

Introduction

"Palladium" is the code name for an evolutionary set of features for the Microsoft® Windows® operating system. When combined with a new breed of hardware and applications, these features will give individuals and groups of users greater data security, personal privacy, and system integrity. In addition, "Palladium" will offer enterprise customers significant new benefits for network security and content protection. This topic reveals the following:

- Examines how "Palladium" satisfies the growing demands of living and working in an interconnected, digital world
- Catalogs some of the planned benefits offered by "Palladium"
- Summarizes the software and hardware components of "Palladium"

The Challenge: Meeting the Emerging Requirements of an Interconnected World

Today's personal computing environment has advanced in terms of security and privacy, while maintaining a significant amount of backward compatibility. However, the evolution of a shared, open network (the Internet) has created new problems and requirements for trustworthy computing. As the personal computer grows more central to our lives at home, work and school, consumers and business customers alike are increasingly aware of privacy and security issues.

Now, the pressure is on for industry leaders to take the following actions:

- Build solutions that will meet the pressing need for reliability and integrity
- Make improvements to the personal computer such that it can more fully reach its potential and enable a wider range of opportunities
- Give customers and content providers a new level of confidence in the computer experience
- Continue to support backward compatibility with existing software and user knowledge that exists with Windows systems today

Together, industry leaders must address these critical issues to meet the mounting demand for trusted computing while preserving the open and rich character of current computer functionality.

The Solution: "Palladium"

"Palladium" is the code name for an evolutionary set of features for the Microsoft Windows operating system. When combined with a new breed of hardware and applications, "Palladium" gives individuals and groups of users greater data security, personal privacy and system integrity. Designed to work side-by-side with the existing functionality of Windows, this significant evolution of the personal computer platform will introduce a level of security that meets the rising customer requirements for data protection, integrity and distributed collaboration.

Users implicitly trust their computers with more of their valuable data every day. They also trust their computers to perform more and more important financial, legal and other transactions. "Palladium" provides a solid basis for this trust: a foundation on which privacy- and security-sensitive software can be built.

There are many reasons why "Palladium" will be of advantage to users. Among these are enhanced, practical user control; the emergence of new server/service models; and potentially new peer-to-peer or fully peer-distributed service models. The fundamental benefits of "Palladium" fall into three chief categories: greater system integrity, superior personal privacy and enhanced data security. These categories are illustrated in Figure 1.

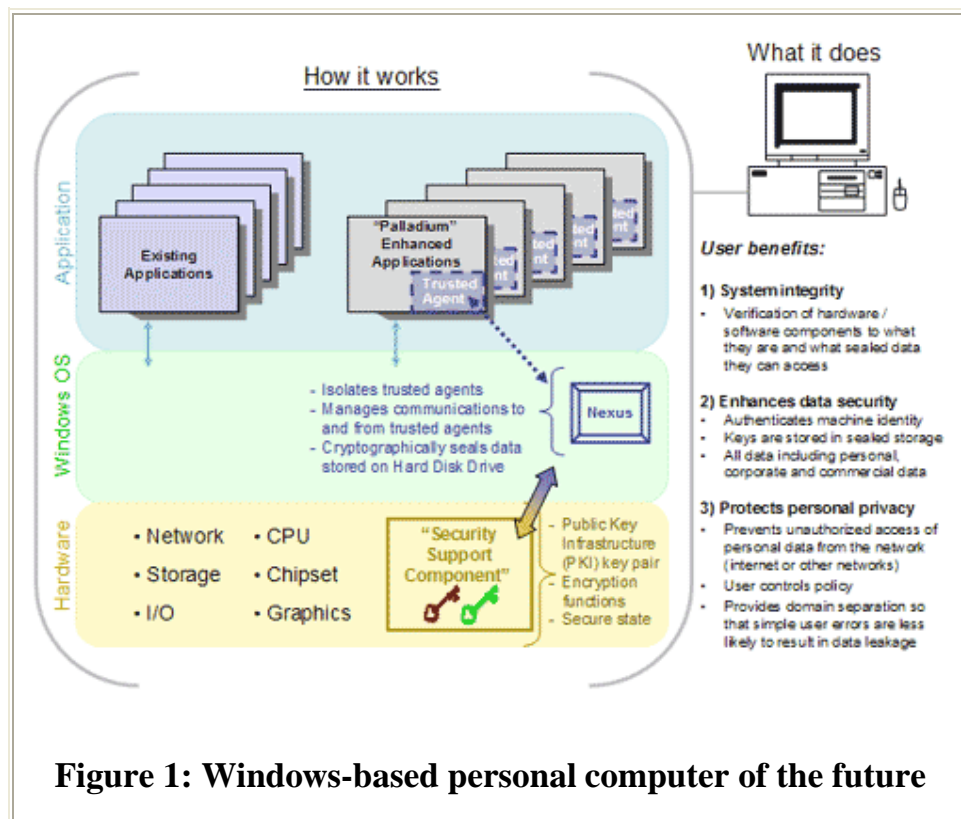


Figure 1: Windows-based personal computer of the future

Core Principles of the "Palladium" Initiative

Development of "Palladium" is guided by important business and technical imperatives and assumptions. Among these are the following:

A "Palladium"-enhanced computer must continue to run any existing applications and device drivers.

"Palladium" is not a separate operating system. It is based on architectural enhancements to the Windows kernel and to computer hardware, including the CPU, peripherals and chipsets, to create a new trusted execution subsystem (see Figure 1).

"Palladium" will not eliminate any features of Windows that users have come to rely on; everything that runs today will continue to run with "Palladium."

In addition, "Palladium" does not change what can be programmed or run on the computing platform; it simply changes what can be believed about programs, and the durability of those beliefs. Moreover, "Palladium" will operate with any program the user specifies while maintaining security.

"Palladium"-based systems must provide the means to protect user privacy better than any operating system does today.

"Palladium" prevents identity theft and unauthorized access to personal data on the user's device while on the Internet and on other networks. Transactions and processes are verifiable and reliable (through the attestable hardware and software architecture described below), and they cannot be imitated.

With "Palladium," a system's secrets are locked in the computer and are only revealed on terms that the user has specified. In addition, the trusted user interface prevents snooping and impersonation. The user controls what is revealed and can separate categories of data on a single computer into distinct realms.

Finally, the "Palladium" architecture will enable a new class of identity service providers that can potentially offer users choices for how their identities are represented in online transactions. These service providers can also ensure that the user is in control of policies for how personal information is revealed to others. In addition, "Palladium" will allow users to employ identity service providers of their own choosing.

"Palladium" will not require digital rights management technology, and DRM will not require "Palladium."

Digital rights management (DRM) is an important, emerging technology that many believe will be central to the digital economy of the future. As a means of defining rules and setting policies that enhance the integrity and trust of digital content consumption, DRM is vital for a wide range of content-protection uses. Some examples of DRM are the protection of valuable intellectual property, trusted e-mail and persistent protection of corporate documents.

While DRM and "Palladium" are both supportive of Trustworthy Computing, neither is absolutely required for the other to work. DRM can be deployed on non-"Palladium" machines, and "Palladium" can provide users with benefits independent of DRM. They are separate technologies. That said, the current software-based DRM technologies can be rendered stronger when deployed on "Palladium"-based computers.

User information is not a requirement for "Palladium" to work.

"Palladium" authenticates software and hardware, not users. "Palladium" is about platform integrity, and enables users - whether in a corporate or home setting - to take advantage of system trustworthiness to establish multiple, separate identities, each to suit specific needs.

For example, an employee logs onto the corporate network from home. A trusted gateway server at the corporate network mediates the remote access connection, allowing only trusted applications to access the network. This ensures that the network is protected against infection from attacks by viruses that the home user might have received through personal e-mail. Once connected, the employee can use Remote Desktop to access the computer at the office or save a file back to the corporate server by using locally active Trusted Agents and sealed storage (see below) on the client.

With this technology, the corporate network is protected, while the individual can also be confident that the company is not using the remote connection as an opportunity to snoop into the contents of the user's home computer.

"Palladium" will enable closed spheres of trust.

A closed sphere of trust binds data or a service to both a set of users (logon) and to a set of acceptable applications. As shown in Figure 2, the nexus (formerly referred to as the Trusted Operating Root, or TOR) does not simply open the vault; the nexus will open only a particular vault, and only for a small list of applications.

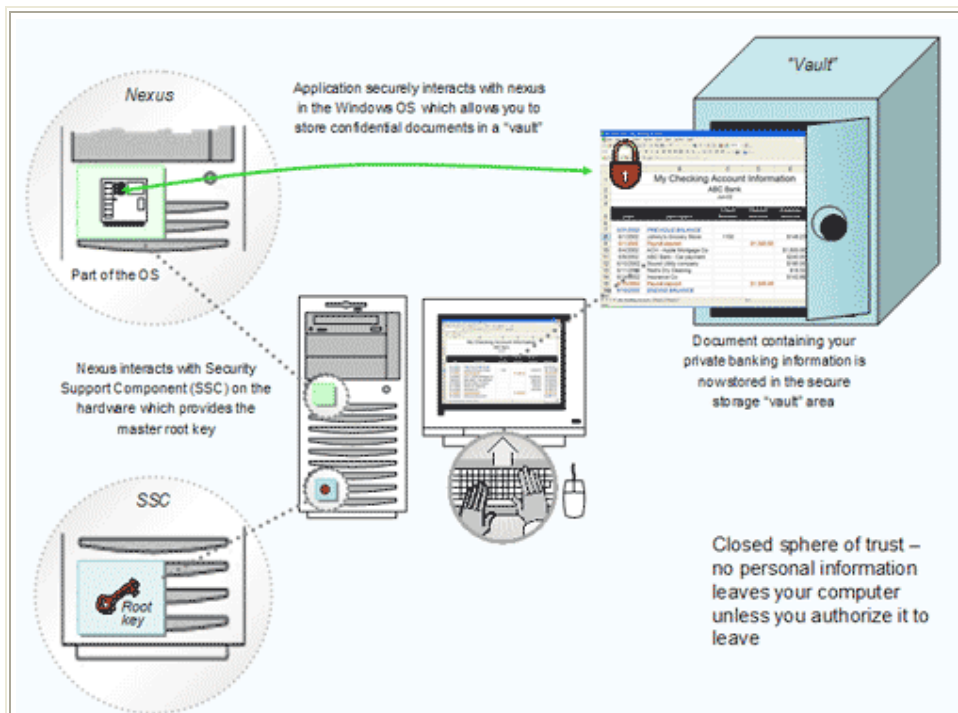


Figure 2: Closed Sphere of Trust

"Palladium" is an opt-in system.

"Palladium" is entirely an opt-in solution; systems will ship with the "Palladium" hardware and software features turned off. The user of the system can choose to simply stay with this default setting, leaving all "Palladium"-related capabilities (hardware and software) disabled.

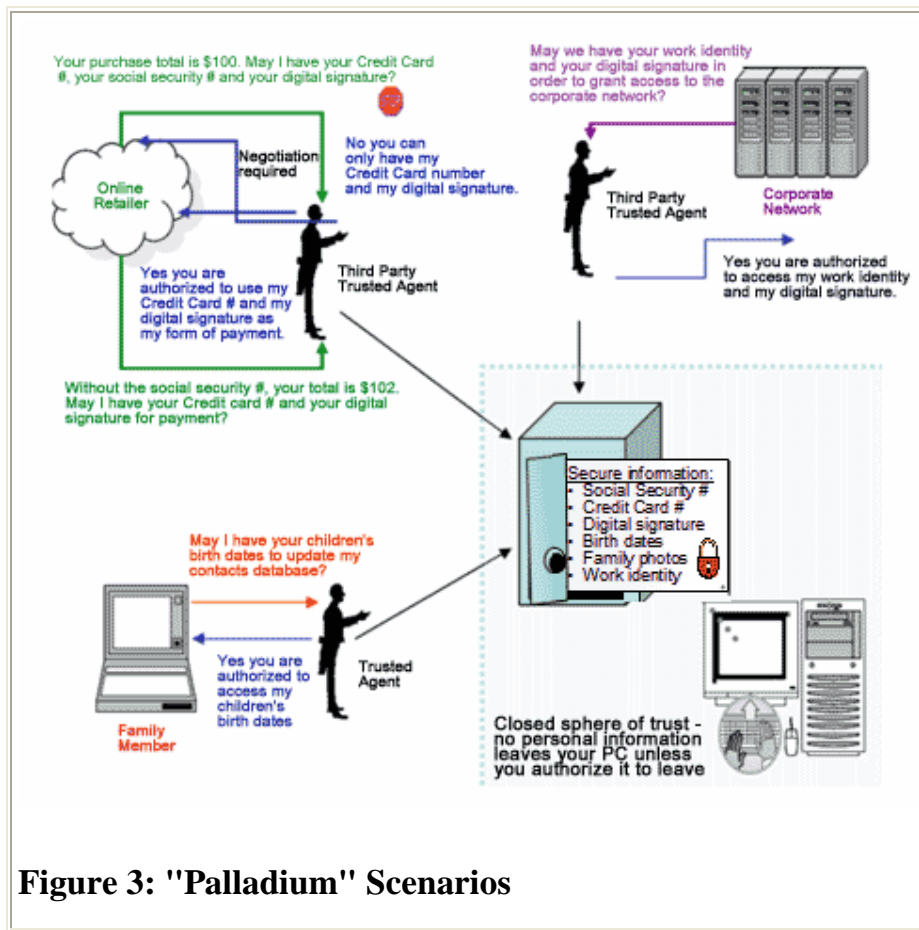


Figure 3: "Palladium" Scenarios

Palladium must be highly resistant to software attacks (such as Trojan horse viruses), and must provide users with the integrity of a protected, end-to-end system across networks.

Palladium provides a trusted processing environment. Trusted code runs in memory that is physically isolated, protected, and inaccessible to the rest of the system, making it inherently impervious to viruses, spy-ware, or other software attacks. With respect to viruses, the contribution from Palladium is fairly straightforward. Since Palladium does not interfere with the operation of any program running in the regular Windows environment, everything, including the native OS and viruses, runs there as it does today. So antivirus monitoring and detection

software in Windows will still be needed. However, Palladium does provide antivirus software with a secure execution environment that cannot be corrupted by infected code, so an antivirus program built on top of a Palladium application could guarantee that it hasn't been corrupted. This grounding of the antivirus software allows it to bootstrap itself into a guaranteed execution state, something it can't do today.

One of the key Palladium building blocks is "authenticated operation". If a banking application is to be trusted to perform an action, it is important that the banking application has not been subverted. It is also important that banking data can only be accessed by applications that have been identified as trusted to read that data. "Palladium" systems provide this capability through a mechanism called sealed storage.

Another capability provided by authenticated operation is attestation. "Palladium" will allow a bank to accept only transactions initiated by the user and that are not viruses or other unknown machines on the Internet. Because "Palladium" software and hardware is cryptographically verifiable to the user and to other computers, programs and services, the system can verify that other computers and processes are trustworthy before engaging them or sharing information. Users therefore can be confident that their intentions are properly represented and carried out, as illustrated in Figure 3. Moreover, the source code for the operating system's critical nexus will be published and validated by third parties.

Finally, interaction with the computer itself is trusted. "Palladium"-specific hardware provides a protected pathway from

keyboard to monitor, and keystrokes cannot be snooped or spoofed, even by malicious device drivers.

"Palladium" data security features will make a Windows-based device a trustworthy environment for any data.

The "Palladium" system is architected with security and integrity as its primary design goals. Trusted code cannot be observed or modified when running in the trusted execution space. Files are encrypted with machine-specific secrets, making them useless if stolen or surreptitiously copied. In addition, machine-specific system secrets are physically and cryptographically locked (the machine's private key is embedded in hardware and never exposed), and the trusted hardware architecture prevents snooping, spoofing and data interception. Core system secrets are stored in hardware, where no software attack can reveal them. Even if exposed by a sophisticated hardware attack, the core system secrets are only applicable to data on the compromised system and cannot be used to develop widely deployable hacks. Finally, a compromised system can likely be spotted by IT managers, service providers and other systems, and then excluded.

A "Palladium" system will be open at all levels.

"Palladium" hardware will run any nexus. Some platforms may allow a user to restrict the nexuses that are allowed to run, but the user will still be in full control of this policy. The "Palladium" TOR will also run trusted agents from any publisher. Again, the user may choose to restrict the trusted agents that run on the system, but the user will remain

in full control of this policy. The "Palladium" nexus will work with any network service provider of the user's choosing.

Aspects of "Palladium"

"Palladium" comprises two key components: hardware and software.

Hardware Components

Engineered for ensuring the protected execution of applications and processes, the protected operating environment provides the following basic mechanisms:

- **Trusted space.** The execution space is protected from external software attacks such as a virus. Trusted space is set up and maintained by the nexus and has access to various services provided by "Palladium," such as sealed storage.
- **Sealed storage.** Sealed storage is an authenticated mechanism that allows a program to store secrets that cannot be retrieved by nontrusted programs such as a virus or Trojan horse. Information in sealed storage cannot be read by other nontrusted programs. (Sealed storage cannot be read by unauthorized secure programs, for that matter, and cannot be read even if another operating system is booted or the disk is carried to another machine.) These stored secrets can be tied to the machine, the nexus or the

application. Microsoft will also provide mechanisms for the safe and controlled backup and migration of secrets to other machines.

- **Attestation.** Attestation is a mechanism that allows the user to reveal selected characteristics of the operating environment to external requestors. For example, attestation can be used to verify that the computer is running a valid version of "Palladium."

These basic mechanisms provide a platform for building distributed trusted software.

Software Components

The platform implements these trusted primitives in an open, programmable way to third parties. The platform consists of the following elements:

- **Nexus** (a technology formerly referred to as the "Trusted Operating Root (TOR)". The component in Microsoft Windows that manages trust functionality for "Palladium" user-mode processes (agents). The nexus executes in kernel mode in the trusted space. It provides basic services to trusted agents, such as the establishment of the process mechanisms for communicating with trusted agents and other applications, and special trust services such as attestation of requests and the sealing and unsealing of secrets.
- **Trusted agents.** A trusted agent is a program, a part of a program, or a service that runs in user mode in the trusted space. A trusted agent calls the nexus for security-related services and critical general services such as memory management. A trusted agent is

able to store secrets using sealed storage and authenticates itself using the attestation services of the nexus. One of the main principles of trusted agents is that they can be trusted or not trusted by multiple entities, such as the user, an IT department, a merchant or a vendor. Each trusted agent or entity controls its own sphere of trust, and they need not trust or rely on each other.

Together, the nexus and trusted agents provide the following features:

- Trusted data storage, encryption services for applications to ensure data integrity and protection
- Authenticated boot, facilities to enable hardware and software to authenticate itself

From the perspective of privacy (and anti-virus protection), one of the key benefits of "Palladium" is the ability for users to effectively delegate certification of code. Anyone can certify "Palladium" hardware or software, and it is expected that many companies and organizations will offer this service. Allowing multiple parties to independently evaluate and certify "Palladium"-capable systems means that users will be able to obtain verification of the system's operation from organizations that they trust. In addition, this will form the basis for a strong business incentive to preserve and enhance privacy and security. Moreover, "Palladium" allows any number of trusted internal or external entities to interact with a trusted component or trusted platform.

- The initial version of Palladium will require changes to five parts of the PC's hardware. Changes will be required to the CPU, the chipset (on the motherboard), the input devices (e.g. keyboard),

and the video output devices (graphics processor). In addition, a new component must be added: a tamper-resistant secure cryptographic co-processor, which Microsoft calls SCP or SPP.

- Although the SCP is tamper-resistant, it is likely that a skilled attacker with physical access to the inside of a Palladium PC can still compromise it or subvert its policies in some way.
- So it is possible that an attacker with physical access can still compromise the system, even though the SCP is meant to be tamper-resistant, partly because other components (like RAM) are less robust against modification. Palladium primarily defends effectively against two classes of attacks: (1) remote network-mounted attacks (buffer overflows and other programming flaws, malicious mobile code, etc.), because even if some malicious code is installed in one part of the system, it still can't effectively subvert the policy of another part of the system, and (2) local software-based attacks, including things like using a debugger to try to read a program's internal state while it's executing or to try to subvert its policy. Thus, Palladium can probably guarantee that you can't write or download any software (and nobody else can write or upload to you any software) which would compromise the policy of software running locally which is making use of Palladium trust features.
- Palladium's changes to the CPU allow it to be placed into a new mode where certain areas of memory are restricted via a technique called "code curtaining" to an ultra-privileged piece of code called the "nub" or "TOR". ("Nub" is the Palladium team's term for this code, and "TOR", for "Trusted Operating Root", is the official

public term.) The nub is a kind of trusted memory manager, which runs with more privilege than an operating system kernel. The nub also manages access to the SCP.

- The SCP is an 8-bit tamper-resistant cryptographic smart-card which contains unique keys, including public keypairs (2048-bit RSA), and symmetric keys for AES in CBC mode. These keys are unique per machine and the SCP does not reveal them to anything outside the SCP's security perimeter. It also contains a variety of other cryptographic functionality, including SHA-1, RSA, AES, and other cipher implementations, a small amount of memory, and a monotone counter. The SCP can do a number of cryptographic protocols. It also contains a thing called a PCR. (I think that stands for "platform configuration register".)
- When you want to start a Palladium PC in trusted mode (note that it doesn't *have* to start in trusted mode, and, from what Microsoft said, it sounds like you could even imagine booting the same OS in either trusted or untrusted mode, based on a user's choice at boot time), the system hardware performs what's called an "authenticated boot", in which the system is placed in a known state and a nub is loaded. A hash (I think it's SHA-1) is taken of the nub which was just loaded, and the 160-bit hash is stored unalterably in the PCR, and remains there for as long as the system continues to operate in trusted mode. Then the operating system kernel can boot, but the key to the trust in the system is the authentication of the nub. As long as the system is up, the SCP knows exactly which nub is currently running; because of the way the CPU works, it is not possible for any other software to modify

the nub or its memory or subvert the nub's policies. The nub is in some sense in charge of the system at a low level, but it doesn't usually do things which other software would notice unless it's asked to.

- The nub interfaces with other software on the system by means of programs (outside the nub) called trusted agents (or TAs). The TAs can implement sophisticated policies and authentication methods, where the nub (and SCP) just implement fairly simple primitives. A TA can also communicate with user-space programs (at least, that will be a feature of Microsoft's nub; other people can write their own nubs which can support different kinds of TAs or even do without TAs entirely). The TAs are protected by hardware from one another and from the rest of the system.
- Even PCI DMA can't read or write memory which has been reserved to a nub's or TA's use (including the nub's or TA's code). This memory is completely inaccessible and can only be accessed indirectly through API calls. The chipset on the motherboard is modified to enforce this sort of restriction.
- The SCP provides a feature called "sealed storage" by means of two API calls (called SEAL and UNSEAL). If a TA running on a system in trusted mode wants to use sealed storage, it can call into the APIs implemented in the nub.
- Sealed storage is implemented by means of encryption (sealing) or decryption (unsealing) with a symmetric cipher. When the SCP is given data to seal, it's given two arguments: the data itself and a 160-bit "nub identifier"

- Sealing is performed by prepending the nub identifier to the data to be sealed, and then encrypting the result with a private symmetric key -- the "platform-specific key", which varies from machine to machine and is secret. That key is kept within the SCP and is a unique identifier for the machine which performed the sealing operation.
- The SCP actually also prepends a random nonce to the data to be sealed before encryption (and discards the nonce upon decryption). This is a clever privacy feature which prevents someone from creating an application which "cookies you" by recording the output of sealing an empty string (and then using the result as a persistent unique identifier for your machine). A program which tried to "cookie you" this way would find that, because of the random nonce, the result of sealing a given string is constantly completely different, and no useful information about the identity of the machine is revealed by the sealing operation.
- After encryption, the SCP returns the encrypted result as the return value of the SEAL operation.
- When an SCP is given encrypted data to UNSEAL, it internally attempts to decrypt the encrypted data using its platform-specific key. This means that, if the encrypted data was originally sealed on a different machine, the UNSEAL operation will fail outright immediately. (You can't take a sealed file and transfer it to another machine and unseal it there; because the platform-specific key is used for encryption and decryption, and can't be extracted from the SCP, you can only UNSEAL data on the same machine on which it was originally SEALED.)

- If the decryption is successful, the SCP performs a second check: it examines the nub identifier which resides within the decrypted data. The nub identifier was specified at the time the data was originally SEALED, and indicates which nub is allowed to receive the decrypted data. If the nub identifier for the decrypted data is identical to the nub identifier which is currently stored in the PCR (which is the SHA-1 hash of the currently-running nub on the machine at the moment UNSEAL was called), the UNSEAL is successful and the decrypted data is returned to the calling nub. However, if the nub identifier does not match the contents of the PCR, the SCP concludes that the nub which is currently running is not entitled to receive this data, and discards it.
- Thus, sealing is *specific to a physical machine* and also *specific to a nub*. Data sealed on one machine for a particular nub cannot be decrypted on a different machine or under a different nub. An application which trusts a particular nub (and is running under that nub) can seal important secret data and then store the resulting sealed data safely on an untrusted hard drive, or even send it over a network.
- If you reboot the machine under a debugger, there is no technical problem, and you can debug the software which created the encrypted file. However, since you aren't running the proper (non-debugger-friendly) nub, the debugger will work, but the UNSEAL call won't. The SCP will receive the UNSEAL call, examine the PCR, and conclude that the currently-running nub is not cleared (so to speak) to receive the sealed data. Your applications can only decrypt sealed data if they are running *under the same machine*

and *under the same software environment* within which they originally sealed that data!

- This is remarkably clever. When you are running under a trusted nub, your applications can use the SCP to decrypt and process data, but you can't run software which subverts a TA's policy (because the nub will not permit the policy to be subverted).
- When you are not running under a trusted nub, you can run software which subverts a TA's policy (because the nub isn't able to prevent it), but your applications will no longer be able to decrypt any sealed data, because the SCP won't be willing to perform the decryption.
- There is a long discussion of how you can make a backup, or upgrade your system, or migrate your software and data to a new system, etc. The default with sealed storage is that any sealed data will be unusable when migrated to a new system. The Microsoft nub provides wrappers around the SCP's sealing features which allow the software which performs the sealing operation to specify a migration policy at the time the sealing operation is originally performed. The migration policy can be (approximately) one of the following, at the software's sole option: (1) Migration is prevented entirely, and the data must die with the current PC where it was created. (2) Migration is permitted upon some kind of authentication by a local user (e.g. a password) which will decrypt or command the decryption of data temporarily in order to permit it to be migrated. (3) Migration is permitted with the assistance and consent of a 3rd party .

- Palladium's modifications to input and output hardware will prevent software from doing certain kinds of monitoring and spoofing, as well as "screen scraping". A program will be able to ask Palladium to display a dialog box which can't be "obscured" or "observed" by other software, and Palladium hardware can enforce these conditions. And there is a way to be sure that input is coming from a physical input device and not spoofed by another program.
- The secure output features also permit, e.g., a DVD player program to prevent other software from making screen captures. The initial version of Palladium does not control audio output in this way, so you can still record all sound output via something like TotalRecorder.
- In principle, nub and kernel are independent, so a non-Microsoft kernel could run on a Microsoft nub, or vice versa. Patent and copyright issues might prevent this from being done in practice, but it is apparently technically possible within the design of Palladium.
- Microsoft's nub, including its source code, will be published for review by anyone who wants to examine it, in order to allow all of Microsoft's claims about its security properties to be verified. There is no part of Palladium's design or code which needs to be kept secret, although each SCP will contain secret cryptographic keys loaded at the time of its manufacture. Microsoft will encourage non-Microsoft people to read and discuss its nub. You will also be able to create your own nub, except that changing the

nub will (as discussed above) prevent previously-sealed data from being decrypted.

- Microsoft suggests that Palladium is flexible enough that many entities could use it to create their own policies, judgments, certification services, etc. Palladium has a more robust technical enforcement mechanism than either of those standards.

Known Elements Of Palladium

The system purports to stop viruses by preventing the running of malicious programs

The system will store personal data within an encrypted folder

The system will depend on hardware that has either a digital signature or a tracking number

The system will filter spam.

The system has a personal information sharing agent called “My Man”.

The system will incorporate Digital Rights Management technologies for media files of all types.

Comparison of TCPA and Palladium

TCPA stands for ‘Trusted Computing Platform Alliance’, an initiative led by Intel. Their stated goal is a ‘new computing platform for the next century that will provide for improved trust in the PC platform’. Palladium is a software that Microsoft says it plans to incorporate in

future versions of Windows; it will build on the TCPA hardware and will add some new features.

The TCPA and Palladium rely on the addition to the hardware of normal PC's. While Palladium calls for more extensive changes, the modifications are remarkably similar. Both call for a new chip to be placed on the motherboard of all future computers. The chip would include new encryption functions as well as a small amount of memory that would act as a digital vault to store important keys to decrypt protected data. The TCPA refers to the chip as the "Trusted Platform Module", a successor to the Intel's processor. Microsoft refers to the hardware components of Palladium as Secure Cryptographic Co-processor or SCP.

Conclusion

Today, IT managers face tremendous challenges due to the inherent openness of end-user machines, and millions of people simply avoid some online transactions out of fear. However, with the usage of "Palladium" systems, trustworthy, secure interactions will become possible. This technology will provide tougher security defenses and more abundant privacy benefits than ever before. With "Palladium," users will have unparalleled power over system integrity, personal privacy and data security.

Independent software vendors (ISVs) that want their applications to take advantage of "Palladium" benefits will need to write code specifically for this new environment. A new generation of "Palladium"-compatible hardware and peripherals will need to be designed and built.

The "Palladium" development process will require industry wide collaboration. It can only work with broad trust and widespread acceptance across the industry, businesses and consumers.

"Palladium" is not a magic bullet. Clearly, its benefits can only be realized if industry leaders work collaboratively to build "Palladium"-compatible applications and systems - and then only if people choose to use them. But the "Palladium" vision endeavors to provide the trustworthiness necessary to enable businesses, governments and individuals to fully embrace the increasing digitization of life.

References

- White paper on “Microsoft Palladium” by Electronic Privacy Information Centre.
- Details about Palladium on the web site Microsoft.com.
- Extensive FAQ on the Palladium system by Prof. Ross Anderson.
- Detailed summary of Palladium by Seth Schoen.
- Interview with John Manferdelli, General Manager of the Windows business unit that is building Palladium.

ACKNOWLEDGMENT

I thank God Almighty for the successful completion of my seminar.

I express my sincere gratitude to Dr. M N Agnisharman Namboothiri, Head of the Department, Information Technology. I am deeply indebted to Staff-in-charge, Miss. Sangeetha Jose and Mr. Biju, for their valuable advice and guidance. I am also grateful to all other members of the faculty of Information Technology department for their cooperation.

Finally, I wish to thank all my dear friends, for their whole-hearted cooperation, support and encouragement.

ABSTRACT

Today's personal computing environment has advanced in terms of privacy and security, while maintaining a significant amount of backward compatibility. However the evolution of a shared, open network has created new problems and requirements for trustworthy computing. As the personal computer grows more central to our lives at home, work and school, consumers and business customers alike are increasingly aware of privacy and security issues.

Palladium is the code name for an evolutionary set of features for the Microsoft Windows Operating system. When combined with a new breed of hardware and applications, these features will give individuals and groups of users greater data security, personal privacy and system integrity.

Palladium provides a solid basis for the user's trust: a foundation on which privacy-and security-sensitive software can be built. There are many reasons why Palladium will be of advantage to users. Among these are enhanced, practical user control; the emergence of new server/service models and potentially new peer-to-peer or fully peer-distributed service models.

CONTENTS

- Introduction
- The challenge: Meeting the Emerging Requirements of an Interconnected world.
- The solution: “Palladium”
- Aspects of Palladium
- Hardware components
- Software components
- Addition of a new secure cryptographic co-processor or SCP
- Comparing Palladium and TCPA aspects
- Known elements of Palladium
- Conclusion