Part I

Side Channel Attacks

Side channel attacks are a class of attacks that gathers sensitive information passively. Generally it does not require direct access to the target in question, and in fact the less direct the access required is, the harder the attack can be to prevent. The attack then makes inferences from the gathered data and generally somehow derives sensitive information from the stream.

Joe: Side channels not necessarily passive, you can have active side channel attacks such as timing attacks versus passwords.
Dan: Good point. (Thanks)

Part II

Keylogging via Side Channels

In this case, were going to look at keylogging via a side channel. There are a variety of ways this has been done: acoustic attacks, power monitoring, lasers (to measure vibration, effectively acoustic – pretty cool) as well as Van Eck phreaking.

Kurt: does the sound map 1:1 to keys or use other time info like three keys presses
Dan: Use a neural network, but yeah, basically 1:1

More on Van Eck Phreaking

Using $15 of equipment we can read the contents of CRT from hundreds of feet, through walls. Effective key logging, via electromagnetic leakage. (Joe: Read more about TEMPEST shielding if you’re interested). Government was obviously very interested in this technology. However, thanks to technology we
don’t even need to spend $15, we can just use our phones to gather side channel data.

Part III
Sound Comber

SoundComber intelligently extracts sensitive info (e.g. credit card numbers, PINs) using microphone. However, because it has access to a restricted sensor it is much more difficult for it to have Internet access, thus captured data cannot leave. However, if we pair it with another application...

Part IV
Covert Android Communication

Because Android apps run inside sandboxes, interprocess communication is difficult. Thus we can use a side channel to convey information between applications. There are a few ways we can do this, we can send global notices via settings changes, volume, wake-lock, and orchestrated file system locking. The latter provides us with the greatest throughput. Both our colluding applications create a series of files. Our application with internet access tries to open a file in a long series, and if the file is locked the colluding application has signaled a “1” bit. Otherwise, its signaled a “0” bit.

Mitigation

Android only performs static permissions checks, maybe should be more contextual. However that presents it’s own set of challenges creating a good context to evaluate security decisions in. Especially when coupled with this next attack.

Part V
Accelerometer Data

The accelerometer in a phone can be used to record vibrations, so if our target phone is set next to a keyboard, we can read the desk vibration and infer keystrokes. However, in this case the sensor is not as fine as others listed to far, and we hit something called a Nyquist rate, which describes the minimum sample rate, and serves as a hard bound on our ability to read data.

Because this data is more noisy, initial efforts of trying to do 1:1 key detection
did not work. Next up, the authors broke down the problem into a set of pairs, using left or right sides of the keyboard to do pattern matching. With a targeted dictionary attack, this yields around 80% accuracy, kinda (as in, it returned a set of 10 words, of which one was the actual word). It’s also worth noting that this relies on proximity data, with the phone exactly 2” away from the keyboard.

Kurt: Really? Seems like things change a lot
Adam: other info could be available
Ben: Passwords aren’t words
Adam: still reduce search space
Dan: Why using neural net as opposed to markov model?
Dan: Trying to emulate acoustic paper
Ben: how do we know really, what word it is. Not just top 10
Adam: we can use predictive stuff based on previous words
Getty: the longer the word, the easier it will be to pick out
Adam: would putting it under be better?
Nick: probably not, looking at distance of vibration
Joe: Also, if you had physical access there are easier ways...
Dan: Mitigation -> don’t set phone next to keyboard
Joe: We can also use a known plaintext attack for calibration purposes

Mitigation

This attack is easier to mitigate because it’s a very fragile system overall. One of the easiest mitigation paths might be to limit the accelerometer data to <30hz which would be below the Nyquist threshold for extracting data. Other work in the form of TouchLogger shows that the method is easier with an on screen keyboard, and gets more accurate as the keyboard gets larger. This does not bode well for tablets. No mitigation is proposed for this attack.

Joe: this works with numbers, probably work worse with keyboard?
Dan: probably, but the data they had was really nice. Earlier data was way worse so this could probably be made better

Part VI

Final Thoughts

Be very careful with the sensors we surround ourselves with because they’re probably not nearly as secure as we would hope. Also, RFID is probably a bad idea.