Looking at Mutex Objects for Malware Discovery & Indicators of Compromise

By Lenny Zeltser

Published: 2012-07-24 · Archived: 2026-04-05 16:14:48 UTC

Mutex (a.k.a. mutant) objects, which are frequently used by legitimate software, can also help defenders discover the presence of malicious programs on the system. Incident responders can examine the infected host or reverse-engineer malware to identify mutex names used by the specimen, which will allow them to define the signs of the infection (a.k.a. indicators of compromise). Let's take a look at how mutex objects are used and what tools are available to identify them on a system.

How Programs Use Mutex Objects

Programs use mutex ("mutual exclusion") objects as a locking mechanism to serialize access to a resource on the system. Consider the following explanation by Microsoft: "For example, to prevent two threads from writing to shared memory at the same time, each thread waits for ownership of a mutex object before executing the code that accesses the memory. After writing to the shared memory, the thread releases the mutex object."

The Use of Mutex Objects by Malware

Malicious software often uses mutex objects for the same purpose as legitimate software. Furthermore, malware might use a mutex to avoid reinfecting the host. For instance, the specimen might attempt to open a handle to a mutex with a specific name. The specimen might exit if the mutex exists, because the host is already infected.

Consider the renowned Flame malware. According to FireEye, one of this specimen's components created "numerous mutexes in order to synchronize copies of itself simultaneously injected into various core Windows processes (e.g., services.exe, iexplore.exe, winlogon.exe) that are already running." FireEye documented the mutex names whose presence indicated that the system was infected with Flame.

As another example, the Pushdo/Cutwail bot created mutex objects that were used to "coordinate its highly multithreaded communication" according to TrendMicro. The mutex objects names were "gangrenb," "germeonb," "crypt32LogOffPortEvent," etc. As yet another example, the default name of the mutex set by the popular Poison Ivy backdoor is ")!VoqA.I4"; this was the case during a targeted attack against a large Swedish company documented by the Internet Storm Center.

In some cases, malware might dynamically generate mutex names in an attempt to evade detection.

Using Mutex Values to Find Malware

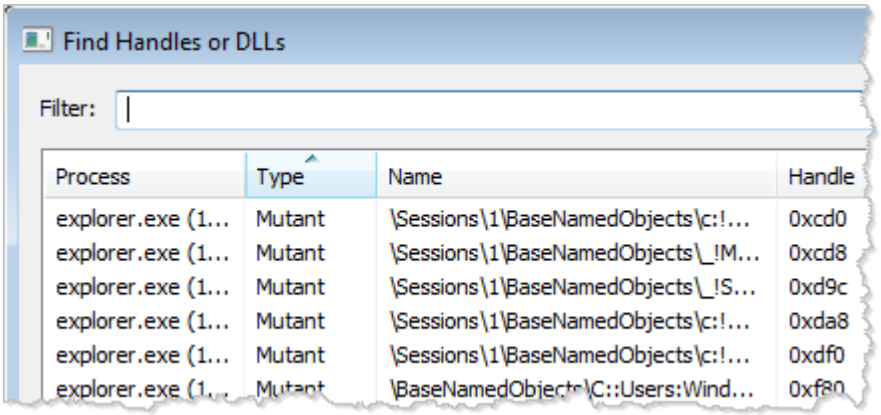
When examining a potentially-infected system, we can look for names of mutex objects known to belong to malicious programs. This approach works particularly well when you've already identified malware on some

enterprise system, determined the names of its mutex objects, and are examining other systems to see whether they are infected. Malware databases such as ThreatExpert include mutex names when describing malware, providing another source of potential signatures based on mutex objects.

Moreover, it's relatively uncommon for legitimate programs to use mutex names that are completely random; you might use this heuristic to identify infected hosts even without searching for a specific mutex names. (See Gary Golomb's post that touches upon this topic.) A command-line tool called CheckMutex can query the local host for the presence of a mutex object with a specific name. The author of CheckMutex, Jaime Blasco, also provides a command-line utility called EnumerateMutex for generating a list of all active mutex objects on the system, you you can examine the list for the names that interest you.

Another way to enumerate all mutex objects from the command line involves Microsoft's Handletool by Mark Russinovich. This utility lists various handle types that are open on the system; to list only mutex objects look for those of type "Mutant" like this:

GUI tools Process Explorer and Process Hacker tools can list open handles on the host, including those that refer to mutex objects. Both tools include an option to search for an open handle or DLL by name. The Performance Monitor tool, built into Windows, also offers these capabilities, as outlined by Mark Baggett. Here's what this feature looks like in Process Hacker:



It is also possible to search for mutex names when examining a memory snapshot of a compromised system. For instance, the popular memory forensics framework Volatility can enumerate mutant values using the "mutantscan" command.

For another potential use of mutex values, consider the possibility of proactively generating mutant objects, so that malware believes it is already active on the host and refuses detection. I discussed this idea in the article Contemplating Malware Immunization via Infection Markers.

As you saw in this article, mutex names can be used for creating indicators of compromise, which would allow incident responders to identify hosts infected with malware that uses those mutex objects. It might also be possible to define heuristics that alert when unusually-random mutex names are discovered on the host, though this approach could produce some false positives. There are several command-line tools to list mutex names, though there is room for maturing this approach to malware discovery.

Lenny Zeltser teaches malware analysis at SANS Institute. At the "day job," Lenny focuses on safeguarding customers' IT operations at NCR Corp. He is active on Twitter and writes a security blog.
