Windows Survey Methodology: Section 1 Transcript

Introduction

Welcome to the Windows Survey Methodology module. When we talk about Windows Survey Methodology, we also refer to this as tactical analysis or tactical forensic analysis. The ability to perform tactical forensic analysis on a host is a critical skill for a cyber-operator. Tactical forensic analysis is a means to quickly acquire situational awareness of a new system. From the offensive perspective, this may mean a system reached through remote access. Defensively, this may be a machine at risk of compromise from an attacker. In either case, quickly assessing the presence of other actors on the system to determine potential threats to tools and methodologies is important.

Tactical forensic analysis, performed to capture a snapshot of an operating system's, or OS's, current state, provides this type of information, which is then analyzed to draw conclusions about a machine. For example, an operator should be able to determine whether or not a system is compromised, the administrators of a system are technologically savvy, or if a system is heavily monitored.

In this module, we will discuss the methodology for performing tactical forensic analysis for a Windows OS on a network. While there are some differences, the methodology for conducting tactical forensic analysis on a host is essentially the same for Windows and UNIX systems. If you have already completed the UNIX module on tactical forensic analysis, much of this content should look very familiar to you.

Throughout this module, you'll be presented with opportunities to assess and apply what you've learned

At the end of this module, you will be able to:

- Apply Windows problem-solving techniques using available resource materials, including related commands and terminology,
- Identify processes and executables for potentially risky behavior using system information, including the process list, network connections, and the registry,
- Analyze suspect processes and system programs using system information and tools,
- Determine possible suspicious behavior, and
- Use timestamps to perform tactical forensics.

Bypass Exam Introduction

If you are already familiar with the subject matter presented in this module, you can choose to take a Bypass Exam to skip this module.

The Bypass Exam option provides a single opportunity to successfully demonstrate your competence with the material presented within the module. If you pass, you'll receive credit for completing the module, unlocking the content within, and you will be free to proceed to the next module. If you do not pass, you will need to successfully complete the module, including all exercises and the Module Exam, to receive credit.

Click the Next Section button to continue.

Windows Survey Methodology: Section 2 Transcript

Overview

When faced with performing cyber operations on an unfamiliar machine, establishing a basic level of situational awareness is important - regardless of mission. Every organization has its own tools, tactics, and procedures that it wants to protect from disclosure. Therefore, it is important to recognize who creates threats to each of these elements. In a nutshell, threats can be broken into three classes of actors: malicious actors, system administrators, and careless users.

Each of these actors represents a potential risk to operations. Our goal when tactically analyzing a machine is to identify measures these actors have taken that may put our operations at risk - whether intentional or not. The best tool we have to perform this task is anomaly analysis. What is anomaly analysis? That is a good question.

Anomaly Analysis

You may have noticed that previous training has primarily focused on *normal* system behavior. In other words, we have examined what we would expect to see in a *normal* system that is not at risk. When we perform tactical forensics on target machines, we collect data and analyze the results. As we analyze our results, we compare this data against our mental checklist of what a *normal* system looks like, identifying a list of potential anomalies.

We complete our tactical analysis by investigating each anomaly in more depth, providing our best assessment as to the reason the anomaly exists. It is possible, and even likely, that any particular anomaly is due to normal system operations; however, an anomaly may be indicative of potentially dangerous activities, such as a malicious actor who has:

- Altered user space to ensure continued access to the system,
- Installed user space malware that forwards data to a remote collector, or
- Modified kernel space with a **rootkit**.

Furthermore, an anomaly, may be indicative of potentially dangerous activities, such as:

- The diligent system administrator who has installed system monitoring tools (for example, auditing and logging tools, packet sniffers, keyloggers, and other related tools),
- Lazy system administrator who has installed shortcuts that allow users to elevate their privileges,
- Lazy system administrator or careless user who has left the machine open to potential compromise, or
- Careless user who is performing unsafe or unauthorized actions on a system.

Triage vs. Prosecution

There are two phases of tactical forensic analysis on a machine: triage and prosecution. Select each

tab to learn about both phases. When you have finished reviewing each phase, click Next Slide to continue with the module.

Triage: When conducting triage, a cyber-operator must find and identify:

- Potentially anomalous running processes
- Unknown, unexpected, or otherwise anomalous network connections
- Potentially anomalous programs on the disk
- Any other indicators of out-of-the-ordinary behavior or activity (on a Windows OS this generally means examining the registry keys)

Prosecution: Prosecution of a Windows OS, on the other hand, generally entails determining the specific actions, behaviors, and origins of the identified anomalous processes and connections, as well as a whole host of other details. In other words, prosecution is an in-depth examination to gather as much information as possible about each anomaly.

Strategy: Trusted Tools

Prior to accessing a machine to perform tactical forensic analysis, developing an effective strategy is very important. With this in mind, there are several items that you should consider; one of the most important is the use of trusted tools. If a target system has been compromised, it is possible that the tools on the system that would normally be used for tactical forensic analysis will give false information. With this in mind, you should always upload your own set of trusted tools.

In order to use trusted tools effectively, it is important that you always enter the full path to your trusted tools. For example, `C:\Windows\Temp\psloglist.exe`. Failure to enter the full path to your trusted tools each time, may result in you running potentially compromised tools on the target system.

Additionally, your trusted tools should ideally be statically-linked binaries, which eliminates the risk of your tools using potentially compromised system libraries. However, keep in mind that bringing your own binaries is not foolproof. If the target system has a rootkit, malware that is installed into the kernel itself, it is possible that the system calls in the kernel of the target system are compromised. If system calls are compromised, no tool can be trusted since the kernel may provide false data. Therefore, should you determine this is your situation, you should exit the machine and conduct off-line forensic analysis, provided you support a defensive mission. If you support an offensive mission, then you will need to seek the appropriate path of inquiry stipulated by your organization.

Strategy: Minimization of Footprint

As part of maintaining a level of stealth, a priority when conducting cyber operations is minimizing your footprint on the remote machine. Therefore, when performing tactical forensic analysis, it is important to be cognizant of the number of commands being run on a host. If running one command can obtain the same information as running five or more separate commands, then you should always run just the one command. Why? Simply put, fewer commands means fewer packets on the wire, which in turn means lower bandwidth. Fewer commands also means less chance of detection in the event that your session is being logged by the host.

Therefore, in an effort to reduce your potential footprint on a system and potential discovery, an excellent strategy is to ensure you only run the commands that will yield the data you desire. One way to accomplish this is to type the commands out in a text file. This will allow you to see everything in one place, provide the opportunity to de-conflict commands that will return duplicate data, and ensure that all commands and options are entered correctly. Not to mention, pre-listing commands will reduce the number of commands run by eliminating typos, forgotten switches, and more. Then, when performing tactical forensic analysis, all you have to do is copy and paste the commands from your text file to your command shell on the remote machine.

This being said, please note that it is not a good idea to script all of your tactical analysis commands to run sequentially without user input. Depending on the configuration of the remote machine, some commands may cause unintended and undesirable results - as a cyber-operator you will need to be prepared and ready to react.

Strategy: Time Management

Another matter to consider, as you prepare to perform tactical forensic analysis on a machine, is time management. When striving to manage your time, an important thing to remember is that some of the data you may wish to collect can take a while to gather, especially if you need to execute searches of the filesystem or other commands which have to traverse a significant portion of the disk. One strategy to effectively make use of your time is to run the most labor intensive commands early in the tactical forensic analysis process. Running these types of commands early will result in the data being available when you are ready to review it. This may seem fairly basic, but efficiency is often one of the most critical components of the tactical forensic analysis process.

Knowledge Check Introduction

It is time for a Knowledge Check. This Knowledge Check will not be scored, but may indicate areas that you need to review prior to the Module Exam.

Windows Survey Methodology: Section 3 Transcript

Triage

The first part of the tactical analysis process is performing triage on a machine. Based on the information they provide about the remote system, the commands run to conduct a triage can be organized into three main groups: System Characterization, Processes, and Network Connections.

Let's begin our discussion of these three groups by examining System Characterization commands and what they offer during the analysis process.

System Characterization Commands

When accessing a remote machine, it is wise to first run System Characterization commands as they will confirm that you are indeed viewing the correct host. They will also provide you with basic information about the environment local to that machine. Take a few moments to review the chart to learn how each of these commands is helpful.

Process Commands

Once you have successfully accessed a remote machine, triage of the machine should almost always begin with a search for anomalous behavior in the process list. On a Windows machine you might try the `/v` and `/svc` options for the tasklist command. The `-x` and `-t` options are also particularly useful when using the Sysinternals tool `pslist`.

Note, there is no comprehensive checklist of what makes a process anomalous; recognizing one when you see it is a skill that you will develop with experience. To identify abnormal processes, you can search for some of the indicators displayed. Click each tab to learn some helpful details for each of these indicators. When you have finished reviewing each indicator, click the Next Slide button to continue with the module.

Start from unexpected locations: Some processes may start from unexpected locations. For example, `explorer.exe` running out of `C:\windows\system32…` (or just about any executable running from `%WINDIR%\temp`)

Start sooner than expected: Generally, there are certain processes, including system services such as `svchost.exe` and `smss.exe`, as well as `winlogon.exe`, that you expect to see when the system is first coming up. Other processes starting around the same time as these could be questionable. You can determine this by reviewing timestamps and PIDs.

Possess unusual options or arguments: Viewing the process list using the syntax `wmic process list` (this may be easier to read if you redirect the output to a text file) enables you to see the Parent Process ID to determine what process spawned the process in question. Examine these processes to look for anomalies, such as a shell process (for example, `svchost.exe` or lsass.exe) descending from a service process.

Possess unexpected ancestors/PPIDs: Finding PPIDs is not as straightforward in Windows as it is in UNIX systems, so some creative command crafting using `wmic` or `pslist` may be necessary to view process lineage.

Run by unexpected users: In some cases you may observe a process being run by an unexpected user. For example, `explorer.exe` or `cmd.exe` seldom run as SYSTEM - if seen doing so, they should be investigated further.

Exhibit strange or intentionally malformed names: Malware will often use randomly generated executable names, but there are also names intentionally chosen to look like and be mistaken for legitimate processes. For example, `explorer.exe` is a legitimate process; however, `explorerer.exe` is not a valid process.

Other known examples of intentional malformed names include:

- `WINW0RD.EXE` (with a zero [0] instead of the letter [O])
- `iexplorer.exe` (`iexplore.exe` is Internet Explorer and `explorer.exe` is the Windows environment)
- `svchst.exe` (the legitimate services process is `svchost.exe`)
- `csrss .exe` (the space between the executable name and the period [`.`] can easily be overlooked)

While it may seem obvious examining a process in this context, keep in mind that it may not be so obvious when you are examining a process list. Entries such as these can be overlooked very easily - especially if there is a large number of services running on the host.

Process List: An Additional Note

Attention to detail is imperative when examining a process list. With practice and experience, you will learn how to recognize questionable and potentially anomalous processes more quickly - but it is always better to be slow and deliberate when reviewing this list.

An anomalous process will not have a name that makes it immediately identifiable (for example, the ILOVEYOU virus wasn't called `ILOVEYOUVIRUS.exe`). Generally, you are not going to be able to determine exactly what your questionable processes are without further investigation. Unless a single process is glaringly bad, we recommend simply creating a list of candidates to inspect more thoroughly. You will be collecting additional information that may prove or disprove their legitimacy, or give you more insight into the processes' behaviors and actions.

Network Connections

The next step in conducting a triage is to examine network connections. While the process list should almost always be the first place to check for the presence of other actors, network connections are often just as important. As previously mentioned, malware processes are frequently named in a manner that often makes them appear legitimate at first glance. Examination of the socket table is one of the best ways to identify anomalous or unusual behavior. Reviewing the socket table will also help you double check processes on your list, as well as identify effectively disguised processes previously missed.

Network connections cannot be hidden easily without the use of a sophisticated rootkit. It is for this reason that the socket table is considered a reliable source of information regarding the activity of processes on a machine. Much like the process list, most network activity on a target host will be normal traffic and totally benign. To successfully develop the knowledge and skills to differentiate between normal and anomalous network traffic, you must become familiar with commonplace processes and services, as well as the ports they communicate over. You must also learn which processes on the target are likely to make connections out to other services. For example, while a shell can make a connection, it is definitely unusual and would warrant further investigation. Experience is generally the best teacher.

Network Commands

Once you have become familiar with routine processes and services, as well as their ports and method of connection to other services, you generally view socket table information with the `netstat` command. This command has a number of options that you can use to view forensically relevant network information and to tie the related open sockets to the processes on the system.

On a Windows machine, the syntax that gives the best info in one shot is `netstat -anob` where the options used are as follows:

- `-a`: Displays all active TCP connections as well as the TCP and UDP ports on which the computer is listening
- `-n`: Displays numerical addresses instead of trying to determine host, port, or user names
- `-o`: Displays active TCP connections and includes the Process ID, or PID, for each connection; you can find the application based on the PID on the Processes tab in Windows Task Manager
- `-b`: Displays which process is involved in the connection; however, this only returns the executable name and not the full path of the executable

You can also view just TCP connections by using `-p TCP`. UDP connections are usually much less of a concern; however, to view them, you can use `-p UDP`. (Note: It is acceptable to leave out the `-p` option.) The Sysinternals Suite contains a command line tool called `tcpvcon` that can be used to return information about the open sockets as well; however, it does not necessarily offer any additional benefit over the built-in Windows tools.

Once you have all of the socket data and associated processes, review the data for any suspicious or unusual behavior.

Suspicious/Unusual Behavior: Host Checks

Some potentially suspicious or unusual behavior to search for on the host includes:

- **Unusual or unexpected ports open to any address (with a local address of * or 0.0.0.0):** Usually, ports listening on localhost only (127.0.0.1) are safe and can be ignored
- **Processes listening on unusual ports:** For example, a webserver probably should not be listening on port 25; and, a telnet service generally should not be listening on a random high

port. (It is important to note that these are configuration specific though. Upon further investigation, it might turn out to be completely legitimate. The fact that these examples are non-standard and unexpected, however, is enough to warrant that further investigation.)
- **Established connections with unusual remote ports:** This can be anything connected to well-known malicious ports, such as 31337 or 6666, or processes/services on the machine connecting out to unusual ports. For example, you probably would not expect to see a web browser process with an established connection that does not include TCP ports 80, 8080, or 443.

As you might imagine, this is just the tip of the iceberg.

Other Host Checks

Other potentially suspicious or unusual behavior to look for on the host includes:

- **Unusual users associated with processes and services that have open sockets:** While it is perfectly normal to observe a user account running processes like `firefox.exe` and `outlook.exe` with open sockets, it is highly unusual to see these same processes being run under SYSTEM. (Note: Many Windows system processes can run and open network sockets as SYSTEM, LOCAL SERVICE, and/or NETWORK SERVICE, so do not be alarmed when you see this type of activity.)
- **Unusual or unexpected parentage of a process with open sockets:** We expect processes that a user would create to spawn from the user's instance of `explorer.exe`. Likewise, system processes should mostly be spawned from `wininit`. If you see either user or system processes that are descended from unexpected parents, chances are the processes are not legitimate. (Note: You may be able to determine this by looking at the process list and process tree in the previous step, but examining open sockets gives you another opportunity to differentiate between valid and invalid processes.)
- **Unusual processes and services making network connections:** While it may be totally normal to see open sockets associated with `chrome.exe` or `services.exe`, seeing them associated with `sol.exe` or `notepad.exe` should raise a red flag of concern and prompt further investigation.
- **The geography of remote addresses with which the host has established connections:** Connections to specific locations are not automatically bad; however, if a connection clearly stands out from the others, this connection may give you some insight as to the behavior of the associated process. For example, if all established connections on your local machine are to US IP addresses except one to an IP address in a country known to engage in adversarial cyber activity, further investigation of the process associated with that connection is advisable.

Suspicious/Unusual Behavior: Disk Checks

Along with searching for anomalous processes on the host, when conducting triage on a remote machine, it is important to search for suspicious or unknown programs on the disk. This being said, searching the entire disk for executables is not realistic. A more measured approach is needed. When analyzing the process list, you may have identified processes running from nonstandard or suspicious locations on the disk. This can give you a starting place in your search. A follow-up question might be, "Are there any other unknown or potentially anomalous programs in the same

directory?"

Generally, it is also worth searching for unknown or potentially anomalous programs in the "temp" directory (`C:\WINDOWS\Temp`), the root of the drive (`C:\`), and common user directories (`[Windows 7] C:\Users\[username] or C:\Users\[username]\Desktop`). Although you are less likely to find unknown executables that are not currently running sitting in these locations, it is advisable to develop the habit of checking. More often than not, to determine where on the disk you should be searching for anomalous executables, you will need to rely on indicators such as the locations of other running processes.

Suspicious/Unusual Behavior: Final Check

The final major step in triaging a machine consists of checking other locations where you might find information about potentially anomalous processes and behaviors on the host. On a Windows machine, this involves reviewing startup processes, services, scheduled tasks, and specific registry keys. Click each tab to learn how to check each of the locations displayed. When you have finished reviewing each location, click the Next Slide button to continue with the module.

Startup Process: To view processes run at startup, you can use the `wmic` command. The most commonly used syntax for the `wmic` command for this purpose is: `wmic startup get caption,command`

Search processes displayed for any unusual entries. You may wish to compare sizes against known good examples for the operating system.

Services: To interact with the Service Control Manager, which will allow you to see the configured services, their status, and when they are run, use the options associated with the `sc` command. If you are unfamiliar with command usage and options, refer to the materials in the Resources tab.

Scheduled Tasks: To investigate scheduled tasks, use the `schtasks` command. The `at` command (now deprecated) can also be used - but this function is typically locked down on systems running Windows Vista and later versions. When searching scheduled tasks, you should keep in mind that it is not uncommon to see a large number of scheduled tasks. Custom (non-Windows system) tasks are usually pretty obvious and easy to pick out.

Registry Keys: When searching the registry, there are several keys with which you should be familiar. First are the Run and RunOnce keys displayed:

- `HKEY_LOCAL_MACHINE\Software\Microsoft\Windows\CurrentVersion\Run`
- `HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\Run`
- `HKEY_LOCAL_MACHINE\Software\Microsoft\Windows\CurrentVersion\RunOnce`
- `HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\RunOnce`

The important distinction between these keys is that any executable listed in the RunOnce key (as the name suggests) will only be run once and then it will be removed from the registry. The Run key, on the other hand, is persistent and will be run each time the system boots up, or the user logs in.

It is also worth noting that if the program under RunOnce does not run successfully, the RunOnce

entry is not removed. It is removed only on successful completion. Additionally, you are more likely to see entries in the HKEY_LOCAL_MACHINE directory (or HKLM hive) as opposed to the HKEY_CURRENT_USER directory (or HKCU hive) - entries in HKLM affect all users on the machine, and will be run at system startup as opposed to waiting for the user to login.

Another registry key you should investigate is:
`HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Winlogon\Shell`

Typically, the value of this key should be `explorer.exe`, however, it can be changed to a different executable or it can be changed to call another program first and then launch `explorer` as normal. All of these can give you valuable insight into the potential presence and actions of other actors on the host. They can also give you a starting point for locations on the disk where you should look for other anomalous executables.

Windows Process List

Welcome to the Investigating a Windows Process List video. I will be your guide for this video presentation.

Analyzing a process list during a survey is a critical part of developing a target's situational awareness.

In triaging a process list, our goal is to identify anomalous, suspicious, or potentially malicious processes. Our background knowledge of the OS and contextual understanding of the target should help us notice things that are out of the ordinary or unexpected.

In this presentation, we will analyze a sample process list and discuss some of the most common indicators that a process is of concern and warrants further investigation.

Ultimately our goal is to help us answer the questions, "How likely is it that my presence will be noticed?" and "Is it safe to operate?" Let's get started!

Here we have the output of some commands from a Windows 7 workstation. If we didn't know that ahead of time we could look at the tasklist and netstat commands to help determine this. Looking through the running processes, we don't see any server type processes, and looking at the netstat, we don't see any server related listening ports. Understanding the type and purpose of a host can help us determine which processes we would expect to see.

Taking a look at the process list, we see many familiar processes that are typical of a Windows 7 system.

At first glance the "iexplorer.exe" process almost blends in, but the Internet Explorer process is actually "iexplore.exe" as we see a few lines below. A quick Internet search can help you determine which processes are normal and which ones are not.

To further investigate, let's take a look at the full path to the executable using WMIC.

We can see in the WMIC output that iexplorer.exe is running out of Windows\Temp. That's very

suspicious.

Let's dig a little deeper and see if we can find the method of persistence in the registry.

As we've discussed with Windows Survey, the Run keys are common locations for persistence, and we see here that the executable in question is using Run key under HKLM for persistence.

We can see from the registry output that it appears to have a listening port. We will always check the netstat anyway, but this is a good indication of what we'll find there.

Now that we're looking at the netstat output, we can see our process in question is in fact listening on port 4444. So this process is definitely abnormal and is most likely malware.

While we're looking at the netstat, we also notice wordpad.exe is listening on port 25. Wordpad is a common windows application and 25 is a common port, but the combination here is a dead giveaway of something suspicious. Wordpad should definitely not be listening for SMTP traffic.

We'll continue investigating this Wordpad process by looking back at the output of the WMIC command from earlier.

We can see that this process is running out of C:\Windows\system32. At first glance, this doesn't appear suspicious since notepad.exe is running out of the same location. A quick Internet search will inform us that notepad.exe does indeed live in System32, but Wordpad does not. Wordpad actually lives in the Program Files directory.

If we look back at the output from the Tasklist command, we can see wordpad.exe actually blends in fairly well. The process name is the same as the legitimate Wordpad program. As we can see, this analysis has clearly indicated the importance of being thorough when surveying a host.

This concludes our video demonstrating how to investigate a Windows process list. We will have an opportunity to gain more experience with analyzing a process list in the upcoming exercise.

Windows 10 Process List

Prior to Windows 10, `svchost.exe` files were used to run various system services in order to reduce memory consumption. Services were often grouped which meant that each instance of `svchost.exe` could contain one or more of the services run by the operating system.

Starting with Windows 10 version 1703 (PC build 15063), PCs with sufficient memory (3.5 GB of RAM or more) do not group services. For this reason, newer versions of Windows 10 will have an increased number of `svchost.exe` instances running on a system.

However, if a Windows 10 Operating Systems has inadequate memory, (less than 3.5 GB), the classic service model will be used, resulting in services being grouped like in previous versions of Windows.

It should be noted that when Windows 10 uses service grouping, there will still be more `svchost.exe` instances than in previous versions of Windows, but not as many as when service grouping is not being used.

Windows Survey Methodology: Section 4 Transcript

Process Prosecution

Once you have completed basic analysis of the operating system and created a list of unknown or questionable processes, you can begin to investigate, or prosecute, the processes on your list, including any suspicious or unknown process you may have identified when looking at the network activity. This encompasses many skills you have already learned: viewing the directory from which a process is running, the permissions with which the process is run, the handles a process has open, and any libraries it uses. You can also use timestamps to find other files and directories that were affected at or around the same time as the process executable in question. While we have not covered this last concept yet, timestamps will be discussed in more detail later in this module.

There are, however, challenges when it comes to process prosecution. Let's examine some of these challenges.

Challenges

One challenge is that generally you will not find a standalone piece of evidence that confirms with 100% certainty whether a process is valid or invalid. It is much more common that you will have to piece together numerous bits of information to make a complete (or nearly complete) picture of the true nature and behaviors of an unknown or anomalous process. However, an example of evidence that could be considered conclusive is the hash of an executable that is identified as malware on a trusted source, such as VirusTotal. Even if you should find conclusive evidence of this nature, you will be responsible for fully prosecuting all suspicious or anomalous processes.

Another particular challenge of process prosecution is that it is not an exact science. There is no specific order in which commands must be run, and sometimes the commands you run are dependent on the output of other commands. There is also no specific order for investigating suspicious processes that you noted during the triage process. Process prosecution is about gathering as much information as you can and examining all the data in context together.

Quick Checks

Sometimes process prosecution is as simple as using your Internet search engine of choice to look up the name of an unknown process. Many times unknown processes are truly legitimate and simply have strange looking names, or are version-specific variations of processes with which you are otherwise familiar. A quick search, combined with a brief analysis to correlate your findings, can give you confidence that a previously unknown process is normal and safe, unless you see some other concerning behavior. You can also store this in your memory banks, so that the process will not be unfamiliar next time.

One thing to watch out for, however, is that just about any executable name can be flagged as malware. When using the Internet to conduct searches on executables, be thorough and wary. Search for other indicators such as installation directories, related files, and even file hashes, if you

can find them. Also, it should be noted that "because the Internet said so" is not a valid defense of the legitimacy of a process. The Internet is just another useful tool to provide indicators and other data to aid in the prosecution process.

In addition to using the Internet, another easy way to do some quick checking on an unknown process is to use the `wmic` command, specifically, `wmic datafile where name='c:\\windows\\system32\\calc.exe'`, to view file properties. This is similar to right-clicking on a file in the Graphic User Interface (GUI) and selecting properties. While it is true that this information can be spoofed fairly easily, most malicious programs do not take this extra step of obfuscation. The `SignTool.exe`, which verifies Digital Signatures in Windows, will tell you if the executable is verified and signed by a Windows trusted source. Again, this information is far from foolproof, but it is one useful piece of information to incorporate in your overall evaluation.

After Quick Checks

After checking the Internet and properties, a good place to start process prosecution on a remote machine is to view the directory in which the process is running. By itself this may not provide a definitive answer as to the legitimacy of the process, but it can give you a good starting point. For example, if an Internet search indicates that an executable should be found running out of a specific directory and it is actually running out of that directory, then that would be a good first indicator of the process' authenticity. Conversely, if the executable is running out of an unexpected or suspicious directory, then this would be a good first indicator that you may be on the track of an anomalous process.

Keep in mind that there can be a great deal of variability between different systems, even of the same Windows vendor and distribution, depending on the system administrator's preferences. However, as a general rule, the most common locations for executables on a Windows system include the following (in order of likelihood):

- `c:\windows\system32`
- `c:\program files`, or
- `c:\windows`

Conversely, a good first indicator that you may be on the track of an anomalous process is when a process is running from locations, such as:

- `c:\windows\temp`, or
- `c:\windows\user`

Does this sound similar to triaging? If it does, that is because many of the steps to prosecute a process are the same ones you might use to identify an anomalous or suspicious process or program. For example, another indicator of a process' behavior can be the effective permissions with which the process is executed - that is, the user account which is associated with the execution of the process.

Most likely you will have already examined this detail when identifying anomalous processes. However, re-examining this detail can help you during process prosecution as well. This is because a system process running with `system` privileges may not initially seem suspicious, but if it is running out of the `temp` directory, then there might be reason to be concerned. As discussed,

prosecuting a process is all about putting incomplete bits of information together to form a more complete picture.

Prosecuting Specific Behaviors

Again, while there is not an exact script or progression to follow when attempting to prosecute a process, you can use what you have learned so far to meet some high level objectives. Select each objective item to learn about specific behaviors that should be investigated when prosecuting a process. When you have finished reviewing items that should be prosecuted for each high level objective, click Next Slide to continue with the module.

Identify the characteristics of the executable: In an effort to identify the characteristics of the executable, search for answers to the following questions:

a. Where does it live on disk?
b. What is it doing? For example, what handles does it have open? What DLLs does it have loaded?
c. Are there any human-readable strings in the binary?
d. What can you find out about it via open source research?

Determine the network capabilities of the processes: When investigating the network capabilities of the process, determine the answers to the following questions:

a. Is it initiating outbound connections?
b. Is it listening locally?
c. Has it established connections? If so, what are they doing? For example, are they ex-filtrating data, receiving instructions, or something else?

Identify the files associated with the process: It is important to identify files associated with the process, so you must determine:

a. What else was created or changed around the same time as the executable? For example, were files, directories, or registry keys created or modified?
b. Is it creating or outputting files or logs on disk? What about configuration- or C2-related files?

If the process is running automatically, determine its persistence method: Oftentimes persistence is a crucial factor; therefore, when investigating the process determine the answers to the following questions:

a. Is it a service?
b. Is it a scheduled task?
c. Are there one or more registry keys?

Perspectives of High Level Objectives

It is worth noting that many of the objectives and questions discussed are from the offensive perspective - geared toward discovering malicious processes implanted by a third party. However, these same principles apply to prosecuting a custom monitoring program compiled by a particularly attentive administrator, or a script that a lazy system administrator placed on a machine to

circumvent corporate policy and make his job easier.

None of these concepts should be unfamiliar to you, but learning to pull it all together to assess an unknown process can be a daunting task, and certainly takes practice. One good way to practice is to spend some time prosecuting processes that you know are legitimate. Since you can already answer the questions for a known process, see if you can identify all of the process' attributes through command line investigation techniques. This will help you be confident in your results when prosecuting an unknown process.

Exercise Introduction

It is time for an Exercise. In order to successfully complete the Exercise, you are expected to use your notes and any available resources presented throughout the course, in addition to conducting your own Internet research.

Windows Survey Methodology: Section 5 Transcript

Windows File Time

On a Windows machine, a **file time**, sometimes referred to as MAC time, is a 64-bit value that represents the number of 100-nanosecond intervals that have elapsed since 12:00 A.M. January 1, 1601 Coordinated Universal Time (UTC). The system records file times when applications create, access, and write to files.

The New Technology File System (NTFS) stores time values in UTC format, so they are not affected by changes in time zone or daylight saving time. The File Allocation Table (FAT) file system stores time values based on local time of the computer. For example, a file that is saved at 3:00pm PST in Washington is seen as 6:00pm EST in New York on the NTFS volume, but it is seen as 3:00pm EST in New York on a FAT volume.

Timestamps

NTFS keeps track of lots of timestamps. Timestamps provide a tool that can be helpful when performing forensic analysis of a target machine. Once an event of interest has been identified, timestamps can be used to determine what file system activity took place both before and after the event. Each Windows file has a timestamp for: create time, modification time, access time, and entry modify time. Select each tab to learn about the different types of timestamps. When you have finished reviewing each type of timestamp, select Forward to continue with the module.

Create Time: The create timestamp is updated anytime a file or directory is created from scratch or a copy is made.

Modify Time: The modification timestamp is updated anytime a file or directory is changed.

Access Time: The access timestamp is updated anytime the contents (including metadata) of a file or directory is touched to perform an action.

Entry Modify Time: The entry modified timestamp refers to the time when the Master File Table (MFT) entry itself was modified.

It is important to note that not all file systems can record creation and last access times, and not all file systems record them in the same manner. Resolution of create time on FAT is ten (10) milliseconds, while write time has a resolution of two (2) seconds and access time has a resolution of one (1) day, which means it is really the access date. The NTFS file system delays updates to the last access time for a file by up to one (1) hour after the last access, while Modify Time is two (2) minutes.

Viewing Timestamps

Windows offers tools to view the timestamps associated with directories and files. For example, the

`dir` command, shows the timestamp for directories and files - in this case, the timestamp `12/05/2014 06:37` for `test.txt` file. We can also get the same result using the `/t:w` option, which enables you to select which timestamp to display. In this case, "w" or the last written time, which is the default timestamp to display with the `dir` command.

Using the option `/t:c` will display the creation time (`12/05/2014 06:34`). Additionally, the option `/t:a` will display the last access time (`12/05/2014 06:34`).

It is important to note that in Vista+, the last access timestamp is not enabled by default. The timestamp exists, but it will not be updated. It is also important to note that FAT32 and VFAT volumes store only the date (not the time) of the last access. On these drives the time of last access will always be 00:00. The registry key for this is:

```
HKLM\System\CurrentControlSet\Control\File
system\NtfsDisableLastAccessUpdate = 1
```

While this registry key has to be added to XP, it defaults to 0 (false), meaning access times are still updated. Starting with Vista+, the setting defaults to 1 (true), meaning access times are NOT updated. Access time updates can be re-enabled by setting this key back to 0 (false), but at the risk of bottle-necking the disk I/O queue.

Along with being viewable via the command line, Windows timestamps can be viewed using Windows Explorer. The system default only displays the date modified, but you can add the other timestamp columns, if you desire.

Analyzing Timestamps

Timestamps can be helpful when performing tactical forensics. For example, you can use the timestamp of a suspicious or anomalous executable to find other things on a disk that were created, added, or modified around the same time and thus may be related, which may provide further insight or information. More specifically, timestamps can be useful when correlated with log files, since they can provide an indication of which actions took place either just before or just after an auditable event.

That's it for timestamps . . . and the fundamentals of tactical forensic analysis. Are you ready to test all of your new knowledge? Great!

Video

Welcome to the Windows Survey video. I will be your guide for this video presentation.

Prior to interacting with your target, it is a good idea to create a list of commands you plan to run. This list of commands acts as a survey to help develop target situational awareness. In order to help you keep track of any unwanted attention you may have brought to yourself, it is important to record all commands with timestamps to easily identify any logging later on. This includes any pre-exploitation or vulnerability scans.

In this presentation, we will conduct an initial survey using basic commands, determine if you should continue your survey by verifying that your actions did not create any logs, and then ultimately

assess your target situational awareness.

As you run through your survey commands you should always ask yourself, "Am I caught?" and "Is it safe to operate?"

Let's get started!

Choosing ipconfig as your first command is useful in determining if this host is the host you expect it to be, generally through the hostname, or sometimes the IP if you know they statically configure them. To begin our survey, we will examine the output from the `wmic process get executablepath, processid, parentprocessid|more` command to begin enumerating processes. This allows us to look for anything unusual running on the system. Note that this can return a lengthy output and be difficult to read in the terminal, so piping it to "more" can help manage the output.

This `wmic` command shows where each process is running from, as well as any arguments. Note that this can also return a lengthy output. Again, the goal here is to look for anything unusual running.

Next, we will run the netstat with a findstring search for "EST." Note that in this case, the command produces no results since there are currently no established connections on this machine. Changing the search string to "LIST" will show all listening ports. The goal here, just as with the process lists, is to look for anything unusual that might determine it's not safe to continue operating here.

Now that we've determined it's safe to continue, we can really start surveying the system. Date with the "t" option shows us the date of the system. Time with the "t" option shows us the time of the system.

The SysInternals command, `psloggedon`, is useful in that it displays any users logged into the box, and any users accessing network shares. The `auditpol` command displays the current audit policies on the box which is useful in determining which events get logged. The SysInternals command, `Pslist`, also displays the running processes, but provides other useful information, such as the idle process having only 1 thread tells us this box only has one processor core. Additionally this is helpful in determining Operating System. For instance, a System PID of 4 means XP or above, Wininit and LSM in the process list equals Vista +.

A registry query of this key shows you the service pack number. The CSD version is service pack one. `nbtstat -n` displays the names that have been registered locally on the system by NetBIOS applications such as the server and redirector.

The routing table could potentially be useful for network mapping purposes. `net use` displays any resources that this host is currently connected to. `net share` displays any resources this host is currently sharing out. `net start` displays all the services currently running.

This `dir` command displays the contents of system32, the creation date of each listing, and then orders the results by that date. This is useful in determining legitimacy of command line tools. The SysInternals command, `autorunsc`, displays programs configured to autostart. The -b option displays boot execute which specifies the applications, services, and commands executed during startup.

Another common location of persistence is the HKLM Run key. HKCU also has a Run key. RunOnce keys can contain pending malware installations. While not truly persistence, it could be a location that installs the malware into a persistent location. HKCU has its own RunOnce key as well. The winlogon key, specifically the shell value is yet another location of persistence.

`net users` displays all the users that have been created on this system. This registry query displays the SIDS of users that are currently logged on to this computer, which is a useful distinction between an account simply existing or having been used.

Keep in mind that there are many different commands that are useful in various situations and environments, but this is just an overview to familiarize you with the general process of a survey and some of the things you'll want to look for.

Exercise Introduction

It is time for an Exercise. In order to successfully complete the Exercise, you are expected to use your notes and any available resources presented throughout the course, in addition to conducting your own Internet research.

Windows Survey Methodology: Section 6 Transcript

Summary

You have completed the Windows Survey Methodology module. You should now be able to:

- Apply Windows problem solving techniques using available resource materials,
- Identify processes and executables for potentially risky behavior using system information,
- Analyze suspect processes and system programs using system information and tools,
- Determine possible suspicious behavior associated with suspect programs and system configurations, and
- Use timestamps to perform tactical forensics.

To receive credit and advance to the next module, you must achieve a passing score on the Module Exam.

Click the Next Section button to begin the Module Exam.