Available online www.jsaer.com

***Journal of Scientific and Engineering Research*, 2019, 6(10):298-309**

Research Article

ISSN: 2394-2630
CODEN(USA): JSERBR

SEDAT: Security Enhanced Device Attestation with TPM2.0

Avani Dave, Monty Wiseman, David Safford

Department of Computer Science and Electrical Engineering,
University of Maryland, Baltimore County

Abstract: Remote attestation is one of the ways to verify the state of an untrusted device. Earlier research has attempted remote verification of a device's state using hardware, software, or hybrid approaches. Majority of them have used Attestation Key as a hardware root of trust, which does not detect hardware modification or couterfiet issues. In addition, they do not have a secure communication channel between verifier and prover, which makes them susceptible to mordern security attacks. This paper presents SEDAT, a novel methodology for remote attesta- tion of the device via a security enhanced communication channel. SEDAT performs hardware, firmware, and software attestation. SEDAT enhances the communication protocol security between verifier and prover by using the Single Packet Authorization (SPA) technique, which provides replay and Denial of Service (DoS) protection. SEDAT provides a way for verifier to get on- demand device integrity and authenticity status via a secure chan- nel. It also enables the verifier to detect counterfeit hardware, change in firmware, and software code on the device. SEDAT validates the manufacturer's root CA certificate, platform cer- tificate, endorsement certificate (EK), and attributes certificates to perform platform hardware attestation. SEDAT is the first known tool that represents firmware, and Integrity Measurement Authority (IMA) event logs in the Canonical Event Logs (CEL) format (recommended by Trusted Computing Group). SEDAT is the first implementation, to the best of our knowledge, that showcases end to end hardware, firmware, and software remote attestation using Trusted Platform Module (TPM2.0) which is resilient to DoS and replay attacks. SEDAT is the first remote verifier that is capable of retrieving a TPM2.0 quote from prover and validate it after regeneration, using a software TPM2.0 quote check. All source code, tools, and kernel patches are open-sourced via BSD 2-Clause License.

Keywords: Secure boot, Reference integrity measurement, remote attestation, platform root of security, TPM, full stack security

Introduction

Traditional computing and Embedded devices are prolif- erating into numerous and diverse aspects of everyday life. These devices are utilizing in different domains ranging from tiny personal gadgets to large industrial systems. the amount of such devices connected to the Internet is growing rapidly and securing these devices and our electronic infrastructure becomes increasingly difficult, in particular because a large fraction of devices cannot be managed by security professional nor can they be protected by firewalls. these devices are most likely to be susceptible to attacks like - hardware modification, or counterfeit can cause serious security challenges as e.g., see Chinese supply chain hardware attack [1]. Malware infestation involves modifying a device's firmware and software and replacing benign code with a malicious one. Which can destroy physical equipment (e.g., see Stuxnet [2]) or enables a more sophisticated attack threatening the safety of the users (e.g., see Jeep hack [3]). This increasing importance confronts developers with new challenges. One of them is the verification of the identity and integrity of a device by the trusted remote attestation tool called remote verifier.

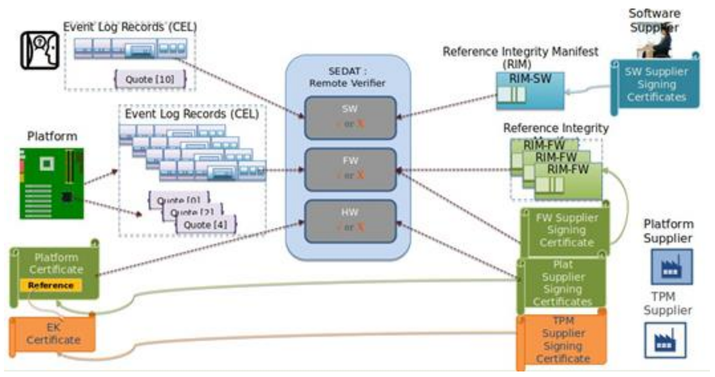

Fig. 1. SEDAT: Remote Verifier

Fig-1 shows high-level design of trusted remote verifier - Security Enhanced Device Attestation with TPM 2.0 (SEDAT) to attest untrusted devices. As can be seen, the verifier has to attest three components: hardware, firmware, and software to fully attest a device. Device suppliers will provide platform root certificates, endorsement certificates, platform attributes certificate. The device will have firmware and software mod- ules that need to be unchanged and executed in the correct order to ensure integrity. This execution of the firmware and software module will create event records in form of firmware and IMA event logs, which will be explained in detail in the coming section. Fig-1 has Reference Integrity Manifest (RIM) for both firmware and software event logs, which will provide golden measurements based on Trusted Computing Group (TCG) 's new recommendations. Details of each module and working will be discussed in the following sections.

Remote Verifier (RV) has to validate the integrity and authenticity of hardware, firmware, and software state on the untrusted device against known good state for attesting the device. If any or all of the states fail, then the device attestation result is failed. RV should be able to perform attestation on- demand to ensure the correct state of remote prover. RV can perform a quote check with TPM2.0 to validate Ek and signing keys.

Goals and Contributions

In this paper, we present SEDAT, the first proof of concept work for remote attestation, to ensure the integrity and au- thenticity of devices via a security-enhanced communication channel. Designing such a verifier is a challenging task.As there are many possibilities of malfunctions, like, there could be counterfeit hardware, modified software or malicious prover will triggering the attestation process. Therefore, we analyzed the requirements for designing a secure attestation protocol and depict how to address them with minimal features and assumptions. Further, we identified possible use-cases where SEDAT can be applied. Our work brings the following contri- butions:

• **Client provisioning:** Provided tool for client provision- ing. it is the process that enables the platform owner to register the device with a remote verifier using a security- enhanced Single Packet Authentication (SPA) technique.

• **Platform attestation:** Provide remote verifier for plat- form which attests platform root CA certificates, endorse- ment certificates, platform bindings.

• **CEL firmware event logs**: Provide tools to convert and verify firmware event logs into canonical event log (CEL) structure. also, validated upstreamed kernel patches for crypto align firmware event logs.

• **CEL IMA event logs:** Provide tools and kernel patches for getting IMA event logs into userspace and converting it into CEL format.

• **Quote check:** Provide tools to check the quote at a remote verifier.

• **Secure authentication:** Provide tools for secure connec- tion and authentication for verifier and protects it from replay and DoS type attacks.

Outline: This is not correct at this time –putting some place- holder flow. Outline, Section 2 reviews related work, adversary models and Replay, DoS attacks on attestation. Section 3 provides required preliminaries and notations, identifies the minimal requirements for secure RV, and describes SEDAT. The prototype implementation of SEDAT has described in Section 4; its application to collective attestation is explained in Section 5. Next, the security of SEDAT has evaluated in Section 7, and the paper concludes in Section 8.

Problem Description

Remote attestation is an interactive process between a trusted remote verifier (denoted as RV) and a potentially untrusted remote device called prover (denoted as RP). It allows a trusted RV to capture the state of a potentially untrusted remote device.Essentially, RV measures and takes hash of the software running on the RP, tranfers it to itself and matches to the golden measurement to attest the device. Remote attestation can be performed by Hardware- only, software-only, or hybrid techniques. Each approach has its merits and demerits. Hardware or hybrid approach provides better security assurance as they use immutable hardware as a root of trust.

One approach to achieve better security is to equip these devices with a root of trust, such as a Trusted Platform Module (TPM), a Trusted Execution Environment (TEE), and Software Guard Extensions (SGX), and then have that root of trust attest the state of the device or computations made. Many devices have a Trusted Platform Module (TPM) that fulfills these tasks. Remote device attestation (RV) can be used to establish static or dynamic root of trust in cyber-physical and industrial controls systems.It can be used as building block for other security services and primitives, such as device provisioning,

firmware updates, kernel software patching.

A. TPM based remote attestation

As shown in Fig-1, to attest a device RV needs to validate hardware, firmware and software all three components of the remote device.In last decade, researcher and industry has tried to solve the remote attestation problem and provided couple of solutions but none of it solves all three sametime. For example National Security Agency (NSA) has open-sourced tool called HIRS [4] for complite platform / hardware attestation. but it does not cover firmware or software attestation, moreover it works on centos 7 only and supports old tpm2-tools and tpm2- tss Intel’s stack for TPM based device attestation.Second,The International Business Machines Corporation (IBM) has open- sourced their implementation called Attestation Client Server (IBM-ACS) [5]. IT does platform certificates, hardware and software attestation but it uses IBM’s TSS and Tools to com- municate with software TPM and its not complitely supported for hardware or firmware TPM. Also, it does not have support to verify firmware and software event logs in Cannonical Event logs (CEL) structure. Google has their open-sourced implementation for remote attestation called go-attestation [6], which does not do platform attestation and start at AK as root of trust. Acadamic implementation called keylime [7] has support for multiple platforms and languages for attestation but it misses platform certificate validations, firmware and software even logs validations. all of the above solutions do not have support for replay or Denial of Service (DoS) attacks as they all work on https client-server protocols.

This movitaved our research to first figure out what threat and adversary model are open to address, followed by looking at limitations of available attestation schemes.

B. Threat and Adversary Model

Following the adversarial models from [8], SEDAT has classified attacks on remote attestation in three categories.

• **Communication Adversary:** Adversary has complete control over all communication channels. it can do eaves- dropping and/or inject/modify packets, delay or drop packets. in case of DoS attack, adversery will flood the remote verifier by sending multiple provisioning request and eventually brings the verifier down.

• **Software Adversary:** Adversary can exploit software vulnerabilities to infect prover or verifier, read its un-protected memory regions, manipulate its software state or fake identity of prover.

• **Mobile Adversary:** In addition to adversaies soft capa- bilities, this sophiesticated mobile adversaries are capable of erasing all traces of its previous presence on prover which makes it not detected by remote verifier. Executing such a sophisticated attack requires the knowledge of the exact execution time of attestation.

C. Limitations of Current Solutions

In recent research papers and in practice TPM has been used for hardware root of trust and remote attestation. where the integrity and authenticity of platform is relied on TPM based attestation key (AIK or AK).

• **AK limitation:** AK is taken as hardware root of trust anchor for remote attestation. problem of taking AK as only hardware root of trust is, generation of AK comes few step later in device manufacturing process. AK is derived from endorsement key (EK) and EK is provided from platform/hardware vendor by fusing private key and exposing associated public key as explained in next section. So, AK based attestation fails to detect malicious or counterfiet hardware, the TPM module or both. Also, some implementations uses software seed and AK key for attestation.

• Firmware eventlogs limitation: System records the firmware events extended into PCR's in form of firmware event logs. Verifying firmware event logs assures boot sequence and firmware code states integrity. However problem with current state of firmware event logs are they are not crypto aliened format and not avaliable in user space. They are only available in UEFI shell. So, it needs to be read from ACPI table to user space. second, problem is they do not have sequence numbers for the events. so, its hard to read the events and keep track of them from the binary blob when transfered to remote verifier. details of the firmware event log generation is explained in Theory and requirement section of SEDAT.

• **Software eventlogs limitation:** Same as firmware event logs, IMA event logs also does not have sequence num- bers. Second major problem is, since it is stored into small firmware memory which is not ment for storing in- cremental large IMA event logs. It needs the mechanisum to free up the space once the blob is read in user space.

• Quote check limitation: Some remote attestation im- plemetations have used quote generation with nonce and checks it at the provers, but it will be more valuable to check quote and get same PCR values at the RV to ensure that there was no tempering in middle.

• Protection to known attacks: Some non TPM based attestation works have tried to enchance the security against replay and DoS attacks but majority of the at- testation framework takes the https protocol as secure communication channel.

Above all reasons combined motivated our research and eventually resulting in implemetation of SEDAT. SEDAT: the first remote attestation scheme, which performs hardware, firmware and software attestation and it is completely secure against Denial of Service (DoS), replay attacks.

Theory, Requirements, Assumptions and Limitations for New SEDAT

A. Theory and remote attestation requrements:

SEDAT has identified remote device attestation problem needs to address five issues. the theory and requirement for each are explained in below sections, Assumptions and limitations considered while desinging SEDAT are discussed in next subsection.

1) Platform Validation: It is common practice in industry that device vendor will not be the one and only vendor for all the hardware software modules compressed in the device. The device vendor will assemble the device by assembling all different parts in the manufacturing unit and send it to warehouse. After receiving the final product, the device vendor needs to validate the authenticity and integrity of all the modules present in the device.

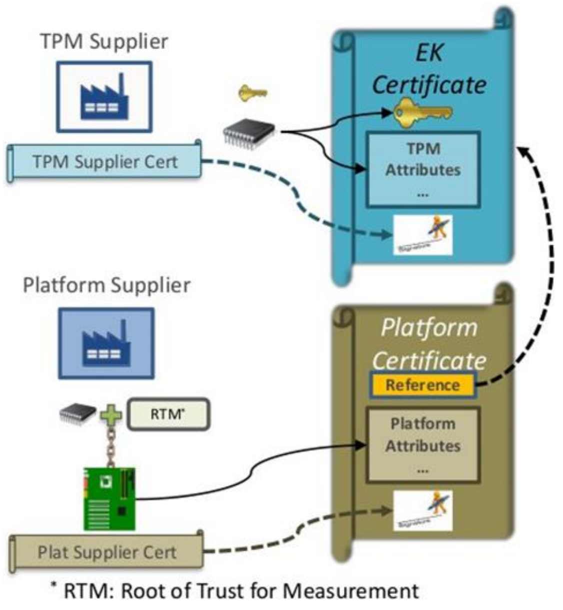

Fig. 2: Platform and TPM supplier certificates binding, establishing platform ownership

All hardware module manufacturers will provide modules' root CA certificate signed by the module manufacturer. The device vendor will validate all module certificates and binds them together with the platform and generates a self-signed platform certificate. The vendor creates a platform attributes certificate, RV needs to verify hardware certificates and bind- ings.

Platform verification process has following steps, as de- picted in Fig.2.

• **Step 1:** TPM vendors will create endorsement private and public keys (EK) for the TPM platform. The private part of the endorsement key will be fused into the hardware, and the public key will be exposed for creating the platform attestation key (AK). This attestation key will act as a trust anchor in the hardware root of trust. EK is used for creating an endorsement certificate.

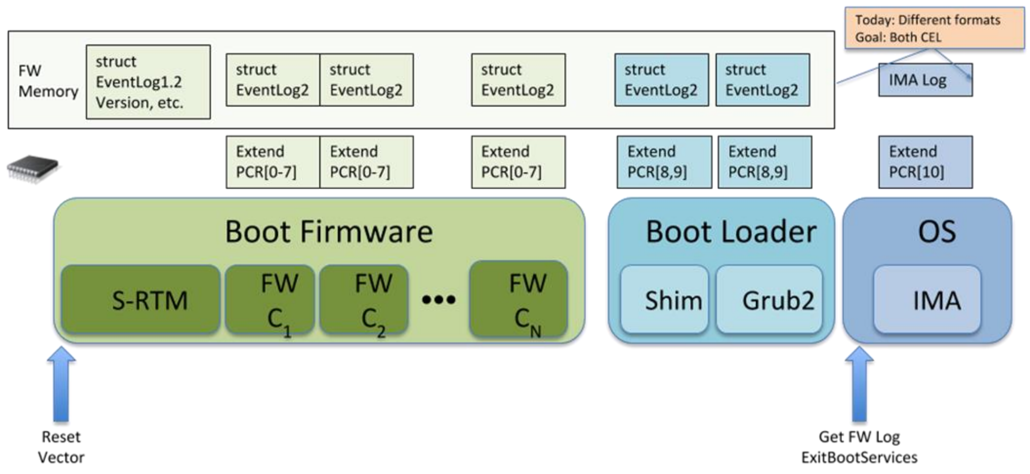

Fig. 3: Boot Sequence of (x86 / UEFI / TPM2.0)

• **Step 2:** The platform supplier will provide a self-signed platform certificate. System user can create platform attributes certificate to get details about hardware modules are parts present on the device.

• **Step 3**: The platform supplier will reference and bind the platform attributes and TPM certificate generated in the previous step. This step will make TPM to be exclusive to the platform. Platform attributes certificate will provide details about what other hardware modules are present onto the platform, which are mutable and non-mutable components.

• **Step 4**: both EK and platform certificates are stored the in NV storage of TPM2.0.

In order to attest platform RV needs to verify all of the above steps.

2) Firmware Validation: Firmware is a collection of codes stored on a small memory chip of a device. It provides the necessary instructions for the device to communicate with other hardware and software modules within and outside the device. The device uses flash ROM to store firmware, and it is semi-permanent unless it is changed or upgraded. Understanding the platform boot sequence (x86 / UEFI / TPM2) is a useful to perform firmware and software remote attestation. Fig.3 shows the boot sequence of X86 / UEFI / TPM2 based hardware device.

Root of trust measurement will be done by following four basic operations namely

LOAD
MEASURE
EXTEND
EXECUTE

as seen from Fig-3, When the systems powers on, reset vector will first LOAD the Static Root of Trust Measurement (S-RTM) component of boot firmware. System MEASURES its code by taking a secure hash, EXTENDS it into PCR, creates first event record into firmware event logs in firmware memory. Next, the system EXECUTEs S-RTM module code and gives the information regarding the next boot image. The program counter will point to the next firmware image location, and the system will perform the same - LOAD, MEASURE, EXTEND and EXECUTE operations on the next firmware component (e.g., FW C1). Each firmware component will follow the same boot sequence and will have an event extended into PCR0- PCR7.

The system will generate an event record in firmware event logs. The last firmware boot component will point to the Shim or shimx64 as the next stage bootloader image. Which, in turn, will call grub or grub2, followed by loading Operating System (OS). These will have events extended in PCR8, and PCR9 and firmware event logs will have record of each event.

As seen in Fig-3, today firmware and software event logs are in different format and does not have sequence numbers, which makes hard to transfer meaningful binary blob over to RV. one of the goal of SEDAT is to convert both event logs into Cannonical Event Logs (CEL) structure as recommended by TCG.

The malware's like ransomware tries to change firmware code or device boot sequence to lock the device and resources. So, the assurance of the integrity of firmware is important for device attestation along with the boot sequence integrity. RV needs to verify firmware event logs.

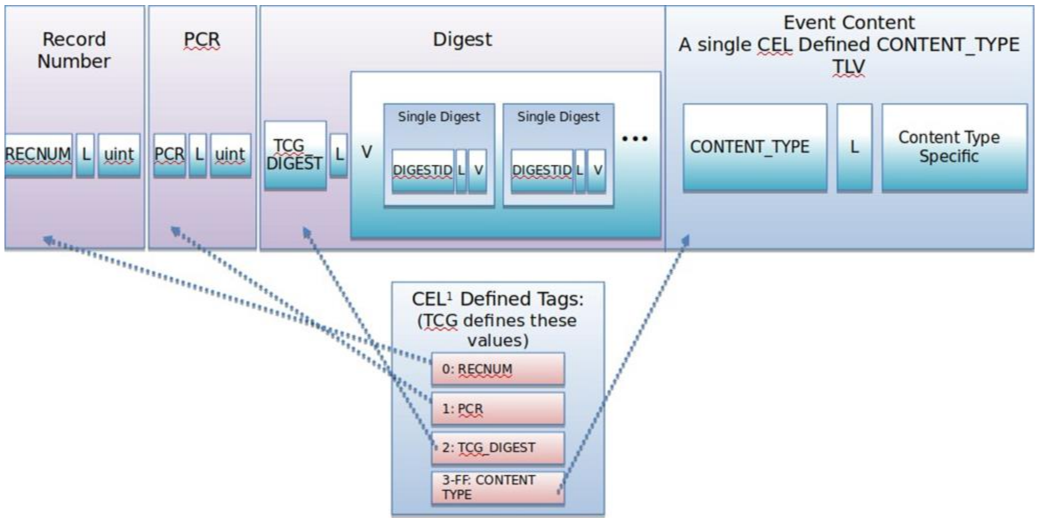

Fig. 4. CEL IMA eventlogs

3) Software Validation: Remote attestation of all the software on a device is a relatively hard and resource-heavy process. Instead, we can have a certain portion of the software modules attested, to ensure the critical portion of the software code and data is intact. Trusted Computing Group (TCG) has recommended a standard for software integrity check- called Integrity Measurement Authority (IMA). Which signs, measures, and extends the IMA secure region of software into PCR10 of TPM2.0. It will also generate entry in IMA event logs. RV needs to verify IMA event logs for the IMA software integrity check.

4) Quote Validation: TPM2.0 has a function called quote generation. The devices' AK, selected encryption algorithm, and PCR values are used to generate a quote. TPM2.0 uses nonce for adding freshness to the quote, which is sent from the verifier for the added layer of security and replay protection. RV needs to validate the quote.

The next section describes the assumptions and limitations we considered while designing SEDAT.

B. Assumptions and Limitations of SEDAT

SEDAT is implemented keeping hardware as root of trust us- ing TPM2.0. All the devices provisioned with SEDAT requires to have hardware or firmware TPM2.0. Devices should have installed intel's latest TPM2-TSS, TPM2-abrmd and TPM2- tools. SEDAT assumes that, there is one time trusted secure channel for verifier to get all the root certificates from the device vendor. SEDAT uses single packet authorization for securing communication channel between prover and verifer. So, SEDAT assumes that pre-shared secret key is loaded at both ends before communication start between varifier and prover. It is protected from replay and DoS attacks but covering some advance attacks are out of scope of this research. All prover devices are required to be patched with provided IMA and firmware patches and latest linux kernel to get firmware and IMA event logs in CEL format. Protecting the prover or verifier from physical, side channel attacks is out of scope. SEDAT can be validated on Software TPM with intels TPM2 stack.

Architecture building blocks for SEDAT

After understanding theory and design requierments SEDAT is designed in moduler fashion.each module is performing dedicated task as follow.

A. Device provisioner

This is the first communication anchor between prover and verifier. Prover sends a hello message to the verifier. Verifier sends counter value to prover in response. Prover uses this counter value and to calculate Hashed One Time Password (HOTP) using pre-shared key and sends the HTOP to the verifier. The verifier compares the received HOTP with the one it has computed, if those matches, SEDAT enrolls the client to remote verifier by sending ack signal else the connection will is dropped. Prover sends its device information, Bios details, OS details to the verifier in the following message and device provisioning task is finished by recording response into database.

B. Platform attestation

This tool perform all the steps listed in platform validation subsection. also it take the ownership of the platform and TPM module as shown in Fig- 5.

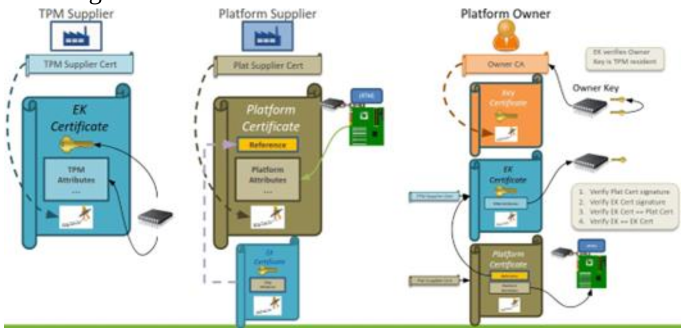

Fig. 5: Platform and TPM supplier certificates binding, establishing platform ownership

C. Firmware and software event logs generation

This are combination of patches and scripts which pulls the firmware and IMA event logs into user space and using tools it converts them into CEL format.

Fig 4 depects the new Canonical Eventlog Structure (CEL) for IMA and Firmware events recommended by Trusted Computing Group (TCG). each field in the eventlog has Tag Lenght and Value (TLV) parameters. for firmware first event is TPM1.2 eventlog format as per pc client specification by TCG which has information about bios firmware versions, events are crypto aligned, supported hashing algorithms etc. second event onwards are actual events for firmware as explained before. it will have PCR number which is 32 bit value between PCR0- PCR9 for firmware eventlogs and PCR10 for IMA. Digest will give information about hashing algorithm used for that event. Length of Digest field will be dependet on hashing algorithm used for that event. if it is SHA1 length will be 20 bytes, SHA256 length will be 32bytes and so on.Value field in the Digest will hold extended PCR value. Event content field will hold the actual data in TLV format. on top of all this eventlog has the sequence number field for firmware and IMA eventlogs to make the records more meaningfull and easy to understand when transfer over to verifier.

D. Quote attestation

Quote check is the mechanism used in TPM based attesta- tion to validate identity and authentication of the platform. Mostly prover generates the quote and validates the quote locally. There is no known solution available, which does quote check at remote verifier. this tool is a quote verifier which runs couple of scripts on prover and gets quote at RV and following same process it regenerates the quote at RV and matches it.

The next section explains workflow of the SEDAT frame- work.

SEDAT: framework design

The framwork of SEDAT works as follow

• **Client provisioning:** In this step untrusted prover will establish secure communication channel with the remote verifier.

• **Get TPM certificate:** Using TPM2.0 command get the TPM EK key and create endorsement certificate from the manufacturers root certificate site.
• **Get platform certificate:** Using TPM2.-0 command to get platform certificate and run paccor to create platform attributes certificate.
• **Store certificates in TPM:** Using TPM2.0 commands store the platform certificate and endorsement certificate into NV storeage of TPM2.0.
• **Take ownership:** Using TPM2.0 command take owner- ship of the platform.
• **Get eventlogs for Firmware and IMA in CEL for- mat:** Using provided scripts get the firmware and IMA event logs from /sys/kernel/security/tpm0/binary* and
/system/kernel/security/ima/binary* respectively and send it to RV.
• **Generate quote:** Using tpm2 quote command and added freshness nonce generate quote.
• **Validate quote at RV:** based on received information at the remote verifier regenerate and check the quote. this is implemented using IBM's software tpm as verifier needs not to have tpm module.

Implimentation/ Tools and Techniques

Verifier and tools are implemented in go language and used postgres as backend database. we took insperation from National Security Authorities (NSA)'s tool for platform varification called HIRS. SEDAT verifier uses NSA's tool called paccor to generate platform certificate and platform attributes. it stores those two certificate along with manufacturer's root CA certificate into the postgres database as golden template. SEDAT transfers the golden CEL firmware and IMA event logs to the remote verifier using replay protected secure channel. we provide full control to verifier to enable or disable certificates, firmware event logs and IMA logs validation.

Use Cases

SEDAT can be applied to different application areas.here we showcase three usecases.

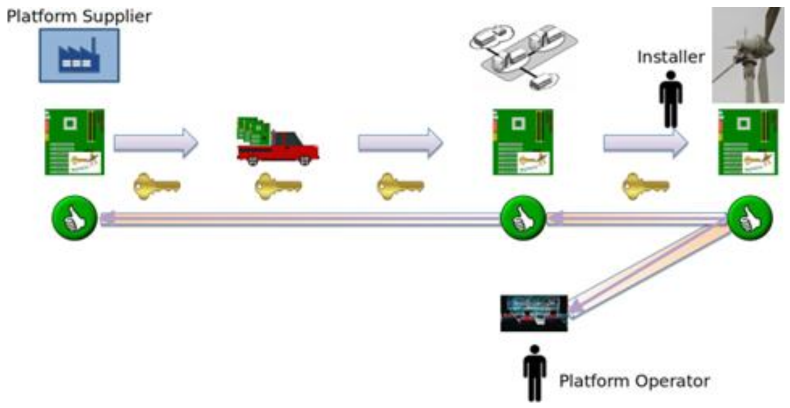

Fig. 6: Supply Chain

• **Supply Chain Validation:**

In supply chain platform supplier will be located in different geolocation and it transports the devices from the assembly line to warehouse, then it will go to retail location or to the installation facility. In this transport process, there are multiple untrusted anchors involved which can lead to counterfiet, substitute the devices.tpm based attestation provides the hardwae root of trust, signed key, certificate validation will increase security and trust. our verifier helps in tracking with reduced cost and increased trust. it also reduces the in situ installation and replacement cost.it is possible to remote key provisioning. key allows trusted remote configuration, trust channels using keys allows multiplexing connections reducing cabling costs.with SPA the communication is replay and DoS protected.

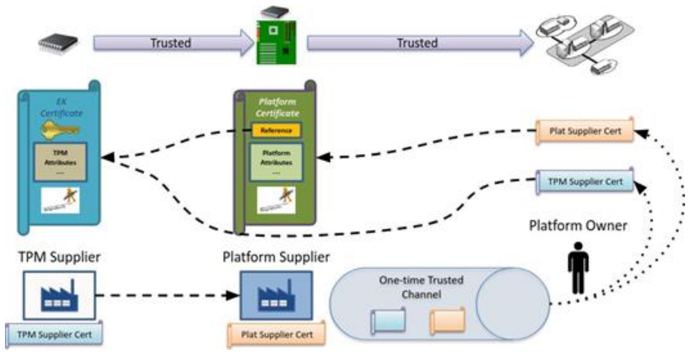

Fig. 7: One-time secure channel

As shown in Fig.7 we need to have one-time trusted channel between platform owner and tpm and platform supplier to transfer securely the platform supplier cert and tpm supplier cert to platform owner and it will be binded and referenced as discussed before. we are using standard https connection for now.

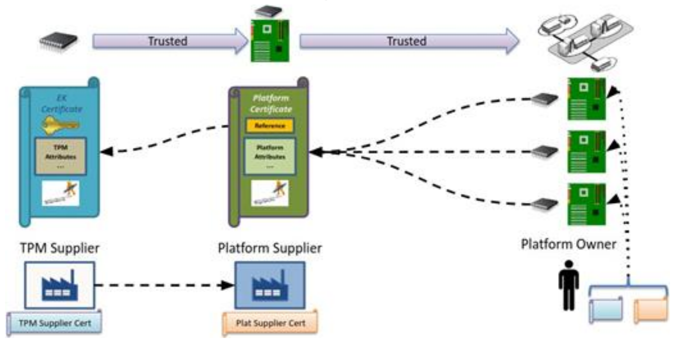

Fig. 8: Supply Chain Validation

As shown in Fig.8 shows supply chain validation uses case. the connection for now.

• **Inventory management:**

In large organization employer will give computers to emplyees which they use for all corporate works so integrity, authenticity and inventory management of those coputer devices are must. SEDAT can be used as remote attestation verifier as most of the computers now a days have TPM chip. SEDAT can also be laverage to list information of mutable and immutable components on computers as it was shown in [4]. this helps in boot time security and inventory management. if some employee has broken display or ethernet port while on vacation or travel and they replaced untrusted replacement part with SEDAT trust anchor installed verifier has golden state of all mutable immutable components on the device with was given to employee on day one. so next time when attestation takes place verifiers platform certs will not match and we can trigger an alert for appropriate action.

• **Industrial control systems:**

SEDAT can be used as remote verifier for controllers and embedded devices with hardware or firmware based TPM module. in those environment verifying the integrity and authenticity of the platform is key factor.

Evaluation

We have compared SEDAT with other solutions and found that SEDAT outperforms others by providing one stop solution for Hardware firmware and software remote verification with single packet authentication to protect from replay and DoS attacks. below figure shows the evaluation report.

Table 1: Comparision between SEDAT v/s other solutions

Parameters	**IBM-ACA**	**Sublime**	**HIRS-NSA**	**SEDAT**
Endorsement certificate	Y	N	Y	Y
Platform certificate	N	N	Y	Y
Platform attributes Certificate	N	N	Y	Y
Platform mutable components	N	N	Y	Y

CEL Firmware Event logs	N	N	N	Y
CEL IMA Event logs	N	N	N	Y
Quote generation	Y	Y	N	Y
Quote check at RA	N	N	N	Y
replay /DoS protection	N	N	N	Y
Multi-OS support	Y	Y	N	Y

A. Future Work

We are planing to enchance our varifier to include secure communication and replay protection. also we are interested in looking into secure communication protocol to transfer the manufacturers root CA and endorsement cert securely to the verifier as currently SEDAT assume that there is a one time secure channel for transfering those to verifier. we are motivated to close the loop of remote varifier and take actions once it detects a problem in attestation.

Related Work

Remote Verifier (RV) allows a trusted entity to securely measure internal state of the remote unstrusted platform (prover). RV can be used to establish static or dynamic root of trust in cyber-physical and industrial controls systems. It can be used as building block for other security services and primitives, such as provisioning, updates, patches. Current attestation approaches fall into two domains namely collective attestation and single device attestation.

• Collective Attestation:

Traditional attestation schemes consider only a single prover and verifier. Swarm/collective attestation aims at scaling existing attestation schemes to networks of embedded devices, by leveraging in-network verification [9], and novel cryptographic primitives [10]. SEDA

[9] investigates the security of swarms of embedded de vices. It presents the first attestation protocol for large swarm, allowing a central verifier to assess the trustworthiness of a million device swarms in order of seconds. It achieves this by distributing the attestation burden across the swarm, allowing neighbors to attest each other, and aggregating the attestation results in a hop- by-hop manner. SANA [10] enables low verification overhead due to the integration of a novel Optimistic Aggregate Signatures (OAS), which is a generalization of aggregate and multi-signatures. Finally, DARPA [11] aims at detecting software compromise and device capture in embedded networks. Since DoS attacks on collective attestation are more significant, as it allows to adversary to disturb a large network by targeting one device, SANA [10] proposes using secure tokens obtained from a trusted third party to prohibt scaling DoS attacks to large networks. However, SANA uses expensive public key cryptography which imposes additional overhead on the Prv, and is vulnerable to DoS attacks based on the digital signature verification procedure.

• Single Device Attestation:

It has three main categories: Software-based attestation schemes [12]–[17]. does not require secure hardware. It enables attestion of legacy and low-end embedded devices with some assumptions. These assumptions are: adversery is not active during the attestation process, the attestation code and implementation are optimal, and presence of an out-of-band authentication channel. Due to this rea- sons, security of software-based attestation schemes has been challenged by [?], [18], [19] and their applicability and reliability was limited. Hardware-based attestation schemes [20]–[25] provide better security guarantees. Software/Hardware Co-design or Hybrid schemes such as [?], [26]–[28] provides examples of minimal hardware- based features required for enabling secure remote attestation. Such security features are as simple as a Read Only Memory (ROM), and a simple Memory Protection Unit (MPU). SEDAT is the first remote attestation scheme, which performs hardware, firmware and software attestation and it is completely secure against Denial of Service (DoS), replay attacks.

Conclusion

In this paper we have presented proof of concept work for remote verifier to perform end to end attestation of hardware, firware and software of untrusted device with tpm2.0. to the best of our knowledge SEDAT is the first solution to demor- nstrate one stop solution for all three components varification with INTEL's /TCG recommended tpm2-tools stack. we are the first one to auther tools for represnting IMA and Firmware eventlogs into CEL format. all codes is open-sourced and available for future research.

Acknowledgment

The authors would like to thank faculties of University of Maryland Baltimore County (UMBC) and research group at General Electric global research for funding, actively partici- pating and guiding the research work.

References

[1]. B. Schneier, "Schneier on Security," in Wwwschneiercom, 2018, p. 336. [Online]. Available: https://www.schneier.com/blog/archives/2018/ 10/chinese{ }supply{ }.html

[2]. J. Vijayan, "Stuxnet renews power grid security concerns," in Stuxnet renews power grid security concerns, june 2010. [Online]. Available: https://www.computerworld.com/article/2519574/ stuxnet-renews-power-grid-security-concerns.html

[3]. D. Schneider, "Jeep Hacking 101," in Jeep Hacking. [On- line]. Available: https://spectrum.ieee.org/cars-that-think/transportation/ systems/jeep-hacking-101

[4]. NSA, "HIRS provisioner," 2017. [Online]. Available: https://github. com/nsacyber/HIRS

[5]. K. Goldman, "IBM-ACS," 2017. [Online]. Available: https://sourceforge.net/p/ibmtpm20acs/activity/?page=0{&}limit= 100{#}5cc3737aee24ca5b73320e9c

[6]. B. Weeks, "No Title," 2019. [Online]. Available: https://github.com/ google/go-attestation

[7]. N. Schear, P. T. Cable, II, T. M. Moyer, B. Richard, and R. Rudd, "Bootstrapping and maintaining trust in the cloud," in Proceedings of the 32Nd Annual Conference on Computer Security Applications, ser. ACSAC '16. New York, NY, USA: ACM, 2016, pp. 65–77. [Online]. Available: http://doi.acm.org/10.1145/2991079.2991104

[8]. A. Ibrahim, A.-R. Sadeghi, and S. Zeitouni, "SeED: Secure non- interactive attestation for embedded devices," in Proceedings of the 10th ACM Conference on Security and Privacy in Wireless and Mobile Networks, ser. WiSec '17. New York, NY, USA: ACM, 2017, pp. 64–74. [Online]. Available: http://doi.acm.org/10.1145/3098243.3098260

[9]. N. Asokan, F. Brasser, A. Ibrahim, A.-R. Sadeghi, M. Schunter, G. Tsudik, and C. Wachsmann, "SEDA: Scalable embedded device attestation," in Proceedings of the 22Nd ACM SIGSAC Conference on Computer and Communications Security, ser. CCS '15. New York, NY, USA: ACM, 2015, pp. 964–975. [Online]. Available: http://doi.acm.org/10.1145/2810103.2813670

[10]. M. Ambrosin, M. Conti, A. Ibrahim, G. Neven, A.-R. Sadeghi, and M. Schunter, "Sana: Secure and scalable aggregate network attestation," in Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security, ser. CCS '16. New York, NY, USA: ACM, 2016, pp. 731–742. [Online]. Available: http://doi.acm.org/10.1145/2976749.2978335

[11]. A. Ibrahim, A.-R. Sadeghi, G. Tsudik, and S. Zeitouni, "Darpa: Device attestation resilient to physical attacks," in Proceedings of the 9th ACM Conference on Security &#38; Privacy in Wireless and Mobile Networks, ser. WiSec '16. New York, NY, USA: ACM, 2016, pp. 171–182. [Online]. Available: http://doi.acm.org/10.1145/2939918.2939938

[12]. R. W. Gardner, S. Garera, and A. D. Rubin, "Detecting code alteration by creating a temporary memory bottleneck," IEEE Transactions on Information Forensics and Security, vol. 4, no. 4, pp. 638–650, Dec 2009.

[13]. R. Kennell and L. H. Jamieson, "Establishing the genuinity of remote computer systems," in Proceedings of the 12th Conference on USENIX Security Symposium - Volume 12, ser. SSYM'03. Berkeley, CA, USA: USENIX Association, 2003, pp. 21–21. [Online]. Available: http://dl.acm.org/citation.cfm?id=1251353.1251374

[14]. A. Seshadri, M. Luk, and A. Perrig, "Sake: Software attestation for key establishment in sensor networks," Ad Hoc Netw., vol. 9, no. 6, pp. 1059–1067, Aug. 2011. [Online]. Available: http://dx.doi.org/10.1016/j.adhoc.2010.08.011

[15]. A. Seshadri, M. Luk, A. Perrig, L. van Doorn, and P. Khosla, "Scuba: Secure code update by attestation in sensor networks," in Proceedings of the 5th ACM Workshop on Wireless Security, ser. WiSe '06. New York, NY, USA: ACM, 2006, pp. 85–94. [Online]. Available: http://doi.acm.org/10.1145/1161289.1161306

[16]. A. Seshadri, M. Luk, E. Shi, A. Perrig, L. van Doorn, and P. Khosla, "Pioneer: Verifying code integrity and enforcing untampered code exe- cution on legacy systems," 2005.

[17]. A. Seshadri, A. Perrig, L. V. Doorn, and P. Khosla, "Swatt: Software- based attestation for embedded devices," in In Proceedings of the IEEE Symposium on Security and Privacy, 2004.

[18]. C. Castelluccia, A. Francillon, D. Perito, and C. Soriente, "On the difficulty of software-based attestation of embedded devices," in Proceedings of the 16th ACM Conference on Computer and Communications Security, ser. CCS '09. New York, NY, USA: ACM, 2009, pp. 400–409. [Online]. Available: http://doi.acm.org/10.1145/1653662.1653711

[19]. G. Wurster, P. C. v. Oorschot, and A. Somayaji, "A generic attack on checksumming-based software tamper resistance," in Proceedings of the 2005 IEEE Symposium on Security and Privacy, ser. SP '05. Washington, DC, USA: IEEE Computer Society, 2005, pp. 127–138. [Online]. Available: https://doi.org/10.1109/SP.2005.2

[20]. X. Kovah, C. Kallenberg, C. Weathers, A. Herzog, M. Albin, and J. Butterworth, "New results for timing-based attestation," in Proceedings of the 2012 IEEE Symposium on Security and Privacy, ser. SP '12. Washington, DC, USA: IEEE Computer Society, 2012, pp. 239–253. [Online]. Available: https://doi.org/10.1109/SP.2012.45

[21]. J. M. McCune, Y. Li, N. Qu, Z. Zhou, A. Datta, V. Gligor, and Perrig, "Trustvisor: Efficient tcb reduction and attestation," in 2010 IEEE Symposium on Security and Privacy, May 2010, pp. 143–158.

[22]. J. M. McCune, B. J. Parno, A. Perrig, M. K. Reiter, and H. Isozaki, "Flicker: An execution infrastructure for tcb minimization," SIGOPS Oper. Syst. Rev., vol. 42, no. 4, pp. 315–328, Apr. 2008. [Online]. Available: http://doi.acm.org/10.1145/1357010.1352625

[23]. N. L. Petroni, Jr., T. Fraser, J. Molina, and W. A. Arbaugh, "Copilot - a coprocessor-based kernel runtime integrity monitor," in Proceedings of the 13th Conference on USENIX Security Symposium - Volume 13, ser. SSYM'04. Berkeley, CA, USA: USENIX Association, 2004, pp. 13–13. [Online]. Available: http://dl.acm.org/citation.cfm?id=1251375.1251388

[24]. R. Sailer, X. Zhang, T. Jaeger, and L. van Doorn, "Design and implementation of a tcg-based integrity measurement architecture," in Proceedings of the 13th Conference on USENIX Security Symposium - Volume 13, ser. SSYM'04. Berkeley, CA, USA: USENIX Association, 2004, pp. 16–16. [Online]. Available: http://dl.acm.org/citation.cfm?id=1251375.1251391

[25]. D. Schellekens, B. Wyseur, and B. Preneel, "Remote attestation on legacy operating systems with trusted platform modules," Sci. Comput. Program., vol. 74, no. 1-2, pp. 13–22, Dec. 2008. [Online]. Available: http://dx.doi.org/10.1016/j.scico.2008.09.005

[26]. F. Brasser, B. El Mahjoub, A.-R. Sadeghi, C. Wachsmann, and P. Koeberl, "Tytan: Tiny trust anchor for tiny devices," in Proceedings of the 52Nd Annual Design Automation Conference, ser. DAC '15. New York, NY, USA: ACM, 2015, pp. 34:1–34:6. [Online]. Available: http://doi.acm.org/10.1145/2744769.2744922

[27]. K. E. Defrawy, D. Perito, G. Tsudik, and et al., "Smart: Secure and minimal architecture for (establishing a dynamic) root of trust," in IN: PROCEEDINGS OF THE 19TH ANNUAL NETWORK AND DISTRIBUTED SYSTEM SECURITY SYMPOSIUM, pp. 5–8.

[28]. A. Francillon, Q. Nguyen, K. B. Rasmussen, and G. Tsudik, "A minimalist approach to remote attestation," in Proceedings of the Conference on Design, Automation & Test in Europe, ser. DATE '14. 3001 Leuven, Belgium, Belgium: European Design and Automation Association, 2014, pp. 244:1–244:6. [Online]. Available: http://dl.acm.org/citation.cfm?id=2616606.2616905