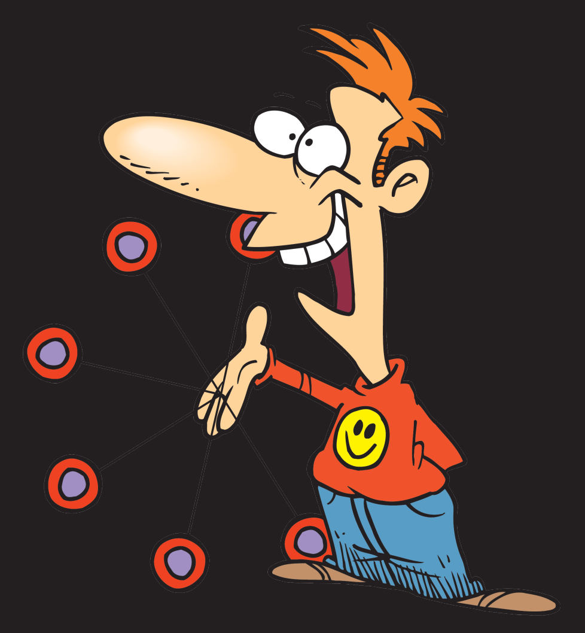
Yo-Yo Attack - Vulnerability in auto-scaling mechanism

Mor Sides [1] Anat Bremler-Barr [1] Elisha Rosensweig [2]

[1]Interdisciplinary Center, Herzliya

[2]Cloudband, Alcatel-Lucent



IDC HERZLIYA

Background: Auto-Scaling Mechanism

- Auto-scaling enables adapting the number of application machines automatically to support changes in user load.
- A scaling policy to guide auto-scaling is defined by the application owner. It commonly consists of a scaling criteria (function) and corresponding thresholds for overload and underload.

Vulnerability in auto-scaling mechanism

- Auto-scaling mechanism can provide protection from many basic Distributed Denial of Service (DDoS) attacks, with the virtually-unlimited resources of a cloud available.
- However, it also opens the door to a new type of attack - the Economic Denial of Sustainability (EDoS) attack, where the application owner pays large sums for virtual machines that yield negligible gains.
- Here we present the '**Yo-Yo attack**', an instance of **EDoS** attack targeting the auto-scaling mechanism, which is difficult to detect while causing economic damage and also performance damage.

Yo-Yo Attack

- Yo-Yo attack cycles between two phases repeatedly:
  - ▷ **On-attack**-the attacker sends a short burst of traffic that causes the auto-scaling mechanism to perform a scale up.
  - ▷ **Off-attack**-the attacker stops sending the excess traffic (after identifies that the scale up has occurred) that causes the auto-scaling mechanism to perform a scale down.
- The Yo-Yo attack can also be considered a Reduction of Quality (**RoQ**) attack. RoQ attacks aim to keep an adaptive mechanism oscillating between over-load and under-load conditions, which in the Yo-Yo attack triggers scale-up and scale-down processes repeatedly.

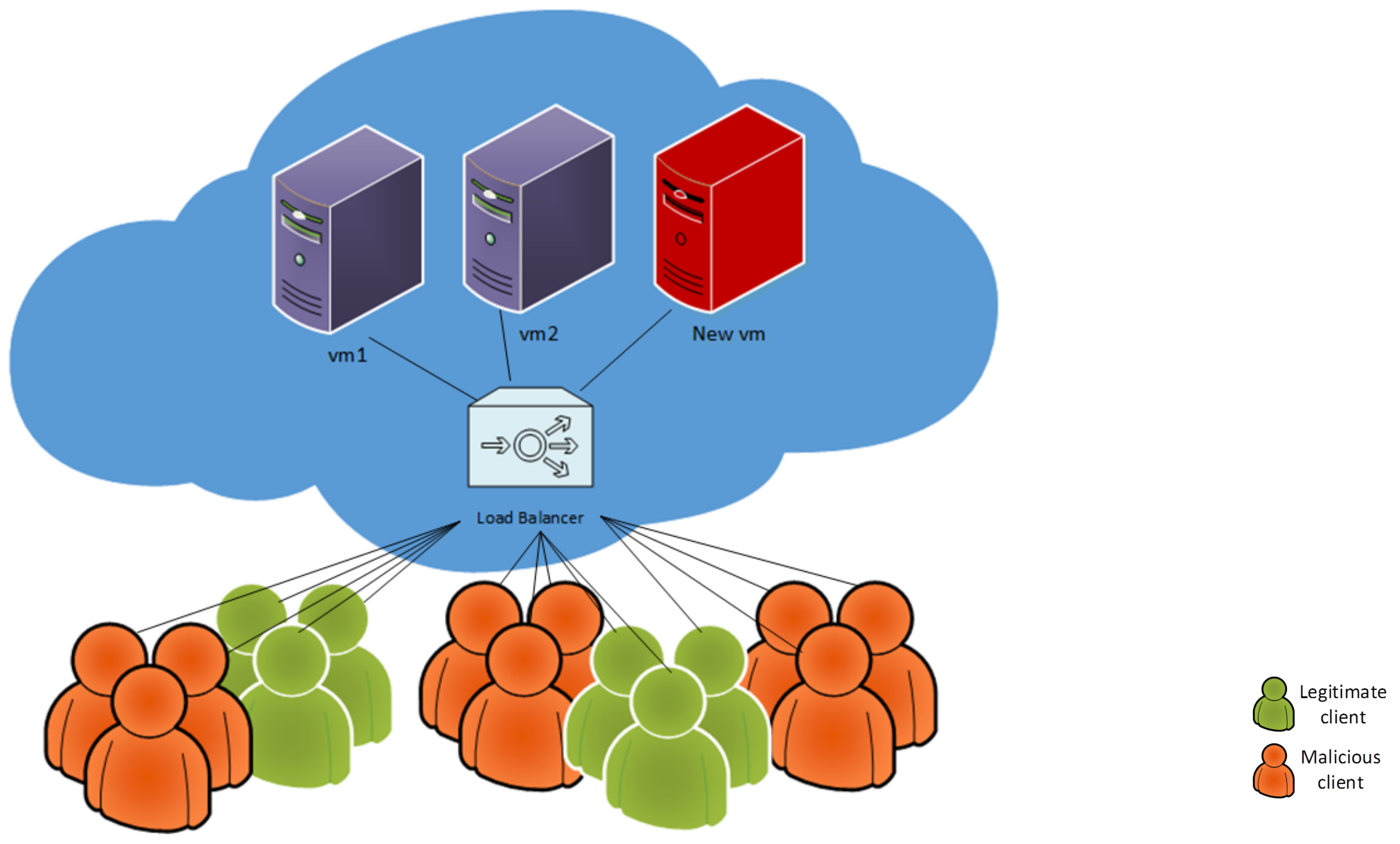


Figure 1: Yo-Yo attack environment

Detecting Scale Up

- The key idea is that scale is done usually in order to improve the response time, thus the response time reveals some information on the state of the auto-scaling mechanism.

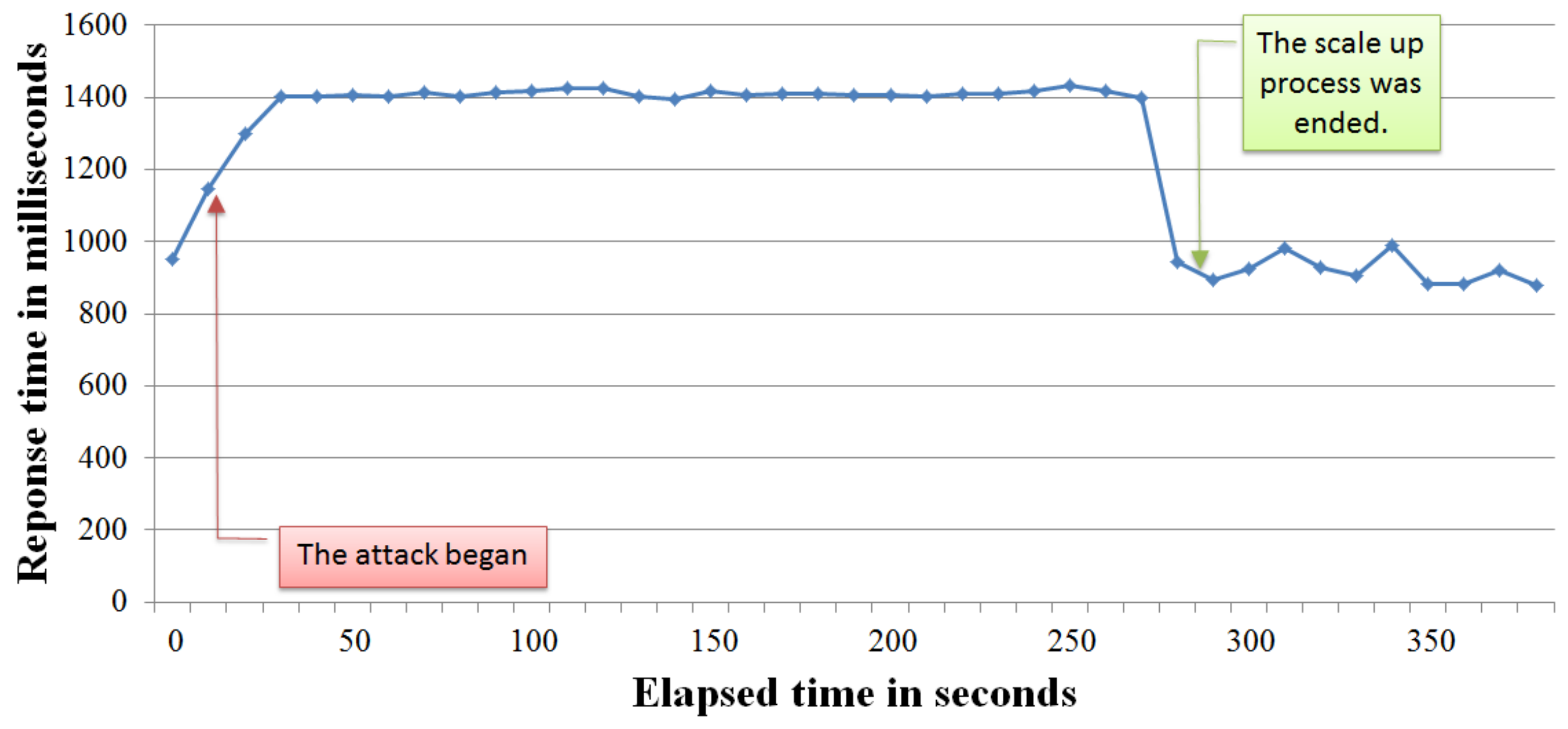


Figure 2: Approximate scale up period according to client response time

Experiment with Yo-Yo attack

We demonstrate the Yo-Yo attack on Amazon cloud service.

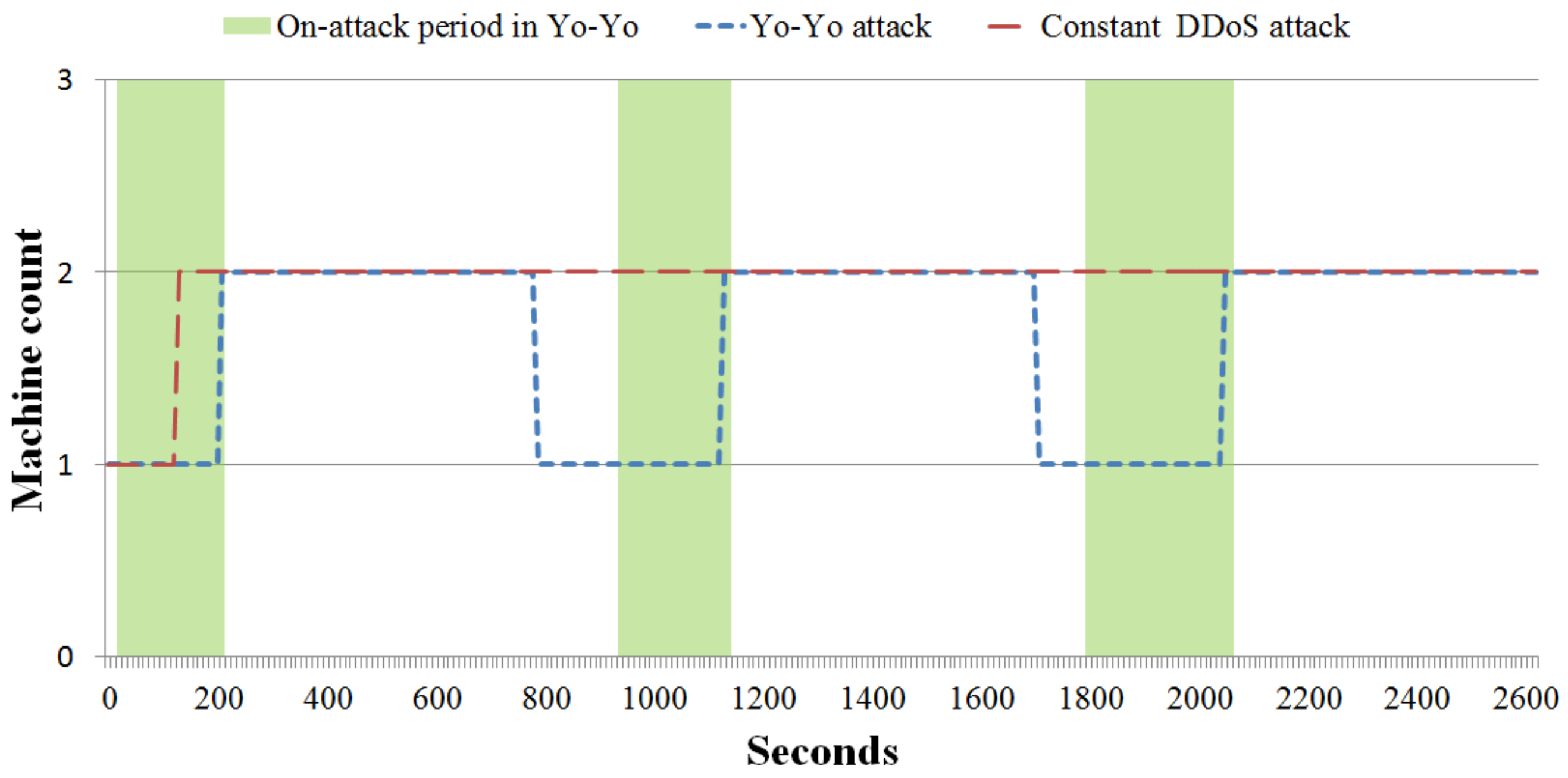


Figure 3: **Economic damage:** machine count comparison

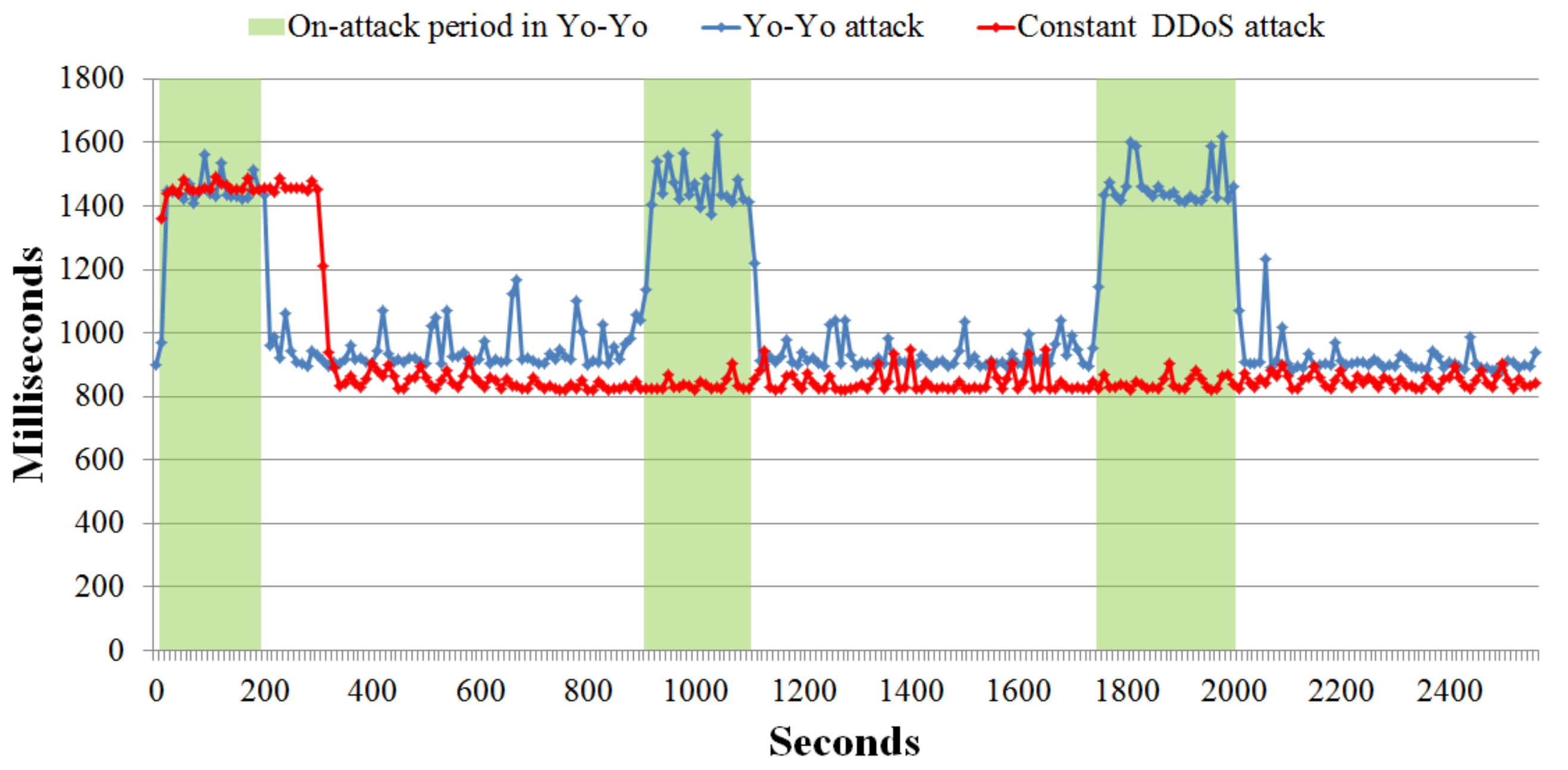


Figure 4: **Performance damage:** response time comparison

Evaluate the power of the attack

- **Attack cost=** The factor between *off-attack* to *on-attack* is almost 3.5, i.e. reducing the attack cost in about 77%.
- The attacker would be interested in maximizing the damage per unit cost, denoted as **potency**.
  - ▷ $potency = m/t$, where:
    - m- the averaged number of extra machines.
    - t- the ratio of *on-attack* duration to the entire attack duration.
  - ▷ In a full DDoS attack the potency is 1, while in our experiment, Yo-Yo attack, the potency is 0.71/0.23 which is 3.08. Thus the attacker is 3 time more effective.

Estimations on Larger Services

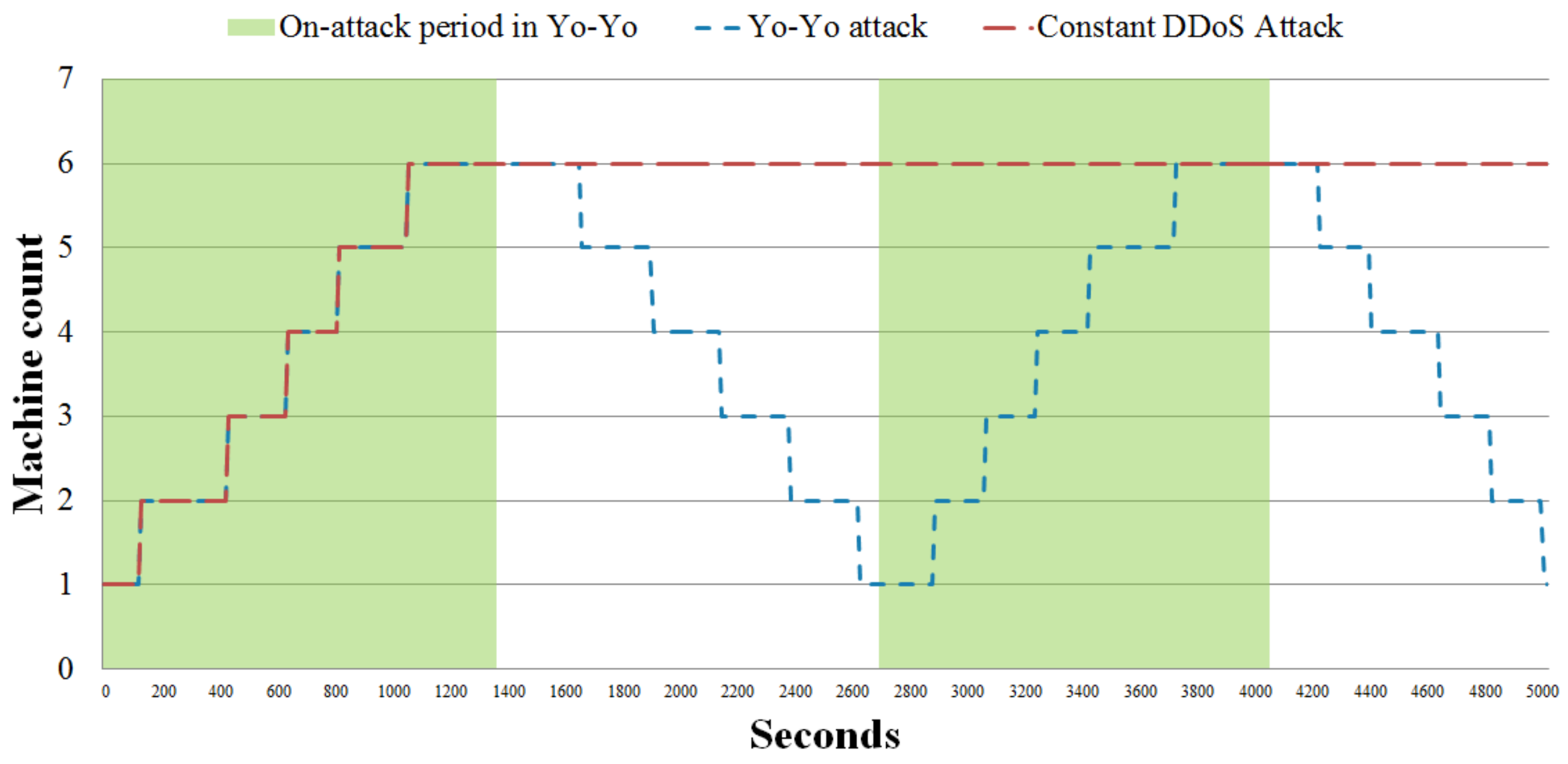


Figure 5: Yo-Yo attack on unlimited scaling group

Future Work

- Future work will focus on:
  - ▷ Execute the attack on more environments, statefull services and middleboxes.
  - ▷ Ways to mitigate the attack.

Acknowledgments

- This research was supported by European Research Council under the European Unions Seventh Framework Programme (FP7/2007-2013) /ERC Grant agreement No259085.

http://www.deepness-lab.org mor.sides@idc.ac.il