MITRE | SOLVING PROBLEMS FOR A SAFER WORLD®

Incorporating Entity Attestation into the P2795 Analytic Exchange Protocol

ENTITY ATTESTATION FOR RESPONDING NODE
CYBER HEALTH AND DATA INTEGRITY

July 18, 2022

Authors:

Mari J. Spina, D.Sc.
Evan J. Rudy
Ben Espey

Abstract

The P2795 Analytic Exchange provides valuable information for interpreting the context of the data communicated when a Requesting Node submits data for analysis and receives the computed analytic from the Responding Node. However, data integrity validation provisions would improve the ability of the Requesting Node to make judgements upon the received data for use in subsequent decision making. This paper summarizes development and test activities conducted by The MITRE Corporation to demonstrate the use and utility of adding Entity Attestation Token (EAT) claims into the P2795 message exchange specifically for the purpose of interpreting the cybersecurity health of the Responding Node and the integrity of the data computed by the Responding Node. If implemented as part of the standard, EAT constitutes a Zero Trust security enhancement to the P2795 protocol [3].

Background

As part of MITRE's Smart Connected Analytic Learning Exchange (SCALE) project, an entity cyber health and computation integrity attestation solution for the P2795 standard development was demonstrated. The demonstrated solution is based upon use of the Internet Engineering Task Force (IETF) Remote Attestation Procedures (RATS) Working Group Entity Attestation Token (EAT)-01 draft standard. The Standard identifies an array of claims for use by cyber assurance systems and provides a means to develop and apply system specific claims.

The National Security Agency (NSA) suggests a "Zero Trust Mindset" for cybersecurity and operational capabilities that "never trust, always verify" [6]. Use of entity attestation for device health and computational integrity validation is a reasonable approach for integrating Zero Trust capabilities into the P2795 standard. For this work, EAT-01 standard device and unique software specific claims were defined to enable the application of Zero Trust principles to the P2795 analytic information exchange as illustrated in **Table *1***.

Table 1. EAT-01 Claims Applied

Para. #	Claim Name	Demo Implementation
3.1	Nonce Claim (cti and jti)	Random Number Generation
3.2	Timestamp claim (iat)	Network Time Protocol Service
3.3	Universal Entity ID Claim (ueid)	Machine ID
3.4	Origination Claim (origination)	Processor Chip Version
3.5	OEM identification by IEEE OUI (oemid)	MAC Address
3.10	The Uptime Claim (uptime)	Device Uptime
3.12	The Submods Claim (submods)	Analytic SW Hash

Demonstration

The SCALE Entity Attestation demonstration involved two Linux compute platforms deployed as virtual machines and configured to handle the P2795 message exchange over SSH v2, using certificate-based authentication. Following the Requesting node's call for analytic results, the Responding node provided an EAT response attached to the results. In the demonstration setup, the Requesting node assumed the role of EAT Verifying Host and the Responding node assumed the role of Attestation Host. An EAT was formulated by the Attesting Host per the EAT-01 draft standard [2]. The EAT claims were formulated and included as payload for transmission inside a JSON Web Token (JWT), as defined by RFC 7519. Creation of the JWT performed a cryptographic hash of the payload using the HMAC-SHA256 hashing algorithm and signed using a pre-shared symmetric key [1]. As illustrated in Figure 1, the JWT was then sent to the Verifying Host using SSH channel encryption. The JWT was then unpacked by the Verifying Host and the original claims were retrieved. The Verifying Host compared the received claims against a set of expected claims to evaluate the attestation content.

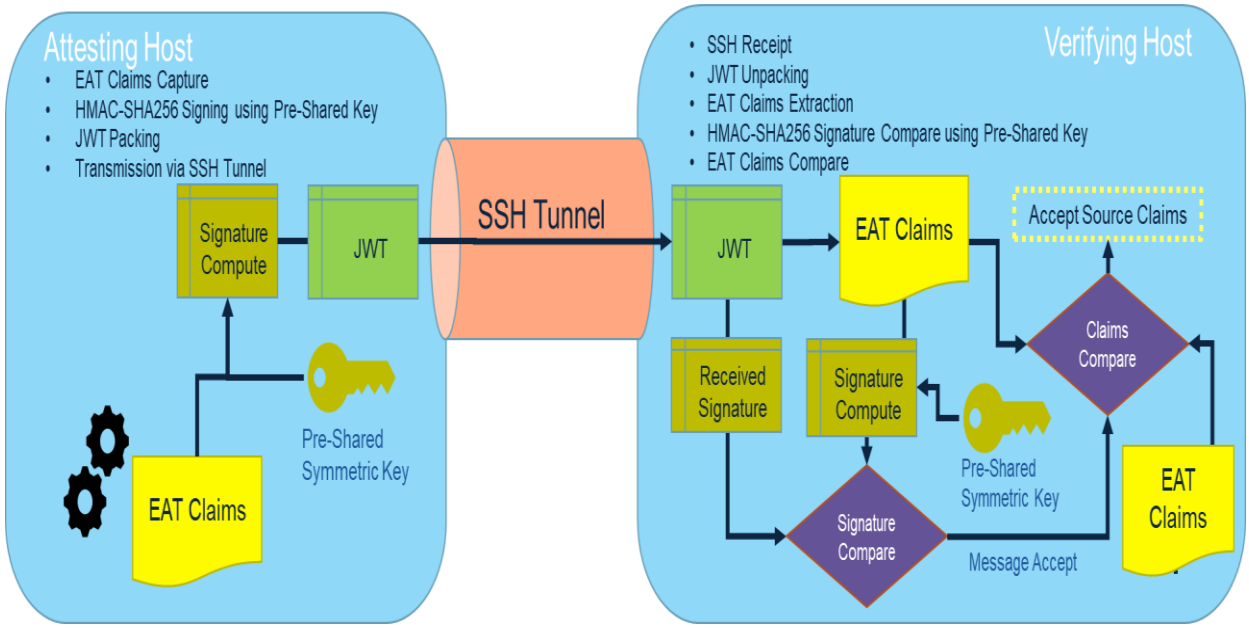

Figure 1. EAT Creation, Transmission, and Unpacking using JWT

Four demonstration runs of the EAT message exchange were performed as indicated in Figure 2. In each run, the EAT claims were created at the Attesting Host to either pass of fail upon comparison at the Verifying Host. Demo #1 represents an "all-clear" use case while demos #2, #3, and #4 represent cases where some inconsistency exists in the received token that causes a comparison failure.

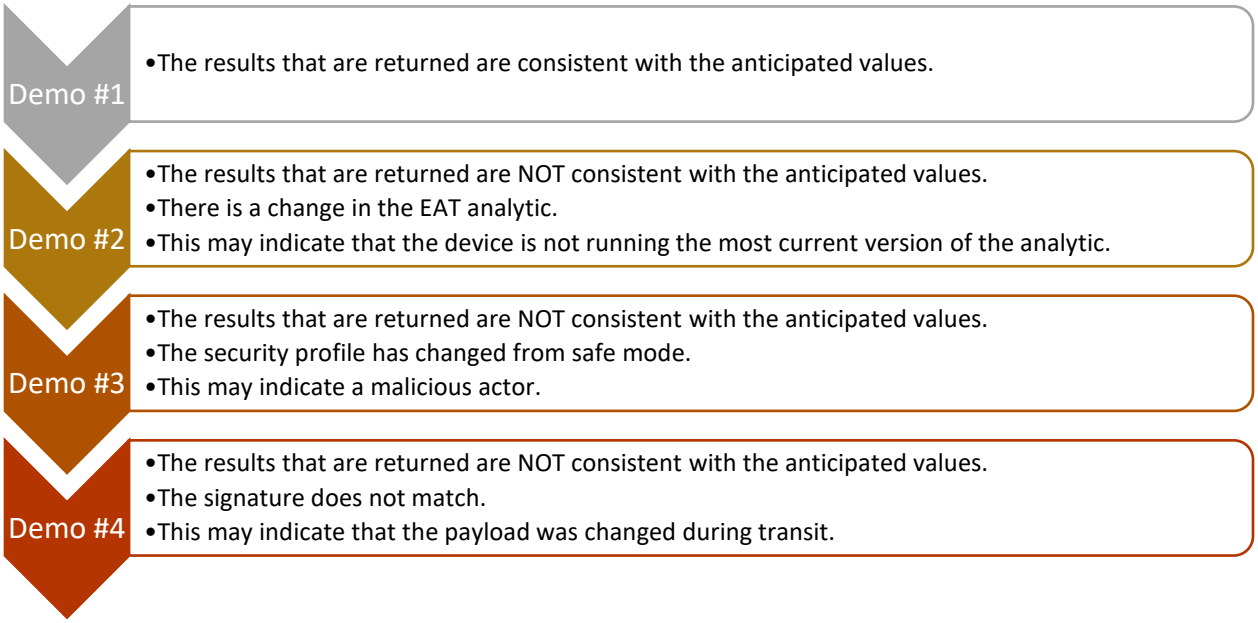

Figure 2. EAT Handling Demonstration Runs

Figure 3 provides an example of the EAT claims at the Attesting Host. The time stamp claim (iat) and computer uptime (uptime) are captured at runtime. The nonce is a random number supplied by the Verifying Host and returned in the EAT unmodified. The other claims used in the demonstration are static values captured from the Attesting Node operating system to include the ueid, oemid, hardware version, and hash of the analytic source. While use of a claim for the security mode upon boot of the machine is recommended as a check on device health, the Attesting Host device did not provide access to the necessary information. This is because the experiment was performed on virtual machines that did not provide access to a trusted protection module, or TPM, that is cryptographically secure and would traditionally be used to perform cryptologic operations. Use of the SHA256 hashing algorithm for the analytic binary was chosen to demonstrate a claim useful for application to the P2795 protocol, and as a means for providing a compute integrity check.

```
{
    "iat": "2022-02-22T14:08:17-05:00",
    "nonce": "8675309",
    "ueid": "4da6af5234774bf4b29fc5360e35dad5",
    "oemid": "00:50:56:82:59:a5",
    "hardware-version-claims": "Intel Xeon Gold 5220R",
    "uptime": "2957944.69",
    "manifests":
    {
        "analytic":
        {
            "id": "1",
            "name": "analytic_v1.py",
            "sha256sum": "9d598e04d0e666a00d4ea527cff0de7fef2bda7c654e2f2b1bb212da7c7cef03"
        }
    }
}
```


Figure 3. EAT Claims Programming

Technical Findings

This work represents a valuable demonstration of the use of entity attestation in the context of the P2795 Analytic Exchange protocol. The EAT standard is observed to be lightweight and sufficiently flexible to address device health and compute integrity attestations proposed for addition to the P2795 protocol.

While compute performance was not measured for this demonstration, EAT represents a computationally lightweight algorithm that, itself, imposes little computational impact specifically when claim collection can be accomplished by file lookup and read operations. However, selection of cryptographic routines and the signing of the JWT token is expected to bear on system performance. It is noted that the completed JWT token is a relatively small data package at an estimated typical 500 bytes. As a result, it is not expected to noticeably burden today's internet communication links. As mentioned below, the JWT token could be made smaller through design choices if link utilization was at a premium (e.g., using different algorithms to compute values, truncation of values, inclusion/exclusion of fields, etcetera).

Flexibility in the EAT standard is derived from its allowance of nested and submodule (submod) claims. The submodule claim structure allows for the definition of sub-system specific claim names, types, and content. It is particularly useful for implementing claims specific to one or more subsystems operating within a single hardware platform. In this demonstration, the "manifest" claim is a submod claim specific to the software payload being allowed to execute on the platform.

The nested claim capability allows for the addition of claims specific to a subsystem in use cases where many claim types are useful. In this demonstration, the "analytic" claim is a nested claim within the manifest submodule. The "analytic" claim was named "analytic" because it represents the analytic software used in the P2795 information

exchange. With such a structure, multiple software specific claims could have been made. Such a capability can be used to capture claims about other software packages or other claim types related to the same software package.

The use of the SHA256 checksum computation for the analytic claim demonstrates the flexibility in which claim content can be defined. Application of a hashing function for operation on a software binary to provide a measure of software integrity is a commonly accepted software security measure. The authors note the SHA256 algorithm was chosen because of its FIPS 140-3 compliance [4] but other integrity verification solutions could have been implemented, depending on system requirements. For example, on an embedded system, it would be possible to use a less computationally intensive hashing algorithm (perhaps at the expense of integrity assurance, however). Exploring these potential tradeoffs is an area for future research.

For applications of device attestation in environments that are both highly compute and communication bandwidth limited, it is noted that use of the EAT standard may prove to be especially advantageous where encryption in transit for claim confidentiality is desired but the overhead processing of an SSH or TLS connectivity is not. An emerging JSON Web Encryption (JWE) standard [5], currently supported by some JSON development libraries, could easily be implemented when the EAT is implemented as a JWT; the approach demonstrated in this research. Such a solution is suggested for future research.

Recommendation for IEEE P2795 Committee Consideration

The authors recommend that an EAT message exchange be added to the P2795 Analytic Exchange protocol to enable Zero Trust-based entity attestation. Attestation addressing cybersecurity health and compute integrity of the P2795 Responding Node can add valuable information to inform the recipient of the P2795 analytic information. Attestation comparison results can be used as a measure of information trustworthiness to guide decisions associated with its use in subsequent activities, analytics, or in support of medical procedures.

References

[1] JSON Web Tokens - jwt.io
[2] https://tools.ietf.org/id/draft-ietf-rats-eat-01.htm
[3] https://sagroups.ieee.org/2795
[4] FIPS 140-2 - Annex A (nist.gov)
[5] RFC 7516 - JSON Web Encryption (JWE) (ietf.org)
[6] CSI_EMBRACING_ZT_SECURITY_MODEL (defense.gov)