

# ELF Malware Analysis 101: Linux Threats No Longer an Afterthought

By Avigayil Mechtinger

Published: 2020-06-16 · Archived: 2026-04-10 02:46:26 UTC

Linux has a large presence in the operating systems market because it's open-sourced, free, and software development oriented—meaning its rich ecosystem provides developers easy access to many different artifacts. Linux is the predominant operating system for [Web servers](#), [IoT](#), [supercomputers](#), and the [public cloud workload](#). Although Linux holds only [two percent](#) of the desktop market share in comparison to the 88 percent share held by Windows, Linux desktop security should not be neglected, evidenced by our discovery of [EvilGnome](#) in July 2019.

Linux is practically everywhere but low Linux threat detection is pervasive across the antivirus industry, encouraging attackers to target this operating system aggressively in recent years. Researchers have disclosed highly sophisticated ELF malware, proving attackers are increasingly adding Linux malware to their arsenal. Currently, there aren't enough companies hunting for and publishing IOCs and other information about the latest Linux threats. There are many undiscovered threats on this operating system and we expect more threats will be exposed over time as Linux continues to gain in popularity. It's crucial that security researchers have the ability to analyze and understand Linux malware as part of their evolving skillset.

We initiated this training to make practical ELF malware analysis more accessible. This multi-part series will provide you with practical knowledge and tools for effective ELF malware analysis. You will gain a better understanding of the ELF format and learn how to analyze ELF files using static and dynamic methods. Also, we will present useful analysis tools and practice [malware analysis](#) hands on. After this series you will be able to analyze an ELF file, determine if it's malicious, and classify the threat.

Before diving into technical ELF analysis practices, this post will serve as an introduction to the ELF malware world. We will review the ELF threat landscape, explain how a Linux machine is initially infected with malware, and elaborate why it's important for you as a security researcher or malware analyst to gain ELF analysis skills.

## The Linux Threat Landscape

The Linux threat landscape is heavily concentrated with DDoS botnets and crypto-miners. It's much more complex than that, however, home to more sophisticated threats developed by APTs and other cybercrime groups. In 2019, our researchers documented over 20 instances of previously undiscovered Linux threats. Those threats included large scale crypto-mining campaigns, botnets, ransomware, and nation-state sponsored attacks.

The following Linux threats are just some of the examples that have been documented by the research community:

- [QNAPCrypt](#) – Ransomware campaign targeting Linux file storage servers. This campaign was later attributed by our researchers to [FullofDeep](#), a Russian cybercrime group.

- ○ [Cloud Snooper](#) – RAT found on Linux servers by researchers at [Sophos](#). The threat was identified on Amazon Web Services EC2 instances and applied by using different tools to bypass security measures. Researchers believe the attack was conducted by an APT due to its toolset and complexity.
- ○ [Winnti](#) – Backdoor tied to the Winnti Umbrella group was discovered on the systems of a German pharmaceutical company named Bayer. Winnti group is a cluster of Chinese government-sponsored activities which contain shared goals and resources including attacking tools. An in-depth research on this malware was conducted by [Chronicle](#). It's the first Winnti Linux variant exposed in the wild.
- ○ [HiddenWasp](#) – RAT targeting Linux servers. It's composed of a rootkit, a trojan, and an initial deployment script. At the time of discovery by our researchers, the malware was undetected despite using code from various open-source projects such as Mirai and the Azazel rootkit. There is evidence that [HiddenWasp](#) is related to a Chinese APT.
- ○ [EvilGnome](#) – Linux desktop backdoor implant with connections to Russia's Gamaredon Group. The malware has many functionalities including file stealing, the ability to capture desktop screenshots, audio recording, and module expansion.
- ○ [Dacls](#) – RAT tied to Lazarus APT group reported by [360 Netlab](#). Researchers found both ELF and PE versions of this malware. This is Lazarus's first exposed Linux malware.
- ○ ManusCrypt – RAT tied to Lazarus group. This malware was reported mainly targeting Windows. Just recently a Linux version of this malware was found, similar to the ManusCrypt variant F PE malware reported by the [US CERT](#) in May 2020.
- ○ [MESSAGETAP](#) – Infostealer discovered by [FireEye](#) on a telecommunications company's Linux servers. These servers operate as a Short Message Service Center (SMSC), which routed SMS messages to recipients. The malware was designed to steal SMS traffic and it was also tied to the Winnti group.

Linux threats are not just established on the risk of a malware taking advantage of a victim's computing resources. They also consist of harmful and intrusive malware that can cause damage to a victim's private domains.

### How does ELF Malware Infect Systems?

Unlike desktops, where phishing is a common method of infection, attackers looking to infiltrate servers and IoT platforms can't rely on end users to install malware on their behalf. There's no user interaction with browsers and email accounts, which makes phishing attacks practically irrelevant in these environments. This means a malware's entry point to the system has to be much more targeted. Here are the main attack vectors used to infect non-desktop Linux machines:

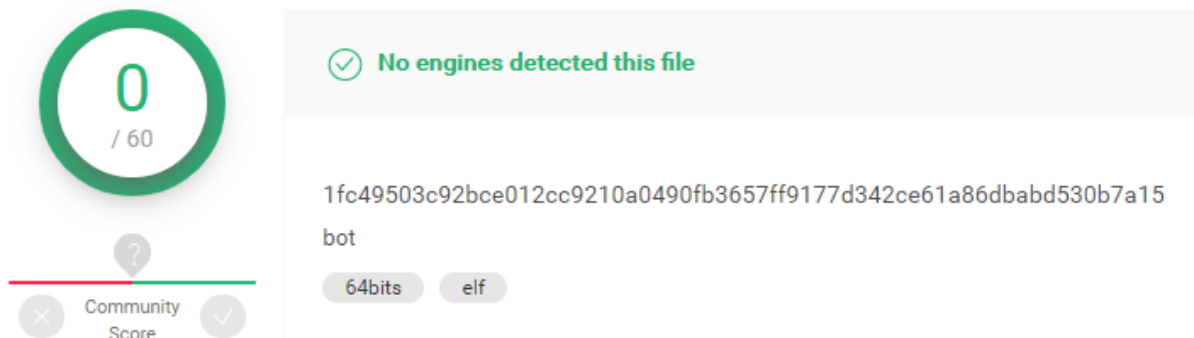
1. **Vulnerability exploit:** attackers will search for exploitable and unpatched publicly faced components in order to access systems. As an example, the attacker behind the [NOTROBIN](#) backdoor exploited CVE-2019-19781, a vulnerability in Citrix NetScaler, to spread the malware. The [Asnarok](#) trojan infection was initiated after the attacker discovered and exploited a zero-day (SQL injection remote code execution) in

Sophos XG firewalls. Misconfigured services can also serve as an entry point for attackers. [Kinsing](#) malware was spread after the attacker took advantage of misconfigured open Docker Daemon API ports.

2. **Use of valid credentials:** default software credentials or compromised credentials. There are different methods in which attackers can steal passwords, including password spraying, credential stuffing, and local discovery. Researchers believe the Cloud Snooper infection was initiated by an attacker accessing the servers through SSH, which is protected with password authentication.
3. **Trusted relationships abuse:** attackers can leverage entry to third party organizations that have direct access to the victim's systems. These organizations may have limited access to the victim's infrastructure in which they maintain but can exist in the same network. For example, an attacker can breach an IT services contractor to then target its clients after gaining valid credentials to these organizations.

### Linux Malware is Off the Radar

It's not only new and sophisticated Linux malware which remain fully undetected by security vendors, but also common ones. Mirai is a prime example. Mirai is a DDoS botnet whose source code was released to the wild and many botnets variants are now based on this code. All that was required for an attacker to bypass detection using this particular [Mirai sample](#) was to make a few signature changes by obfuscating the file's strings.



This sample was uploaded to VirusTotal in March and had zero detections. Since then, we've published a blog post which discusses the [effectiveness of code reuse analysis vs. signature-based detection for detecting this malware](#) and other Linux threats. To this day, the file's VirusTotal report lists only [one detection](#).

When it comes to investigating ELF malware, the current antivirus solutions are not reliable. That's one of the reasons why it's important to add analyzing ELF files to your skillset.

If you want to learn more about why traditional solutions do not detect ELF properly, check out [this webinar](#) profiling the Linux threat landscape.

### The Challenge with ELF File Analysis

So you have a suspicious ELF file that you want to analyze? Where do you start?

The internet is full of information about PE file analysis and there are also various easy-to-use tools and tutorials. However, when searching for information about ELF analysis, one can easily get lost. The shortage of relevant and unified information about analysis methodology, verdict determination, and malware evasion techniques, together with the lack of up-to-date open source tools can be frustrating.

We can list at least six publicly available sandboxes which support Windows PE files. However, currently there is no online sandbox solution available for executing ELF. The few Linux sandboxes out there—[Limon](#), [detux](#), and [LiSa](#)—require creating a sandbox instance and aren't actively maintained. In this series we will present you with relevant ELF analysis tools for performing both [static](#) and dynamic analysis.

## Conclusion

Linux is used broadly and the threat is both real and emerging. Winnti and Lazarus are just a few examples of APT groups that have recently been documented using ELF in their malware toolset. Due to the lack of ELF malware visibility, poor detection from security vendors, and the shortage of relevant publicly available resources about ELF malware, we believe there are many unexposed Linux threats still waiting to be discovered.

Our main goal in initiating this series is to unify a knowledge base and relevant tools for researchers to use when analyzing ELF malware.

## Coming Up

In the next article we will review the ELF format, its static artifacts, and explain how to practically leverage them in your malware analysis together with useful tools.

Here's what you'll need for the next blog:

- Make sure you have a Linux virtual machine.

Once you're set up with your Linux VM, you can read on to Part 1 here:

- [Part 1 – Sections and Segments](#)
- [Part 2 – Symbols](#)
- [Part 3 – Relocations](#)
- [Part 4 – Dynamic Linking](#)

---

Source: <https://intezer.com/blog/linux/elf-malware-analysis-101-linux-threats-no-longer-an-afterthought>