With Machoc and Victorious Weapons

*Using CFG hashing for the lazy reverser*

*Présenté le* : 13 / 02 / 2020  
*Pour* : Bière Sécu - Lyon  
*Par* : Tristan (@contact_out)
Introduction
Introduction

1. Introduction
2. With Machoc
3. and Victorious Weapons
4. Conclusion
With Machoc
What is Machoc?

- First generation pokemon
- *It can hurl around 100 adult humans before it gets tired*
- Not really the subject of this talk
What REALLY is Machoc?

### CFG hashing algorithm
- Original idea by Stefan Le Berre (*Heurs*)
- Designed for helping during malware analysis @ ANSSI
- Presented at SSTIC in 2016

### Objectives
- Fast to calculate
- Resistant to small changes, (recompilation, C&C update, ...)
- *Le reste marche pas, donc on se sort les doigts et on le code*
Control Flow Graph

Figure: Typical CFG
Calculation - step 1

Figure: Simplified CFG
Calculation - step 2

Graph translation
For each block:
- Number of the block, and destination blocks
- Call instruction
- Example: 1:2,3; or 7:c,5,9;

Hashing
- Murmurhash of the graph string

Generalization
- Calculate for each function of the binary
- Concatenation of all tuples (addr, hash)
and Victorious Weapons
Clusterization

**Calculation**
- Calculate a jaquard distance between the machoc hashes of two samples
- If > 0.8, it’s a match!
- Group samples by links

**Identified problems**
- Some hashes must be blacklisted (ex: empty functions)
- Some false positives otherwise
Clusterization: APT1 example

Datasets
APT1 archive from ContagioDump

Figure: APT1 Clusterization: TARSIP
Bindiffing

**Identifying unchanged functions**
- If two functions have the same signature, they are probably the same.
- This behavior can also be used as a malware signature (e.g., unique function, etc).

**Identifying changed functions**
- Function 1 same hash == unchanged
- Function 2 different hash == changed
- Function 3 same hash == unchanged

=> Function 2 has probably changed
## Bindiffing: WEBC2-HEAD

### Samples
- One sample fully reversed
- One similar sample not reverse

### Results
- 3 functions renamed with a direct match
- 3 functions renamed with an indirect matched
- The new sample is almost entirely reversed!
Conclusion
Conclusion

- Machoc is cool
- Can be used in different contexts
- Complementary to other heuristics / tools
- According to twitter, I’m not the only one to use it
Bibliography

- Machoc original article
  https://www.sstic.org/2016/presentation/demarche_d_analyse_collaborative_de_codes_malveillants/

- Machoc spec
  https://github.com/ANSSI-FR/polichombr/blob/dev/docs/MACHOC_HASH.md

- Title inspiration
  https://www.youtube.com/watch?v=IyH0ub0gDbg

- Possible use case
AVEZ-VOUS DES QUESTIONS?

MERCI DE VOTRE ATTENTION

SYNACKTIV
DIGITAL SECURITY