T-dominance: Prioritized Defense Deployment for BYOD Security

Wei Peng*, Feng Li†, Keesook J. Han§, Xukai Zou*, and Jie Wu‡
*Department of Computer and Information Science
†Department of Computer, Information, and Leadership Technology
Indiana University-Purdue University, Indianapolis, Indianapolis, IN, U.S.A.
§ Air Force Research Laboratory, Rome, NY, U.S.A.
‡Department of Computer and Information Sciences
Temple University, Philadelphia, PA, U.S.A.

***Abstract*—Bring Your Own Device (BYOD) is an enterprise information technology (IT) policy that encourages employees to use their own devices to access sensitive corporate data at work through the enterprise IT infrastructure. Many current BYOD security practices are costly to implement and intrusive to employees, which, to some degree, negate BYOD's perceived benefits. To address such tension, we propose *prioritized defense deployment*: Instead of employing the same costly and intrusive security measures on each BYOD smartphone, more stringent threat detection/mitigation mechanisms are deployed on those representative smartphones, each of which represents, security-wise, a group of smartphones in the whole BYOD device pool. To this end, we propose a concept and a distributed algorithm, both named T-*dominance*, to capture the temporal-spatial pattern in an enterprise environment. We identify a few desirable properties of prioritized defense deployment, and analytically show that T-dominance satisfies such properties. We complement our analysis with simulations on real Wi-Fi association traces.**

***Index terms*—BYOD, prioritized defense deployment, security representativeness, temporal-spatial pattern**

I. Introduction

Bring Your Own Device (BYOD) is an enterprise information technology (IT) policy that encourages employees to use their own devices to access sensitive corporate data at work through the enterprise IT infrastructure. Employees' demand/satisfaction, decreased IT acquisition and support cost, and increased use of cloud/virtualization technologies in enterprise IT infrastructure are common justifications for adopting BYOD [1]. With the consumerization of smartphones and tablet computers (smartphones for brevity) in recent years, the demand for using personal smartphones in the workplace has brought BYOD to the attention of enterprise IT professionals as one of the "tech trends for 2013 [2]."

Despite the commonly cited benefits, BYOD presents significant security challenges. On the one hand, forwarding corporate e-mails to public Web mail services, using public cloud-based storage services (e.g., Dropbox and Apple's iCloud) to store corporate documents, or even interacting with smartphones through voice in the workplace may leak sensitive corporate information assets [3]; moreover, employees may inadvertently or maliciously introduce malware to the enterprise network behind the firewalls through their own malware-infected smartphones. On the other hand, forcing employees to disable common applications such as Dropbox [3], though may be necessary security-wise, significantly worsen employees' BYOD experience; frequently auditing the use of employees' smartphones not only intrudes on their convenience, but is also costly to implement.

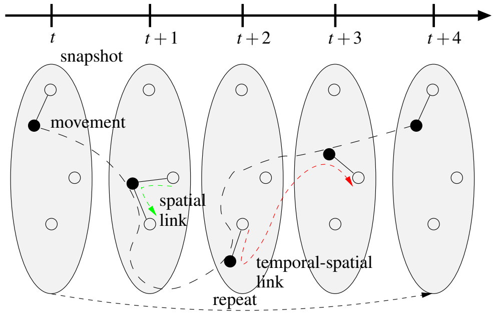

Fig. 1: T-dominance exploits temporal-spatial patterns of BYOD devices to implement prioritized defense deployment. The black node T-dominates the white ones for $T > 4$.

To address such tension, we propose *prioritized defense deployment*: Instead of employing the same costly and intrusive security measures on each BYOD smartphone, more stringent threat detection/mitigation mechanisms are deployed on those representative smartphones, each of which represents, security-wise, a group of smartphones in the whole BYOD device pool.

In this paper, we interpret and measure security representativeness through the temporal-spatial pattern inherent in an enterprise environment: Those BYOD smartphones that connect with *many* other smartphones *often* are representative security-wise, because they are exposed to more attacks and have more severe consequences if compromised.

More specifically, we interpret and measure security representativeness with a novel temporal-spatial structural property and propose a distributed algorithm (running distributedly on individual smartphones) that robustly preserves that property. We name both the property and the algorithm T-*dominance*, in which T is a temporal bound. Each BYOD smartphone executes the T-dominance algorithm and, based on potentially outdated information from proximate smartphones (as briefly discussed in Section II, such information is readily available on many consumer smartphones), estimates its security representativeness. If a smartphone considers itself as representative, it turns into

Approved for Public Release; Distribution Unlimited: 88ABW-2012-4117, 25-Jul-2012.

an *agent*. The algorithm needs no central coordination, which reduces maintenance overhead for enterprise IT administration and is less intrusive to BYOD employees. After running the algorithm for awhile, the whole BYOD smartphone pool will be T-dominated by the agents: Each smartphone is either an agent, or is highly likely to be proximate to an agent with a delay not exceeding T. The idea of T-dominance is illustrated in Figure 1. A more intrusive and costly defense mechanism will be deployed on the agents.

Prioritized defense deployment based T-dominance provides an adjustable (through T) balance between security provision and mechanism intrusiveness/cost. We define the concept of T-dominance and present an algorithm to implement it (Section III). We show the temporal robustness and the effectiveness of the proposed algorithm through analysis (Section IV) and trace-driven experiments (Section V), and put our works in the context of previous research (Section VII).

In summary, we make the following contributions.

- We propose prioritized defense deployment based on security representativeness as a solution to the tension between the demand for BYOD security practices and the intrusiveness/cost of such practices.
- We propose a novel interpretation of security representativeness, based on the inherent temporal-spatial structures in an enterprise environment, and illustrate the application of the concept: strategic sampling to detect malware, prioritized patching to prevent or recover from damage.
- We propose a method, T-dominance, to capture the temporal-spatial dynamics of BYOD smartphone networks in a graph structure (Definition 1) and maintain such a structure with an algorithm that does not incur extra administration cost, and is less intrusive to employees (Section III).
- We show the temporal robustness and the effectiveness of the proposed algorithm through analysis (Section IV) and trace-driven experiments (Section V). The temporal robustness ensures that the T-dominance algorithm will maintain the T-dominance structural property on potentially outdated information, due to the absence of constant, central coordination.

II. Model

Due to the wide deployment of Wi-Fi infrastructure in enterprise networks and the wide availability of Wi-Fi co-location information on smartphones (to support, for example, location-based services), we consider a threat model that includes, besides the common drive-by download attack, smartphone malware that can infect Wi-Fi co-located smartphones through techniques such as ARP poisoning; we briefly discuss the feasibility and current state of such proximity malware attacks in Section VI.

Each smartphone maintains a connectivity log of past access point associations, with entries in the form of $(ST = s, ET = e, APID = \text{AP}_i)$ indicating that the smartphone is associated with access point AP_i from time s to e. Connectivity logging is a standard feature on major mobile platforms, such as the consolidated.db in iOS's location-aware services [4].

Given the connectivity log of a pair of smartphones, u and v, we can find the maximal temporal intervals during which the two smartphones are co-located within the temporal window $[t-W, t]$ of size W^1: $[s_1, e_1], [s_2, e_2], \ldots, [s_k, e_k]$. Let $s_{k+1} = s_1 + W$; we have $s_1 < e_1 < \ldots < s_i < e_i < \ldots < s_k < e_k \leq s_{k+1} = s_1 + W$.

At a particular moment m ($t - W \leq m \leq t$), the waiting time $g(m)$ before the next encounter between u and v is:

$$g(m) = \begin{cases} 0 & \exists i, \text{s.t. } s_i \leq m \leq e_i, \\ \min_{s_i \geq m}(s_i - m) & \text{otherwise.} \end{cases} \quad (1)$$

Thus, we define the expected delay $\mathrm{r}(u, v)$ till next encounter between u and v at time t as their *reachability*, computed by:

$$\mathrm{r}(u, v) = \frac{\int_{s_1}^{s_{k+1}} g(m)\mathrm{d}m}{W} = \frac{\sum_{i=1}^{k}(s_{i+1} - e_i)^2}{2W}. \quad (2)$$

As a special case, if the two smartphones are not co-located between $t - W$ and t (reflected by the lack of common intervals in l_1 and l_2 during that temporal window), their reachability is defined to be $+\infty$. The definition of reachability in Equation (2) has implications (Lemma 1) on our design (Section IV)^2.

Given a set of smartphones $P = \{u, v, w, \ldots\}$ along with their connectivity logs, we define the *reachability graph* $G(P)$ of P to be a weighted undirected graph with P as the vertices and $\mathrm{r}(u, v)$ as the weights on the edges between two smartphones u and v. Given a threshold T, we define the *filtered reachability graph* $G^T(P)$ to be the subgraph of $G(P)$ consisting of all the vertices along with those edges with weights no greater than T.

III. Design

A. Motivation: prioritized defense deployment

Threat detection/mitigation in an enterprise network is an ongoing, rather than a one-shot, process. Threat detection/mitigation mechanisms, such as malware detection and vulnerability patching, need to be deployed on BYOD smartphones and regularly updated to defend against evolving and emerging threats. Doing so constantly on all BYOD smartphones is costly for the enterprise, and intrusive to the employees. *Random sampling* is less costly and intrusive, but is oblivious to the temporal diversity of BYOD employees' connectivity patterns and, thus, presents challenges such as how many and how often devices shall be checked for security vulnerabilities and receive updates, as well as how to quantify the security provision.

Prioritized defense deployment addresses these challenges by assigning each BYOD smartphone one of two mutually exclusive roles, *agents* and *non-agents*, according to its security representativeness, and prioritizing the agents for defense mechanism deployment. The use of the neutral terms (agent and non-agent) to differentiate the security representativeness brings forth the essence of such distinction without confining prioritized defense deployment to one narrow scenario. For example, in the context of proximity malware attacks, prioritized defense deployment can support *strategic sampling* for detecting

[1] Temporal window is used in the definition of reachability to phase out old information that may be outdated. An example is that, for an employee who transferred from one department to another two days ago, a temporal window W of 2 days will exclude the information before the transfer when computing reachability.

[2] In Equation (2), we use $\int_{e_1}^{s_{k+1}} g(m)\mathrm{d}m$, instead of $\int_{t-W}^{t} g(m)\mathrm{d}m$, as the numerator; effectively, we cut the temporal interval $[t-W, s_1]$ and paste it to the right of $[t-W, t]$; then we take an interval of length W from the right to form the interval $[s_1, s_1 + W]$, i.e., $[s_1, s_{k+1}]$. This ensures the temporal robustness of the reachability metric in Theorem 1 (more specifically, in Lemma 1).

malware, and *prioritized patching* for preventing/recovering from malware attacks.

- In strategic sampling, the agents resemble traditional Internet *honeypots* for intrusion detection [5]: They attract and expose propagating malware. The agents are periodically checked for malware infection by enterprise IT security staff. Prioritized defense deployment will choose those security-wise representative smartphones as agents, and hence, provide a quantifiable security provision for detecting malware.
- In prioritized patching, the agents resemble the high-risk population (prior to their immunization) and vaccine depot (after their immunization) in human epidemiology: They are high-risk target of malware (prior to being patched against the malware) due to their temporal-spatial importance in connecting the network; they are also good deliverers of the security patches (after being patched) for the same reason.

Strategic sampling is reactive, and prioritized patching is proactive: Whereas an agent in the former waits for a co-located smartphone to infect it, an agent in the latter actively distributes patches to co-located smartphones. Nevertheless, in both applications of prioritized defense deployment, a smaller number of agents lowers the sampling/patching cost for enterprise IT management, and reduces intrusiveness to employees; it is, therefore, more desirable.

In this paper, we propose T-dominance as an approach to implement prioritized defense deployment. In the rest of the section, we define the concept of T-dominance (Section III-B) and design a localized and temporally robust algorithm for electing a T-dominating agent set in a BYOD network for prioritized defense deployment (Section III-C).

B. T-dominance: the concept

The concept of T-dominance is defined on the filtered reachability graph $G^T(P)$ (Section II) over a network of smartphones P as follows.

Definition 1 (T-dominance)**.** *Let P be a set of smartphones and A be a subset of P called the* agents. *We say that the agents A T-dominates the smartphones P at moment t if, for any $u \in G^T(P)$, either $u \in A$ or u is a neighbor of an agent $a \in A$ in $G^T(P)$.*

By definition, P trivially T-dominates itself. We are interested in a non-trivial A that T-dominates P. For prioritized defense deployment based on T-dominance, a small A is desirable.

T-dominance quantifies the security provision in both strategic sampling and prioritized patching (Section III-A). For example, consider that a Wi-Fi co-location-based epidemic malware starts to propagate at the moment t.

In strategic sampling, if the agents T-dominate the network, it is highly likely that one of the (T-dominating) agents will co-locate with an infected smartphone, and thus, be infected before $t+T$. Thereafter, the infection will be detected the next time the infected agent is checked. If the periodic check is scheduled at a cycle of T, the epidemic is highly likely to be detected before $t+2T$, which is controllable by the choice of T. Comparing to both constant monitoring and random sampling, strategic sampling through the T-dominating agents provides control over the trade-off between cost/intrusiveness, in terms of the scale and frequency of the sampling, and the security provision, in terms of the maximal detection delay.

In prioritized patching, when a piece of smartphone malware is detected or a system vulnerability is uncovered, patches for preventing further exploitation can be first issued to the agents at the moment t. The agents will then become immune to this particular threat and will, therefore, slow down the malware's epidemic propagation. Furthermore, the agents can distribute the patches to their co-located smartphones. T-dominance ensures that most BYOD smartphones will receive the patches by $t+T$. Like in strategic sampling, prioritized patching through T-dominating agents provides control over the trade-off between cost/intrusiveness, in terms of the scale and frequency of the initial patching, and the security provision, in terms of maximal patching delay.

C. T-dominance: the algorithm

Now we have seen that the T-dominating agents serve a specific role in prioritized defense deployment. In this section, we present a local algorithm that runs on individual smartphones to elect agents without central coordination. The algorithm consists of two decision processes: activation and deactivation. We present deactivation before activation, because activation contains deactivation as a sub-process.

1) Agents vs. non-agents: Agents and non-agents differ in the amount of auxiliary information they maintain: An agent keeps track of other smartphones it shares co-location opportunities with, while a non-agent does not. The auxiliary information helps smartphones make informed activation/deactivation decisions without central coordination; the differentiation in the amount of maintained auxiliary information reduces prioritized security deployment's overhead for those non-agent smartphones. The auxiliary information maintained by the agents includes co-located smartphones' IDs, agent/non-agent status, and connectivity logs; each record is time-stamped for later consolidation.

When two smartphones are co-located (or, meet) and at least one of them is an agent, the agent will collect information from the other smartphone. When an agent u meets another smartphone v, there are two scenarios.

- When agent u meets non-agent v, since a non-agent only maintains its own auxiliary information, u can only obtain v's own information from v. After the meeting, u's auxiliary information expands to include v.
- When agent u meets another agent v, they share information on other smartphones they directly met with. After the meeting, u's auxiliary information expands to include v and v's direct acquaintance, as illustrated in Figure 2.

In both cases, agent u forms a filtered reachability graph $G^T(P)$ from all the phones P within its expanded scope, and takes the largest connected component containing itself, $G_D(u)$, as its *domination graph*. Later operations will be conducted on this domination graph $G_D(u)$.

2) Deactivation: Each agent first collects at least a time window's intelligence before it is eligible for deactivation. When an agent u meets another agent v, and *only after u has been an agent for at least a temporal window W's time*[3], u makes a decision of whether it will deactivate itself: A deactivated agent changes into a non-agent. Deactivation reduces the number

[3] The intuitive explanation for this is to let the agent be well informed before making a decision. We provide a technical justification in Section IV.

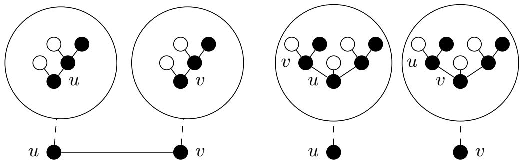

Fig. 2: After exchanging auxiliary information during their encounter, agent u's scope expands to include another agent v's direct acquaintance and vice versa.

of agents and, hence, the overall sampling/patching cost and intrusiveness of the deployed defense mechanism.

u makes its deactivation decision based on its domination graph $G_D(u)$. Let $N(w)$ be the neighbors of a vertex w in $G_D(u)$ and $N[w] = N(w) \cup \{w\}$ be the closed neighbor set of w. Depending on the security context/corporate policy, u may choose to be either aggressive or conservative in deactivating itself. Accordingly, we propose two alternative rules that u can follow to decide whether to deactivate itself:

- **Individual.** u deactivates itself if there exists an agent w with higher priority in $G_D(u)$ so that $N[u] \subseteq N[w]$.
- **Group.** u deactivates itself if there exists a connected set of agents U in $G_D(u)$, *each* of which has a higher priority than u, so that $N[u] \subseteq \bigcup_{w \in U} N[w]$. Such a U is said to be a replacement of u.

The definition of the two rules implies that agents under the Group rule are more aggressive in deactivation that those under the Individual rule. The two rules provide a trade-off between cost (in terms of number of agents) and responsiveness (in terms of delay between malware infection and detection in strategic sampling, or between patch release and distribution in prioritized patching) in a prioritized defense deployment scheme. The requirement for connectedness in Group is to enlist the bridging nodes in the BYOD smartphone network (such as an inter-departmental courier on a large corporate site) for security defense due to their critical role in connecting the network and, hence, higher chances of being attacked.

To complete the previous rules, u needs to decide whether w has a higher priority than itself. Again, u can be either aggressive or conservative: There are two alternative criteria that u can apply to decide whether w has a higher priority than itself (let $N_\cap = N(u) \cap N(w)$):

- **Strong.** w has a priority higher than u if 1) $N_\cap \neq \emptyset$; 2) $\exists x \in N_\cap, \mathrm{r}(x, w) < \mathrm{r}(x, u)$; 3) $\forall x \in N_\cap, \mathrm{r}(x, w) \leq \mathrm{r}(x, u)$.
- **Weak.** w has higher priority than u if 1) $N_\cap \neq \emptyset$; 2) $\sum_{x \in N_\cap} \mathrm{r}(x, w) < \sum_{x \in N_\cap} \mathrm{r}(x, u)$.

By definition, if an agent decides that one of its peers has a higher priority than itself under the Strong rule, it will reach the same conclusion under the Weak rule. Similar to Individual and Group, Strong and Weak provide a trade-off between cost and responsiveness in a prioritized defense deployment scheme. The absence of equivalence in the second clauses in both criteria is to eliminate the case that a pair of agents (wrongfully) assume that the other party will take over the responsibility for their dominated nodes and, hence, deactivate themselves during the same encounter.

3) Activation: When an agent u meets a non-agent v, u makes the decision of whether it should activate v.

One possible strategy is to activate every co-located non-agent. Given enough contact opportunities, such a strategy leads to an epidemic activation: Every smartphone gets activated at least once, no matter whether it is representative. However, since some of the agents are to be deactivated later, a more discreet strategy is desirable to avoid thrashing, i.e., employees' smartphones get repeatedly activated and deactivated in cycle, which consumes computational and energy resources on the smartphones without much security benefits.

The insight is that a non-agent should be activated *unless it is highly likely to be deactivated later*. Thus, the activation decision process comes down to measuring the likelihood of the non-agent being deactivated *later* if it is activated *now*.

Let us consider how an agent u can decide whether to activate a non-agent v. The activation process consists of two consecutive stages, *deactiviablity* and *coverage*.

Deactiviability. u computes a filtered reachability graph on its scope *along with that of* v*,* and invokes the deactivation strategy *for* v on the graph (in other words, u assumes v's perspective and decides, if v is to make the deactivation decision, whether v will deactivate itself). We say v is *deactivable* if the result (computed by Agent u) turns out to be that v will deactivate itself.

If v is *not* deactivable, u will activate v and terminate the activation decision process.

Otherwise, if u is deactivable, Agent v will proceed to the next stage.

Coverage. Let $A(u)$ be the set of *agents* that u knows of (including u itself). Agent u estimates v's *unique* coverage contribution to $A(u)$ and activates v with a corresponding probability.

The unique coverage contribution of v to $A(u)$ is those periods of time (within the temporal window) that none of the agents in $A(u)$ covers, but only v does.

Let the total length of v's *unique* coverage be $c(v \backslash A(u))$, and let the total length of $A(u)$'s coverage be $c(A(u))$. u activates v with a probability:

$$1 - \exp(-\frac{c(v \backslash A(u))}{c(A(u))}). \tag{3}$$

Thus, the probability is close to 0 if v contributes little unique coverage ($c(v \backslash A(u)) \to 0$) and is close to 1 if v contributes significant unique coverage ($\frac{c(v \backslash A(u))}{c(A(u))} \to \infty$). In other words, the more unique coverage v contributes, the more likely it will be activated by u. The unique temporal coverage contribution of the newly activated agent may help expose malware infection in strategic sampling or deliver patches in prioritized patching.

4) T-dominance-based prioritized defense deployment: By the activation and deactivation processes, a subset of the whole BYOD smartphone pool are elected as agents, and the rest are non-agents. Since the enterprise has much less central control over employees' BYOD devices than traditionally enterprise-issued ones, the T-dominating agent set allows security measures to be prioritized for those security-wise representative devices in order to reduce security mechanisms' cost/intrusiveness under a quantified security provision. For example, during each round of strategic sampling, an agent will have a higher probability of being sampled than a non-agent; similarly, in prioritized patching, when a new vulnerability is found, an agent will have a higher priority of being patched early than a non-agent. Thus, prioritized defense deployment can be formalized as follows.

Prioritized defense deployment. In deploying a repeatedly executed/upgraded defense mechanism, let the *priority* of, or the *probability* of deploying a security mechanism in *one round* on, the agents and non-agents be p and q respectively. *Prioritized defense deployment* is to have $p > q$.

A relatively small q reduces security overheads for those devices that have less contacts with others and, therefore, are less likely to spread or be infected with malware, while at the same time not completely neglecting the security of these relatively reclusive devices.

As a special case, $p = 1$ and $q = 0$: All the agents, and only the agents, are sampled in each round in strategic sampling, or patched (directly by corporate IT security staff) against each new vulnerability in prioritized patching. Suppose there is a bounded maximal rate of sampling or patching (due to technological, economic, or organization-political restriction), T-dominance-based prioritized defense deployment provides an ordering that favors more security-wise representative devices in the sampling/patching request queue. In Section V-B2, we simulate this scenario over real Wi-Fi association traces.

IV. Analysis

We identify a few desirable properties for an algorithm that implements T-dominance-based prioritized defense deployment and shows that the algorithm presented in Section III satisfies them.

A desirable algorithm should maintain the T-dominance structural property on a BYOD smartphone network, or, in other words, be correct: The effectiveness of strategic sampling and prioritized patching is contingent on the premise that the delay (as estimated by the reachability metric) to reach most smartphones from the agents through co-location is bounded by T.

Property 1 (Correctness)**.** *The T-dominance structural property is maintained by the algorithm.*

If an algorithm that implements T-dominance-based prioritized defense deploy requires employees to forfeit their co-location records for centralized planning, the algorithm will still be costly for the enterprise (due to the collection and central planning) and intrusive to the employees (due to the forfeiture). A distinction of a BYOD enterprise network, in comparison with a traditional enterprise-issued device network, is that it is more costly for the enterprise to provide security support the diverse set of devices and that the employees are more reluctant to intrusive security measures initiated by the enterprise (since, by definition, they belong to the employees). A key idea of prioritized security deployment is the observation that what matters to the enterprise is that which BYOD smartphones are representative security-wise (so that they will be prioritized for security mechanism deployment), rather than the detailed co-location information; employees may prefer not going through the chore of periodically updating with the enterprise about their whereabouts, but only sharing the information locally with co-located smartphones when needed. Thus, a desirable algorithm should be localized.

Property 2 (Localization)**.** *An agent makes its activation/deactivation decisions based on its own status and the connectivity logs from other smartphones it co-locates with.*

While localization (Property 2) decentralizes information collection process among opportunistically co-located smartphones, the information collected by agents about its past co-located neighbors may be outdated, and the reachability computed from such information may be different from the actual one at that moment. Requiring employees to constantly or on-demandly update such information with their neighbors induces great overheads and, therefore, negates the benefits of decentralization. Thus, a desirable algorithm should be able to handle outdated information while electing agents for prioritized defense deployment.

Property 3 (Temporal robustness)**.** *Property 1 is achieved even if the connectivity logs obtained from other smartphones during Wi-Fi co-location is outdated.*

In the rest of the section, we will show that the algorithm presented in Section III satisfies the Properties 1–3. Because only Deactivation (Section III-C2) may violate the properties, and Group-Weak is the most aggressive deactivation rule, we prove, in Theorem 1, that all three properties are satisfied by the design under the Group-Weak rule; the cases for other less aggressive deactivation rules are corollaries to Theorem 1. In addition, we complement our analysis here with simulations on real AP-association traces in Section V.

Theorem 1. *If an agent a deactivates itself in its local (and potentially outdated) view at the moment t, then, in the global (and updated) view, each of the smartphones T-dominated by a, including a itself, is still T-dominated by some agent at t.*

We break the proof of Theorem 1 down to a series of lemmas. Before proceeding, we need to make some extension to the notation to be more precise. The reachability metric, as defined in Equation (2) for two smartphones u and v, are actually defined on snapshots of u and v connectivity logs l_u and l_v, respectively. Therefore, we make this explicit by writing $\mathrm{r}(l_u, l_v)$ in place of $\mathrm{r}(u, v)$.

Lemma 1 is a property of the reachability metric defined in Equation (2).

Lemma 1. *Let l_u and l'_u (l_v and l'_v) be two snapshots of the connectivity log of smartphone u (v). If the common intervals of l'_u and l'_v are all contained in those of l_u and l_v in the temporal window $[t - W, t]$, then:*

$$r(l_u, l_v) \leq r(l'_u, l'_v).$$

Proof. In the same notation in Equation (2), let the common intervals of l_u and l_v within the window $[t - W, t]$ be $[s_1, e_1], \ldots, [s_k, e_k]$; $s_{k+1} = s_1 + w$. By Equation (2), $\mathrm{r}(l_u, l_v) = \sum_{i=1}^{k} (s_{i+1} - e_i)^2 \Big/ 2W$.

Since the common intervals of l'_u and l'_v are all contained in the common intervals of l_u and l_v in the temporal window $[t - W, t]$, the common intervals of l'_u and l'_v within the window $[t - W, t]$ can be represented as $[s_i, e_i], [s_{i+1}, e_{i+1}], \ldots, [s_j, e_j]$ for some $1 \leq i \leq j \leq k$. By Equation (2), $\mathrm{r}(l'_u, l'_v) = \sum_{n=i}^{j} (s'_{n+1} - e'_n)^2 \Big/ 2W$.

We have:

$$\mathrm{r}(l'_u, l'_v) - \mathrm{r}(l_u, l_v) = \Big[(s_i + W - e_j)^2 - \sum_{n=1}^{i-1}(s_{n+1} - e_n)^2 - \sum_{n=j}^{k}(s_{n+1} - e_n)^2\Big] \Big/ 2W. \tag{4}$$

Since $s_1 < e_1 < \ldots < s_i < e_i < \ldots < s_j < e_j < \ldots < s_k < e_k < s_{k+1} = s_1 + W$,

$$\begin{aligned} &\sum_{n=1}^{i-1}(s_{n+1} - e_n)^2 + \sum_{n=j}^{k}(s_{n+1} - e_n)^2 \\ \le &\left[\sum_{n=1}^{i-1}(s_{n+1} - e_n)\right]^2 + \left[\sum_{n=j}^{k}(s_{n+1} - e_n)\right]^2 \\ \le &(s_i - e_1)^2 + (s_{k+1} - e_j)^2 \\ \le &(s_i - e_1 + s_{k+1} - e_j)^2 = (s_i - e_1 + W + s_1 - e_j)^2 \\ \le &(s_i - s_1 + W + s_1 - e_j)^2 = (s_i + W - e_j)^2. \end{aligned} \tag{5}$$

By Equations (4) and (5), $\mathrm{r}(l_u, l_v) \le \mathrm{r}(l'_u, l'_v)$. □

In the following discussion, we use $l^t_{u(v)}$ to denote the snapshot of smartphone v's connectivity log stored on an agent u (only agents store other smartphones' connectivity logs) at time t, or in other words, u's local view of v at t. By definition, $l^t_{u(u)}$ is u's latest connectivity log at t, which is exactly u's connectivity log at t from the global view; therefore, we write $l^t_{u(u)}$ simply as l^t_u. We use $l^t_{u(u)}$ and l^t_u in different contexts to emphasize the different perspectives: The former is from u's local view, and the latter is from the global view.

Lemma 2 shows that, after collecting a window's intelligence, an agent's local view on the set of smartphones that is T-dominated by it agrees with the global view. This is the technical justification for requiring an agent to collect a window's intelligence before deactivating itself.

Lemma 2. *Suppose a is an agent during $[t - W, t]$. For each smartphone u with $r(l^t_a, l^t_u) < +\infty$, we have*

$$r(l^t_{a(a)}, l^t_{a(u)}) = r(l^t_a, l^t_u).$$

Proof. Since $\mathrm{r}(l^t_a, l^t_u) < +\infty$, by Definition (2), a has met u at least once during $[t - W, t]$; since a is an agent during $[t - W, t]$, $l^t_a = l^t_{a(a)}$ includes a record on the last meeting between a and u. Thus, the common intervals of $l^t_{a(a)}$ and $l^t_{a(u)}$ are exactly the same with those of l^t_a and l^t_u in the temporal window $[t - W, t]$. By Lemma 1, $\mathrm{r}(l^t_{a(a)}, l^t_{a(u)}) \le \mathrm{r}(l^t_a, l^t_u)$ and $\mathrm{r}(l^t_{a(a)}, l^t_{a(u)}) \ge \mathrm{r}(l^t_a, l^t_u)$. Hence $\mathrm{r}(l^t_{a(a)}, l^t_{a(u)}) = \mathrm{r}(l^t_a, l^t_u)$. □

Proof of Theorem 1. a deactivates itself at t if a is an agent during $[t - W, t]$ and finds, in its local view, a group of agents A with higher priorities, so that each smartphone T-dominated by a (including a itself) is T-dominated by at least one agent from A.

By Lemma 2, a's local view on the set of smartphones T-dominated by itself agrees with the global view. Hence, we only need to show that a non-agent u, which is T-dominated by both a and another agent $v \in A$ at t in a's local view, is actually T-dominated by some agent at t in the global view.

The proof is concluded if a's local view on v agrees with the global view. However, two possible discrepancies between a's local view and the global view demands further discussion: connectivity log and agent status of v.

The first case is straightforward to resolve. Suppose v is still an agent at t in the global view. Since $l^t_{a(u)}$ and $l_{a(v)}$ are past snapshots of l^t_u and l^t_v respectively, the common intervals of $l^t_{a(u)}$ and $l_{a(v)}$ are all contained in l^t_u and l^t_v in the temporal window; by Lemma 1, $\mathrm{r}(l^t_u, l^t_v) \le \mathrm{r}(l^t_{a(u)}, l^t_{a(v)})$. Since u is T-dominated by v in a's local view, we have $\mathrm{r}(l^t_{a(u)}, l^t_{a(v)}) \le T$. Thus, $\mathrm{r}(l^t_u, l^t_v) \le T$: u is T-dominated by the agent v at the moment t in the global view.

The latter case is more involved. Suppose v is no longer an agent at t in the global view. v must have deactivated itself and delegated the dominance of u to a replacement w at an earlier time $t' < t$ *after the last encounter between a and v* (w must be an agent at the moment t' for this to happen). Thus, $l^t_{a(v)}$ is a past snapshot of $l^{t'}_{v(v)}$. Since $l^{t'}_{v(u)}$ contains all the encounters between u and v up to the moment t', the common intervals of $l^t_{a(u)}$ and $l^t_{a(v)}$ are all contained in those of $l^{t'}_{v(u)}$ and $l^{t'}_{v(v)}$, thus $\mathrm{r}(l^{t'}_{v(u)}, l^{t'}_{v(v)}) \le \mathrm{r}(l^t_{a(u)}, l^t_{a(v)}) \le T$ by Lemma 1.

Since v deactivated itself at t' for w, $\mathrm{r}(l^{t'}_{v(u)}, l^{t'}_{v(w)}) \le \mathrm{r}(l^{t'}_{v(u)}, l^{t'}_{v(v)}) \le T$. Since $l^{t'}_{v(u)}$ and $l^{t'}_{v(w)}$ are both past snapshots of l^t_u and l^t_w, the common intervals of $l^{t'}_{v(u)}$ and $l^{t'}_{v(w)}$ are contained in those of l^t_u and l^t_w, thus $\mathrm{r}(l^t_u, l^t_w) \le \mathrm{r}(l^{t'}_{v(u)}, l^{t'}_{v(w)}) \le T$ by Lemma 1. That is to say, even though v may be deactivated at t, u is still T-dominated by the agent w delegated by v.

Thus, by the same argument on v's replacement w at t', even if v has deactivated itself by t, either the replacement w actually T-dominates u at t, or it has further delegated u to other agents at an earlier time. By tracing back this chain of delegation, we can eventually find, in the global view, an agent that T-dominates u at t.

We now show that there is no loop in the chain of delegation. Along with the fact $T \ge \mathrm{r}(l^t_{a(a)}, l^t_{a(u)}) \ge \mathrm{r}(l^t_{a(u)}, l^t_{a(v)}) \ge \mathrm{r}(l^{t'}_{v(u)}, l^{t'}_{v(v)}) \ge \mathrm{r}(l^{t'}_{v(u)}, l^{t'}_{v(w)}) \ge \ldots$ we have just proved, the non-equality requirement in the priority comparison rule ensures that there is no loop in the chain of delegation. □

V. Verification

We complement our analysis on T-dominance with simulations driven by real-world collected datasets.

A. Dataset and methodology

The dataset is from the Wireless Topology Discovery (WTD) project [6]. The dataset consists of traces collected from over 150 UC San Diego freshmen using hand-held mobile devices for an 11-week period. Periodic Wi-Fi AP scanning and association results were recorded every 20 seconds. The students participating in this experiment, though coming from different majors, resided in the same university housing facility. This setting resembles the arrangement in a large enterprise site, with employees working in their designated office spaces (corresponding to students' dormitories). The traces capture the mobility and connectivity patterns of a group of users in a relatively short period of time [6].

Given the frequency of data recording (once every 20 seconds), we transformed the periodic records into a series of sessions (a session is defined as a device associating with an AP during a period of time) by the following method: Consecutive records of the same device associating with the same AP within 20 seconds were combined to form a single session.

The transformed traces were then fed into an event-driven simulator implemented in Perl. Each session in the transformed traces triggers two events along the time line: an association event and a de-association event. We took the first 200 thousands entries in the records and used the first 30% of the data for the 190 nodes to accumulate connectivity logs, which allowed them to simulate the agent election process later. Then, some nodes were randomly selected as initial agents; the agents made activation/deactivation decisions, based on the algorithm specified in Section III. In the following scenarios, the simulation process was repeated with different psuedo-random number generator (PRNG) seeds to obtain the means and quartiles.

B. Scenario and results

1) T-dominating agent election: We simulated the agent election process under the different T-dominating strategies (Group-Strong, Group-Weak, Individual-Strong, and Individual-Weak) with different numbers of initial agents and T. Since for a given set of initial agents, no proximity-based activation strategy can activate more agents than the epidemic one (the epidemic strategy is one in which agents unconditionally activate their co-located neighbors), we normalized the results with the epidemic strategy to make them comparable: At a particular moment, the number of agents elected by a T-dominance strategy is divided by the number of agents activated by the epidemic strategy, to obtain the normalized agent set size. We computed the means of the results from multiple rounds of simulation, with different PRNG seeds, to account for the potential bias introduced by a peculiar initial setting. Figure 3 shows a representative result with 5, 10, and 15 initial agents (out of the 190 nodes) with $T = 18,000s$ (5 hours). The following are a few notes on Figure 3:

- In terms of agent set reduction through self-deactivation, Group-Weak is the most aggressive strategy, and Individual-Strong is the most conservative one, while the other two come in between, and are comparable. This confirms our design intuition in Section III.
- The size of the initial agent set has little influence on the size of the agent set eventually elected. The small differences are mostly at the beginning of the process and are difficult to notice without zooming in. One explanation is that this dataset, like in many closed-world networks such as in an enterprise, is well connected: Except for maybe a few peculiar cases, an agent election process originating from a small set of agents spreads to the whole network quickly; the activation/deactivation process since that moment will then converge.
- An agent election process election process consists of two consecutive phases. The first phase (0 to around 5000 seconds in Figure 3) corresponds to the general trend of decreases (with occasional small increases) in normalized agent set size (NASS), and reflects the overwhelming effect of deactivation in search of the T-dominating agent set. The second phase (after 5000 seconds in Figure 3) is characterized by the dynamic balance between activation and deactivation when the T-dominanting agent set has been activated.

2) Prioritized defense deployment: We simulated prioritized defense deployment based on the T-dominance-elected agents. We consider the following scenario. Following the election of T-dominating agents as in Figure 3, the elected agents were periodically checked for malware infection; once an infection is detected, the infected agent would enroll itself, along with the non-agents T-dominated by it, in a malware patching pool. For simplicity, we considered the case in which there was no delay in detecting agent infection: The agents were constantly monitored for malware infection.

Independently, a smartphone was randomly selected from the patching pool at the rate of once every ten seconds and, if the smartphone was indeed infected, it would be patched. Patched smartphones would then become immune to malware infection.

We compared the T-dominance-based prioritized defense deployment, instantiated by this strategic sampling/patching (strategic s/p) strategy, with a random sampling/patching (random s/p) strategy. The latter periodically selected a smartphone randomly for malware infection checking, at the same rate as in the prioritized defense deployment (i.e., once every ten seconds). If the selected smartphone was indeed infected, it would be patched immediately.

We considered both epidemic and static malware models, which correspond to proximity malware attacks and drive-by download attacks, respectively. We assumed that an agent could detect malware infection in co-located smartphones, and, if malware infection was detected, would enroll its T-dominating smartphones in the malware patching pool.

Boxplots of the results[4] are shown in Figures 4 and 5 for different numbers of initial agents (corresponds to the value in Figure 3) and initial malware-infected smartphones.

Figure 4 shows the delay between the initial malware outbreak and the first patching of a malware-infected smartphone. A few notes on Figure 4:

- Strategic s/p has a shorter delay than random s/p. In other words, the former is more responsive to malware infection than the latter. This justifies the adoption of T-dominance for prioritized defense deployment: By having an agent set that T-dominants the whole smartphone pool to serve as sampling points, malware outbreaks will be detected more promptly.
- The delay under the static malware model with small numbers of initial malware-infected smartphones is relatively long; the delay under the epidemic malware model with large numbers of initial malware-infected smartphones is relatively short. The explanation is that more smartphones will be infected by the malware shortly in the latter case, so initial sampling and ensuing patching will take less time than the former case.

Figure 5 shows the number of malware-infected smartphones averaged through the whole infection process (from the malware outbreak to the moment that all malware-infected smartphones were patched). A few notes on Figure 5:

- Under the epidemic malware model, T-dominance-based strategic s/p has significant less average malware infections

[4] Boxplots [7] show the max/min, 75%/25% quartiles, and the median of a group of observations.

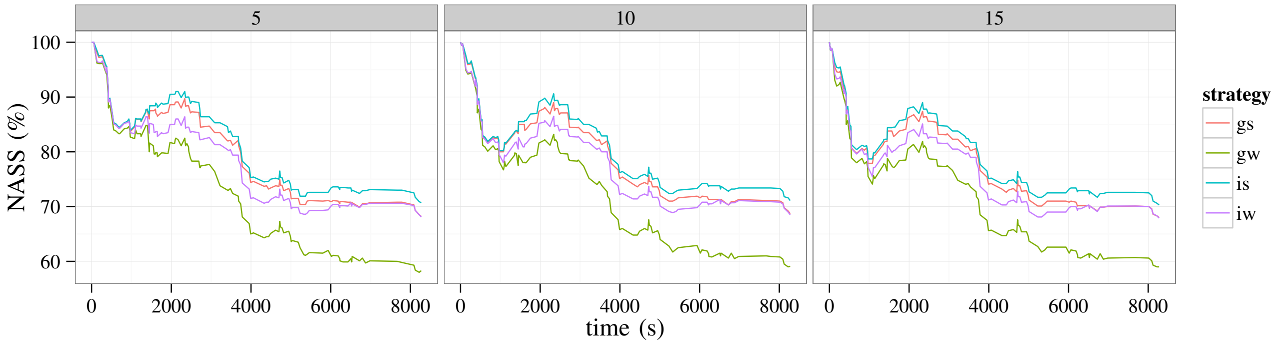

Fig. 3: A representative T-dominating agent election process with 5, 10, and 15 initial agents (out of the 190 nodes) and $T = 18,000s$ (5 hours). Agent set size is normalized by epidemic activation strategy: the y-axis is shown in normalized agent set size (NASS). Strategy notations: gs (Group-Strong), gw (Group-Weak), is (Indivdual-Strong), iw (Individual Weak).

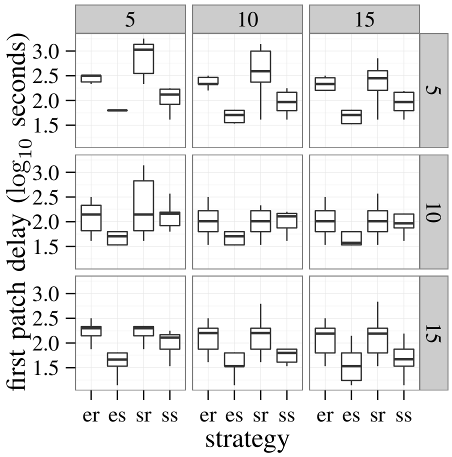

Fig. 4: Delay from the malware breakout to the first patching of a malware-infected smartphone. The patching rate is once per ten seconds. The row heading shows initial agent number *before* malware election; the column heading shows the number of malware-infected smartphone at the malware breakout. Strategy notation: er (epidemic malware, random sampling/patching), es (epidemic malware, strategic sampling/patching), sr (static malware, random sampling/patching), ss (static malware, strategic sampling/patching). The y-axis is shown in a $\log_{10}$ scale.

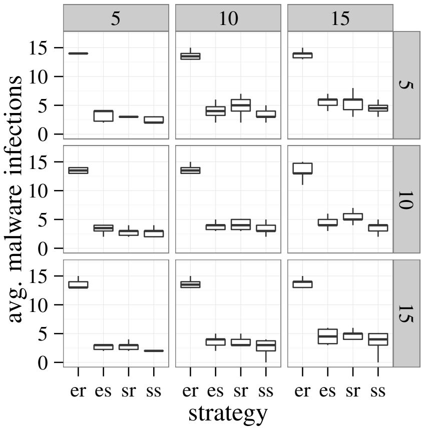

Fig. 5: Average malware number. The notations are the same as in Figure 4.

than that of random s/p: A typical number of average infections is 3 for strategic s/p and 13 for random s/p. The difference is even more pronounced when there are more initial malware infections as in the upper rightest column with 15 initial infections.

- Even under the static malware model, where the malware would not propagate from infected smartphones to others and, hence, the average number of malware-infected smartphones over time shall be less than the initial number, the strategic s/p based on the T-dominating agent set has 1 to 3 less average infections than the random s/p.

Results shown in Figures 4 and 5 collectively show the responsiveness and effectiveness of T-dominance-based prioritized defense deployment, as instantiated by the strategic s/p, in detecting and mitigating BYOD smartphone malware.

VI. Extended discussion

Currently, we are not aware of any real-world report of smartphone malware propagating through Wi-Fi co-location. However, this does not mean that the attack model is purely hypothetical or impractical. For example, a report [8] on hijacking hotel Wi-Fi hotspots for drive-by malware attacks on laptops comes close to what we have in mind; practical man-in-the-middle attacks against Wi-Fi co-located devices was demonstrated in a recent BlackHat security conference [9]. We note that enabling environments and techniques for Wi-Fi co-location-based smartphone malware are already in place.

- Given the complexity of the commercial smartphone software platforms and the diversity of security awareness and experience of application developers, it is reasonable to believe that remote exploitable vulnerabilities will be discovered and exploited.
- The privilege authorization frameworks on smartphone platforms, which supposedly prevent the malware from obtaining unwarranted privilege, are often ignored for convenience, or circumvented for customization by the users. Rootkits, like iOS Jailbreak[5], are routinely used by users for installing third-party applications, whose trustworthiness is often assumed, but not verified.
- Commercially-available Wi-Fi honeypots like Wi-Fi Pineapple[6] enable DNS spoofing, ARP poisoning, and man-in-the-middle attacks.
- The concentration of mobile application development on the two major smartphone platforms (iOS and Android) greatly

[5] http://www.jailbreakme.com/
[6] http://hakshop.myshopify.com/products/wifi-pineapple

reduces device heterogeneity, and thus, makes malware epidemics possible.

Given these considerations, Wi-Fi-co-location-based smartphone malware is likely to emerge in the near future; even worse, such malware may have already been deployed in the real world. This makes the study in mitigating it for a comprehensive BYOD network security model relevant and worthwhile.

VII. Related works

Although BYOD features numerous recent IT industry analyses and news reports as one prominent enterprise IT trend in the coming years [1, 2, 3], academic studies on the security implications of BYOD are scarce and still at an early stage [10, 11, 12]. One explanation is that while it is agreed that BYOD brings many benefits as well as management/security challenges, approaches to modeling and resolving the challenges are still being explored. In this paper, we identify the tension between security provision and employee intrusiveness/security mechanism deployment cost as one of the challenges for BYOD security and propose prioritized defense deployment as a solution.

Proximity malware has been studied previously in the context of sensor, ad hoc, P2P, or mobile networks, with a focus on either identifying the critical point for long-term malware survival/extinction under various epidemiological models [13, 14, 15], or extracting and exploiting mobility pattern and community structure for malware mitigation [16, 17, 18]. Studies on Android, one of the dominating smartphone software platforms, show that many mobile applications are vulnerable to attacks and malware on the Android smartphone software platform [9, 19, 20] and that malware is rampant [21].

The fascinating topic of capturing temporal dynamics in complex networks is studied by many previous works, often in the context of human mobility patterns captured by telecommunication service traces [22, 23, 24]. T-dominance is our attempt to capture the temporal dynamics in the BYOD enterprise environment. The exploitation of temporal dynamics for mitigating BYOD malware threat is novel.

The T-dominance algorithm is inspired by previous works on the Connected Dominating Set (CDS) problem of topology and routing in ad hoc and sensor networks [25, 26, 27]. However, the interpretation of CDS for temporal dynamics, the application in electing security-wise representative nodes in a BYOD network, and the issue of temporal robustness are all novel.

VIII. Conclusion

Evidence indicates that many enterprises have adopted or are considering adopting a BYOD IT policy. However, research on BYOD enterprise network security is still at an early stage; many issues are yet to be clearly identified. In this paper, in the context of smartphone malware attacks and widely deployed enterprise Wi-Fi infrastructures, the tension between security provision and intrusiveness/cost is identified as one such issue; prioritized defense deployment based on security representativeness is one approach to address the tension; prioritization by temporal-spatial structure through T-dominance is one interpretation of security representativeness. Other issues/approaches/interpretations are to be explored.

Independent from the application of T-dominance in prioritizing defense deployment, we briefly discuss the possibility of abusing T-dominance in making BYOD malware attacks stealthy. This shows, from another perspective, the importance of understanding the temporal-spatial structure for BYOD enterprise network security. For example, a study on the competition between a strategic sampling/prioritized patching scheme and an instance of stealthy malware, running the T-dominance algorithm with different T, would be interesting.

References

[1] J. Rains. (2012) Bring your own device (BYOD): hot or not? Show survey results on BYOD adoption. [Online]. Available: http://goo.gl/AV7mj

[2] E. Harrison. (2012) Five enterprise tech trends for 2013: BYOD, VPNs, AaaS, Big Data and Business Intelligence. [Online]. Available: http://goo.gl/T1JqO

[3] B. Bergstein. (2012) IBM faces the perils of "bring your own device". [Online]. Available: http://goo.gl/Dnlpe

[4] "Location histories for location aware devices," U.S. Patent 20 110 051 665, 2011.

[5] N. Provos, "A virtual honeypot framework," in *Proc. of USENIX Security*, 2004.

[6] M. McNett and G. Voelker, "Access and mobility of wireless PDA users," *ACM SIGMOBILE Mob. Comput. and Comm. Rev. (MCCR)*, vol. 9, no. 2, pp. 40–55, 2005.

[7] R. McGill, J. Tukey, and W. Larsen, "Variations of box plots," *The American Statistician*, vol. 32, no. 1, pp. 12–16, 1978.

[8] A. Lee. (2012) FBI warns: New malware threat targets travelers, infects via hotel Wi-Fi. [Online]. Available: http://goo.gl/D8vNU

[9] Trustwave Holdings, Inc. (2012) BeEF injection with MITM. [Online]. Available: http://goo.gl/P97Au

[10] K. Miller, J. Voas, and G. Hurlburt, "BYOD: Security and privacy considerations," *IT Professional*, vol. 14, no. 5, pp. 53–55, 2012.

[11] G. Thomson, "BYOD: enabling the chaos," *Network Security*, vol. 2012, no. 2, pp. 5–8, 2012.

[12] M. Potts, "The state of information security," *Network Security*, vol. 2012, no. 7, pp. 9–11, 2012.

[13] D. Chakrabarti, J. Leskovec, C. Faloutsos, S. Madden, C. Guestrin, and M. Faloutsos, "Information survival threshold in sensor and p2p networks," in *IEEE INFOCOM 2007. 26th IEEE International Conference on Computer Communications*, 2007, pp. 1316–1324.

[14] N. Valler, B. Prakash, H. Tong, M. Faloutsos, and C. Faloutsos, "Epidemic spread in mobile ad hoc networks: Determining the tipping point," in *Proc. of IFIP NETWORKING*, 2011.

[15] B. Prakash, D. Chakrabarti, M. Faloutsos, N. Valler, and C. Faloutsos, "Threshold conditions for arbitrary cascade models on arbitrary networks," in *Proc. of IEEE ICDM*, 2011.

[16] C. Gao and J. Liu, "Modeling and restraining mobile virus propagation," *IEEE Trans. Mob. Comput.*, vol. preprint, no. 99, p. 1, 2012.

[17] Y. Li, P. Hui, L. Su, D. Jin, and L. Zeng, "An optimal distributed malware defense system for mobile networks with heterogeneous devices," in *Proc. IEEE SECON*, 2011.

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

[19] S. Fahl, M. Harbach, T. Muders, M. Smith, L. Baumgärtner, and B. Freisleben, "Why eve and mallory love android: an analysis of android SSL (in) security," in *Proc. ACM CCS*, 2012.

[20] M. Grace, Y. Zhou, Z. Wang, and X. Jiang, "Systematic detection of capability leaks in stock android smartphones," in *Proc. ISOC NDSS*, 2012.

[21] Y. Zhou and X. Jiang, "Dissecting Android malware: Characterization and evolution," in *Proc. IEEE S&P*, 2012.

[22] S. Strogatz, "Exploring complex networks," *Nat.*, vol. 410, no. 6825, pp. 268–276, 2001.

[23] J. Tang, M. Musolesi, C. Mascolo, and V. Latora, "Characterising temporal distance and reachability in mobile and online social networks," *ACM SIGCOMM Comput. Comm. Rev. (CCR)*, vol. 40, no. 1, pp. 118–124, 2010.

[24] H. Kim and R. Anderson, "Temporal node centrality in complex networks," *Phys. Rev. E*, vol. 85, no. 2, p. 026107, 2012.

[25] J. Wu, F. Dai, and S. Yang, "Iterative local solutions for connected dominating set in ad hoc wireless networks," *IEEE Trans. Comput. (TC)*, vol. 57, pp. 702–715, 2008.

[26] S. Yang, J. Wu, and F. Dai, "Efficient directional network backbone construction in mobile ad hoc networks," *IEEE Trans. Parallel and Distrib. Syst. (TPDS)*, vol. 19, no. 12, pp. 1601–1613, 2008.

[27] K. Sakai, S. Huang, W. Ku, M. Sun, and X. Cheng, "Timer-based CDS construction in wireless ad hoc networks," *IEEE Trans. Mob. Comput. (TMC)*, vol. 10, no. 10, pp. 1388–1402, 2011.