Trusted Computing:

The Linux Front

June 29, 2004

Trusted Computing — Helping the Linuxer

As any Linux user knows, Linux is much more secure than Windows. However, we still suffer from various security issues:

- Worms and Trojan horses – just like in any other OS.
- Unsecure Internet – unsecure network protocols.
- Problems in identification of servers – man in the middle attacks, spoofing, etc.
- Denial of Service (DoS) attacks.
- SPAM.

Trusted computing can help us in reducing these problems and threats to almost a zero.

Trusted Computing — Helping the Linuxer (By Products)

Some of the by products of the trusted computing initiative can also help the everyday Linuxer:

- Faulty OS causing bad networking – some of the attacks on other OSes, cause network overload. When trusted computing will reduce the hazards on those system, Linuxers could have better networking.
- Cryptographic hardware on every computer – will make cryptographic computations faster, and thus, cryptography will be widely deployed and used.

Trusted Computing – Why Not?

Attestation is the source of all evil (TM).

Attestation is useful for the sake of preventing the state of the computer from changing without the knowledge and the consent of the owner.

Currently, the attestation model of the TCG design can also be used to prevent the state of the computer from changing, even if the owner wishes to have this change.

It is true that the owner is free to change and alter her software. However, in the world of attestation this might have a side-effect: this will eliminate the ability of the computer to interact with other systems (the systems that require attestation of trust).

Today, third parties have no reliable way to tell what software you are using. Thus, they have no reliable way to force you to use the software of their choice.

The current state can be viewed as an advantage for the computer owners – in terms of improved competition, software choice, software interoperability.

More important is the fact that owners have the ability to control their own computers.

On Privacy and Freedom of Choice

As far as security is concerned, the TCG assures that the computer would be secure. So where is the problem?

Many accuse the TCG initiative of causing problems while solving security issues. These problems have much to do with privacy, freedom of choice, etc.

One main problem is the initial trust. You need to trust the CEs and PEs used in the TCG model, to start the trust chain. What if no CEs and PEs are truly trusted?

As the task of validation of the security requirements is not computationally feasible, even the most dedicated CE could not sign its statement with 100% assurance.

Another main problem is the fact that the TCG could be used to reduce the freedom of choice an owner has on her system.

On Privacy and Freedom of Choice (cont.)

Problem: Third-party uncertainty about your software environment is normally a feature, not a bug

The remote attestation is one of the features that jeopardize an honest user. This can be used against the user (or owner) to impose restrictions on what she can (or cannot) do.

When we design a security solution, we must specify the threat model. The TCG assumes that the owner is also a threat. It is usually not the case.

Another concern is the fact that preventing one attack may lead to new attacks. For example, adding heavy cryptographic computations for verifying trust, can be used for denial-of-service attacks.

Forcing Users to Use a Specific Program

Many sites today support Internet Explorer only. Not only this is against any Internet standard, it prevents people from using other browsers. However, many browsers know to present themselves as Internet Explorer, thus, bypassing the browser check imposed by the web site. In a TCG-world this will not work.

If websites could ask for validating trust, guess which web server is going to require Internet Explorer as the trusted browser?
Today, if a web designer is annoying enough, he can close his website for non-IE browsers. But how he can know what browser you are using?

You tell him!

So, instead of telling him you use Mozilla/Opera/Konquerer/lynx, tell him you are using IE (really easy to do). However, once attestation will be mandatory, everyday hacker could not lie to him and tell him she is running IE, her computer could not attest that she is running IE, if she is running Opera.

Preventing Interoperability and Tethering

By demanding attestation that the OS is of one kind and not another, we can prevent running programs in the "wrong" OS. We can also lock data to be opened only by a specific platform in a specific environment. Thus, preventing migration from one OS to another.

Moreover, there are mechanisms in the TCG design, allowing to bind data to a given application (or a given platform).

By binding all content created in any of her products to Windows, Microsoft can force content users to use Windows.

What you will do in Linux, if you cannot open MS-Word documents? How could you interact with non-Linux users?

Preventing Interoperability and Tethering (cont.)

If a DVD drive could require the OS to be trusted before the data is decrypted and sent to the OS, DVD could be designed to be played only on one OS.

No more WINE and other Non-emulation projects – the software (even the most friendly open-source compiled for windows) might not work (assuming trust checking).

Preventing Interoperability and Tethering (cont.)

If all debugging for reverse engineering can be detected and reported (once you change the CPU into debugging mode, the TPM will be alerted), reverse engineering could become very-very-very hard.

Even if the content has not been bound to one OS (to rule them all), if the format is kept closed, no reverse engineering means no possibility to use it in other OSes.

No DVD-players for Linux, no next-generation-NTFS support in Linux Kernel, no next-generation-WMA/DOC/PPT support in Linux applications.

Preventing privacy

Today there is anonymity to some extent in the Internet. A Chinese person who uses the Internet to surf into "democratic" sites can do so without the risk of the government finding that. In the TCG-world, as communications will be authenticated, then the government can find this out.

Automatic document destruction

We can add a feature saying that any email 6 month old is deleted forever from the records. Some companies would like that ability (Enron case, Microsoft anti-trust case, etc.). TC can also implement fancier controls: for example, if you send an email that causes embarrassment to your boss, she can broadcast a cancellation message that will cause it to be deleted wherever it has gotten to.

But I'm Secure — I'm using Linux+OpenSource

Consider the case when a person wishes to recompile her kernel.
If the "trust" certificates are linked to the source, then she can use whatever binary she wishes and claim it's the output of the source (and thus run "un-trusted" applications as "trusted").
If the "trust" certificates are linked only to binary kernel (already complied), then people could not compile their kernel again, with different options than the ones used by the distributor.
In both cases – we can not really get trust.

How to Solve Linux Problems

The main problem is the fact that open source environment is hard to certify as trusted. Actually, in some cases, it can be considered less trusted than closed-source.

A basic solution is to have the RPMs (or any other **binary** packages) be signed and verified just like any other software.

This is sufficient for many Linux users, but will actually decrease the Linux ability to adopt — no longer people could compile their own kernel and use it!

First Solution – We Sign All

Today you can find over the Internet, many services for free:

1. Free Email services,
2. Free DNS services,
3. Free hosting services,
4. . . .

One day, some Linux-friendly service of giving certificates-to-all, for free, will emerge, and sign certificates for any binary.

Problem: Someone might not trust this Linux-friendly person, and all his certificates will worth nothing (at least with respect to the untrusting).

Second Solution – You Pay – We Sign

Distributions will sell you certification services (like they sell support).
You can buy 100 certification processes, in which they'll take your compiled programs, and certify them as trusted.

Problem: Someone might trick them to sign untrusted binaries. Once this will happen, they are likely to be less trusted, ...

Third Solution – You Pay – We Compile and Sign

Distribution will sell compilation and certification services.
The source code will be held by the distributions. You will download the source, will run *./configure* or any other mechanism, and get a set of compilation instructions.
Then, you will send those instructions to the distribution, which will compile for you the program using these options.

Problem: Do you trust them?
Answer: Compile locally (under the same compiler), and verify that the binary is what you wanted.

Problem: And if I want to change the source code?
Answer: You'll probably pay more. (another solution: buy yourself certificate as VE)

Fourth Solution – Change Trusted Computing

The best solution is to allow the owner of the machine to have it's TPM secret key (endorsement key). Allowing the owner to have this key, she can sign whatever trust "certificate" she wishes on the system.
This can be used to "prove" that she is running IE 6.0 (when she is actually using Mozilla).

Problem: Attestation has no longer have a meaning.
Answer: Partially correct. In the industry, or when the user of the machine is not the owner – the user cannot bypass the mechanisms, and thus, for organizations, there is ability to prove trust. In your home, you could be able to do whatever you wish.

Fifth Solution – Let the Windowers Suffer

In case Linux, OpenSource, free speech, and other related movements will gather enough influence, almost nobody would require attestation.

In that case, we could (as Linuxers) enjoy the good side of trusted computing. The ones that use OSes that requires them to have the TPM activated (Longhorn is expected to use NGSCB/Palladium – Microsoft's trusted computing base), will suffer also the disadvantages.

This will strengthen Linux, but only if:

- The US senate will not pass laws requiring the activation of TPMs.
- The related movements will have enough power.

Sixth Solution – Change TCG's Design

If the remote attestation will be removed from the design criteria, trusted computing might not be that bad...

It would be easier to track and remove viruses and worms, reduce spam (as many spammers today use other computer as zombies), etc.

Dedicated cryptographic hardware could be used to enhance general Internet trust and security, and even reduce spam in other ways (various mechanisms to reduce spam are based on cryptographic ideas).

Problem: I can't do it. I am not related to the TCG.

Answer: Yes, but we (IBM) are.

More Reading

Whether you think trusted computing is the solution, or you think it is the problem, there are many voices out there:

- Trusted Computed Platform Alliance — http://www.trustedcomputing.org
- Trusted Computed Group — http://www.trustedcomputinggroup.org
- Microsoft's Next Generation Secure Computing Base – Technical FAQ — http://www.microsoft.com/technet/treeview/default.asp?url=/technet/security/news/NGSCB.asp
- Ross Anderson's TCPA FAQ — http://www.cl.cam.ac.uk/~rja14/tcpa-faq.html
- EFF's Trusted Computing: Promise and Risk — http://www.eff.org/Infra/trusted_computing/20031001_tc.php

Acknowledgments

I would like to thank Prof. Eli Biham for his great help in writing the lecture. Many talks and conversations among us have helped me to grasp the trusted computing initiative.