Advanced Wireless Fuzzing to Exploit New Bluetooth Escalation Targets

Jan Ruge, Jiska Classen, Francesco Gringoli, Matthias Hollick
Motivation

- Radio protocols have huge attack surface
- We focus on Bluetooth firmware
- Large unauthenticated attack surface
  - Devices are connectable by default
  - Lower protocols terminated in firmware
Why Emulation?

- Reverse engineering is hard
- Embedded device debugging is hard
- Static analysis does not give context
  - ~300 functions with *RX*
  - How are those invoked?
- Narrow down relevant code
Firmware Emulation - Modifications

- Added debug output
- Supports
  - Threading
  - HCI injection/extraction
  - Raw wireless frames injection/extraction
- Attach emulator as Bluetooth device
- Read wireless frames from stdin
Firmware Emulation - Concept

- Hooking by rewriting function prologue
- Extract clean firmware state
  - Memory
  - Memory Mapped I/O
  - Registers
- Restore memory layout
- Restore registers
- Go?

```
xmit_state:
push {r0-r12,lr}
ldr r0, =saved_regs
str sp, [r0]
bl xmit_memory
cont:
ldr r0, =saved_regs
ldr sp, [r0]
pop {r0-r12,lr}
bx lr
```
Firmware Emulation - Concept

- Link C code against firmware
- Assemble to ELF file
- Execute using `qemu-arm`
Oops, that’s not my bug...
CVE-2019-11516 (Broadcom)

- Heap corruption during device inquiry
  - Affects devices ~2010 - 2018
  - Exploited for chip RCE
  - Used by: Samsung, Apple, ...

- Bug located in Link Manager and BCS
Outcome & Conclusion

- Large scale firmware emulation possible
- Advantages
  - Same layer of abstraction – More readable code
  - No API calls – More performance
  - Same code base for patching device*
- Disadvantages
  - Process remains (less) tedious

* and find Android 0-Click RCE by accident

<table>
<thead>
<tr>
<th>Type</th>
<th>Severity</th>
<th>Updated AOSP versions</th>
</tr>
</thead>
<tbody>
<tr>
<td>DoS</td>
<td>Moderate</td>
<td>10</td>
</tr>
<tr>
<td>RCE</td>
<td>Critical</td>
<td>8.0, 8.1, 9</td>
</tr>
</tbody>
</table>

[git://github.com/seemoo-lab/frankenstein](git://github.com/seemoo-lab/frankenstein)