Living Off the Land: How threat actors use your system to steal your data

By Barracuda Networks

Published: 2025-03-03 · Archived: 2026-04-06 03:15:29 UTC

Almost every advanced threat actor has added Living off the Land (LotL) techniques into their attacks. LotL is an attack strategy where threat actors conduct malicious activities by exploiting legitimate tools and features already present in a target. The phrase "living off the land" means surviving on resources you find in an existing environment. If the environment is a physical ecosystem like a forest, it means sustaining yourself on what you can forage, grow, etc. If the environment is a digital network, it means conducting an attack with the binaries, scripts, and other tools that are already at work in the victim's digital environment. The term was applied to these techniques in 2013.

Traditional malware, fileless attacks, and LotL

Before we get into the details, we need to understand the difference between traditional malware, fileless attacks, and LotL techniques.

Traditional malware relies on external malicious files to move through a computer or network and damage the systems. Let's use WannaCry ransomware as an example. WannaCry ransomware was the notorious cryptoworm that infected over 230,000 computers in 150 countries in just one day. It accessed and took control of computers vulnerable to the EternalBlue exploit. Once established, WannaCry installed the ransomware and used the host computer to replicate and infect other vulnerable machines. Technically, WannaCry installed three pieces of malware to the machine.

A fileless attack is one that executes malicious code directly from memory. It does not write any files to disk, and it often uses system tools and macros to carry out the attack. Fileless attacks may or may not be LotL attacks, and this distinction comes down to a strict definition of LotL. A browser-based JavaScript attack like SocGholish is fileless because it runs in browser memory and doesn't write to disk. However, JavaScript is not a system administration tool, and the malicious commands are normally introduced from an external source like an infected website. There are some grey areas around this, but it's enough to know that some fileless attacks are not LotL.

LotL attacks may combine these two types of attack by leveraging system tools like PowerShell with files that are written to the disk for delayed execution. For example, an LotL attack could be launched by someone opening a malicious file that was previously downloaded or dropped in a previous attack.

LotL has been widely adopted by threat actors and is now included in most advanced attacks.

A Brief History of LotL Techniques

Living-off-the-Land is nothing new. Although the LotL terminology did not exist at the time, the 1989 Disk Operating System (DOS) virus 'Frodo' is considered one of the first to use LotL techniques to remain stealth until the payload was activated. Once launched, Frodo was memory-resident and intercepted DOS interrupt calls to hide its presence. The 2001 Code Red worm targeted Microsoft IIS servers with buffer overflow and denial-of-service (DoS) attacks. This malware exploited CVE-2001-0500 and operated entirely in memory with no writing to the disk. Code Red defaced websites and slowed sites and network electronics with excessive traffic.

The 2003 'SQL Slammer,' also known as the Sapphire Virus, was a worm that spread via port 1434, commonly found open on Microsoft SQL Server 2000, Microsoft SWL client-side applications, and MSDE 2000 systems. Once a system was infected, it replicated the worm to every vulnerable computer it could find. SQL slammer generated over 25,000 infection packets per second, and infected about 75,000 systems within the first hour. SQL Slammer was the first widespread fileless and LotL attack.

LotL attacks have grown rapidly since then. Almost every new capability added to operating systems led to new advancements in cyberthreats. Eventually LotL techniques grew to the point that it earned its own terminology:

- **LOO – Living off the Orchard**: A reference to LotL attacks that target MacOS. The 'orchard' is a play on the Apple logo.
- **LOLBins – Living off the Land Binaries**: This term was introduced by security researcher Oddvar Moe in 2018. LOLBins refers to legitimate system binaries that can be exploited for malicious purposes. Common examples:
  - **Microsoft Windows**: PowerShell, Rundll32, Regsvr32, Certutil, Bitsadmin.
  - **MacOS**: Curl, OpenSSL, Nscurl, Xattr, Launchctl
  - ***nix**: Curl, OpenSSL, Bash, Python, Nc (Netcat)
- **LOLScripts – Living off the Land Scripts**: Like it sounds, this is term refers to the legitimate scripts and scripting languages. Examples:
  - **Microsoft Windows**: PubPrn.vbs, CL_LoadAssembly.ps1, CL_Mut3exverifiers.ps1, Pester.bat, winrm.vbs
  - **MacOS**: osascript, bash, python, ruby, perl
  - ***nix**: bash, python, perl, awk, sed

Now, let's put this together with the top five ways that threat actors use LotL techniques:

LotL Use	**Windows**	**macOS**	**Linux/Unix**
Lateral Movement	- PsExec - WinRM - PowerShell - WMI	- SSH (Secure Shell) - Osascript - Bash scripts	- SSH - Bash scripts - Python scripts

Privilege Escalation	- PowerShell - Rundll32 - Reg.exe	- Sudo - Dscl - Osascript	- Sudo - Setuid binaries - Cron jobs
Data Exfiltration	- Bitsadmin - Certutil - PowerShell	- Curl - Rsync - SCP (Secure Copy Protocol)	- Curl - Rsync - SCP - Netcat
Persistence	- Schtasks - Reg.exe - WMIC	- Launchctl - Cron jobs - Plist files	- Cron jobs - Systemd services - Init scripts (Initialization scripts)
Execution of Malicious Payloads	- PowerShell - Mshta - Rundll32	- Python - Perl - Bash	- Python - Perl - Bash - Awk

What kind of threat actor lives off the land?

LotL attacks are common in ransomware and espionage, but you don't typically find them in DDoS or phishing attacks. Infostealers and banking trojans both use LotL, while cryptocurrency wallet stealers do not. LotL allows threat actors to blend in with normal system activities, making the attack more difficult to detect, especially in the absence of threat intelligence and other advanced security measures. However, LotL does have its drawbacks:

- Limited functionality: Custom malware can provide more flexibility and control over an attack than system tools designed for a specific purpose.
- Environmental variability: LotL techniques depend on the victim's environment having the right set of tools. If the environment doesn't have these tools, the attack will not be effective.
- Attacker expertise: LotL attacks require an understanding of system architecture and behavior.
- Speed v stealth: LotL attacks may require patience, and many attackers prioritize speed and additional functionality over the stealth of LotL.
- Improved detection: Monitoring and anomaly detection techniques are advancing rapidly. Threat actors are willing to mix techniques and try new things to stay ahead of defenders.

Let's go back to the cryptocurrency wallet stealer. This is malware designed to locate and extract the sensitive data needed to access the digital assets. This data includes private keys, wallet files, and sometimes even passwords or

seed phrases. The wallet stealer specifically scans for infected systems for wallet information and copies and exfiltrate this information back to the attacker's system. The attacker will then attempt to access or transfer funds from the wallet. This malware has to work fast before a victim can disrupt the attack or transfer funds out of the wallet. This malware targets a broad range of systems and often follows a larger phishing or malware attack. For these reasons, LotL techniques are not a good fit for wallet stealer malware.

Defend yourself from LotL tactics

Detecting LotL attacks is challenging because they exploit trusted tools, but a proactive defense is possible with some planning. This should be part of the company cybersecurity strategy.

Use solutions like Barracuda Managed XDR to monitor systems for behavioral anomalies and uncommon network activity. Make sure your systems are logging script executions and unusual process creation. Limit the use of high-risk LOLBins and LOLScripts through whitelisting or other measures.

Maintain a strong patch management system and conduct regular vulnerability assessments.

Segment networks to isolate sensitive environments and limit possibilities for lateral movement. Determine the normal traffic and network activity and configure security solutions to flag deviations. Maintain strong patch management and conduct regular vulnerability and risk assessments.

It's critical to use the principle of least privilege (PolP) and require multi-factor authentication (MFA) for all users. Configure behavioral analytics and flag activity that may indicate abnormal user behavior.

Barracuda can help

Barracuda Managed XDR is an extended visibility, detection, and response (XDR) platform, backed by a 24×7 security operations center (SOC) that provides customers with round-the-clock human and AI-led threat detection, analysis, incident response, and mitigation services.
