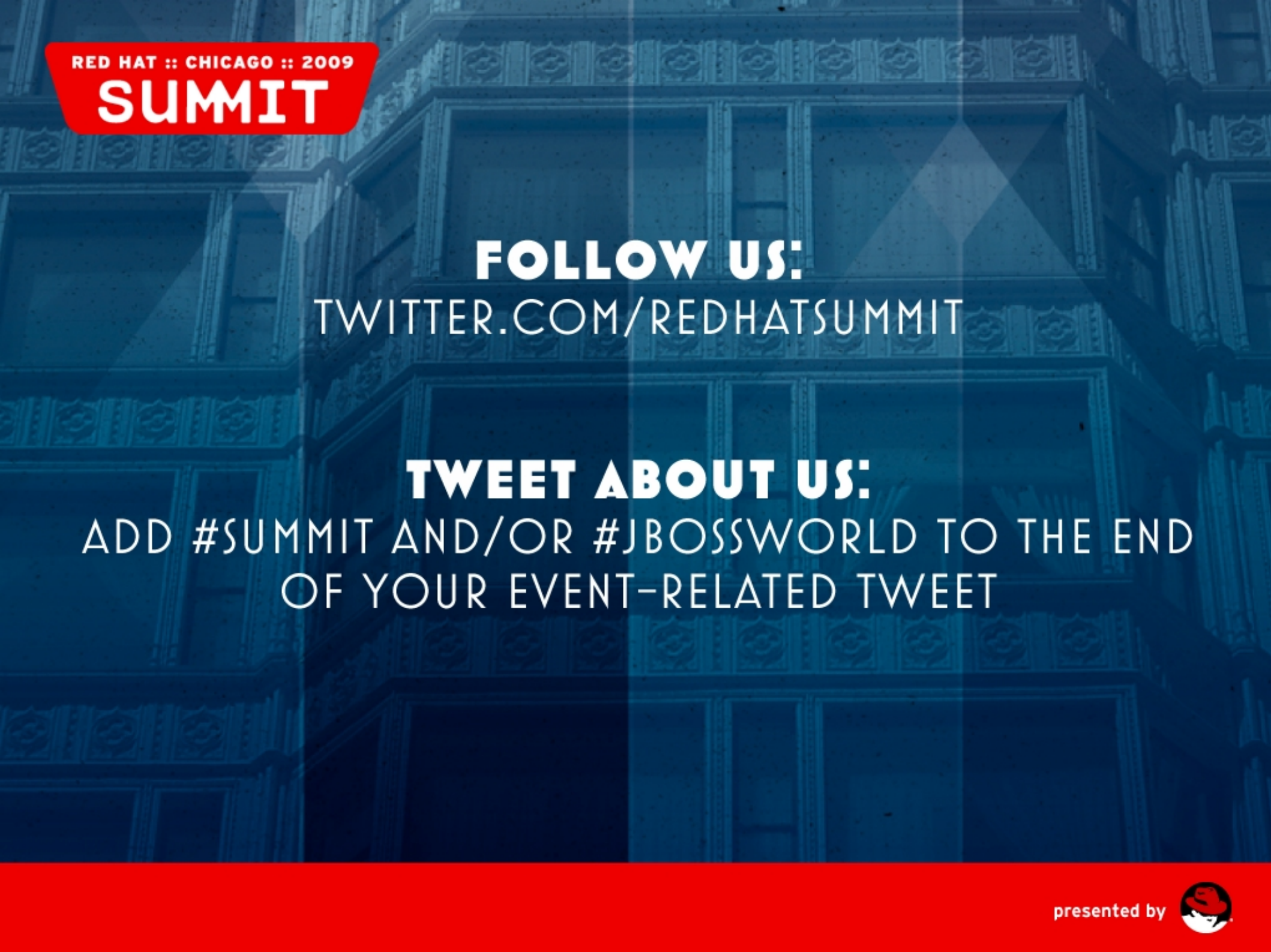
RED HAT :: CHICAGO :: 2009
SUMMIT
FOLLOW US:
TWITTER.COM/REDHATSUMMIT
TWEET ABOUT US:
ADD #SUMMIT AND/OR #JBOSSWORLD TO THE END
OF YOUR EVENT-RELATED TWEET
presented by

RED HAT :: CHICAGO :: 2009
SUMMIT

Unmatched Security Is Manageable

Presenter

Spencer Shimko
Senior Security Engineer, Tresys Technology
Date

Agenda

- Secure configuration at deployment time
  - Creating a secure state from the get go
  - Methods and tools for deployment
- Monitoring systems after deployment
  - How has my security state changed over time?
- Updating systems at run-time
  - Management of security configuration
  - Management of system updates

Typical Secure Deployment Technique - Legacy Model

- Start with a general-purpose RHEL install
- Identify “lock-down” configuration
  1. Deploy a system in test environment
     - Turn off SELinux (*cough cough*)
  2. Configure by hand
  3. Roll custom RPMs?
- Port these config changes to kickstart
- Create/modify package repository
- Rebuild ISO or add to PXE environment
- Rinse-wash-repeat until “correct”
-

Problems with Legacy Model

- Do you meet your compliance requirements?
  - PCI, HIPPA, STIG (guidelines), DCID 6/3, NIST 1253
- Is there a gap between your intentions and reality?
  - Gap between requirements and OS functions
- What level of assurance do you have?
  - Evidence to support config == reality
- How will you perform updates as state changes?
-

A Modest Proposal for Deployment

- Start with a vetted, *security-focused* platform
- Avoid general-purpose OS w/ larger gap
- Reduce error prone hand configuration
- SELinux should be enabled and enforcing
- Many security requirements should be addressed
  - Gap analysis & assurance evidence for base platform
- Admins should be able to focus on
  - Application configuration
  - Environment requirements
-

Certifiable Linux Integration Platform (CLIP)

- CLIP Goals (simply stated)
  1. Mapping between security reqs and OS functions
  2. Configuration of RHEL that meets security reqs
  3. Evidence to support 1 & 2 (aka documentation)
-
- But CLIP is much, much more…
  -

CLIP pt. 2

- Provides secure platform focused on security
-
- Tight, trimmed down RHEL (RHEL 4 & 5)
-
- Remains useful in general-purpose applications
  - Not just for the government!
-

CLIP pt. 3

- Common security tools automagically utilized
-
- SELinux enabled, enforcing, and analyzed
-
- DAC, iptables, auditing, caps, role separation etc
-
- Tailor to suit tastes Areas of security addressed
-
- Confidentiality, integrity, availability, accountability

CLIP pt. 4

- Implements recommendations and guidance to meet requirements
-
- Gap analysis between RHEL and security reqs performed
  - Gap addressed
    - Configuration, custom packages, custom policy
-
-

CLIP pt. 5

- Mapping created (security reqs -> implementation)
  - Evidence available
  -
- Easy to deploy
  - PXE boot (ala Cobbler), ISO (revisor)
  -
- Utilizes Puppet for configuration
  - Easily extended
  - Easily re-apply configuration at run-time
  -

CLIP pt. 6

- Docs used for reference (randomized jargon)
  - DCID 6/3, NIST 1253 v4, DoD 8500.2, NIST 800-53, STIG
  - Yeah, I know, those are memorable/meaningful names
  -
- Completely open source!

OK, We've deployed securely. Now what?

System Monitoring

- How do we know things are still “all good”?
-
- auditd & syslog report events, not status
  - Necessary but not sufficient
  -
- Need to know if system state is sane
-
- Configs can change with time

System Monitoring pt. 2

- Systems need to be patched or updated
- How can I check CVEs against my systems?
- Are the systems even vulnerable to said exposure?
- Do the patches violate my security reqs?

Security Content Automation Protocol (SCAP)

- High-level specification for expressing security guidance & information
- Two languages
  - XCCDF
    - Express “security guidance”, benchmarking and doc generator
    - Think checklists for opting into best-practices tests
  - OVAL – XML schemas for
    - Representing config info, executing analysis, reports results
    - Think “CVE” testing
    -

Open Vulnerability Assessment Language (OVAL)

- Simply an XML Schema used to express tests
  - against vulnerabilities
    - eg CVEs
  - against known mis-configuration
    - eg world-readable shadow, open mail relay
- Open standard
- Lotsa “content”
  - Content tests specific cases of vulnerabilities & exposures

OVAL pt. 2

- NIST maintains repo of content
  - Automate compliance activities
  - Check for app misconfigurations (general)
  - Check for software flaws (general)
  -
- Red Hat maintains OVAL content for RHSAs
  - Run content on system, know if RHSAs apply to you
  -

OVAL pt.3 - So?

- Interpreting OVAL
  - Requires OVAL content + interpreter (OVALDI)
  - Gives you a report of your security state
  - Does not make changes
- Value
  - It can be automated
  - You have evidence of you security state
  - You can react appropriately to this evidence
  - It is an growing industry standard

Things aren’t “all good.” Respond?

Run-Time Security Config & Updates

- Up and coming area, but some thoughts…
-
- Combine Puppet and OVAL capabilities
  - Map Puppet configuration into OVAL tests
  - Tie OVAL results back into Puppet
-
  -
  -

Run-Time Security Config & Updates pt. 2

- Query homogenous Linux environments
  - Later heterogenous environments
    - OVAL not necessarily OS specific
    -
- Provide centralized interface
  - For reviewing OVAL results
  - Eventually… react to results
  -
  -

Run-Time Security Config & Updates pt. 3

- Puppet configs
  - Can be centralized
  - Can be used to re-apply configs at run-time
  -
- OVAL
  - Content needs to be centralized (in intranet sense)
  - Needs centralized reporting and presentation
  -
- Need to tie OVAL content to Puppet configs

(Envisioned) Order of Operations

1. Deploy using CLIP as base
2. Run OVAL tests
3. Collect OVAL results & collate
4. Update systems as necessary
   - Don’t update unless vulnerable
     - eg, mail servers w/out apache don’t need Apache updates
   - CVEs, etc
5. Re-run Puppet
   - Ensures system state still sane after updates

Demonstration

Random Linkage

- http://oss.tresys.com/projects/clip
- http://reductivelabs.com/products/puppet/
- http://scap.nist.gov/
- http://www.redhat.com/oval
- http://oval.mitre.org/

QUESTIONS?

TELL US WHAT YOU THINK:
REDHAT.COM/SUMMIT-SURVEY