

# Testing Control Systems with Microsoft's Attack Surface Analyzer

{ Digital Bond, Inc  
Michael Toecker, PE  
*[toecker@digitalbond.com](mailto:toecker@digitalbond.com)*

ICSJWG – October 15<sup>th</sup> – 18<sup>th</sup>

Track III



## { Michael Toecker, PE

- ⌘ Professional Engineer
- ⌘ 8 Years in Control System Security and NERC CIP Compliance
- ⌘ Began ICS Work at a Major Power Engineering Firm
- ⌘ Cyber Security and Compliance for Owner-Operator of Critical Infrastructure (Electric Power)

## { Digital Bond, Inc

- ⌘ Founded in 1998
- ⌘ Focused on Control System Security in 2004
- ⌘ Perform:
  - ⌘ Consulting
  - ⌘ Research
  - ⌘ Outreach
- ⌘ Known For:
  - ⌘ Bandolier
  - ⌘ S4 Conference
  - ⌘ Project Basecamp

# About



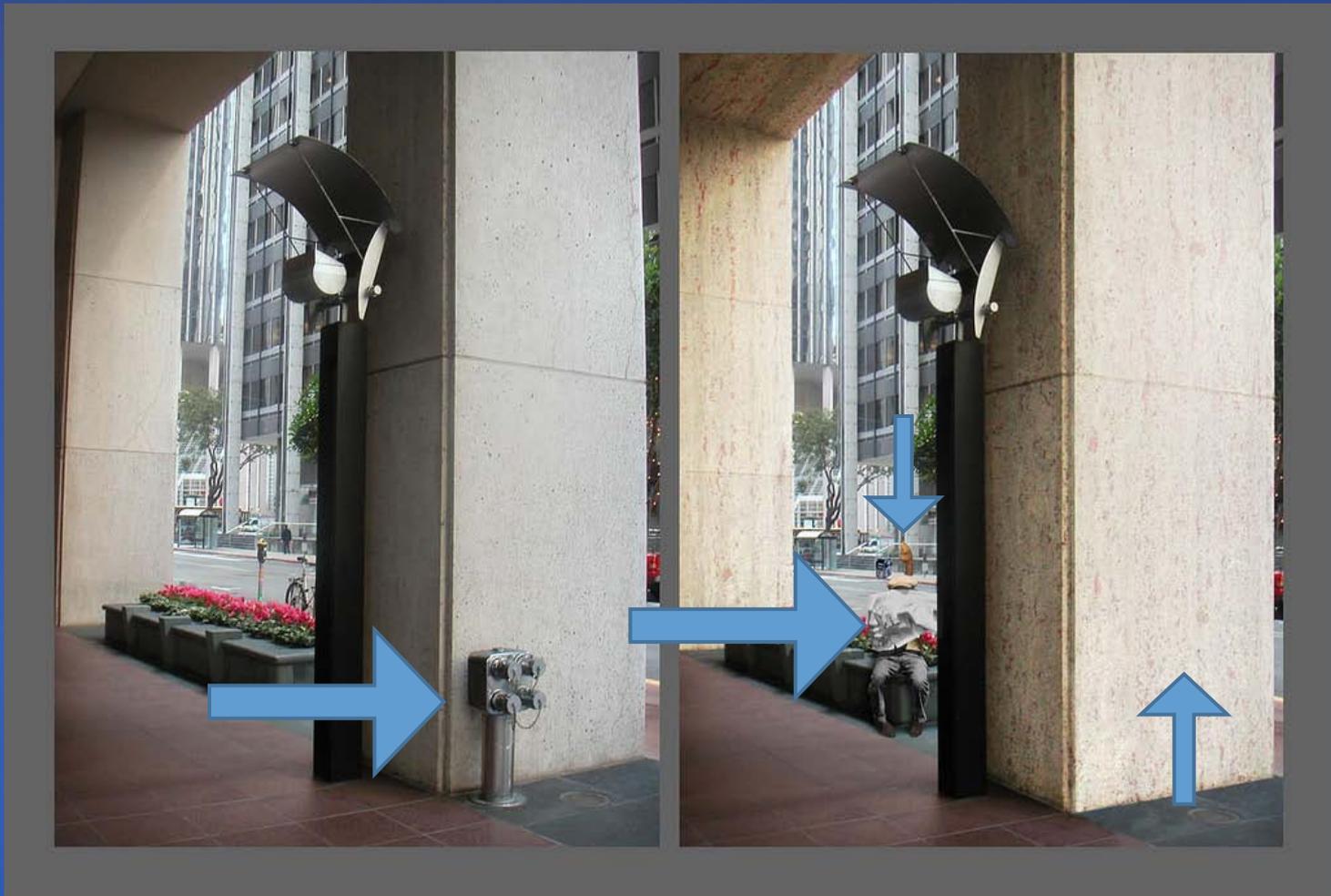
- ⌘ Microsoft Attack Surface Analyzer (ASA)
  - ⌘ Developed internally by the Trustworthy Computing Security Group
  - ⌘ One of several other tools that are used in the Security Development Lifecycle at Microsoft
  
- ⌘ Works with the following Microsoft Operating Systems:
  - ⌘ Windows 7, 8, and Vista
  - ⌘ Windows Server 2008, 2008R2, 2012
  - ⌘ Windows Server Core 2008, 2008 R2, 2012
  
- ⌘ Evaluates security changes that have been made by new software and updates.
- ⌘ Snapshot model, where a set of options is captured and stored
- ⌘ Snapshots are compared to one another, and differences enumerated
- ⌘ Includes Ports, Services, Users, Groups, Registry, others
  
- ⌘ Available at:  
<http://goo.gl/SAmUZ>

# Meet The Tool

- ⌘ **Security Issues** are known to be insecure practices and configuration. Examples:
  - ⌘ Weak Access Controls on directories, files, and registry keys
  - ⌘ Services vulnerable to tampering
  - ⌘ Vulnerable COM and DCOM
  
- ⌘ **Attack Surface** is a listing of changes made to the system since the selected baseline. Examples:
  - ⌘ New Users, Groups, and Group Memberships
  - ⌘ New TCP/UDP Ports in Use
  - ⌘ New Network Shares

# Meet The Tool





Spot the 8 differences?

Some rights reserved by [Sir Frog](#)



# { Security Issues

- ↳ Executables with Weak ACLs
- ↳ Directories Containing Objects with Weak ACLs
- ↳ Registry Keys with Weak ACLs
- ↳ Processes with Weak ACLs
- ↳ Process Threads with Weak ACLs
- ↳ Processes with NX Disabled
- ↳ Services Vulnerable to Tampering
- ↳ Services with Fast Restarts
- ↳ Vulnerable Named Pipes
- ↳ Vulnerable COM Classes
- ↳ Vulnerable DCOM Classes
- ↳ Memory Mapped Sections with Weak ACLs

# { Attack Surface

- ↳ System Information
  - ⌘ Processes, Objects, Modules
- ↳ Service Information
  - ⌘ Services and Drivers (DLLs)
- ↳ ActiveX, DCOM, COM, File Extensions
  - ⌘ New Registered, and Permissions
- ↳ Internet Explorer
  - ⌘ Zones, Silent Handlers, others
- ↳ Network Information
  - ⌘ Ports, Pipes, RPC, Shares
- ↳ Firewall
  - ⌘ Rules, Profiles, Authorized Apps
- ↳ System Environment, Users, Groups
  - ⌘ System Path, New Users and Changes

# Snapshot Components\*



⌘ Regulations require Testing for Each Significant Change

- ⌘ Must ensure that the security of the system is not adversely impacted due to change
- ⌘ Baseline approach is ideal for this, as only the changes are reported in the Attack Surface report

⌘ Examples are:

- ⌘ NERC CIP-007 R1
- ⌘ NIST SP-800-82 (Voluntary)
- ⌘ ISA 99 Patch Management Guidelines
- ⌘ Internal Guidelines and Policies



**Owners of Control  
Systems Have to  
Meet Regulatory  
or Voluntary  
Standard  
Challenges**

# Why Test at All?

## Identify Conditions that can Affect Reliability and Productivity

- ⌘ Review of Changes can identify conditions that could affect production
- ⌘ Case Study: Firewall Rule Changes
  - ⌘ Custom Firewall Rules were in place for an application that allowed specific communications to specific systems
  - ⌘ New Application fully rewrote those firewall rules, removing capability to communicate
  - ⌘ On next reboot, system failed to connect, causing operational workarounds to be activated
  - ⌘ Could not re-establish communication for nearly 48 hours due to operational requirements, technical issues
  - ⌘ Had MS-ASA existed, could have noticed the changes to the firewall rules, and been able to take action before this failure occurred.

# Why Test At All?

- ⌘ Verify Secure Coding Practices are being followed
  - ⌘ Require an explanation for each change in Attack Surface
  - ⌘ Require Changes for Security Issues
- ⌘ Hedge Against Security Mistakes
  - ⌘ ASA Identifies Common Mistakes in Software Installations
- ⌘ Keep Test Code out of Production
  - ⌘ Things like Test Users, Debugging Ports, etc could be identified from ASA Reports



**Help Facilitate  
Security in  
Software  
Development**

# Why Test At All?

## Identify Changes to Systems Over Time



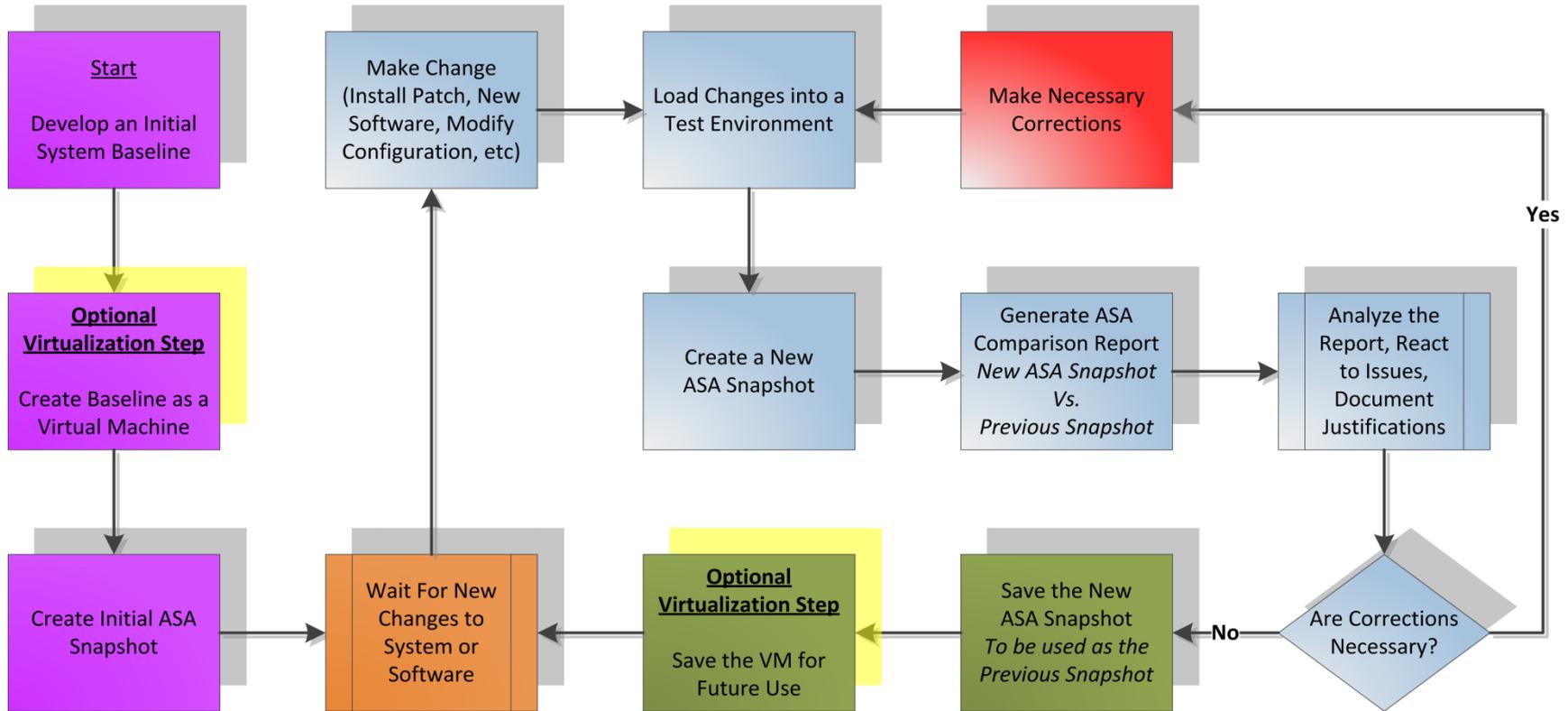
- ⌘ New Software
  - ⌘ Keeps track of changes, and impact of those changes
- ⌘ Updates to Existing Software
  - ⌘ Identifies changes that are outside of usual scope, such as addition of Flash, DLLs, etc.
- ⌘ Removed Software
  - ⌘ Ensures that all pieces of software are removed

# Why Test at All?

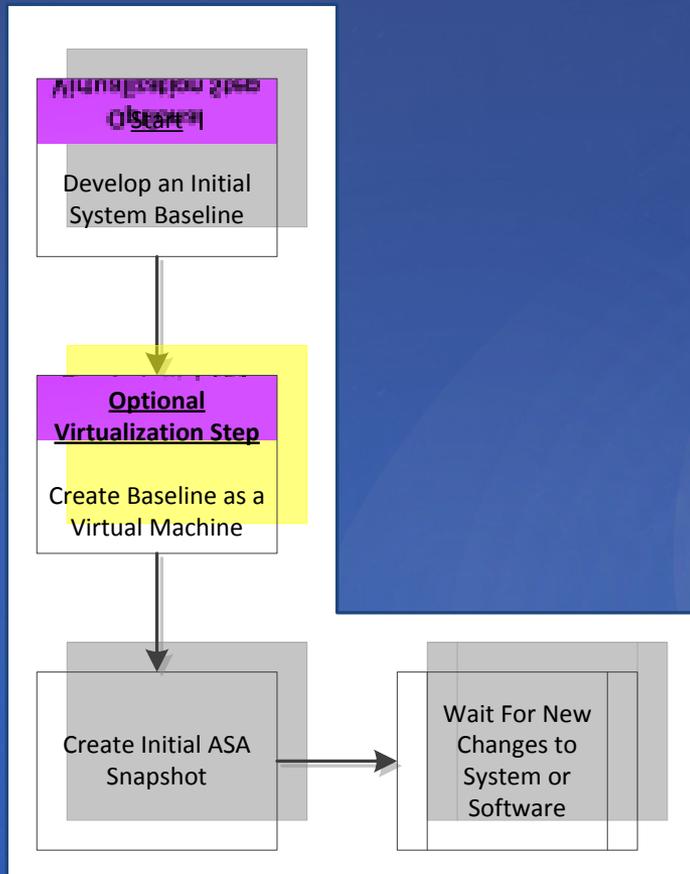
- ⌘ Outside Service Providers and Contractors may improve systems, and add new components
- ⌘ Often require some 'modification' of existing security controls or operating system configuration to work
  - ⌘ Firewall Rules
  - ⌘ Adding of Users
- ⌘ Identifies changes that go above and beyond the scope of the authorized work

**Ensure Work is Completed to Specification**

# Why Test At All?



# The Testing Process



## Initial Baseline

- ⌘ This is a hardened operating system, locked down to the minimum necessary to have a functioning operating system
- ⌘ Many such images exist already (i.e. DISA, FDCC, etc), and can also be developed internally

## Optional Virtualization – Coming up...

### Initial Attack Surface Analyzer Snapshot

- ⌘ Snapshot must be of the hardened OS
- ⌘ Before even standard programs are installed
- ⌘ Domain Membership would be fine, as this often sets many hardened settings

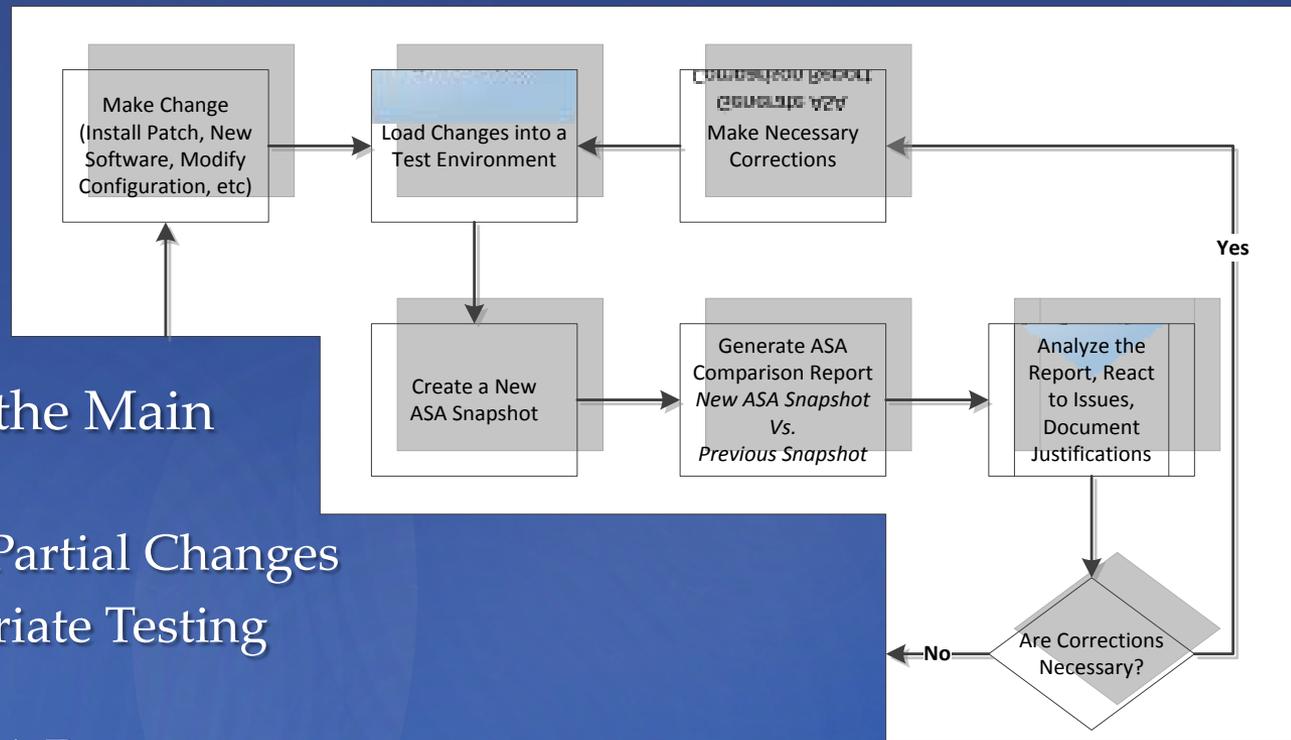
### Wait for Changes

- ⌘ Basically a pause point, where changes come in and get fed into the main process.

# Setup Activities

## & Important Parts of the Main Process

- ⌘ Full Changes vs. Partial Changes
- ⌘ Using an Appropriate Testing Environment
- ⌘ Analyzing the ASA Report



# Main Process

- ⌘ Some changes will not be detectable by ASA without taking extra steps
- ⌘ Examples:
  - ⌘ Applications that communicate with devices often need a device IP Address to enable detectable functionality
  - ⌘ Often includes OPC Servers and DNP Servers, but others exist
- ⌘ Not fully making a change can affect important areas of the Final ASA Report, such as:
  - ⌘ Listening and Established Ports and Services
  - ⌘ Running or Stopped Services
  - ⌘ Running Processes
- ⌘ Requires some knowledge and experimentation with change
  - ⌘ Or vendor assistance?

# Full Change vs Partial Change

⌘ The Attack Surface Analyzer report is only as accurate as your Test Environment's Accuracy

⌘ Major Issues that Affect Accuracy

- ⌘ Not Running appropriate Applications while taking a Snapshot
- ⌘ Not Communicating with Devices or Other Systems
- ⌘ Using Administrator Level accounts instead of Limited Permission Accounts

$$\left\{ \begin{array}{l} \text{ASA-Report}_{\text{Accuracy}} \\ = \\ f(\text{TestEnv}_{\text{Accuracy}}) \end{array} \right.$$

⌘ Mitigation

- ⌘ Know how your test environment differs from an actual production deployment
- ⌘ Gauge the impact of these differences, and determine if the cost of fixing the difference provides a suitable reward

# Testing Environment

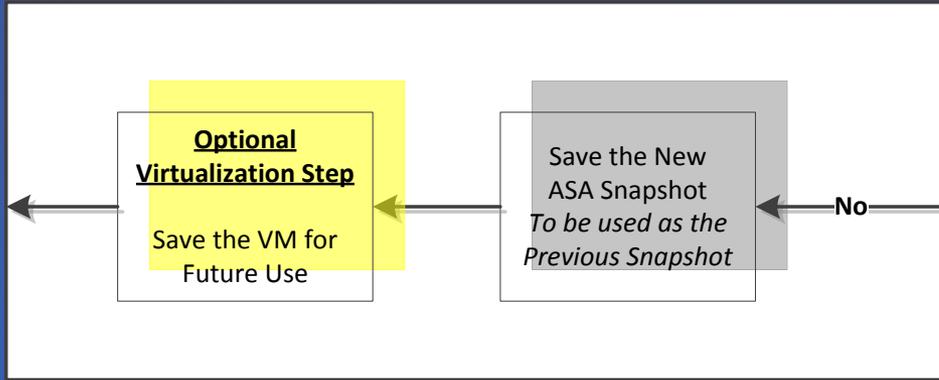
- ⌘ What sections are you most concerned with?
  - ⌘ For NERC CIP, most concerned with new listening ports, new users, removing logging and auditing
- ⌘ What sections are you least concerned with?
  - ⌘ For Automation pros, maybe less concerned about Internet Explorer if it's not enabled in my environment.
- ⌘ What sections can you safely reduce in severity due to other controls, and which ones will bypass existing controls?
  - ⌘ Granting Everyone permissions often bypasses controls
  - ⌘ Firewall rule permissiveness may be less of an issue due to a perimeter firewall
- ⌘ Make this a defined process.

# Analyzing the ASA Report

- ⌘ Questions should be asked about each change
  - ⌘ New Listening Port? Why is it there, what process is bound to it? Is that a legitimate process?
  - ⌘ New Network Share? Why does it have Everyone Permissions? Are those Permissions necessary for the application?
  - ⌘ New 3<sup>rd</sup> Party Applications? Why does my control system need an outdated version of Flash.OCX? Or Adobe Reader 7.0?

- ⌘ **Honest Opinion** – Sending this report to a vendor for their justification is a valid tactic, as they should know what changes are necessary and which aren't. Plus, maybe they will start using the tool to avoid issues.

# Analyzing the ASA Report



⌘ Saving the new snapshot is necessary

⌘ This is your new Baseline, used for subsequent changes

⌘ Establishes a chain, where all changes can be examined back to the original Baseline

# Change Closeout

- ⌘ Testing has a cost, both in time and in materials
  - ⌘ Personnel to perform the tests may have other responsibilities
  - ⌘ License Dongles, Hardware, Software, Operating Systems all cost money
- ⌘ Virtualization can reduce these costs by minimizing the hardware and software commitment
- ⌘ Virtualization also lowers the cost of owning certain test environments
  - ⌘ Windows and other PC based operating systems can be easily virtualized
  - ⌘ Devices are not virtualization capable (They also aren't MS-ASA compatible anyway.)
- ⌘ Use Virtualization to manage and store previous versions of your test systems
  - ⌘ Allows investigation and rollback
  - ⌘ Re-imaging a system back to the Hardened Baseline is a mouse click

# Optional Virtualization

- ⌘ No Support for Windows XP or Older
  - ⌘ Major stopping point for use in existing automation, but developers will find handy
- ⌘ Captures Only What it Currently Sees
  - ⌘ If you don't run an application, changes associated with that application won't be recorded.
- ⌘ System must be in a near production state
  - ⌘ Applications we are concerned with must be running
  - ⌘ Device Communication should be Active/Simulated
- ⌘ Requires a Secure Baseline to Start
  - ⌘ Without the initial secure baseline, changes made by the software will be missed
  - ⌘ Allows comparison to a "Gold Standard"

# Concerns and Issues

## Report Summary

[Explain...](#)

### Report Details

<b>Report Analyzer Version</b>	1.0.0.0
<b>Baseline Data File</b>	C:\Users\mtoecker\Attack Surface Analyzer\Baseline Test - Initial Scan.cab
<b>Baseline Collection Time</b>	2012-08-14T13:04:07
<b>Baseline Scanner Version</b>	1.0.0
<b>Product Data File</b>	C:\Users\mtoecker\Attack Surface Analyzer\After Some_Vendor-ABB Install.cab
<b>Product Collection Time</b>	2012-08-14T13:30:28
<b>Product Scanner Version</b>	1.0.0

### System Details

<b>Machine Name</b>	WIN-0HCVFPP7ARN
<b>OS</b>	Microsoft Windows 7 Enterprise
<b>OS Version</b>	6.1.7601
<b>Service Pack Version</b>	1.0
<b>Computer Role</b>	Standalone Workstation

# Sample ASA Report



{ Thanks,  
Mike

Questions?



↳ Digital Bond's S4  
Conference in Miami  
Beach, January 2013

↳ Speakers Include:

- ⌘ Travis Goodspeed
- ⌘ Billy Rios and Terry  
McCorkle
- ⌘ atlas Of d00m

↳ Details on  
[DigitalBond.com](http://DigitalBond.com)



# More Research at S4

