

Building a Whole Security Solution

Granite Key, Inc.

Aug 22, 2005

Contents

- Contents 2
- Summary 3
- The Security Wholution (Whole / Holistic Solution) 4
- Cost of failure and level of security required..... 6
- Ability to Detect failures..... 6
- Threats / Security Objectives 7
- Computing device Environment (PC, PDA, phone, etc) 8
 - Data is encrypted on disk, but not in operating system 9
 - Copy written data is encrypted on disk, but not in media player..... 10
 - How secure is a VPN? 11
- Security Hardware 12
 - Summary: Requirements are essential for excellence and integrity 14
 - Summary: Requirements are essential for excellence and integrity 14
- The Future of security solutions 15
- Industry examples of the application of security 16
 - Legal – digital signature of documents 16
 - Medical security 16
 - Credit cards 17
 - Copyright protection of content 17
 - POS terminals 17
- Summary 18

Summary

Granite Key has helped Fortune 100 customers integrate security technologies into their new and existing PC, Wireless, Network, and Internet solutions. Our customers typically are adding authentication, platform security, encryption, auditing into their applications. Our philosophy is that a good technical solution must start with a deep understanding of business requirements, business processes, and competing solutions, followed by design, implementation, and in many cases process engineering/re-engineering. Thus, GraniteKey frequently gets involved with customers at the earliest stages of a project, in order to help ensure the business success of the solution

A security solution may include technologies that are used internally in your company, and/or may be included in products which you ship to customers. Today, security is on the minds of most technology users. Concerns range from:

- Competitive parity – “I just need security technology XYZ because other people have it”
- The need to meet regulatory requirements
- A well thought out need for security to help alleviate specific threats
- A need to help alleviate a wide range of threats.

Before designing the technical solution, it’s important to first understand what is driving the demand for security, the environment in which your or your customers are operating, and the level of security needed. Security is not unlike other products or projects, in that it’s important to start with requirements. However, because of the highly technical and complex nature of typical security products, it is a very common mistake to allow the technology to drive the requirements rather than the other way around.

Designing security into your solution is like purchasing insurance: What level of protection do you want, and what can you afford to pay? (Or perhaps you are told what you need by a regulatory body or a partner). Basic protection might or might not be enough, depending on your objectives, risk profile, resources, etc.

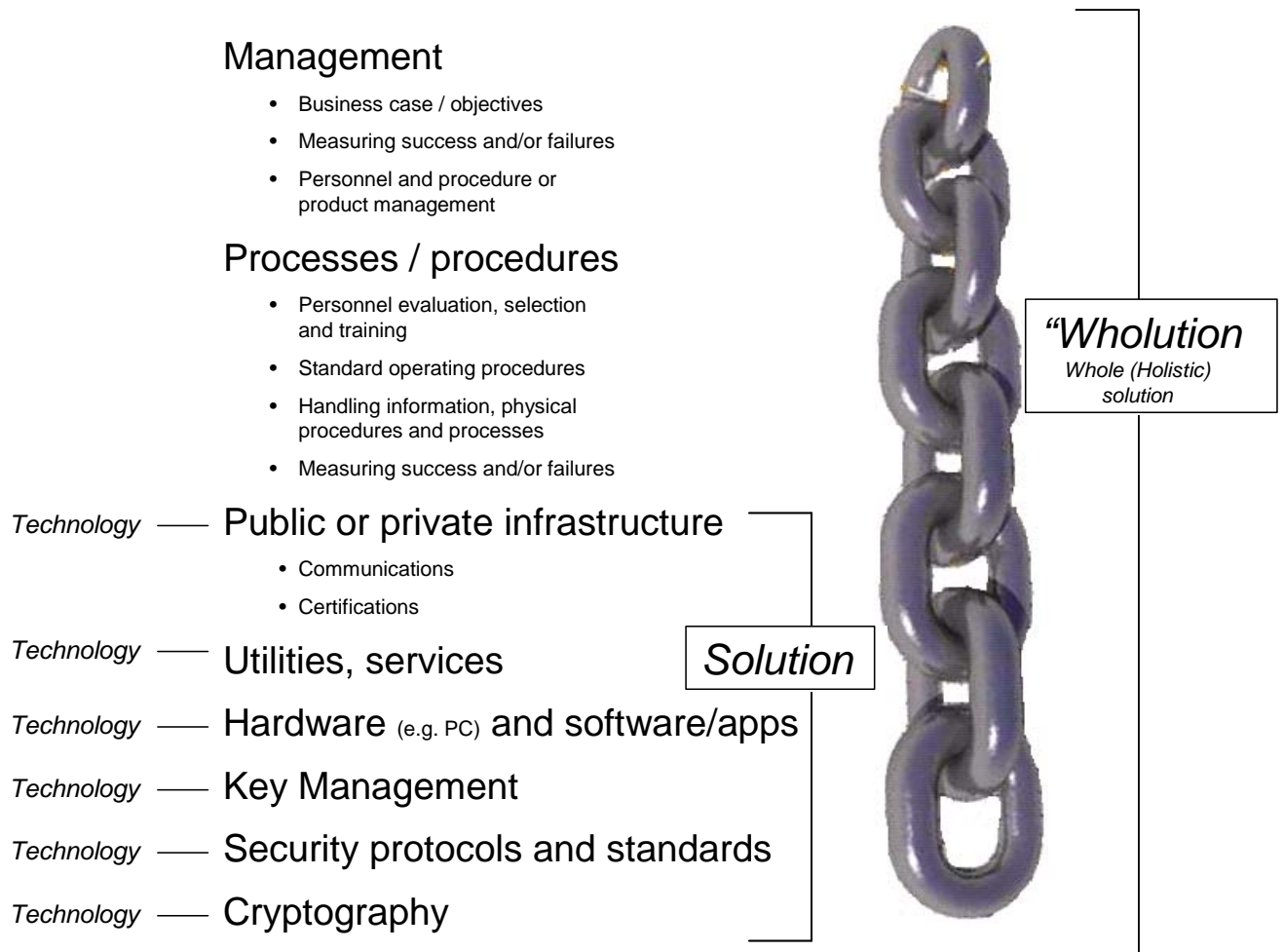
This paper focuses on those who need to design security into their system to meet a specific set of threats or to understand the threats that they may face with a given security solution. Understanding your environment, the drivers of security requirements, the critical weaknesses in today’s technology environment, and considerations for designing secure solutions will be addressed.

The Security Wholution (Whole / Holistic Solution)



In many technologies, if you find a hole in the system or a break somewhere, it can be fixed or patched up or simply avoided. Unfortunately, secure solutions do not have this luxury, any more than you can afford to have a hole in the hull of your spacecraft, no matter how small.

Thus when considering any technology or solution that requires security, it's especially important to start by understanding your whole environment in order to achieve excellence and integrity.



Moving from bottom to top: A security solution might start with cryptography (e.g. encrypting data), which is then used in security protocols which require key management, which is then implemented on software/hardware (e.g. PC), and so on up the chain. Cryptography and security protocols are technologies, but the integration of technologies into a solution is what is typically

delivered to customers. In order to deliver and integrate solutions into your business or your working life, various processes and procedures are put into place by management or by individuals users. Typically, these processes and management procedures are NOT part of the solution.

The lower you are in the chain, the more you are on clear and solid ground. Cryptography today is extremely robust, as are security protocols. However, key management technologies are not as robust. Thus even with an extremely robust way of encrypting data (cryptography), your security is compromised if you have holes in your key management technology. (I.e., the most secure lock in the world is rendered impotent if you lose the keys). As you go up the chain, the level of robustness typically goes down, however, the complexity from a holistic perspective goes up. For example, what is more difficult: implementing a cryptographic algorithm which is widely available, or figuring out how to manage your employees in terms of who to hire and whether you can trust them? What is more likely to compromise your security: Somebody cracking your encrypted file in transit over the Internet, or you accidentally losing or copying the file from your computer after it has been decrypted? As you go up the chain, tasks such as making sure that procedures are secure, that you're able to detect security failures, and so on, are even more complex. Yet ironically, much attention is focused on the lower parts of the chain, simply because they are easier to see, manage, and measure.

Take airport security, for example. The X-ray machine and metal detectors are technologies. The solution is the combination of the equipment, the personnel, training, and procedures for running the security checkpoint. The Wholution includes asking the question "where is the real weak link in the security?" and addressing the key issues (e.g. Standard operating procedures on the plane, policies for pilots using the bathroom thus opening the cockpit door, luggage policies, preventing shoulder missiles from being fired, etc). You get the idea.

The weak link in security is typically your environment (e.g. your computer) and management practices (technical and procedural). It is easy for management to deal with that which is easy to see and measure. Unfortunately that which is most important in this world is typically difficult to see and measure – it is nebulous and holistic (has to be viewed from the perspective of the whole). For more detailed discussion, see GraniteKey paper "*Management Challenges in Security*".

In summary, from most important to least, the factors affecting security design are:

Most challenging

Least challenging

People → Operating Environment → Technology
 (e.g. PC and physical environment)

You may or may not be able to control the people and the operating environment, but understanding these parameters are essential to building the appropriate technology to allow you to achieve your objectives.

Cost of failure and level of security required

$$\left(\text{Cost of Failure} \times \text{Probability of Failure} \right) \text{ VS } \text{Cost of Security}$$

While this equation doesn't illustrate the complexity of the decision to be made regarding what type of security to deploy, it captures the essence of the decision process.

Credit card companies run failure rates (bad debt and unpaid debts as a percentage of revenue) of approximately one percent, more or less. And if the failure rate goes up a little, the loss goes up a little. But the failure rate for commercial airliners is orders of magnitude lower, because failure cannot be tolerated at any level. As the failure rate goes down, the cost goes up. How far you go, depends on your objectives and the various factors listed above. In order to achieve excellence, these must be articulated in the product / technology requirements.

What level of security do you need (90%, 95%, 99%, 99.9%, 99.99%, 99.999%, etc). For example:

Lower security

- Prevent an average user from compromising security (i.e., hackers can still compromise the security)
- Prevent software attacks only (i.e., you assume that hackers don't have physical access to your computer)
- Prevent hardware attacks (i.e., hackers do have physical access to your computer)
- Prevent all potential security attacks

Higher security

One of the first things to consider is whether the user is trusted. If the user is not trusted (e.g. when you are protecting copy-righted material), you have to worry about issues such as hardware attacks, running invalid copies of software, and/or people stealing information using a variety of techniques. The system has to be extremely robust. However, if the user is trusted, many of these issues go away, and the largest issue is whether you trust the user's competency (and some would argue more so - incentive) at following procedures. Thus you could profile 3 types of users:

- Untrusted (e.g. protecting copy-righted material)
- Trusted intentions, but not competency (e.g. a loyal employee)
- Trusted intentions AND competency (e.g. a well trained military personnel).

Ability to Detect failures

Another factor determining your security solution is the ability to detect failures. For example in a financial system (e.g. credit cards), failure can be detected as a type of leakage. For example, money paid out minus money received. However, detecting the amount of *information* which

has leaked out (e.g. missing documents) is virtually impossible, because you generally don't know when this happens, and copies can be made without being detected.

As a result, in the credit industry, a very sophisticated technology called Risk Management has evolved over a long period of time. Because it is feasible to detect and track most failures, Risk Management technology can be used to set charge limits on vendors and customers, and policies on how merchandise is shipped and claimed, and is able to contained failures to within reasonable values (i.e., enough to allow credit companies to stay in business and make good profits). Thus the amount of security required in the credit card and point of sale itself is (or appears) more lax than one might expect.

Furthermore, failures are easier to detect and fix in a private system, vs. a public system. For example if a failure is detected in a meal card/payment system for a University, the amount of infrastructure and the size of the installed base is much smaller than in a global credit card system.

Unfortunately, in most solutions, the failures cannot be easily detected and measured – the quality of the solution cannot be assessed. One can provide budget for the QA team, or a 3rd party to try to hack into your system as part of the QA process (but unfortunately, the best hackers are not always for hire). And getting a certification (e.g. FIPS-140), may provide a certain level of proof that your system is secure, but such certificates typically only test components of the solution, not the Wholution.

Your ability to measure failure is important in your calculation of the effectiveness and value of your solution, however, this cannot always be readily done. A careful assessment should be done as to whether this is important and/or feasible.

Threats / Security Objectives

Security objectives should be focused on what is being protected and the benefits of that protection. Specific technologies should not enter the picture at this point, and should only be used to illustrate examples. (e.g. biometrics, smart cards, types of encryption, TPMs are technologies not security objectives)

Threats / Security objectives can be put into several broad categories. For example:

1. Proving your identity for retrieving information or making a transaction (banking, network access, server, peer)
 - a. Something you know (e.g. password)
 - b. Something you have (e.g. biometric, smart card, USB token)
2. Granting Access to a resource (based on identity)
 - a. Network access (e.g. VPN)
 - b. Control (e.g. web configuration site)
3. Protecting data that you own and share with others from theft. (User is trusted)
 - a. Stolen laptop, CD, flash memory

- b. Protecting transmission of data over Internet
 - c. Protecting data residing on a server (e.g. mail server, file server)
4. A 3rd party protecting data that they own (i.e., User is not trusted.)
 - a. Media (Music, video)
 - b. Documents
5. Protecting software
 - a. Licensing
 - b. Ensuring it has not been modified
6. Ensure data has not been changed and comes from a known entity (person or company).
 - a. May include a secure timestamp
7. Securing transactions (e.g. financial, legal)
 - a. Proving identity of both parties
 - b. Ensuring data has not changed, include the timestamp of the transaction
 - c. Encrypting if necessary
8. Ensuring system (PC, PDA, server) has not been compromised (virus, trjoans, compromised OS or BIOS)
 - a. Sniffing data (e.g. stealing online banking passwords)
 - b. Hijacking network resources (e.g. using your bandwidth and/or hard drive for peer to peer traffic)
 - c. Stealing data from your network or PC
9. Telemetry integrity
 - a. E.g. security devices
10. Improper use of equipment
 - a. Computer won't run with unauthorized devices (e.g. aftermarket battery)
 - b. Devices won't run on unauthorized system
11. Physical security is sometimes integrated with the above
 - a. Access to building
 - b. Tickets (e.g. sports games, movies)

Computing device Environment (PC, PDA, phone, etc)

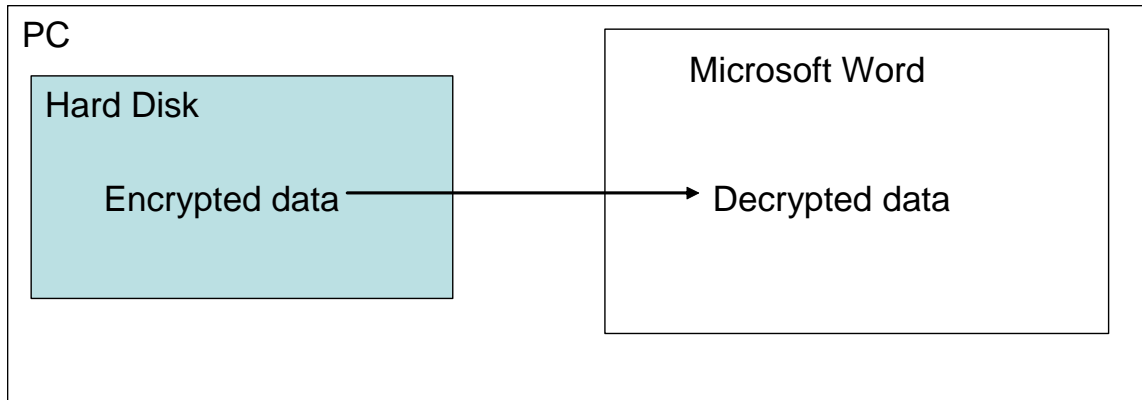
The level of security of the Wholution may driven more by your environment than by anything else (other than the security level of the people involved). For example the PC environment is not a highly secure environment. This is in sharp contrast to an embedded system, for example a point of sale (POS) terminal, manufactured by a POS vendor used in credit card transactions.

The following examples illustrate some of the challenges on a PC platform. Note that it's much easier to provide *some* level of security vs. *excellent* security (where you are keenly aware the risks and cost of failure). How you choose to view these risks is entirely up to you in terms of your risk profile, cost of failure, the sensitivity of the work you do, etc. Most of the time, *some* security is good enough, and better than no security, but other times it's not good enough.

In the following examples, all items shaded are secure. Items not shaded are not secure.

There are many other examples which could be illustrated, but the important point is to have a clear understanding of where your weak points are, and that these weak points are commonly found in the platform that you are using (i.e., not in the security technologies being deployed).

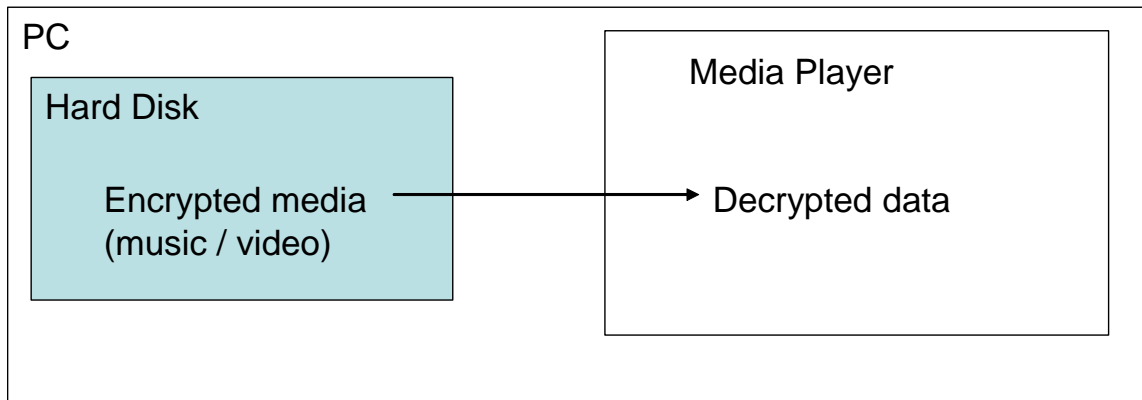
Data is encrypted on disk, but not in operating system



This protects against theft of the data after the system is turned off, but does not protect against viruses as they could scan the system. This is generally not a problem with most applications, as the user is trusted, and the likelihood of a virus stealing this data is very low. However, if the data is highly confidential, or a matter of national security, one might not accept this risk. But then, should we even allow such a document to be printed? Not only is the printed document a risk, but that document may be sent over an unsecure network to the printer. But of course, this all depends on whether the user is trusted, for if they aren't, much of the above is moot.

Note: Microsoft has added "Full Volume Encryption" into their new Vista OS. This is expected to be superior to current technologies, because it ensures that vulnerabilities like buffers and swap space which is stored on the disk is also secure (i.e., not just the file being stored, but other copies of it that might be floating around in the OS).

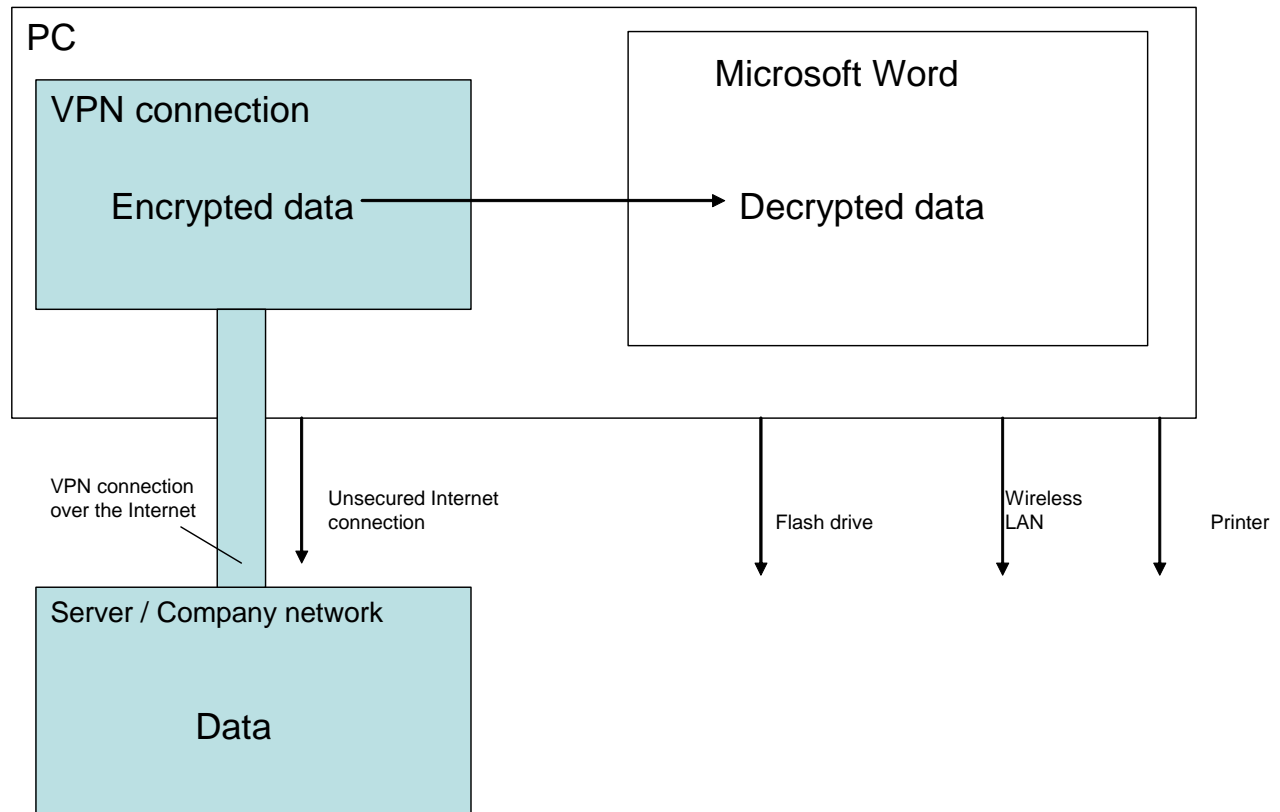
Copy written data is encrypted on disk, but not in media player



This is similar to the above example, with one important difference – the user is not trusted. With the number of hackers out there, it is naive to assume that the data could not be compromised. Even if the data inside the media player is encrypted, somewhere inside the player it would have to be decrypted, and the key to do that would be in its software. In addition, even if a hacker doesn't crack the media player, it is common to find various other illegal copies of content on the Internet from a variety of original sources. (One tactic the industry is using to address this is “ensuring” that low quality copies of media are out on the Internet “polluting” the free content). It is possible to provide a higher level of security in a dedicated non-windows device like a portable media player, but it is very common for the user to want to run their media across multiple devices such as PC's. Thus the PC might still be the weak link.

Protecting copy written material is one of the most difficult issues to address. However, one approach is simple, just prevent casual hacking (i.e., make it difficult for the average person to steal the music), put legal pressure on file sharing services, and accept a certain level of hacking. There are approaches using technologies like hardware based secure devices (e.g. smart cards, TPM's) to reduce the level of hacking, but without a trusted operating system to secure the media player itself (trusted operating systems to handle this issue are not available today), it will be a while before a higher level of robustness can be achieved. There are also approaches like embedding credit card information in purchased media to reduce the probability that you will distribute copies to your friends.

The challenge of copyright protection of software is similar, however, there has been more success in this area, as it is more difficult to create an illegal copy.

How secure is a VPN?

VPN's provide excellent security between your PC and your company's server and/or network. Even though VPN's use the Internet, they automatically encrypt all data. However, the security risk is generally on your PC, not the quality of the VPN. For example:

- Viruses on your PC could compromise security of data
- If the disk is not encrypted, or data is copied to flash drive, important data could accidentally (or intentionally) be copied off the system and lost
- Simultaneous Internet connections with other servers or applications (e.g. peer to peer) provide a path where data could be copied off the system. Wireless LAN's are particularly vulnerable because it's common for security not to be active; wireless LAN's make the PC especially vulnerable if drives are made sharable

Thus, VPN security has its limitations. However, given your objectives, it may be that the VPN itself is good enough without other considerations.

Security Hardware

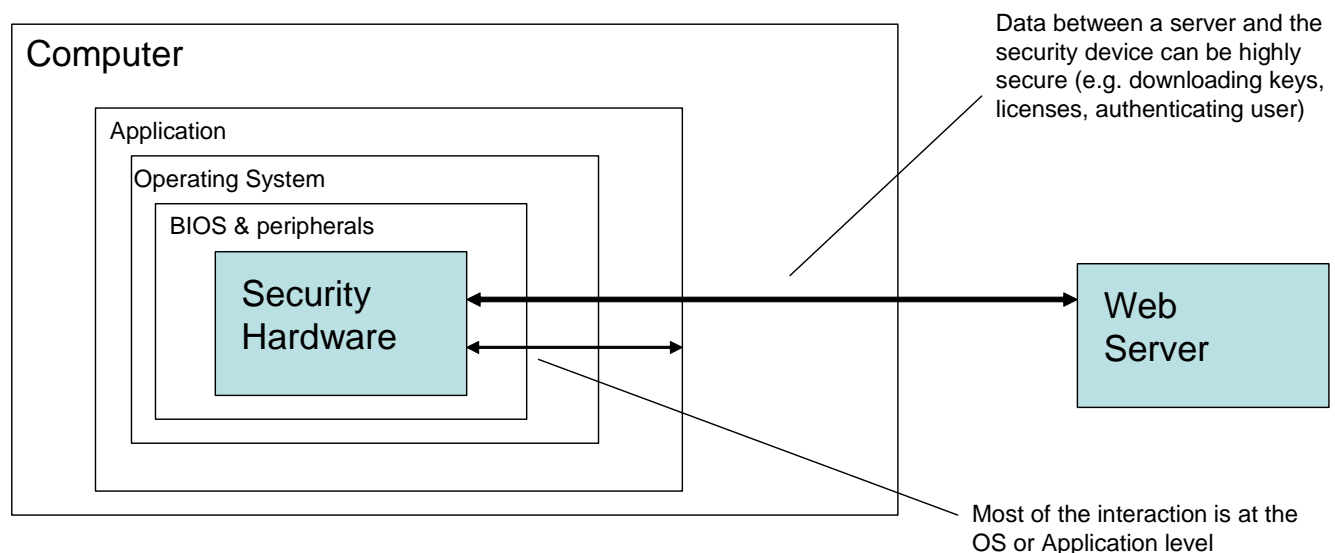
Increasingly, dedicated security hardware is being included in platforms such as PC's, mobile phones, PDA, embedded systems. For example:

- Smart Cards / SIM cards (SIMs have been used in cell phones for over a decade).
- USB versions of Smart Cards and SIM cards, typically for user identification.
- TPM's (Trusted Platform modules) are increasingly being shipped in PC's. They are designed to ensure that essential components of the platform have not been modified (e.g. BIOS, operating system), and are designed to store critical data such as keys. TPM's can also be used for other functions.
- Embedding biometrics and security chip functionality on USB drives to protect the data.
- Other dedicated security hardware. Typically with functionality similar to Smart Cards.

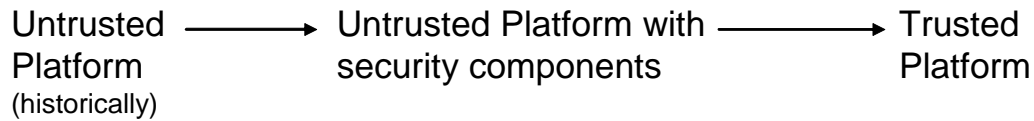
These dedicated security devices are more secure, for example, than storing data in your PC/PDA/Cell phone because the hardware is tamperproof. Some of the hardware (e.g. Smart Cards) is specifically certified by the US government (FIPS) or EU (Common Criteria). These devices typically perform the following types of functions:

- Storage of keys which can only be released if a password is given.
- Storage of unique identity keys so you must physically have the device to prove your identity. The device is challenged with data (different data each time), and its response proves that you have that device. This can be used in conjunction with a password or biometrics to add additional security.
- Secure storage and reporting of system measurements (e.g. ensuring BIOS or OS has not been changed). Can be done in conjunction with storage of keys (e.g. keys will only be released if the system has not been changed).
- Cryptographic acceleration – typically needed for embedded systems.
- Keys which are needed to run licensed software.

It is important to note that, even with the use of security hardware, security is still limited by the security of the host environment (e.g. PC). Security is improved, but the limitations must be understood.:

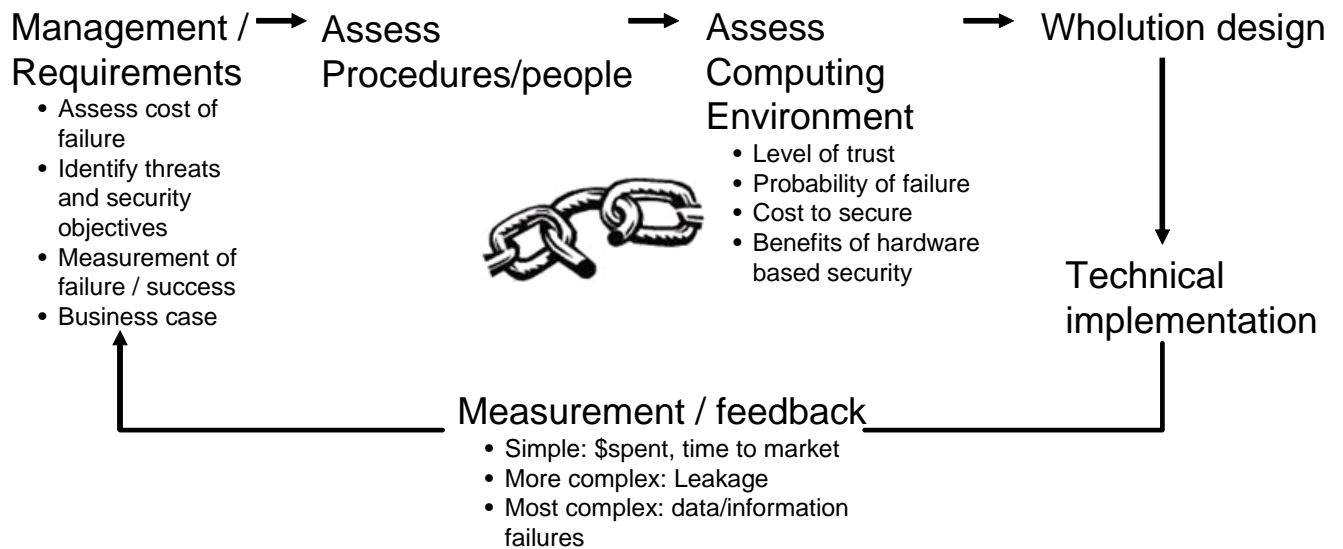


There are various emerging technologies to address the weakness of the PC platform, these will evolve over time:



- Integrating security hardware into the PC as a standard component (e.g. TPMs).
- Encrypting the hard drive.
- A secure mode for the keyboard or display to protect critical tasks such as providing passwords and approving financial transactions.
- Integrating security more closely with media devices (e.g. speakers, video).
- Trusted operating system (all software components are trusted). It is not clear when this technology will be delivered.
- Encrypting all software is something that we may see even further down the road.

Summary: Requirements are essential for excellence and integrity



Security involves complex technology, but the need for first defining requirements and assessing people, procedures, and the computing environment cannot be emphasized enough, especially in an application where a high level of security is needed. For example a military Wholution may require the highest level of security and careful consideration of all the above, however, a financial solution might require less, a health care solution even less, and a consumer application even less.

It's important to start with Management requirements, and understand what is driving those requirements. Regulations? Because everybody else is using a particular piece of technology? Is it important to the business? Then an assessment of the process/people, and the environment will lead to a Wholution design, then the selection of the technology. Finally, it's important to understand the way success or failure is measured against not only the original objectives, but to ensure ongoing excellence.

Note again the focus on traditional security technology like cryptography is not the most error prone. It's the whole system, the Wholution. Ideally it must be designed so that even the designer can't break it (the designers of the cryptographic technology cannot break those technologies – the source code is widely available).

Once the requirements are defined, one can begin to look at the specific security technologies to be used. Defining the requirements first, makes the design of the Wholution much simpler and much more effective. See the paper "*A Look at Security Technologies*" for an overview of some of the security technologies used in designing a wholution.

The Future of security solutions

As we look towards the future, it's important to focus on the most important aspects of the Wholution. The assumptions about the people who are using the technology are the strongest drivers of the technology needs and how they need to evolve. One of the most important considerations is whether the user is trusted. For example if the user is trusted:

- You can trust that the user is not going to compromise the hardware of their device (e.g. sniffing a bus with a hardware device, or disabling a trusted keyboard)
- You can trust that they won't run any hacker utilities (at least not on purpose). Thus the odds of an application getting sniffed are lower than with an untrusted user.
- You can trust that they won't run pirated applications. It's possible to do this because each application can be digitally signed by the software vendor. Windows already checks this. Many people ignore the warnings, but people can be trained to not run pirated applications. This eliminates some of the risks associated with applications getting hacked in order to steal the data. Although this requires training and adherence (easier said than done), if it's important enough it can be done.
- This assumes that the vendor who wrote the software is trusted, and there aren't any security holes in the software caused by bugs or rouge engineers. Note: Some argue that open source ultimately is more secure than software from traditional vendors because the source code to the software is examined by many people in an open forum.

If the user is NOT trusted (e.g. using copyrighted material), then it is more difficult to provide security via a trusted application:

- The user cannot be trusted to not run rouge applications
- The user cannot be trusted to not run hacking utilities to sniff into application memory
- The user cannot be trusted to not hack into the hardware

Providing strong security on the host platform (e.g. PC) can be difficult when dealing with an untrusted user. PC's will evolve over time to be more trusted, but benefits will initially accrue more to applications where the user is trusted. To improve security for untrusted uses will require more robust evolution of computing platforms. Some suggest, such security can be achieved by encrypting all data and software where decryption is done inside the CPU (both code and memory). The decryption key will be released to the CPU if the software license mechanism detects that the software is valid (e.g., a TPM could release the decryption key to the CPU through a secure channel) if the software passes a digital signature check. However, this requires much change in the technology, and requires special handling of audio and video. Stay tuned....

Industry examples of the application of security

Note that many of the key issues when creating a wholution are not core technology issues. They are issues such as:

- Lack of proper incentive, or conflicts of interest
- Lack of standards
- Too much possibility of failure due to potential leaks across the Wholution

Legal – digital signature of documents

What would it take to replace notaries and signed contracts with digital signatures? The benefits are that the document cannot be tampered with after it is signed, and the process of getting the document signed or notarized would be very quick (can be done electronically)

Issues:

- No legal precedent. Legal professionals have little incentive to push this through.
- Need standards for who should sign a document.
- How do you prevent a person from generating a digital signature with a back timestamp?
- How do you prevent a person from claiming that their private key was compromised? (For example, they could just say, “I lost my private key years ago”, invalidating all the contracts they signed.)

However, the use of signatures in vertical applications like signing of medical documents or other documents in the corporate world is evolving because it doesn’t require broad standards across a global environment.

Medical security

HIPAA (regulations for ensuring security of patient data) basically requires that a “reasonable degree” of security is used on all patient records. The regulations generally do not stipulate the type of security (with the exception for example of providing identity). The industry does not have the incentive to pay to ensure security, so they do the minimum (which may include not making patient data readily available). Insurance companies have the incentive to prevent fraud, but not necessarily to protect patients’ data (other than the minimum required) and even less incentive to make that data available to patients.. Patients have the most incentive to make sure their data is secure and widely available to them, but they have the least influence. Thus we can expect medical security to evolve as insurance companies look for ways to reduce fraud and costs (e.g. ensuring that a patient was present when a test was done, ensuring that a test is not done twice unnecessarily because the data was lost, etc.).

Credit cards

Magnetic stripe and risk management (setting credit limits for vendors and customers) are very mature technologies. We haven't seen much change (in the US), even though there has been a push to have more advanced technology such as smart cards. Why?

In addition to the cost of retrofitting the system, many transactions do not require the vendor to physically have the card:

- Internet purchases
- Recurring purchases (e.g. your health club membership charges you every month).

These would need to be retrofitted as well, adding to the cost and complexity of the solution (e.g. what percentage of your credit card transactions, on a dollar basis, involves someone who has physical presence of your card, for example at a retail outlet?). Given the maturity and effectiveness of risk management, and the cost to retrofit and retrain, is it worth it? Are there any other benefits that can be integrated with the solution to make it more economically attractive (e.g. integration with electronic cash, parking meters, etc.)

Copyright protection of content

As discussed, the challenges are:

- Given the ease of getting content over the Internet, is it worth it?
- How important is it that somebody can play their media on different device, including a PC, and does it have to work when the PC is not connecting to the Internet?
- What is an acceptable level of failure in a solution that runs on a PC?
- What are the acceptable tradeoffs in cost, complexity, and security?
- Can ways be created to make the media bundles more interesting to give users a greater incentive to pay (e.g. bundling music with video, information on the artist, interactive experiences)?
- Can a relationship be established with the user to create a recurring monthly fee in return for continuing to provide new and innovative content? (i.e., focusing resources more on developing content than security)

POS terminals

Point-of-sale terminals (e.g. credit card swiping devices) are a good example of a system that is trusted. Many of them are not based on traditional operating systems, and the vendors have made special effort to ensure security within the context of their operating environment and risk management technology.

Summary

Adding security to your solution, like any product, first requires clear understanding of customer needs as well as your own business objectives. It is perfectly reasonable to create a security Wholution that is simple, and prevents only some types of security threats, provided that this is consistent with your objectives. It is also perfectly reasonable to spend a lot of time and resources analyzing every possible threat and designing a Wholution which prevents every possible threat, if this is consistent with your objectives.

The complexity in implementation a security solution successfully is highly variable (from simple to more complex):

- The easiest solution to implement is encrypting data on your hard drive or flash drive, in order to protect against theft when the device is shut off.
- Solutions which prove your identity can readily be implemented. The simplest are those which provide you access to your PC, PDA, etc. Securing network access (corporate network, wireless network) are somewhat more complicated because they involve infrastructure. Over time they will be more sophisticated still as network access may also require that the operating system attests to whether or not the proper anti-virus and operating system patches are installed. (And in some cases, certain parts of the system must be disabled while connected to the network).
- Providing your identity for the purposes of transactions such as online banking or on-line credit card transactions are somewhat more difficult, because they involve large scale infrastructure (applications, servers, tokens or equipment may be involved).
- Digital signatures are not difficult to implement, but legal challenges can come up, as there are many cases where one of the parties who signed the document is not trusted.
- Protecting your system against viruses and other attacks is mature, but constantly evolving technology. Attacks are getting more sophisticated and can create threats such as sniffing passwords (e.g. a threat to online banking), and even hacking into applications.
- Any application where the user is not trusted is the most challenging. This includes software licensing, copyright protection of media, telemetry devices, equipment security, and even peer to peer applications (e.g. filesharing, messaging), etc. The problem is more acute if running on a general purpose operating system vs. an embedded system.

The most significant challenge to achieving true security integrity and excellence is getting management to understand and to commit to the task. As we have seen in the past, until a CIO position is created at a company, it is difficult to create excellence in IT, making it a strategic part of the company. The same is true in security. If security is not your core business, then a CSO (Chief Security Officer) will make it easier for a company to spend the appropriate time to achieve security excellence rather than it being treated as an orphan, a necessary evil that people are scrambling to get done as quickly as possible. This turns it into an exercise in managing perception, not security. However, perhaps that's OK, depending on your objectives.

However, once the management challenges are addressed and the Wholution is defined, one can begin to focus on the task of designing, building, and managing a secure product. Spending the time up front to design it well makes the task of building it much easier and success more likely.