The Need for TCPA

David Safford, IBM Research, October, 2002

Introduction

The Trusted Computing Platform Alliance (TCPA) has produced open specifications for a security chip and related software interfaces. The TCPA chip is designed to provide client machines with a minimal but essential hardware base for client side security. Recent papers have attacked the TCPA specification out of ignorance of its goals and capabilities. This paper summarizes what the TCPA chip does, gives examples of important applications of the chip, and shows why these applications are critical to client side security. This paper is organized into three parts:

- A description of modern security threats
- A description of the TCPA chip's basic functions
- How TCPA helps defend against the threats

The Modern Threats

Roger Schell, USAF said in Preliminary Notes on the Design of Secure Military Computer Systems" on January 1, 1973,

> ***"From a practical standpoint, the security problem will remain as long as manufacturers remain committed to current system architecture, produced without a firm requirement for security. As long as there is support for ad hoc fixes and security packages for these inadequate designs and as long as the illusory results of penetration teams are accepted as demonstrations of a computer system security, proper security will not be a reality."***

Nearly thirty years ago, Roger Schell accurately predicted the exact situation we find ourselves in: systems not designed for the modern Internet threats, poorly implemented, forcing the installation of nearly daily security patches, and many millions of systems being compromised on an ongoing basis.

In January 2000, Roger Schell and Michael Thompson wrote a paper, "Platform Security: What is Lacking" [2], which analyzes the modern internet threats, and shows how the lack of platform defenses impede important potential applications, such as electronic commerce. The article goes on to suggest minimal hardware along the lines of TCPA, that can help defeat these threats. Roger's paper is well worth reading.

Threat Methods

What are the modern internet threats? There are many categories of methods the hackers use to attack a system. Three of the major categories are:

- Unsafe programs
- Misconfigured programs
- Buggy programs
    - Buffer Overflows
    - Parsing Errors

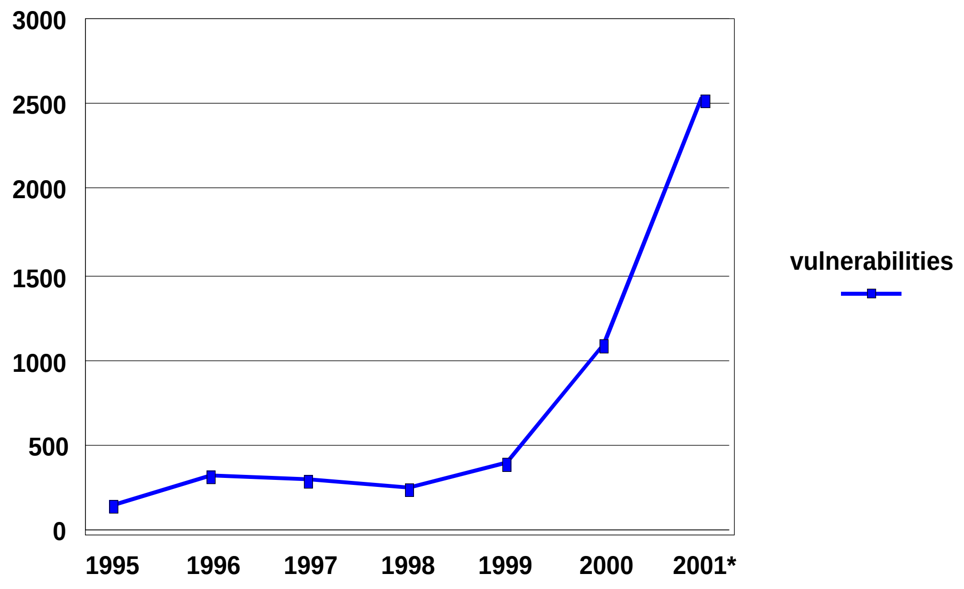
Unsafe programs are ones whose designs are inherently unsafe in the Internet environment. Examples include telnet and ftp, which normally send usernames and passwords over the network unencrypted. Hackers can easily record the username and passwords as they go over the network, and replay them to steal a person's access. Another example is rlogin which uses the sender's IP address for authentication, which is easily forged. These programs were not designed with security in mind, and are inherently exploitable by hackers.

Misconfigured programs are ones that have adequate security features, but that require the features be turned on or configured properly. For example, NFS can be configured to export a filesystem in read-only mode, which is very safe, but it can also be configured to export a filesystem read-write to everyone, which allows the hackers to break in easily.

Buggy programs are ones which are designed and configured properly, but which have implementation or coding flaws which can be exploited remotely by the hacker by sending carefully crafted malicious data that exploits the bug. The two most common program vulnerabilities are buffer overflows and parsing errors.

A buffer overflow is a programming error in which the programmer creates a fixed size buffer to store remote input data, but fails to check the length of the input data to make sure it fits in the buffer. The remote hacker can then send more data than can fit in the buffer, in which case the excess data overwrites whatever follows the end of the buffer. The malicious data will typically include malicious executable code, which gives the remote hacker access to the machine. Buffer overflows have been well understood for over thirty years, and are easily avoided, yet they consistently are the most frequently discovered and exploited vulnerability.

Parsing errors occur when the contents of remote input data is not properly checked. For example, a web server accepts requests to view html files by name, verifying that the specified file is allowed to be viewed. If the parsing of the filename is not done correctly, the server may improperly allow the hacker to view data (such as the password file). One example of this was the "directory traversal" bug in which web servers did not recognize that requests for ../../../etc/passwd would allow access outside the allowed directory.



Threat rates

This chart (compiled from CERT data), shows the number of different exploitable program bugs (such as buffer overflows, and parsing errors) discovered in software each year. In 2001, the rate was an amazing seven bugs **per day.** So far in 2002, the rate has been running at double this, or an astounding fourteen bugs per day! The problem is that with the bugs being discovered at this rate, it becomes almost impossible to apply all the necessary patches to fix them. For the hackers, this is an ideal situation, as most systems will have at least one of these known vulnerabilities. Client systems are particularly vulnerable, as they typically do not have security aware administrators to keep up with the patches.

The problem with software

Why are there so many bugs being discovered? Well, first of all, modern system are incredibly complex. A typical Unix or Windows system, including major applications, represents something around 100 Million lines of source code. Several recent studies [3,4] have shown that typical product level software has roughly one security related bug per thousand lines of source code. Thus a typical system will potentially have a **hundred thousand security bugs**. It is not surprising that we are finding five thousand of these bugs per year, and that the rate of finding them is increasing so dramatically.

Threat Trends

One trend in the hacking threats of particular interest from the client perspective, is the trend of hackers to focus on attacking the client. In the '80's, hackers largely attacked the network, passively sniffing passwords, and actively hijacking network sessions. As applications increasingly encrypted data going across the network, hackers then turned their attention largely to attacking servers directly, mainly through misconfigured or buggy services, like web servers. As companies have responded with firewalls, intrusion detection, and security auditing tools to protect their servers, hackers have increasingly turned to hacking clients.

The hackers are interested in client PC's for two reasons. First they are an easily available source of machines from which to launch Distributed Denial of Service (DDoS) attacks. But even more importantly, they frequently contain usernames and passwords used to access servers. Thus hackers will break into clients machines, simply to get usernames and passwords to get into a target server. A good example of this is the QAZ virus, which targets client machines through email and windows file shares, and which has a payload that captures usernames and passwords, and emails them off to the hackers.

In the past no one has been concerned with client side security, but as these client attacks increase, they will pose increasing threats to both users and servers in electronic business. Protection of sensitive authentication data, such as passwords will become critical for electronic business to succeed.

What Is TCPA?

The Trusted Computing Platform Alliance was formed to establish an industry standard for a trusted computing subsystem to be added to PC's. IBM has been shipping a predecessor, called the "Embedded Security Subsystem" (ESS) chip on the PL300 desktop and T23 Thinkpad systems for over two years. The ESS chip was basically a public key smartcard chip placed directly on the motherboard's SMB bus. The concept was to make public key hardware tokens available at very low cost, by embedding them, and eliminating the need for separate smart cards and readers. Other companies were looking at similar solutions, and it became clear that there needed to be a single common standard. The TCPA organization has published open, freely downloadable specifications for all of TCPA[1]. The TCPA main specification defines a chip that meets the security requirements of all the member companies. In addition, other TCPA specifications cover PC specific interface and software details.

The TCPA chip itself has three main groups of functions:

- public key functions
- trusted boot functions
- initialization and management functions

The public key functions are very similar to IBM's original ESS chip design, (which already has GPL'ed driver code in active use by several projects.) They provide for on-chip key pair generation using a hardware random number generator, along with public key signature, verification, encryption and decryption.

The "trusted" boot functions provide the ability to store in Platform Configuration Registers (PCR), hashes of configuration information throughout the boot sequence. Once booted, data (such as symmetric keys for encrypted files) can be "sealed" under a PCR. The sealed data can only be unsealed if the PCR has the same value as at the time of sealing. Thus, if an attempt is made to boot an alternative system, or a virus has back-doored the operating system, the PCR value will not match, and the unseal will fail, thus protecting the data.

The initialization and management functions allow the owner to turn functionality on and off, reset the chip, and take ownership. This group of functions is somewhat complex, to provide strong separation of what can be done at BIOS (boot) time, and what can be done at normal run-time, so that sensitive operations (like reading the endorsement key) can't be performed by malicious applications trying to compromise one's privacy.

What TCPA is Not

Some of the papers critical of TCPA claim that TCPA is primarily intended to support Digital Rights Management (DRM), such as the copy protection of music or video data, on behalf of the content owners. They argue that TCPA would take away user rights on their own machines, preventing backup, time and space shifting of legally purchased content. Debating the merits of DRM is a complex, controversial topic, and won't be covered here. However, it is easy to point out that the TCPA chip is not well suited to DRM tasks, and IBM's implementation of the chip was neither designed not evaluated for the necessary tamper resistance needed to provide effective copy protection (Personally, I do not believe it is possible do provide effective copy protection at all, but that's another paper).

The TCPA chip is not particularly suited to DRM. While it does have the ability to report signed PCR information, and this information could be used to prevent playback unless a trusted operating system and application were in use, this type of scheme would be a nightmare for content providers to manage. Any change to the BIOS, the operating system, or the application would change the reported values. How could content providers recognize which reported PCR values were good, given the myriad platforms, operating system versions, and frequent software patches?

Second, the IBM version of the TCPA chip, while evaluated to FIPS and Common Criteria security standards, has specifically omitted tamper resistance from the evaluation target. The IBM chip sits on the LPC bus, which is easily monitored. The chip is not defended against power analysis, RF analysis or timing analysis. The bottom line is that the physical owner of the machine could easily recover any DRM secrets from the chip. This apparent lack of security in the chip makes perfect sense when you realize that the purpose of the chip is to defend the user's data against remote (software) attack. When the goal is to protect the user's keys and data against external attack, we simply are not concerned with threats based on the user attacking the chip, as the user already has secure access to his own data.

Important Applications of TCPA

Protection of user authentication keys

Given the large number of vulnerabilities in client system, and the trend of hackers to target client machines looking for passwords, it is vital to provide some way to protect sensitive authentication information such as passwords and private keys. TCPA provides exactly this protection.

A user can generate an RSA public/private key pair on the TCPA chip. The private key can be configured never to leave the chip. This private key can be used with protocols like SSL to provide strong user authentication over the internet. The TCPA chip does not prevent the hacker from exploiting vulnerabilities in the client, but it does protect the user's private key. No matter what the remote hacker does, he cannot get a copy of the private key out of the TCPA chip.

Protection of user file and filesystem keys

Another important application of TCPA is to protect a user's sensitive files or data. Using TCPA, a user can seal a master encryption key for an encrypted file system under a TCPA PCR register. So long as the operating system environment remains unchanged, the encrypted filesystem master key can be retrieved from the TCPA chip. If, however, a hacker successfully attacks the client, and installs a back door, or keyboard monitoring software, the TCPA system will notice that the operating system has changed, and will not release the key, thus protecting the sensitive files from attack.

Summary

Hackers on the internet present a threat to clients and to the authentication used in electronic commerce applications. Our client side operating systems and application are so complex, that bugs and security vulnerabilities in software are virtually inevitable. It is therefore critical to provide some hardware base protection for sensitive authentication and encryption keys, that protects them from hackers even in the presence of vulnerable software. TCPA provides this critical hardware security function, protecting an individual's authentication and encryption keys from remote software attack.

References

[1] TCPA website/specifications:

http://www.trustedcomputing.org
http://www.trustedcomputing.org/docs/main%20v1_1b.pdf
http://www.trustedcomputing.org/docs/TCPA_PCSpecificSpecification_v100.pdf

[2] Roger Schell, and Michael Thompson, "Platform Security: What is Lacking" Elsevier Science, Information Security Technical Report, January 2000 (http://www.elsevier.nl/inca/publications/store/3/1/1/8/5/)