Global Information Assurance Certification Paper

How to use Forensic Toolkit v2.0 on Windows NT 4.0 Server

Written by: Maarten van Essen – Landis ICT Services & Consultancy

GIAC – NT certification practical assignment

Index

How to use Forensic Toolkit v2.0 on Windows NT 4.0 Server

Written by: Maarten van Essen – Landis ICT Services & Consultancy

GIAC – NT certification practical assignment

Introduction

This document was written to fulfil requirements for th e practical assignment portion of the GIAC -NT certification. The subject of this paper is the Forensic Toolkit 2.0 by Foundstone (www.foundstone.com). I experienced that there is not a lot of information about the Toolkit on the web or anywhere else. Al the information you see is saying it is a very useful tool and then gives a very short description. I feel that it would be useful to write about this Toolkit. I hope after reading this paper you can work with the T oolkit and maybe fill me up on some subjects. Next to that I hope you learn something about NT security. The references I used can be found at the end of this paper.

Disclaimers

All efforts have been made to ensure the accuracy and completeness of the information contained in this document. However, discovery of new software revisions, new or revised fixes, and new or revised vendor documentation may, at any time, make portions of this document invalid in terms of its applicability in a computing environm ent. Before using the Toolkit in an production environment test it on a test machine.

1. The Forensic Toolkit 2.0

The Forensic Toolkit 2.0 is a suite of very useful tools to help you examine the files on a NTFS disk partition for unauthorized activity. This is a Win32 Command line tool. This version is the latest and can be downloaded at:

http://www.foundstone.com

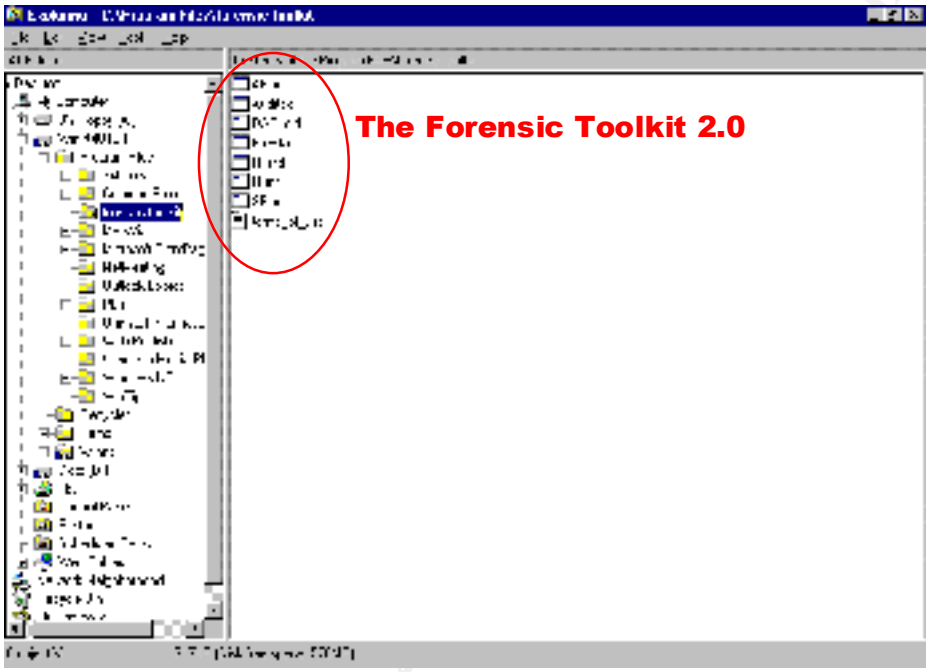
Before using the Forensic Toolkit 2.0, read the Terms_of_use file enclosed with the Toolkit. The Toolkit can be used on Microsoft Windows NT 4.0 server and workstation and also on Microsoft Windows 2000 professional and the server versions. In this paper I will use it on a Windows NT 4.0 Server machine with Service Pack 6a. Keep in mind that it only works on NTFS partitions. Earlier troubles with Service Pack 4 and 5 are solved with this version of The Forensic Toolkit. The minimum requirement for your system are:

Windows NT 4.0 with SP 3 or higher
16 MB Memory
Administrator privileges
Audit log enabled with searchable records
Set NT command line buffer to 500 or more line (recommended 1200)
The command prompt must be a minimum of 80 characters.

2. Installation

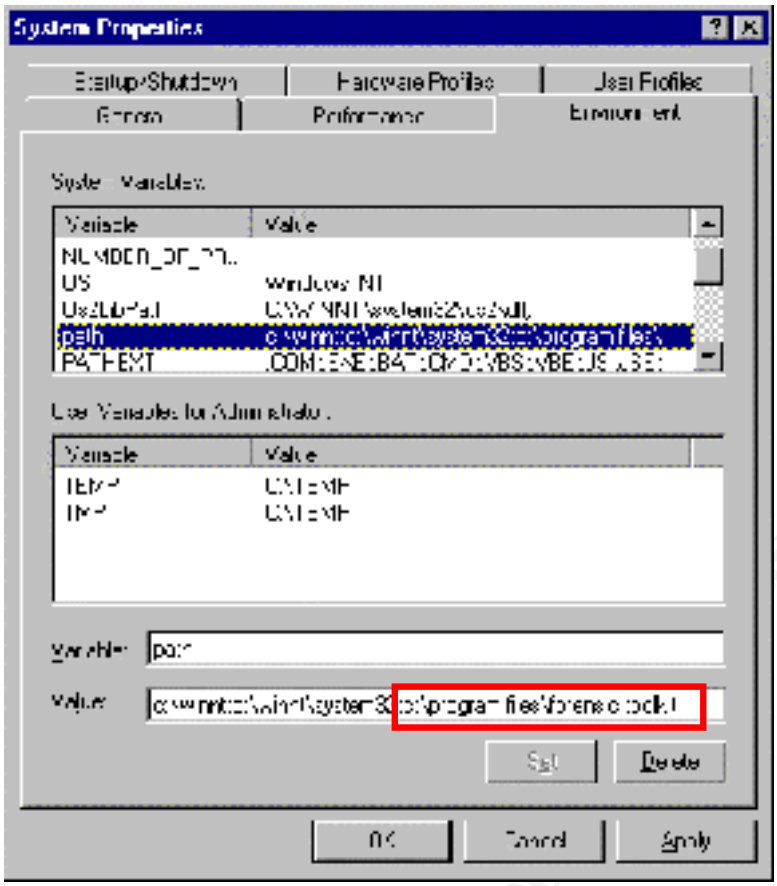
Installation File: Forensicoolkit20.zip, 329KB

As you can see the installation file is an Zip file. So to extract this file you need an unzip program. If you don't have one download it from the Internet. I use WinZip to extract the file and located the file in the directory c: \program files\forensic toolkit. A trail version of WinZip can be downloaded from http://www.winzip.com



Because the Toolkit consists of command line tools we want to be certain that it works within any directory path, so we put the path to the Toolkit directory in the System variables. Here is how we do that.

1. Go to start => Settings =>Control Panel => System
2. On the Environment Tab under System variables choose path
3. At the bottom of the Tab behind Value you add the path to the toolkit directory and click set. I chose the default directory so it looks like this.

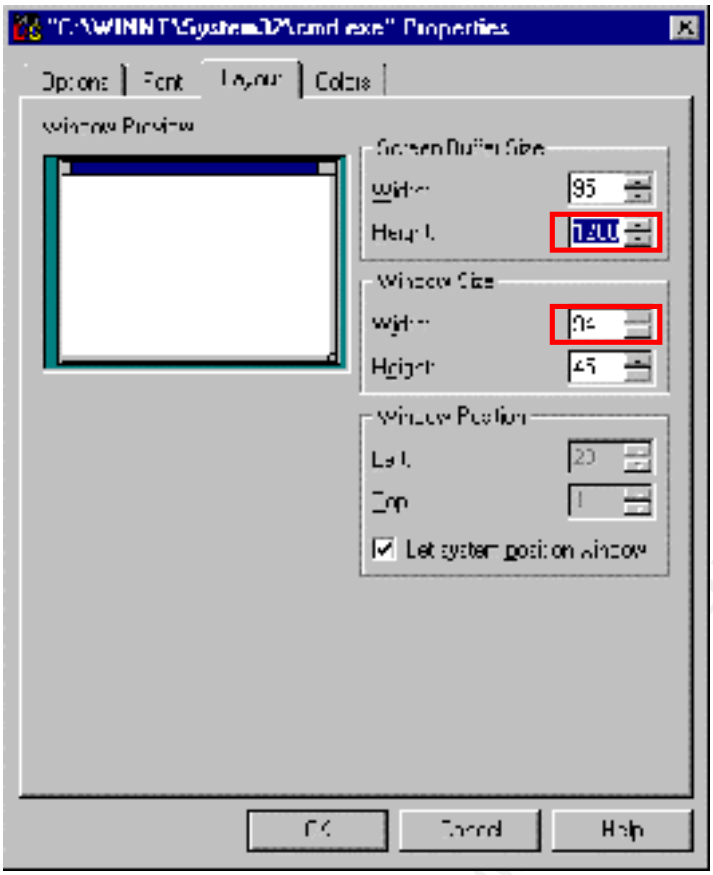


To get the minimum requirements we need to make a few other adjustments to the system. We will start by setting the right windows size width. The command prompt box must have minimum width of 80 characters. How to apply this is shown next.

1. Open a command prompt by going to start => Run and typing CMD (this start up the command prompt.
2. Right click on the MS -DOS logo in the right corner and click on properties.
3. Select the Tab Layout and under Window Size choose a minimum of 80 for Width. I choose 94.

To set the NT command line buffer size to 500 or more lines

4. On the Layout Tab hold down the up -arrow behind the height of the screen buffer size, until it says 500 or more , I used 1200.



5. When you click OK it will ask you if you want to apply it to the current window only or to save the properties for future windows with same title. Choose the last one.

Now you are ready to use the Forensic Toolkit 2.0

Now lets take a look at what the toolkit consists of. There are eight files within the Toolkit. Seven executables and one text document. These are listed below.

C:\>Program Files \forensic Toolkit

File Name	Description
Afind.exe v2.0	Tool that lists files by there last access time
Hfind.exe v2.0	Scans disk for hidden files
Sfind.exe v2.2	Scans disk for hidden data streams
FileStat.exe v1.4.1	Makes a dump of all file and security attributes
Hunt.exe v1.2	Provides a quick way to see if a server reveals too much info via NULL sessions.
Audited.exe v1.5	Generates a list of files being audited by the system
DACLchk.exe v1.2	Dumps any ACL that has Denied and Allowed ACE's in reverse order
Terms_of_use.txt	Terms of use

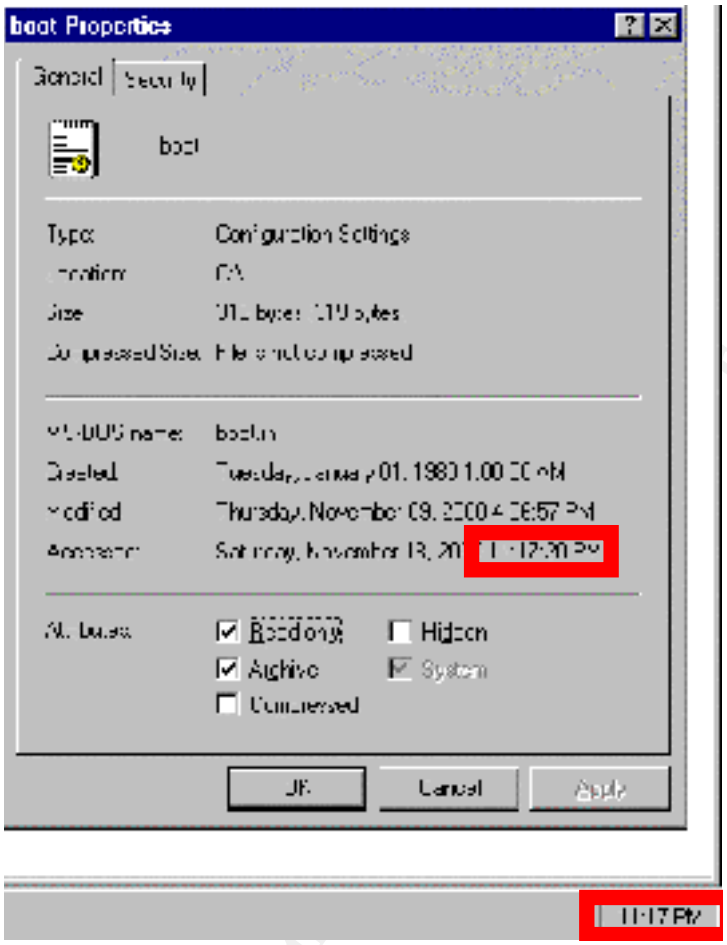
Lets take a closer look at the toolkit commands.

3. Forensic Toolkit Command options and syntaxes.

For all commands applies that " –" as well as "/ " can be used at the beginning of an option. Sometimes it is useful to save the output of the commands to a text file. This can be done with every command in the toolkit. To do this use **> [filename.txt]** at the end of a command. The result can then be viewed in the text file.

4. Afind.exe

As listed above this tool can be used for listing files by there last access time. The nice thing is that it does this without tampering the date the way that windows Explorer.exe does.



As you can see is the Accessed time of this file the same as the time on the computer. This is no coincidence. If you access the proper ty tab of a file, you really access the file, so the date and time are not very useful. If we use afind on the other hand, it does not change anything.

Here you can see the syntax and the options you can use for Afind.exe

Command options	**What it does**
Afind [dirname]	Specifies the directory to search
Afind –ns	Exclude the subdirectories of the directory specified
Afind –f [filename]	Lists the last access time of the file specified.
Afind –s [seconds] (***x***) or ([***y***-***z***])	This option is used to list files accessed less then ***x*** seconds ago or files accessed between ***y*** and ***z*** seconds ago.
Afind –m [minutes] (***x***) or ([***y***-***z***])	This option is used to list files accessed less then ***x*** minutes ago or files accessed between ***y*** and ***z*** minutes ago.
Afind –h [hours] (***x***) or ([***y***-***z***])	This option is used to list files accessed less then ***x*** hours ago or files accessed between ***y*** and ***z*** hours ago.
Afind –h [days] (***x***) or ([***y***-***z***])	This option is used to list files accessed less then ***x*** days ago or files accesse d between ***y*** and ***z*** days ago.
Afind –a **(**[day/month/year-hours:minutes:seconds]**)** or **(**[day/month/year-hours:minutes:seconds]-[day/month/year-hours:minutes:seconds]**)**	This option is used to list files that have been accessed after a certain date and time **or** between certain dates and times. (You must use the whole syntax for the command to work. You can not specify only a date.)
Afind or Afind /?	Get help. A summary of the options

5. Hfind.exe

Hfind.exe looks for hidden files on a disk or a specific directo ry. The syntax for Hfind.exe is:

Hfind [path] –ns

The path is the directory you want to search in. If you use the –ns option it will not search the subdirectories of the given path. The option /? Can also be used to get information about the command.

6. Sfind.exe

Sfind.exe helps you finding hidden data streams. First I will explain what these are and we will talk about the command sfind.exe

6.1 Streams

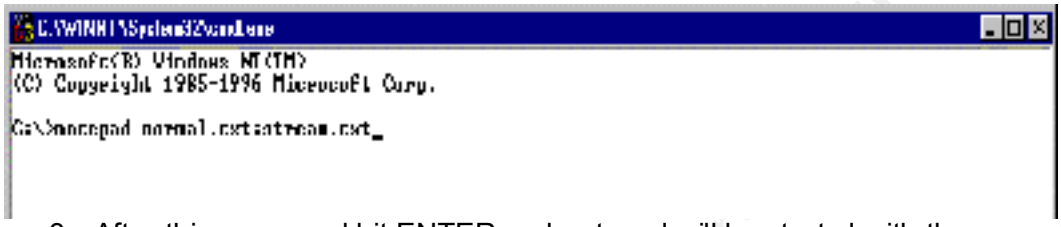
The file system of Windows NT, NTFS has a feature that is not well documented. This feature is called Alternate Data Streams (ADS). There are more names for this, but I will use ADS.

On an NTFS volume files consist of at least one stream, these are normal viewable files. The streams that we are going to talk about are alternate named streams that are not viewable to ordinary NT Tools.

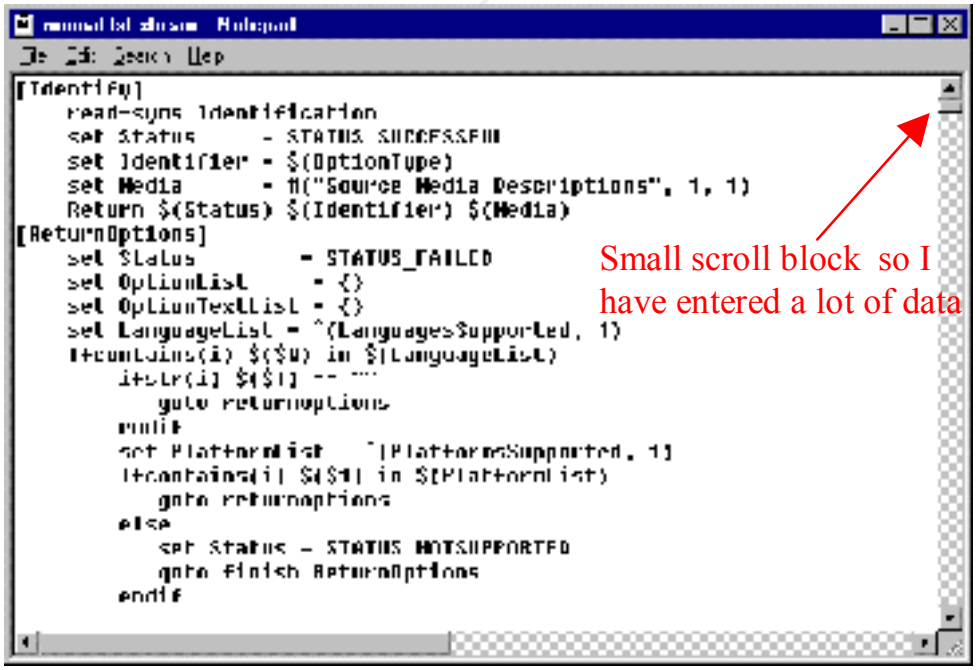
What it means is that we can hide information on a disk without anyone knowing, because we can not see the data.

I will now show you how it works.

1. open a command prompt by going to start => run and type in cmd and clic k OK
2. Now we are going to create a hidden data stream with the use of a normal application: Notepad. At the command prompt type notepad normal.txt:stream.txt

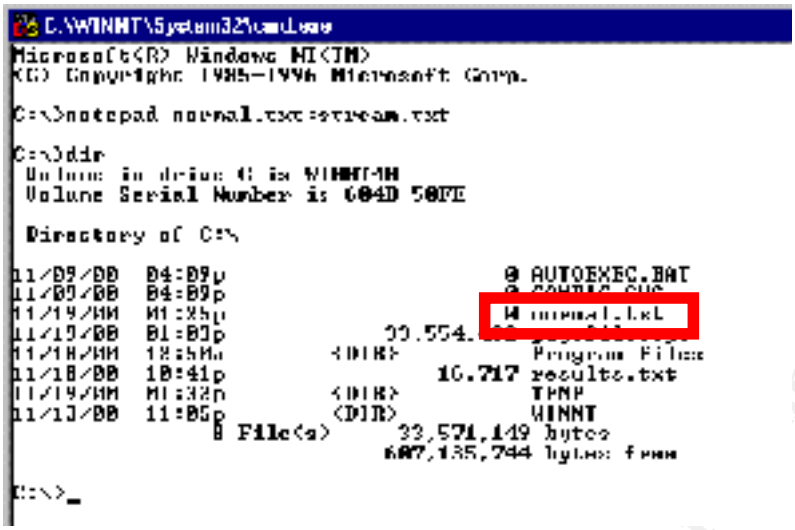


3. After this command hit ENTER and notepad will be started with the ques tion *Cannot find the normal.txt:stream.txt. Do you want to create a new file?* Answer YES.
4. Notepad is ready now. Now we enter a large amount of text in notepad. The way I did it was to use a *.INF file and copied and pasted it a few times. See here the resu lt.



5. Now we are going to save the file. Go to File => Save and the file will be saved.

6. Now lets take a look at the file. In the command prompt box type DIR and look for the file normal.txt. If you did it right you see the follo wing



7. It seams that there is no data in normal.txt because the size is 0 bytes. This is true. If you open the file with notepad there is no text to read. There is only a sort of link with your ADS through this file. You can try this by typing notepad normal.txt:stream.txt and now you see you text again.

You can understand that this is a perfect hiding place for data. Users who are aware of this feature can use it to come around disk quota, without an administrator ever noticing . Attackers can use it as a hiding place for data also, for example a virus or some batch files. They can also use it as a form of a Denial of Services and flood the hard disk.

The solution to this problem is to look for ADS with the use of Sfind.exe. Sfi nd.exe looks for ADS on a NTFS partition or a specific directory on that partition. The syntax for Sfind.exe is:

Sfind [path] –ns

The path is the directory you want to search in. If you use the –ns option it will not search the subdirectories of the give n path. The option /? Can also be used to get information about the command. Lets take a look how it works

1. Open an command prompt (how to? Look at earlier demonstrations)
2. At the command prompt type sfind c: \ and it will go searching for hidden streams. Here is the output that I got.

What we see here is the ADS we created in the previous demonstration. As you can see I experimented a bit with it before I wrote this paper, because there is an other hidden stream normal.txt:hidde n.txt.

To delete these ADS just delete the file that it is associated with. In this case it is normal.txt. Even if a, in this case normal.txt file is hidden, sfind will find the ADS. Another thing to remember is that a attacker can use one single referen ce file (for example normal.txt) to hide more then one document. The way to do this is to use for example the command notepad normal.txt:hidden.txt and put text in and after that you use notepad normal.txt:stream.txt and put data in that to. Then to open them type notepad normal.txt:hidden.txt or notepad normal.txt:stream.txt and the data from both files can be used.

7. Filestat.exe

Filestat.exe dumps NTFS security, file and stream attributes. The syntax for the command is:

Filestat [Filename]
Do not forget the path if you are not in the same directory as the file, except when the path is already in the system variables path.

Example:

I made a dump of the file c: \normal.txt, the file we used to make ADS. I used this file to show you that Filestat also d umps stream information (look at the bottom of the dump).

Tip: Use an output file to dump the information, because there are more lines then your command line box can handle. How to do this is described in the beginning of this paper.

Dump output.txt (th is is a copy out of the text file)

```
Dumping c:\normal.txt...
SD is valid.
SD is 184 bytes long.
SD revision is 1 == SECURITY_DESCRIPTOR_REVISION1
SD's Owner is Not NULL
SD's Owner-Defaulted flag is FALSE
  SID = BUILTIN/Administrators  S-1-5-32-544
SD's Group-Defaulted flag is FALSE
  SID = NT4/Domain Users  S-1-5-21-664811115-334825431-2087665911-513
SD's DACL is Present
SD's DACL-Defaulted flag is FALSE
    ACL has 5 ACE(s), 120 bytes used, 0 bytes free
    ACL revision is 2 == ACL_REVISION2
  SID = BUILTIN/Administrators  S-1-5-32-544
    ACE 0 is an ACCESS_ALLOWED_ACE_TYPE
    ACE 0 size = 24
    ACE 0 flags = 0x00
    ACE 0 mask = 0x001f01ff -R -W -X -D -DEL_CHILD -CHANGE_PERMS -TAKE_OWN
  SID = /Everyone  S-1-1-0
    ACE 1 is an ACCESS_ALLOWED_ACE_TYPE
    ACE 1 size = 20
    ACE 1 flags = 0x00
    ACE 1 mask = 0x001301bf -R -W -X -D
  SID = BUILTIN/Administrators  S-1-5-32-544
    ACE 2 is an ACCESS_ALLOWED_ACE_TYPE
    ACE 2 size = 24
    ACE 2 flags = 0x00
    ACE 2 mask = 0x001f01ff -R -W -X -D -DEL_CHILD -CHANGE_PERMS -TAKE_OWN
  SID = BUILTIN/Server Operators  S-1-5-32-549
    ACE 3 is an ACCESS_ALLOWED_ACE_TYPE
    ACE 3 size = 24
    ACE 3 flags = 0x00
    ACE 3 mask = 0x001301bf -R -W -X -D
  SID = NT AUTHORITY/SYSTEM  S-1-5-18
    ACE 4 is an ACCESS_ALLOWED_ACE_TYPE
    ACE 4 size = 20
    ACE 4 flags = 0x00
    ACE 4 mask = 0x001f01ff -R -W -X -D -DEL_CHILD -CHANGE_PERMS -TAKE_OWN
SD's SACL is Not Present
Stream 1:
  Type: Security
  Stream name = (null) Size: 184

Stream 2:
  Type: Alternate Stream
  Stream name = jera.txt Size: 17

Stream 3:
  Type: Alternate Stream
  Stream name = maarten.txt Size: 20

Creation Time - 19/11/2000  15:39:58
Last Mod Time - 19/11/2000  15:40:35
Last Access Time - 19/11/2000  15:44:11
Main File Size - 0
File Attrib Mask - Arch Hid
Dump complete...
```

There are a few things I want to call attention to. These things are:

- Security Descriptors (SD)
- Security IDentifiers (SID)
- Discretional Access Control List (DACL)
- System Access Control List (SACL)
- Access Control Entries (ACE)
- Streams

All these things are listed in the dump files.

7.1 Security Descriptors (SD)

A security descriptor contains the security information of an object with Filestat the object is a file. The DS consists of an SD structure and it's information. An SD can contain the following information:

- An owner's Security IDentifier.
- A primary group's Security IDentifier
- Discretional Access Control List
- System Access Control List

If you look at the dump file you can see that the SD is 184 bytes long and that it contains:

- An owner's SID: SID = BUILTIN/Administrator S -1-5-32-544
- A primary group's SID: SID = NT4/Domain Users S -1-5-21-664811115-334825431-2087665911-513
- A DACL: SD's DACL is Present.
- NO SACL: SD's SACL is Not Present

7.2 Security IDentifiers (SID)

Windows NT assigns every user, group and computer account a SID. A SID is a unique value of variable length used to identify trustees. If you logon to a NT machine and your username and password are validated you get an access token based on you SID and the SID's of the groups you are member of. Every time you attempt to access an object such as a file the SID's in your access token are compared to the SID's in the ACL of that object. If one of the SID's in your access token is present in the ACL of the object you are granted a certain level of access. If you look in the dump file you see a few SID's within ACL's.

7.3 DACL and ACE's

A DACL is an Access Control List (ACL) that identifies trustees. With the use of Access Control Entries a t rustee can be granted or denied access to an object. If we look at the dump file we can see a example of an DACL.

```
SD's DACL is Present
SD's DACL-Defaulted flag is FALSE
   ACL has 5 ACE(s), 120 bytes used, 0 bytes free
   ACL revision is 2 == ACL_REVISION2
SID = BUILTIN/Administrators  S-1-5-32-544
  ACE 0 is an ACCESS_ALLOWED_ACE_TYPE
   ACE 0 size = 24
   ACE 0 flags = 0x00
   ACE 0 mask = 0x001f01ff-R -W -X -D -DEL_CHILD -CHANGE_PERMS -TAKE_OWN
SID = /Everyone  S-1-1-0
   ACE 1 is an ACCESS_ALLOWED_ACE_TYPE
   ACE 1 size = 20
   ACE 1 flags = 0x00
   ACE 1 mask = 0x001301bf-R -W -X -D
SID = BUILTIN/Administrators  S-1-5-32-544
   ACE 2 is an ACCESS_ALLOWED_ACE_TYPE
   ACE 2 size = 24
   ACE 2 flags = 0x00
   ACE 2 mask = 0x001f01ff-R -W -X -D -DEL_CHILD -CHANGE_PERMS -TAKE_OWN
SID = BUILTIN/Server Operators  S-1-5-32-549
   ACE 3 is an ACCESS_ALLOWED_ACE_TYPE
   ACE 3 size = 24
   ACE 3 flags = 0x00
   ACE 3 mask = 0x001301bf-R -W -X -D
SID = NT AUTHORITY/SYSTEM  S-1-5-18
   ACE 4 is an ACCESS_ALLOWED_ACE_TYPE
   ACE 4 size = 20
   ACE 4 flags = 0x00
   ACE 4 mask = 0x001f01ff-R -W -X -D -DEL_CHILD -CHANGE_PERMS -TAKE_OWN
```

In this DACL there are 5 Access Control Entries, have put a red box around it.

An ACE is an element in the ACL. An ACE consist always of the following element:

- A SID to which the ACE applies (see the blue box)
- An access Mask that specifies the Access rights (see the yellow box)
- A Flag that indicates the type of ACE (see the green b ox)
- A set of bit flags that determine whether child containers or objects can inherit the ACE. (see the gray box)

In a DACL there are two types of ACE's:

Access-denied ACE	This type is used to deny access to a SID
Access-allowed ACE	This type is used to allow access to a SID

7.4 System Access Control List (SACL)

A SACL enables administrators to log attempts to access a object. An ACE specifies the type of access attempts that must been logged. The type of access attempts are: Failed login, Success log on or both. What we are talking about here is AUDITING. The ACE is System -audit ACE. This says to generate an audit record when a person attempts to exercise the specified rights. This record is stored in the event log of NT for example.

7.5 Streams

Filestat also gives you information about streams. Filestat does not give you more information as Sfind.exe Only with Filestat you can also see the first stream the normal one. If you look at the size of that one, in my case 184 bytes you can see that it is the same size as the Security Descriptor and that the type of stream is called security.

8. Hunt.exe

Hunt is a very useful tool to get an idea of what information is given away way with NULL Session access. Before I go any further I will explain what "Null Sessions" are.

8.1 Null Sessions

A Null session is created when a client logs into a Windows NT host and both the username and password for this session are each Null characters. Null session users are considered to be members of the Everyone and Networ k groups, but not Authenticated Users. Security Access Tokens (SATs) for null sessions users always have a SID of S -1-5-7. (SANS network security 2000 – Track 5.1/5.2/5.3, securing windows NT, Step -by-Step, Parts 1-3, by Jason Fossen and Jeniffer Kolde)

Null sessions are mainly used for administrative purposes. It is also used by certain network services when they communicate across a network.

An attacker can obtain for example a list of usernames and groups from a remote domain controller with a null ses sion.

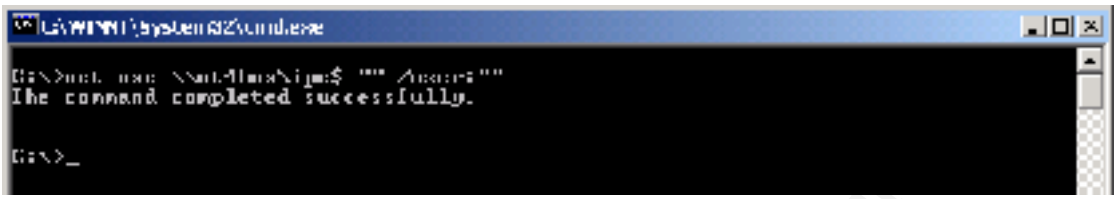
How this works I will show you now.

In my network I have an NT4 Server called NT4BOX and a Windows 2000 professional box. To demonstrate a null session I will use the command net view to obtain a list of resources being shared on a remote computer (NT4BOX).

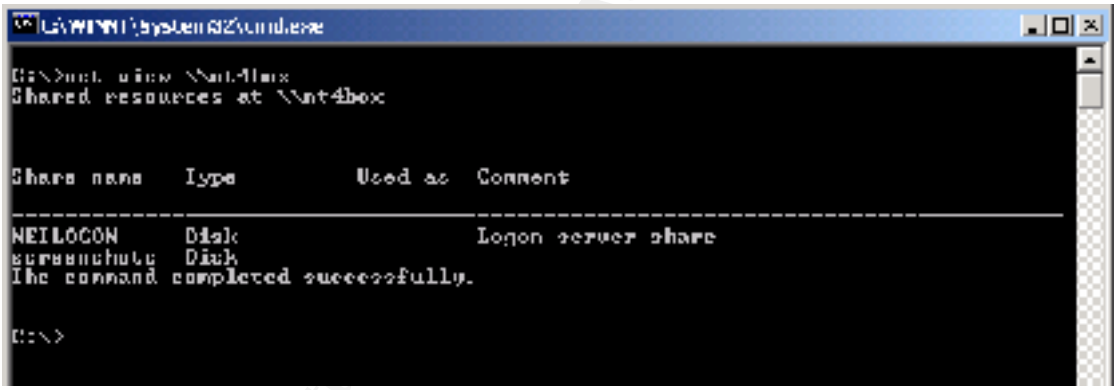
Lets see what happens.

As you can see access is denied and a system error occurred.

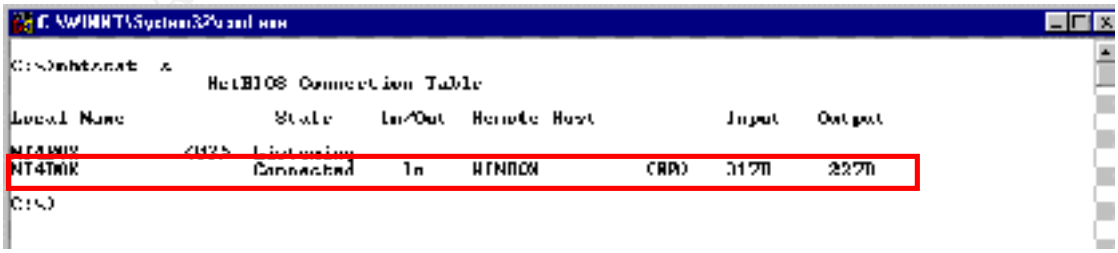
Now lets establish a Null user session. Here is how it works.



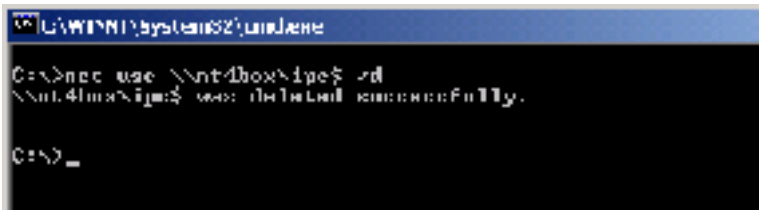
As you can see the connection is established successfully. I connect to the shared named pipe IPC$. This is a resource sharing the named pipes that are essential for communication between programs. This named pipe is for example used during remote administration of a computer and to view shared resources on a computer. I'm now allowed to use th e net view command. Watch this.



As you can see I have just obtained the shares of the NT4BOX. If we look at the NT4BOX we can see the Null Session.



Now we are going to delete the null session and see if it it really gone.



The command was successfully so if we now look at the NT4box, the Null session should be gone.

Now we know what a Null Session is I will explain what Hunt.exe does, but first let look at the syntax an options for Hunt.

Hunt \\servername

Where \\servername is the name of the server you want to examin. You can also use the /? Option to get information about the command.

Here is what you get when you enter the command.

As you can see I typed in the hunt command with the local computer name behind it. I get a list of shares hidden or not and I get a list of Users. The nice thing is that it picks out the BUILTIN/Administrator account. As you can see it is not of any influence that I renamed the BUILTIN/Administrator account to mvessen.

What hunt does is actually establishing a null user session and then looks what it can get. If you would use Hunt and looked at the NetBIOS connection Table

(nbtstat –s) of the computer you will see that a connection is established. I will not show this here, but it is easy to try.

9. Audited.exe

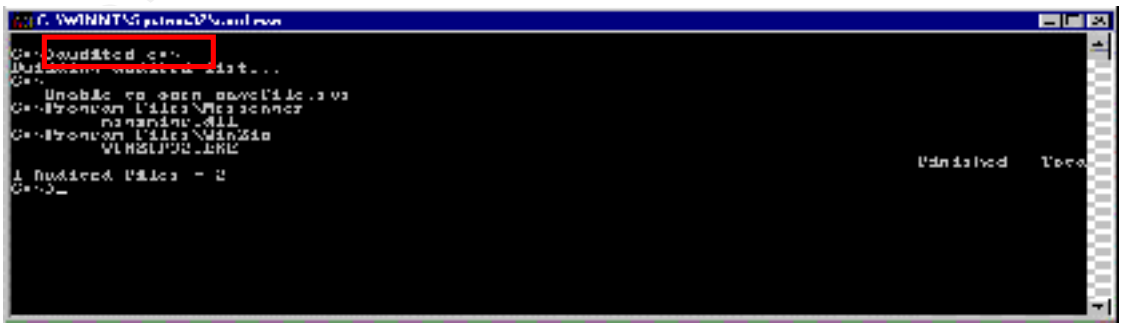
This tool reports all files in a specific directory that you are auditing and what actions are audited. Here are the options you can use and the syntax.

Command options	What it does
Audited [dirpath]	[dirpath] specifies the directory you want to look in and in its subdirectories.
Audited [dirpath] –ns	Looks in the specified directory but not in the subdirectories.
Audited [dirpath] –d	Also dump the file audit attributes
Audited [dirpath] –v	This enables the verbose mode

NOTE: *There are more option. The are for searching for audits in the registry. Unfortunately I could not get it working. If anyone has any information about this please let me now and maybe I can release a new edition of my paper that is mo re complete. (mailto: mvessen@go.com)*

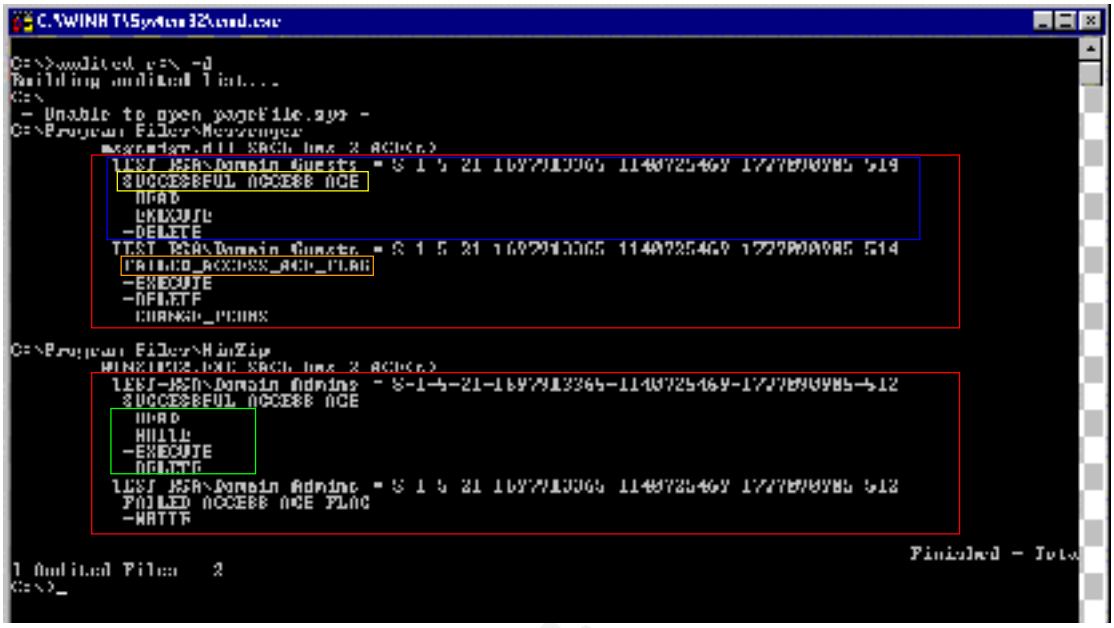
To try audited.exe you must have enabled auditing on Windows NT and you must set an audit for a few files. How to enable auditing on NT and how to set an audit for a file I refer to the practical assignment *"Centralized Auditing of a Windows NT computer" – By Steven Toy (www.sans.org/y2k/practical/Steven_Toy.doc) and "Section 3 Security Event Auditing" – By Martin Golias (http://www.sans.org/y2k/practical/Martin_Golias.doc)*

If you have set the audits we can start to u se audited.exe. First I will show the command audited c:\ This simple gives you all file names that have an audit on it.



As you can see I have an audit set on the files msgsmigr.dll and WINZIP32.EXE At the bottom you see total o f audited files. In my case 2.

Now lets look at the –d option. I use the command audited c: \ -d



Before we look at the screenshot I will explain a bit more about the SACL.

What SACL's are I already explained with the command F ilestat.exe. Let's take a look at the structure of a SACL.

If a SACL is present in a Security Descriptor of an object and it contains ACE's then the file is being audit. As I explained with the command Filestat.exe an ACE consists of:

- A SID to which the ACE applies
- An access Mask that specifies the Access rights
- A Flag that indicates the type of ACE
- A set of bit flags that determine whether child containers or objects can inherit the ACE.

A SACL (see red box) can contain one type of ACE's (See blue box) :

- SYSTEM_AUDIT_ACE_TYPE (see yellow grey)

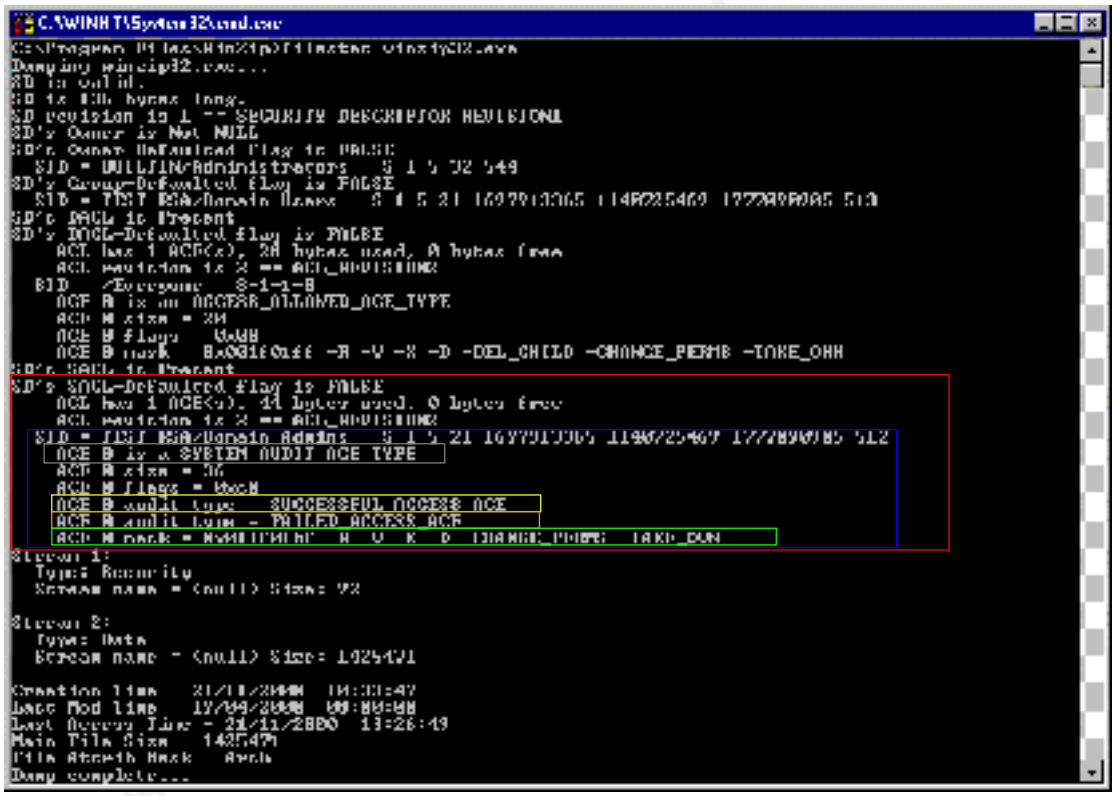
This type can contain one of the following two audit types:

- SUCCESSFUL_ACCESS_ACE_FLAG (see yellow box)
- FAILED_ACCESS_ACE_FLAG (see orange box)

The ACE mask in a SACL can contain the following attributes (see g reen box) :

- R, -W, -X, -D, -CHANGE_PERMS , - TAKE_OWN

To make it more easy to understand I will explain it with a screenshot of the Filestat command.



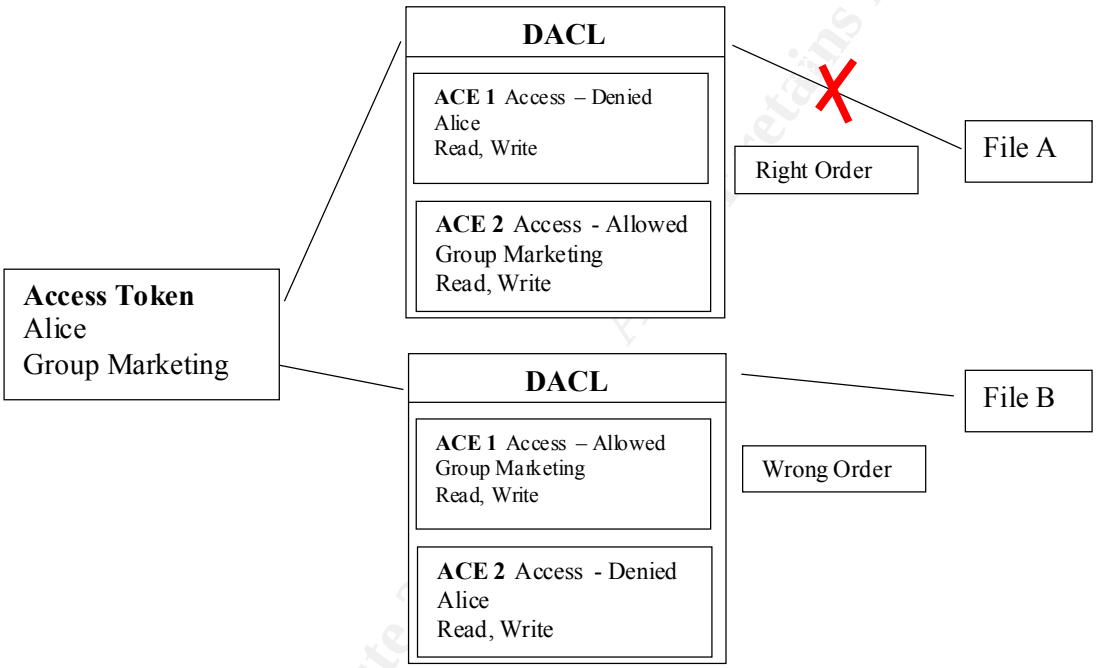
With audited –d you see almost the same information, but more simplified. I f you asked yourself why there were coloured boxes in the screenshot of the audited –d command, now you know, I used them to show where everything is with the audited command. I used the same colours as the above screenshot.

10. DACLchk.exe

NOTE: *I experienced a few problems using DACLchk. A lot of times it looped on certain files. Because of this problem I only explain what it does and how it should work.*

DACLchk reports any secured files that have denied ace's after allowed ace's. This is a security h ole that could allow unwanted access since denies should always be first.

To make it easier to understand I have made a drawing where you can see the effect of wrong ordered ACE's in an DACL.



As you can see, with the right ordered ACE's Alice has no access to a file, but with wrong ordered ACE's she can have access. I tried to force wrong ordered ACE's but could not get it done with Explorer. This may be a nice practice to do at home.

The syntax and options that can be used are:

Command options	**What it does**
Daclchk [dirpath]	[dirpath] specifies the directory you want to look in and in its subdirectories.
Daclchk [dirpath] –ns	Looks in the specified directory but not In the subdirectories.
Daclchk –d	Dumps all DACL's, does not detect reversed ACE's
Daclchk -?	Shows you the options and other information. The syntax it shows is not right. It says sfind, it should say DACLchk.

11. Conclusion

The Forensic Toolkit 2.0 can be a very useful tool. Try to use it in combination with other tools.

What is good about this Toolkit is that you can use it in script and automate it. I did not write anything about this possibility, because I think that that should be an paper on it's own. Maybe this is a good subject for an other GIAC course attendant.

I hope you learned something about the Forensic Toolkit 2.0 and about Windows NT security.

Maarten van Essen
Landis ICT Services & Consultancy

With special thanks to:

Jera Lysen, Harry and Truus van Essen, Landis ICT Services & Consultanc y and my colleagues

Thanks for giving me the time to write this paper.

12. References

Websites

- http://msdn.microsoft.com
- http://www.microsoft.com
- http://www.securiteam.com
- http://www.ntique.org
- http://www.foundstone.com
- http://www.sans.org
- http://www.whatis.com
- http://www.urec.cnrs.fr/securite/outils/nt/index.html
- http://www.winzip.com

Books

- SANS Network Security 2000 Track 5: Windows Security 5.1/5.2/5.3 Securing Windows NT, Step -by-Step, Parts 1-3 – By Jason Fossen and Jennifer Kolde
- Windows Nt Server 4.0 In The Enterprise Mcse Study Guide – By Alan R. Carter. (ISBN – 0764532480)

Last Updated: July 22nd, 2017

Upcoming Training

SANS Virginia Beach 2017	**Virginia Beach, VA**	**Aug 21, 2017 - Sep 01, 2017**	**Live Event**
SANS Network Security 2017	**Las Vegas, NV**	**Sep 10, 2017 - Sep 17, 2017**	**Live Event**
SANS vLive - SEC505: Securing Windows and PowerShell Automation	**SEC505 - 201709,**	**Sep 18, 2017 - Nov 13, 2017**	**vLive**
Secure DevOps Summit & Training	**Denver, CO**	**Oct 10, 2017 - Oct 17, 2017**	**Live Event**
SANS San Francisco Winter 2017	**San Francisco, CA**	**Nov 27, 2017 - Dec 02, 2017**	**Live Event**
San Francisco Winter 2017 - SEC505: Securing Windows and PowerShell Automation	**San Francisco, CA**	**Nov 27, 2017 - Dec 02, 2017**	**vLive**
SANS Cyber Defense Initiative 2017	**Washington, DC**	**Dec 12, 2017 - Dec 19, 2017**	**Live Event**
SANS Southern California- Anaheim 2018	**Anaheim, CA**	**Feb 12, 2018 - Feb 17, 2018**	**Live Event**
SANS OnDemand	**Online**	**Anytime**	**Self Paced**
SANS SelfStudy	**Books & MP3s Only**	**Anytime**	**Self Paced**