Expliot – IoT Security Testing and Exploitation Framework

BY ASEEM JAKHAR
About Me

• Aseem Jakhar
  • Co-Founder/Director R&D Payatu IoT Security Lab
  • Co-Founder
    ◦ null – The open security community
    ◦ nullcon Security Conference
    ◦ hardwear.io Security Conference

• Open source Developer
  • Jugaad - https://bitbucket.org/aseemjakhar/jugaad/src/master/README.TXT
  • Indroid - https://bitbucket.org/aseemjakhar/indroid/src/master/README.TXT
  • Dexfuzzer - https://bitbucket.org/aseemjakhar/dexfuzzer/src/master/
  • DIVA Android - https://github.com/payatu/diva-android

• Linux/Android/IoT
• Speaker/Trainer
  • Brucon, Hack in Paris, Defcon, Blackhat, Hack.lu, PHDays, Xcon, etc
  • Practical IoT Hacking
• Email: aseem@payatu.com
Agenda

• IoT Security issues
• IoT Attack Surface
• Problem Statement
• Meet expliot!
• Demos
IoT Security Issues

• Speed to market
• No/low motivation for security
• Little awareness about security issues
• Power/Cost limitation of security implementation
• Protocol
  • Custom protocols
  • No or default Authentication
  • Discovery mechanisms aid in recon
• Implementations still not that mature
• Cloud
  • Trust in telemetry data
IoT Attack Surface

- High level view
IoT Attack Surface

- Device

IoT Attack Surface

- **Device**
- **Services**
  - SSH, Telnet
  - Web
  - Proprietary services
- **Storage**
  - SD Card
  - USB
- **Firmware**
  - Encryption
  - Modification
- **Hardware**
  - Debug ports - UART/JTAG
  - Memory - Flash/EEPROM
  - Radio

IoT Attack Surface

- Cloud

Image source: https://s-media-cache-ak0.pinimg.com/564x/2f/3e/52/2f3e520f1b0465388d85732e6b2367a6.jpg
IoT Attack Surface

- **Cloud**
  - Communication
  - Storage

- **Business logic flaws**
  - Domain specific flaws

- **Owasp Web Top 10**
  - Standard Web security issues

Image source: https://s-media-cache-ak0.pinimg.com/564x/2f/3e/52/2f3e520f1b0465388d85732e6b2367a6.jpg
IoT Attack Surface

• Mobile

IoT Attack Surface

- **Mobile**
- Communication
- Storage
- Business logic flaws
  - Domain specific flaws
- **OWASP Mobile Top 10**
  - Standard mobile security issues

---

Getting ready for an IoT Pentest

Image Source: http://cdn.memecats.com/media/thumbs/embedded/398.jpg
What you think?

What actually happens!
Problem Statement

• Too many interfaces
• Tooooo many tools
Meet expliot!

- Pronounced expliot (expl-aa-yo-tee)
- Framework
  - IoT exploitation
  - IoT Penetration Testing
- Design Goals/Motivation
  - Simple to use
  - Extendable
  - Easy to write test cases
- Source code
  - https://gitlab.com/expliot_framework/expliot

Image Source: http://www.funnycatsite.com/pictures/robo_cat.jpg
Expliot Architecture

- Framework
- Cli
- TestSuite
- Test
- Protocols

Plugins Module
- exploitTest1
- ReconTest1
- TestFoobar

efconsole
Expliot Plugin

- Simple to implement
- Inherit from the relevant base Test class
- Import protocols only from expliot protocols package
- Only import required stuff from default python library
- Define members
  - Plugin information
  - command-line arguments
- Override 3 methods
  - pre() – Optional. Setup etc.
  - execute() – Mandatory. The main plugin execution code
  - post() – Optional. Cleanup etc.
from exploit.core.tests.test import Test, TCategory, TTarget, TLog

class Sample(Test):  
    def __init__(self):
        super().__init__
        self.name = "Sample name"
        self.summary = "Sample Summary"
        self.description = "Sample Description"
        self.author = "Sample author"
        self.email = "email@example.com"
        self.refs = ["https://example.com", "https://example.dom"]
        self.category = TCategory(TCategory.COAP, TCategory.SW, TCategory.EXPLOIT)
        self.target = TTarget(TTarget.GENERIC, TTarget.GENERIC, TTarget.GENERIC)

        self.argparser.add_argument("-r", "--rhost", required=True, help="IP address of the target")
        self.argparser.add_argument("-p", "--rport", default=80, type=int, help="Port number of the target. Default is 80")
        self.argparser.add_argument("-v", "--verbose", action="store_true", help="show verbose output")

    def pre(self):
        TLog.generic("Enter {}.pre()").format(self.id))
        # Only implement this if you need to do some setup etc.
        TLog.generic("Exit {}.pre()").format(self.id))

    def post(self):
        TLog.generic("Enter {}.post()").format(self.id))
        # Only implement this if you need to do some cleanup etc.
        TLog.generic("Exit {}.post()").format(self.id))

    def execute(self):
        TLog.generic("Sending request to server\{\} on port\{\}".format(self.args.rhost, self.args.rport))
        TLog.trydo("Searching imaginary database")
        TLog.success("Found matching entry in db - \{\}\".format("FooEntry"))
        self.snd = "GET / HTTP/1.1"
        TLog.generic("Sending command to server \{\} on port \{\}".format(self.args.rhost, self.args.rport))
        if self.args.verbose is True:
            TLog.generic("More verbose output. sending payload \{\}".format(snd))
        TLog.fail("No response received")
        TLog.generic("Re-sending command")
        self.rcv = "Response received from the server"
        # In case of failure (Nothing to do in case of success)
        self.result.setstatus(passed=False, reason="Server is not vulnerable")
Current Plugins

• BLE
  • Ble scanner
  • Read | Write | Fuzz characteristic values

• MQTT
  • Publish | Subscribe | Auth cracker

• Modbus
  • Read | Write

• CAN
  • Read | Write

• Serial
  • Brute Force | Fuzz

• Exploits
  • Tapplock - Unlock
  • Kankun Smartplug ON/OFF
MQTT – Security issues

• $SYS/# topics
• DoS
• Auth bruteforce
• Malicious telemetry data
MQTT – Demo Auth Bruteforce
MQTT – Demo Malicious Telemetry data

- Pwn cloud ~ Pwn ecosystem
BLE – Security Issues

• Characteristic value read/write
BLE – Plugins demo
Tapplock – Demo of tappunlock plugin

Unbreakable design

Bold. Sturdy. Secure. Tapplock one is crafted for the practical. Forged with Zamak 3 zinc alloy metal body and reinforced stainless steel shackle, strengthened by double-layered lock design with anti-shim and anti-pry technologies. The lock features unparalleled industrial design finished with electroplating.
UART – Security issues

- Root shell
- Custom shell with no input validation
- Hidden commands
UART – Demo Brute force commands

- DIVA IoT Board – Damn Insecure and Vulnerable App IoT
Road map

- Hardware interface for JTAG, SPI, I2C, etc.
- Radio protocol support – zigbee, IoRA etc.
- Firmware analysis test cases
- More IoT exploits
References

- Mosquitto - https://mosquitto.org
- MQTT php example - https://github.com/bluerhinos/phpMQTT/issues/6
Image Credits

- Jtagulator GUI - https://www.invincealabs.com/images/2016/05/jtagulator.png
- gqrx - http://farm6.staticflickr.com/5510/10912568935_e38bcfe964_z.jpg
- mutt client form - https://dutchtechy.files.wordpress.com/2014/03/csharp2.png
- eeprom reader - http://img.dxcdn.com/productimages/sku_336830_1.jpg
- Tapplock - https://scontent-mrs1-1.xx.fbcdn.net/v/t31.0-8/20247592_135954506993037_5653104370729346590_o.png?_nc_cat=0&oh=af98b07853a69891e5e87e9d14b46b63&oe=5BEC1FE9
Image Credits

- xds debugger - http://processors.wiki.ti.com/images/1/1e/CCS_XDS110_Debug_options.png
- limesdr - https://www.crowdsupply.com/img/e9d0/limesdr-2.jpg_project-body.jpg
- ubertooth - https://cdn.sparkfun.com/assets/parts/5/2/0/6/10573-01.jpg
- minicom - http://spritesmods.com/autobaud/minicom.png
- Burp suite - https://portswigger.net/sc/InstallingandConfiguring_LaunchingBurp_7.png
- nmap - https://nmap.org/movies/matrix/trinity-nmapscreen-hd-cropscale-418x250.jpg
- sqlite3 - https://dab1nmslvntp.cloudfront.net/wp-content/uploads/2015/02/142013979createTable.png
Thanks | Q&A

- You can reach me at aseem@payatu.com
- Expliot home to be – www.expliot.io
- Currently working on Contributor License agreement
- Use, Test, Suggest improvements