, ChronoLog, DataLog++, db4o, EOS, Firebird, GiST, Ingres, InterBase, MetaKit, MIND), (3) the operating system (Windows, DOS, Mac, UNIX, LINUX) and (4) the various software applications (JavaScript, ActiveX, Browser) being used.

Once a firm chooses it's IT profile, the vulnerabilities related with each component of the IT profile are represented in the form of a vulnerability profile [21]. Based on this vulnerability profile and the budget constraints, the firm chooses its security components. The major categories of security components are the authentication mechanisms to be used, the firewalls, log Systems and encryption mechanisms. Various commercial products are available for each of these categories. The organization chooses one among these based on its budget. Based on the security rules chosen, a security profile is generated. These rules will not help overcome all of the vulnerabilities. Hence a risk profile indicating the risks that a firm faces when exposed to these vulnerabilities is generated. This is the profile the attacker uses to understand a firm’s weak points and exploit them based on his capabilities.

5. AGENT ARCHITECTURE

The perpetrators are represented as artificial agents and learning mechanisms implemented to make the agents intelligent enough to represent a real world hacker. The agents learn by gaining intelligence from the knowledge bases that contain data from the real world environment.

As shown in Figure 3, the agent-based architecture of a hacker consists of

Agent brain
Interface module

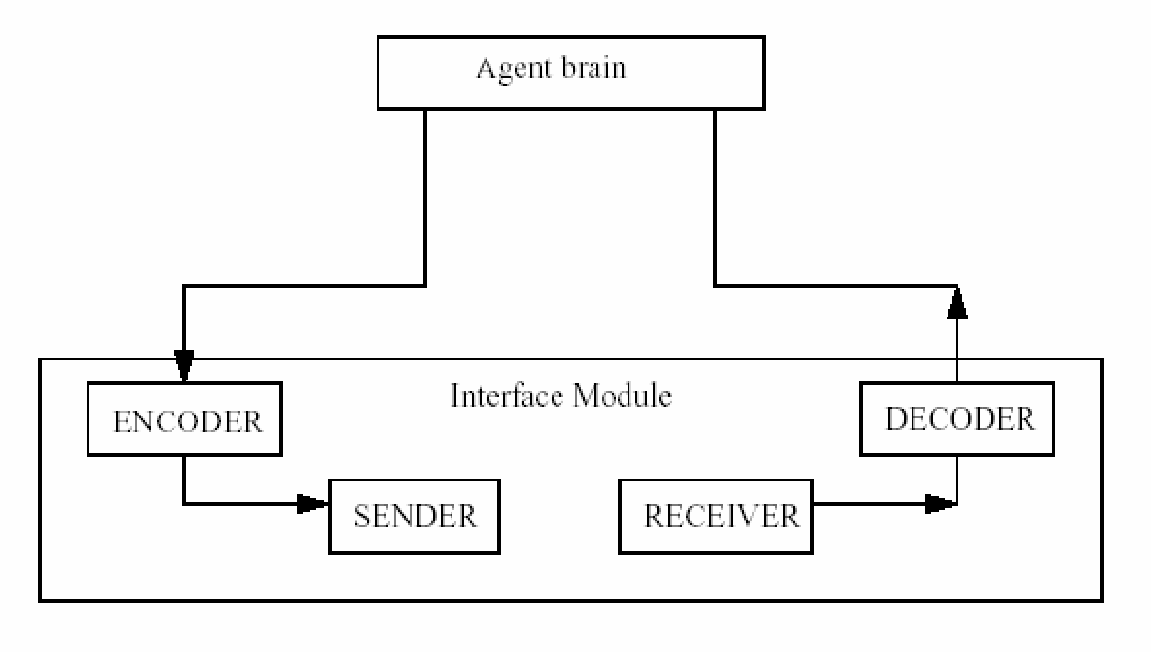

Figure 3: Agent Based Architecture of Hacker

Interface module

This module provides the agent interface with the external world. One of the main functionalities is to transform messages into the format agreed upon as an interface between agents and its transfer to the other agents. The other functionality is to make the inverse transformation of the messages received from the external world and handing over them to the coordinator module. This module is further subdivided into Encoder, Decoder, Sender and Receiver.

Encoder: Encodes the message to be sent to external agent, as defined by the communication
Protocol.
Decoder: It decodes the external message received into the format expected by the agent brain.
Sender: It sends the message to target agent.
Receiver: It receives the message from an external agent and hands over to the decoder to process it.

Agent brain

The agent brain is representative of the adaptive attackers. It helps attackers gain utility from every successful attack and exploitation of specific vulnerabilities in the firms. It also helps attackers choose strategies for attack based on the success or failure of previous attacks. It also will dynamically adapt to the security the firms are providing.

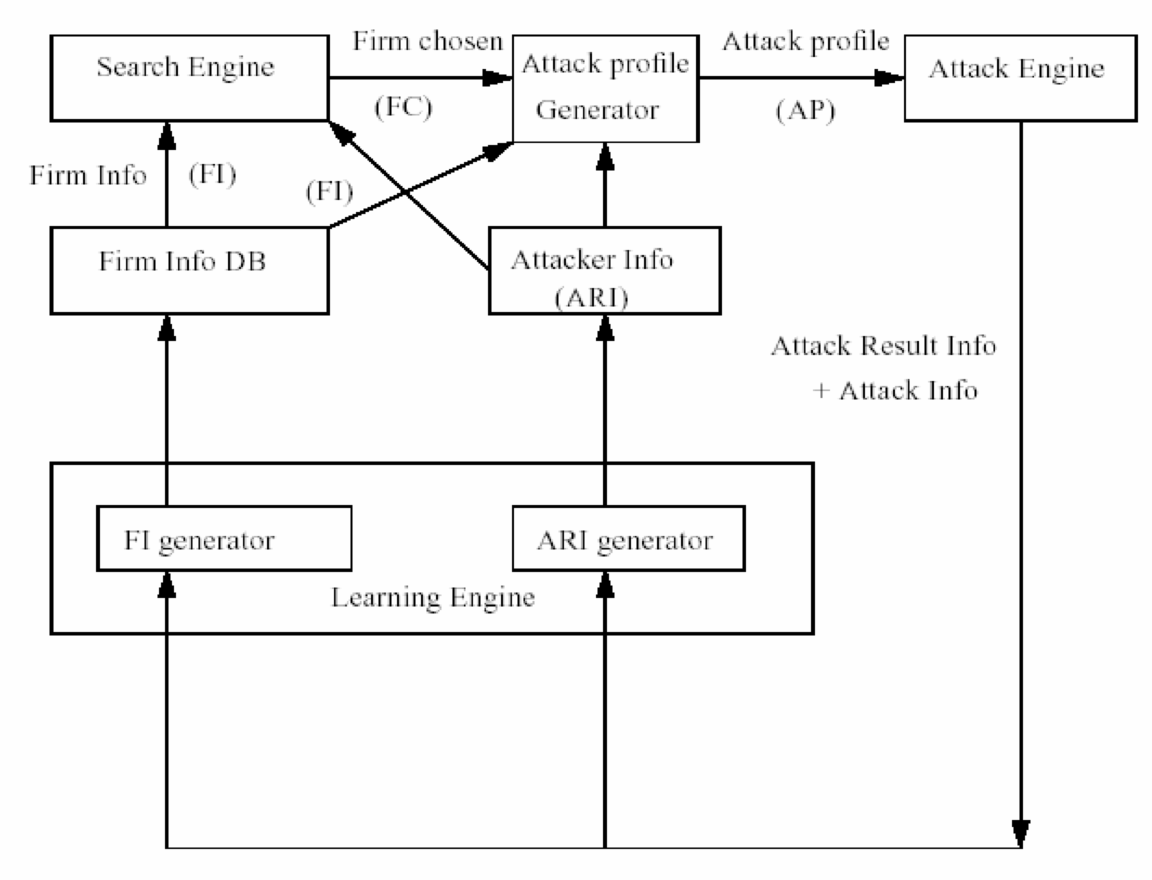

Figure 4: Agent Brain

The definitions of the information flow parameters in the agent brain module are as follows:

Firm Info: Firm Info *(FI)* is representative of knowledge attacker has about the firm. This includes the knowledge about the IT profile and the vulnerability profile and is represented as a duple <*AIP*, *AVP*>.AIP (AVP) is a 4b (4k)-bit string represented in a format similar to the IT profile (vulnerability profile) of the firm.

Attack Profile: Attack Profile *(AP)* is a *4k/μ* bit binary string where the bit values represent the capability of the attacker to exploit particular vulnerabilities. It has same format as the risk profile (*RP*) of the firm mentioned earlier. The probability of success of attack perpetrated by any attacker l with profile AP_l on a firm with risk profile RP is given by,

$$p_{lik} = \begin{cases} zero & \\ very\ low & \text{if} \\ low & \\ high & \end{cases} \begin{array}{c} AP_{lk} = 0 \\ (AP_{lk} = 1) and (RP_k = 1) \\ (AP_{lk} = 1) and (RP_k = 0) \\ (AP_{lk} = 1) and (RP_k = -1) \end{array} \qquad (6)$$

Where, k represents the bit at which the vulnerability of the firm is exploited

If the attack against a firm is successful, the firm incurs losses associated with the vulnerabilities that were exploited. The security rule adopted by this firm also has an associated decrease in strength due to failure in defending firm's resources. However, if the attack against a chosen firm is unsuccessful, the adopted security rule gains strength.

Attacker Info (ARI): Attacker Info (*ARI*) is a list of resources the attacker possesses (RR) to exploit the vulnerabilities of the firm.

Attack Info: Attack Info (AI) is the information of the firm obtained by the attacker after the attack has taken place. It is represented as a duple <firm-id, losses-incurred-by-firm>
firm-id: Each firm in the environment is assigned a unique id, firm-id.
losses-incurred-by-firm: This parameter represents loss incurred by the firm due to the attack.

Attack Result Info: Attack Result Info (*ARI*) indicates result of the attack success (1) or failure (0).

Agent brain (Figure 4) comprises of the following four components

- Search Engine
- Attack Profile Generator
- Attack Engine
- Learning engine

Search Engine

The basic functionality of Search Engine is to choose a firm to attack. It can be modeled as an optimization function, which maximizes benefits incurred by the attacker, has maximum utilization of the attacker resources, and minimizes risks involved. The Firm Info from Firm Info Database(FIDB) and attacker info (ARI) are inputs to Search Engine, and its output is firm chosen (FC).

Attack Profile Generator

FC given as input from Search Engine this module uses the knowledge of the firm obtained from FIDB, along with attacker info (ARI) and generates attack profile.

Learning Engine

It can be subdivided into two major sub components

- FI generator
- ARI generator

Learning Engine receives attack info and attack result info obtained as feedback from the attack engine after the attack has been placed and updates AI and knowledge about the target firm. It also receives the attack related information of the attacks performed by other agents on the target firm and enhances its knowledge about that firm.

FI generator updates FI in FIDB, while ARI generator updates attacker resource info based on the feedback of the attack.

Attack Engine

This modules places an attack on the FC, determines the result of the attack, losses to the firm due to attack, benefits to the attacker, new knowledge of vulnerabilities of the firm. All the above is given as feedback to learning engine.

5. CONCLUSIONS

This report presents a simple design of hacker, assuming the Firm is not adaptive. Moreover many psychological and humane aspects which lead to hacker behavior in real world are not yet embedded into the architecture. This design gives the direction one has to proceed when trying to simulate any human behavior. Moreover the proposed agent architecture can be enhanced and used in the simulation project going on at "SEAS" laboratory at Management Dept, Purdue University, which right now deals with simulation of Firms as intelligent agents, while the Hackers are dumb.

6. FUTURE WORK

This module can be enhanced to incorporate other human behaviors like perception, visualization and perception, helping build an socially intelligent agent. It can be used to know predict the unknown behavior of the hackers as mentioned in [9].The sub modules in the interface module can be enhanced to handle more complicated, descriptive and informative message interchange between agents. More complicated scenarios can be simulated if intelligence can be built into firms too. In this context, the agent architecture can also be enhanced to incorporate reconnaissance into the agent behavior about the firms. As mentioned earlier, design can be evolved to involve detailed and meticulous assignment of mistrust factor to the information provided by other agents, based on the results obtained by trusting them in the past. Work can also be extended by taking into consideration the fact that information from other agents could be incomplete or in some cases incorrect too.

7. REFERENCES

[1] J.S Balasubramaniyam, Jose Omar Garcia-Fernandez, David Isacoff, Eugene spafford, and Diego Zamboni (1998) "An Architechture for Intrusion Detection using Autonomous Agents", COAST Technical Report 98/05.

[2] Salvatore Stolfo, Andreas L. Prodromidis, Shelley Tselepis, Wenke Lee, Dave W. Fan and Philip K. Chan, "JAM: Java Agents for Meta-learning over Distributed Databases", Technical Report, Department of Computer Science, Columbia University, 1997.

[3] Gheorghe Tecuci, "Building intelligent agents, An Apprenticeship Multilstrategy Learning Theory, Methodology, Tool and Case Studies", Academic press, 1998.

[4] Nikos Karacapilidis, Pavlos Moraitis, "Intelligent Agents for an Artificial Market System", proceedings of ACM, Agents'01, pp 592-599.

[5] Edmund S.Yu, Ping C.Koo, Elizabeth D. Liddy, "Evolving Intelligent Text-based Agents", proceedings of ACM, Agents'00, pp 388-395, 2000.

[6] Henry Lieberman, Christopher Fry, and Louis Weitzman, "Exploring the Web with Reconnaissance Agents", communications of ACM, vol. 44, No. 8, pp 69-75.

[7] Micheal schroeder, Julie McCann, Dan Haynes, "Rough Traders and Intelligent Clients", proceedings of ACM, Agents'00, pp 259-415.

[8] J. Shavlik, M. Shavlik, M. Fahland, "Evaluating Software Sensors for Actively Profiling Windows 2000 Computer Users", Presented at the Fourth International Symposium on Recent Advances in Intrusion Detection, Davis, CA(2001).

[9] Shailendra Raj Mehta, Alok Chaturvedi, Mukul Gupta, Bharat Bhargava, "Behavior Based artificial agents for information security", 2001.

[10] David E. Goldberg, "Genetic Algorithms in Search, Optimization, and Machine learning",
Addison-wesley publishing company, 1989.

[11] Alok Chaturvedi, Shailendra Mehta, Mukul Gupta, "fighting the willy hacker::Modeling Information Security, Issues for on-line Financial institutions using the seas environment",paper presented at, The annual Meeting of the Internet Society, July 18'th-21'st, I.NET 2000, Yokohama, Japan.

[12] Souymo D moitro, Sureh L Konda, "A simulation model for Managing surviability of Networked Information systems", Technical Report, CMU/ SEI-2000.

[13] J.D. Howard, "An Analysis of Security Incidents on the Internet", Ph.D. Thesis, Carnegie Mellon University, 1995, Chapter 6.
(http://www.cert.org/research/JHThesis/Chapter6.html).
[14] B. Schmidt, "The Modelling of Human Behavior", SCS Publications, 2000.

[15] Donn B. Parker, "Fighting computer crime, A New framework for protecting Information". Wiley (NY) ISBN 0-471-16378-3. xv + 500 pp 1998.

[16] K.M.C. Tan, "The Application of Neural Networks to UNIX Computer Security", Department of Computer Science, University of Melbourne, Parkville 3052, Australia.

[17] T. Lane and C.E. Brodley, "Temporal Sequence Learning and Data Reduction for Anomaly Detection", In Proceedings of the Fifth ACM Conference on Computer and Communications Security, pages 150-158, 1998.

[18] S. Forrest, S.A. Hofmeyr and A. Somayaji, "Computer Immunology", Communications of the ACM, 40, 10, 88-96, October 1997.

[19] S.A. Hofmeyr, "An Immunological Model of Distributed Detection and its Application to Computer Security", Ph.D. Thesis, University of New Mexico, May 1999.

[20] M. Crosbie, "Applying Genetic Programming to Intrusion Detection", In Proceedings of 1995 AAAI Fall Symposium on Genetic Programming, November 1995.

[21] S. Potluri, Technical Report, Krannert School of Management, October 2001