GovOps Web Security

Mike Nescot

Web Operations and Security Manager, JBS International

Agenda

- Government IT Security Policy and Focus
- DevOps Revolution and Promise
- Open Source: Acceptance in Government
- Version Control: Git, GitHub, Gerrit
- Configuration Management: Puppet, Ansible
- Virtualization: Vagrant, Docker
- Continuous Integration: Jenkins
- Monitoring: OSSIM, Prometheus
- Communication: Slack, Social Networking
- Cloud: Amazon and Azure
- Future DevOps and Security

Government IT Security

Security Controls: FISMA Standard, SP-53 Rev 4

NIST Special Publication 800-53
Revision 4

Security and Privacy Controls for Federal Information Systems and Organizations

JOINT TASK FORCE
TRANSFORMATION INITIATIVE

http://dx.doi.org/10.6028/NIST.SP.800-53r4

NIST
National Institute of Standards and Technology
U.S. Department of Commerce

SP800-53 Rev 4
Security Controls: 18 Families

- Access Control
- Awareness & Training
- Audit & Accountability
- Security Assessment & Authorization
- Configuration Management
- Contingency Planning
- Identification & Authorization
- Incident Response
- Maintenance

SP800-53 Rev 4
Security Controls: 18 Families (cont.)

- Media Protection
- Physical and Environmental Protection
- Planning
- Personnel Security
- Risk Assessment
- System and Services Acquisition
- System & Communications Protection
- System & Information Integrity
- Program Management

SP800-53 Rev 4
Privacy Controls: 8 Families (FEA)

- Authority & Purpose
- Accountability, Audit, & Risk Management
- Data Quality & Integrity
- Data Minimization & Retention
- Individual Participation & Redress
- Security
- Transparency
- Use Limitation

New Focus

- Insider threat
- Application security (Build it Right & Continuously Monitor)
- Supply chain risk
- Security assurance and trustworthy systems
- Mobile and cloud computing technologies
- Advanced persistent threat
- Tailoring guidance and overlays
- Privacy

New Focus

- Simplify
- Specialize
- Integrate
- Strategic Risk-Based Approach

Anatomy of a Control

AC-2 ACCOUNT MANAGEMENT

Control: The organization:

a. Identifies and selects the following types of information system accounts to support organizational missions/business functions: [*Assignment: organization-defined information system account types*];

b. Assigns account managers for information system accounts;

c. Establishes conditions for group and role membership;

d. Specifies authorized users of the information system, group and role membership, and access authorizations (i.e., privileges) and other attributes (as required) for each account;

e. Requires approvals by [*Assignment: organization-defined personnel or roles*] for requests to create information system accounts;

f. Creates, enables, modifies, disables, and removes information system accounts in accordance with [*Assignment: organization-defined procedures or conditions*];

g. Monitors the use of, information system accounts;

Anatomy of a Control

h. Notifies account managers:

1. When accounts are no longer required;
2. When users are terminated or transferred; and
3. When individual information system usage or need-to-know changes;

i. Authorizes access to the information system based on:

1. A valid access authorization;
2. Intended system usage; and
3. Other attributes as required by the organization or associated missions/business functions;

j. Reviews accounts for compliance with account management requirements [*Assignment: organization-defined frequency*]; and

k. Establishes a process for reissuing shared/group account credentials (if deployed) when individuals are removed from the group.

Supplemental Guidance: Information system account types include, for example, individual, shared, group, system, guest/anonymous, emergency, developer/manufacturer/vendor, temporary, and service. Some of the account management requirements listed above can be implemented by organizational information systems. The identification of authorized users of the information system and the specification of access privileges reflects the requirements in other security controls in the security plan. Users requiring administrative privileges on information system accounts receive additional scrutiny by appropriate organizational personnel (e.g., system owner, mission/business owner, or chief information security officer) responsible for approving such accounts and privileged access. Organizations may choose to define access privileges or other attributes by account, by type of account, or a combination of both. Other attributes required for authorizing access include, for example, restrictions on time-of-day, day-of-week, and point-of-origin. In defining other account attributes, organizations consider system-related requirements (e.g., scheduled maintenance, system upgrades) and mission/business requirements, (e.g., time zone differences, customer requirements, remote access to support travel requirements). Failure to consider these factors could affect information system availability. Temporary and emergency accounts are accounts intended for short-term use. Organizations establish temporary accounts as a part of normal account activation procedures when there is a need for short-term accounts without the demand for immediacy in account activation. Organizations establish emergency accounts in response to crisis situations and with the need for rapid account activation. Therefore, emergency account activation may bypass normal account authorization processes. Emergency and temporary accounts are not to be confused with infrequently used accounts (e.g., local logon accounts used for special tasks defined by organizations or when network resources are unavailable). Such accounts remain available and are not subject to automatic disabling or removal dates. Conditions for disabling or deactivating accounts include, for example: (i) when shared/group, emergency, or temporary accounts are no longer required; or (ii) when individuals are transferred or terminated. Some types of information system accounts may require specialized training. Related controls: AC-3, AC-4, AC-5, AC-6, AC-10, AC-17, AC-19, AC-20, AU-9, IA-2, IA-4, IA-5, IA-8, CM-5, CM-6, CM-11, MA-3, MA-4, MA-5, PL-4, SC-13.

Control Enhancements:

(1) *ACCOUNT MANAGEMENT | AUTOMATED SYSTEM ACCOUNT MANAGEMENT*

The organization employs automated mechanisms to support the management of information system accounts.

Supplemental Guidance: The use of automated mechanisms can include, for example: using email or text messaging to automatically notify account managers when users are terminated or transferred; using the information system to monitor account usage; and using telephonic notification to report atypical system account usage.

Anatomy of a Control

(2) *ACCOUNT MANAGEMENT | REMOVAL OF TEMPORARY / EMERGENCY ACCOUNTS*

The information system automatically [*Selection: removes; disables*] temporary and emergency accounts after [*Assignment: organization-defined time period for each type of account*].

Supplemental Guidance: This control enhancement requires the removal of both temporary and emergency accounts automatically after a predefined period of time has elapsed, rather than at the convenience of the systems administrator.

(3) *ACCOUNT MANAGEMENT | DISABLE INACTIVE ACCOUNTS*

The information system automatically disables inactive accounts after [*Assignment: organization-defined time period*].

(4) *ACCOUNT MANAGEMENT | AUTOMATED AUDIT ACTIONS*

The information system automatically audits account creation, modification, enabling, disabling, and removal actions, and notifies [*Assignment: organization-defined personnel or roles*].

Supplemental Guidance: Related controls: AU-2, AU-12.

(5) *ACCOUNT MANAGEMENT | INACTIVITY LOGOUT*

The organization requires that users log out when [*Assignment: organization-defined time-period of expected inactivity or description of when to log out*].

Supplemental Guidance: Related control: SC-23.

(6) *ACCOUNT MANAGEMENT | DYNAMIC PRIVILEGE MANAGEMENT*

The information system implements the following dynamic privilege management capabilities: [*Assignment: organization-defined list of dynamic privilege management capabilities*].

Supplemental Guidance: In contrast to conventional access control approaches which employ static information system accounts and predefined sets of user privileges, dynamic access control approaches (e.g., service-oriented architectures) rely on run time access control decisions facilitated by dynamic privilege management. While user identities may remain relatively constant over time, user privileges may change more frequently based on ongoing mission/business requirements and operational needs of organizations. Dynamic privilege management can include, for example, the immediate revocation of privileges from users, as opposed to requiring that users terminate and restart their sessions to reflect any changes in privileges. Dynamic privilege management can also refer to mechanisms that change the privileges of users based on dynamic rules as opposed to editing specific user profiles. This type of privilege management includes, for example, automatic adjustments of privileges if users are operating out of their normal work times, or if information systems are under duress or in emergency maintenance situations. This control enhancement also includes the ancillary effects of privilege changes, for example, the potential changes to encryption keys used for communications. Dynamic privilege management can support requirements for information system resiliency. Related control: AC-16.

(7) *ACCOUNT MANAGEMENT | ROLE-BASED SCHEMES*

The organization:

(a) Establishes and administers privileged user accounts in accordance with a role-based access scheme that organizes allowed information system access and privileges into roles;

(b) Monitors privileged role assignments; and

(c) Takes [*Assignment: organization-defined actions*] when privileged role assignments are no longer appropriate.

Supplemental Guidance: Privileged roles are organization-defined roles assigned to individuals that allow those individuals to perform certain security-relevant functions that ordinary users are not authorized to perform. These privileged roles include, for example, key management, account management, network and system administration, database administration, and web administration.

(8) *ACCOUNT MANAGEMENT | DYNAMIC ACCOUNT CREATION*

The information system creates [*Assignment: organization-defined information system accounts*] dynamically.

Supplemental Guidance: Dynamic approaches for creating information system accounts (e.g., as implemented within service-oriented architectures) rely on establishing accounts (identities) at

Anatomy of a Control

run time for entities that were previously unknown. Organizations plan for dynamic creation of information system accounts by establishing trust relationships and mechanisms with the appropriate authorities to validate related authorizations and privileges. Related control: AC-16.

(9) *ACCOUNT MANAGEMENT | RESTRICTIONS ON USE OF SHARED GROUPS / ACCOUNTS*

The organization only permits the use of shared/group accounts that meet [*Assignment: organization-defined conditions for establishing shared/group accounts*].

(10) *ACCOUNT MANAGEMENT | SHARED / GROUP ACCOUNT CREDENTIAL TERMINATION*

The information system terminates shared/group account credentials when members leave the group.

(11) *ACCOUNT MANAGEMENT | USAGE CONDITIONS*

The information system enforces [*Assignment: organization-defined circumstances and/or usage conditions*] for [*Assignment: organization-defined information system accounts*].

Supplemental Guidance: Organizations can describe the specific conditions or circumstances under which information system accounts can be used, for example, by restricting usage to certain days of the week, time of day, or specific durations of time.

(12) *ACCOUNT MANAGEMENT | ACCOUNT MONITORING / ATYPICAL USAGE*

The organization:

(a) Monitors information system accounts for [*Assignment: organization-defined atypical use*]; and

(b) Reports atypical usage of information system accounts to [*Assignment: organization-defined personnel or roles*].

Supplemental Guidance: Atypical usage includes, for example, accessing information systems at certain times of the day and from locations that are not consistent with the normal usage patterns of individuals working in organizations. Related control: CA-7.

(13) *ACCOUNT MANAGEMENT | DISABLE ACCOUNTS FOR HIGH-RISK INDIVIDUALS*

The organization disables accounts of users posing a significant risk within [*Assignment: organization-defined time period*] of discovery of the risk.

Supplemental Guidance: Users posing a significant risk to organizations include individuals for whom reliable evidence or intelligence indicates either the intention to use authorized access to information systems to cause harm or through whom adversaries will cause harm. Harm includes potential adverse impacts to organizational operations and assets, individuals, other organizations, or the Nation. Close coordination between authorizing officials, information system administrators, and human resource managers is essential in order for timely execution of this control enhancement. Related control: PS-4.

References: None.

Priority and Baseline Allocation:

P1	**LOW** AC-2	**MOD** AC-2 (1) (2) (3) (4)	**HIGH** AC-2 (1) (2) (3) (4) (5) (12) (13)

Anatomy of a Control

- Control count: from 198 to 267, or 600 to 850
- More tailoring guidance, overlays, focus on assurance controls, strategic, privacy
- Overlays

SANS Top 20

- Inventory of Authorized and Unauthorized Devices
- Inventory of Authorized and Unauthorized Software
- Secure Configurations for Hardware & Software on Laptops, Workstations, & Servers
- Continuous Vulnerability Assessment and Remediation
- Malware Defense
- Application Software Security
- Wireless Device Control
- Data Recovery Capability
- Security Skills Assessment & Training
- Secure Configurations for Firewalls, Routers, & Switches

SANS Top 20 (cont)

- Limitation & Control of Network Ports, Protocols, & Services
- Controlled Use of Administrative Privileges
- Boundary Defense
- Maintenance, Monitoring, & Analysis of Audit Logs
- Controlled Access Based on Need to Know
- Account Monitoring & Control
- Data Loss Prevention
- Incident Response & Management
- Secure Network Engineering
- Penetration Testing & Team Exercises

SANS Top 20

The five critical tenets of an effective cyber defense system as reflected in the Critical Controls are:

- Offense informs defense: Use knowledge of actual attacks for defense
- Prioritization: Invest first in controls that will provide the greatest risk reduction and protection
- Metrics: Establish common metrics to measure effectiveness
- Continuous monitoring: Test and validate the effectiveness of current security measures.
- Automation: Automate defenses, achieve reliable, scalable, and continuous measurements

State of Required Security Controls

- Newly updated: NIST SP-53 Rev 4
- SANS Top 20 Controls
- Build it Right (SDLC), Continuous Monitoring
- 2011: NIST SP 800-137

Information Systems Continuous Monitoring (ISCM)

- Maintaining ongoing awareness of information security, vulnerabilities, and threats to support organizational risk management decisions.
- From compliance driven to data driven risk management

Conventional

- Hostile cyber attacks
- Natural disaster
- Structural failures
- Human errors of omission or commission
- Strong Foundation

Advanced Persistent Threat

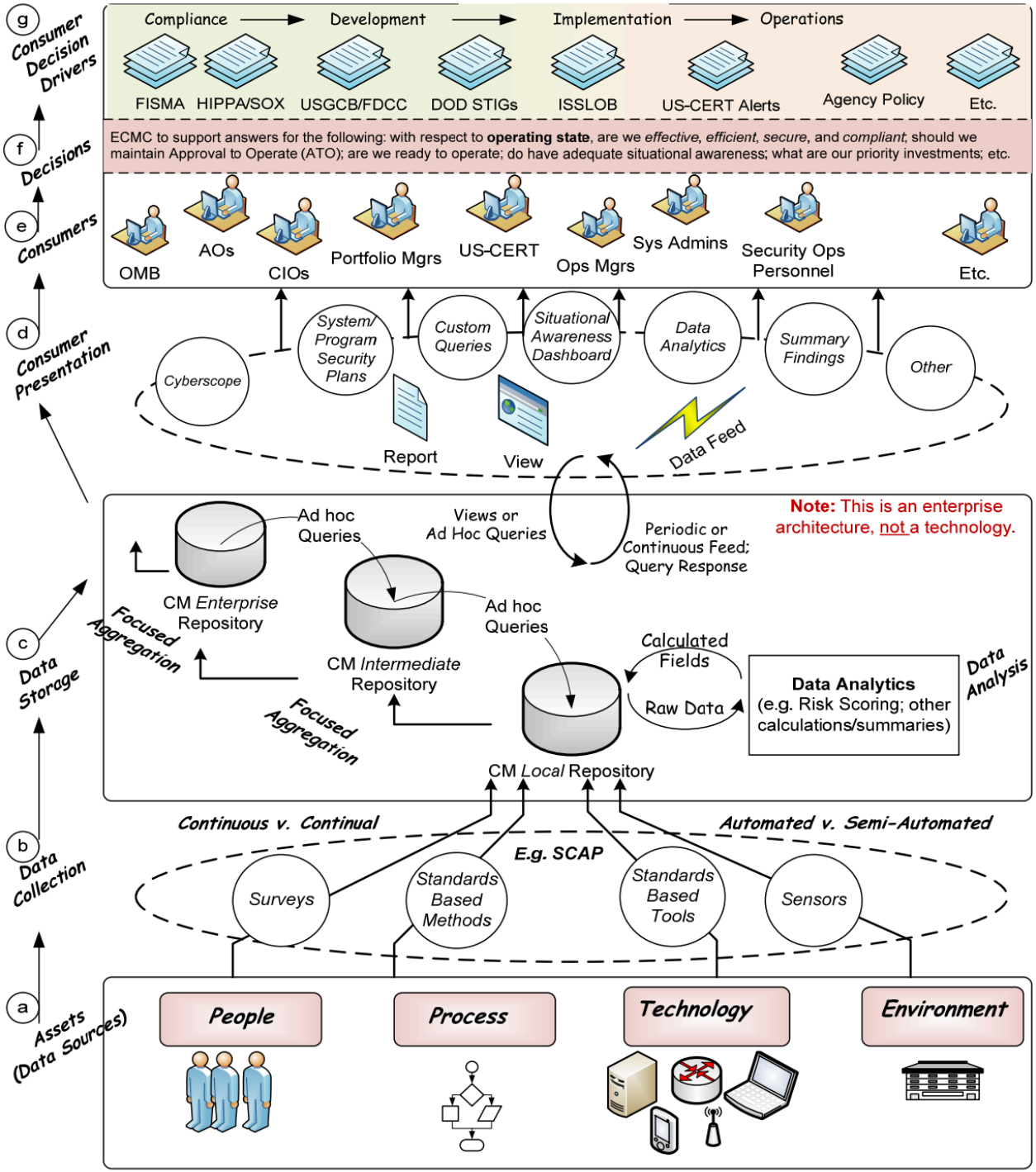
- Significant expertise
- Multiple attack vectors
- Establishes footholds

Continuous Asset Evaluation, Situational Awareness, and Risk Scoring (CAESARS)

- Reference Architecture: Security Automation Standards
- Data Sources
- Data Collection
- Data Storage & Analysis
- Consumer Presentation
- Decisions

CAESARS Subsystems

- Sensor (Assets, devices, servers, devices, appliances)
- Database Sub (repository of configuration and inventory baselines)
- Analysis/Scoring
- Presentation (variety of views, query capabilities)


g
Consumer Decision Drivers
Compliance
Development
Implementation
Operations
FISMA
HIPPA/SOX
USGCB/FDCC
DOD STIGs
ISSLOB
US-CERT Alerts
Agency Policy
Etc.
f
Decisions
ECMC to support answers for the following: with respect to operating state, are we effective, efficient, secure, and compliant; should we maintain Approval to Operate (ATO); are we ready to operate; do have adequate situational awareness; what are our priority investments; etc.
e
Consumers
OMB
AOs
CIOs
Portfolio Mgrs
US-CERT
Ops Mgrs
Sys Admins
Security Ops Personnel
Etc.
d
Consumer Presentation
Cyberscope
System/ Program Security Plans
Custom Queries
Situational Awareness Dashboard
Data Analytics
Summary Findings
Other
Report
View
Data Feed
c
Data Storage
Ad hoc Queries
Views or Ad Hoc Queries
Periodic or Continuous Feed; Query Response
Note: This is an enterprise architecture, not a technology.
CM Enterprise Repository
Focused Aggregation
CM Intermediate Repository
Ad hoc Queries
Focused Aggregation
Calculated Fields
Raw Data
Data Analytics
(e.g. Risk Scoring; other calculations/summaries)
Data Analysis
CM Local Repository
b
Data Collection
Continuous v. Continual
E.g. SCAP
Automated v. Semi-Automated
Surveys
Standards Based Methods
Standards Based Tools
Sensors
a
Assets (Data Sources)
People
Process
Technology
Environment

CAESARS

The end goal of CAESARS FE is to enable enterprise CM by presenting a technical reference model that allows organizations to aggregate collected data from across a diverse set of security tools, analyze that data, perform scoring, enable user queries, and provide overall situational awareness.

Security Capability

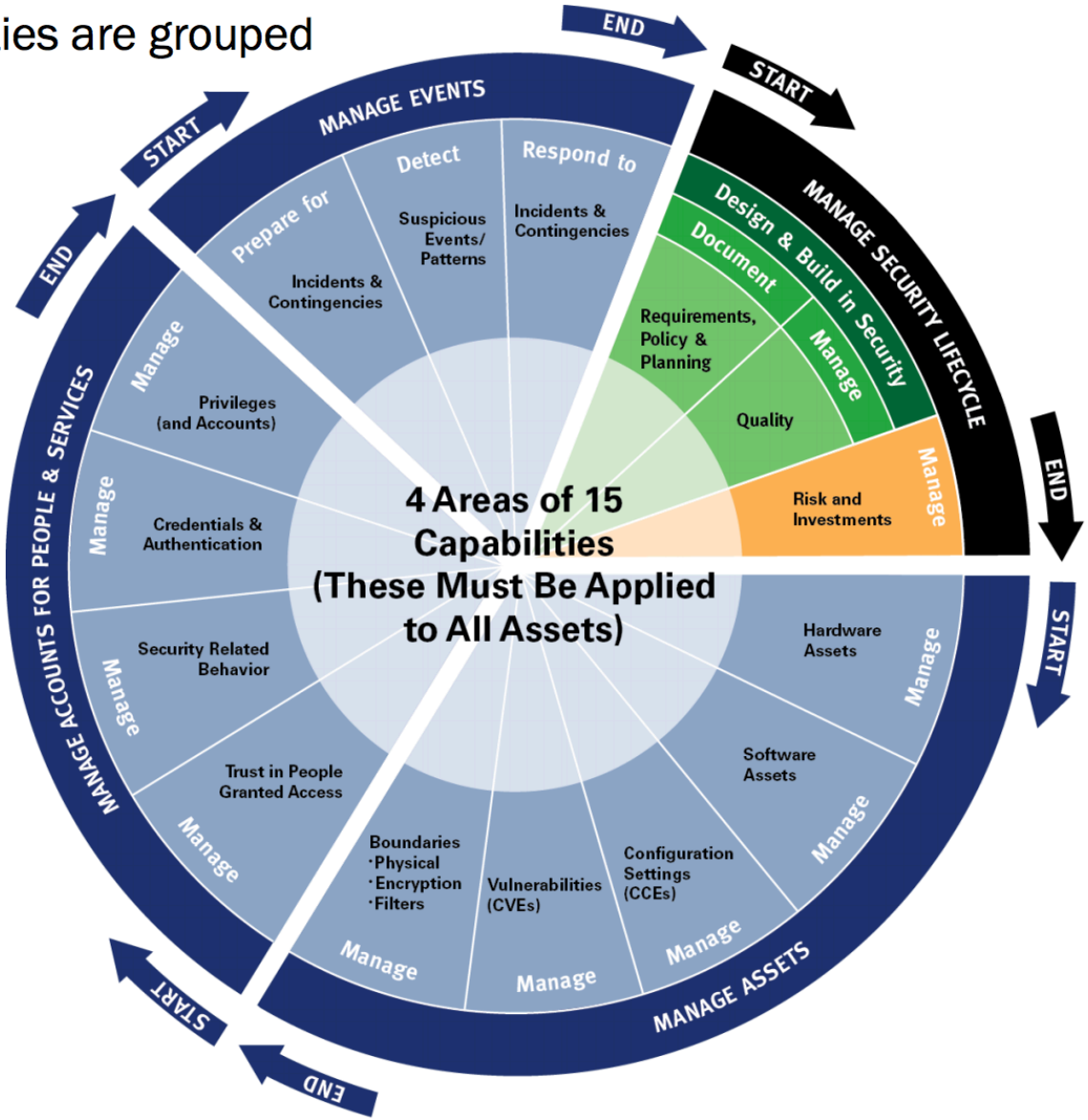
- A collection (set) of security controls that work together to achieve an overall security purpose. (NIST 800-53 Rev4, p. 21.)

- Improves/supports risk management:

  - The failure of multiple controls may not affect the overall security capability needed by an organization.
  - Moreover, employing security capabilities allows an organization to determine if the failure of a particular security control affects the overall capability needed for mission/ business protection.
  - Ultimately, authorization decisions (i.e., risk acceptance decisions) are made based on the degree to which the desired security capabilities have been effectively achieved and are meeting the security requirements defined by an organization.

CDM Capabilities: Capability Wheel

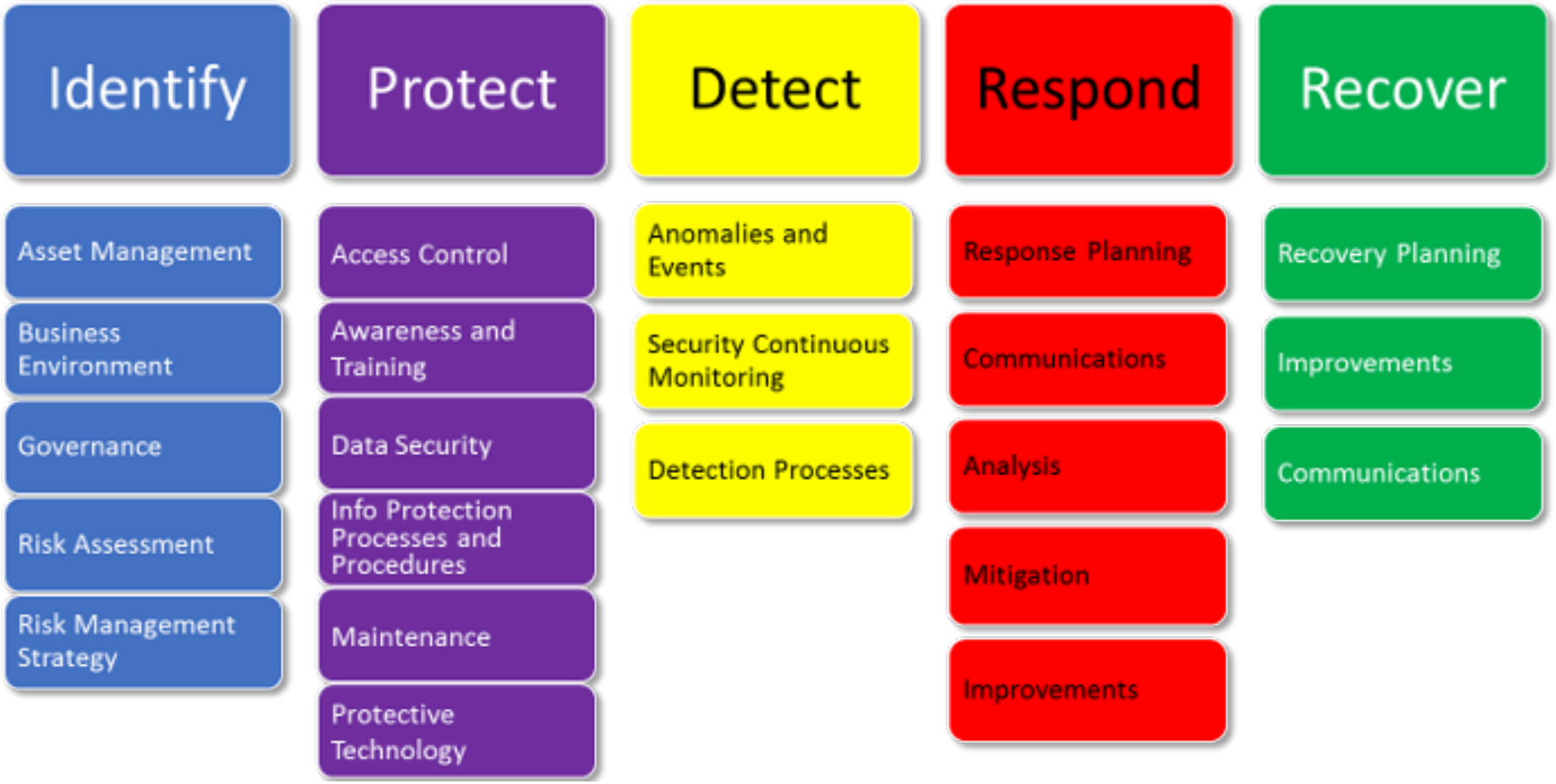
Identifies all CDM Capabilities
Related Capabilities are grouped into “Families”



Cybersecurity Framework

- Building from those standards, guidelines, and practices, the Framework provides a common
- taxonomy and mechanism for organizations to:
- 1) Describe their current cybersecurity posture;
- 2) Describe their target state for cybersecurity;
- 3) Identify and prioritize opportunities for improvement within the context of a
- continuous and repeatable process;
- 4) Assess progress toward the target state;
- 5) Communicate among internal and external stakeholders about cybersecurity risk.

NIST Cyber Security Framework



Cybersecurity Framework

Tier 1: Partial

Risk Management Process – Organizational cybersecurity risk management practices are not formalized, and risk is managed in an ad hoc and sometimes reactive manner. Prioritization of cybersecurity activities may not be directly informed by organizational risk objectives, the threat environment, or business/mission requirements.

- Integrated Risk Management Program – There is limited awareness of cybersecurity risk at the organizational level and an organization-wide approach to managing cybersecurity risk has not been established. The organization implements cybersecurity risk management on an irregular, case-by-case basis due to varied experience or information gained from outside sources. The organization may not have processes that enable cybersecurity information to be shared within the organization.
- External Participation – An organization may not have the processes in place to participate in coordination or collaboration with other entities.

Cybersecurity Framework

Tier 2: Risk Informed

- Risk Management Process – Risk management practices are approved by management but may not be established as organizational-wide policy. Prioritization of cybersecurity activities is directly informed by organizational risk objectives, the threat environment, or business/mission requirements.
- Integrated Risk Management Program – There is an awareness of cybersecurity risk at the organizational level but an organization-wide approach to managing cybersecurity risk has not been established. Risk-informed, management-approved processes and procedures are defined and implemented, and staff has adequate resources to perform their cybersecurity duties. Cybersecurity information is shared within the organization on an informal basis.
- External Participation – The organization knows its role in the larger ecosystem, but has not formalized its capabilities to interact and share information externally.

Cybersecurity Framework

Tier 3: Repeatable

- Risk Management Process – The organization's risk management practices are formally approved and expressed as policy. Organizational cybersecurity practices are regularly updated based on the application of risk management processes to changes in business/mission requirements and a changing threat and technology landscape.
- Integrated Risk Management Program – There is an organization-wide approach to manage cybersecurity risk. Risk-informed policies, processes, and procedures are defined, implemented as intended, and reviewed. Consistent methods are in place to respond effectively to changes in risk. Personnel possess the knowledge and skills to perform their appointed roles and responsibilities.
- External Participation – The organization understands its dependencies and partners and receives information from these partners that enables collaboration and risk-based management decisions within the organization in response to events.

Cybersecurity Framework

Tier 4: Adaptive

- Risk Management Process – The organization adapts its cybersecurity practices based on lessons learned and predictive indicators derived from previous and current cybersecurity activities. Through a process of **continuous improvement** incorporating advanced cybersecurity technologies and practices, the organization actively adapts to a changing cybersecurity landscape and responds to evolving and sophisticated threats in a timely manner.
- Integrated Risk Management Program – There is an organization-wide approach to managing cybersecurity risk that uses risk-informed policies, processes, and procedures to address potential cybersecurity events. Cybersecurity risk management is part of the organizational culture and evolves from an awareness of previous activities, information shared by other sources, and continuous awareness of activities on their systems and networks.
- External Participation – The organization manages risk and actively shares information with partners to ensure that accurate, current information is being distributed and consumed to improve cybersecurity before a cybersecurity event occur

Services

- Data Sales
- State Portal
- 1.800.georgia
- Professional Services
  - Training and Education
  - Office of Information Security
    - Security Program Reviews
    - Security Services
  - Enterprise Portfolio Management Services
- Unified Messaging
- Spectrum Management

Home » Services » Professional Services » Office of Information Security

Office of Information Security

The Office of Information Security (OIS) is a component of the Enterprise Governance and Planning (EGAP) division of the Georgia Technology Authority (GTA). It operates in a similar manner to a central information security program as defined by the National Institute of Standards and Technologies (NIST), Special Publication 800-12, An Introduction to Computer Security: The NIST Handbook. Each agency of the state is required to run its own information security program in compliance with the information security policies and standards issued by GTA. To assist the agencies with this responsibility, OIS performs the following activities.

Security Program Reviews

GTA has adopted the security requirements created by the Federal Information Security Management Act (FISMA) of 2002 and the FISMA Implementation Project conducted by NIST. GTA's policies and standards were developed in accordance with FISMA, and OIS conducts program reviews to help agencies identify and remediate deficiencies. These reviews are based on federal guidance from Program Review for Information Security Management Assistance (PRISMA). The reviews are focused on the agency's security management and operational processes based on requirements established by statewide security policies, the Federal Information Security Management Act (FISMA), and the National Institute of Standards and Technology (NIST) Computer Security Division.

The ultimate goal of these reviews is to assist agencies in:

- Building robust information security and risk management programs
- Preparing for future reporting and audit requirements
- Responding to audit or assessment findings
- Improving their and the state's overall security posture

Education, Training and Awareness

In coordination with the University System of Georgia (USG), GTA has developed a training program for agency information security staff. These classes are delivered by USG on a rotating basis and prepare the security staff to create and operate an information security program. In addition, OIS works with the agencies to develop security awareness training programs and to educate agency leadership about information security issues and responsibilities.

Information Security

+ Cybersecurity Framework

Security Services

+ Education, Communication, Collaboration, and Awareness

The Archer GRC Portal

Hot Topics

» IT and Contracting Legislation Information: Updated

» DIR Provides Front Line Telecom Support during Hurricane Season, Emergencies

» Texas Launches First-Ever One Page State Government Website

» Easy-to-Use Geographic Imagery for Government Entities across Texas

» DIR Introduces New VoIP Capability for Capitol Complex Customers

» See all Hot Topics

Customer Spotlight

» DFPS and TxDOT represented in Hackathon

» Red River Rivalry? Not in State Technology!

» Texas Workforce Commission Welcomes Veterans to Texas

TAC 202

First, some background.

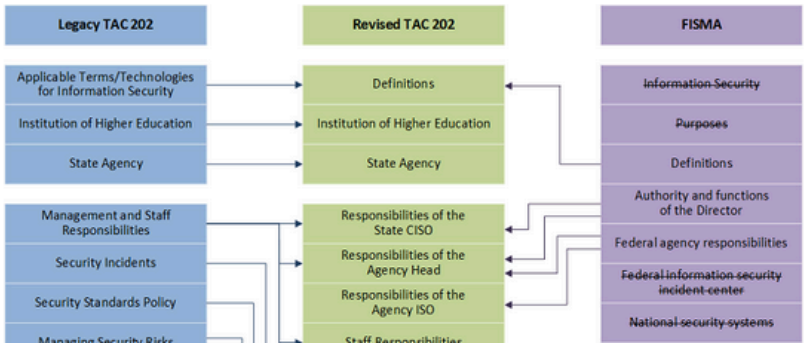
Back when it was proposed in 2002, Legacy TAC 202 established a baseline of security standards for Texas state agencies and institutions of higher education.

Setting security standards at the federal level is FISMA, which stands for the Federal Information Security Management Act. FISMA requires federal agencies and their contractors to safeguard their information systems and assets. The National Institute of Standards and Technology, known as NIST, helps develop standards and guidelines for FISMA.

FISMA was updated in 2013.

The committee of state ISOs and others have revised TAC 202 to move it closer to FISMA and NIST 800-53. The Revised TAC covers agency responsibilities and includes a Control Standards Catalog.

The graphic below shows how the Revised TAC 202 aligns more closely with FISMA.



Security Control Standards Catalog

Version 1.2

Texas Department of
Information Resources

April 3, 2015

Contents

About the Security Control Standards Catalog 1

- Document Life Cycle 1
- Revision History 2
- Scope 2
- Exceptions 2
- Control Details and Sample Format 2
- Notes on the Control Details and Sample Format 3

Security Controls Standards 5

- AC–Access Control 5
- AP–Authority and Purpose 22
- AR–Accountability, Audit, and Risk Management 24
- AT–Awareness and Training 30
- AU–Audit and Accountability 33
- CA–Security Assessment and Authorization 44
- CM–Configuration Management 50
- CP–Contingency Planning 58
- DI–Data Quality and Integrity 67
- DM–Data Minimization and Retention 69
- IA–Identification and Authentication 72
- IP–Individual Participation and Redress 80
- IR–Incident Response 83
- MA–Maintenance 90
- MP–Media Protection 95
- PE–Physical and Environmental Protection 101
- PL–Planning 114
- PM–Program Management 119
- PS–Personnel Security 132
- RA–Risk Assessment 138
- SA–System and Service Acquisition 142
- SC–System and Communication Protection 156
- SE–Security 182
- SI–System and Information Integrity 184
- TR–Transparency 196
- UL–Use Limitation 199

Appendix A. NIST Control Families 201

Appendix B. Acronyms and Abbreviations 212

Appendix C. Glossary of Terms 214

Impact System Level	FISMA Assessment	FedRAMP Assessment
Low-	115	116
Moderate-	252	297
High-	329	N/A*

IA-2 IDENTIFICATION AND AUTHENTICATION (ORGANIZATIONAL USERS)

Control: The information system uniquely identifies and authenticates organizational users (or processes acting on behalf of organizational users).

Supplemental Guidance: Organizational users include employees or individuals that organizations deem to have equivalent status of employees (e.g., contractors, guest researchers). This control applies to all accesses other than: (i) accesses that are explicitly identified and documented in AC-14; and (ii) accesses that occur through authorized use of group authenticators without individual authentication. Organizations may require unique identification of individuals in group accounts (e.g., shared privilege accounts) or for detailed accountability of individual activity. Organizations employ passwords, tokens, or biometrics to authenticate user identities, or in the case multifactor authentication, or some combination thereof. Access to organizational information systems is defined as either local access or network access. Local access is any access to organizational information systems by users (or processes acting on behalf of users) where such access is obtained by direct connections without the use of networks. Network access is access to

organizational information systems by users (or processes acting on behalf of users) where such access is obtained through network connections (i.e., nonlocal accesses). Remote access is a type of network access that involves communication through external networks (e.g., the Internet). Internal networks include local area networks and wide area networks. In addition, the use of encrypted virtual private networks (VPNs) for network connections between organization-controlled endpoints and non-organization controlled endpoints may be treated as internal networks from the perspective of protecting the confidentiality and integrity of information traversing the network.

Organizations can satisfy the identification and authentication requirements in this control by complying with the requirements in Homeland Security Presidential Directive 12 consistent with the specific organizational implementation plans. Multifactor authentication requires the use of two or more different factors to achieve authentication. The factors are defined as: (i) something you know (e.g., password, personal identification number [PIN]); (ii) something you have (e.g., cryptographic identification device, token); or (iii) something you are (e.g., biometric). Multifactor solutions that require devices separate from information systems gaining access include, for example, hardware tokens providing time-based or challenge-response authenticators and smart cards such as the U.S. Government Personal Identity Verification card and the DoD common access card. In addition to identifying and authenticating users at the information system level (i.e., at logon), organizations also employ identification and authentication mechanisms at the application level, when necessary, to provide increased information security. Identification and authentication requirements for other than organizational users are described in IA-8. Related controls: AC-2, AC-3, AC-14, AC-17, AC-18, IA-4, IA-5, IA-8.

controls: AC-2, AC-3, AC-14, AC-17, AC-18, IA-4, IA-5, IA-8.

Control Enhancements:

(1) *IDENTIFICATION AND AUTHENTICATION | NETWORK ACCESS TO PRIVILEGED ACCOUNTS*

The information system implements multifactor authentication for network access to privileged accounts.

Supplemental Guidance: Related control: AC-6.

(2) *IDENTIFICATION AND AUTHENTICATION | NETWORK ACCESS TO NON-PRIVILEGED ACCOUNTS*

The information system implements multifactor authentication for network access to non-privileged accounts.

(3) *IDENTIFICATION AND AUTHENTICATION | LOCAL ACCESS TO PRIVILEGED ACCOUNTS*

The information system implements multifactor authentication for local access to privileged accounts.

Supplemental Guidance: Related control: AC-6.

(4) *IDENTIFICATION AND AUTHENTICATION | LOCAL ACCESS TO NON-PRIVILEGED ACCOUNTS*

The information system implements multifactor authentication for local access to non-privileged accounts.

(5) *IDENTIFICATION AND AUTHENTICATION | GROUP AUTHENTICATION*

The organization requires individuals to be authenticated with an individual authenticator when a group authenticator is employed.

Supplemental Guidance: Requiring individuals to use individual authenticators as a second level of authentication helps organizations to mitigate the risk of using group authenticators.

(6) *IDENTIFICATION AND AUTHENTICATION | NETWORK ACCESS TO PRIVILEGED ACCOUNTS - SEPARATE DEVICE*

The information system implements multifactor authentication for network access to privileged accounts such that one of the factors is provided by a device separate from the system gaining access and the device meets [*Assignment: organization-defined strength of mechanism requirements*].

Supplemental Guidance: Related control: AC-6.

(7) *IDENTIFICATION AND AUTHENTICATION | NETWORK ACCESS TO NON-PRIVILEGED ACCOUNTS - SEPARATE DEVICE*

The information system implements multifactor authentication for network access to non-privileged accounts such that one of the factors is provided by a device separate from the system gaining access and the device meets [*Assignment: organization-defined strength of mechanism requirements*].

Control Enhancements:

(1) *IDENTIFICATION AND AUTHENTICATION | NETWORK ACCESS TO PRIVILEGED ACCOUNTS*

The information system implements multifactor authentication for network access to privileged accounts.

Supplemental Guidance: Related control: AC-6.

(2) *IDENTIFICATION AND AUTHENTICATION | NETWORK ACCESS TO NON-PRIVILEGED ACCOUNTS*

The information system implements multifactor authentication for network access to non-privileged accounts.

(3) *IDENTIFICATION AND AUTHENTICATION | LOCAL ACCESS TO PRIVILEGED ACCOUNTS*

The information system implements multifactor authentication for local access to privileged accounts.

Supplemental Guidance: Related control: AC-6.

(4) *IDENTIFICATION AND AUTHENTICATION | LOCAL ACCESS TO NON-PRIVILEGED ACCOUNTS*

The information system implements multifactor authentication for local access to non-privileged accounts.

(5) *IDENTIFICATION AND AUTHENTICATION | GROUP AUTHENTICATION*

The organization requires individuals to be authenticated with an individual authenticator when a group authenticator is employed.

Supplemental Guidance: Requiring individuals to use individual authenticators as a second level of authentication helps organizations to mitigate the risk of using group authenticators.

(6) *IDENTIFICATION AND AUTHENTICATION | NETWORK ACCESS TO PRIVILEGED ACCOUNTS - SEPARATE DEVICE*

The information system implements multifactor authentication for network access to privileged accounts such that one of the factors is provided by a device separate from the system gaining access and the device meets [*Assignment: organization-defined strength of mechanism requirements*].

Supplemental Guidance: Related control: AC-6.

(7) *IDENTIFICATION AND AUTHENTICATION | NETWORK ACCESS TO NON-PRIVILEGED ACCOUNTS - SEPARATE DEVICE*

The information system implements multifactor authentication for network access to non-privileged accounts such that one of the factors is provided by a device separate from the system gaining access and the device meets [*Assignment: organization-defined strength of mechanism requirements*].

(8) *IDENTIFICATION AND AUTHENTICATION | NETWORK ACCESS TO PRIVILEGED ACCOUNTS - REPLAY RESISTANT*

The information system implements replay-resistant authentication mechanisms for network access to privileged accounts.

Supplemental Guidance: Authentication processes resist replay attacks if it is impractical to achieve successful authentications by replaying previous authentication messages. Replay-resistant techniques include, for example, protocols that use nonces or challenges such as Transport Layer Security (TLS) and time synchronous or challenge-response one-time authenticators.

(9) *IDENTIFICATION AND AUTHENTICATION | NETWORK ACCESS TO NON-PRIVILEGED ACCOUNTS - REPLAY RESISTANT*

The information system implements replay-resistant authentication mechanisms for network access to non-privileged accounts.

Supplemental Guidance: Authentication processes resist replay attacks if it is impractical to achieve successful authentications by recording/replaying previous authentication messages. Replay-resistant techniques include, for example, protocols that use nonces or challenges such as Transport Layer Security (TLS) and time synchronous or challenge-response one-time authenticators.

(10) *IDENTIFICATION AND AUTHENTICATION | SINGLE SIGN-ON*

The information system provides a single sign-on capability for [*Assignment: organization-defined information system accounts and services*].

Supplemental Guidance: Single sign-on enables users to log in once and gain access to multiple information system resources. Organizations consider the operational efficiencies provided by single sign-on capabilities with the increased risk from disclosures of single authenticators providing access to multiple system resources.

(11) *IDENTIFICATION AND AUTHENTICATION | REMOTE ACCESS - SEPARATE DEVICE*

The information system implements multifactor authentication for remote access to privileged and non-privileged accounts such that one of the factors is provided by a device separate from the system gaining access and the device meets [*Assignment: organization-defined strength of mechanism requirements*].

Supplemental Guidance: For remote access to privileged/non-privileged accounts, the purpose of requiring a device that is separate from the information system gaining access for one of the factors during multifactor authentication is to reduce the likelihood of compromising authentication credentials stored on the system. For example, adversaries deploying malicious code on organizational information systems can potentially compromise such credentials resident on the system and subsequently impersonate authorized users. Related control: AC-6.

(12) *IDENTIFICATION AND AUTHENTICATION | ACCEPTANCE OF PIV CREDENTIALS*

The information system accepts and electronically verifies Personal Identity Verification (PIV) credentials.

Supplemental Guidance: This control enhancement applies to organizations implementing logical access control systems (LACS) and physical access control systems (PACS). Personal Identity Verification (PIV) credentials are those credentials issued by federal agencies that conform to FIPS Publication 201 and supporting guidance documents. OMB Memorandum 11-11 requires federal agencies to continue implementing the requirements specified in HSPD-12 to enable agency-wide use of PIV credentials. Related controls: AU-2, PE-3, SA-4.

(13) *IDENTIFICATION AND AUTHENTICATION | OUT-OF-BAND AUTHENTICATION*

The information system implements [*Assignment: organization-defined out-of-band authentication*] under [*Assignment: organization-defined conditions*].

Supplemental Guidance: Out-of-band authentication (OOBA) refers to the use of two separate communication paths to identify and authenticate users or devices to an information system. The first path (i.e., the in-band path), is used to identify and authenticate users or devices, and generally is the path through which information flows. The second path (i.e., the out-of-band path) is used to independently verify the authentication and/or requested action. For example, a user authenticates via a notebook computer to a remote server to which the user desires access, and requests some action of the server via that communication path. Subsequently, the server contacts the user via the user's cell phone to verify that the requested action originated

Risk Management Framework

Source: NIST SP 800-37 Rev.1

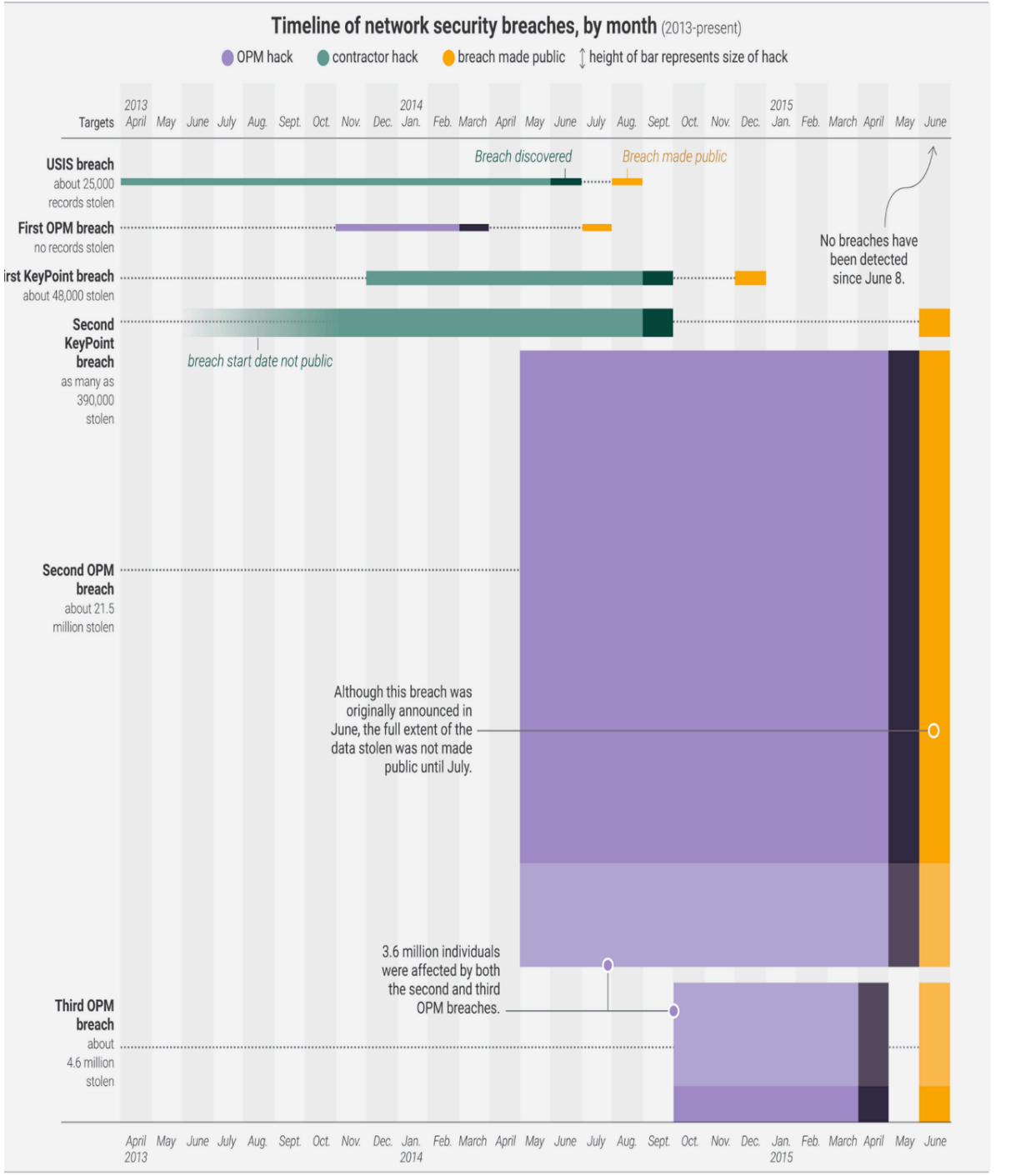

Timeline of network security breaches, by month (2013-present)
OPM hack
contractor hack
breach made public
height of bar represents size of hack
Targets
2013 April May June July Aug. Sept. Oct. Nov. Dec. 2014 Jan. Feb. March April May June July Aug. Sept. Oct. Nov. Dec. 2015 Jan. Feb. March April May June
USIS breach
about 25,000 records stolen
Breach discovered
Breach made public
First OPM breach
no records stolen
irst KeyPoint breach
about 48,000 stolen
No breaches have been detected since June 8.
Second KeyPoint breach
as many as 390,000 stolen
breach start date not public
Second OPM breach
about 21.5 million stolen
Although this breach was originally announced in June, the full extent of the data stolen was not made public until July.
3.6 million individuals were affected by both the second and third OPM breaches.
Third OPM breach
about 4.6 million stolen
April 2013 May June July Aug. Sept. Oct. Nov. Dec. Jan. 2014 Feb. March April May June July Aug. Sept. Oct. Nov. Dec. Jan. 2015 Feb. March April May June

Table 3: ISCM Capabilities FY 2013 & FY 2014

Agency	Information Security Continuous Monitoring Average FY 2013 (%)	Information Security Continuous Monitoring Average FY 2014 (%)
HHS	90	80
EPA	57	82
DOT	52	88
Commerce	69	88
NRC	95	89
DOD	76	90
USAID	97	90
HUD	85	91
Energy	86	92
Interior	86	94
DHS	94	95
State	82	95
VA	77	96
NSF	95	96
NASA	88	96
OPM	97	97
SSA	96	98
Treasury	84	98
ED	95	98
GSA	98	98
Labor	97	99
Justice	99	99
USDA	100	100
SBA	63	100
CFO Act Agency Average*	81	92

Table 4: Strong Authentication Capabilities FY 2013 & FY 2014

Agency	Strong Authentication FY 2013 (%)	Strong Authentication FY 2014 (%)
Labor	0	0
HUD	0	0
NRC	0	0
SBA	0	0
State	1	0
OPM	0	1
USAID	0	3
USDA	6	6
VA	4	10
NSF	0	19
Energy	9	29
DOT	7	31
Interior	0	36
Treasury	9	43
Justice	30	44
EPA	0	69
HHS	66	69
DHS	30	80
NASA	17	82
ED	75	85
SSA	85	85
DOD	89	87
Commerce	30	88
GSA	94	95
CFO Act Agency Average*	67	72

DevOps Revolution

Manifesto for Agile Software Development

We are uncovering better ways of developing
software by doing it and helping others do it.
Through this work we have come to value:

Individuals and interactions over processes and tools
Working software over comprehensive documentation
Customer collaboration over contract negotiation
Responding to change over following a plan

That is, while there is value in the items on
the right, we value the items on the left more.

Kent Beck	James Grenning	Robert C. Martin
Mike Beedle	Jim Highsmith	Steve Mellor
Arie van Bennekum	Andrew Hunt	Ken Schwaber
Alistair Cockburn	Ron Jeffries	Jeff Sutherland
Ward Cunningham	Jon Kern	Dave Thomas
Martin Fowler	Brian Marick	

INDIVIDUALS AND INTERACTIONS	OVER	PROCESSES AND TOOLS
WORKING SOFTWARE	OVER	COMPREHENSIVE DOCUMENTATION
CUSTOMER COLLABORATION	OVER	CONTRACT NEGOTIATION
RESPONDING TO CHANGE	OVER	FOLLOWING A PLAN

John Willis: 3 Threads: Agile Infrastructure

- Patrick Dubois and Andrew Schafer 2008 Agile Infrastruture conferecne BOF
- Marcel Wegerman had started an agile-sysadmin mailing list: promoted bridge between development and systems administration.
- DevOps Days Ghent 2009 : (DevOps Manifesto lowering the risk of change through tools and culture
- DevOps Days in Moutainview in 2010

John Willis: 3 Threads: Velocity

- John Aspaw 2009 Velocity presentation: "10+ Deploys Per Day," operations needed to change
- Jessie Robbins, 2007: "Operations is a competitive advantage… (Secret Sauce for Startups!)"
- Tim O'Reilly, 2006: "Operations: The New Secret Sauce."

John Willis: 3 Threads: Lean Startup

- Stephen Blank: Four Steps to Epiphany Eric Reis: Lessons Learned, Wealthfront, Lean Startup
- Jes Humble: Continuous Delivery, Lean Enterprise

Gene Kim

- Visible Ops: Agile ITIL
  - Release Management
  - Repeatable, automated build process
  - Golden builds stored in DSL (definitive software library)
  - Rebuild instead of Repair:
    - Known amount of time
    - Less config variance
    - Less complicated, junior staff
    - Senior staff work on systemic issues
    - Eliminate unplanned work

Gene Kim

- Visible Ops: Agile ITIL
- DSL-CMDB (configuration management database)
- Create list of supported components :
  - a. infrastructure and operating systems
  - b. applications
  - c. business rules
  - d. data

- No developers should be part of the build process both for security (segregation of duties) and to advance the operations staff's ability to provision and maintain infrastructure unaided

  No more "patch and pray," test patches on pre-production systems
  - patching in production can easily create errors, accumulate latent errors
  - automated detection of changes

Gene Kim

- Visible OPS Security:
  - Create situational awareness
  - Embed security controls in operations
  - Risk-based approach, most valuable assets
- Phoenix Project

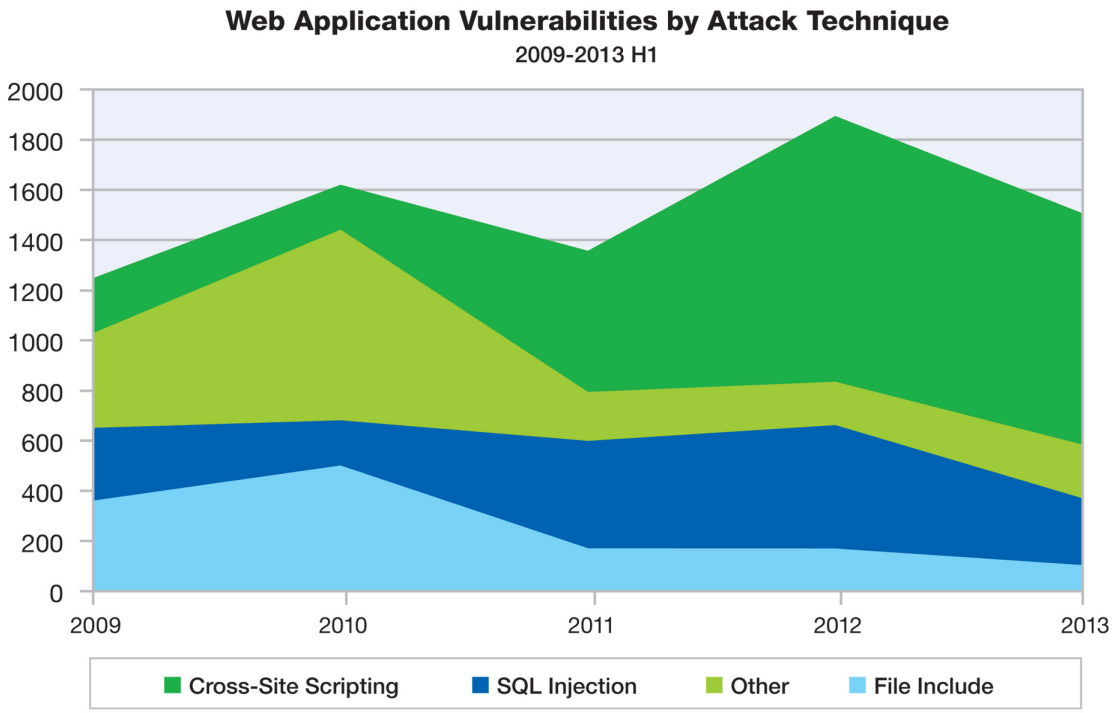
Continuous Improvement

- Continuous quality improvement (CQI)
- Total quality management (TQM
- Deming, Toyota
- Team accountability and contribution

Continuous Delivery

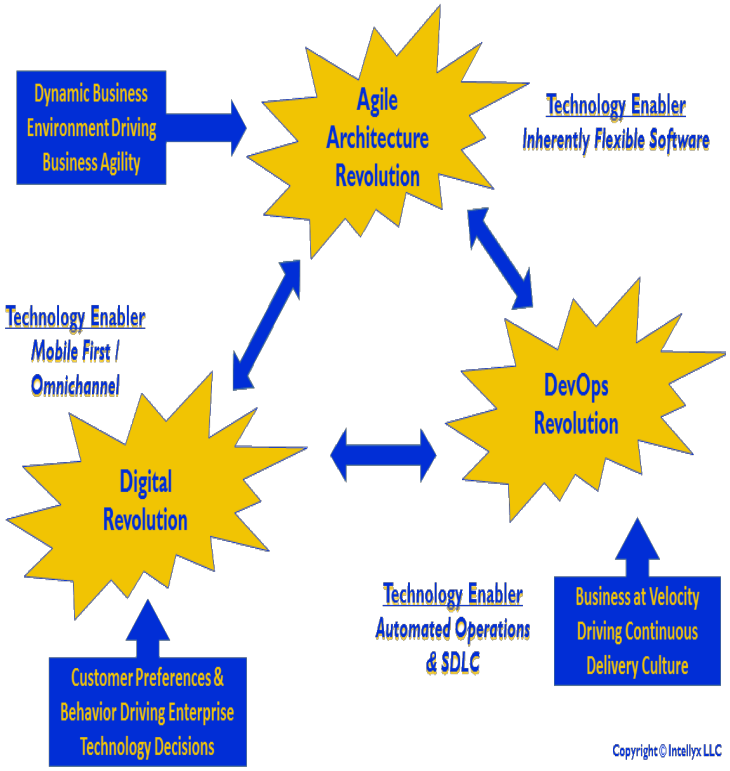
- Netflix: 100's of deployments per day
- Amazon: Changes to production every 11.6 seconds on average in May of 2011
- Facebook: Releases to production twice
- Google: Developed on mainline

DevOps Security and Continuous Failure: Lessons From Heartbleed, Shellshock, and Countless Other Security Flaws



Source: IBM X-Force® Research and Development

DevOps Security Potential



- Software defined sec
- Isolation
- Collab, control
- Situational awareness, rapid response

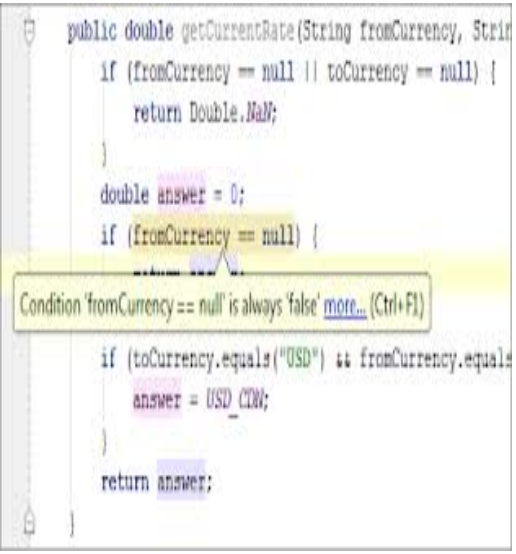
Security Island

- Security community
- Fear and liability
- Unique goals
- Black box, reactive tools

Spiritual Rebirth

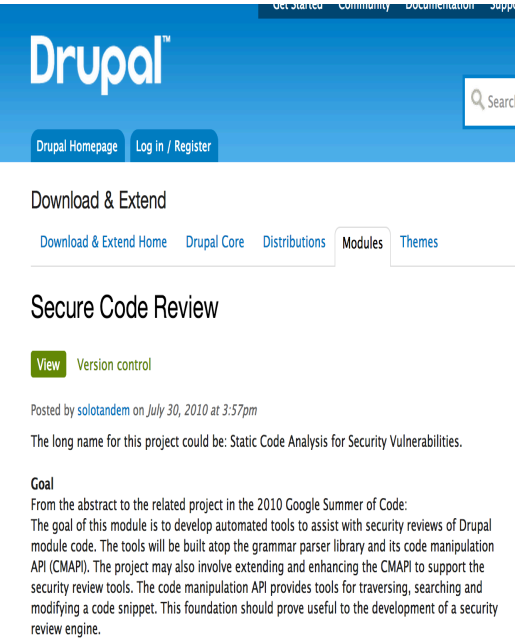
- Integrate security in devops
- Shared mindset
- Goals and metrics, tools and tests in pipeline

Static Analysis (SAST)



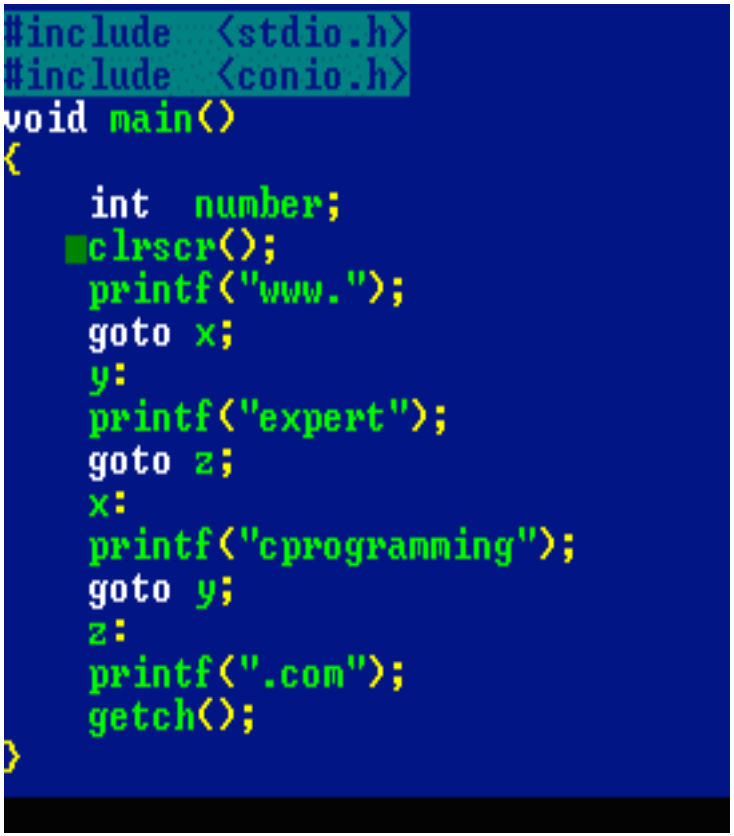
- Syntax checking
- Structure/complexity
- Bugs, duplication
- Standards
- Security configuration

Example: PHP/Drupal



- PHP Code sniffer
- Coder/Secure Coding Modules
- PHPUnit/SimpleTest, Jenkins (STDD/SDD)

Problems/Limitations



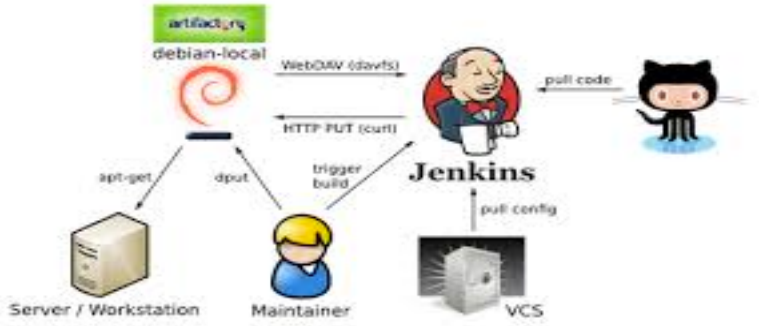
```
#include  <stdio.h>
#include  <conio.h>
void main()
{
    int  number;
   clrscr();
    printf("www.");
    goto x;
    y:
    printf("expert");
    goto z;
    x:
    printf("cprogramming");
    goto y;
    z:
    printf(".com");
    getch();
}
```

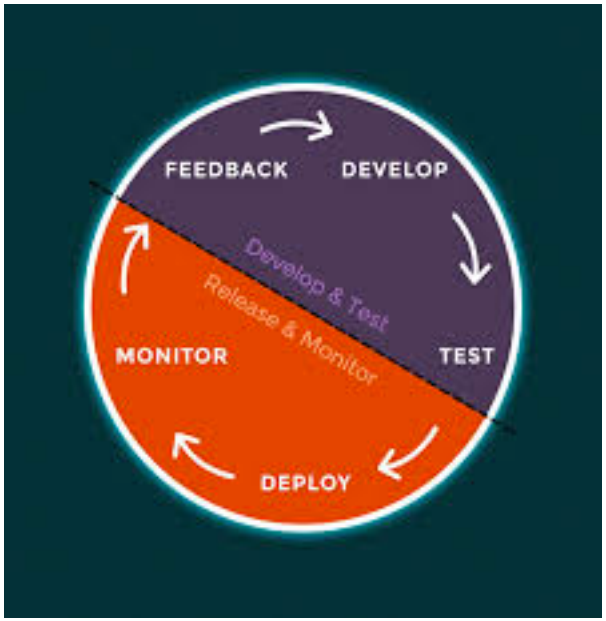
- Failed in Heartbleed
- Old/complex languages
- Bad/complex code
- Dynamic languages

Dynamic App Analysis (DAST)

- Pen tests/vuln scans
- Web app scanner: ZAP
- False positives

DevOps Pipeline vs. Lifecycle





- Continuous feedback
- Event aggregation, normalization, correlation
- Trickle testing, deployment

Code Review/Collaboration

- Code review app
- Review board
- CI: Jenkins
- Paired programming

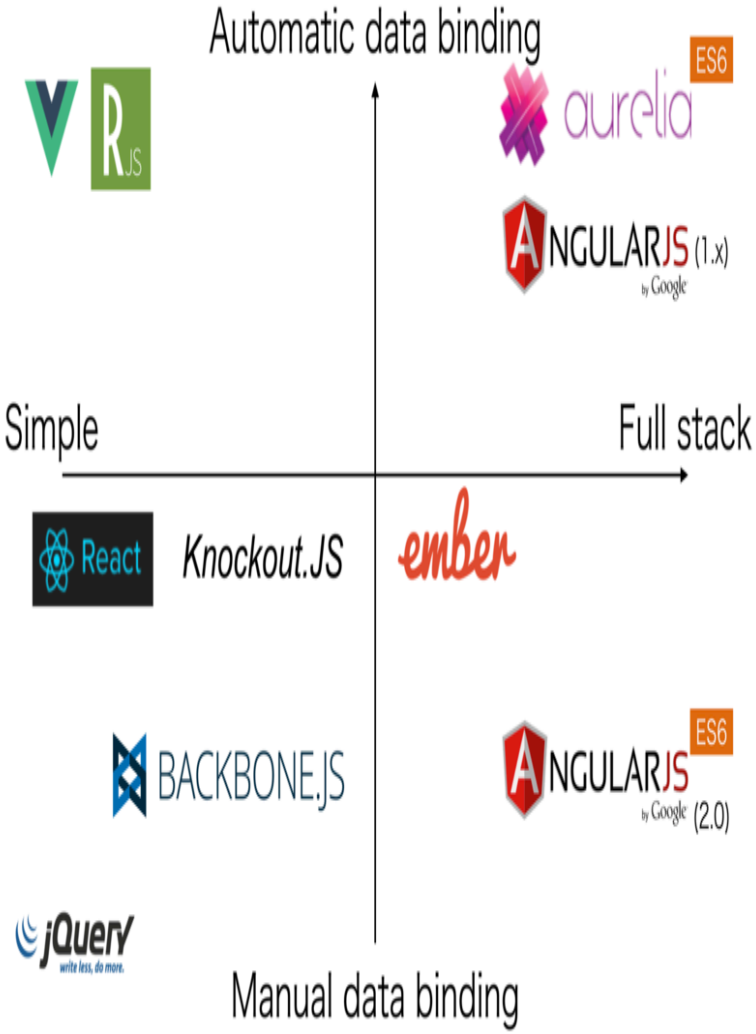
Architectural Risk Analysis

- Design flaws vs. bugs
- Scope of impact, risk potential
- Safeguards

Code Security: Human Factors

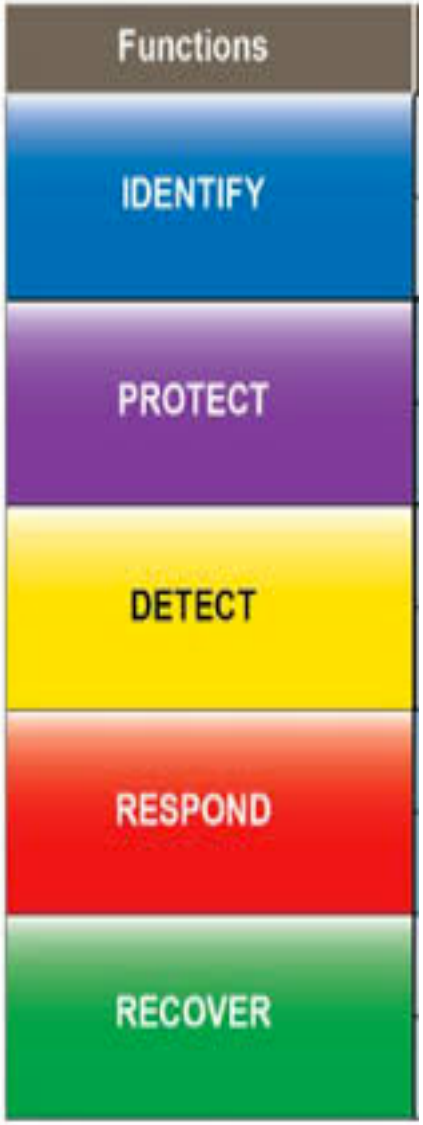
- Technology, programmer, community
- Secure coding guidelines
- Dedicated security team
- Gateways

Vendor Management



- Check third-party code
- Creds, dependencies
- Limitations/unknowns

Security Frameworks/Process

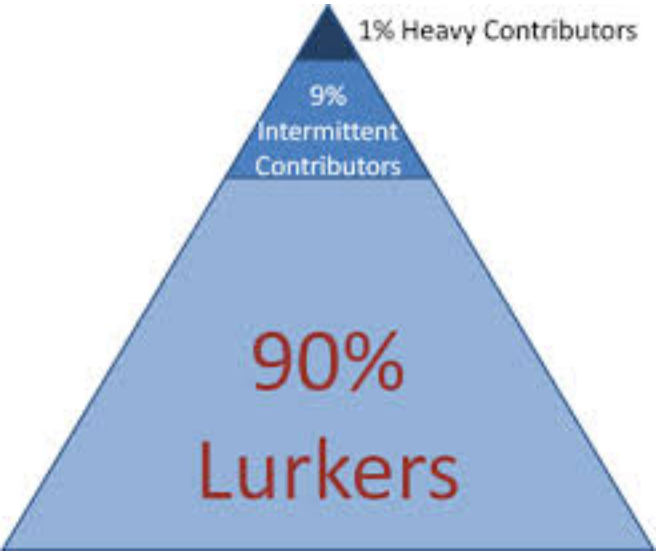


- CMMI: Security Extension
- OWASP ESAPI
- FISMA
- NIST Cybersecurity
- ITIL
- ISO 2700
- COBIT

Risk Management

- Leanest stack, minimal privileges
- Simplicity: Occam
- Open standards
- API Security

Community Support



- Developers and ops in security
- Contribute time and $

Show value and cost

Technology to the Rescue: People are People

- Developers are developers
- Better languages: Go and Elixir
- Safeguards (e.g., MFA)

Security Metrics & Marketing

- App security
- Config management
- Financial
- Incident management
- Patch management
- Vulnerability management
- Balanced Sec Scorecard

DevOps Revolution

- *C*ulture – Own the change to drive collaboration and communication
- *A*utomation – Take manual steps out of your value chain
- *L*ean – Use lean principles to enable higher cycle frequency
- *M*etrics – Measure everything and use data to refine cycles
- *S*haring – Share experiences, successful or not, to enable others to learn

DevOps Agile Continuum

- Continuous Integration
- Continuous Delivery
- Continuous Release
- Continuous Improvement
- Continuous Monitoring

DevOps Security: Potential Benefits

- Software Defined Security
- Isolation
- Precise control
- Immutability
- Situational awareness
- Rapid response
- Collaboration

DevOps Tools

- Version Control and Code Review (Git, GitHub, Gerrit)
- Configuration Management (Puppet, Ansible)
- Virtualization (Vagrant, Docker)
- Microservices
- Cloud Computing (AWS, Azure)

Open Source

Open Source

- Collaboration
- Governance/Review
- Extendable
- Modular
- Security Infrastructure As Code
- COTS

Honorary" Open Source/DevOps Community Members

- Apple
- Google
- Amazon
- GitHub

DevOps Open Source

- Google (Kubernetes, Go)
- Netflix (Security Monkey)
- Docker (Docker)
- HashiCorp (Vagrant, Packer)
- Cloud Bees (Jenkins)
- Sound Cloud (Prometheus)

Open Source Adoption

- Microsoft
- IBM
- Oracle
- HP

Government Open Source

- NSA SELinux
- Open Source Software
- White House

Federal Government

- OSS should be leveraged by the U.S. federal government to realize open government and save tax payers money, especially at a time when controlling wasteful spending could not be more important. In this regard, there are six reasons for the government to move to OSS:
  - 1. Improved Security and Privacy over proprietary software;
  - 2. Increased procurement speed so agency's can get their programs deployed faster, particularly for rapid prototyping, experimentation, and where the ability to "test drive" the software is important;
  - 3. No lock into one vendor, support can be provided by anyone since the code is in the public domain;
  - 4. Reduced cost of license and support, on average, open source products provide same functionality at a 80-90% lower cost to the taxpayers;
  - 5. Improved quality, normally, supported open source products go through three times more quality reviews than proprietary software as part of community review, indemnification review, and then productizing; and,
  - 6. The Government can become part of the open source community and directly inject their specific requirements into the product.

Open Source

- Linux
- Package Management
- Modular

U.S. Digital Services Playbook

The American people expect to interact with government through digital channels such as websites, email, and mobile applications. By building digital services that meet their needs, we can make the delivery of our policy and programs more effective.

Today, too many of our digital services projects do not work well, are delivered late, or are over budget. To increase the success rate of these projects, the U.S. Government needs a new approach. We created a playbook of 13 key “plays” drawn from successful practices from the private sector and government that, if followed together, will help government build effective digital services.

DIGITAL SERVICE PLAYS

1. Understand what people need
2. Address the whole experience, from start to finish
3. Make it simple and intuitive
4. Build the service using agile and iterative practices
5. Structure budgets and contracts to support delivery
6. Assign one leader and hold that person accountable
7. Bring in experienced teams
8. Choose a modern technology stack
9. Deploy in a flexible hosting environment
10. Automate testing and deployments
11. Manage security and privacy through reusable processes
12. Use data to drive decisions
13. Default to open

Open Source: Infrastructure As Code

- Configuration as Code
- Docker (Dockerfile), Vagrant (Vagrantfile)
- Puppet (Manifest, Module)
- Git (Text-based)
- Linux (Open source, single purpose)
- Packages or Repositories (Git to install)
- Light-weight IDE (Vim, Sublime, Atom)

Version Control

Centralized vs. Distributed Version Control Systems

- Centralized:
  - CVS, Subversion, TFS,
- Distributed:
  - Git, Mercurial, Bazaar

FreeBSD Compromise

vs.

Linux Kernel.org Compromise

» About
» Features
» Applications
» Internetworking
» Advocacy
» Marketing
» Administration
» News
» Events
» Press
» Multimedia
» Artwork
» Logo
» Donations
» Legal Notices
» Privacy Policy

FreeBSD.org intrusion announced November 17th 2012

Security Incident on FreeBSD Infrastructure

From: FreeBSD Security Officer <security-officer@FreeBSD.org>
To: FreeBSD Security <FreeBSD-security@FreeBSD.org>
Bcc: freebsd-announce@freebsd.org, freebsd-security-notifications@FreeBSD.org
Reply-To: secteam@FreeBSD.org
Subject: Security Incident on FreeBSD Infrastructure

On Sunday 11th of November, an intrusion was detected on two machines within the FreeBSD.org cluster. The affected machines were taken offline for analysis. Additionally, a large portion of the remaining infrastructure machines were also taken offline as a precaution.

We have found no evidence of any modifications that would put any end user at risk. However, we do urge all users to read the report available at http://www.freebsd.org/news/2012-compromise.html and decide on any required actions themselves. We will continue to update that page as further information becomes known. We do not currently believe users have been affected given current forensic analysis, but we will provide updated information if this changes.

As a result of this event, a number of operational security changes are being made at the FreeBSD Project, in order to further improve our resilience to potential attacks. We plan, therefore, to more rapidly deprecate a number of legacy services, such as cvsup distribution of FreeBSD source, in favour of our more robust Subversion, freebsd-update, and portsnap models.

More information is available at http://www.freebsd.org/news/2012-compromise.html

Saturday November 17th, 2012

Table of Contents

- Announcement
- Update: 30th April 2013
- Update: 23rd March 2013
- Update: 3rd March 2013
- Update: 29th December 2012
- Update: 27th November 2012
- Update: 22nd November 2012
- Update: 18th November 2012
- Initial Details: 17th November 2012
- What is the Impact?
- What has FreeBSD.org done about this?
- Recommendations

Update: April 30th, 2013

Port managers and cluster administrators have completed the restoration of binary package building in the last few weeks. This has brought us back the continuous updates for the old-style binary packages on the 8.x and 9.x -STABLE branches. Note that, as beneficial consequences, Release Candidate builds for the 8.4 release cycle can now include binary packages on the install media, and the Project was able to add the missing binary packages retroactively for 9.1-RELEASE on i386 and amd64 platforms.

Port managers are currently working on introducing new-style (as known as "pkgng") binary packages in the coming months, please check the FreeBSD ports announcements list for further gradual status updates.

This is planned to be the last status update to this page. An official announcement will be sent to the FreeBSD announcements mailing list with the further details soon.

Update: March 23rd, 2013

Port managers have successfully restored some of the Project's binary package building capacity. There are some issues left still to resolve, e.g. how to publish the resulting package sets in a secure manner or how to build packages seamlessly for 8.x and 9.x systems on a recent 10.x system that the head node ("pointyhat") is running, but we are very close to finish with the preparations required for providing binary packages for the upcoming 8.4 and further releases.

Update: March 3rd, 2013

Redports underwent a full security audit, and as a result could be brought back on line. This took place on the 5th February, and since then more backend hardware has been added to bring it back up to full strength. On 11th February, sanity checks for ports have been turned back on, reenabling generation and update of the INDEX files used. The portsnap(8) service has been switched from CVS to SVN on 25th February. The binary package building infrastructure has undergone a major security review, and as a result many changes have been made to the code. The review completed on the 16th February and we are now in the process of bringing it up on new hardware. At this point, we expect new binary packages to be available in 2-4 weeks.

Update: December 29th, 2012

With the exception of systems relating to the building and testing of packages, all FreeBSD.org infrastructure has now been brought back online. A full audit of the third party package build infrastructure code ("pointyhat") and package testing infrastructure ("redports") continues, and neither system will be brought back online until audits are complete.

As a result, FreeBSD 9.1-RELEASE will be published with only minimal i386 and amd64 (x86_64) precompiled package sets available, and with no packages available for other architectures. This package set will be available on the DVD image, and are sufficient to install either the GNOME or the KDE desktop environment. For any other uses, or for any packages not included on the CD, either using the most recently available -stable package collection or compiling ports from the ports tree are recommended. Packages for 9.1-RELEASE will be made available at a later date. Instructions for obtaining and updating the ports tree can be found in the FreeBSD Handbook.

Update: November 27th, 2012

Due to the legacy third-party package build controller head nodes being offlined pending reinstall, we have been unable to build new package sets over the last two weeks. As a result, FreeBSD 9.1-RELEASE has been delayed as it was felt that we should not ship the release without at least a minimal package set available. We are now in a position where we are once again able to build third-party packages for both of our Tier-1 architectures (i386 and amd64), and are planning on releasing it within the next few days with only a slightly limited set of packages. Please note that historically we have also provided packages on a best-effort basis for some of our Tier-2 architectures such as sparc64, ia64 and powerpc. We are not currently expecting to be in a position to build any Tier-2 packages before FreeBSD 9.1 ships, so initially no precompiled packages will be available for these platforms. We may be in a position to provide some packages for these architectures shortly after the release.

A few reports covering this incident on external tech news websites have confused details relating to how this incident was discovered. Over the last few weeks, many of our primary cluster servers have been either physically relocated and/or replaced with new hardware as part of work planned several months in advance. The discovery of this incident was unrelated to this ongoing cluster maintenance. Several service outages in the days surrounding the incident were correctly attributed to ongoing cluster work, and were not related in any way to the compromise. In parallel with the physical upgrades and relocation of servers, we are also reworking the network layout in order to provide better functionality, security, resilience, and to reduce any

Kernel.org hacked, but Linux kernel safe thanks to git

By Fahmida Y. Rashid
2011-09-01

Article Rating: ★★★★★ / 9

Rate This Article:	Add This Article To:
Poor ○ ○ ○ ○ ◉ Best Rate	SHARE

Attackers compromised several servers at kernel.org using an off-the-shelf Trojan that appears to have entered via a compromised user credential. However, the source code for the Linux kernel does not appear to have been altered, thanks to its "git" distributed revision control system, say kernel maintainers.

Attackers have compromised several key servers at kernel.org, which houses the source code for the Linux kernel. The likelihood of attackers modifying the actual source code is very low, since the code is distributed across thousands of computers, according to developers who help maintain the code.

Attackers modified a number of files and logged user activity on the compromised servers, according to a message posted on the kernel.org website Aug. 31. The attackers were able to modify the OpenSSH client and server software installed on the compromised server. However, the attackers did not change the actual OpenSSH source code.

The attack happened "some time" in August and was discovered by Linux Kernel Organization officials on Aug. 28, according to the security notice on the site. The attackers used a Trojan to compromise the servers on Aug. 12, according to an email from John "'Warthog9" Hawley, the chief administrator of kernel.org. That email was sent to developers and posted on the text-sharing website Pastebin.

"Earlier today discovered a trojan existing on HPA's personal colo machine, as well as hera," the email said. HPA refers to kernel developer H Peter Anvin.

Other kernel.org boxes were discovered to have been hit by the same Trojan. The Trojan startup file was inserted into the startup scripts on the compromised server so that it would execute whenever the machine was started.

Site administrators have taken the compromised servers offline and are creating backups as well as reinstalling the systems, according to the message on the site. The investigation is ongoing.

Intruders apparently gained root access on one of the servers using a compromised user credential, the email said. It's not yet known how the attackers exploited the credentials to become root, according to the security notice.

A not so stupid git

While the intruders were able to compromise kernel.org servers, that doesn't translate to modifying the actual kernel code, Jonathan Corbet, a kernel developer, wrote on *Linux.com*.

There are thousands of copies of the kernel source code housed on developer machines around the world, and if someone tries to check in corrupted or modified code, the changes would be flagged by a distributed revision control system called git, according to Corbet.

Git calculates a cryptographically secure SHA-1 hash for each of the nearly 40,000 files that make up the Linux kernel. The name of each version of the kernel depends on the complete development history leading up to that version, and once it is published, it's not possible to change the old versions without someone noticing. Any changes to the source code would be noticed by anyone updating their personal copy of the code, according to the site's security notification.

Kernel.org is "just a distribution point" and no actual development happens on the server, according to Corbet. "When we say that we know the kernel source has not been compromised on kernel.org, we really know it," Corbet wrote.

Phalanx a prime suspect

The Trojan appeared to be a self-injecting rootkit known as Phalanx, Jon Oberheide, one of the Linux security researchers briefed by Linux Kernel Organization about the breach, told *The*

Git vs. GitHub

- https://training.github.com/kit/downloads/github-git-cheat-sheet.pdf
- https://github.com/WhiteHouse
- Government GitHub: https://government.github.com/
- Non-code uses

Git Conventions and Jargon

- "Pull Request"
- "Fork"
- "Clone"
- "Commit"
- "Push"
- "Repo"

Code Review

- Trigger code review on check in custom or add third-party extension
- -Many eyes
- -Know what's out there
- Dev on duty

Configuration Management

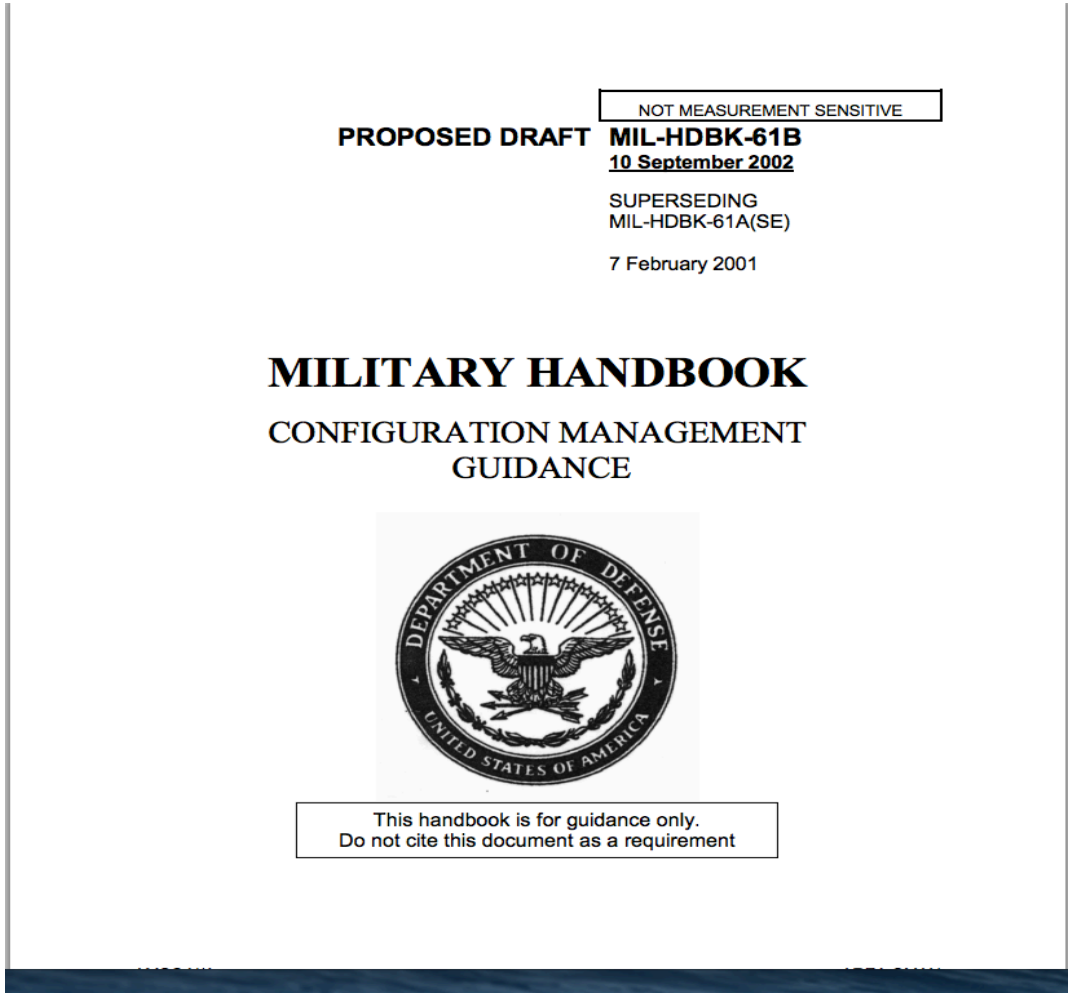
Configuration Management

- Major Control Family in FISMA
- Standardize Secure Platforms
- Prevent Configuration Drift
- Monitor, Auto Correct configuration
- Create users, manage permissions andfiles

Configuration Management

- Planning
- Identification and Control
- Accounting
- Verification and Audit
- Change Management

Configuration Management Roots



NOT MEASUREMENT SENSITIVE

PROPOSED DRAFT **MIL-HDBK-61B**
10 September 2002

SUPERSEDING
MIL-HDBK-61A(SE)

7 February 2001

MILITARY HANDBOOK

CONFIGURATION MANAGEMENT GUIDANCE

This handbook is for guidance only.
Do not cite this document as a requirement

Configuration Management

- CFEngine
- Puppet
- Chef
- Ansible
- Salt
- SCCM
- DCS

QUICK LOOK

	Ansible	SaltStack	Chef	Puppet
Terminology	Directive: Task Script: Playbook Master: Control Machine Children: Hosts	Directive: State Script: SLS Formula (SLS = SaLt State) Master: Master Children: Minions	Directive: Resource Script: Recipe (cookbook plural) Master: Server Children: Clients	Directive: Resource Script: Manifest Master: Master Children: Agents
Execution Order	Sequential	Sequential (since 0.17)	Sequential	Random, requires explicit ordering
Directive Language	YAML with Jinja2 templating	YAML with Jinja2 templating	Ruby DSL	Custom
Bootstrap Child Nodes	Not needed	Required (unless using salt-ssh)	Required	Required
Remote Execution	Built-in, easy	Built-in, easy	knife, challenging	mcollective, challenging

Puppet

- Install software
- Manage files and folders within the filesystem
- Manage cron jobs
- Run commands
- Manage users and groups
- Create configurable classes
- Use third-party Puppet modules and Puppet Forge
- Manually or automatically run Puppet to provision a machine

Puppet

- Manifests
- Modules
- DSL
- Provision Many Things (VM's containers)

Ansible

- Playbook
- Simpler DSL: SSH
- YAML

Virtualization

Virtualization

- Bare metal
- Virtual machines
- VirtualBox
- VMWare
- HyperV
- Xen
- KVM

Virtualization: Kernel-level

- BSD Jails
- Linux Containers
- Solaris Zones
- Windows Containers

DevOps-Friendly Virtualization

- Vagrant (Vagrant Box)
- Docker (Docker Container)
- Git Conventions

Containers

- Process isolation
- Logical segregation of duties
- Lower attack surface
- Control groups and namespaces: stronger isolation
  - Immutability
  - Different update and patching model
  - Precise control
  - Enhanced insight

Containers

“By using cgroups, system administrators gain fine-grained control over allocating, prioritizing, denying, managing, and monitoring system resources. Hardware resources can be appropriately divided up among tasks and users, increasing overall efficiency”.

“Multiple separate hierarchies of cgroups are necessary because each hierarchy is attached to one or more subsystems. A subsystem [2] represents a single resource, such as CPU time or memory”.

Cgroup Subsystems in Red Hat Enterprise Linux

- blkio — this subsystem sets limits on input/output access to and from block devices such as physical drives (disk, solid state, USB, etc.).
- cpu — this subsystem uses the scheduler to provide cgroup tasks access to the CPU.
- cpuacct — this subsystem generates automatic reports on CPU resources used by tasks in a cgroup.
- cpuset — this subsystem assigns individual CPUs (on a multicore system) and memory nodes to tasks in a cgroup.
- devices — this subsystem allows or denies access to devices by tasks in a cgroup.
- freezer — this subsystem suspends or resumes tasks in a cgroup.
- memory — this subsystem sets limits on memory use by tasks in a cgroup, and generates automatic reports on memory resources used by those tasks.
- net_cls — this subsystem tags network packets with a class identifier (classid) that allows the Linux traffic controller (tc) to identify packets originating from a particular cgroup task.
- net_prio — this subsystem provides a way to dynamically set the priority of network traffic per network interface.
- ns — the *namespace* subsystem
-

Infrastructure as Code

- Precisely Defined
- Auditable
- Controllable
- Reproducible
- Predictable
- Executable Documentation (!!!!!!)

Software Defined Defined Infrastructure

- SDIM: Machine Configuration (Virtualization, Chef & Puppet), AWS, VMWare & OpenStack
- SDN: Software Defined Networking
- SDS: Software Defined Storage
- Software Defined Application Security?

Vagrant

- Development Environments
- Up to Production
- Customize Virtual Machines
- Providers and Provisioners
- Packer and Atlas
- Infrastructure as Code

Vagrant Fundamentals: Vagrantfile

- Base Box
- Memory
- Network
- Provisioners
- Providers

Vagrant Fundamentals: Vagrant Commands

- Up
- Halt
- Rebuild
- Destroy
- Package

Development environments made easy.

Create and configure lightweight, reproducible, and portable development environments.

DOWNLOAD GET STARTED

VAGRANT WILL CHANGE HOW YOU WORK

Docker Fundamentals: Dockerfile

- FROM: Base image
- MAINTAINER: Author name
- RUN: Execute a shell or exec command on top of existing image
- ADD/COPY: Copy files from a URL or local area to container path
- EXPOSE: Open ports for listening
- CMD/ENTRYPOINT: What the container runs
- VOLUME: Storage area, files or folders created, mutable
- USER: Run as non-root user
- WORKDIR: Path for commands

Docker Fundamentals: Commands

- Build
- Run
- Commit
- Pull and push

Advanced Bash-Scripting Guide

An in-depth exploration of the art of shell scripting

Mendel Cooper

<thegrendel.abs@gmail.com>

10

10 Mar 2014

Revision History

Revision 6.5	05 Apr 2012	Revised by: mc
'TUNGSTENBERRY' release		
Revision 6.6	27 Nov 2012	Revised by: mc
'YTTERBIUMBERRY' release		
Revision 10	10 Mar 2014	Revised by: mc
'PUBLICDOMAIN' release		

This tutorial assumes no previous knowledge of scripting or programming, yet progresses rapidly toward an intermediate/advanced level of instruction . . . *all the while sneaking in little nuggets of UNIX® wisdom and lore*. It serves as a textbook, a manual for self-study, and as a reference and source of knowledge on shell scripting techniques. The exercises and heavily-commented examples invite active reader participation, under the premise that **`the only way to really learn scripting is to write scripts`**.

This book is suitable for classroom use as a general introduction to programming concepts.

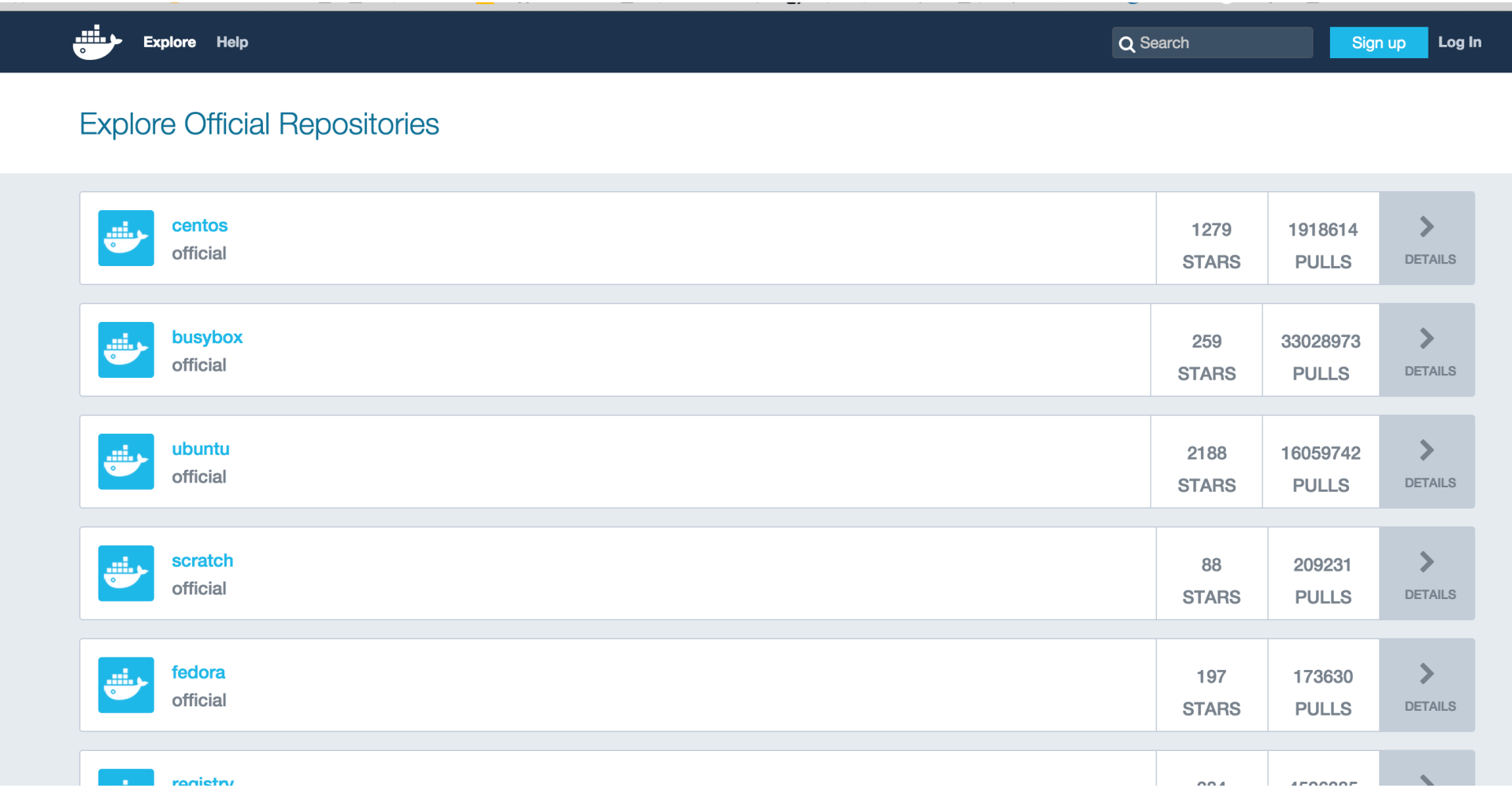
Dedication

For Anita, the source of all the magic

Table of Contents

Docker Fundamentals: DockerHub

- Repositories and images
- Operating systems and applications
- Instructions on Use
- Link to image manifest, history, and development repository (GitHub)


Explore
Help
Search
Sign up
Log In
Explore Official Repositories
centos
official
1279
STARS
1918614
PULLS
DETAILS
busybox
official
259
STARS
33028973
PULLS
DETAILS
ubuntu
official
2188
STARS
16059742
PULLS
DETAILS
scratch
official
88
STARS
209231
PULLS
DETAILS
fedora
official
197
STARS
173630
PULLS
DETAILS

15 lines (10 sloc) | 0.351 kB

Raw | Blame | History

```
FROM scratch
MAINTAINER The CentOS Project <cloud-ops@centos.org> - ami_creator
ADD centos-7-20150616_1752-docker.tar.xz /
# Volumes for systemd
# VOLUME ["/run", "/tmp"]

# Environment for systemd
# ENV container=docker

# For systemd usage this changes to /usr/sbin/init
# Keeping it as /bin/bash for compatability with previous


CMD ["/bin/bash"]
```

Executable File | 33 lines (25 sloc) | 1.124 kB

Raw Blame History

```
FROM php:5.6-apache
MAINTAINER Michael Babker <michael.babker@joomla.org> (@mbabker)

# Enable Apache Rewrite Module
RUN a2enmod rewrite

# Install PHP extensions
RUN apt-get update && apt-get install -y libpng12-dev libjpeg-dev zip unzip && rm -rf /var/lib/apt/lists/* \
        && docker-php-ext-configure gd --with-png-dir=/usr --with-jpeg-dir=/usr \
        && docker-php-ext-install gd
RUN docker-php-ext-install mysqli

VOLUME /var/www/html

# Define Joomla version and expected SHA1 signature
ENV JOOMLA_VERSION 3.4.3
ENV JOOMLA_SHA1 cd35ed61029d2ed0dc38cc70073944786bba7979

# Download package and extract to web volume
RUN curl -o joomla.zip -SL https://github.com/joomla/joomla-cms/releases/download/${JOOMLA_VERSION}/Joomla_${JOOMLA_VERSION}-Stabl
        && echo "$JOOMLA_SHA1 *joomla.zip" | sha1sum -c - \
        && mkdir /usr/src/joomla \
        && unzip joomla.zip -d /usr/src/joomla \
        && rm joomla.zip \
        && chown -R www-data:www-data /usr/src/joomla

# Copy init scripts and custom .htaccess
COPY docker-entrypoint.sh /entrypoint.sh
COPY makedb.php /makedb.php

ENTRYPOINT ["/entrypoint.sh"]
CMD ["apache2-foreground"]
```

Container Security

- Microservices
- Security: Service Architecture vs. Network Architecture
- Application and service based

Microservices

- Loosely coupled
- Limited footprint
- Docker security model
- Application level security

Microservices Application

- Python web application
- JavaScript service and
- Key-value store
- Reverse-proxy
- Monitoring service
- Logging service

Minimal Stack

- Remove dependencies
- Minimal OS: CoreOS, Atomic, Rancher (5 mb)
- Minimal container: Alpine Linux (alternative C library) or no operating system (Go)
- Performance, security, and usability

Disposable Infrastructure

- Immutability
- Rebuild rather than fix
- Cows vs. pets
- Scalability
- Resilience
- "Golden image"
- Configuration drift

Orchestration

- Kubernetes
- Heat
- Fleet
- TOSCA
- MaestroNG
- Fig/Docker Compose

Orchestration Authoring

- Docker Compose
- Vagrant

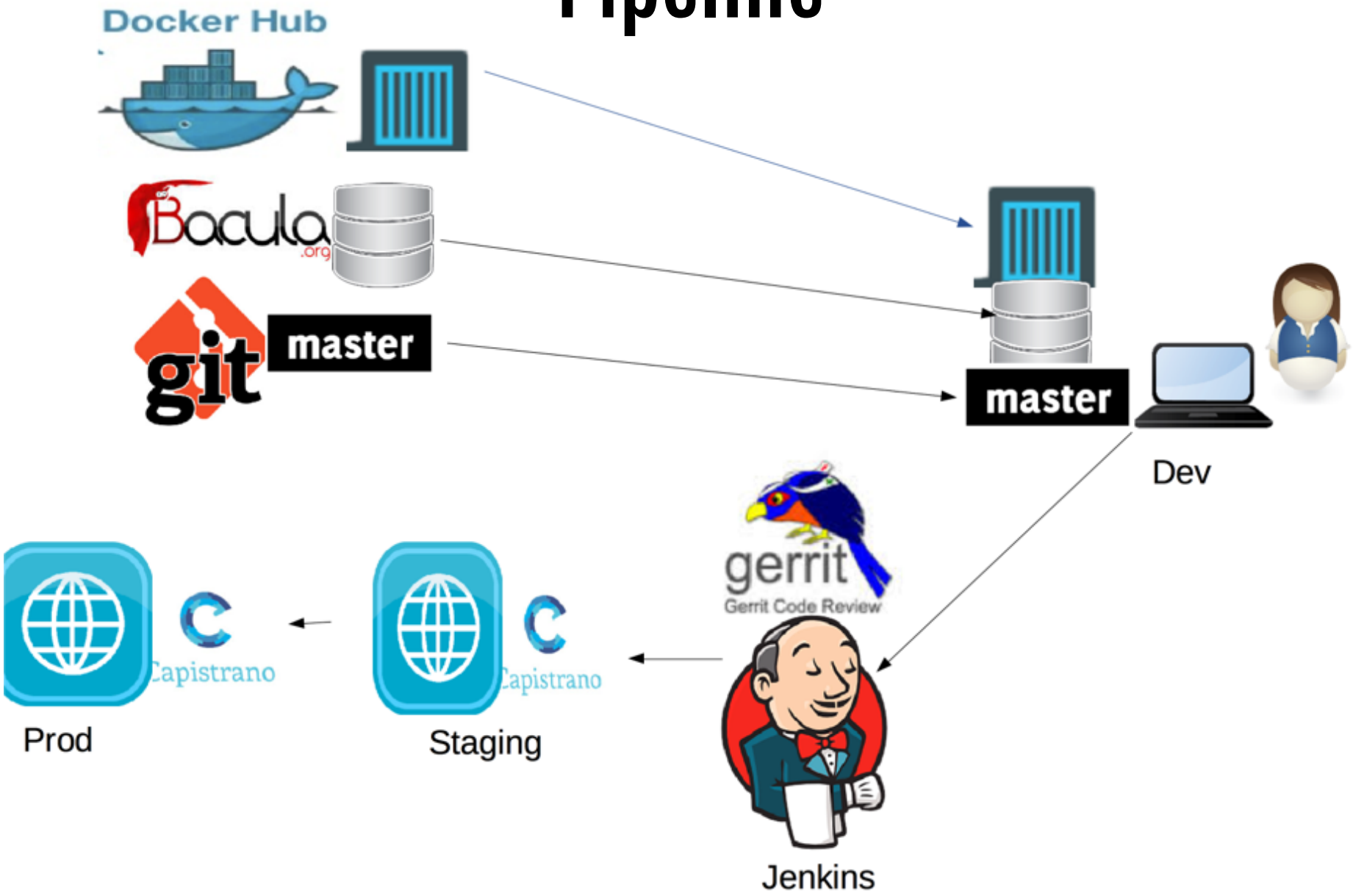
Congratulations! Your file is valid. Dismiss

Reset Workspace

```
InfluxSrv:
  image: "tutum/influxdb:0.8.8"
  ports:
    - "8083:8083"
    - "8086:8086"
  expose:
    - "8090"
    - "8099"
  environment:
    - PRE_CREATE_DB=cadvisor
cadvisor:
  image: "google/cadvisor:0.14.0"
  volumes:
    - "/:/rootfs:ro"
    - "/var/run:/var/run:rw"
    - "/sys:/sys:ro"
    - "/var/lib/docker/:/var/lib/docker:ro"
  links:
    - "InfluxSrv:influxsrv"
  ports:
    - "8080:8080"
  command: "-storage_driver=influxdb -storage_driver_db=cadvisor -storage_driver_host=influxsrv:8086"
grafana:
```

Continuous Integration

Pipeline


Docker Hub
Bacula
.org
git
master
master
Dev
gerrit
Gerrit Code Review
Capistrano
Capistrano
Prod
Staging
Jenkins

Continuous Integration

- Jenkins
  - Compose Pipeline
  - Automation
  - Plugins
  - Communication
  - Auditing

Pipeline

- Gerrit Trigger
- Phing (Static code analysis)
- SSH scripts, Platform tools: e.g., Drush
- Docker image build
- ZAP Proxy (vulnerability, penetatration testing)

Monitoring

Monitoring

- Open Source Monitoring/Metrics
  - Monitorama
  - Notifications, metrics, awareness
  - Security
  - ELK (Elasticsearch, Logstash, Kibana)

Monitoring

- Nagios
- Sensu
- CAdvisor
- Promethus
- OSSIM (SIEM)

Monitoring

- Traditional: Computer systems and services, monolithic, static
- Next generation: Application systems and services, customized, adaptable,

Prometheus

- Time series database
- Extensions
- Visualization and alerts

Communication

DevOps Communications

- Slack
- HipChat
- People and Bots
- Integrations: Jenkins, JIRA, Redmine
- Insight
- Audit

Communications

- GitHub
- Threat Exchanges

Cloud

Cloud Computing

- Amazon Web Services (AWS): market leader, FedRAMP, GovCloud (ITAR)
- Microsoft Azure, Google, IBM, VMWare, Digital Ocean
- Infrastructure as a service (IAAS)
- Platform as a service (PAAS)
- Software as a service (SAAS)

FedRAMP

- (FedRAMP),3 was designed to focus on three main areas: authorization, continuous monitoring, and federal security requirements. The initial goal of FedRAMP was to establish a unified risk management process that:
- · Increased security of cloud solutions through a common assessment approach.
- · Eliminated duplication of effort and achieved cost-savings through efficiency.
- · Enabled rapid acquisition through leveraged authorizations.
- · Improved reuse of authorization packages based on a common set of security requirements.
- · Facilitated use of shared services across multiple federal agencies.
- · Integrated a government-wide security approach.

Cloud

- AWS CLI and SDK
- S3
- RDS
- EC2
- DynamoDB
- ECS

Summary & Future

SANS Top 20: DevOps

- Inventory of Authorized and Unauthorized Devices: **Cloud manages hardware**
- Inventory of Authorized and Unauthorized Software: **Virtualization and containerization constrict and define software**
- Secure Configurations for Hardware & Software on Laptops, Workstations, & Servers: **Infrastructure as code and configuration management define and enforce security**
- Continuous Vulnerability Assessment and Remediation: **Microservices enable continuous service-oriented, application-centric monitoring and assessment**
- Malware Defense: **Immutable infrastructure prevents malware installation**

SANS Top 20: DevOps (cont).

- Application Software Security: **Continuous integration includes application security tests and checks**
- Wireless Device Control: **Cloud computing limits wireless access**
- Data Recovery Capability**: Infrastructure as code is readily archived and reproduced**
- Security Skills Assessment & Training: DevOps **communication builds security skills**
- Secure Configurations for Firewalls, Routers, & Switches: **Network infrastructure as code defines and enforces security**

SANS Top 20 (cont)

- Limitation & Control of Network Ports, Protocols, & Services: **Network infrastructure as code manages and restricts connectivity**
- Controlled Use of Administrative Privileges: **Virtualization and containerization create sandbox environments. Microservices support least privilege**
- Boundary Defense: **Virtualization creates additional, stronger boundaries**
- Maintenance, Monitoring, & Analysis of Audit Logs: **Virtualization manages and integrates logging**
- Controlled Access Based on Need to Know: **Automation manages access**

SANS Top 20 (cont)

- Account Monitoring & Control: **Continuous integration improves account auditability**
- Data Loss Prevention: **Immutable infrastructure and microservices help concentrate and control dynamic and sensitive data**
- Incident Response & Management: **Continuous integration and infrastructure as code allow quicker identification of and more information on incidents**
- Secure Network Engineering: **Composable infrastructure supports secure architectural design**
- Penetration Testing & Team Exercises**: Infrastructure as code and continuous integration support continuous testing and team collaboration**

Containers key as Cisco looks to "open" data center OS

A rack of servers in CERN's Geneva data center, where it stores 160PB of data on disk and tape drives. *Credit: CERN*

NX-OS Linux utility vital for easily spun up DevOps microservices

RELATED

Review: Container wars: Rocket vs. Odin vs. Docker

First look: Run Docker in Windows Server 2016

What are containers and why do you need them?

on IDG **Answers**

If I buy a Chromebook and can't get to grips with OS can I convert to windows?

https://console.aws.amazon.com/ecs/home?region=us-east-1#/getStarted

Apps Bookmarks DISCOVER LOGO mlbpirates.crx Suggested Sites https://www.tastingr http://osp.od.nih.gov jQuery MultiSelect W Modules Google Read Later Other Bookm

AWS Services Edit Michael Nescot N. Virginia Support

Amazon EC2 Container Service (ECS)

Amazon ECS makes it easy to deploy, manage, and scale Docker containers running applications, services, and batch processes. Amazon ECS places containers across your cluster based on your resource needs and is integrated with familiar features like Elastic Load Balancing, EC2 security groups, EBS volumes and IAM roles.

Get started

Learn more about Amazon ECS



```
- name: Check test pecl module
  shell: "pecl list | grep test | awk '{ print $2 }'"
  register: pecl_test_result
  ignore_errors: True
  changed_when: False

- name: Ensure test pecl module is installed
  command: pecl install -f test-1.1.1
  when: pecl_test_result.stdout != '1.1.1'
```

This is one of Ansible's most powerful tools, but unfortunately Ansible also relies on this for pretty basic functionality. Notice in the above what's happening. The first task checks the status of a shell command then registers it to a variable so that it can be used in the next task. I was displeased to see it took this much effort to do very basic functionality. This should be a feature of the DSL. Puppet, for instance, has a much more elegant syntax for this:

```
exec { 'Ensure redis pecl module is installed':
  command => 'pecl install -f redis-2.2.4',
  unless  => 'pecl list | grep redis | awk \'{ print $2 }\'';
}
```

Windows Container Service

- Window Server Technical Preview
- Azure
- Visual Studio
- Docker
- Powershell

Get-WindowsFeature

Gets information about Windows Server roles, role services, and features that are available for installation and installed on a specified server.

Syntax

```
Parameter Set: Default
Get-WindowsFeature [[-Name] <String[]> ] [-ComputerName <String> ] [-Credential <PSCredential> ] [-LogPath <String> ] [-Vhd <String> ] [ <CommonParameters>]
```