A[1] Moving-target Defense Strategy for Cloud-based Services with Heterogeneous and Dynamic Attack Surfaces

Wei Peng[1] Feng Li[1] Chin-Tser Huang[2] Xukai Zou[1]

[1]Indiana University-Purdue University Indianapolis (IUPUI.edu)
[2]University of South Carolina (SC.edu)

[1]note: not "the" here; we will explain it further on the "caveat" of the "findings in brief" page

Staticity and Homogeneity in IaaS Clouds

deep automation leads to (largely) static and homogeneous IaaS Cloud service (e.g., relatively small number of choices of AWS OS instances)

$\Rightarrow$

decrease a potential attacker's uncertainty about the target

$\Rightarrow$

increased chance of the service being compromised

Moving-target Defense (MTD) to the Rescue

- from military: "a moving target is hard to hit"
- common attacker-defender relation
  - defenses are **reactive** to a preceding **proactive** attack attempt
  - the attacker has the upper hand
- goal
  - **decrease** the **utility** (for attacking) of **attackers' existing intelligence** on the **old** target...
  - ...and **increase attackers' uncertainty** on the **new** target

The Question

deploying MTD entails additional overheads

whether and **to what extend**
MTD is effective
in securing a Cloud-based service
with **heterogeneous** and **dynamic** attack surface

Findings in Brief[2]

- MTD is more effective when the service deployment is dense in the replacement pool and/or when the attack is strong
- attack-surface heterogeneity-and-dynamics awareness helps in improving MTD's effectiveness.

[2] caveats: the findings are based on a particular MTD strategy with a particular attack surface model; alternative models require separate studies. This work shows a way on how to do this.

Contributions

- formulate a Cloud-based service security model that incorporates Cloud-specific features such as VM migration/snapshotting and the diversity/compatibility of migration
- consider the accumulative effect of the attacker's intelligence on the target service's attack surface
- model the heterogeneity and dynamics of the service's attack surfaces, as defined by the (dynamic) probability of the service being compromised, as an S-shaped generalized logistic function
- propose a probabilistic MTD service deployment strategy that exploits the dynamics and heterogeneity of attack surfaces for protecting the service against attackers

The Model

Overview

a Cloud-based service whose deployment migrates among VM inst.

vs.

an **accumulative** attacker with limited attack budget

The Model

Service Model

- the service is deployed on several **active** virtual machine (VM) instances
- replacement VM instances are standing by . . .
  - . . . the service can be migrated from active inst. to compatible replacements
  - we assume attacker cannot differentiate active/in-active VM inst.
  - $\Rightarrow$ inactive VMs protect active ones by increasing attackers' uncertainty
- replacements are subject to compatibility requirements
  - e.g., some Windows and Linux services are not compatible
  - replacements are different (in configuration) and similar (in migration feasibility) to the active instance at the same time
  - $R(j)$: the compatible replacement set of VM instance j
- snapshot-and-restore service migration model. . .
  - . . . instead of a refreshing model
  - more realistic
  - $\Rightarrow$ attacker's advantage is preserved
  - more challenging for MTD

The Model

Attacker Model

- attacker can probe arbitrary VM instances but not knowing their active status
  - inactive VM instance may appear real through fake service response (e.g., honeypot)
- attacker is constrained by an attack budget
  - upper limit on the rate of probing
- attacker's intelligence on the target is **accumulative**
  - hit: probe an active inst.; miss: probe an inactive inst.
  - probability of attacker compromising inst.

The Model

Attack Surface Model

- **heterogeneous** and **dynamic** attack surface
- $P_{a,j}(t)$: Probability that the active VM inst. j be compromised by attacker a after t hits
- intuitively, $P_{a,j}(t)$ has an S-shape
  - begin with reconnaissance, characterized by low success probability, but with fast intelligence growth rate
  - end with a high success probability but saturated intelligence

The Model

Attack Surface Model

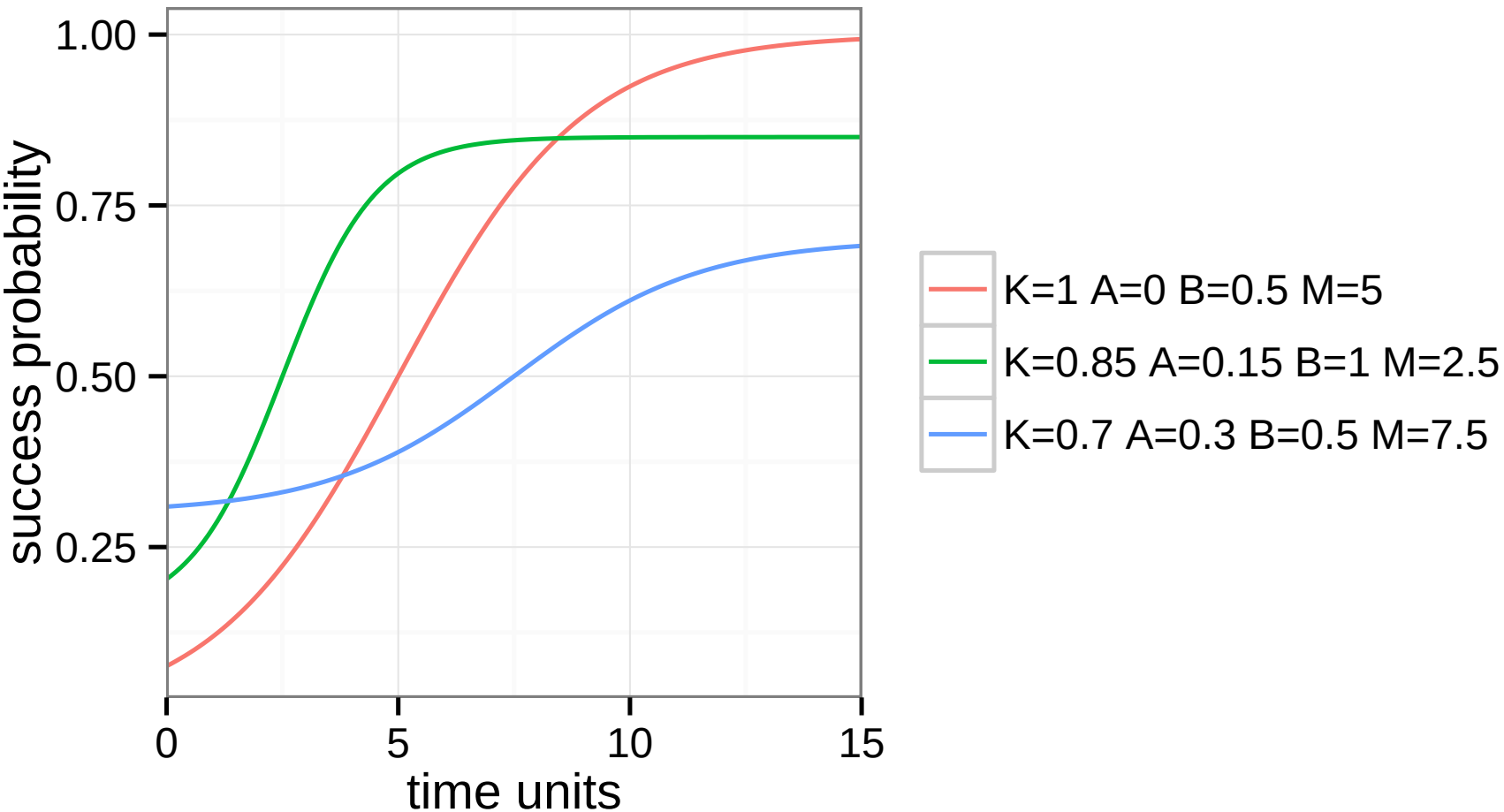
a generalized logistic function

$$P_{a,j}(t) = A_j + \frac{K_j - A_j}{1 + e^{-B_j(t-M_j)}}.$$

- A_j: lower asymptote
  - a's first hit has a success probability of (very close to) A_j
- K_j: upper asymptote
  - a's success probability is never over K_j
- B_j: the growth rate
  - B_j determines the growth in the attacker's success probability between subsequent hits
- M_j: the time of maximal growth
  - the period before M_j has an increasing growth rate, whereas the period after M_j has a decreasing growth rate

The Model

Attack Surface Model



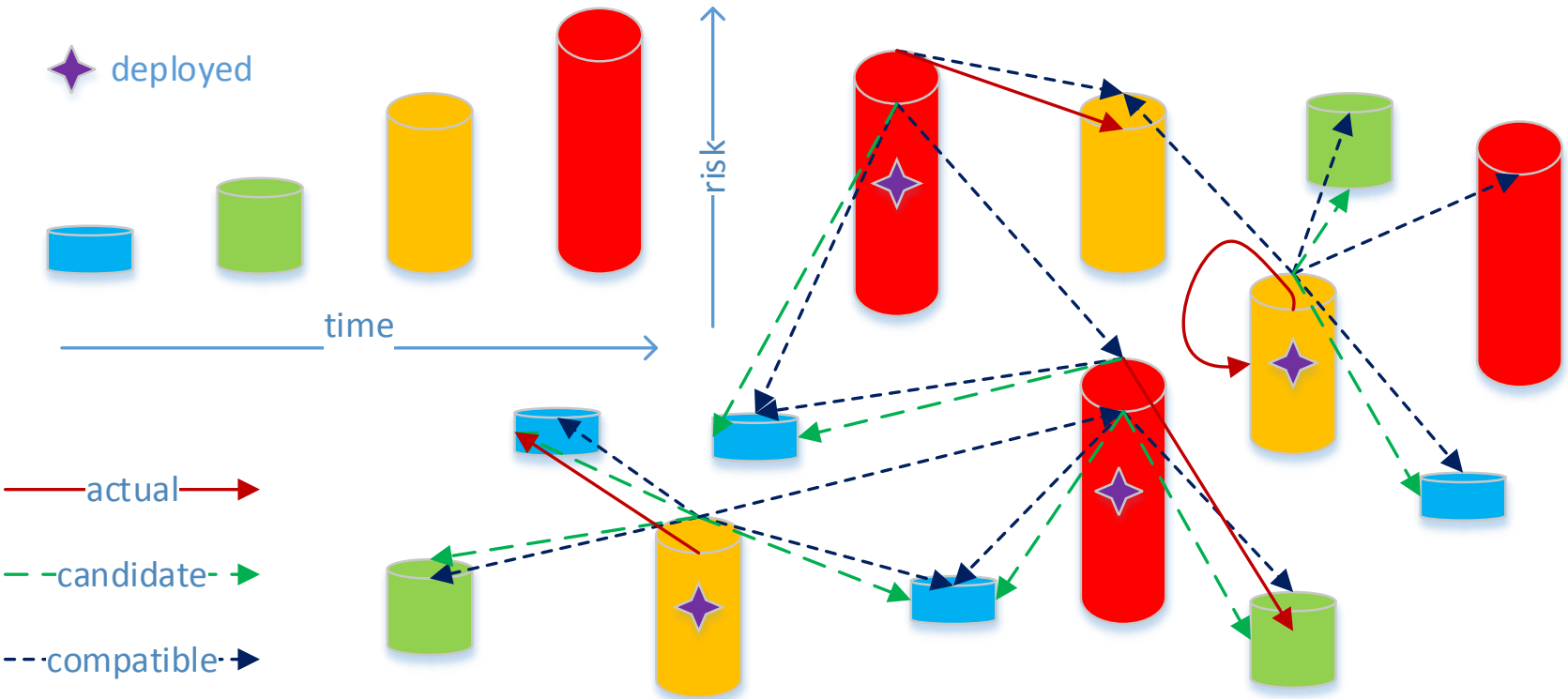
The Proposed MTD Strategy

The Intuition

- "wishing for the best by preparing for the worst"
  - estimate risk accumulation by the attack surface function...
  - ...even though may have suffered less hits
- avoid bad migrations...
  - ...by requiring replacements have less risk accumulation by a pre-specified margin
  - less migrations reduce MTD overheads
- probabilistically migrate to one of replacements (including itself)...
  - ...in proportion with risk estimation

The Proposed MTD Strategy

to Put the Intuition in a Diagram



The Proposed MTD Strategy

The Details

- $E_{a,j}(i) = P_{a,i}(t_i)$: j's risk estimation of migrating to $i \in R(j) \cup \{j\}$
  - conservative estimate: "wishing for the best by preparing for the worst"
- eligible replacements

$$R'_\delta(j) = \{i | i \in R(j) \text{ and } E_{a,j}(i) \leq E_{a,j}(j) - \delta\}.$$

  - migrations entail costs
  - this helps avoid bad migrations

The Proposed MTD Strategy

The Details

j makes a probabilistic decision to migrate to $i \in R'_\delta(j) \cup \{j\}$ as follows:

- If $R'_\delta(j) = \emptyset$, the decision is (trivially) "migrate to j", i.e., not migrating.
- Otherwise $R'_\delta(j) \neq \emptyset$, j migrates to $i \in R'_\delta(j) \cup \{j\}$ with a probability of

$$M^{\rightarrow}_{a,j}(i) = \frac{\sum_{k \in R'_\delta(j) \cup \{j\} \setminus \{i\}} E_{a,j}(k)}{|R'_\delta| \sum_{k \in R'_\delta(j) \cup \{j\}} E_{a,j}(k)}.$$

properties

- $\sum_{i \in R'_\delta(j) \cup \{j\}} M^{\rightarrow}_{a,j}(i) = 1$: $M^{\rightarrow}_{a,j}(i)$ $(i \in R'_\delta(j) \cup \{j\})$ constitutes a probabilistic partition of 1
- $M^{\rightarrow}_{a,j}(i)$ is proportional to risk estimation

The Proposed MTD Strategy

Numerical Examples

- risk estimation: 3 (by definition, this must be j), 2, and 1
  - probabilities of being selected as the replacement are $1/4$, $1/3$, and $5/12$
  - $1/4 + 1/3 + 5/12 = 1$; $1/4 : 1/3 : 5/12 = 3 : 4 : 5$
- risk estimation: 5 (by definition, this must be j), 0, and 0
  - probabilities of being selected as the replacement are 0, $1/2$, and $1/2$

Evaluation

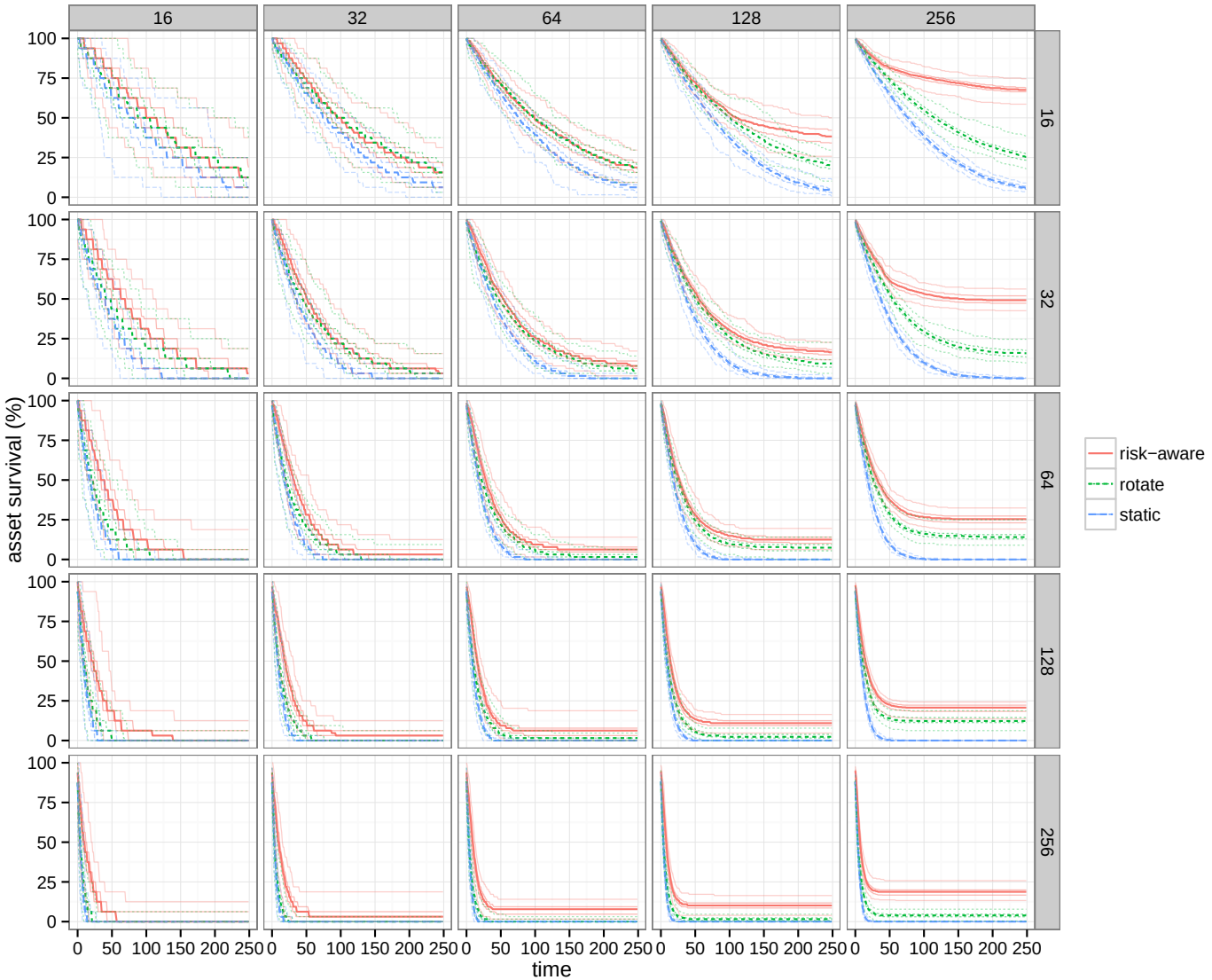
Setup

- Common Lisp model simulator[3]
  - S-shaped heterogeneous and dynamic attack surface parameters randomly generated
- comparison between 3 service deployment strategies
  - static
    - do not migrate once deployed
  - rotate
    - probabilistically choose a lesser used replacement
  - attack-surface heterogeneity-and-dynamicity aware ("risk-aware")
    - the proposed startegy
- what questions do these comparison address?
  - static vs. rotate/risk-aware
    - does MTD help?
  - rotate vs risk-aware
    - does risk awareness help?

[3] publicly available at `https://github.com/pw4ever/pw-sim-mtd`

Evaluation

min/25%/median/75%/max summary of survival rate for 512 nodes/100 rounds



column header: initial defender assets

row header: attacker budget

Findings in Detail

- When the service is dense and/or the attacker is strong, there is a high probability that an attacker (even a random one) will hit a static service.
- Although an MTD service is equally likely to be hit as a static one, the risk (of the service being compromised by the attacker) is amortized among the pool of replacements through migration.
- Therefore, over the same period of time, although more assets have been activated (and hence potentially been probed and attacked), fewer has reached a risk level high enough to be compromised.
- When the service is sparse *and* the attacker is weak, the very sparsity serves as adequate camouflage for a static service against the weak attacker, so that MTD does not show significant benefits.
- Nevertheless, risk awareness helps in improving security.
- Risk awareness helps avoid poor decisions such as migrating an asset from a lowly risky but more used node to a highly risky but less used one.

Q&A

thank you