Capture of Beacon Frame and Probing Requests for Identification of Frequently Requested Access Points

Abstract:

Network security analysts are tasked with gathering and compiling information pertaining to unknown wireless networks. Many different tools can be utilized to provide this information, which may require manual intervention by the network security analyst to conduct a successful assessment. This manual interaction with the software tools may require prior working knowledge of not only the tools, but the protocols encountered during the network security evaluation. Our research is intended to allow an analyst to conduct their assessment independent of the previous working knowledge of the network under analysis, but of the wireless protocols encountered within the test environment. We intend to utilize existing open source software and build upon this foundation. Our objective is to construct a file representation of the network under test which will includes access point information, clients, and a collection of beacon probe requests broadcast by the clients. This file will be fabricated through the dissection of IEEE 802.1x frames encountered during the examination. This information can then be utilized to create rouge access points for further security assessments.

Tools used:

TP-Link TL-WN821N wireless dongle

RALINK USB Wifi RT5370 wireless dongle

Github iSniff source code [1]

Kali Linux

VirtualBox VM software

Python

Scapy

Github

Wireshark

Methodology:

1. Log into VM
2. Open terminal
   1. Run the following commands
      1. ifconfig
         1. determine if USB dongle detected and mounted.
         2. the USB dongle was reported as wlan0 and will be referred to as such for the duration of the document.
      2. ifconfig wlan0 down
         1. bring the wlan0 interface down to set it into monitor mode
      3. macchanger –r wlan0
         1. change the factory interface MAC address to a random value
      4. iwconfig wlan0 mode monitor
         1. set the usb dongle into monitor mode
      5. ifconfig wlan0 up
         1. bring the wlan0 interface back up
      6. wireshark &
         1. verify the interface is in monitor mode and capturing non-IP 802.1x traffic
3. Find and load script (saved as Tester.py)
   1. python Tester.py
      1. verify script works and is capturing traffic.

Works Cited:

[1] https://github.com/0x90/iSniff/blob/master/iSniff.py