Pawn Storm Update: iOS Espionage App Found

By By: Feike Hacquebord, Fernando Merces Feb 04, 2015 Read time: 4 min (962 words)

Published: 2015-02-04 · Archived: 2026-04-05 15:23:41 UTC

Updated February 6, 2015, 10:30 AM PST Trend Micro™ Mobile Security protects users' iOS devices and stops threats before they reach them. Trend Micro Mobile Security offers protection and detects these malware using the cloud-based Smart Protection Network™ and Mobile App Reputation technology.

Updated February 11, 2015, 7:52 PM PST In a previous version of this blog posting, we stated that the iOS device doesn't have to be jailbroken per se for the malware to be installed. We revisited this finding and found that the iOS device indeed needs to be jailbroken. The exact way how the actors install the espionage malware on iOS devices is currently unknown to us. It is very likely that social engineering is an important part.

In our continued research on Operation Pawn Storm, we found one interesting poisoned pawn—spyware specifically designed for espionage on iOS devices. While spyware targeting Apple users is highly notable by itself, this particular spyware is also involved in a targeted attack.

Background of Operation Pawn Storm

Operation Pawn Storm is an active economic and political cyber-espionage operation that targets a wide range of entities, like the military, governments, defense industries, and the media. The actors of Pawn Storm tend to first move a lot of pawns in the hopes they come close to their actual, high profile targets. When they finally successfully infect a high profile target, they might decide to move their next pawn forward: advanced espionage malware.

The iOS malware we found is among those advanced malware. We believe the iOS malware gets installed on already compromised systems, and it is very similar to next stage SEDNIT malware we have found for Microsoft Windows' systems. We found two malicious iOS applications in Operation Pawn Storm. One is called *XAgent* (detected as IOS_XAGENT.A) and the other one uses the name of a legitimate iOS game, *MadCap* (detected as IOS_ XAGENT.B). After analysis, we concluded that both are applications related to SEDNIT. The obvious goal of the SEDNIT-related spyware is to steal personal data, record audio, make screenshots, and send them to a remote command-and-control (C&C) server. As of this publishing, the C&C server contacted by the iOS malware is live.

Analysis of XAgent

The XAgent app is fully functional malware. After being installed on iOS 7, the app's icon is hidden and it runs in the background immediately. When we try to terminate it by killing the process, it will restart almost immediately. Installing the malware into an iOS 8 device yields different results. The icon is not hidden and it also cannot restart automatically. This suggests that the malware was designed prior to the release of iOS 8 last September 2014.

Data Theft Capabilities

The app is designed to collect all kind of information on an iOS device. It is able to perform the following routines:

- Collect text messages
- Get contact lists
- Get pictures
- Collect geo-location data
- Start voice recording
- Get a list of installed apps
- Get a list of processes
- Get the Wi-Fi status

Figure 1. XAgent code structure

C&C Communication

Besides collecting information from the iOS device, the app sends the information out via HTTP. It uses POST request to send messages, and GET request to receive commands.

Formatted Log Messages

The malware's log messages are written in HTML and color coded, making it easier for human operators to read. Error messages tend to be in red, while others are in green as shown in the figure below.

Figure 2. Color-coded HTML log messages

A Well-Designed Code Structure

We can see that the code structure of the malware is very organized. The malware looks carefully maintained and consistently updated.

Figure 3. XAgent code structure

The app uses the commands *watch*, *search*, *find*, *results*, *open*, and *close*.

Figure 4. List of base URIs

Randomly Generated URI

The full uniform resource identifier (URI) for C&C HTTP requests is randomly generated, according to a template agreed upon with the C&C server. The base URI can be seen in Figure 4, and parameters are chosen from the list below and appended to the base URI.

Figure 5. List of parameters used with URIs

Here are corresponding implementations we got during our reversing:

Figures 6 and 7. Code for URI generation

Token Format and Encoding

The malware uses a token to identify which module is communicating. The token is Base64 encoded data, but padded with a 5-byte random prefix so that it looks like valid Base64 data. See the first line “ai=” part in the figure below.

Figure 8. Client (XAgent) request

Reverse engineering also revealed additional communication functions.

Figure 9. HTTP communication functions

Figure 10. C2 server

FTP Communication

The app is also able to upload files via FTP protocol.

Figure 11. FTP communication functions

Analysis of "MadCap" "

Madcap" is similar to the XAgent malware, but the former is focused on recording audio. "Madcap" can only be installed on jailbroken devices.

Figure 12. Code structure of Madcap

Possible Infection Methods

The exact methods of installing these malware is unknown. As far as we can tell the iOS device has to be jailbroken to install the Xagent malware. However we have seen one instance wherein a lure involving XAgent simply says “Tap Here to Install the Application."

Figure 13. Site used in downloading XAgent

It is good to note that it is still possible to install the malicious app into non-jailbroken devices if the app is signed using Apple's enterprise certificate. Another possible scenario is infecting an iPhone after connecting it to a compromised or infected Windows laptop via a USB cable.

To learn more about this campaign, you may refer to our report, Operation Pawn Storm Using Decoys to Evade Detection. The hashes of the related files are:

- 05298a48e4ca6d9778b32259c8ae74527be33815
- 176e92e7cfc0e57be83e901c36ba17b255ba0b1b
- 30e4decd68808cb607c2aba4aa69fb5fdb598c64
