Hunting for GetSystem commands in offensive security tools

By susannah.matt@redcanary.com

Archived: 2026-04-05 20:12:02 UTC

Due to its privileges, the Windows NT AUTHORITY\SYSTEM account is a juicy target for adversaries across all versions of Windows operating systems. The SYSTEM account is the highest level administrator for a host. When a user is a standard member of the Administrators Windows group, User Account Control (UAC) and certain security boundaries still apply to slow down potentially unauthorized activity. No such boundaries apply to the SYSTEM account on a local system, allowing it to make changes to a system as needed.

What is GetSystem?

Windows Services often need this level of privilege for system management. Client management and deployment products often use SYSTEM to allow software installations. Security software often uses SYSTEM to peer into the activity of other users on a system, a use case that also appeals to adversaries. When using SYSTEM, an adversary can monitor and manipulate data from any other user on that local computer. While this account doesn't allow an adversary network access to log on to other computers, it does allow the adversary to execute credential access attacks against files and memory on a computer to compromise credentials for network access. This is commonly seen with attacks that use tools like Mimikatz. In the really unfortunate cases where adversaries gain access to the SYSTEM account on Active Directory domain controllers, they can grab credentials for any users within the domain and manipulate Active Directory to add accounts for themselves.

This is why many offensive security tools include a command named `getsystem` or similar. These commands make those tools try one or more things to elevate privileges to that SYSTEM account so the adversary can own everything on the victim host.

GetSystem in Meterpreter & Cobalt Strike's Beacon

Two of the most prevalent adversary tools that Red Canary sees on a weekly basis are Metasploit's Meterpreter payload and Cobalt Strike's Beacon. These payloads serve as malicious agents for adversaries to manage and control victim computers. Interestingly, both of them implement a `getsystem` command into their payloads in an incredibly similar manner using multiple methods.

Both tools first attempt to use "named pipe impersonation" to achieve SYSTEM privileges. This involves creating a Windows Service to execute as `NT AUTHORITY\SYSTEM` and feeding data to it through a named pipe that is randomly created by the malicious payload. An in-depth explanation of this technique can be found here.

Hunting tips

In the case of Cobalt Strike's Beacon, the Windows `services.exe` process will execute `cmd.exe` with a command line like this:

```
cmd.exe /c echo ba80ae80df9 > \\.\pipe\66bee3
```

Metasploit's Meterpreter also presents itself in a predictable way spawning from `services.exe`:

```
cmd.exe /c echo fvxens > \\.\pipe\fvxens
```

You can easily hunt for this behavior with two evidence sources: process monitoring data or Windows Event Logs.

With process monitoring, hunt for processes matching these criteria:

- parent process is `services.exe`
- process name is `cmd.exe`
- command line includes `echo` AND `\pipe\`

With Windows Event Logs, search for events with the ID 7045 that match these criteria:

- ServiceFileName contains `cmd.exe` OR `%COMSPEC%`
- ServiceFileName contains `echo` AND `\pipe\`

Both of these hunts will reliably find adversaries using named pipe impersonation from both tools. The second GetSystem method uses `rundll32.exe` and a few hardcoded command line options to execute a DLL for privilege escalation. Thankfully, the command line options are consistent and appear similar to this:

```
rundll32.exe C:\Users\user\AppData\Local\Temp\fvxens.dll,a /p:fvxens
```

As with named pipe impersonation, you can use process monitoring to hunt for this. Look for processes matching these criteria:

- process name is `rundll32.exe`
- command line includes `,a /p:`

In addition to these methods, both tools also support a third method that involves token manipulation. The first two methods seem to be more prevalent than the third. However, the token manipulation method isn't readily observable via process monitoring data, so we don't have a great deal of visibility into it. It's not entirely clear whether a perceived lack of token manipulation is representative of reality or the result of our blindspots.

GetSystem in Empire & PoshC2

As with Metasploit and Cobalt Strike, we see GetSystem commands in Empire and PoshC2—but to a slightly lesser extent. Both advanced and opportunistic adversaries use these tools in the wild and they implement `getsystem` using PowerShell. Both tools have adopted the Get-System.ps1 script from the PowerSploit project, and it also has a distinct command line to conduct named pipe impersonation:

```
cmd.exe /C start %COMSPEC% /C `"timeout /t 3 >nul&&echo TestSVC > \\.\pipe\TestSVC`
```

The beautiful thing about this similarity is that you can reuse the first hunts for named pipe impersonation in Metasploit and Cobalt Strike to also search for named pipe impersonation from Empire and PoshC2!

What you can look for now

If you're looking for a reliable, high-fidelity way to alert on Metasploit Meterpreter, Cobalt Strike Beacon, Empire, or PoshC2 GetSystem activities you can implement these hunts today:

- parent process is `services.exe`
- process name is `cmd.exe`
- command line includes `echo` AND `\pipe\`
- Event ID 7045
- ServiceFileName contains `cmd.exe` OR `%COMSPEC%`
- ServiceFileName contains `echo` AND `\pipe\`

Happy hunting!
