Introduction of CERT-Conix and tooling
CERT-Conix, BTG and Machoke

Meeting TF-CSIRT - September 21th 2017

Auteurs : Robin Marsollier
Who am i

Robin Marsollier
robin.marsollier@conix.fr

• CERT-Conix

• ArchLinux security team
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

Consulting Skill Center
Risk analysis
Cybersecurity Policy
Cybersecurity Organization
KPI & Dashboard

Audit Skill Center
Organizational audits
Technical audits
Penetration testing

CERT Skill Center
Cybersecurity watch
Threat intelligence
Malware analysis
Forensics & Incident response

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

DATE
CLIENT – Réponse à incident – Réunion d'ouverture
Document Confidentiel
CERT-Conix

- Commercial CERT
- Accredited team

- Active in:
  - Threat intelligence
  - Cybersecurity watch
  - DFIR
  - Malware analysis
  - R&D
  - Etc.
Tools published:
- BTG
- Machoke
- Bl2ru2
- Zer0m0n
- And more

Contributions to:
- MISP
- Cve-search
- Radare2
- Etc.

https://github.com/conix-security/
• 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
CLIENT – Réponse à incident – Réunion d’ouverture

Document Confidential

DATE

MACHOKE
Machoke

• CFG-based fuzzy hash for malware classification

• Designed to be easily correlated with other machoke

• Creates a hashes from the « tree » of jumps inside a sample
1. Label nodes/code blocks

```assembly
[0x660]; [gb]
; -- main:
    (fcn) main 54
     main ()
    ; var int local_14h @ rbp-0x14
    ; var int local_4h @ rbp-0x4
    ; CALL XREF from 0x000006e6 (sym.function1)
     ; DATA XREF from 0x0000054d (entry0)
     push rbp
     mov rbp, rsp
     sub rsp, 0x20
     mov DWORD [local_14h], edi
     mov DWORD [local_4h], 0
     mov DWORD [local_4h], 0
     jmp 0x689; [ga]

0x689; [ga]
     ; JMP XREF from 0x00000679 (main)
     ; [0x4:4]=0x10102
     cmp DWORD [local_4h], 4
     jle 0x67b; [gd]

0x67b; [gd]
     ; JMP XREF from 0x0000068d (main)
     mov eax, DWORD [local_4h]
     mov edi, eax
     call sym.function1; [gc]
     add DWORD [local_4h], 1

0x68f; [ge]
     mov eax, 0
     leave
     ret
```
Machoke

1. Label nodes/code blocks
2. Convert « tree » to a string :
   1:2;
1. Label nodes/code blocks
2. Convert « tree » to a string:
   1:2;2:3,4;
1. Label nodes/code blocks

2. Convert « tree » to a string:
   1:2;2:3,4;3:c,2;4;
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

Questions ?

Robin Marsollier
robin.marsollier@conix.fr
@rbnctl

CERT-Conix
contact-cert@conix.fr
https://github.com/conix-security/
conixsecurity.fr
@CONIXSecurity