Why can't I clean a hacked machine

from cPanel Technical Support

When a root account is compromised, users often ask how they can “clean” their server. To put it as succinctly as possible: without knowing every action that has ever taken place on a server, it is impossible to prove that the server is completely clean. While it is simple to show a compromised server, showing the opposite, for all intents and purposes, is not.

After a root-level compromise, the only determinations that can be made about the server’s integrity are the following:

1. The server has been hacked.
2. The server may still be hacked.

Once a user gains root access, they can manipulate the server in any way they wish. This means that a hacker can install multiple backdoors, which allow them to regain access to the server. Just because one backdoor is found and removed does not mean that others do not exist. For example, a cron job may run as the `root` user and download a backdoor to the `/bin` directory each day. You may find the backdoor in the /bin directory, but miss the cron job that will allow backdoor access to the server again.

Let’s say that 100,000 root-owned files exist on your Linux server. If three of those files are backdoors that grant root access, how will you know? In addition, many rootkits hide the presence of backdoors. If a rootkit instructs your operating system to hide a file, it is unlikely that you will see the file on the disk. Backdoors can also reside in memory only. Most users do not have the resources necessary to continually audit gigabytes of memory for suspicious activity.

Third-party rootkit hunters

Utilities like rkhunter and chkrootkit can be just as harmful as they are helpful. While they may provide information about known rootkits, they may also create a false sense of trust and security. If rootkit detection performed flawlessly every time, there would be no need for multiple products in order to do so. Remember, these utilities check for known malware only. While they can conduct some heuristics, they can also provide false positives. Most importantly, it is both simple and commonplace for malware developers to evade detection by downloading these utilities and learning how they work.

There will always be unknown malware that has never been and will never be detected. Malware often has variants that operate in many different ways. Without knowing every possible variant, it is impossible to conclusively address the issue.

No official documentation exists for malware because its stealth is how it survives. While independent researchers and antivirus companies provide information about their findings in some cases, no guarantee can be made that the information is entirely accurate or complete. Once that information is released to the public, malware authors may alter their programs to function in a new manner in order to remain undetected.

Solutions for dealing with a compromised server

The only viable solutions for handling a hacked server are the following:

1. Migrate the accounts to a clean server and reinstall the hacked server.
2. Restore the server from a snapshot. However, the server could have been compromised long before the issue was known. If so, this solution may still leave the server compromised.

> **Important:**
> If you believe your server has been compromised, we recommend you contact cPanel Support. If Support determines that your server is compromised, you or your system administrator will need to follow the solutions above to resolve the issue.

Additional documentation

Suggested documentation For cPanel users For WHM users For developers

- How to Purchase a KernelCare License
- How to Purchase an Imunify360 License
- How to Install KernelCare
- Tips to Make Your Server More Secure
- Recommended Security Settings

- How to Install WHMCS

- How to Use cPanel API Tokens
- How to Open a Technical Support Ticket
- Security
- Man-in-the-Middle Attacks

- How to Purchase a KernelCare License
- How to Purchase an Imunify360 License
- How to Install KernelCare
- Tips to Make Your Server More Secure
- Recommended Security Settings

- WHM API 1 Functions - accesshash
- WHM API 1 Functions - setminimumpasswordstrengths
- WHM API 1 Functions - create_user_session
- WHM API 1 Functions - get_tcp4_sockets
- WHM API 1 Functions - get_tcp6_sockets