Towards Network Containment in Malware Analysis Systems

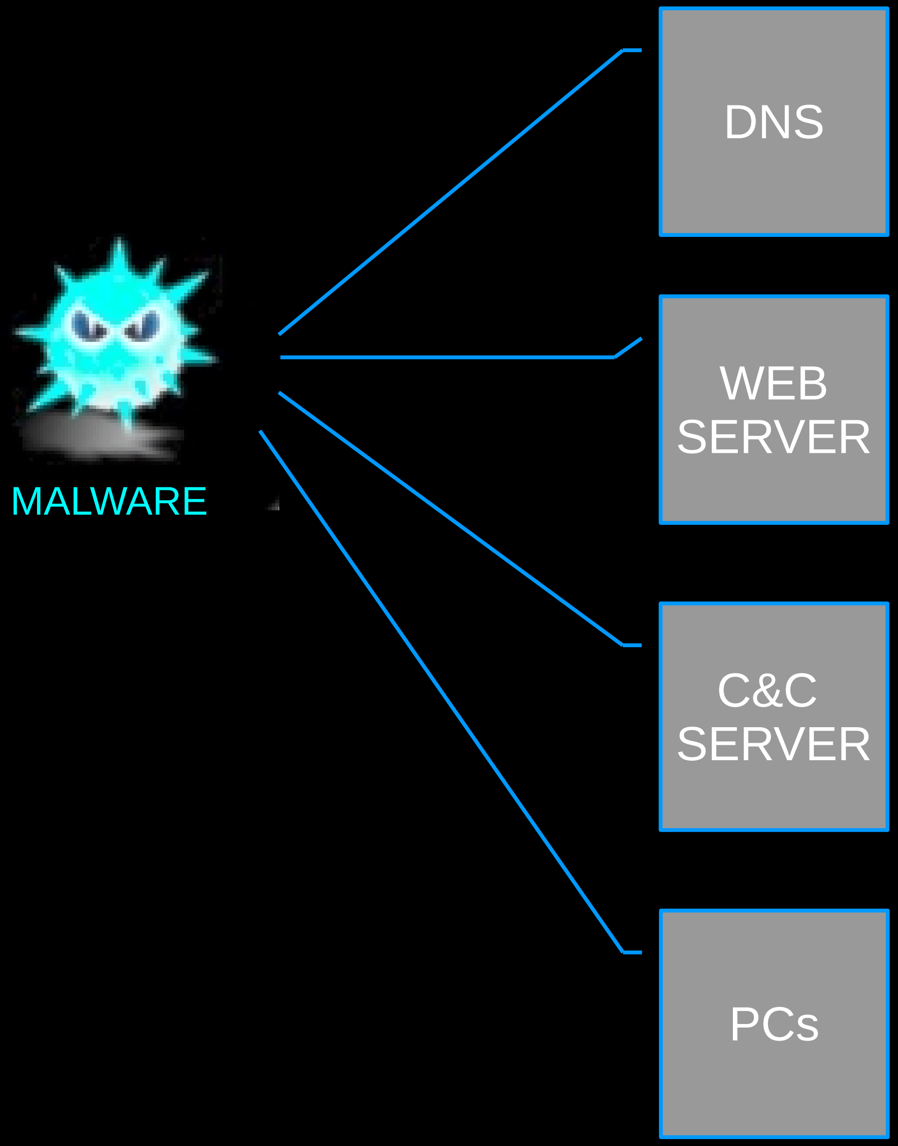
Mariano Graziano, Corrado Leita, Davide Balzarotti

ACSAC, Orlando, Florida, 3-7 December 2012

Malware Analysis Scenario

- Analysis based on Sandboxes (API Hooking, Emulation)
- Complex and distributed Security Companies Infrastructure
- Malware behavior often depends on external factors (C&C servers)
- Sophisticated attacks involve multiple stages

Malware Execution Stages



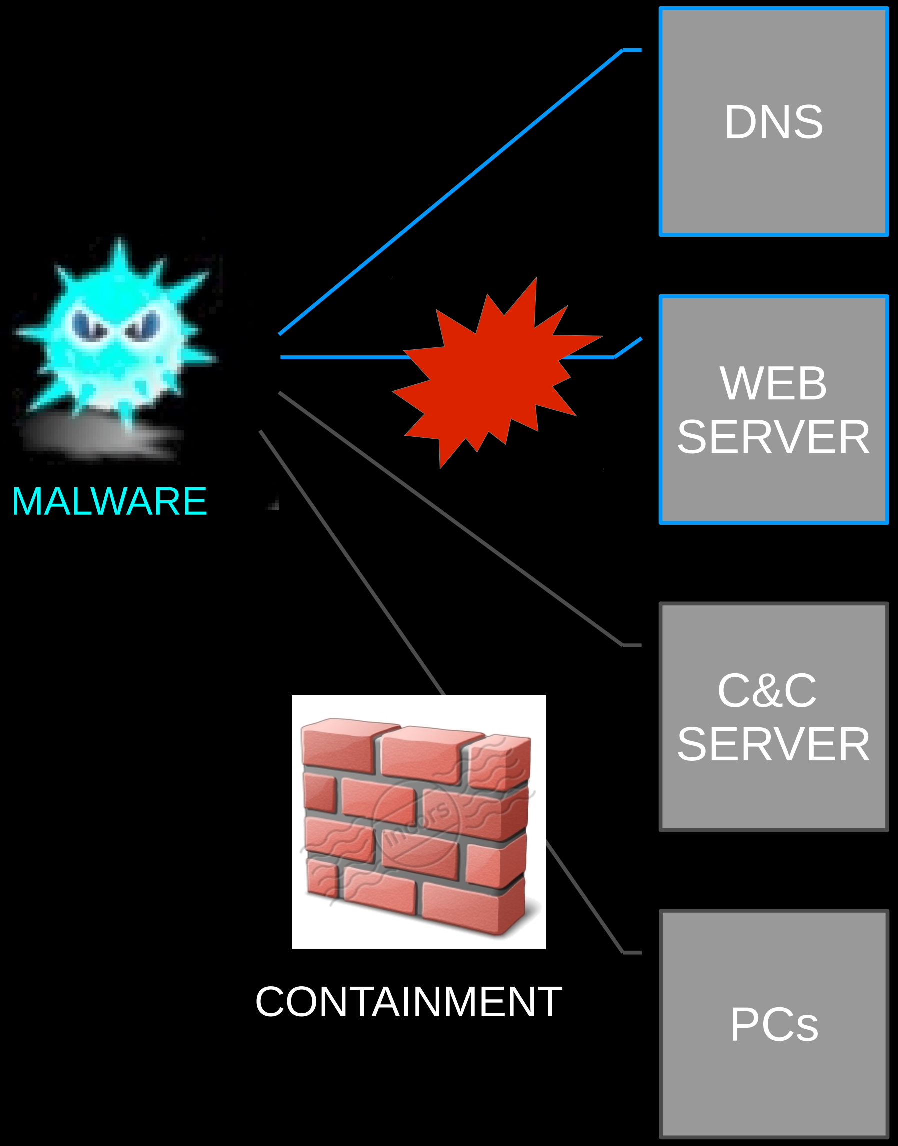
DNS name resolution

Download additional components, check Internet connectivity

Receive commands, exfiltrate information

Extend infected population

Repeatability & Containment



DNS name resolution

Web Server Unreachable, Impossible to download the components

Receive commands, exfiltrate information

Impossible to harm other machines

Goal

- Goal:
  - Model/Replay the network traffic for malware containment and experiment repeatability

- Motivation:
  - Malware behavior often depends on the network context
  - Experiments are not repeatable over time
  - Sandbox containment of polymorphic variations

Malware Containment

- Only possible in case of:
  - Polymorphic variations
  - Re-execution of the same sample
- Full containment → Repeatable execution

- Current containment solutions:

APPROACH	CONTAINMENT	QUALITY
Full Internet Access	x	~
Filter/Redirect specific ports	~	~
Common service emulation	v	~
Full Isolation	v	x

Roadmap

- Introduction
- Protocol Inference
- System Overview
- Evaluation

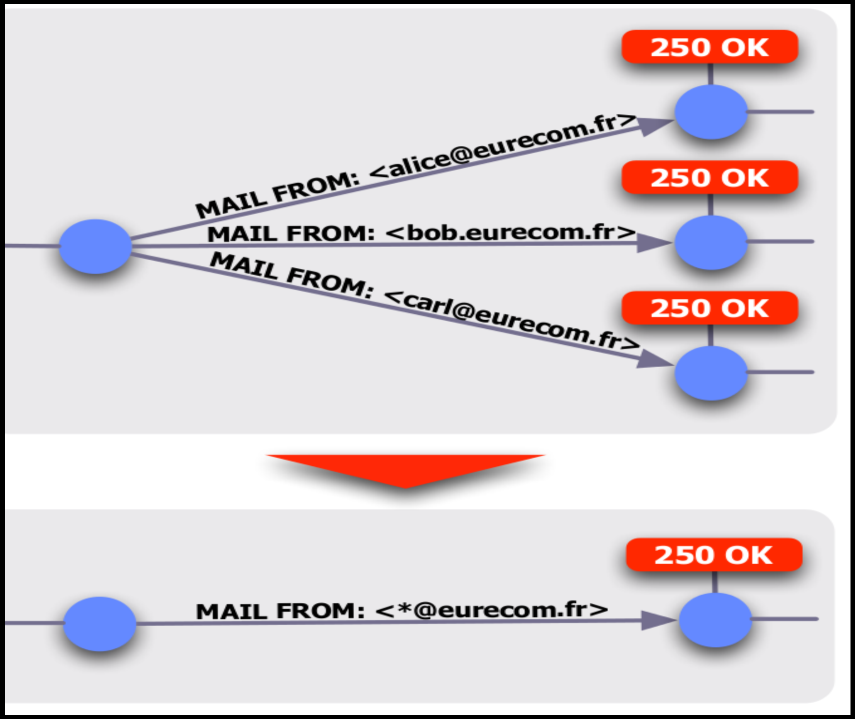
ScriptGen[1]

- Existing suite of protocol learning techniques developed for high interaction honeypots
- It aims at rebuilding portions of a protocol finite state machine (FSM) through the observation of samples of network interaction between a client and a server implementing such protocol
- No assumption is made on the protocol structure, and no a priori knowledge is assumed on the protocol semantics

1 Leita Corrado, Mermoud Ken, Dacier Marc - “ScriptGen: an automated script generation tool for honeyd” - ACSA 2005, 21st Annual Computer Security Applications Conference, December 5-9, 2005, Tucson, USA

Finite State Machine

- It is a tree:
  - The vertices contain the server's answer
  - The edges contain the client's request



SMTP Finite State Machine

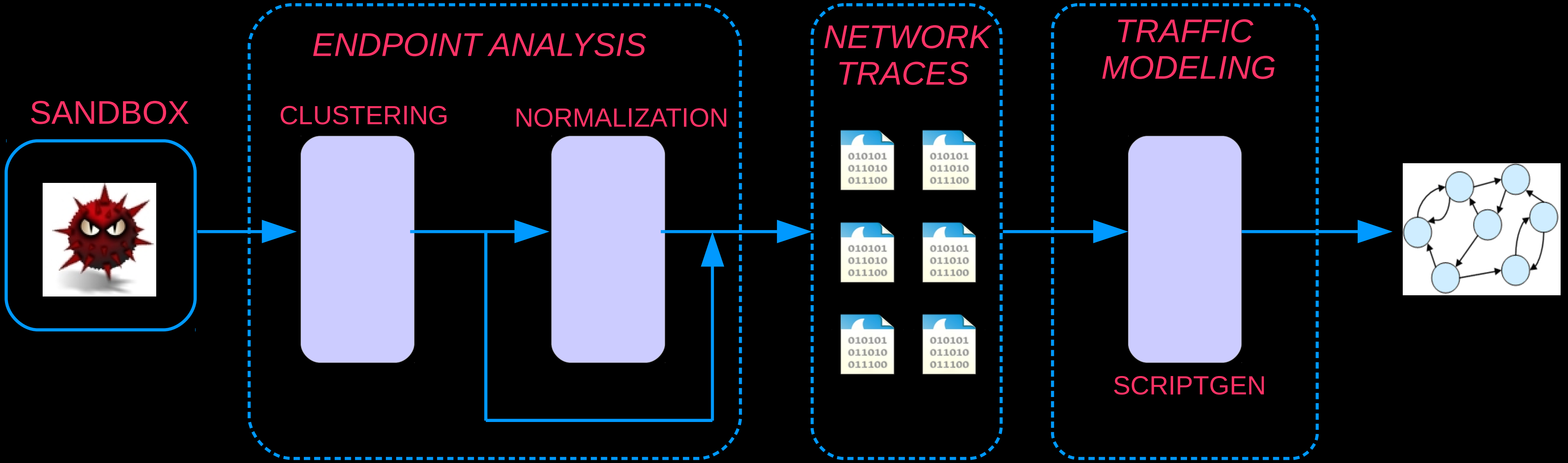
Roadmap

- Introduction
- Protocol Inference
- **System Overview**
- Evaluation

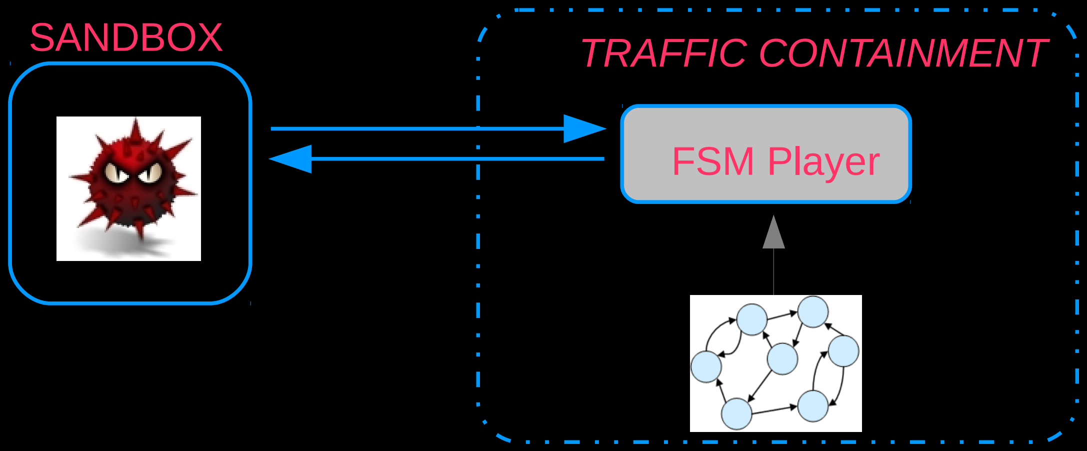
System Overview

- Traffic Collection
    - By running the sample in a sandbox or by using past analyses
- Endpoint Analysis
    - Cleaning and normalization process
- Traffic Modeling
    - Model generation (two ways: incremental learning or offline)
- Traffic Containment
    - Two modes (Full or partial containment)

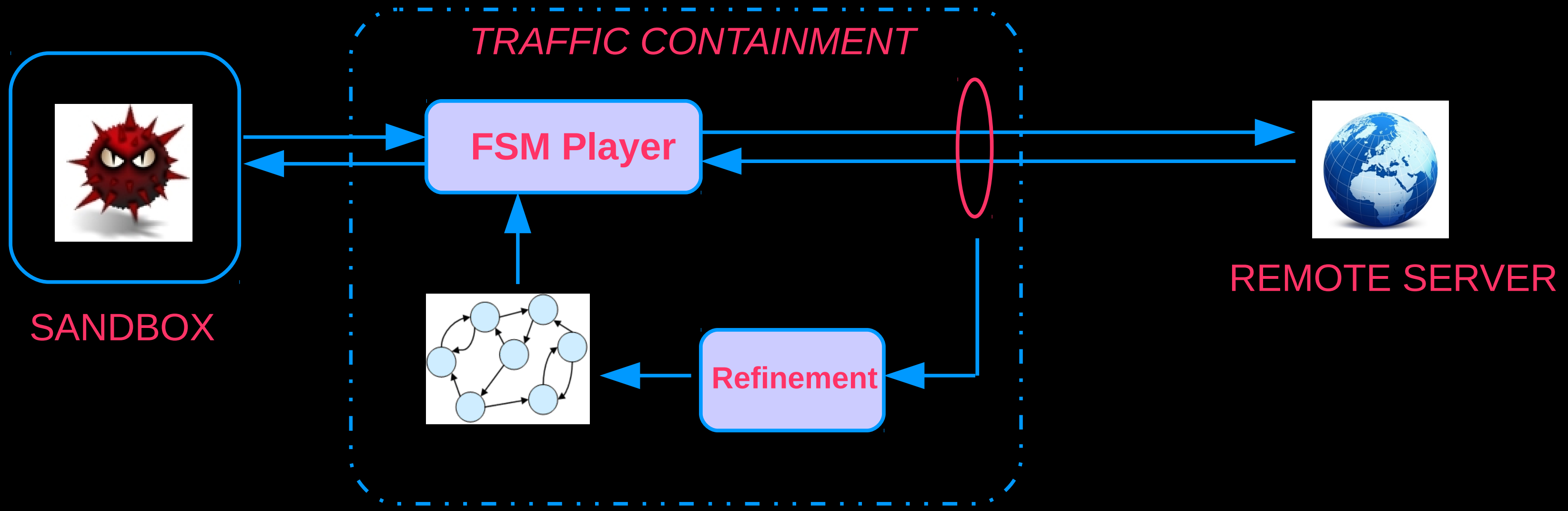
Traffic Model Creation



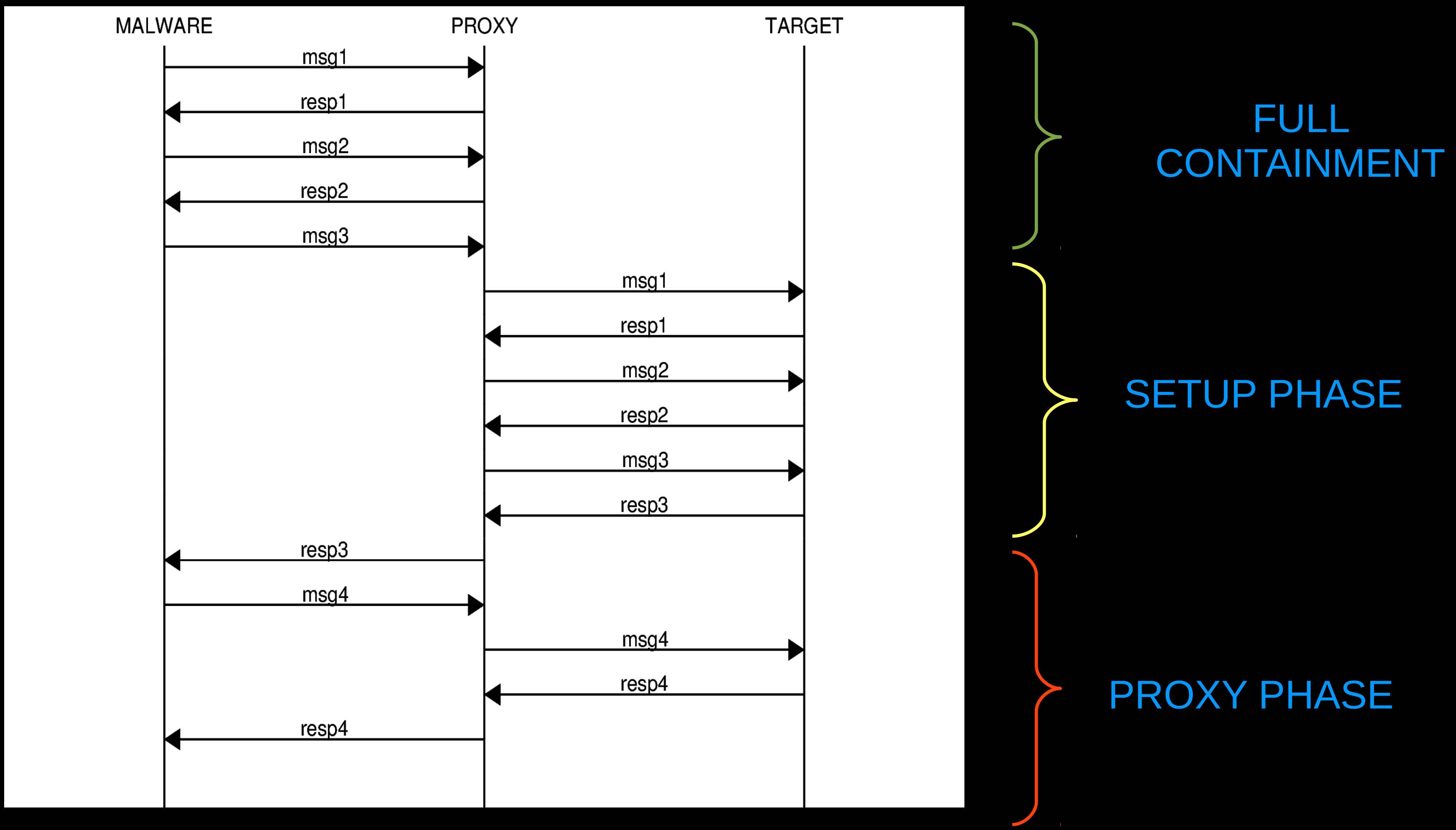
Mozzie – Full Containment



Mozzie – Partial Containment



Partial containment



Roadmap

- Introduction
- Protocol Inference
- System Overview
- Evaluation

Experiments

- Goals
  - Find minimum number of network traces to generate a FSM to fully contain the network traffic
  - Learning optimal parameters for commonly used protocols (HTTP, IRC, DNS, SMTP) + custom protocols
- Two groups of experiments
  - Offline
  - Incremental learning

Offline Experiments

Sample	Category	Containmnet	Normalization	Traces
W32/Virut	IRC Botnet	FULL	NO	15
PHP/PBot.AN	IRC Botnet	FULL	NO	12
W32/Koobface.EXT	HTTP Botnet	72%	YES	9
W32/Agent.VCRE	Dropper	FULL	NO	23
W32/Agent.XIMX	Dropper	FULL	YES	10

Incremental Learning Experiments

Sample	Category	Runs	Containment	Normalization
W32/Banload.BFHV	Dropper	23	FULL	NO
W32/Downloader	Dropper	25	FULL	NO
W32/Troj_generic.AUULE	Ransomware	4	FULL	NO
W32/Obfuscated.X!genr	Backdoor	6	FULL	NO
SCKeylog.ANMB	Keylogger	14	FULL	YES

Results

- Tested samples: 2 IRC botnets, 1 HTTP botnet, 4 droppers, 1 ransomware, 1 backdoor and 1 keylogger
- Required network traces ranging from 4 to 25 (AVG 14)
- DNS lower bound (6 traces)
- On AVG the number of traces is reasonable (Polymorphism, packing techniques)

Limitations

- Protocol agnostic approach
  - ✓ Find a good trade-off
- Analysis of encrypted protocols is impossible
  - ✓ API level solution
  - ✓ MITM solution
- Malware with different behaviors (Domain flux)
  - ✓ Improve the training set
  - ✓ Protocol-aware heuristics

Use Cases

- Repeat the analysis after weeks/months
- Analysis of similar variations (polymorphic) of the same sample
- Provide network containment for privacy/ethical issues
- Analysis of sophisticated attacks (Stuxnet/SCADA systems)

The end

THANK YOU

graziano@eurecom.fr