

Ticked Off: Upatre Malware's Simple Anti-analysis Trick to Defeat Sandboxes

unit42.paloaltonetworks.com/ticked-off-upatre-malwares-simple-anti-analysis-trick-to-defeat-sandboxes/

Richard Wartell

October 6, 2015

By [Richard Wartell](#)

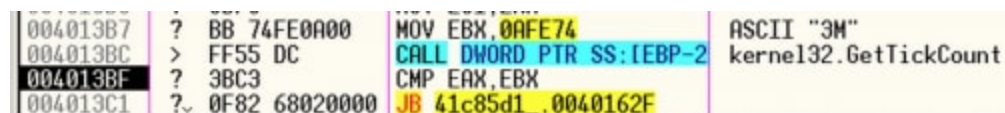
October 6, 2015 at 3:30 PM

Category: [Malware](#), [Unit 42](#)

Tags: [AutoFocus](#), [GetTickCount](#), [Upatre](#), [WildFire](#)

The Upatre family of malware is frequently updated, with the authors adding new features and protecting the malware from detection in various ways. If you aren't yet familiar with Upatre, it's one of the most common downloaders in the wild today, typically infecting systems through phishing e-mails and downloading the Dyre banking Trojan to steal victim's credentials. Recently, the authors of Upatre added a very simple anti-analysis measure in an attempt to defeat sandboxes, which dynamically analyze executables to identify malicious behavior.

The new anti-analysis trick involves using the Windows API [GetTickCount](#). `GetTickCount` returns the number of milliseconds that the system has been alive, up to a maximum of approximately 49 days. Programs can use this value to determine how long a system has been running and make decisions based on that value. The following image shows Upatre executing these instructions inside of a debugger:



```
004013B7 ? BB 74FE0A00 MOV EBX, 0AFE74 ASCII "3M"
004013BC > FF55 DC CALL DWORD PTR SS:IEBP-2 kernel32.GetTickCount
004013BF ? 3BC3 CMP EAX, EBX
004013C1 ? 0F82 68020000 JB 41c85d1, 0040162F
```

The code calls `GetTickCount` and compares the returned value to `0xAFE74` (720,500 milliseconds, or ~12 minutes). If `GetTickCount` returns a value less than `0xAFE74`, Upatre determines that the system has been running for less than 12 minutes and exits.

To understand why this is an effective anti-analysis technique, we have to look at the normal startup procedure for a sandbox:

1. Start up a virtual machine with the target operating system on it
2. Copy the malicious binary to the virtual machine
3. Let the malware run for an extended period of time (usually about 5 minutes)

4. Retrieve a report of what the malware did from the VM
5. Shut down the virtual machine and clean up

Since this whole process usually takes around 6 minutes at most, GetTickCount typically returns a value much lower than 12 minutes. If this was an actual user's system the value would almost certainly be larger than 12 minutes, so a lower value gives the malware a clear indication that it is executing in a sandbox. By exiting at this point in the execution flow, the malware doesn't perform any malicious actions and will be flagged as benign in a sandbox.

In the Palo Alto Networks [WildFire](#) analysis system, we modify the value returned by GetTickCount to make it appear as though the machine has been running for hours. By doing this, Upatre is fooled into continuing to execute its malicious routine, resulting in a malicious verdict.

We recently detected a surge of new Upatre malware samples exhibiting this behavior and Brendan Griffin at Malcovery [published a report](#) on the activity recently. Palo Alto Networks customers using WildFire are fully protected from this anti-analysis technique and Palo Alto Networks [AutoFocus](#) users can find new Upatre samples using the [Upatre](#) tag.

**Get updates from
Palo Alto
Networks!**

Sign up to receive the latest news, cyber threat intelligence and research from us

By submitting this form, you agree to our [Terms of Use](#) and acknowledge our [Privacy Statement](#).