Suricata File Extraction API

SuriCon 2016

Zach Rasmor

Lockheed Martin CIRT

@ZachRasmor

whoami

- ~2 year member of Lockheed Martin CIRT
- Software Engineer focusing on network sensors

Evolution of IDS

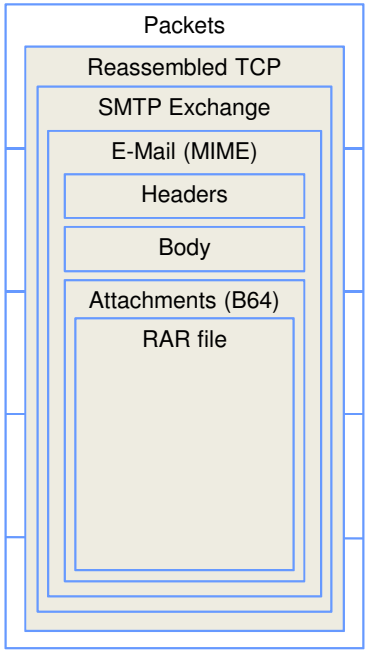
- **IDS has evolved to provide increasing insight into application layer data**
- **There has been a shift in frequency from network exploits to file/application exploits**
- **How does traditional network IDS (NIDS) address this threat vector?**
- **Most major IDS now support some level of file extraction to disk**

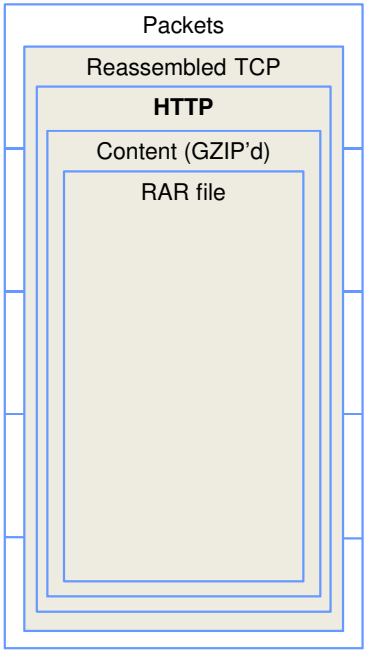
Value of External File Analysis

- **Adversaries have virtually limitless methods to obscure and embed malicious payloads**
- **Real-time constraints of NIDS make deep file analysis challenging**
- **Dedicated file analysis frameworks currently exist**
  - Laika BOSS (Lockheed Martin CIRT)
  - FSF (Emerson)
  - Stoq (Punch Cyber)

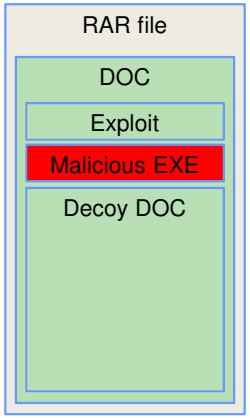
A File-Centric Approach

NIDS Reassembly

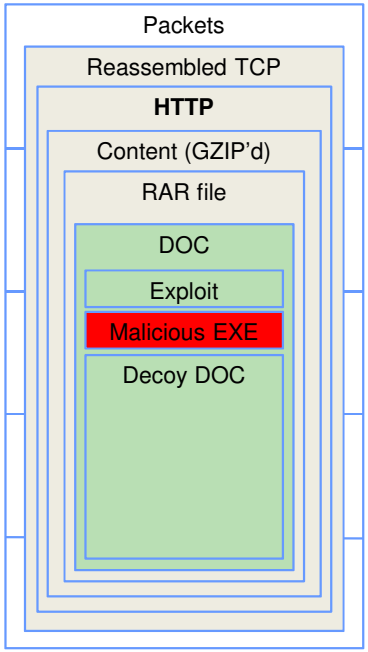




File Analysis



Can we leverage both?



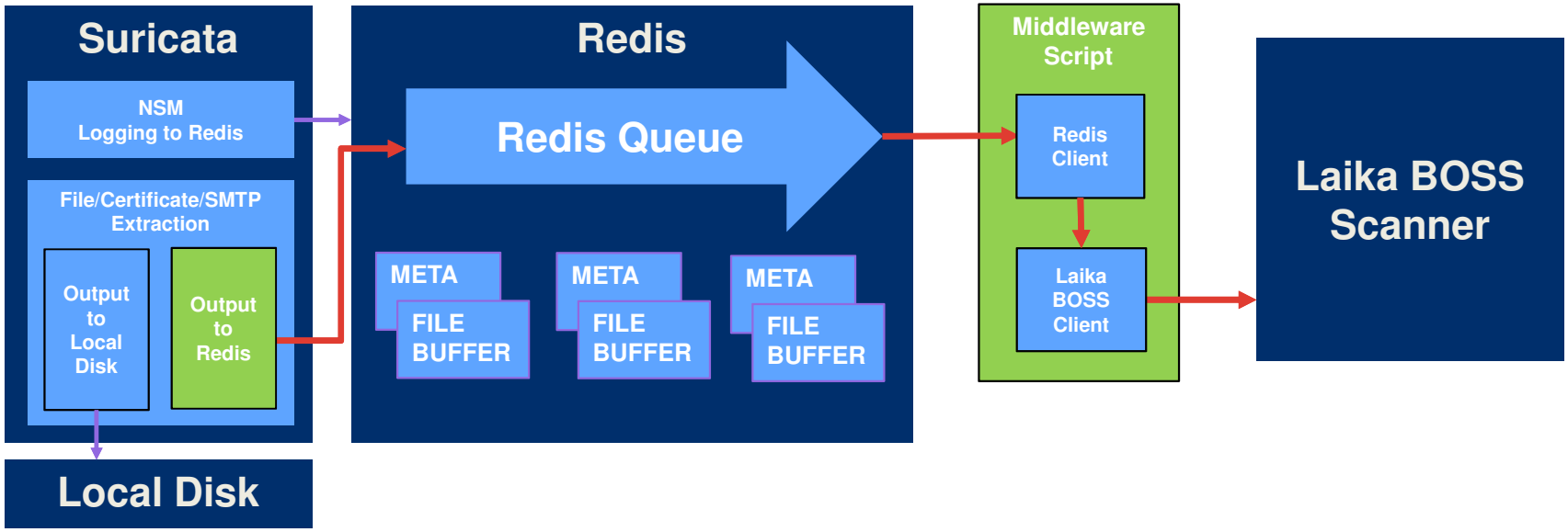
How can we get the best of both worlds?

- **Currently not a straightforward means to offload extracted files to file analysis frameworks**
  - Other attempts to solve this problem have had limitations
    - Example: running inotify on the file extraction directory

- **Can we leverage the file extraction framework currently built into Suricata?**

- **Our design goals:**
  - Generic interface
  - Minimally invasive
  - Fast
  - Tight coupling of metadata and files

Our Approach



Laika BOSS – Key Components

- **Input/Output: Client Library connects to Redis, Milter, ICAP, passive network sensor, malware analysis lab, etc.**

- **EXPLODE: Break compound objects into atomic elements**
  - Unzip, Unrar, etc.
  - OLE = Object Linking and Embedding
  - PDFs have multiple embedded streams

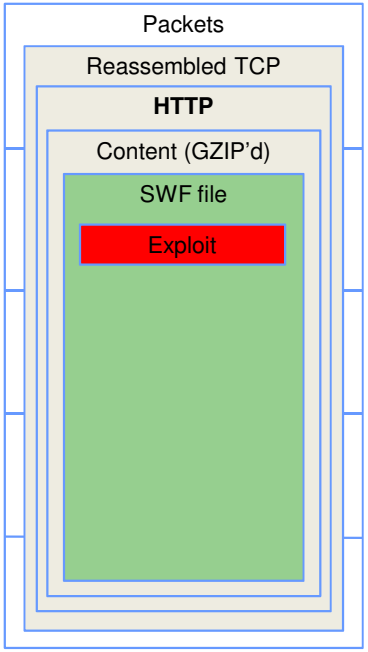
- **SCAN: Real-time inspection for badness. YARA, ClamAV, etc…**

- **META: Extract metadata at any depth for logging and future analysis (and anything else you can think of)**

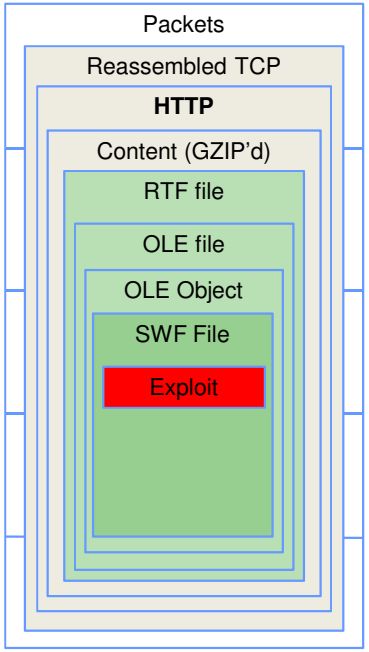
- **More info: https://github.com/lmco/laikaboss**

Demo Time!!!

PCAP #1



PCAP #2



Acknowledgements

- **Matt Arnao – Lockheed Martin CIRT**
  - Twitter: @mattarnao

- **Charles Smutz – Sandia National Labs**
  - Blog: http://smusec.blogspot.com/

Links

- **Suricata – Redis Prototype Branch**

https://github.com/lmco/suricata/tree/file_extract_redis_prototype_v1

- **Laika BOSS – Redis Middleware Script**

https://github.com/lmco/laikaboss/blob/master/laika_redis_client.py

Pros/Cons of Our Approach

- **Advantages**
  - Leverages current file extraction framework
  - Redis integration already supported for Eve logging
  - Straightforward to implement caching to limit duplicate work
  - Generic enough to be adopted by other IDS or File Scanning platforms

- **Disadvantages**
  - Tailored for passive scanning, not as straightforward to apply inline

/* TODO */

- **Create configurable parameters in suricata.yaml**
  - Redis Instance
  - Redis Queue Name
  - Redis Key Prefix
  - Hashing Algorithm
  - Key Expiry
  - Caching Enable/Disable
  - Caching Expiry
- **Apply similar concept to TLS Certificate extraction**
- **Ensure compatibility with future SMTP extraction**
- **Cleaner separation from FileStore module**
  - Register separate FiledataModule
- **Better handling of truncated files**
- **Unit tests**