Examination of Microsoft Baseline Security Analyzer

By Megan Bell

Table of Contents

Introduction

This examination presents an examination of Microsoft Baseline Security Analyzer (MBSA) software to scan a host computer running Microsoft Windows Operating System (OS) for a set of pre-defined secure system configurations. Testing was conducted in a virtualized environment on a host computer running Windows 7. Since the virtualized environment restricted network access, experimental settings to evaluate MBSA were limited to passwords changes, registry settings and Windows firewall use.

Many organizations use MBSA and other tools such as the Microsoft Security Assessment Tool (MSAT) to routinely audit computer systems for organizational compliance (Cobb, 2010). MBSA can be used to audit one specific system or run across a network for a broader look at organizational compliance. MBSA examines several criteria including system configuration issues, Windows registry settings, and system updates. When combined with other monitoring processes such as firewall log analysis using tools such as Splunk and appropriately implemented organizational policies such as group policy implementation through Microsoft Active Directory, MBSA can assist with bridging the gap between high-level organizational compliance and local computer system configurations.

This examination briefly outlines the environment and host configuration for test analysis. Next, an initial scan is conducted of the host computer with its default configurations. Then, firewall and registry settings are modified to assess how MBSA scoring adjusts for changes in system configurations. Finally, a review of MBSA and its use are reviewed in a broader context such as virus detection and evasion.

Setup

To complete the experiment, a host VM was created and assigned a unique IP address. The VM was then accessed through a secure Virtualized Private Network (VPN) connection and Microsoft Remote Desktop Protocol (RDP) software. An experiment was conducted with MBSA and differing Windows system configurations available through Windows control Panel and the Windows System registry.

MBSA Software Overview

A review of the MBSA program files within file path C:\Program Files\Microsoft Baseline Security Analyzer 2 was performed to identify the version of MBSA. According to release notes within the file readme.html in file path C:\Program Files\Microsoft Baseline Security Analyzer 2\Help, version 2.3 of MBSA was installed and was identified as being able to scan Windows OS versions including "Windows XP, Windows Server 2003, Windows Vista, Windows Server 2008, Windows 7, Windows Server 2008 R2, Windows 8, Windows Server 2012, Windows 8.1, Windows Server 2012 R2 - including 64-bit (x64 and ia64) support" (Microsoft, 2015; Johnson, 2009).

MBSA was opened and the "Scan the Computer" option was clicked. Figure 1 displays this option as permitting a scan of a single computer.

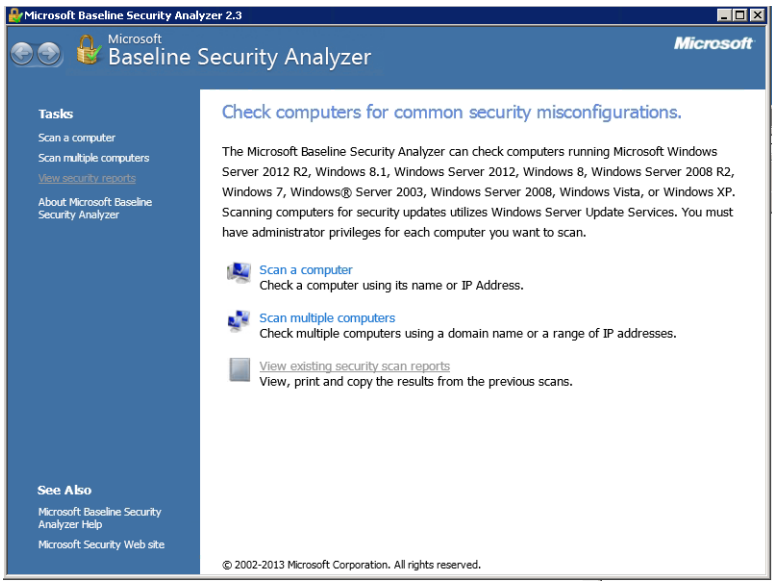

Figure 1. Main menu of MBSA application

The scanning menu presented MBSA default configurations for scanning. MBSA identified the target computer being scanned, the report name and scanning options. By default, all options are selected. As observed in Figure 2, options for IIS vulnerabilities, SQL vulnerabilities and security updates were de-selected. The default host computer and report name were not changed. With the adjustments to the default settings, the scan was run.

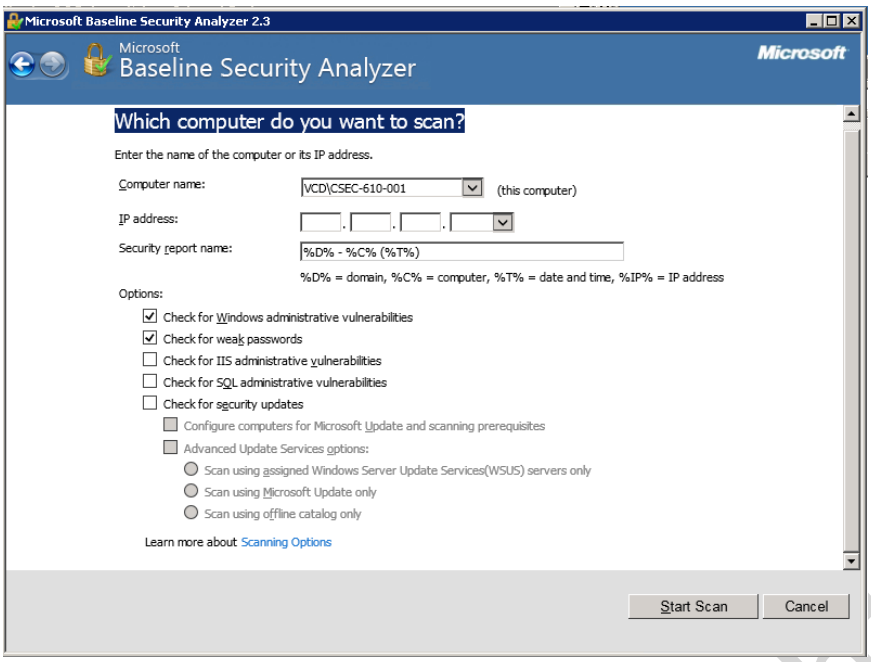

Figure 2. MBSA menu options with SQL and IIS vulnerabilities unchecked

A second test was run to examine the implication of firewall activation. Microsoft Windows OS 7 has a built-in firewall that can be accessed through Control Panel Settings. The firewall was activated, and then an MBSA scan was performed. Results were saved.

The final test was a set of scans to assess how MBSA determines appropriate settings for Restrict Anonymous. The Registry Editor was opened to Within Registry Editor, registry keys RestrictAnonymous and EveryoneIncludesAnonymous were found within the System registry location HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\LSA. The two keys were tested together to determine the impact on MBSA security scores. Table 1 illustrates the key combinations used for testing. For each combination, a scan was run, and the MBSA output was recorded.

Table 1

Windows System Registry Anonymous User Settings

RestrictAnonymous	EveryoneIncludesAnonymous
0	0
0	1
1	0
1	2
2	1

Results

Scans

In the first scan on default settings, MBSA reported a security assessment score of "Severe Risk". Within the security assessment report presented in Appendix A, several administrative vulnerabilities were listed, and one vulnerability regarding Windows Automatic Updates was reported as s critical. Evaluation of the Control Panel settings for settings menu Windows Update Change indicated that the "Important updates" option was set to "Never check for updates (not recommended)", and the "Recommended updates" option was not checked. With Automatic Updates disabled, MBSA returned a critical vulnerability rating. Most vulnerabilities including firewall disablement were informational.

In the second MBSA scan, the Windows firewall was activated within Control Panel. The caution message associated first scan where the firewall was disabled changed to a successful audit response of the firewall being activated.

The final scan evaluated MBSA's adjustment to registry settings. For Windows 7, the Windows System registry settings for registry keys RestrictAnonymous and EveryoneIncludesAnonymous were collectively examined to assess the impact on MBSA scans. According to Miscosoft (2015), "We do not recommend setting RestrictAnonymous to 2 on domain controllers or on computers running Small Business Server (SBS) unless they are in pure Windows® 2000 environments and have been tested for application compatibility." While this setting was not recommended, a change of the RestrictAnonymous registry value alone to this setting did not translate to an MBSA warning message for the host VM. As observed in Table 2, there were no changes in MBSA scan messages for RestrictAnonymous registry value changes while EveryoneIncludesAnonymous key was set to "0". When EveryoneIncludesAnonymous was set to "1", changes in RestrictAnonymous values resulted in MBSA returning information messages concerning the risks with allowing anonymous access to the host system.

Table 2

MBSA Scan Results for Windows Registry Anonymous Settings

RestrictAnonymous	EveryoneIncludesAnonymous	MBSA message resulting message
0 (default)	0	Computer is properly restricting anonymous

RestrictAnonymous	EveryoneIncludesAnonymous	MBSA message resulting message
		access. =
0	1	Computer is running with RestrictAnonymous = 0. This level allows basic enumeration of user accounts, account policies, and system information. Set RestrictAnonymous = 2 to ensure maximum security.
1	0	Computer is properly restricting anonymous access.
1	2	Computer is running with RestrictAnonymous = 1. This level prevents basic enumeration of user accounts, account policies, and system information. Set RestrictAnonymous = 2 to ensure maximum security.
2	1	Computer is properly restricting anonymous access.

Summary of Use

MBSA interrogates a target computer for a pre-defined set of scans options. The default scan options comprehensively perform all available tests on a target. The issue with the default setting is that not all Windows computer host an IIS server for Internet services or a SQL server database and related services. Testing for non-existent options could extend the time to perform scans and result in audit results that are not appropriate for analysis.

MBSA scans Windows system configurations that are controlled through Control Panel access and also evaluates registry settings. Changing the firewall setting and registry settings for anonymous logon both resulted in changed MBSA information messages, but did not results in the reporting of a sever vulnerability. In this experiment, the only severe vulnerability was the lack of Windows Automatic Updates. Since the target host did not have network connectivity, changes to Automatic updates could not be examined for the impact on MBSA. Overall, MBSA is responsive to Windows system configurations changes. Further analysis would be required to fully assess the rating assignments for severe versus non-severe security assessment ratings.

Follow-up Analysis Related to MBSA

Assessment of Security Measures

The resulting MBSA security assessment report displayed an overall assessment score of "Severe Risk" with documentation indicating that one or more critical checks failed. A review of the report's listed vulnerabilities was conducted, and the disablement of Windows Automatic Updates was reported as s critical administrative vulnerability. To confirm the system

configurations for Windows Automatic Updates, a review was conducted of the settings within Control Panel > System and Security > Windows Update Change settings menu. As observed in Figure 3, the "Important updates" option was set to "Never check for updates (not recommended)", and the "Recommended updates" option was not checked. This configuration indicated that Windows updates could only be performed manually. Microsoft recommends the use of automatic updating of software patches for Microsoft Windows updates that are determined to be critical or important (Microsoft, 2013). With Automatic Updates disabled, Windows could be vulnerable if routine patching is not completed. For this reason, MBSA determined that the local host computer was a "Severe Risk".

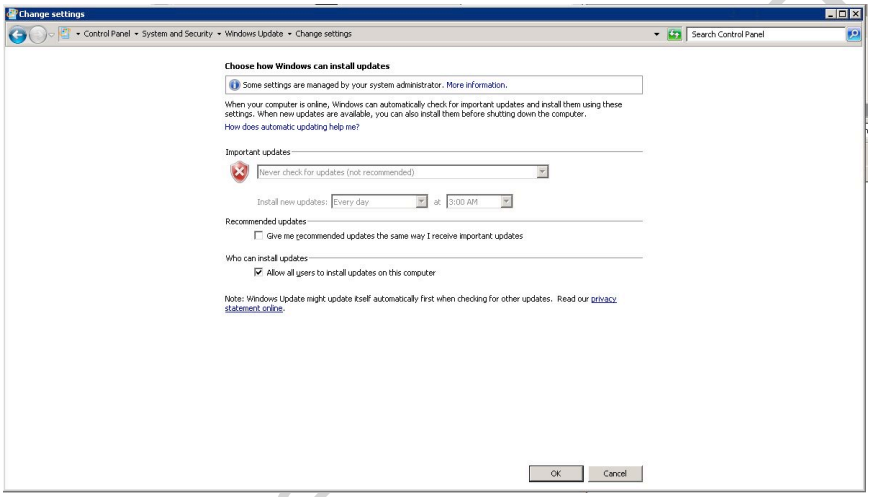

Figure 3. Windows Automatic Update settings in Control Panel

The National Institute of Standards and Technology (NIST) defines patch management as the process for identifying, implementing and maintaining updates for software management (Souppaya & Scarfone, 2013). This includes software such as Microsoft Windows and firmware updates such as print driver software for an Epson printer. According to Vacca (2013), patch management can be cost prohibitive but is necessary for maintaining the integrity of an

information system's environment. Vacca states, "A good practice is to release patches once a month, and then to release them roughly a week after the vendor has published the patch." Additionally, patch management best practices may also include centralized management of software patching and in some cases, the use of specialized tools to facilitate testing and implementation of patching management (Souppaya & Scarfone, 2013). In accordance with patching best practices, MBSA examines computer systems for automatic patching capability and the status of currently installed Windows patches.

Even with tools such as MBSA to facilitate auditing for patching best practices, outdated software continues to pose significant vulnerabilities for security incidents. An information security analysis study published by Solutionary (2015) found that 70% of organizations in the study (n = 18,000) had vulnerabilities that were more than two years old. Older vulnerabilities increase the potential attack surface of an organization and are more likely to result in successful exploitation of an organization's system resources. For example, an Iranian hacking group recently claimed responsibility for the 2013 hacking incident of a New York dam. The Iranian hacking group was able to infiltrate computer systems and locate files with usernames and passwords. Investigations into this incident reveal that lack of software updates is a likely contributor to the breach (Francescani, 2016).

MBSA Local Account Password Check and Strong Password Enforcement

MBSA offers automated scan and command-line options for evaluating password integrity. The automated scanning tool begins its password scanning with the enumeration of user accounts. After user accounts are identified, MBSA performs weak password analysis checks such as testing for blank passwords or a user name as a password. In this experiment,

MBSA performed analysis of weak or blank passwords and determined that two passwords were blank and one password was weak. As reported in Appendix A, results from the first experimental MBSA scan, audit results were informational. This would make sense since the password do not directly affect the performance of the Windows OS. Additionally, MBSA is not limited to Windows accounts. It will also test SQL server accounts if configured to do so. (Microsoft, 2007). SQL server analysis was beyond the scope of this experiment.

Strong password policies on local computer reduce the risk of potential compromise. According to Vacca (2013), strong passwords use multiple words or mixed word and alphanumeric combinations. The purpose of strong password should be to increase the complexity in order to reduce the chance of compromise by such tactics as dictionary attacks. Organizations such as SANS Institute for the Internet, use policies for strong password construction in combination with other guidelines to restrict password use (SANS, 2014). Recommended policy restrictions include limiting use of the same passwords across multiple accounts and requiring different passwords for system-level accounts such as root in Linux. Vacca (2013) also advocates for the use of technologies such as Virtual Private Networks (VPN) to further secure access to restricted computing resources. Through the use of supplemental polices that define limitations on password use and technologies that provide further security, strong password policies can reduce the risk of compromise through dictionary, brute-force and simple password guessing tactics.

Password expiration policies can provide added security to strong password, but current support of password expiration policies has diminished due to the risks created by such policies (Scott, 2016; Knudsen, 2016). Password expiration policies reduce the opportunity for hackers to

use stolen or compromised credentials. If expiration is combined with enforcement of unique passwords over a period of renewals, then systemic password integrity may be further improved since this protocol adds variation to the set of passwords employed by one specific user. Recently, Martin (2013) spearheaded research concerning the effectiveness of passwords and the respective vulnerabilities that could negate policies such as strong password enforcement. One argument put forth is that users are experiencing password overload. By requiring users to adopt and remember more passwords, users are more likely to adapt poor password practices. Since poor practices conflict with use of better passwords, Martin cites using passwords only where needed and allowing users to securely store passwords.

Malware Disablement of Windows Defenses

Malware uses multiple avenues to infect victim computers. Methods of infection include email attachments, direct file upload to a host computer, remote code execution through direct system access or carefully designed network data packets, and USB device transfer. Once malware installs, the next step is to gain a foothold within the victim computer. On Windows computer, one modern method to obtain account credentials is through a software script known as Mimikatz which may be run to extract Windows local user account credentials. With credentials, an attacker may be attempt access to administrative credentials or creation of separate user account (Dalpy, 2014). While Mimikatz was originally created for IT personnel, it is publically available on GitHub and has been cited in use by attackers (Mimoso, 2016). Beyond gaining access and administrative-level access on a host, malware may also need to remove threats and use obfuscation to maintain persistence and the capability to perform tasks.

Antivirus and local firewall software are two of the biggest threats to malware's existence. For this reason, malware is known to incapacitate defensive software in order to

establish full control within a victim computer. Within Windows software, common avenues to interrupt firewall and antivirus software include disruption of processes and services as well as re-configuration of the Windows registry. An example to illustrate this approach is the worm W32/Chode-J (Fitzgibbon & Wood, 2009). W32/Chode-J spreads to its victims through instant messaging services such as MSN Instant Messanger. It begins installation through a series of Windows registry entry creations or updates:

- HKCU\Software\Microsoft\Windows NT\CurrentVersion\Runcsrss "<System>\<random>\csrss.exe"
- HKCU\Software\Microsoft\Windows\CurrentVersion\Runcsrss "<Program Files>\<Messenger>\msmsgs.exe /background"
- HKLM\Software\Microsoft\Windows\CurrentVersion\Runcsrss "nwiz.exe /installquiet"
- HKCU\Software\Microsoft\Windows NT\CurrentVersion\WindowsLoad "<System>\<random>\csrss.exe"
- HKCU\Software\Microsoft\Windows\CurrentVersion\Explorer\AdvanceHidden 2
- HKCU\Software\Microsoft\Windows\CurrentVersion\Explorer\AdvanceShowSuperHidden 0
- HKCU\Software\Microsoft\Windows\CurrentVersion\Explorer\AdvanceSuperHidden 0
- HKLM\SOFTWARE\Microsoft\WindowsNT\CurrentVersion\SystemRestoreDisableSR 1
- HKLM\SYSTEM\CurrentControlSet\Services\SharedAccessStart 4
- HKLM\SYSTEM\CurrentControlSet\Services\srservice Start 4

The registry changes quietly set up W32/Chode-J on the victim host and then establish persistence within Windows. Next, W32/Chode-J sets local IP addresses for antivirus software such as Kasperspky to 127.0.0.1 which removes the ability for antivirus software to contact parent providers for software definition updates. IP Address 127.0.0.1 is a default IP address for a local computer, and creates a loopback effect that restricts access to outside network addresses. Finally, Windows services such as msconfig and regedit are disabled in order to limit access to Windows resources that permit corrections to registry and services (Microsoft, 2016). W32/Chode-J also attempts to disable name-brand antivirus and firewall software processes such as Kaspersky (Fitzgibbon & Wood, 2009). By disabling processes, defensive software cannot function in Windows.

As observed in this examination, Windows System registry keys RestrictAnonymous and EveryoneIncludesAnonymous were evaluated for manipulation. These types of keys could be adjusted by malware to facilitate user account access with limited tractability. MBSA did not report anonymous user access as a severe issue, but highlighted the issue as a warning in audit results as described in Table 2.

Malware may also creatively use services within a host computer for installation and functionality. One example is the trojan Win32/Jowspry. Win32/Jowspry uses a built-in Microsoft service known as Background Intelligent Transfer Service (BITS) to download data to a victim computer from the Internet (Microsoft, 2007). Since Win32/Jowspry uses the Windows-native BITS service, it can bypass firewalls and other security software. BITS is used by Windows Update to maintain software-patching updates, but when employed by malware, it becomes a valid way to bypass firewall security (Reimer, 2007).

Malware can also manipulate software within the Windows environment to legitimately install and then take actions to limit the capabilities of anti-virus software. One example is Vonteera adware that is a type of spyware. The goal of spyware software is to persist on a computer in order to interact with users and steal information. Vonteera installs and set up through Internet browser software. The use of Internet browser software offers a legitimate method to gain access to a victim computer (Williams, 2015). Vonteera installs within the plug-ins or extensions software of an Internet browsers. Plug-in or extensions add software functionality to a browser. Vonteera then goes beyond an Internet browser and embeds itself in Windows shortcut files. Through shortcuts to an Internet browser software, Vonteera established the default-landing page on browser launch. Most cleverly, Vonteera limits access to anti-virus companies by placing anti-virus company security certificates in "Untrusted Certificates" status. This action prevents the running of antivirus software. Vonteera is able to manipulate legitimacy without having to disable antivirus or firewall software.

Examination of Conficker and Role of MBSA

According to the National Vulnerability Database, Conficker, also known as CVE-2008-4250, is described as a vulnerability in Microsoft's Server service "in Microsoft Windows 2000 SP4, XP SP2 and SP3, Server 2003 SP1 and SP2, Vista Gold and SP1, Server 2008, and 7 Pre-Beta allows remote attackers to execute arbitrary code via a crafted RPC request that triggers the overflow during path canonicalization (NIST, 2008). Remote Procedural Call (RPC) is a protocol where one computer requests programmatic resources on a remote computer without the need to communicate using network protocols. While RPC may be used for legitimate purposes such as file sharing or printing, it could also be a severe vulnerability if an attacker gained such access. In the case of Conficker, attackers used RPC to directly interact with a remote host and execute

programmatic code that manipulated file path memory calculations (canonicalization). The specific error that occurs is the copying of wide-character strings across multiple computer memory locations to a point where computing resources fail, and an attacker is able to implement malicious code (Burton, 2009; Hypponen, 2009). It should be noted that Conficker went on to have several predecessors and became a Swiss-army knife of tools to compromise Windows computers

According to Microsoft (2008), using Conficker, an "attacker could exploit this vulnerability without authentication to run arbitrary code. It is possible that this vulnerability could be used in the crafting of a wormable exploit." One of Conficker's most powerful capabilities is the ability to infect other computers networked computers. To accomplish this task, Conflicker installs itself as a service within its Windows host (Burton, 2009). Once installed, it identifies its victim host to a remote computer server and then downloads and installs an HTTP server. Using the HTTP server, Conflicker is able to scout for other potential victim hosts. Once target hosts are identified, Conficker is able to direct infection of other hosts.

In addition to use of HTTP, Conficker has other infection-spreading capability (Fitzgibbon & Wood, 2009). Conficker determines the Windows version of its victim and adjusts its functionality accordingly. For earlier Windows versions, Conficker is able connect anonymously to other victim computers using the Windows Server service to access Network Shares. Later variants of the Conficker virus spread infection by compying to connected USB devices. Early Conficker have brute-force capability to conduct dictionary attacks in an effort to compromise victims. Conficker eventually grew into a bot-net network due to its successful implementation (Fitzgibbon & Wood, 2009).

Finally, Conficker has surveillance, obfuscation, and data exfiltration capability. Conficker is able to detect virtualized machine (VM) characteristics which may suggest malware analysis (Fitzgibbon & Wood, 2009). If Conficker identifies characteristics of being locked within a VM, it stops functioning. Conficker uses several obfuscation techniques to mask its presence. One technique is random-naming of the service installed on a victim host (Burton, 2009). Conficker has file-copy capability for self-distribution, and the same copy capability could be used to copy and exfiltrate data.

Since 2008, Conficker continues to be a persistent threat. One example of Conficker continued persistence is its infection of police body cameras (Goodin, 2015). Cameras manufactured by Martel Electronics were found to have a Conficker variant installed. Security firm iPower tested the cameras and confirmed Conficker's functionality. Further, Conficker's global presence continues to be part of Microsoft's security program. In Ireland for example, the Conficker virus was recently found to be a prominent threat of infected devices in Dublin (O'Brien, 2016).

Conficker continues to rely on faulty configurations and design issues within Windows in order to successfully spread across victim computers (Asghari, Ciere & Van Eaten 2015). Use of tools such as MBSA to evaluate Windows computers for system configurations and patching provides the means to achieve a high-level health check within an organization. Used across a network, MBSA can be used to identify particular computers with suspect changes or disablement of Windows system settings and configurations. An example of disablement would be turning off Automatic Updates, and an example of configuration changes would be adjusting Windows System registry RestrictAnonymous keys to permit anonymous logon. Since

Conflicker continues to evolve in its capabilities, recurring use of MBSA is necessary to monitor recurring security policies within Windows environments.

MBSA and Feature Suggestions

MBSA addresses core security configurations within Windows, but could be extended in functionality. First, MBSA could be updated to incorporate the use of customizable rules within an organization. One example of custom configuration would be adjusting for the use of Roaming Profiles. Roaming Profiles enabled employees to access their data anywhere within an organization. While beneficial to users, Roaming Profiles can pose security issues if the profiles are not routinely deleted. If MBSA could be customized to check for Roaming Profile user folders, IT personnel could determine whether group policies are properly set a the local level. Secondly, MBSA could be customized to further examine Windows system artifacts for suspicious activity. One example would be the examination of the Windows Task Scheduler. The Windows Task Scheduler may be used by viruses to maintain a persistent presence on a victim machine. By baselining Task Scheduler, an organization may be able to locate suspicious activity not consistent with an organization's typical task scheduler functionality. This is particularly important in server environments where servers have consistent patterns of activity but are also highly valuable targets for external attackers. Third, MBSA should conduct a deeper analysis of Windows registry keys where changes would be unusual. AN example is a change to System registry keys for keyboard language which can be highly unusual and can indicate suspicious intruder activity. Finally, MBSA should examine Windows Security audit logs for Remote Logon Type 10. This event may be acceptable to an organization but it is also classically associated with brute-force attack attempts on Windows computers. In incident response on

Windows computers, this Logon Type may be one of the first indicators reviewed. For this reason, it would be a good candidate for MBSA review.

References

Asghari, H., Ciere, M., & Van Eaten, M. (2015). Post-mortem of a zombie: Conficker cleanup after six years (Working paper). Retrieved https://www.usenix.org/system/files/conference/usenixsecurity15/sec15-paper-asghari.pdf

Burton, K. (2009). The Conficker worm. Retrieved from https://www.sans.org/security-resources/malwarefaq/conficker-worm.php

Cobb, M. (2010). Microsoft security tools: MBSA and MSAT explained. Retrieved from http://www.computerweekly.com/tip/Microsoft-security-tools-MBSA-and-MSAT-explained

Dalpy, B. (2014). Mimikatz (Version 2) [Computer software]. Retrieved from https://github.com/gentilkiwi/mimikatz

Fitzgibbon, N., & Wood, M. (2009). Conficker.C: A technical analysis. 45-64. Retrieved from https://www.sophos.com/medialibrary/PDFs/marketing%20material/confickeranalysis.pdf

Francescani, C. (2016). U.S. Infrastructure Can Be Hacked With Google, Simple Passwords: Experts. Retrieved from http://www.nbcnews.com/news/us-news/u-s-infrastructure-can-be-hacked-google-simple-passwords-n548661

Goodin, D. (2015). Police body cams found pre-installed with notorious Conficker worm. Retrieved from http://arstechnica.com/security/2015/11/police-body-cams-found-pre-installed-with-notorious-conficker-worm/

Hypponen, M. (2009). The Conficker mystery. Retrieved from http://www.blackhat.com/presentations/bh-usa-09/HYPPONEN/BHUSA09-Hypponen-ConfickerMystery-PAPER.pdf

Johnson, S. 1. (2009). *MCITP Windows Server 2008 Enterprise Administrator: Study Guide*. Indianapolis, IN: Wiley Pub.

Knudsen, J. (2016, March). Create strong passwords for small business. Small Business Computing.com. Retrieved from http://www.smallbusinesscomputing.com/tipsforsmallbusiness/create-strong-passwords-for-your-small-business.html

Martin, C. (2016). Simplifying your approach: Password guidance. Retrieved from: https://www.gov.uk/government/uploads/system/uploads/attachment_data/file/458857/Password_guidance_-_simplifying_your_approach.pdf

Microsoft. (2013). Description of the standard terminology that is used to describe Microsoft software updates. Retrieved from https://support.microsoft.com/en-us/kb/824684

Microsoft. (2007). How To: Use the Microsoft Baseline Security Analyzer. Retrieved from https://msdn.microsoft.com/en-us/library/ff647642.aspx

Microsoft. (n.d.). How to use the RestrictAnonymous registry value in Windows 2000. Retrieved from https://support.microsoft.com/en-us/kb/246261

Microsoft. (2015, January 9). Microsoft baseline security analyzer (Version 2.3) [Program documentation]. Retrieved from https://www.microsoft.com/en-us/download/details.aspx?id=7558

Microsoft. (2008). Microsoft security bulletin MS08-067 – critical: Vulnerability in server service could allow remote code execution (958644). Retrieved from https://technet.microsoft.com/library/security/ms08-067

Microsoft. (2007). TrojanDownloader:Win32/Jowspry. Retrieved from https://www.microsoft.com/security/portal/threat/encyclopedia/entry.aspx?ThreatId=-2147383707

Microsoft. (n.d.). Using system configuration (msconfig) - Windows help. Retrieved from http://windows.microsoft.com/en-us/windows/using-system-configuration#1TC=windows-8

Mimoso. (2016). Metel bank robbers borrowing from APT attacks. Retrieved from https://threatpost.com/spree-of-bank-robberies-show-cybercriminals-borrowing-from-apt-attacks/116173/

National Institute of Standards and Technology (NIST). (2008). National vulnerability database: Vulnerability summary for CVE-2008-4250. Retrieved from https://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2008-4250

O'Brien, C. (2016). Microsoft leading the fight to counter cybercrime. Retrieved from http://www.irishtimes.com/business/technology/microsoft-leading-the-fight-to-counter-cybercrime-1.2566206

Reimer, J. (2007). Symantec: Malware can hijack Windows Update, bypass firewall. Retrieved from http://arstechnica.com/business/2007/05/symantec-malware-can-hijack-windows-updateThe /

Scott, B. (2016). The end of password expiry. SC Magazine UK. Retrieved from http://www.scmagazineuk.com/the-end-of-password-expiry/article/482484/

Souppaya, M., & Scarfone, K. (2013). NIST special publication 800-40: Guide to enterprise patch management technologies (Revision 3). 16-17. doi:10.6028/nist.sp.800-40r3

Solutionary. (2015). 2015 NTT global threat intelligence report. Retrieved from https://www.solutionary.com/threat-intelligence/threat-reports/annual-threat-report/ntt-group-global-threat-intelligence-report-2015/

Vacca, J. (2013). *Computer and Information Security Handbook (Second Edition)*. Waltham, MA: Morgan Kaufmann.

Williams, M. (2015). Malwarebytes exposes adware that disables antivirus. Retrieved from http://betanews.com/2015/11/25/malwarebytes-exposes-adware-that-disables-antivirus/

Appendix A

First MBSA Scan of Host Computer

<table>
<tr>
<td></td>
<td>Password Expiration</td>
<td>All user accounts (3) have non-expiring passwords.
<table>
<tr><th>User</th></tr>
<tr><td>Administrator</td></tr>
<tr><td>Guest</td></tr>
<tr><td>User1</td></tr>
</table>
</td>
</tr>
<tr>
<td></td>
<td>Windows Firewall</td>
<td>Windows Firewall is disabled and has exceptions configured.
<table>
<tr><th>Connection Name</th><th>Firewall</th><th>Exceptions</th></tr>
<tr><td>All Connections</td><td>Off</td><td>Programs, Services</td></tr>
<tr><td>Local Area Connection</td><td>Off*</td><td>Programs*, Services*</td></tr>
</table>
</td>
</tr>
<tr>
<td></td>
<td>Incomplete Updates</td>
<td>No incomplete software update installations were found.</td>
</tr>
<tr>
<td></td>
<td>Local Account Password Test</td>
<td>Some user accounts (1 of 3) have blank or simple passwords, or could not be analyzed.
<table>
<tr><th>User</th><th>Weak Password</th><th>Locked Out</th><th>Disabled</th></tr>
<tr><td>Guest</td><td>Weak</td><td>-</td><td>Disabled</td></tr>
<tr><td>Administrator</td><td>-</td><td>-</td><td>-</td></tr>
<tr><td>User1</td><td>-</td><td>-</td><td>-</td></tr>
</table>
</td>
</tr>
</table>

	File System	All hard drives (1) are using the NTFS file system. Drive Letter: C: File System: NTFS
	Guest Account	The Guest account is disabled on this computer.
	Autologon	Autologon is not configured on this computer.
	Restrict Anonymous	Computer is properly restricting anonymous access.
	Administrators	No more than 2 Administrators were found on this computer. User Administrator User1

Additional System Information

Score	Issue	Result
	Windows Version	Computer is running Microsoft Windows 7.

<table>
<tr><td></td><td>Auditing</td><td>Neither Logon Success nor Logon Failure auditing are enabled. Enable auditing and turn on auditing for specific events such as logon and logoff. Be sure to monitor your event log to watch for unauthorized access.</td></tr>
<tr><td></td><td>Shares</td><td>2 share(s) are present on your computer.
<table>
<tr><th>Share</th><th>Directory</th><th>Share ACL</th><th>Directory ACL</th></tr>
<tr><td>ADMIN$</td><td>C:\Windows</td><td>Admin Share</td><td>NT SERVICE\TrustedInstaller - F, NT AUTHORITY\SYSTEM - RWXD, BUILTIN\Administrators - RWXD, BUILTIN\Users - RX</td></tr>
<tr><td>C$</td><td>C:\</td><td>Admin Share</td><td>BUILTIN\Administrators - F, NT AUTHORITY\SYSTEM - F, BUILTIN\Users - RX, NT AUTHORITY\Authenticated Users - D</td></tr>
</table></td></tr>
<tr><td></td><td>Services</td><td>No potentially unnecessary services were found.</td></tr>
</table>

Desktop Application Scan Results

n/a

Administrative Vulnerabilities

Score	Issue	Result
	IE Zones	Internet Explorer zones have secure settings for all users.
	Macro Security	No supported Microsoft Office products are installed.