Cracking Cell Phones

By Glenn Davisson

The easiest way to compromise a cell phone is to get the user to install malware. This is fairly trivial from a technical sense – the attacker wants to do things that the phones are designed to let app writers do. The trick is to get the target to install it, and that's a roleplaying situation, not covered here.

There are three approaches to remotely breaking into a cell phone:

Attack the phone operating system through the cell network. This is the most difficult attack, and incurs a -40 to the skill check to find an exploit that will allow access. Once access is obtained, there will be a -30 to all skill checks to access information at the user operating system level (this is pretty much everything other than any calls or texts currently incoming or outgoing). The phone simply isn't designed for the hardware level to control the user level.

Such an attack can come from anywhere in the world, if the attacker knows the identity of the phone (the IMEI number, which can be gotten from the phone number – if one has access the phone company's records) and has access to the cell network it is connected to.

Attack the user interface through the WiFi system. (Note: Not all cell phones have WiFi, but nearly all smart phones do.) This is the most common method of attack, and provides no modifiers for either the initial attack or retrieving data.

Such an attack can come from anywhere on the local network, but requires WiFi be on. A -10 penalty applies

Note: This was originally written for Covert Ops. It has been genericized to work with any percentage based game. For a d20 based game, divide everything by 5.

It does assume the game system in use has some kind of critical success/critical failure mechanism.

Cell phones have two completely separate sets of hardware[1] and software that operate independently:

The **phone system level** handles the cellular connection. The hardware and software are both very simple, making it easier to avoid bugs that allow attacks.

The user never interacts with it directly; it is accessed only through the user interface.

The **user interface level** runs on separate hardware, with a separate operating system, such as iOS or Android, and is basically a small computer not unlike a desktop system. Both hardware and software are much more complicated than the system level, and bugs and vulnerabilities are inevitable.

[1]Some phones also have a separate **encryption subsystem** that runs on its own hardware, but this is represented by a negative modifier to skill checks to break in or retrieve data.

to the skill check if the WiFi is not actually connected to anything (but is turned on).

Attack the user interface through Bluetooth. (Like WiFi, not all cell phones have Bluetooth, but nearly all smart phones do.) This is the easiest method of attack, giving a +20 to the skill check to gain access, but no modifier for checks to retrieve data. While Class 2 Bluetooth devices (nearly all cell phones) have a theoretical range of 10 meters (33 feet), in real world conditions it is often a lot less. Even normal wood/drywall walls will significantly reduce the range.

In addition to the attack vector, other modifiers result from the type of phone. No opinions will be offered here on the quality of security offered by real life phone models, so types will be classified as Cheap, Moderate, and Expensive, and Old, Average, and New. Cheap and Old each offer +10 to skill checks. Expensive and New each offer -10 to skill checks. Some phones will also have a dedicated encryption subsystem, which offers an additional -20 to skill checks. These modifiers apply to both the attack roll and rolls to retrieve specific information.

To attack a phone, the attacker makes a skill check with appropriate modifiers. This takes 1d10 minutes per attempt (or turns, or whatever seems appropriate – this is something that can be done even under combat conditions, but not easily). If the attack critically succeeds, any security system present is deactivated, and further skill checks on that phone are at +10. If the attack is successful (but not critically), access is granted. If the attack fails (but not critically), there is no access, but further attacks are unmodified. If the attack critically fails, an immediate second skill check (with a -20 penalty) is necessary to avoid triggering any security systems that are present (not all phones have them). This could also render the phone inoperable.

Security Sytems: Anti-malware features are increasingly common on smart phones and tablets. A typical app would simply block the attempt and log that it has done so. Better quality apps might signal the user that is has occurred, perhaps by beeping or posting a notice on the screen. High security apps might send an alert to a remote monitoring system. The app might also lock the phone, or wipe its memory entirely.

Once a phone has been compromised, the attacker has several options. Each takes 1 turn and requires a skill check (with modifiers for attack vector and type of phone, as noted above) – downloading data may take longer, depending on how much there is:

1. Deactivate the security monitoring system. (Until this is done, any other actions requires a second skill check (with a -20 penalty) to prevent triggering an alarm.)

2. Retrieve a specific body of information (as a single dump), such as all texts, all call records, address book, all emails, data from a specific app, etc.

3. Retrieve an encryption key. (If the phone has an encryption subsystem, this is impossible to do directly, but can be done by feeding known data into the encryption subsystem, retrieving the encrypted version, and reverse engineering the key. This is with the usual modifiers for the phone, but takes 1d10 turns to generate enough data to be feasible, followed by a second skill check (without phone modifiers – this is done on the attackers own system), taking 2d10 turns, to generate the key.

4. If the phone is not in use, the microphone can be activated. If it is in use, the call can be tapped. If the phone is connected to a Bluetooth headset, there is an additional +20 modifier on this roll.

5. Delete or alter stored data in any one app.

6. Activate and control any app. (-20 to do so without the user being aware.)

7. Turn WiFi or Bluetooth on or off.

8. Use the phone's WiFi signal characteristics to track the phone's location to within half a meter, relative to a known access point (-30 to skill check, but +20 if Bluetooth is turned on, and +20 if the access point is also controlled).

9. Access the phone's GPS location. An additional skill check will allow this to be altered to provide false data to the phone's user, or to anyone else (like the 911 operator).

In some cases (such as a burner phone), once data is retrieved, an appropriate skill check (by the intruder or someone else) may be needed to extract useful information out of it. This may also require access to other resources, such as tracking a phone's location through metadata records, and coordinating it with similar records for other phones to get a name. This would require access to phone company records, and can take hours, days, or months. Often, stored texts, browser history or app data will give clues to the user's identity as well.