

RED HAT :: CHICAGO :: 2009

SUMMIT

FOLLOW US:

[TWITTER.COM/REDHATSUMMIT](https://twitter.com/redhatsummit)

TWEET ABOUT US:

ADD #SUMMIT AND/OR #JBOSSWORLD TO THE END
OF YOUR EVENT-RELATED TWEET

presented by



RED HAT :: CHICAGO :: 2009

SUMMIT

Solving the Threat of Dirty Devices

Presenter

Spencer Shimko

Senior Security Engineer, Tresys Technology

09.03.09

presented by



Agenda

- Quick introduction to the problem and solution
- Delve into the technical rabbit hole
 - Hey, I'm a techie too
 - Technical challenges
 - Technical solutions
 - Security – our cup of tea
 -



The Problem with USB Thumbdrives

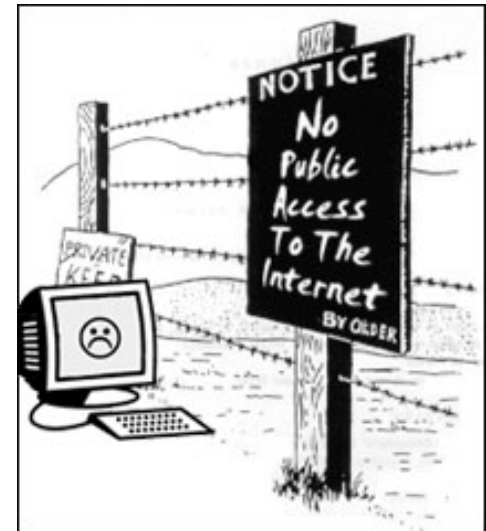
- Store data
 - Visible data intentionally stored, and
 - Meta-data not clearly visible by user
-
- Mobile – very mobile
 - Easily move data between different environments
 - Home, work, random friend's laptop, etc
-
- Often comes with pre-installed software
 - Is this software safe?
-
- Computers often “just run” code
 - autorun.inf



More specifically.... customer concerns

- USB thumbdrives have become integral in workflow
 - Transfer data between systems/environments
 - Used to move data through air gaps
 - Sometimes the **only** way to transfer data
- Cyber-terrorists are pretty darn smart
 - Couldn't connect to disconnected networks but...
 - Realized prevalence of thumbdrives
 - Targeted attacks leverage
 - USB drives
 - Known standard data formats (image formats, PDF, etc)
 - Hide instructions using "hidey-holes"
 -

* All of this resulted in a complete ban of USB drives across the US Government



How can we fix this *quickly*?

- Open source - Linux
 - Rapid devel. and system integration
- COTS
 - Red Hat Enterprise Linux
 - Well supported, vetted by the feds
 - VM Fortress + VMware
 - Secure virtualization, easy integration and deployment
- GOTS – Assured File Transfer
 - Robust filtering platform, RHEL 4 based



How quickly is quickly?

- Combining OSS, COTS, and GOTS
 - Functional prototype demonstrated in 1 week
 - Not just blinking lights, *functional* being the keyword
 - First fielded device in 1 month
 - Full featured, larger scale deployment in 4 months
 -
 -



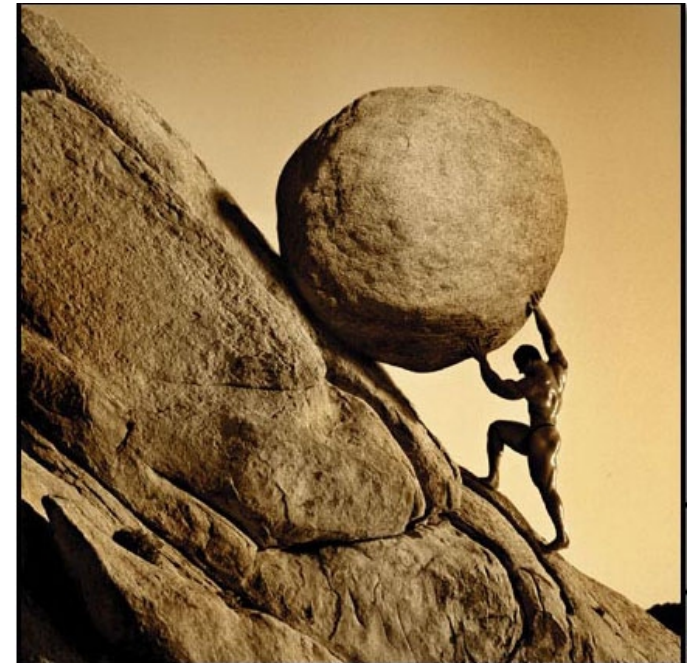
So what'd you do again?

- Kiosk-style system that cleans thumbdrives
 - Utilizes enterprise-class filtering and content inspection engines
 - Best practices in security applied at all levels
 - After all, this thing expects “the malicious” all the time
- Features
 - Dual mode, single mode, erase mode, virus updating
 -
 -

Now BRING ON the fun stuff!

Major Challenges

- System stability
 - Filters are enterprise-class, not laptop friendly
- Filesystem support
 - NTFS, HFS+ painful
 - Windows does what it wants, ignores expectations
- Performance
 - Laptop form-factor restrictive
 - Virtualization IO bottleneck



Major Challenges pt. 2

- Security
 - Malicious content and targeted attacks the norm
- Believe it or not: power saving settings
 - Laptops never been Linux's cup-o-tea
 - Screen blanking, low power mode for CPU
- Non-technical end-users (and administrators!)
 - Created endless corner cases
 - You removed the thumbdrive when?!?! (HAL hangs)
 -

Major Challenge Solutions

- System stability
 - Be as stateless as possible
 - Filter snapshotting supported for marking “known-good”
 - Generously apply iostat as necessary
- Filesystem support
 - fdisk *still* > parted
 - Partition tableless devices common, deal with it
 - Tons of cross-platform testing
- Performance
 - Hundreds of tweaks applied and benchmarked

Major Challenges Solutions pt. 2

- Security
 - Nauseating detail later
- Believe it or not, power saving settings
 - Force high-performance every way possible
 - vbetool + xorg.conf to fix screen blanking
- Non-technical end-users (and administrators!)
 - Replace HAL, it randomly hangs on device removal
 - Dead simple GUI (not even a window manager)
 - Reparent dialogs and buttons when necessary
 - Attendantless installation, less error prone

Securing against known malicious content and the users!

Trimmed OS

- Trimmed OS footprint
 - Remove kernel modules, almost all of them
 - {sound, net, input, drivers}
 - Remove unneeded packages (or subcomponents)
 - Eliminate almost all services
- Have RPM dependencies you need to break?
 - “--nodeps” or just remove the files by hand
 - ssshhh – don’t tell RPM
 - Only safe if you “own” the whole system

Partitioning

- Custom partition layout – mostly read-only
- readonlyroot kernel command line option is broke
 - Implemented via rc.sysinit trickery
 - No SELinux support & other things break (GDM)
 - Led to patching of rc.sysinit
 - empty rwtab (defines rw mounts)
 - disable mounting of stateless partition (which fails)
 - create our own implementation for rwtab & stateless
-

Partitioning pt. 2

- unionfs would have made our lives easier but
 - not upstream
 - xattr support was broken
 - COW LVM an alternative
- Resorted to
 - bind mount magic for persistence & rw
 - /etc/{passwd,group,...} & /etc/{adjtime,blkid} respectively
 - tmpfs when read/write absolutely necessary
 - context-based mounts for SELinux
 - eg mount -o context=system_u:object_r:xserver_log_t
 - doesn't work with bind mounts, so add labels for those
-

SELinux

- SELinux on RHEL 5 Host and RHEL 4 Guest
- Custom policy supporting pipeline and virtualization
- Custom udev rules support per-USB port labels
 - Clean, dirty, forensic, etc.
 - Differing requirements, different label + policy
 - ie writer can access clean port but not dirty port
-
-

SELinux pt. 2

- Leveraging tmpfs to support readonlyroot
 - Context-based mounting
 - rootcontext & context
 - rootcontext requires relabeling early in init
- Customized SELinux boolean configuration
- Clean room policy analysis
 - Kiosk developers not involved in policy analysis
-
-

Random Security Thoughts

- Users like to pound head on keyboard
 - Custom keyboard maps loaded
 - GDM & vncviewer caused lotsa problems
 - Best effort to trap interrupts in shell scripts
 - Attendantless installation, less error prone
- Disable all unused features in BIOS
 - Do not distribute BIOS password ;)
- Disable “Interactive” boot prompt
 -



Other Random Security Thoughts

- Lockdown
 - GRUB via randomized password
 - securettys
 - root account (randomized password, require root auth for single user mode)
- Stateless virtualization (forked snapshots)

**Great, you did stuff.
What about Open source?**

Open Source

- OSS for rapid integration
- Examples
 - Complete system configuration via Kickstart
 - Easily “trim” system footprint
 - Ideal for 1-off solutions
 - Use what you can
 - Replace/remove what you can't use
 - Try replacing HAL in 7-days on Windows
 -

Open Source pt. 2

- OSS provided for extensibility
- Examples
 - Extend functionality through fine-grain configuration
 - Adding modular components is a cakewalk
 - Custom KS modules, custom firstboot modules
 - Don't forgot 3rd party kernel modules (double edged sword?)
 - Open source and extensible security + policy
 - OSS friendly vendors
 - Easily to extend the platform with 3rd party components
 - Yes, provide closed source products, but OSS *friendly*
 -

QUESTIONS?

**TELL US WHAT YOU THINK:
[REDHAT.COM/SUMMIT-SURVEY](https://redhat.com/summit-survey)**