SKYAGENT PROTOCOL DESCRIPTION

Matt Mastracci, Joshua Wise, Eric Smaxwill, Matthew Fogle

skyagent is a piece of apparent malware shipped on some Android phones (primarily HTC's Sprint phones). It is stored in /system/bin/skyagent and is setuid root on all known devices. By default, it listens on port 12345, but it has a few mandatory command line options to get it running.

The skyagent process, normally, is invoked as such:

```
/system/bin/skyagent logFileLocation=/data/local/ linuxCommandDevice=/dev/
ptmx
```

When invoked as such, the skyagent process will be listening on TCP port 12345 on all interfaces (including the mobile wireless interface). skyagent's initialization sequence also permits a few other parameters (specified in a similar fashion; i.e., param=value). These parameters have not been studied in depth, but for the sake of completeness, they are listed below:

- keyCaptureDevices
- penCaptureDevices
- frameBufferDevice (contains vulnerability, noted below)
- icdCommandDevices
- linuxCommandDevice
- keyInjectDevices
- penInjectDevice
- logFileLocation (contains vulnerability, noted below)
- penInject
- port

Any other parameters will result in the message "<param> is unknown" being printed to stderr, and will otherwise be ignored. logFileLocation contains a vulnerability that allows memory in .bss to be overwritten (there is a print to a fixed-size buffer), and frameBufferDevice contains a vulnerability that allows the stack to be overwritten. (As it turns out, neither of these can be used to execute code, as best we understand.)

The main body of the backdoor executes in a thread, and receives packets into another fixed-size stacked buffer. (A theme is sensed here.) The best we know, commands conform to the following structure:

```
struct packet {      /* All fields little-endian. */
     uint32_t magic; /* Must be SKYAGENT_MAGIC (0x573A3E14) */
     uint32_t len;   /* Length of data after command packet. */
     uint16_t cmd;   /* described below */
};
```

Currently known commands for the cmd field are:

- 0x0000: Undecoded -- believed to be an echo-type packet.
- 0x0001: Initialize input hooks.
- 0x0002: Shut down input hooks.
- 0x0003: Inject keystroke.
- 0x0004: Inject pen tap.
- 0x0008: Capture screen.
- 0x0009: Get process list.
- 0x000A: Unknown behavior -- based on symbols, appears to relay ICD (?) information to remote system.
- 0x000C: Shut down skyagent process, reboot system.
- 0x000D: getprop ro.build.version.incremental
- 0x000E: getprop ro.version
- 0x0010: Spawn child process as root.
- 0x0013: Inject pen drag.
- 0x00FF: Shut down skyagent process.

We believe this list to be complete. The only investigation performed beyond rough descriptions of each command has been on command 0x0010, the mechanism by which to spawn a process. The length is the length of the command to execute (no trailing null needed), and the data is appended directly after the packet. By example:

```
00000000 14 3e 3a 57 0b 00 00 00  10 00 2e 2f 73 74 61 67  |.>:W.... .../stag|
00000010 65 32 2e 73 68                                    |e2.sh|
```

This packet executes the command "./stage2.sh" as root. The magic number is **bolded**, the length is *italicized*, and the length is underlined.

These data should be sufficient to further investigate the intent and behavior of the skyagent binary.