Another OSX.Dok dropper found installing new backdoor

blog.malwarebytes.com/threat-analysis/2017/05/another-osx-dok-dropper-found-installing-new-backdoor/

Thomas Reed

May 1, 2017

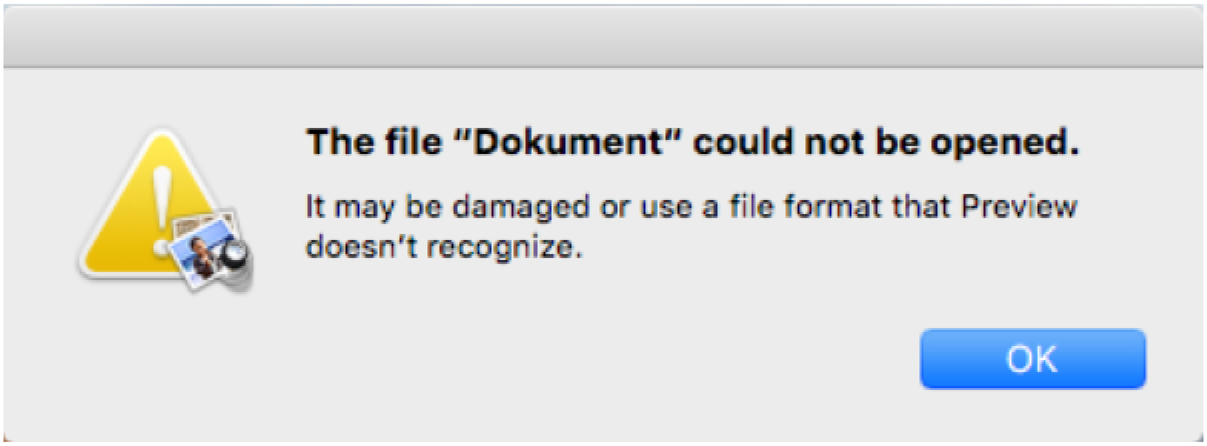
On Friday a sophisticated Mac Trojan was discovered, called OSX.Dok, which installs malware designed to intercept all HTTP and HTTPS traffic. This morning, Adam Thomas, a Malwarebytes researcher, found a variant of the OSX.Dok dropper that behaves altogether differently and installs a completely different payload.

Distribution method

This variant has the same form as the dropper for OSX.Dok – a zipped app named Dokument.app, masquerading as a document. It is signed with the same (now revoked) certificate as the previous OSX.Dok dropper and it was first uploaded to VirusTotal around the same time.

```
OSX.Dok.B SHA-256:
54ee71f6ad1f91a6f162bd5712d1a2e3d3111c352a0f52db630dcb4638101938
```

As with the previous variant, this one also copies itself to */Users/Shared/AppStore.app*, and displays the same alert claiming that the app is damaged:



The file "Dokument" could not be opened.

It may be damaged or use a file format that Preview doesn't recognize.

However, this variant never displays the fake “OS X Updates Available” window, covering the entire screen. After a minute or so, it simply closes and deletes itself.

Instead of installing OSX.Dok, this dropper installs an open-source backdoor named Bella, created by someone who identifies himself on GitHub only as “Noah.”

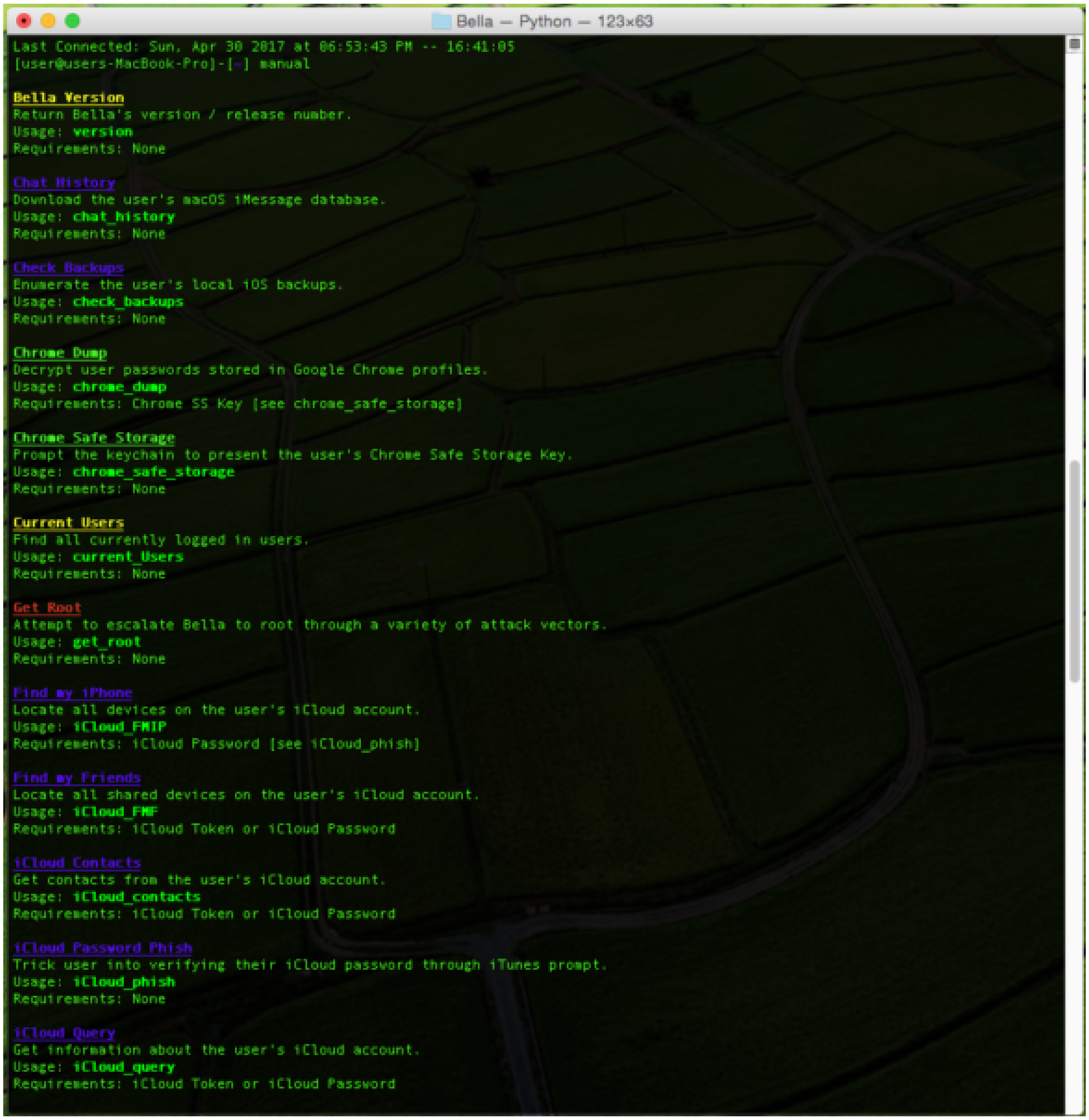
Behavior analysis

Noah first joined GitHub back in 2015 but was not active there until August of 2016, when he began creating Python scripts to attack various macOS data, such as stealing iCloud authorization tokens, or password and credit card information from Chrome.

In February of this year, he published the code for Bella, a Python script with some frightening capabilities, including:

- Exfiltration of iMessage and SMS chat transcripts
- Location of devices via Find My iPhone and Find My Friends
- Phishing of passwords
- Exfiltration of the keychain
- Capture of data from the microphone and webcam
- Creation and exfiltration of screenshots
- Remote shell and screen sharing

Bella even includes the capability to escalate to root privileges via vulnerabilities in the system (which only work on macOS 10.12.1 and earlier) or phishing to obtain an admin user password. Some of the above capabilities rely on gaining root privileges, while others do not.



```
Last Connected: Sun, Apr 30 2017 at 06:53:43 PM -- 16:41:05
[user@users-MacBook-Pro]-[~] manual

Bella Version
Return Bella's version / release number.
Usage: version
Requirements: None

Chat History
Download the user's macOS iMessage database.
Usage: chat_history
Requirements: None

Check Backups
Enumerate the user's local iOS backups.
Usage: check_backups
Requirements: None

Chrome Dump
Decrypt user passwords stored in Google Chrome profiles.
Usage: chrome_dump
Requirements: Chrome SS Key [see chrome_safe_storage]

Chrome Safe Storage
Prompt the keychain to present the user's Chrome Safe Storage Key.
Usage: chrome_safe_storage
Requirements: None

Current Users
Find all currently logged in users.
Usage: current_Users
Requirements: None

Get Root
Attempt to escalate Bella to root through a variety of attack vectors.
Usage: get_root
Requirements: None

Find my iPhone
Locate all devices on the user's iCloud account.
Usage: iCloud_FMIP
Requirements: iCloud Password [see iCloud_phish]

Find my Friends
Locate all shared devices on the user's iCloud account.
Usage: iCloud_FMF
Requirements: iCloud Token or iCloud Password

iCloud Contacts
Get contacts from the user's iCloud account.
Usage: iCloud_contacts
Requirements: iCloud Token or iCloud Password

iCloud Password Phish
Trick user into verifying their iCloud password through iTunes prompt.
Usage: iCloud_phish
Requirements: None

iCloud Query
Get information about the user's iCloud account.
Usage: iCloud_query
Requirements: iCloud Token or iCloud Password
```

Bella comes with a script named BUILDER that can be used to customize some aspects of its behavior. This particular copy of Bella has been configured to connect to the following C&C server:

```
host = '185.68.93.74' #Command and Control IP (listener will run on)
port = 4545 #What port Bella will operate over
```

This address is owned by a hosting company located in Moscow, Russia.

The malware has also been set to install the script, database, and launch agent files in the following locations:

```
~/Library/Containers/.bella/Bella
~/Library/Containers/.bella/bella.db
~/Library/LaunchAgents/com.apple.iTunes.plist
```

If root access can be achieved, it will instead be placed in the corresponding locations in the root Library folder, rather than the user's Library folder.

Conclusion

Of course, since the code signing certificate on the Dokument.app dropper for this malware has been revoked, no one can be newly-infected by this particular variant of this malware at this point. However, since Bella is open-source and surprisingly powerful for a Python script, it's quite likely it will be dropped by other malicious installers in the future.

It is unknown whether there is any connection between Noah, the author of Bella, and the creators of the OSX.Dok malware. Bella may simply have been used by unrelated hackers since it is freely available as open-source software.

Malwarebytes for Mac detects this malware as OSX.Bella. If you've been infected with this malware, after removing it, be sure to change all your passwords as well.

Business users should be aware that this malware could exfiltrate a large amount of company data, including passwords, code signing certificates, hardware locations and much more. If you've been infected, contact your IT department.