Citrus: Orchestrating Security Mechanisms via Adversarial Deception

Ryan Mills
Lancaster University
Lancaster, England
r.mills2@lancaster.ac.uk

Nicholas Race
Lancaster University
Lancaster, England
n.race@lancaster.ac.uk

Matthew Broadbent
Lancaster University
Lancaster, England
m.broadbent@lancaster.ac.uk

***Abstract*—Despite the Internet being an apex of human achievement for many years, malicious activity and cyber attacks are becoming more prevalent than ever before. Large scale data collection using threat sources such as honeypots have recently been employed to gather information relating to these attacks. While this data naturally details attack properties, there exists challenges in extracting the relevant information from vast data sets to provide valuable insight and a standard description of the attack. Traditionally, threats are identified through the use of signatures that are crafted manually through the composition of IOCs (Indicators of Compromise) extracted from telemetry captured during an attack process, which is often administered by an experienced engineer. These signatures have been proven effective in their use by IDSs (Intrusion Detection Systems) to detect emerging threats. However, little research has been made in automating the extraction of emerging IOCs and the generation of corresponding signatures which incorporate host artefacts. In this paper we present Citrus: a novel approach to the generation of signatures by incorporating host based telemetry extracted from honeypot endpoints. Leveraging this visibility at an endpoint grants a detailed understanding of bleeding edge attack tactics, techniques, and procedures gathered from host logs.**

***Index Terms*—honeypot, threat intelligence, signature generation**

I. Introduction

The evolving threat landscape is a back and forth battle between malware authors incorporating innovative methods of infection and systems administrators repelling attacks by improving their detection systems. The defense mechanism of choice within enterprise network is usually an IDS, which typically performs pattern matching of behaviour observed inside the local network in comparison to known malicious signatures. Consequently, the effectiveness of an IDS is dictated by the accuracy of the signatures within its database. This method requires periodic signature updates in order to keep abreast of emerging threats.

Honeypots are systems under observation which contain components that masquerade as legitimate enterprise infrastructure in order to catch unsuspecting adversaries leveraging previously unobserved exploits, attack tactics and patterns used for infiltration [1]. Utilising these honeypots grants unrestricted access to emerging threat log data, which is otherwise extremely difficult for the wider community to access due to corporations limiting the exposure of breaches. The benefits of this approach relate to the fact that activity relating to these systems is extremely likely to be malicious as all communication is unsolicited. Therefore, data garnered from honeypots offers an invaluable source of signatures which can be utilised by an IDS to provide detection against emerging threats.

Recent literature [2]–[5] has identified the benefits of extracting attack characteristics from honeypots and generating signatures such that attacks in the same vein are prevented. While providing effective signatures for NIDSs (Network Intrusion Systems), this research does not consider events which transpire at a *host* level, and only provides signatures suitable for *network* based defense mechanisms. Gaining visibility at a host level when uncovering stages within an attack grants a greater understanding of malware behaviour and provides multiple IOCs which capture malicious authentication, registry, and process operations [6], which would otherwise be omitted if using traditional network telemetry, thus enabling each stage within an attack to be detailed and available to an IDS. The proposed approach in this work orchestrates the retrieval of emerging threat data from myriad sources to a centralised storage platform, where the data is then processed, analysed, and contextualised through requests to external services in order to craft malicious behavioural signatures which provide rapid defense measures of the latest attacks from a host perspective. In this paper, we present the design and implementation of Citrus, a novel honeypot signature generation framework which enables the identification and prevention of emerging threats by considering behavioural operations at a host level. The detection capabilities Citrus provides are discussed in the Evaluation section. This is achieved by Citrus orchestrating the subscription and digestion of information from a variety of sources and applying generated signature rules to a policy engine which prevents bleeding edge attacks causing compromise within the internal network.

II. Related Work

A. Threat Intelligence

Threat Intelligence refers to the behaviour and information derived from observation of threat sources. Research into this sphere attempts to extract IOCs to gain an understanding of attack properties so attacks of the same type can be prevented. In contemporary literature, this is often achieved by an evaluation of network activity initiated by malware.

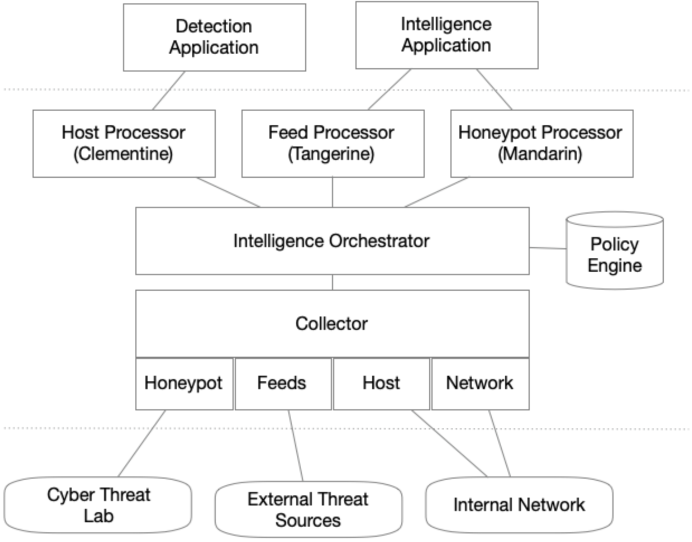
Vasilomanolakis et al. [5] used bespoke ICS (Industrial Control System) honeypots in order to generate signatures of multi stage attacks by modeling each disparate protocol from the same host as a separate stage in the attack. For each of these stages, a signature is generated based upon characteristics of the network packet involved in the attack which is then used by Bro IDS[1] to evaluate the detection capabilities. However, this approach lacks an understanding of activity from a host perspective. In order to tackle the most sophisticated threats, defenses which incorporate both host and network telemetry detection mechanisms have been proposed to provide greater accuracy and visibility [7]. To the best of our knowledge, there has been no research which generates and implements defense signatures via host activity garnered from a honeypot. For the purposes of integration into industry standard defense mechanisms, the generated host based signatures should be easily understandable, accessible and translatable to bespoke IDS and SIEM (Security Information and Event Management) search terms. Sigma[2], a generic signature format to describe log events, was chosen for this purpose. Sigma's utility is based upon its open and universal event description so that defense mechanisms which incorporate a host based understanding are able to compare internal behaviour to known malicious signatures. The focus of Citrus is the generation of host signatures based on event logs extracted from medium and high interaction honeypots as well as behavioural analysis of malware extracted from sandbox services.

B. Honeypot

Honeypots are systems emulating production service with the intention to attract adversaries who wish to infiltrate infrastructure. These type of systems can be loosely classified into two distinct types: medium-interaction and high-interaction. Medium-interaction honeypots aim to masquerade as a realistic service such as Telnet and provide an underlying emulated environment which allow attackers to further interact and capture exploits without them being successful in infiltration. High-interaction honeypots further add complexity as they deploy real operating systems and applications in which an attacker has the ability to completely gain access. While the risk of infiltration is much greater, the data gathered via logs allows a richer understanding of the extent of their behaviour.

The nature of honeypots allow observation of developing attack mechanisms which yield a wealth of detailed log information. In order for these systems to become useful in a production environment, these logs must be analysed to uncover specific characteristics of the attack. This is often undertaken by a network engineer who can interpret the results and administer policies to remedy the threat. In recent times, Elasticsearch[3] has proved a popular approach for the purpose of threat hunting [8], [10]. This approach enables a scalable environment in which search queries are utilised to provide an illustrated evaluation of intrusion attempts. While providing the means for rapid visualisation and attribution of threat behaviour across distributed data sets, this process relies heavily on expert analysis by a human actor which remains a slow and tedious process [2], [3]. Furthermore, implementation of prevention mechanisms is also a manual process in which the data mined characteristics of the attacks are used by systems such as a firewall. This is not suitable in cases of time-critical importance, where any delay in the implementation of mechanisms may allow unauthorised access to corporate assets. Hence the need for automated routines which digest this data and provide response in the form of accurate security measures [10].

Fig. 1. Architecture overview of Citrus and its components



III. Design

The aim of Citrus is to bolster existing IDS by providing an automated framework in which host based signatures of emerging attacks are generated. The architecture of Citrus is outlined in Figure 1. As depicted, the Tangerine module is responsible for the analysis of information derived from intelligence feeds. This process first digests data from the deployed low and medium interaction honeypots, then normalises this into a common format, and gathers further context about the attack via queries to external sources. The dynamic run time behaviour of executed malware is captured for entry into the policy engine so that internal hosts displaying signs of similar behaviour are rapidly identified. The Mandarin module within Citrus is responsible for the collection and processing of data emanating from the selection of high interaction honeypots. In the same ilk, Clementine orchestrates the retrieval of host log data from internal network hosts for correlation and comparison to signatures generated by Citrus.

Gaining insight into the methodology of malicious actors allows greater understanding of the way in which to detect them [8]. A number of honeypots have been deployed for this purpose within a research facility, the Cyber Threat Laboratory, located inside Lancaster University. These honeypots vary in the level of interaction and placement within the network topology in order to capture a range of potentially interesting

[1] Bro IDS https://www.zeek.org/
[2] Sigma https://github.com/Neo23x0/sigma
[3] Elasticsearch https://elastic.co

malicious activity. For example, deployment within a local network limits the scope of people able to interact with internal hosts. The plurality of these systems require centralised logging infrastructure capable of correlation between vast data sets. For this purpose, Citrus contains an intelligence module instrumented to subscribe to published honeypots and collate their telemetry data for further analysis and correlation.

IV. Implementation

A. Intelligence Orchestration

Citrus generates Sigma signatures from behaviour observed on high and medium interaction honeypots. In order to generate these signatures, logs describing mechanisms of attacks must be digested and analysed. Citrus achieves this by subscribing to a Kafka[4] broker where streams of events from the honeypots are stored. As well as receiving raw log inputs, Citrus also maintains a record of malware dropped onto the medium interaction honeypots. If a malware variant has not previously been encountered by Citrus, the hash is calculated and stored. Citrus also orchestrates the correlation with external sources, such as VirusTotal, autonomously in order to generate a rich understanding of malware behaviour. VirusTotal[5] is a malware intelligence service which provides a wealth of information pertaining to malicious files such as target operating system and malware family, and in some cases rich runtime behaviour which relates to the implementation detail [9]. In the event that a log is transferred from a honeypot to Citrus, it must be initially ensured that the event which the log describes is malicious. This is achieved by modelling normal behaviour on the honeypots, and any event which deviates from this pattern is deemed malicious and a corresponding signature is generated.

B. Honeypot Processing & Signature Generation

As discussed in the previous sections, there is a need for host based signatures of emerging threats. Utilising Sigma as the method of describing the malicious actions occurring on the deployed honeypots grants a rich understanding of attacks suitable for detection using heterogeneous defense solutions. Citrus receives logs from disparate sources such as Windows Event Tracing and honeypots such as Dionaea, and converts the extracted malicious events into Sigma format for inclusion into IDSs signature database. Citrus orchestrates the generation of Sigma signatures upon notification by the Tangerine module that a malicious action was taken on a honeypot. Citrus is capable of generating signatures which describe a range of malicious mechanisms including file creation and modification, process creation, registry activity, service creation, and authentication. By specifying which mechanism is used in the attack, the details which uniquely identify it are extracted from raw event log data and translated into Sigma format using a template signature file. An example signature is documented in Listing 1, and describes the series of process creation when a WannaCry variant was executed on a high interaction honeypot.

```
logsource:
   category: process_creation
   product: windows
detection:
   selection:
      CommandLine:
         - ’*\cmd.exe /c
         "C:\\ProgramData\\loads.exe"’
      ParentImage:
         - ’*\wscript.exe’
   condition: selection
```

Listing 1. Process creation signature of WannaCry variant

C. Policy Engine Integration

Sigma provides a standard signature format to describe events from log sources. It is intended to be flexible in nature and will seamlessly integrate with any IDS, SIEM, or bespoke security framework. To show the feasibility of this integration and the detection capabilities provided, Citrus was instrumented with a bespoke policy engine which uses the Sigma format. Furthermore, a detection application was built upon Citrus which performs a comparison of each log emanating from internal hosts to signatures present within the policy engine. If a known signature matches with any log, Citrus detects an intrusion and an alert is issued.

V. Evaluation

In order to evaluate Citrus, a relevant network test-bed was developed and is outlined below. For the purposes of hunting for emerging threats a Debian VM (Virtual Machine) containing TPot[6] was deployed within the Cyber Threat Laboratory. The VM was granted an externally accessible WAN (Wide Area Network) IP address in order to capture a wide variety of potential attacks including credential bruteforcing, exploit observation and malware capture over a large surface area. TPot is composed of a number of medium interaction container based honeypots enabling multiple vulnerable services to be exposed to the internet. High interaction honeypots were deployed which are based on Windows Docker containers. Individual attacks are allowed to propagate fully and complete their objective in order to capture the entire attack session and its corresponding events. Once this has been accomplished, the Docker containers are destroyed and reprovisioned, ready for another adversary to attempt infiltration. The intricately detailed host and networking logs from high interaction honeypot and emerging threat data extracted from medium interaction honeypots are transferred to Citrus in the same manner: Kafka. A Kafka broker was instantiated which receives a stream of events from the honeypots. Citrus maintains a constant connection to this Kafka broker to receive updates about threat data in real-time using the Confluent Kafka[7] library for python.

[4] Kafka https://kafka.apache.org

[5] VirusToal https://www.virustotal.com

[6] TPot https://github.com/dtag-dev-sec/tpotce

[7] Confluent https://www.confluent.io/

A. Signature Generation

In this evaluation, malware dropped onto medium interaction honeypots and Windows Event logs from high interaction honeypots are used for signature generation purposes. As Citrus maintains an authoritative record of files dropped onto the medium interaction honeypots, Sigma signatures are generated based upon the hash of the file and also the dynamic run time behaviour gathered from external sources. As the hash value is trivially changed via minute code modification, Citrus attempts to extract behavioural artefacts which are unlikely to change upon recompilation to defend against similar variants.

The hash and file behaviour of a PE file obtained through an exploit captured on the Dionaea honeypot is used to evaluate the efficiency of policy application. The time taken for the generation of the signature is illustrated in Figure 2. During the period of observation, a number of distinct attacks were captured on the high interaction honeypots, their attack mechanisms were recorded and the corresponding signatures were generated. To evaluate the efficiency of signature generation based on event logs extracted from high interaction honeypot, the time taken within each stage of the process was calculated and is also shown in Figure 2.

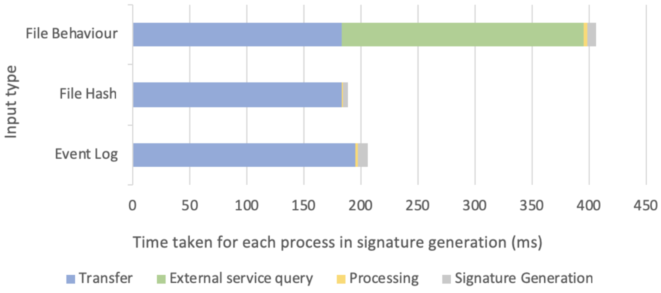
B. Detection Capabilities

The first aspect to evaluate the effectiveness of the signatures generated by Citrus involves ensuring the signatures are specific enough to not generate false positive detection. To test this hypothesis, attacks were injected into the high interaction honeypots Citrus governs. These attacks involve remote authentication and execution of malware which generate Windows Event logs, which are in turn received by Citrus where signatures are generated and the corresponding logs are stored for future playback. This test utilises publicly available dataset of host logs [11] in order to determine the effectiveness of the signatures. The signatures generated in the previous stage were imported into Citrus' policy engine and the dataset was replayed by sending each log to the collector module. As expected, no alerts were issued by Citrus when processing the benign dataset. To further test that the signatures generated are accurate enough to successfully detect the attacks, the corresponding logs which represent the attacks in the previous stage were subsequently merged with the dataset and replayed to Citrus. Each of the injected attacks were correctly identified by the detection application within Citrus without false positive assessment.

VI. FUTURE WORK

The successful identification of attacks suggest this avenue is worth exploring further to exploit the defense benefits encompassing host based honeypot signatures. In our evaluation, primitive attacks were used to detect the evaluate the detection capabilites. It would be valuable to incorporate sophisticated attacks which leverage multiple stages, such as lateral movement, to extensively test the intricate host details available to Citrus, and the corresponding accuracy of the signatures.

Fig. 2. Time taken for signature generation of attack mechanism



VII. CONCLUSION

This paper presents the design, implementation, and evaluation of a honeypot signature generation framework, Citrus, which is adept at observation and mitigation of emerging threats by the analysis of data collected from honeypots of varying interaction levels. The work outlined in this paper showcases the accuracy of the detection capabilities, and the flexibility to integrate these signatures to support existing IDS via the open event description format, Sigma.

REFERENCES

[1] Muhammet Baykara. 2015. A Survey on Potential Applications of Honeypot Technology in Intrusion Detection Systems. *International Journal of Computer Networks and Applications*

[2] Urjita Thakar. 2005. HoneyAnalyzer - Analysis and Extraction of Intrusion Detection Patterns Signatures Using Honeypot. In *Proceedings of the IEEE Infocom.*

[3] Daniel Silalahi, Yudistira Asnar. 2017. Rule generator for IPS by using honeypot to fight polymorphic worm. In *International Conference on Data and Software Engineering*. 1-5.

[4] Varan Mahajan, Sateesh Peddoju. 2017. Integration of network intrusion detection systems and honeypot networks for cloud security. In *IEEE International Conference on Computing, Communication and Automation*. 829-834. DOI: https://doi.org/10.1109/CCAA.2017.8229911

[5] Emmanouil Vasilomanolakis, Shreyas Srinivasa. 2016. Multi-stage attack detection and signature generation with ICS honeypots. In *Proceedings of the NOMS 2016 - 2016 IEEE/IFIP Network Operations and Management Symposium*. 1227-1232. DOI: https://doi.org/10.1109/NOMS.2016.7502992

[6] Saurabh Singh, Pradip Kumar Sharma, Seo Yeon Moon. 2016. A comprehensive study on APT attacks and countermeasures for future networks and communications: challenges and solutions. In *The Journal of Supercomputing*. 1-32. DOI: https://doi.org/10.1007/s11227-016-1850-4

[7] Adel Alshamrani, Sowmya Myneni, Ankur Chowdhary. A Survey on Advanced Persistent Threats: Techniques, Solutions, Challenges, and Research Opportunities. 2019. *IEEE Communications Surveys & Tutorials.* 1851-1877. DOI: https://doi.org/10.1109/COMST.2019.2891891

[8] Amos O. Olagunju and Farouk Samu. 2016. In Search of Effective Honeypot and Honeynet Systems for Real-Time Intrusion Detection and Prevention. In *Proceedings of the 5th Annual Conference on Research in Information Technology (RIIT '16).* 41-46. DOI: https://doi.org/10.1145/2978178.2978184

[9] Gerrado Fernandez, Ana Nieto (2017). Modeling Malware-driven Honeypots Modeling Malware-driven Honeypots. 2018. In *International Conference on Trust and Privacy in Digital Business*. DOI: https://doi.org/10.1007/978-3-319-64483-7

[10] Oscar Navarro. 2011. Gathering Intelligence Through Realistic Industrial Control System Honeypots. *Critical Information Infrastructures Security*. 143-153. DOI: https://doi.org/10.1007/978-3-642-21694-7

[11] Melissa Turcotte, Alexander Kent. 2018. Unified Host and Network Data Set. In *Data Science for Cyber-Security*. 1-22.