Operation Triangulation: iOS devices targeted with previously unknown malware


Authors

- Igor Kuznetsov
- Valentin Pashkov
- Leonid Bezvershenko
- Georgy Kucherin

While monitoring the network traffic of our own corporate Wi-Fi network dedicated for mobile devices using the Kaspersky Unified Monitoring and Analysis Platform (KUMA), we noticed suspicious activity that originated from several iOS-based phones. Since it is impossible to inspect modern iOS devices from the inside, we created offline backups of the devices in question, inspected them using the Mobile Verification Toolkit's mvt-ios and discovered traces of compromise.

We are calling this campaign "Operation Triangulation", and all the related information we have on it will be collected on the Operation Triangulation page. If you have any additional details to share, please contact us: triangulation[at]kaspersky.com.

What we know so far

Mobile device backups contain a partial copy of the filesystem, including some of the user data and service databases. The timestamps of the files, folders and the database records allow to roughly reconstruct the events happening to the device. The mvt-ios utility produces a sorted timeline of events into a file called "timeline.csv", similar to a super-timeline used by conventional digital forensic tools.

Using this timeline, we were able to identify specific artifacts that indicate the compromise. This allowed to move the research forward, and to reconstruct the general infection sequence:

- The target iOS device receives a message via the iMessage service, with an attachment containing an exploit.
- Without any user interaction, the message triggers a vulnerability that leads to code execution.
- The code within the exploit downloads several subsequent stages from the C&C server, that include additional exploits for privilege escalation.
- After successful exploitation, a final payload is downloaded from the C&C server, that is a fully-featured APT platform.
- The initial message and the exploit in the attachment is deleted

The malicious toolset does not support persistence, most likely due to the limitations of the OS. The timelines of multiple devices indicate that they may be reinfected after rebooting. The oldest traces of infection that we discovered happened in 2019. As of the time of writing in June 2023, the attack is ongoing, and the most recent version of the devices successfully targeted is iOS 15.7.

The analysis of the final payload is not finished yet. The code is run with root privileges, implements a set of commands for collecting system and user information, and can run arbitrary code downloaded as plugin modules from the C&C server.

Forensic methodology

It is important to note, that, although the malware includes portions of code dedicated specifically to clear the traces of compromise, it is possible to reliably identify if the device was compromised. Furthermore, if a new device was set up by migrating user data from an older device, the iTunes backup of that device will contain the traces of compromise that happened to both devices, with correct timestamps.

Preparation

All potential target devices must be backed up, either using iTunes, or an open-source utility idevicebackup2 (from the package libimobiledevice). The latter is shipped as a pre-built package with the most popular Linux distributions, or can be built from the source code for MacOS/Linux.
To create a backup with idevicebackup2, run the following command:

```
idevicebackup2 backup --full $backup_directory
```

You may need to enter the security code of the device several times, and the process may take several hours, depending on the amount of user data stored in it.

Install MVT

Once the backup is ready, it has to be processed by the Mobile Verification Toolkit. If Python 3 is installed in the system, run the following command:

```
pip install mvt
```

A more comprehensive installation manual is available the MVT homepage.

Optional: decrypt the backup

If the owner of the device has set up encryption for the backup previously, the backup copy will be encrypted. In that case, the backup copy has to be decrypted before running the checks:

```
mvt-ios decrypt-backup -d $decrypted_backup_directory $backup_directory
```

Parse the backup using MVT

```
mvt-ios check-backup -o $mvt_output_directory $decrypted_backup_directory
```
This command will run all the checks by MVT, and the output directory will contain several JSON and CSV files. For

the methodology described in this blogpost, you will need the file called timeline.csv.

Check timeline.csv for indicators

1. The single most reliable indicator that we discovered is the presence of data usage lines mentioning the process named "BackupAgent". This is a deprecated binary that should not appear in the timeline during regular usage of the device. However, it is important to note that there is also a binary named "BackupAgent2", and that is not an indicator of compromise. In many cases, BackupAgent is preceded by the process "IMTransferAgent", that downloads the attachment that happens to be an exploit, and this leads to modification of the timestamps of multiple directories in the "Library/SMS/Attachments". The attachment is then deleted, leaving only modified directories, without actual files inside them:

```
2022-09-13 10:04:11.890351Z Datausage
IMTransferAgent/com.apple.datausage.messages (Bundle ID:
com.apple.datausage.messages, ID: 127) WIFI IN: 0.0, WIFI OUT: 0.0 - WWAN IN:
76281896.0, WWAN OUT: 100956502.0
2022-09-13 10:04:54.000000Z Manifest Library/SMS/Attachments/65/05 - MediaDomain
2022-09-13 10:05:14.744570Z Datausage BackupAgent (Bundle ID: , ID: 710) WIFI IN:
0.0, WIFI OUT: 0.0 - WWAN IN: 734459.0, WWAN OUT: 287912.0
```

2. There are also less reliable indicators, that may be treated as IOCs if several of them happened within a timeframe of minutes:
   - Modification of one or several files: *com.apple.ImageIO.plist, com.apple.locationd.StatusBarIconManager.plist, com.apple.imservice.ids.FaceTime.plist*
   - Data usage information of the services *com.apple.WebKit.WebContent, powerd/com.apple.datausage.diagnostics, lockdownd/com.apple.datausage.security*

Example:
```
2021-10-30 16:35:24.923368Z Datausage IMTransferAgent/com.apple.MobileSMS (Bundle
ID: com.apple.MobileSMS, ID: 945) WIFI IN: 0.0, WIFI OUT: 0.0 - WWAN IN: 31933.0,
WWAN OUT: 104150.0
2021-10-30 16:35:24.928030Z Datausage IMTransferAgent/com.apple.MobileSMS (Bundle
ID: com.apple.MobileSMS, ID: 945)
2021-10-30 16:35:24.935920Z Datausage
IMTransferAgent/com.apple.datausage.messages (Bundle ID:
com.apple.datausage.messages, ID: 946) WIFI IN: 0.0, WIFI OUT: 0.0 - WWAN IN:
47743.0, WWAN OUT: 6502.0
2021-10-30 16:35:24.937976Z Datausage
IMTransferAgent/com.apple.datausage.messages (Bundle ID:
com.apple.datausage.messages, ID: 946)
2021-10-30 16:36:51.000000Z Manifest
Library/Preferences/com.apple.locationd.StatusBarIconManager.plist - HomeDomain
2021-10-30 16:36:51.000000Z Manifest Library/Preferences/com.apple.ImageIO.plist
- RootDomain
```

Another example: modification of an SMS attachment directory (but no attachment filename), followed by data usage of com.apple.WebKit.WebContent, followed by modification of com.apple.locationd.StatusBarIconManager.plist. All the events happened within a 1-3 minute timeframe, indicating the result of a successful zero-click compromise via an iMessage attachment, followed by the traces of exploitation and malicious activity.
```
2022-09-11 19:52:56.000000Z Manifest Library/SMS/Attachments/98 - MediaDomain
2022-09-11 19:52:56.000000Z Manifest Library/SMS/Attachments/98/08 - MediaDomain
```

```
2022-09-11 19:53:10.000000Z Manifest Library/SMS/Attachments/98/08 - MediaDomain
2022-09-11 19:54:51.698609Z OSAnalyticsADDaily com.apple.WebKit.WebContent WIFI
IN: 77234150.0, WIFI OUT: 747603971.0 - WWAN IN: 55385088.0, WWAN OUT:
425312575.0
2022-09-11 19:54:51.702269Z Datausage com.apple.WebKit.WebContent (Bundle ID: ,
ID: 1125)
2022-09-11 19:54:53.000000Z Manifest
Library/Preferences/com.apple.locationd.StatusBarIconManager.plist - HomeDomain
2022-06-26 18:21:36.000000Z Manifest Library/SMS/Attachments/ad/13 - MediaDomain
2022-06-26 18:21:36.000000Z Manifest Library/SMS/Attachments/ad - MediaDomain
2022-06-26 18:21:50.000000Z Manifest Library/SMS/Attachments/ad/13 - MediaDomain
2022-06-26 18:22:03.412817Z OSAnalyticsADDaily com.apple.WebKit.WebContent WIFI
IN: 19488889.0, WIFI OUT: 406382282.0 - WWAN IN: 66954930.0, WWAN OUT:
1521212526.0
2022-06-26 18:22:16.000000Z Manifest Library/Preferences/com.apple.ImageIO.plist
- RootDomain
2022-06-26 18:22:16.000000Z Manifest
Library/Preferences/com.apple.locationd.StatusBarIconManager.plist - HomeDomain
2022-03-21 21:37:55.000000Z Manifest Library/SMS/Attachments/fc - MediaDomain
2022-03-21 21:37:55.000000Z Manifest Library/SMS/Attachments/fc/12 - MediaDomain
2022-03-21 21:38:08.000000Z Manifest Library/SMS/Attachments/fc/12 - MediaDomain
2022-03-21 21:38:23.901243Z OSAnalyticsADDaily com.apple.WebKit.WebContent WIFI
IN: 551604.0, WIFI OUT: 6054253.0 - WWAN IN: 0.0, WWAN OUT: 0.0
2022-03-21 21:38:24.000000Z Manifest
Library/Preferences/com.apple.locationd.StatusBarIconManager.plist - HomeDomain
```

3. An even less implicit indicator of compromise is inability to install iOS updates. We discovered malicious code that modifies one of the system settings file named *com.apple.softwareupdateservicesd.plist*. We observed update attempts to end with an error message "Software Update Failed. An error ocurred downloading iOS".

Network activity during exploitation

On the network level, a successful exploitation attempt can be identified by a sequence of several HTTPS connection events. These can be discovered in netflow data enriched with DNS/TLS host information, or PCAP dumps:

- Legitimate network interaction with the iMessage service, usually using the domain names *.ess.apple.com
- Download of the iMessage attachment, using the domain names .icloud-content.com, content.icloud.com
- Multiple connections to the C&C domains, usually 2 different domains (the list of known domains follows). Typical netflow data for the C&C sessions will show network sessions with significant amount of outgoing traffic.

Time	Server Name	Destination	Destination Port	Protocol
222.577175	init.ess.apple.com	62.115.253.208	443	TLSv1.3
223.248546	kt-prod.ess.apple.com	17.145.0.2	443	TLSv1.3
250.471089	p113-caldav.icloud.com	17.250.84.36	443	TLSv1.2
301.339923	edge-102.sesto4.icloud-content.com	17.250.84.37	443	TLSv1.3
302.194211	p31-content.icloud.com	17.250.84.22	443	TLSv1.3
314.766744	setup.icloud.com	17.250.84.19	443	TLSv1.2
339.869951	backuprabbit.com	104.21.21.154	443	TLSv1.3
359.630968	gsa.apple.com	17.32.194.2	443	TLSv1.2
360.605764	backuprabbit.com	104.21.21.154	443	TLSv1.3
361.092903	pds-init.ess.apple.com	62.115.253.218	443	TLSv1.3
368.065719	cloudsponcer.com	104.21.79.172	443	TLSv1.3
377.414078	backuprabbit.com	104.21.21.154	443	TLSv1.3
423.442812	gateway.icloud.com	17.250.84.4	443	TLSv1.3
426.333906	identity.ess.apple.com	17.138.176.4	443	TLSv1.3
427.062256	identity.ess.apple.com	17.138.176.4	443	TLSv1.3
428.386581	identity.ess.apple.com	17.138.176.4	443	TLSv1.3
429.057571	identity.ess.apple.com	17.138.176.4	443	TLSv1.2
437.411511	iphone-ld.apple.com	62.115.253.233	443	TLSv1.3
500.156738	init.itunes.apple.com	184.51.132.49	443	TLSv1.3
760.068442	courier.push.apple.com	17.57.146.133	5223	TLSv1.3
761.825773	iphone-ld.apple.com	62.115.253.219	443	TLSv1.3
762.498958	cloudsponcer.com	104.21.79.172	443	TLSv1.3
765.125499	gs-loc.apple.com	17.36.206.4	443	TLSv1.3

Network exploitation sequence, Wireshark dump

The iMessage attachment is encrypted and downloaded over HTTPS, the only implicit indicator that can be used is the amount of downloaded data that is about 242 Kb.

Encrypted iMessage attachment, Wireshark dump

C&C domains

Using the forensic artifacts, it was possible to identify the set of domain name used by the exploits and further malicious stages. They can be used to check the DNS logs for historical information, and to identify the devices currently running the malware:

addatamarket[.]net
backuprabbit[.]com
businessvideonews[.]com
cloudsponcer[.]com
datamarketplace[.]net
mobilegamerstats[.]com
snoweeanalytics[.]com
tagclick-cdn[.]com
topographyupdates[.]com
unlimitedteacup[.]com
virtuallaughing[.]com
web-trackers[.]com
growthtransport[.]com
anstv[.]net
ans7tv[.]net