


# Not another RET

YaraRET

- ▶ IT Security Analyst at  S2 GRUPO
- ▶ Honeypots & Forensics
- ▶ Interested in malware analysis (Linux/ARM)

Why am I up here?

## Why am I up here

- ▶ YaraRET - carving tool based in Radare2 & Yara, written in Go
- ▶ Show you why I realized I need this kind of tool
- ▶ Resolve a forensics case using YaraRET
- ▶ Show you what kind of problems I had to deal with

Why I need this tool?

## Why I need this tool

- ▶ Some forensics cases are based in unreliable evidence
- ▶ We have to deal with a raw disk, losing all flexibility of other tools provide
- ▶ I wrote a carving tool which solves some of this problems

# The Case



- ▶ It's FAKE, és fals, es falso, 这是假的, c'est un faux...
- ▶ It's a mixture of different cases I had to handle
- ▶ Realize together why I needed this tool



- ▶ A computer did some requests to an APT 33 related IP
- ▶ It's a critical PC, which had been working as a gateway of the industrial system
- ▶ Had access to critical information
- ▶ Industrial protocols

- ▶ ¿It's infected?
- ▶ What kind of information could have been exfiltrated

- ▶ Our client didn't do a good IR
- ▶ Computer is infected with generic malware
- ▶ Can't communicate with industrial system
- ▶ Spread with USB stick
- ▶ This malware is not related with APT 33
- ▶ Lot of industrial libraries
- ▶ We have no logs

The Tool

- ▶ Match a yara rule over a raw disk
- ▶ Extract the file which holds the offset
  - Simple job with Radare2
  - <http://radare.today/posts/carving-bins/>



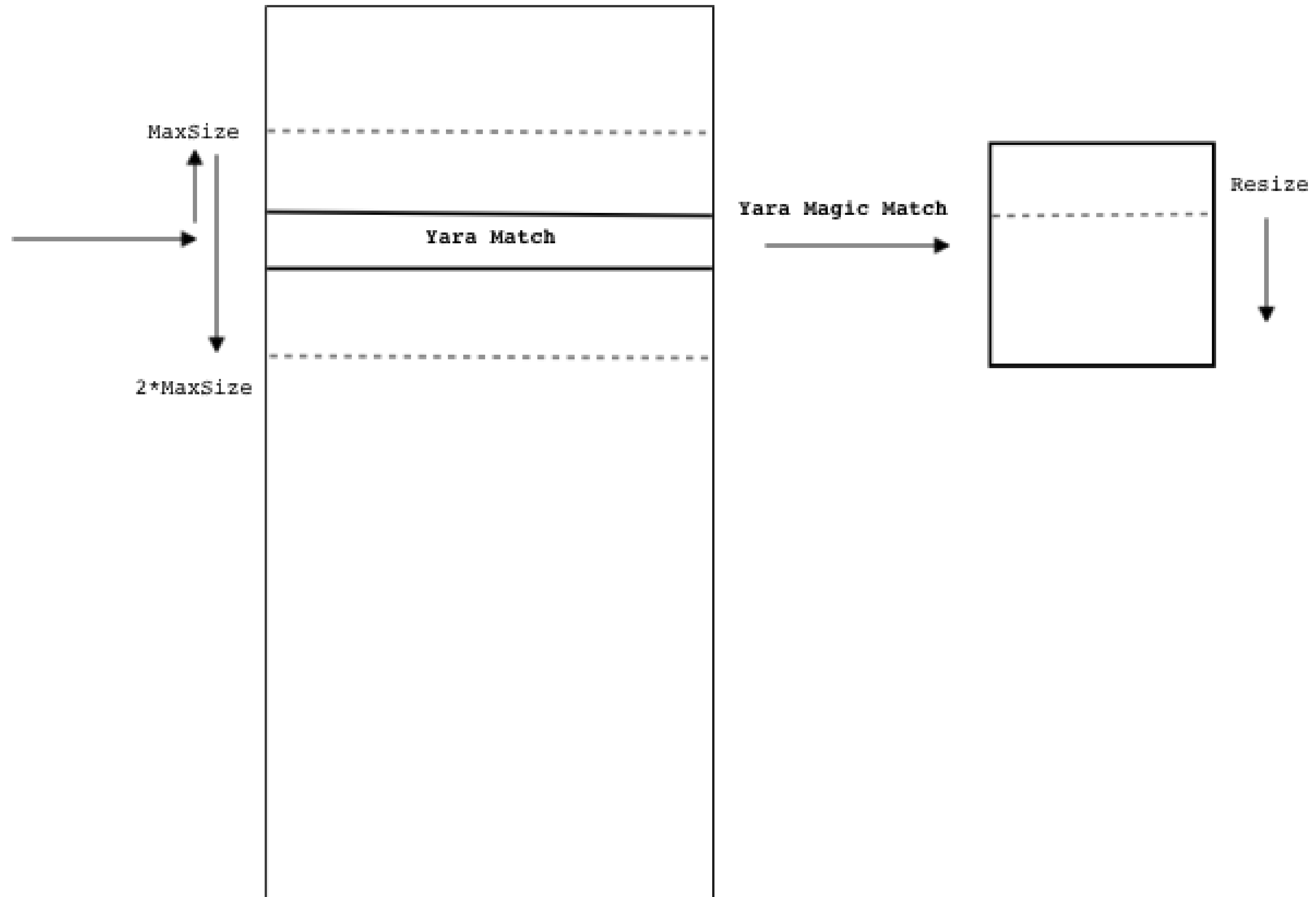
## Carving bins

JUNE 20, 2014

Radare was initially developed as a forensic tool. Nowadays most people use it for static code analysis or binary patching, but the framework and the tools still provide functionalities for analyzing disk partitions or filesystems..

In this post I'm going to explain how to use r2 to extract some ELF's files from a raw memory dump or unknown format firmware image.

# The Tool





- ▶ Done! We get a file
- ▶ It's related with TRISIS malware
- ▶ It's only a support library, we still can't know the extent of the breach



► What more do we have?

## What more do we have?



Solutions Services Partners

Home > FireEye Blogs > Threat Research > Attackers Deploy New ICS Attack Framework “TRITON”...

## Attackers Deploy New ICS Attack Framework “TRITON” and Cause Operational Disruption to Infrastructure

December 14, 2017 | by [Blake Johnson](#), [Dan Caban](#), [Marina Krotofil](#), [Dan Scali](#), [Nathan Bru](#)

### Introduction

[Mandiant](#) recently responded to an incident at a critical infrastructure organization where attackers deployed malware designed to manipulate industrial safety systems. The targeted system had the capability for industrial processes. We assess with moderate confidence that the attackers had the capability to cause physical damage and inadvertently shutdown operations. TRITON, is an attack framework built to interact with Triconex Safety Instrumentation. The attackers have not attributed the incident to a threat actor, though we believe the attackers are in the state preparing for an attack.

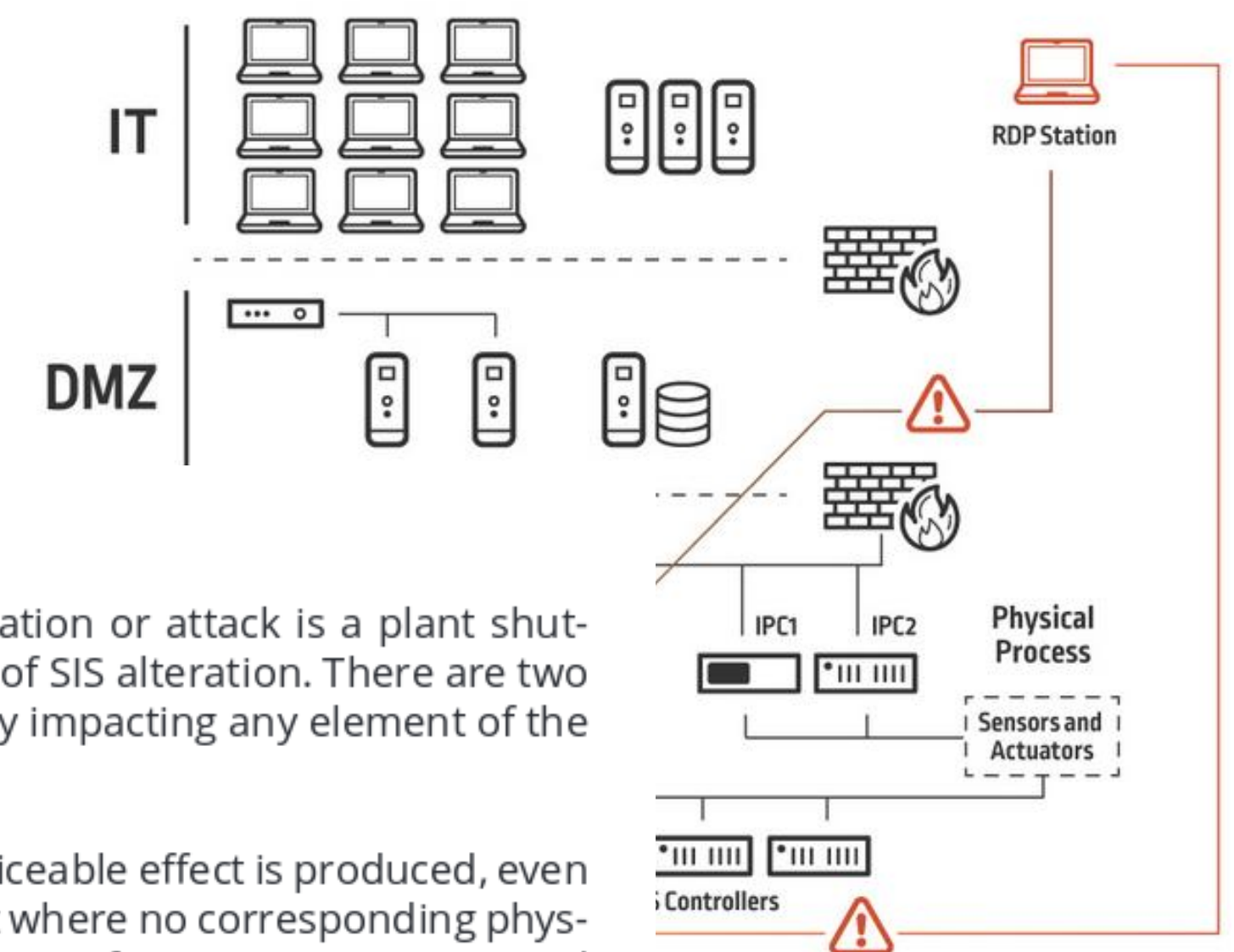
### Attack Scenario #1: Plant Shutdown

The most likely and operationally easy impact scenario from SIS manipulation or attack is a plant shutdown – and not necessarily due to follow-on physical damage as the result of SIS alteration. There are two general methods of achieving an operational ‘mission kill’ without physically impacting any element of the target environment:

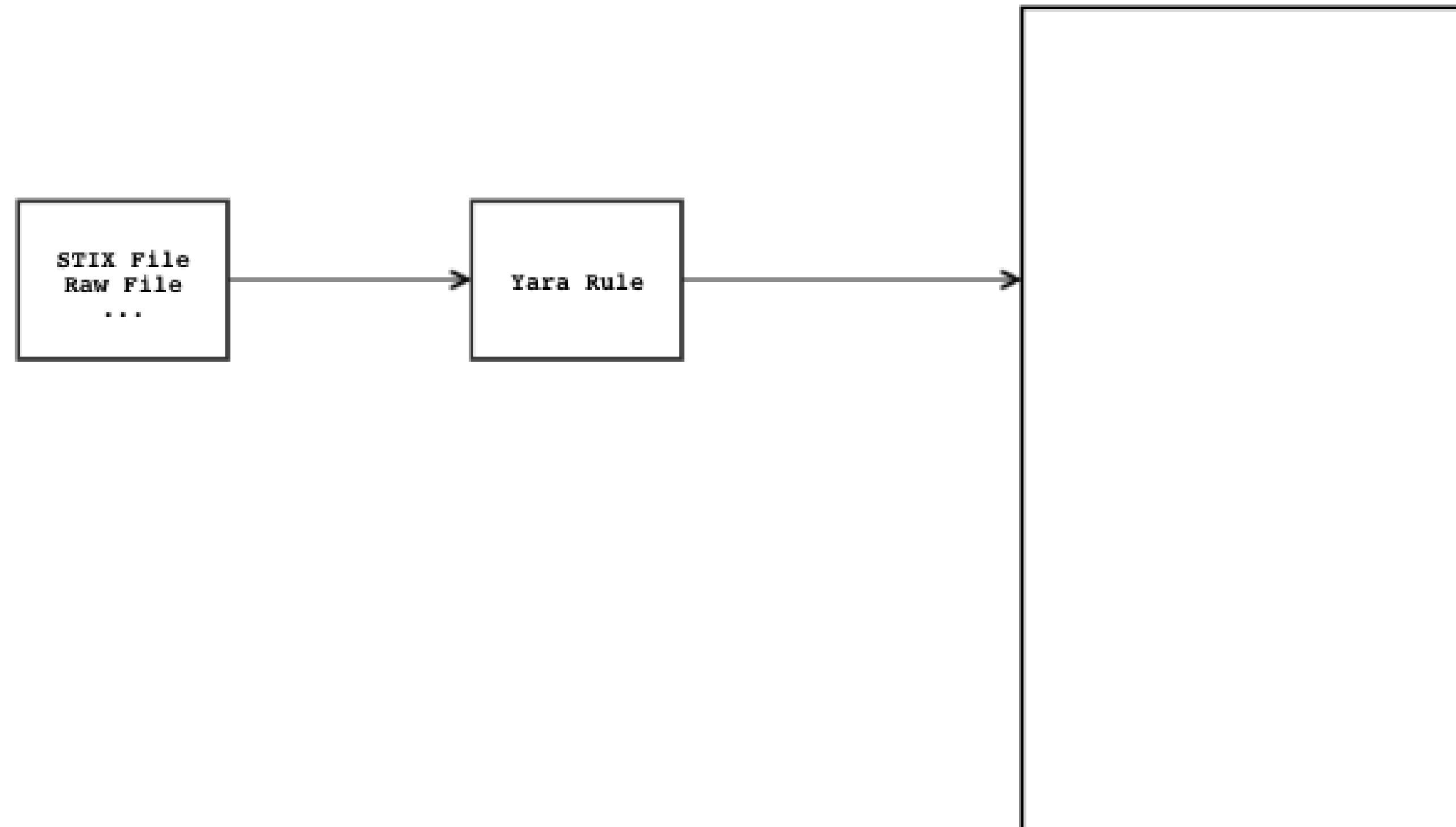
1. Create operational uncertainty. By altering an SIS where some noticeable effect is produced, even if only recognizing a configuration change or tripping a safety fault where no corresponding physical condition is observed, doubt is introduced into operations as to safety system accuracy and reliability. While the problem is investigated and troubleshooting takes place, operations will likely be significantly reduced if not outright stopped.
2. Trip safety ‘fail-safes’ to halt operations. Changing underlying logic to enter safety-preserving conditions during normal operations can trip SIS-managed equipment to enter ‘fail-safe’ modes when such conditions are not actually present. This will lead to a likely halt or stop to the affected process, and likely bring about a much longer shutdown as this scenario rapidly transitions to the item outlined in no. 1 above due to extensive troubleshooting.

RATs are computer programs designed to provide attackers with complete control over the victim’s system. They can be used to steal sensitive information, spy on victim’s system, and ultimately remotely control infected devices.

### How the Triton RAT Made its Way to the Tricon Engineering Station



- ▶ Network IoC (Domains, IPs, URIs...)
  - Parse openioc and generate rules for looking for them





- ▶ TRISIS malware
- ▶ It's able to communicate with industrial systems
- ▶ It's only a support library

Improve the tool

- ▶ The priority was the speed
- ▶ Spending more time at the beginning...
  - Complex analysis
  - Flexibility
  - Correlation

Shell Mode



- ▶ Run magic numbers over all disk
- ▶ Fields for defining each structure
  - Filetype
  - Index
  - Header's offset

- ▶ Start/Select filetype
- ▶ Run Yara
- ▶ Get Hash
- ▶ Run Radare2



- ▶ Starting all filetype info of 300 Gb ~ 30 min
- ▶ Starting single filetype 300 Gb ~ 15 min

- ▶ Starting all filetype info of 300 Gb ~ 30 min
- ▶ Starting single filetype 300 Gb ~ 15 min
- ▶ Save/Open Information



Back to the case

- ▶ As we know, TRISIS is based in pyc libraries
- ▶ We have a lot legit files
- ▶ Those files have a company's signature



- ▶ OK, we got pyc malware, but we still can't find trilogy.exe or similar
- ▶ Run yara for generic malware behaviour





- ▶ We got the binary, so why don't use it?
- ▶ And...

# The Case

- We got the binary, so why don't use it?
- And...



► SSdeep generation and distance checking



Case closed

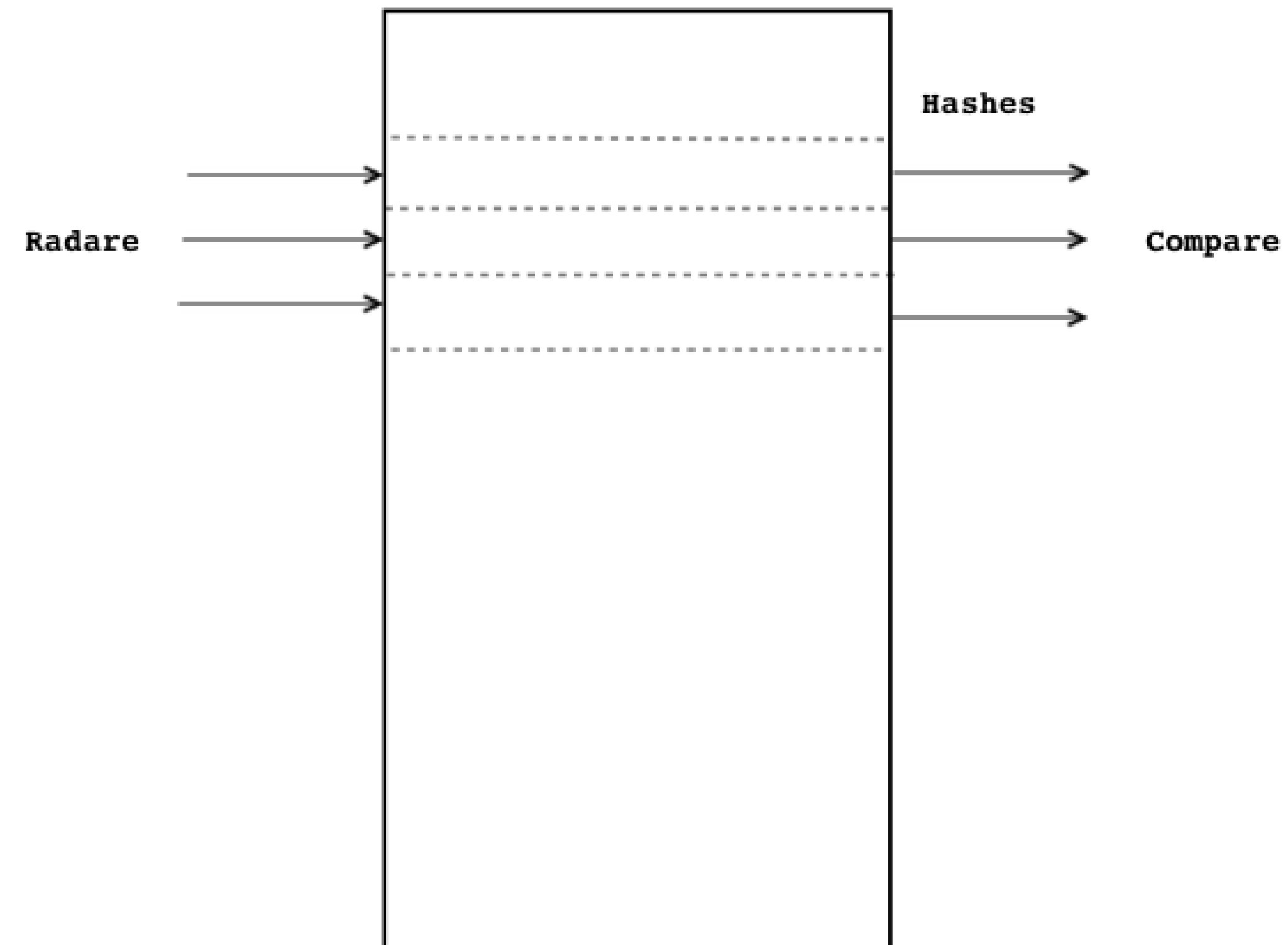
Let's see another features

- ▶ VirusTotal API integration
  - Check for a hash, ioc, upload a file...



## ► Hashes

- Generating hashes with Radare2 for every file of the selected file type





► Boot sector as a filetype



## ► Yara Forensics

- ./magicnumbers/
- Currently, YaraRET only supports 60 file types
- Yara Forensics it's a repo owned by Xumeiquer which contains some magic numbers
- <https://github.com/Xumeiquer/yara-forensics.git>



### ► Scripting

- If an analysis is going to take too much time, we can run a simple script with YaraRET commands
- Setting history var, every command is saved at `./yaraRET_{unixtime}`
- With run option, we can execute those scripts



### ► Upload files

- Radare2 is able to send data through TCP
- YaraRET uses Radare2 to send selected structures



Issues

- ▶ Still have failings in footer detection
- ▶ Some files have embedded files in them
- ▶ Memory management



- I'm not a developer so... code could be improved



Future work

## Future work

- ▶ Improve memory management
- ▶ Improve footer detection
- ▶ More magic numbers
- ▶ Improve algorithm
- ▶ Check information with MFT

# Where?

► At my repo

— <https://github.com/wolfvan>



@w0lfvan



@joanbt1

- ▶ Do you have an idea?
- ▶ Would you like to help?



Thanks !

