T-dominance: Prioritized Defense Deployment for BYOD Security

IEEE CNS 2013

Wei Peng[1] Feng Li[1] Keesook J. Han[2] Xukai Zou[1] Jie Wu[3]

[1]Indiana University-Purdue University Indianapolis

[2]Air Force Research Laboratory

[3]Temple University

14 October 2013

bring your own device (BYOD)

- an enterprise IT policy rising with blackberry/smartphones. . .
- . . . that encourage employees to user their own devices to access the enterprise IT infrastructure at work
- some cited justifications
  - employees' demand/satisfaction
  - decreased IT acquisition and support cost,
  - increased use of virtualization
- security concerns
  - "bring your own virus"
  - inadvertenly or maliciously bring malware on a personal device to other devices. . .
  - . . . through the enterprise network behind firewalls

prioritized defense deployment

motivation

- BYOD devices need to be monitored and audited for malware protection. . .
- . . . but constantly doing so on all devices:
  - negates the perceived convenience
  - is costly to implement

idea

- observation: some device are more **security-wise representative**
- **prioritize** these devices for defense deployment

question

- How to define security-wise representative?
- How to find these users?

T-dominance

as a structural property on temporal-evolving topology

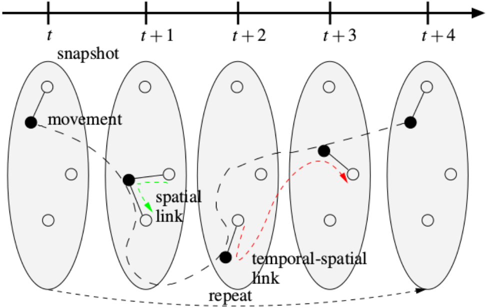

Fig. 1: T-dominance exploits temporal-spatial patterns of BYOD devices to implement prioritized defense deployment. The black node T-dominates the white ones for $T > 4$.

the black node is security-wise representative. . .

. . . because it **T-dominants the white nodes with $T = 4$**

T-dominance

as a distributed algorithm that constructs a T-dominating set

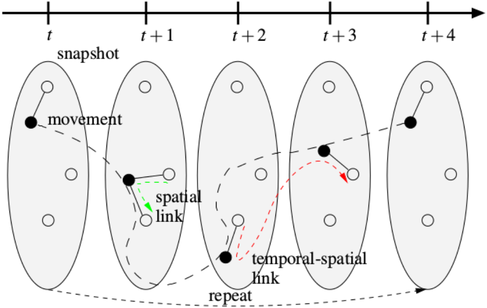

Fig. 1: T-dominance exploits temporal-spatial patterns of BYOD devices to implement prioritized defense deployment. The black node T-dominates the white ones for $T > 4$.

the T-dominating set election process is carried out by **individual** nodes. . .

. . . with knowledge of **local** (rather than global) neighborhood

T-dominance

as a prioritized defense deployment strategy

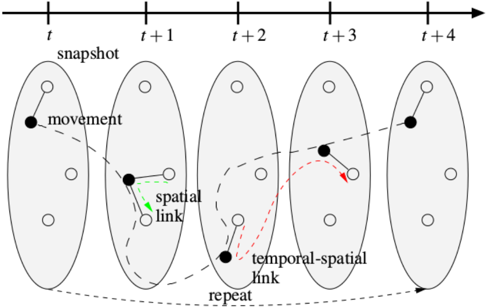

Fig. 1: T-dominance exploits temporal-spatial patterns of BYOD devices to implement prioritized defense deployment. The black node T-dominates the white ones for $T > 4$.

more stringent security mechanism deployed on the T-dominating set...

...provides a **quantified** (by T) security trade-off...

...between **deployment cost** and **detection delay**

T-dominance structural property

- given connectivity history[1], expected encounter delays (reachability) $r(u, v)$ between devices $u, v \in P = \{u, v, w, \ldots\}$ can be estimated ▸ details
- $G^T(P)$ (reachability graph filtered by T): undirected graph with P as vertices and $r(u, v)$ as weight on edge (u, v), and **all edges with weight greater than T removed**

Definition (T-dominance)

Let P be a set of devices and A be a subset of P called the agents. Agents A are said to T-dominate 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 $G^T(P)$.

- example: prioritizing a T-dominating set for deploying a security patch will have the patch reach all devices within a maximal delay of T with a high probability

[1] a built-in feature of many smartphones

T-dominance distributed algorithm

overview

info exchange upon encounters...

- agent keeps info on encountered devices; non-agent does not
- time-stamped info: device ID, agent/non-agent status, connectivity history
- info helps make the following activation/deactivation decisions
- u constructs its domination graph $G_D(u)$, based on exchanged info

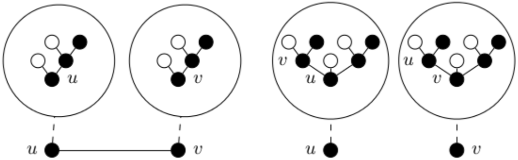

Fig. 2: After exchanging auxiliary information during their encounter, agent u's scope expands to include another agent v's direct acquaintance and vice versa.

...plus 2 circumstances

- agent meets agent: deactivation
- agent meets non-agent: activation

T-dominance distributed algorithm
deactivation

- when agent u meets another agent (after u has been an agent for at least a period of W), u decides whether to deactivate itself
- $N[w] = N(w) \cup \{w\}$: the closed neighborhood of $w \in G_D(u)$

2 alternative decision rules for u

- **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.

2 alternative priority comparisons

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

T-dominance distributed algorithm

activation

- when agent u meets non-agent v, u decides whether to activate v
- problem: indiscriminate activation wastes resources in thrashing
- solution: activate v **unless it is highly likely to be deactivated later**

2 consecutive stages

- **Deactiviability.** u pretends v is an agent, plays v's role in u's own perspective $G_D(u)$
  - if v is not to be deactivated, then u activates v
  - if v is to be deactivated, then u proceeds to the next stage.
- **Coverage.** u estimates v's *unique* coverage (in addition to the agent set $A(u)$ that u knows of) and activates v with a corresponding probability
  - $c(v \backslash A(u))$: v's unique coverage; $c(A(u))$: $A(u)$'s total coverage
  - u activates v with a probability:

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

T-dominance algorithm properties

3 properties

Property (Correctness)

The T-dominance structural property is maintained by the algorithm.

Property (Localization)

An agent makes its activation/deactivation decisions locally.

Property (**Temporal robustness**)

Correctness is achieved even if the info obtained from other devices is outdated.

T-dominance algorithm properties

the key to temporal robustness

Theorem

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 devices T-dominated by a, including a itself, is still T-dominated by some agent at t.

dataset

- from the Wireless Topology Discovery (WTD) project[2]
- collected from over 150 UC San Diego freshmen using hand-held mobile devices over an 11-week period
- periodic Wi-Fi AP scanning and association results were recorded every 20 seconds

preprocessing

- consecutive association records (every 20 seconds) are combined into a single session
- took the first 200 record entries
- use the first 30% of the data (with 190 nodes) to accumulate connectivity history
- some nodes are randomly selected as initial agents
- simulate the activation/deactivation processes

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

evaluation

agent election results

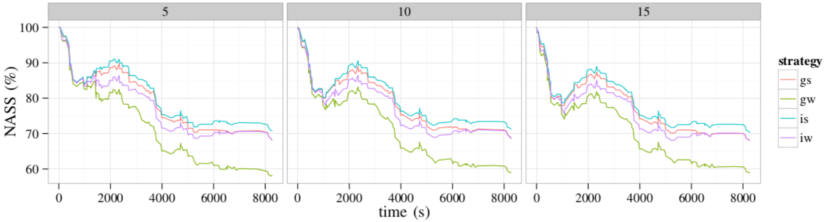

Fig. 3: A representative T-dominating agent election process with 5, 10, and 15 initial agents (out of the 190 nodes) and $T = 18,000s$ (5 hours). Agent set size is normalized by epidemic activation strategy: the y-axis is shown in normalized agent set size (NASS). Strategy notations: gs (Group-Strong),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

agent election is normalized by the epidemic activation strategy

evaluation

prioritized defense deployment effectiveness

compare at the same rate

- T-dominance-based strategic malware sampling/patching
- random sampling/patching

on different malware propagation model

- epidemic propagation
- static/no propagation

evaluation

prioritized defense deployment effectiveness

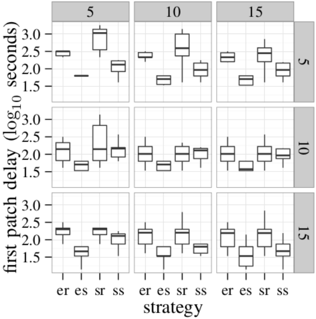

Fig. 4: Delay from the malware breakout to the first patching of a malware-infected smartphone. The patching rate is once per ten seconds. The row heading shows initial agent number *before* malware election; the column heading shows the number of malware-infected smartphone at the malware breakout. 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.

the delay till first detection

T-dominance strategic sampling can detect malware **faster** than random sampling

evaluation

prioritized defense deployment effectiveness

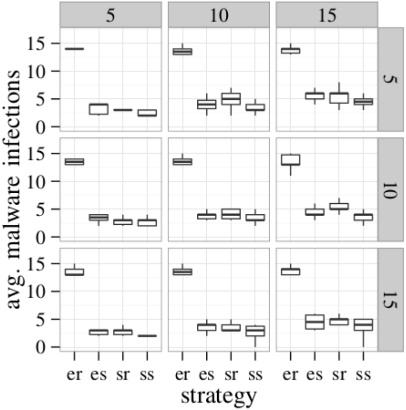

Fig. 5: Average malware number. The notations are the same as in Figure 4.

the number of malware infected nodes averaged over the whole time period

T-dominance strategic patching is **more effective in preventing malware epidemic** than random patching

take-aways

- prioritized defense deployment provides a less-intrusive BYOD security solution
- T-dominance provides a quantified trade-off between defense deployment cost and time-to-full-coverage
- the activation/deactivation distributed algorithm preserves the T-dominance structural property with temporal robustness
- T-dominance-based strategy sampling/patching is more effective than random sampling/patching

thank you

- connectivity log entry $(ST = s, ET = e, APID = AP_i)$: the device is associated with access point AP_i from time s to e
- given u and v's connectivity logs, find encounter durations in time window $[t - W, t]$ to be $[s_1, e_1], [s_2, e_2], \ldots, [s_k, e_k]$ (define $s_{k+1} = s_1 + W$)
- at time m, delay until the next encounter:

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

- reachability between u and v as expected delay:

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

back to T-dominance definition