Malicious Modification of Software Defined Radios
Using Hardware Breakpoints

Jatin Thakkar, Alexandru Turcu, Jung-Min “Jerry” Park
{jatinst, talex, jungmin}@vt.edu

The unprecedented flexibility of Software Defined Radios (SDRs) raises new security concerns not applicable to current handsets.

One serious threat in this context is the unauthorized modification of SDR to change its operating characteristics.

We demonstrate an attack on an SDR implementation (i.e., GNU Radio) that:

1. can make controlled changes to the transmission frequency w/o making changes to the signal processing software itself;
2. exploits hardware features common to modern GPPs (viz, debug registers and floating point registers); and
3. circumvents tamper resistance provided by checksumming-based integrity verification mechanisms.

Assumptions and attack model:
Our attack technique is based on the following assumptions:
1. Availability of debug registers
2. Presence of an operating system and access to it
3. Malicious host model
4. Checksumming-based tamper resistance protecting the SDR

Attack Mechanism:

1. Pinpointing the code of attack
   - We exploited the fact that GNU Radio’s code for dealing with the transmission frequency uses floating point numbers.
   - We obtained a log of all floating point operations performed by the processor, and were able to identify those that handle the initialization of transmission frequency in hardware.
   - Steps for obtaining the log are explained in Figure 1.

2. Inserting and handling breakpoints
   - Once we determined the instruction that manipulates frequency from the obtained log, we proceeded to place a breakpoint on the vulnerable instruction.
   - Since the particular library that contains it is not loaded at the time the execution starts, we delayed placing the breakpoint until the library is available.
   - We placed a breakpoint temporarily in the dynamic library loader’s code and checked for our desired library. When the library is available, the final breakpoint is placed on the target code.
   - This process is illustrated in Figure 2.

3. Changing the transmission frequency by altering the floating point register values
   - At a particular point in the operation, the argument to manipulate frequency (i.e., numerically controlled oscillator (NCO) ratio) is present on the top of FPR stack.
   - To modify the transmission frequency, we changed the FPR value at this point.

Experimental Results:
Figures 3, 4, and 5 show the received signal’s spectral density (as observed on the receiving USRP), without frequency modification, with frequency shifted down, and with frequency shifted up, respectively. One can clearly observe that the modifications made by the attack resulted in the change of the radio’s carrier frequency.

Conclusion:
A software-only tamper resistance approach is inadequate to harden SDRs against unauthorized modification. A combination of both software and hardware based approaches is needed.