Work-in-Progress: Boot Sequence Integrity Verification with Power Analysis

Arthur Grisel-Davy, Amrita Milan Bhogayata, Srijan Pabbi, Apurva Narayan, Sebastian Fischmeister

Department of Electrical and Computer Engineering
University of Waterloo. Waterloo, Ontario, Canada
Email: agriseld@uwaterloo.ca

***Abstract*—The current security mechanisms for embedded systems often rely on Intrusion Detection System (IDS) running on the system itself. This provides the detector with relevant internal resources but also exposes it to being bypassed by an attacker. If the host is compromised, its IDS can not be trusted anymore and becomes useless. Power consumption offers an accurate and trusted representation of the system's state that can be leveraged to verify its integrity during the boot sequence. We present a novel IDS that uses the side-channel power consumption of a target device to protect it against various firmware and hardware attacks. The proposed Boot Process Verifier (BPV) uses a combination of rule-based and machine-learning-based side-channel analysis to monitor and evaluate the integrity of different networking equipment with an overall accuracy of 0,942. The BPV is part of a new layer of cybersecurity mechanisms that leverage the physical emissions of devices for protection.**

I. Introduction

The boot sequence of an embedded system contains many security-critical operations. Two examples are loading the firmware and activating hardware components. Firmware loading can be vulnerable to many attacks [1], [2], including downgrading firmware, loading malicious firmware, and cancelling firmware updates. Hardware components also provides a means of entry for attackers who can leverage malicious peripherals [3], for traffic-sniffing, key-logging, or altering the system's behaviour.

The standard countermeasures to firmware and hardware attacks [4] share the common flaw of being performed by the protected machine itself, allowing an attacker to bypass them after infecting the machine. Intrusion Detection Systems (IDSs) face a trade-off between accessing relevant information and keeping the detection mechanism separated from the target machine. Our solution addresses this trade-off by leveraging unforgeable side-channel information.

This paper presents a novel solution for firmware verification using side-channel analysis. Building on the assumption that every security mechanism operating on a host is vulnerable to being bypassed and that any deviation from a normal boot sequence operation is a reason for concern, we propose to use the device's power consumption signature during the boot sequence to assess its integrity. The integrity evaluation leverages unforgeable power consumption data collected independently of the host. A distance-based outlier detector can learn the expected pattern and detect any variation in a new boot sequence. Our solution can detect various attacks centred around firmware manipulation. This novel detector is versatile, retrofittable to any embedded system, and requires a theoretic minimum of four training examples, well below current data requirements for state-of-the-art methods [5].

Many hardware and firmware attacks leverage machine-specific designs to provide an access point to the attacker. This paper focuses on attacks relying on firmware modifications, but the method for detecting hardware modifications remains the same. Because the firmware is responsible for the initialization of the components, the low-level communications, and some in-depth security features, executing adversary code in place of the expected firmware is a powerful capability [6]. A firmware modification is defined as deploying a new firmware code. Modifications include implementing custom functions, removing security features, or changing the firmware for a different version (downgrade or upgrade), as well as bypassing firmware procedures via hardware tampering. Any loading of a non-approved firmware (including a maliciously modified one) is considered an attack. This type of attack can result in the attacker gaining full control of the device.

Manufacturers have implemented different security mechanisms to guarantee the integrity of the firmware. The first and most common is to cryptographically sign, or compute a checksum of the code. This method suffers many possible bypasses, even with dedicated hardware [7].

Historically, Side-Channel Analysis (SCA) is mainly used for attacks. However, defense is also a promising application for this technology with runtime anomaly detection [8] or specific attack detection [9]. These mechanisms are powerful at protecting systems that cannot host security software.

II. Boot Process Verifier

To enable firmware verification, we design a training and testing pipeline that performs anomaly detection on a boot-up sequence power trace. A boot-up power trace is a time series corresponding to the power consumption of the machine during one complete boot-up sequence. The Boot Process Verifier (BPV) takes as input a power trace and verifies its validity against valid boot-up traces (see Figure 1). The Interquartile Range (IQR) is a measure of dispersion of samples. It is based on the first and third quartiles and defined as $IQR = Q_3 - Q_1$ with Q_3 the third quartile and Q_1 the first quartile. This value is commonly used [10] to detect outliers as a more

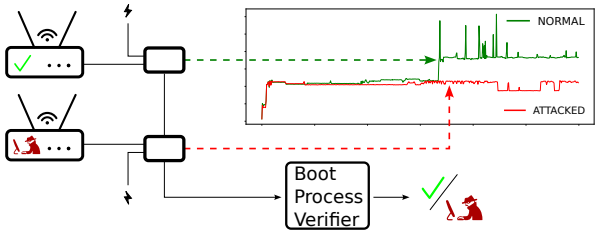

Fig. 1: Overview of the Boot Process Verifier pipeline.

robust alternative to the 3σ interval of a Gaussian distribution. The training phase consists in first computing the IQR of the Euclidean distances from each training trace to their average. Then, the distance threshold takes the value $Q3 + 1.5 \times IQR$. The distance of each new trace to the reference average is computed and compared to the threshold in the detection phase. If the distance is above the pre-computed threshold, the new trace is considered anomalous.

III. Experiment

To verify the performance of the proposed detector, we designed an experiment to detect firmware modifications on networking devices. These devices are bespoke for transmitting information as fast as possible. We consider four machines representing consumer-available products for different prices and performances: Asus Router RT-N12 D1, Linksys Router MR8300 v1.1, TP-Link Switch T1500G-10PS, HP Switch Procurve 2650 J4899B. As part of the experiment, each device undergoes firmware modifications using OpenWRT for the routers and downgraded firmware for the switches.

We use a hardware device [11] placed in series with the power cable of the target device. The capture box's shunt resistor generates a voltage drop representative of the global power consumption of the machine. This voltage drop value is recorded at a sampling rate of 10 KSPS. A managed Power Distribution Unit (PDU) enables turning each machine ON or OFF automatically. To account for randomness and gather representative boot-up sequences of the device, we performed 500 boot iterations per machine and per firmware version. The complete dataset is publicly available [12].

The output of each measurement is a $\approx$ 24 hours power trace containing 500 boot-up sequence event. A threshold-based algorithm extracts the boot-up sequences from the complete trace. The algorithm leverages the rising edge at the start of the boot sequence to detect the start time accurately. We use two hyperparameters T the consumption threshold, and L the length of the boot-up sequence controls the detection and are tuned per machine. When a sample crosses the threshold on a rising edge, the next L samples are saved as a boot-up sequence. The value of T is taken just above the maximum consumption when the machine is off in order to be crossed during the initial consumption rise. The boot time L is around 20 s and the choice of the value is discussed in III-A. The extracted traces are resampled at $50ms$ using a median aggregator, and median and average filters are applied to remove noises that could falsely trigger the detection.

A. Results

Table I shows the experimental results. For each machine, we compute the distance threshold using ten known-good traces and classify ten normal and ten abnormal traces. The procedure is repeated 20 times, and the results averaged per machine. We compute the overall F_1 score using arithmetic mean.

Machine	Detection F_1 Score	Overall F_1 Score
TP-Link switch	0.866	0.942
HP switch	0.983	
Asus router	1	
Linksys router	0.921	

TABLE I: Results of detection.

Two hyper-parameters require tuning to achieve the best performance. The length of the extracted sequences needs to cover the whole boot-up while including no post-boot operations that introduce noise. Because the IQR method is based on quartiles, a theoretical minimum of four traces is required. Collecting additional traces offers a more robust IQR threshold placement, but too many traces (> 20) offer marginal improvements as the boot-up sequence is usually consistent. Other parameters, such as sampling rate or pre-processing values, do not show to significantly affect the results.

IV. Conclusion

This study illustrates the application of side-channel analysis to detect firmware attacks. The proposed side-channel-based IDS can reliably detect firmware tampering from the power consumption trace. Moreover, distance-based models leveraged in this study allow minimal training data and time requirements. Deploying this technology to production networking equipment requires minimal downtime and hardware intrusion. Finally, it applies to any clientless equipment.

References

[1] "CVE-2019-19642.." MITRE, CVE-ID CVE-2019-19642., 2019.
[2] "CVE-2020-15046.." MITRE, CVE-ID CVE-2020-15046., 2020.
[3] Hack5, "Rubber ducky, lan turtle, key croc," 2021.
[4] L. McMinn and J. Butts, "A firmware verification tool for programmable logic controllers," in *Critical Infrastructure Protection VI* (J. Butts and S. Shenoi, eds.), (Berlin, Heidelberg), pp. 59–69, Springer Berlin Heidelberg, 2012.
[5] H. Ismail Fawaz, G. Forestier, J. Weber, L. Idoumghar, and P.-A. Muller, "Deep learning for time series classification: a review," *Data mining and knowledge discovery*, vol. 33, no. 4, pp. 917–963, 2019.
[6] "Mitre attck® t1542.001 pre-os boot: System firmware." https://attack.mitre.org/versions/v10/techniques/T1542/001/. Accessed: 2022-03-31.
[7] C. Cimpanu, "Thrangrycat flaw lets attackers plant persistent backdoors on cisco gear," *Accessed: Sep*, vol. 15, p. 2019, 2019.
[8] S. Dunlap, J. Butts, J. Lopez, M. Rice, and B. Mullins, "Using timing-based side channels for anomaly detection in industrial control systems," *International Journal of Critical Infrastructure Protection*, vol. 15, pp. 12–26, 2016.
[9] A. Xu, Y. Jiang, Y. Cao, G. Zhang, X. Ji, and W. Xu, "Addp: Anomaly detection for dtu based on power consumption side-channel," in *2019 IEEE 3rd Conference on Energy Internet and Energy System Integration (EI2)*, pp. 2659–2663, 2019.
[10] J. Han, J. Pei, and M. Kamber, *Data mining: concepts and techniques*. Elsevier, 2011.
[11] J. Doe, "Source hidden for double blind review," *Journal*, 2022.
[12] A. Grisel-Davy, "Dataset of bootup power consumption traces for four networking equipment https://doi.org/10.5281/zenodo.6419214," Apr. 2022.