RansomWhere?

download

Let's try to generically thwart OS X ransomware via math!

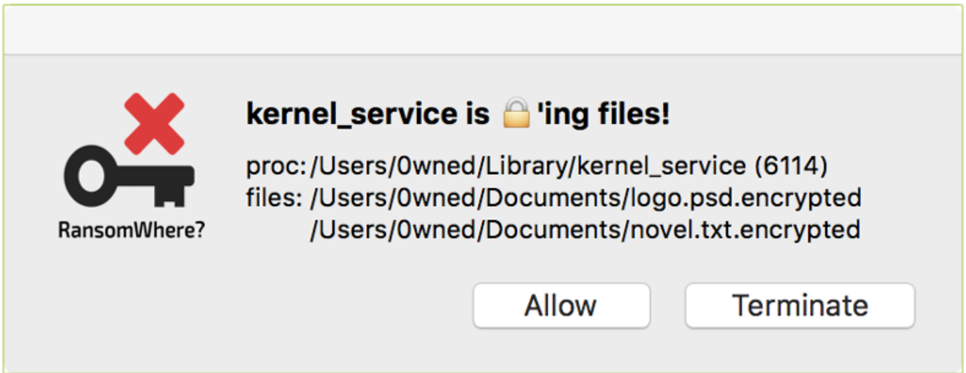
By continually monitoring the file-system for the creation of encrypted files by suspicious processes, RansomWhere? aims to protect your personal files, generically stopping ransomware in its tracks.

Supported OS: OS X 10.8+

Current version: 1.2.5 (**change log**)

Zips's SHA-1: AA566400D8933A463BA8C045BFFAB59499AB8D72

Source Code: **RansomWhere?**



kernel_service is 🔒'ing files!

proc: /Users/Owned/Library/kernel_service (6114)
files: /Users/Owned/Documents/logo.psd.encrypted
/Users/Owned/Documents/novel.txt.encrypted

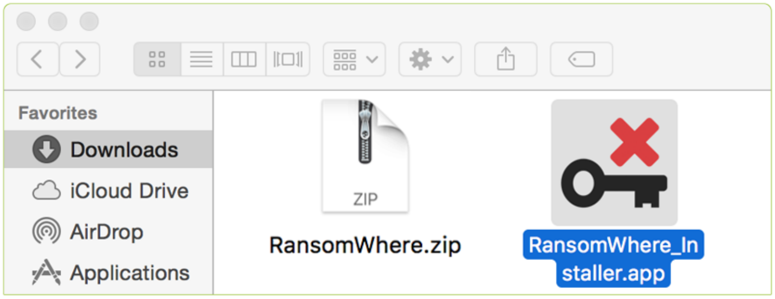
Interested in the background research and design of this tool? See the blog post; **'Towards Generic Ransomware Detection?'**

Also, as with any security tool, direct or proactive attempts to specifically bypass RansomWhere?'s protections will likely succeed. A concerted effort has been made to fully transparent about this, and to articulate the limitations of this tool. See the **'limitations'** section below for more details.

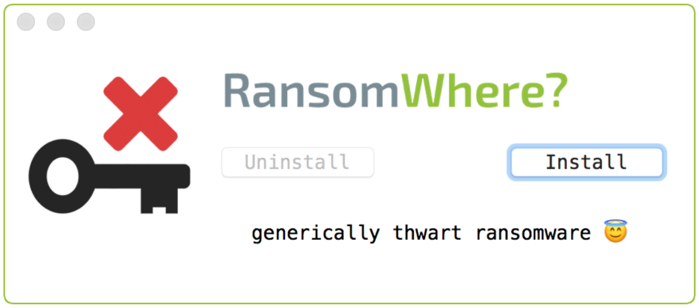
RansomWhere? is a utility with a simple goal; generically thwart OS X ransomware. It does so by identifying a commonality of essentially all ransomware; the creation of encrypted files. Generally speaking, ransomware encrypts personal files on your computer, then demands payment (the ransom) in order for you to decrypt your files. If you fail to pay up, and don't have backups of your files, they may be lost forever - that sucks!

This tool attempts to generically prevent this, by detecting untrusted processes that are encrypting your personal files. Once such a process is detected, RansomWhere? will stop the process in its tracks and present an alert to the user. If this suspected ransomware, is indeed malicious, the user can terminate the process. On the other hand, if its simply a false positive, the user can allow the process to continue executing.

To install RansomWhere? and gain continual protection, first **download** the zip archive containing the application. Depending on your browser, you may need to manually unzip the application by double-clicking on the zipped archive:



Then, simply double-click on `'RansomWhere_Installer.app'` and enter your password to authenticate. Click 'Install' to install the tool:



ℹ️ As malware may encrypt files at any location, RansomWhere? needs full disk access (FDA).

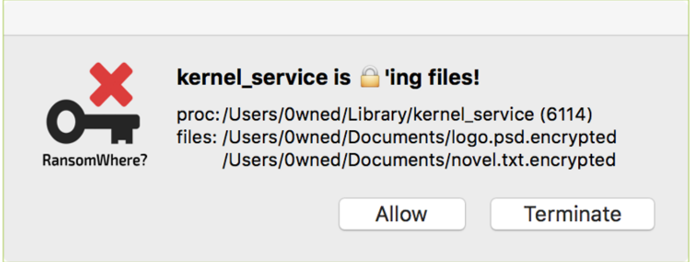
You may have to manually grant RansomWhere? such acccess via "`System Settings`", under "`Privacy`

Now, there are also other ways to install RansomWhere? that may be more conducive to automated or managed installations. First, the RansomWhere? installer app can be executed directly from the commandline (i.e. non-UI), with the `-install` or `-uninstall` flag:

```
//install
$ sudo RansomWhere_Installer.app/Contents/MacOS/RansomWhere -install
RANSOMWHERE?: install ok!

//uninstall
$ sudo RansomWhere_Installer.app/Contents/MacOS/RansomWhere -uninstall
RANSOMWHERE?: uninstall ok!
```

Once installed, RansomWhere? will attempt to block any untrusted processes that are detected quickly creating encrypted files (a la ransomware). Specifically it will suspend the suspect process and alert the user. For example; here's the alert for the OS X ransomware **KeRanger**:



kernel_service is 🔒'ing files!

proc: /Users/0wned/Library/kernel_service (6114)
files: /Users/0wned/Documents/logo.psd.encrypted
/Users/0wned/Documents/novel.txt.encrypted

As RansomWhere? attempts to generically prevent ransomware encryptions purely thru heuristics, its important to understand such alerts. Why? Well it's possible (though unlikely) that RansomWhere? has simply detected a legitimate application or binary that is not ransomware (for example, a legitimate encryption tool you are running to secure various sensitive files).

Alerts shown by RansomWhere? contain two important pieces of information; the process that RansomWhere? has suspended (until one allows or terminates it), and the list of encrypted files that the process has created. If you trust the process, or the files created by the process are legitimate, click 'allow' to allow the program to continue executing in an unabated manner. On the other hand, if you don't recognize that process or the files it is creating, click 'terminate' to kill it.

The following list summarizes the 'allow' and 'terminate' actions

- 'allow'
  Tells RansomWhere? it's ok to let the process continue running. This will be persistently remembered; you'll never be alerted about this binary again.

- 'terminate'
  Tells RansomWhere? to kill the process. As this action is a little more drastic, RansomWhere?, (by design) will not remember such actions. Thus if the terminated process is ran again, it will cause another alert.

It is important to understand how RansomWhere? determines what (it thinks) is ransomware -as this can also help understand its alerts and how to effectively respond to them. Essentially RansomWhere? must decide that the answer is 'yes' to the following two questions in order to display an alert:

1. Is the the process trusted?
   RansomWhere? trusts processes that are signed by Apple proper, from the official Mac App Store (and sandboxed), or where already present when the tool was installed, or have be explicitly approved by the user (i.e. you clicked 'allow' in a previous alert). Also, some 3rd-party applications (such as password managers) that legitimately create encrypted files have been explicitly whitelisted.

2. Is the process quickly creating encrypted files?
   RansomWhere? uses mathematical calculations to determine if a created/modified file is encrypted. If an untrusted process creates several of these quickly, RansomWhere? will generate an alert.

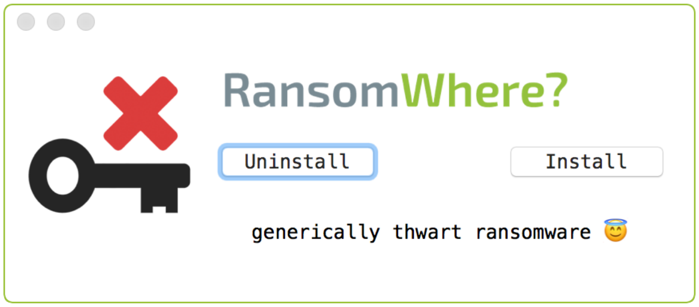
For technical details on how RansomWhere? answers the above questions (e.g. the mathematical constructs used to determine if something is encrypted), see the blog post; **'Towards Generic Ransomware Detection?'**.

As previously mentioned, RansomWhere? will trust most (all?) binaries that were already present when it was installed, or have be explicitly approved by the user. However, one can 'reset' RansomWhere? which will clear its 'memory' of these trusted applications:

```
//reset RansomWhere?
$ sudo /Library/Objective-See/RansomWhere/RansomWhere -reset
```

```
    b) stopped, then (re)started the launch daemon
```

To uninstall RansomWhere? simply re-run the `'RansomWhere_Installer.app'`. Clicking the 'Uninstall' button will stop RansomWhere? and completely remove it from your system:



One can also manually uninstall RansomWhere? via the following commands:

```
//stop launch daemon
$ sudo launchctl unload /Library/LaunchDaemons/com.objective-see.ransomwhere.plist

//remove launch daemon, its plist, and supporting files
$ sudo rm -rf /Library/LaunchDaemons/com.objective-see.ransomwhere.plist

$ sudo rm -rf /Library/Objective-See/RansomWhere
```

Limitations

As with any security tool, it is important to understand the tool's limitations. RansomWhere? was designed to generically stop OS X ransomware. However several design choices were consciously made (to facilitate reliability, simplicity, and speed) that may impact its protection capabilities. First, it is important to understand that the protections afforded by any security tool, if specifically targeted, can be bypassed. That is to say, if new piece of OS X ransomware was designed to specifically bypass RansomWhere? it would likely succeed.

Other limitations include:

- Reactivity
  RansomWhere? detects and blocks ransomware by detecting untrusted processes that are rapidly creating encrypted files. This is inherently reactive; and as such, the ransomware will likely encrypt a few files (ideally only two or three), before being detected and blocked.

- Trust
  RansomWhere? explicitly trusts binaries signed by Apple proper (though not ones signed with an Apple developer ID). As such, if ransomware abuses an signed Apple binary (or process, perhaps via injection), RansomWhere? may not detect this. Moreover, the tool inherently trusts most applications that are already present on the system when it is installed. Thus is ransomware is already present on the system (before RansomWhere? is installed), it may not be detected.

- False-positive reductions
  RansomWhere? attempts to reduce false positives, specifically in terms of misclassifying encrypted files. For example, it tries to filter out files and file types that may be legitimately encrypted, or appear to be encrypted (when in reality they are not). If ransomware was aware of this fact, it could in theory abuse this to avoid detection.

Again, see the aforementioned **blog post** for more information of the technical details on these limitations.

Components/Capabilities/Footprints

The following table briefly summarizes RansomWhere?'s components, capabilities, and system footprint:

Executable Component	Capability	System Footprint/Impact
`RansomWhere_Installer.app`	Installs or uninstalls RansomWhere?	Install: a) creates the `/Library/Objective-See/RansomWhere` directory and copies in various components (such as the RansomWhere daemon) b) creates `/Library/LaunchDaemons/com.objective-see.ransomwhere.plist` c) starts RansomWhere? daemon Uninstall: a) stops RansomWhere? daemon b) removes `/Library/LaunchDaemons/com.objective-see.ransomwhere.plist` c) removes the `/Library/Objective-See/RansomWhere` directory

`/Library/Objective-See/RansomWhere/→RansomWhere`	The persistent RansomWhere? daemon, which continually monitors the system for ransomware	As such, once initialized, it utilizes a small percentage (<0.5%) of the CPU. It also may create several files in the `/Library/Objective-See/RansomWhere/` directory such as: a) `/Library/Objective-See/RansomWhere/installedApps.plist`, a list of applications already present on the system. b) `/Library/Objective-See/RansomWhere/approvedBinaries.plist`, a list of binaries, explicitly approved by the user. c) `/Library/Objective-See/RansomWhere/whitelist.plist`, a list of safe binaries (that often legitimately create encrypted files). d) `/Library/Objective-See/RansomWhere/graylist.plist`, a list of system binaries, that are not explicitly trusted.

In terms of networking code, each time RansomWhere? starts, it queries `https://objective-see.com/products/versions/ransomwhere.json` to see if there is a new version of the tool. Other than this simple version check, it contains no other networking capabilities.

FAQs

Q: Why does RansomWhere? need my password?
A: In order to continually monitor the file-system for encrypted files, RansomWhere? requires system privileges. As such, the tool requests a password (via a standard authorization prompt) during installation/uninstallation.

Q: How can I tell if RansomWhere? is installed and running?
A: By design, RansomWhere? keeps a low profile and chooses not to clutter up the UI. To check if it's running, simply use `Activity Monitor.app`, select `View->All Processes`, and look for a running process named 'RansomWhere'

Q: Why is RansomWhere? using so much of the CPU?
A: By design, RansomWhere? strives to use as few system resources as possible. However, when RansomWhere? is installed for the first time the tool will crunch away for a few minutes (processing running processes, installed/baselined applications, etc.). This shouldn't last to long, and soon RansomWhere? will move into 'monitoring' mode, which consumes barely any CPU (~0.2%).

Q: RansomWhere? displayed an alert. It this ransomware?
A: In order to generically detect and thus thwart OS X ransomware, RansomWhere? alerts on processes it does not recognize that are quickly creating encrypted files. As such, its possible that the flagged process is not ransomware. Closely examine the alert; if you recognize (and trust!) the flagged process and files it is creating, simply click 'allow' to tell the tool to trust (and thus ignore) the process from now on out.

Q: Why does RansomWhere? access the network?
A: When RansomWhere? is started, it connects to Objective-See.com (specifically `https://objective-see.com/products/versions/`) to check if there is a new version of the product. It reads the file `ransomwhere.json`, which contains the latest version number of RansomWhere. No information is collected or transmitted and, other than version checks, RansomWhere? has no other networking code, nor makes any other network connections.

Q: I installed RansomWhere? and (knowingly) ran some ransomware to test it. Why didn't it detect the ransomware?
A: When RansomWhere? is first installed, it enumerates all installed applications in order to 'baseline' the system. This process can take a few minutes and until its complete, RansomWhere? is not monitoring the system. Also, if the ransomware is dormant, or not encrypting files, RansomWhere? will not flag it (until the ransomware is 'activated' and starts encrypting). All this, and much more is explained in see the blog post; **'Towards Generic Ransomware Detection?'**.

Q: Any other questions?
A: Feel free to shoot me an email at **patrick@objective-see.com**.