In an intrusion case, spotting the difference between abnormal and normal is often the difference between success and failure. Your mission is to quickly identify suspicious artifacts in order to verify potential intrusions. Use the information below as a reference for locating anomalies that could reveal the actions of an attacker. 

In-depth:

Memory Artifacts

Rogue Processes
Malware authors generally pick one of two strategies for obscuring their malicious processes: hide in plain sight and attempt to appear legitimate, or use code injection and/or kernel methods to hide from the view of normal analysis tools. See below for more on code injection and rootkits. 

When searching for malware attempting to hide in plain sight, look for process names that appear legitimate but originate from the wrong directory path or with the wrong parent process or SID. Look for misspellings like "lsass.exe", commonly used by malware authors to avoid detection. 

A rootkit is a broad term for describing ways of subverting the operating system's control over processes and system resources. Malware authors generally pick one of two strategies for obscuring their malicious processes: hide in plain sight and attempt to appear legitimate, or use code injection and/or kernel methods to hide from the view of normal analysis tools. 

Software developers use the built-in "tasklist", "taskkill", and "whoami" commands to query installed services. Try parameters such as "queryex", "rundll32.exe", "svchost.exe", "winrm.vbs", "powershell.exe -wim", or "winrm.exe -c runkey". The keys here provide the parameters for such tasks as gathering service, the service name, display name, the service's executable file path, start type, required privileges, dependencies, and more. Each service key has a unique set of parameters and can be used to start, stop, or pause services, or to obtain information about a specific running service.

For offline analysis, investigate services in the registry using "regedit". Malware authors may hide code or data injection, so inspect the registry under "HKEY_LOCAL_MACHINE\CurrentControlSet\Services". 

Code Injection and Rootkit Behavior
Code injection and rootkit detection tools are widely available, such as "avcsoft.exe", "lsass.exe", and "svchost.exe". These tools can be used to detect malware attempting to hide in plain sight and attempt to appear legitimate, or use code injection and/or kernel methods to hide from the view of normal analysis tools. 

Typical code injection techniques provide an effective way to hide without relying upon kernel-level programming knowledge, thus making it very popular among malware authors. Code injection is almost never legitimate, with the one exception being software developer tools. Therefore, finding evidence of code injection on a standard system is almost always worth looking into further.

When programs are running, a process may be creating a rootkit that can be used to hide a process or thread from the view of normal analysis tools. Rootkits are installed by malware authors to prevent forensic analysis. Rootkits are installed by malware authors to prevent forensic analysis. Rootkits may be used to hide from the view of normal analysis tools. Rootkits may be used to hide from the view of normal analysis tools.

Code Injection and Rootkit Behavior
Code injection and rootkit detection tools are widely available, such as "avcsoft.exe", "lsass.exe", and "svchost.exe". These tools can be used to detect malware attempting to hide in plain sight and attempt to appear legitimate, or use code injection and/or kernel methods to hide from the view of normal analysis tools. 

Typical code injection techniques provide an effective way to hide without relying upon kernel-level programming knowledge, thus making it very popular among malware authors. Code injection is almost never legitimate, with the one exception being software developer tools. Therefore, finding evidence of code injection on a standard system is almost always worth looking into further.

When programs are running, a process may be creating a rootkit that can be used to hide a process or thread from the view of normal analysis tools. Rootkits are installed by malware authors to prevent forensic analysis. Rootkits may be used to hide from the view of normal analysis tools. Rootkits may be used to hide from the view of normal analysis tools.

Posters

SPRING 2014 – 29TH EDITION

digital-forensics.sans.org

Know Abnormal…Find Evil

SANS DFIR CURRICULUM

CORE

FOR108 Digital Forensic Foundations
FOR008 Computer Forensic Investigations - Windows In-Depth
FOR908 Computer Forensic Analyst - Incident Response GCFA
FOR508 Advanced Computer Forensic Analyst - Incident Response GCFA
FOR726 Windows Memory Forensics - In-Depth
FOR098 Forensics for Cyber Investigators
LEARN AEM

IN-DEPTH

FOR106 Memory Forensics
FOR104 Advanced Computer Forensic Analysts - Incident Response GCFA
LEARN R2M: Malware Analysis Tools and Techniques GREM
FOR555 Advanced Smartphone Forensics

SPECIALIZATION

FOR408 Windows In-Depth
FOR526 Windows Memory Forensics - In-Depth
FOR572 Advanced Forensic Analysis: Windows Services and Analysis
FOR410 Windows Memory Forensics - In-Depth
FOR610 Digital Forensic Foundations

GCIH Incident Handling, Incident Response & Digital Forensics
GCFE Digital Forensics & Incident Response
PCCF Computer Forensic Framework
FCFF Forensic Fundamentals
IN-DEPTH

FOR408 Windows In-Depth
FOR526 Windows Memory Forensics - In-Depth
FOR572 Advanced Forensic Analysis: Windows Services and Analysis
FOR410 Windows Memory Forensics - In-Depth
FOR610 Digital Forensic Foundations

GCIH Incident Handling, Incident Response & Digital Forensics
GCFE Digital Forensics & Incident Response
PCCF Computer Forensic Framework
FCFF Forensic Fundamentals

unknown services

Windows services are designed to run applications in the background without user interaction. Many services are required at system boot, including the DNS, Network, Internet Tuning, Remote Access, and Workstation services. These services provide critical functionality for the OS and must be started immediately without requiring user input.

Services can be implemented as standard services or as so-called "DLLs". In order to consume resources, most services will be grouped together and run under a single identity. The avcsoft.exe process is a Windows generic service based process, and it is typical to see several running instances of avcsoft.exe (up to 5 is common).

Service configurations, as well as service start type, are stored in the registry under "HKEY_LOCAL_MACHINE\CurrentControlSet\Services". The keys here provide the parameters for such tasks as gathering service, the service name, display name, and the service's executable file path, start type, required privileges, dependencies, and more. Each service key has a unique set of parameters and can be used to start, stop, or pause services, or to obtain information about a specific running service.

For offline analysis, investigate services in the registry using "regedit". Malware authors may hide code or data injection, so inspect the registry under "HKEY_LOCAL_MACHINE\CurrentControlSet\Services".

Unusual OS Artifacts
Malware does not need to be present as a system is to be compromised. We need to also look for unusual OS-based artifacts that would not exist on a typical workstation in the organization. When looking for program execution, focus on critical, dynamic, unauthorized registry keys, and even user input. Many of these artifacts can result from an adversary using your system but not exploiting malware. Look for evidence showing odd behavior such as running a program on your computer that you don't recognize, and even user input. Many of these artifacts can result from an adversary using your system but not exploiting malware. Look for evidence showing odd behavior such as running a program on your computer that you don't recognize, and even user input.

Evidence of Persistence

Malware commonly accomplishes persistence using a variety of techniques. The most often used capability to achieve persistence with elevated rights is through scheduled tasks using the "taskschecl.exe" command. Windows executables can contain a trigger to invoke a program or script or run an executable file with a new malicious executable. The next most common malware persistence mechanism is using the registry, auto-start mechanisms to load malware at boot or during user logon. Some of the latest techniques include DELL Search Order Hijacking and using local group policy editor. Attackers usually malware can also be integrated into a Microsoft Office Add-In where MS Word starts, the malware is executed.

Poster References
- Rootkit Analysis, 2nd Edition
- Windows System Administration Reference
- Security - Windows (SEC035)
- Advanced Incident Response (FOR908)
- Memory Forensics (FOR526)
- REM: Malware Analysis (FOR610)

Evidence of Suspicious Network Activity
Many core processes in Windows utilize the network, including avcsoft.exe, lass.exe, etc.exe, and even the operating system. Since you can't rule out the possibility of legitimate network activity from these processes, you need an effective way to identify malicious network activity. Network analysis tools can parse through existing network traffic and even residual network connections and system established by the system when you are connecting to the internet. You need to identify unusual network behavior, keep an eye out for the following:

- Any process communicating over port 80, 443, or 8080 is not a browser
- Any browser that communicates over ports 80, 443, or 8080
- Connections to unassigned external or external IP addresses. For example, why did a process have a TCP connection to a system in Moldova?
- Web requests directly to an IP address rather than a domain name
- RDP connections (port 3389), particularly originating from odd IP addresses. External RDP connections are typically routed through a VPN concentrator.
- DNS requests for unusual domain names
- DNS requests for unusual domain names
- DNS requests for unusual domain names
- DNS requests for unusual domain names
- DNS requests for unusual domain names

Evidence of Persistence

Malware commonly accomplishes persistence using a variety of techniques. The most often used capability to achieve persistence with elevated rights is through scheduled tasks using the "taskschecl.exe" command. Windows executables can contain a trigger to invoke a program or script or run an executable file with a new malicious executable. The next most common malware persistence mechanism is using the registry, auto-start mechanisms to load malware at boot or during user logon. Some of the latest techniques include DELL Search Order Hijacking and using local group policy editor. Attackers usually malware can also be integrated into a Microsoft Office Add-In where MS Word starts, the malware is executed.

Scheduled Tasks
- Auto-Start Registry Keys
- SQL Search Order Hijacking
- Trusted Systematic Software Libraries

More Advanced - PowerShell background job, Local Group Policy, MS Office Add-In, or RDS Sharing

Knowledge and tools are the most important factor success in an investigation. This poster was created by SANS instructors Mike Pilkington and Rob Lee.
When searching for malicious processes, look for any of these anomalous characteristics:

- Started with the wrong parent process
- Image executable is located in the wrong path
- Misbehaved processes
- Processes that are running under the wrong account (incorrect SID)
- Processes with unusual start times (i.e., starts minutes or hours after boot when it should be running within seconds of boot)
- Unusual command-line arguments
- Packaged executables

Use the information below as a reference to know what's normal in Windows and to focus your attention on the outliers.

Know what's normal on a Windows host helps cut through the noise to quickly locate potential malware.