Formalizing Agent's Beliefs for Cyber-Security Defense Strategy Planning

8th International Conference on Computational Intelligence in Security for Information Systems

Karsten Martiny‡ **Alexander Motzek*** Ralf Möller*

* Universität zu Lübeck
Institute of Information Systems
Ratzeburger Allee 160, 23562 Lübeck, Germany
{motzek,moeller}@ifis.uni-luebeck.de

‡ Hamburg University of Technology
Institute for Software Systems
Am Schwarzenberg-Campus 3, 21073 Hamburg, Germany
karsten.martiny@tuhh.de

June, 16th 2015

Introduction

- **Proactive** measures sometimes not possible.
  e.g. proactive removal of known vulnerabilities
  Critical infrastructure remains exposed.

→ **Reactive** strategies at a last stage of an attack.
→ maintain a model of an intruder's belief state
  formalized using Probabilistic Doxastic Temporal (**PDT**) Logic.

Considerations and Assumptions

- We start at a **last point of defense**.
- **Highly skilled** attacker with extensive knowledge about our network.
- **Two stages** of an attack.

- Stage of escalated privileges (“Shell”).
- Code execution of (handcrafted) malware (“Exec”).

Shell

- "From stock"
- Obtained through (standard, **known**) **patterns**.
- Can be detected through an IDS: $Obs_D(attack(X))$.
- Does not expose attacker's identity.

→ Can be **defended()**. OR

→ If **monitored**, can allow observation of executed code.

Code execution

- Will be **handcrafted**.
- Might allow **identification** of attacker.
- Might reveal **intention** of attack.
- Can seriously *harm* our system.
- If not immediately observed, **no traces left.**

→ So we should **always defend?** but

Always defend?

- **No.** No IDS is perfect.
- Would perfectly tell the attacker if **she is undetected**.
- Defense might be dangerous as well.

Minimal Example - Introduction

- We have two devices.
- **Productive** system and **Backup** system.
- Backup can be **sacrificed**.
- Only defend productive?

Minimal Example - Introduction

- We have two devices.
- **Productive** system and **Backup** system.
- Backup can be **sacrificed**.
- Only defend productive?

  No. We perfectly tell her which device to attack.

We can belief...

- At every timestep one attack (maybe).
- **Observe nothing** at t → **missed an attack** at t or at $t - 1$.
- **Observe attack** at t, → missed a previous attack at $t - 1$.

→ We consider multiple *worlds* possible.

Attacker can also belief...

- Wants to execute her code stealthy, but wants to execute it.
- If she gains a shell, she can be...

(i) Stealthy: Execute directly and **disappear**.

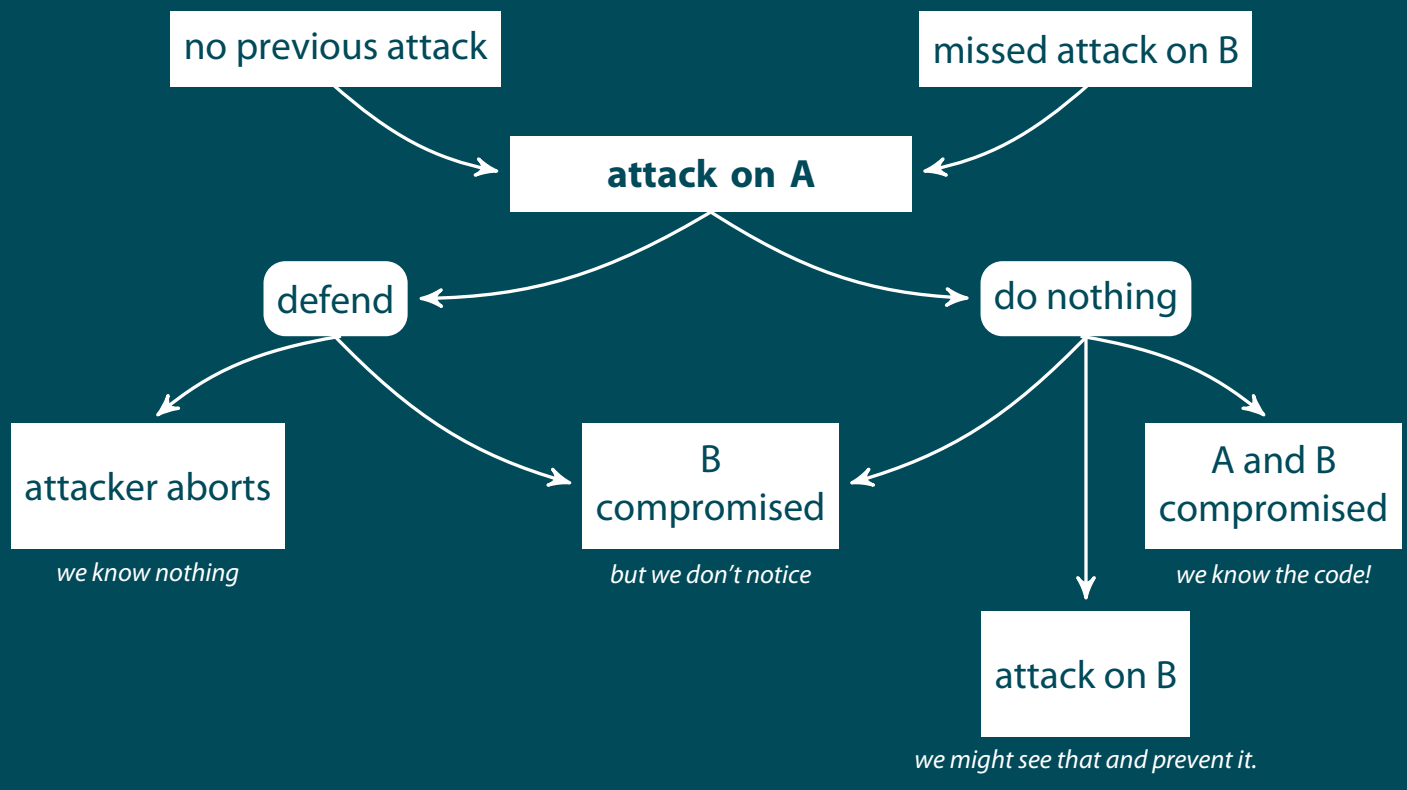
(ii) Aggressive: Gain another shell to **broaden** attack.

(iii) Lure: **Distract** with attack to other host

PDT Logic

- **PDT logic** can **formalize** all considerations into one model.
- Reason about beliefs in **ranges**.
- World: *sequence of actions* + *associated probabilities*.
- Agents consider *multiple* worlds possible.
- Beliefs might *divulge*.

Minimal example - IDS says "attack on A"



Minimal example - PDT applied

- **Worst Case**: $\varphi = exec(B) \wedge \neg Obs_D(exec(B))$

$$\neg defend(A) \models B_D(\varphi) : [0,\ 0.05], \text{and}$$
$$defend(A) \not\models B_D(\varphi) : [0,\ 0.05].$$

- I.e. we are better off not defending.

Minimal example - PDT applied

- Say, we don't defend $\neg defend(A)$.
- Opportunities to **analyze** the intruder's malicious **code**, i.e., $\varphi = Obs_D(exec(A)) \vee Obs_D(exec(B))$

$$\neg defend(A) \quad \models B_D(\varphi) : [0.9, 1].$$

- I.e. our belief is rather high, that we will be able to get a grasp at her code.
- Nested beliefs allow even deeper understanding.

Contributions

- Well-defined theory to formalize multi-agent beliefs.
- Formal analysis of an adversary's belief evolution.
- Enables advanced defense strategies.