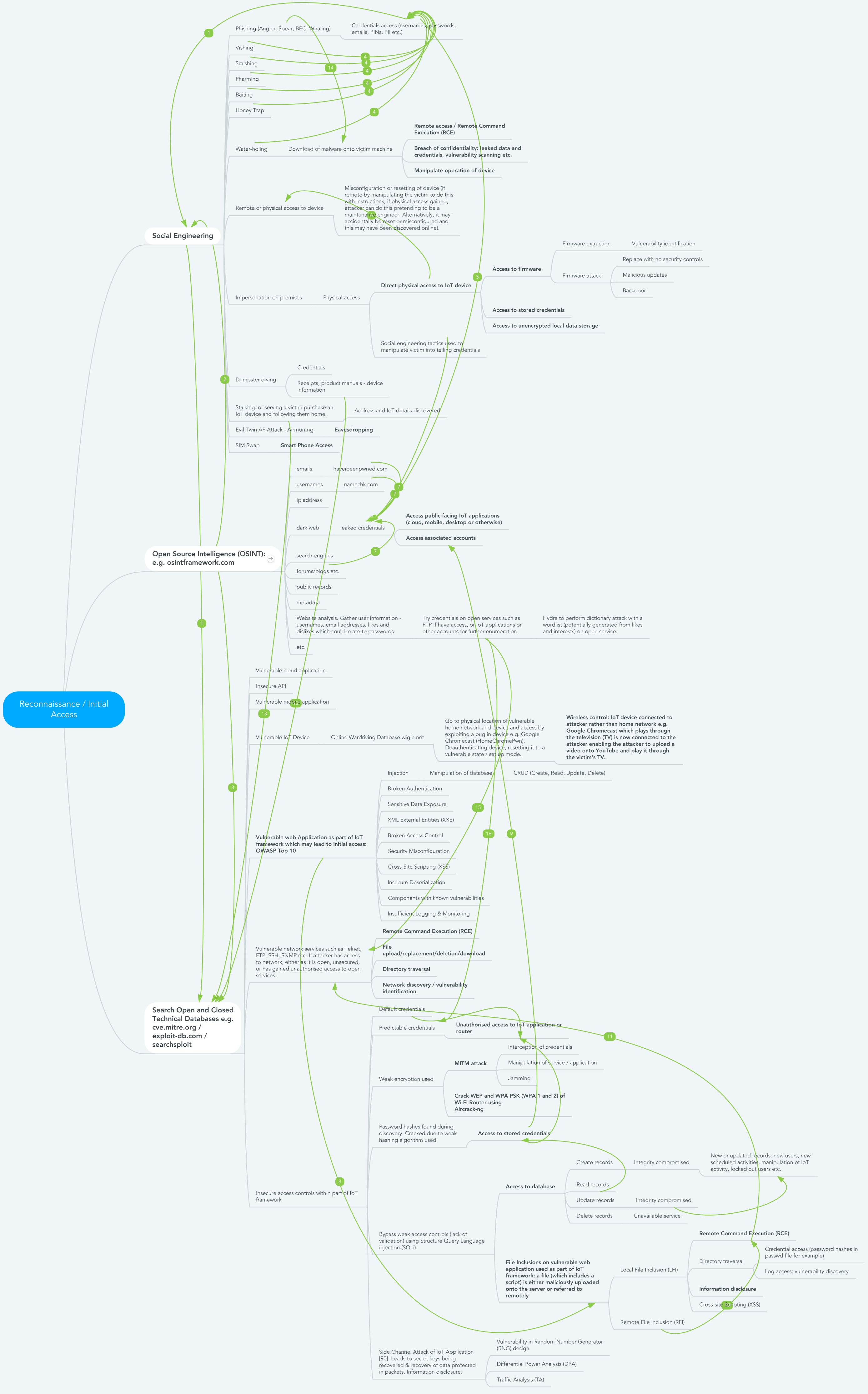

Reconnaissance / Initial Access

Social Engineering

Phishing (Angler, Spear, BEC, Whaling)

Credentials access (usernames, passwords, emails, PINs, PII etc.)

Vishing

Smishing

Pharming

Baiting

Honey Trap

Water-holing

Download of malware onto victim machine

**Remote access / Remote Command Execution (RCE)**

**Breach of confidentiality: leaked data and credentials, vulnerability scanning etc.**

**Manipulate operation of device**

Remote or physical access to device

Misconfiguration or resetting of device (if remote by manipulating the victim to do this with instructions, if physical access gained, attacker can do this pretending to be a maintenance engineer. Alternatively, it may accidentally be reset or misconfigured and this may have been discovered online).

Impersonation on premises

Physical access

**Direct physical access to IoT device**

**Access to firmware**

Firmware extraction

Vulnerability identification

Firmware attack

Replace with no security controls

Malicious updates

Backdoor

**Access to stored credentials**

**Access to unencrypted local data storage**

Social engineering tactics used to manipulate victim into telling credentials

Dumpster diving

Credentials

Receipts, product manuals - device information

Stalking: observing a victim purchase an IoT device and following them home.

Address and IoT details discovered

Evil Twin AP Attack - Airmon-ng

**Eavesdropping**

SIM Swap

**Smart Phone Access**

Open Source Intelligence (OSINT): e.g. osintframework.com

emails

haveibeenpwned.com

usernames

namechk.com

ip address

dark web

leaked credentials

**Access public facing IoT applications (cloud, mobile, desktop or otherwise)**

**Access associated accounts**

search engines

forums/blogs etc.

public records

metadata

Website analysis. Gather user information - usernames, email addresses, likes and dislikes which could relate to passwords

Try credentials on open services such as FTP if have access, or IoT applications or other accounts for further enumeration.

Hydra to perform dictionary attack with a wordlist (potentially generated from likes and interests) on open service.

etc.

Search Open and Closed Technical Databases e.g. cve.mitre.org / exploit-db.com / searchsploit

Vulnerable cloud application

Insecure API

Vulnerable mobile application

Vulnerable IoT Device

Online Wardriving Database wigle.net

Go to physical location of vulnerable home network and device and access by exploiting a bug in device e.g. Google Chromecast (HomeChromePwn). Deauthenticating device, resetting it to a vulnerable state / set up mode.

**Wireless control: IoT device connected to attacker rather than home network e.g. Google Chromecast which plays through the television (TV) is now connected to the attacker enabling the attacker to upload a video onto YouTube and play it through the victim's TV.**

**Vulnerable web Application as part of IoT framework which may lead to initial access: OWASP Top 10**

Injection

Manipulation of database

CRUD (Create, Read, Update, Delete)

Broken Authentication

Sensitive Data Exposure

XML External Entities (XXE)

Broken Access Control

Security Misconfiguration

Cross-Site Scripting (XSS)

Insecure Deserialization

Components with known vulnerabilities

Insufficient Logging & Monitoring

Vulnerable network services such as Telnet, FTP, SSH, SNMP etc. If attacker has access to network, either as it is open, unsecured, or has gained unauthorised access to open services.

**Remote Command Execution (RCE)**

**File upload/replacement/deletion/download**

**Directory traversal**

**Network discovery / vulnerability identification**

Insecure access controls within part of IoT framework

Default credentials

Predictable credentials

**Unauthorised access to IoT application or router**

Weak encryption used

**MITM attack**

Interception of credentials

Manipulation of service / application

Jamming

**Crack WEP and WPA PSK (WPA 1 and 2) of Wi-Fi Router using Aircrack-ng**

Password hashes found during discovery. Cracked due to weak hashing algorithm used

**Access to stored credentials**

Bypass weak access controls (lack of validation) using Structure Query Language injection (SQLi)

**Access to database**

Create records

Integrity compromised

New or updated records: new users, new scheduled activities, manipulation of IoT activity, locked out users etc.

Read records

Update records

Integrity compromised

Delete records

Unavailable service

**File Inclusions on vulnerable web application used as part of IoT framework: a file (which includes a script) is either maliciously uploaded onto the server or referred to remotely**

Local File Inclusion (LFI)

**Remote Command Execution (RCE)**

Directory traversal

Credential access (password hashes in passwd file for example)

Log access: vulnerability discovery

**Information disclosure**

Cross-site Scripting (XSS)

Remote File Inclusion (RFI)

Side Channel Attack of IoT Application [90]. Leads to secret keys being recovered & recovery of data protected in packets. Information disclosure.

Vulnerability in Random Number Generator (RNG) design

Differential Power Analysis (DPA)

Traffic Analysis (TA)

## Reconnaissance / Initial Access

### 1. Social Engineering

1.1. Phishing (Angler, Spear, BEC, Whaling)

1.1.1. Credentials access (usernames, passwords, emails, PINs, PII etc.)

1.2. Vishing

1.3. Smishing

1.4. Pharming

1.5. Baiting

1.6. Honey Trap

1.7. Water-holing

1.7.1. Download of malware onto victim machine

1.7.1.1. Remote access / Remote Command Execution (RCE)

1.7.1.2. Breach of confidentiality: leaked data and credentials, vulnerability scanning etc.

1.7.1.3. Manipulate operation of device

1.8. Remote or physical access to device

1.8.1. Misconfiguration or resetting of device (if remote by manipulating the victim to do this with instructions, if physical access gained, attacker can do this pretending to be a maintenance engineer. Alternatively, it may accidentally be reset or misconfigured and this may have been discovered online).

1.9. Impersonation on premises

1.9.1. Physical access

1.9.1.1. Direct physical access to IoT device

1.9.1.1.1. Access to firmware

1.9.1.1.1.1. Firmware extraction

1.9.1.1.1.1.1. Vulnerability identification

1.9.1.1.1.2. Firmware attack

1.9.1.1.1.2.1. Replace with no security controls

1.9.1.1.1.2.2. Malicious updates

1.9.1.1.1.2.3. Backdoor

1.9.1.1.2. Access to stored credentials

1.9.1.1.3. Access to unencrypted local data storage

1.9.1.2. Social engineering tactics used to manipulate victim into telling credentials

1.10. Dumpster diving

1.10.1. Credentials

1.10.2. Receipts, product manuals - device information

1.11. Stalking: observing a victim purchase an IoT device and following them home.

1.11.1. Address and IoT details discovered

1.12. Evil Twin AP Attack - Airmon-ng

1.12.1. Eavesdropping

1.13. SIM Swap

1.13.1. Smart Phone Access

## 2. Open Source Intelligence (OSINT): e.g. osintframework.com

**Link:** https://osintframework.com/

2.1. emails

2.1.1. haveibeenpwned.com

2.2. usernames

  2.2.1. namechk.com

2.3. ip address

2.4. dark web

  2.4.1. leaked credentials

    2.4.1.1. Access public facing IoT applications (cloud, mobile, desktop or otherwise)

    2.4.1.2. Access associated accounts

2.5. search engines

2.6. forums/blogs etc.

2.7. public records

2.8. metadata

2.9. Website analysis. Gather user information - usernames, email addresses, likes and dislikes which could relate to passwords

  2.9.1. Try credentials on open services such as FTP if have access, or IoT applications or other accounts for further enumeration.

    2.9.1.1. Hydra to perform dictionary attack with a wordlist (potentially generated from likes and interests) on open service.

2.10. etc.

## 3. Search Open and Closed Technical Databases e.g. cve.mitre.org / exploit-db.com / searchsploit

3.1. Vulnerable cloud application

3.2. Insecure API

3.3. Vulnerable mobile application

3.4. Vulnerable IoT Device

3.4.1. Online Wardriving Database wigle.net

3.4.1.1. Go to physical location of vulnerable home network and device and access by exploiting a bug in device e.g. Google Chromecast (HomeChromePwn). Deauthenticating device, resetting it to a vulnerable state / set up mode.

3.4.1.1.1. Wireless control: IoT device connected to attacker rather than home network e.g. Google Chromecast which plays through the television (TV) is now connected to the attacker enabling the attacker to upload a video onto YouTube and play it through the victim's TV.

3.5. Vulnerable web Application as part of IoT framework which may lead to initial access: OWASP Top 10

3.5.1. Injection

3.5.1.1. Manipulation of database

3.5.1.1.1. CRUD (Create, Read, Update, Delete)

3.5.2. Broken Authentication

3.5.3. Sensitive Data Exposure

3.5.4. XML External Entities (XXE)

3.5.5. Broken Access Control

3.5.6. Security Misconfiguration

3.5.7. Cross-Site Scripting (XSS)

3.5.8. Insecure Deserialization

3.5.9. Components with known vulnerabilities

3.5.10. Insufficient Logging & Monitoring

3.6. Vulnerable network services such as Telnet, FTP, SSH, SNMP etc. If attacker has access to network, either as it is open, unsecured, or has gained unauthorised access to open services.

- 3.6.1. Remote Command Execution (RCE)
- 3.6.2. File upload/replacement/deletion/download
- 3.6.3. Directory traversal
- 3.6.4. Network discovery / vulnerability identification

3.7. Insecure access controls within part of IoT framework

- 3.7.1. Default credentials
- 3.7.2. Predictable credentials
    - 3.7.2.1. Unauthorised access to IoT application or router
- 3.7.3. Weak encryption used
    - 3.7.3.1. MITM attack
        - 3.7.3.1.1. Interception of credentials
        - 3.7.3.1.2. Manipulation of service / application
        - 3.7.3.1.3. Jamming
    - 3.7.3.2. Crack WEP and WPA PSK (WPA 1 and 2) of Wi-Fi Router using Aircrack-ng
- 3.7.4. Password hashes found during discovery. Cracked due to weak hashing algorithm used
    - 3.7.4.1. Access to stored credentials
- 3.7.5. Bypass weak access controls (lack of validation) using Structure Query Language injection (SQLi)
    - 3.7.5.1. Access to database
        - 3.7.5.1.1. Create records
            - 3.7.5.1.1.1. Integrity compromised

3.7.5.1.1.1.1. New or updated records: new users, new scheduled activities, manipulation of IoT activity, locked out users etc.

3.7.5.1.2. Read records

3.7.5.1.3. Update records

3.7.5.1.3.1. Integrity compromised

3.7.5.1.4. Delete records

3.7.5.1.4.1. Unavailable service

3.7.5.2. File Inclusions on vulnerable web application used as part of IoT framework: a file (which includes a script) is either maliciously uploaded onto the server or referred to remotely

3.7.5.2.1. Local File Inclusion (LFI)

3.7.5.2.1.1. Remote Command Execution (RCE)

3.7.5.2.1.2. Directory traversal

3.7.5.2.1.2.1. Credential access (password hashes in passwd file for example)

3.7.5.2.1.2.2. Log access: vulnerability discovery

3.7.5.2.1.3. Information disclosure

3.7.5.2.1.4. Cross-site Scripting (XSS)

3.7.5.2.2. Remote File Inclusion (RFI)

3.7.6. Side Channel Attack of IoT Application [90]. Leads to secret keys being recovered & recovery of data protected in packets. Information disclosure.

3.7.6.1. Vulnerability in Random Number Generator (RNG) design

#### 3.7.6.2. Differential Power Analysis (DPA)

#### 3.7.6.3. Traffic Analysis (TA)