Abstract

- Existing security solutions for rapidly-changing, modern clouds still struggle with too many false-positive alarms.
- The lack of a true, causal link between IoCs makes correlations error-prone.
- We propose an idea for a framework that automatically and strategically injects lures and decoys, so that we can span an attack graph onto which alarms are projected for reconstruction.

Introduction

- Recent work focuses on correlating many weak indicators by IP addresses, alarm types, or time windows.
- Cyber deception reduces false-positives, but they are not as automatic, nor adaptive to scale well with modern cloud environments.

We focus on three aspects:

- **Cyber Deception.** Use honeypots and honeytokens for stronger IoCs.
- **Automatic Injection.** Strategically and automatically place tripwires in existing applications, and react to changes.
- **Attack Graphs.** Causally connected deceptive components naturally span an attack graph onto which incoming alarms can be projected, which provides clearer insights into multi-step attacks.

We ask the following research question:

- **„Are automatically injected tripwires suitable to reconstruct multi-step cyber attacks in modern cloud environments?"**

Conclusion

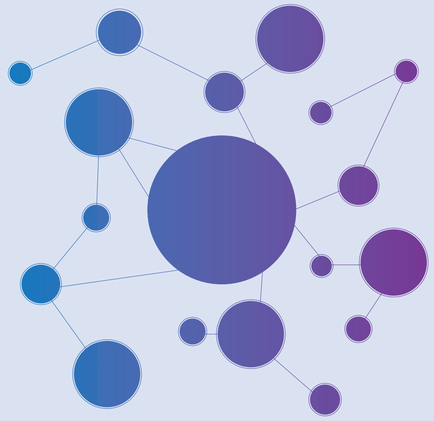
- We describe a framework and tripwires.
- Future work implements such a system and evaluates attack reconstruction.

Towards Reconstructing Multi-Step Cyber Attacks in Modern Cloud Environments with Tripwires

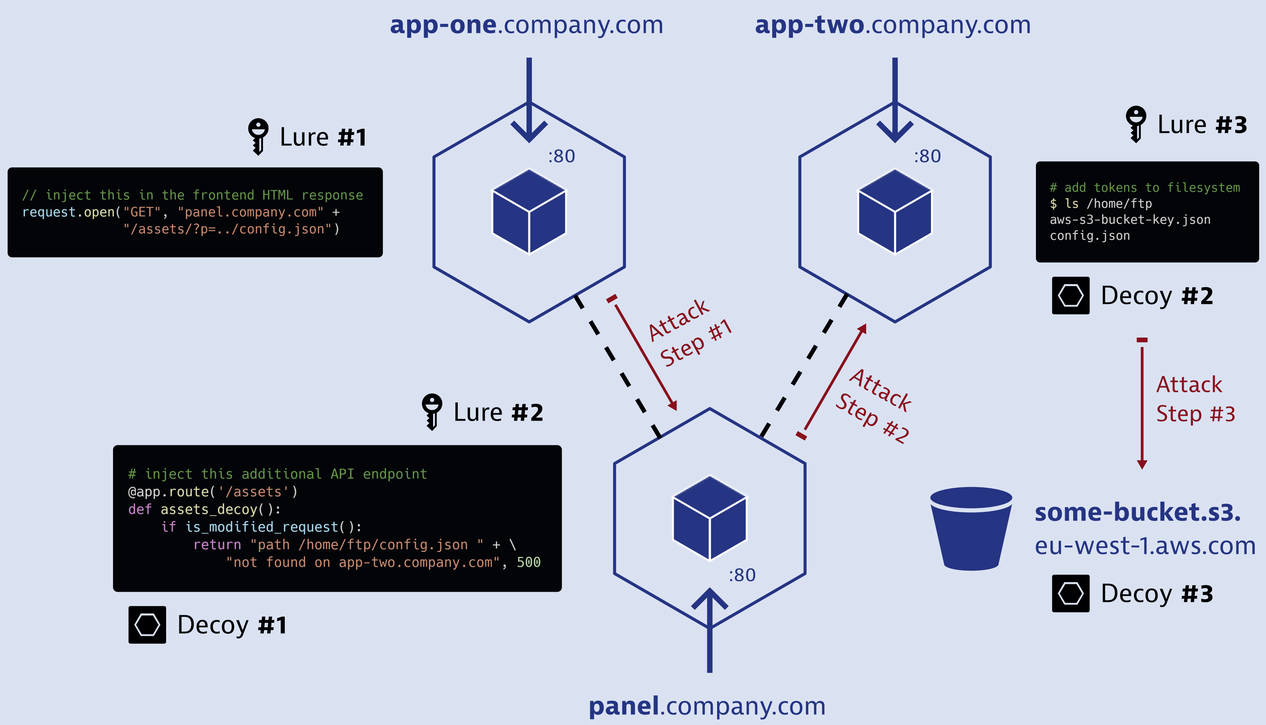
MARIO KAHLHOFER
MICHAEL HÖLZL
ANDREAS BERGER

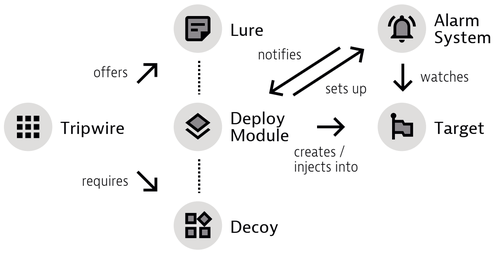
dynatrace

TRIPWIRES



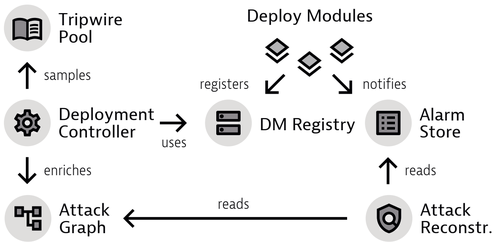
Let's reconstruct **multi-step** attacks in modern **clouds** with automatic cyber **deception** by spanning **attack graphs**





A **tripwire** describes the relation between lures, decoys, their deployment on some targets via a deploy module, and its associated alarm system.

- **Connected.** Each tripwire comes with a set of lures and decoys that enforce strong causal dependencies.
- **Managed.** Deployment and clean-up of lures, decoys, and alarm systems is taken care of accordingly.
- **Automatic.** Injection points in libraries of existing applications are detected and automatically populated with tripwires.
- **Strategic.** Tripwires are placed to efficiently cover the environment, and to discover relevant attack phases.
- **Adaptive.** Tripwires are re-deployed when the environment changes.



The **framework** describes the life cycle of tripwires in cloud environments, from deployment, alarm and attack graph storage, to attack reconstruction.

- **Deploy Module.** Process hooks identify application libraries and then provide a DM that can inject tripwires and associate an alarm system with it.
- **Tripwire Pool.** Holds multiple definitions of tripwires that could be deployed.
- **Deployment Controller.** Manages the deployment of tripwires in the cloud.
- **Attack Graph.** Stores the relationships between deceptive components.
- **Attack Reconstruction.** Uses backward and forward tracking algorithms to reconstruct multi-step cyber attacks.