# Introduction of CERT-Conix and tooling

CERT-Conix, BTG and Machoke

**Meeting TF-CSIRT - *September 21th 2017***

**Auteurs :**
**Robin Marsollier**

CONIX Security

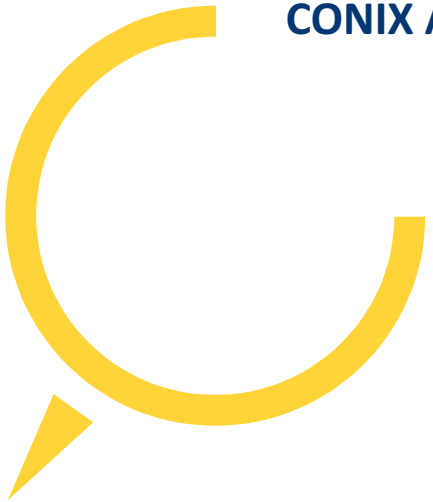
## Who am i

# Robin Marsollier

*robin.marsollier@conix.fr*

- CERT-Conix
- ArchLinux security team

# CONIX AND CERT-CONIX



## Identity

CONIX is a French company created in 1997

220 employees

## Operational organization

Transformation & Innovation (30 employees)

Risk & Regulation (30 employees)

Cybersecurity

(55 employees)

Digital Solutions (60 employees)

Business Intelligence (25 employees)

## INTRODUCTION OF CONIX CYBERSECURITY

Risk analysis
Cybersecurity Policy
Cybersecurity Organization
KPI & Dashboard

**Consulting**
Skill Center

**Audit**
Skill Center

Organizational audits
Technical audits
Penetration testing

**CERT**
Skill Center

**SOC**
Skill Center

Cybersecurity watch
Threat intelligence
Malware analysis
Forensics & Incident response

SOC building | BUILD
SOC services | RUN
SOC infrastructure MOC
SOC infrastructure MSC

## CERT-Conix

- Commercial CERT
- Accredited team
- Active in:
  - Threat intelligence
  - Cybersecurity watch
  - DFIR
  - Malware analysis
  - R&D
  - Etc.

## CERT-Conix : R&D

- **Tools published:**
  - BTG
  - Machoke
  - Bl2ru2
  - Zer0m0n
  - And more
- **Contributions to:**
  - MISP
  - Cve-search
  - Radare2
  - Etc.

**https://github.com/conix-security/**

## BTG

## BTG

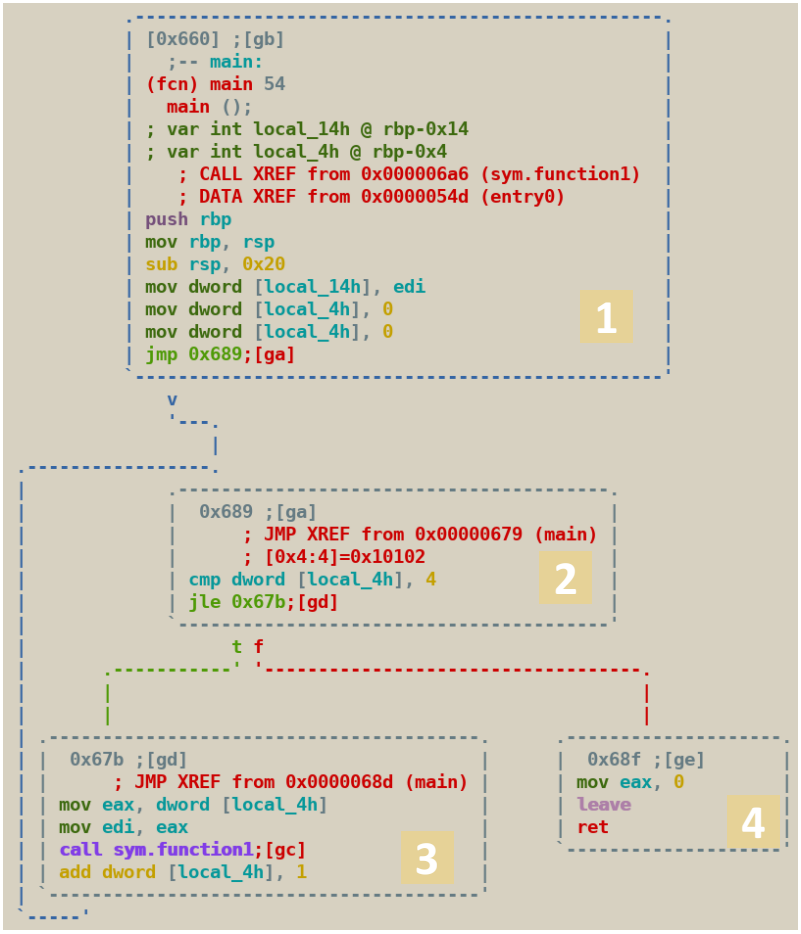
- CLI tool to quickly qualify an observable (usually found in SIEM or logs)
- Very useful for SOC and DFIR analysts
- Can handle:
  - IP
  - Domains
  - URL
  - Hashs

## MACHOKE

## Machoke

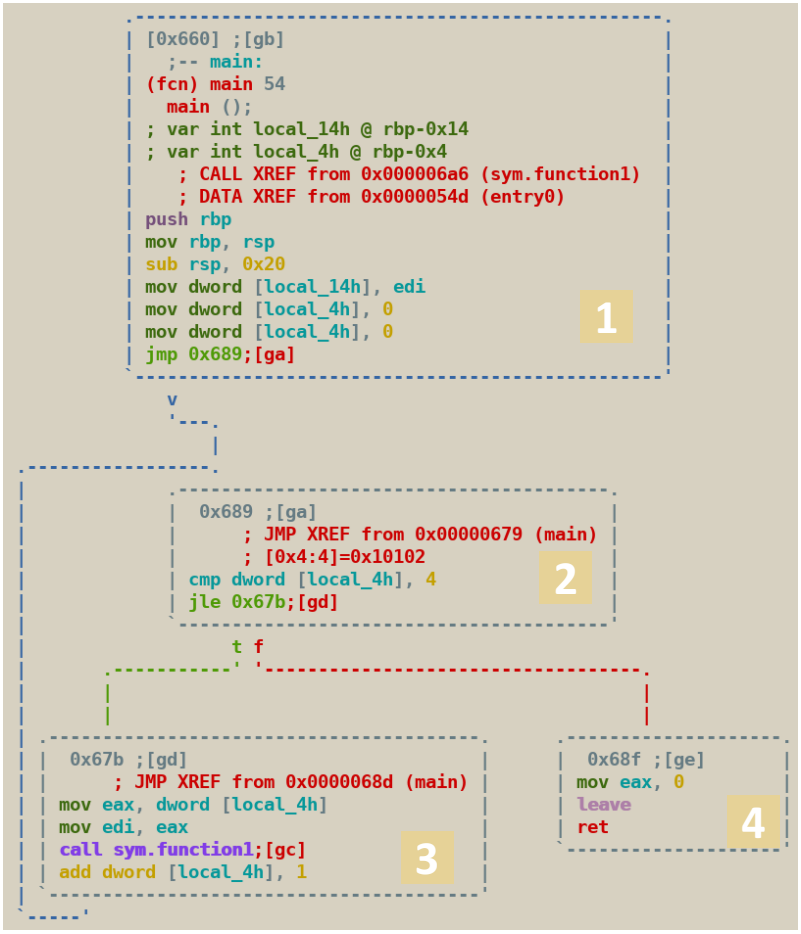
- CFG-based fuzzy hash for malware classification
- Designed to be easily correlated with other machoke
- Creates a hashs from the « tree » of jumps inside a sample

# Machoke
## Algorithm




1 . Label nodes/code blocks

# Machoke

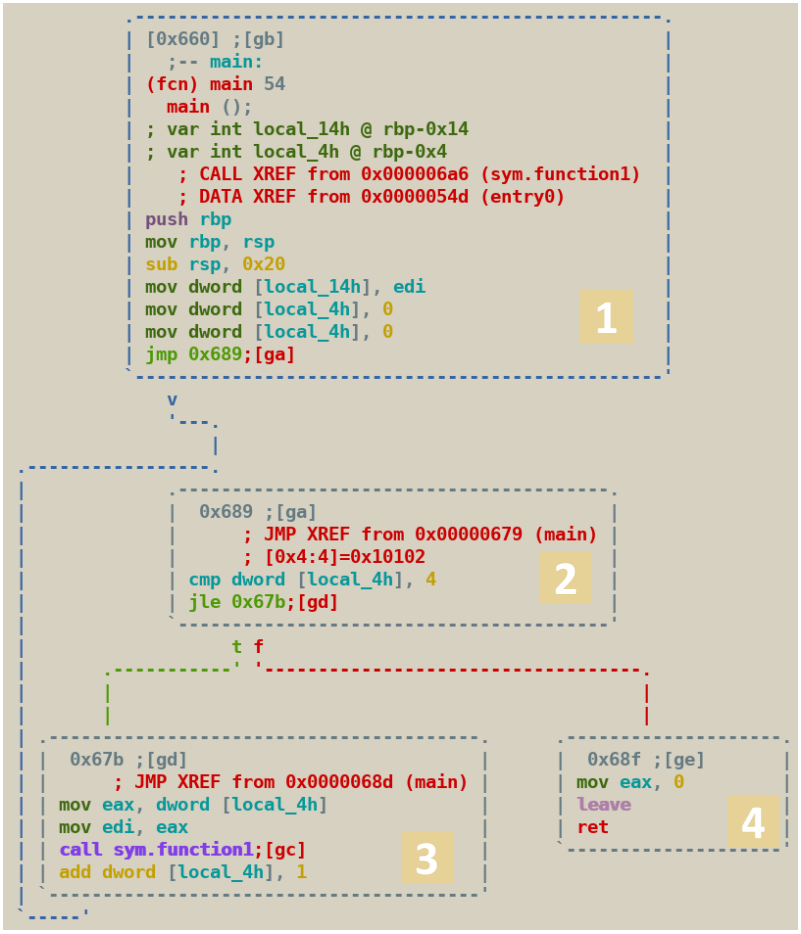



1 . Label nodes/code blocks

2 . Convert « tree » to a string :

1:2;

# Machoke

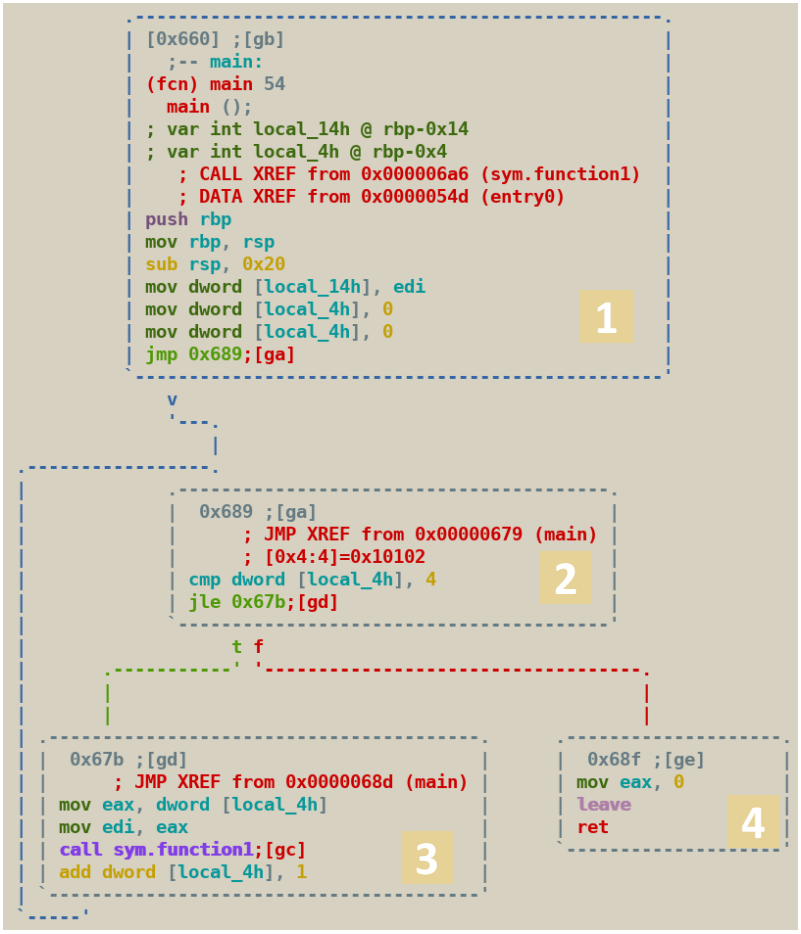



1 . Label nodes/code blocks

2 . Convert « tree » to a string :

1:2;2:3,4;

# Machoke

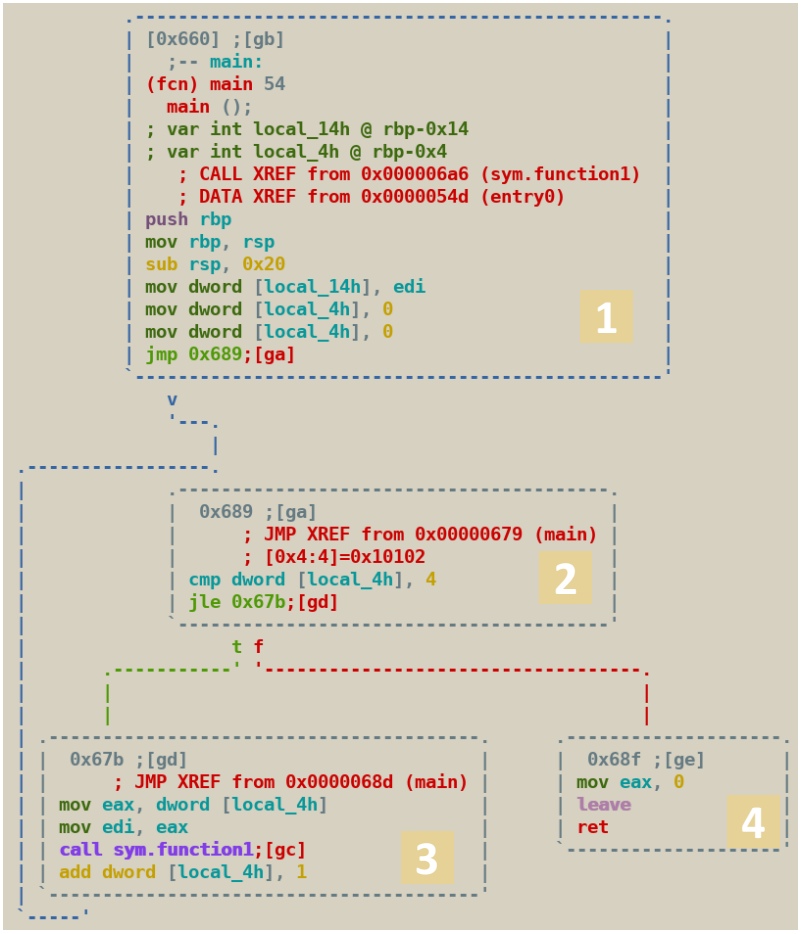



1 . Label nodes/code blocks

2 . Convert « tree » to a string :

1:2;2:3,4;3:c,2;4:;

# Machoke




1 . Label nodes/code blocks

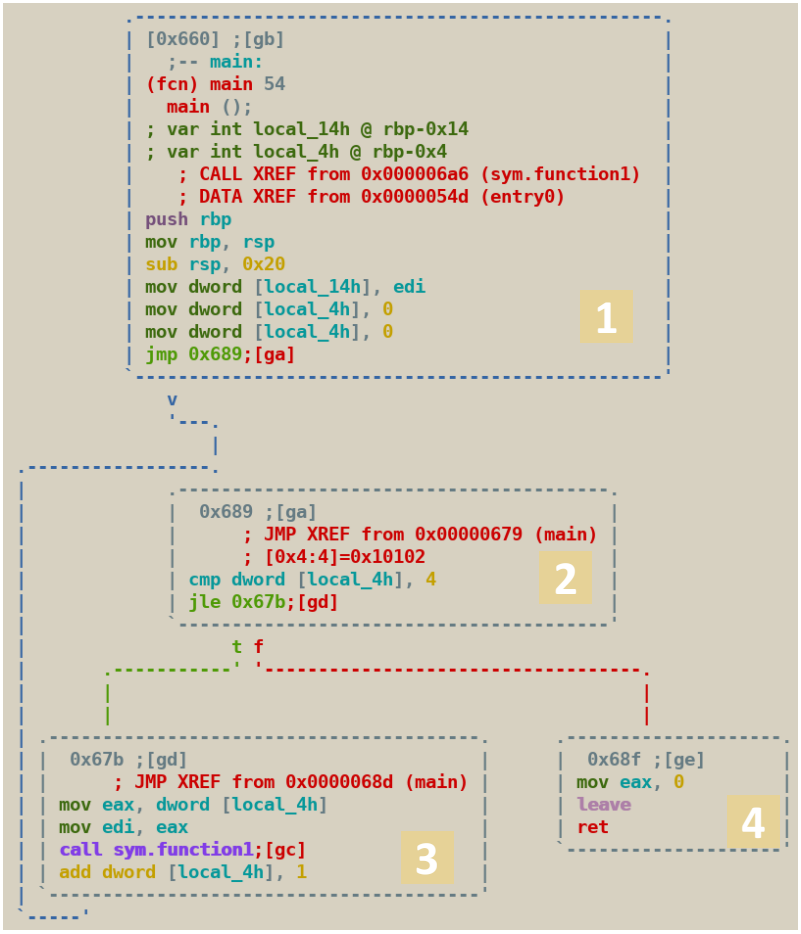
2 . Convert « tree » to a string :

1:2;2:3,4;3:c,2;4:;

3 . Hash this string

4 . Repeat for each functions and concatenate

# Machoke




1 . Label nodes/code blocks

2 . Convert « tree » to a string :

1:2;2:3,4;3:c,2;4:;

3 . Hash this string

4 . Repeat for each functions and concatenate

-> Variable size hash

-> If one function changes, only a small part of the machoke changes

-> easy to cluster

## Happy ending

**CERT-Conix**

*contact-cert@conix.fr*

https://github.com/conix-security/

conixsecurity.fr

@CONIXSecurity

# Questions ?

**Robin Marsollier**

*robin.marsollier@conix.fr*

@rbnctl