Song Yang (sy540)

Xin Yang (xy213)

Zhuohang Li (zl299)

Paper Review

WindowGuard: Systematic Protection of GUI Security in Android

**Problem statement**

The number of malware that are using graphic user interface (GUI) systems to attack Android devices is increasing rapidly. Android GUI attack refers to harmful behavior that attempts to adversely affect the integrity and availability of GUIs belong to other applications in order to achieve malicious purposes, including launching unwanted windows, tapjacking, taskjacking, etc.

**Motivation**

Android GUI attacks are real threats and can cause severe consequences, such as sensitive user information leak, user device denial of service, etc. Given the serious and rapid growth of GUI attacks, there is a pressing need for a comprehensive defense solution. However existing defense methods are all lack of defense coverage, effectiveness, and practicality.

**Solution approach**

Android window integrity (AWI) model clearly designates user session and continuously checks for window integrity to prevent the attacker from taking over the user’s screen. The AWI model is implemented in WindowGuard which is developed as a Xposed module to be usable to a larger number of users.

**Pros and Cons of the paper**

* **Pros:**

1. It is very difficult for previous static-analysis-based defense solution to distinguish from the usage of the same API from a malware or from an app that is useful to the user. However by enforcing clear designation of user session and legitimacy of GUI system, AWI model is able to make normal user session less vulnerable, and therefore detect malware with a much higher while preserving the original user experience.
2. The AWI model provides a comprehensive protection for the user. It works for all known GUI attacks, including malicious screen locker, GUI confusion attacks, tapjacking attacks, etc.
3. WindowGuard, the implementation of AWI model, compared with existing solutions which require modification on both apps and systems, is more practical and easier to deploy on a massive number of user devices.
4. Once deployed, WindowGuard can work systematically in the background and do not require user involvement until a malicious behavior is detected. WindowGuard does not have any impact on most apps. As for the 1.03% of apps that trigger security alert, only a single tap is needed from the user to make a decision for that app and will not distract the user from that on. Compared with existing solutions which continuously require user’s attention, this method does not bring much better user experience but is also bound to deliver higher accuracy instead of relying on user’s experience to make judgments.

* **Cons:**

1. It’s based on Xposed framework, so the device has to be rooted first, which gives full permission to all apps, and exposes the system to a more dangerous environment.
2. For new Android devices, warranties will be invalid after rooting and installing the Xposed framework.
3. Also, this paper failed to clarify how many malicious attacks can happen even when the device is not rooted. Referring to the consequences of rooting, there should be a lot of attacks that can be avoided if they are not given enough permission. It’s better to detect the malicious behaviors without rooting your device and develop another version for rooted users.
4. This paper is tested on Google Nexus devices, which is a pure Android system, and the whitelist includes several system apps and works fine. But when it comes to third-party devices e.g. Samsung or Sony, more manufactures pre-installed apps need to be added to the whitelist like Samsung Pay or Xperia Home launcher, which needs much more efforts for the long-term maintenance. For the system apps, they should extract some common features so that devices from other manufacturers can easily adapt to.
5. WindowGuard only steps in when malicious attack affects the integrity and accessibility of the GUI of other applications. Therefore it will not work on spoofing malware that misleads user to launch itself. On the other hand, WindowGuard also places demands on app developers to follow AWI security principles. Otherwise, the functionality of such apps could be affected on deployed devices.

**Potential Improvement Point**

Since WindowGuard still requires judgment from users, we think it could be of practical use to interfacing WindowGuard with malware analysis websites. For example, a single prompt of the rate of an app marked as malware by the community can largely help users to make the decision. Conversely, information provided by the user can help to increase malware detection rate on a large scale of devices.

Song Yang (sy540)

Xin Yang (xy213)

Zhuohang Li (zl299)

Malware Review

Monero Mining Malware embeded in KMSpico v10.2.2

**Introduction**

KMSpico is a well-known Windows operating system activation tool, but the latest version has been embedded with a Monero miner in some unofficial sites.

Installing KMSpico would extract four files under the directory “C:\Program Files\KMSPico 10.2.2 Final”, a batch file named INSTALL\_KMS.bat would run the activation program and the win32.exe file. Win32.exe is exactly the Monero mining malware, which utilizes CPU resources for heavy hash computing tasks.

Comparing with the most famous Bitcoin, the speed of mining Monero using CPU is still acceptable for some high-end home PCs, also the Monero mining is completely anonymous thanks to the algorithms behind, which raises the interests of many malware developers.

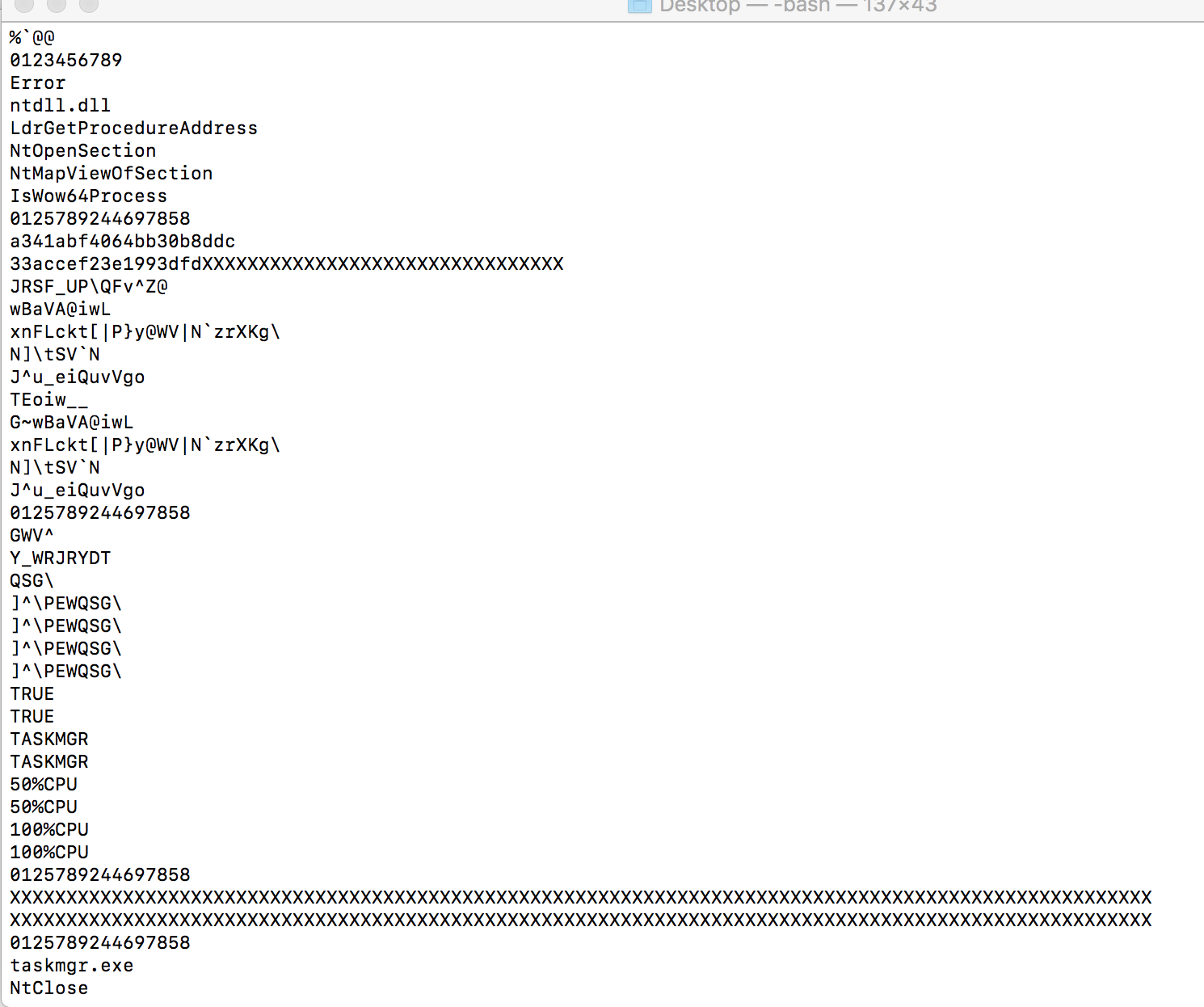
**Reverse Engineering**

* **Static Analysis**

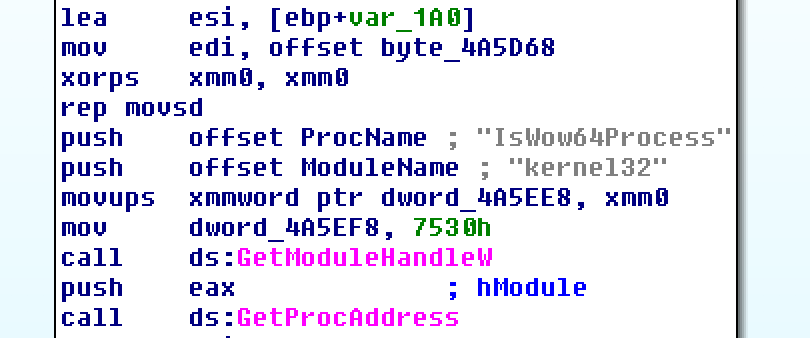
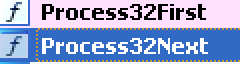
Uploading the win32.exe to VirusTotal, 57/67 engines report it as a malware, tags appeared mostly are shown as Trojan, CoinMiner, MoneroMiner, etc.. At the behavior page, we can see that under the Processes Created section, a fake svchost.exe process is created with the arguments standing for a Monero mining pool and Monero wallet email address.



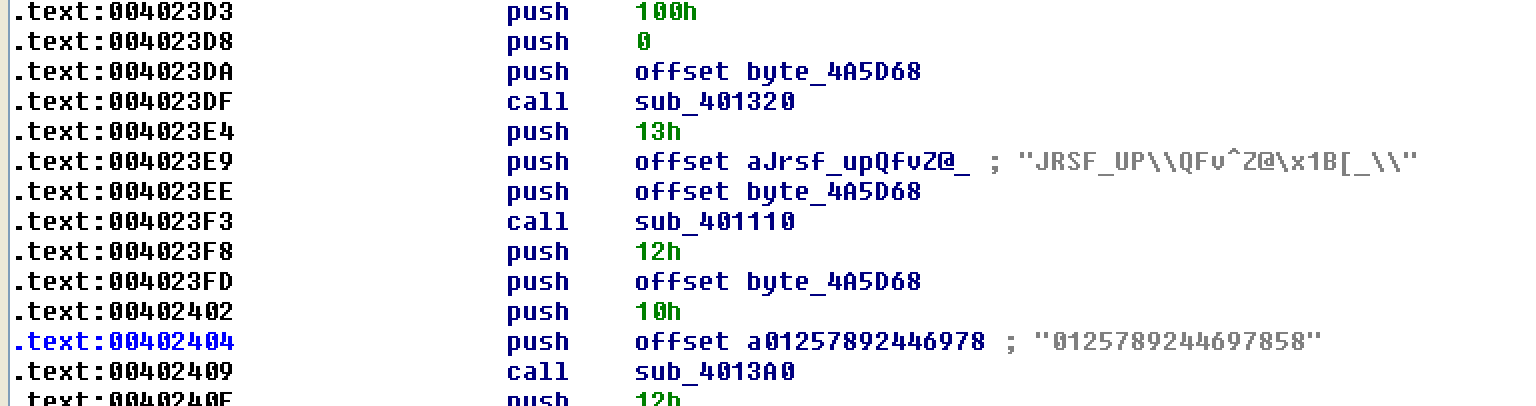
From DependencyWalker, PEID, and PEView, we couldn’t find much significant information. But when we run strings command, lots of hints infer this is a miner malware. As can be seen from the screenshots below, “50%CPU” and “100%CPU” infer the CPU usage. Such an abnormally high rate definitely links it to a miner tool considering this is an activation tool. Also, we can see there are words like “TASKMGR” and “taskmgr.exe”, which might mean that this malware can detect the existence of task manager, and take some operations accordingly. Most of the rest parts are gibberish.



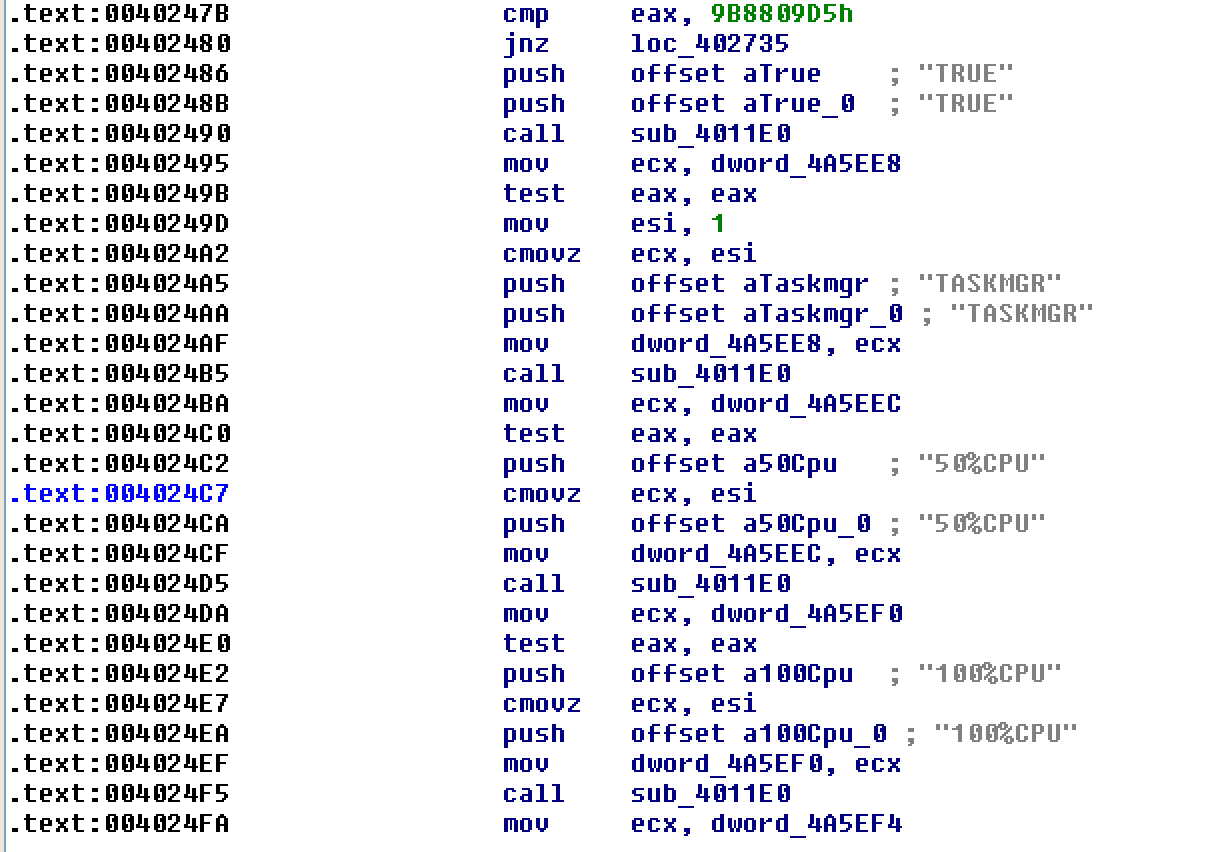
From IDA Pro, we can find the functions Process32First, Process32Next and Windows Version related operations e.g. “IsWow64Process”, which determines the current version of victim machine and take operations differently.



From the screenshots below, we can see the gibberish looking string and a number string, which might infer to an encrypted Monero wallet and the encryption password.

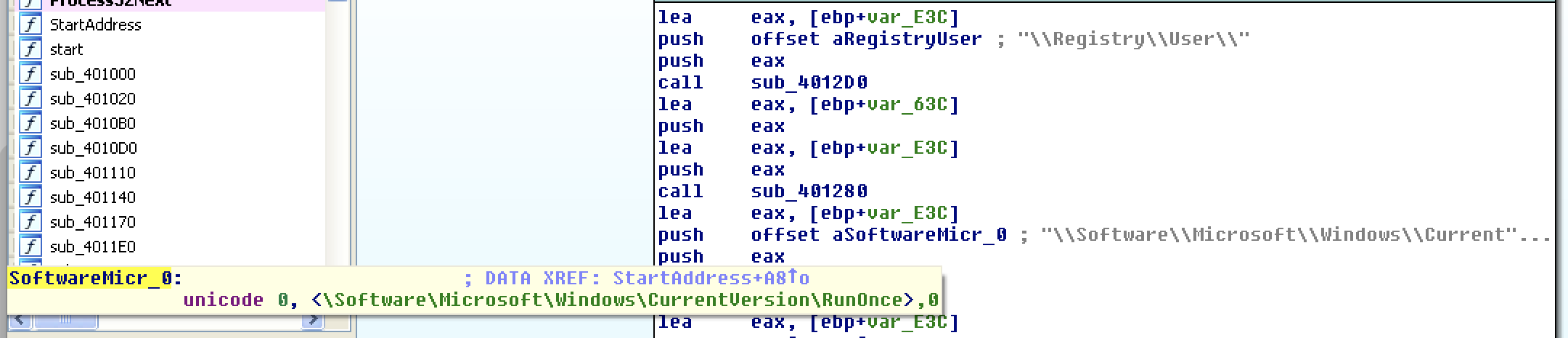


This malware will detect the windows task manager in a loop and pause the miner if user aware something is wrong. A report of current CPU usage when it comes to 50% and 100% can also be found.



We can also find the registry operations to the RunOnce record, which can make this malware auto launching each time windows is started.

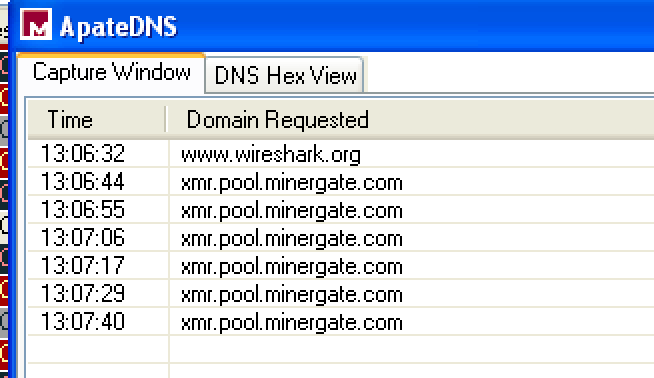




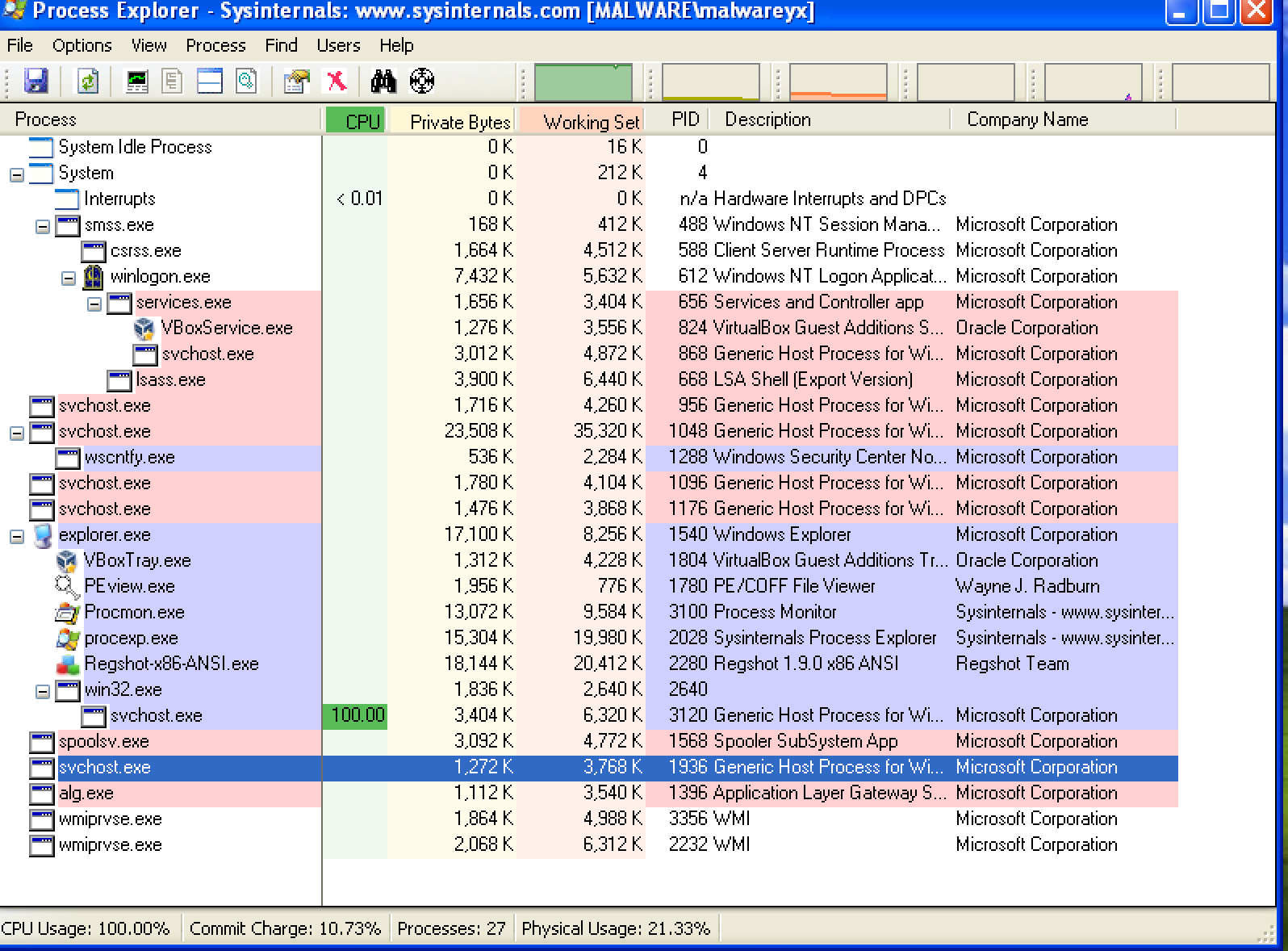
* **Dynamic Analysis**

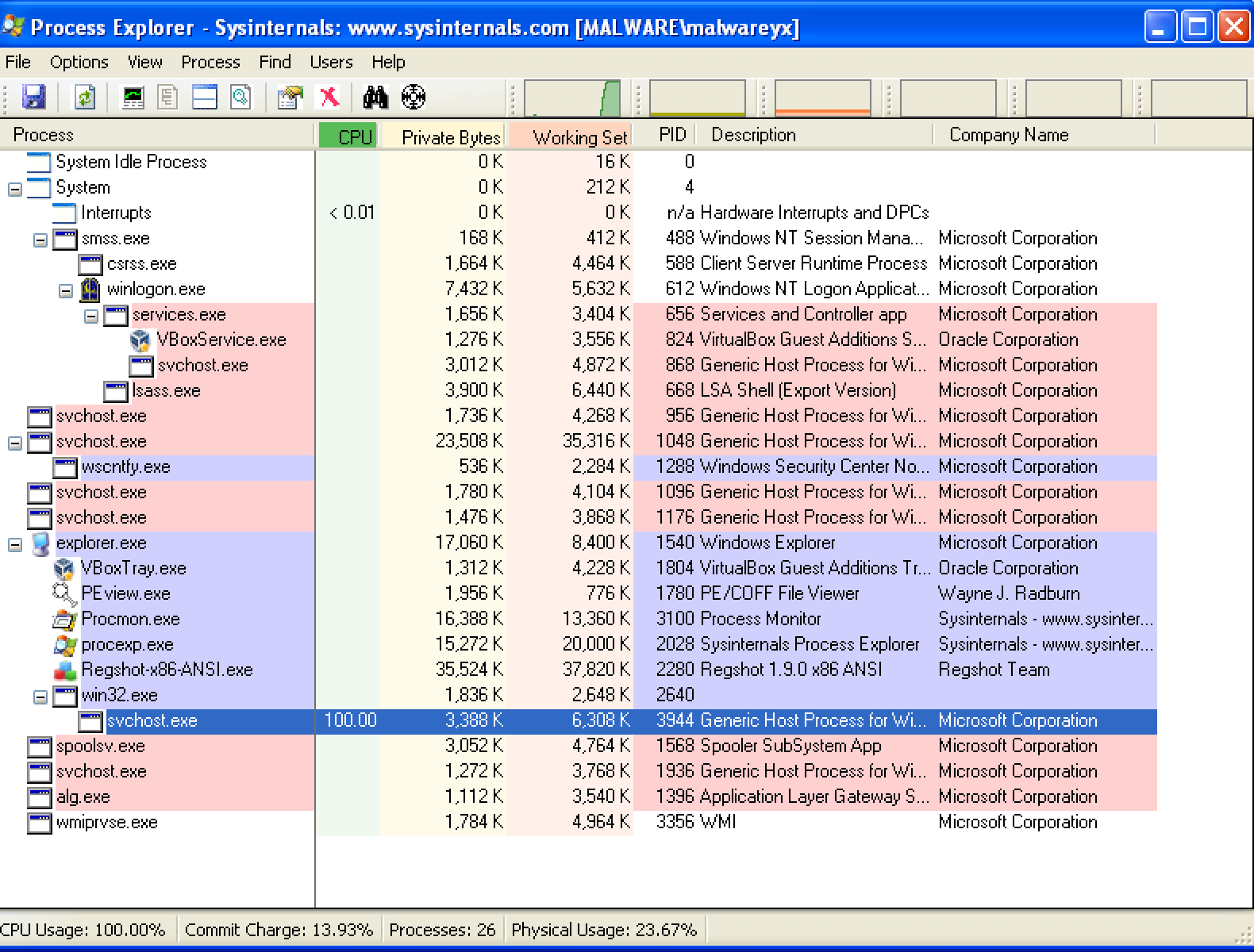
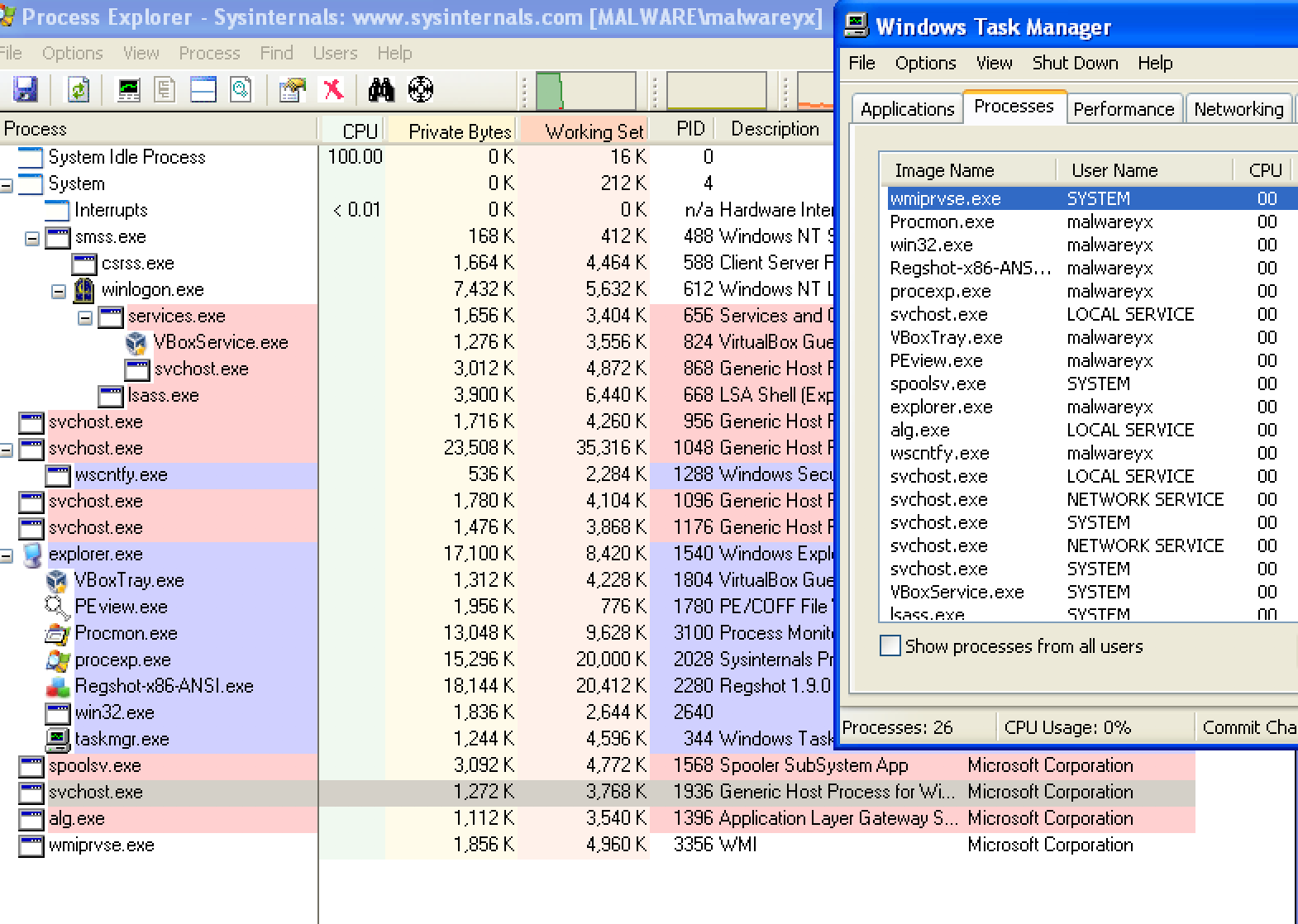
First, we set the ApateDNS and opened Wireshark, and prepare the Regshot, process monitor and process explorer.

Once launched win32.exe, we can see this malware tries to connect with xmr.pool.minergate.com. As known to all, “xmr” refers to Monero and this is a Monero mining pool address. But from Wireshark we couldn’t get any valuable information.



In Process Explorer, we can see there’s a new process under the win32.exe and takes 100% of CPU, so this is the Monero mining process.



We opened the Windows task manager to see if this malware will react to taskmgr.exe. And from the below screenshot, the CPU rate falls to a low level after task manager is launched, meanwhile the svchost.exe is killed. Then we closed the task manager, the svchost.exe was created again and took full advantage of the CPU resources.

From the Regshot comparison, we can see this malware added a record in the “RunOnce” to achieve auto-running when Windows is started. And the directory is under “C:\Documents and Settings\[User]\Local Settings\Application Data\kAUNCUkNWH\win32.exe”, which means this malware copied itself to another directory to prevent user delete the installation files. The massive registry operations and file operations from the Process Monitor also proves this.



**Intrusion detection**

If infected by this malware, users would feel the heat caused by massive CPU computing, and by launching Process Monitor (not task manager), users can see an svchost.exe under the win32.exe that takes 100% of CPU usage. Meanwhile, under the directory “C:\Documents and Settings\[User]\Local Settings\Application Data\kAUNCUkNWH\” will exist a win32.exe file, as well under the “C:\Program Files\KMSPico 10.2.2 Final\”. There will also be a win32.exe registered inside the “\Microsoft\Windows\CurrentVersion\RunOnce”.

**Intrusion recovery**

Process Monitor can find the win32.exe and the svchost.exe processes which consume 100% CPU, killing the processes should stop the malware from mining. Then users can delete the win32.exe from the two directories mentioned above, and also remember to delete the KMSpico related directories completely, then restart the computer and it shall be fine.