Registry Analysis with CrowdResponse »

By Chad Tilbury

Published: 2014-08-28 · Archived: 2026-04-05 16:37:13 UTC

The third release of the free CrowdResponse incident response collection tool is now available! This time around we include plugins that facilitate the collection of Windows registry data. Our inspiration for this release was one of those vulnerabilities that just won't die – Windows Sticky Keys. We'll show how to identify this attack while demonstrating the new additions.

New Plugins

```
@RegDump [-ds] <reg key>
```

RegDump recursively extracts Windows registry key and value data.

```
-d  Nested output format
 -s  Recursive dump
 <reg key> Registry key to start dump from
```

Valid registry hive names are: HKLM, HKCU, HKCR, HKU, and HKAU (pseudo key representing all users)

```
@RegFile [-scmh] <reg key>
```

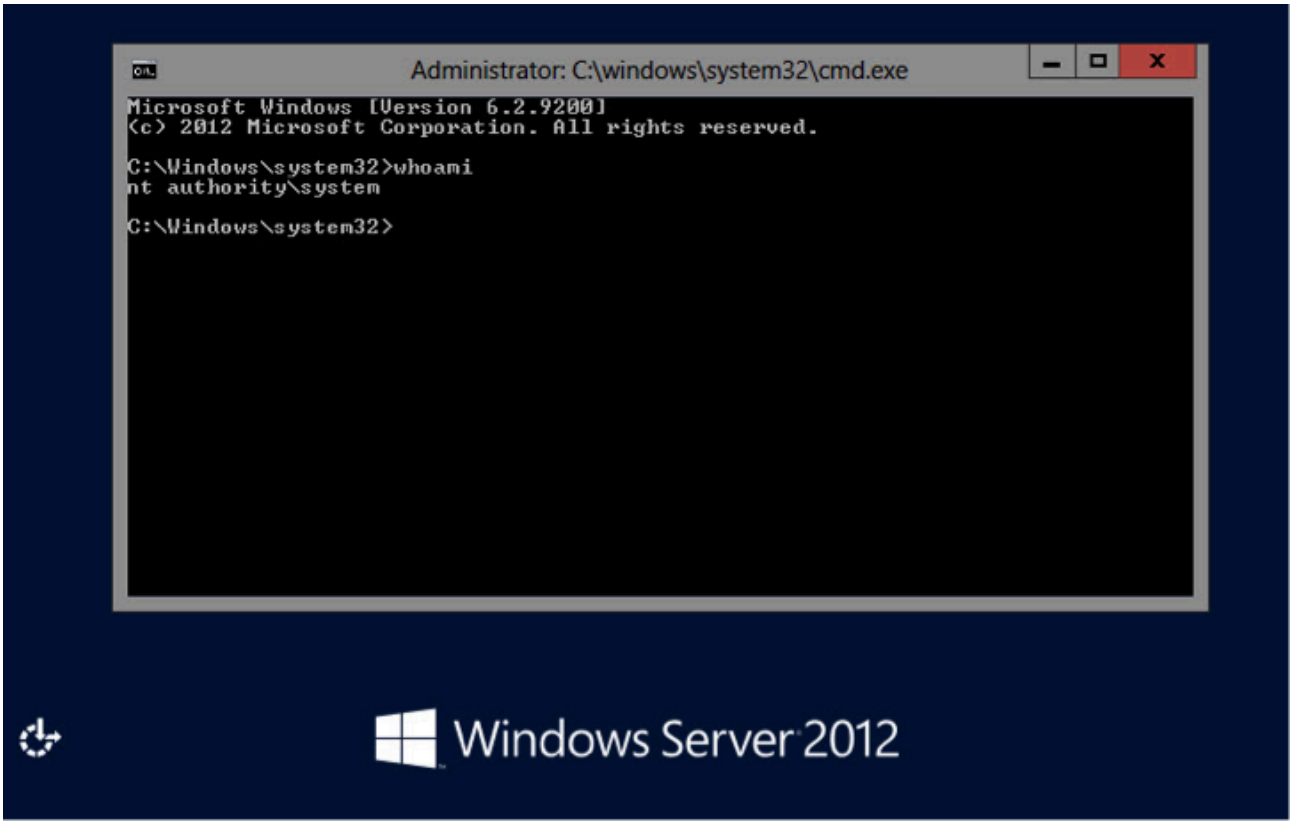
RegFile searches for registry string values (REG_SZ and REG_EXPAND_SZ) and identifies file path data. If the file exists on disk, file information, hash, and digital signature details are recorded.

```
-s  Recursive dump
 -c  Verify digital signature of discovered files
 -m Compute MD5 hashes
 -h  Compute SHA256 hashes
 <reg key> Registry key to start dump from
```

The Attack

The Sticky Keys attack is one of those vulnerabilities that is nearly too simple to believe. With a high success rate in most Windows environments, it is not surprising that we still see even some of our more advanced adversaries putting it into play. The original Sticky Keys attack involved replacing the C:\Windows\System32\sethc.exe binary with something that could provide access to the underlying OS, such as cmd.exe. After the switch, all it takes is five presses of the Shift-key from the logon screen and cmd.exe is executed. Since sethc.exe is executed pre-login, the attacker effectively gets a shell without needing to authenticate. This reduces the logging footprint

(no compromised account logon necessary!) and gives the added bonus of providing a shell running with LOCAL_SYSTEM privileges.



As new versions of Windows introduced slightly better protection mechanisms for the System32 folder, a new variant emerged – setting cmd.exe as a debugger to the sethc.exe process. One simple addition to the Windows registry and the attack works just as before, except there is no longer a need to perform file replacement.

```
REG ADD "HKLM\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Image File Execution Options\sethc.exe" /
```

Another common variant takes advantage of a different part of the accessibility suite, Utilman. The attack is identical to the sethc.exe registry debugger modification seen above, except the binary is now Utilman.exe and a simple Windows key + U combination will present a LOCAL_SYSTEM privileged shell.

```
REG ADD "HKLM\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Image File Execution Options\utilman.exe"
```

An important detail about this vulnerability is the attacker must have prior access to the system. That can be either via an administrator account (necessary to modify HKLM or write files to C:\Windows\System32) or via physical access to the machine, with modifications done via a bootable USB. Hence it is largely used post-compromise in conjunction with RDP as a convenient means for attackers to pivot through the network or to regain access after a password reset. Finding evidence of a Sticky Keys attack may lead the responder to the initial compromise of a system, but it will not be the first malicious activity to occur.

Finding Sticky Key Attacks

File Replacement

One of the clever things that make the Sticky Keys attack difficult to identify is that the file replacement uses a legitimate Windows binary (cmd.exe). Thus, the resultant file will still nicely match your list of known good hashes. The trick of course is to not only compare hashes, but to ensure you are matching the filename with the correct hash. Using the CrowdResponse @DirList plugin, we can pull filenames, hashes, and digital signatures for all files in the C:\Windows\System32 folder.

From the CrowdResponse config file:

```
@dirlist "%windir%\system32" -h -m -r -s -t -p 2 -z 30 -i "\.(exe|dll|sys)$"
```

With the output in hand (or in a backend database), a simple query for well-known sethc.exe or cmd.exe hashes can be constructed to easily identify anomalies. In this example we see cmd.exe and sethc.exe both having the same (cmd.exe) hash.

name	size	created	md5
C:\Windows\system32\cmd.exe	302592	2012-08-31T19:34:40Z	ad7b9c14083b52bc532fba5948342b98
C:\Windows\system32\sethc.exe	302592	2012-08-31T19:34:40Z	ad7b9c14083b52bc532fba5948342b98

Registry Analysis

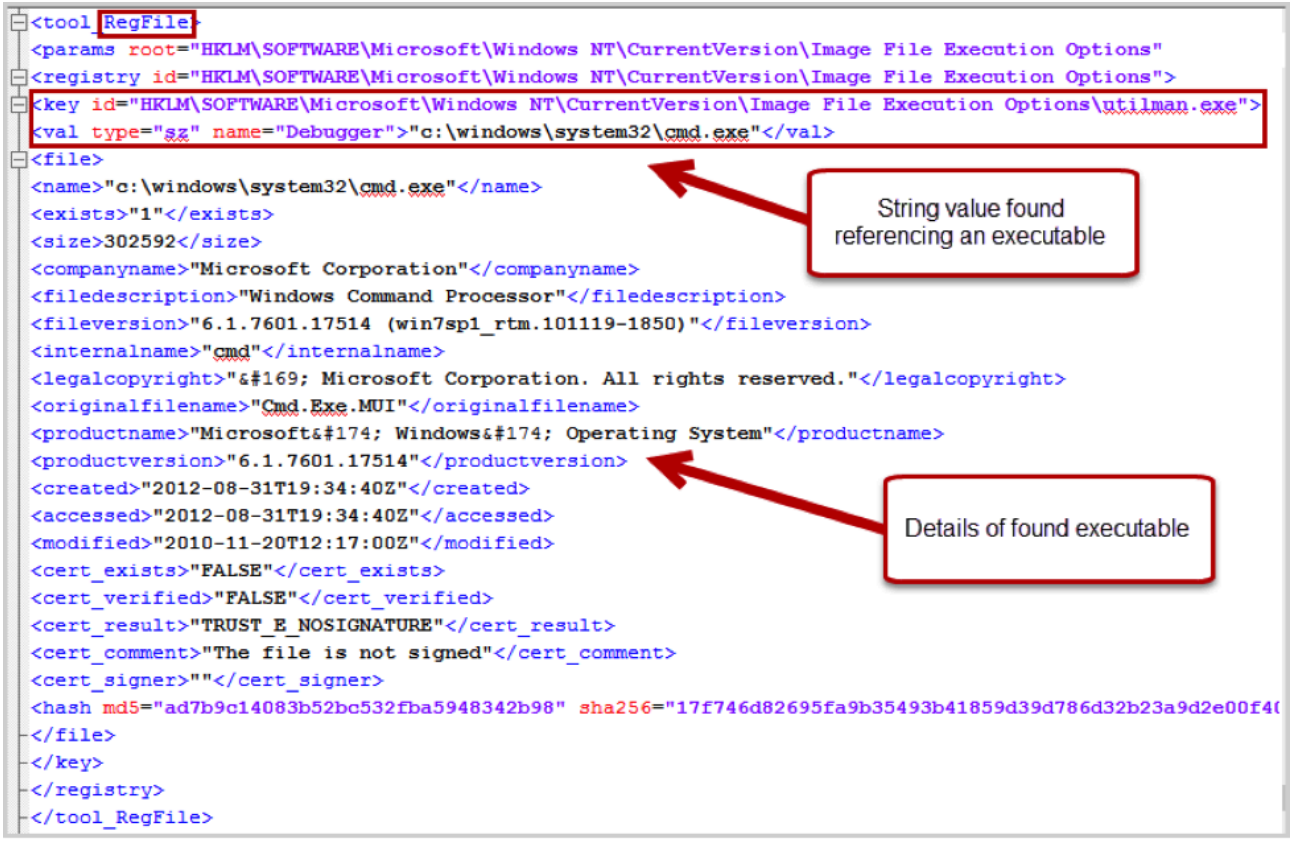
Imagine a tool that dumps all of your favorite registry locations for inspection along with validating any binaries that are referred to within those registry locations. This use case nicely covers the capabilities of CrowdResponse's new @RegDump and @RegFile plugins. Let's see how they work when searching for Sticky Key modifications.

From the CrowdResponse config file:

```
@RegDump "HKLM\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Image File Execution Options" -s
```

@RegFile "HKLM\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Image File Execution Options" -s -c -h -m

Regdump key	type	name	value
HKLM\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Image File Execution Options\ielowutil.exe	qword	MitigationOptions	256
HKLM\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Image File Execution Options\iexplore.exe	qword	MitigationOptions	256
HKLM\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Image File Execution Options\msfeedssync.exe	qword	MitigationOptions	256
HKLM\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Image File Execution Options\mshta.exe	qword	MitigationOptions	256
HKLM\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Image File Execution Options\utilman.exe	sz	Debugger	c:\windows\system32\cmd.exe




```
<tool_RegFile>
<params root="HKLM\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Image File Execution Options"
<registry id="HKLM\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Image File Execution Options">
<key id="HKLM\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Image File Execution Options\utilman.exe">
<val type="sz" name="Debugger">"c:\windows\system32\cmd.exe"</val>
<file>
<name>"c:\windows\system32\cmd.exe"</name>
<exists>"1"</exists>
<size>302592</size>
<companyname>"Microsoft Corporation"</companyname>
<filedescription>"Windows Command Processor"</filedescription>
<fileversion>"6.1.7601.17514 (win7sp1_rtm.101119-1850)"</fileversion>
<internalname>"cmd"</internalname>
<legalcopyright>"&#169; Microsoft Corporation. All rights reserved."</legalcopyright>
<originalfilename>"Cmd.Exe.MUI"</originalfilename>
<productname>"Microsoft&#174; Windows&#174; Operating System"</productname>
<productversion>"6.1.7601.17514"</productversion>
<created>"2012-08-31T19:34:40Z"</created>
<accessed>"2012-08-31T19:34:40Z"</accessed>
<modified>"2010-11-20T12:17:00Z"</modified>
<cert_exists>"FALSE"</cert_exists>
<cert_verified>"FALSE"</cert_verified>
<cert_result>"TRUST_E_NOSIGNATURE"</cert_result>
<cert_comment>"The file is not signed"</cert_comment>
<cert_signer>""</cert_signer>
<hash md5="ad7b9c14083b52bc532fba5948342b98" sha256="17f746d82695fa9b35493b41859d39d786d32b23a9d2e00f4(
</file>
</key>
</registry>
</tool_RegFile>
```

The @RegDump output clearly shows utilman.exe as an outlier (it was the only one in the list with a “debugger” value) and @RegFile nicely identified the full path in that registry value and provided hash and digital signature information. While this example shows us using the new registry plugins to identify Sticky Key modifications, they are equally useful to collect known hotspots in the registry, such as the infamous “Run” keys and other registry-based ASEPs.

YARA

You may recall that one of the first features of CrowdResponse was the ability to scan memory and disk for arbitrary YARA signatures. YARA's power lies in its simplicity, and there are some great signatures available to help you find all kinds of evil. Florian Roth at BSK Consulting shared a clever YARA signature for detecting sethc.exe file replacement. The signature looks for well-known strings within sethc.exe, and if they are not found, flags the binary for potential file replacement. A slightly modified version can be used with CrowdResponse:

```
CrowdResponse -o out.xml @yara -m sethc.yar -f sethc.exe -t        %windir%\System32
```

Contents of sethc.yar:

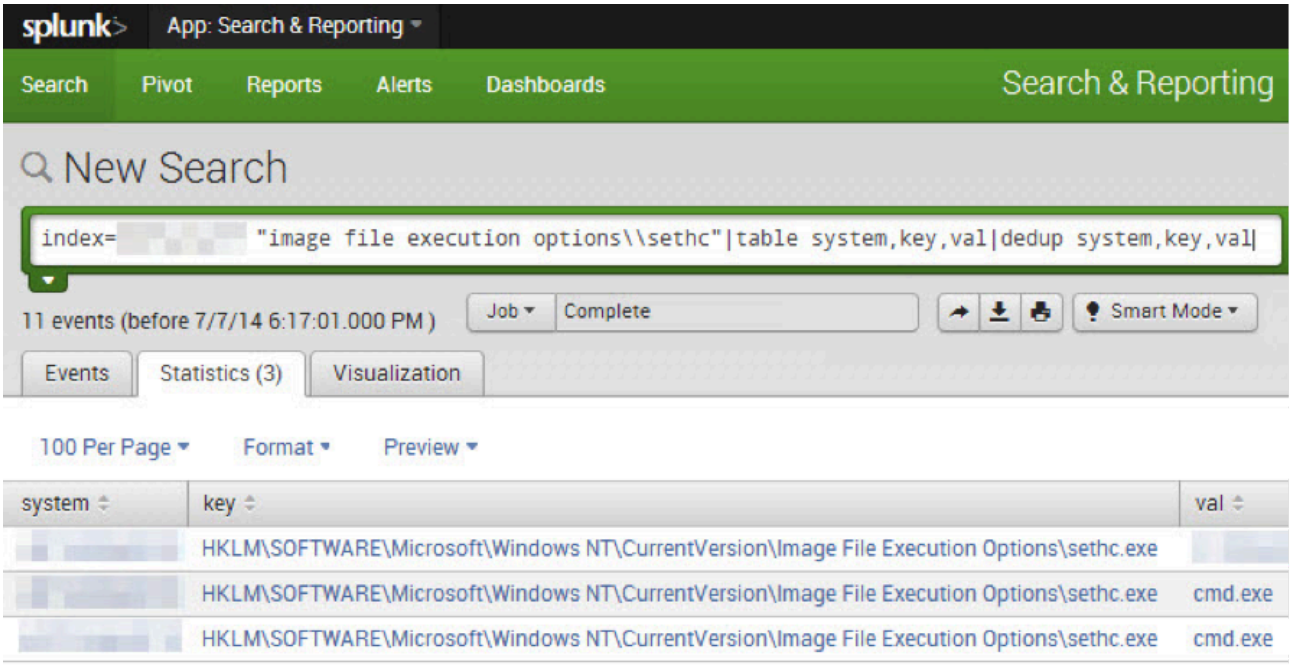
Rule Sethc_Replace

```
{
strings:
$s1 = "stickykeys" fullword nocase
$s2 = "stickykeys" wide nocase
$s3 = "Control_RunDLL access.cpl" wide
$s4 = "SETHC.EXE" wide
}
```

The command uses the CrowdResponse @YARA module to scan anything named sethc.exe in the \Windows\System32 folder, flagging the binary if specific strings are not present.

Scaling Your Efforts

It is increasingly necessary to collect and analyze incident response data at scale. Scaling analysis to hundreds or thousands of systems requires a process that can collect the results of a tool like CrowdResponse and get the data into a searchable repository. Once accomplished, you can easily detect the Sticky Keys attack with a search similar to the one seen below.



In this example an analyst with knowledge of the Sticky Keys vulnerability created a simple query and identified three new compromised systems. Other possible searches include the presence of a "Debugger" value and hash comparisons of sethc.exe and utilman.exe.

Remediation

What can be done about this attack once it is identified? Like many Windows vulnerabilities, the most effective mitigation is to prevent adversaries from achieving administrative privileges. Modifying the necessary keys or files in this attack requires administrative rights. Beyond that, there are some things you can do to make the attack slightly more difficult and potentially easier to detect.

- Upgrade systems to at least Win7/Win2008R2. This does not remove the vulnerability, but at least makes file replacement slightly more difficult.
- Turn off sticky keys via the registry. An obvious solution, but unfortunately an attacker with administrative rights can easily re-enable.
- Enforce Network Level Authentication for RDP connections. Available starting with Windows 7 and Server 2008, NLA requires account authentication before a RDP session is established. NLA could be an elegant fix, except that an adversary with administrative rights can easily disable it.
- Require IPsec to pre-authenticate RDP sessions. This solution mitigates any advantages Sticky Keys give to an attacker, but requires significant up-front work.
- Set real-time triggers looking for changes to file system or registry in your security tools. Tune end-point monitoring or HIPS to look for telltale signs of file replacement and registry modification.

Want to try it yourself? Download the latest version of CrowdResponse here. A big thank you to Robin Keir and Danny Lungstrom for their work on this release.

The CrowdStrike team is committed to developing and delivering free community tools like CrowdResponse, CrowdInspect, Tortilla, and the Heartbleed Scanner. If you have any questions or would like additional

information on our services, products, or intelligence offerings, please reach out to us via our contact page. For immediate assistance with an Incident Response, the CrowdStrike Services team can be contacted at 1.855.CROWDIR (276-9347).
