Digital Forensics and Security Toolkit

Mattias Huber • Michael Soltys • COMP 499

Overview

As society becomes more reliant on technology to facilitate every aspect of our lives, vulnerabilities in the systems we confide in expose a risk to continued stability and prosperity. The importance of protecting sensitive data has brought information security to the forefront. Malicious individuals and criminal organizations have become increasingly brazen in seeking to steal or ransom valuable information. In order to secure their interests, nation states are pouring resources into developing cyber warfare capabilities which has resulted in a militarization of cyber-space. What were once a rarity, large breaches that cause drastic amounts of damage now occur regularly.

Unfortunately attackers are employing progressively complex techniques to discover vulnerabilities, break into systems, and cause damage. This is paired with the increased reliance on technology which presents a larger attack surface. Attackers are also extremely thorough in removing any trace of their presence, or may masquerade as other known attacker groups to throw off investigators.

Fortunately the information security community works tirelessly to defends our critical infrastructure from this constant barrage. This toolkit was built as an open-source solution to augment and aid information security professionals.

Windows Forensics Toolkit

What is Digital Forensics?

When investigating a possible breach, investigators are given the monumental task of piecing together what little information is left in order to build a profile of the events that occurred. This process is called digital forensics, and encompasses a wide field of techniques, such as auditing logs, recovering deleted files, and extracting registry information.

Objective:

This toolkit aids in a forensic investigation on Windows by pulling important information from the registry and filesystem, and compiling them into a report.

Open/Save Most Recently Used (MRU):

- When an open/save Windows dialog box is opened, if a file is opened or saved an entry is stored here.
- NTUSER.DAT\Software\Microsoft\Windows\CurrentVersion\Explorer\ComDlg32\OpenSavePIDlMRU

Last Visited MRU:

- When a file is opened from the Open Save MRU, the executable that opened the file is stored here, along with the path of the file.
- NTUSER.DAT\Software\Microsoft\Windows\CurrentVersion\Explorer\ComDlg32\LastVisitedPidlMR

Run MRU:

- When using Start->Run command, the command is stored here.
- NTUSER.DAT\Software\Microsoft\Windows\CurrentVersion\Explorer\RunMRU

Application Compatibility Cache:

- Windows checks every process executed for compatibility issues (such as applications needing to be run in legacy mode) and stores that information here. When the system has a graceful shutdown, all of the executed processes are stored here, up to 1024 entries on Windows 7+.
- SYSTEM\CurrentControlSet\Control\Session Manager\AppCompatCache

Jump Lists:

- Windows 7+, when one right clicks on an item on the task bar, a jump list appears that shows pinned files, frequently used files, and recently used files. All that information is stored here.
- C:\%USERPROFILE%\AppData\Roaming\Microsoft\Windows\Recent\ AutomaticDestinationsOn

Browser Artifacts:

- Internet Explorer, Firefox, and Chrome have folders which hold browser history (if enabled).
- Internet Explorer:
  - IE8-9 %USERPROFILE%\AppData\Roaming\Microsoft\Windows\IEDownloadHistory\index.dat
  - IE10-11 %USERPROFILE%\AppData\Local\Microsoft\Windows\WebCache\WebCacheV*.dat
- Firefox:
  - v3 %userprofle%\AppData\Roaming\Mozilla\Firefox\Profles\<random text>.default\downloads.sqlite
- Chrome:
  - Win7/8/10 %USERPROFILE%\AppData\Local\Google\Chrome\UserData\Default\History

Windows Backdoor Detection

What is a Backdoor?

Once an attacker has compromised a system, in order to have a persistent connection they may choose to install a 'backdoor' to facilitate future access. A backdoor is a means of bypassing authentication, such as circumventing the user login screen on Windows. This can be useful as it allows an attacker to pivot across systems through the network, or even regain access after a password reset. A very common post-compromise backdoor on Windows systems exploits a vulnerability in the Accessibility Suite.

What is the Accessibility Suite?

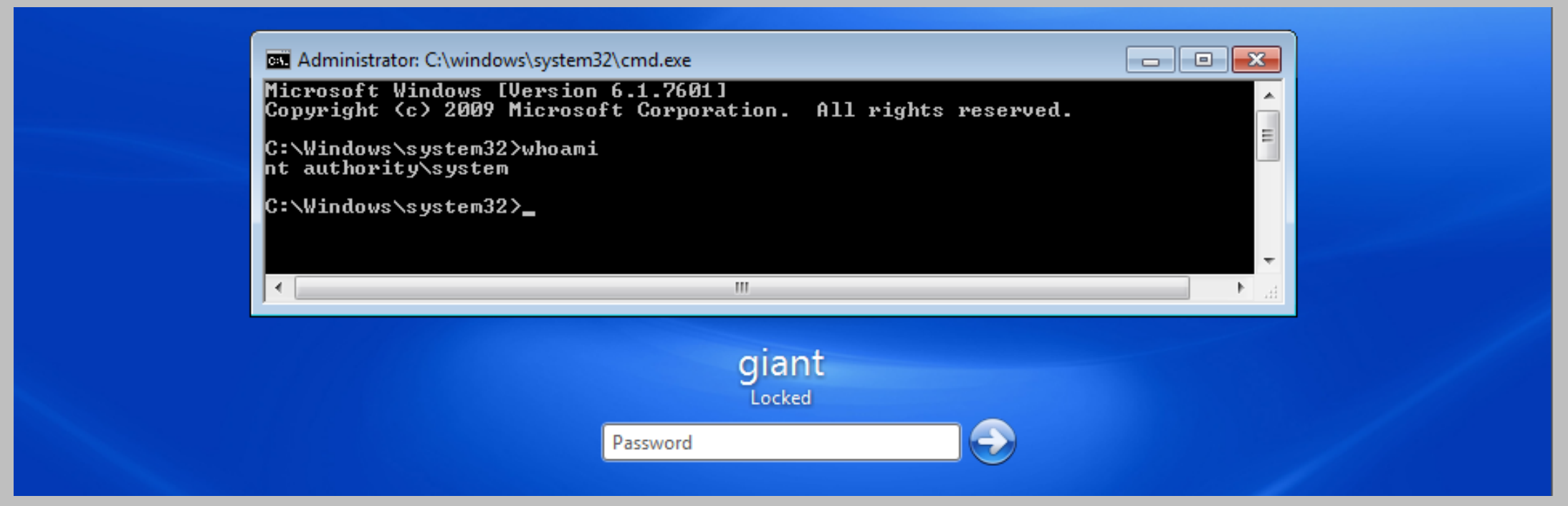
In order to facilitate users with disabilities, Windows comes with a set of tools called the Accessibility Suite. From a user accessibility standpoint, these tools must be available at the login screen as they may be crucial in assisting a user with disabilities to login. However from a security standpoint, since these tools are exposed at the login screen before authentication they are vulnerable to compromise.

Some of the Tools that are Exposed at the Login Screen are:

- Sticky Keys (sethc.exe) – Press shift 5 times in order to toggle shift-lock, ctrl-lock, or alt-lock.
- Display Switch (displayswitch.exe) – Toggles between multi-monitor modes by pressing Windows Key + P.
- Narrator (narrator.exe) – Reads text on the UI to the user, accessed through Windows Key + Ctrl + Enter.
- Utility Manager (utilman.exe) – Control panel for utilities, accessible by pressing Windows Key + U.
- On Screen Keyboard (osk.exe) – Displays a clickable on screen keyboard, accessible through Utility Manager.
- Magnifier (magnifier.exe) – Pops up a magnified Window, accessed through Utility Manager.

Objective:

Since the aforementioned exes are runnable at the login screen, replacing them with Command Prompt (cmd.exe), PowerShell (powershell.exe), or Explorer (explorer.exe) would result in unauthorized access to the underlying system from outside the login screen. Since there is technically no user logged in, the resulting Command Prompt would be running under SYSTEM account privileges, which is the account that the Windows operating system itself uses and is granted full control of all files on the system. Also since there is no user login, there is no logging footprint left behind. Lastly, cmd.exe, powershell.exe, and explorer.exe are legitimate Windows binaries do not show up as malicious files. There are two methods to install these backdoors on Windows by exploiting the Accessibility Suite.



```
Administrator: C:\windows\system32\cmd.exe
Microsoft Windows [Version 6.1.7601]
Copyright (c) 2009 Microsoft Corporation.  All rights reserved.

C:\Windows\system32>whoami
nt authority\system

C:\Windows\system32>_
```

Method I – File Replacement:

All of these files reside in the C:\Windows\System32\ folder. Replacing one of these exes with cmd.exe, powershell.exe, or explorer.exe would result in those exes being run instead of the exe that was replaced. Newer versions of Windows have enabled stricter protections of the System32 folder which prevents overwrite access, but it is still possible by using Takeown.exe to take ownership of the file and then replacing it. However, checking the hash of the file (its digital fingerprint) with known hashes of cmd.exe can be a sign of file replacement.

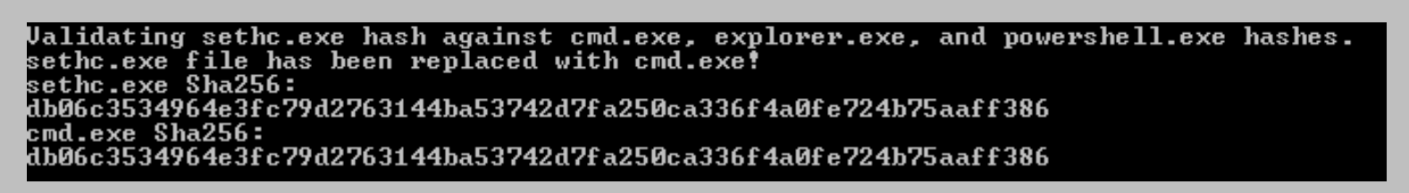
Method II – Debugger Registry Key:

A debugger is a program that attaches to another program to facilitate in the finding of bugs in the debugged program. A debugger can be specified for a program by adding a key to the Windows registry under Image File Execution Options, and when the debugged program is launched so is the debugger. The following example command shows how to attach cmd.exe to setch.exe:

```
REG ADD "HKLM\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Image File Execution
Options\sethc.exe" /t REG_SZ /v Debugger /d “C:\windows\system32\cmd.exe” /f
```

Windows Backdoor Detection and Removal

In order to detect file replacement, first the tool checks if setch.exe, utilman.exe, narrator.exe, osk.exe, magnifier.exe, and displayswitch.exe have been replaced by comparing their hashes against cmd.exe, explorer.exe, and powershell.exe.



```
Validating sethc.exe hash against cmd.exe, explorer.exe, and powershell.exe hashes.
sethc.exe file has been replaced with cmd.exe!
sethc.exe Sha256:
db06c3534964e3fc79d2763144ba53742d7fa250ca336f4a0fe724b75aaff386
cmd.exe Sha256:
db06c3534964e3fc79d2763144ba53742d7fa250ca336f4a0fe724b75aaff386
```

The tool then inspects the Image File Execution registry key for each exe and removes any found debugger keys. For investigative purposes only, the tool can also be used to install the backdoors by either adding registry keys or file replacement. Even though the tool will backup replaced exes, it is highly recommended that these files are backed up manually before. **Use at your own risk!**

Note:

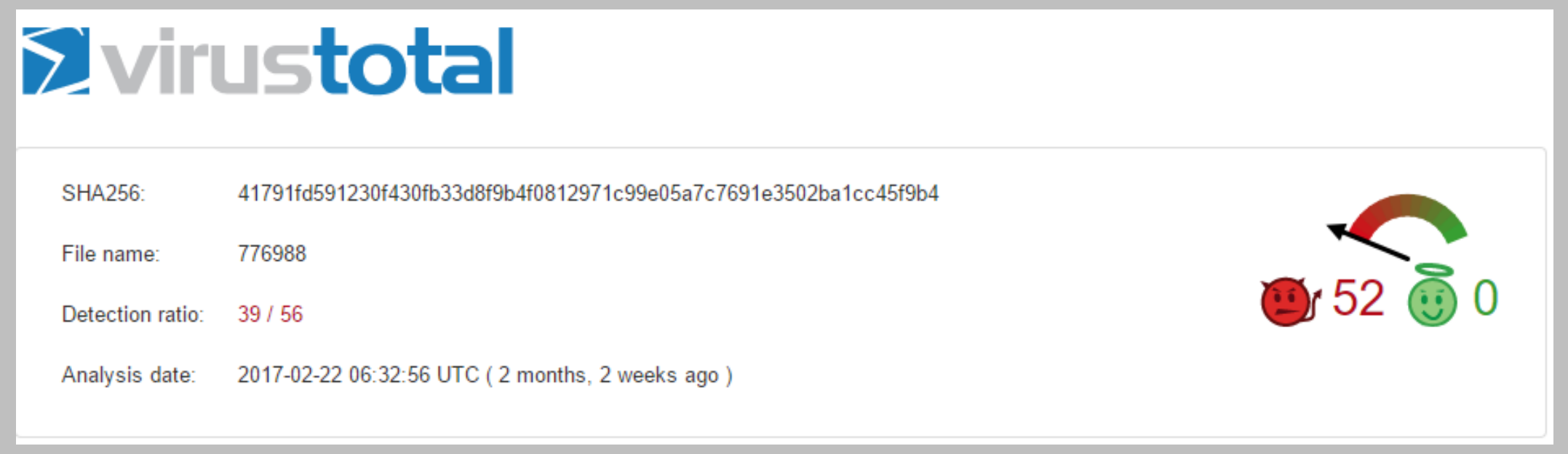
Installing these backdoors requires administrative access in order to replace files in the System32 folder or add keys to the registry. They are mostly used by attackers post compromise to regain access to a system. Therefore a found installed backdoor is not necessarily evidence the first malicious activity to occur, but may lead investigators to the initial compromise of a system.

Virus Total Automated Upload System

What is Virus Total?

Virus Total is a website (www.virustotal.com) where you can upload a file or supply a URL, and it will scan the target with around 60 different virus scanners from industry. It has become an essential tool for the information security community, as it not only aids in detection but also in false positives. Antivirus engine developers can gain valuable information about the performance of their scanners, such as receiving alerts when a known legitimate file is given a false positive.

Uploading to Virus Total expands the database, discovers more malicious files and false positives, and helps the security community. Users can even comment on files, detailing important information such as in-the-wild location, distribution vector, and deep malware analysis.



virustotal

SHA256:	41791fd591230f430fb33d8f9b4f0812971c99e05a7c7691e3502ba1cc45f9b4
File name:	776988
Detection ratio:	39 / 56
Analysis date:	2017-02-22 06:32:56 UTC (2 months, 2 weeks ago)

Objective:

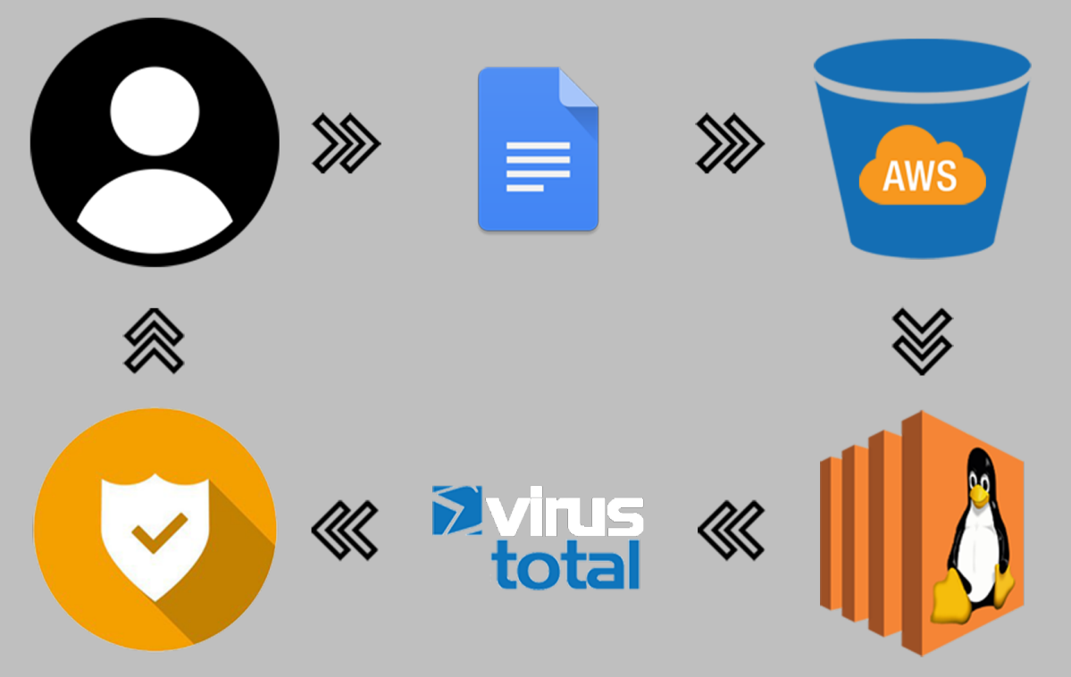
This tool can be used to upload a target file, directory, or URL to Virus Total. If the target is not already in the Virus Total database, the scan will be queued and completed shortly. Queued scans can take from a few seconds to a few minutes to complete, based on file size and current traffic. As this is an asynchronous process, this tool is useful in uploading jobs, checking if jobs have completed, and to display reports on completed jobs. The system also keeps track of all files uploaded, performs checks on already uploaded files to save bandwidth, saves all completed reports in a list, and all positive reports in a separate list. The system can be set to watch a certain directory and upload any new files that it encounters. It can also be used in 'scrape' mode, where given a URL, the system will find all other URLs on that website and upload them to Virus Total.

AWS Functionality:

Utilizing Amazon Web Services (AWS), Elastic Compute Cloud (EC2), and Simple Storage Service (S3), this system can be set up allow users to place files into a S3 bucket which will then be scanned automatically and user can be notified of any possible positives found. All of this is possible utilizing AWS free tier.

Elastic Compute Cloud allows users to rent a virtual computer in which they can use to automate their tasks. Simple Storage Service is a web facing cloud storage service that allows users to upload files from anywhere.

1. The User places a file they wish to scan into the S3 bucket, such as http://mybucket.s3.amazonaws.com
2. A dedicated EC2 instance watches the bucket, detects the new file, and uploads the file to Virus Total.
3. The EC2 instance waits until Virus Total returns a completed report.
4. If any positives are found the instance notifies the user, otherwise the report is added to the completed list.



Application Program Interface (API) Rules:

Interacting with Virus Total is done through their API, and must follow their guidelines. In order to interact with the API one must set up a Virus Total account in order to receive an API key which offers access with two tiers. The free tier of API access is limited to 6 requests a minute, while the paid for tier is limited to 600 requests per minute. The system is set up by default for the free tier request rate but can be increased to the paid tier if needed. Educational institutions and security researchers are eligible for an increased rate quota.

Final Note

All code is open source and available with instructions on how to setup and install at https://github.com/gitgiant

All code was written in Python 3+.