The TPM and OVAL

Using the Trusted Platform Module to Enhance OVAL Driven Assessments

Introduction

Recently, MITRE/SEDI developed a new draft component schema for OVAL to support interaction with the Trusted Platform Module (TPM). At a basic level, this is no different than most other component schemas: it simply expands OVAL's ability to collect system state information into a new component. However, because of some of the unique features of the TPM, this expansion of OVAL also supports a greater degree of trust in the results provided by an OVAL assessment. This document is intended to educate the OVAL community about TPMs in general and about the exciting possibilities made possible by OVAL interactions with the TPM.

The TPM

The TPM is a hardware chip permanently attached to the motherboard of a computer. The TPM was originally designed by the Trusted Computing Group[1], which publishes the official TPM specifications[2]. Today, TPM chips are built by several vendors and are included in virtually all desktop and laptop machines sold today. (A notable exception is Apple, whose machines do not include TPMs.)

The TPM is a tamper-resistant piece of hardware intended to provide a trusted base for certain activities. It contains built in memory and logic to support activities like encryption and data signing in a way that prevents certain sensitive pieces of information from being exposed under most reasonable circumstances. While the TPM is not 100% proof against tampering (physical access to a TPM chip along with the right tools can allow one to compromise its secrets) the TPM is strongly protected against any remote attempt to compromise its functioning.

TPM Features

The TPM has a number of special features that can be used to enhance a host's security. The first is on-chip storage of the root keys for storage and identity key chains. These are the Storage Root Key (SRK) and Endorsement Key (EK), respectively. These keys never leave the TPM and, because of a TPM's tamper-resistance, can never be discovered under normal circumstances. A TPM also contains several Platform Configuration Registers (PCRs). PCRs, which are reset with every boot, can be used to store measurements of system software. PCRs cannot be set directly, but are instead "extended" using the hash of a combination of the previous PCR value and some new value. Although it is possible for applications on a system to change PCR values, the use of the "extend" operation makes it impractical to turn an arbitrary PCR current value (e.g., one indicating a compromised piece of software) into some desired value (e.g., one indicated that

[1] http://www.trustedcomputinggroup.org/

[2] http://www.trustedcomputinggroup.org/developers/trusted_platform_module

piece of software was uncompromised). As such, if a PCR is holding a "good" value, one can have confidence that this was the result of an actual measurement rather than the result of tampering.

Use of certain keys and data is tightly restricted by the TPM. For example, as noted above, the EK is a key (technically a public-private key pair) whose private part never leaves the TPM, and which serves as the root of a chain of identity certificates. (The public part can be exported in a certificate for certificate-chain verification.) However, the TPM only allows the EK to be used to sign certain other keys created on the TPM as well, specifically Attestation Identity Keys (AIKs). In turn, the TPM will only allow AIKs to be used to sign content that the TPM itself has generated. As a result, if one receives AIK-signed content, one can have strong confidence in the provenance of that content because that content can only have come from the TPM itself. The strict data protection and controls the TPM can impose by having its own, protected logic and memory, allows the TPM to be used to provide a trust base for a number of activities.

TPM Uses

The aforementioned features allow the TPM to be used in several ways. The first is to "seal" secrets on the system to specific PCR values. Since PCRs contain measurements of software, this allows certain pieces of information (such as keys to encrypted data) to only be released (i.e., decrypted) if the software binaries in question are in the same state as they were when the data was initially sealed. Thus, access to sensitive information can be prohibited if measurements indicate possible compromise.

A second use is to create "quotes". A quote consists of some subset of PCR values along with a nonce all signed by one of the TPM's AIKs. A remote party may request a quote as a means of assessing whether a particular host's software was in a particular state at the time of measurement. The requester can determine this by comparing the reported PCR values against some "golden values" that the requester has. The fact that this quote is signed by a particular AIK not only protects the integrity of the quote message in transit, but also serves to prove that the measurements are from the TPM associated with that AIK since, as mentioned earlier, the TPM will only use the AIK to sign data it generates. Since the AIK is part of an identity key chain rooted in a TPM's EK, and since the private (signing) parts of these keys never exist unencrypted off of the TPM, a quote gives reliable information about a host's state in a format that is tightly bound to that particular device.

OVAL and the TPM

At a basic level, the TPM represents another piece of system state that OVAL can be used to test. TPMs have inherent information, such as its version and manufacturer, which might be of interest in an OVAL inventory check. In addition, TPMs must be initialized and configured and, as such, a TPM's state could be the subject of configuration policy recommendations. For this reason, expanding the OVAL Language to collect TPM information is a natural step in OVAL's role of supporting inventory, vulnerability, and configuration checking.

At another level, however, one notes that OVAL and the TPM are doing very similar things. Specifically, both technologies support the reporting on aspects of a machine's state to a remote party. TPM quotes are designed to be attributable and reliable. This comes from the combination of the TPM's secure processing and the signing of quotes using the AIK, which can ultimately be tied back to a secret known only to the source TPM. However, the reports provided by the TPM have a limitation in that they lack detail. Each PCR is just a 20 byte register containing a hash of some measurement. As such, while the PCR values can be compared against one or more "correct" values, mismatched values provide no information about why the mismatch occurred. Moreover, most TPMs have no more than a few dozen PCRs, severely limiting the number of distinct pieces of information that can be reported in this way.

OVAL, in contrast, provides a great deal of information. It is capable of interrogating a wide variety of system entities, combining checks into complex assertions of machine state, and returns not only true-false results but all the underlying findings that produced this result. However, OVAL has no inherent root of trust for its assessments. If one were to compromise a system's OVAL interpreter, it would be trivial to falsify results undetectably. Similarly, OVAL results are associated with a host through the use of the <system_info> element, which records information such as the hostname, interfaces, etc. Again, it would be relatively trivial for an adversary to change this data to have one host masquerade as another. While using signature blocks in OVAL files would prevent adversaries from undetectably modifying content after it was created, the signing keys would likely exist on the client's hard drive where an adversary might be able to compromise them.

The strengths and weaknesses outlined above suggest that finding a way to unify TPM and OVAL technologies might be highly beneficial. The next sections outline how such unions might be achieved and what they might be able to accomplish.

Using the TPM and OVAL Together

MITRE/SEDI has developed prototypes for a number of tests that collect information from or about the TPM. Most of these collect TPM state information and would be primarily of interest in normal inventory, vulnerability, and configuration checks. However, MITRE/SEDI also developed a test, named quote_report, which collects a TPM quote. Specifically, it uses an Object that identifies a nonce, an AIK blob, a list of PCRs of interest, and a type of quote (modern TPMs can create both a regular and a legacy quote structure). The test includes a State that, in addition to the properties found in the Object, contains individual PCR values, the signature blob, and a few other pieces of information that a TPM quote reports.

The quote_report structures allow all the standard OVAL assertions to be tested (equality, greater than, etc.) for each of the collected fields. However, it should be noted that some reasonable uses of a TPM quote are outside OVAL's ability to check in a State element. For example, the signature blob cannot be verified as part of a State's assertions. For these types of tests, one would rely on mechanisms outside of OVAL to perform processing on the data returned by the associated OVAL System Characteristics Item. It is this processing of the quote_report data, rather any OVAL assertions about this data, that we will focus on here.

Conveyance of a TPM Quote in OVAL

At the most basic level, the quote_report structures can be thought of as a means to request and receive TPM quotes within the conventions and structures of the OVAL Language. While other structures do exist for encapsulating quote information in XML for transport (most notably the

Trusted Computing Group's Integrity Report Schema[3]), the creation of OVAL structures allows this information to be collected and, with the caveats noted above, processed as an integral part of existing OVAL assessment frameworks. While fully processing of some fields (i.e., the signature checks) are outside the scope of OVAL, the collection of this information for audit purposes can be shown to have benefits.

For example, fielded tools are already using PCR values to detect changes in the state of low level system structures, such as the BIOS and boot loaders. This is accomplished without any need to know any "golden" PCR values, but simply by recording the PCR values at regular intervals and watching for changes, which turn out to be relatively infrequent for the boot stack even in operational environments. The OVAL quote_report structures could be used to collect this information as part of one's regular OVAL assessments. The fact that the TPM's AIK signature automatically provides both integrity and proof-of-provenance for the information in the quote_report means that such audit information is highly reliable even in the absence of any signature infrastructure for the OVAL framework itself. In fact, even if the OVAL software on a client was compromised, the attacker would still be unable to plausibly falsify this part of the OVAL results.

Using the TPM Quote to Secure OVAL Assessments

An even greater degree of security assurance can be achieved by using the TPM to attest to the integrity of the OVAL software stack itself. This can be done by including measurements of the client's OVAL interpreter, along with any libraries that it utilizes, in a PCR. This, combined with measurements of the underlying framework (from the BIOS up to the operating system – components that are already measured by default into the lower numbered PCRs on many systems) provides a base set of measurements of the OVAL interpreter and its dependencies. These measurements are expressed in the quote_report structure, which cannot be modified without resulting in a signature verification failure based on the TPM's AIK. The result is that, even if the OVAL interpreter is compromised by an attacker, the attacker-controlled OVAL interpreter will be unable to forge a good measurement in the PCRs.

On the other hand, a good PCR measurement will provide good assurance that the OVAL interpreter is uncompromised and therefore, the findings it collects and the subsequent evaluations of these findings is reliable. Again, since it is virtually impossible for an attacker to forge a good PCR measurement even with complete control of a system's software, the level of trust in the regular OVAL results in the presence of a good PCR value is increased substantially.

As an added bonus, the TPM-stored measurements of the OVAL interpreter and its dependencies can also be used to protect any signing keys used by the interpreter to integrity protect its results. (Such integrity protection would obviously be necessary or corruption of the results by a man-in-the-middle attack during transmission would nullify any increased trust provided by good PCR measurements). This can be done by using the TPM's sealing feature to protect signing keys. By sealing the keys to good PCR values, the signing keys will be unavailable if any of the OVAL interpreter's dependencies are compromised in a manner that results in changes to these PCR

[3] http://www.trustedcomputinggroup.org/files/temp/6427CBE0-1D09-3519-AD519217F523C9CB/IWG%20IntegrityReport_Schema_Specification_v1.pdf

measurements. This, in turn, prevents a compromised interpreter from signing its forgeries or leaking the key. One could even theoretically use this mechanism to provide assurance of an OVAL interpreter's integrity without even transmitting the quote_report as signed OVAL results would only be possible when the state of the interpreter allowed unsealing of its signing keys.

OVAL Enhancements of TPM Objectives

The result of this use of the TPM together with OVAL provides us with a significantly enhanced ability to trust OVAL results. This not only benefits the existing OVAL assessment framework, but can be used to augment the TPM as well. As noted earlier, TPM PCR measurements are highly reliable but virtually without detail. However, if the integrity of OVAL assessment results can be trusted, then the OVAL results can provide the detail that TPM PCRs lack allowing greater situational awareness. In some cases, OVAL can completely remove the need to store measurements into a TPM's limited supply of PCRs if OVAL checks can be written to assess the application(s) in question. Similarly, OVAL checks can be written to handle the many variations that a particular piece of software might express (e.g., variations in patch application, patch order, configuration, etc.). This variation can lead to combinatorial explosions in the number of "good" PCR values an application might have, but can be handled easily by OVAL. Thus the result of integrating TPM data into OVAL is not only a more secure OVAL infrastructure, but a more expressive and flexible reporting mechanism than the TPM could handle alone.

Remaining Challenges

Unfortunately, while the aforementioned potential is great, the full realization of this model awaits the addressing of a few challenging research questions. The ability to use OVAL as a means to convey TPM quotes, as well as other TPM configuration and state information, is currently realizable, awaiting only the community review of the proposed TPM component schema. However, the other outlined capabilities, which hinge on using the TPM to enhance the trustworthiness of TPM results, require that a few remaining research challenges be addressed. Progress is being made in all the described areas, but the community should be aware of what is left to be done.

Golden Measurement Maintenance

Ultimately, the assurance provided by any measurement infrastructure, the TPM included, rests on that system's ability to maintain the integrity of its measurement functions, the efficacy of those measures in collecting all relevant information, and the ability to adequately interpret these measurements to discern "good" indicators from "bad" ones. With regards to the integrity of the measurement functions, the TPM itself does not provide a root of trust for measurement tools and so must rely on external capabilities in this regard. Fortunately, solutions already exist for this problem in the form of Intel's Trusted Execution Technology (TXT) and AMD's Secure Virtual Machine (SVM), both of which provide a hardware-based root of measurement that has similar protection characteristics to that of the TPM. While not quite as ubiquitous as the TPM itself, these technologies currently enjoy relatively wide deployment today.

Knowing what to collect, especially when one cannot necessarily know a priori the indicators that a new form of attack might have, will always be a challenge. The best we can do is to collect on the basis of historical and probable indicators of attack. The measurements tools that provide data to the TPM are no different and, as such, must be viewed as evidence, but never proof, of system integrity. This is further complicated on TPMs as the limited number of PCRs forces large amounts of evidence to be combined together into a single PCR value. Moreover, because the PCRs lack any details as to what system state contributed to their values, measurements must be limited to structures that are expected to be largely static, lest the natural variation of a structure lead to a combinatorial explosion of "good" PCR values.

Finally, there is the challenge of knowing what PCR values constitute acceptable measurements. If all machines are identical, both in their configuration and their maintenance, then this can be accomplished simply by reading measurements from a machine that is known to be uncompromised. However, subtle differences in how a machine is managed (e.g., the order in which patches are applied) can produce variations that are reflected in PCR values. To further complicate things, different computer vendors have made different decisions as to how to combine measurements in PCRs, meaning that an IBM laptop will have different PCR values from a Dell laptop, even if the software is identical. All of these variations must be tracked and managed in order to make meaningful judgments as to what individual PCR values tell one about a machine's integrity. Such management is certainly not beyond current capabilities, but it represents an additional burden that enterprise administrators must absorb.

Recency of Measurements

Most PCRs in the TPM can accept measurement updates at any time. However, currently, most measurements are taken and stored into PCRs during machine boot. Unfortunately, given that there may be a substantial gap between the time a machine is booted and the time an assessment is requested, the attacker has a window to compromise a system, corrupting either the underlying binaries or code in runtime memory, such that "good" PCR results (from the time of boot) will be delivered for software that has been corrupted prior to its performance of an assessment. As such, the current meaning of a good set of PCRs delivered with an OVAL assessment can only be that the OVAL interpreter and its dependencies *were* in a good state at some point in the past, rather than providing any assurance that the OVAL results were produced by an interpreter that *is currently* in a good state.

This should not be taken to mean that PCR measurements of the OVAL interpreter and its dependencies are currently worthless – it would be hoped that good PCR values of this software would reflect a security posture that was resistant to subsequent compromise and thus provided some degree of assurance. However, the ultimate goal of attesting to the integrity of the assessment infrastructure *at the time of assessment* has not yet been demonstrated. Research is progressing, both in terms of usefully updating TPM PCRs with timely measurements as well as collecting those measurements from both binary files and runtime memory, but these results have yet to be applied in production environments. As such, at this time, the TPM can only provide historical assurance of OVAL interpreter integrity.

Conclusion

This paper provided a basic introduction to the Trusted Computing Group's TPM technology and outlined the synergies between it and the assessment infrastructure supported by the OVAL Language. It is hoped that this will encourage vendors to support expansions of OVAL to include TPM information as well as consider infrastructure enhancements that could lead to greater security of the OVAL process.