Workshop Proceedings of

# ATIS 2011

Melbourne, November 9th, 2011.

*Second* **Applications and Techniques in Information Security Workshop**

**Edited by:**
**Matthew Warren**
**ISBN** 978-0-9872298-0-9

**Proceedings of**

# ATIS 2011

**Edited by**

Matthew Warren
ISBN 978-0-9872298-0-9

Published by the School of Information Systems, Deakin University, Melbourne, Victoria, 3125, Australia.

All papers published in the conference proceedings have been blind refereed by at least two of the ATIS 2011 **Organising** committee.

## Welcome

The ATIS 2011 workshop is the second workshop of its series and is being held at Deakin University, Burwood Campus, Melbourne, Australia. This workshop looks at the continued development of Information Security and related technologies taking into account the issues that impact everyone in a global context.

Members of the workshop organising committee accepted each paper in the proceedings after a careful review; this took the form of a **blind review** by at least **two** members of the workshop organising committee. The papers were subsequently reviewed and developed where appropriate; taking into accounts the comments of the reviewers. The aim of this conference is to further the work already achieved within Australia and bring together researchers in the field to discuss the latest security issues.

We commend the authors for their hard work and sharing their results, and the reviewers of the workshop for producing an excellent program.

*Contents*

**Paper 1: An Application of Novel Clustering Technique for Information Security**

Gleb Beliakov and Andrei Kelarev, School of Information Technology, Deakin University.

John Yearwood, Centre for Informatics and Applied Optimization School of Science, Information Technology and Engineering, University of Ballarat.

# An Application of Novel Clustering Technique for Information Security

Gleb Beliakov*, John Yearwood† and Andrei Kelarev*
*School of Information Technology
Deakin University, 221 Burwood Hwy, Burwood 3125, Australia
Email: {gleb,andrei}@deakin.edu.au
†Centre for Informatics and Applied Optimization
School of Science, Information Technology and Engineering
University of Ballarat, P.O. Box 663, Ballarat, Victoria 3353, Australia
Email: j.yearwood@ballarat.edu.au

***Abstract*—This article presents experimental results devoted to a new application of the novel clustering technique introduced by the authors recently. Our aim is to facilitate the application of robust and stable consensus functions in information security, where it is often necessary to process large data sets and monitor outcomes in real time, as it is required, for example, for intrusion detection. Here we concentrate on the particular case of application to profiling of phishing websites. First, we apply several independent clustering algorithms to a randomized sample of data to obtain independent initial clusterings. Silhouette index is used to determine the number of clusters. Second, we use a consensus function to combine these independent clusterings into one consensus clustering. Feature ranking is used to select a subset of features for the consensus function. Third, we train fast supervised classification algorithms on the resulting consensus clustering in order to enable them to process the whole large data set as well as new data. The precision and recall of classifiers at the final stage of this scheme are critical for effectiveness of the whole procedure. We investigated various combinations of three consensus functions, Cluster-Based Graph Formulation (CBGF), Hybrid Bipartite Graph Formulation (HBGF), and Instance-Based Graph Formulation (IBGF) and a variety of supervised classification algorithms. The best precision and recall have been obtained by the combination of the HBGF consensus function and the SMO classifier with the polynomial kernel.**

***Keywords*-consensus functions; clustering; classification; phishing websites**

## I. Introduction

This article deals with the experimental investigation of various combinations of consensus functions and supervised classification algorithms for the profiling of phishing websites. There are many clustering algorithms known in the literature. However, their outcomes depend on the random selection of initial seeds. Our approach has been designed to enable the application of consensus functions, since they can be used to increase stability and robustness of the obtained clusterings. The readers are referred to Section V for preliminaries and background information on consensus functions, see also [5] and [25] for additional references and examples of recent results.

The data sets in information security are very large and often require real-time monitoring, which is necessary, for example, for intrusion detection. This is why direct applications of sophisticated consensus functions in this area are computationally expensive. To overcome this difficulty, in [5] the authors have introduced a new approach combining consensus functions with fast supervised classification algorithms.

The present paper is devoted to experimental investigation of the effectiveness of this technique for the new application to profiling phishing websites. This application has not been considered before and belongs to the information security domain that has been actively investigated recently, as described by the Anti-Phishing Working Group [1] and OECD Task Force on Spam [16], see also [5] and [25]. We hope that the outcomes of our experiments can also provide guidance for choosing directions of future studies in other branches of information security.

This novel technique makes it possible to utilize slow and most reliable consensus functions at the initial stages to obtain more robust clusterings. On the other hand, it increases the speed of processing the whole large data set and new samples by incorporating fast classification algorithms in the final stage.

The paper is organised as follows. An outline of the combined approach to clustering is given in Section II. Section III is devoted to the preprocessing of data and extraction of features for clustering algorithms. Section IV contains a brief outline of clustering algorithms applied to obtain an initial clustering ensemble for a small randomized sample of the data set. Section V contains background information on consensus functions and concise preliminaries on graph formulations and heuristics used to combine the ensemble into one final consensus clustering. Section VI deals with the supervised classification algorithms trained on the consensus clustering. Experimental results are summarized in Section VII. Section VIII presents the conclusions.

## II. Outline of the Combined Approach to Clustering

We investigated various combinations of consensus functions in conjunction with fast supervised classification algorithms. This approach to clustering has several steps or stages. First, a variety of independent clustering algorithms are applied to a randomized sample of the data. Second, a consensus function is used to combine these independent clusterings into one common consensus clustering. (In fact, we investigated the effectiveness of several consensus functions in this scheme.) Third, the consensus clustering of the randomized sample is used as a training set to train fast supervised classification algorithms. Finally, these fast classification algorithms can be applied to classify the whole large data set.

Our experiments investigated this approach for the particular case of profiling of phishing websites. Algorithms for classification and clustering of phishing emails and websites have been actively investigated recently, as discussed for example, by the Anti-Phishing Working Group [1] and OECD Task Force on Spam [16], see also [5] and [25] for examples of recent results. Phishing usually involves acts of social engineering attempting to extract confidential details by sending emails with false explanations urging users to provide private information that will be used for identity theft. The users are then directed to a phishing website, where they are asked to enter personal details, such as credit card numbers, tax file numbers, bank account numbers and passwords. For more comprehensive information on phishing the readers are referred to the Anti-Phishing Working Group [1] and OECD Task Force on Spam [16].

## III. Feature Extraction

A flexible preprocessing and feature extraction system has been implemented in Python for the purposes of this investigation. It has been used to extract features describing the content and html structure of the websites.

We used the *term frequency–inverse document frequency* word weights, or TF-IDF weights, to select features for the clusterings. These weights are well known in text categorization, see [24]. They are defined using the following concepts and notation. Suppose that we are extracting features from a data set $E$, which consists of $|E|$ websites. For a word $w$ and a website $m$, let $\mathrm{N}(w,m)$ be the number of times $w$ occurs in $m$. Suppose that a collection $T = \{t_1, \ldots, t_k\}$ of terms $t_1, \ldots, t_k$ is being looked at. The *term frequency* of a word $w \in T$ in a website $m$ is denoted by $\mathrm{TF}(w,m)$ and is defined as the number of times $w$ occurs in $m$, normalized over the number of occurrences of all terms in $m$:

$$\mathrm{TF}(w,m) = \frac{\mathrm{N}(w,m)}{\sum_{i=1}^{k} N(t_i,m)}. \tag{1}$$

The *document frequency* of the word $w$ is denoted by $\mathrm{DF}(w)$ and is defined as the number of websites in the given data set where the word $w$ occurs at least once. The *inverse document frequency* is used to measure the significance of each term. It is denoted by $\mathrm{IDF}(w)$ and is defined by the following formula

$$\mathrm{IDF}(w) = \log\left(\frac{|E|}{\mathrm{DF}(w)}\right). \tag{2}$$

The *term frequency–inverse document frequency* of a word $w$ in website $m$, or TF-IDF weight of $w$ in $m$ is defined by

$$\text{TF-IDF}(w,m) = \mathrm{TF}(w,m) \times \mathrm{IDF}(w). \tag{3}$$

We collected a set of words with highest TF-IDF scores in all websites of the data set. For each website, the TF-IDF scores of these words in the website were determined. These weights and additional features were assembled in a vector. In order to determine the TF-IDF scores we used Gensim, a Python and NumPy package for vector space modelling of text documents. In addition, we used features reflecting the html structure of the websites and links to different URL domains or numeric IP addresses.

We have also included several features related to the structure of the websites, the number and quality of images, hidden fields or graphics, and properties of the links on the web page, full HTML substitution in the links, inline embedding of scripting content, unicode-obfuscated URLs, loading external scripting code, URLs beginning with IP addresses in links, and other html obfuscating techniques.

These features were assembled in an algebraic vector space model representing the data set. A number of independent initial clusterings were then obtained for the feature vectors of the websites in the sample using the following clustering algorithms.

## IV. Independent Initial Clusterings

Looking at the features extracted as described in Section III, we used four clustering algorithms implemented in WEKA, SimpleKMeans, Cobweb, EM and FarthestFirst, and obtained an ensemble of independent initial clusterings $C = \{C^{(1)}, C^{(2)}, \ldots, C^{(k)}\}$, where, for each clustering $C^{(i)}$, the whole data set $D$ is a disjoint union of the classes in this clustering so that

$$C^{(i)} = \{C_1^{(i)}, C_2^{(i)}, \ldots, C_{k_i}^{(i)}\} \text{ and} \tag{4}$$

$$D = C_1^{(i)} \dot\cup C_2^{(i)} \dot\cup \ldots \dot\cup C_{k_i}^{(i)}, \tag{5}$$

for all $i = 1, \ldots, k$. We refer to [12], Section 3.3.2, and [22], Section 4.8, for more details on these algorithms. Their outcomes in WEKA depend on the value of the input parameter "seed". To overcome the dependence of the outcome on the random choice of this parameter we run each algorithm with 10 random selections of the "seed", as recommended in [13].

Cobweb, EM, FarthestFirst and SimpleKMeans produce clusterings given a fixed number of clusters as an input parameter. In order to determine the appropriate number

of clusters we used Silhouette index. The *Silhouette index* of a clustering is a robust measure of the quality of the clustering introduced in [18]. The Silhouette index $\mathrm{SI}(x)$ of each observation $x$ is defined as follows. If $x$ is the only point in its cluster, then $\mathrm{SI}(x) = 0$. Denote by $a(x)$ the average distance between $x$ and all other points of its cluster. For any other cluster $C$, let $d(x, C)$ be the average distance between $x$ and all points of $C$. The minimum

$$b(x) = \min\{d(x, C) \,:\, x \notin C\} \tag{6}$$

is the distance from $x$ to its nearest cluster $C$ to which $x$ does not belong. Finally, put

$$\mathrm{SI}(x) = \frac{b(x) - a(x)}{\max\{a(x), b(x)\}} \tag{7}$$

The Silhouette index of the whole clustering is found as the average index over all observations. The Silhouette index always belongs to $[-1, 1]$. The partition with highest Silhouette index is regarded as optimal.

For each initial clustering algorithm and each value of the "seed", we repeated it several times increasing the number of clusters, as recommended in [18]. The clustering with the best Silhouette index was included in the set of initial clusterings to be processed by consensus clustering algorithm at the next stage. The same procedure of determining the number of clusters was applied for other initial clustering algorithms too.

All these initial clustering algorithms can process our data without any additional data transformations or encoding. The outcomes of all of these clustering algorithms often depend on the initial random selections made during the start of their iterations. A standard method is to run them for several random selections of initial parameters, as in [13]. In WEKA, the outcomes of these algorithms depend on their input parameter "seed". We run each of these algorithms for 10 random selections of the "seed" and obtained a total of 40 initial clusterings. This provided sufficient input for the consensus clustering algorithms considered in the next section. Thus, we have used multiple start versions of the Cobweb, EM, FarthestFirst and SimpleKMeans, which could process our sample directly and produced sufficient input for the next stage of our approach.

## V. Consensus Functions for Ensemble Clusterings

The following three consensus functions have been applied:

CBGF – Cluster-Based Graph Formulation,
HBGF – Hybrid Bipartite Graph Formulation,
IBGF – Instance-Based Graph Formulation

The final consensus clustering obtained by these consensus functions was used to train fast classification algorithms. It is therefore natural to assume that a consensus function performs better in our scheme, if the supervised classification algorithms are able to produce higher precision and recall at the final stage.

*Cluster-Based Graph Formulation*, CBGF, is a graph-based consensus function. It defines a complete weighted undirected graph on the set of vertices consisting of all the given clusters. The weight of each edge of this graph is determined by a measure of similarity of the clusters corresponding to the vertices. Namely, for two clusters $C'$ and $C''$ the weight of the edge $(C', C'')$ can be set equal to

$$w((C', C'')) = \frac{|C' \cap C''|}{|C' \cup C''|}, \tag{8}$$

known as the *Jaccard index* or *Jaccard similarity coefficient*, see [20], Chapter 2. In order to ensure that clusters that have a lot of elements in common are grouped together, the edges with lowest weights are then eliminated by applying a graph partitioning algorithm. Each element is then allocated to the new final cluster where it occurs most frequently.

*Hybrid Bipartite Graph Formulation*, HBGF, is a consensus function proposed in [8] and based on a bipartite graph. It has two sets of vertices: clusters and elements of the data set. A cluster $C$ and an element $d$ are connected by an edge in this bipartite graph if and only if $d$ belongs to $C$. An appropriate graph partitioning algorithm is then applied to the whole bipartite graph. The final clustering is determined by the way it partitions all elements of the data set. We refer to [8], [19], [21] for more details.

*Instance-Based Graph Formulation*, IBGF, is also a consensus function based on a complete undirected weighted graph. Vertices of the graph are all elements of the data set. The edge $(d', d'')$ has weight given by the formula

$$w((d', d'')) \;=\; \sum_{i=1,\ldots,k;\; C_i(d')=C_i(d'')} 1/k,$$

where $C_i(x)$ stands for the cluster containing $x$ in the $i$-th clustering. This means that $w((d', d''))$ is the proportion of clusterings where the clusters of $d'$ and $d''$ coincide. Then IBGF applies an appropriate graph partitioning algorithm to divide the graph into classes. These classes determine clusters of the final consensus clustering.

We used METIS graph partitioning software described in [14]. The weights of edges in the input files of METIS must all be strictly greater than zero, which means that it can handle only complete weighted graphs. In order to apply it to a bipartite graphs, we had to set the weights of all edges not present in the graph to 1 and to rescale the weight of all other edges by multiplying them with a constant to make them larger than 10,000. This ensured that METIS removed all nonexistent edges from the graph and then continued analysing the resulting bipartite graph.

We used *feature ranking* to select the most essential features for use in consensus functions. It ranks all features with respect to their relevance and importance to the problem. We investigated the well-known and widely used measures: the Rank Correlation Coefficient, RCC. It assesses how well the relationship can be described using a monotonic function, as explained in the book [15]. The Rank Correlation Coefficient $\varrho$ is a measure of association based on the ranks of the data values. It is given by the formula

$$\varrho = \frac{\sum(R_i - \overline{R})(S_i - \overline{S})}{\sqrt{\sum(R_i - \overline{R})\sum(S_i - \overline{S})}}, \qquad (9)$$

where $R_i$ is the rank of the $i$-th $x$-value, $S_i$ is the rank of the $i$-th $y$-value, $\overline{R}$ is the mean of the ranks of $x$-values, and $\overline{S}$ is the mean of the ranks of $y$-values. The values of $\varrho$ belong to the segment $[-1; 1]$. Values close to 1 indicate that there is a good correlation (described by a monotonically increasing function). In our experiments better results have been obtained using the Rank Correlation Coefficient, and further we include only tables with the results based on this coefficient. First, we obtained initial clusterings for the small randomized sample and all original features as described in Section IV. Then we used these initial clusterings to find the Rank Correlation Coefficients. For each numerical feature, we numbered all clusters according to the mean value of this feature for all instances of the cluster, and after that ranked all values of the feature and the cluster numbers. Numbering clusters in the order of the mean values of the feature for all instances of each cluster is essential, since it ensures that we only have to look at values of RCC close to 1 in our case. Having found the RCC for each feature, we ordered the original features by the values of their Rank Correlation Coefficients. The features with higher values were selected for the next stages of our procedure.

We ranked all the preliminary variables according to the values of their ranked correlation coefficients. Different testing data sets or clustering algorithms will produce different ranking lists of the preliminary variables. The principle is that, the higher the ranking of the feature, the more relevant it is to the clustering result. This means that not all of the features make the same contribution to the clustering result. The least important features can be regarded as redundant features and can be removed. The quality of the clusters can be improved by eliminating the influence of the redundant features, and the efficiency of clustering algorithm can be increased by reducing dimensionality and removing irrelevant features.

In order to determine the appropriate number of clusters for the final consensus clustering we used Silhouette index described in Section IV. We ran each consensus clustering increasing the number of clusters from 2 to 30. The final consensus clustering with the best Silhouette index was then regarded as the final output of the whole process.

## VI. Supervised Classification Algorithms

We have compared the performance of these three consensus functions and their combinations with several supervised classification algorithms. The resulting consensus clustering described in Section V was used to train supervised classification algorithms. We investigated the performance of all classifiers implemented in WEKA, and have included in the tables of this paper the outcomes of the following algorithms, which worked well in our scheme: AdaBoostM1, BayesNet, DecisionTable, IBk, J48, JRip, HyperPipes, LibLINEAR, LibSVM, NaiveBayes, PART, RBFNetwork, Ridor, SMO and VFI. More information on these algorithms is given by [3], [4], [6], [7], [9], [22], [23].

The performance of the SMO, LibSVM and LibLINEAR depends on the SVM type, the kernel and several numerical parameters. For each of them, we used the optimization procedure explained in [11]. More advanced optimization techniques presented in [2] can also be applied here.

## VII. Experimental results

We have carried out experimental investigation of all combinations of the CBGF, HBGF and IBGF consensus functions and classification algorithms listed above for a sample of 1024 websites randomly selected from a very large data set supplied by the industry partners of the Centre for Informatics and Applied Optimization at the University of Ballarat. Analogous data sets are available to all researchers from the downloadable databases at the PhishTank [17]. We used tenfold cross validation to evaluate the weighted average precision and recall of these classification algorithms comparing them with the classes of the corresponding consensus clustering.

The results of our experiments are summarized in Tables I, II, III and IV. The precision and recall for all choices of kernels of the SMO, LibSVM and LibLINEAR classifiers are assembled in Tables I and II. Their best results have been also included in Tables III and IV for convenience of the readers. The outcomes show that the combination of HBGF consensus function and the SMO classifier with the polynomial kernel achieved the best precision and recall in this scheme.

## VIII. Conclusion

This article investigated a new application of our novel approach to clustering for profiling phishing websites. Our method is based on combining reliable consensus functions with fast supervised classification algorithms.

Our experiments compared the effectiveness of CBGF, HBGF and IBGF consensus functions in conjunction with various classification algorithms. The experimental results have shown that the combination of HBGF consensus function and the SMO classifier with the polynomial kernel achieved the best precision and recall in this scheme and can be recommended for the future implementations.

Table I
PRECISION OF SMO, LIBSVM AND LIBLINEAR

| | CBGF | HBGF | IBGF |
|---|---|---|---|
| SMO | | | |
| - normalized polynomial | 84.770 | 91.331 | 88.101 |
| - polynomial kernel | 87.441 | 94.759 | 90.726 |
| - Pearson universal | 84.236 | 91.936 | 87.557 |
| - RBFKernel | 76.766 | 82.844 | 79.028 |
| LibSVM C-SVC | | | |
| - linear kernel | 68.173 | 74.573 | 71.797 |
| - polynomial kernel | 69.648 | 74.671 | 72.252 |
| - radial basis function | 65.949 | 71.525 | 68.961 |
| - sigmoid kernel | 2.506 | 2.042 | 1.883 |
| LibSVM nu-SVC | | | |
| - linear kernel | 67.515 | 72.831 | 70.459 |
| - polynomial kernel | 60.460 | 65.254 | 63.405 |
| - radial basis function | 55.358 | 60.206 | 57.913 |
| - sigmoid kernel | 2.290 | 2.113 | 2.591 |
| LibLINEAR | | | |
| - L2 loss svm (dual) | 40.040 | 43.750 | 41.823 |
| - L1 loss svm (dual) | 41.113 | 44.675 | 42.891 |
| - multi-class svm | 61.012 | 65.286 | 63.267 |

Table II
RECALL OF SMO, LIBSVM AND LIBLINEAR

| | CBGF | HBGF | IBGF |
|---|---|---|---|
| SMO | | | |
| - normalized polynomial | 84.763 | 91.326 | 88.096 |
| - polynomial kernel | 87.443 | 94.760 | 90.727 |
| - Pearson universal | 84.222 | 91.921 | 87.542 |
| - RBFKernel | 76.707 | 82.783 | 78.966 |
| LibSVM C-SVC | | | |
| - linear kernel | 68.185 | 74.585 | 71.806 |
| - polynomial kernel | 69.608 | 74.632 | 72.216 |
| - radial basis function | 65.697 | 71.270 | 68.709 |
| - sigmoid kernel | 2.624 | 2.159 | 2.000 |
| LibSVM nu-SVC | | | |
| - linear kernel | 67.494 | 72.809 | 70.439 |
| - polynomial kernel | 60.534 | 65.329 | 63.477 |
| - radial basis function | 55.219 | 60.064 | 57.772 |
| - sigmoid kernel | 2.404 | 2.229 | 2.709 |
| LibLINEAR | | | |
| - L2 loss svm (dual) | 39.995 | 43.703 | 41.779 |
| - L1 loss svm (dual) | 41.041 | 44.606 | 42.819 |
| - multi-class svm | 61.004 | 65.274 | 63.259 |

Table III
PRECISION OF CLASSIFIERS WITH CBGF, HBGF, IBGF

| | CBGF | HBGF | IBGF |
|---|---|---|---|
| AdaBoostM1 | 47.036 | 51.142 | 49.385 |
| BayesNet | 73.952 | 81.112 | 77.297 |
| DecisionTable | 67.505 | 73.394 | 70.635 |
| IBk | 80.792 | 87.195 | 83.775 |
| J48 | 76.266 | 83.397 | 79.640 |
| JRip | 74.195 | 81.131 | 77.477 |
| HyperPipes | 55.380 | 59.686 | 57.198 |
| LibLINEAR | 61.012 | 65.286 | 63.267 |
| LibSVM | 69.648 | 74.671 | 72.252 |
| NaiveBayes | 66.115 | 71.053 | 68.312 |
| PART | 73.584 | 79.476 | 76.201 |
| RBFNetwork | 58.726 | 63.716 | 61.639 |
| Ridor | 64.307 | 69.406 | 66.633 |
| SMO | 87.441 | **94.759** | 90.726 |
| VFI | 61.678 | 66.935 | 64.590 |


## ACKNOWLEDGEMENTS

The authors are grateful to three referees for thorough reports with comments and corrections, which have helped to improve the text of this article.


## REFERENCES


[1] Anti-Phishing Working Group, APWG, http://apwg.org/, viewed 20 September 2011.

[2] G. Beliakov and J. Ugon. Implementation of novel methods of global and non-smooth optimization: GANSO programming library, *Optimization*, 56 (2007), 543–546.

[3] R.R. Bouckaert, E. Frank, M. Hall, R. Kirkby, P. Reutemann, A. Seewald and D. Scuse, WEKA Manual for Version 3-7-3, http://www.cs.waikato.ac.nz/ml/weka/, viewed 15 August 2011.

[4] C.-C. Chang and C.-J. Lin, LIBSVM – A Library for Support Vector Machines, 2001. Software available at http://www.csie.ntu.edu.tw/~cjlin/libsvm/, viewed 12 June 2011.

[5] R. Dazeley, J.L. Yearwood, B.H. Kang and A.V. Kelarev, Consensus clustering and supervised classification for profiling phishing emails in internet commerce security, In: Knowledge Management and Acquisition for Smart Systems and Services, PKAW2010, *Lecture Notes in Computer Science* 6232 (2010), 235–246.

Table IV
RECALL OF CLASSIFIERS WITH CBGF, HBGF, IBGF

| | CBGF | HBGF | IBGF |
|---|---|---|---|
| AdaBoostM1 | 46.981 | 51.085 | 49.330 |
| BayesNet | 73.941 | 81.103 | 77.287 |
| DecisionTable | 67.505 | 73.392 | 70.633 |
| IBk | 80.795 | 87.196 | 83.774 |
| J48 | 76.269 | 83.397 | 79.644 |
| JRip | 74.195 | 81.131 | 77.476 |
| HyperPipes | 55.298 | 59.605 | 57.121 |
| LibLINEAR | 61.004 | 65.274 | 63.259 |
| LibSVM | 69.608 | 74.632 | 72.216 |
| NaiveBayes | 66.110 | 71.046 | 68.307 |
| PART | 73.582 | 79.473 | 76.195 |
| RBFNetwork | 58.726 | 63.715 | 61.641 |
| Ridor | 64.312 | 69.406 | 66.638 |
| SMO | 87.443 | **94.760** | 90.727 |
| VFI | 61.660 | 66.918 | 64.570 |


[6] R.O. Duda, P.E. Hart and D.G. Stork, *Pattern Classification*, 2nd Edition, Wiley-Interscience, 2001.

[7] R.-E. Fan, K.-W. Chang, C.-J. Hsieh, X.-R. Wang and C.-J. Lin, LIBLINEAR - a library for large linear classification. Software available at http://www.csie.ntu.edu.tw/~cjlin/liblinear/, viewed 10 August 2011.

[8] X. Fern, C. Brodley, Solving cluster ensemble problems by bipartite graph partitioning, In: 21st International Conference on Machine Learning, ICML'04, Vol. 69, ACM, New York, NY, USA, 2004, pp. 36–43.

[9] E. Frank and I. H. Witten, Generating accurate rule sets without global optimization. In: *15th Internat. Conf. on Machine Learning*, 144–151, 1998.

[10] Y. Hong, S. Kwong, Y. Chang and Q. Ren, Consensus unsupervised feature ranking from multiple views, *Pattern Recognition Letters*, 29 (2008), 595–602.

[11] C.-W. Hsu, C.-C. Chang and C.-J. Lin, A practical guide to support vector classification, Dept. Computer Science, National Taiwan University, `http://www.csie.ntu.edu.tw/~cjlin` (Initial version: 2003, last updated: April 15, 2010).

[12] A. Jain and R. Dubes, *Algorithms for Clustering Data*, Upper Saddle River, NJ, USA: Prentice-Hall, 1988.

[13] A. Jain, M. Murty and P. Flynn, Data clustering: a review, *ACM Computing Surveys*, 31 (1999), 264–323.

[14] G. Karypis and V. Kumar, Metis: A software package for partitioning unstructured graphs, partitioning meshes, and computing fill-reducing orderings of sparse matrices, Technical report, University of Minnesota, Department of Computer Science and Engineering, Army HPC Research Centre, Minneapolis (1998).

[15] M.G. Kendall and J.D. Gibbons, *Rank Correlation Methods*, 5th Edition, London: Oxford University Press, 1990.

[16] Organisation for Economic Cooperation and Development, OECD task force on spam, OECD anti-spam toolkit and its annexes, http://www.oecd.org/dataoecd/63/28/36494147.pdf, viewed 12 August 2011.

[17] PhishTank, Developer information, http://www.phishtank.com/developer_info.php (viewed 20 September 2011).

[18] P. Rousseeuw, Silhouettes: a graphical aid to the interpretation and validation of cluster analysis, *J. Comp. Appl. Math.*, 20 (1987), 53–65.

[19] A. Strehl, J. Ghosh, Cluster ensembles – a knowledge reuse framework for combining multiple partitions, *J. Machine Learning Research*, 3 (2002), 583–617.

[20] P. Tan, M. Steinbach and V. Kumar, *Introduction to Data Mining*, Boston, MA, USA: Addison-Wesley, 2005.

[21] A. Topchy, A. Jain and W. Punch, Combining multiple weak clusterings, in: IEEE International Conference on Data Mining, 2003, pp. 331–338.

[22] I. Witten, E. Frank, *Data Mining: Practical Machine Learning Tools and Techniques*, Amsterdam: Elsevier/Morgan Kaufman, 2005.

[23] X. Wu, V. Kumar, J.R. Quinlan, J. Ghosh, Q. Yang, H., Motoda, G.J. McLachlan, A. Ng, B. Liu, P.S. Yu, Z.H. Zhou, M. Steinbach, D.J. Hand, D. Steinberg, Top 10 algorithms in data mining. *Knowledge Inf. Systems*, 14 (2007)(1), 1–37.

[24] J.L. Yearwood and M. Mammadov, *Classification Technologies: Optimization Approaches to Short Text Categorization*, IGI Global, 2010.

[25] J. Yearwood, D. Webb, L. Ma, P. Vamplew, B. Ofoghi and A. Kelarev, Applying clustering and ensemble clustering approaches to phishing profiling, *Data Mining and Analytics 2009*, Proc. 8th Australasian Data Mining Conference: AusDM 2009, (1-4 December 2009, Melbourne, Australia) CRPIT, Vol.101, pp. 25–34.

## Paper 2: Virtual World Security Inspection

Nicholas C. Patterson and Michael Hobbs, School of Information Technology, Deakin University.

# Virtual World Security Inspection

## A software inspection process carried out on popular virtual world environments

Nicholas C. Patterson
School of Information Technology
Deakin University
Pigdons Rd, Waurn Ponds 3215, Australia
ncp@deakin.edu.au

Michael Hobbs
School of Information Technology
Deakin University
Pigdons Rd, Waurn Ponds 3215, Australia
mick@deakin.edu.au

*Abstract*—**Virtual property theft is a serious problem that exists in virtual worlds. Legitimate users of these worlds invest considerable amounts of time, effort and real-world money into obtaining virtual property, but unfortunately, are becoming victims of theft in high numbers. It is reported that there are over 1 billion registered users of virtual worlds containing virtual property items worth an estimated US$50 billion dollars. The problem of virtual property theft is complex, involving many legal, social and technological issues. The software used to access virtual worlds is of great importance as they form the primary interface to these worlds and as such the primary interface to conduct virtual property theft. The security vulnerabilities of virtual world applications have not, to date, been examined. This study aims to use the process of software inspection to discover security vulnerabilities that may exist within virtual world software – vulnerabilities that enable virtual property theft to occur. Analyzing three well know virtual world applications *World of Warcraft*, *Guild Wars* and *Entropia Universe*, this research utilized security analysis tools and scenario testing with focus on authentication, trading, intruder detection and virtual property recovery. It was discovered that all three examples were susceptible to keylogging, mail and direct trade methods were the most likely method for transferring stolen items, intrusion detection is of critical concern to all VWEs tested, stolen items were unable to be recovered in all cases and lastly occurrences of theft were undetectable in all cases. The results gained in this study present the key problem areas which need to be addressed to improve security and reduce the occurrence of virtual property theft.**

***Keywords: virtual worlds, virtual property theft, real money trading, keylogging, vulnerability, software inspection***

### I. INTRODUCTION

Virtual World Environments (VWEs) are computing simulation environments that allow users to socialize, play, compete and even work in an immersive on-line virtual world. VWEs have their heritage in the text-based multi-user computer games (MUDs) of the 1980s [1], while modern versions are commonly visually rich 3D, extensive environments that range from fantasy and space based realms, to life-like real world environments. The number of people actively participating in these environments has grown dramatically over recent years and current reports indicate the number of registered virtual world users exceed 1 billion world-wide [1]. It is common for users to pay a subscription fee to access these worlds and then, over a period of time, through completing various tasks are able to collect items that are owned by the VWE character, representing the player. It is also possible in many VWEs for users to spend real-world money to purchase items as well. The investment made by users in terms of their time, effort and real-world money, places a value on these virtual property items, which can then be traded with or sold to other users for either virtual-world or real-world currency. Virtual worlds expert Marcus Eikenberry estimates the market value of virtual property as high as US$50 billion dollars [2].

The ability to convert virtual property into real-world money has enabled the rise of a serious problem faced by many in VWEs, that of virtual property theft (VPT) [3]. The problem of VPT is complex as it envelopes many diverse areas, such as *legal issues* – lack of laws to support prosecution, especially in cases that span international borders; *social issues* – such as identity theft and harassment; and *technological issues* – the appropriate use of security methods and tools (within the software used to access VWEs) to protect resources.

Users access VWEs through software running on their computers. Commonly, this is in the form of a client-application that connects to a remote server application holding the data associated with the VWE. Although security in client-server based and other forms of distributed system applications has been researched extensively, to date, there has been no research on the security aspects of VWE software. It is this software (technology) that is used to access VWEs, and thus, it is important to understand the security vulnerabilities of such software. Having information on potential security vulnerabilities will help identify approaches needed to address the problem of VPT. This information can be used to form recommendations to VWE developers on how to improve their software and ultimately providing users with an appropriate level of trust to actively participate in VWEs.

The goal of this paper is to discover what vulnerabilities and threats may exist in software used to access popular VWEs that enable the problem of virtual property theft (VPT). One method to discover these vulnerabilities would be to analyze and test the actual source code for the VWEs. Due to the commercial nature of VWEs, companies will not provide the source code for research by an independent party; therefore vulnerabilities must be identified through an external examination of the executable VWE software. Therefore, to achieve our goal an operational software inspection technique [4] is employed as a method to externally assess the quality of software (in this case VWEs) and to reduce the number of defects (security vulnerabilities).

There are many hundreds of VWEs that exist and since it is not feasible to analyze all, a representative selection of three VWEs was made. These included: *World of Warcraft*, *Guild Wars* and *Entropia Universe* and were chosen based on their popularity among player (number of registered users), length of time they have been active, known issues with VPT, and examples of real money trading (RMT) of virtual property [5].

The inspection of the VWE software involves discovering functional and design problems that may exist related to VPT. These relate to issues including: authentication, virtual property trading and recovery. As far as can be discovered from published literature an inspection process of this nature has not been conducted on a collection of software of this nature. The results gained from this study can be used to identify problem areas that exist and the factors that cause them. This study provides a foundation to the development of a solution to the problem of VPT. It is envisaged that a solution to VPT could be incorporated into current and future virtual world software.

This paper is structured as follows. Section II provides background on the area of software inspection approaches and requirements. Section III present and details the three VWE selected for inspection, while Section IV describes inspection process and environment used. The set of inspection categories and objectives are presented in Section V. The inspection results are tabled and analyzed in Section VI. Section VII provides conclusions on this work.

## II. RELATED WORK

The general security issues associated with VWE have been examined in the past. This work has focused primarily on the game-play and social effects that cheating and fraud have on users of such systems. These issues and their effects on the virtual economy within VWEs was discussed by Cikic et al. [6]. The security vulnerabilities of existing VWE client software has not been a focus of current research.

Livshits and Lam [7] conducted an analysis of nine popular open source applications, they used a method of static analysis to perform their inspection and testing. They found that there were a total of 41 potential security violations in the nine benchmarks and 29 of those turned out to be security errors and 12 were false positives. However this study did not focus on VWEs specifically but provides support that this technique is applicable to the testing of VWEs.

A study to test the resilience of commercial virus scanning software packages was conducted by Christodorescu and Jha [8]. The aim of this study was to present architecture for detecting malicious patterns in executable files that are resilient to code-obfuscation attacks. To determine if an executable was resilient or not, they performed tests against three commercial virus scanners, their results showed that a combination of nop-insertion and code transportation was all that was required to render a malicious executable undetectable by these virus scanners [8]. This study was useful from a security analysis point of view however did not cover VWEs specifically.

Hole et al. [9] conducted a study on Norwegian internet banks from 2003 to 2004. Their aim was to determine if a false sense of security existed within bank customers and whether this contributes to an additional security risk in using online banking. They discovered customer authentication methods in many Norwegian Internet banks were weak, which allows simple but powerful attacks possible. This study presented an example of a successful attack that involved a PIN calculator (that was used by many Norwegian banks to generate new PINs for customers – based on certain period of time). In this attack it is possible to generate a timeline to associate a PIN number with a certain time interval and then employ brute force search to access customer accounts. This relates to our study in that one form of security may lead to users having a false sense of security.

From existing literature, it appears that no work has been conducted on the security vulnerabilities of the VWE client software. A study of this nature would assist in determining the potential for security problem to exist within VWE software. In doing so help identify if such flaws are common with other software testing results and potentially showing that no software package can be completely secure.

## III. VIRTUAL WORLD ENVIRONMENT CHOICES

There exists many hundreds of VWEs available to the public. The choice of VWEs for this study was based upon their popularity in terms of users (relevancy in terms of are they a good representation of the many available); does the VWE have virtual property available to users, which can be traded or potentially stolen. The three VWEs selected to be inspected for this study included: *World of Warcraft* [10], *Guild Wars* [11] and *Entropia Universe* [12]. They are online VWEs for personal computer (PC) compatible computers.

### A. World of Warcraf (WoW)

*World of Warcraft* (WoW) [10] is an online role playing VWE released for personal computers in 2004 with a user base consisting of 10's of millions of users worldwide. As of late 2010 WoW had over 12 million subscribers – when they launched their second expansion named Wrath of the Lich King [13]. This VWE utilizes a subscription based model where users pay approximately US$15 a month in order to access the world. This VWE uses a client-server based model; where the user through a client interface application executing on their personal computer connects to a WoW server over the internet. In this VWE users can take on the role of a fantasy based character, through which they can explore and quest across a large virtual world. *World of Warcraft* can allow thousands of users to interact with each other in the same virtual world. Users can form relationships with other users and compete against each other for virtual currency or virtual property such as armor or weapons.

### B. Guild Wars (GW)

*Guild Wars* (GW) [11] is also an online role playing environment which was released on the personal computer in 2005. In 2009 this VWE had sold over 6 million copies of the client software [14]. This VWE uses a client-server based model; where the user will utilize a client interface on their personal computer and then will connect to the server over the internet. A user is required to purchase the software but can play for free with no monthly subscription fee. *Guild Wars* is popular as it takes all the best aspects of other online games and combines them into a mission based design. *Guild Wars*

allows users to create a fantasy based character in a virtual world and supports cooperative play. It also allows users to compete against each other for virtual currency and virtual property.

### C. *Entropia Universe (EU)*

*Entropia Universe* (EU) [12] is a VWE that was designed by the company MindArk for the personal computer. It has grown to more than 1,000,000 registered accounts from over 200 countries or territories [15]. This VWE uses a client-server based model; where the user will utilize a client interface on their personal computer and then will connect to the server over the internet. The key reason for selecting this VWE is that the virtual economy is backed by real world money. It is currently the only VWE with a true Real Cash Economy (RCE) [15]. *Entropia Universe* employs a micropayment business model where players can play in the world for free but the company allows users to buy virtual currency, called Project Entropia Dollars or PED, which can then be traded back to the company for real world money. In *Entropia Universe* this means that virtual property and virtual currency has real world value, allowing users at any time to 'cash out' of the VWE. The better the user is at collecting virtual property in the world, the more money they can make outside of it. This VWE is an excellent choice for software inspection purposes since virtual property in this VWE has distinct value and many virtual property transaction occur in the VWE but also many real world transactions occur, in terms of 'cash-outs'.

## IV. SOFTWARE INSPECTION PROCESS

The software inspection process has been used extensively as a common process for debugging and improving source code quality [16]. A related method is that of usability inspection, which is a popular way to evaluate user interfaces [16].

The method of inspection used in this paper is a combination of software and usability inspection techniques. The software inspection process used in this study will be that of *operational software inspection*, a process that typically involves a group of individuals. In this instance one individual (the first author) is used to examine the software to find defects or security vulnerabilities, which are often the result of one or many design or operational faults.

The test environment used for this operational software inspection process took place in the School of IT at Deakin University (Waurn Ponds campus). The test computer system (PC based system running a default Deakin University installation of Windows XP) is located on this campus, utilizing the universities network connection to the internet.

The VWEs being inspected were installed into the ~/Program Files/ directory on the test computer. Additional software based tools that were used for testing included: Sniphere, Wireshark and Actual Keylogger (discussed more in Section 5).

For inspection purposes no security measures will be in place apart from those inbuilt into the VWE's being inspected. All firewalls and antivirus software will be disabled on the inspection computer. This is to try and mimic the greater population of personal computers running VWE clients, where some may have security and some may not.

This operational software inspection consists of a clearly defined agenda and set of requirements for the inspection process. The set of inspection categories and the sets of tools utilized are presented next.

## V. INSPECTION CATEGORIES

This section will outline the categories of testing that will occur on the selected VWEs. These categories were chosen as a result of a literature review conducted in the overall research project and determined as the most crucial areas of concern that relate to VPT and how it occurs. Firstly authentication will be focused on, as this is the users primary method of gaining access to a VWE, and in order to conduct theft; thieves have to break the authentication to gain access to victims accounts. Secondly detection will be focused on; in order to conduct theft there is often a specific signature of events that occur for this to be successful. If you can detect when theft is occurring you can stop it and prevent the virtual property from being stolen. Lastly recovery will be focused on, this is the last resort. If theft does occur and virtual goods are stolen; can users get them back effectively and in a timely fashion or are they lost forever?

### A. *Authentication*

Authentication is the process of proving or confirming by the VWE server that an individual attempting to login to an environment is authentic and the actual owner of the account being accessed. Authentication is a vital component of most online or offline digital environments including VWEs, as it directly relates to being able to access a user's account and the virtual property within it. Authentication can come in the form of passwords, biometrics and digital signatures. The aspects we wish to inspect here are password sniffing, password robustness and keylogging.

#### 1) *Network Password Sniffing*

This technique works by attempting to view the password as it is sent from the client to the server. This test will determine if encryption is used to send the username and password to the login server. The software used to do this testing will be Sniphere version 2.0 [17]. The process will involve launching the virtual world software and then starting the sniffing software, then proceeding to login to the server and analyzing if the password is sent unencrypted.

#### 2) *Password Robustness*

This process will not be automatic and will work by manually entering in commonly used passwords until one is accepted. This is to determine the robustness required for client passwords in VWE's. This works due to people in general choosing easy to remember words as their password; for example pets name or family name. The technique used for this will be getting an assistant to set the password to the test account; making it unknown to the tester. The tester will then use a dictionary file of commonly used passwords and using human input to see if one is accepted. Risk can be determined in the results by looking at if the VWE operator requires the user to set a specific kind of password, for example 8 characters and combination of letters and numbers.

*3) Keylogging*

This technique is executed by utilizing a Trojan type program to monitor keystrokes on a user's computer system. This is a popular attack used to gain unauthorized access to user accounts in many popular VWEs. The testing application for this will involve the use of legitimate key logging software named Actual Keylogger [18], it will involve the monitoring of keystrokes as the authentication procedure is performed on each VWE, to determine if username and password can be captured.

*B. Virtual Property Trading*

Virtual property trading relates to the trade between users of virtual property within the VWE. Trade can occur in many different ways such as direct trade between two avatars (virtual characters), sending virtual property through an in-world mail system as well as buying and selling at an in-world auction house or multi user trade interface; which is a common feature in most of these VWEs. Once a computer criminal gains unauthorized access to a user account, these transactions allow them to steal virtual property from one account and send it off to another which they own. The aspects we wish to inspect here are mail trading, direct trading and aspects of multi-user trade mechanisms.

*1) Direct Trading*

Direct trading is the process whereby user X will open up a trade window dialog box with user Y and transfer virtual property directly. This requires both user X and Y to be online within the world at the same time and essentially provides a real time way of transferring virtual property. The aim of this test is to determine if virtual property items can be directly traded effectively without any security mechanisms or restrictions in place to ensure they are not being stolen.

*2) Mail Trading*

Mail trading involves user X wishing to send an item to user Y; commonly they send an electronic mail from within the VWE which often contains a message and the virtual property item/s. This mail system is often used when user Y is offline or not available for a normal trade window scenario. This is considered a quick and convenient option for players but can provide an easy avenue for unauthorized users to transfer virtual property to another account which they own without requiring them to be logged in on two different accounts at the same time. The aim of this test is to determine if virtual property items can be traded through mail based systems effectively without any security mechanisms or restrictions in place to ensure they are not being stolen.

*3) Multi-user Interface Trading*

In many of the VWEs looked at for this study; multi user trade interfaces such as an auction house are prevalent within them. An auction house is usually a virtual building within the world, where users can walk in and talk to a NPC (Non Player Character) avatar and place virtual property up for auction in order to sell and potentially make some profit. A user might utilize this means; whereby they will log into a compromised account; access one of the characters owned by that user and proceed to place virtual property up for auction for a small price and then buy it on their own legitimate account; providing a means to launder the property to make it seem like an innocent transaction. The aim of this test is to determine if virtual property items can be traded through multi-user trade interface based systems effectively without any security mechanisms or restrictions in place to ensure they are not being stolen.

*C. Intruder Detection*

Intruder detection is the process of detecting intruders or in this case potential thieves as they aim to gain access to unauthorized accounts. If a thief or hacker is not detected by the VWE software, they can break into many accounts and steal virtual goods without being noticed until the owner logs in to discover this theft has occurred. If an unauthorized user can be detected before entry to the VWE is permitted, many of the thefts can be stopped. The aspects we wish to inspect here will be looking at failed login attempts, unusual internet protocol addresses, unusual MAC addresses, and the software version the time of login and can they log onto an account at the same time as the user is currently logged in.

*1) Login Attempts*

This test will be performed during the authentication or login process, whereby the numerous failed attempts at logging in will be conducted and then determine if the VWE actually locks the user out of the account all together for conducting numerous failed attempts. Detection will be asserted TRUE if the account is locked after a certain amount of logins at that time.

*2) IP Address*

This test will conduct logging into a test account of a particular VWE numerous times from one class-B IP address, and then proceed to logon with a completely different class-B IP address using a foreign proxy address and then analyze if this is detected in any way. Therefore detection will be asserted TRUE if the account is locked at that exact moment or a number of days later.

*3) MAC Address*

This test will conduct logging into a test account of a particular VWE numerous times from a test computer with a specific MAC address and then proceed to logon from a completely different computer with a different MAC address and then analyze if that is detected in any way. Therefore detection will be asserted TRUE if the account is locked at that exact moment or a number of days later.

*4) Login Time / Zone*

This test will be performed by logging into a test account of a particular VWE a number of times at a set time each day for a period of time, and after that period proceed to login at completely different times. Analysis will determine if the account gets suspended due to usage times being drastically different time zones.

*5) Concurrent Access*

One important factor is to determine, can a potential thief login to your account at the same time you are logged in? If so this could present some dangers in terms of having all your items stolen, while you're actually still logged in. This test will be conducted by logging into a test account of a particular VWE, then attempting to login to that same VWE with the

exact same username and password. This is to determine if there are measures in place to stop two individuals from logging into the same account twice concurrently.

### D. Recovery

Recovery relates to the reacquisition of stolen virtual property by the original owner. Virtual property often has great value associated with it by the owner; this is due to it often taking great amounts of time and effort to gather, not to mention the user is often paying a subscription fee to play within the VWE. When an item is stolen from a user's account, it is highly beneficial to return the property back to the original owner without a lengthy process as this involves human interaction by staff, costing money to the company since there could be many of these recovery sessions to do per day. The process of recovering stolen virtual property needs to be done accurately so that all individuals involved in the theft are compensated for any innocent transactions or example if a thief is selling stolen items on a virtual world auction house system and innocent users are spending virtual currency to buy these goods without knowing they are stolen.

The aspects inspected here relate to using a scenario based method where theft will be simulated and then requests will be issued to determine if VWE operators can recover the stolen property. More precisely, this series of tests looks at recovery of virtual property after it has been reported or detected as stolen. The only test here will consist of using VWE operators (administrators) to assist in the recovery of the stolen property. The aim of this test is to discover if virtual property can be recovered in the VWE or if it will remain stolen for good.

This experiment was designed to replicate a real VPT (and in need of recovery) situation as much as possible. Our aim was to design the experiments in a way that they would appear to the VWE operator to be a legitimate theft and recovery situation and not a mock scenario. The VWE operators from each individual VWE did not have any affiliation with the tester or knew of this experiment beforehand. This experiment breaks down to essentially, conducting a theft of a number of rare virtual property items between two individual unassociated accounts (thief and victim), then placing a request on the victims account using an online help system featured in-world; asking if the stolen virtual property items could be recovered and returned. Whereby the VWE operator would outright deny the request or conduct some investigation and recover the stolen virtual property items returning them back to the victim account.

As discussed above the inspection process for this test was essentially the same for all three VWEs. This involved having two independent user accounts for each VWE analyzed. These accounts were not related in any way to ensure that the act of VPT (performed entirely by the tester) was viewed as a legitimate act of theft between two separate entities by the VWE operator. This was achieved by registering these accounts under acquaintances of the tester.

These two individual accounts are logged into from two different Internet Protocol (IP) addresses. Each account will have an avatar created for it, and each avatar will be setup with a number of virtual property items of varying quality. The process will involve the avatar from the first account trading; two to three virtual property items and virtual currency to the avatar from the second account using a direct trade mechanism. The first avatar will wait 24 hours and then report these items as stolen and request recovery. Due to the recovery process being textual conversing (through the online help feature) between mock victim and VWE operator entity, to validate if a test was successful or not, we utilized simple visualization to analyze whether or not the VWE operator was able to complete the recovery of our stolen virtual property items. This was answered by the operator entity either outright denying the request by using such phrases as "Sorry we are unable to complete your request" or accepting the request, doing some investigation and placing the stolen goods back into our inventory. An assertion value of TRUE or FALSE was recorded if the property is returned or not, respectively.

## VI. Results

There are two main methods of risk analysis and one hybrid method. First we have qualitative; this aims to improve the awareness of information systems security problems and the position of the system being analyzed [19]. Secondly we have quantitative; this is the identification of where security controls should be implemented, as well as the cost it will take to implement them [19]. Lastly the hybrid method is a combination of both the first two methods, and can be used to implement the components and use the available information all while minimizing the metrics to be collected and calculated [19]. The hybrid method is generally considered a less intensive and expensive method, compared to in-depth analysis.

This study will use qualitative analysis, it is considered much simpler and more widely used [19]. The goal of this study is to identify the parts of VWEs that are at risk and the vulnerabilities that might allow those threats to be realized, so this makes qualitative analysis perfect for this situation. The analysis in this study will use simple calculations and procedures which will determine the impact, probability and overall risk evaluation associated with these threats.

Each test category was broken down into authentication, virtual property trading, intruder detection and virtual property recovery. Each test involved specifying what was evaluated, if the test was successful or not, what probability of this threat is, what the overall impact will be if it occurs, and then provide the results in terms of risk value and evaluation. The tables of results in this study will list the risk outcomes which eventuated from a calculation of assertion, probability and impact.

### A. Assertion

This is a simple test and is evaluated by determining if the aim of the test failed or was successful. A value of 'true' will be given if the test was successful and a value of 'false' will assigned if unsuccessful. The assertion results for each of the inspection categories: Authentication, Unauthorized Trade, and Intruder Detection; as presented in the previous section are shown in Table I.

*B. Risk*

This is evaluated by how easily the assertion was achieved, combining the probability of the attack occurring and the impact if the attack occurs. A value of 'low', 'medium' or 'high' will be given. Figure 1 shows a risk matrix applicable in qualitative risk analysis; the threat level comprises of the probability or likelihood and impact of a risk. Probability is the chance that the risk will occur. Impact is the amount of damage that it would do were it to occur [20].

TABLE I. AUTHENTICATION TESTS – ASSERTION RESULTS

| | **Authentication Assertion** | | | **Unauthorized Trade Assertion** | | | **Intruder Detection Assertion** | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | *Password Sniffing* | *Password Robust* | *Key-logging* | *Mail Trade* | *Direct Trade* | *MUT* | *Failed Logins* | *Multi-IPs* | *Multi-MACs* | *Login Time* | *Concurrent Access* |
| **WoW** | False | True | True | True | True | True | False | True | True | True | False |
| **GW** | False | False | True | True | True | True | False | True | True | True | False |
| **EU** | False | True | True | False | True | True | False | True | True | True | False |

TABLE II. AUTHENTICATION TESTS – ATTRIBUTED RISKS

| | **Authentication Risk** | | | **Unauthorized Trade Risk** | | | **Intruder Detection Risk** | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | *Password Sniffing* | *Password Robust* | *Key-logging* | *Mail Trade* | *Direct Trade* | *MUT* | *Failed Logins* | *Multi-IPs* | *Multi-MACs* | *Login Time* | *Concurrent Access* |
| **WoW** | Medium | High | Critical | Critical | Critical | Medium | Medium | Critical | High | High | Medium |
| **GW** | Medium | High | Critical | Critical | Critical | Medium | Medium | Critical | High | High | Medium |
| **EU** | Medium | High | Critical | Critical | Critical | Medium | Medium | Critical | High | High | Medium |

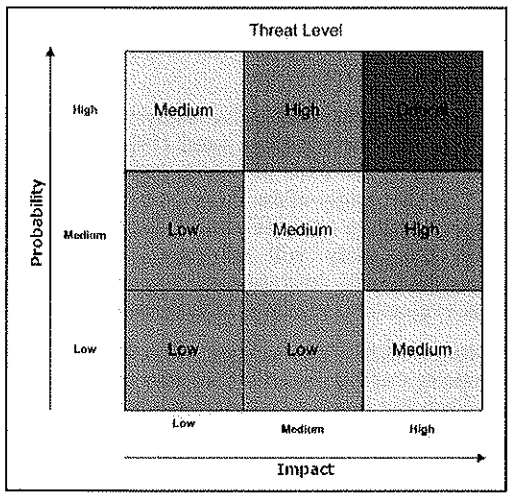



Figure 1. A decision matrix to determine risk rating [20]

- A value of 'low' will be assigned when the risk is at a point where it is so low that there is no apparent threat or danger to the user in terms of VPT. When determining a risk rating of low, both the probability and impact will go towards giving an ultimate value. In Figure 1 these are shown as "green" risks; these are insignificant and most likely will not result in VPT.
- A value of 'medium' will be assigned when the risk is at a point where it is considered a possibility that the user in danger of theft. When determining a risk rating of medium, both the probability and impact will go towards giving an ultimate value. In Figure 1 these are shown as "yellow" risks; they can have a major impact in terms of VPT but if they are managed well by the VWE operators, they can be mitigated.
- A value of 'high' will be assigned when the risk is at a point where it is considered very likely to occur and will provide an impact in terms of VPT. When determining a risk rating of high, both the probability and impact will go towards giving an ultimate value. In Figure 1 these are shown as "orange" risks, these are of high concern which can lead to virtual property theft, but not of critical concern.
- A value of 'critical will be assigned when the risk is of such a high value that the user is very likely to be in danger of this threat. When determining a risk rating of high both the probability and impact will go towards giving an ultimate value. In Figure 1 these are shown as "red" risks and often occur when the VWE software or VWE operators are either unfamiliar with the risk or have no way of stopping it at all, so therefore resulting in a high frequency of VPT.

*C. Authentication Risk*

Authentication deals with processes that relate to when the user is logging into a VWE. The tests associated with authentication, look at password sniffing, password robustness and keylogging and how much of a risk exploiting vulnerabilities in these areas present.

In Table II the authentication risk of each of these security concerns is shown, categorized with the VWE that the test was performed on. As you can see from this table the risk associated with password sniffing for each VWE is quite low,

so there is very small chance that a users password could be intercepted between the client and server by a third party. The next risk being password robustness displayed that the password requirements and complexity required by each VWE is of mid level, thus reducing the chance a user's password could be gathered via brute force techniques. When it comes to the last risk being key-logging, as shown in the graph this is a large problem for all VWE's and users computers alike. Keyloggers fall under the category of malicious software and thus users should keep up to date antivirus and anti malware software on their personal computers as well as not visiting strange websites where key-loggers could be automatically downloaded. Overall the risk associated with authentication presents a high risk (medium probability and high impact) concern to all VWEs tested and can be dealt with some small provisions such as awareness and security software for user's personal computers such as anti-virus and anti malware

*D. Trade Risk*

Unauthorized trading deals with mechanics within the VWE which allow virtual property to be traded from one avatar to another. The tests associated with unauthorized trading involved mechanisms such as at mail trade, direct trading between avatars and multi-user trading.

In Table II the risk associated with each of these trading techniques is shown, categorized with each VWE that was analyzed. From the graph; mail trade is a critical risk (high probability and high impact) for *World of Warcraft* and *Guild Wars*, but is a low risk for *Entropia Universe*, due to the fact EU has no active mail trading mechanism. Mail trading can be used by for virtual property thieves to trade virtual property items without requiring a second avatar to be logged in at the same time to be used as a form of bank for stolen goods.

The next risk, direct trading, allows direct avatar to avatar trading and represents a critical risk (high probability and high impact) for all VWEs. This option in the testers belief is the most likely option to be used as a form of theft mechanics in VWE as it allows for real time trading of stolen goods, enabling thieves to log into a potential stolen account and at the same time be logged into a separate account they own, then send valuable virtual goods from the stolen account in real time and then log out. The last risk, multi-user trading, deals with looking at trade mechanisms which allow thieves to send virtual items to many different users, and be used as a form of laundering or attempting to bring legitimacy to the trade of stolen goods. Overall the risk of unauthorized trading presents a medium (medium probability and medium impact) concern for all VWEs looked at and represents unauthorized trading can occur quite easily once an unauthorized user has gained access to an account.

*E. Intruder Detection*

Intrusion detection deals with being able to detect unauthorized users that attempt to gain access to a legitimate users account for the purpose of virtual property theft. The tests associated with intrusion detection are failed login attempts, unusual internet protocol address (IP), unusual media access control address (MAC), unusual login times and concurrent login attempts.

In Table II the risk associated with each of these intrusion detection techniques is shown, correlated with each VWE that was analyzed. For all the VWEs tested it was shown that when an individual attempts a login a number of times and fails, it was detected and the account suspended for a time period, presenting medium risk from attacks such as automated brute force or a thief trying to guess a users password by hand. The next test was to determine if the VWE detected that a user had an unusual IP address than what had been used in the past and initializes any measures to accommodate that. The result of that test was that the risk is very high and no measures were taken by the VWE to stop a user from logging in from a completely different IP address.

The next test shown in Table II was similar to the IP address test but in fact looked at the MAC address, which is a unique identifier for network interfaces and each computer has a unique one of these. The result of the test was that no measures were taken by the VWE to stop a user from logging in from another computer with a completely different MAC address. Therefore the risk is classified as high. The next test as shown in Table II looked at if the VWE took any measures to detect if a user was logging in at an unusual time, differing than what they usually are on at, say logging in at 3AM as opposed to what they normally log in at being 6 PM for example. The result of this test was the risk was of high concern and no measures were taken by the VWE to alert VWE operators of strange login times on a users account.

The final test as shown in Table II analyzed if two individuals could login to the same account on the same VWE, at the same time concurrently. In all VWEs this proved to be secure, no two accounts can be logged in at the same time. What occurs is simply the current person logged in, is disconnected once the second attempt is successful in authentication. The result here is therefore of medium risk, due to the fact it does actually stop two individuals from being logged in at the same time but does not prevent a hacker from logging into a stolen account, then whereby the system disconnects the owner off the account and the hacker can steal virtual property items of his choosing until getting disconnected. A better developers may implement would be to not disconnect the current active user if say a hacker is trying to login to the account in question. If the owner of an account is logged in and becomes disconnected from the internet, a timer of inactivity could be issued, whereby the account will become automatically logged out after say 6 minutes; then they could log back in.

Overall intrusion detection is for the most part a very high risk for all VWEs looked at; in most instances there are no detection mechanisms in place or there is no follow up when flags are triggered (such as unusual IP address, unusual MAC address, strange login times, concurrent logins); allowing thieves to freely venture in and out of stolen accounts without risk of being detected.

*F. Scenarios: Virtual Property Theft Recovery*

Recovery mechanics looks at the evaluation of the recovery tests and if they were able to be achieved or not. In Table III a series of recovery tests were performed on all VWEs chosen and as a result a success or failure measure was given.

TABLE III. VIRTUAL PROPERTY RECOVERY SCENARIO TESTS

| | Virtual Property Recovery Scenarios | |
|---|---|---|
| | *Success* | *Failure* |
| **WoW** | (0 from 3) 0% | (3 from 3) 100% |
| **GW** | (0 from 3) 0% | (3 from 3) 100% |
| **EU** | (0 from 3) 0% | (3 from 3) 100% |

As shown in Table III each of the virtual property recovery tests were performed at varying times and they all failed. This represents a high degree of inability for VWE operators to recover virtual property once it is stolen by thieves.

### G. *Scenarios: Virtual Property Theft Detection*

A set of virtual property theft scenarios were performed to determine if once a theft occurred; if the VWE software or VWE operator was able to detect it occurring, and stop it from resulting in theft. In Table IV a series of theft tests were performed on all VWEs chosen and as a result a success or failure measure was given.

TABLE IV. VIRTUAL PROPERTY THEFT SCENARIO TESTS

| | Virtual Property Theft Scenarios | |
|---|---|---|
| | *Success* | *Failure* |
| **WoW** | (4 from 4) 100% | (0 from 4) 0% |
| **GW** | (4 from 4) 100% | (0 from 4) 0% |
| **EU** | (4 from 4) 100% | (0 from 4) 0% |

As shown in Table IV all the theft scenarios were successful, representing that theft was able to be performed without being detected by the VWE software or VWE operator whilst it is occurring.

## VII. CONCLUSION

Virtual world software is considered one of the most complex forms of software that exists today; it is essentially a large piece of enterprise software that consists of databases, specialized servers, client software, often millions of users and a huge amount of content. This complexity has presented points of vulnerability to many security problems that have existed for some time in VWE software for with the most part with no effective solutions being produced. People indulging in personal entertainment through buying VWE software and often paying a subscription fee per month should view these results, and then demand VWE operators improve security before the software is given global availability. Solutions to most of these problems can be moderately simple for a VWE development team. This small investment of time and effort and can quite potentially reduce VPT significantly.

There are some key intrinsic factors existing within VWEs which can lead to security compromise. One of these fundamental flaws with VWE software is primitive authentication features which allow key logging to be one off, if not the most fruitful, technique for stealing VWE accounts (which often then leads to VPT). From then on the ability for VWE software or operators to not only detect an account intrusion or detect a VPT situation is essentially nonexistent, allowing the theft to occur with no resistance. Lastly the ability for VWE operators to be able to recover and return victims (often hard earned) virtual property items is nonexistent according to our investigation results. Therefore we have in all areas of authentication, unauthorized trading, intrusion detection and recovery mechanisms we have inherent flaws ranging from medium to critical risk to users of these VWEs.

Until better security development practices are in place and thorough testing of VWE software (as shown in this study) from a security point of view occurs on VWE software by their creators, users should take valid measures to protect their 'investment'. Some of these measures that can be implemented by users to enhance security and avoid VPT are as simple as up to date anti-virus or anti malware software to prevent key-loggers and Trojans, changing your password frequently (bi-weekly, dependant on how much virtual property items you own) so if a potential thief does obtain a users login details, they will have changed the password hopefully before the thief gains access. Measures which can be utilized by VWEs to improve security and protect their community base are more up to date and secure authentication mechanisms along with effective intrusion detection and VPT detection systems, to not only avoid account intrusion but also if required detect VPT and stop it before it can occur. This study is crucial as it highlights the flaws and points out what needs to be fixed in VWEs. Overall the results gained from the set of tests presented in section 6 demonstrated that there are key areas of VWE software that require focus to improve security and reduce the chances of virtual property theft occurring.

To conclude we present a concise list of the most significant findings of this study, which represented a selection of three popular VWEs which were assessed in the year 2010 by a hybrid software inspection process.

- All three (100%) VWEs were highly susceptible to key logging methods, which is used in order to gain unauthorized access to user accounts.
- Mail and direct trading methods were to be the most likely method for intruders to transfer stolen virtual property items.
- Intrusion detection or lack thereof, is of critical risk to all three VWEs and is considered of extreme concern.
- Concurrent logins are not permitted but still don't prevent an account being compromised and virtual property stolen. Simply due to the fact, the current active user is disconnected upon a second successful login. This can potentially result in an ongoing loop of authentication between owner and potential thief.
- When virtual property was actively being stolen, this was not detected nor blocked.
- Stolen virtual property items were unable to be recovered in all scenarios tested – which is also of high concern.

## REFERENCES


[1] A. Watters. (2010). *Number of Virtual World Users Breaks 1 Billion, Roughly Half Under Age 15.* Available: http://www.readwriteweb.com/archives/number_of_virtual_world_users_breaks_the_1_billion.php

[2] M. Eikenberry. (2011). *Real Money Trade is a Billions Dollar a year Industry.* Available: http://www.youtube.com/watch?v=rtZY3fVwlgw

[3] N. Patterson and M. Hobbs, "A Multidiscipline Approach to Governing Virtual Property Theft in Virtual Worlds," in *What Kind of Information Society? Governance, Virtuality, Surveillance, Sustainability, Resilience*. vol. 328, J. Berleur, *et al.*, Eds., ed: Springer Boston, 2010, pp. 161-171.

[4] C. Boogerd and L. Moonen, "Prioritizing Software Inspection Results using Static Profiling," in *Sixth IEEE International Workshop on Source Code Analysis and Manipulation (SCAM'06)*, 2006, pp. 149-160.

[5] DFC.Intelligence, "Virtual Property and Real Money Trade: A Business and Legal Survey," DFC Intelligence, San Diego, California2009.

[6] S. Cikic, *et al.*, "Cheat-prevention and -analysis in online virtual worlds," presented at the 1st International Conference on Forensic Applications and Techniques in Telecommunications, Information, and Multimedia and Workshop (e-Forensics '08), Adelaide, Australia, 2008.

[7] V. B. Livshits and M. S. Lam, "Finding security vulnerabilities in java applications with static analysis," presented at the Proceedings of the 14th conference on USENIX Security Symposium - Volume 14, Baltimore, MD, 2005.

[8] M. Christodorescu and S. Jha, "Static analysis of executables to detect malicious patterns," presented at the Proceedings of the 12th conference on USENIX Security Symposium, Washington, DC, 2003.

[9] K. J. Hole, *et al.*, "Case study: online banking security," *Security & Privacy, IEEE,* vol. 4, pp. 14-20, 2006.

[10] Blizzard.Entertainment. (2004). *World of Warcraft Community Site*. Available: http://www.worldofwarcraft.com

[11] NCSoft. (2009). *NCSoft.* Available: http://global.ncsoft.com/global/

[12] MindArk. (2009). *Entropia Universe.* Available: http://www.mindark.com/entropia-universe/

[13] W. Yin-Poole. (2010). *World of Warcraft hits 12m subscribers* Available: http://www.eurogamer.net/articles/2010-10-07-world-of-warcraft-hits-12m-subscribers

[14] NCsoft. (2009, 21/9/2011). Guild Wars Surpasses Six Million Units Sold. *Guild Wars Press Release* [Online Article]. Available: http://www.guildwars.com/events/press/releases/pressrelease-2009-04-24.php

[15] C. Donatello. (2011). *'Entropia Universe' Boasts Improved Land Grab System.* Available: http://sciencefiction.com/2011/03/15/%E2%80%98entropia-universe%E2%80%99-boasts-improved-land-grab-system/

[16] J. Nielsen, "Usability inspection methods," presented at the Conference companion on Human factors in computing systems, Boston, Massachusetts, United States, 1994.

[17] SecureSphere. (2010). *SecureSphere - Free IT Security Software.* Available: http://www.securesphere.net/

[18] Actual.Spy.Software. (2010). *Actual.Key.Logger.* Available: http://www.actualkeylogger.com

[19] J. W. Meritt. (1999). *A Method for Quantitative Risk Analysis.* Available: http://csrc.nist.gov/nissc/1999/proceeding/papers/p28.pdf

[20] B. Witzel. (2005). *Bad things that can happen to good people: Identifying project risks.* Available: http://www.charityvillage.com/cv/research/rom18.html

**Paper 3: A Comparison of the Classification of Disparate Malware Collected in Different Time Periods**

Rafiqul Islam, Ronghua Tian, Veelasha Moonsamy and Lynn Batten, School of Information Technology, Deakin University.

# A Comparison of the Classification of Disparate Malware Collected in Different Time Periods

Rafiqul Islam, Ronghua Tian, Veelasha Moonsamy, Lynn Batten
*School of Information Technology*
*Deakin University, Melbourne, Australia*
*(rafiqul.islam, rtia, v.moonsamy, lmbatten)@deakin.edu.au*

## Abstract

*It has been argued that an anti-virus strategy based on malware collected at a certain date, will not work at a later date because malware evolves rapidly and an anti-virus engine is then faced with a completely new type of executable not as amenable to detection as the first was.*

*In this paper, we test this idea by collecting two sets of malware, the first from 2002 to 2007, the second from 2009 to 2010 to determine how well the anti-virus strategy we developed based on the earlier set [14] will do on the later set. This anti-virus strategy integrates dynamic and static features extracted from the executables to classify malware by distinguishing between families.*

*The resulting classification accuracies are very close for both datasets, with a difference of only 5.4%, the older malware being more accurately classified than the newer malware. This leads us to conjecture that current anti-virus strategies can indeed be modified to deal effectively with new malware.*

**Keywords**: malware, classification, static, dynamic.

## 1. Introduction

The work in [12, 3, 10, 2, 1, 16, 11, 8] supports the argument that an anti-virus strategy which has been successful in a given time period will not work at a much later date; this, they argue, is due to changes in malware design which evolves with time and eventually becomes unrecognizable from the original form.

The aim of this paper is to test this argument. We do so by considering two sets of malware, one collected during the period 2002 to 2007 (881 samples) and the other during the period 2009 to 2010 (1517 samples). All samples come from CA Technologies VET Zoo (www.ca.com) and all have been pre-classified as members of particular families.

The key contribution of this paper is the provision of strong evidence that anti-virus techniques which work well on malware developed at a certain time may continue to be effective on malware developed at a much later time.

The rest of the paper is organized as follows: Section 2 includes a summary of the related work on comparison of malware classification over a time period. In Section 3 we elaborate on the data preparation while Section 4 describes the classification process. Section 5 provides an analysis of the results and lastly, in Section 6 we discuss these results and present some ideas for future work.

## 2. Related Work

In [1], the authors classify malware based on behavioral features, considering the interactions between the executable files and the operating system. Testing about 8,000 malware samples collected over the period 2004 to 2007, they achieve an overall classification accuracy of almost 92 %.

Zheng and Fang [17] propose a novel infrastructure for malware detection that can be implemented into a cloud system thus relieving a local end system of strenuous processing of suspicious files. They compare detection rates on malware sets which differ in age by up to3 months, with diminishing levels of accuracy.

The authors of [5] use a combination of static and dynamic analysis to achieve a high level of malware accuracy over an eight year time period, demonstrating that including 'older' malware in the set for feature selection can assist in identifying 'new' malware.

Roundy and Miller [9] showed that by applying a hybrid of dynamic and static analysis, the probability of correctly detecting malicious programs can be significantly increased. Testing their algorithm on 200 malware samples, the authors found that 33% of the malicious code analysed by the combined methods would not have been identified by dynamic analysis only.

## 3. Data Preparation

In this section we provide a detailed explanation of the experiment. Section 3.1 describes the datasets that are used; Section 3.2 introduces our feature extraction method.

### 3.1. Datasets

In our experiment, we use the term 'Old Dataset' to refer to the malicious files collected between 2002 and 2007 and the term 'New Dataset' for those collected between 2009 and 2010. Tables 1 and 2 list the families in the Old and New Datasets respectively, along with the numbers of files per family.

**Table 1. Malware files in Old Dataset.**

| Type | Family | Number of files |
|---|---|---|
| *Virus* | Emerleox | 75 |
| | Looked | 66 |
| | Agobot | 283 |
| *Trojan* | Clagger | 44 |
| | Alureon | 41 |
| | Bambo | 44 |
| | Boxed | 178 |
| | Robknot | 78 |
| | Robzips | 72 |
| | ***TOTAL*** | **881** |

**Table 2. Malware files in New Dataset.**

| Type | Family | Number of files |
|---|---|---|
| *Worm* | Frethog | 174 |
| | SillyAutorun | 87 |
| *Trojan* | Addclicker | 65 |
| | Gamepass | 179 |
| | Banker | 47 |
| | SillyDI | 439 |
| | Vundo | 80 |
| | Bancos | 446 |
| | ***TOTAL*** | **1517** |

The date used was the date assigned to the malware as it was received by CA Technologies VET Zoo. It is of interest to note that within the total of 2398 malware files, no family is represented in both Old and New Datasets. This was not done intentionally but is a true representation of the data in the Zoo.

### 3.2. Feature Extraction

We extract both static and dynamic features to be used for malware classification.

#### (i) Static Features

We unpack all malware using a command line anti-virus engine provided by CA Technologies. The software allows us to unpack the executables in batch mode which considerably reduces the unpacking time. We then consider two static features which are: (a) Function Length Frequency (FLF) and (b) Printable String Information (PSI).

In extracting the FLF features, we follow the methodology described in [13] using IDAPro to define the functions.

The lengths of each function (as a bitstring) are computed and these lengths are divided into 50 'bins' based on the methodology of [13]. Each bin is correlated with the total number of function lengths (with repeats) it contains and this number is called the *function length frequency* of any function which has a length in the bin.

As an example, we consider a malicious file, $F$, which includes 12 functions which have the following lengths (represented in bytes and in increasing order of size): 4, 5, 5, 12, 15, 15, 18, 19, 23, 45, 60 and 90. For the purpose of illustration, let us create 10 exponentially spaced bins based on the function length ranges. The distribution of frequencies across the bins is depicted in Table 3.

**Table 3. FLF bin distribution.**

| Length of functions per bin | FLF Vectors |
|---|---|
| 1-2 | 0 |
| 3-8 | 3 |
| 9-21 | 5 |
| 22-59 | 2 |
| 60-166 | 2 |
| 167-464 | 0 |
| 465-1291 | 0 |
| 1292-3593 | 0 |
| 3594-9999 | 0 |
| >=10000 | 0 |

Finally, we select all the entries from the last column of the above table to form the following FLF vector for the file $F$: (0, 3, 5, 2, 2, 0, 0, 0, 0, 0). In our actual experiment, we fix the bin size to 50 based on a global list of all function length frequencies.

The second type of static feature, PSI, is extracted from the disassembled malicious executables. For each dataset, the printable strings are collected and combined to build a global string list. The example below explains the steps used in generating the PSI vector for a particular malicious executable.

Let us consider the following global string list consisting of 7 ***distinct*** strings: {"GetProcAddress",

"RegQueryValueExW", "CreateFileW", "OpenFile", "FindFirstFileA", "FindNextFileA", "CopyMemory"}, where the order of strings within the list is fixed. Assume that the list of printable strings extracted from a particular executable file *F*, including repeats, is: {"GetProcAddress", "RegQueryValueExW", "CreateFileW", "RegQueryValueExW"}. We then track the presence of the strings in *F* against the global list using a '1' to indicate that a string is present (at least once) in the global string list and a '0' to denote the absence of the string. Table 4 presents the corresponding information where we also include the total number of strings (with repeats) in the file in the first row. Hence, the corresponding PSI vector for *F* is (4,1,1,1,0,0,0,0).

**Table 4. PSI data for file *F*.**

| | |
|---|---|
| Number of strings | 4 |
| "GetProcAddress" | 1 |
| "RegQueryValueExW" | 1 |
| "CreateFileW" | 1 |
| "OpenFile" | 0 |
| "FindFirstFileA" | 0 |
| "FindNextFileA" | 0 |
| "CopyMemory" | 0 |

### (ii) Dynamic Features

The dynamic features are obtained from runtime behaviour of packed malware executables. We execute both datasets in a controlled virtual machine (VM) environment and record the behaviours in log files. To generate the log files, the executables were run for 30 seconds and then stopped. Below is an excerpt of a log file for executable *F*, where the API calls and the parameters are shown in italics:

2010/09/02 11:24, *RegQueryValueExW, Compositing*
2010/09/02 11:24, *RegOpenKeyExW, 0x54, Control Panel\Desktop*
2010/09/02 11:24, *RegQueryValueExW, LameButtonText*
2010/09/02 11:24, *LoadLibraryW, .\UxTheme.dll*
2010/09/02 11:24, *LoadLibraryExW, .\UxTheme.dll*

Let us assume that the global API list of distinct features is as follows: {"RegOpenKeyEx", "RegQueryValueExW", "Compositing", "RegOpenKeyExW", "0x54", "Control Panel\Desktop", "LameButtonText", "LoadLibraryW", ".\UxTheme.dll", "LoadLibraryExW", "MessageBoxW"}. We then compare the API features from the log file of *F* with the global list and count the frequencies of each item to generate the dynamic vector.

In this case, the dynamic vector for *F* is (0,2,1,1,1,1,1,2,1,0), drawn from the frequency column of Table 5.

**Table 5. Dynamic feature data for *F*.**

| Dynamic Features in global list | Frequency |
|---|---|
| "RegOpenKeyEx" | 0 |
| "RegQueryValueExW" | 2 |
| "Compositing" | 1 |
| "RegOpenKeyExW" | 1 |
| "0x54" | 1 |
| "Control Panel\Desktop" | 1 |
| "LameButtonText" | 1 |
| "LoadLibraryW" | 1 |
| ".\UxTheme.dll" | 2 |
| "LoadLibraryExW" | 1 |
| "MessageBoxW" | 0 |

### (iii) Integrated Features

The integrated feature vector is a combination of the FLF, PSI and dynamic vectors. The motivation behind combining the different types of features described in the previous subsections is to prevent a malware writer from bypassing anti-virus technologies based on a single component of a malware file.

Hence, we collect the three vectors, FLF, PSI and Dynamic, and merge them into a single vector for each malicious executable, as shown in Figure 1.

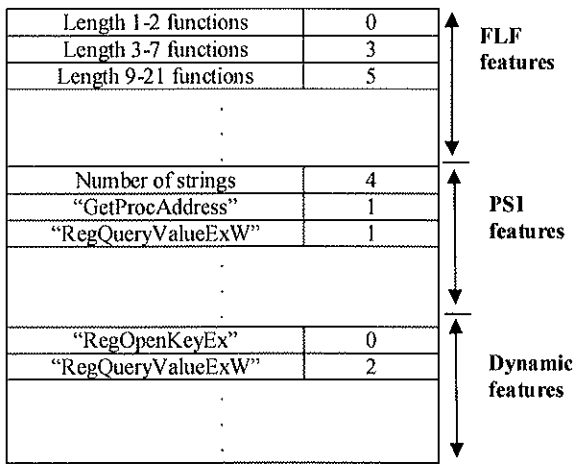



**Figure 1. Example of an integrated vector.**

## 4. Classification

For the classification process, we use four base classifiers from WEKA [4] representing a broad spectrum of classifier types: Sequential Minimal Optimization (SMO), Instance-Based (IB1), Decision Table (DT) and Random Forest (RF), and apply the

statistical method known as 10-fold cross-validation [6] to classify the data.

In the cross-validation phase, we select files from one particular malware family and choose the same number of files at random from other families, using a random function. The files are then divided into 10 groups, where one group is used as a testing set and each of the remaining nine groups as a training set.

A step-by-step breakdown of our methodology is given below:

1. Extract and generate the vectors for the FLF, PSI and dynamic features from all files in the Old Dataset, as described in Section 3.
2. Build the integrated vector from the 3 feature vectors in Step 1 and use these to generate the WEKA (arff) files.
3. Select a malware family, $M$, and from the remaining families, randomly choose a set of size $|M|$ of malware files.
4. For the malware files chosen in Step 3, consider the set of corresponding arff files; break this set into 10 groups of equal size using the procedure, 'Making 10 groups of equal size', discussed below.
5. Select one of the constructed groups as a testing set and the union of the remaining nine as a training set.
6. Call WEKA libraries to train the classifiers using the training set.
7. Evaluate the classifiers using the testing set.
8. Repeat steps 3 to 7 for the remaining families. Then go to Step 9.
9. The first time, return to Step 1 and repeat for the New Dataset. The second time, exit.

**Making 10 groups of equal size:**

In Step 4 from the above described ethodology, we split the set of arff files, $A$, into 10 groups of equal size as follows:

- If $10 \mid |A|$, then each group has size $\frac{|A|}{10}$.
- If $10 \nmid |A|$, then we first generate 9 groups using the following equation,

  $$|A| = 9 * B + r, \text{ where } 0 \leq r < 9,$$

  whereby we take 9 groups of size $B$ and place the remaining $r$ arff files in a $10^{th}$ group along with $(B\text{-}r)$ randomly chosen arff files from $A$.

We also apply the meta classifier, AdaboostM1, to each of the base classifiers and rerun the tests. The meta classifier enhances the capabilities of the base classifiers by operating on the output of those classifiers. The classification accuracies produced by AdaboostM1 represent the correctness of each file (also referred to as an instance) classified by each of the four base classifiers, as described in [15]. In all cases, the boosted base classifiers perform better than the base classifier and therefore we present only the meta-classifier results in the next section.

## 5. Analysis of Results

The classification results for the Old and New Datasets are presented in Table 6 and in Figure 2.

**Table 6. Weighted Average Accuracy (with Adaboost).**

| Boosted Classifiers | Old Dataset | New Dataset |
|---|---|---|
| SMO | 98.9% | 83.9% |
| IB1 | 99.2% | 90.7% |
| DT | 99.2% | 92.4% |
| RF | 99.8% | 94.4% |

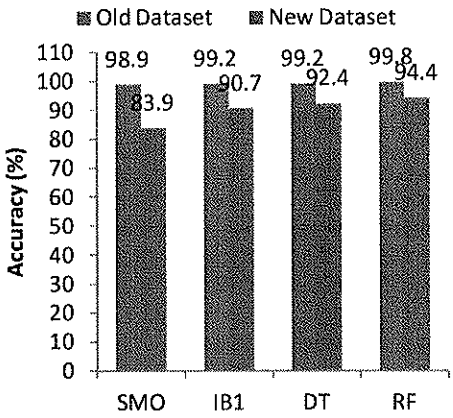



**Figure 2. Weighted Average Accuracy.**

Overall, RF outperformed the other classifiers with 99.8% classification accuracy for the Old Dataset and 94.4% for the New Dataset. These numbers also have the smallest difference (5.4%) across classifiers. We believe that the high accuracies obtained from RF can be attributed to the fact that this classifier runs capably on large datasets which incorporate diverse features, hence confirming the effectiveness of using integrated features.

On the other hand, the classifier SMO shows worst performance for new data set which is almost 15% drop in accuracy compared to old data set. The reason for this drop in accuracy between old and new data set are not clear and remain for further investigation. However, in

the classification process, while SMO build training model, it determines optimal tangential hyperplanes which can separate the data points (support vectors) into the categories. Then the SMO attempts to classify these support vectors into groups by determining on which side of a hyperplane a point of data lies. Therefore, it could be one of the reasons to build the optimal hyperplane with minimum support vectors, hence reduce the classification accuracy.

Moreover, it has been shown from Figure 2 that the classification accuracy of old malware data set show significance performance compared to new malware data set for all classifiers. It has been observed that our old experimental data set, consists approximately 51% malicious files from Trojan families and 49% virus families however, the new experimental data set consists 83% malicious files from Trojan families and only 17% from worm families but no files virus families. Therefore, it is assumed that the dissimilarities of malware types between old and new families could impact the drop of classification accuracy.

## 6. Discussion and Future Work

While our malware classification strategy worked very well on the old malware set, its results were much more moderate on the new malware set. This second weaker result was almost certainly due to the difference in malware in the samples. Some malware families in the New Dataset require the user's input along with an Internet connection in order to execute some of the in-built functions, and so these functions would not have been extracted into our classification test using our method.

On the other hand, the boosted RF test gave reasonable results and this indicates that older classification techniques should not be abandoned en masse but that they could be adapted to cope with malware as it evolves. One such adaptation might be to include both old and new malware in the same test; another might be to combine the features for the datasets in other ways as, for example, in [9].

Moreover, we have demonstrated in this paper that it is possible to develop an anti-malware technique which can maintain consistent performance with more advanced and future malware. The main approach we have used was to combine all feature types, derived from FLF, PSI and dynamic API calls and API parameters, into a single vector allows the classifier algorithm to identify complex patterns which span multiple feature types. Our empirical study indicates that our strategy performed well in new malware data set with 5.4% accuracy drop. Therefore it is expected that our proposed method can deal with future malware i.e. 2011 malware. However, it is difficult to predict whether the detection rate will maintain the same performance or not. In our future work we will investigate our system using more advanced and challenging data set.

## 7. References


[1] Michael Bailey, Jon Oberheide, Jon Andersen1, Z. Morley Mao1, Automated Classification and Analysis of Internet Malware, Chapter Recent Advances in Intrusion Detection 178-197, 2007.

[2] Barford, P., Yagneswaran, V.: An inside look at botnets. In: Series: Advances in Information Security, Springer, Heidelberg (2006)

[3] Cyveillance, accessed on 19th May 2011, http://www.cyveillance.com/web/news/press_rel/2010/2010-08-04.asp

[4] Mark Hall, Eibe Frank, Geoffrey Holmes, Bernhard Pfahringer, Peter Reutemann, Ian H. Witten (2009); The WEKA Data Mining Software: An Update; SIGKDD Explorations, Volume 11, Issue 1.

[5] Islam, R., Tian, R., Moonsamy, V., Batten, L., Versteeg, S.: A cumulative timeline approach to malware detection. Submitted September 2011.

[6] Kohavi, R., A study of cross-validation and bootstrap for accuracy estimation and model selection. In IJCAI (1995), pp. 1137–1145

[7] Lee, J.; Im, C. & Jeong, H. (2011), A study of malware detection and classification by comparing extracted strings, in 'Proceedings of the 5th International Conference on Ubiquitous Information Management and Communication', ACM, New York, NY, USA, pp. 75:1--75:4.

[8] Nair, V. P.; Jain, H.; Golecha, Y. K.; Gaur, M. S. & Laxmi, V. (2010), MEDUSA: Metamorphic malware dynamic analysis using signature from API, in 'Proceedings of the 3rd international conference on Security of information and networks', ACM, , pp. 263--269.

[9] Roundy, K. & Miller, B. (2010), Hybrid Analysis and Control of Malware, in 'Recent Advances in Intrusion Detection', Springer Berlin / Heidelberg, pp. 317-338.

[10] Sukwong, O.; Kim, H. & Hoe, J. (2010), 'An Empirical Study of Commercial Antivirus Software Effectiveness', Computer 44 (3), 63--70.

[11] Tang, H.; Zhu, B. & Ren, K. (2009), A New Approach to Malware Detection, in Jong Park; Hsiao-Hwa Chen; Mohammed Atiquzzaman; Changhoon Lee; Tai-hoon Kim & Sang-Soo Yeo, ed., 'Advances in Information Security and Assurance', Springer Berlin / Heidelberg, , pp. 229-238.

[12] Takeshi Yagi, Naoto Tanimoto, Takeo Hariu adn Mistsutaka Itoh, Investigation and analysis of malware on websites, IEEE 2010

[13] Tian, R., Batten, L., and Versteeg, S. Function length as a tool for malwareclassification. In Proceedings of the 3rd International Conference on Malicious and Unwanted Software: MALWARE 2008 (2008), pp. 69–76.

[14] Tian, R., Islam, R., Batten, L., and Versteeg, S. Differentiating malware from cleanware using behavioural analysis. In Proceedings of the 5rd International Conference on Malicious and Unwanted Software: MALWARE 2010 (2010)

[15] Witten, I.; Frank, E.; Trigg, L.; Hall, M.; Holmes, G. & Cunningham, S. (1999), Weka: Practical machine learning tools and techniques with Java implementations, *in* 'ICONIP/ANZIIS/ANNES', pp. 192--196.

[16] You, I. & Yim, K. (2010), Malware Obfuscation Techniques: A Brief Survey, in 'Broadband, Wireless Computing, Communication and Applications (BWCCA), 2010 International Conference on', pp. 297 -300.

[17] Zheng, X. & Fang, Y. (2010), An AIS-based cloud security model, *in* 'Intelligent Control and Information Processing (ICICIP), 2010 International Conference on', pp. 153 -158.

**Paper 4: Microphone Identification using One-ClassClassification Approach**

Huy Quan Vu, Shaowu Liu, Zhi Li, and Gang Li, School of Information Technology
Deakin University.

# Microphone Identification using One-Class Classification Approach

Huy Quan Vu*, Shaowu Liu†, Zhi Li‡, and Gang Li§
School of Information Technology
Deakin University, 221 Burwood Highway
Vic 3125, Australia
* Email: quan@tulip.org.au
† Email: swliu@tulip.org.au
‡ Email: zhilimailbox@yahoo.com.au
§ Email: gang.li@deakin.edu.au

***Abstract*—Rapid growth of technical developments has created huge challenges for microphone forensics - a sub-category of audio forensic science, because of the availability of numerous digital recording devices and massive amount of recording data. Demand for fast and efficient methods to assure integrity and authenticity of information is becoming more and more important in criminal investigation nowadays. Machine learning has emerged as an important technique to support audio analysis processes of microphone forensic practitioners. However, its application to real life situations using supervised learning is still facing great challenges due to expensiveness in collecting data and updating system. In this paper, we introduce a new machine learning approach which is called *One-class Classification* (OCC) to be applied to microphone forensics; we demonstrate its capability on a corpus of audio samples collected from several microphones. Research results and analysis indicate that OCC has the potential to benefit microphone forensic practitioners in developing new tools and techniques for effective and efficient analysis.**

***Index Terms*—Machine Learning, Data Mining, Audio Forensics, Microphone Forensics, One-Class Classification**

## I. Introduction

Microphone forensics is a sub-category of audio forensic science, which aims to establish whether an obtained audio recording is original, or to verify whether it was made on a given recorder. The determination of microphone model of arbitrary recording can help assure the actual ownership of that recording in the case of multiple claims of ownership, and thus provides a valuable mechanism to resolve copyright disputes. Rapid growth of technical developments in the past decade has created huge challenges with availability of numerous digital recording devices and massive amount of recording data. These digital media make undetected forgeries and manipulations easy, and might thereby encourage criminals. Demand for fast and efficient methods to assure integrity and authenticity of information is becoming more and more important in criminal investigation nowadays.

Recently, machine learning has emerged as an important method of automated audio analysis processes which provides supportive tools for microphone forensic practitioners. A first attempt of practical evaluation on recording devices and environment classification was performed by Kraetzer et al. in 2007 [1]. They incorporated the *K-means* and *Naive Bayes* as classifiers, and evaluated their classification capability on a set of audio *steganalysis* features. Later on, they proposed an *Unweighted Fusion* framework using a *Decision Tree* and *Linear Logistic Regression* models that achieved higher performance on microphone detection task [2]. In 2009, another attempt using supervised machine learning methods (*Simple Logistic, J48 decision tree, K-nearest neighbor, Support Vector Machine−SVM*) was performed by Buchholz et al. with *Fourier* coefficient histogram extracted from near-silence segments of the recording as the feature vectors [3]. Similar approach was utilized by Garcia-Romero and Espy-Wilson [4] with SVM classifier to assess the performance of *linear-cepstral coefficients* and *mel-scaled cepstral coefficients* (MFCCs) as audio features. The audio samples were obtained from two classes of acquisition devices, land-line telephone handsets and microphones. Recently, Kraetzer et al. [5] designed a context model to devise empirical investigation to identify suitable classification algorithm and appropriate audio features as a supportive guidance for microphone forensic researchers, the experiment involved 74 supervised classification algorithms and 8 clusters

developed in Weka and 590 intra-frame audio features.

Although, great effort has been spent on this task of microphone forensics, only a limited number of approaches were found. Current techniques using supervised machine learning are still facing great challenges due to expensiveness in collecting data and updating system. Research results are still far from standard for real life application. In this paper, we introduce a relatively new approach of machine learning - *One-class classification* (OCC), into microphone forensics, which has potential to benefit microphone forensic practitioners in developing new tools and techniques for more efficient analysis.

Having introduced the research motivation, section II provides a critical analysis of supervised learning problems in microphone forensics, following by formulation of research objective for this work. Section III is devoted to describe a set of audio features and present several *OCC* techniques which are selected for our experiments. In Section IV we present our audio data collection and analysis of empirical results. Section V concludes the paper with a summary and offers practical implication as well as future research suggestions.

## II. Problem Definition and Research Objective

Traditionally, microphone identification is considered as an $n$-class supervised learning problem, where an audio sample is classified into one of $n$-classes of microphone models in the training data. A single model is trained on a data set consisting of audio samples for $n$ microphone models. For microphone verification, the task can be considered as a binary classification problem which aims to determine whether the audio sample is really recorded by a particular microphone models or not. The training data contains of two classes label: *positive* (audio samples of claimed microphone) and *negative* (audio samples of all other microphones), each microphone requires a unique model to verify its identity. Agreed that binary classification is usually easier than multi-class classification, microphone verification would come out to be simpler than microphone identification. However, real world application of microphone forensics using supervised learning approach is a very challenging problem because of their *open* nature. It is impractical, if not impossible, to construct a complete database with audio samples recorded from all available microphones models in the world as training data set. In addition, when new models are being produced continuously, the entire supervised classification model needs to be retrained frequently, and hence makes supervised learning an expensive approach. This aspect of supervised learning approach is demonstrated in Fig. 1.

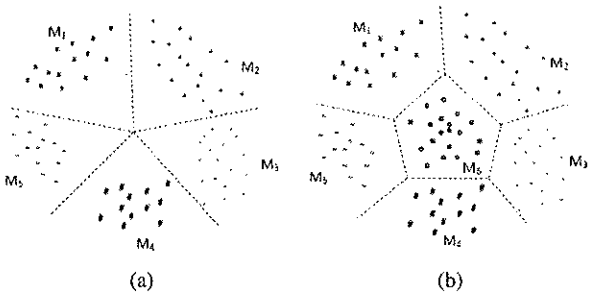



Fig. 1. Supervised Machine Learning Approach

Fig. 1(a) shows a decision boundary on a data set containing 5 microphones ($M_1 - M_5$) constructed by supervised machine learning approach, data extracted from different microphones is represented by different symbols. This decision boundary can be used to classify a new data samples into one out of 5 classes corresponding to 5 microphone models. However, when a new microphone becomes available ($M_6$), the decision boundary need to be retrained in order to correctly classify the microphone models as shown in Fig. 1(b), For these reasons, it is worth to analyze this task under a different approach that should be more practical, cost effective and easier for system maintenance.

Due to recent advances in machine learning, an approach has emerged to be more suitable for application for microphone forensics, which is referred to as *One-class classification*. It can be simple to obtain audio samples from a particular microphone models as *positive* (target) class but impossible to collect from all other available microphones as *negative* (outlier) class. The idea behind OCC approach is to design a classifier so that only the target class is characterized, and consequently can distinguish it from all counter-examples from outlier class. Note that, OCC approach characterizes only the target class, only data samples of target microphone are required during training process of OCC model. Each microphone requires a model to be trained on its own data samples, therefore, to identify $n$ microphones, $n$ OCC models are constructed. An example of OCC model construction for a microphone is demonstrated in Fig. 2.

Fig. 2(a) shows a tight decision boundary surrounding *target* microphone data using OCC approach, which can separate it from *outlier* data samples. Here, the data

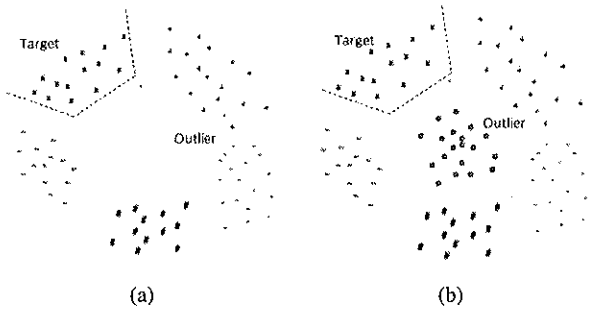



Fig. 2. One-Class Classification Approach

samples of *outlier* classes are shown for demonstration purpose, not input to OCC model. When new microphones becomes available as shown in Fig. 2(b), the existing OCC model is not required to be re-trained. As a result, the updating process of microphone authentication system is simpler and cheaper than supervised learning approach. Despite the fact that the OCC approach has been successfully employed in several audio forensic tasks such as *sound classification* [6], [7], *scene classification* [8], and *speaker verification* [9], so far no work has been found in the field of microphone forensics where its application scenario is considered appealing.

The objective of this paper is to introduce the application of OCC into the field of Microphone Forensics. Our goals are to evaluate OCC algorithms performance whether it can accomplish the task of identifying and verifying microphone models, and which OCC algorithm can achieve the best results. Hereafter, we use the term "*microphone identification*" to refer to both *identification* and *verification* tasks of microphone forensic category. In order to achieve such goals, we apply a range of relatively new OCC algorithms to audio samples collected from different scenarios using a sets of digital microphones. Empirical results and analysis are promising to provide forensic practitioners with an overview about OCC approach. This can support researchers in this area to make microphone forensics a more practical science.

## III. Methodology

In this section, we first select a set of audio features to capture characteristics of audio recordings, then, we outline several OCC algorithms which are used in this study.

### A. Audio Features

Audio features are mathematical representations reflecting characteristics of audio signal that are used in statistical pattern recognition based approach for audio forensics. Depending on the tasks of audio forensic practitioners, different sets of audio features could be employed. In [10], AlQahtani et al. made use of MPEG-7 audio low level descriptors along with temporal zero crossing as features vector for automatic recognition of environment sounds. Recently, Sen et al. proposed a new feature extraction technique coming from a new transformation which is based on the *Nyquist* filter bank and achieved significant result in speaker identification [11]. Besides, feature sets extracted from *Linear Predictive coefficients* (LPC) and *mel-frequency cepstral coefficient* (MFCC) also have powerful descriptive capability which are used frequently in gunshot detection [12], audio clips classification [13] and environment sound recognition [14]. In microphone forensic area, different feature sets have been tested in identifying microphone models of recorded audio samples, which include features in time domain, frequency domain and Mel-cepstrum domain. Recent studies conducted by Garcia-Romero and Espy-Wilson [4], and Kraetzer et al. [5] have confirmed that MFCCs are among the best candidates for microphone identification due to its ability of capturing microphone characterizes as well as low dimensionality of 13 coefficients (features). Therefore, we employ MFCCs as our audio features to assess the performance of OCC algorithms in this study. A detail description of MFCCs can be obtained from [15].

### B. One-class Classification algorithms

The area of OCC is considerably well adapted to the problems of microphone identification where sampling every microphone model is an impossible task. For OCC approach, a boundary around the well-sampled target distribution is constructed that may discard a small percentage of targets examples, and consequently hopes to be able to identify majority of target while throwing out as many of the counter-class examples as possible. A sample is classified as member of target class if its assigned scored by OCC model lies above a given threshold, otherwise, it is considered belonging to outlier class.

In the context of this study, we use a number of OCC algorithms implemented in *Data Description toolbox* by Tax [16] which are described as follows:

- **One-class Gaussian Model (1-GN):** uses a simple Gaussian to characterize target class. When instances of target class are input to this algorithm, a model is constructed using one Gaussian distribution, where $\mu$ is estimated mean and $\Sigma$ is estimated

covariance matrix of target "points":

$$f(x) = (x - \mu)^T \Sigma^{-1} (x - \mu) \quad (1)$$

An instance $x$ is classified as target class if $f(x)$ is greater than a specified target error $\theta$, otherwise, it is considered belonging to outlier class.

- **One-class Gaussian Mixture Model (1-GNM):** uses a mixture of $K$ Gaussians to construct a more flexible description for target class. The model can be presented as follows:

$$f(x) = \sum_{i=1}^{K} P_i \exp(-(x - \mu_i)^T \Sigma_i^{-1} (x - \mu_i)) \quad (2)$$

In this model, EM algorithm is utilized to optimize the parameters $P_i$, $\mu_i$ and $\Sigma_i$.

- **One-class K-Means (1-KM):** describes the target data by $k$ clusters with center $c_i$ to be estimated such that the average distance to a cluster center is minimized. The model for 1-KM is described as follows:

$$f(x) = \min_i (x - c_i)^2 \quad (3)$$

- **One-class K-Nearest Neighbors (1-KNN):** evaluates a new object $x$ by computing the distance to its $k$ nearest neighbors in target class. Then this distance is compared to a threshold to evaluate the prediction output of $x$.
- **One-class Principal Component Analysis (1-PCA):** describes the target data by a linear subspace $W$ which is defined by the $k$ eigenvector of covariance matrix of target class data. The fitness of new object $x$ to the target subspace is evaluated by comparing it to reconstruction error:

$$f(x) = ||x - x_{proj}||^2 \quad (4)$$

where the projection $x_{proj}$ is calculated by:

$$x_{proj} = W(W^T W)^{-1} W^T x \quad (5)$$

- **One-class Incremental Support Vector Machine (1-ISVM):** fits a hypersphere around the target class without using an external quadratic programming optimizer or kernel. The employment of Support Vector Machine technique into context of one-class was originally proposed by Tax and Duin [17], which is known as *Support Vector Domain Description* (SVDD). It was proven to achieve significant results in application to audio surveillance system [18] and speaker verification [9]. Therefore, we would like to evaluate this technique in this study using 1-ISVM as a more advanced version of SVDD.

## IV. Experiment and Analysis

This section first describes our audio samples collection process and the evaluation metrics to assess the performance of OCC algorithms. Then, detail analysis of experiment results will be presented together with discussion and implication analysis.

### A. Data and Experiment Setup

In this study, we collected a set of digital recording devices with microphone built in to collect audio samples. Totally, 5 devices are gathered as shown in Table I.

TABLE I
A set of collected Microphones

| Microphone ($M_i$) | Device Type | Model | Manufacturer |
|---|---|---|---|
| $M_1$ | Audio Recorder | NWZ-B142F | Sony |
| $M_2$ | Audio Recorder | LX100 | Creative |
| $M_3$ | Audio Recorder | GoGear Mix | Philips |
| $M_4$ | Camera | PowerShot G5 | Canon |
| $M_5$ | Camera | OptioS6 | Pentax |

These microphones were used to collect audio samples at different locations such as *indoor*, *quiet park* and *busy street* between $9am$ to $5pm$ to form an audio corpus for our study. All audio samples were recorded as a mono signal at 8kHz sampling frequency with 16-Bit quantization. Then, MFCCs are extracted from each segment of 0.25 seconds in length (non-overlapping) to form an instance with 13 vector features. In each experiment, a number of instances (1000 in our case) are extracted from each microphone samples equally and combined together to form the test data set.

At the training stage, one model $o_i$ of OCC algorithm $O$ is built for each microphone, thus, $m$ microphones require $m$ OCC models $o_1, \ldots, o_m$ to be built. Then, they are applied to the test set to detect audio samples belonging to their target class. Since we are only interested in the absolute performance of OCC algorithms on "target" data, *accuracy* rate and *error* rate are used to measure their detection capability.

Suppose there are $P$ audio samples of target class and $N$ audio samples of outlier class in the test set for microphone $M_i$, among which $P_a$ target samples are recognized correctly, and $N_r$ outliers are recognized incorrectly as target class, then the *accuracy* and *error* of OCC model $o_i$ on $M_i$ are calculated as:

$$accuracy = \frac{P_a}{P} \quad (6a)$$

$$error = \frac{N_r}{N} \quad (6b)$$

To evaluate the overall performance of $O$, we use *overall accuracy* and *overall error* which are averages of *accuracy* values and *error* values over $m$ models $o_1, \ldots, o_m$ of $O$. In a classification task, the prediction result is expected to be better than a random guess (above 0.5 for *accuracy* score). The *accuracy* is considered to be higher if it is closer to 1.0.

In the experiment, the OCC algorithms mentioned in section III will be used. Parameters of each classifier are kept as default values in the OCC toolbox [16]. On the training set, the rejection rate of the target class was set to 10% in order to provide a tight decision boundary around the target class.

### B. Result Analysis

To assess the performance of OCC for microphone detection, we perform three major experiments with audio samples at different noise levels corresponding to different recording locations. In these cases, the training sets are sampled to be equal to testing set of 1000 instances. Considering the fact that noisy environment is usually more difficult than quiet environment, we perform another experiment for the case of *busy street* with different number of training instances to examine if the performances of OCC algorithms can be improved by incorporate more training samples.

*1) Microphone Identification for indoor recordings:* The recorded audio samples in this case were taken from *indoor* environment such as small room, big room, building and lecture theatre where little background noise is presenting.

All 6 OCC algorithms were trained and tested on the same test set extracted previously. The experiment results are shown in Table II.

TABLE II
MICROPHONE IDENTIFICATION RESULT FOR INDOOR ENVIRONMENT RECORDINGS

| Algorithm | | $M_1$ | $M_2$ | $M_3$ | $M_4$ | $M_5$ | *Overall* |
|---|---|---|---|---|---|---|---|
| **1-GN** | Accuracy | 0.873 | 0.892 | 0.897 | 0.685 | 0.867 | **0.843** |
| | Error | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 |
| **1-GNM** | Accuracy | 0.830 | 0.800 | 0.858 | 0.543 | 0.842 | 0.774 |
| | Error | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 |
| **1-KM** | Accuracy | 0.885 | 0.881 | 0.896 | 0.623 | 0.893 | **0.836** |
| | Error | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 |
| **1-KNN** | Accuracy | 0.867 | 0.894 | 0.896 | 0.626 | 0.897 | **0.836** |
| | Error | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 |
| **1-PCA** | Accuracy | 0.908 | 0.807 | 0.897 | 0.767 | 0.915 | **0.859** |
| | Error | 0.061 | 0.052 | 0.003 | 0.148 | 0.000 | 0.053 |
| **1-ISVM** | Accuracy | 0.907 | 0.818 | 0.916 | 0.750 | 0.872 | **0.853** |
| | Error | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 |

In general, all algorithms achieved high *overall accuracy* in identifying target microphones as indicated by the values of over 0.8, only except for 1-GNM with lower detection rate of 0.744. Importantly, almost no error was made across the models as indicated by *overall error* value of 0, only a small error is shown for 1-PCA.

As each microphone $M_i$ is identified by a unique model $o_i$, the assessment of OCC algorithm $O$ needs to be considered together with the performance of each individual model corresponding to each microphone. All six algorithm achieved high *accuracy* and no *error* in detecting microphones $M_1$, $M_2$, $M_3$ and $M_5$, while it is harder to recognize microphone $M_4$ with significantly lower *accuracy* values. Even so, 1-PCA and 1-ISVM achieve considerably good *accuracy* for $M_4$ of above 0.750, especially for 1-ISVM which has no error at all.

These results indicate that the OCC algorithms are suitable for detecting and verifying microphone models in indoor environment. Further evaluation for OCC will be carried out in the next experiment with more noisy audio samples.

*2) Microphone Identification for quiet park recordings:* In this case, the audio records were taken from a *quiet park* with considerable background noise. Training and testing data sets were both extracted from these audio records, and then similar evaluation process was applied with the same parameter values for each algorithm. The results are shown in Tables III.

TABLE III
MICROPHONE IDENTIFICATION RESULT FOR QUIET PARK ENVIRONMENT RECORDINGS

| Algorithm | | $M_1$ | $M_2$ | $M_3$ | $M_4$ | $M_5$ | *Overall* |
|---|---|---|---|---|---|---|---|
| **1-GN** | Accuracy | 0.750 | 0.916 | 0.956 | 0.911 | 0.893 | **0.885** |
| | Error | 0.001 | 0.090 | 0.001 | 0.174 | 0.008 | 0.055 |
| **1-GNM** | Accuracy | 0.506 | 0.751 | 0.822 | 0.756 | 0.838 | 0.735 |
| | Error | 0.000 | 0.016 | 0.000 | 0.029 | 0.000 | 0.009 |
| **1-KM** | Accuracy | 0.768 | 0.907 | 0.926 | 0.897 | 0.923 | **0.884** |
| | Error | 0.001 | 0.056 | 0.003 | 0.135 | 0.001 | 0.039 |
| **1-KNN** | Accuracy | 0.648 | 0.810 | 0.866 | 0.820 | 0.887 | **0.806** |
| | Error | 0.000 | 0.024 | 0.000 | 0.032 | 0.000 | 0.011 |
| **1-PCA** | Accuracy | 0.795 | 0.831 | 0.932 | 0.873 | 0.841 | **0.854** |
| | Error | 0.229 | 0.121 | 0.002 | 0.201 | 0.069 | 0.124 |
| **1-ISVM** | Accuracy | 0.871 | 0.969 | 0.954 | 0.973 | 0.878 | **0.929** |
| | Error | 0.003 | 0.300 | 0.221 | 0.353 | 0.066 | 0.188 |

It is interesting to see that the *overall accuracy* of the algorithms are not significantly different from previous experiment. However, we notice that more errors were made as indicated by higher *overall error* values. In particular, 1-SVM outperformed other algorithm in detecting microphones with *overall accuracy* of 0.929,

however, it also made signification number of wrong prediction as indicated by *overall error* value of 0.188.

In view of individual detection model, none of them failed to detect their target microphones as shown by high *accuracy* and low *error* values for $M_1 - M_5$. Even for 1-GNM scoring lowest *accuracy* of 0.506 on $M_1$, it is still accepted as we are interested in the detection capability for target class of OCC models.

Despite some errors were made, OCC algorithms were able to identify microphone models for audio samples taken from outdoor (*quiet park*) environment. Further evaluation of OCC approach will be performed for extremely noisy audio records in the next section.

*3) Microphone Identification for busy street recordings:* In this experiment, the audio samples were obtained from a *busy street* environment with significant amount of noise presenting. Audio features were extracted and input to OCC algorithms as in previous experiments. The detection results are shown in Table IV.

TABLE IV
MICROPHONE IDENTIFICATION RESULT FOR BUSY STREET ENVIRONMENT RECORDINGS

| Algorithm | | $M_1$ | $M_2$ | $M_3$ | $M_4$ | $M_5$ | *Overall* |
|---|---|---|---|---|---|---|---|
| **1-GN** | Accuracy | 0.787 | *0.263* | *0.332* | 0.743 | 0.688 | 0.563 |
| | Error | 0.247 | 0.082 | 0.132 | 0.435 | 0.234 | 0.226 |
| **1-GNM** | Accuracy | 0.320 | 0.143 | 0.188 | 0.306 | 0.331 | *0.257* |
| | Error | 0.011 | 0.001 | 0.002 | 0.047 | 0.011 | 0.014 |
| **1-KM** | Accuracy | 0.555 | *0.481* | *0.384* | 0.514 | 0.625 | 0.512 |
| | Error | 0.108 | 0.016 | 0.036 | 0.111 | 0.092 | 0.073 |
| **1-KNN** | Accuracy | 0.175 | 0.149 | 0.185 | 0.259 | 0.262 | *0.206* |
| | Error | 0.007 | 0.001 | 0.003 | 0.026 | 0.006 | 0.008 |
| **1-PCA** | Accuracy | 0.787 | *0.263* | *0.332* | 0.743 | 0.688 | 0.563 |
| | Error | 0.247 | 0.082 | 0.132 | 0.435 | 0.234 | 0.226 |
| **1-ISVM** | Accuracy | 0.804 | 0.868 | 0.506 | 0.876 | 0.865 | **0.784** |
| | Error | 0.230 | 0.027 | 0.081 | 0.322 | 0.255 | 0.183 |

From Table IV, we can see that 1-GNM and 1-KNN failed in identifying their target microphones as indicated by *overall accuracy* of below 0.5. Although, the *overall accuracy* values of 1-GN, 1-KM and 1-PCA are higher than 0.5, they failed to detect microphone $M_2$ and $M_3$ as shown by *accuracy* values of less than 0.5.

On the contrary, 1-ISVM outperformed all others and was able to detect all microphone models with *overall accuracy* of 0.784, especially, none of its individual models has *accuracy* value lower than 0.5. This evidence shows that 1-ISVM is a potential candidate for microphone forensics for signals obtained from noisy environment.

*4) Effects of training sample size:* In this experiment, we assess the performance of OCC algorithms in case of noisy environment (*busy street*) with increasing number of training samples in each iteration to examine if bigger training set can improve their performance. The training set was started from 1000 instances and increased by 100 instances in each iteration. In this case, we would like to evaluate the overall performance of those algorithms, therefore, only *overall accuracy* and *overall error* values are presented, as shown in Fig. 3.

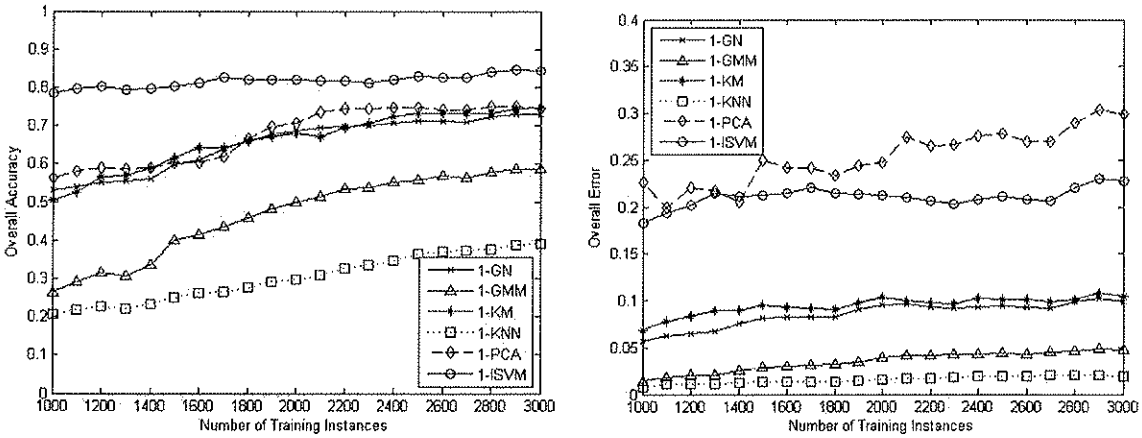
From Fig. 3(a), we can see that, generally, the *overall accuracy* of OCC algorithms are increasing steadily as the number of training instances increasing. Whilst, 1-GN, 1-KM and 1-PCA are stabilizing from training size of 2400 instances, the *overall accuracy* values of 1-GNM and 1-KNN continue to growth slightly. In contrast, no significant improvement is found for 1-ISVM as its detection rate stays around 0.8 even when the training set is tripled to 3000 instances.

In Fig. 3(b), roughly growth in *overall error* rate is found for 1-PCA, while that of 1-SVM is increasing gradually until training size reach 1300 instances then stylizing between 0.2 and 0.25. Similarly, the *error* rates of 1-GA and 1-KM increase slightly until training size reach 2000 instances then become stable of around 0.1. At the same time, error rates of 1-GNM and 1-KNN are also increased but remained lower than 0.05 even when the training data is tripled.

In summary, the above result shows that the increase of training size can improve performance on most OCC algorithms on noisy audio recordings. Especially for 1-KNN and 1-GNM, their performance is significantly improved as their accuracies keep increasing while their error rates are stabilizing at low value. Although, the performance of 1-SVM is not improved considerably, it still achieved highest accuracy rate among the tested algorithm.

### C. Discussion

The experiment results in section IV-B1 and IV-B2 support the claim that *One Class Classifier* (OCC) is a suitable approach to microphone forensics. Most algorithms achieve high performance on audio samples that are recorded from little noise environment, as the recorded signals are not influenced significantly by outside signal, and can thus reflect better microphone characteristics. Although, the quality of OCC algorithms are reduced when applying to noisy audio signal as presented in IV-B3, their performance can be improved by increasing the number of instances in training set as demonstrated in IV-B4.




(a) (b)

Fig. 3. Effects of different training sample sizes

The key advantage of the OCC approach is that they focus on the detection of target class instances by using tight decision boundary obtained from training data. As consequence, only a small number of fault detection cases are made as indicated by low *error* rate across the OOC models in previous experiments. In addition, these OCC models only need to be trained once, and are independence from each other. In other words, when a new microphone become available, only a new OCC model trained on audio samples of that microphone is required without retraining existing OCC models. For such reasons, the approach using OCC is more practical, cost effective and easier for system maintenance than traditional supervised learning.

Besides, we notice that the OCC algorithms can perform well on little noise audio signal, a pre-processing step comes out obviously for microphone forensic practitioners is to extract only silence frames from testing audio records for detection purpose. In cases of noisy environment where very few or no silence frame exists, a method to identify least noisy frames should be developed to improve the detection capability of these algorithms.

## V. Conclusions

Microphone identification and verification are important tasks in integrity and authenticity assurance of information which are becoming more and more crucial nowadays in criminal investigation. However, only a limited number of approaches have been used in the literature which utilize the machine learning methods to support for microphone forensic practitioners in doing this work. Current applications of supervised learning are still facing huge challenges due to time and associated cost of collecting audio samples from a large number of microphones for model training purposes; In addition, frequent retraining of existing classification models is required. In this paper, we present the first attempt in automated microphone detection using One-Class Classification approach which exhibits to be effective in alleviate the challenges of microphone forensics.

To be precise, we presented an evaluation of 6 relatively new OCC algorithms in detecting microphone models under different conditions of noise level. Experiment results indicate that the tested OCC algorithms are able to detect microphone model with high accuracy and low error rate. Among them, 1-SVM presented to be the best algorithm for microphone identification, especially when the recorded audio signal is noisy. Further experiment on the different training sample sizes showed that the capability of OCC algorithms can be improved by increasing number of instances. This study provides evidence for effectiveness and efficiency of the OCC approach, which we have shown to be suitable for application in real life situations.

A natural extension of this work will be testing OCC algorithms on a wide range of microphone models and under more sophisticated recording scenarios. Furthermore, we will investigate the extent to which silence frame detection, determination of the least noisy frame, can assist in improving the performance of OCC algorithms.

## ACKNOWLEDGMENT

The authors would like to thank Professor Lynn Batten, of School of Information Technology at Deakin University, for proposing and funding this research project.

## REFERENCES

[1] C. Kraetzer, A. Oermann, J. Dittmann, and A. Lang, "Digital Audio Forensics : A First Practical Evaluation on Microphone and Environment Classification," in *Proceedings of the 9th workshop on Multimedia and security*, Dallas, Texas, September 2007, pp. 63–74.

[2] C. Kraetzer, M. Schott, and J. Dittmann, "Unweighted fusion in microphone forensics using a decision tree and linear logistic regression models," in *Proceedings of the 11th workshop on Multimedia and security*. Princeton, New Jersey, USA: ACM Press, September 2009, pp. 49–56.

[3] R. Buchholz, C. Kraetzer, and J. Dittmann, "Microphone Classification Using Fourier Coefficients," *Lecture Notes in Computer Science*, vol. 5806, pp. 235–246, 2009.

[4] D. Garcia Romero and C. Y. Espy Wilson, "Automatic acquisition device identification from speech recordings," in *Proceeding of IEEE International Conference on Acoustics Speech and Signal Processing*, Dallas, Texas, US, March 2010, pp. 1806–1809.

[5] C. Kraetzer, K. Qian, M. Schott, and J. Dittmann, "A context model for microphone forensics and its application in evaluations," *Media Watermarking, Security, and Forensics III*, vol. 7880, p. In Press, 2011.

[6] A. Rabaoui, H. Kadri, Z. Lachiri, and N. Ellouze, "One-Class SVMs Challenges in Audio Detection and Classification Applications," *EURASIP Journal on Advances in Signal Processing*, vol. 2008, pp. 1–14, 2008.

[7] A. Rabaoui, M. Davy, S. Rossignol, and Z. Lachiri, "Improved one-class svm classifier for sounds classification," in *Proceeding of IEEE Conference on Advanced Video and Signal Based Surveillance*, London, United Kingdom, September 2007, pp. 117–122.

[8] G. FengJuan, S. ShuQian, and W. XiaoHui, "Using One-Class SVMs and MP for Audio Recognition of Action Scenes," in *Proceeding of 2nd International Workshop on Education Technology and Computer Science*, Wuhan, China, March 2010, pp. 401–404.

[9] A. Brew, M. Grimaldi, and P. Cunningham, "An evaluation of one-class classification techniques for speaker verification," *Artificial Intelligence Review*, vol. 27(4), pp. 295–307, 2008.

[10] M. O. AlQahtani, G. Muhammad, and Y. A. Alotaibi, "Environment Sound Recognition using Zero Crossing Features andl MPEG-7," in *Proceeding of Fifth International Conference on Digital Information Management*, Thunder Bay, Ontario, Canada, July 2010, pp. 502–506.

[11] N. Sen, T. K. Basu, and H. A. Patil, "Significant Improvement in the Closed Set Text- Independent Speaker Identification Using Features Extracted from Nyquist Filter Bank," in *Proceeding of 5th International Conference on Industrial and Information Systems*, India, Jul 2010, pp. 303–308.

[12] I. L. Freire and J. A. Apolinario Jr., "Gunshot detection in noisy environments," in *Proceeding of the 7th International Telecommunications Symposium*, Manaus, Brazil, September 2010.

[13] P. Dhanalakshmi, S. Palanivel, and V. Ramalingam, "Classification of audio signals using AANN and GMM," *Applied Soft Computing*, vol. 11, no. 1, pp. 716–723, Jan. 2011.

[14] M. O. Alqahtani and A. S. Al mazyad, "Environment Sound Recognition for Digital Audio Forensics Using Linear Predictive Coding Features," in *Proceeding of International Conference on Digital Information Processing and Communications*, Ostrava, Czech Republic, July 2011, pp. 301–309.

[15] J. Ye, "Speech recognition using time domain features from phase space reconstructions," Ph.D. dissertation, Marquette University, Milwaukee, Wisconsin, May 2004.

[16] D. Tax, "Data description toolbox dd tools 1.7.5," Delft University of Technology, Delft, The Netherlands, Tech. Rep., May 2010.

[17] D. M. Tax and R. P. Duin, "Support vector domain description," *Pattern Recognition Letters*, vol. 20, pp. 1191–1199, 1999.

[18] A. Rabaoui, M. Davy, S. Rossignol, and N. Ellouze, "Using one-class svms and wavelets for audio surveillance," *IEEE Transactions on Information Forensics and Security*, vol. 3(4), pp. 763–775, 2008.

**Paper 5: Image Source Detection: A Case Study on Facebook Images Taken by iPhones**

Lei Pan and Nijaz Trepanic, School of Information Technology, Deakin University.

# Image Source Detection: A Case Study on Facebook Images Taken by iPhones

Lei Pan
*School of Information Technology*
*Deakin University*
*Burwood, VIC 3125, Australia*
*Email: l.pan@deakin.edu.au*

Nijaz Trepanic
*School of Information Technology*
*Deakin University*
*Burwood, VIC 3125, Australia*
*Email: trepanicn@gmail.com*

## Abstract

*Forged and tempered digital images become increasingly common on Facebook to aid computer frauds. The situation is worsened as many users can use a phone to take a photo and upload it to Facebook within two clicks, which highlights the need of image forensics for the cyber fraud cases. In this paper, we show the existence of the Facebook image filter which automatically changes the Facebook photos and consequently challenges the validity of forensic results. We aim to enable forensic investigators to relate a seized camera and a Facebook image. Specifically, we utilize intrinsic sensor pattern noise produced by a camera's lens to derive forensically useful information as Photo Response Non-Uniformity (PRNU) patterns. We propose to compare the PRNU patterns of a Facebook image and the flat field images produced by the candidate cameras. And we conclude this method to be effective by successfully identifying the correct iPhone from a list of four for a given Facebook image.*

## Index Terms

*Image Forensics, Photo Response Non-Uniformity (PRNU), Facebook Image Filter, Apple iPhone.*

## 1. Introduction

Social networking has exploded over the last 10 years; the euphoric rise in popularity of hundreds of social networking sites is astonishing. Facebook perhaps more than others stands out as the most popular social networking site on the planet in 2011; Facebook have over 600 million registered and active users [1]. With the popularity come the concerns, Facebook, much like all other social networking sites is a publicly accessible user-profile based social networking web site. With billions of photos uploaded to Facebook every month the question of image integrity and validity is one that arises frequently in the world of digital image forensics.

Digital image forensics becomes prosperous due to the exponential increase of image fraud cases. The planning and process of exploitation used by cyber frauds or certain group and individuals are beyond anything that was thought possible 15 to 20 years ago. Digital image forensics exploits the traces of image processing algorithms or the characteristics that are introduced during the acquisition process when a digital image is captured according to [2]. Digital image forensics is a sub division of computer forensics, [3] observes most definitions for computer forensics include the identification, prevention and the analysis of digital information that is stored and transmitted by a computer system. The main purpose of computer forensics is to establish and to verify the validity of the hypothesis testing used in an attempt exploit the condition of an incident that is under investigation [3].

However this theory is disagreed by [4] stating that image forensics is merely a component of multimedia forensics. Computers can be diversely used in various crimes, and can be used as an effective instrument to commit crimes in the real world, hence the differentiation between computer forensics and multimedia forensics. According to [5], there are three common problems in digital image forensics: Image source identification determines which acquisition device acquired a given digital image and to match the image to an individual source device; discrimination of images distinguishes the real-life images from computer generated images; and image forgery detection determines whether an image has been manipulated or processed after it was captured by an acquisition device like a digital camera.

Among the three problems, image source detection is most valuable to digital forensic investigators because it offers the connection between a digital image and a physical device.

We propose a theory for determining the authenticity and integrity of images downloaded from a social networking site. Our fundamental work is based on the Facebook image filter. We seek to determine whether we can relate the processed Facebook images to their respective source acquisition device. We propose to use sensor pattern noise in particular photo-response non-uniformity noise to identify correlation patterns that link an image to its source acquisition device. We implement the use of camera fingerprint reference images as a comparison tool for natural images to draw parallels to.

Given the recent popularity of social networking sites, and the overwhelming use of mobile based cameras, uploading images to Facebook can be done instantly, in fact Facebook phone applications have options that allow for taking images whilst on Facebook on the mobile device, moreover the posting of the images is done instantly. It is for this particular reason that the Apple iPhone 3GS camera was predominantly used within this study.

Our approach compares a variety of cameras against natural and manipulated images; we use the Apple iPhone 3GS camera to conduct a majority of our testing, due to the high volume in numbers of the phone camera in existence. Our method for identifying an image involves comparing fingerprint reference images against the image under investigation. We conduct four image comparison tests when attempting to gather a correlation between a given image under investigation and the fingerprint reference images which serves as the basis for comparison. The four detailed tests include:

1) Comparing an image under investigation against the fingerprint reference images;
2) Comparing an image under investigation that is resized to a resolution of $(720 \times 540)$ against the fingerprint reference images also resized to a resolution of $(720 \times 540)$;
3) Comparing an image under investigation that has been through the Facebook image filter against the fingerprint reference images that have also been through Facebook image filter; and,
4) Comparing an image under investigation that has been through the Facebook image filter against the fingerprint reference images resized to a resolution of $(720 \times 540)$.

The rest of this paper is organized as follows: Section 2 lists the related work on device source identification and PRNU methods; Section 3 shows the existence of the Facebook image filter and presents our method of detecting the source device; Section 4 describes our experiments and the results; and we conclude in Section 5.

## 2. Related Work

Source device identification is a major component of digital image forensics. Contemporary cyber frauds utilize the complexity of digital images. But we can link an image to both the camera make and the camera model. Chi and Hongbin [6] propose to inspect the EXIF header of a digital image to identify the source camera of an image. Generally speaking, all the relevant information about any digital image can be stored in the EXIF header such as date, focal length, resolution and so on. However, the EXIF header information is unreliable for forensic investigators due to various reasons: Many image editing software tools can modify or erase the EXIF information [6]; secondly, EXIF header is not a compulsory part of a digital image and hence can be omitted without causing any display issue; thirdly, cyber criminals deliberately use image filtering, modification enhancement and tampering to change the EXIF header.

Source camera identification helps us to identify the digital acquisition device which includes digital cameras, camcorders, scanners, mobile phones and so on [5]. Different outcomes can be measured with source camera identification. The detection, classification and measurement of the qualities associated with spatial structures including colour, texture and edge structures of an image may be used to trace an image acquisition device [7]. Many different techniques and methods can be used to exploit the source camera identification of an image. The first method proposed by [8] emphasizes the use of intrinsic lens radial distortion for automatic source camera identification. This method focuses on the use of radial distortions that serve as a unique fingerprint, and the extraction of parameters from irregularities are also measured and compared in order to obtain the source acquisition device of a digital camera. The second method proposed for source device identification focuses on a set of characteristics of a specific digital camera such as average pixel values, RGB pairs correlation, neighbor distribution center of mass, RGB pairs energy ratio, and wavelet domain/algorithm statistics [9].

To identify a source acquisition device, sensor pattern noise has been proven to produce more accurate and reliable results than the above methods. Many researchers use some form of sensor noises such as Photo Response Non-Uniformity (PRNU) for source device identification. [10]. PRNU can be

used to accomplish the following forensic tasks — source device identification, device linking, recovery of processing history, and detection of digital forgeries [10]. PRNU noise pattern is regarded an almost true signature of digital cameras [11], [12]. PRNU is so representative because each digital image consists of imperfect digital signals due to the fact that digital camera sensors contain defects [13]. Therefore, PRNU is treated as a fundamental property contained in all digital imaging sensors [10].

Li and Li [11] describe a two-step approach using PRNU noise pattern for source camera identification in which, firstly the PRNU noise patterns are extracted from a host of low contrast digital images taken from an imaging device, then averaging these images to calculate a reference fingerprint pattern so that PRNU comparisons can be drawn upon. This process can also be described as flat-field imaging, where we take a set of 'flat' images all in the same orientation and usually consisting of the same backdrop. For example if we take a set of ten flat field images with a single camera of a white background, all images should look very similar in our bare eyes. According to [11], we can extract a PRNU pattern from an image under investigation and compare this pattern with the reference fingerprint pattern derived from the flat field images. Using the process of flat fielding we can theoretically link every image to the source camera that took the particular image that is under investigation, provided that we have access to flat field images of every camera. Therefore, if we had a sample fingerprint reference of every camera model that exists in this world, in the form of flat field images, we would be able to link the PRNU noise pattern extraction of images to that particular camera. A similar version of this theory is described in [14] as linking the averaged residues in the form of reference patterns of a digital camera to a digital camera whose noise pattern is extracted.

We believe that the use of flat fielding can be effective for identifying images based on the comparative results of the reference pattern obtained from the flat field images and a natural image but can yet be problematic. Hu, Jian and et al. [15] describe that a camera fingerprint reference pattern may suffer from one unavoidable problem arose in PRNU-based method, that is, camera fingerprint references are constructed in single color channel instead of all possible color channels. This defect indicates that the detection process is incomprehensive and thus the verdict is less reliable. The primary issue of this process is that image sensors such as the Charge Coupled Device (CCD) may not be able to comprehensively compare the noise patterns that are extracted from within the image.

To produce more reliable results, wavelet-based algorithms can be used to extract of PRNU noise patterns within both natural and flat-field images. The research by [16] establishes the theory of using the wavelet based algorithm for the use in flat-fielding also known in the digital imaging field as a camera fingerprint reference pattern. The use of the wavelet-based algorithm produces much more accurate results than any other de-noising filters, because the noise residuals that have been obtained, contain the least amount of scene traces (scene traces are those areas around the edges of images, which are usually misinterpreted by other de-noising filters) [16]. Furthermore, wavelet-based transformations allow for individual frequency components of a digital images to be extracted producing a range of high and low frequency bands [17].

The aptitude to compare flat-field images as a reference to natural images gives a moderate to strong indication of which camera takes the respective digital image once compared to the other images and a group of candidate cameras. Highlighted in the experiments conducted in our research, the ability to easily relate an image to its respective source camera is a unique feature of the wavelet based algorithm, alongside with the use of flat-fielded images as a reference pattern. The results are comprehensive in that we successfully identify the camera by comparing the images to the fingerprint reference images.

The next Section will show the existence of the Facebook image filter and present our research methodology.

## 3. Facebook Image Filter and Research Methodology

We discover the existence of Facebook image filter by examining its changes to an image once the image is uploaded to the Facebook server. To determine the exact extent the Facebook image filter performs, we examine and compare the images in terms of visual quality, binary contents of the image file and textual information in the image file header.

To demonstrate the results of the above tests, we use an iPhone photo shown in Fig 1 as an example. By viewing the photos in the Windows image viewer program, we observe an obvious reduction in visual quality because this iPhone photo's resolution is reduced from $2048 \times 1536$ to $720 \times 540$ after being uploaded to Facebook.

Important details extracted by an EXIF reader from the above image are make, model, orientation, software, date and time, exposure time, exposure program, ISO speed ratings, shutter speed value, aperture value, flash, focal length, subject location, color space, exposure mode, white balance and

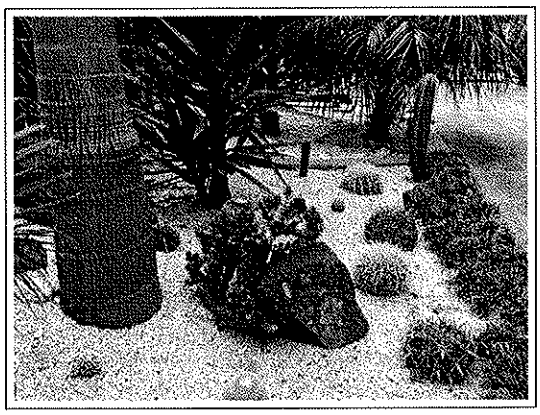


Figure 1: An iPhone Image

sharpness. The complete EXIF header of the photo shown in Fig 1 is displayed as follows:

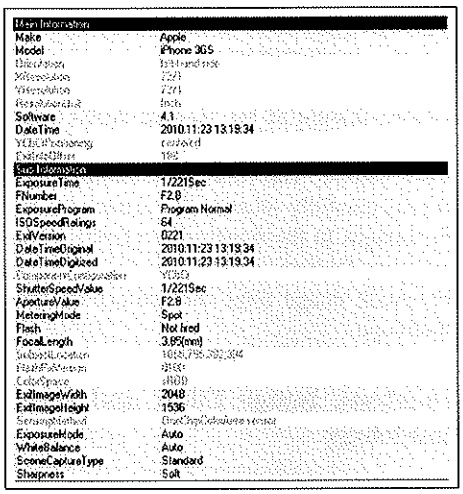



Figure 2: The EXIF Header of the iPhone Image

Unfortunately, we note that the image downloaded from Facebook does not contain any EXIF header file information. Furthermore a comparison of the properties boxes was analyzed to show how the Facebook image filter alters the information. We note that the vendor information is present in Fig 3a, and absent in Fig 3b after the photo is uploaded to Facebook.

We use a HEX editor to collect bit-level information of the images. We visualize how the image is constructed with respect to the camera settings when the photo is taken. We note that important meta-information obtained in reading the original photo include camera type, camera model, time and date, and current firmware version in Fig 4a; and Fig 4b shows evident changes to the image downloaded from Facebook as meta-information disappears.

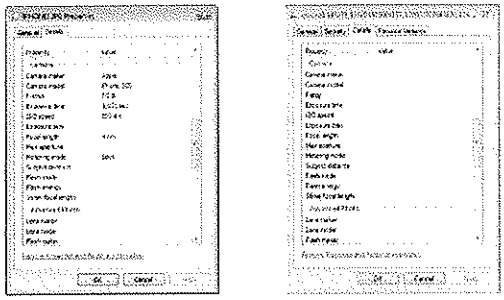


Figure 3: Properties Boxes of the Images

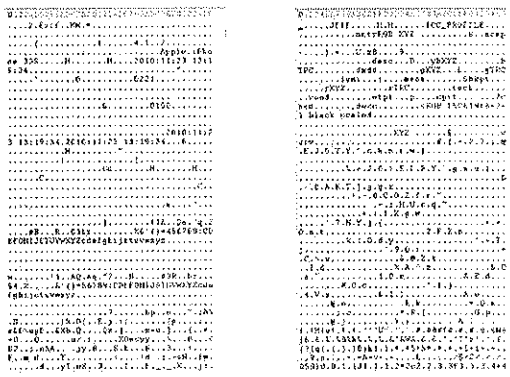


Figure 4: HEX Content Readings

The above results clearly show that Facebook employs an image filter altering natural images. Observing these differences helps us little to identify the source device. As described in Section 2, PRNU patterns reveal a given camera's intrinsic features. We want to test and establish how the specific PRNU noise patterns of a camera are affected by resizing the image and uploading the image through the Facebook image filter. Hence we develop the following methodology to identify the source device of a Facebook image —

Step 1. Take a `natural image` and ten `flat field images` by using a camera.

Step 2. Compare PRNU patterns of `natural images` against of `flat field fingerprint reference images`.

Step 3. Resize the `natural image` and the `flat field images` to $720 \times 540$ by using the *ImageMagick* program [18] and compare PRNU patterns of `resized natural images` against of `resized flat field fingerprint reference images`.

Step 4. Upload the natural image to Facebook, download the `Facebook natural image`, and compare PRNU patterns of `Facebook natural images` against of `resized flat field fingerprint reference images`.

Step 5. Upload the ten flat-field images to Facebook, download `Facebook flat field fingerprint reference images`, and compare PRNU patterns of `Facebook natural images` against of `Facebook flat field fingerprint reference images`.

This procedure is demonstrated in a flowchart in Fig 5.

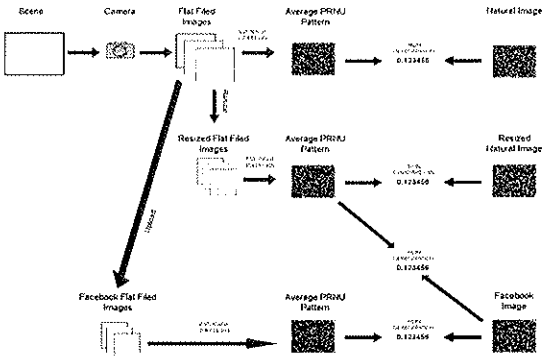



Figure 5: Flowchart of Image Comparisons

The flowchart shows the process of how the average patterns are obtained, how the respective patterns are compared to the natural images, and which comparisons are made. Note we compare the `resized flat filed fingerprint images` to both the `resized natural image` and the `Facebook natural image`. The next section reports an experiment that identifies source camera.

## 4. Experiment and Analysis

The following experiment distinguishes which camera (from a set of 4 cameras, all of the same model) took a particular Facebook image. We test the PRNU noise patterns of a particular camera and then compare PRNU noise patterns to another camera with same make and model but a different device. We follow the procedure in Fig 5.

### 4.1. Devices and Photos

We choose to use four Apple iPhone 3GS cameras for this experiment, because iPhone is dominant in the mobile phone market and can directly take a photo and upload it to Facebook. We name these four cameras as $Cam1$ (Serial No. 83562523FHI), $Cam2$ (Serial No. 870208533NQ), $Cam3$ (Serial No. 880287XQEDG), and $Cam4$ (Serial No. 8793962D3NR). And we take 40 `natural images` by using these 4 cameras, that is, 10 photos for each camera.

The *NFI PRNU compare* program [19] reports a fairly even correlation between all four Apple iPhone 3GS cameras as shown in Table 1. The table presents the three color channels which are reflected in the color filter array of every digital camera, the higher the reading in each color channel the higher the correlation of between the comparisons with respect to the same channel. Overall, we concentrate on the Sum value which gives the best indication of the correlation between our comparisons.

| Comparison | Red | Green | Blue | Sum |
|---|---|---|---|---|
| Cam1 & Cam2 | 0.003892 | 0.004981 | 0.004587 | 0.013460 |
| Cam1 & Cam3 | 0.003432 | 0.002805 | 0.002684 | 0.008921 |
| Cam1 & Cam4 | 0.004439 | 0.004380 | 0.003431 | 0.012250 |
| Cam2 & Cam3 | 0.002651 | 0.003458 | 0.003615 | 0.009724 |
| Cam2 & Cam4 | 0.004117 | 0.004397 | 0.004189 | 0.012702 |
| Cam3 & Cam4 | 0.007502 | 0.007375 | 0.007333 | 0.022210 |
| Average | 0.004339 | 0.004566 | 0.004307 | 0.013212 |

Table 1: PRNU Correlation between Apple iPhone 3GS Cameras

The table presents the readings of the three primary color channels that are present in every single digital camera; a higher correlation in each color channel represents a higher degree of linear similarity between the two patterns. The overall sum indicated the comparison correlations between the two extracted patterns. Thus, the comparison results between the four cameras are close to the average value of 0.013212. Similarly, we obtain results for the three color channels (red, green, blue), in all camera comparisons which are also close to the average reading of (0.004339, 0.004566, and 0.004307) respectively. We note that the two cameras which have less correlation are $Cam1$ and $Cam3$ (0.008921) as well as $Cam2$ and $Cam3$ (0.009724). However $Cam1$ and $Cam2$ (0.013460) have a good correlation that is bearing a resemblance to the average of 0.013212. This could explain that $Cam3$ tends to have a poor correlation with other cameras, though $Cam3$ and $Cam4$ have a relatively strong correlation, in fact the strongest correlation of (0.022210), which is well above the average.

To generate a fingerprint reference pattern, we shoot `flat field images` of a white board in standard day-time lighting with default camera settings on all four cameras. To ensure reproducibility, we use each camera to take ten reference fingerprint images. Once the flat fielding images from the four cameras are obtained, we extract the average pattern of each camera by applying the *NFI PRNU compare* program (developed by the *Netherlands Forensic Institute* [19] and implemented by [13]). When the average PRNU noise patterns are obtained from each camera we are presented with similar readings. There

is an even spread of noise right across the scale and the four cameras to the human eye appear to be the same in content. The average patterns from all four cameras give a similar visual reading; this is common due to all cameras capturing ten images of the exact same content. For example, a PRNU pattern of a flat-field image taken by $Cam1$ is shown in Fig 6.

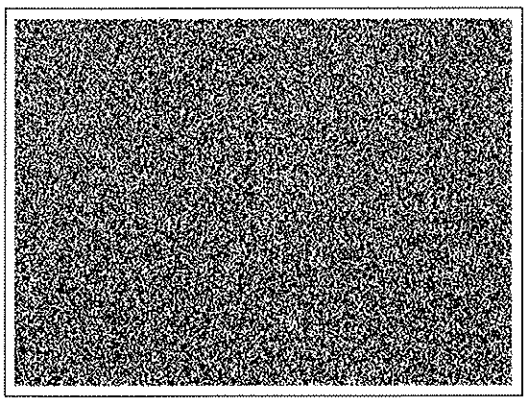


Figure 6: PRNU of a Flat-field Image by $Cam1$

In order to ensure the reliability and to simplify the experiment, we select the wavelet algorithm implemented in the *NFI PRNU compare program* to extract the PRNU pattern from the reference images. In particular, the extractions are a result of ten flat field images averaged out using the wavelet-based algorithm to obtain the PRNU noise value of each image. Such fingerprint reference patterns serve as a comparison, as there is no value stored in them. Furthermore, these patterns are only useful when we compare something against them. In order to effectively determine how the four cameras differ in PRNU noise we run a comparison of all the cameras against each and determine if we can establish any correlation between the four Apple iPhone 3GS cameras.

Having derived the fingerprint patterns, we resize the captured photos to the Facebook resolution of 720 × 540. Specifically, we firstly resize every flat field image from a resolution size of 2048 × 1536 to 720 × 540 as this is the default image resolution size that Facebook accepts; we then extract the average PRNU noise pattern from the 10 resized flat field images to create another fingerprint reference pattern (these will be referred to as `resized flat field fingerprint reference images`). We then upload all ten flat field images to Facebook; these images are downloaded in a resolution size of 720 × 540. Furthermore we extract the average PRNU noise patterns of the ten flat field images that were uploaded to Facebook and use these as a fingerprint reference pattern for the images from Facebook, (these images will be referred to as `Facebook flat field fingerprint reference images`).

### 4.2. Identifying an "Unknown" Image

Using one of the iPhone cameras, we take a natural image in a golf course shown in Fig 7. Clearly visible is the amount of green color present in the image.

Figure 7: An Unknown Natural Image

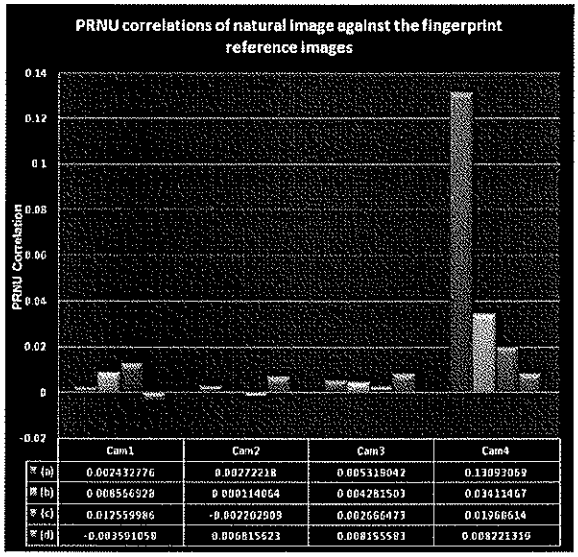
We extract this natural image's PRNU noise levels and compare it against the `flat field fingerprint reference images` generated in the previous subsection. The natural image is then resized to the Facebook standard resolution size of 720 × 540; then the resized image had its PRNU noise level extracted, and a comparison is run against the `resized Facebook flat field fingerprint reference images`. Finally to complete the experimental testing, the `natural image` is put through the Facebook image filter, and the PRNU noise levels are extracted and a comparison against the `Facebook flat field fingerprint reference images` is conducted.

The experiment has four sets of results, and a comparison is conducted between the following groups:

(a) `Flat field fingerprint reference images` and the `natural image`
(b) `Resized flat field fingerprint reference images` and the `natural image resized`
(c) `Resized flat field fingerprint reference images` and the `natural image Facebook`

(d) `Facebook flat field fingerprint reference images` and the natural image Facebook

And the comparison results are shown in Fig 8 — group (a) in red , group (b) in green, group (c) in purple and group (d) in blue. We note that the high PRNU results of *Cam4* strongly suggest that *Cam4* is the source device.




| | Cam1 | Cam2 | Cam3 | Cam4 |
|---|---|---|---|---|
| (a) | 0.002432776 | 0.0027221R | 0.005318042 | 0.13093069 |
| (b) | 0.008566928 | 0.000114064 | 0.004281503 | 0.03411467 |
| (c) | 0.012559986 | -0.002202909 | 0.002666473 | 0.01968614 |
| (d) | -0.003591058 | 0.006815623 | 0.008155583 | 0.008221319 |

Figure 8: Correlations between the natural image and the fingerprint reference images

In the above four comparisons it is evident that *Cam4* captured the natural image in Fig 7. The best results are obtained in group (a) by comparing `flat field fingerprint reference images` against `natural images`, which fits our expectation; the second best results are obtained in group (c) by comparing `resized flat field fingerprint reference images` against `natural images Facebook`, which is feasible in real computer forensics settings where the law reinforcement personnel can easily compare a Facebook image against a set of resized flat field images obtained from a seized camera. On the other hand, the worse case is group (d) where we compare `Facebook flat field fingerprint reference images` against `natural images Facebook`. Hence, we identify the vital element for correct identification is the flat field images which should be used without being uploaded to Facebook.

## 5. Conclusions and Future Work

This paper investigates the effect of Facebook's image filter regarding image forensics. We show the existence of the undocumented Facebook image filter by examining its changes made to uploaded images. We confirm that a digital image is altered as going through the Facebook filter. We test the effects on the internal aspects of the image and find that Facebook's image filter severely changes the internal image contents, particularly many inconsistencies arise with images with respect to the EXIF header, namely the fact that numerous images downloaded from Facebook do not contain any EXIF header information. Our method of identifying source camera from a sample range of cameras is similar to Filler's approach in [20]. But we test the Facebook image filter which does not appear in any published literature to our best knowledge.

We note the importance of the filter with respect to the PRNU noise. Our experiments identify image sources acquisition devices through the use of PRNU. We determine that PRNU noise patterns are affected in terms of obtaining strong results as an outcome of the Facebook image filter, though substantial results can be still obtained through PRNU noise comparisons. Furthermore, we note a standardized image resolution size $720 \times 540$ that the Facebook image filter accepts; this significantly impacts our tests as we have to resize our flat field fingerprint reference images to the same size, which ultimately affects the PRNU correlation comparison between natural images and reference fingerprint reference images. Thus, we achieve reliable results by comparing the `resized flat field fingerprint reference image` against `Facebook images` both of which are accessible by law enforcement.

Regarding future works, we would like to extend our testing work on newer and more devices as we have used four Apple iPhone 3GS cameras which are relatively homogenous. Finally, we note that iPhone cameras employ the CCD image sensors only, and the CMOS-based cameras are becoming popular in the market. So we would like to further test the CMOS-based cameras, similar to the approach of Lukas, Fridrich and et al. [16].

## References

[1] S. Raik-Allen, "Platforms, not content, are now king," 2011, http://www.abc.net.au/technology/articles/2011/03/10/3160464.htm.

[2] T. Gloe, M. Kirchner, A. Winkler, and R. Bohme, "Can we trust digital image forensics?" in *Proceedings of the 15th international conference on Multimedia*, 2007.

[3] G. Francia and K. Clinton, "Computer forensics laboratory and tools," *Consortium for Computing Sciences in Colleges*, vol. 20, no. 6, p. 143, 2005.

[4] R. Bohme, F. Freiling, T. Gloe, and M. Kirchner, "Multimedia forensics is not computer forensics," in *Proceedings of the 3rd International Workshop on Computational Forensics*, 2009.

[5] H. Sencar and N. Memon, "Overview of state-of-the-art in digital image forensics," *Statistical Science and Interdisciplinary Research*, pp. 1–19, 2008.

[6] Z. Chi and Z. Hongbin, "Digital camera identification based on canonical correlation analysis," in *Proceeding of IEEE 10th Workshop on Multimedia Signal Processing*, 2008, pp. 769–773.

[7] S. Dehnie, T. Sencar, and N. Memon, "Digital image forensics for identifying computer generated and digital camera images," in *Proceeding of IEEE International Conference on Image Processing*, 2006, pp. 2313–2316.

[8] K. Choi, E. Lam, and K. Wong, "Automatic source camera identification using intrinsic lens radial distortion," *Optical Express*, vol. 14, no. 24, pp. 11 551–11 565, 2006.

[9] K. Mehdi, H. Sencar, and N. Memon, "Blind source camera identification," in *Proceeding of International Conference on Image Processing ICIP'04*, vol. 1, 2004, pp. 709–712.

[10] J. Fridrich, "Digital image forensics: Introducing methods that estimate and detect sensor fingerprint," *IEEE Signal Processing Magazine*, vol. 26, no. 2, pp. 26–37, 2009.

[11] Y. Li and C.-T. Li, "Decomposed photo response non-uniformity for digital forensic analysis," *Forensics in Telecommunications, Information and Multimedia*, vol. 8, no. 1, pp. 166–172, 2009.

[12] M. Chen, J. Fridrich, M. Goljan, and J. Lukas, "Determining image origin and integrity using sensor noise," *IEEE Transactions on Information Forensics and Security*, vol. 3, no. 1, pp. 74–90, 2008.

[13] A. Puppe, V. Ding, A. Schutijser, and M. Timmers, "Robustness of prnu fingerprints," *Student Project for Security of Systems and Networks*, pp. 1–25, 2009.

[14] Y. Sutcu, S. Bayram, H. Sencar, and N. Memon, "Improvements on sensor noise based source camera identification," in *Proceedings IEEE International Conference on Multimedia and Expo*, 2007, pp. 24–27.

[15] Y. Hu, C. Jian, and C.-T. Li, "Using improved imaging sensor pattern noise for source camera identification," in *Proceedings of Multimedia and Expo (ICME)*, 2010, pp. 1481–1486.

[16] J. Lukas, J. Fridrich, and M. Goljan, "Digial camera identification from sensor pattern noise," *IEEE Transactions on information forensics and security*, vol. 1, no. 2, pp. 205–214, 2006.

[17] L. Cheng-Liang and C. Yi-Shiang, "The application of intelligent system to digital image forensics," in *Proceedings of International Conference on Machine Learning and Cybernetics*, vol. 5, 2009, pp. 2991–2998.

[18] ImageMagick Studio LLC, "ImageMagick: Convert, Edit, and Compose Images," available at http://www.imagemagick.org.

[19] NFI, "NFI PRNU compare," 2010, available at http://prnucompare.sourceforge.net.

[20] T. Filler, J. Fridrich, and M. Goljan, "Using sensor pattern noise for camera model identification," *Image Processing*, pp. 1296–1299, 2008.

**Invited Paper 1: RFID Security/Privacy in Computer-Integrated Manufacturing and Supply Chains**

Selwyn Piramuthu, Information Systems and Operations Management, University of Florida, & RFID European Lab, Paris, France.

# RFID Security/Privacy in Computer-Integrated Manufacturing and Supply Chains

Selwyn Piramuthu
Information Systems and Operations Management
University of Florida, Gainesville, FL 32611-7169, USA
&
RFID European Lab, Paris, France
Email: selwyn@ufl.edu

*Abstract*—**Security and privacy issues related to RFID implementations in sensitive areas (e.g., health care) are beginning to be addressed by researchers. We consider knowledge based system applications of a few such scenarios in computer-integrated manufacturing and supply chains.**

## I. Introduction

The decrease in unit price of RFID tags with increasing volume and technological advances has the strength to provide the impetus for further growth in RFID applications. The primary thrust toward more RFID (vs. bar code and other auto-identification technologies) applications is due to their capabilities that are a result of their (albeit only minimal) memory and processing power. However, therein lies their vulnerability to security and privacy of the RFID-tagged item and (possibly) the owner (or the bearer) of the item. In fact, security and privacy issues have been cited (along with cost and form factor) as among the primary deterrents to explosive spread of RFID applications.

RFID implementations to-date span a wide spectrum of domains and applications. Among them, supply chain applications dominate while manufacturing applications remain rather rare. The reason for this may just be sheer timing or mandate bias. At least in the U.S., although RFID tags have been used for decades in other applications (e.g., highway toll payment systems), the surge in supply chain applications occurred primarily due to mandates from Wal-Mart and the Department of Defense. These mandates have resulted in spill-over effects in related applications and domains. Manufacturing, on the other hand, has not had the opportunity to experience a similar dynamic. Regardless, prospective manufacturing applications are continually being envisioned by researchers and manufacturing firms (e.g., Zhou 2009, Zhou et al. 2009, Zhou et al. 2010).

While the business case for RFID is rather compelling, its case from a security and privacy perspective is filled with anecdotal evidence of serious (possible or even imagined) violations. We discuss several such scenarios in manufacturing and supply chain applications where RFID tags have been or are seriously being considered for implementation.

## II. RFID in Supply Chains [incl. Manufacturing]

Supply chains generally begin upstream at the manufacturer or supplier of raw materials and end downstream at the end-user or ultimate customer. From a security/privacy perspective, the dynamics are different at different stages in the supply chain. These variations are due to (a) security of the immediate vicinity, (b) extent of mobility of the RFID-tagged item at this location, (c) location-specific constraints on RF-interference, (d) the tolerance limit of parties involved at any given location, and (e) the characteristics/strengths of the communication protocols. We consider these variations for a few stages in a supply chain: up-stream (represented by a computer-integrated manufacturer), mid-stream (represented by a retailer), and down-stream (represented by the ultimate customer).

Privacy and security violations can come from various sources (adversaries) as well as means. Mitrokotsa et al. (2010) provide an overview of several RFID attacks. These attacks include permanently disabling tags (tag removal, tag destruction, KILL command), temporarily disabling tags (active jamming, passive interference), removal or destruction of RFID readers, relay attacks, tag attacks such as cloning and spoofing, reader attacks such as impersonation and eavesdropping, covert channels, denial of service attacks, traffic analysis, crypto. attacks, side channel attacks, replay attacks, and a few others.

### A. *Computer-integrated Manufacturer Perspective*

While RFID tags are used at the pallet-level in manufacturing shop floors, its use at the individual item-level is somewhat uncommon. For example, Volkswagen uses item-level RFID tags in its Emden (Germany) plant on the 20,000 parts supplied by its press shop every day for its Passat saloon and coupe automobiles (automation, 2010). These item-level tags are used to uniquely identify items as they are processed by the manufacturer. Other applications include component parts matching (Zhou, 2009) whereby detailed specifications of each component (as stored in the attached RFID tag) are used to match those that belong together in the final product. In assembling the final product, generally speaking, instances from each of its component part bins are randomly selected. Even though each of these component parts are within necessary tolerance levels, the finished product has better quality when components that best match one another are selected to be used together. This is readily accomplished by the use of RFID-tagged components with detailed component specifications that can be read and utilized for matching appropriate component parts.

In a majority of cases, during computer-integrated manufacturing, the component parts remain in a secure (both physical and RF) environment and do not travel outside this environment. Moreover, there are usually no restrictions on RF strength used in these environments and therefore the authentication protocols used may not have any restrictions on signal-strength. However, restrictions on complexity of these protocols may exist due to the speed at which these component parts move from one physical location to another on the manufacturing shop floor. Faster movements dictate simpler protocols with fewer messages that are passed between prover and verifier since the protocol must generally be completed while the prover and verifier are in close physical proximity to each other. Among the attacks listed in Mitrokotsa et al. (2010), the most relevant in a computer-integrated manufacturing scenario with RFID-tagged component parts may be relay attacks since it is relatively difficult to mount any of the other attacks in such a secure environment that is physically isolated from the surrounding environment. Relay attacks, however, can be fairly easily mounted in this (or, most other environments for that matter) environment. Unfortunately, with existing state-of-the-art crypto. protocols, it is extremely difficult if not impossible to thwart such attacks since the adversary need not necessarily be physically be present in close proximity to the RFID tag that is under attack and the there is no need to modify any message passed between prover and verifier.

### B. *Retailer Perspective*

A majority of RFID tags used in the retail supply chain are at the pallet level. Of course, there are exceptions such as the recent introduction of item-level RFID tags on Wrangler jeans sold at Wal-Mart since August 2010. Security/privacy issues are a bit tricky at the retailer level since the general retailing environment is not as secure as an isolated manufacturing environment where entry by personnel is more or less strictly controlled. In most cases, the security of RFID tags is relatively hard to guarantee in a retailing (vs. manufacturing) environment given the difficulty in RF securing an area that is physically not restricted to entry by outside personnel (e.g., customers). Most retail locations do not have restrictions on RF-interference as well as on characteristics/strengths of communication protocols used. The RFID-tagged items themselves also do not physically change locations frequently within the retail environment. The tolerance limit, in regard to security/privacy issues, of parties at this location is rather low since these limits vary across customers and the retailer also needs to follow related rules and regulations.

Several RFID attacks mentioned in Mitrokotsa et al. (2010) can be readily mounted on RFID-tagged items at a retailer setting. While mechanisms that accomplish permanently disabling of RFID tags (tag removal, tag destruction, KILL command) are meant to be used by the ultimate customer who has legal possession of the RFID-tagged item, the same can be accomplished by an adversary for other purposes. For example, permanent disabling of a tag can facilitate removal of the RFID-tagged item from the retail premises without going through necessary transaction processes (e.g., automated check-out scenario). Similar end-results can be accomplished through temporarily disabling tags (active jamming, passive interference), especially when the adversary is interested in communicating with the RFID tag upon removal of the tagged item from the retail setting. While temporary disabling is relatively easy to accomplish without raising suspicion, permanent disabling of an RFID tag (e.g., tag removal or destruction) may involve effort on the part of the adversary that could trigger suspicion. Moreover, neither of these tag disabling methods may be used with impunity when the tag is in continuous contact with a reader in a retail setting.

Removal or destruction of RFID readers may not be possible in a retail setting since the back-end system may be in constant communication with these readers. The loss of any of these readers would most likely trigger an immediate exception

handling mechanism that would precisely identify the exact reader, its location, and time when it was disconnected from the rest of the system. This may not bode well for the adversary mounting such an attack.

As in the manufacturing setting, relay attacks can be mounted in a retail setting. However, unlike in a manufacturing setting, tag attacks such as cloning and spoofing can be relatively easily accomplished and the extent of damage to the retailer depends on the characteristics of the RFID-tagged item that is cloned or spoofed. The damage from cloning/spoofing could be minimal in terms of monetary value to something that's life-threatening (e.g., involving pharmaceutical items).

Reader attacks such as impersonation and eavesdropping can also be mounted in a retailer setting. Moreover, covert channels, denial of service attacks, traffic analysis, crypto. attacks, side channel attacks, replay attacks, among others can also be accomplished. All these attacks depend on the retail environment and the RFID-tagged items present in this environment and the extent to which an adversary is willing and able to mount any of these attacks. For example, a competitor might be interested in determining the inventory level of a given class of item at a retail environment (e.g., a Wal-Mart store) and this competitor (i.e., adversary) could readily monitor inventory as they are taken from the display shelf as well as when they are moved from the back of the store to the shelf (e.g., Gaukler 2010, Gaukler 2011)

With the increasing use of RFID tags at the pallet-level or case-level, and in rare cases at the item-level, in perishable food chains (e.g., Hsu et al. 2008, Kärkkäinen 2003, kelepouris et al. 2007), securing food supply chains becomes important.

Ownership transfer (e.g., Kapoor et al. 2011) is of paramount importance in supply chains where an RFID-tagged item changes ownership as it passes from upstream to downstream. The criticality of ownership transfer in supply chains is underscored by the fact that physical ownership transfer may not necessarily coincide with its RF counterpart. When these don't go together, a previous owner of an RFID-tagged item may be able to communicate with the tag without explicit permission from or knowledge of the current owner. While this may not lead to deleterious consequences in some applications, the fact is that (physical and RF) ownership transfer was not completely accomplished. A deleterious consequence might occur, for example, when a distributor is able to communicate with items even after they are sold to a retailer. Here, the distributor may be able to determine the retailer's inventory level of this item and can use it to either (1) reduce bullwhip effect or (2) adjust price and/or stock depending on inventory and demand from retailers, or both to its advantage.

### *C. Customer Perspective*

Although currently there are relatively fewer products that are RFID-tagged at the item-level, almost all forecasts show an increasing trend in this regard. The introduction of RFID readers in mobile telephones and the increasing prevalence of such phones would certainly provide the impetus for more applications for item-level RFID tags. Since RFID tags can be read from a distance and as a batch, it facilitates automating (e.g., Baja Beach Club in Barcelona) entry as well as payment systems. However, privacy and security issues at the customer level can lead to rather negative consequences. For example, with RFID-tagged item in possession, the owner of these items can be readily tracked/traced by adversaries. And, being able to be traced/tracked without one's knowledge is rather serious to individual privacy/security. Although incidents involving tracking/tracing of individuals through item-level RFID tags is not known or absent altogether thus far, it is not hard to envision such a scenario as and when item-level RFID tags become ubiquitous.

From the perspective of an individual customer, the immediate vicinity is generally not secure at all times and the individual is most likely mobile. Moreover, individuals are most likely not much tolerant to being tracked/traced (although studies indicate that most individuals are tolerant to giving out private information even when the associated 'reward' is rather minimal). As to one's tolerance to RF-interference, it depends on the individual (e.g., those with artificial pacemaker may be completely intolerant). The strength and complexity of the protocols used to communicate with RFID tags may similarly be affected by individual circumstances.

When the ultimate customer is a health care institution (e.g., hospital), there are clearly strict restrictions and regulations that govern RF-interference since this affects devices that are present in such environments. The strength of RF communication as well as the complexity of communication protocols used in such environments need to be lowered to levels that may not be able to sustain continuous communication nor even the completion of an authentication protocol. Yet another complexity with such scenarios is that as the RFID-tagged (say, health care) item moves from upstream in the supply chain to the health care provider, there may be need for more complex protocols and high-intensity RF communication in mobile environments that are not necessarily secure. Communication and authentication of RFID tags that pass through

such (sometimes highly secure and low security at other times) environments need to be adaptive and the RFID tag and related systems need to be able to seamlessly switch among such environments.

Individual customers who won't need to communicate with the RFID-tagged item through RF means can easily remove or permanently destroy the tag as well as accomplish the same using the KILL command. The customer can also temporarily disable the tag through active jamming or passive interference while in transit in an environment that is not deemed to be secure. The customer can then enable the tag at a later time while at a secure environment (e.g., at home when an RFID-tagged perishable item is placed in an RFID-reader-enabled refrigerator). An issue with the use of cell phones or PDAs as readers is that when such a reader is stolen, it has the keys to all related RFID tags. With access to those keys, an adversary can impersonate the reader to the tags. Attacks such as cloning or spoofing may not be of much interest or use to an adversary when individual customers are involved. However, in other cases (e.g., health care environments), cloning/spoofing (of life-saving pharmaceutical items) has the potential to compromise someone's health.

An adversary can mount relay attacks on such RFID-tagged items. For example, relay attacks can be used to remotely and from a distance open doors to buildings or start an automobile. Covert channels may be utilized by adversaries to retrieve information from the tags, and therefore about the owner of the tags, unbeknownst to the owner. The adversary can also conduct traffic analysis to gain information on the communicating parties. When appropriate, the adversary can overload the communication channels and prevent the tag from communicating with a legitimate reader. The adversary can also mount replay attacks, when the protocols are relatively easily broken, to impersonate the tag to the reader and vice versa.

## III. Discussion

It is difficult enough to ensure security and privacy of individuals and organizations when dealing with tangible entities. It is even worse in terms of security and privacy when dealing with RF communications, especially when such communications (1) could occur without implicit permission from the owner of the RFID-tagged entity, and (2) have the potential to cause irrevocable damage. A primary means to alleviate or reduce such violations related to security and privacy of RFID-tagged entities is through the use of strong authentication/communication protocols and by ensuring that RFID tags do not communicate with unauthenticated entities. It is difficult to develop such protocols, especially under extreme memory and processing power constraints such as those in low-cost passive RFID tags. Yet, it is a challenge that has to be addressed before the worst case scenarios become reality leaving everything in chaos.

Moreover, RFID-tagged entities are not static during their lifetime. It is highly likely that these entities change locations (either through a supply chain or when the tagged entity is moved when its owner moves) and when that happens, the security/privacy levels should not deteriorate. However, ambient conditions may dictate lower levels of RF communication power (e.g., in health care environments) or simpler (i.e., less complex) protocols when the RFID-tagged objects move quickly in and out of the field of the reader. These environments expose the RFID-tagged items to vulnerabilities from resourceful adversaries. For RFID tags to be widely accepted across various applications, security/privacy issues need to be addressed regardless of the resource constraints in these tags as well as the ever-changing ambient conditions under which they operate. We discuss some knowledge-based approaches to addressing some of these issues.

While researchers have made great strides in this area during the past decade, there still remain several unaddressed issues. Some of these issues include:

- Interference from RF signals: This is not really an issue in a majority of applications. However, where this is an issue (e.g., health care environments where RF signal has been shown to interfere with devices that are present in the immediate vicinity), existing literature does not present an acceptable solution.
- Relay attacks: None of the existing solutions completely address these attacks. While minimal progress has been made in (a) identifying the possibility of these attacks and (b) proposing protocols that measure the round-trip times of messages, given the relatively short distances that are covered by these messages and the accuracy of measuring devices, this still remains an unsolved vulnerability.
- Ultra-lightweight implementations (e.g., without the use of hash functions): All existing ultra-lightweight protocols are vulnerable to attacks from (active/passive) adversaries. Although developments in RFID technology and the sheer increase in volume would eventually bring down the unit cost of RFID tags with sufficient capabilities, there likely would always be a need for tags with the most basic capabilities.

- Protocols for other scenarios: A majority of authentication protocols address (either one-way or mutual authentication of) the single-tag/single-owner scenario. While this is the most prevalent case in reality, there is a need to develop protocols for other scenarios that are not so uncommon (e.g., ownership transfer, ownership sharing, multiple tags, multiple readers).
- Protocols and their security characteristics: There is also a claim that protocols need not be completely secure since multiple-authentications are performed in critical cases (e.g., passports, where crypto. authentication is done in addition to magnetic scan and physical matching of photograph and person). Can we derive security bounds based on the necessity/sufficiency as it relates to the application domain/person/institution?
- RFID tag design: While RFID tags can be embedded in objects (unlike bar codes, which need to be on the outside to be scanned), harsh environments (e.g., surgical instruments that are sterilized after each use) dictate appropriate design and placement of these tags. Some application areas that involve liquid or metal also necessitate the use of either buffers between the tag and the substrate item or some means to improve read rate accuracy. There is an urgent need to address related issues.
- Improving read-rate accuracy: A related issue is the need to improve the read-rate accuracy, which is not necessarily anywhere near 100% in all RFID applications. With the incorporation of RFID tags in supply chains, there is a strong push toward automating various processes. Automation of processes will result in poor quality output when read-rate accuracy is low since, unlike in the bar code case, manual checks are most likely not that common in automated systems.
- RFID in component manufacturing: RFID tagging items that are finished products are fairly straight-forward. However, how does one tag items as they are being processed (e.g., in a manufacturing shop floor)? Are there means to temporarily remove them while they're in-process?

The above list is by no means meant to be exhaustive. Rather, these are only a subset of some of the obvious issues that arise in RFID implementations. Given the reaction among the general public toward anything that could possibly violate security/privacy, it is critical to address some of these issues before RFID tags become even more prevalent in our everyday lives.

## REFERENCES


[1] Automation. 2010. Light Barriers Keep VW Safe. 30 April. (http://www.connectingindustry.com/story.asp?sectioncode=663&storycode=194499)

[2] Gaukler, G.M. 2010. Preventing avoidable stockouts: the impact of item-level RFID in retail. *Journal of Business & Industrial Marketing*, 25(8), 572-581

[3] Gaukler, G.M. 2011. Item-level RFID in a Retail Supply Chain with Stock-out-based Substitution. *IEEE Transactions on Industrial Informatics*, 7(2), 362-370.

[4] Hsu, Y.-C., A.-P. Chen, C.-H. Wang. 2008. A RFID-Enabled Traceability System for the Supply Chain of Live Fish. *Proceedings of the IEEE International Conference on Automation and Logistics*, 81-86.

[5] Kapoor, G., W. Zhou, S. Piramuthu. 2011. Multi-tag and Multi-owner RFID Ownership Transfer in Supply Chains. *Decision Support Systems*, 2011.

[6] Kärkkäinen, M. 2003. Increasing Efficiency in the Supply Chain for Short Shelf Life Goods using RFID Tagging. *International Journal of Retail & Distribution Management*, 31(10), 529-536.

[7] Kelepouris, T., K. Pramatari, G. Doukidis. 2007. RFID-enabled Traceability in the Food Supply Chain. *Industrial Management & Data Systems*, 107(2), 183-200.

[8] Mitrokotsa, A., M.R. Rieback, A.S. Tanenbaum. 2010. Classifying RFID Attacks and Defenses. *Information Systems Frontiers*, 12, 491-505.

[9] Zhou, W., G. Kapoor, S. Piramuthu. 2009. RFID Enabled Item-level Product Information Revelation. *European Journal of Information Systems*, 18, 570-577.

[10] Zhou, W., S. Piramuthu. 2010. Framework, strategy and evaluation of health care processes with RFID. *Decision Support Systems* 50(1), 222-233.

[11] Zhou, W. 2009. RFID and Item-level Information Visibility. *European Journal of Operational Research*, 198(1), 252-258.

## Invited Paper 2: Smart Phone Security

Bernard Colbert, Resolve Partners.

# Smart Phone Security

Bernard Colbert

## I. Introduction

This paper provides a brief overview of the security of mobile devices. It focuses on capable platforms such as smart phones and tablets.

## II. Change in Landscape

Mobile devices have evolved rapidly, mainly in response to consumer demand. Devices compete on applications, style and integration with other devices, networks and systems.

Mobile devices, also known as *Smart Phones* have become very popular, and the use of these devices have also diversified. They are now used to:

- read mail,
- read documents — including commercial documents such as contracts,
- consuming other media — music, movies and magazines, and
- playing games.

Beyond these applications they are also being used in other ways:

- banking and trading, and
- being used as a means for authentication.

These have implicit security functions.

## III. Operating Systems

There are two paradigms of application environments:

- using the device's native operating system;
- running the applications on a virtualised platform.

The second approach is used in the Android Platform, which runs applications within a Java environment, and by the Microsoft X360 platforms which runs each application on it own virtual machine.

Due to the constrained nature of mobile devices, they have specialised operating systems, although many are based on existing operating systems. The operating systems used include:

- Linux and Unix variants:
  - iPhone,
  - Mobile Ubuntu;
- Windows Mobile —- the latest being Windows Mobile 7;
- Symbian 9; and
- Blackberry Operating System.

The Windows Mobile, Symbian and Blackberry operating systems are proprietary systems. Interfaces to Windows Mobile and Symbian are available to developers.

Note, however, that Symbian is no longer supported.

## IV. Change in Security Posture

There are many consequences of the changes in mobile platforms that significantly change their security profile. Some of these consequences are detailed below.

### A. Standarisation of Application Environments

In the past there were many mobile device operating systems. Often they were proprietary, and none gained market dominance.

This had two consequences.

1) There was a small number of applications for any particular device, and these often needed proprietary interfaces to be developed. This limited the opportunity for malicious applications to be developed. The small number also provided limited opportunity to exploit bugs in the code.
2) If an application was exploited, then it would affect only a small proportion of mobile devices, since no environment had dominant position. Thus, there was little to gain by exploiting any weakness.

As mobile operating systems have become more standardised, it has allowed the development of many more, and useful, applications. It has also meant that these operating environments have a greater footprint. Thus, it becomes attractive to exploit these environments.

### B. Data

Due to their utility, mobile devices have significantly more personal data. This includes:

- mail messages;
- personal documents;
- address books and contacts, including medical and other contacts;
- personal details, such as drivers licence numbers, banking details, *et cetera*.

In a corporate environment, these device also hold this type of information and corporate documents.

### C. Use

Mobile devices are also being used for security functions — for example, sending an SMS with confirmation codes, or using security functions on the SIM. This is a continuation of use of mobile devices.

They are also able to use other technology, such as Near Field Communications, which can be used as access tokens.

Consequently, the utility of mobile devices has increased significantly.

### D. Target Profile

The two aspects — increased data and utility — makes these devices significantly more attractive to malicious users.

This has been seen in a corresponding targeting of mobile devies to spam, viruses, Trojans and other malicious software.

The devices are also attractive targets for theft and physical attack. The value of the information on the device often exceeds the value of the device.

## V. Security Models

Security models have been developed for mobile devices.

### A. iPhone

The iPhone and iPad provide a centralised security model. The operating system restricts the method of loading applications onto the device. Only applications from the *App Store* can be installed onto the device.

Developers are required to have a relationship with Apple, and to submit the applications for review before they are made available on the App Store.

One the device an application loader exists to install all applications. This application loader can also remove applications: and the removal of applications may be initiated by either the user or from an update from the App Store.

Consequently, malicious software is controlled by:

- developers having a relationship with Apple,
- application being reviewed before being put into App Store, and
- applications being removed.

The first two mechanisms limit the opportunity for malicious software to be loaded onto the platform. The third allows remediation of malicious software.

Note that this will not remove the usual threats of allowing bugs in the code allowing a third party access to the platform. The most common example of this is seen in the method for users to gain control and root access of the device — which is known as *jail breaking* the phone.

In this, the user loads a particular page into the browser on the phone, which causes a buffer overflow, and provides root access. The user then follows the instructions to gain access to the root user account on the phone.

When a device is *jail broken* arbitrary software can be installed and run. It also removes the capability for the centralised controls to remove software. In this case, the mobile device is as vulnerable to being exploited as other Linux/Unix based platforms.

### B. Windows Mobile

Windows Mobile 7 provides a sophisticated and comprehensive security model:, based on a policy enforcement approach.

All applications are put into one of four security containers. These are called:

- Trusted Computing Base
- Elevated Rights Chamber
- Standard Rights Chamber
- Least Privileged Chamber

The container will determine what privileges will be granted to the application.

The functions controlled include:

- access to HTTP/S;
- access to other networking;
- location information;
- access to phone functions — calling;
- access to phone functions — messaging;
- access to user data; and
- recording and managing media.

The most privileged chamber is the Trusted Computing Base — which is based on the work of the Trusted Computing Group. This has the controls to the platform.

At the other end, applications downloaded from the *Marketplace Hub*, the application store for Windows Mobile, will be placed by default into the Least Privileged Chamber.

Along with these chambers are the capabilities and utilities to define and enforce policies. This includes policy and account databases, policy loader and policy engines. Authentication and authorisation framework for the device.

Processes have also been defined for developing applications and software which require greater access to the device.

Users also have the option of synchronising the device to a Microsoft Exchange server: the server used by the device for mail. The primary function of this server is for mail messages, but it also provides two security functions: *Remote Lock* and *Remote Wipe*. These functions allow a lost device to lock, so that data on the device can be access. The remote wipe function will delete the data and disable the device.

In a corporate environment, the device would be synchronised to a corporate Exchange Server, and the *remote lock* and *remote wipe* functions would be managed by the administrators of the server.

### C. Android

Android is a Java based application framework, running on top of a Linux 2.6 kernel. APIs have been published to allow applications to be developed to run within Android, and will be run on a Java Virtual Machine (JVM). Each application will be placed into its own JVM.

The security model for Android is based on each application having a set of permissions that the user grants to the application. These permissions are then enforced by the underlying platform.

The set of permissions is expansive. When an application is installed, the user is queried what permissions are to be granted to the application.The permissions cover what functions will be granted to the application, what other applications are required to be used, and what other applications can use the application being installed.

The approach is flawed for a number of reasons.

- The user is required to have a comprehensive understanding of security and permissions.
- Users are expected to go through a large number of permissions when installing an application. On many devices this requires the user reading through a number of screens, and being able to allocate individual permissions.

In most cases, users will not read the permissions and give whatever permissions are being asked.
- It assumes that the permission model will not lead to any conflicts.
- Developers will asks for maximum access to the device — since they may want to extend their application and not ask for revalidation of permissions.
- The only way to change the permission for an application is to remove and reinstall it.

Without a facility for abstracting and managing permissions, the cluster of permissions becomes very complex. This will allow malicious applications to effectively obfuscate any malicious behaviour. For example, a logger may ensure that it will only collect information while other applications are also running, making it difficult to identify which application was actually collecting information maliciously.

Even if the permissions were effective, the risk of poorly written applications providing access to the underlying platform and other applications.

## VI. Other Considerations and Security Measures

In analysing the security of mobile devices, external aspects will influence security.

- Networking provides access to the devices. While this is fundamental to the device, it also provides an opening for malicious parties to exploit.
  For example, when the iPhone was first released, it was configured to roam automatically onto any open wireless LAN hot spot. Thus, a malicious person could set up a hot spot and wait for devices to roam onto it. It would then redirect all web traffic to a page that would jail break the device and take control of the device.
- Other protocols may have intrinsic weakness. Security researchers are regularly finding vulnerabilities in protocols. These, however, affect all platforms, not just mobile devices.
- More than any other platform, mobile devices are prone to physical exploitation. Thus, attacks which require physical access present a significant problem to mobile devices, since they are more vulnerable to physical access and tampering.

### A. Open Mobile Alliance

The Open Mobile Alliance has published standards, OMA 2, that allows carriers to be able to manage mobile devices over the air — that is, carriers are able to update and reconfigure mobile devices while are connected to the mobile network.

Essentially, the mobile device has an OMA agent which communicates with a management server in the network. The agent is able to install and configure updates passed over the data link.

### B. DRM Solutions

Digital Rights Management has been developed to control the copying and consuming of digital content, such as movies and music. However, it was also extended to be applied to documents, in particular corporate documents.

These solutions can be implemented on mobile devices (and other platforms) to protect any documents on the device.

### C. Platform Security

Security measures can be applied to mobile devices as well as other platforms. These include:

- antivirus scanners,
- platform layer data encryption,
- virtual private networking and other communications security, and
- document management for access control to documents and information.

These are not specific to mobile devices.

## VII. Conclusion

While the security of mobile device is important, it is not one of the drivers in the development of mobile devices. Other features, such as applications and style are considered more commercially important.

The increased capability and utility of mobile devices has meant that they are more widely used and have much more data. Consequently, they are being targeted more by malicious parties, and attacks against mobile devices are becoming more common.

Currently, several security models exist for mobile devices: the Apple model is very centralised, the Windows Mobile model is commercially focused, and the Android model is limited.

However, these model will only form part of security mobile devices. Other considerations, such as communications and physical security need to be considered.