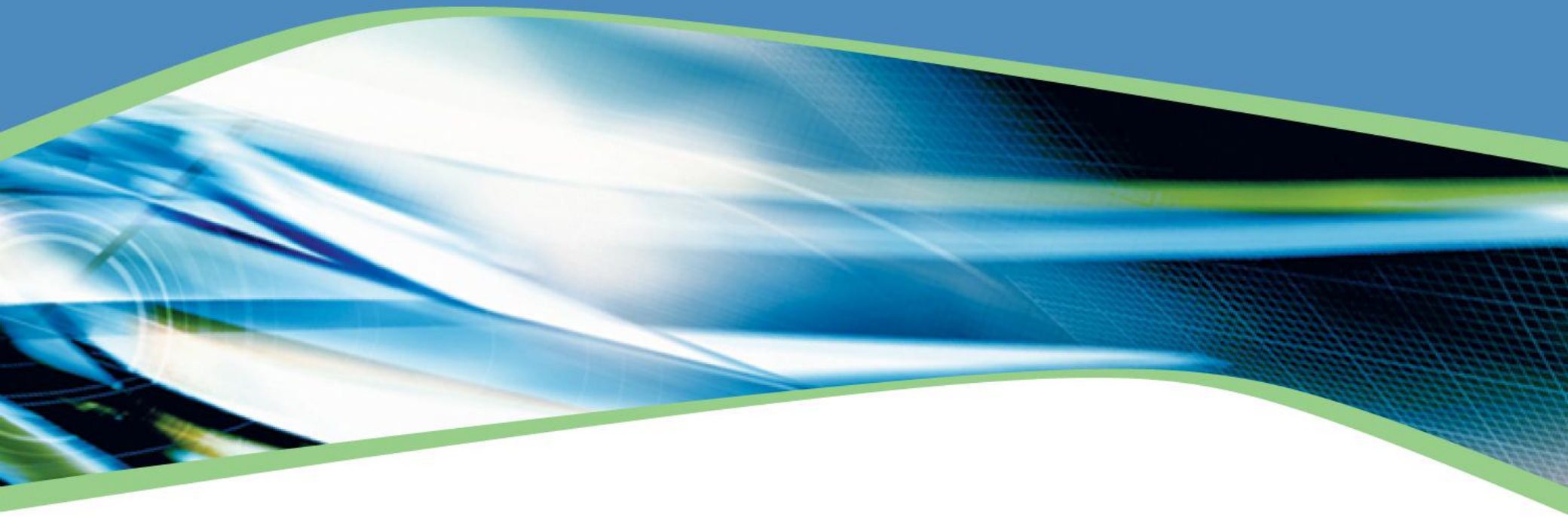
Booz | Allen | Hamilton

OVAL for Mobile Devices



OVAL for Mobile Devices

Session Agenda

- Introduction to Mobile Device Security
  - Policy & Configuration Guidance Landscape
  - Mobile Operating System (MOS) Security
  - Mobile Device Management (MDM) Suites
- MDM-based Approach to Mobile Security Automation
- Development of OVAL Schemas for Mobile Devices
- Android OVAL Schema

Introduction to Mobile Device Security

- Mobile device ubiquity
  - Comparable to laptops in terms of internal hardware specifications, available applications ("apps"), and data/network connectivity
- Limited enterprise control
  - Prioritization of features, visuals and social network interoperability over security and manageability
- We need to fill in the gaps
  - MOS and MDM platforms have matured in the past two years to provide a greater level of security configurability and integrability, but still lack some important capabilities
  - Lack of MOS security functionality creates implications across the entire mobility ecosystem, including limiting capabilities of MDM suites
  - SCAP may be a way to supplement existing security measures and provide additional reporting and continuous monitoring capabilities

Policy & Configuration Guidance Landscape

- Our team analyzed current national- and DoD-level security configuration policies and guidance as well as industry configuration recommendations
  - Configuration guides such as the Defense Information Systems Agency's (DISA) Security Technical Implementation Guides (STIGs), Security Requirements Guides (SRGs) and Interim Security Configuration Guides (ISCGs) provide a basis for achieving consistent and secure configuration implementation across the DoD
  - Perhaps most importantly (to us) they are expressed in XCCDF
- Guidance currently exists for:
  - Android 2.2 (Dell) STIG Bundle
  - BlackBerry STIG Bundle
  - Apple iOS ISCG Bundle
  - Windows Mobile 6.5 STIG Bundle, Windows 7 Tablet
  - General Mobile Device STIG and Draft Mobile OS SRG

Source: http://iase.disa.mil/stigs/net_perimeter/wireless/smartphone.html

Mobile Operating System (MOS) Security

- RIM BlackBerry OS
  - BlackBerry Enterprise Server (BES) and BlackBerry together provide thousands of mature configuration capabilities
  - MDM vendors typically do not provide much support for BlackBerry devices
- Fixmo Sentinel and Sentinel Server Compliance Checker (SCC)
  - Originated from NSA's AutoBerry and AutoBES technology
  - Pre-configured with STIG definitions and references when testing a device/server configuration (though not SCAP-based)

BlackBerry

Mobile Operating System (MOS) Security

- Apple iOS
  - Provides enterprise features out-of-the-box through the Apple MDM API
  - An administrator can deploy one or more *configuration profiles* to manage iOS devices
  - iOS has a special *restrictions profile* which can disable certain device functionality, such as the camera, screen capture, or app store
  - Disablement of Wi-Fi, Bluetooth, and other capabilities are not available through the Apple MDM API and iOS does not provide capabilities/APIs to supplement
- NSA security guide & associated SCAP content*

Supported configurable settings

Accounts
- Exchange ActiveSync
- IMAP/POP Email
- Wi-Fi
- VPN
- LDAP
- CardDAV
- CalDAV
- Subscribed calendars

Passcode policies
- Require passcode on device
- Allow simple value
- Require alphanumeric value
- Minimum passcode length
- Minimum number of complex characters
- Maximum passcode age
- Time before auto-lock
- Passcode history
- Grace period for device lock
- Maximum number of failed attempts

Security and privacy
- Allow diagnostic data to be sent to Apple
- Allow user to accept untrusted certificates
- Force encrypted backups

Other settings
- Credentials
- Web clips
- SCEP settings
- APN settings

Device functionality
- Allow installing apps
- Allow Siri
- Allow use of camera
- Allow FaceTime
- Allow screen capture
- Allow automatic syncing while roaming
- Allow voice dialing
- Allow In-App Purchase
- Require store password for all purchases
- Allow multiplayer gaming
- Allow adding Game Center friends

Applications
- Allow use of YouTube
- Allow use of iTunes Store
- Allow use of Safari
- Set Safari security preferences

iCloud
- Allow backup
- Allow document sync and key-value sync
- Allow Photo Stream

Content ratings
- Allow explicit music and podcasts
- Set ratings region
- Set allowed content ratings

Source: http://images.apple.com/ipad/business/docs/iOS_MDM.pdf

* Available at *http://www.nsa.gov/ia/mitigation_guidance/security_configuration_guides/operating_systems.shtml*

Mobile Operating System (MOS) Security

- Google Android
  - Open-source mobile OS maintained by the Open Handset Alliance
- Unique to the competition for a variety of reasons
  - Google is not in the business* of manufacturing or selling hardware or mobile devices, as are Apple and BlackBerry
  - Various manufacturers—such as Motorola Mobility, Samsung and HTC—each distribute their own set of Android-based devices that can potentially run very different versions of the OS
  - When discussing Android, it is important to differentiate between "Vanilla" Android (or the stock Google kernel and associated software) from the different vendor-released "flavors"

** This may be changing with the Nexus devices unveiled at Google I/O*

Mobile Operating System (MOS) Security

- "Vanilla" Android Security APIs
  - Standard set of device security settings which can be managed by an organization
  - The Device Administration API provides 16 configurable security settings and 3 actionable commands (force password reset, wipe device/restore to factory defaults, and lock device)
  - Additionally, apps can be installed and granted special permissions that allow administrative access to other hardware components not covered by the Device Administration API, such as enabling/disabling Bluetooth and Wi-Fi, and configuring use of Near-Field Communications (NFC)

- Policies supported by the Android-native Device Administration API
  - Password enabled
  - Minimum password length
  - Alphanumeric password required
  - Complex password required
  - Minimum letters required in password
  - Minimum lowercase letters required in password
  - Minimum non-letter characters required in password
  - Minimum numerical digits required in password
  - Minimum symbols required in password
  - Minimum uppercase letters required in password
  - Password expiration timeout
  - Password history restriction
  - Maximum failed password attempts
  - Maximum inactivity time lock
  - Require storage encryption
  - Disable camera

Sources: http://developer.android.com/guide/topics/admin/device-admin.html#policies
http://developer.android.com/guide/topics/security/security.html#permissions

Mobile Operating System (MOS) Security

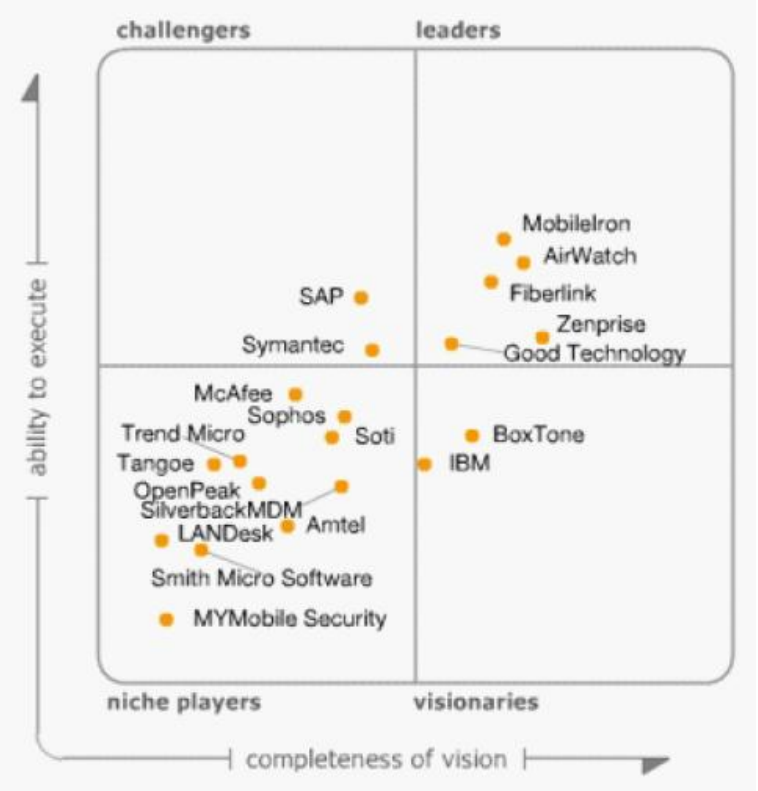
- More on Android "flavors"…
  - Large amount of variance between devices and OEMs/handset vendors
  - Device manufacturers put their own "spin" on the OS/device builds and have been known to strip out (or replace) some vanilla Android functionality to save on resource usage and increase battery life
- The differences between devices makes it difficult for an organization to trust that each device is running at a known security configuration baseline
  - For example, if Android deployments are highly diverse, and the MDM administrator issues a configuration that disables the camera, there will likely be devices that will not implement the policy

Mobile Operating System (MOS) Security

- Filling in the gaps – Enhanced APIs
  - Samsung Approved For Enterprise (SAFE)
  - Three Laws of Mobility (3LM)
- How they work
  - Support extremely granular and powerful system-level security hooks for device and application management
  - APIs lie dormant on the device and activate when the device is associated and enrolled with a compatible MDM
- Supported devices
  - 3LM works with various OEMs, such as parent company Motorola Mobility, HTC, Sony Ericsson, Sharp and Pantech
  - The SAFE API is available on certain Samsung devices
- Using the APIs
  - 3LM provides its own MDM server (3LM Device Manager)
  - Both Samsung and 3LM are working with third-party niche MDM vendors (e.g., AirWatch, BoxTone, MobileIron) to include support for their Enhanced APIs

Mobile Device Management (MDM) Suites

- The MDM market has ballooned tremendously over the past year, with well over 100 participating vendors
- Only a handful of vendors truly differentiate their products with unique capabilities and security features
- Mobile OSs provides limited capabilities, so MDMs are challenged to find ways that make their product stand out



Gartner's 2012 Magic Quadrant for Mobile Device Management (MDM) Software

Source: http://www.gartner.com/DisplayDocument?doc_cd=230508

Mobile Device Management (MDM) Suites

- Push Notification Services (PNS)
  - Push technology allows an MDM server to initiate communication with a client and deliver commands rather than having to wait for the client to "check in"
  - Mobile devices are heavily reliant on access to these Internet-hosted services
  - Apple iOS and Google Android employ Apple Push Notification Service (APNs) and Cloud to Device Messaging (C2DM), respectively, to enable push notifications for system services and mobile apps
  - All MDMs require these services in some capacity and, although feasible alternatives exist to C2DM for modified Android builds, over-the-air (OTA) device management of Apple iOS *must* make use of APNs

Mobile Device Management (MDM) Suites

- Android "MDM" support through Exchange ActiveSync (EAS)
  - EAS provides mobile device users with synchronized access to organizational e-mail, calendars and contacts
  - Well-documented protocol and implementation guidance from Microsoft
  - Many MDMs manage Androids *through the use of* EAS
- Concerns
  - No guarantee of compliance as the EAS protocol simply requires a client to acknowledge receipt of the policy
  - Each vendor implements EAS differently
  - Only products identified in the Microsoft Exchange ActiveSync Logo Program can be trusted to conform to the EAS protocol (and no Android devices currently qualify)

Sources: http://msdn.microsoft.com/en-us/library/hh285606(v=exchg.140)
http://msdn.microsoft.com/en-us/library/hh509085(v=exchg.140)
http://technet.microsoft.com/en-us/exchange/gg187968.aspx

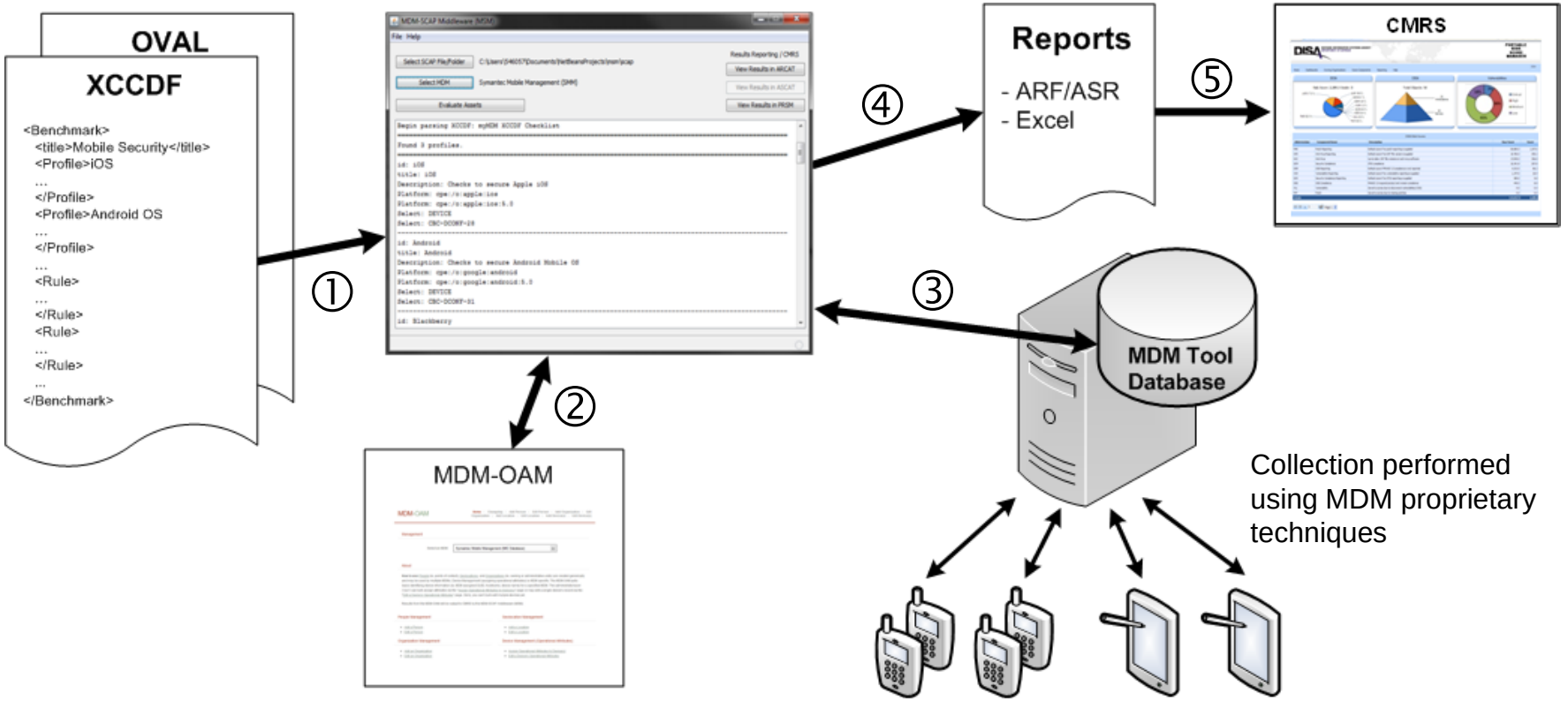
MDM-based approach to Mobile Security Automation

- One of our project's goals is to "demonstrate the ability to leverage configuration guidance, as expressed in SCAP, to extract configuration compliance data of mobile devices from MDM(s)"
- Why MDMs?
  - Doesn't involve development of applications to specific mobile OSs
  - MDMs can support multiple MOS platforms
  - Central repository of device configuration data
  - "Low hanging fruit"

MDM-based approach to Mobile Security Automation

- Approach
  - Analyzed MDMs and how they stored information – found a commonality in that many have SQL backend databases
  - Can query using the *sql57_tests* from the independent definitions schema
- Initial Hurdles
  - Lack of *sql57* support from ovaldi and other scanners
  - Tailoring the SQL query to target a specific device
  - Do we really need to develop another tool?
- Strictly a "proof-of-concept"
  - Prove its possible, capture requirements and lessons learned, and influence vendors/industry to adopt/implement SCAP support into their products
- The MDM-SCAP Middleware (MSM)
  - Java-based
  - Ingests SCAP content, evaluates assets, and outputs results to Excel and (DoD) ARF/ASR
  - Currently supports 2 MDMs

MDM-based approach to Mobile Security Automation



① Mobile device configurations and policies, expressed in SCAP content (XCCDF, OVAL, CPE), instruct the middleware on how to evaluate the security of MDM-managed mobile devices

② Additional device attributes (metadata) are tagged through an external interface

③ Middleware uses OVAL's sql57 independent definitions to query MDM database for asset information

④ Reports are generated from the middleware as an Excel document or in DoD ARF/ASR format

⑤ Results are ingested into DISA's Continuous Monitoring and Risk Scoring (CMRS) System

MDM-based approach to Mobile Security Automation

- Issues
  - Level of effort: database analysis, cooperation from MDM vendors
  - Adaptability: OVAL content scaffold can be reapplied to other MDMs, but the queries are still MDM-specific. Could be better to develop a schema and let vendors determine how to implement the configuration tests
  - Remote / multiple device scanning: how to target an individual device? With SQL, that's a WHERE clause and the MSM has to modify the query
- What we'd like to see
  - MDMs able to read and interpret SCAP content, scan MDM-managed devices
  - Output in a standard format (ARF, ASR, OVAL Results)

MDM-based approach to Mobile Security Automation

- In other words, we'd like to turn this…

```
<ind-def:sql57_object id="oval:com.bah.smmd:obj:1" version="1">
     <ind-def:engine>sqlserver</ind-def:engine>
     <ind-def:version operation="equals">2008</ind-def:version>
     <ind-def:connection_string>jdbc:sqlserver://127.0.0.1\SQLServer:1433;databaseName=
     MDMDatabase;user=username;password=password</ind-def:connection_string>
     <ind-def:sql>SELECT [AllowCamera] FROM [DeviceConfiguration]
     WHERE [guid] = 'device_guid'</ind-def:sql>
</ind-def:sql57_object>
```

- Into this…

```
<android-def:camera_object id="oval:com.bah.smmd:obj:1" version="1">
</android-def:camera_object>

<android-def:camera_state id="oval:com.bah.smmd:ste:1" version="1">
     <android-def:current_status>disabled</android-def:current_status>
</android-def:camera_state>
```

Development of OVAL Schemas for Mobile Devices

- Discussion points
  - What configurations should be included in the OVAL Schemas?
    - For Android, just "vanilla" device configurations, additional APIs (3LM/SAFE), or other hardware/software configurations that may be monitored/enforced by an agent?
    - Is there interest in other(s)? For iOS, should the Mac OS plist definitions be extended? How different will Windows 8 RT (ARM) be from Windows 8?
  - Where should device compliance assessments be performed?
    - On a device, with scarce resources, traversing networks, and potential bandwidth/connectivity issues?
    - Off device, but obtain configuration from the device or MDM?
    - On an MDM, but keeping in mind concurrency and trust issues – when was it last updated/synced and is the policy actually being enforced?