Document: Risk Rating Details

1. Introduction

This document gives specific explanations for the risk rating metrics used in this project, and how the evaluation result come into being for every threat considered in this situation.

2. Risk Rating Metric -- Variant of Microsoft DREAD

At the beginning, I considered to refer DREAD risk rating methods from Microsoft.
The original DREAD consists of 5 parts,

- Damage potential(D)
- Reproducibility(R)
- Exploitability(E)
- Affected users(A)
- Discoverability(D)

with details shown in the form below:

	Rating	High (3)	Medium (2)	Low (1)
D	Damage potential	The attacker can subvert the security system; get full trust authorization; run as administrator; upload content.	Leaking sensitive information	Leaking trivial information
R	Reproducibility	The attack can be reproduced every time and does not require a timing window.	The attack can be reproduced, but only with a timing window and a particular race situation.	The attack is very difficult to reproduce, even with knowledge of the security hole.
E	Exploitability	A novice programmer could make the attack in a short time.	A skilled programmer could make the attack, then repeat the steps.	The attack requires an extremely skilled person and in-depth knowledge every time to exploit.
A	Affected users	All users, default configuration, key customers	Some users, non-default configuration	Very small percentage of users, obscure feature; affects anonymous users
D	Discoverability	Published information explains the attack. The vulnerability is found in the most commonly used feature and is very noticeable.	The vulnerability is in a seldom-used part of the product, and only a few users should come across it. It would take some thinking to see malicious use.	The bug is obscure, and it is unlikely that users will work out damage potential.

DREAD defines each parameter with different level(generally High/Medium/Low) and assigns corresponding score ranges, then calculates the sum of all five parameters as the final result. The final score will be mapped into different ranges which refer to different level of severity.

DREAD is a very simple and classical method to produce evaluation for security risks. However, it was designed to be used for assessment of high-level security areas,

especially for applications' security. It is even barely used for threat modeling in network layer, so does the second layer which is the core research environment of this project. Thus, lots of definitions, such as the five parameters and detailed description for each of them, can not be referred directly. Changes need to be made so that it can be more fitable in public wireless network scenarios.

Firstly, I considered to alter the definitions for some parameters, mainly for following two parameters:

- Damage potential: How great is the damage if the vulnerability is exploited?
  - Low: Leaking trivial information, e.g. existence, MAC address, general information of user's devices, request packets header/content(no sensitive information exposed)
  - Medium: Leaking sensitive information(in plaintext), fake authentication and association.
  - High: Attacker can get full trust authorization, run as administrator, fully control all communications, access to user's local directory/system privilege, etc.
- Affected Users
  - Low: Targeted at random user
  - Medium: Targeted at certain user
  - High: Targeted at AP, which will affect all users connected to it.

Then, I realized that some additional parameters need to be added into the original metrics, since they need much more concentration when talking about public wireless hotspots than generic application security:

- Additional One: Time Consuming (Since most of public Wi-Fi users connect to AP temporarily, if an attack can be successful in theory but need to take very long time to exploit, it will have less risk for normal users).
  - Very Low: > 1 hr
  - Low: 30 mins - 1 hr
  - Medium: 15 mins - 30 mins
  - High: 5 mins - 15 mins
  - Very High: < 5 mins

- Additional Two: Potential usage for further attack, further attack's severity and contribution to further attacks.
  - Very Low: Hard and less severe
  - Low: Hard and severe

- Medium: Not hard and not so severe
- High: Easy and not that severe
- Very High: Easy and very severe

Furthermore, I thought it is not that reasonable to use the sum of all parameters as the final result, Since among the current 7 parameters, 3 are for damage, 4 are for possibility, also for some special reason with wireless network, there are many situations that a threat always has very high possibility to be exploited but it's effects could be limited or just be as a trivial step for other threats. It's the tight relationship between all threats that makes DREAD not that practical in this occasion. So I consider to combine the extensioned DREAD parameters with classical formula "Risk = Damage * Possibility", and give this final rating method:

Damage Potential	Affected Users	Further Attack	Exploitability	Reproducibility	Discoverability	Time Consuming

Risk = (D+A+F)(E+R+D+T)*

And I assigned various weights for different parameters. In sum, total score for damage counts the same as total score for possibility. All parameters of possibility share same weights. Damage Potential and Further attack counts a little more than Affected Users, since range of affection in this situation is not that important due to the radio wave feature of wireless network.

Calculation for total score:
(3.5+3+3.5) * (2.5+2.5+2.5+2.5) = 100

Score Distribution of Severity Levels: S>A>B>C>D

0-19	D
20-39	C
40-59	B
60-79	A
80-100	S

3. Security configuration

The evaluation result is given by analyzing potential risks with each kind of threats for every certain security configuration. The security configuration consists of six key vectors, three from local device, “DisconnectionOnLogout”, “JoinMode/JoinModeFallback, “Root-Security”, and two from public access point, “SSID property”, “Authentication/Encryption”.

“DisconnectionOnLogout”: Some devices defaultly disable this configuration with concerns that achieves high efficiency and supports to some background applications. However, this could lead an extension of attacking vectors. Most directly it gives attackers more time and opportunity to exploit some vulnerabilities even if the user is not using the device while still staying under such Wi-Fi environment. Also, it weakens some local security tools that only gives defenses during user login.
Note This parameter is independent and has effects to any kind of threat model. So when rating the risks, I simply make it as a factor multiplied to the total score calculated from other configurations.(10% increase if not enabled)

JoinMode/JoinModeFallback(Auto-connection): By default, user’s devices always choose to automatically connect to available known access points when starting Wi-Fi function or suddenly losing previous Wi-Fi connection. They will keep a history table for previous associated access points and automatically send probe request and make further authentication/association steps with available ones in that table, to keep a reliable Wi-Fi connection. This setting, however, can be utilized by attackers to make attacks like fake AP, Evil Twins, etc.

Root-Security: This stands for the security mechanism used by user’s local devices, generally as root user-password based authentication/authorization. Sometimes, it may be unavoidable that attacker may take advantage of the Wi-Fi pure insecure nature to break the WLAN authentication or other layer-2 related security, however to prevent the devices from terrible damage, local security is the last safeguard preventing attackers from breach into local system and get root privilege to take overall control of the system. Here, I mainly focused on whether the local machine enables the SSH remote login function.(System password is hard to be analyzed via the program)

SSID property: Generally there are three situation for user’s customized SSID.

- Default SSID: Default SSID usually expose extra information about Wi-Fi access points’ producer and model, or even version of firmware, which will ease attacker’s attempts to find some known vulnerabilities related to certain access points.

- Public normal SSID: This is the general situation that the administrator changes the default SSID into some other names, this can be better than the former one, however essentially they are similar since device's mac address always contains information about the devices. Another situation could be the admin named access point with a meaningful content, which actually expose some private information like location, owner's personal information, etc, leading to potential risks like social engineering attacks.
- Hidden SSID: This is not normal for public wireless access points since it contradicts with the purpose that offering user's convenience when using the wireless network. However it can be a method to trivially reduce the attack vectors, since it increase the workload for attackers to discover and link the hidden access point with its other physical information.

Authentication/Encryption: Finally, the biggest part for evaluation is still about encryption and authentication. Some authentication protocols like EAP only associates with enterprise-level devices which is not very practical to discuss for public Wi-Fi. So basically I only categorized it here as:
Open/Open+RADIUS/WEP/WPA/WPA+WPA2/WPA2
in which Open+RADIUS is not easy to be detected, however for most threat models it has trivial difference with Open ones since RADIUS only improve the security at higher layers.

4. Detail Description for Calculated Risk Ratings.
The result of risk rating for all well known threat models were given in another document(spreadsheet) named "Threat Modeling".
The spreadsheet gives score for every threat model considering every possibility of security configuration of access point and user device. These data are the key elements of evaluation model, and will be combined together and merged into the program to give detailed ratings for the data sets captured by wifi scanning model.
This part will be discussed through all threat models, and will be presented with [Concern] and [Result] pairs that explains reason for every detail decision of scoring.

[Sniffing]
Generally, sniffing can be performed only using passive sniffer(wireshark, airodump-ng)

[Concern]
The direct damage from sniffing is trivial, especially for AP using WEP/WPA/WPA2. Open AP may leak some sensitive info or credentials from users.
[Result]

Damage Potential: 2/3.5 (OPEN), 1/3.5 (WEP/WPA/WPA2).

[Concern]
Sniffing is very easy to perform. Hidden ssid will make it a little inconvenient, but very trivial effects.
[Result]
For APs that are not hidden:
Exploitability: 2.5/2.5
Reproducibility: 2.5/2.5
Discoverability: 2.5/2.5
Time consuming: 2.5/2.5

[Concern]
Sniffing can contribute to some further attacks, WEP/WPA cracking, other attacks that However sniffing is almost the very first step of all other complex attacks, it's necessary but not that decisive.
[Result]
Further Attack: 1.5/3.5 Open, 1/3.5 Not Open

[Concern]
If SSID is hidden, attacker needs to firstly discover hidden ssid, then sniffing can be targeted to the clients. This is the reason that hidden SSID relatively decreases the risk.
Methods from attacker to handle with Hidden SSID:
1. Passively monitor clients connecting to AP.
2. De-auth clients and monitor reconnection.
[Result]
For models with *hidden SSID*, mainly affected parameters are:
Affected User: 2/3
(Users of hidden AP should be less than public ones, but among all its users, the attack can be targeted on any of them.)
Reproducibility: 2/2.5
(Passively monitor method has low efficiency and it depends on whether clients connect to the target AP during the exploit time, De-auth method also depends on client's device configuration, so this attack cannot guarantee to be succeed every time. However attacker still has a high possibility to get it)
Time Consuming: 2/2.5
(Because of the extra step for the attacker that to firstly discover the hidden SSID, time consuming should be longer than public APs, but still it's not a significant increase)

[Fake/Rogue Access Point]

When talking about fake access point as a threat, it only means that attacker creates a fake access point in the wave range and wait clients to connect to it without further achievements.

[Concern]
It has little relations with AP's configuration. User's setting that whether their devices connect APs automatically will somehow affect the ease for attackers.
[Result]
Not distinguish between different configuration of APs.

[Concern]
Fake AP itself can lead to little damage. It can affect all users in the range. It can used to perform Evil Twin, which is a severe threat.
[Result]
Damage Potential: 0.5/3.5
Affected Users: 3/3
Further Attacks: 2.5/3.5

[Concern]
Creating a fake AP is essentially easy but needs some knowledges and experiences of certain tools, like airbase-ng.
[Result]
Exploitability: 2.5/2.5
Reproducibility: 2/2.5
Discoverability: 2.5/2.5
Time Consuming: 2.5/2.5

[Concern]
A device that turn down the auto-connect configuration will make attacker harder to lure user to connect. It's exposed there, but only if the user himself/herself want to connect to it, does this threat succeed.
[Result]
With Auto-Connect configurations, results varies as follow:
Exploitability: 1.5/2.5, 2/2.5, 2.5/2.5
Time Consuming: 1.5/2.5, 2/2.5, 2.5/2.5

***Special Note:**
***
No more discussion about SSID because it's generally only related to sniffing steps.
The final result for each security setting will combine all possible threats' risk ratings.
***

[Break WLAN Authentication]
[Concern]
For open AP, authentication only has two packets, attacker can easily craft a fake auth packet to crack it. Open-Radius makes no big difference with open at this layer, attacker can easily associate with the AP, Radius only works in further network layer access control.
[Result]
For Open and Open+Radius
Exploitability: 2.5/2.5
Reproducibility: 2.5/2.5
Discoverability: 2.5/2.5
Time Consuming: 2.5/2.5

[Concern]
For shared key auth, more complex request/responses verification occurs during the authentication.
1. client -Auth request -> AP
2.AP -128 byte plaintext(challenge) -> client
3. client encrypts the challenge with IV, RC4 stream cipher, shared key, - encrypted challenge -> AP
4. AP verify the encrypted challenge, confirm/deny access - >client

Attacker need to firstly sniff enough packets and extract/compute the key stream using airodump-ng. Then use aireplay-ng to send fake auth packets based on key stream obtained.
Also, WPA/WPA2 make it much harder to crack than WEP
[Result]
For WEP:
Exploitability: 2/2.5
Reproducibility: 2/2.5
Discoverability: 2/2.5
Time Consuming: 2/2.5

For WPA/WPA2
Exploitability: 1/2.5
Reproducibility: 1/2.5
Discoverability: 1/2.5
Time Consuming: 1/2.5

[Concern]
Breaking WLAN Authentication leads to unauthorized user connect to AP. Which is actually fine in the context of public hotspots, since shared-key of public AP can always be easily propagated by users. However, a weak authentication will more likely attract attackers and give them convenience to do further harm.
Directly, this threat doesn't affect normal users and has little damage.
[Result]
Damage Potential: 0.5/3.5
Affected Users: 1/ 3

[Concern]
It can lead to further Damage: Attacker can try to (take control of) get access to AP settings. If SSID exposes the producer even type of AP obviously, it make attacker easier to find some general known vulnerabilities, ip address, default account/password, etc. However, the possibility of success cracking AP configuration and the damage it will lead to is limited.
[Result]
Further Attacks: 2/3.5, 2.5/3.5 if SSID exposes extra information about AP.

[Hotspot Evil Twin]
Steps to create a Evil Twin:
1. Create Fake AP(with higher radio strength)
2. Send De-authentication Packets
3. Monitoring ARP packets that informing client's IP address.
4. Change at0 into same subnet with client.
5. Get connected with client by network layer.
6. Further Attacks

Evil Twin by itself means to lure the clients to connect to the fake software based AP supposing it is the legitimate one they want to connect with, by standing beside the original one and interfere/attract user to connect.

[Concern]

It leads to damage like: User can be communicated by attacker via network layer, ping(DoS), ssh(remote access), wifi phishing, etc. The likelihood depends on user's local security settings(root password, ssh)
[Result]
Damage Potential is high, and determined by Root Privilege
3.5 / 3.5 with weak root security
2.5 / 3.5 with strong root security

[Concern]
It can also be used to contribute to MITM attack by relaying it to AP or directly to Internet.
[Result]
Further Attacks:
Generally it's high.
Specifically:
OPEN > WEP > WPA/WPA2
Weak Root > Strong Root

[Concern]
It is always targeted at certain client due to the step of de-authentication, but can be extended to affect more(de-authenticate all connect users).
[Result]
Affected Users: 2/3

[Concern]
This attack is a little complicate, not the one novel attacker can easily achieve. But with help of some tools, airodump-ng, aireplay-ng..., it can be learnt quickly.
It is easy to perform it with Open AP, but since attacker need to make the user thinking connected with the right one. There should be little difference between fake one and original one superficially, which also includes the authentication/encryption methods. A successful evil twin should let the user connect to it automatically, not simply a fake AP, so if original AP uses WEP/WPA/WPA-WPA2/WPA2, user's device will use the share-key to request connection. Attacker need to firstly crack the key, then craft a fake one with that key. Or accept any encrypted challenge from clients to confirm the successful authentication. Both ways need complex steps and time consuming. WPA/WPA2 are harder than WEP.
[Result]
Generally,
Exploitability and Time consuming are not extremely high.

Specifically,
For Exploitability, Reproducibility, Discoverability, Time Consuming
OPEN > WEP > WPA/WPA2
Auto-Connection > No Auto-Connection

[Cracking Shared Key]
[Concern]
This is easy to evaluation due to the different encryption algorithm and complexity
[Result]
The possibility for succeed such attacks:
OPEN>WEP > WPA>WPA/WPA2>WPA2

[MITM]
MITM attack can be easily performed when attacker successfully create an evil twin and lure clients connecting to it. The attacker vectors and related security configurations of AP/user device has no differences, so here I just combine its risk into Evil Twin, as part of potential further attack.

[Bypass MAC Filter]
No practical meaning for public hotspots

Note: More detailed scores are presented in the spreadsheet.