Smartphone Strategic Sampling in Defending Enterprise Network Security

Feng Li*,Wei Peng†, Chin-Tser Huang‡, Xukai Zou†
*Department of Computer Information and Graphics Technology
†Department of Computer and Information Science
Indiana University-Purdue University Indianapolis, Indianapolis, IN, U.S.A.
‡Department of Computer Science and Engineering
University of South Carolina, Columbia, SC, U.S.A.

***Abstract*—Smartphones have made their inroads in enterprise environment, manifested in the Bring Your Own Device (BYOD) policy: More employees are bringing their own smartphones to work and are using them to access enterprise information assets. The susceptibility of smartphones to mobile malware makes them a liability in enterprise network security. The dilemma between responsiveness to security incidents and convenience/cost-effectiveness demands BYOD security solutions beyond the straightforward all-inclusive full-scanning or uniformly random sampling approaches. In this paper, we propose a carefully planned but otherwise random, or *strategic*, sampling approach out of this dilemma. Strategic sampling provides a balance between security responsiveness and cost effectiveness by identifying and periodically sampling those *representative* smartphones (security-wise): Epidemic smartphone malware can be more efficiently detected than in the uniformly random approach, while the sampling is less annoying and intrusive for most users than the all-inclusive approach. Smartphones' security representativeness is measured by the unique traits of smartphones: co-location communication channels in addition to the cellular links, readily available connectivity information, and regular mobility/connectivity pattern of users in the enterprise environment. The probabilities used in strategic sampling are derived from a *lottery tree* that reflects the smartphones' representativeness. We validate the efficiency and effectiveness of the proposed strategic sampling via simulations driven by publicly available, real-world collected traces.**

***Index terms*—Enterprise network, lottery tree, smartphone security, social network, strategic sampling, probabilistic algorithm.**

I. Introduction

With recent report of approximately 1.3 million new Android devices being activated worldwide everyday [1], smartphones have made inroads in enterprise environment, manifested in the *Bring Your Own Device* (BYOD) policy. While this policy appeals to employees' convenience and employers' budgets, smartphones, which are susceptible to abuses such as the Dream Droid mobile malware [2], can be used to compromise an otherwise secure enterprise network. Figure 1 illustrates one such example.

To prevent such security breaches, BYOD smartphones need to be checked for vulnerabilities and malware infections, like their traditional desktop or laptop counterparts. A straightforward approach is to periodically check *all* BYOD smartphones. However, this has a number of issues: Running constantly scanning anti-malware software on smartphones is too costly energy-wise; checking all the smartphones is inconvenient for both the employees and the employer. This security dilemma for BYOD smartphones calls for innovative solutions that balance between security responsiveness and cost effectiveness.

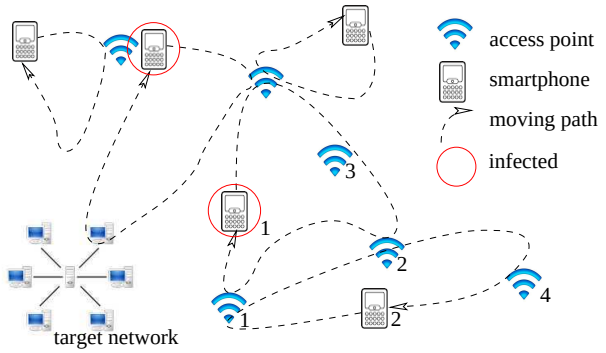


Fig. 1: A susceptible smartphone may be used to compromise an otherwise secure enterprise network. Due to BYOD, malware can be smuggled into the enterprise network through a susceptible smartphone without the owner's awareness.

In this paper, we propose a *carefully planned but otherwise random* sampling approach to address the aforementioned security dilemma. We call such method *strategic sampling*. More specifically, BYOD smartphones will be periodically, randomly *sampled* (i.e., subject to malware scanning) by probabilities commensurable to their *representativeness* in the enterprise network: If the owner of a smartphone shares similar interests or frequently meets with other BYOD employees, this smartphone is *representative* in the enterprise network and, therefore, is more likely to be sampled. Strategic sampling is less intrusive when comparing with the all-inclusive full-scanning approach, and can detect security incidents more promptly than a naive uniform random sampling.

The contributions of this paper are three-fold. First, we identify a few BYOD smartphones' threats to enterprise network security based on the unique characteristics of smartphones. Second, we introduce a method to measure smartphones' (security) representative in an enterprise network based on their owners' interests and the co-location logs on these smartphones. Third, we propose to use strategic sampling to address the BYOD smartphone security problem, which balances between security responsiveness and cost effectiveness;

the probabilities used for strategic sampling are derived from a *lottery tree* that reflects the smartphones' representativeness. We validate the efficiency and effectiveness of the proposed strategic sampling via simulations driven by publicly available, real-world collected traces.

II. Threat Model

The new threats from the BYOD malware are tied with the unique characteristics of the smartphones. Two defining characteristics of smartphones emerge as the market expands: diverse connectivity capabilities and rich auxiliary information. New generations of smartphones are shipped with short-range connectivity technologies (e.g., Wi-Fi, NFC, and Bluetooth) in addition to traditional cellular telecommunication technology (e.g., 3G). The recent convergence to a few dominating mobile software platforms (e.g., iOS and Android) greatly eases mobile application development by providing standard interfaces to access information stored on the smartphone (e.g., contact list and location information). Based on these characteristics, the malware will comprehensively exploit the feasible attack vectors on the smartphones.

A commonly used mobile malware propagation channel is through inadvertently downloading and installing a malware-infected application by unsuspecting users. This malware attack vector is the most effective one in practice as demonstrated by numerous recent malware attacks (e.g., Dream Droid [2] and DroidKungfu [3]). However, the centralized nature of this propagation channel is also the "Achilles' heel" of this kind of malware.

Triggered by the diverse connectivity capabilities of the smartphones, two additional types of attack vectors on the smartphones, *proximity propagation* [4, 5] and *co-location attack* [6], should be taken into consideration. In proximity propagation, the malware propagates through a direct (e.g., Wi-Fi, Wi-Fi Direct, and Bluetooth) communication channel in close physical proximity. It is demonstrated to be a promising alternative to the centralized paradigm. The absence of a single point of failure in the proximity paradigm is especially desirable for malware's survival. Moreover, the recent development of direct local link-based applications, including games (e.g., Flight Control and Real Racing on iOS), file sharing (Goomeo on WindowsMobile), and social network applications (Bump on iOS), opens more opportunities for this type of attack vector. The feasibility of this attack vector is also confirmed by recent development of jailbreak technology (e.g., PDF file exploit on iOS) on the smartphone platform [7]. Therefore, the proximity propagation model can be viewed as the second feasible attack vector for BYOD smartphones.

The third feasible attack vector, co-location attack, is through the indirect local connections (e.g., connecting through the same Wi-Fi Access Point). Recently reported attack techniques [6, 8] show that Wi-Fi co-location (i.e., smartphones associated with the same Wi-Fi access point at the same time) is a feasible malware injection channel. An example attack is the ARP-poisoning-based man-in-the-middle attack [6]. Besides the existing threats and attacks, these three attack vectors also cover and characterize the possible attack models of the zero-day threat on the BYOD smartphones.

For the attacking phase, the infected smartphone, which can reach the targets, e.g., isolated enterprise networks or enterprise networks with strict access control, could become a *wormhole* to the target network, or directly bring the malicious payload to the target network. In an example attack scenario, the bot-master may connect to a backdoor on the smartphone through the Wi-Fi/Cellular interface, inject the malicious payload through the USB interface to the enterprise network, and bypass the access control mechanisms. The feasibility of the wormhole capability has been proven by third-party tethering applications (e.g., PdaNet for Android) [9].

In this paper, we focus on the infection and interaction of mobile elements (i.e. smartphones) in the investigations. In reality, a malware with hybrid infection capabilities (i.e. infect both smartphones and static computers) will be more effective to achieve the attack goals. However, the malware propagation and detection in the static computer network part in this hybrid environment would not bring significant differences to the existing research. Therefore, we omit this part in the discussion and focus on the investigation of BYOD smartphone sampling.

III. Design

BYOD smartphones need to be periodically checked for their risk to the enterprise's information assets. In this section, we present *strategic sampling* as a solution that shares the effectiveness of universal checking (check all the smartphones) and the efficiency of random sampling (sample the smartphones with equal probabilities). Strategic sampling consists of three components: 1) comprehensive social topology extraction; 2) social-aware lottery tree formation; 3) strategic sample selection.

A. Comprehensive Social Topology Extraction

Social connections between BYOD employees can be represented by a social graph, with nodes representing the employees and links between nodes representing social connections. For example, two BYOD employees who are in the same department or the liaisons between two departments may share a link. Intuitively, social connections have different strength. Concretely, we derive the social connection strength from two BYOD-specific sources: 1) Wi-Fi co-location logs from enterprise access points (APs); 2) smartphone owner's interests reflected by installed apps.

Co-location log interpretation. Smartphone users' mobility and social connections have patterns [10]: Social closeness, reflected by the presence and frequency of *future* contact opportunities, can be estimated from *past* connections captured by enterprise Wi-Fi APs or location-based services like Foursquare and Google Latitude [11].

In an enterprise IT environment, association and deassociation events of BYOD devices with enterprise Wi-Fi APs are logged and readily available for processing. More specifically, connectivity log captures the *temporal order* and *duration* of a smartphone's connection with APs. A connectivity log consists of entries with three fields: start timestamp (ST), end timestamp (ET), and access point identifier (APID).

From connectivity logs, a metric, *reachability*, can be derived to estimate how soon a smartphone can expect to reach others. Given a pair of connectivity log instances, l_a and l_b,

for two different smartphones, we can find, within the time window $[t-w, t]$, the temporal intervals during which the two smartphones are co-located. We can define the reachability, $r(a, b)$, to be the *expected* waiting time of the next encounter, computed by:

$$r(a,b) = \int_{e_1}^{t_{k+1}} g(m)dm/w = \sum_{i=1}^{k}(t_{i+1} - e_i)^2/2w. \quad (1)$$

Here, we assume the common temporal intervals of l_a and l_b in the time window $[t-w, t]$ to be $[t_1, e_1]$, ..., $[t_k, e_k]$, and $t_{k+1} = t_1 + w$. At a particular moment, m ($t_1 \leq m \leq t_{k+1}$), the waiting time, $g(m)$, until the next encounter of the two smartphones is: 0 when the smartphones are co-located; and $\min_{t_i \geq m}(t_i - m)$ otherwise. As a special case, if the two smartphones are not co-located during $[t-w, t]$, we define their reachability to be $+\infty$.

Closeness information abstraction. We can further augment the reachability metric based on the similarities of users' interests. Sociology studies suggest that individuals with similar interests tend to share information more often [12]. This has been confirmed by studies on the relationship between mobility patterns and users' interests [13]. On the other hand, users with similar interests are susceptible to similar vulnerabilities. An BYOD employee's interests can be represented by a point in an m-dimensional interest space, with each dimension representing a different topic and the projection on that dimension representing the level of interest in that topic. From the install apps on a BYOD employee's smartphone, the interest in each of the pre-specified topics can be estimated. We use cosine similarity [14] to measure the similarity of interests of two BYOD employees, a and b (let V_a and V_b be a and b's points in the interest space, respectively):

$$s(a,b) = \cos(\theta) = \frac{\sum_{i=1}^{n}(V_a)_i \cdot (V_b)_i}{\sqrt{\sum_{i=1}^{n}((V_a)_i)^2} \cdot \sqrt{\sum_{i=1}^{n}((V_b)_i)^2}}. \quad (2)$$

We combine reachability and interest similarity by weighted sum to derive the *closeness* on a social link. More specifically, closeness $c(a, b)$ is the weighted sum of the similarity metrics and the multiplicative inverse of the reachability: $c(a,b) = \alpha \cdot s(a,b) + (r(a,b))^{-1}$, with α being a system parameter depending. Closeness reflects the similarity in the vulnerabilities (and, hence, the risk to the enterprise information assets) of two BYOD smartphones.

B. Social-aware Lottery Tree Formation

Best-friend identification. Enterprise social networks are often clustered. If we rank the persons that an employee meet in the company by frequencies, the top few are relatively stable. For example, the top contacts may be the employee's team/office mate or boss. For convenience, we call these top contacts the *friends* of the employee. Inspired by this observation, given an enterprise social graph, we define the *best friend* of a node a to be the node b that is most close to a, i.e., has the largest closeness value; this is denoted by $b = BF(a)$. In case of a tie, we use the minimal node ID (or consistently using some arbitrary criterion) to break the tie. The *directed* link from a to b is the *best-friend link*.

Algorithm 1 Strategic Sampling Process

```
Calculate r(a,b), s(a,b), and c(a,b) for each link (a,b);
Identify b = BF(a) for each node a and construct the tree;
Calculate S_a = Σ_{b∈N(a)} c(a,b) for each node a;
Calculate S_T = Σ_{a∈T} S_a;
WeightCalculation (root, tree);
Sample node a with probability Q_a / Σ_{i∈E} Q_i;
Mark sub(a) as sampled if node a is selected;

procedure WEIGHTCALCULATION(i, sub(i))
    if chd(i) = ∅ then Q_i = f(S_i / S_T);
    else
        for each node l ∈ chd(i) do
            WeightCalculation(l, sub(l));
        end for;
        Calculate S_{sub(i)} = Σ_{l∈sub(i)} S_l;
        Calculate Q_i = f(S_{sub(i)} / S_T) − Σ_{l∈chd(i)} f(S_{sub(l)} / S_T);
    end if;
end procedure
```

Therefore, each node has at most one best friend out-link, but may have multiple best-friend in-link. From the best-friend links, we construct a *lottery tree* based on best-friend links as follows. Due to the stability of a node's best friend, the lottery tree requires few updates after construction.

Node placement and subtree formation. For a node a, if $b = BF(a)$, and $BF(b) \neq a$, then b is the parent of a in the lottery tree. Therefore, for any node, its number of direct descendants (children) depends on the number of best-friend in-links, which partially reflect the importance (security representativeness) of the node in the social topology.

For the case where $b = BF(a)$ and $a = BF(b)$, the number of best-friend in-links is used to break the tie: The node with the larger number of best-friend in-links will be the parent node; the parent will then become a child of the root of the lottery tree, i.e., a level-1 decedent of the root. The root of the lottery tree is a virtual node that links all the subtrees of the level-1 decedents together.

Property 1 (Loop-Freeness) There is no loop in the best-friend chain: If $b = BF(a)$ and $u = BF(b)$, a can *not* be $BF(u)$.

Proof: The closeness metric is symmetric: $c(a,b) > c(b,a)$. $b = BF(a)$ implies $c(a,b) > c(a,u)$ (or $c(a,b) = c(a,u)$ and $ID_b < ID_u$). $a = BF(u)$ implies $c(a,u) > c(b,u)$ (or $c(a,u) = c(b,u)$ and $ID_a < ID_b$). These implications contradict with $u = BF(b)$, which implies $c(b,u) > c(a,b)$ (or $c(a,b) = c(a,u) = c(b,u)$ and $ID_u < ID_a$). Therefore, a best-friend loop among three or more nodes cannot exist. □

By Property 1, the structure constructed by the above rule is indeed a tree, i.e., a single-parented acyclic graph.

C. Strategic Sample Selection

The sampling probability should be assigned according to the representativeness of each BYOD smartphone. The representativeness depends on two factors: 1) relative importance in the social topology; 2) diversity based on clustering.

Each node's individual relative importance in the social topology is reflected by the *closeness degree centrality*, which is the sum of all the closeness values on the social links

from this node: $S_a = \sum_{b \in N(a)} c(a,b)$. Intuitively, a owner of smartphone a with large S_a shares similar interests and meets with many other BYOD employees. Such a smartphone has a higher chance of being infected by their contacts, which makes it a good candidate for sampling.

Since the lottery tree is constructed based on the best-friend relationship, nodes in the different subtrees are more intuitively farther away from each other than nodes in the same subtrees. On the other hand, for a larger and more socially active group/subtree, it is desirable to assign larger sampling probability to this group/subtree, since they are more likely to be infected through frequent contacts. Therefore, in addition to the closeness degree centrality, the sampling probabilities should reflect a node's placement in the tree to achieve the social group diversity.

In our design, the sampling probability of any root of a subtree will be pushed up according to the total relative importance of the nodes in that subtree, defined by the sum of the closeness degree centrality: $S_{sub(a)} = \sum_{i \in sub(a)} \sum_{b \in N(i)} c(i,b)$. The following property reflects the intuitively desirable features described above.

Property 2 (Social-Representativeness) The sampling probability Q_a of a node a should satisfy:
1) If S_a increased to S'_a, then $Q_a(S_a) < Q_a(S'_a)$;
2) If $S_a = S_b$ and $S_{sub(a)} > S_{sub(b)}$, then $Q_a > Q_b$.

The first clause in Property 2 ensures that sampling probabilities reflect the relative importance in the social topology. The second clause in Property 2 promotes the diversity based on the group structure: Larger $S_{sub(a)}$ leads to larger sampling probability. However, we also don't want the sampling probability of the root of the subtree to increase too fast because of $S_{sub(a)}$, since that will make the smartphones on the leaf positions highly unlikely to be selected for sampling and, hence, make them ideal targets for an attacker.

To balance the relative importance and diversity requirements formalized by Property 2, we use a strictly convex function $f(x)$ in assigning sampling probabilities: $f(x) = \beta \cdot x^{(1+\delta)} + (1-\beta) \cdot x$. Here, $\beta \in [0,1]$ and $\delta \in (0, +\infty)$ are configurable system parameters. We design the sampling probability assignment for node a as follows:

$$Q_a = f(\frac{S_{sub(a)}}{S_T}) - \sum_{l \in chd(a)} f(\frac{S_{sub(l)}}{S_T}) \qquad (3)$$

Here, S_T is the sum of the closeness degree centrality of all the nodes in a tree T, $chd(a)$ includes all children of a. There are two spacial cases in this process. First is the virtual root, its sampling probability is 0. Second, if node a is a leaf node without any child in the tree, Equation 3 still holds and $Q_a = f(\frac{S_a}{S_T})$. The sampling probability assignment based on Equation 3 guarantees Property 2. In addition, the design leads to the following desirable property.

Property 3 (Sampling-Fairness) There exists a $\phi > 0$, such that for any smartphone a, a's sampling probability should be no less than $\phi \cdot \frac{S_a}{S_T}$.

Proof: For a node a, we have:

$$\begin{aligned} Q_i &= f(\frac{S_{sub(a)}}{S_T}) - \sum_{l \in chd(a)} f(\frac{S_{sub(l)}}{S_T} \\ &= (\frac{1}{S_T})^{(1+\delta)} \cdot \beta((\sum_{l \in chd(a)} S_{Sub(l)} + S_a)^{(1+\delta)} \\ &\quad - \sum_{l \in chd(a)} (S_{Sub(l)})^{(1+\delta)}) + (1-\beta)\frac{S_a}{S_T}. \end{aligned}$$

Since $((\frac{1}{S_T})^{(1+\delta)} \cdot \beta((\sum_{l \in chd(a)} S_{Sub(l)} + S_a)^{(1+\delta)} - \sum_{l \in chd(a)} (S_{Sub(l)})^{(1+\delta)}) \geq 0$, we have: $Q_a \geq (1-\beta)\frac{S_a}{S_T}$. Therefore, our design satisfies property 3 with $\phi = (1-\beta)$.□

In the case in which multiple nodes are selected for one round of strategic sampling, the same sample selection process can be applied multiple times. When a smartphone a has been selected to be a sampling point, the subtree rooted at a will be marked and excluded from subsequent rounds of selection. If all subtrees have been marked but more sampling points are needed, the selection will restart with the marked nodes (not the whole subtree) removed. In each round of strategic sampling, since the sampling probabilities for root and excluded nodes are 0, each eligible node will be selected with an adjusted probability of $\frac{Q_a}{\sum_{i \in E} Q_i}$ with E being the set of all nodes in T that have not been excluded yet. The adjustment of sampling probabilities only affects those eligible nodes: The adjusted assignment still satisfies the desirable properties. Algorithm 1 summarizes the strategic sampling process.

IV. Simulations

We validate the proposed lottery-tree-based strategic sampling mechanism with simulations driven by a real-world collected dataset.

A. Methodology

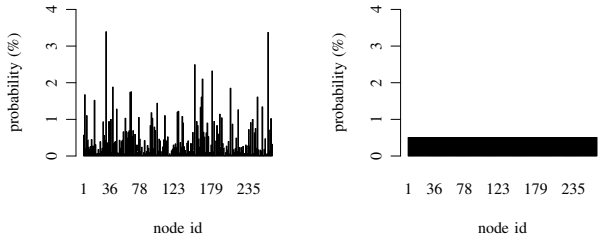
We used the dataset from the Wireless Topology Discovery (WTD) project[1] in our simulation.

The dataset consists of traces collected from UC San Diego freshmen for an 11-week period. The traces were collected with Wi-Fi-enabled PDAs, periodically recording the MAC address of associated APs. The students participating in this experiment, though coming from different majors, resided in the same university housing facility. Therefore, the traces capture the mobility and connectivity patterns of a group of users in a relatively short period of time [15].

We converted the periodic beacon records into a series of association/disassociation events; these events were then replayed in a customized simulator implemented with Perl. The simulator allowed us to obtain the reachability between devices, find the best-friend links, and construct the lottery tree.

We compared the proposed strategic sampling based on lottery tree with the purely random sampling, which samples each smartphone with equal probability. Figure 2 contrasts the probabilities derived from the lottery tree with that of the uniform distribution: Unlike the uniform distribution used in

[1] http://sysnet.ucsd.edu/wtd/data_download/wtd_data_release.tgz



(a) Strategic sampling by lottery-tree probabilities. (b) Random sampling by uniform distribution.

Fig. 2: Unlike the uniform distribution used in random sampling, the probabilities used in strategic sampling is derived from the lottery tree and vary among different nodes.

the random sampling, the probabilities derived from the lottery tree, which is used in the strategic sampling, vary among different nodes.

We adopted the Susceptible-Infectious-Recovered (SIR) model [16] from epidemiology in simulating malware propagation and sampling. Sampling nodes were first randomly selected with the probabilities specified by the (strategic or random) sampling rule; initially infected nodes were uniformly selected from the rest of the nodes. All other nodes were initially susceptible to the malware. When the simulation began, two types of status change took place when nodes came into contact with each other through AP co-location: 1) A susceptible node that came into contact with an infected node would be infected; 2) an infected node that came into contact with a sampling node would be sampled and patched, and became immune to the malware. The simulation were repeated over 200 times for each setting (a given number of sampling/initially infected nodes) to bring forth the effect of different sampling probabilities.

B. Results

We compared the efficiency and effectiveness, as defined and explained below, of the different sampling methods.

1) Efficiency: We measured the *efficiency* of a sampling method by the delay between the epidemic outbreak and the first detection: The shorter the delay is, the more efficient the sampling method is. Figure 3 shows the simulation result, in boxplot [17], with different number of sampling/initially infected nodes.

The number of initially infected nodes increases from left to right; the number of sampling nodes increases from top to bottom. The result indicates that increasing sampling/initially infected nodes led to a decrease of the delay: Sampling nodes and initially infected nodes were more likely to come into contact early if there were more of them initially.

In all cases shown in Figure 3, strategic sampling based on lottery tree had an overall shorter detection delay in comparison with random sampling based on uniform distribution. The difference is more pronounced with more sampling nodes, i.e., from top to down in Figure 3. An explanation is that, with more sampling nodes, more topologically important nodes would be chosen for checking by the strategic sampling rule

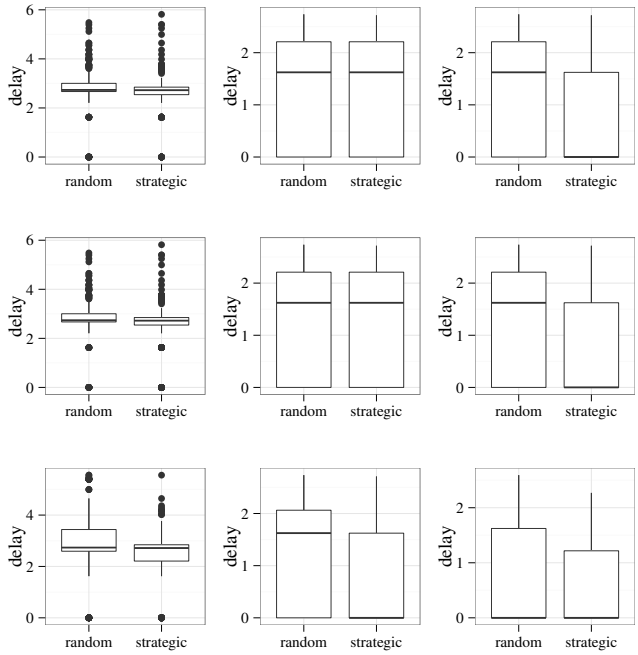


Fig. 3: Efficiency measured by the delay between the epidemic outbreak and the first detection. Log-scale is employed for clearer presentation: If the epidemic breaks out at time 0 and the first infection is detected at time T, delay on the y axis is defined as $\log_{10}(T+1)$. The shorter the delay is, the more efficient the sampling method is. The number of initial infected smartphones is 1, 10, and 20 from left to right; the number of chosen smartphones for sampling is 1, 10, and 20 from top to bottom.

based on the lottery tree and, hence, the epidemic could be detected sooner.

2) Effectiveness: We measured the *effectiveness* of a sampling method by the number of infected nodes still undetected, or missed, by the end of a round of simulation: The fewer get missed, the more effective the sampling method is. Figure 4 shows the simulation result in a way similar to Figure 3.

The result indicates that more sampling nodes led to fewer infected nodes were missed in detection, and more initially infected nodes led to more infected nodes were missed in detection; both agree with intuition.

Similar to the efficiency result, in all cases shown in Figure 4, strategic sampling based on lottery tree missed fewer infected nodes in comparison with random sampling based on uniform distribution. The difference is more pronounced with more initially infected nodes: Infected nodes were more likely to evade detection if there were more of them initially.

In summary, simulation results indicate that strategic sampling based on lottery tree are more efficient and effective than random sampling based on uniform distribution.

V. Related works

Employees desire BYOD [18]. However, security challenges [19, 20] for BYOD demand novel solutions [21]. This paper presents an efficient and effective solution to the mobile malware problem [22] in BYOD.

Recently, mobile malware has received attention from both the general public [2] and the research community [22] due to recent security incidents, such as the Dream Droid mobile malware [2]. The mobile malware can be used for a variety of

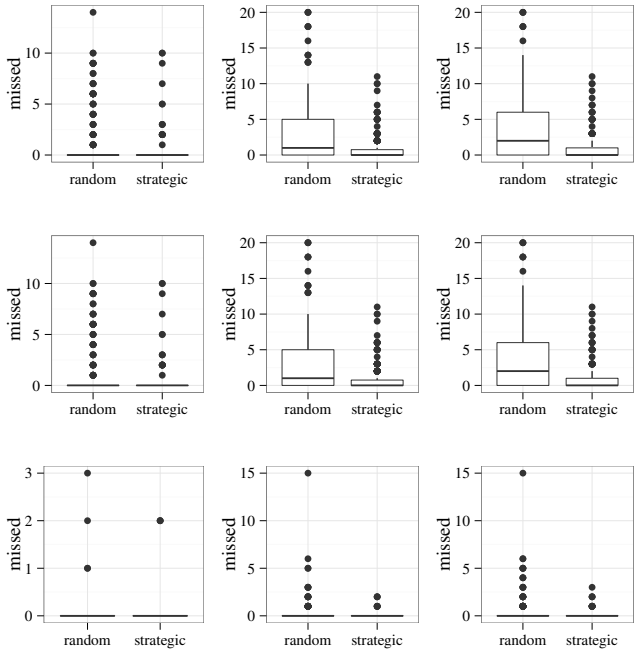


Fig. 4: Effectiveness measured by the number of infected smartphones that are still undetected (missed) by the end of the simulation. The fewer get missed, the more effective the sampling method is. The number of initial infected smartphones is 1, 10, and 20 from left to right; the number of chosen smartphones for sampling is 1, 10, and 20 from top to bottom.

criminal activities, including identity and password theft [23], launching denial-of-service attacks [24], and sending spam-emails [25]. With emergence of BYOD, mobile malware is likely to become a major issue in enterprise network security.

Random sampling schemes were previously used in intrusion and malware detection [26, 27, 28]. We propose to identify and sample representative nodes and derive the sampling probabilities based on a lottery tree. Lottery tree is inspired by previous works on incentive design in distributed systems [29, 30].

VI. Conclusion

In this paper, we present a strategic sampling scheme for enterprise network security. We introduce the threat model based on the unique features of the BYOD smartphones. We present methods to extract two types of social intelligence, reachability and similarity, from the available data in the enterprise environment. Each smartphone is assigned a sampling probability commensurate to its security representativeness in the enterprise network based on a lottery tree. The efficiency and effectiveness of strategic sampling are validated against publicly available, real-world collected traces.

References

[1] W. Mangalindan. (2012) Today in Tech: 1.3 million Android devices activated daily, says Eric Schmidt. http://tech.fortune.cnn.com/2012/09/06/1-3-million-android-devices-activated/.

[2] O. Krehel. (2011) Worse than Zombies: the Mobile Botnets are Coming. http://www.idt911blog.com/2011/06/worse-than-zombies-the-mobile-botnets-are-coming/.

[3] X. Jiang. Security alert: New sophisticated android malware droidkungfu found in alternative chinese app markets. [Online]. Available: http://www.csc.ncsu.edu/faculty/jiang/DroidKungFu.html

[4] G. Zyba, G. Voelker, M. Liljenstam, A. Méhes, and P. Johansson, "Defending mobile phones from proximity malware," in *Proc. of INFOCOM*. IEEE, 2009.

[5] F. Li, Y. Yang, and J. Wu, "Cpmc: an efficient proximity malware coping scheme in smartphone-based mobile networks," in *Proc. of INFOCOM*. IEEE, 2010.

[6] B. Zdrnja, "Malicious JavaScript Insertion through ARP Poisoning Attacks," *IEEE Security and Privacy*, vol. 7, no. 3, pp. 72–74, 2009.

[7] T. Nouveau. (2011) New browser-based iOS jailbreak uses PDF exploit. http://www.tgdaily.com/software-features/57063-new-browser-based-ios-jailbreak-uses-pdf-exploit.

[8] C. Papathanasiou and N. Percoco, "This is not the droid you're looking for..." in *Presentations of DEF CON 18*, 2010.

[9] M. Joire. (2012) ClockworkMod Tether serves free Android USB tethering, no root required. http://www.engadget.com/2012/01/03/clockworkmod-tether-serves-free-android-usb-tethering-no-root-r/.

[10] M. Afanasyev, T. Chen, G. Voelker, and A. Snoeren, "Usage patterns in an urban WiFi network," *IEEE Transactions on Networking*, vol. 18, pp. 1359–1372, 2010.

[11] N. Li and G. Chen, "Sharing location in online social networks," *IEEE Network*, vol. 24, no. 5, pp. 20 –25, september-october 2010.

[12] M. McPherson, "Birds of a feather: Homophily in Social Networks," *Annual Review of Sociology*, vol. 27, no. 1, pp. 415–444, 2001.

[13] A. Noulas, M. Musolesi, M. Pontil, and C. Mascolo, "Inferring interests from mobility and social interactions," in *Proc. Workshop on Analyzing Networks and Learning with Graphs*, 2009.

[14] M. Deza and E. Deza, *Encyclopedia of Distances*. Springer, 2009.

[15] M. McNett and G. Voelker, "Access and mobility of wireless pda users," *ACM SIGMOBILE Mobile Computing and Communications Review*, vol. 9, no. 2, pp. 40–55, 2005.

[16] F. Brauer, P. Van den Driessche, J. Wu, and L. Allen, *Mathematical epidemiology*. Springer Verlag, 2008, vol. 1945.

[17] M. Frigge, D. Hoaglin, and B. Iglewicz, "Some implementations of the boxplot," *The American Statistician*, vol. 43, no. 1, pp. 50–54, 1989.

[18] S. Mansfield-Devine, "Interview: BYOD and the enterprise network," *Computer Fraud & Security*, vol. 2012, no. 4, pp. 14–17, 2012.

[19] G. Thomson, "BYOD: enabling the chaos," *Network Security*, vol. 2012, no. 2, pp. 6–8, 2012.

[20] J. Burt, "BYOD trend pressures corporate networks," *eWeek*, vol. 28, no. 14, pp. 30–31, 2011.

[21] Y. Wang, K. Streff, and S. Raman, "Security Threats and Analysis of Security Challenges in Smartphones," *Computer*, vol. PP, no. 99, p. 1 8, 2012.

[22] C. Xiang, F. Binxing, Y. Lihua, L. Xiaoyi, and Z. Tianning, "Andbot: towards advanced mobile botnets," in *Proc. USENIX Conference on Large-scale Exploits and Emergent Threats*, 2011.

[23] P. Wang, L. Wu, B. Aslam, and C. Zou, "A systematic study on peer-to-peer botnets," in *Proc. of the International Conference on Computer Communications and Networks (ICCCN)*, 2009.

[24] M. J. S. Kandula, D. Katabi and A. Berger, "Botz-4-sale: Surviving organized ddos attacks that mimic flash crowds," in *Proc. of Second Symposium on Networked Systems Design and Implementation (NSDI)*, 2005.

[25] L. Zhuang, J. Dunagan, D. Simon, H. Wang, and J. Tygar, "Characterizing botnets from email spam records," in *Proc. of the 1st Usenix Workshop on Large-Scale Exploits and Emergent Threats*, 2008.

[26] D. Brauckhoff, B. Tellenbach, A. Wagner, M. May, and A. Lakhina, "Impact of packet sampling on anomaly detection metrics," in *Procs. of the 6th ACM SIGCOMM conference on Internet measurement*, 2006.

[27] C. Kolbitsch, P. Comparetti, C. Kruegel, E. Kirda, X. Zhou, and X. Wang, "Effective and efficient malware detection at the end host," in *Procs. of the 18th conference on USENIX security symposium*, 2009.

[28] G. Gu, R. Perdisci, J. Zhang, and W. Lee, "Botminer: clustering analysis of network traffic for protocol- and structure-independent botnet detection," in *Procs. of the 17th conference on Security symposium*, 2008.

[29] F. Li and J. Wu, "Frame: An innovative incentive scheme in vehicular networks," in *Proc. of the IEEE International Conference on Communications (ICC)*, 2009.

[30] J. Douceur and T. Moscibroda, "Lottery trees: motivational deployment of networked systems," in *Proc. of the ACM SIGCOMM*, 2007.