Global Information Assurance Certification Paper

Implementing a PC Hardware Configuration (BIOS) Baseline

GIAC (GSEC) Gold Certification

Author: David R. Fletcher Jr., 6fletch9@gmail.com
Advisor: Rich Graves

Accepted: October 13, 2013

Abstract

High level operating system features such as patch management, full disk encryption, virtualization, and malware protection are increasingly reliant on properly configured Basic Input Output System (BIOS) firmware settings and support. Varying configuration settings complicate the implementation process and subsequent troubleshooting sessions. This paper presents a solution to these issues through implementation of a hardware configuration policy, a BIOS firmware features baseline, and hardware configuration standards. This is accomplished by folding hardware selection and configuration into comprehensive lifecycle, operations, and change management programs to ensure predictable support for required features. To support the development of necessary documentation a survey of typical BIOS firmware configuration options is presented. Security implications for each of these options are explored to identify settings that are both beneficial and detrimental to security. Finally, vendor options and support for BIOS firmware settings automation are explored.

Introduction

This paper provides a road map for implementation of the recommended phases identified in NIST SP 800-147, BIOS Protection Guidelines. NIST outlines a five phase process for organizations to follow in order to properly manage BIOS firmware and settings (Cooper, Polk, Regenscheid & Souppaya, 2011). However, they do not identify specific actions that an organization should take as a comprehensive management program nor do they address specific settings that may enhance or weaken overall system security. In an attempt to bridge that gap this paper will provide recommendations to develop a BIOS standardization policy, identify a BIOS baseline, and create BIOS settings standards. The overall goal will be integration of BIOS standardization into existing acquisition, life-cycle management, and operations programs. Vendor automation support for BIOS firmware deployment and settings will also be explored in an effort to streamline the process.

This paper is not a panacea and cannot cover all of the various combinations of BIOS settings produced for each manufacturer and model. Examples of features and settings that both enhance and reduce security, functionality, and/or performance will be presented. The intent of these examples is to provoke the reader to investigate the particular makes and models employed in their organization. In addition, the reader is encouraged to research options for each employed BIOS package to define appropriate standards based on baseline features required by the organization.

The automation examples presented in this paper are focused on Microsoft Windows operating systems in the enterprise. These systems currently have the largest market share and as such the greatest support for automation. Since the firmware settings are independent of the operating system, the principles presented will still hold true for other operating system software. The greatest limitation for non-Windows systems is available manufacturer support for automation on those specific platforms.

David R. Fletcher Jr, 6fletch9@gmail.com

Background

NIST SP 800-147 recommends a process for proper BIOS firmware and settings management throughout the life of a computer (Cooper, Polk, Regenscheid & Souppaya, 2011). Other organizations such as the Center for Internet Security (CIS), the Defense Information Services Agency (DISA), and the National Security Agency (NSA) provide comprehensive guidance on hardening network devices and operating systems. In the interest of avoiding duplication of effort, the hardening guides published by the aforementioned organizations were reviewed for BIOS firmware settings recommendations. In the guides reviewed, the most common settings recommendations were restriction of the boot order to the primary operating system drive and that a BIOS administrator password is assigned. This is likely due to the myriad options that are presented as new motherboard and computer makes, models, and features are released. In addition, the intended role of the computer in the organization must be factored into configuration decisions. To understand the impact that BIOS configuration settings have on computer performance and security the role that BIOS firmware plays in a computer's operation must be explored.

A computer's motherboard provides boot support and access to low level hardware settings through its BIOS firmware. At the most basic level the BIOS firmware loads device drivers, coordinates hardware self-tests, and prepares the computing environment for execution of an operating system. During this process, user selectable options are interpreted to determine the hardware and features that will be made available to the booting operating system (Mueller, 2013). As computing technology has advanced new features have been made available within the BIOS to support higher level functions which enhance processing capability, security, and increase system flexibility. Many of these settings can be used to protect the computing device and its stored data from malware, theft, and unauthorized access. However, as is usually the case, a small subset of options contradicts sound security practice.

The Visible Ops Handbook and Visible Ops Security both advocate strong configuration management to produce a high-performing information technology

organization. Both books employ a four phase approach to improving an organization's performance. The Visible Ops Handbook deals with Information Technology Infrastructure Library (ITIL) implementation and producing a controlled work environment (Behr, Kim & Spafford, 2005). Visible Ops Security deals with integration of security into daily operations (Kim, Love & Spafford, 2008). Both approaches have key elements that can be used to support implementation of the NIST process. By integrating the phases identified in NIST SP 800-147 into the computing lifecycle an organization will realize benefits in the form of:

- An added layer in their defense-in-depth strategy
- Increased availability through reduced configuration variance
- A predictable operating environment

The end goal of this effort is to develop a strategy to manage BIOS firmware settings and versions and to develop supporting documentation. Before this can be accomplished it would be wise to survey the landscape to identify the options and challenges that lie ahead.

Evaluating BIOS Settings

The first phase of Visible Ops Security suggests that an organization gain awareness of the current situation and integrate into change management (Kim, Love & Spafford, 2008). Evaluation of the current landscape with regard to BIOS firmware settings corresponds to this phase. Standardization would be a very difficult task to undertake without a comprehensive knowledge of the existing landscape.

As a result, the implementing organization should evaluate the computer makes and models that have already been deployed (or are likely to be purchased) within their enterprise. This will allow the Change Advisory Board (CAB) to perform research and make informed decisions about the technologies that must be supported and the configuration options that must be evaluated. To illustrate the type of information an organization is likely to encounter examples are presented categorically as Standard, Security Enhancing, Application Supporting, and Potentially Hazardous.

As these options are being evaluated it is important not to lose sight of the context in which the hardware will be employed. For instance, a standard network workstation will have different requirements than a server, a standalone workstation, or a telecommuter workstation. Without proper consideration, the functionality of deployed server platforms may be reduced or a telecommuting worker may be left with degraded capability out of simple oversight. Another issue to consider is determining the level of effort that will be placed into securing the equipment based on this context. Server and desktop BIOS can typically be reset by changing a physical jumper on the motherboard (Mueller, 2013). While a closed case is a deterrent for a normal user this is likely not the case for someone determined to reduce the security posture of the asset.

Standard BIOS Settings

1.1.1. BIOS Administrator Password

The BIOS administrator password restricts access to the BIOS setup utility. This functionality is also referred to as the supervisor or setup password on some firmware (Mueller, 2013). The BIOS administrator should have a password assigned if there is any hope of maintaining a standard configuration. The CIS, DISA, and NSA hardening guides were in agreement that this setting should be included as a basic operating system hardening step.

This password should meet the standard password complexity requirements of the enterprise depending on the support the firmware provides. It is a good idea to test any selected password on a representative sample of computer and firmware combinations prior to implementation. This will ensure that any shortfalls can be addressed as early as possible. Support for password length and character set vary a great deal across manufacturers, makes and models. As such, it may be impossible to enforce long password lengths or use of special characters universally.

Another consideration for the BIOS administrator password is how the password will vary across the enterprise. Standardizing the password across functional lines allows lower level administrators to easily manage systems but a single compromise requires a password change for all of the systems in the same functional group. Completely random

passwords present a challenge because management of credentials and support for automation become nearly impossible. A potential middle ground might be a common password root with a seemingly random suffix such as an organizational indicator, part of the serial number, or identification tied to the platform such as the MAC address. When making this decision, keep in mind that the BIOS administrator password may be easily subverted by using a jumper on the motherboard.

1.1.2. Boot Order

The boot order identifies to the computer which devices might contain an authorized bootable partition. The computer will search each identified device sequentially until it finds an active bootable partition. At that point the computer will attempt to boot the operating system or utility found on the device (Mueller, 2013). Unless there is a compelling need, the only boot device identified and active in the boot order should be the partition on which the primary operating system is installed. This setting is another that had overwhelming support in the CIS, DISA, and NSA hardening guides.

1.1.3. Integrated Devices

Modern firmware provides a great deal of flexibility with regard to configuration of the integrated devices on the motherboard. Most modern computers allow the BIOS administrator to determine exactly which device functionality to expose to the operating system and consequently the user. The CAB should review the list of integrated devices included in a particular make and model to determine which should be disabled. Some of the devices that are likely to be encountered include:

- IDE/SATA Ports – Unless the computer model in question is likely to have additional disks installed it is a good idea to disable any unused controller ports.
- USB Ports – Some computer models allow you to disable USB ports individually while others allow you to disable groups of ports. If USB devices are a concern for the organization it may be a good idea to disable all but the minimum number of ports required. Realize that this will only stop a casual violator as port replication is simple and inexpensive using a USB hub. This measure should only

be considered an enhancement of a more rigorous data loss prevention program.

- Unused Ports/Bays – Modern computers can be ordered with a plethora of integrated devices that may go unused or may violate company security policy. It would be a sound decision to disable any device that may not be necessary. Devices that are likely to fall into this category include; smartcard readers, fingerprint readers, serial ports, parallel ports, FireWire ports, and media readers such as SD, Micro SD, Memory Stick, and ExpressCard. This is another area where testing is prudent. For instance, disabling the media bay on a laptop may render the CD/DVD-RW drive unusable. However, these devices are meant to be hot-swappable and the media bay will accommodate a hard drive as well. As with USB Ports, this measure is an enhancing feature for data loss prevention.
- Wireless Capabilities – Laptop computers are likely to have a heavy wireless presence unless ordered without these devices. Typical BIOS firmware will allow the administrator to disable these features individually. If there is no corporate wireless network and no authorized telecommute capability it would be wise to disable WLAN and WWAN features. Likewise, if no Bluetooth devices are authorized in the enterprise this adapter can be disabled as well. If a device is identified as truly standalone then all three options should be disabled to prevent network communication.
- Network Adapter – Computers with multiple network adapters (typically server platforms) should have any unused network adapters disabled. In addition, standalone devices should have wired network adapters disabled as well.
- Modem – While modem use grows less and less common, many laptop makes and models still come with an integrated modem as a standard option. Unless there is a demonstrated need for a modem connection, this device should be disabled.

Security Enhancing Features

1.1.1. Wireless Switching

Wireless switching prevents network bridging at the hardware level. Once link is detected on the wired network adapter the wireless adapter is automatically disabled.

This feature is an excellent measure to prevent backdoor network access. Effects on higher level technologies such as operating system level or third party wireless switching prevention should be determined prior to adoption.

1.1.2. Execute Disable/Enhanced Virus Protection

Intel and AMD have both taken measures to attempt to provide enhanced exploit protection through the Intel Execute Disable (XD) and AMD Enhanced Virus Protection (NX-bit) options. Neither technology is meant to detect or eradicate viruses or malware. Instead, it allows the operating system to mark pages of memory as non-executable. This capability prevents some buffer overflow exploits by preventing execution of instructions on marked memory pages (Kubicki, 2004). This feature will not provide comprehensive virus or malware protection. However, it should be considered yet another layer of protection in exploit defense and should be enabled.

1.1.3. Intel Trusted Execution Technology (TXT)

In response to "blue pill" style virtualization attacks, Intel developed their Trusted Execution Technology (Haletky, 2010). This feature, used in conjunction with the Trusted Platform Module (TPM), provides a Root of Trust for virtual machine execution. A hypervisor or operating system that supports Measured Launch Environment (MLE) can be used to TXT to ensure the integrity of the boot environment. The MLE records hashes of the known good hypervisor in the TPM for comparison on launch based on the Launch Control Policy. This technology is primarily meant to protect the data center ("Intel trusted execution," 2012). An organization with a virtualized data center that wishes to use virtualization technology extensions, employs an MLE supported hypervisor, and has TXT capable hardware should enable this feature.

1.1.4. Trusted Platform Module (TPM)

Trusted Computing Group defines the TPM as a computer chip that can securely store artifacts used to authenticate the platform. This capability can provide numerous services such as boot and system integrity validation or secure key/signature storage ("Trusted platform module," 2013). When running a Windows operating system, the encryption keys employed with Microsoft BitLocker are stored in the TPM. If an

organization is entertaining advanced boot integrity measures or full disk encryption the TPM and resulting implications of using it should be explored fully. The Trusted Computing Group website provides a great deal of information on the applications supported by the TPM.

As an example, when using Microsoft BitLocker in the enterprise, the BIOS Settings change procedure must be adjusted to take into consideration the protections that BitLocker affords to the BIOS. Certain settings changes will require an administrator to provide the BitLocker recovery key to authorize and subsequently recover the data from the encrypted drive. As a result, when modifying the BIOS, procedures will have to be adjusted to suspend BitLocker protection prior to making any changes. In addition, recovery key protection becomes a concern due to the fact that the recovery key itself is sensitive. Once key compromise occurs, the encrypted disk can be decrypted using this key on another computer. If this occurs the compromised key should be deleted from the volume and a new key must be generated and backed up (Jumelet, 2010). Finally, support considerations must be taken into account as users will encounter new errors and recovery key prompts may cause a work stoppage.

1.1.5. Hardware Recovery/Protection Services

Some BIOS firmware is delivered with hardware recovery and data protection services support pre-installed. This feature allows the organization to pay for a subscription service which provides a number of capabilities to the subscriber. The features provided include lost/stolen device location services, device disable, remote wipe, and device recovery. This can come at a considerable cost for an enterprise as rates for one such service currently start at $40/year. However, this may be a reasonable cost to provide additional protection to executives who may be targets of intelligence gathering efforts. If data loss is a concern, this service should be implemented to augment other existing data protection technologies such as data loss prevention and full disk encryption.

Options available at the hardware level vary with the make and model of computer. The implications of different options should be explored prior to selection. For instance, some implementations allow the option to simply turn off the service or

completely disable it on the device. In some cases, the disable option is irreversible without a firmware update.

Application Supporting Features

1.1.6. Unified Extensible Firmware Interface (UEFI) Support

Newer BIOS Firmware, typically produced in 2011 or later, provides UEFI capabilities. These capabilities extend standard BIOS features and provide boot level utilities and operating system support features. The goal of UEFI is to modernize the boot process and standardize the boot environment. UEFI was born out of the Extensible Firmware Interface (EFI) standard which was developed by Intel and has been around since 2000. Until 2008 the UEFI firmware standard was not widely deployed. The key drivers behind adoption were boot support for disks with capacity greater than 2.2 TB and the GUID Partition Table (GPT) scheme. This new partition scheme is capable of supporting up to 9.4 ZB which more than accommodated new drive capacities. Legacy PC BIOS cannot support either of these features on bootable disk volumes (Mueller, 2013).

In addition to GPT support, the new firmware is written in either 32 or 64-bit code versus 16-bit code which is used in standard PC BIOS. UEFI firmware may also include extensive onboard diagnostics, live update features, on-board hard disk cloning, advanced overclocking features, and embedded applications. One catch is that the operating system running on the hardware must include UEFI support in order to boot from a GPT partition. This is yet another area to extensively explore before adoption. In a Microsoft Windows enterprise UEFI support is not available on 32-bit versions of the operating system. In addition, only 64-bit versions produced after Microsoft Vista Service Pack 1 support UEFI features (Mueller, 2013).

The UEFI boot feature takes on different names based on the BIOS firmware in question. On some Hewlett Packard computers it is labeled as UEFI Boot Mode while on some Dell computers it is identified in the Boot List option setting.

1.1.7. Virtualization Technology

This BIOS firmware setting provides hardware support to virtualized operating

systems and applications. Both AMD and Intel claim performance and security benefits from enabling hardware assisted virtualization technology. This setting should be considered if the organization is entertaining desktop or application virtualization. In addition, if server platforms are virtualized this setting may enhance support on the physical hardware running the virtual machine hypervisor.

If the organization is risk-averse, it may be wise to investigate the implications of "blue pill" type malware which takes advantage of the VMX instruction set to insert itself between the operating system and hardware. While the capability has a working proof of concept, it has yet to be widely adopted due to exploit complexity and ease of exploitation by other means (Mcmillan, 2011). If Virtualization Technology is enabled, it would be wise to keep abreast of advancements in "blue pill" malware. Use of this feature could represent a tradeoff between security and performance.

1.1.8. Wake On LAN

Wake On LAN is a feature used to recover a computer from a powered off or sleep state to boot the computer's operating system and enable network activity to take place. This feature is used by patch management systems like Microsoft's System Center Configuration Manager to wake computers for patch installation during off-peak hours. Inconsistent application of this setting can contribute to poor patch installation metrics and cause confusion in troubleshooting patch management systems. An enterprise that employs a configuration management system in this fashion should have Wake On LAN universally enabled.

If an organization is contemplating employing a configuration management system that has Wake On LAN capability then network level implementation details and impact should be given first consideration. Some systems rely on broadcast traffic to deliver magic packets (used to wake the computer). These packets will be filtered by routers (which restrict broadcast domains) and may prohibit Wake On LAN from working properly.

Also, when implementing this feature it is important to ensure that the BIOS firmware employed boots the computer properly. Some BIOS firmware does not provide

the user an option for specifying the boot procedure and just follows the standard boot sequence. In contrast, other firmware requires the user to select a boot preference (such as standard boot sequence, boot to network, etc.). This requirement is not universal and must be investigated on each unique make and model.

1.1.9. Wake Up Timer

This feature takes many names across BIOS implementations. However, the purpose of the Wake Up Timer is to power on the computer based on a time and/or date that the user specifies in the BIOS. Where a computer model may not support Wake On LAN this feature can act as a stop-gap measure to ensure that patch management systems can more reliably contact the computer. This feature is should be considered for use in the absence of Wake On LAN.

1.1.10. Wake On AC

This feature is used to turn the computer on and boot the operating system after a power loss situation has occurred. This is another feature that can assist in increasing reliability of patch management. Like Wake On LAN, some BIOS firmware implementations allow the user to specify the boot sequence after power restoration. This feature should be enabled if the organization expects to be able to connect to computers at all times.

Potentially Hazardous Features

1.1.11. Pre-Boot Content/Application Access

Many computers ship with BIOS firmware that allows pre-boot content and/or application access. Web browsing, calendar, and e-mail access appear to be the most commonly supported features. These environments lack the full support that an operating system can provide, can be configured for unauthenticated access, and don't have a great deal of technical information that can be easily found.

Exploring Hewlett Packard's QuickLook 3 software reveals that e-mail and calendar data from Outlook is cached for access from the pre-boot environment. Access can be protected with a pin or windows credentials and the data is encrypted using 192-bit AES. Data is stored on a separate partition on the hard disk to avoid conflicts with

existing data at rest encryption solutions ("Hp quicklook," 2009). Of primary concern is circumvention of traditional forensic artifacts such as modify, access, and create timestamps as well as repudiation capability through standard credentialing features.

Another Hewlett Packard tool, QuickWeb, allows access to the internet using a pre-boot environment web browser. This tool carries the same problems as QuickLook but also provides network access to the computer ("Hp quickweb," n.d.). Browser and e-mail exploits plague applications that are updated continuously. This feature places a web browser in the computer's firmware which is likely to get updated much less frequently. If the application is updated frequently, it will increase the overall footprint for application updates and likely complicate the process. Exploitation would place compromised software in the firmware for the computer.

Policy, Baseline, and Standards Development

After surveying the landscape of BIOS options, educated decisions can be made as to which BIOS firmware features will or will not be supported and which settings are appropriate. Armed with this information an appropriate policy, enterprise baseline, and accurate configuration standards can be developed.

Enterprise BIOS Standardization Policy

The first document that should be developed is a policy outlining, in general terms, how BIOS standardization will be implemented across the enterprise. A policy represents management's endorsement of the effort and gives the implementers the authority to complete the work. This document should not delve into the specifics of implementation, such as individual settings, but instead should direct the overall accomplishment and direction of the program. In addition, the policy should be forward looking with a goal of being in place for several years. Finally, all policies should address the same set of common elements. These elements include; stating the purpose of the policy, identifying related documents, providing background discussion, determining the scope of the policy, outlining the actual policy details, identifying responsible parties, and identifying actions that must be taken (Sans security essentials, 2013). A sample policy containing all of these elements can be seen in Appendix A.

Since computers are touched by many different offices throughout an organization, the BIOS standardization policy will cover the full lifecycle of a computing device. The process outlined in NIST SP 800-147 recommends managing the BIOS through 5 phases; provisioning, deployment, operation and maintenance, recovery, and disposition (Cooper, Polk, Regenscheid & Souppaya, 2011). An additional acquisition phase should be included to ensure that new platforms provide a standard set of features which will be identified in the enterprise baseline.

1.1.12. Acquisition

The Visible Ops Handbook advocates that investment early in the IT lifecycle results in greater performance as an organization and lower cost for defect correction (Behr, Kim & Spafford, 2005). With regard to computer configuration there is no earlier time in the IT lifecycle than acquisition. During this phase, candidate device characteristics will be compared to the enterprise minimum baseline to ensure that a minimum set of hardware features is available. In addition, if the organization intends to use automation to maintain BIOS firmware and settings, manufacturer support for this capability should be evaluated. This will ensure that any newly acquired equipment will integrate into the computing environment as seamlessly as possible. It is likely that multiple platform specific baselines will be necessary to ensure comprehensive coverage. For instance, a baseline might be required for laptops, desktops, standard servers, blade servers, computing appliances, virtual machines, etc.

This phase is particularly critical because it is typically difficult and/or costly to add integrated features after a computer has been purchased and delivered. As an example, if an enterprise performs patching during non-business hours and wishes to take advantage of power saving features then Wake On LAN is likely necessary. A platform that does not support Wake On LAN will not be able to support this process and will require either additional hardware or exemption from the process.

1.1.13. Provisioning

After baseline conformance has been confirmed and the computer has been purchased it must be provisioned for use on the network. This represents the earliest time that the organization has to ensure that the device firmware version and settings are

configured to standard. New computers should be configured using settings for the make, model and BIOS revision from a comprehensive build library (Cooper, Polk, Regenscheid & Souppaya, 2011). Ideally this process should be automated but only after the manual configuration process has been fully documented. At this point, the organization must identify who is required to perform and confirm provisioning configuration. In most cases, this should be the department responsible for lifecycle management and delivery of assets to their intended recipients.

1.1.14. Operation and Maintenance

Once a computer has been placed into service, configuration standards are likely to change and devices are likely to malfunction. The operation and maintenance phase places emphasis on ensuring that the prescribed configuration standards are enforced during the device's in-service period (Cooper, Polk, Regenscheid & Souppaya, 2011).

With regard to evolution of firmware configuration standards, management must identify the department tasked with implementation. In most situations this will be the IT operations department. First, thorough testing should be accomplished prior to deployment. Next, using the manually documented standard, the operations department should employ automation tools such as third-party BIOS management tools, Active Directory Group Policy, and/or in-place configuration management systems to deploy changes with as little disruption possible. These same tools should be used to audit configuration settings to ensure that the standard is not being violated. Once a violation has been identified it should be reported and corrective action, as outlined in this policy, should be applied in a standard fashion. Finally, any identified violation should be remediated prior to returning the asset to service (Cooper, Polk, Regenscheid & Souppaya, 2011).

In the event of equipment malfunction, especially those malfunctions that cannot be remotely remediated, conformance with the BIOS firmware settings standard should be verified. If possible, this confirmation should be performed using a diff style tool to compare the expected settings to the settings as configured. This will streamline the process and reduce the opportunity for a configuration setting to be missed.

Conformance validation should also be identified as a standard practice whenever a computer requires physical administrator presence. By performing these checks, configuration variance is identified early on in the troubleshooting process. These checks also serve to validate that automation tools used for deployment, remediation, and compliance checks are working properly.

1.1.15. Recovery

The organization must also have a plan for recovery should the most current BIOS firmware standard cause instability or compatibility issues (Cooper, Polk, Regenscheid & Souppaya, 2011). Management must identify the process by which the situation will be assessed. As an example, instability may be due to an overlooked setting or feature rather than the firmware itself. A much greater level of effort may be required to roll back firmware rather than to adjust a single setting. This process should take these situations into account to ensure the most acceptable recovery actions are applied.

Regardless of the recovery process or decisions, the goal of recovery is to ensure that the target computers are recovered to full operation and uniformly configured to an approved standard. The previous standard was likely superseded by the failing standard. In addition, changes to the settings in the failed standard will require approval prior to implementation. For this purpose, an emergency change advisory board meeting should likely be called to approve any necessary actions.

1.1.16. Disposition

Once a device has reached the end of its lifecycle it should be returned to the lifecycle management office for final disposition. To facilitate this process, management should direct development of sanitization procedures that must be followed to ensure proper clearing of BIOS artifacts prior to permanent disposal (Cooper, Polk, Regenscheid & Souppaya, 2011). Without proper sanitization, BIOS firmware settings can be analyzed for weakness, BIOS passwords can be recovered, and any certificates or encryption keys may be retrieved from a computer equipped with a Trusted Platform Module.

Enterprise BIOS Firmware Features Baseline

In order to support acquisition decisions the organization should develop a BIOS firmware features baseline. This baseline will identify the minimum set of features that a computing platform must support to be considered for purchase. The baseline serves as a common configuration reference point which will ensure that necessary features are not overlooked (Sans security essentials, 2013). This feature set should be identified by an enterprise configuration management body such as a CAB. The CAB membership should be diverse enough to ensure proper feature consideration from acquisition, lifecycle, operations, maintenance, and security perspectives (Behr, Kim & Spafford, 2005). The BIOS firmware features baseline should be reviewed at least annually to address current trends, upcoming projects/initiatives, and new features. An example baseline can be seen in Appendix B.

BIOS Version and Settings Standards

The final documentation element will likely be the largest and most time consuming to develop. After surveying the landscape of laptops, desktops, servers, and virtual machine platforms the time has come to document the specific settings that need to be configured on each make and model. These standards are intended to be applied to the entire organization and are mandatory unless an exception has been identified in the governing policy. As defined, a standard specifies the specifics about how a task is to be carried out (Sans security essentials, 2013). In this instance, the standard will define the details regarding BIOS firmware configuration.

Each standard should be developed from a common template and use nearly identical language in order to avoid confusion. Basic template elements that should be included are: the originating body, an authority statement, target departments, computer make/model, BIOS firmware revision, a supersession statement, the specific BIOS settings, and the date that the standard was approved. An example BIOS Version and Settings Standard can be seen in Appendix C.

The largest portion of each document will invariably be the BIOS settings section. Within this section the structure of the BIOS menu system should be represented exactly by a bulleted or numbered list. Every configurable BIOS setting should appear in this list

with the corresponding approved setting value. This will make the standard explicit leaving no single option open to interpretation. Some options will require research for complete understanding. These settings should be commented to provide documentation regarding the background information and setting selection decision.

Taken together, this group of documents becomes a repeatable build library for the hardware that the organization employs. The Visible Ops Handbook recommends this capability to standardize the way that hardware is provisioned and deployed. Enforcement of this standard ensures that deployed hardware is configured consistently across the enterprise. The organization should attempt automation only after the manual process has been documented in a standard. The Visible Ops Handbook warns, "When dealing with IT operational processes, whether it is related to change, configuration, or release or security, take heed. If you cannot run these processes manually, do not attempt to automate it – all you will do is automate confusion," (Behr, Kim & Spafford, 2005).

If an automation capability exists for a particular make/model then the standard should identify where the automation procedure can be found. It may also be necessary to identify situations in which the standard can be circumvented. For instance, the standard will likely restrict the boot devices that are authorized for use. During a reimage operation it may be necessary to change this setting to allow CD/DVD, USB, or network boot. The standard should provide authorization for these types of non-persistent changes to be made in order to perform standard maintenance actions. It should also indicate that these temporary setting changes must be reverted prior to placing a device back into production.

New BIOS firmware is published infrequently and typically in an unscheduled fashion. As a result, BIOS version review should become a standard element of CAB meetings. The release notes for new BIOS firmware for a particular make/model should be reviewed to identify security enhancements, new functionality, or bug fixes. Based on the criticality or functionality that is provided in the update the CAB should make a recommendation to either adopt or delay upgrade of a particular platform. Once a new BIOS firmware revision is adopted a new BIOS setting standard must be developed prior to deployment.

Firmware Update and Standardized Settings Automation

As UEFI firmware matures and operating system support increases automation capabilities will likely improve and standardize. As mentioned previously, no 32-bit Microsoft Windows operating system has native UEFI support. This is likely to cause problems in standardizing automation efforts. In addition, manufacturers offer a wide range of support for BIOS firmware version and settings management automation. To complicate the matter further, support provided by one manufacturer typically only pertains to a subset of their specific platforms.

A survey completed in January of 2009 by Ian Godfrey of 1E Consulting indicated wide ranging support for BIOS automation. Manufacturers such as Asus, Acer, Sony, and Toshiba had no indicated support for BIOS automation whatsoever. In contrast, manufacturers like Dell, HP, and IBM provide several different tools for this purpose (Godfrey, 2009). Where automation support is available, manufacturer's offerings include free standalone tools, integrated operating system support, and enterprise system management solutions. A quick survey of the same field of manufacturers revealed little change since this white paper was published.

In a completely homogeneous environment a manufacturer supported system management solution is likely the best product for automation. Unfortunately, not all manufacturers offer a product that provides this type of support. Of the manufacturers' support offerings that were reviewed; only HP and Fujitsu provide a complete BIOS management solution. Searching for third party software solutions providing BIOS firmware settings and version support also revealed limited offerings. Those products that were found suffered the same limitations as vendor products; single vendor support, limited model support, etc.

Outside of the homogeneous environment it is likely not feasible to run multiple system management solutions to solely manage BIOS firmware and settings. The manufacturers that support BIOS firmware settings automation provide multiple avenues for settings update including Windows Management Instrumentation (WMI) scripting interfaces and command line tools. Basic examples of each of these techniques can be seen in Appendix D.

While these offerings appear primitive, existing tools can be used in conjunction to support automation on an enterprise level. In most cases, the command line tool can be used with a text or binary file to identify option setting values. In addition, configuration files can be created from an existing computer's configuration. As a result, a computer that has been thoroughly tested can be used to produce the "gold master" BIOS firmware settings file. Once the "gold master" settings file has been created, replicating the changes to another computer is a simple matter of replaying the settings.

A word of caution is necessary about use of these tools. This process is very detailed in nature and requires significant attention to detail. Readme files must be consulted to ensure that the tool in use supports the computer make and model being targeted. In addition, if using host operating system support (command line or WMI) the readme must be consulted to ensure that the operating system version and architecture are compatible. Finally, some manufacturers name utilities alike (some exactly) despite supporting different models, operating systems, and architectures. It is critical to ensure that the right tool version is being used with a particular model to ensure settings are applied properly. Some tools will fail outright while others will simply implement the changes that they support providing unexpected results. Without proper attention, this process can be frustrating and seem haphazard.

During the provisioning phase, the appropriate command line tool coupled with the "gold master" settings file can be executed in a standalone fashion as a standard component of system setup. Provisioning technicians should first check the BIOS firmware version, update if necessary, and apply the automated settings from the template. To ensure that settings are being applied appropriately command output should be analyzed or a diff-style tool should be employed to spot-check proper tool operation.

Support for the operation and maintenance phase will be the most complex and difficult scenario to automate. Since computers are already in operation update is a difficult prospect. In a heterogeneous environment hosting multiple manufacturer supported enterprise configuration management solutions may not be efficient. In this case, the same command line tools used for provisioning can be used to support the enterprise.

David R. Fletcher Jr., 6fletch9@gmail.com

Using existing enterprise system management tools such as Microsoft System Center Configuration Manager (SCCM) can create great efficiencies in this effort. For instance, collections of supported makes and models can be created for each supporting tool. After the collections have been created task sequences are developed to run the tool, provide credentials, and identify the appropriate settings file. An SCCM task sequence also works well with options that typically require a restart such as TPM settings since the task sequence tracks state between restarts. In addition, SCCM provides inherent support to track progress.

These settings files can also be used with integrated no-cost solutions such as Active Directory Group Policy. This solution will likely be more complex. The computer make/model will need to be identified through a Windows Management Instrumentation filter or script, the appropriate utility must be selected, and the correct command line syntax and options file must be specified. Reporting will also be more complicated as command output will need to be collected and analyzed to identify success or failure.

These same techniques can be used to confirm compliance with BIOS firmware settings and version standards. Using a diff tool, the output from the appropriate settings utility can be compared with the gold master to identify deviations. Once identified, these deviations can be corrected using one of the processes detailed above. This verification capability may take some initial effort since each text file collects some system-specific information that will differ from computer to computer.

Unless there is a catastrophic failure requiring touch maintenance, the recovery phase will rely on the same tools and techniques outlined above. If a catastrophic event does occur, then the organization can employ the same techniques used in the provisioning phase to return systems back to production capability.

Finally, during the disposition phase, an additional settings file can be created to return the system to default settings and destroy any sensitive artifacts. This process represents potentially sensitive information leaving the organization so an automated verification process such as the diff-style validation described above should be employed on every computer.

David R. Fletcher Jr., 6fletch9@gmail.com

Caution should be exercised when using any of these tools for automation. NIST SP 800-147 speculates that BIOS firmware malware may increase as a result of the UEFI standardization effort. The resulting standardized interfaces may represent a prime target for widespread infection (Cooper, Polk, Regenscheid & Souppaya, 2011). In addition, BIOS credentials should be passed in a secure manner. All of the command line tools surveyed include encrypted password support. Details of the encryption standard employed by any individual tool should be investigated prior to use to ensure that industry standard algorithms and key lengths are employed.

Conclusion

Application of the phases outlined in NIST SP 800-147 results in a managed and predictable computing environment at the hardware level. This result is achieved through rigorous application of configuration management principles. By initially surveying the target environment an organization can identify existing capabilities and vulnerabilities. Once this survey is complete support for standardization can be addressed. The most important document in establishing a standardization process is the Enterprise BIOS Standardization policy. This policy is management's endorsement commit resources to solve the problem. Next, baseline features should be identified and integrated into the acquisition process. Finally, standards must be developed to address the specific details for each individual computer make and model. Once manual processes have been developed and documented, focus can be placed on automation. Automation capabilities should be developed to support each phase of the NIST process.

Creating a standardized and predictable hardware configuration environment will eliminate firmware configuration variance in the troubleshooting process. In addition, standardized hardware configuration will provide predictable support to higher-level efforts. Projects such as data at rest encryption, patch management, virtualization and virus/malware control can all benefit from careful consideration of pertinent BIOS firmware options. This results in higher availability and security since BIOS firmware settings and versions are managed throughout the lifecycle of a computing device. Investment early in the computing lifecycle, preferably acquisition, also results in higher

availability and lower cost due to full out of the box support.

References

(2013). *Sans security essentials bootcamp style*. (V2013_0202 R2 ed., Vol. 401.2, pp. 43-71). Bethesda MD: The SANS Institute.

Behr, K., Kim, G., & Spafford, G. (2005). *The visible ops handbook: implementing ITIL in 4 practical and auditable steps*. Eugene OR: IT Process Institute.

Cooper, D., Polk, W., Regenscheid, A., & Souppaya, M. U.S. Department of Commerce, National Institute of Standards and Technology. (2011). *BIOS protection guidelines* (SP 800-147). Retrieved from National Institute of Standards and Technology website: http://csrc.nist.gov/publications/nistpubs/800-147/NIST-SP800-147-April2011.pdf

Godfrey, I. (2009, January). *Remote management of BIOS configuration*. Retrieved from http://www.1e.com/fr/wp-content/plugins/1E-Resources_fr/includes/doc_download.php?id&86865831&ei=XvAlUs60D8TqyQHX2YDwDQ&usg=AFQjCNGITIaBXDMTjHAySoV_srIKzyKOGQ&bvm=bv.51495398,d.aWc&cad=rja

Greene, J. (2012). *Intel trusted execution technology*. Retrieved from http://www.intel.com/content/dam/www/public/us/en/documents/white-papers/trusted-execution-technology-security-paper.pdf

Haletky, E. (2010, June 29). *vSecurity gets a boost from tpm/txt*. Retrieved from http://www.virtualizationpractice.com/vsecurity-gets-a-boost-from-tpmtxt-6179/

Hp quicklook. (2009, June). Retrieved from http://h41112.www4.hp.com/promo/professional-innovations/pdf/uk/en/Hp_professional_innovation_2009_technology_spot_light_easeofuse_quicklook3_us_en.pdf

Hp quickweb. (n.d.). Retrieved from http://h71036.www7.hp.com/hho/us/en/pclc/articles/quickweb.html

Intel trusted execution technology. (2012). Retrieved from http://www.intel.com/content/dam/www/public/us/en/documents/white-papers/trusted-execution-technology-security-paper.pdf

Jumelet, A. (2010, November 12). [Web log message]. Retrieved from http://blogs.technet.com/b/arnaud_jumelet/archive/2010/11/12/how-to-regenerate-the-bitlocker-numerical-recovery-password.aspx

Kim, G., Love, P., & Spafford, G. (2008). *Visible ops security: achieveing common security and it operations objectives in 4 practical steps*. Eugene OR: IT Process Institute.

Kubicki, K. (2004, October 11). *A bit about the nx bit; virus protection woes*. Retrieved from http://www.anandtech.com/show/1507

Mcmillan, R. (2011, August 5). *The undetectable malware that real hackers don't seem to want*. Retrieved from http://www.pcworld.com/article/237437/the_undetectable_malware_that_real_hackers_dont_seem_to_want.html

Mueller, S. (2013). *Upgrading and repairing PCs*. (21 ed.). Indianapolis, IN: Que Publishing.

Trusted platform module summary. (2013). Retrieved from http://www.trustedcomputinggroup.org/files/resource_files/4B55C6B9-1D09-3519-AD916F3031BCB586/Trusted Platform Module Summary_04292008.pdf

Appendix A – BIOS Settings Sample Policy Letter

1.0 Purpose

The purpose of this policy is to provide guidance for standardization of Basic Input/Output System (BIOS) firmware settings across the <Company Name> enterprise in order to ensure uniformity, increase security, and to detect and remediate unauthorized configuration changes. This policy prescribes a process by which the recommendations found in National Institute of Standards and Technology (NIST) Special Publication (SP) 800-147 BIOS Protection Guidelines, Section 3.2 will be implemented.

2.0 References

<Company Name> Security Policy

NIST SP 800-147 BIOS Protection Guidelines

3.0 Scope

This policy applies to all <Company Name> employees, contractors, workforce members, vendors and agents with a <Company Name>-owned or personal workstation connected to the <Company Name> network.

4.0 Background

Many performance and security enhancing technologies rely on correct BIOS firmware settings for proper operation. Applying standardized settings across the <Company Name> enterprise will ensure optimal security and availability in addition to streamlining troubleshooting efforts. NIST SP 800-147 prescribes a five phase process for managing BIOS software and settings; Provisioning, Deployment, Operation/Maintenance, and Recovery.

5.0 Policy

<Company Name> will implement the NIST five phase process to ensure that settings are properly applied and uniform across the enterprise.

This policy augments existing lifecycle management and operations policies.

3.1 Product acquisition – Prior to purchase of a particular computer make/model the model will be presented to the Change Advisory Board (CAB) for consideration. Board members will ensure that this make/model supports a minimum set of features identified

as the <Company Name> BIOS features baseline.

3.2 Platform Deployment – Upon receipt of a new computer make/model the CAB will develop a BIOS firmware settings standard. This standard will identify the make/model of the computing platform, the approved BIOS revision, and acceptable values for each individual setting. If possible, the CAB will also create a "gold master" BIOS settings file for automation should the platform support this operation.

3.3 Operation/Maintenance – Automation tools will be used to assess compliance with prescribed BIOS firmware settings standards on a <Interval> basis. In addition, whenever hands-on maintenance is performed on a particular computer, BIOS firmware settings compliance will be evaluated by the maintenance or helpdesk technician. The CAB will review existing BIOS firmware settings standards on an annual basis to determine if BIOS firmware update and/or settings changes are necessary. If a change is found to be necessary the prescribed BIOS firmware settings standard will be tested according to operations policy and all documentation will be updated accordingly. Once policy and documentation are updated, the IT operations department will deploy the updated standard using existing automation tools to the applicable make/model within <Interval> days. Beyond this date, any remaining systems will be quarantined from network access and updated manually.

3.4 Recovery – Should a BIOS firmware settings change cause widespread avialability or security concerns, the CAB will be convened to determine corrective action. This may require rollback to the most recent BIOS firmware revision and BIOS firmware settings standard utilizing the escrowed gold master or adjustments to the failed standard. Once corrective action has been identified the IT operations department will test and initiate the change accordingly. This action will be completed within the same timeframe as standard operation/maintenance changes.

3.5 Disposition – As computing platforms are removed from operation due to standard lifecycle management processes these retired platforms will be reset to factory defaults ensuring that all data indicating custom configuration are removed. In addition, any hardware encryption keys and other sensitive information will be removed from firmware-related devices.

6.0 Enforcement

Any employee found to have violated this policy may be subject to disciplinary action, up to and including termination of employment.

7.0 Effective Period

This policy will remain in effect until superseded or cancelled by <Company Name>

David R. Fletcher Jr., 6fletch9@gmail.com

management.

8.0 Definitions

Computer – A computer is any device or software that employs a configurable BIOS firmware and runs an operating system. This includes, but is not limited to, servers, laptops, desktops, and virtual machines.

9.0 Revision History

None

Appendix B - BIOS Features Baseline

<Date>

MEMORANDUM FOR <Company Name> LIFECYCLE MANAGEMENT OFFICE

FROM <Company Name> CHANGE ADVISORY BOARD

SUBJECT: Computing BIOS Features Baseline

1. In order to conform with requirements set forth in the <Company Name> BIOS Settings Standardization Policy any server, laptop and/or desktop computer must support the following minimum BIOS features:

a. Complex passwords conforming to <Company Name> password policy.

b. Trusted Platform Module 2.0.

c. Ability to disable the following peripherals (if equipped):

i. PATA/SATA Ports

ii. USB Ports

iii. Serial Port

David R. Fletcher Jr., 6fletch9@gmail.com

iv. Parallel Port

v. IEEE 1394 Port

vi. Flash Media Reader(s)

d. Wake On LAN

e. Advanced Configuration Power Interface

i. Power on After Power Failure

ii. Scheduled Power On

f. Wireless Switching Capability (disable wireless when wired cable is present)

g. Hardware virtualization support

2. This baseline document will remain in effect until superseded. For clarification of any of the above features, please consult the Change Advisory Board.

Appendix C – BIOS Settings Standard (HP EliteBook 6930p)

<Date>

MEMORANDUM FOR <Company Name> Network Operations

<Company Name> Computer Maintenance

<Company Name> Lifecycle Management

<Company Name> Helpdesk

FROM: <Company Name> Change Advisory Board (CAB)

SUBJECT: BIOS Baseline Configuration (HP Elitebook 6930p, v F.17)

SUPERSEDES: None.

AUTHORITY: <Company Name> Enterprise BIOS Standardization Policy, dtd <Date>

1. This document describes the Basic Input Output System (BIOS) settings standard that will be employed on all HP Elitebook 6930p laptops running BIOS software revision F.17. This document will remain in effect until the HP Elitebook 6930p laptop model has been completely life-cycle replaced or a new BIOS software revision becomes the standard. Deviations from this baseline will be considered on a case by case basis and must be approved by the <Company Name> CAB prior to implementation. This baseline does not apply to non-persistent changes (i.e. boot order change to re-image computer). However, any non-persistent changes must be reversed to adhere to this standard prior to inclusion in <Company Name> production environments.

2. The following configuration settings constitute the BIOS baseline for the HP Elitebook 6930p laptop running BIOS software revision F.17:

1. *File Menu:*
    i. Set System Date and Time to Local Time
    ii. ***All other options default.***

David R. Fletcher Jr., 6fletch9@gmail.com

2. *Security Menu:*
    i. *HP Sparekey – **Disabled***
    This option allows the user to reset the BIOS password after answering three questions correctly.
    ii. *Always Prompt for SpareKey Enrollment – **Disabled***
    iii. *Fingerprint Reset on Reboot (If Present) – **Disabled***
    iv. *Allow reset of HP ProtectTools Security Keys – **Disabled***
    v. *Change Password – **Change to standard BIOS password scheme***
    vi. *TPM Embedded Security:*
    The following options control the Trusted Platform Module, used by BitLocker to manage keying material for full disk encryption at rest.
        i. *Embedded Security Device Availability – **Available***
        ii. *Embedded Security Device State – **Enabled***
        iii. *Factory Defaults – **No***
        iv. *Power-On Authentication Support – **Disabled***
        v. *Reset Authentication Credential – **No***
        vi. *OS Management of TPM – **Enabled***
        vii. *Reset of TPM from OS – **Enabled***
    vii. ***All other options set as default.***
3. *Diagnostics Menu (No Configuration Settings Defined)*
4. *System Configuration:*
    i. *Language – **English***
    ii. *Boot Options:*
        i. *Startup Menu Delay (Sec.) – **0***
        ii. *WWAN Initialization Delay – **Disabled***
        iii. *Custom Logo – **Disabled***
        iv. *Display Diagnostic URL – **Disabled***
        v. *CD-ROM boot – **Disabled***
        vi. *SD card boot – **Disabled***
        vii. *Floppy boot – **Disabled***
        viii. *PXE Internal NIC boot – **Disabled***
        ix. *MultiBoot Express Boot Popup Delay (Sec) – **0***
        x. *Boot Order – **Notebook Hard Drive***
        Notebook Ethernet
        Notebook Upgrade Bay
        Dock Upgrade Bay
        SD Card

USB CD-ROM

USB Floppy

USB Hard Drive

Note: *Although the boot order indicates a list of devices beyond the Notebook Hard Drive these devices will not be listed in the boot menu as long as they are disabled in the BIOS.*

iii. *Device Configurations:*
   i. *USB legacy support –* ***Enabled***
   ii. *Parallel port mode –* ***ECP***
   iii. *Fan Always on while on AC Power –* ***Disabled***
   iv. *Data Execution Prevention –* ***Enabled***
   This setting disallows code from being executed in the data portion of memory.
   v. *SATA Device Mode –* ***AHCI***
   vi. *Secondary Battery Fast Charge –* ***Enabled***
   vii. *HP QuickLook 2 –* ***Disabled***
   This setting caches Microsoft Outlook e-mail and calendar information for pre-boot access.
   viii. *Virtualization Technology –* ***Enabled***
   This setting is useful for devices expected to run virtualization software such as VMWare. It is not necessary on 127 WG workstations.
   ix. *TXT Technology –* ***Enabled***
   This setting enables Intel malware protection available in the CPU core.
   x. *Dual Core CPU –* ***Enabled***
   xi. *UEFI Boot Mode –* ***Disabled***
   UEFI = Unified Extensible Firmware Interface. This setting provides host operating system access to BIOS.
   xii. *Wake on USB –* ***Disabled***
   xiii. *Numlock state at boot –* ***Off***
iv. *Built-In Device Options:*
   i. *Wireless Button State –* ***Enabled***
   ii. *Embedded WLAN Device –* ***Enabled***
   iii. *Network Interface Controller (LAN) –* ***Enabled***
   iv. *LAN/WLAN Switching –* ***Enabled***
   This setting physically disables the wired/wireless network adapter when the opposite device is connected.
   v. *Wake on LAN –* ***Follow Boot Order***

vi. *Ambient Light Sensor* – ***Enabled***
vii. *Notebook Upgrade Bay* – ***Enabled***
This setting controls activation of the notebook bay which hosts the CD/DVD-RW drive.
viii. *Fingerprint Device* – ***Disabled***
ix. *Audio Device* – ***Enabled***
x. *Modem Device* –***Disabled***
xi. *Microphone* – ***Enabled***

v. *Port Options:*
i. *Serial Port* – ***Disabled***
ii. *Parallel Port* – ***Disabled***
iii. *Flash Media Reader* – ***Disabled***
iv. *USB Port* – ***Enabled***
v. *1394 Port* – ***Disabled***
vi. *Express Card Slot* – ***Disabled***
vii. *Smart Card* – ***Enabled***

vi. *Set Security Level (**Leave all settings as default**)*

3. Where possible, the configuration described above will be applied via automated means through use of the HP supplied BIOS Configuration Utility which uses a pre-formatted text file as setting arguments. Use of this tool will be covered in a separate instruction.

4. This configuration baseline will remain in effect until this laptop model has been completely lifecycle replaced or a newer BIOS revision is necessary.

5. Questions or concerns regarding the actions identified above should be directed to the <Company Name> CAB.

Appendix D - BIOS Firmware Settings Automation Examples

The examples that follow serve to illustrate techniques to automate the potentially time-consuming and error-prone process of configuring BIOS firmware settings in support of standardization. These examples only cover settings automation on a single computer level. Once capable of automating a single computer, the target organization can use existing tools typically found within the enterprise for mass deployment. These tools are beyond the scope of this paper but could include; VBScript, PowerShell, Active Directory Group Policy, System Center Configuration Manager, or any of the other well-known configuration management tools. This appendix specifically covers settings automation on the Lenovo T420i. The methods illustrated include a command line tool provided by the manufacturer in addition to use of Windows Management Instrumentation via VBScript. These two capabilities are representative of offerings supported by both Dell and Hewlett-Packard.

This content is meant to generate ideas for automation in the reader's target environment. The current computing landscape is very diverse regarding supported BIOS automation tools, computing platforms, processor architectures, and operating systems. The target network should be surveyed to determine the level of support possible for automation in general to determine if the task will be worthwhile or even possible.

Lenovo T420i BIOS Settings Automation (SRSETUP)

The Windows compatible SRSETUP utility can be obtained from the Lenovo support website. SRSETUP supports both 32 and 64-bit Windows up to Windows 8. This utility has a very simple option set and is used to record BIOS settings on one computer and play them back on a target computer. Before recording BIOS settings, the BIOS standard for the target computer should be completed manually. The general process will be; record master BIOS settings, copy master file to target, replay master BIOS settings on target, record target BIOS settings, and compare master settings to target settings for verification.

1. Set up and test settings for the make/model according to the applicable BIOS firmware settings standard.

2. Download the SRSETUP utility from Lenovo and extract the software on the target computer.

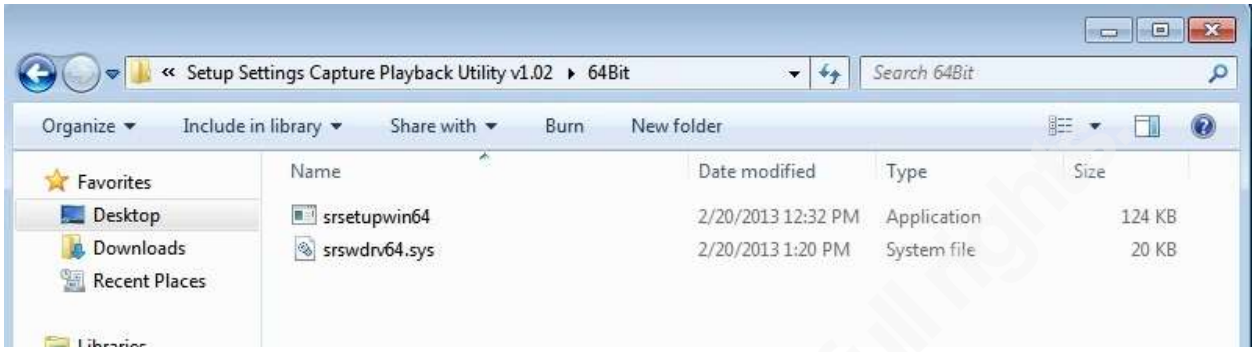


Figure 1: *SRSETUP software after download and extraction.*

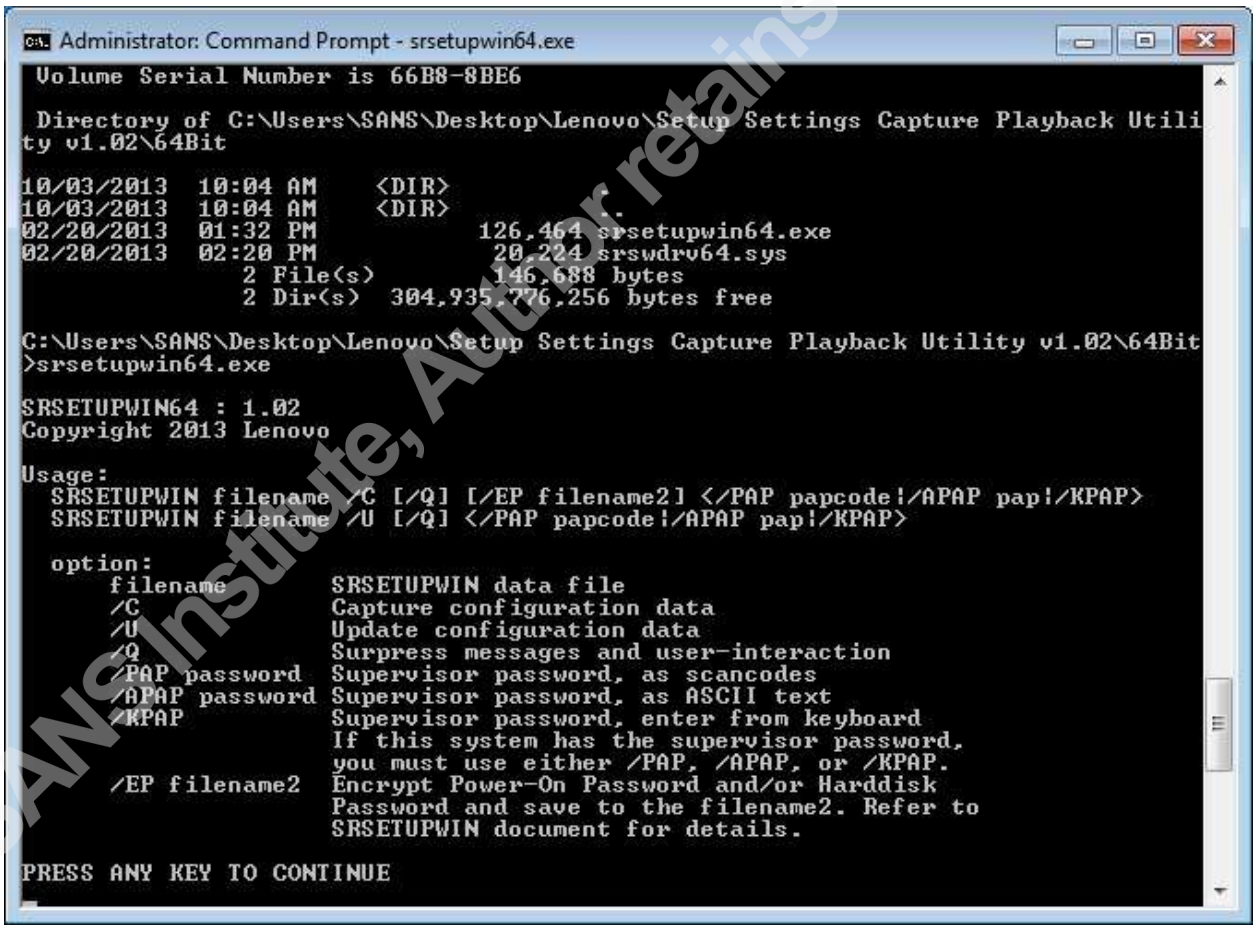


Figure 2: *SRSETUP command line options.*

3. Record the pre-configured BIOS settings from the source computer into a "master" BIOS firmware settings file.

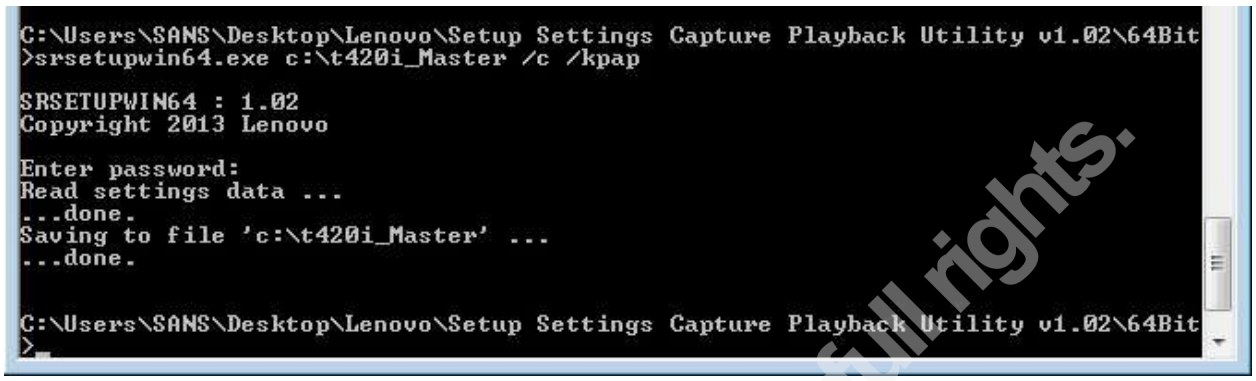


Figure 3: *Record desired BIOS firmware settings from source computer.*

4. Copy the "master" BIOS firmware settings file to the target computer.

5. Replay the "master" BIOS firmware settings file on the target computer using SRSETUP.

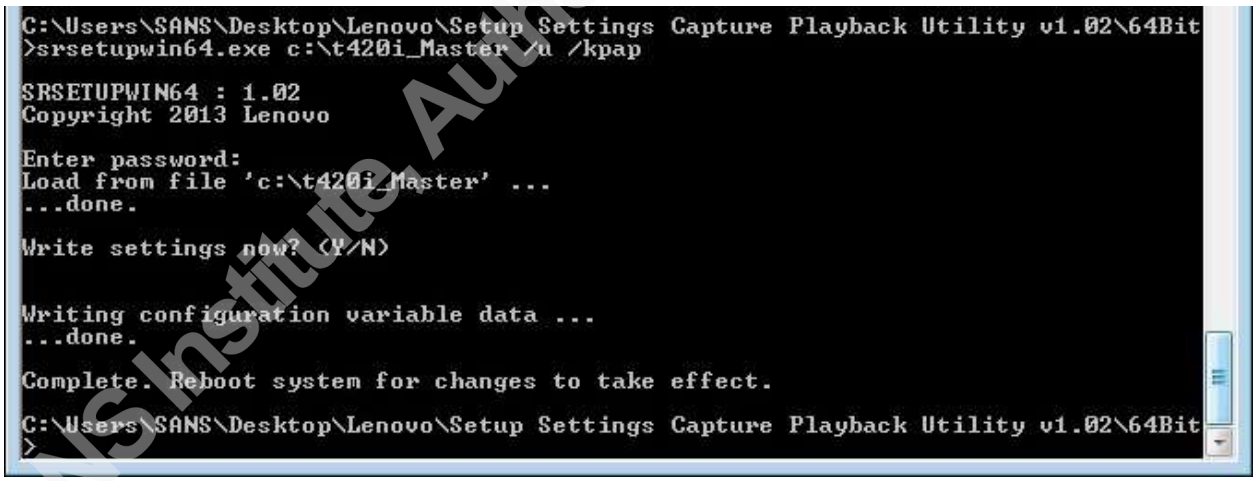


Figure 4: *Replay pre-recorded BIOS settings on the target computer.*

6. Capture the "cloned" BIOS firmware settings from the target computer using SRSETUP.

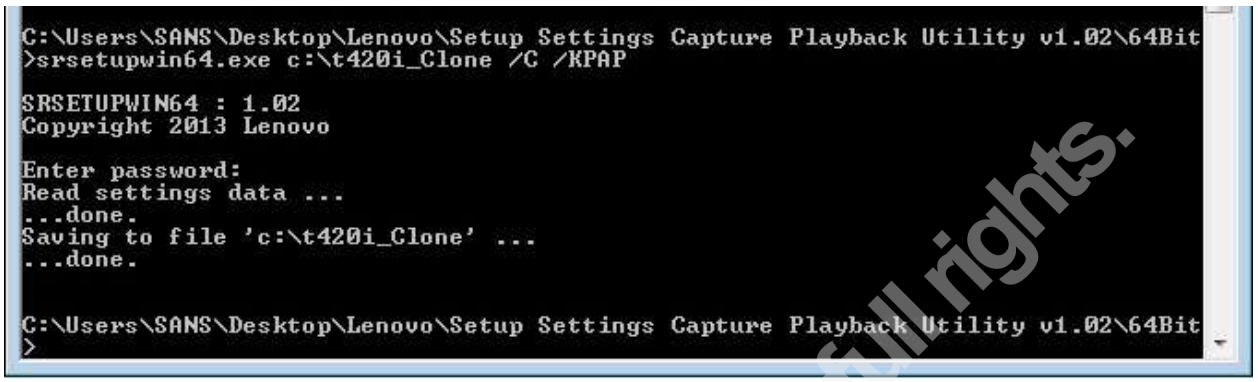


Figure 5: *Capture cloned BIOS settings from the target computer.*

7. Compare the "cloned" BIOS settings file to the "master" BIOS settings file to confirm proper configuration. While this is accomplished with the GUI WinDiff application in this example, the command line utility would be used for full automation.

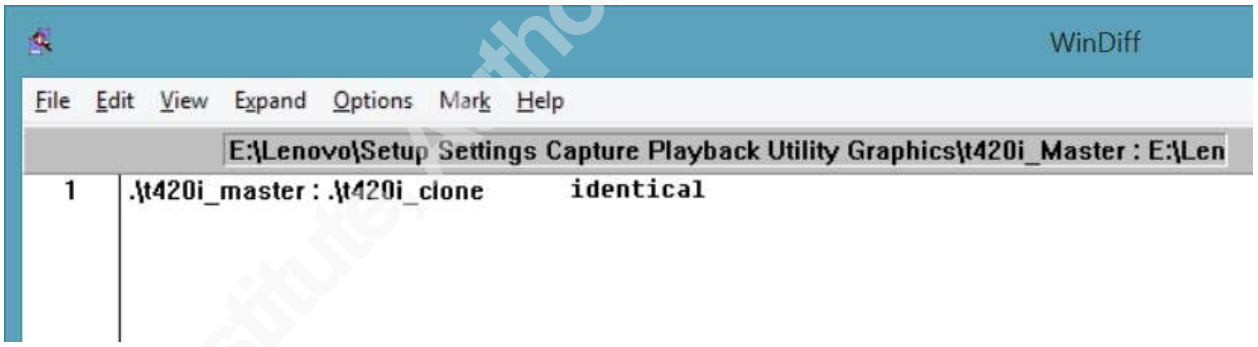


Figure 6: *Comparison of master and clone binary settings files to ensure settings conformance.*

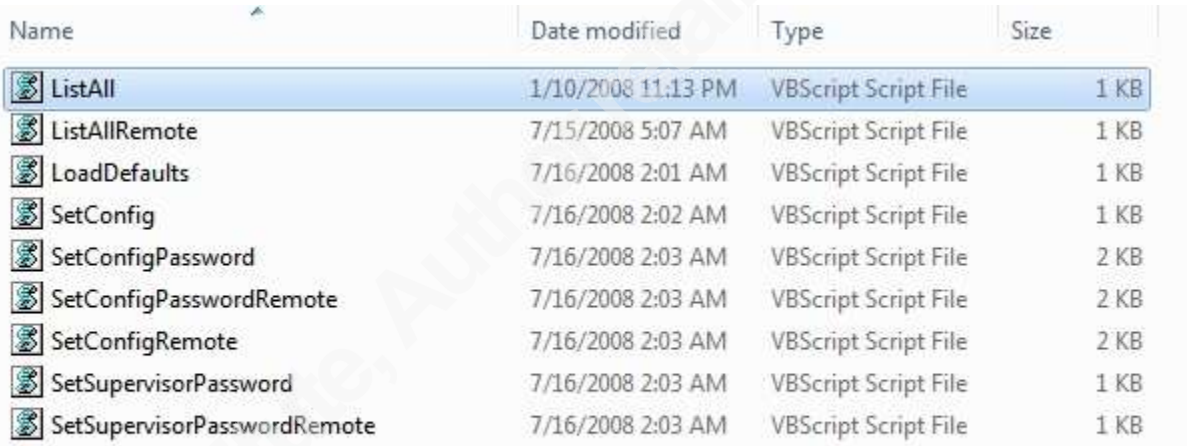
It should be noted that the SRSETUP utility does not have an option for encrypting the BIOS administrator password. This can be a hindrance to BIOS automation in some cases. For example, Active Directory Group Policy or scripts could transmit the password in clear text without transport layer encryption. This tool would be acceptable for provisioning, touch maintenance, and disposition but may be dangerous during operation and maintenance phases depending on transport layer encryption support. In addition, local usage would likely be within a Windows batch script using either the /APAP or /PAP options. In this case the password may be recovered and compromised with simple file recovery. A tool like SDELETE from the Sys Internals tool suite should be used to ensure that file recovery is impossible.

Lenovo T420i BIOS Settings Automation (WMI Provider)

The Lenovo T420i BIOS firmware settings are exposed to the Microsoft Windows operating system through the Lenovo_BiosSetting Windows Management Instrumentation (WMI) provider. Using this provider, the BIOS settings can be adjusted both locally and remotely using VBScript, JavaScript, and Windows PowerShell. A primer on using the WMI provider and basic scripts can be obtained from the Lenovo

website. This method has the same options as SRSETUP for securing and communicating the BIOS administrator password. However, the password can be communicated securely by using impersonation and packet privacy options with WMI. In addition, instead of dealing with a binary settings file as output ASCII text names and setting values are used. This allows setting verification using ASCII text which can give an indication of which exact setting is not configured to standard. The general process for automation is as follows; configure the source computer according to the BIOS settings standard, query the computer to determine target settings and values, develop a script or batch file using this output, deploy settings to a target computer using the script, verify conformance by querying all settings.

1. Obtain the WMI provider documentation and sample scripts from the Lenovo support site.



Name	Date modified	Type	Size
ListAll	1/10/2008 11:13 PM	VBScript Script File	1 KB
ListAllRemote	7/15/2008 5:07 AM	VBScript Script File	1 KB
LoadDefaults	7/16/2008 2:01 AM	VBScript Script File	1 KB
SetConfig	7/16/2008 2:02 AM	VBScript Script File	1 KB
SetConfigPassword	7/16/2008 2:03 AM	VBScript Script File	2 KB
SetConfigPasswordRemote	7/16/2008 2:03 AM	VBScript Script File	2 KB
SetConfigRemote	7/16/2008 2:03 AM	VBScript Script File	2 KB
SetSupervisorPassword	7/16/2008 2:03 AM	VBScript Script File	1 KB
SetSupervisorPasswordRemote	7/16/2008 2:03 AM	VBScript Script File	1 KB

Figure 7: *Sample WMI Provider Scripts from Lenovo Support.*

2. Manually configure the computer according to the BIOS firmware settings standard.

3. Run the "ListAll.vbs" script to capture settings and values that conform to the standard. Manually verify that settings displayed match the BIOS firmware settings standard.

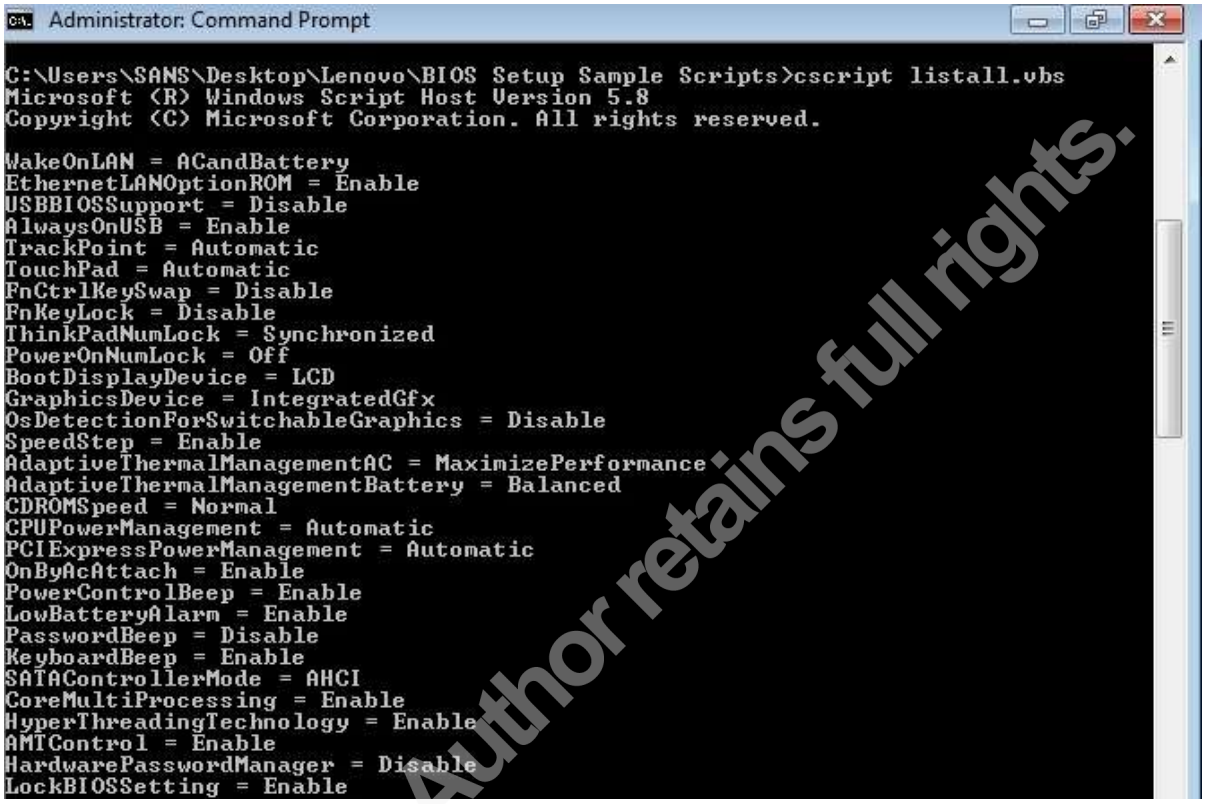


Figure 8: *Abbreviated ListAll.vbs output showing configured BIOS firmware settings.*

4. Capture validated settings by using output redirection to create a "master" settings file for future comparison.

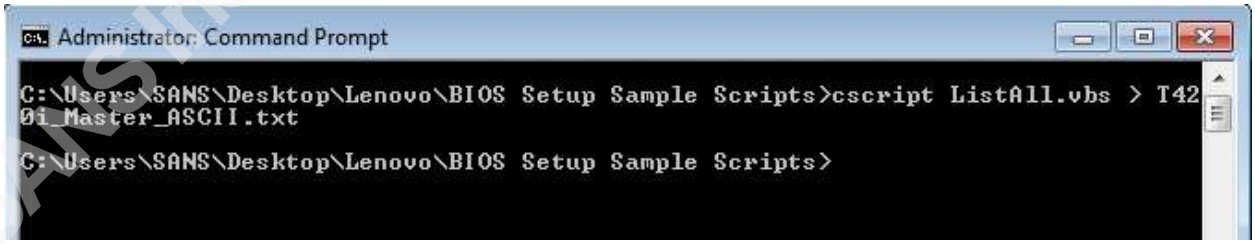


Figure 9: *Export settings from master configuration using command line redirection.*

5. Create a batch file (or new VBScript) to process all of the desired settings. Batch file development uses one of the sample SetConfig.vbs, SetConfigRemote.vbs, SetConfigPassword.vbs, or SetConfigPasswordRemote.vbs scripts. This tutorial will illustrate use of the SetConfigPassword.vbs script in a batch file. A new VBScript or PowerShell script would allow setting of a collection of values rather than calling the sample script for each setting. The master text file created in the previous step can be used very efficiently to create this file. Simply append the

script call to the beginning of each line, replace the " = " with a single space, and append credentials to the end of each line.

Figure 10: *BIOS firmware settings batch file using example scripts from Lenovo support site.*

6. Deploy the settings to the target computer by using the newly created batch file.

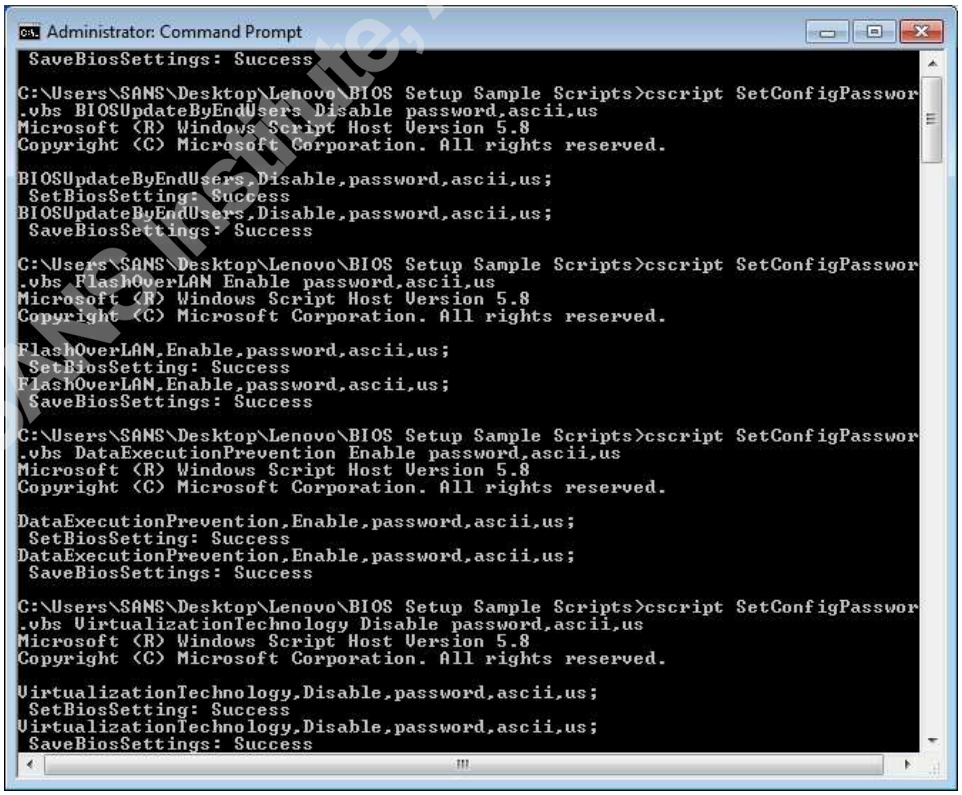


Figure 11: *Execution of the BIOS firmware settings batch file. Notice the setting/save status messages.*

7. Export settings changes on the cloned computer for comparison with the master settings file.

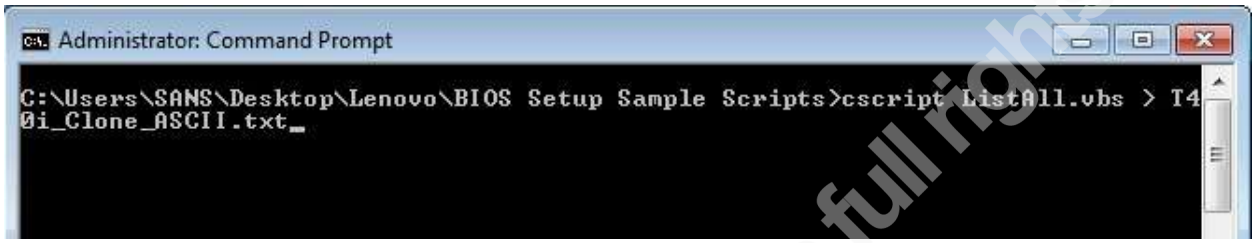


Figure 12: *Export cloned system settings to a file for comparison with master settings file.*

8. Verify configuration settings by comparing the master settings to the target settings. While this is accomplished with the GUI WinDiff application in this example, the command line utility would be used for full automation.

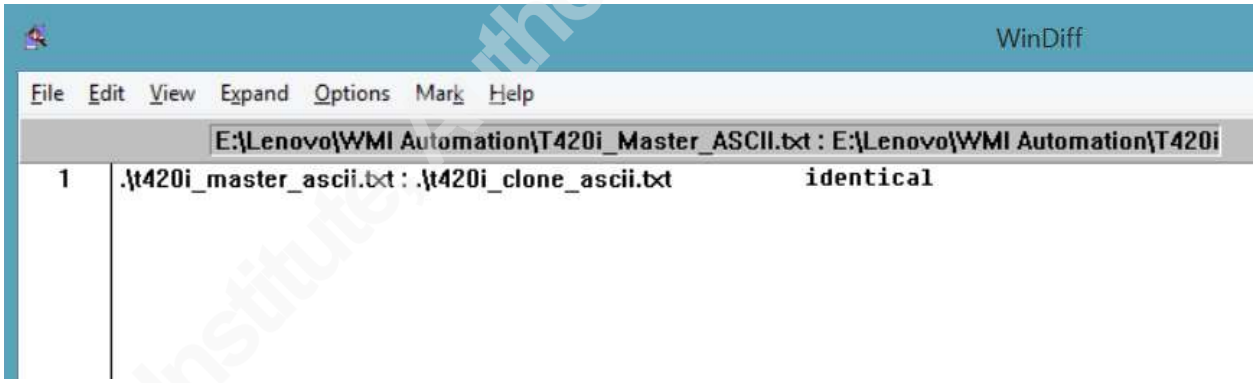


Figure 13: *File comparison to ensure export of master configuration matches export of clone.*

Extension of this technique for enterprise use would involve completing steps 1 through 5 during the adoption phase of a particular hardware platform. Steps 6 through 8 would be automated using an existing configuration management platform for use throughout the enterprise. The script created in step 5 would most likely encompass all of the operations identified in steps 6 through 8. This would provide the greatest level of automation. In addition, these same techniques can be used to perform periodic audits of the environment using just the processes outlined in steps 7 and 8. As described with the SRSETUP command line tool, if files containing the BIOS password are copied to target computers, a secure delete tool should be used to ensure credentials are not compromised. Finally, the LoadDefaults.vbs script can be used in the disposition phase to ensure that all settings are returned to their manufacturer supplied values.