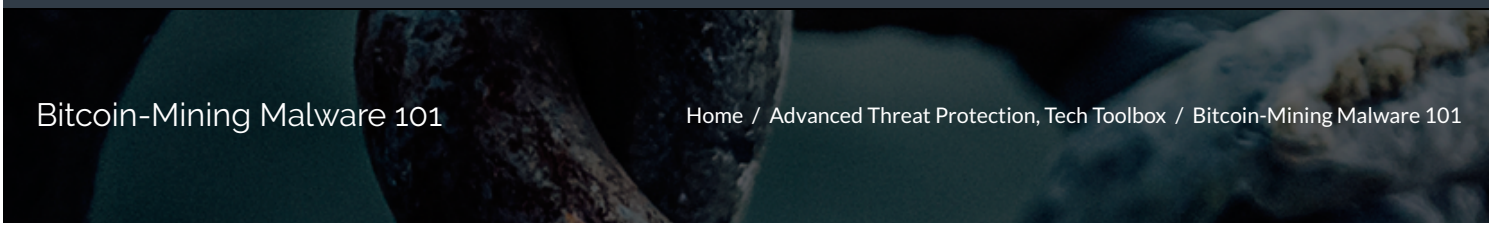

Bitcoin-Mining Malware 101

08
JUL

Bitcoin-Mining Malware 101

July 8, 2014 / Ryan Nolette / Advanced Threat Protection, Tech Toolbox

About the only thing that using Bitcoin-mining malware has in common with real mining is how dirty you should feel while doing it.

With Bitcoin mining, the work isn't about digging a hole in the earth and pulling out the raw materials, as with normal mining. Rather it's about about finding holes in someone's security and digging in to set up a long-term mining operation.

To understand the impact of this type of malware, we first need to understand what the attackers are trying to do and why.

What are Bitcoins and what is Bitcoin mining?

Bitcoins are "a peer-to-peer payment system introduced as open source software." The digital currency created and used in the system is alternatively referred to as a virtual currency, electronic money, or a cryptocurrency.

Bitcoin mining is roughly defined as the processing of transactions in a digital currency system, in which the records of current Bitcoin transactions (blocks) are appended to the record of past transactions (block chain).

With this new knowledge we can look at the process of a Bitcoin mining malware operation. The most common questions I get asked about Bitcoin mining malware are:

"How does Bitcoin malware get into users' systems?"

"Why are they using my machine?"

"How do I know they are there?"

"What can I do to stop them from getting on my systems?"

"What does a Bitcoin mining malware look like on my system?"

Why are they targeting my machine?

Bitcoin mining is a computationally demanding process that gets more and more intense over time. Bitcoin is mined in blocks, and since it takes significant computing power to mine each block, the malicious miners join up and form what are referred to as mining pools/networks. The idea behind this mining pool is that each participant who provides some computing power gets in return their share of the revenue proportional to the amount they participated. In the example of Bitcoin mining malware, the attacker would be the sole beneficiary of the mining efforts instead of a team of willing participants.

How do Bitcoin miners get into users' system?

Bitcoin mining malware uses the same methods as most other malware to gain access to an endpoint. Techniques like malicious downloads, emails with malicious links or attachments, and already-installed malware are the most common and effective methods of delivery.

A well-known example of one of these techniques, the watering hole attack method, was reported on in January 2014. In this example, malicious ads were served to Yahoo! users as they visited a site. The malware downloaded from those malicious ads was designed to transform computers into a Bitcoin

mining operation.

It leveraged known vulnerabilities in Java to install itself on computers that visited the ads.yahoo.com site. The payload downloaded to each successfully exploited computer varied in its contents. Some payloads were just Bitcoin mining malware, while others contained credential-stealing Trojans like Zeus or more common generic remote access tools (RATs).

The malicious ads reportedly lasted from December 31 through January 3, when Yahoo! took them down. "According to security firms, the malware that took advantage of a Java flaw in Yahoo ads infected some 27,000 machines per hour during the four days it was active on the site."

How do I know they are there?

There are a few ways to know a processor-intensive malicious application has been installed on a system. The most obvious to the user is a significantly noticeable performance impact on the infected host. I have read about Bitcoin malware variants that are "user aware" and only run when there are no users logged in or when CPU usage is below a specified threshold.

Another indicator would be copious amounts of network connections made between the infected host and the server. This communication will contain new commands from the server, possibly new malware to install on the host, information on the blocks mined from the host, and other information defined by the attacker.

Finally, there will also be file artifacts. A few artifacts from the sample I analyzed (to be presented in future blog posts) were executables and dll's in the %appdata% directory.

What can I do to stop them from getting on my systems?

Application whitelisting or similar security software in a high-enforcement mode (block unknown) would prevent these unknown and unapproved executables from running and installing. In my next post, I will provide a high-level analysis of the behavior of a sample of Bitcoin mining malware.