RawSec

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

Github / Twitter: 0xrawsec

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

Outline

1. Introduction to WHIDS
   1. Generalities
   2. Possible Deployments
2. WHIDS Installation and feature exploration
3. Writing rules: methodology and practical exercises
4. Putting everything together: one case study of your choice will be given to you and the objective will be to write your own rule(s)

?I ma ohw

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

- Originally doing Incident Response, digital forensics, malware oriented digital forensics …
- 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**)

RawSec

Why ?

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

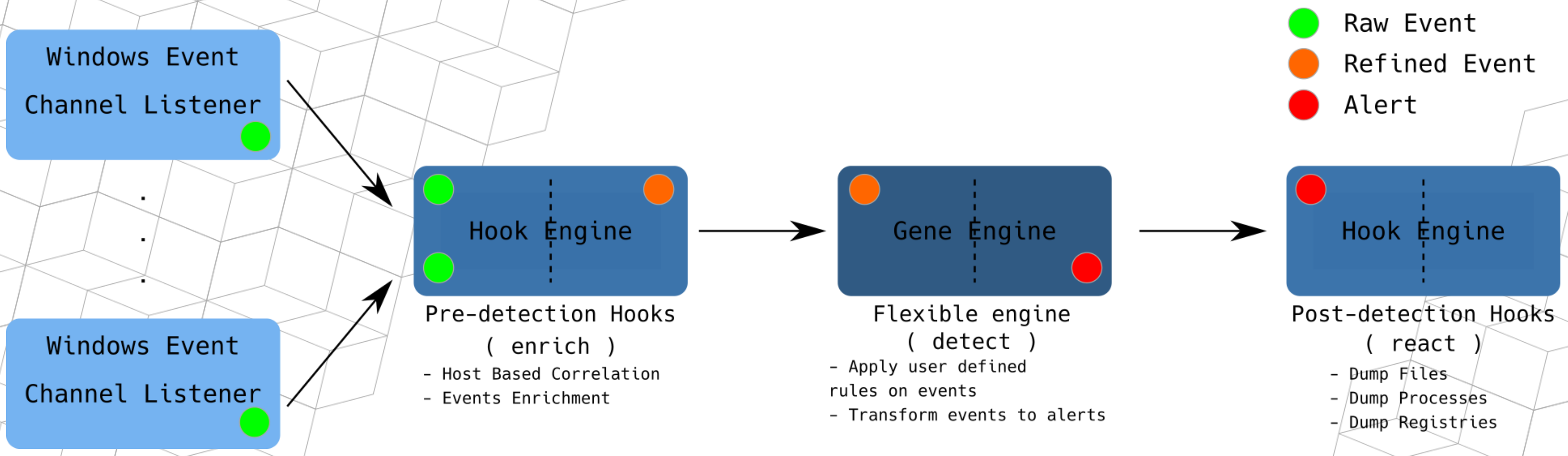
- They can craft detection rules specific to their environment

  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

RawSec

Gene Hands On

Exercises 1.X

Standalone Deployment

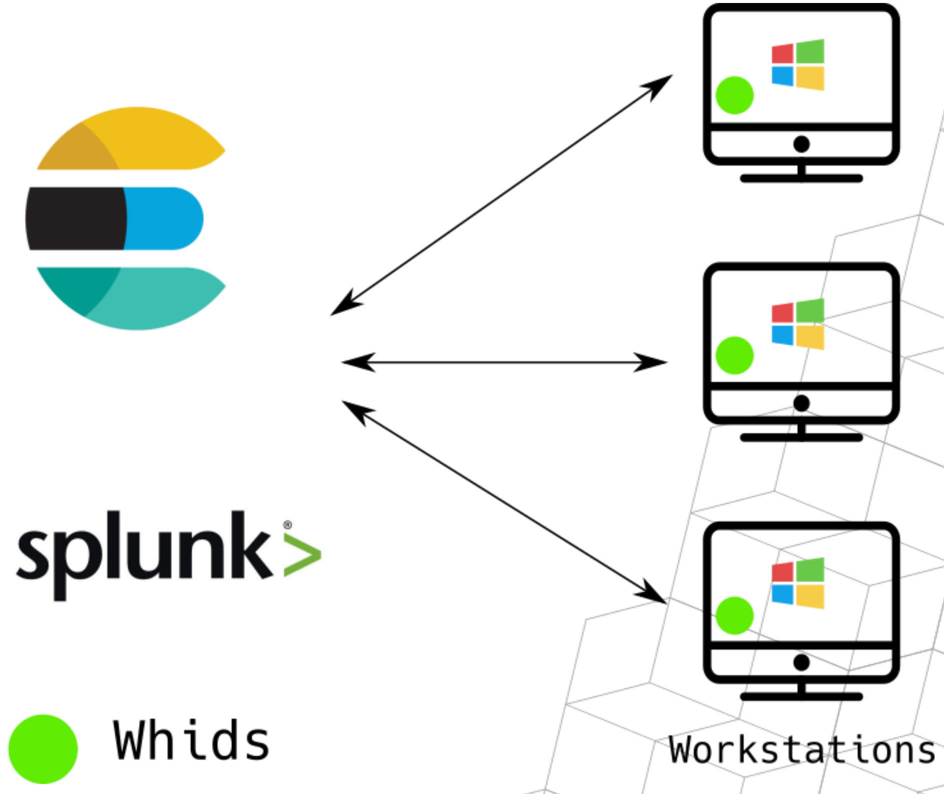
- Installation of WHIDS on each endpoint
- Log collection done directly on the endpoint

Pro:

- Solution for a single machine

Cons:

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



Centralized Management

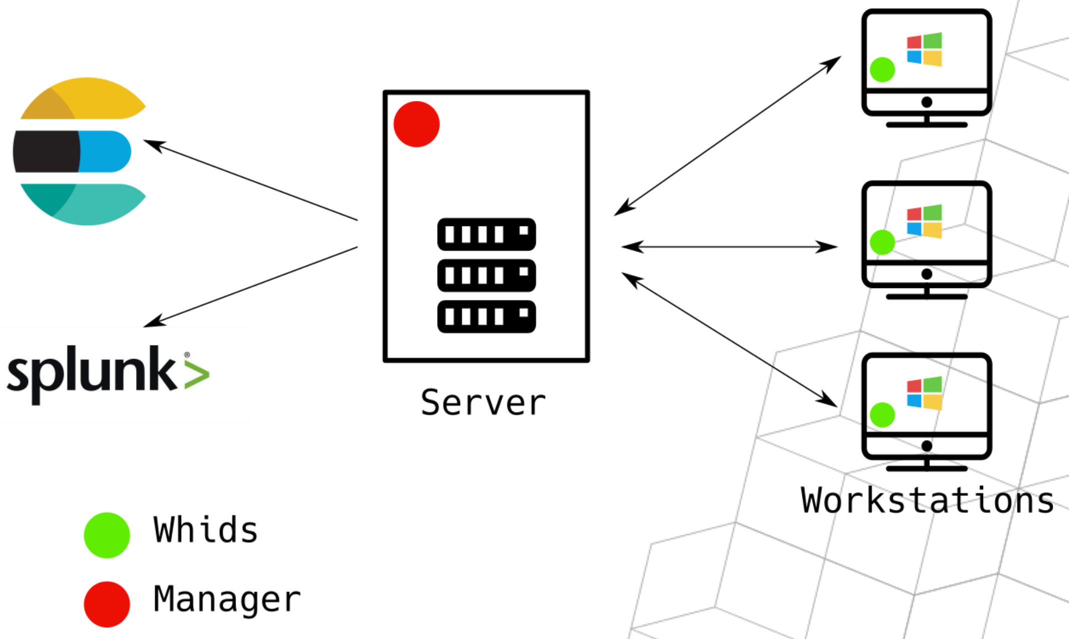
- WHIDS is installed on every endpoint
- All entities are managed centrally

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



WEC Deployment

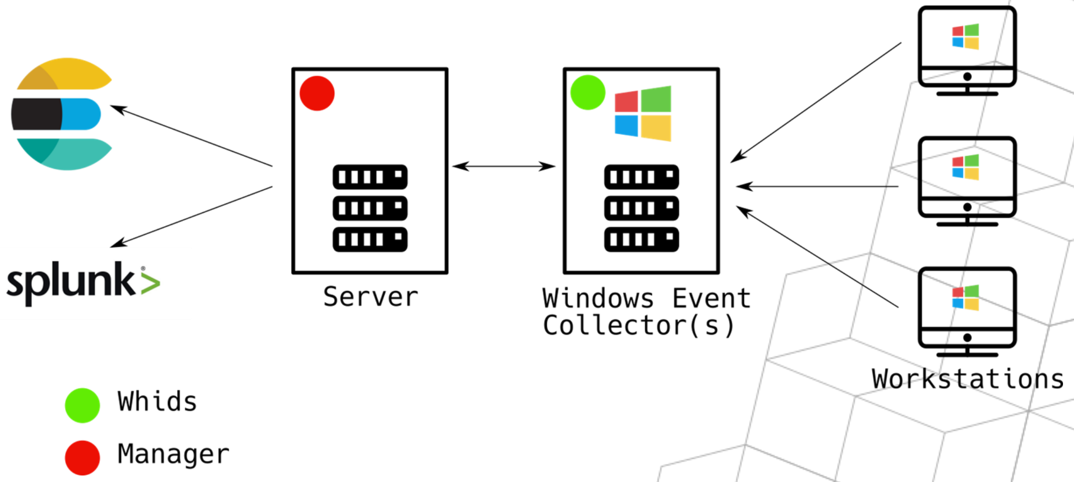
- Endpoints use Windows protocol to send logs to WEC(s)
- WHIDS is installed on WEC(s)
- If only one WEC can run without manager

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 ☺)



WHIDS Hands On

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 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

Features 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