IT Audit:
Security Beyond the Checklist

Copyright SANS Institute

Author Retains Full Rights

This paper is from the SANS IT Audit site. Reposting is not permited without express written permission.

Interested in learning more?

Check out the list of upcoming events offering
"20 Critical Security Controls: Planning, Implementing and Auditing (SEC440)"
at http://it-audit.sans.orghttp://it-audit.sans.org/events/

Simple Windows Batch Scripting for Intrusion Discovery

GIAC (GSNA) Gold Certification

Author: Tim Proffitt, tim@timproffitt.com

Advisor: Jeff Groman

Accepted:

Abstract

A universal saying in the security world is that there is no completely secure system. With that realization, security practitioners should have a reoccurring procedure in place to determine if their information systems are being compromised by unauthorized individuals. This paper will discuss a procedure that utilizes common tools in conjunction with automated batch scripting to identify successful intrusions into a Microsoft Windows environment.

© SANS Institute 2009, As part of the Information Security Reading Room Author retains full rights.

1. Introduction

With the multitude of different attacks against the Windows operating system and an ever-changing attack landscape, it is just a matter of time before a successful attack will compromise your environment. Malicious entities are steadily crafting new attacks that can utilize an approved firewall rule, evade IPS sensors, elevate its rights in order to obtain sensitive data and ultimately own your information resource. This paper will discuss some common, free tools in conjunction with batch scripting that can be used to identify a successful intrusion on Windows environment. The paper will walk through some logical intrusion areas and add commands until a working script automatically produces auditable data for an investigator to review.

2. Why use Windows batch script?

There are many successful scripting languages that are available to the Windows administrator. Perl, PHP, VBScript, Python, PowerShell and Ruby are all good scripting tools any of which would have the capacity to perform our intrusion discovery effort. Do security administrators really want to introduce complexity when it is not necessary? For the work performed in this paper the simple Microsoft batch scripting language will work easily with no additional software installations, license agreements or compatibility requirements to address.

Batch files are a series of MS-DOS commands typed in a file with one command per line. The batch file uses the MS-DOS character set, has an extension of .bat and runs automatically if you type the file name without the extension.

The philosophy of batch programming is that nearly all of the batch constructs are ordinary commands that can also be used outside of batch scripts. Although some of these commands are virtually never used outside batches, they are still

© SANS Institute 2009, As part of the Information Security Reading Room Author retains full rights.

there in DOS. Most commands used in this script are ordinary DOS commands (Tatila).

3. Generate Trending Data

There are two major reasons for the creation and management of the intrusion discovery script. First, the data that is generated and reviewed can immediately uncover a successful penetration into an information system. Good examples of this would be the identification of a local administrator account titled “0wn3d” or the discovery of a port listening on a known backdoor trojan location. For obvious reasons, these would warrant the immediate attention of an incident response team or security staff.

Second, the script will allow the generation of, and analysis against, a security baseline. The baselining of an information resource can be a great asset to any administrator or auditor. A baseline can be described as a “snap shot” of what an information system should look like in a known good state. A security baseline in the context of this paper would consist of (SANS Institute):

- documentation of all approved users
- running processes
- started and installed services
- expected established connections
- OS version
- ASEP registry settings
- Scheduled Tasks
- Events of interest in the local event logs
- Unusual files and file system configurations

© SANS Institute 2009, As part of the Information Security Reading Room Author retains full rights.

Third, the script will collect the generated data and transfer it to a common location for an administrator to review and archive.

The ability to compare accumulated audit data against a documented baseline will give the administrator more information (Ross 2007). Reviewing the intrusion data from a single audit may not necessarily pick up on a crafty penetration. However, compare audit data with a known security baseline and the administrator has a much better chance of pin pointing the malicious activity. In most cases, developing the baseline of a system as the resource is being deployed into production can be a great starting point. Keep in mind that the usefulness of this discovery effort relies on having an accurate baseline in the first place (Hoelzer, 2008). When baselining a system that has been in service, the administrator may not be able to provide 100% confidence the system is not already compromised in some manner. Malicious entities can go to great lengths in obscuring accounts, renaming unauthorized process, hiding directories or listening on known standard ports.

4. Checking for Version

With the goal to remove any false positives, the first thing to be inserted into the script is the identification of the version of the operating system. Microsoft has been known to significantly change processes, services, local accounts and other operating system variables with the installation of service packs. The footprint for a Windows 2000 RC1 system will look drastically different after the installation of service pack 4. Operating Systems are continually being upgraded by Microsoft and knowing what operating system is being audited and when it was changed will be important when the data is analyzed. Once the operating system has been identified, Administrators can have an understanding of expected, common services and processes. Having the audit script identify the FTP service on a

© SANS Institute 2009, As part of the Information Security Reading Room Author retains full rights.

Windows Vista workstation is significantly different than identifying a Windows 2008 server running FTP on the same segment. Would an administrator be interested when a baseline shows a domain controller running an older service pack than it had been previously?

Microsoft has provided two simple command line tools that are adequate for the job in '**ver**' and '**systeminfo**'**.** The '**ver**' command offers a simple listing of the OS information. Similarly, the '**systeminfo**' command provides a greater amount of info about the OS build, patches installed, physical resources, etc. Either command will meet the needs of this scripting effort. For our intrusion discovery script we will be using the '**ver**' command because of the simple output and ease of use in baselining.

5. Unusual Services

Checking for unusual processes and unauthorized services should be at the top of any investigators list. Be aware that many of the default services can pose a potential entry point into an information system. In 1999, a very popular buffer overflow was identified in the Microsoft IIS FTP service (Insecure.org). The buffer overflow could be used to denial of service the system of physical resources and in some cases execute code remotely. If the FTP server allowed anonymous connections (which was the default setting), a remote attacker could compromise

Tim Proffitt, tim@timproffitt.com

© SANS Institute 2009, As part of the Information Security Reading Room Author retains full rights.

the server. By running the FTP service at this time, the organization's ability to defend itself was at risk until a patch was released. However, not running the FTP service when only WWW was needed resulted in an alternate method to remediate this vulnerability. The unnecessary service should never have been enabled.

Alternatively, the running of the FTP service on a dedicated file server where it had not been recorded on the baseline in the past could be a warning sign that an unauthorized user may be looking to place files on the system. Some viruses and malware will try to install themselves as a service to ensure it gets started again after a reboot. The recently popular Win32/Conficker.D can load itself as a service that launches when the 'netsvcs' group is loaded by svchost.exe. It accomplishes this by adding the generated service to the default list of services found in the registry. The name of the service is randomly generated (Microsoft 2009).

A priority for us will be to review for unauthorized, unknown or unnecessary services that could be generated by a worm like Conficker. Being familiar with the started services and having a baseline can reduce a tremendous amount of effort for the investigator during a review. Running a service that is not necessary for the system should be identified and potentially removed. The removal action is performed on a malicious service so that it may not be started again intentionally or inadvertently.

The built in net commands offer simple and clear output for this section. The '**net start**' command provides the started services of the local system.

© SANS Institute 2009, As part of the Information Security Reading Room Author retains full rights.

```
C:\>net start
These Windows services are started:

   Apple Mobile Device
   Application Layer Gateway Service
   Ati HotKey Poller
   Automatic Updates
   Background Intelligent Transfer Service
   Bluetooth Support Service
   Cisco Systems, Inc. VPN Service
   COM+ Event System
   Cryptographic Services
   Cyberlink RichVideo Service(CRVS)
   DCOM Server Process Launcher
   DHCP Client
   Distributed Link Tracking Client
   DNS Client
   Error Reporting Service
   Event Log
   Help and Support
   HID Input Service
   IBM KCU Service
```

For XP and later systems Microsoft has provided service control (SC) that provides a bit more detail than the '**net start**' command. The '**sc query**' command will provide additional information beyond just the started services that will be desirable for the intrusion discovery review. Using 'sc' commands in the script will provide not only the running service but a description (DISPLAY_NAME), the type of service (TYPE) and its current running state (STATE).

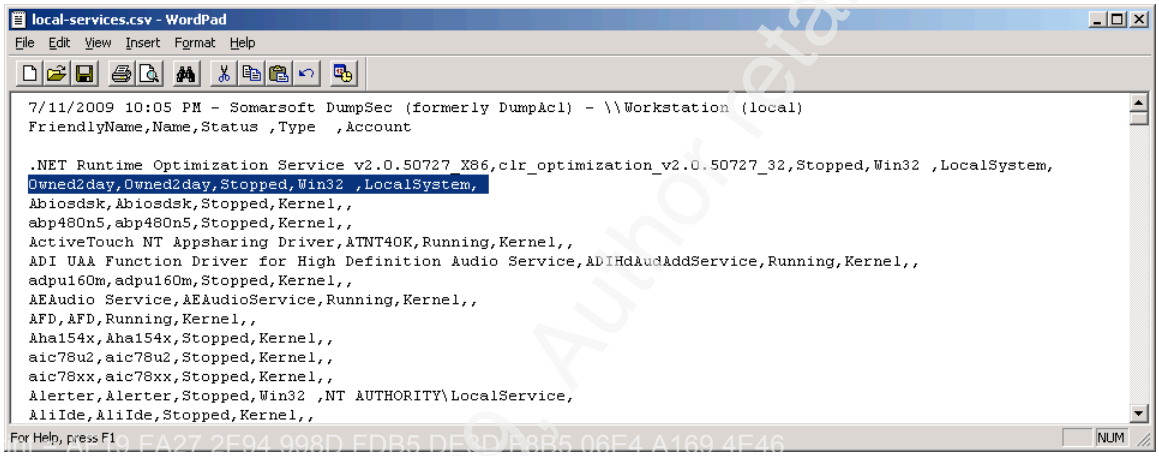
```
C:\WINDOWS\system32\cmd.exe
DISPLAY_NAME: SavRoam
        TYPE               : 10  WIN32_OWN_PROCESS
        STATE              : 4  RUNNING
                                (STOPPABLE,NOT_PAUSABLE,IGNORES_SHUTDOWN)
        WIN32_EXIT_CODE    : 0  (0x0)
        SERVICE_EXIT_CODE  : 0  (0x0)
        CHECKPOINT         : 0x0
        WAIT_HINT          : 0x0

SERVICE_NAME: Schedule
DISPLAY_NAME: Task Scheduler
        TYPE               : 20  WIN32_SHARE_PROCESS
        STATE              : 4  RUNNING
                                (STOPPABLE,PAUSABLE,ACCEPTS_SHUTDOWN)
        WIN32_EXIT_CODE    : 0  (0x0)
        SERVICE_EXIT_CODE  : 0  (0x0)
        CHECKPOINT         : 0x0
        WAIT_HINT          : 0x0

SERVICE_NAME: seclogon
DISPLAY_NAME: Secondary Logon
        TYPE               : 120  WIN32_SHARE_PROCESS (interactive)
        STATE              : 4  RUNNING
                                (STOPPABLE,PAUSABLE,IGNORES_SHUTDOWN)
        WIN32_EXIT_CODE    : 0  (0x0)
        SERVICE_EXIT_CODE  : 0  (0x0)
        CHECKPOINT         : 0x0
        WAIT_HINT          : 0x0
```

A popular alternative to the default Windows **NET** and **SC** commands is the dumpsec tool (formerly known as DumpACL) at www.systemtools.com. Dumpsec is a Windows auditing tool that can query security information such as users, shares, security policy information, permissions, group memberships and more.

Tim Proffitt, tim@timproffitt.com

© SANS Institute 2009, As part of the Information Security Reading Room Author retains full rights.

The script can easily utilize dumpsec to pull our running services with the '**dumpsec /rpt=services /saveas=csv /outfile=c:*.csv**' command**.** The advantages of the dumpsec tool, when querying services, is that it will produce all the services (running or not) in a CSV format for analysis. As you can see below, a service called "0wned2day" has been installed on a target machine but is not currently running.



Because of the additional information the dumpsec tool provides and the output in CSV format, this will be the tool utilized in the script for populating the information on the services of the target.

6. Unusual Processes

The analysis of running processes on an information system can yield significant information about a potential compromise. When targeting systems, malicious entities will utilize attack tools of one sort or another. Regardless of what is used, the tool will need to execute if it is to be utilized by the unauthorized user (Chick 2008). Since malicious code must be executed, it will typically be identified when reviewing running processes. There are some common concepts when investigating unusual processes:

© SANS Institute 2009, As part of the Information Security Reading Room Author retains full rights.

- Trace running processes back to the directory where they were installed. This will provide insight as to whether this is a valid process or an exception.
- Examine the Description, Company Name, and Command Line information for each process. You should be able to identify most of the programs associated with processes as software installed or as a system component of Windows.
- Any processes running from a temp folder should raise a red flag. Malware tends to install itself in and run from locations such as temp (Brandt 2009).
- If a running process points to a DLL in a temp folder, be wary. Typically when something is running from this location, an installation is being conducted.
- smss.exe, winlogon.exe, services.exe, alg.exe, and lsass.exe are critical Windows XP files. Although these files can be compromised, they are system files and will usually be seen in your running list.
- Some forms of malware, distributed as DLL files, hide themselves by using rundll32.exe as a launching pad. The rundll32.exe process is responsible for running DLLs and placing its libraries in the memory. This valid process is normally located at \Windows\System32\rundll32.exe but sometimes spyware uses the same filename and runs from a different directory in order to disguise itself (Chick 2008).
- Check in Microsoft's DLL Help Database to see if an unusual DLL is legitimate. A warning sign will be the administrator identifying a DLL that is not in the DLL Help Database. Although legitimate files can be compromised, identifying an undocumented DLL should cause the administrator to take a closer look at what the DLL may be.

Tim Proffitt, tim@timproffitt.com

© SANS Institute 2009, As part of the Information Security Reading Room Author retains full rights.

For producing the running process list, there are several tools from both Microsoft and other third parties. A popular default Windows command is '**tasklist**'. Used with the "/v" argument, '**tasklist**' will provide a list of the running processes, their PID, memory usage, user account used for starting the process and CPU time. Due to the simplicity and output of the command, **'tasklist'** will be added into the script for documenting the running processes.

```
Image Name                     PID Session Name     Session#    Mem Usage Status          User Name
window Title
========================= ====== ================ ======== ============ =============== ==============================================
==================================================================================
System Idle Process              0 Console                 0         28 K Running         NT AUTHORITY\SYSTEM
System                           4 Console                 0        256 K Running         NT AUTHORITY\SYSTEM
smss.exe                      1260 Console                 0        412 K Running         NT AUTHORITY\SYSTEM
csrss.exe                     1344 Console                 0      4,160 K Running         NT AUTHORITY\SYSTEM
winlogon.exe                  1380 Console                 0      6,916 K Running         NT AUTHORITY\SYSTEM
services.exe                  1424 Console                 0      3,968 K Running         NT AUTHORITY\SYSTEM
lsass.exe                     1436 Console                 0      2,144 K Running         NT AUTHORITY\SYSTEM
ibmpmsvc.exe                  1600 Console                 0      1,516 K Running         NT AUTHORITY\SYSTEM
```

When investigating the audit results of process list, the administrator's interest will be not only unidentified processes but also anomalies where there is high memory usage or CPU utilization. Once again, the use of a baseline when reviewing data from this section can quickly highlight a malicious process.

7. Network Usage

The main purpose for reviewing network usage with the script is to see who is connecting to the system. A file share, a terminal session, an established TCP/UDP connection or any other method to communicate with a resource can be a means for an unauthorized user to manipulate the target. A well documented technique to break into Windows systems when on the same local network is to attack the Windows file and print sharing service (SMB). Assuming that SMB is accessible, the most effective method to compromise this service is the password guessing attempt by connecting to the IPC$ or C$ share (McClure 2005). It is desirable for a malicious entity, once they have valid credentials, to

© SANS Institute 2009, As part of the Information Security Reading Room Author retains full rights.

create a back door which they are able to return to at will. For this reason, administrators will also want to know who has an open share, terminal, TCP port or UDP listener open with the server. Administrators using a baseline will generally have an idea of typical connections to a serving resource. Anything outside the baseline should raise a flag.

Once again, Windows provides a set of default commands that will yield the results needed for this effort. The first on the list is the '**net session**' command. This command will display any active connections to the system. The '**nbtstat**' and '**netstat**' commands can also be used to display additional information about network connections and listening applications.

Win32 Command	Nbtstat	Netstat
Protocol	NetBIOS	TCP and UDP
Displays	Connections to MS File &Print Sharing	Connections to a listening application or service

The '**nbtstat**' command is designed to provide NetBIOS connection information if NetBIOS is enabled on the system. NetBIOS was developed in the early eighties as an API for software communication over IBM's LAN technology. Later, Microsoft created a NetBIOS implementation for its MS-NET networking technology that is still used today (RFC 1001) mostly in Microsoft's file and printer sharing. Windows NT and its successors have achieved a well-deserved reputation for giving away free information to remote pilferers (McClure 2005). This is almost singularly due to the vulnerability of the Windows null session / anonymous connection attack (McClure 2005). The NetBIOS protocol has been identified to be susceptible to DoS style attacks and password cracking; more specifically SMB password guessing (Scambray 2001). As an investigator, interest should be shown in any connections that do not appear to be utilized by the organization. A connection over SMB from an unknown workstation could be

© SANS Institute 2009, As part of the Information Security Reading Room Author retains full rights.

a password brute force attack or a legitimate legacy applications relying on NetBIOS.

The '**nbtstat –s**' command will display each adapter on the system and any NetBIOS connections established with that interface.

```
C:\WINDOWS\system32\cmd.exe
Microsoft Windows XP [Version 5.1.2600]
(C) Copyright 1985-2001 Microsoft Corp.

C:\Documents and Settings\tgprof>nbtstat -s

VMware Network Adapter VMnet8:
Node IpAddress: [192.168.40.1] Scope Id: []

    No Connections

VMware Network Adapter VMnet1:
Node IpAddress: [128.226.1.1] Scope Id: []

    No Connections

Wireless Network Connection:
Node IpAddress: [192.168.1.6] Scope Id: []

    No Connections

\Device\NetBT_Tcpip_{0EA8E39F-D806-4585-80C7-81C6A1395A84}:
Node IpAddress: [0.0.0.0] Scope Id: []

    No Connections
```

The counterpart to '**nbtstat**' is the '**netstat**' command which is designed to provide TCP and UDP information. The netstat command will display:

- Whether a connection is configured for TCP or UDP
- The local systems source port that is configured to communicate
- The destination system the port is communicating with
- Whether the conversation is established or listening

```
C:\WINDOWS\system32\cmd.exe
Microsoft Windows XP [Version 5.1.2600]
(C) Copyright 1985-2001 Microsoft Corp.

C:\

Active Connections

  Proto  Local Address          Foreign Address        State
  TCP            :1157          localhost:27015        ESTABLISHED
  TCP            :1472          localhost:62514        ESTABLISHED
  TCP            :27015         localhost:1157         ESTABLISHED
  TCP            :62514         localhost:1472         ESTABLISHED
  TCP            :3517

C:\Documents and Settings
```

Tim Proffitt, tim@timproffitt.com

© SANS Institute 2009, As part of the Information Security Reading Room Author retains full rights.

The '**netstat**' command would display for the investigator a port that has been opened by a malicious entity over a TCP or UDP port. The command would show an active connection or a port waiting (listening) for the unauthorized user to remote back into.

The information provided from these network commands will be vital to the investigator as the data will provide insight into what the resource will have a conversation with. Understanding what the common configuration is or obtaining a hardening template (such as CIS) used on the resource, will greatly aide in the intrusion discovery review. For most Windows systems, epmap, microsoft-ds, TCP 3389, netbios-ssn, and NTP will be running by default (Microsoft 2009). Each time a new enterprise application is utilized by the organization, you may see a new connection established. However, identifying a TCP connection from an IP address residing in China (using www.dnsstuff.com) established on an ethereal port may warrant further investigation. It is at this point that our baselining information will come in handy. By comparing the latest audit data against a known baseline, administrators can determine unauthorized connections that may not be obvious intrusions. For identifying network connections in our script, '**net session**', '**nbtstat**' and '**netstat**' will be incorporated.

8. Mapping Executables to Listening Ports

Understanding which executable translates to a running process can provide actionable data on whether a process is authorized or not. For typical enterprise systems, it is often difficult to identify what executable is connected to a specific process. In some scenarios, attackers find that their executable hacking tools cannot be renamed or otherwise repackaged. This allows for the investigator to easily identify common / known attack tools such as WINVNC.exe (McClure

© SANS Institute 2009, As part of the Information Security Reading Room Author retains full rights.

2005). As administrators are reviewing the audit data produced by this script, mapping the executables to listening or established ports will help identify authorized applications or malicious code.

'Fport' by Foundstone is a tool for helping identify which executable is tied to a listening port. '**Fport'** reports all open TCP/IP and UDP ports and maps them to the owner application. This is similar information that one would see using the '**netstat –an**' command, but the reason we are using Fport is that it additionally maps ports to running processes with the PID, process name and path. As you can see in the below extract, PID 3045 should draw our attention. In this example we have an unknown process, running on an ethereal TCP port, tied to a suspicious executable in the root of the C: drive.

```
Pid   Process          Port  Proto Path
3045  fgvdq       ->   52500 TCP   C:\atcv.exe
2012  cvpnd       ->   62514 TCP   C:\Program Files\Cisco Systems\VPN Client\cvpnd.exe
3380  MsnMsgr      ->  123   UDP   C:\Program Files\Windows Live\Messenger\MsnMsgr.Exe
6544  iexplore     ->  123   UDP   C:\Program Files\Internet Explorer\iexplore.exe
1492  svchost      ->  123   UDP   C:\WINDOWS\system32\svchost.exe
6544  iexplore     ->  137   UDP   C:\Program Files\Internet Explorer\iexplore.exe
68    mysqld-nt    ->  1306  UDP   C:\Program Files\Cisco Systems\Cisco IPS Manager Express\MYSQL\bin\r
1132  vmware-authd ->  1426  UDP   C:\Program Files\VMware\VMware Player\vmware-authd.exe
2704  iTunesHelper ->  1628  UDP   C:\Program Files\iTunes\iTunesHelper.exe
560   mDNSResponder ->  1840  UDP   C:\Program Files\Bonjour\mDNSResponder.exe
2012  cvpnd        ->  3556  UDP   C:\Program Files\Cisco Systems\VPN Client\cvpnd.exe
```

'**Fport'** will be added to the intrusion discovery script for identifying what applications are using TCP or UDP ports on our target system.

9. Local Accounts

Having the ability to log back into a compromised server after a successful installation of malicious code is at the top of the list of any attacker. The addition of local privileged accounts is a clear sign that should warrant an investigation. Is there a business process that allows for local accounts with elevated rights? Does the organization utilize a server or workstation image? Are local accounts

© SANS Institute 2009, As part of the Information Security Reading Room Author retains full rights.

or service accounts authorized for specific applications to use? In some cases the unauthorized user will create local accounts to run malicious code. As investigators, be aware of local accounts created to masquerade running processes into looking legitimate. For example, the W32/Nimda worm created local accounts on the target (with administrative privileges) and ran the TFTP server to self propagate (Chien 2001). Having answers to these questions is important when identifying new local accounts, specifically those with elevated rights. All user accounts should be enumerated and reviewed to ensure they belong to valid users or approved applications. Intrusion discovery involves not only knowing that someone is trying to break into the system, but also identifying who the intruder is (Escamilla, 1998).

For documenting local accounts, there are several tools at our disposal. The Swiss army knife '**dumpsec'** is already being used in the script and can be utilized here as well. Using the 'users' switch will extract local accounts. The script command will look like:

'dumpsec /rpt=users /saveas=csv /outfile=c:\lsusers.csv /showaudit'

UserName	FullName	AccountTy	Comment
__vmware_user__	__vmware_	User	VMware User
administrator		User	Built-in account for administering the computer/domain
00w300me		User	
user	User	User	Local User Account
ASPNET	ASP.NET	User	Account used for running the ASP.NET worker process (aspnet_wp.exe)
Guest		User	Built-in account for guest access to the computer/domain
HelpAssistant	Remote De	User	Account for Providing Remote Assistance
SUPPORT_388945a0	CN=Micros	L=Redmor	S=Washington

In the above example, the investigator would pay attention to the '00w300me' account. Why would this account be utilized? Is this a service account? Is this account being utilized by an attacker to authenticate to the system? The investigator would want to review the security event log and an established baseline to determine how long the account has been utilized.

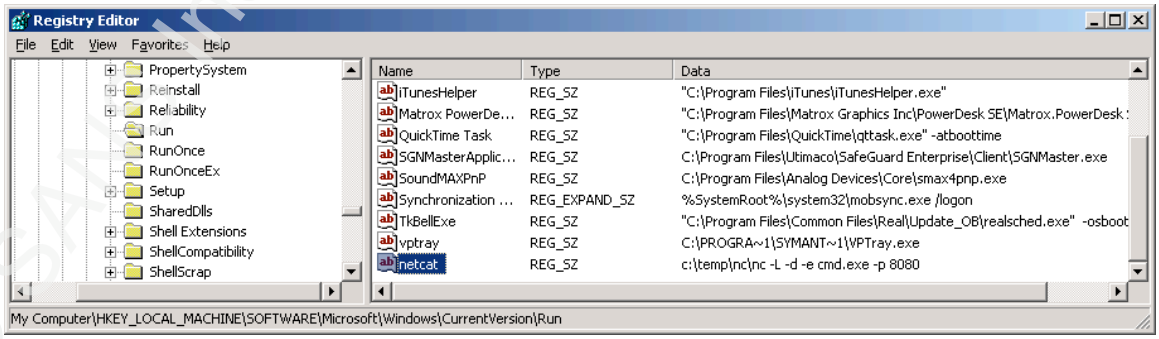
Tim Proffitt, tim@timproffitt.com

© SANS Institute 2009, As part of the Information Security Reading Room Author retains full rights.

10. Registry Entries

The Windows registry is a proprietary database that stores setting and options for the Windows family of operating systems. It contains configuration data about hardware, software, user profiles, and the kernel itself. The registry consists of two basic elements in keys and values which are stored in a logical hierarchy of "hives" (Honeycutt, 1997). Now at this point an entirely different paper can be written on the complexities of this Microsoft configuration database. For the focus of this paper we will take away the knowledge that the registry contains configuration information on which processes and applications will automatically execute on boot up or logon of a user account. Microsoft has started to refer to these locations as "autostart extensibility points" (McClure, 2005).

The two hives we are interested in are HKEY_LOCAL_MACHINE (HKLM) and HKEY_CURRENT_USER (HKCU). The keys 'run' and 'runonce' indicate applications defined to execute automatically. These areas should be checked regularly for the presence of malicious or strange looking commands. It would be advantageous for an attacker to place a netcat listener starting on port 8080 to boot under HKLM\..\..\run. The attacker would now have a perpetual back door into the system (McClure, 2005).



Tim Proffitt, tim@timproffitt.com

© SANS Institute 2009, As part of the Information Security Reading Room Author retains full rights.

To obtain the applications designated to run in the systems registry keys, we will be using the '**reg.exe'** tool provided by Microsoft. This is the command line version of the GUI '**regedit'**. The switch 'export' will create a copy of the specified sub key entries and values into a text format that will be easy to evaluate. Much like the above sections on running processes or services, the administrator should be familiar with each item listed in these registry keys and investigate any entities that are not a specific function of that resource. Here again, this data would be more meaningful if compared against existing baseline.

To run this command to provide the evidence required, the export switch is used. **'Reg export HKLM\Software\Microsoft\Windows\CurrentVersion\Run c:\Run.reg'**

```
[HKEY_LOCAL_MACHINE\Software\Microsoft\Windows\CurrentVersion\Run]
"ccApp"="\"C:\\Program Files\\Common Files\\Symantec Shared\\ccApp.exe\""
"vptray"="C:\\PROGRA~1\\SYMANT~1\\VPTray.exe"
"SGNMasterApplication"="C:\\Program Files\\Utimaco\\SafeGuard Enterprise\\Client\\SGNMaster.exe"
"AppleSyncNotifier"="C:\\Program Files\\Common Files\\Apple\\Mobile Device Support\\bin\\AppleSyncNotif
"SoundMAXPnP"="C:\\Program Files\\Analog Devices\\Core\\smax4pnp.exe"
"iTunesHelper"="\"C:\\Program Files\\iTunes\\iTunesHelper.exe\""
"Matrox PowerDesk SE"="\"C:\\Program Files\\Matrox Graphics Inc\\PowerDesk SE\\Matrox.PowerDesk SE.exe\
"TkBellExe"="\"C:\\Program Files\\Common Files\\Real\\Update_OB\\realsched.exe\"  -osboot"
"H@XXOr"="C:\\A.exe"
```

In this example, the last entry in the CurrentVersion\Run key should set off a flag. The placement of this entry into the run key will allow the a.exe file to execute each time the system is started. To collect data from the most significant startup keys in the registry, add the below locations into the script:

- HKLM\Software\Microsoft\Windows\CurrentVersion\Run
- HKLM\Software\Microsoft\Windows\CurrentVersion\RunOnce
- HKLM\Software\Microsoft\Windows\CurrentVersion\RunOnceEx
- HKCU\Software\Microsoft\Windows\CurrentVersion\Run
- HKCU\Software\Microsoft\Windows\CurrentVersion\Run

Tim Proffitt, tim@timproffitt.com

© SANS Institute 2009, As part of the Information Security Reading Room Author retains full rights.

- HKCU\Software\Microsoft\Windows\CurrentVersion\RunOnceEx

11. Scheduled tasks

With scheduled tasks, administrators can schedule any script, program, or document to run at a time that is most convenient. Scheduled tasks can execute every time Windows starts, at some arbitrary point in the future, be reoccurring or run in the background. Attackers will commonly use the scheduler service to start rogue processes. The scheduler can also be used as a method to gain remote control of a system and to start processes running as the ultra privileged system account (McClure, 2005). Therefore, a very popular method of ensuring a piece of malicious code continues to run on a system is to use the scheduling service. For example, an administrator could find a piece of running malware and promptly kill the task. If the malware has been added into the scheduler, it will simply execute again when the schedule calls the application. As part of this discovery procedure a review of the scheduled tasks on the system will be warranted.

To accomplish a review of scheduled tasks, use the command line features of the built in Windows tools '**at**' and '**schtasks**'. By piping the output to a text file, administrators can obtain the necessary data.

Schtasks > c:\ schtasks.txt

AT >c:\scheduledATtasks.txt

Tim Proffitt, tim@timproffitt.com

© SANS Institute 2009, As part of the Information Security Reading Room Author retains full rights.

```
C:\WINDOWS\system32\cmd.exe
C:\>schtasks

TaskName                                 Next Run Time            Status
======================================== ======================== ===============
sl                                       At system start up

C:\>_
```

In this example the sl.exe has been scheduled to execute at each startup. Here is another good place to point out the importance of the baseline being created. Is this a new item? What business process is this serving? Was this authorized? These are good questions that can help determine if this is a malicious application scheduled to run at each system startup.

12. Events of Interest

Event logs have been a feature in Windows since its original release of NT. The operating system and any application can make use of the log service to report events that have taken place. Examples include failures to start a component, authentication status and completing an action. The system defines three log sources, "System", "Application", and "Security" (Microsoft 2007). The system and application log sources are intended for use by applications and occasionally the operating system, where the security log is only written to by the operating systems local security authority subsystem service.

Event IDs are used to define the uniquely identifiable events that a Windows computer can encounter. The event log is an important part of determining the scope of any breach. In most cases the event logs will provide insight into the type of activities that have been conducted by unauthorized individuals. Even the

© SANS Institute 2009, As part of the Information Security Reading Room Author retains full rights.

deletion of the event logs to cover someone's tracks provides some information about the type of incident administrators would be facing.

When reviewing log events there can be several indications of malicious behavior:

- **When an event service was stopped.** We are interested when services stop because attackers may choose to kill certain services that will make the system easier to attack / launch attacks from. A good example here is an antivirus or HIPS service. Events ID 7023-7042 are most, but not all, events that indicate a service has been stopped.
- **Windows File Protection is not active**. Windows File Protection is in place to ensure Windows system files are not renamed or changed. If an attacker is able to suspend this service, they could replace system files with malicious ones. Event ID 64001-64009 can indicate when WFP is encountering issues.
- **The Telnet Service is started**. The telnet service is an excellent way for a malicious entity to control your system using a remote command line session. Best practices typically keep telnet disabled on Windows platforms as it is not necessary. Event ID 1000 indicates that the Telnet Service has started.
- **The TFTP Service is started**. The trivial file transfer protocol (TFTP) is a simple method to transfer files between two systems. TFTP at the command line will make it simple for a malicious user to transfer files to and from the compromised system. Event ID 257 indicates TFTP was initialized.
- **Failed logon attempts or locked accounts**. System administrators will be interested in failed logon attempts. Massive numbers of failed logon attempts against privileged accounts can indicate a brute force attack. Numbers of failures against a disabled account can indicate a terminated user attempting to gain access. There are many examples that can be inserted here but the point is that monitoring authentication failures can go a long way into

© SANS Institute 2009, As part of the Information Security Reading Room Author retains full rights.

identifying malicious behavior. Event ID 675, 676,672 are common IDs to review.

- **Security log has been cleared**. Suspicions should always be raised on the clearing of the security log as this would be an easy way to cover the tracks of malicious activities. Event ID 517 indicates a cleared log.

Dump Event Log (**dumpel**), a resource kit tool in Windows 2000, is a command-line tool that dumps an event log into a tab-separated text file. The command can also be used to filter for or filter out certain event types. The '-f' switch specifies the filename our script will be creating and the 'l' script can specify any one of the event log repositories. By dumping the event log into a delimited text file, we can easily open the data in a spreadsheet program such as Excel.

C:\dumpel -f securitylog.txt -l security

Since the purpose of the discovery script is to identify security anomalies, administrators will only be extracting the security log. However, if an organization has custom applications that write to one of the other repositories, an additional line in the script can extract those logs as well.

13. File System

In 1993 Microsoft released the NTFS file system with NT 3.1. NTFS supersedes the FAT file system as the preferred file system for Microsoft's Windows operating systems. The primary reason for the switch was NTFS has several improvements over FAT. NTFS improvements include (Microsoft 2007):

- Improved support for metadata
- Use of advanced data structures
- Improved performance and reliability

© SANS Institute 2009, As part of the Information Security Reading Room Author retains full rights.

- Better disk space utilization
- Security access control lists (ACL)
- File system journaling

NTFS allows security from permissions on the share of a file system all the way down to the security access control list on an individual file. The reason administrators are interested in the file system of the target is FAT file systems will be accessible with little to no security applied. A malicious entity could take several approaches from creating a new partition running FAT to the more difficult task of converting of an NTFS partition to FAT. This would allow the intruder to place files or take files from your system without dealing with access control lists. To check the status of the resources file system, the script will utilize the Microsoft command '**chkntfs**'. A '**chkntfs c: > ntfs.txt**' will display whether the disk is running NTFS. Additionally, administrators will want the script to extract any shares established on the system. It would be reasonable to consider an intruder, when taking control of the resource, could create a private share for their own nefarious uses. Dumpsec, being used already, can provide share information.

'dumpsec /rpt=shares /saveas=csv /outfile=c:\shares.csv'

Administrators will want to review the disk file system configuration and any shares being offered by the resource. Each piece of information may help identify a system that has been compromised.

14. Unusual Files

While looking at disk systems administrators should also be checking into space usage. High on the list for the auditing effort should be decreases in major free space. By looking for unusually large files, administrators can find a good

© SANS Institute 2009, As part of the Information Security Reading Room Author retains full rights.

indicator of a malicious entity storing files on your disks. Searching for files that are greater than 10,000 KB may be an indicator of a system that is containing malicious content. This would be a perfect example of how to find pirated movies stored on the disk system from a peer to peer file sharing service.

To perform this task, we can use some simple command line kung fu that will produce a CSV file for review (Skoudis, 2009). Using the FOR command will meet these requirements. The parameters are 'FOR /R [[drive:]path] %variable IN (set) DO command [command-parameters]'. This command walks the directory tree rooted at [drive:]path, executing the FOR statement in each directory of the tree. If no directory specification is specified after /R then the current directory is assumed. If set is just a single period (.) character then it will just enumerate the directory tree. The @echo is the command and will display the file (variable) and the %~zi is the optional syntax that provides the size of the file. The command to list the files on our disk will look like:

'for /r c:\ %i in (*) do @echo %~zi, %i > files.csv'

This 'for' command will display all the files on the system, with their size, into a CSV file called files.csv.

15. Rootkit detection

As any security practitioner worth his salt would tell you, all of the methods being used in this paper can be manipulated to provide false data and cover the tracks of an attacker. With the successful installation of a rootkit on the system, the detection methods described by this paper can be rendered useless. Rootkits are programs designed to infiltrate a system, hide their own presence and provide administrative control or monitoring functionality to an unauthorized attacker (McClure, 2005). Rootkits can manipulate what the task scheduler shows as a

© SANS Institute 2009, As part of the Information Security Reading Room Author retains full rights.

running process, hide administrator accounts, remove files from a directory listing, falsely report listening ports, erase entries in the event logs and alter output from many of the tools we would utilize. An attacker may use a rootkit to replace vital system executables, which may then be used to hide processes and files. It can consist of a program or combination of several programs designed to hide or obscure the fact that a system has been compromised. Typically rootkits require prior access (by exploiting a vulnerability) to install. Typically, a rootkit acts to obscure its presence on the system through subversion or evasion of standard operating system security scan and surveillance mechanisms such as anti-virus or anti-spyware scan. Often rootkits fool users into believing they are safe to run on their systems. One of the most widely used rootkits is known as Hacker Defender. It can be openly downloaded from http://rootkit.hodt.sk. The primary technique utilized by Hacker Defender is to use the Windows API function *WriteProcessMemory* and *CreateRemoteThread* to create a new thread within all running processes. The intent of this thread is to alter the Windows kernel by patching itself in memory to rewrite information returned by API calls to hide hxdef's presence (McClure, 2005).

There are several other popular rootkits that can be found on Windows systems

- FU Rootkit. Uses a user-mode dropper and a kernel-mode dropper.
- Vanquish. A DLL injection-based Rootkit that hide files and registry entries.
- AFX. Replaces iexplore.exe and explorer.exe with compromised files.

Rootkit Revealer by Sysinternals is a functional rootkit detection tool that is actively being updated and works well with operating systems newer than Windows 2000. Rootkit Revealer is an advanced rootkit detection utility. It runs on Windows and its output lists registry and file system API discrepancies that

© SANS Institute 2009, As part of the Information Security Reading Room Author retains full rights.

may indicate the presence of a user-mode or kernel-mode rootkit. Rootkit Revealer successfully detects many persistent rootkits including AFX, Vanquish and Hacker Defender (Cogswell, 2006). After an initial installation on the target system, Revealer can be called via command line (using the '–a' switch) and provide us the data we need for our investigations.

'**rootkitrevealer.exe -a rootkit.log**'

Knowing whether the system is reporting accurate information is vital to our intrusion discovery effort. By successfully landing a rootkit on a resource, a malicious entity can continue to pry deeper into the organization with little risk of detection. It is for this reason the discovery script will utilize a rootkit detection tool.

16. Create the Script

The simplicity of the Windows batch scripting allows placement of the tools described in this document into a simple yet effective script to be executed on a predetermined schedule. To create the simple script, call each of these tools from a text file with a '.bat' extension. Here is a version of the script created in this paper:

```
@echo Running Intrusion Detection Script v1.0
@echo ******************************************

rem **** version, services and processes********************
ver > ver.txt
dumpsec /rpt=services /saveas=csv /outfile=c:\ local-services.csv
Tasklist /v >Tasklist.txt
Net Start >networkservices.txt
Net view \\127.0.0.1 >openshares.txt
Net Session >sessions.txt
nbtstat -s >nbtstat1.txt
```


© SANS Institute 2009, As part of the Information Security Reading Room Author retains full rights.

```
nbtstat -S >nbtstat2.txt
netstat -na >netstat.txt
fport > fports.txt

rem **** shares and users ********************************
dumpsec /rpt=shares /saveas=csv /outfile=c:\shares.csv
dumpsec /rpt=users /saveas=csv /outfile=c:\lsusers.csv /showaudit

rem **** registry ****************************************
Reg export HKLM\Software\Microsoft\Windows\CurrentVersion\Run c:\ Run.reg
Reg export HKLM\Software\Microsoft\Windows\CurrentVersion\RunOnce c:\ RunOnce.reg
Reg export HKLM\Software\Microsoft\Windows\CurrentVersion\RunOnceEx c:\ RunOnceEx.reg
Reg export HKCU\Software\Microsoft\Windows\CurrentVersion\Run c:\ UserRun.reg
Reg export HKCU\Software\Microsoft\Windows\CurrentVersion\Run c:\ UserRunOnce.reg
Reg export HKCU\Software\Microsoft\Windows\CurrentVersion\RunOnceEx c:\
UserRunOnceEX.reg

rem **** scheduled tasks and event logs **********************
AT >c:\ scheduledATtasks.txt
Schtasks > c:\schtasks.txt
dumpel -f securitylog.txt -l security

rem **** file system and large files **************************
chkntfs c: > ntfs.txt
 'C:\> for /r c:\ %i in (*) do @echo %~zi, %i > files.csv'

rem **** rootkit detection ********************************
rootkitrevealer.exe -a rootkit.log

rem **** Copy extracted data files to the repository ************
net use z: \\%your file server and path%
copy *.txt z:\%your file server and path%
copy *.csv z:\%your file server and path%
copy *.reg z:\%your file server and path%
copy *.log z:\%your file server and path%
```

Tim Proffitt, tim@timproffitt.com

© SANS Institute 2009, As part of the Information Security Reading Room Author retains full rights.

```
@echo ************************************************
@echo Script Complete!
```

17. Test, Test and Test Again

Since one objective of this effort is to utilize extracted data to drive investigations, the information produced must accurate, repeatable and the script created in a way as to produce the same results consistently. The above script is pretty straight forward, but it is possible to encounter scenarios where the script will hang or not produce a data file you think it should. The administrator should thoroughly test on each of the target systems before placement into a schedule and creation of the baseline. It will be a much easier investigation if administrators are not chasing false positives when a true incident is discovered.

18. How often to execute the script?

Hourly, once a day, biweekly, monthly, quarterly or each time you interactively logon, are all valid answers to the question of how often to run the script. Information systems tend to be like snowflakes in their uniqueness. Weekly for one organization may be overkill but a different organization will run daily due to the sensitive nature of the target systems. If the organization has a history of risk assessments, this may be a good place to start to help determine the frequency of running the script. It is at this point where the organization should also consider the task of reviewing of the produced data. The script can automatically produce audit data but at some point management must allocate a human brain to analyze it. Is there an advantage when the script is run daily but can only be reviewed bi-monthly due to a busy schedule? Management must find the equilibrium between running the script and analyzing the data. A general

© SANS Institute 2009, As part of the Information Security Reading Room Author retains full rights.

recommendation for a typical organization is to start with a monthly schedule and work toward shortening the interval to a level that can be tolerated. How quickly would an administrator want to know if a service called “nukem” was running on a Windows 2003 domain controller? How quickly should the CSIRT know there are unauthorized privileged local accounts on the Exchange server? When would those responsible for end point protection want to know when the antivirus service was stopped?

19. Analysis

The batch script has been created, executed, and generated audit data. How can this data be useful? Performing analysis on the data sets can be conducted in any number of ways but there are two major camps on how to perform this objective.

The first method is to utilize the power of the spreadsheet. By creating a spreadsheet for each target system and importing the extracted data into it, one can easily create a baseline using basic spreadsheet knowledge. For example, having one spreadsheet tab dedicated to the target system’s processes where each column is a different snap shot in time. Using this method it would be easy to spot when a new service, running process or user account was introduced to the system. More advanced spreadsheet users can macro the importing of the raw data files directly into the identified tabs. By aligning each snapshot in column format, it is clear when a row of data has a new element. Additionally, Excel has a function to display differences in columns in its advanced filter ‘unique records only’.

Tim Proffitt, tim@timproffitt.com

© SANS Institute 2009, As part of the Information Security Reading Room Author retains full rights.

Microsoft Excel - Baseline-Server-3.xls

	A	C	E
1	Quarter 1 - 3/2008	Quarter 2 - 6/2008	Quarter 3 - 9/2008
2	Application Experience Lookup Service	Application Experience Lookup Service	Application Experience Lookup Service
3	Automatic Updates	Automatic Updates	Automatic Updates
4	Background Intelligent Transfer Service	Background Intelligent Transfer Service	Background Intelligent Transfer Service
5	COM+ Event System	COM+ Event System	COM+ Event System
6	COM+ System Application	COM+ System Application	COM+ System Application
7	Cryptographic Services	Cryptographic Services	Cryptographic Services
8	DCOM Server Process Launcher	DCOM Server Process Launcher	DCOM Server Process Launcher
9	DHCP Client	DHCP Client	DHCP Client
10	Distributed Link Tracking Client	Distributed Link Tracking Client	Distributed Link Tracking Client
11	Distributed Transaction Coordinator	Distributed Transaction Coordinator	Distributed Transaction Coordinator
12	DNS Client	DNS Client	DNS Client
13	Error Reporting Service	Error Reporting Service	Error Reporting Service
14	Event Log	Event Log	Event Modifier Service
15	HP Insight Foundation Agents	HP Insight Foundation Agents	Event Log
16	HP Insight NIC Agent	HP Insight NIC Agent	HP Insight Foundation Agents
17	HP Insight Server Agents	HP Insight Server Agents	HP insight NIC Agent
18	HP Insight Storage Agents	HP Insight Storage Agents	HP Insight Server Agents
19	HP ProLiant Remote Monitor Service	HP ProLiant Remote Monitor Service	HP Insight Storage Agents
20	HP ProLiant System Shutdown Service	HP ProLiant System Shutdown Service	HP ProLiant Remote Monitor Service

In this case, where 'Event Log' had been row 14 in the past baseline, the new Event Modifier Service is now an exception. This should raise the red flag for an item to be investigated. This technique can be performed for each section of this paper. Place each data set in a tab of the spreadsheet, analyze for differences and consequently create the baseline.

The second method is to utilize a tool like 'fc' to compare a file for changes. File Compare (**fc**) is a default Windows command designed to compare ASCII files and display the differences. When changes are found, '**fc /N**' will indicate which lines the changes can be found on for the analyst to review. To analyze two different outputs from the tasklist /v command the administrator will run:

fc /N c:\tasklist1.txt c:\tasklist2.txt > taskdiff.txt

Tim Proffitt, tim@timproffitt.com

© SANS Institute 2009, As part of the Information Security Reading Room Author retains full rights.

Note the taskdiff.txt file is the output of the comparison of two different tasklist outputs. In the below example line 114 indicates GoogleUpdate.exe is now a running process when it was not in the past.

```
taskdiff.txt - Notepad
File  Edit  Format  View  Help
Comparing files C:\tasklist1.txt and C:\TASKLIST2.TXT
***** C:\tasklist1.txt
  113:                    0:00:00 DDE Server Window
  114:  GoogleUpdate.exe            3696 Console                    0      3,356 K Running
  115:
  116:  communicator.exe            4056 Console                    0     23,692 K Running

***** C:\TASKLIST2.TXT
  113:                    0:00:00 DDE Server Window
  114:  communicator.exe            4056 Console                    0     23,692 K Running
*****
```

An alternate way to perform this file comparison would be to concatenate all of the extracted data files (tasklist, local users, services, sessions, ports, reg entries, etc.) into one large combined file (using **copy /b** or **cat**) and utilizing the power of '**fc**'. This action can be included as part of the script so that the administrator is working with a single file rather than multiple. In this way the administrator would only be running '**fc**' once and the difference file would list the exceptions all the extracted data. Once the administrator has concatenated the data sets that have been identified as correct, a baseline can be set aside. Each new set of audit data can be compared against this approved baseline and '**fc**' can identify to exactly which line the audit set differs from the baseline.

Regardless of which option is chosen for analyzing the data, the end result is to provide the administrator with audit data to identify when new entities are different than the baseline. Much of the extraction, sorting, correlating, moving and presenting of the data can be automated but in the end it still takes an administrator to review the results.

20. Conclusion

Tim Proffitt, tim@timproffitt.com

© SANS Institute 2009, As part of the Information Security Reading Room Author retains full rights.

A universal saying in the security world is that there is no completely secure system. Knowing that even the systems we may think are highly secured can be compromised; security practitioners need to have a mechanism to determine that their information systems are not being utilized by unauthorized individuals. By utilizing simple, common and free command line tools in a Windows batch script we can position our organization into promptly reacting when an intrusion is successful. Automation with a reoccurring schedule can greatly contain the impacts of a breach and show due diligence in the area of technology security.

Tim Proffitt, tim@timproffitt.com

© SANS Institute 2009, As part of the Information Security Reading Room Author retains full rights.

References

Veli-Pekka Tatila. A Modern Batch Programming Tutorial. Website http://vtatila.kapsi.fi/batch_tutorial.html

SANS Institute. Intrusion Discovery Cheat Sheet v1.4. Pocket Reference Guide. Website http://www.sans.org/resources/winsacheatsheet.pdf

Ron Ross, Stu Katzke, Arnold Johnson, Marianne Swanson, Gary Stoneburner, George Rogers (December 2007). Recommended Security Controls for Federal Information Systems. NIST Special Publication 800-53 Revision 2, Website http://csrc.nist.gov/publications/nistpubs/800-53-Rev2/sp800-53-rev2-final.pdf

David Hoelzer (2008). Auditing Networks, Perimeters and Systems, Book 5, Page 145

Mark@eeye.com. IIS FTP Exploit/DoS Attack. Insecure.org, Website http://seclists.org/bugtraq/1999/Jan/0291.html

Microsoft (July 2009). Article ID 832017. Service overview and network port requirements for the Windows Server system. Website http://support.microsoft.com/kb/832017/

Douglas Chick (2008). 2008s Most Popular Viruses, and Hacking Tools. TheNetworkAdministrator.com. Website http://www.thenetworkadministrator.com/top2004hackertools.htm

Andrew Brandt (2009). Security Tips: Identify Malware Hiding in Windows' System Folders. Website http://pcworld.about.com/magazine/2307p164id120795.htm?p=1

McClure, Scambray and Kurtz (January 2005). Hacking Exposed : Network Secret & Solutions 5th Edition.

Scambray, McClure (2001). Hacking Exposed Windows 2000: Network Security Secrets and Solutions, page 102.

Microsoft (March 2009). Malware Protection Center. Worm: Win32/Conficker.D Website http://www.microsoft.com/security/portal/Threat/Encyclopedia/Entry.aspx?Name=Worm%3aWin32%2fConficker.D

Tim Proffitt, tim@timproffitt.com

© SANS Institute 2009, As part of the Information Security Reading Room Author retains full rights.

Eric Chien (September 2001). Symantec Security Response. W32.Nimda.A@mm. Website http://www.symantec.com/security_response/writeup.jsp?docid=2001-091816-3508-99

Terry Escamilla (September 1998). Intrusion Detection : Network Security Beyond the Firewall , p29.

Jerry Honeycutt (1997). Windows 95 and NT 4.0 Registry and Customization Handbook

Microsoft (May 2007). How to view and manage event logs in Event Viewer in Windows XP. Website http://support.microsoft.com/kb/308427

Microsoft (May 2007). Overview of FAT, HPFS, and NTFS File Systems. Q100108. Website http://support.microsoft.com/kb/100108

Ed Skoudis (June 2009), Command Line Kung FU, Website http://blog.commandlinekungfu.com/2009/02/episode-4-listing-files-and-their-sizes.html,

Bryce Cogswell and Mark Russinovich (November 2006), RootkitRevealer v1.71, Wesite http://technet.microsoft.com/en-us/sysinternals/bb897445.aspx,

Tim Proffitt, tim@timproffitt.com

© SANS Institute 2009, As part of the Information Security Reading Room Author retains full rights.

Last Updated: May 23rd, 2013

Upcoming SANS IT Audit Training

Community SANS @ CANHEIT	**Ottawa, ON**	**Jun 13, 2013 - Jun 14, 2013**	**Community SANS**
SANSFIRE 2013	**Washington, DC**	**Jun 14, 2013 - Jun 22, 2013**	**Live Event**
Community SANS Chicago	**Chicago, IL**	**Jul 08, 2013 - Jul 12, 2013**	**Community SANS**
SANS vLive - SEC566: Implementing and Auditing the Twenty Critical Security Controls - In-Depth	**SEC566 - 201307,**	**Jul 08, 2013 - Aug 07, 2013**	**vLive**
SANS Rocky Mountain 2013	**Denver, CO**	**Jul 14, 2013 - Jul 20, 2013**	**Live Event**
Critical Security Controls Summit	**Washington, DC**	**Aug 12, 2013 - Aug 18, 2013**	**Live Event**
Community SANS Dallas	**Dallas, TX**	**Aug 19, 2013 - Aug 22, 2013**	**Community SANS**
SANS vLive - AUD507: Auditing Networks, Perimeters, and Systems	**AUD507 - 201309,**	**Sep 02, 2013 - Oct 16, 2013**	**vLive**
SANS CyberCon Fall 2013	**Online, VA**	**Sep 09, 2013 - Sep 14, 2013**	**CyberCon**
Network Security 2013	**Las Vegas, NV**	**Sep 14, 2013 - Sep 23, 2013**	**Live Event**
Community SANS Miami	**Miami, FL**	**Sep 16, 2013 - Sep 19, 2013**	**Community SANS**
Network Security 2013 - SEC566: Implementing and Auditing the Twenty Critical Security Controls - In-Depth	**Las Vegas, NV**	**Sep 16, 2013 - Sep 20, 2013**	**vLive**
Mentor Session - SEC 566	**Troy, MI**	**Oct 01, 2013 - Dec 10, 2013**	**Mentor**
Community SANS Washington @ GWU	**Washington, DC**	**Oct 07, 2013 - Oct 10, 2013**	**Community SANS**
SANS Baltimore 2013	**Baltimore, MD**	**Oct 14, 2013 - Oct 19, 2013**	**Live Event**
SOS SANS October Singapore 2013	**Singapore, Singapore**	**Oct 21, 2013 - Nov 02, 2013**	**Live Event**
SANS Chicago 2013	**Chicago, IL**	**Oct 28, 2013 - Nov 02, 2013**	**Live Event**
Community SANS New York	**New York, NY**	**Nov 04, 2013 - Nov 07, 2013**	**Community SANS**
SANS Sydney 2013	**Sydney, Australia**	**Nov 11, 2013 - Nov 23, 2013**	**Live Event**
SANS London 2013	**London, United Kingdom**	**Nov 16, 2013 - Nov 25, 2013**	**Live Event**
Community SANS Toronto	**Toronto, ON**	**Nov 18, 2013 - Nov 21, 2013**	**Community SANS**
SANS vLive - SEC566: Implementing and Auditing the Twenty Critical Security Controls - In-Depth	**SEC566 - 201312,**	**Dec 02, 2013 - Jan 15, 2014**	**vLive**
Community SANS Vancouver	**Burnaby, BC**	**Dec 09, 2013 - Dec 12, 2013**	**Community SANS**
North America ICS Security Summit & Training 2014	**Lake Buena Vista, FL**	**Mar 11, 2014 - Mar 20, 2014**	**Live Event**
SANS OnDemand	**Online**	**Anytime**	**Self Paced**
SANS SelfStudy	**Books & MP3s Only**	**Anytime**	**Self Paced**