Dynamic Authentication by Typing Patterns
TJHSST Senior Research Project Proposal
Computer Systems Lab 2009-2010

Luke Knepper
luke@lukeknepper.com

October 22, 2009

Abstract

This project will analyze and test the accuracy of dynamic typing pattern authentication methods. The program will generate a dynamic set of text that the user will be prompted to type, and then it will feed the user's typing characteristics through neural network structures. Experimentation will be done to determine the most accurate neural network structures and data collection conditions. This process will be an improvement on normal typing pattern authentication techniques, which use static passwords rather than dynamic text.

Keywords: authentication, security, typing patterns, neural networks

1 Introduction

1.1 Background

Current authentication techniques span all three tiers of security:

- Tier 1 - Identification (usernames)
- Tier 2 - Knowledge and Possession (passwords, ID cards, security questions, etc.)

- Tier 3 - Skills and Capabilities (captchas, voice recognition)

In the online world, usernames can be stolen, passwords and security questions can be guessed, and captchas can be cracked. The process of analyzing typing patterns, a tier 3 security method, has been around for decades, but only recently has been put to use commercially. Currently, commercial products only analyze the user's typing patterns when the user is typing their password, a simple, static word. Any computer hacker can easily write a keylogger to record the user's keystrokes when typing this word and then simulate that process to gain access to the user's account.

1.2 Purpose

Dynamically generating content for the user to type and then analyzing that content will make hacking the algorithm considerably harder, especially when more advanced typing characteristics are used. This project will study the accuracy of authenticating users by their typing characterisitics using dynamic text blocks.

1.3 Procedure

To refine the neural network, an online system will be created to collect massive amounts of typing data from volunteers. This data will be used to train the different neural network structures in order to determine the most accurate structure for this purpose. Variations that will be tested include the use of backpropagation to train the network, genetic techniques to breed the weight vectors, and a Hebbian learning system to do so efficiently. To refine the user set-up and log-in processes, a series of tests will be completed using a sample of user volunteers on multiple sittings. The users will create dummy accounts and then attempt to log in to their own accounts and the accounts of others. The accuracy of different set ups will be measured.

1.4 Hypothesis

Based on research, the hypothesis is that a multi-layered genetically-bred neural network using averages of mid-sized typing data will be the best trade-off for accuracy and efficiency, but that this set-up will not offer the accuracy needed to be a stand-alone authentication system, and will instead have to

work in conjunction with older techniques, such as passwords and security questions.

2 Background

There is currently a patent (US 6151593) for an authentication scheme by typing pattern analysis. This method reads in the time between keystrokes for a user when typing their password and then trains a three-layered neural network to this combination. It does not allow for a dynamically-generated content to be used, and does not test the different lengths of passwords. This is the most basic application of typing techniques for authentication, and this project will extend beyond these simple methods.

Multiple typing pattern log-in software packages exist, such as Psylock, but they all have the same weakness as the patent above: they rely on a static password instead of dynamic typing content and therefore can be easily hacked.

An independent team of researchers, headed by Peacock et al., tested the effect of many variations, including neural network set-up, password length, acceptance stringency, data used, and function used. They found the most effective neural network structure from their tests was to use a set-up where many independent neural networks are trained on different cores (i.e. parallel processing) using randomly generated starting weight vectors. During the training, the best weight vectors are picked and created using genetic algorithms. They found the smaller (more stringent) acceptance ranges came up with a good amount of false alarms (when it didn't let the correct user in, happened 22% of the time) but also minimized break-ins (when the incorrect user was let in, happened 3% of the time). They also found the most effective password length was 7 characters, a mid-sized password (the longer passwords had no break-ins but many false alarms, and the shorter passwords had many break-ins). They concluded that a linear evaluation function was more effective than a quadratic function and that averaging was more effective than counting each training run. They suggest their results can be improved (75% success, 22% false alarm and 3% break-in for their best algorithm).

Another team working under L. Maisuria compared the accuracy of neural networks compared to cluster algorithms. A multi-layered perceptron-based neural network which learned on the Hebbian learning theory was used, as

were ten different metrics to compare the clusters for the clustering.
They tested the different algorithms by recruiting twenty volunteers to participate in three different sittings. In the first sitting, they all trained their neural networks by typing in their password sixty times. In the next sittings, they attempted to log in to their accounts and break into the accounts of others. The sittings were spaced out by one week.
The study found that the clustering methods were slightly more accurate than the neural networks in rejecting impostors. They only found an average of 80% to 90% accuracy in rejection rate, not enough to comprise a stand-alone security system but certainly good enough to be used in conjunction with traditional methods. They found that all keystrokes should be measured, including the beginning and end strokes to the enter key, for the highest accuracy. They found that allowing impostors to observe the users typing before attempting to break in to their accounts had little effect on the accuracy.

3 Design Criteria

The final version will have two stages:

- The set-up phase, where users will be prompted to type a longer amount of text with which to train the system.
- The log-in phase, where users will be prompted with a dynamically generated text block in order to gain access to their account

At completion, this project should be composed of the following:

1. Neural Network
    - The neural network should dynamically adapt to allow for an increase or decrease in layers, easily allowing for a multi-layered neural network structure.
    - The weight vector will be trained by starting with a population of randomly-generated vectors, training them slightly via back propagation, then breeding them to form a new generation, training them again, and repeating this process until an optimal vector is created. This process should be done across multiple processors or cores to speed it up.

2. Test Data Collection
   The test data will be collected via an online application, which will prompt subjects to enter in a paragraph and then store their keystroke information for future experimentation.

3. GUI
   The GUI will be optimized for the optimal size found by the previous testing and experimentation, whether this be one word, sentence, paragraph, or a combination of those.

4. Authentication
   The application will determine what other tiers of authentication are needed in order to have the most accurate system. It is predicted to include a username and possibly a password as well, in addition to the passage which the user is prompted to type.

4 Procedure

The development phase will be separated into three parts:

- Proof of Concept
- Neural Network Development
- Input Optimization

4.1 Proof of Concept

A simple proof-of-concept was completed in October '09. The program prompts two users to both type a sentence and uses their data to train a simple single-layer neural network. It then prompts the users with a third sentence and instructs one of them, whose identity is unknown to the computer, to type the third sentence. It runs the final data through the trained neural network and determines which user typed the third sentence. It has been tested twenty times with different users and has a 90% accuracy to this date, proving that this concept does work. This simple implementation leaves much to be desired.

4.2 Neural Network Development

In order to test the accuracy of different neural network structures, large amounts of test data must be collected. A Java applet will be constructed and posted online to collect this data. The applet will consist of a text box in which the user can type and a text area which will display messages to the user to prompt them to type certain text. They will be prompted for different kinds of text, including a word (repeated times), a sentence, and a short paragraph. The typing data will be collected and stored on to the server.

This data will then be used in order to optimize the neural network. An automated testing algorithm will be created which will repeatedly train a given neural network type with one set of typing data, and then run typing data from the same person and from others through the network to comput the accuracy. The program will repeat this process for many different typing pattern sets, and then compute the average accuracy of the given network type. Different networks will be tested to determine which is the most accurate.

4.3 Input Optimization

Using the data collected in the previous stage, the automated testing system will be modified to determine the most accurate data size. Instead of varying the neural network structure, now the corpus size will be varied. Different sizes include single word, sentence, and short paragraph. It has been shown that a bigger corpus increases accuracy, but is less efficient. By comparing the accuracy and efficiency of the different lengths, the experimenter can determine the optimal length.

The consistency over time will then be tested using a small group of about twenty volunteers. The test will be broken up into three sittings, similar to the research discussed above. The first sitting will have the volunteers creating dummy accounts in the optimized system by repeatedly typing dynamically generated text. In the next sittings, spaced out by one week and one month, the subjects will attempt to log in to their accounts and into the accounts of others (simulating impostors). The accuracy in allowing a correct log-in and rejecting an impostor will be computed to determine if the method holds up over periods of time.

References

[1] Cho, S. and Han, D. "Apparatus for authenticating an individual based on a typing pattern by using a neural network system." http://www.freepatentsonline.com/6151593.html

[2] Peacock, A. et al. "Typing Patterns: A Key to User Identification." http://www2.computer.org/portal/web/csdl/doi/10.1109/MSP.2004.89

[3] Maisuria, L. "A COMPARISON OF ARTIFICIAL NEURAL NETWORKS AND CLUSTER ANALYSIS FOR TYPING BIOMETRICS AUTHENTICATION."