Tushar Jois
2018-10-16
Specification v0.3

WDCHK

Wireless Data Collection Harness & Kit

Motivation

Modern homes have increasing amounts of IoT devices, with many of these consumer devices communicating over the air via IEEE 802.11 (otherwise known as “Wi-Fi”). Wi-Fi, as an open-air protocol, has inherent privacy risks; encryption schemes, such as WPA, reduce this attack surface, but do not eliminate it. While disparate Wi-Fi tools are available, there currently is no readily-available toolkit which provides holistic data collection and analysis for 802.11 networks.

Functionality

WDCHK (pronounced “*woodchuck*”) is a targeted data collection framework written in Python. It aims to make 802.11 data collection easier, with a future goal of developing a wireless fingerprinting framework. All results (other than frame/packet traces) are returned as JSON, which allows for other programs to manipulate the data collected by WDCHK.

- **Passive traffic capture**. At a very basic level, WDCHK uses the supported monitor mode of a wireless card in order capture raw 802.11 frames, as well as metadata such as source, destination, and timestamps. WDCHK returns the captured frames as pcap files.
- **Traffic analysis**. Based on captured frames, WDCHK can perform rudimentary analysis of the state of the wireless environment, such as:
    - number of packets sent in/out,
    - partitioning the found devices as clients, bridges, or APs,
    - which clients are associated to a given AP,
    - which SSIDs a client has probed,
    - the number of frames sent over the air over time, and
    - AP management information.
- **Active traffic decryption**. If the user knows the Wi-Fi password, WDCHK can decrypt packets, using a combination of a WPA2-PSK (colloquially, a “Wi-Fi password”), injection of a deauthentication frame, and capture of the resultant 4-way handshake. This process yields the Pairwise Temporal Key (PTK), which, when combined with its passive traffic capture abilities, allows WDCHK to decrypt traffic.
- **Client location mapping**. WDCHK uses the *received signal strength indicator* (RSSI) metric in order to calculate distance to a wireless client or AP. It can also be supplied with GPS coordinates (latitude, longitude, altitude) in order to convert this RSSI metric into the estimated coordinates for nearby devices.

- **Data labeling**. WDCHK allows for data to be labeled properly, keeping a persistent mapping of known MAC addresses with their given names so users can keep track of wireless access points in reports.

Design

WDCHK is designed to incorporate modern software design principles, with an emphasis on object-oriented techniques. Much of WDCHK is implemented using the powerful Scapy framework, which simplifies the sniffing and spoofing of protocol primitives, including 802.11 frames. The backend for the active traffic decryption uses the pyDot11 project.

Classes

The following list enumerates the classes of WDCHK.

- TODO

Test Cases

WDCHK, when implemented, should incorporate the below series of unit tests in order to validate functionality.

Interface

WDCHK is a command-line application. As such, it should have a robust command-line interface that exposes relevant functionality to the user. This interface should be modular, in case a GUI version should ever be developed.