6.857 Computer and Network Security	October 17, 2002

Lecture Notes 12 : TCPA and Palladium

Lecturer: Pato/LaMacchia Scribe: Barrows/DeNeui/Nigam/Chen/Robson/Saunders/Walsh

Joe Pato of Hewlett-Packard presented the Trusted Computing Platform Alliance (TCPA). Brian LaMacchia of Microsoft presented Palladium. Barrows, DeNeui, and Nigam scribed the notes on TCPA. Chen, Robson, Saunders, and Walsh scribed the notes on Palladium. Slides from both speakers are available on the 6.857 Web site.

TCPA

Outline

- Why Trusted Computing Platforms
- The Trusted Computing Platform Alliance
- TCPA Concepts
- TCPA Feature Set
- Benefits of TCPA

1 Why Trusted Computing Platforms

The overall goals of a trusted computing platform are to increase business and customer confidence with the security of a platform, to reduce business risks associated with insecurely storing data, and additionally to protect end-user private data.

A trusted computing platform should address questions such as: Can I trust a target machine to behave in an expected manner (maybe based on past performance)? Can I have confidence in interacting with the platform? Can I trust you (the user) to be what you say you are?

A Trusted Computing Platform should:

- Recognize that a platform has known properties
- Identify that a system will behave as expected
- Enable a user to have more confidence in the behavior of the platform in front of them

- Reduce business risks by enabling trust in the behavior of critical information systems
- Protect end user private data and information by enabling trust in end systems (unknown if current technology trajectory will lead to this result)

2 The Trusted Computing Platform Alliance (TCPA)

Doomsayers claim the TCPA is the conspiracy to prevent artistry, anonymity, or assembly. Others wonder if TCPA is the conspiracy in prelude to the apocalypse, and wonder if this is the end of free computing. Some skeptics question how the TCPA will know the end has been reached and wonder if we are getting on the slippery slope to 'Big Brother' baked into a computer. Joe Pato said that his lecture will demonstrate that TCPA is none of these.

History

The TCPA is an industry group started in 1998. It was founded by Compaq, HP, IBM, Intel, and Microsoft. Currently the group has 180 members from the hardware, software, communications, and security technology industries. The group is focused on defining and advancing the concept of trusted computing. Competition in the security space and the need for cheap cryptography prompted creation of this group. The companies also needed to bypass crypto export regulations, and as a result wanted to work towards this goal with other players in the field.

The TCPA Charter

- Provide a ubiquitous and widely adopted means to address trustworthiness of computing platforms
- Publish an open specification for public review
- Define a technology specification that can be applied to any type of computing platform

3 TCPA Concepts

Definition: *A platform can be trusted if it behaves in the expected manner for the intended purpose.*

TCPA Technology provides the mechanisms for:

- Platform authentication and attestation — is this platform actually a TCPA platform?
- Platform integrity reporting — has this TCPA platform been modified in any fashion?
- Protected storage — enabling secure stable storage in the presence of adversaries, architecture enables root of trust that allows third parties to rely on this trust

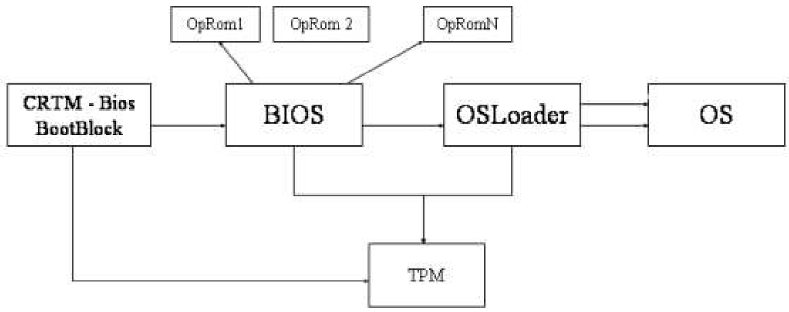

Figure 1: The Authenticated Boot Process (courtesy of Joe Pato, HP Labs)

To achieve this, TCPA relies on the concept of a root of trust. A third party can rely on information provided by a platform's root of trust. The root of trust must be able to report on software that has been executed, and must be able to keep secrets from the rest of the platform. There are two roots of trust and it is necessary to trust these roots of trust for TCPA mechanisms to be relied upon.

- *A root of trust for reporting* — The component that can be trusted to store and report reliable information about the platform
- *A root of trust for measurement* — The component that can be trusted to reliably measure and report to the root of trust for reporting what software executes on platform boot

The Trusted Platform Module (TPM)

The TPM is the Root of Trust for Reporting and is uniquely bound to a single platform. TPM functions and storage are isolated from all other components of the platform. The TPM is tamper resistant and tamper evident. It also contains various cryptographic functions and properties including PRNG, key storage, and some cryptographic functions. However, there is no bulk cryptography built into the TPM.

The Core Root of Trust for Measurement (CRTM)

The CRTM is the first piece of code that executes on a platform at boot time. It must be trusted to property report to the TPM what software executes after it. The CRTM reports a hash of the BIOS to the TPM, the TPM stores this, and then CRTM passes off control to the BIOS. The BIOS hashes various ROMS associated (i.e. the OS Loader) with bootup, TPM securely stores this, the BIOS then loads and executes ROM procedures.

Q: How does CRTM ensure that the boot is authentic?

A: The CRTM builds a chain of hash codes for each portion of the boot. This chain is used to ascertain exactly what software was loaded on boot, the user can then check this with past boot chains and gauge if the boot sequence has been tampered with.

4 The TCPA Feature Set

- Platform Authentication
- Integrity Reporting
- Protected Storage

Platform Authentication

TCPA provides for the TPM to have control over *multiple pseudonymous* attestation identities. TPM attestation identities do not contain any owner or user related information. A platform identity attests to platform properties. No single TPM *identity* is ever used to digitally sign data, this provides privacy protection. A TPM identity certification is required to attest to the fact that they identify a genuine TCPA platform. The TPM identity creation protocol allows for the choice of different Certification Authorities (Privacy-CA) to certify each TPM identity to prevent correlation of the TPMs.

Integrity Reporting

To trust that the TPM is a genuine TPM on a genuine trusted platform, the measurements reported to the TPM during (and after) the boot process cannot be removed or deleted until reboot. Adding each step in the boot process to the TPM hash vector ensures that no hiding code can execute on a platform. The TPM will use an attestation identity to sign the integrity report. The recipient of integrity information can evaluate trustworthiness of the information based on the certificated of this attestation identity.

Protected Storage

The TCPA allows for protected storage, but no generic encryption device is required. Cryptographic keys can be created that are protected by the TPM. Data can be encrypted using the TPM and can only be decrypted using this same TPM. Additionally, the root TPM key can be used to create a hierarchy of sealed keys, of which only the root key lives in the TPM while others live (encrypted) on the hard drive. This allows the user to build new keys from the original TPM key and ensures that the TPM public key is not released. Keys in this hierarchy-space can be migrateable, or not, depending on how they are created by the software/OS or by the manufacturer.

Privacy-Positive design

The ultimate TPM functionality control goes to the owner (i.e. platform administrator). TPM activation is controlled by the owner, while TPM deactivation is available to the individual users. Additionally, to ensure privacy no single TPM *identity* is ever used to digitally sign data and multiple pseudonymous IDs are allowed, which limits correlation. Remote control of the TPM is enabled by challenge response protocols for authorization mechanisms. Unfortunately, since the CA knows all the keys that have been generated, the CA can correlate identities to platforms.

Conformance

The parties involved have various responsibilities. The TCPA's role is that the TPM protection profile is to be completed and will include CRTM and connection to platform. The manufacturers' role is to create a security target, and produce a product design evaluation.

5 Benefits of TCPA

In the short and middle term, TCPA allows for more securely encrypted data and provides for the measurement of integrity metrics of the software environment on the TCPA platform. In the long term, we can learn what software is running on a machine and have confidence in the information about the software environment and identity of a remote party, enabling higher levels of trust when interacting with this party.

Palladium

Brian LaMacchia of Microsoft talked about Palladium during the second half of the lecture. Brian joined the Palladium team this May, having worked the past 3 years on the .NET Framework Security. Before that, he worked at AT&T labs, performing research on policy regarding technology's impact on society.

6 Motivation

What is Palladium?

Palladium (Pd) is a set of new security-oriented capabilities in Windows. Palladium is enabled by new hardware. The goal is to "protect software from software." Protecting against hardware attacks (dual ported memory, physically tampering with the PC, etc.) is not Palladium's goal. Palladium's goal is to protect against malicious software.

Palladium is built on top of TCPA hardware and share some characteristics with TCPA, but is NOT Microsoft's implementation of TCPA.

Peter Biddle of Microsoft Research conceived of and started working on Palladium in 1997.

New Security Features

Palladium offers four categories of new security features.

1. Sealed Storage: seal off storage so that only some programs can get at it
2. Attestation: software and hardware makes a signed statement about some part of the process stack
3. Curtained memory: the ability to segment the physical memory of the machine into standard and trusted modes.
4. Secure Input/Output: user input (i.e. mouse, keyboard)/output (i.e. monitor) are encrypted and thus cannot be sniffed or spoofed

Sealed storage and attestation are characteristics shared with TCPA. Curtained memory and secure I/O are not.

Trusted open systems

Currently, Windows operating systems are designed first for features and performance, then for plug and play and ease of use, and only last for security.

Contrast this to smartcard OS or other closed system; security is higher in closed systems.

Nightmare Scenarios

> "The hackers are standing on a pile of bricks spray-painting glass windows. They haven't yet discovered the bricks."

Although there have been damaging viruses in the past, viruses can do a lot more damage than they have been demonstrated to do. Imagine a virus/Trojan that launches something worse than a denial of service attack:

- Trades a random stock
- Posts tax-records to newsgroup
- Orders random book from Amazon
- Grabs user/password for the host/websites and posts them to newsgroup
- Posts personal documents to a newsgroup

7 Architecture

Palladium at 50,000 ft

	Standard (left-side)	Trusted (right side)
User	Applications	Agents
Kernel	OS	Nexus

Concern: Because one of the priorities of the current Windows OS is to be able to plug and play, there is the concern that the kernel can be corrupted by a plug-in (e.g., trojaned driver). So, the question is: How do you preserve the flexibility and extensibility of pluggable kernel modules while providing security?

Solution: Subdivide the execution environment by adding a new mode flag to the CPU. The CPU is either in standard or trusted mode. Pages of physical memory can be marked as "trusted". Trusted pages can only be accessed when the CPU is in trusted mode.

Thus, the execution environment is divided up into the standard environment (left side) and the trusted environment (right side). The right side has to run without disrupting the left side, since we don't want to break anything that's currently running.

The trusted parallels to the OS and applications are the nexus and the agents. The nexus (a.k.a. trusted operating root (TOR) and nub) is a security kernel. Agents are user programs running in the secured environment. Since agents need to let the user enter secrets, and display secrets to the user, we need trusted I/O.

Input is secured by a trusted USB hub, for the keyboard and mouse, that carries on a protected conversation with the nexus. Output is secured by a trusted GPU that carries on a crypto-protected conversation with the nexus. This gives us "fingertip to eyeball security."

Hardware Security Support

- Security Support Component (SSC)
  - The security support component (SSC) is basically a smartcard soldered onto the motherboard. It is like TCPA's TPM.
  - To make Palladium easily adoptable, the SSC costs $1 a unit so that it can be installed in all machines.
  - It must contain at least an AES key and an RSA key pair. In reality, it may contain two to three AES keys and two to three RSA key pairs. These AES keys & RSA private keys never leave the chip.
  - It also contains registers: the PCR (platform configuration register) that contains the digest of the running Nexus. The Nexus *does not have to be loaded at boot.* It can be loaded later. All we care about Palladium is hashing the nexus. This can happen any time after the machine has booted. So when you boot the nexus you hash the nexus and put it in a hard register.
  - The SSC must be close to the chipset (e.g. not a real smartcard) because it must be involved in nexus initialization.
- Other security goodness
  - RNG, counters, other key storage, crypto operations.
  - There is no secure real-time clock. Best thing to do is to have a monotonically increasing counter, which can be used to detect rollback.

What Palladium Provides

Palladium provides:

- Separate protected execution environments for apps (computing agents) that need higher security, through hardware-based memory isolation (i.e. the left hand side has no access to the right hand side).
- Agents can be standalone, and provide services to other apps. The nexus is the gatekeeper that lets messages pass back to the left hand side.

Palladium Core Features

All Palladium capabilities build off of four key features:

- Strong process isolation
- Root key for persistent secret protection
- Secure path to and from user
- Attestation

The first three are needed to protect against malicious code (viruses, Trojans, etc.). Attestation breaks new ground. With attestation, facts about "things" (software, users, machines, services) can be proved to (and believed by) remote entities.

Code Identity

OS Identity:

- Keep hardware simple
- The SSC/chipset measures the digest of the nexus on "secure initiation"

Application Identity:

- Could use a digest, but we actually use a "manifest" which simplifies management (A "manifest" is a signed statement of hashes)

Sealed Storage

Sealed storage allows software to keep long-lived secrets safe from other software running on the host.

- Sealed storage uses an encryption technology, but it's more than simple encryption
  - The security chip has very little storage (just room for keys)
  - Let's say I have a banking application that I want to protect. How can it encrypt its stuff and then hide its key?
- An OS/nexus can keep secrets from other OS's
  - We involve the secret key in the chip (that no one ever gets to see since it's baked in at build) in the encryption. Forgetting the banking application for now, let's say I'm a Nexus and I want to encrypt something, so I ask the SSC. It uses its AES key and it brands it with my hash value (the hash of myself, the nexus) and now it will only give the data back to me since I'm the only one that hashes to the proper value. The SSC holds the key and so it only gives content to the code which sealed it.
- If an OS can keep a secret, it can provide a similar service to applications

How do we do this?

- Use the PCR value to "brand" encrypted secrets with the identity of the code that "owns" them
- Owners of secrets can also designate alternate recipients (necessary for update and Palladium migration)

Attestation

Attestation lets a remote client know what software is running. This authentication technology, is more than a "simple signing;" Attestation enables authentication of a software configuration (nexus, application, process).

- Code authenticates itself using the SSC quote function:
  - Quot(S) $\rightarrow$ Sign[$S|PCR$]
- This provides a protocol building block:
  - For example, in a Server/peer protocol
    - The server checks the signature, checks certificates on signing key, checks nexus digest is as expected
    - Then, it knows the client is a "MS Nexus on Acme Trusted Platform"
- Implementation: RSA using SSC key pair
- No anonymity at this low level. Can only use the hardware key ONCE per power cycle. To preserve anonymity, use the hardware key to create pseudo-identities to provide indirection/anonymity while still providing platform attestation. Being able to create pseudo-identities requires the presence of trusted third parties which do not exist at present. Microsoft is looking into encouraging such markets.

8 Policy Issues

Some of the technical issues we have to solve are policy issues.

You want to have a piece of technology that you can hold up any time anyone starts to complain about social problems. For instance, you're worried about a child viewing pornography on net? Don't make pornography illegal; make a chip that protects kids.

- So how do we build an attestable TCB (trusted computing base): open, auditable, and comprehensible?
- Since the Palladium RSA key is unique what steps should we take to defend against traffic analysis and behavior?

Nexus Policy

- Everything that runs today will run on Palladium systems
- The platform will run any nexus
- The user will be in charge of what nexuses he chooses to run
- The MS nexus will run any application
- The user will be in charge of the applications that he chooses to run
- The MS nexus will interoperate with any network service provider
- The MS nexus source code will be made available for review

"The security mode will be off by default. You can't have it on by default. This is a hard lesson for Microsoft. Users always click 'yes, sure, go ahead format my hard drive — I don't have time to read this! I have work to do.' " Somehow you have to tell Microsoft what to run on Palladium.

The talk was interrupted by Q&A. See the Palladium slides (24-29) posted on the course handout page for information about:

- Privacy of Machine Identity
- Pseudo-Identities
- Registering a Pseudo-Identity
- Summary

9 Questions & Answers

Q: Have you thought about user interface for designating right hand side application windows vs left hand side applications?

A: You could store a secret (user's favorite fruit plus number) in Palladium, and display that secret every time Palladium window comes up. Alternately, you could use a hardware indicator.

Q: Can I run pirated Microsoft Word on this? - Randall Davis

A: Well yes, now, but it is possible to use Palladium to interfere in the future maybe, i.e. if part of Word runs in the right hand (it never makes sense to run it all in the right hand because it's too big and takes too much work to migrate over). So it is possible to write applications that do that. We [Palladium group] are not involved with the app guys, though. MS is a tribe of 200 groups that dislike each other. Word sits on the left hand side now. Anything I send out to word in the left hand is insecure. - Brian LaMacchia

Q: Would it be possible to design the crypto so that the user can always force a decryption – is it technically possible (if not smart)? - Hal Abelson

A: Yes, because at the end of the day decryption always happens and you have a choice how to store it. Think of it as a big red override button. There's nothing that prevents that at the hardware level. - Brian LaMacchia

Side note: There are certainly customers of ours that don't care about certified stuff – they want their own keys in there. Certain 3 lettered agencies want an override. Who ends up owning the machine? - Brian LaMacchia

Q: Is this effort truly aimed as benefit to users, or is it more for copyright protection, DRM, and Hollywood?

A: There are two main concerns driving Palladium:

1. Piracy
2. That the PC might be bypassed as medium for digital document distribution

Furthermore, this is also aimed at enterprises. Enterprises are concerned about keeping documents private.

Q: In terms of needing to build a market for certificate authorities of pseudo-identities, who are your competitors?

A: Retailers like Blockbuster who may want to do (and care only about) their own authentication.

Q: What will be the first things to move to the trusted side?

A:
- anything in OS that works with secrets
- core crypto
- lightweight viewers
- further down the road: full LHS apps (browser, OE, Word, etc.) move to RHS

Q: What happens if your SSC dies?

A: You'd have to have had a migration scenario upfront (need to have backed up software key). Otherwise, you lose your data. But, the same is true with the non-Palladium technology.

Q: I, as a user, don't know what's running on the right hand side. How do I protect my left hand side stuff from the right hand side stuff? - Ron Rivest

A: Policy for rights of agents is more impoverished in terms of resources. Furthermore, the nexus/OS are the gatekeepers. Agents will first get very restricted rights and build up from there.

Q: With firewalls, people were duped into thinking the firewall solved all their security problems. "Who needs end-to-end security when you have a firewall?" was the attitude. How will you prevent the same attitude from developing if there is wide-spread use of Palladium? People will still have to worry about buffer overflows, etc. -Kevin Fu

A: Train the marketeers. That's why there's a difference between the white paper and the technical paper. Unfortunately, the security business involves a lot of hype.

Q: Regarding small devices. What's your vision for the future (will they also run Palladium?) - Ron Rivest

A: Currently, small devices are closed and thus do not need Palladium. If devices become more open, that may be a possibility. But, devices can also become more closed.