Sentinel: A Robust Intrusion Detection System for IoT Networks Using Kernel-Level System Information

Adrien Cosson, Amit Kumar Sikder, Leonardo Babun, Z. Berkay Celik, Patrick McDaniel, A. Selcuk Uluagac

University of Minnesota
Driven to Discover℠

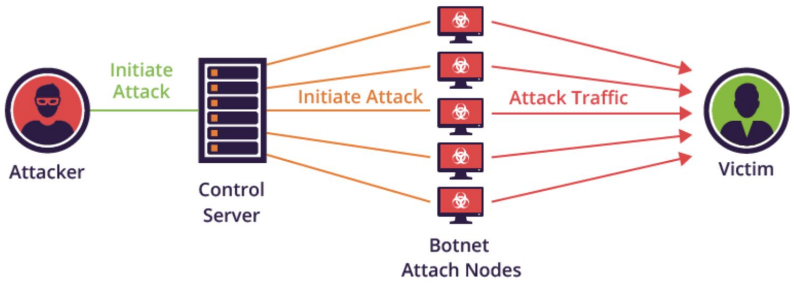
Authors

- **Adrien Cosson** - Penn State University, Graduated with Master's
- **Amit Kumar Sikder** - Florida International University, Postdoctoral Fellow
- **Leonardo Babun** - Florida International University, Received Doctoral Degree
- **Z. Berkay Celik** - Purdue University, Assistant Professor
- **Patrick McDaniel** - Penn State University, Professor, Director of the Institute for Networking and Security Research (INSR)
- **A. Selcuk Uluagac** - Florida International University, Assistant Professor

IoTDI 2021 - ACM/IEEE Conference on Internet of Things Design and Implementation, 2021

IoT & Cyber Attacks

- IoT devices becoming more common
- Influenced by economics and speed to market
- Devices are resource-constrained
- Developers don't have direct access to the hardware to integrate security measures
- Attacks
  - Node-level
  - Network-level
  - Application-level
- Mirai Botnet: launched a series of DDoS attacks



Intrusion Detection

- Intrusion detection detects a system for malicious behavior
  - Architectures
    - Network-based IDS (NIDS): monitor the state of an entire network
    - Host-based IDS (HIDS): run on a specific host and search for malware operating inside of it through the use of system-level and process-level information
  - Approaches
    - Signature-based: compares the collected data pattern to a list signatures of known threats
    - Anomaly-based: builds an internal representation of the system compared to an expected baseline state
    - Specification-based: has set of baseline and threshold values and compares to the current situation

Sentinel Overview

- The idea of using low-level host data for intrusion detection is not new, but it hasn't been implemented for IoT environments
- Sentinel uses a Linux-based kernel module (SKM) to collect low-level host data which is used to detect node and network level attacks
- The heavy work of analyzing the data using ML is offloaded to the hub to differentiate between benign and malicious attacks

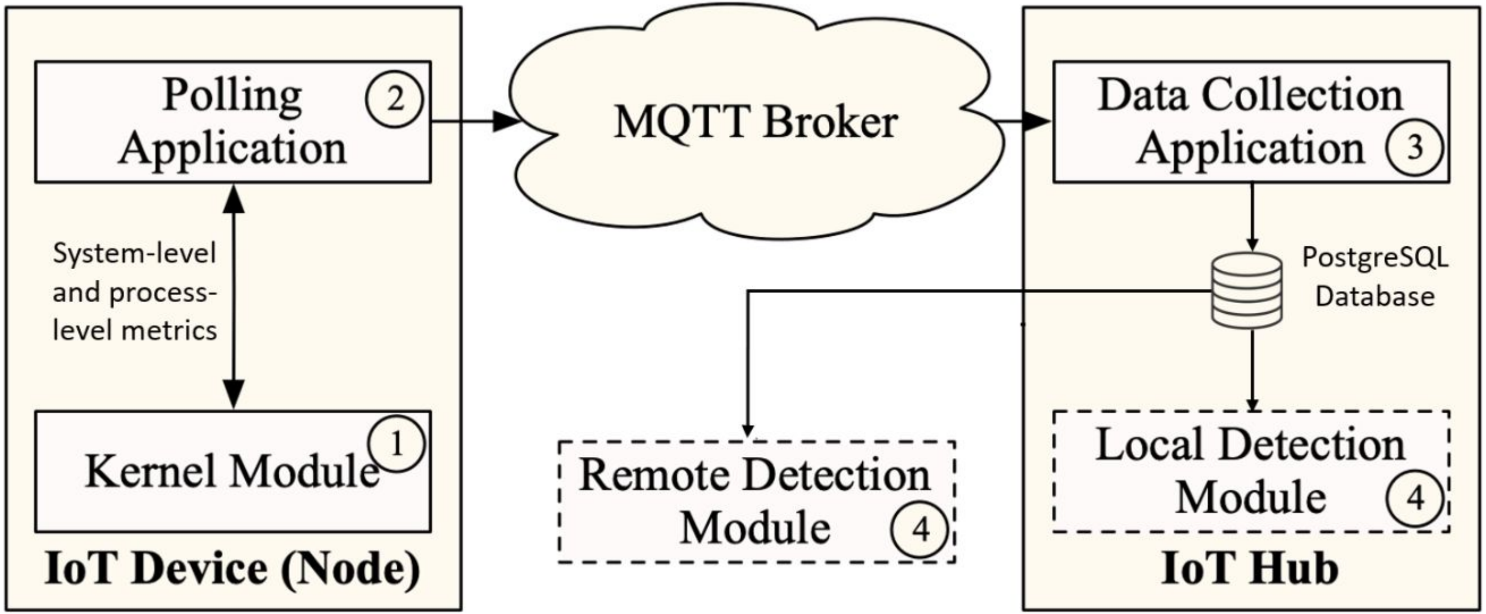
Sentinel Architecture

- Uses Linux, which has high market share for IoT devices (43%)
- SKM is lightweight and easily implemented on other OS platforms
- File-based view of kernel data structures provides an easier interface for developers
- SKM is low overhead and needs less computing power
- Uses commonly found pub-sub protocol (MQTT) to make information accessible to the hub
    - Naming convention example: home/mqtt_lock/available

Sentinel Features

- Configurable polling rates: low-high, dynamic polling rate
- PostgreSQL database collects data and allows for concurrent access
- ML-based detection techniques used: Naïve Bayes, Rule-Based, Regression Model. Neural Network, Tree-Based Classifiers
- IDS collects data from the database, trains the ML model, learns benign device behavior, pushes a notification to the user interface via the hub in case of a malicious attack

Sentinel Framework



Using Mirai Effects to Test Sentinel

- **Network scan/pivoting**
  - Attack 1: the attacked device continuously scans a server to find other devices
- **Exfiltration**
  - Attack 2: send large UDP packets to a server that discards them
- **C&C Keep-alive**
  - Attack 3: periodically ping an infected device that responds with an empty payload
- **Black/Grey Hole Attack**: disrupt the network by compromising a device
  - Attack 4: server floods network with large message
  - Attack 5: send out random messages to simulate the partial packet drops

Evaluation Setup & Methodology

- 2 IoT Platforms: Home Assistant and WebThings
- Binary Classification
  - The datasets contain samples recorded every second over a time window and are labeled if there is an attack or not
  - 7 performance metrics: True Positive Rate (TPR), False Negative Rate (FNR), True Negative Rate (TNR), False Positive Rate (FPR), Accuracy, F-score, and Average Computation Time (Avg. CT)
- Multi-Class Classification
  - 5 Attacks + No Attack
  - For each device/attack/framework combination, run each device for 20 min. of traces for attack scenarios and record metrics

Figure 3: Floor plan of the experimental testbed

Impacts

- Model Parameters
- Platform Configurations
- Power Consumption
- Polling Rate

Results - Binary and Multi-Class Classification

DT & RF have highest accuracies

ML Algorithm	WebThings TPR	FNR	TNR	FPR	Acc.	F-Score	Avg. CT (s)	Home Assistant TPR	FPR	TNR	FNR	Acc.	F-score	Avg. CT (s)
Naive Bayes	0.8	0.2	0.94	0.06	0.87	0.864	21.6	0.77	0.23	0.92	0.08	0.845	0.838	27
PART	0.85	0.15	0.94	0.06	0.895	0.892	24.5	0.75	0.25	0.88	0.12	0.815	0.809	34.6
LR	0.91	0.09	0.9	0.1	0.905	0.905	34	0.88	0.12	0.91	0.09	0.895	0.894	48
MP	0.89	0.11	0.95	0.05	0.92	0.919	68.5	0.86	0.14	0.94	0.06	0.9	0.898	81.7
DT	0.95	0.05	0.97	0.03	0.96	0.959	35.6	0.92	0.08	0.95	0.05	0.935	0.934	51.5
RF	0.95	0.05	0.98	0.02	0.965	0.964	87.9	0.91	0.09	0.97	0.03	0.94	0.939	94
LMT	0.94	0.06	0.92	0.08	0.93	0.92	102.5	0.92	0.08	0.95	0.02	0.93	0.929	112

Table 3: Performance of SENTINEL in binary classification.

RF has high CT

97% average accuracy of detecting attack

	Decision Tree Attack 1	Attack 2	Attack 3	Attack 4	Attack 5	No Attack	Random Forest Attack 1	Attack 2	Attack 3	Attack 4	Attack 5	No Attack
Attack 1	98.76	0.17	0.02	0.00	0.00	1.06	98.51	0.42	0.05	0.00	0.00	1.02
Attack 2	0.167	96.13	0.74	0.20	0.11	2.65	0.27	97.42	0.63	0.17	0.11	1.40
Attack 3	0.00	0.00	96.19	0.35	0.02	3.33	0.00	0.00	96.84	0.47	0.02	2.67
Attack 4	0.00	0.17	0.48	96.56	0.15	2.65	0.00	0.17	0.89	96.71	0.15	2.08
Attack 5	0.02	0.00	0.04	0.07	97.46	2.41	0.00	0.00	0.14	0.15	97.03	2.69
No Attack	0.05	0.26	0.18	0.17	0.20	99.15	0.08	0.39	0.12	0.17	0.27	98.97

Table 4: Confusion matrix for WebThings multi-class classification.

Highest accuracy detecting network scan/pivoting actions

Low FPR & FNR

Lowest accuracy detecting exfiltration

96% average accuracy of detecting attack

	Decision Tree Attack 1	Attack 2	Attack 3	Attack 4	Attack 5	No Attack	Random Forest Attack 1	Attack 2	Attack 3	Attack 4	Attack 5	No Attack
Attack 1	99.35	0.13	0.00	0.00	0.00	0.52	99.12	0.13	0.00	0.00	0.00	0.75
Attack 2	0.00	91.31	0.41	0.00	0.00	8.28	0.00	93.87	0.74	0.00	0.00	5.39
Attack 3	0.04	0.43	96.67	0.04	0.00	2.83	0.06	1.06	97.08	0.12	0.00	1.74
Attack 4	0.00	0.00	0.13	99.11	0.02	0.74	0.00	0.00	0.17	98.75	0.14	0.94
Attack 5	0.00	0.00	0.00	0.00	98.15	1.85	0.00	0.00	0.06	0.07	98.09	1.78
No Attack	0.04	1.36	0.15	0.06	0.09	98.31	0.09	0.87	0.16	0.08	0.12	98.68

Table 5: Confusion matrix for Home Assistant multi-class classification.

Results - Model Parameters

- DT: accuracy increases with the number of tree depths
- RF: accuracy increases with number of trees, but computation time increases significantly with number of trees
- Accuracy is insignificant compared to the computation time

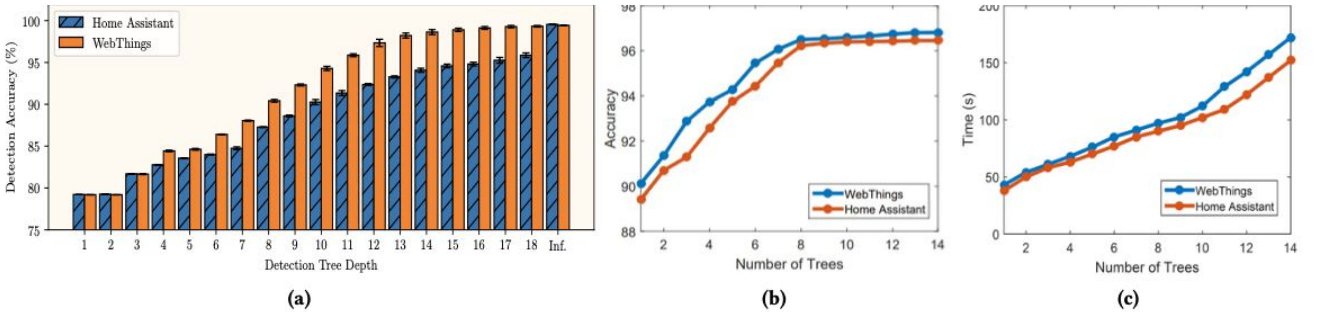


Figure 5: Impact of model parameter in SENTINEL: (a) tree depth vs accuracy using decision tree, (b) number of tree vs accuracy using random forest, and (c) number of tree vs computation time using random forest.

Results - Platform Configurations

- Accuracy drops as sampling rate increases
- Sentinel can effectively run on a low core-count IoT device

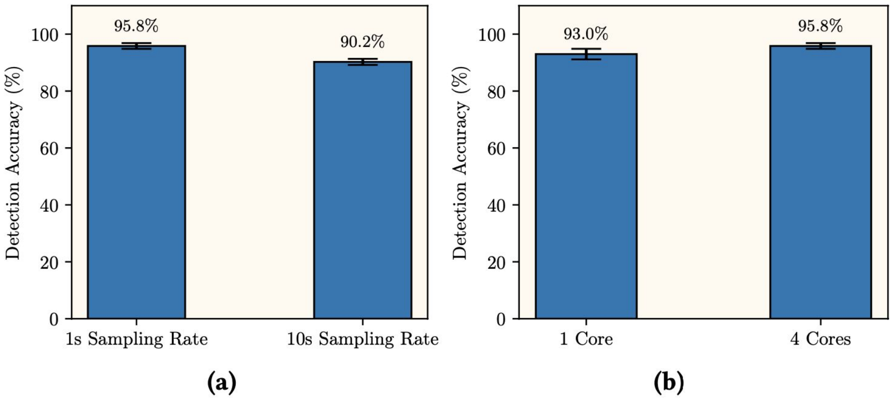


Figure 6: Detection Accuracy for (a) different polling rate (1s and 10s), (b) different computation power (1 and 4 cores).

Results - Power Consumption

- As polling frequency decreases, the power consumption overhead incurred decreases
- Inactive devices have large overhead because of sleep mode
- Can correlate the running processes to reduce overhead by reducing the polling rate

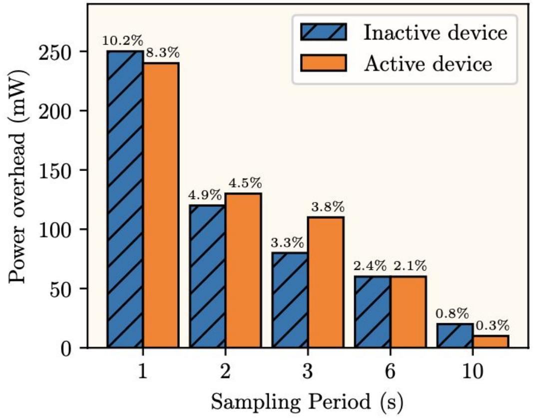


Figure 7: Power overhead caused by Sentinel for various polling periods, expressed as absolute and relative values

Results - Polling Rate

- Accuracy and power consumption are proportional for different polling rates
- Small tradeoff between accuracy and power consumption

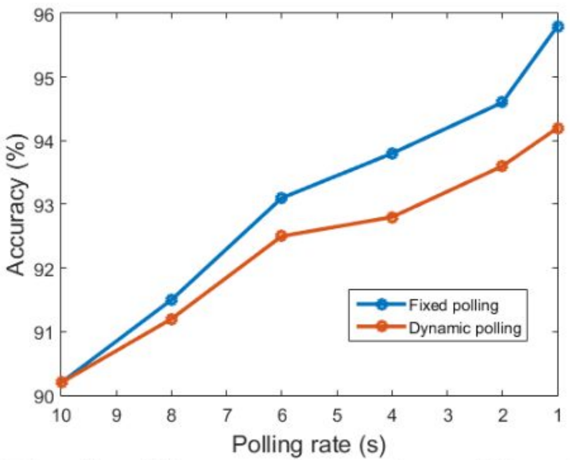


Figure 8: Fixed polling vs dynamic polling in SENTINEL

Positive Points

- Low-Cost
- Lightweight Framework
- Scalable for different configurations

Negative Points

- Device Malfunctions
- Attackers could falsify SKM data
- Any user on device can access the exposed data

Discussion

- How secure is the system?
- What are important features for the customer that Sentinel should have in terms of security?
- Is ~95% accuracy good enough?
- Are there any other metrics that could be considered, in addition to low-level system information?