# Understanding the Risks of LOLBAS in Security - Pentera

By Nir Chako

Archived: 2026-04-05 15:54:11 UTC

**Living Off the Land Binaries and Scripts (LOLBAS)** represent a stealthy and growing threat in cybersecurity. By using trusted system utilities for malicious purposes, LOLBAS exploits challenge security teams, making it harder to detect and mitigate these advanced attack methods. This makes it hard for security teams to distinguish between legitimate and malicious activities, since they are all performed by trusted system utilities. Since LOLBAS are one of the growing trends in cyber-security attacks and they are also very hard for security solutions to detect, we set out to find new official LOLBAS. In this blog post, we'll show how we found 12 new LOLBAS that security professionals should protect against. To read a more in-depth explanation of the process, as well as our proposed framework for LOLBAS identification, you can read the entire research paper here.

## Discovering New LOLBAS Threats: Step-by-Step"

On our quest to find new LOLBAS, we started by using Oddvar Moe's approach. Oddvar is the founder of the official open-source LOLBAS project. He suggests a two-step process:

1. List all the binaries.
2. Try them one by one

So that's what we did. Starting specifically by looking for new LOLBAS downloaders from the Microsoft Office suite.

### Step 1: The Manual Approach

1. We listed all the binaries in the Office suite installation folder.
2. We tried to run the executables with a URL to download a file from as the argument.
3. We initiated an HTTP server that will give an indication about a successful download attempt.

This manual process highlights how effective LOLBAS can be in evading detection by using trusted binaries. Each time we executed a binary with a URL as the argument we waited for a GET request in the HTTP server, which means that the triggered binary wanted to GET something from the HTTP server, i.e trying to download a file. After finding the LOLBAS download trigger, it's easy to find the location of the downloaded files by tracking the downloader with ProcMon.

Within two hours, we found three (!) new LOLBAS downloaders from the Microsoft Office suite.

### Step 2: The Automated Approach

Since Windows OS contains more than 3,000 executable files, running them manually is not a practical approach. Therefore, we decided to build an automated solution, The automated solution needs to list all the binaries, and then go over them one-by-one to and try to trigger a potential downloader. To do so, we ran the simplest command structure that could initiate a download from an HTTP server. Its structure includes only two parts:

- The path of the potential downloader
- A URL to download the file from

The code itself looks something like this:

Then, the tools need to receive feedback on the download attempt. This part includes an HTTP server, like the one we used in the manual approach. The HTTP server log records provide an indication about the file download attempt.

Using this automated method, we managed to find six more downloaders. All in all, we discovered nine new downloaders. That’s almost a 30% increase in the official LOLBAS downloaders list!

### Step 3: Finding LOLBAS Executors

We were so excited about our findings that we decided to continue to find new LOLBAS with other functionalities. This time, we focused on LOLBAS executors. In a complete attack chain, a hacker will use a LOLBAS downloader to download more robust malware. Then, they will try to execute it. LOLBAS executors allow attackers to execute their malicious tools as part of a seemingly legitimate looking process tree on the system. Just like before, we started by iterating over all the binary files on the system, trying to execute a “FILE_TO_EXECUTE” by passing it as an argument of the executor.

Then, we added cli execution flags using hyphen, dash and slash, while iterating over all the ABC letters (lowercase and uppercase). We added these because using different flags affects the execution flow of the program. See examples below:

You might notice there is no HTTP server as an indicator of a trigger. In this scenario, our feedback is based on the way that the operating system manages the process tree. For example, if we run the notepad from cmd, we can look up the process parent and get a clear indication of the executor by its name. To implement this logic in our

research, we developed a helper with the task of finding the name of its process parent and writing it to a log file, if it was executed. This enabled us to get our much-needed indication of the test file execution by the executor.

We managed to find **three new executors** by using this approach! SCP, sftp and our beloved MsoHtmEd. Sftp was found as an executor with the '- D' flag.

Afterwards, we tried reversing the process and found out that the usage of –D leads to the use of the CreateProcess API, which is the windows API call for running a new process. In our case – it ran the Exe helper. In potential future cases,  it might execute malware as part of a cyber attack campaign.

Understanding how LOLBAS executors function is crucial for identifying potential vulnerabilities in your system.

## Key Takeaways for Defending Against LOLBAS for Red Teamers, Blue Teamers & Researchers

Prior knowledge about LOLBAS tactics can help organizations proactively strengthen their defenses. Let's not wait to hear about an unknown LOLBAS taking part in the next cyber attack campaign. Prior knowledge about these threats can help organizations effectively boost their security measures and mitigate potential risks. We hope our research and our findings help Red/Blue Teamers and Security Researchers protect against existing LOLBAS and discover new ones. One last thing before we go. The official LOLBAS project has criteria that dictate that a LOLBAS must be either a Microsoft signed file that is native to the OS or downloaded directly from Microsoft. But theoretically, attackers could use other platforms, like Zoom, Slack or PyCharm, as Downloaders and Executors. Just something for you to think about…

## Living Off the Land Binaries and Scripts (LOLBAS): A Persistent Challenge

Living Off the Land Binaries and Scripts (LOLBAS) remain a stealthy attack method, leveraging native tools to bypass traditional detection mechanisms. These tactics are particularly challenging for security solutions because they exploit trusted system utilities. Proactively identifying threats from these native tools can significantly enhance resilience. For example, organizations focusing on ransomware readiness assessments can uncover hidden exposures that LOLBAS might exploit. Similarly, a deeper dive into the trends from the Verizon 2024 DBIR highlights how enhancing visibility into LOLBAS-related activity is critical for modern threat detection and mitigation strategies. **To read more information about how we found these LOLBAS, as well as our proposed framework for security professionals for finding new LOLBAS, read the complete research paper here.**

---