The stealthiness of Linux/Cdorked: a clarification

welivesecurity.com/2013/05/02/the-stealthiness-of-linuxcdorked-a-clarification/

May 2, 2013

We clarify that the Linux/Cdorked backdoor malware leaves no traces on the hard drive "other than its modified httpd binary" which can be scanned for detection in several ways.

2 May 2013 - 08:01PM

We clarify that the Linux/Cdorked backdoor malware leaves no traces on the hard drive “other than its modified httpd binary” which can be scanned for detection in several ways.

We wanted to clarify that the Linux/Cdorked backdoor malware we reported on last week, and which has been widely reported in the press this week, leaves no traces on the hard drive **other than its modified httpd binary**. We said this explicitly in the original post but some reports have suggested that the malware leaves no trace at all on the hard drive. While that is not the case, we do think “stealthy” is still an appropriate term to use with respect to Linux/Cdorked.

How stealthy is Linux/Cdorked?

Few blog posts on We Live Security have attracted the amount of attention accorded to Linux/Cdorked.A: New Apache backdoor being used in the wild to serve Blackhole which Pierre-Marc Bureau and his team posted last Friday. Their analysis of malware which they dubbed Linux/Cdorked.A, revealed "a sophisticated and stealthy backdoor meant to drive traffic to malicious websites."

After reports of the malware appeared in numerous publications, questions were raised about just how stealthy it was. Pierre-Marc and his team are preparing a follow-up post with additional details about Linux/Cdorked and we think that will address a number of key questions, but in the meantime we wanted to clarify the stealth factor. Here is what the original blog post said:

> The backdoor leaves no traces of compromised hosts on the hard drive other than its modified httpd binary, thereby complicating forensics analysis. All of the information related to the backdoor is stored in shared memory. The configuration is pushed by the attacker through obfuscated HTTP requests that aren't logged in normal Apache logs. This means that no command and control information is stored anywhere on the system.

Now, I think you will agree that's pretty stealthy. Nowhere on the hard drive of the infected server will you find the Linux/Cdorked payload, that is, the malicious code that the bad guys want to run. In the sample we analyzed, the payload–which resides in memory, not on the hard drive–was redirecting web traffic, from the web server infected by Linux/Cdorked, to websites that seek to infect visitors via the Blackhole Exploit kit.

We also noted that the messages used by the bad guys to remotely activate and modify the payload are not logged in normal Apache logs. Furthermore, rebooting the system removes all trace of the payload. However, as stated earlier, we did not say that there is no trace of Linux/Cdorked.A on the server's hard drive. We said there were no traces on the hard drive "**other than its modified httpd binary**". In other words, there is a trace of this infection that can be detected by monitoring changes to binaries on the server, or by scanning the files on the server with an anti-malware program that can spot Linux/Cdorked.A (such a scan, for example with ESET File Security for Linux, will flag the httpd binary).

An open Apache range

Something that emerged in the many valuable comments on Dan Goodin's widely read article about Linux/Cdorked in Ars Technica is the differing levels of expertise and resources applied to administering Apache web servers in different contexts.

On the one hand you have experienced sysadmins who were quick to point out that changes to httpd would be detected by something like Tripwire, software that detects unauthorized changes to files by creating checksums of them in a known good state, then

routinely comparing those checksums to production files. Some folks in this camp can't imagine running an Apache server without such security measures in place (including storing copies of the checksums somewhere other than the server).

On the other hand there are potentially millions of websites running on Apache servers that come preconfigured from hosting providers in ways that seriously complicate the use of such security measures by the server "owners," many of whom have little knowledge of the finer points of managing a Linux server. I would not be surprised to find that this particular subset of Apache servers is being targeted by bad guys who well understand the implications: greater probability of successful and persistent compromise.

Please stay tuned for more coverage.

2 May 2013 - 08:01PM

Newsletter

Discussion