Interesting Research

according to Adam

Classics

“Error Cost Escalation Through the Project Life Cycle” by NASA

The reason to find bugs early.

"Agile" seems to question this, but I have not seen metrics like this to back up the "Agile" claims.

Classics

"Zero Days, Thousands of Nights" by Ablon (RAND)

Incredible metrics on zero-days, a difficult to study topic. How long do they last? How long do vulnerability researchers stick around? It’s all in here.

Classics

"Opportunities and Limits of Remote Timing Attacks" by Crosby et al

The best paper I've ever seen on remote timing attacks

Almost 10 years old!

Classics

"On The Limits of Steganograph" by Anderson

Similar to the above, but on covert channels

Over 20 years old!

Mobile Devices

"Practical Attacks Against Privacy and Availability in 4G/LTE Mobile Communication Systems" by Shaik et al

Awesomeness can be summed up in two pictures...

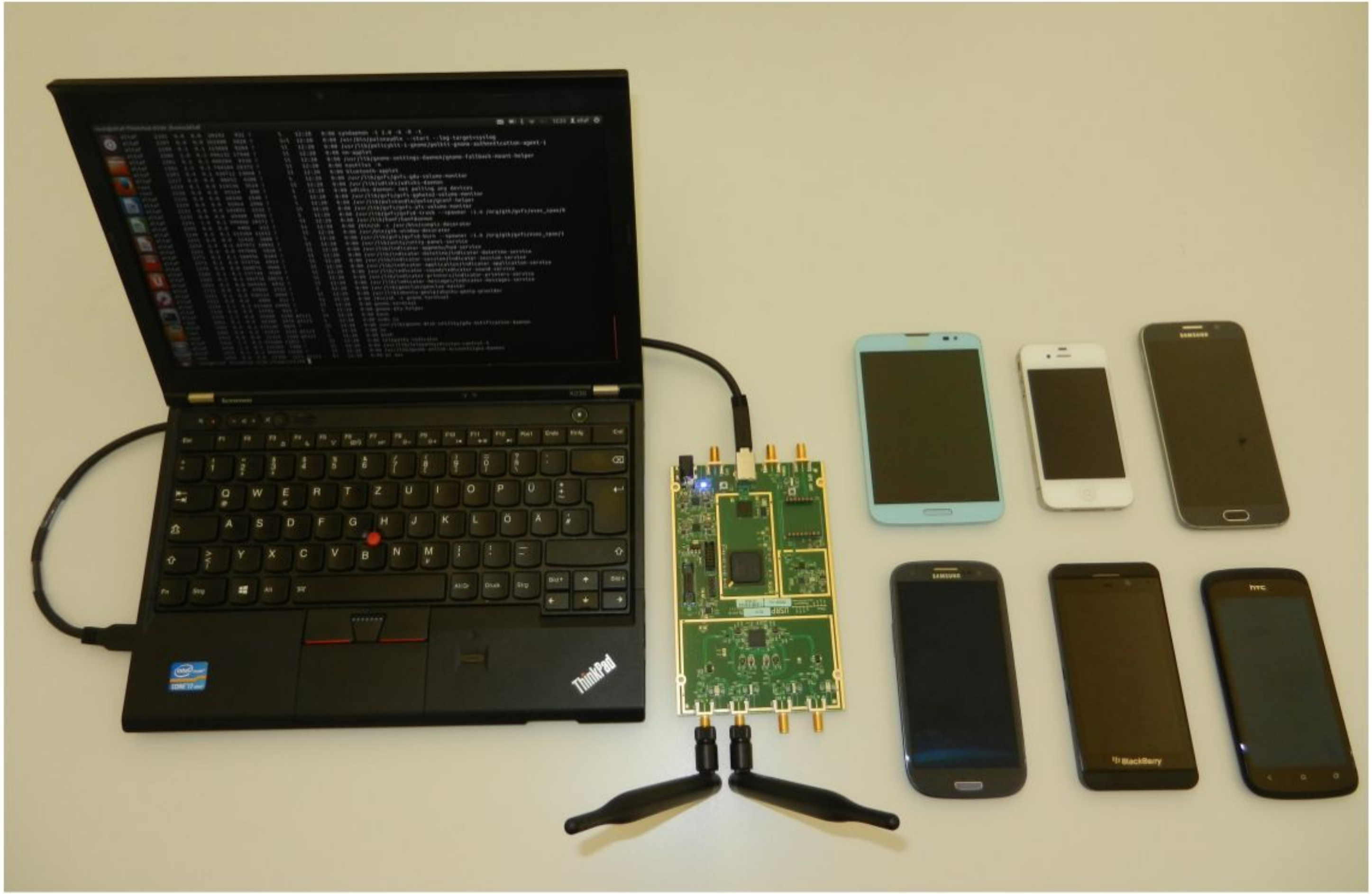


Fig. 3. Experimental setup

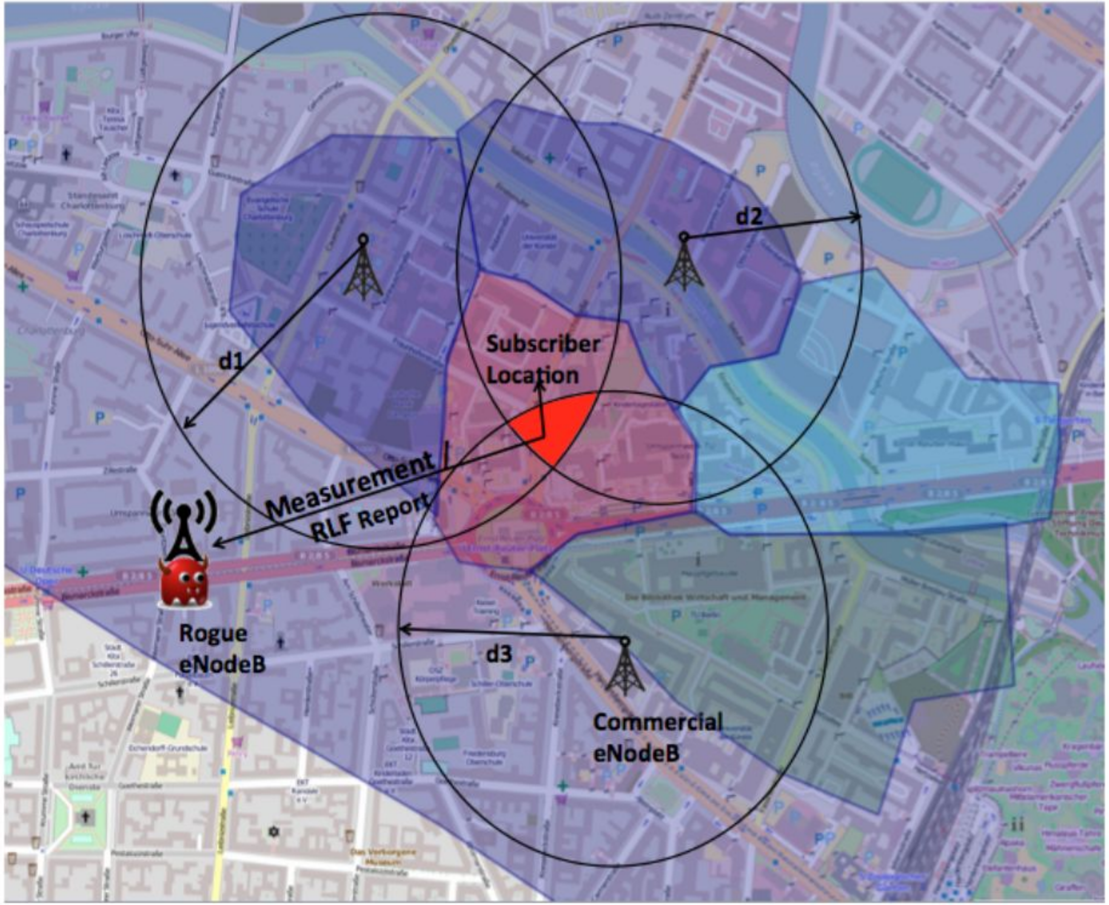


Fig. 7. Determining subscriber's precise location using trilateration (L3)

Mobile Devices

"ARTist: The Android Runtime Instrumentation and Security Toolkit" by Backes et al (Saarland University)

Taint tracking on Android 5+

Previous versions of Android covered by TrainDroid

Mobile Devices

"Challenges for Dynamic Analysis of iOS Applications" by Szydlowski & UCSB

Dynamic analysis on iOS & dealing with user interactions

Mobile Devices

"Behind the Scenes with iOS Security" by Krstić (head of Security, Apple)

Great primer on all Apple's iOS security mechanisms

Mobile Devices

"Unauthorized Cross-App Resource Access on MAC OS X and iOS" by Xing et al

Abusing URL Scheme to steal passwords, tokens, and so forth…

all from a sandboxed app!

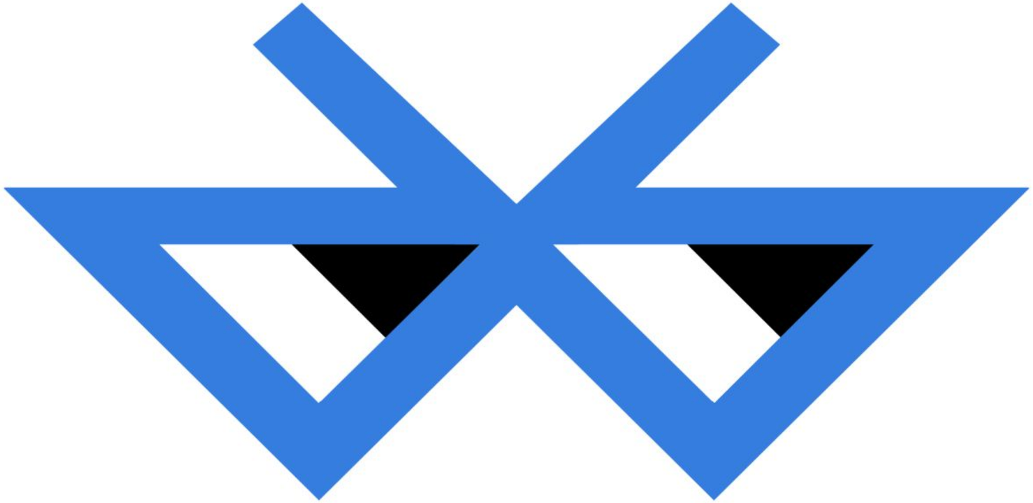
Mobile Devices

"BlueBorne" by Seri

Shows how poor the code is in Android, iOS, Linux, Bluetooth Pineapple

Also a good overview of how to map out attack surface

Fun fact: they changed their logo, my local PDF has the old one



BlueBorne

BlueBorne™

Hardware

"Implementation and Implications of a Stealth Hard-Drive Backdoor"
by Zaddach et al

- Rent a co-location box
- Infect the hard drive
- Let someone else rent it
- Exfil their data using your hard drive malware

Hardware

"Project Maux Mk.II" by Arrigo

Running a shell on a NIC instead of computer (in the normal sense) (2008!)

Hardware

"Attacking and Defending BIOS in 2015" by Yuriy et al

Yuriy is a legend when it comes to low-level security issues

Hardware

"BARing the System - New vulnerabilities in Coreboot & UEFI based systems"

by Yuriy et al

Hardware

"BitWhisper: Covert Signaling Channel between Air-Gapped Computers using Thermal Manipulations" by Dr. Mordechai Guri et al (Beer-Sheva)

The title pretty much says it all.

There's a whole series of these papers...

https://arxiv.org/search/cs?searchtype=author&query=Guri%2C+M

Hardware

- AirHopper: Bridging the Air-Gap between Isolated Networks and Mobile Phones using Radio Frequencies
- Fansmitter: Acoustic Data Exfiltration from (Speakerless) Air-Gapped Computers
- VisiSploit: An Optical Covert-Channel to Leak Data through an Air-Gap
- DiskFiltration: Data Exfiltration from Speakerless Air-Gapped Computers via Covert Hard Drive Noise
- USBee: Air-Gap Covert-Channel via Electromagnetic Emission from USB
- LED-it-GO: Leaking (a lot of) Data from Air-Gapped Computers via the (small) Hard Drive LED

Hardware

- HVACKer: Bridging the Air-Gap by Attacking the Air Conditioning System
- xLED: Covert Data Exfiltration from Air-Gapped Networks via Router LEDs
- aIR-Jumper: Covert Air-Gap Exfiltration/Infiltration via Security Cameras & Infrared (IR)
- MAGNETO: Covert Channel between Air-Gapped Systems and Nearby Smartphones via CPU-Generated Magnetic Field
- ODINI : Escaping Sensitive Data from Faraday-Caged, Air-Gapped Computers via Magnetic Fields
- MOSQUITO: Covert Ultrasonic Transmissions between Two Air-Gapped Computers using Speaker-to-Speaker Communication

Hardware

- PowerHammer: Exfiltrating Data from Air-Gapped Computers through Power Lines
- BeatCoin: Leaking Private Keys from Air-Gapped Cryptocurrency Wallets

Hardware

"SPEAKE(a)R: Turn Speakers to Microphones for Fun and Profit"
by Dr. Mordechai Guri et al (Beer-Sheva)

Again, the title pretty much says it all

Hardware

"A Large-Scale Analysis of the Security of Embedded Firmwares" by Coustin et al

Finding vulnerabilities in embedded devices at scale

Hardware

“Firmalice - Automatic Detection of Authentication Bypass Vulnerabilities in Binary Firmware" by Yan (Shellphish)

This is interesting because it's not finding memory corruption bug, it's finding backdoors (e.g. hard coded passwords, missing authentication, etc.)

Binary Analysis and Exploitation

"AEG: Automatic Exploit Generation" by Brumley et al

Given a binary before and after it was patches for a security issue, produce an exploit for the vulnerability.

Binary Analysis and Exploitation

"Unleashing MAYHEM on Binary Code" Brumley et al

This one blew my mind the first time I found it

Binary Analysis and Exploitation

"(State of) The Art of War: Offensive Techniques in Binary Analysis" by Shellphish

An excellent survey on machine code analysis and exploitation

Binary Analysis and Exploitation

"Coverage-based Greybox Fuzzing as Markov Chain" by Bohme et al

Attempts to escape the "hot" paths of execution to exercise new code

Site is currently returning 403 - Forbidden, I've emailed the author about it

Binary Analysis and Exploitation

"Directed Greybox Fuzzing" by Bohme et al

Attempts to guide fuzzing to a particular piece of code, requires target source

Binary Analysis and Exploitation

"Smart Greybox Fuzzing" by Bohme et al

More intelligent mutation and scheduling

Binary Analysis and Exploitation

"CollAFL: Path Sensitive Fuzzing" by Gan et al

AFL's code coverage system has collisions ("up to 75% of edges could collide with others in some applications"), this research shows a fast, collision resistant algorithm

Binary Analysis and Exploitation

"VUzzer: Application-aware Evolutionary Fuzzing" by Rawat et al

Uses taint tracking to find interesting inputs in fewer iterations

Only works on Linux on i386 and AMD64

Execution is so slow in practice that it performs worse than stock AFL

Could be improved it the fork server were implemented in their code

Binary Analysis and Exploitation

"Driller: Augmenting Fuzzing Through Selective Symbolic Execution"
by Shellphish

Describes the system which came in 3rd place for the Cyber Grand Challenge

Binary Analysis and Exploitation

"Enhancing Symbolic Execution with Veritesting" by Brumley et al

The Brumley paper that came after MAYHEM

Binary Analysis and Exploitation

"Angora: Efficient Fuzzing by Principled Search" by Chen & Chen

Attempts to get the benefits of symbolic execution, without using symbolic execution. Just released their code on 12/27/2018

Binary Analysis and Exploitation

"Full-speed Fuzzing: Reducing Fuzzing Overhead through Coverage-guided Tracing" by Stefan Nagy

Released on Monday from a frequent poster on the AFL mailing list, looks good

If that’s not enough...

- Phrack.org
- PoC || GTFO
- https://arxiv.org/list/cs.CR/recent (Cryptography & Security)
- Conferences: RECon, Infiltrate, and others