RawSec

Introduction to WHIDS, an Open Source Endpoint Detection System for Windows

Github / Twitter: 0xrawsec

Project: https://github.com/0xrawsec/whids

RawSec

Outline

1. Introduction to WHIDS
2. Writing rules: Gene Hands On
3. WHIDS Possible Deployments
4. Installation and feature exploration
5. Case Study: choose a case study and write your own detection rule(s)

Introduction

?I ma ohw

Freelance Security Consultant working in Luxembourg, running for my own company

- Originally doing Incident Response, digital forensics, malware oriented digital forensics …
- I start to accumulate a lot of knowledge (relatively to what is available online) about Sysmon
- Also Open-Source developer (in my free time) mainly Go, C, Python. At the origin of several projects:
  - Golang-evtx
  - Golang-misp
  - Gene
  - WHIDS

Doing other stuffs as well: software RE, bug hunting ...

What ?

Stands for: Windows Host IDS (even though it is more than just an IDS)

To be more accurate, it **combines** IDS features with detection based Incident Response Capabilities.

WHIDS strongly relies on the existence of **Microsoft Sysmon** since most of its nice features are built on to of Sysmon events

Features:

- **Correlate** Windows Event on host
- **Detect** in real time suspicious events (raw/correlated) based on user defined rules
- **React** to the detection:
  - Dump files
  - Dump process
  - Dump registry
- Can send all the information collected to a central point (a.k.a **manager**)

Why ?

I want people who cannot afford expansive solutions (EDR, SIEM …) to have something:

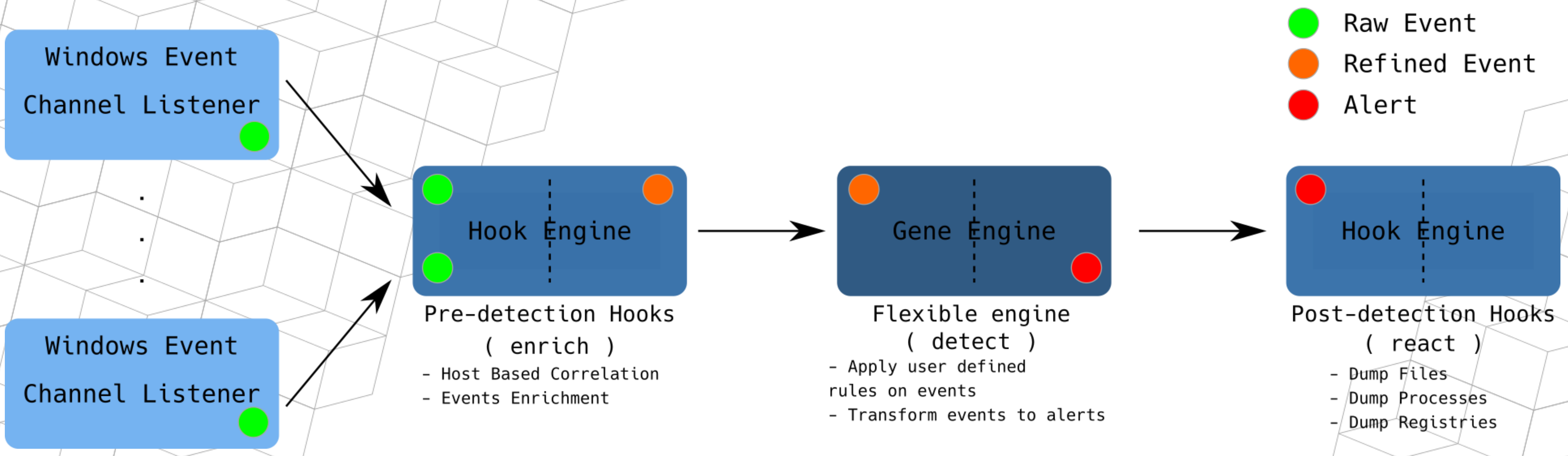
- They can craft detection rules specific to their environment

  Spoiler Alert: vendors often sell generic products, in the end not customizable as you would like it to be. May be it can be customized … but you will have to pay ☺

- That scales
- Which can also be plugged in with the other open source tools they are using

I also want to save time to analysts and allow them to have the data collected in real time

How WHIDS Engine Works



🟢 Raw Event
🟠 Refined Event
🔴 Alert

Windows Event Channel Listener

.
.
.

Windows Event Channel Listener

Hook Engine

Pre-detection Hooks
(enrich)

- Host Based Correlation
- Events Enrichment

Gene Engine

Flexible engine
(detect)

- Apply user defined rules on events
- Transform events to alerts

Hook Engine

Post-detection Hooks
(react)

- Dump Files
- Dump Processes
- Dump Registries

Hook: a function that takes a Windows Event as input and process it either to enrich it or to take information from it to enrich future events

NB: you can listen on absolutely any Windows Event Log channel you want and create detection rules for those

Few Words about Gene

Gene is the detection engine of WHIDS so I need to explain you what it is.

Gene is at the origin of everything…

- **What**: an engine and a rule format designed to detect patterns in Windows Event Logs. It was developed **prior to WHIDS** for Incident Response purposes.
- **Why**: any Windows Event can be considered as an **IOC** so it make sense to have a tool / rule format, to catch them

You can see it as a Yara engine but to match against Windows Event Logs

https://github.com/0xrawsec/gene

Writing Rules: Gene Hands On

Exercises 1.X

Methodology

1. Identify a possible entry point event to look at
   - A process created
2. Follow the track of events from the entry point and look for suspicious (requires Sysmon and a bit of OS knowledge)
   1. At this point it might happen you did not find anything of interest
   2. Dig a bit around the time of the entry point event
      - Look at process created, files created …
   3. Go back to step 1 taking the new event(s) you found as entry point(s)
3. Create rule(s) for the suspicious event(s) you have found
4. Verify your rules against a bunch of known good events (a reference WHIDS trace recorded from a clean system)
5. If previous steps reported some events, it means you have to refine your rule in order to avoid false positives

The toolset

Sorry for you guys, I am not using any GUI for that

- Use **sysmon-search.py** to filter in/out some events. You can find it in the exercises folder
- Use **jq** for visualization and advanced filtering
- A text editor (this one may have a GUI actually) to write the rules
- Gene executable to verify the rules

WHIDS Possible Deployments

RawSec

Standalone Deployment

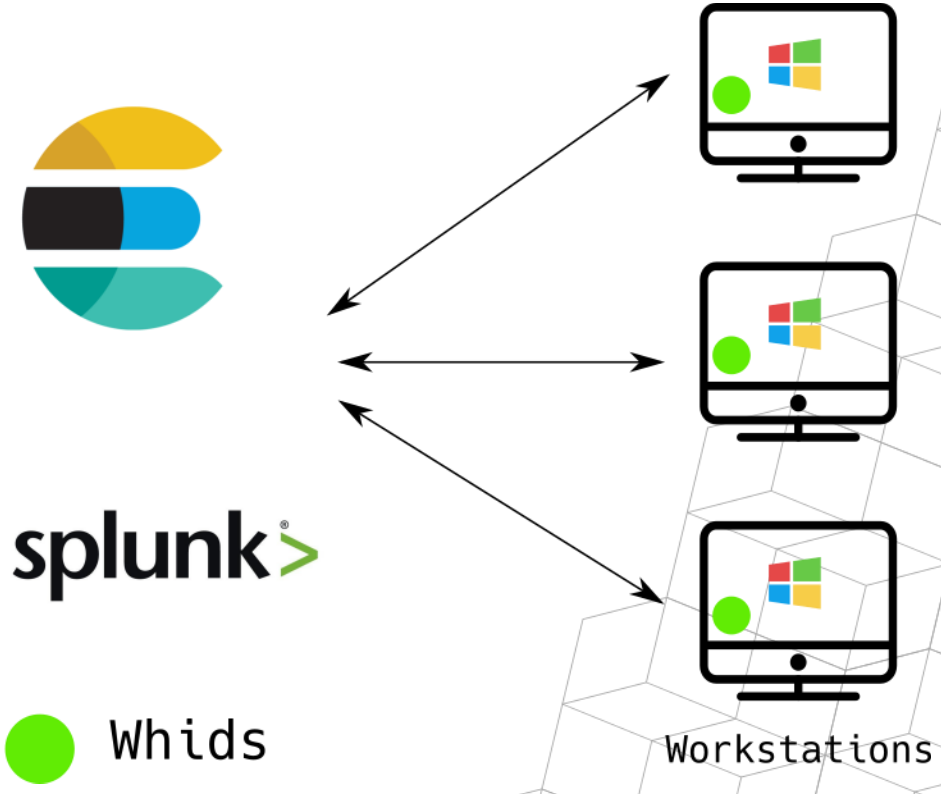
- Installation of WHIDS on each endpoint
- Log collection done directly on the endpoint

Pro:

- Solution for a single machine

Cons:

- Difficult to manage several machines
- Don’t benefit of manager centralization



Centralized Management

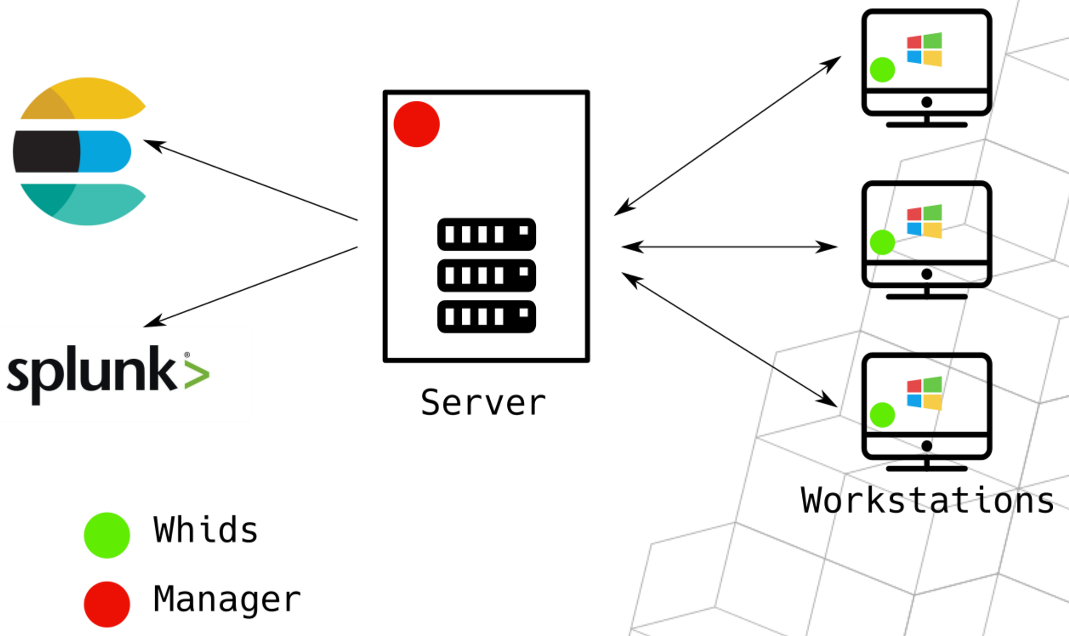
- WHIDS is installed on every endpoint
- All entities are managed centrally

Pros:

- Single point to update rules / containers
- Single point to collect logs from
- Maximizes amount of logs which can be analyzed

Cons:

- Rules / containers are pushed on endpoints



Server

Workstations

● Whids

● Manager

WEC Deployment

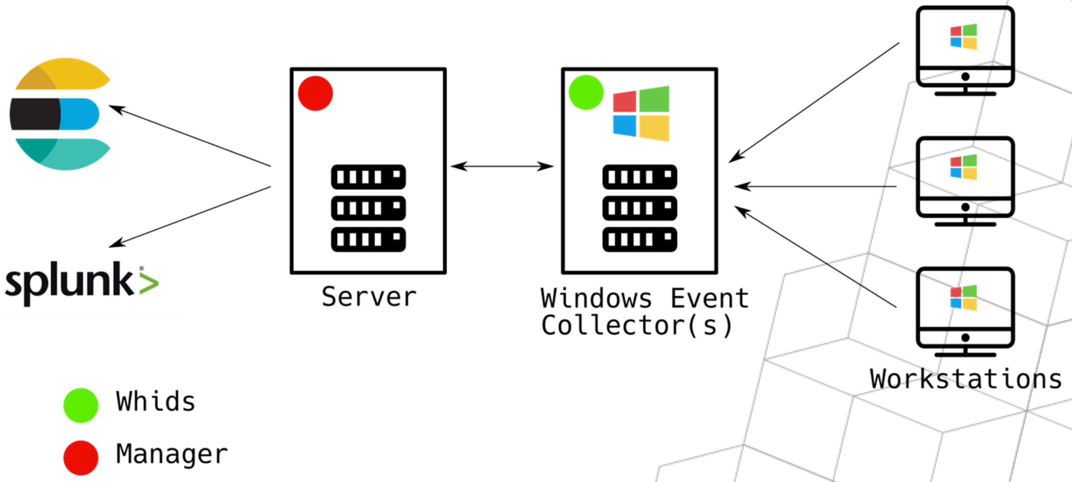
- Endpoints use Windows protocol to send logs to WEC(s)
- WHIDS is installed on WEC(s)
- If only one WEC can run without manager

Pros:

- Rules / containers not on endpoints
- Non invasive deployment

Cons:

- Cannot benefit from the same correlations as it is on endpoint
- Cannot benefit from artifact collection features (I have an idea for a workaround though ☺)



Installation and Feature Exploration

Manager Installation

We are going to cheat, instead of installing the manager on a remote machine, we will install it on the local machine but under WSL (Windows Subsystem for Linux) so simulate a Linux server.

Manager Installation:

- Generate TLS certificate and key for server
- Modify the configuration file to make it listen on 127.0.0.1
- Add rules / containers you’d like to be pushed on the endpoint
- Start the manager and let it run

NB: the manager needs to be rebooted in case of rule / container updates

WHIDS Installation

We are going to install it with a central manager (no WEC).

Endpoint installation steps:

1. Install Sysmon
2. Install WHIDS with the help of **manage.bat**
   - Do not import rules shipped with project (we are going to pull them from the manager)
   - Do not start the services, we are going to configure stuff first
3. Edit configuration file to configure connection to the manager we have just set up
   - Do not forget to set **unsafe** to **true** under **manager-client** config (we have auto generated a TLS cert)
   - Do not forget to set **local** to **false** under **forwarder** config
4. Start the services and check if you see connections in your manager’s logs

Feature Exploration

- Explore dumping capabilities
  - File dumping
  - Process memory dumping
  - Registry dumping
- Alert forwarding capabilities: alerts are regularly forwarded to the manager
- On host log correlation
- MITRE ATT&CK integration
- Offline mode: even though configured with a manager the logs and dumps are never lost in case connection is lost.
- MISP IOC checks (left as homework)

Case Study

Exercise 2.X or whatever technique / malware you want to assess the tool with

RawSec

Thank you