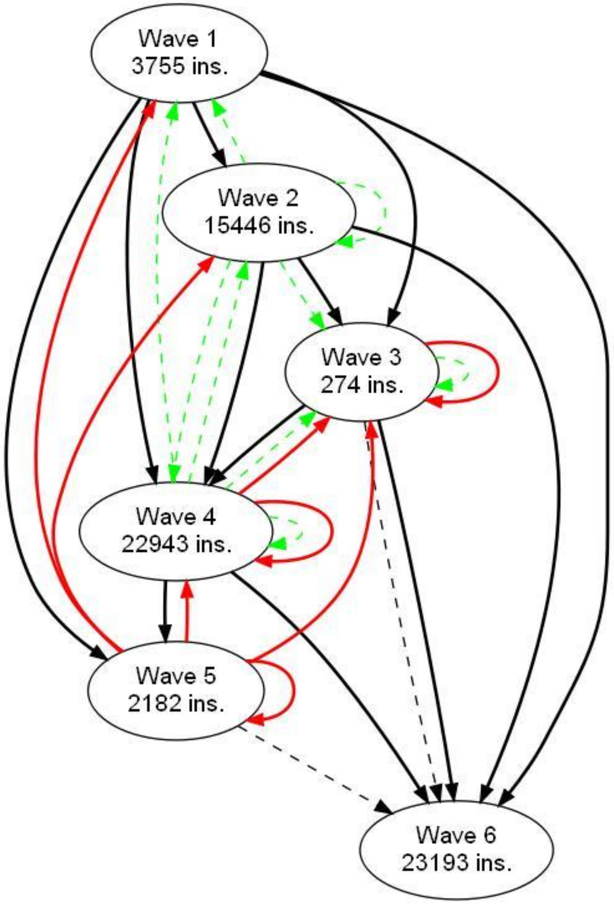
Large-scale Malware Experiments: Why, How, and So What?

Joan Calvet, Pierre-Marc Bureau,
Jose M. Fernandez, Jean-Yves Marion

École Polytechnique de Montreal, ESET, LORIA

Who We Are

- Academic Researchers
  - Botnet research
  - Program analysis and reverse engineering
  - Formal methods
- Industry researcher with interest in botnet mitigation
- Canadian government funding for large scale security experiments (Polytechnique)



Presentation Outline

- Why?
  - Ethics
  - Scientific soundness
  - An interesting case study: Waledac
- How?
  - Physical infrastructure
  - Software infrastructure
  - Attack scenarios
  - Measurements
- So What?
  - Experiment baseline
  - Experimental results
  - Lessons learned
- Where is this going?

Large-scale Malware Experiments

WHY ?

Botnet Research

- **Scale**
  - Understand malware at the botnet level
  - Interaction between thousands of infected hosts
- **Control**
  - Botnet
  - Environment
  - Attack
- **Reproducibility**

Ethics

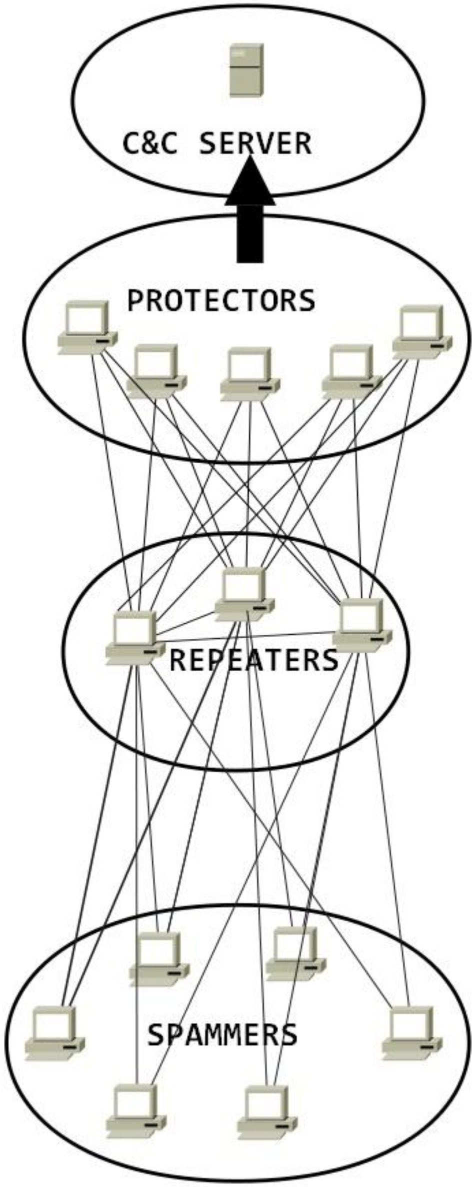
- We can not create our own botnet on the Internet

- We can not play with existing botnets and innocent victims

- We should not tip off botnet operators (trigger arms race)

An Interesting Target: Waledac Botnet

- Peer-to-peer protocol
- Good understanding of the binaries
- No replication
- Interesting weaknesses in p2p implementation



Peer-to-peer Protocol (1)

- Each peer maintains a list of known peers (*RList*)
- Bots exchange parts of their *RList* on a regular basis to maintain connectivity
- Fallback mechanism over HTTP to fetch new peers

```
<lm>
<localtime>1244053204</localtime>
<nodes>
<node ip="W.X.Y.Z" port="80"
time="1244053204">469abea004710c1ac0022489cef03183</node>
<node ip="A.B.C.D" port="80"
time="1244053102">691775154c03424d9f12c17fdf4b640b</node>
…
</nodes>
</lm>
```

Peer-to-peer Protocol (2)

- Vulnerable to sybil attack:

```
<lm>
<localtime>0</localtime>
<nodes>
<node ip=“myIP" port="80“ time=“0">
00000000000000000000000000000001
</node>
...
<node ip=“myIP" port="80" time=“0”>
000000000000000000000000000001F4
</node>
</nodes>
</lm>
```

Large-scale Malware Experiments

HOW ?

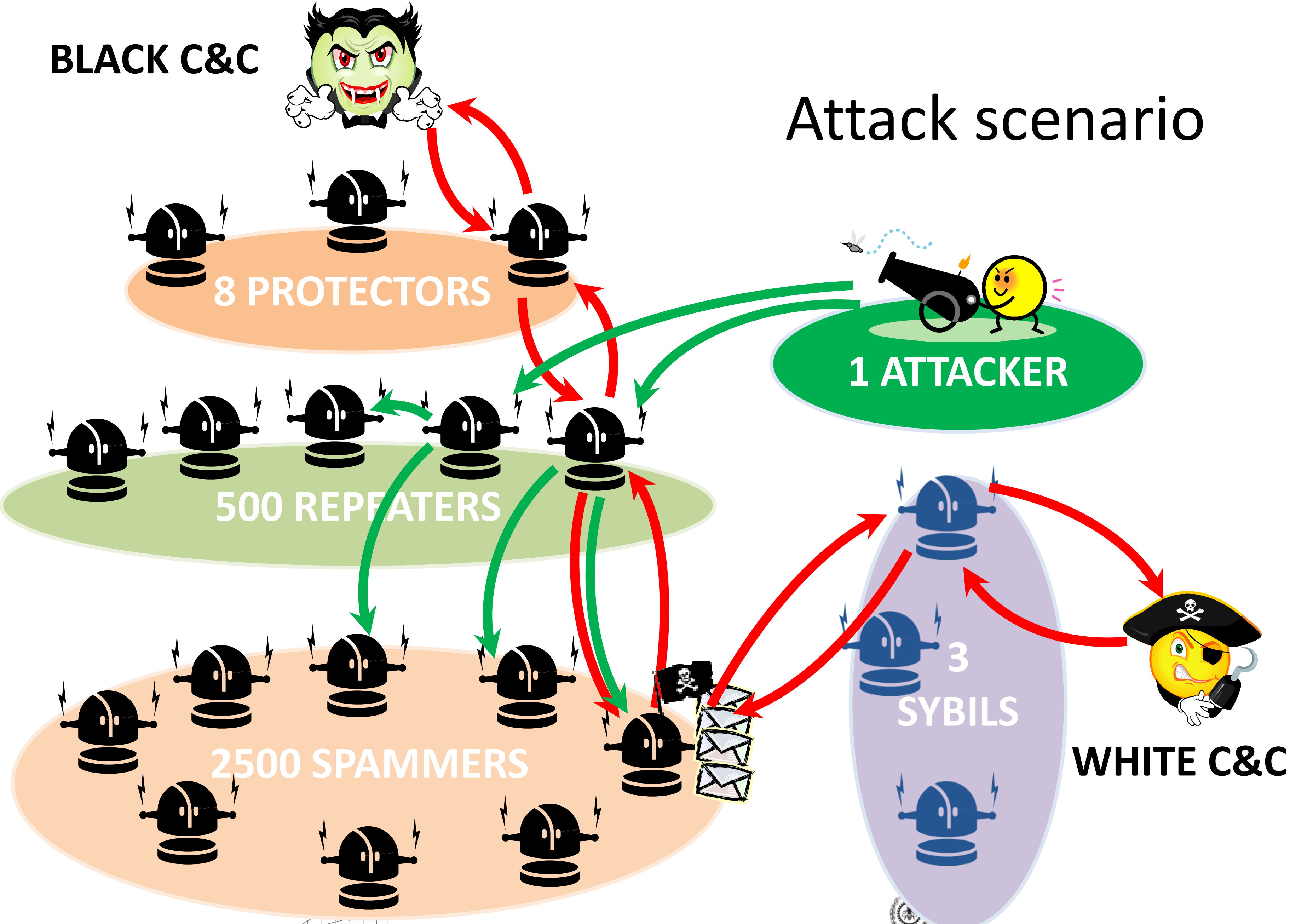
Experimental Environment

- Cluster with 98 blades
- Quad core processors
- 137GB storage
- 8GB RAM
- 4 x gigabit ethernet (network separation)
- No Internet connection

Experimental Environment (2)

- VMWare Virtual Machines
- Deployed using xCAT
- 30 VMs per blade (~3000 bots)
- Windows XP SP3
- Python script to have a remote control on the bots (infection/disinfection/measure)
- HTTP, DNS and SMTP servers

Attack scenario



BLACK C&C

8 PROTECTORS

1 ATTACKER

500 REPEATERS

3 SYBILS

WHITE C&C

2500 SPAMMERS

Measurements

1. Botnet activity
   – Number of spam sent by the botnet over a fixed period of time (botnet efficiency)

2. Attack penetration
   – Percentage of sybils in peer lists

3. Connectivity of the botnet
   (for details check paper)

Large-scale Malware Experiments

SO WHAT ?

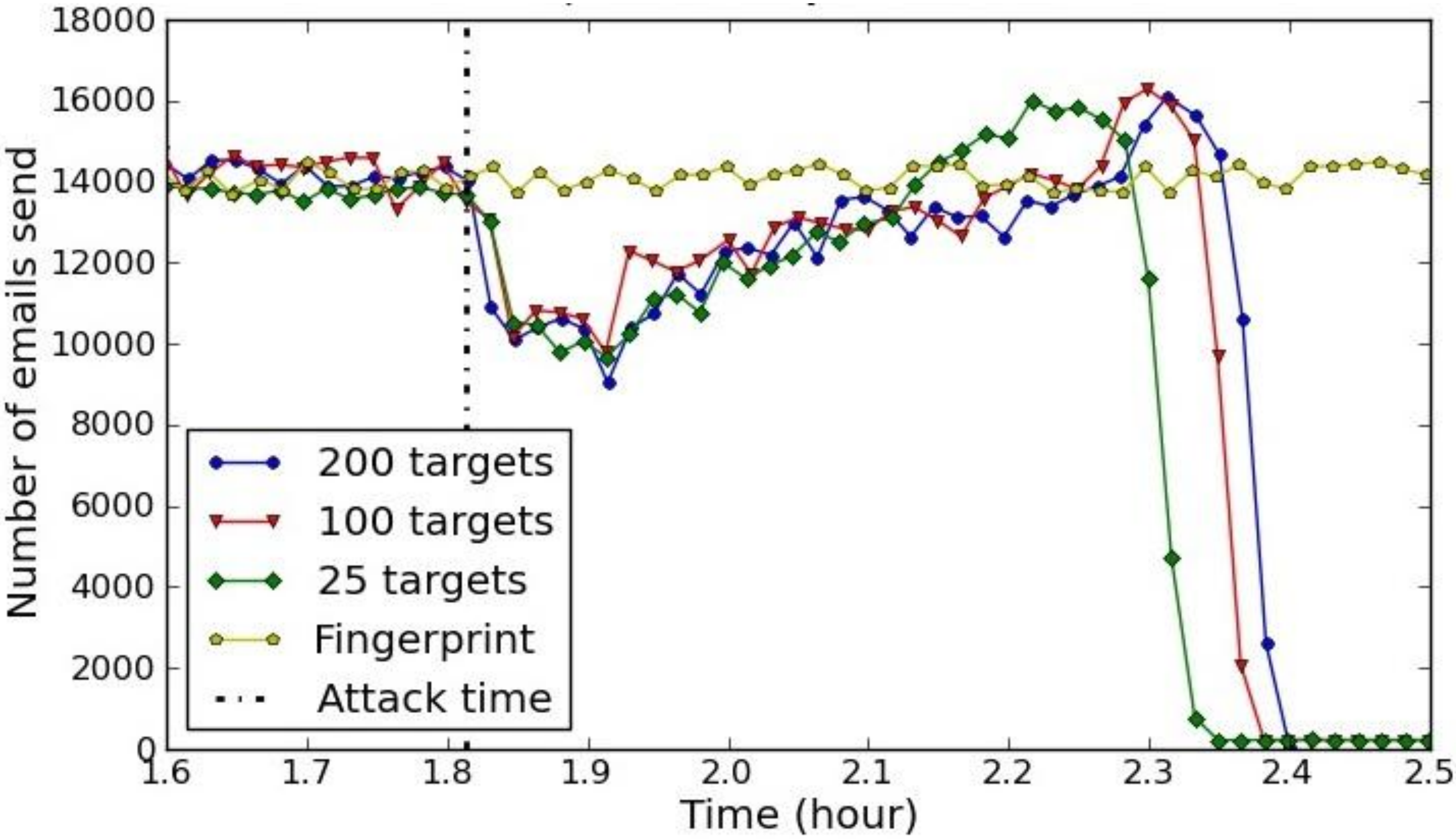
Launch the experiment

- Experiment baseline (without attack):

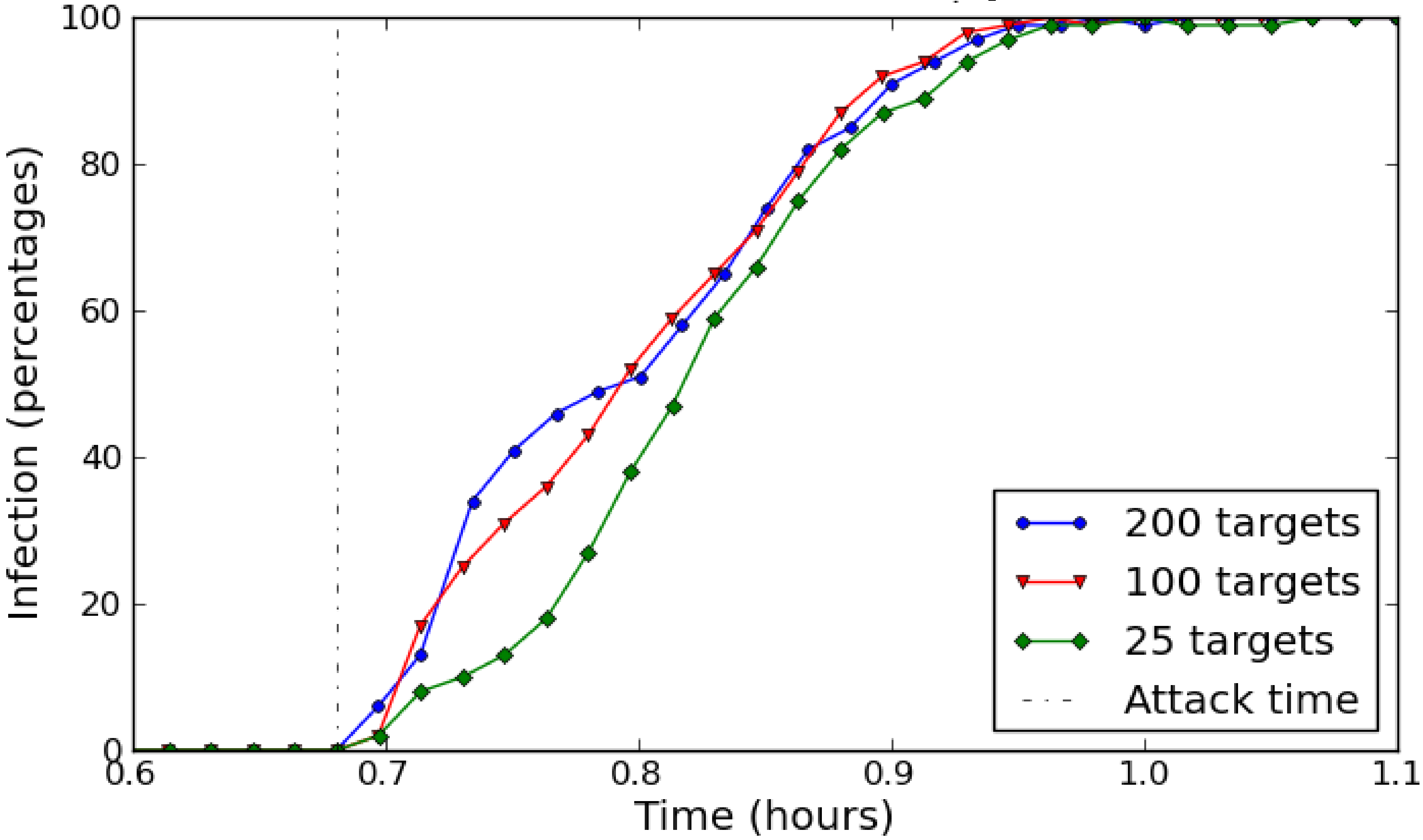
Emails	**13 200 per minutes**
Sybil ratio in peer list	**0%**
Dialog between bots and the C&C server	**120 per minutes**

- Experimental variable:
  - number of direct targets (Repeaters) : 25,100,200.

Spam Sent by the Botnet



RList Infections for Repeaters



Lessons Learned

- “Bad” use of cryptography was not a mistake!
- More aggressive attacks are not necessarily faster!
- Nobody specializes in booting thousands of identical VMs:
  - Microsoft genuine advantage
  - Hostname collisions
  - Make sure you have decent air conditioning

Future Work

- Improve the realism of network latencies in relation to network topology
- Play “cat and mouse”, where we can apply real time reaction from the botmaster and its effect on botnet performance (game theory ?)
- Add dynamic infection/disinfection, diurnal effect…

Conclusions

- Demonstrated viability of safe at scale malware experiments
- Learn new facts about Waledac operation (otherwise hard to find out)