

AusCERT 2007

Ivan Krstić, OLPC

`krstic@solarsail.hcs.harvard.edu`

This slide deck differs from the version presented. Slides introducing the OLPC project have been elided.

My key point today

We have failed as an industry, and modern desktop security is completely broken as a result.

Put differently, it's about *the user*, not about TCP/IP, or SSL, or AES, or IPSEC.
The user.

IDC: 75% of all corporate machines are infected with spyware and malware.

Known virus count surpassed 100 thousand
in 2004, and *1 million* since then.

This doesn't count spam or phishing or related problems.

Key symptom: desktop security relies on the user to make informed, sensible choices...

About things they don't at all understand.

Web Site Certified by an Unknown Authority



Unable to verify the identity of www.URL.com as a trusted site.

Possible reasons for this error:

- Your browser does not recognize the Certificate Authority that issued the site's certificate.
- The site's certificate is incomplete due to a server misconfiguration.
- You are connected to a site pretending to be www.URL.com , possibly to obtain your confidential information.

Please notify the site's webmaster about this problem.

Before accepting this certificate, you should examine this site's certificate carefully. Are you willing to to accept this certificate for the purpose of identifying the web site www.URL.com ?

Examine Certificate...

- Accept this certificate permanently
- Accept this certificate temporarily for this session
- Do not accept this certificate and do not connect to this web site

Help

Cancel

OK

Office 2.0



Words, words, words. Technical explanations to things that you don't really want to understand but we didn't want to make the choice for you, so here we are.

- Click "Do What I Want" to ignore our worry that you don't understand this application well enough to make good choices
- Click "Don't Do What I Want" to not do the thing you intended to

Do What I Want

Don't Do What I Want

Do not show this warning again

Why do we show these dialogs? Because we're too scared – legally, or as technologists – to make a decision for the user.

So we weasel off responsibility to the user,
stick our head in the sand, and pretend this
was the right thing to do.

But the user knows as much about computer security as they do about gravitational waves, closed strings and D-branes.

To users, security dialogs are a black box where clicking 'Permit' or 'Allow' maximizes the likelihood of getting their work done.

We've been using Pavlovian conditioning to train users to ignore security.

And here we are. It's 2007, we've failed, but
how on Earth did we get here?

Follow the #1 fundamentally broken
assumption of desktop security.

“All programs should execute with the full permissions of the executing user.”

Or: “security should exist between different users, but not different programs.”

Fun result #1: a program can erase or corrupt your documents, mail them to Russia, read your mail, pretend to be you, spy on you with your microphone or camera, hand over control of your computer to a third party...

And that program is Minesweeper.

Fun result #2: to have any semblance of security, we must maintain blacklists – one of the dumbest ideas in the entire field.

We have a pretty bad track record about it
as an industry.

Viruses? Blacklist them.

Spyware? Blacklist it.

Malware? Blacklist it.

And then we're surprised when this doesn't
work.

Here's the issue: our fundamental assumption is 35 years old.

Ritchie, Thompson, 1971: the program *is*
the user.

1971: 7 years before the first international packet-switched network, 12 years before first TCP/IP WAN, 20 years before the web.

No conceivable way for untrusted code to “appear” on a machine. You had to physically put it there via tape or punched card.

Compare with today: untrusted code on
every website you visit.

Sticks and stones – we're using a security model that predates personal computing!

The kicker: we knew about wholly superior approaches as far back as 1959.

Alan Kay: “for today’s programmers, it’s not computer science, it’s computer pop culture.”

For security people, it's security pop
culture, and it's toxic.

Rice University Computer, B5000, BLI,
Dennis and van Horn supervisor, PDP-1,
Magic Number Machine, CAL-TSS, System
250, CAP, Hydra, StarOS, System/38,
IAPX 432

And that's just 1959-1981!

Know the past. It often holds answers to questions that we're just now asking. Ask the physicists!

OLPC security

Means six core things to us.

- Prevent hardware damage by software.

- Prevent hardware damage by software.
- Provide recoverability and openness (“learner’s machine”)

- Prevent hardware damage by software.
- Provide recoverability and openness (“learner’s machine”)
- Prevent permanent data loss.

- Prevent hardware damage by software.
- Provide recoverability and openness (“learner’s machine”)
- Prevent permanent data loss.
- Protect the user’s privacy.

- Prevent hardware damage by software.
- Provide recoverability and openness (“learner’s machine”)
- Prevent permanent data loss.
- Protect the user’s privacy.
- Discourage the laptops being a platform for attacks.

- Prevent hardware damage by software.
- Provide recoverability and openness (“learner’s machine”)
- Prevent permanent data loss.
- Protect the user’s privacy.
- Discourage the laptops being a platform for attacks.
- Keep the laptop under control of its owner.

Goals while doing so

- No user passwords.

Goals while doing so

- No user passwords.
- Out-of-the-box security.

Goals while doing so

- No user passwords.
- Out-of-the-box security.
- Open design.

Goals while doing so

- No user passwords.
- Out-of-the-box security.
- Open design.
- No reading required.

Goals while doing so

- No user passwords.
- Out-of-the-box security.
- Open design.
- No reading required.
- No lockdown.

Our solution

I designed a new platform called Bitfrost.

Instead of protection *from* executing untrusted code, Bitfrost protects the machine *while* executing untrusted code.

Main idea: run each application in its own virtual machine (really, OS container or zone).

Give each program only the permissions it needs.

With this approach, viruses and spyware
largely “go away”.

Hardware damage: NAND flash is rate-limited, BIOS is cryptographically protected.

Can restore full factory system by resetting
a few links.

Privacy: microphone and camera LEDs are present and wired in series in hardware, explicit user action is required to access documents.

Full network stack and filesystem isolation.

Result #1: minesweeper can no longer do anything bad.

Result #2: Bitfrost *never* presents dumb prompts to the user.

So, is this the future?

Well, OS isolation isn't exactly new, and virtualization has been around since the 60s.

But we've actually decided to break backwards compatibility to provide strong security.

I firmly believe there is no other way.

Only time will tell, but I'm hopeful that Bitfrost will be a leap forward in systems security.

The real question is how much systems security will matter a few years down the line.

Things you should fear

“Web 2.0” has already turned the browser
into a micro-OS.

Firefox 3.0 is adding complex offline storage
and execution primitives.

Already, web applications are drastically changing what desktop security means. You don't run the code, you don't store the data.

It's about to get worse. Welcome to Web 3.0, the “let's kill all compatibility and openness to try and impose vendor lock-in on the web again” version.

Silverlight: Microsoft's runtime, Microsoft's markup, Microsoft's graphics/animation framework, Microsoft's development tools, Microsoft's audio/video codecs.

Adobe Apollo/AIR: Adobe's runtime,
Adobe's markup, Adobe's
graphics/animation framework, Adobe's
development tools, Adobe's audio/video
codecs.

JavaFX: Sun's "like AJAX, but minus junky HTML, DOM, CSS, all that stuff."

OpenID: “why bother phishing individual accounts when you can phish the single sign-on?”

No idea what's going to happen to the web,
but the security picture is getting scary, and
browser security teams are already having a
hard time keeping up with issues.

To make things more fun a few OSI layers below, the network is changing beneath our feet.

IPv6 is coming: OLPC will be deploying it on millions of units. Heard of Teredo tunneling? Know the original name was “Shipworm”?

IPv6 also kills NAT, probably the most widely-deployed “drive-by” security measure on today’s networks.

PDAs, smartphones, crackberries: data flowing everywhere. Who's protecting the data flow?

We've seen limited for-profit malware on mobile devices. Now there's universal malware for Symbian.

So what do we do?

All of this can be dealt with. We can avert
much of the impending doom.

But not with a 1971 security model.

We can get our hands dirty now and fix things properly, or be condemned to another decade of empty promises and lousy “solutions.”

My call to you: let's wake up, shake off 35-year old assumptions, work less on sexy problems and more on hard ones, and let's fix this.

We've started the discussion with Bitfrost.
Join us.