**Alcatraz, an end-to-end SaaS product for complete computer and network security**

**Dr. Annapurna P Patil J Geetha Reddy**

Professor, Assistant Professor,

Department of Computer Science and Engineering, Department of Computer Science and Engineering,

Ramaiah Institute of Technology Ramaiah Institute of Technology

Bangalore - 560054 Bangalore - 560054

**Ayush Kumar Bineet Kumar**

Student, Department of CSE, RIT Student, Department of CSE, RIT

**Aman Mehta Deepak Jayaprakash**

Student, Department of CSE, RIT Student, Department of CSE, RIT

***Abstract--***The demand for modern tools and techniques to restrict access to applications and services which contain delicate data is increasing exponentially each year. People can no longer rely on older methods of computer and network security as they fail in dealing with the challenges posed.

The product is aimed at providing an architecture and open source code to developers so that they can embed these features into their applications to enhance the security. The services provided are top notch and cover the broad spectrum of computer and network security. All the features of the product involve the application of **Data Mining** and **Machine Learning** techniques onto the domain of **Computer Security**.

The most prominent features of the product are listed below:

- **Keystroke Biometric Authentication**: Ensembles result of 3 statistical models and classifies user based on Dynamic Thresholding

- **Network Intrusion Detection:** Data mining on NSL-KDD data set, build 3 models, accuracy comparison and data visualization.

- **Anti-key logging:** Uses authentication based on GET request exchange and adds a double layer of AES + Hashing to guarantee both Data Integrity and Security.

- **O-Auth with Hashing:** Traditional O-Auth modified with additional layer of Hash to impart Digital Signature.

The complete product with the features is based on dynamic-dispatch, a kind of “on-demand software” inspired ideology, hence the term SaaS. The entire product is released as open source with detailed documentation on GitHub. It is released as a stable live version with a guide for developers on how to contribute.

The product when compared to existing software applications like Snort, SpyShelter and other biometrics have advantages like

- No external hardware

- One complete encapsulated bundle

- Open source code

- Cloud Integration

- Detailed documentation and support

*Keywords: Keystroke dynamics, zero variance removal, O-Auth, Network Intrusion Detection, Anti-key logging, Statistical modeling*

1. INTRODUCTION

The biggest necessity of a product of this nature arises from the fact that developers prioritize attributes such as functionality, UI, memory and efficiency while designing and creating applications of both small and large scale. But they comparatively turn a blind eye or assign a low level priority to security. This is understandable because making an application/ service secure requires a lot of programming effort and expertise. The objective of the product is hence to provide an overall architecture, design and code to application developers so that they can embed these onto their modules.

The features are also on a demand and get basis so only those services which are required can be taken. Each feature is an application of statistical modeling and ML algorithms on layers of computer security. The services use highly efficient mechanisms ranging from Dynamic Thresholding, Zero-variance removal, AES, Hashing, Data Mining to Ensemble learning, asynchronous communication and accuracy analysis.

The entire product is versioned and released on GitHub. Each feature is a novel implementation because all of them are not a single model-classifier service. They are an ensemble learnt multi classifier accumulation, wherein each classifier/ statistical model is designed to outperform one another so that the drawbacks of one are overcome in another thereby leading to a more accurate result. Enough mathematical support is given to all the algorithms used. Some of the concepts used are linear programming, statistical mean, median, variance, k-cross validation, iterative averaging, etc.

1. LITERATURE SURVEY

The features offered in the product belong to well researched and hot topics lying in the domains of Biometrics, Data Mining, Machine Learning, Statistical modelling and Computer/ Network Security. Prominent work has been done in the field of keystroke biometrics. This is because of the rising significance in the area of digital signatures, data integrity and security. Many existing approaches implement the static text method of keystroke dynamics [5]. The existing live implementations for “free text” are very hard to find, which can be attributed to the low accuracy rate and other problems like bias, over-fitting, false recognition rates and non-significance

First true work on keystroke dynamics originated from Monrose and Rubin [23]. Their learning models were implemented by using statistics like digraph/ trigraph mean and variance

Researchers have presented their work on choosing different distance metrics ranging from Manhattan to Euclidian and combined them with mean, median and tested their suitability on the biometric authentication. For the implementation both traditional and modern classifiers have been used, including knn[5], Bayes Point Machines, K-mediods/means methods [11], genetic algorithms, neural networks , and SVMs if the attributes are non linearly dependant.

Network Intrusion Detection happens to be one of hot topics in the domain of Network (both wired and wireless) security. Prominent data mining algorithms like Naïve Bayes, Decision Tree, K-nearest neighbor, K-Means and Fuzzy C-Mean clustering algorithms and machine learning strategies like Support Vector Machine (SVM), neural nets have been applied to study the classification. The existing work can be broadly classified into:

- Machine Learning Based IDS

- The Multiple Classifiers (MCS)

- Boosting based IDS

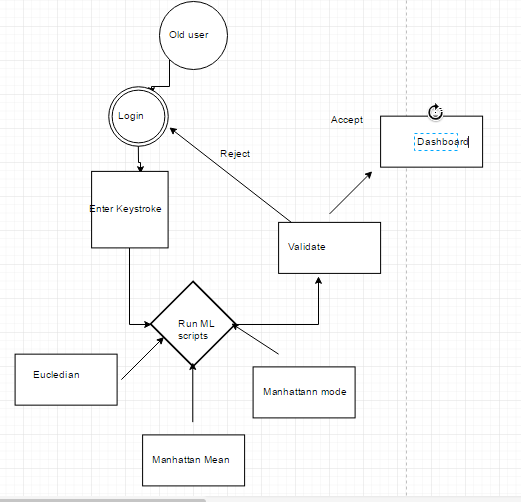
- Data Mining and Rule based IDS

There has been quite few implementations of Anti-Key logging systems and have been offered as service or an application supporting some Operating Systems. The research done on this include, Behavior based detection technique using KLIMAX (Kernel- Level Infrastructure for Memory and execution profiling), Anomaly Based Detectors, Signature based detectors, Subnet Mask Scanners and Host Analyzers.

III. PROPOSED SYSTEM

The problem with the existing software and services available is that they are not bundled properly. The right combination of features and a documented source of reference are nowhere presented. Therefore we propose a complete encapsulated bundle of features and open source code to make integration of security modules much easier.

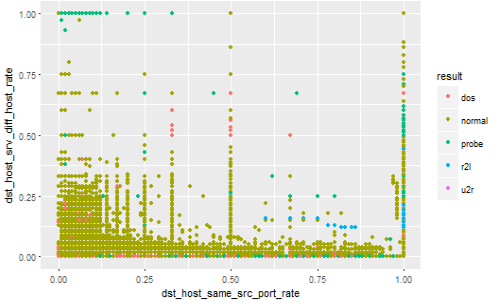
The design of one of the modules is shown in figures **Fig 3.1.**



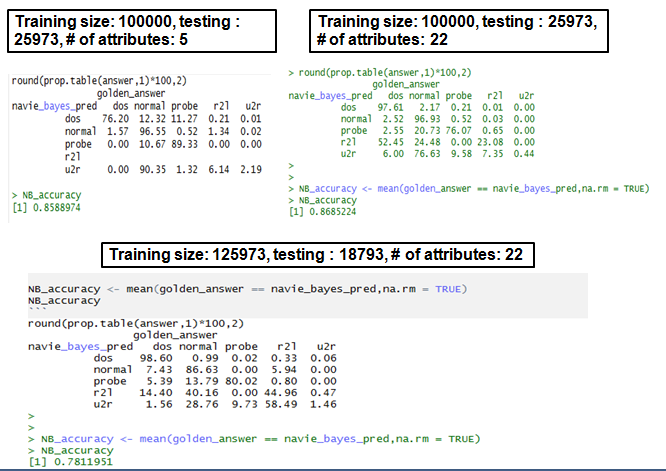
**Fid 3.1: Design of Keystroke Authentication**

Some key milestones in the architectural design:

* **Data and feature extraction:** In keystroke authentication it corresponds to the derivation of tuples taken from the user to train the ML models. In NID it maps to the labeling of data, attribute selection, zero-variance removal, dimensionality reduction, etc.
* **Pre-processing and metric selection:** This includes selection of statistical parameters like Manhattan, Mahalanobis, Euclidean, etc. The plotting of attributes for k-cross comparison to analyze the interdependencies between them is also done in this phase. Fig 3.2 shows one of the plots.
* **Building Classifiers (statistical and DM/ML models):** The heart of the product lies in this stage. This involves using multiple algorithms like Naïve Bayes, Decision Trees and Random Forests for NID and Manhattan mean, Euclidian mean and Manhattan median for Keystroke Authentication.
* **Accuracy Comparison and Analysis**: The idea of building multiple models is that one can overcome the drawbacks of the other, then ensembling it into one cumulative result. A set of accuracies arise due to the combination of number attributes, data size length, etc. One set of accuracies are as shown in Fig 3.3
* **Visualization and version control:** The last phase involves data visualization and handling multiple versions of the code and documentation.



**Fig 3.2: Plot to analyze interdependency between attributes**



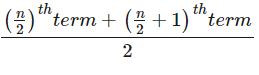
**Fig 3.3: Accuracy comparison for one of the models**

***3.1 Metrics for statistical models***

Distance measurements are in Manhattan and Euclidian statistic. Manhattan can be expressed as the modulus of the absolute difference between two points in space. Euclidian is the RMS distance of two points.

The arithmetic metrics used to train models is on mean and median. Equations (1) and (2) represent the formulae to calculate the mean and median of n points. The points that are being referred to are the data points for the training set. Assume that the space is made up of 6 data points each of them which would stand for a set of attribute array in our database. Now pick a data point a random and calculate the distance of this particular point from each of the rest of data points.

That would give the threshold that must exist for this data point. Now choose another data point and repeat this calculation. At the end we would end up with six different difference vectors. Now take the average of the vectors and conclude to a single point in space. This point is cumulative distance equivalent of all the vectors combined.

….(1) ….(2)

***3.2 Algorithmic flow for modified O-Auth and Anti-key logging***

The architecture is divided into client and server side protocols. The host will a set S of forms in N number of web pages. Only s subset s of S will need anti key logging. This information will be stored in a JSON document maintained at the server of the form

{ form\_name: [Name],

isAuthenticationRequirred:[True/False]

}

The steps flow as shown below:

i. For the set {S-s} proceed with the default mechanism.

ii. For the set s which requires Anti-Key logging, capture all the form elements and encrypt them using the standard AES encryption algorithm.

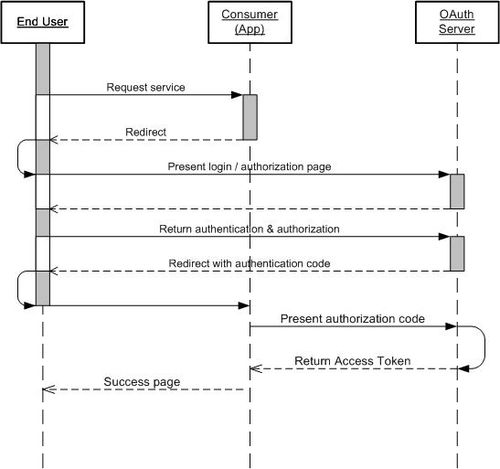
iii. The trick to the anti key logging is the key used in AES. Here, the key is dynamically generated by hashing a random r out of the R form elements.

iv. In OAuth terms a GET request is sent to the server, with a [client\_id, {subset of form elements}]. The server has a has function which replies with a hashCode.

v. This value is passed as the key to AES. This guarantees data integrity.

|  |
| --- |
|  |
|  |  |

Fig 3.4 shows the sequence diagram of the flow of communication.



**Fig 3.4: Sequence diagram for Anti-Keylogging and modified OAuth**

1. IMPLEMENTATION

The product follows a factory design pattern with very systematic modular oriented highly coherent features. It consists of 5 major modules namely:

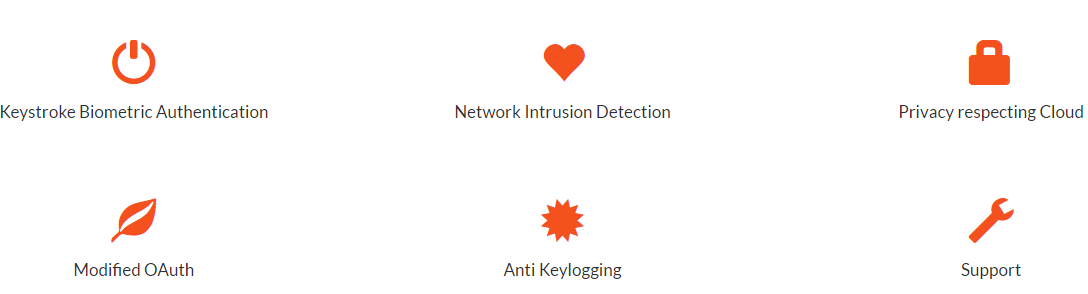
i. **DB module**: MySql, JSON in conjugation with PHP. Responsible for maintaining user profiles, keystroke timing features, .csv to data frame transformation, etc

ii. **UI module:** Involves data visualization for decision trees and plotting attributes, along with Dashboard for each feature, front-end JS validations, etc

iii. **Back-end scripts:** Profiling user information, connecting DB with front-end, asynchronous communication for GET, POST requests, etc

iv. **Machine Learning module:** The three classifiers for keystroke authentication belong in here. It also involves the Ensemble algorithm.

v. **Data Mining module:** The classifiers used in Network Intrusion Detection along with dimensionality reduction, pre-processing and building models.



**Fig 4.1: Services offered in the product**