Applying AI to improve the performance of client honeypots

Applying AI to Improve The Performance of Client Honeypots

Van Lam Le, Peter Komisarczuk, Xiaoying Sharon Gao
School of Engineering and Computer Science, Victoria University of Wellington
P.O. Box 600, Wellington 6140, New Zealand
{van.lam.le, peter.komisarczuk, xiaoying.gao}@mcs.vuw.ac.nz

ABSTRACT
Victoria University has developed a capability around the detection of drive by download attacks using client honeypot technology [1-3]. Two types of client honeypot, low-interaction and high-interaction honeypots, have been developed to inspect malicious web pages. A new client honeypot model, called a hybrid system, has also been proposed to improve the performance of client honeypots [2]. These client honeypots have made significant contributions to Internet security through detection of malicious servers. However, their performance has shown there are areas where artificial intelligence (AI) technology can add value to create more adaptable client honeypots. In this workshop, we briefly present client honeypots which have been developed by Victoria University and how we can apply AI to improve their performances.

Keywords
Client Honeypots, Artificial Intelligence, HoneyC, Capture-HPC

Client Honeypots at Victoria University
Client honeypots are measurement systems that actively measure the Internet for malicious web servers and malicious content. Malicious servers respond to client requests for a URL with the requested web page but may also include crafted exploit code which aims to compromise the client system resulting in loss of integrity or loss of data. These attacks are called "drive-by-downloads" [5-6].

Client Honeypots can be classified into two main types: low-interaction client honeypots and high-interaction client honeypots. Low-interaction client honeypots are developed by emulating system services which are attractive to the intruders. On the other hand, high-interaction client honeypots use real systems that interact with potential web servers [2].

To detect malicious web pages, Victoria University has developed both client honeypots: HoneyC - a low-interaction client honeypot and Capture – high interaction honeypot. HoneyC inspects malicious web pages by analyzing the responses from web servers directly. It uses static method such as pattern matching, static code analysis algorithm to detect malicious web pages [3]. Capture, on the other hand, detects malicious pages by deploying a real operating system in a virtual machine to interact with potential malicious web servers and monitoring for any unauthorized state changes during surfing web pages [1]. In performance, HoneyC has high false positive rate while Capture has a false positive rate of zero but may miss attacks that detect for the presence of a virtual machine. However, Capture consumes large computing resources and time to detect malicious servers and exploits [2].

To take advantages of both client honeypots, a hybrid client honeypot system has been proposed. In this system, there is a set of client honeypot nodes operating collaboratively to detect malicious web servers. Each node uses both client honeypots: low-interaction and high-interaction honeypots. Low-interaction honeypot are first used to analyze the responses from web servers and suspicious web servers are forwarded to high-interaction honeypots for final classification [2].

PAM 2009, 1-3 April 2009, Seoul, Republic of Korea.

Performance of Client Honeypots: Need of AI Technology.
The detection performances of client honeypots has benefited from applying AI technology. First of all, HoneyC uses static methods to detect malicious web pages. Signatures from known malicious contents are used to classify web pages. Therefore, it misses unknown malicious contents because the drive-by-download attacks are mutating over time, called "concept - drift" and so pattern matching is not effective long term [2]. In addition, Capture also misses trigger attacks which need users' interactions to make state changes in the system [4]. Using AI technology to study potential web page contents can be used to overcome these issues.

In hybrid client honeypots, AI technology is very important in order to get significant performances. First off all, AI technology can be used to analyze the responses from potential web servers. It can reduce the set of potential URLs sent to high-interaction honeypots for inspecting. Moreover, high-interaction honeypots can use AI to study the activities of intruders. The outcome of this study can be some generated patterns or signatures which can be used in low-interaction honeypots and be continuously updated to overcome concept-drift.

Current Works in Applying AI for Client Honeypots
Christian et al. propose static heuristics to classify malicious web pages. The main idea of this method is to classify web pages using static heuristics before they are inspected by high-interaction honeypots. The web pages classified as malicious were then forwarded to high-interaction honeypot for evaluating. To implement this method, common elements of malicious web pages were studied and some potential attributes were chosen. These attributes were extracted from both malicious and benign web pages and were fed into J4.8 decision tree learning algorithm implementation of Weka Machine Learning Library [8]. This

method has been proposed to improve the known false negative rate in high-interaction honeypot. It also improves the speed of performance as reducing the set of inspected URLs [4].

In addition, Vicky K. [7] used clustering to classify intruders' intention in high-interaction honeypots. She used log files generated from Capture – a high-interaction honeypot which instruments the Windows XP operating system to detect events. The main task is to classify malicious web pages with similar behaviors into the same groups. These groups are then studied to find the common characteristics which can be used to generate the general signatures to detect malicious web pages. To implement this research, all actions which are made by malicious web pages are monitored and stored into log files by high-interaction honeypots. These actions are encoded as sequences of characters and indexes. The characters and indexes presents for actions and affected objects. These sequences are then fed into hierarchical agglomerative clustering algorithm for estimating the number of clusters. They are finally fed into K-means clustering algorithm. Smith Waterman similarity measuring techniques are used to measure the similarity of sequences of actions and indexes because of the similarity of events captured to DNA pattern matching [7].

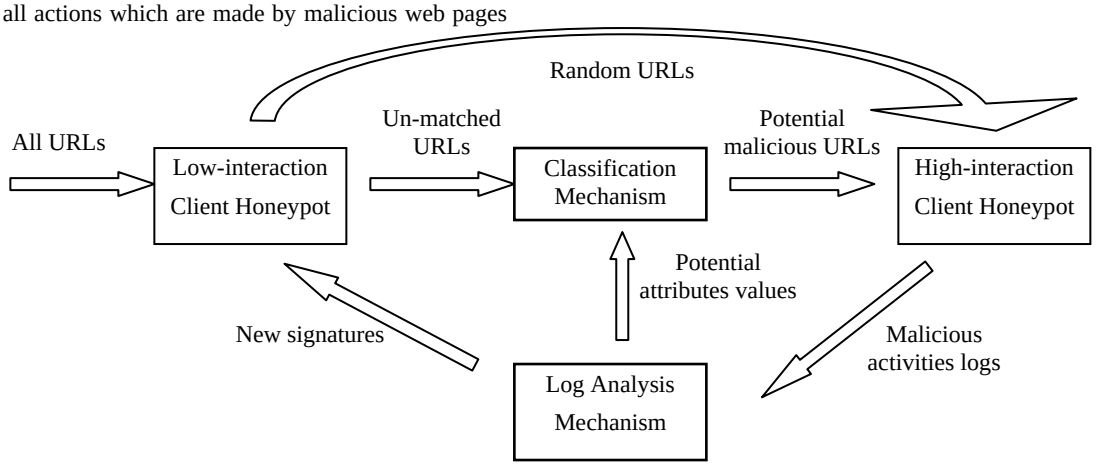

Figure 1. Client Honeypot system with AI implementation

Further Researches on Applying AI Technology for Client Honeypots

Christian et al. has used J4.8 decision tree learning algorithm to classify web pages while Vicky K. used clustering to analyzing unauthorized activities for classifying intruders' behaviors. There are still many challenges to use AI technology for client honeypots. These challenges can be addressed as workshop discussion topics:

- Which available learning algorithms can do better performances than J4.8 decision tree learning algorithm?
- How to study activities of malicious webs to overcome missing trigger attacks at high-interaction honeypots?
- What is the relationship between unauthorized activities monitored by low-interaction honeypots and attributes used by high-interaction honeypots to classify potential malicious web pages?

REFERENCES

[1] Seifert, C., Steenson, R., Welch, I., Komisarczuk, P., Endicott-Popovsky, "B. Capture - A Behavioral Analysis Tool for Applications and Documents", Proceedings of the 7th Digital Forensics Research Workshop Conference, Pittsburgh, August 2007.

[2] Seifert, C. "Improving Detection Accuracy and Speed with Hybrid Client Honeypots, PhD Proposal", Victoria University of Wellington, Wellington, New Zealand, available at http://www.mcs.vuw.ac.nz/~cseifert/publications/publications/Cseifert_phd_proposal-Hybrid_Client_Honeypots.pdf, accessed on 30th January 2009.

[3] C. Seifert, I. Welch, and P. Komisarczuk, "Honeyc – the low-interaction client honeypot", in NZCSRCS, Hamilton, 2007.

[4] Seifert, C., Komisarczuk, P., Welch, I., "Identification of Malicious Web Pages with Static Heuristics", in the Austalasian Telecommunication Networks and Applications Conference, Adelaide, 2008.

[5] Seifert, C., " Know Your Enemy: Behind The Scenes Of Malicious Web Servers", The Honeynet Project, 2007, available at http://old.honeynet.org/papers/wek/KYE-Behind_the_Scenes_of_Malicious_Web_Servers.pdf, accessed on 30th January 2009

[6] Seifert, C., Steenson, R., Holz, T., Yuan, B., Davis, M.A., "Know Your Enemy: Malicious Web Servers", The Honeynet Project, 2007, available at http://old.honeynet.org/papers/wek/KYE-Malicious_Web_Servers.pdf, accessed on 30th January 2009.

[7] Vicky, K., "Clustering Malicious Networking Attacks", Msc thesis, Victoria of Wellington, Wellington, New Zealand, 2008.

[8] I. H. Witten and E. Frank, "Data mining: Practical machine learning tools and techniques", 2nd ed, San Francisco: Morgan Kaufmann, 2005