Extensible Regular Humanistic Observable Impression Scooping Multi-various Physical Facts

Sneha D Patil[1] Dinesh D Patil[2]
[1]ME Student [2]Professor
[1,2]Department of Computer Science
[1,2]SSGB College of Engineering and Technology, Bhusawal India

Abstract— *Despite the advent of wearable devices and the proliferation of smart phones, there still is no ideal platform that can continuously sense and precisely collect all available contextual information. Mobile sensing data collection approaches should deal with uncertainty and data loss originating from software and hardware restrictions. We have conducted life logging data collection experiments from many users and created a rich dataset (7.5 million records) to represent the real world deployment issues of mobile sensing systems. We create a novel approach to identify human behavioral motifs while considering the uncertainty of collect data objects. Our work benefits from combinations of sensors available on a device and identifies behavioral patterns with a temporal granularity similar to human time perception. Employing a combination of sensors rather than focusing on only one sensor can handle uncertainty by neglecting sensor data that is not available and focusing instead on available data. Moreover, we demonstrate that using a sliding window significantly improves the scalability of our analysis, which can be used by applications for small devices such as smart phones and wearable.*

Keywords— Frequent Pattern Mining, Temporal Granularity, Multivariate Temporal Data, Human-Centric Data, Human Pattern

I. Introduction

The proliferation of smart phones and, more recently, wearable devices such as fitness trackers and smart watches equipped with sensors, has led to a significant expansion of possibilities to study human behavior. Computing and networking capabilities of these devices within their multiple sensors makes them capable enough so we can easily observe and collect useful contextual information (mobile sensing). For instance, mobile health, which benefits from mobile sensing, offers the possibility of a shift from treatment to prevention in medical care systems. Researchers show that 69% of U.S. adults monitor and track their health status and 21% of them use technology for this purpose [8]. Unlike wearable devices, which are still quite new in the market, the smart phone platform has benefited from a significant amount of scientific work ranging from personal air pollution footprint trackers applications [15] to wellbeing [13]. Both wearable devices and smart phones are very capable of sensing and collecting basic patterns of human behavior and collecting contextual information.

While human behaviors are predictable, at least in aggregate [1], traditional approaches for detecting human behavioral patterns (which are not digital) are often difficult. However, the advent of these ubiquitous devices enables researchers to identify human behavior to an extent that was not previously possible. On one hand, this information collection paradigm should be moved from simple data collection tools to intelligent systems with cognition capabilities [4]. On the other hand, there is still a lack of wide acceptance of mobile sensing applications in real world settings.

There are several reasons for this mismatch of capability and acceptance. First is the resource limitation and lack of accuracy in the collected contextual data, especially with regard to the battery life [24]. The size of sensors that are dealing with radio frequency, i.e., Bluetooth, Wi-Fi and GPS, affects the quality of their data [22] (smaller devices have less accurate data). The next reason, which has been noted but has not been widely explored, is the proximity of the smart phone to users [5]. Smart watches and wearable are body mounted and thus the proximity problem has been resolved in those devices, but they still suffer from a lack of accuracy [12]. The third reason for this problem is operating system restrictions of mobile devices, which removes background services when the CPU is under a heavy load in order to preserve the battery life. As a result, there is no ideal data collection approach that can sense and record individuals information 24/7 with no data loss. The uncertainty of these data objects is a major challenge that limits the applications that can benefit from them.

This Thesis deals with the problem of analyzing smart malware for smart devices, providing specific methods for improving their identification. The Thesis is strongly biased towards smartphones, since they currently are the most extended class of smart devices and the platform of choice for malware developers and security researchers. However, our discussion and conclusions apply to other devices as well, and can help to better understand the problem and to improve upon current defense techniques.

We next describe the main motivation and objectives of this work. Firstly, we state that current methods aiming at analyzing smart malware are ineffective and we question the role that security analysts play during the study of large amounts of complex software. Secondly, we establish the need of systematic approaches and automated tools for analyzing smart malware.

II. Motivation

This Thesis identifies two fundamental open issues where research is needed: There is more malware than ever before, and it is increasingly sophisticated. P1: Sustained growth in the number of malicious apps targeting smart devices. As discussed before, malware has become a rather profitable business due to the existence of a large number of potential targets and the availability of reuse-oriented malware development methodologies that make exceedingly easy to produce new samples. The impressive growth both in malware and benign apps is making increasingly unaffordable any human-driven analysis of potentially dangerous apps. This is especially critical as current trends in malware engineering suggest that malicious software will continue to grow both in number and sophistication. As a result, market operators and malware analysts are overwhelmed by the amount of newly discovered samples that must be analyzed. This is further complicated by the fact that determine Increase in the sophistication of malicious apps and the rise of a new generation of smart malware.

Malware for current smartphone platforms is becoming increasingly sophisticated and developers are progressively using advanced techniques to defeat malware detection tools. On one hand, smartphone malware is becoming more and more stealthy and recent specimens are relying on advanced code obfuscation techniques to evade detection. These techniques create an additional obstacle to malware analysts, who see their task further complicated and have to ultimately rely on carefully controlled dynamic analysis techniques to detect the presence of potentially dangerous pieces of code. On the other hand, the presence of advanced networking and sensing functions in the device is giving rise to a new generation of smarter malware. These malware instances are characterized by a more complex situational awareness, in which decisions are made on the basis of factors such as the location, the user profile, or the presence of other apps.

This state of affairs has consolidated the need for smart analysis techniques to aid malware analysts in their daily functions. This challenge has to be tackled by novel methods to efficiently support market operators and security analysts. In some cases, this problem cannot be solved by market operators alone or by enhanced security models, as they really depend on each user's privacy preferences. For example, a leakage of data such as one's location or the list of contacts might well constitute a serious privacy issue for many users, but others will simply not care about it. The situation described above inevitably leads to the need for more sophisticated analysis techniques. This, however, poses an important challenge: many devices suffer from strong limitations in terms of power consumption, so certain security tasks executed on the platform may be simply unaffordable. External analysis performed on the cloud in near real time can constitute an alternative. Such a strategy seeks to save battery life by exchanging computation and communication costs, but it still remains unclear whether this is optimal or not in all circumstances. Furthermore, the rise of targeted—user-specific—malware poses one additional challenge: conducting particularized analysis for specific user and execution context.

III. Objectives

The main goal of this Thesis is to study methods, tools and techniques to assist security analysts and end users in the analysis of untrusted apps for smart devices and automate the identification of smart malware.

To achieve this goal, we will focus in the following three general objectives:

- Study the evolution and current state of malware for smart devices, as well as recent progress made to analyze and detect it.
- Develop techniques aiming at better analyzing malware in large scale software markets, with particular emphasis on intelligent instruments to automate parts of the analysis process.
- Facilitate the analysis of complex smart malware in scenarios with a constant and large stream of apps on target. Examples of such sophistication include malware targeting user-specific actions, malware hindering detection with advance obfuscation techniques, or malware exploiting the battery limitations of current devices, to name a few.

A. Naïve Bayes

D: Set of tuples

- Each Tuple is an 'n' dimensional attribute vector
- X : (x1,x2,x3,…. xn)

Let there be 'm' Classes: C1,C2,C3…Cm

Naïve Bayes classifier predicts X belongs to Class Ci iff

- P (Ci/X) > P(Cj/X) for 1<= j <= m , j <> i

Maximum Posteriori Hypothesis

- P(Ci/X) = P(X/Ci) P(Ci) / P(X)
- Maximize P(X/Ci) P(Ci) as P(X) is constant

With many attributes, it is computationally expensive to evaluate P(X/Ci).

Naïve Assumption of "class conditional independence"

$$p(X/Ci) = \prod_{k=1}^{n} p(x_k/Ci)$$

$$P(X/Ci) = P(x1/Ci) * P(x2/Ci) *\ldots* P(xn/ Ci)$$

B. Apriori Algorithm for mining frequent Itemset

Association rule generation is usually split up into two separate steps:

1) First, minimum support is applied to find all frequent itemsets in a database.
2) Second, these frequent itemsets and the minimum confidence constraint are used to form rules.

While the second step is straight forward, the first step needs more attention. Finding all frequent itemsets in a database is difficult since it involves searching all possible item sets (item combinations). The set of possible itemsets is the power set over I and has size 2n− 1 (excluding the empty set which is not a valid itemset). Although the size of the powerset grows exponentially in the number of items n in I, efficient search is possible using the downward-closure property of support (also called anti-monotonicity) which guarantees that for a frequent itemset, all its subsets are also frequent and thus for an infrequent itemset, all its supersets must also be infrequent. Exploiting this property, efficient algorithms (e.g., Apriori and Eclat) can find all frequent itemsets.

C. Apriori Algorithm Pseudocode

```
Procedure Apriori (T, minSupport) {//T is the database and minSupport is the minimum support
L 1 = {frequent items};
for ( k=2; $L_{k-1}$ ! = Ø;k++) {
$C_k$ = Candidates generated from $L_{k-1}$
// that is Cartesian product $L_{k-1}$ x $L_{k-1}$ and eliminating any k-1 size itemset that is // not frequent
```

```
        for each transaction t in database do {
        #increment the count of all candidates in $C_k$ that are
contained in t
        $L_k$ = candidates in $C_k$ with minSupport
}// end of each
        return $U_k L_k$ ;
}
```

As is common in association rule mining, given a set of itemsets (for instance, sets of retail transactions, each listing individual items purchased), the algorithm attempts to find subsets which are common to at least a minimum number C of the itemsets. Apriori uses a "bottom up" approach, where frequent subsets are extended one item at a time (a step known as candidate generation), and groups of candidates are tested against the data. The algorithm terminates when no further successful extensions are found.

Apriori uses breadth-first search and a tree structure to count candidate item sets efficiently. It generates candidate item sets of length k from item sets of length k−1. Then it prunes the candidates which have an infrequent sub pattern. According to the do wnward closure lemma, the candidate set contains all frequent k-length item sets. After that, it scans the transaction database to determine frequent item sets among the candidates.

Apriori, while historically significant, suffers from a number of inefficiencies or trade-offs, which have spawned other algorithms. Candidate generation generates large numbers of subsets (the algorithm attempts to load up the candidate set with as many as possible before each scan). Bottom-up subset exploration (essentially a breadth-first traversal of the subset lattice) finds any maximal subset S only after all.

D. Sybil attack Detection in Social networks

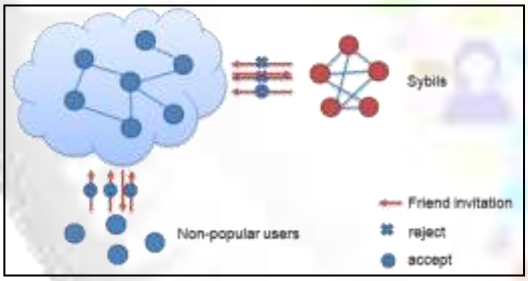


Fig. 1: Sybil attack Detection in Social networks

E. VoteTrust: An Overview

1) *Basic idea*

- Considering invitation feedback as *voting*

2) *Key techniques*

- Trust-based votes assignment
- Global vote aggregation

3) *Properties*

- High precision in Sybil detection
- Efficient in limiting Sybil’s attack ability

F. Graph Model

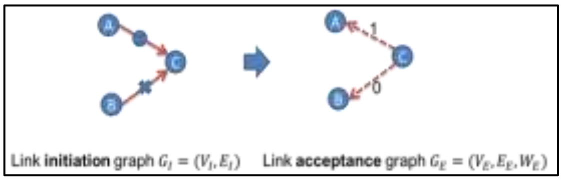

A
B
C
A
B
C
1
0
Link initiation graph $G_I = (V_I, E_I)$
Link acceptance graph $G_E = (V_E, E_E, W_E)$

Fig. 2: Graph Model

1) *Proposed Methodology*

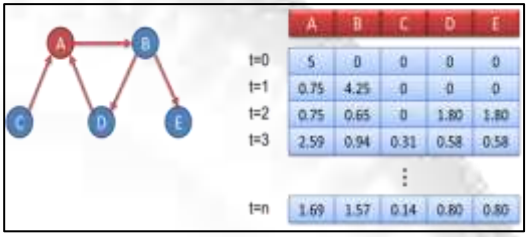
- Select trust seed – high reliable users
- Distribute votes
- Collect votes and computing score

2) *Trust-based Votes Assignment*

- Step1: Assigning votes to little human-selected reliable seeds
- Step2: Propagating to whole users across the Link initiation graph

$$vote(u) = d \cdot \sum_{v:(v,u)\epsilon E1} \frac{vote(v)}{out\_degree(v)} + (1-d) \cdot init(u)$$

3) *Example*



	A	B	C	D	E
t=0	5	0	0	0	0
t=1	0.75	4.25	0	0	0
t=2	0.75	0.65	0	1.80	1.80
t=3	2.59	0.94	0.31	0.58	0.58
			⋮		
t=n	1.69	1.57	0.14	0.80	0.80

Fig. 3: Example

- Node A is reliable seed
- Total votes =5

G. Vote Aggregating

1) *Trust-based votes assignment*

The goal of trust-based votes assignment is to assign low vote capacity to Sybils, so that we can limit the number of votes that Sybils could cast for each other. To achieve this goal, we first select some trusted users as seeds, and then propagate the vote capacity from the seeds to others along the links of friend invitation graph G(V;E). As Sybil region has a limited number of in-links, the total vote capacity entering the Sybil region is constrained.

Selecting Trusted Seeds. The goal of seed selection is to find real users that will be the most useful in identifying other real users. A heuristic for selecting seeds is to give preference to those from which trust can be propagated to many other real users. Note that real users prefer to send requests to their real-life acquaintances, so we use the inverse PageRank method like TrustRank [15]. The basic idea is to build the seed set from real users that point to many real users that in turn point to many others and so on. In particular, we can reverse the links in the friend invitation graph, and compute the PageRank. Through manually inspecting a few users of high inverse PageRank scores, OSN providers can easily identify those real users to seed trust.

2) *Global Vote Aggregation*

Vote assignment gives low vote capacity to not only Sybils but also non-popular real users with few incoming links. We thus introduce the global vote aggregating phase to get the global acceptance rate $p(u)$ of a node u. This phase further leverages the sign of outgoing links (i.e., the user feedback) for higher accuracy, as Sybils have a higher percentage of negative links to real region.

3) *Global rating computation*

For a node u, VoteTrust computes the $p(u)$ by combining all the votes from its outgoing neighbors. As neighbors of high global acceptance rates are more likely to be real users, we should bias towards their votes. Based on the above intuition

4) Limiting the collusion votes

When aggregating votes of outgoing neighbors, an important problem we should address is how to prevent the attacker from increasing the total number of collusion votes by enlarging the Sybil set? Considering the case illustrated in Fig. 6. Initially, the Sybil region has 3 Sybils that receive a total of 1 vote capacity from the real region. The vote capacity of each Sybil is 1=3, and each Sybil can collect at most 1=3 collusion votes. However, if the attacker adds another two Sybils, the vote capacity of individuals drops to 1=5 as the total vote capacity is constant. But each Sybil can collect at most 2=5 collusion votes. This means that the attacker can increase collusion votes for Sybils by enlarging the Sybil region. In fact, a complete-connected subgraph with N Sybils and c total capacity could create c(N¡1) 2 collusion votes, which increases as N grows.

- Step1: Set all users' initial score as 0.5;
- Step2: Iteratively computing each user's trust score according to aggregated votes.

$$score(u) = \frac{\sum vote(v).score(v).X_{v,u}}{\sum vote(v).score(v)}\ ,(v,u) \in E_E$$

H. Algorithm for Sybil Detection

```
        Procedure VOTETRUST-D(G,$V_\delta$)
        if u ∈ $V_\delta$ then            vote assignment
        I(u) ← N / | $V_\delta$ | ;
else
        I(u) ←0 ;
end if
while  $\Delta > \varepsilon_1$  do
        for u  ∈ V  do
       $\vartheta(u) = d \cdot \sum_{v:(v,u)\epsilon E} \frac{\vartheta(v)}{\omega(v)} + (1-d) \cdot I(u)$
        end for
end while
$p^{(0)}$  ← 0.5 ;              vote aggregation
  while  $\Delta > \varepsilon_2$  do
        for u ∈ V do

$\hat{p}(u) = \frac{\sum_{v:(v,u)\in E} + v(v) \cdot p(v)}{\sum_{v:(v,u)\in E} v(v) \cdot p(v)}$
p ← WilsonScore ($\hat{p}$);
        end for
   end while
end procedure
```

1) Small-sample Problem

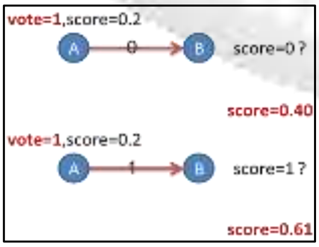


Fig. 4: Number of votes is too small

2) Wilson Score

$$p = \frac{\hat{p} + \frac{1}{2N} z_{1-\alpha/2}}{1 + \frac{1}{N} z_{1-\alpha/2}}$$

Weighted average of $\hat{p}$ and ½.

a) Security Properties (I)

- Theorem 1: The Number of Sybil's attack-link needs to satisfy the following upper bound

$$N_{out} \leq \rho N_{in} \cdot \frac{\delta_f - \delta_f^2}{\delta_f - r}$$

Where $\boldsymbol{\delta_f}$ is Detection Threshold

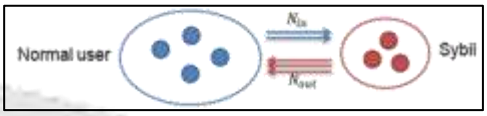


Fig. 5: Security Properties

b) Security Properties 2

- Theorem 2: Sybil community size need to satisfy the upper bound

$$N_s \leq \sigma \cdot \frac{N_{in}}{\delta_v}$$

Where δ_v is Vote Collection Threshold

IV. Conclusion

In this paper, we have proposed a scalable approach for daily behavioral pattern mining from multiple information sources. This work benefits from a realistic dataset and users who use different smart phone brands. We use a novel temporal granularity transformation algorithm that makes changes on timestamps to mirror the human perception of time. Our behavioral motif detection approach is generic and not dependent on a single source of information; therefore, we reduce the risk of uncertainty by relying on a combination of sensors to identify behavioral motifs and patterns. Our app also identifies health deficiencies in user according to the behavior user is opting or recording in our app. We also generate a probabilistic results from the data generated by user. We investigate the efficiency of our work by evaluating it from three different perspectives: the execution time performance, the effect of threshold changes on motif detection, and the validity of the identified behavior from a temporal perspective. This approach is scalable enough to be used in several types of applications such as mobile health, context-aware recommendations and other quantified-self applications

References

[1] S. Foell et al., "Micro-navigation for Urban Bus Passengers: Using the Internet of Things to Improve the Public Transport Experience," Urb-Iot '14.

[2] R. Dobbins and R. Rawassizadeh, "Clustering of Physical Activities for Quantified Self and mHealth Applications," in IUCC 15.

[3] A. Campbell and T. Choudhury, "From Smart to Cognitive Phones," IEEE Pervasive Computing '12, vol. 11, no. 3, pp. 7–11.

[4] D. Ferreira et al., "Understanding Human-Smartphone Concerns: a Study of Battery Life," Pervasive Computing '11.

[5] R. Rawassizadeh et al., "Wearables: Has the Age of Smartwatches Finally Arrived ?" Communications of the ACM '15, vol. 58, no. 1, pp. 45–47.

[6] A. Dey et al., "Getting Closer: An Empirical Investigation of the Proximity of User to Their Smart Phones," UbiComp '11.

[7] N. Eagle and A. Pentland, "Reality Mining: Sensing Complex Social Systems," Personal and Ubiquitous Computing '06, vol. 10, no. 4, pp. 255–268.

[8] N. Kiukkonen et al., "Towards Rich Mobile Phone Datasets: Lausanne Data Collection Campaign," ICPS' 10.

[9] S. Nath, "ACE: Exploiting Correlation for Energy-Efficient and Continuous Context Sensing," MobiSys '12.

[10] H. Ma et al., "A Habit Mining Approach for Discovering Similar Mobile Users," WWW '12.

[11] R. Rawassizadeh et al., "Ubiqlog: a generic mobile phone-based life-log framework," Personal and Ubiquitous Computing '13, vol. 17, no. 4, pp. 621–637.

[12] D. Wagner et al., "Device Analyzer: Large-scale Mobile Data Collection," SIGMETRICS Perform. Eval. Rev. '14, vol. 41, no. 4, pp. 53–56.

[13] R. Rawassizadeh and A. Tjoa, "Securing Shareable Life-Logs," SocialCom '10.

[14] R. Rawassizadeh et al., "Lesson Learned from Collecting Quantified Self Information via Mobile and Wearable Devices," Journal of Sensor and Actuator Networks, vol. 4, no. 4, p. 315, 2015.

[15] J. Paek et al., "Energy-efficient Positioning for Smartphones Using Cell-ID Sequence Matching," in MobiSys '11.

[16] K. Farrahi and D. Gatica-Perez, "A Probabilistic Approach to Mining Mobile Phone Data Sequences," Personal and Ubiquitous Computing, vol. 18, no. 1, pp. 223–238, 2014.

[17] J. Zheng and L. Ni, "An unsupervised framework for sensing individual and cluster behavior patterns from human mobile data," UbiComp '12.

[18] T. Bhattacharya et al., "Automatically Recognizing Places of Interest from Unreliable GPS Data Using Spatio-temporal Density Estimation and Line Intersections," Pervasive and Mobile Computing '14.

[19] N. Deblauwe and P. Ruppel, "Combining GPS and GSM Cell-ID Positioning for Proactive Location-based Services," in MobiQuitous '07.

[20] C. Zhou et al., "Discovering Personal Paths from Sparse GPS Traces," in JCIS '05.

[21] N. Mamoulis et al., "Mining, Indexing, and Querying Historical Spatiotemporal Data," in KDD '04.

[22] Y. Li et al., "Mining Probabilistic Frequent Spatio-Temporal Sequential Patterns with Gap Constraints from Uncertain Databases," in ICDM '13.

[23] M. Ye et al., "On the Semantic Annotation of Places in Location based Social Networks," in KDD '11.

[24] D.Wang et al., "Human Mobility, Social Ties, and Link Prediction," in KDD '11.

[25] S. Isaacman et al., "Identifying Important Places in Peoples Lives from Cellular Network Data," in Pervasive Computing '11, pp. 133–151.

[26] R. Poidevin, "The Experience and Perception of Time," 2009, http://plato.stanford.edu/entries/time-experience.

[27] C. Bettini et al., Time Granularities in Databases, Data Mining, and Temporal Reasoning. Springer, 2000.

[28] V. Srinivasan et al., "Mobileminer: Mining your Frequent Patterns on Your Phone," in UbiComp '14.

[29] R. Agrawal and R. Srikant, "Fast Algorithms for Mining Association Rules in Large Databases," in VLDB '94, 1994, pp. 478–499.

[30] J. Han, J.Pei, and Y. Yin, "Mining Frequent Patterns without Candidate Generation," in SIGMOD '00, 2000, pp. 1–12.