Understanding Security Implications of Using Containers in the Cloud

Byungchul Tak

Kyungpook National University, Daegu, Korea

Canturk Isci, Sastry Duri, Nilton Bila, Shripad Nadgowda, James Doran

IBM TJ Watson Research Center, Yorktown Heights, NY, USA

Container Cloud

IBM **Bluemix**
Container Service

Google Container Engine

Azure Container Service

Oracle Container Cloud

Amazon EC2
Container Service

CARINA™
BY RACKSPACE

- Why container cloud?
  - Speed
    - rapid deployment
    - faster time-to-market
  - Convenience, Simplicity
  - Portability
  - Resource efficiency

- Challenges
  - **Security**
  - Robustness/Stability
    - isolation guarantee
  - Management
    - Increased deployment density
    - Monitoring becomes more complex

Goal of This Work

- Many excellent container scanning tools available
  - IBM Vulnerability Advisor, Twistlock, Amazon inspector, Docker security scanning, Aqua, CoreOS Clair, OpenSCAP …
- Goal

To **share our findings** of real-world container cloud in security aspect

To promote discussion and comparison with other production clouds

- To start understanding what the real cloud looks like in terms of cloud security
  - What does the security posture look like?
  - How do people use containers?
    - As regular VMs?
    - As immutable objects?

…

Non-goal: introducing new container/image scanning tool

Compliance Rules

Compliance Rules
Selected from Internal Rules
+ Newly Added rules

Password Restrictions	• Maximum password age must be set to 90 days • Minimum password length must be 8. • Reuse of password must be restricted to eight
File System Integrity	• Read/write access of ~root/.netrc only by root • Permission setting of /var/log for other must be r-x or more restrictive. • Syslog file permissions must be set: rwx r-x r-x (or more restrictive) • /var/log/faillog must exist for all systems not using pam_tally2.so
remote access	• rsh server must not be installed
SSH-related Rules	• SSH must not be installed • SSH password authentication must be disabled • Password must not be weak

Characteristics of Analyzed Data

- Data Collection Period
  - 2016 January to October
  - Image scan data
    - 2016 Jan to Oct
  - Live container scanning data
    - two weeks period in Oct, 2016
- From two development sites
  - Referred to as Site A and Site B in this talk

Container Image Scanning

- Need for the image scanning
  - Unknown image pedigree – lurking vulnerabilities
  - Even with modification history (e.g. docker history) available, security implication unclear
  - Vulnerability Amplification Effect – unforeseen synergy of independent updates

- Sample Image Scan Summary

Top 5 Compliance Violations

Rank	Compliance Rule Description
1	Minimum days that must elapse between user-initiated password changes should be 1
2	Minimum password length must be 8
3	Maximum password age must be set to 90 days
4	SSH server must not have been installed
5	SSH password authentication should not be enabled

Average Vulnerabilities Per Image

However, our study suggests that **simple scan and report is not enough** to understand the root cause of vulnerabilities (or non-compliances)

Case Study

- Highly vulnerable container images
  - We focus on SSH-related rules from the set of compliance rules

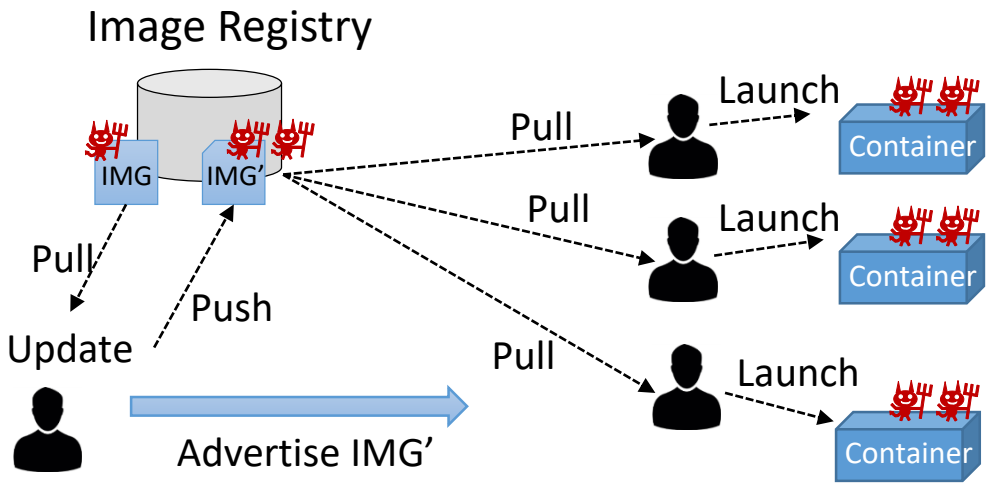
[9A] SSH server must not be installed
[9F] SSH password-based authentication must be disabled
[9G] Password must not be weak

  - If all 3 are violated, it is considered ‘highly vulnerable’

registry.private.net/17_??ma?o/**myappsr**:latest
registry.private.net/ak_??me?pace/**myappsrv**:latest
registry.private.net/all??na?espace/**myappsrv**:latest
registry.private.net/am_??la?i/**myappsrv**:latest
registry.private.net/an_??ue/myibmliberty:latest
registry.private.net/ch_??_1?01_dev/**myappsrv**:latest
registry.private.net/ck_??g/**myappsrv**:latest
registry.private.net/co_??oy?ham_app/**myappsrv**:latest
registry.private.net/de_??on/**myappsrv**:latest
registry.private.net/do_??r_?ode00/**myappsrv**:latest
registry.private.net/do_??r_?ode/**myappsrv**:latest
registry.private.net/do_??on?ain/**myappsrv**:latest
registry.private.net/dy_??cl?ud/**myappsrv**:latest
registry.private.net/es_??nd?ox_01/grafana:latest
registry.private.net/ex_??am?space/**myappsrv**:latest
registry.private.net/gr_??it?/**myappsrv**:latest
registry.private.net/hs_??bm?container/**myappsrv**:latest
registry.private.net/hu_??ev?dev/**myappsrv**:latest
registry.private.net/ja_??19?52/**myappsrv**:latest
registry.private.net/jh_??am?space/**myappsrv**:latest
registry.private.net/jo_??am?space/**myappsrv**:latest

Case Explained

- ‘myappsrv’ image
  - Searched for this image in the Dockerhub
  - Inspecting the image contents reveals that
    - ‘docker inspect’ shows postgres start-up command as the entry
    - opened ports: 22 (ssh), 5432 (postgres port), 7276, 7268, 9080, 9443 (websphere ports)
    - list of installed packages shows many postgres packages



- Our Scanning Tool Reported that:
  - Is SSH is installed? **Yes**
  - Is SSH password access enabled? **Yes**

```
In /etc/ssh/sshd_config
# Change to no to disable tunnelled clear text passwords
#PasswordAuthentication yes
```

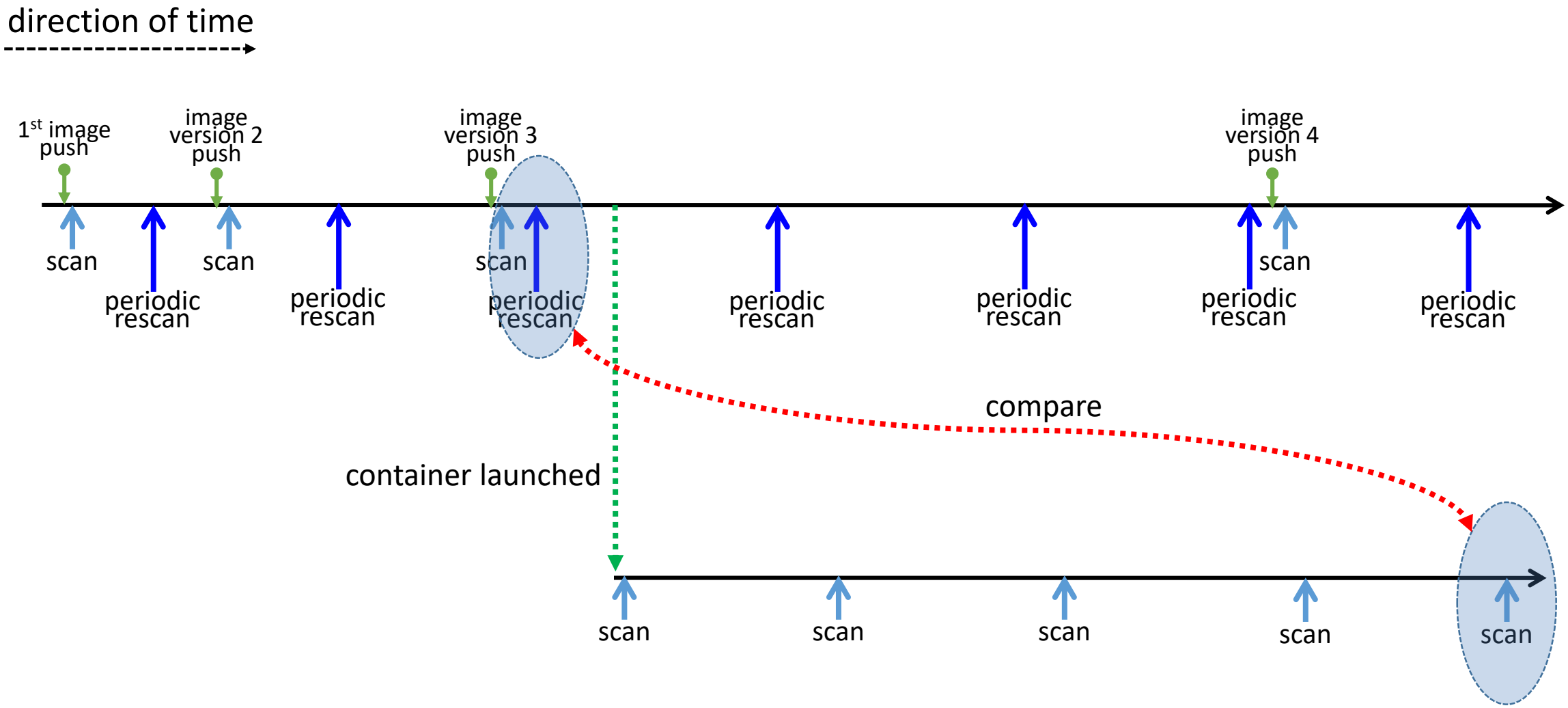
  - Is there any ID with default password? **Yes**

Overall aggregate analysis may be needed to understand the root cause of vulnerabilities (or non-compliances)

Why Scanning Live Containers is Critical?

- Image security is only one part of overall container security
  - Having secure image is great, but …
  - What if container instance gets directly modified?
    - Image scanning results become invalid!
  - DevOps builds upon the assumption of immutability
    - Updates should be applied to the source image and the instance should be launched again
- Question
  - Does security posture change after images instantiate to containers?
  - That is, are there any **drifts**?
  - If so, what does it look like?

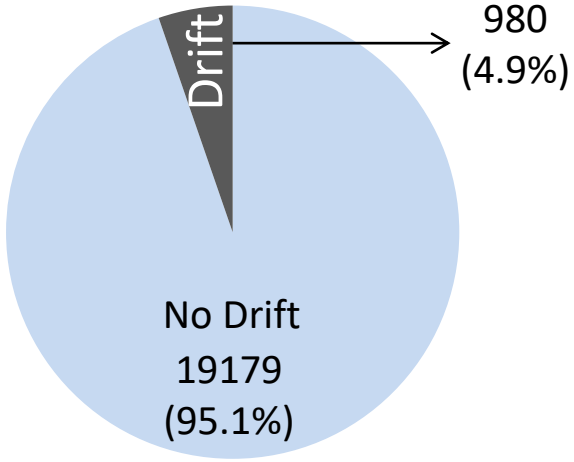
Definition of Drifts

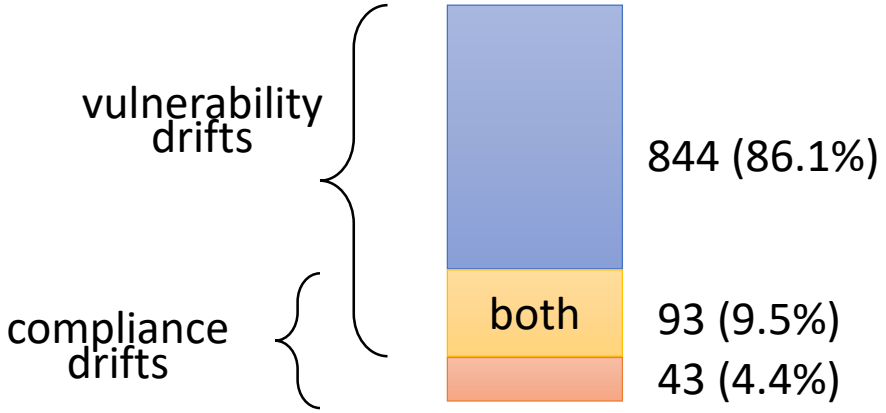


Drift Findings

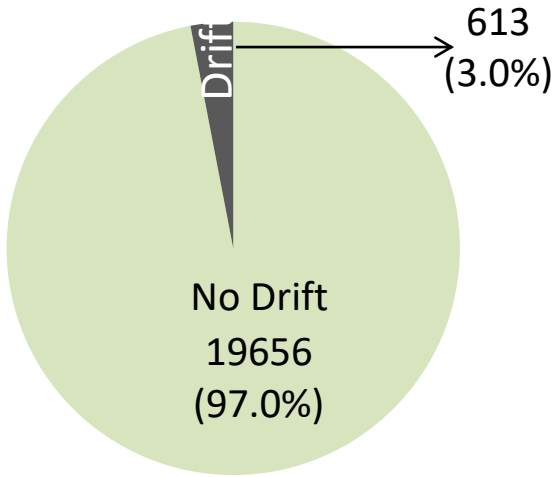
Site-specific differences exists. Absolute # of drift does not imply higher/lower security level.

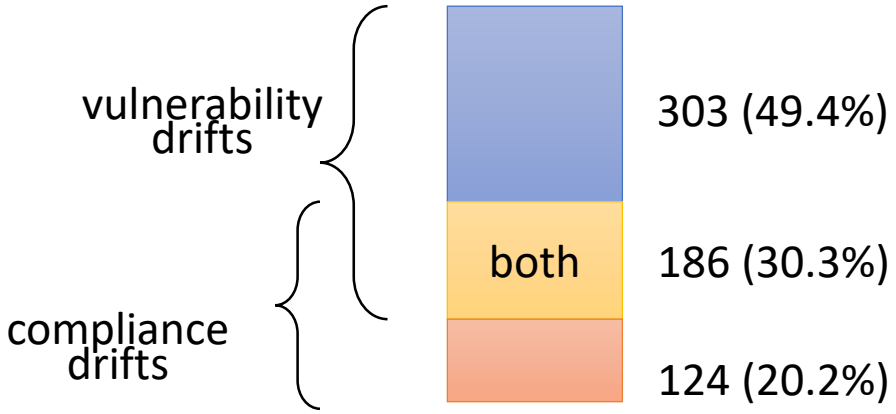
Site A





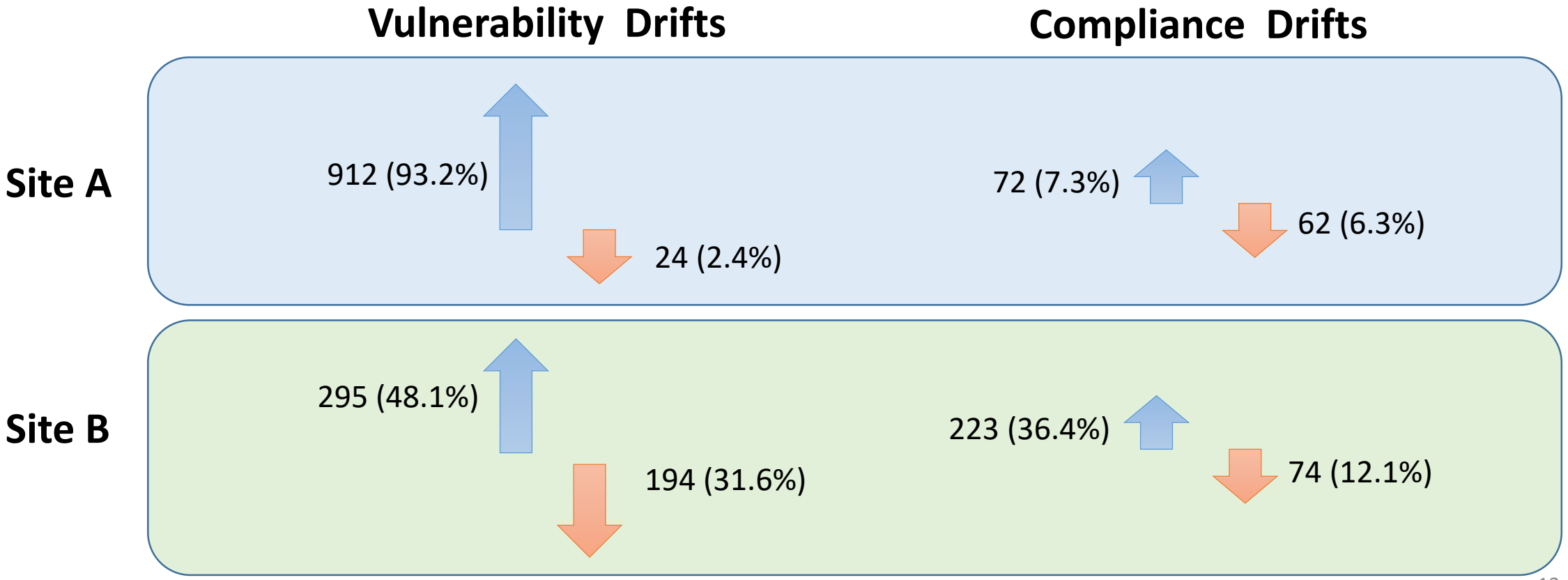
Site B





Drift Findings - continued

Drifts are not always in the increasing direction.
In-place updates does happen in both benign and disruptive ways.



Drift Findings - SSH

- Recall 3 SSH-related compliance rules

[9A] SSH server must not be installed
[9F] SSH password-based authentication must be disabled
[9G] Password must not be weak

		Site A		Site B	
SSH Vulnerability Increased	No SSH, but password becomes weak	1	1.3%	1	1.2%
	SSH gets installed	31	39.2%	19	23.5%
	SSH gets installed with weak password ID	3	3.8%		
	Password becomes weak	1	1.3%		
SSH Vulnerability Decreased	Password becomes strong	3	26.6%	26	32.1%
	Password-based authentication gets disabled	13	16.5%	2	2.5%
	No SSH, but password becomes strong	1	1.3%		
	SSH gets removed	8	10.1%	33	40.7%

Why Drifts Happen?

- Benign drifts
  - Newly added definitions of vulnerable packages
  - Updated/added compliance rules
  - Implementation changes
  - Introduction of bugs
  - System Anomalies
- Disruptive drifts
  - Update via Remote access: SSH login or ‘docker exec’
  - Automated S/W update
  - Software configured at Runtime

Conclusion

- First look at the security postures of real-world container cloud
- Achieving secure container cloud requires
  - Automated image scanning
    - Vulnerability amplification
    - In addition, aggregate analysis needed to determine the true source of vulnerabilities
  - Live container scanning
    - Drifts do exist in the real world: about 5%
    - Different type of drifts
      - Increase/decrease
      - Benign/disruptive
        - Majority are benign drifts of vulnerable package increase
        - But, disruptive manual in-place updates do exist
          - could lead to serious problem
    - Must be accompanied with static image scanning