Analysis Notes (1)

A Grounded Theory Based Approach to Characterize Software Attack Surfaces

Memo#1: Initial Concepts Emerged from Open Codes for Entry Points

Description:
This memo shows the emergence of concepts from open codes and constant comparison processes for defining Entry Points (where).

Targets are software components or parts of software applications that attackers try to access.

1. Operating system commands (system calls)
2. Software application files: lock files, tmp files,
3. System files: node catalog in distributed systems
4. HTML/Webscripts
5. Memory allocation/deallocation/tampering/access: socket buffer, kernel memory, kernel stack memory, loops counting buffer size, uninitialized memory, check boundary
6. Marshalling/unmarshalling data objects to/from json: deserializing class, deserializing polymorphic class
7. Android activity
8. Type casting parts
9. Executable code
10. Database
11. Software configuration parts
12. Special Objects or Classes: Cryptographic objects, Gadget Classes
13. Sensitive/private information: Credentials, userdata, metadata

Memo#2: Initial Concepts Emerged from Open Codes for Targets

Description:
This memo shows the emergence of concepts from open codes and constant comparison processes for defining Targets (what).

Entry points are parts of software system that attacker can leverage to access targets.

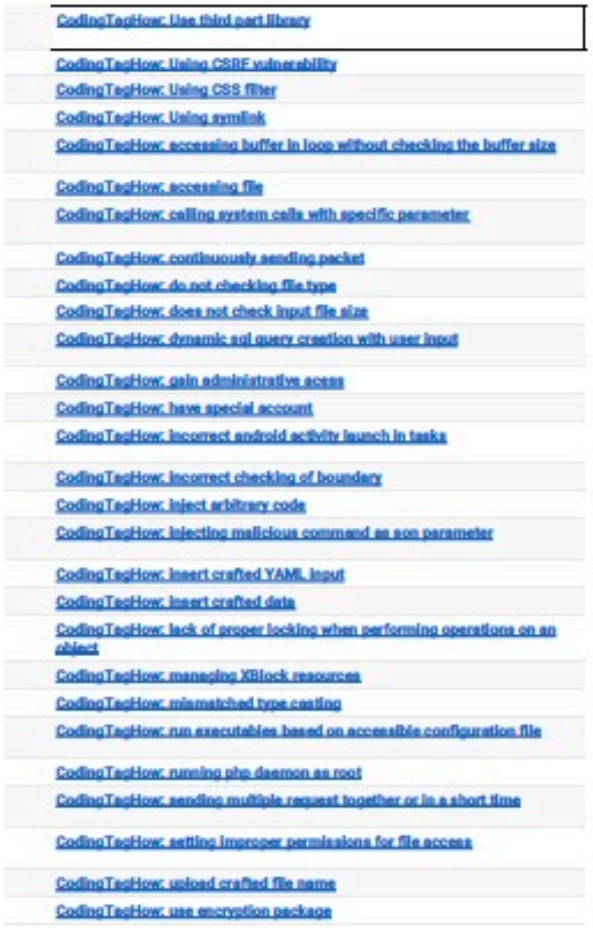
1. System files: account information file (etc/passwd),
2. dll files, eds files
3. Command line arguments
4. Web requests: http post request, http get request, files in post requests
5. Packets: IPv6 packets
6. Network sockets
7. REST APIs
8. Device related arguments
9. Service requests: such as inter procedural communication

Design-level (system sub-modules):

1. Application Configuration
2. File system handling
3. Installer components
4. Update component
5. Editor
6. Chat
7. User Console
8. Web console
9. Plugin administration
10. Port management interface
11. File upload
12. Access to Local system

Memo#3: Initial Concepts Emerged from Open Codes for Mechanisms

Description:
This memo shows the emergence of concepts from open codes and constant comparison processes for defining Mechanisms (How).



1. Using third party library: encryption package
2. Improper permission: for accessing files, running scripts
3. Run executables based on accessible configuration files
4. Mismatched type casting
5. Dynamic SQL query creation
6. Do not checking input file: type, size
7. Using dangerous technology: CSS filters, symlinks
8. Using incorrect (unsafe) technology: http instead of https
9. Unauthenticated access to account or services
10. Sending multiple requests or packets together in short time

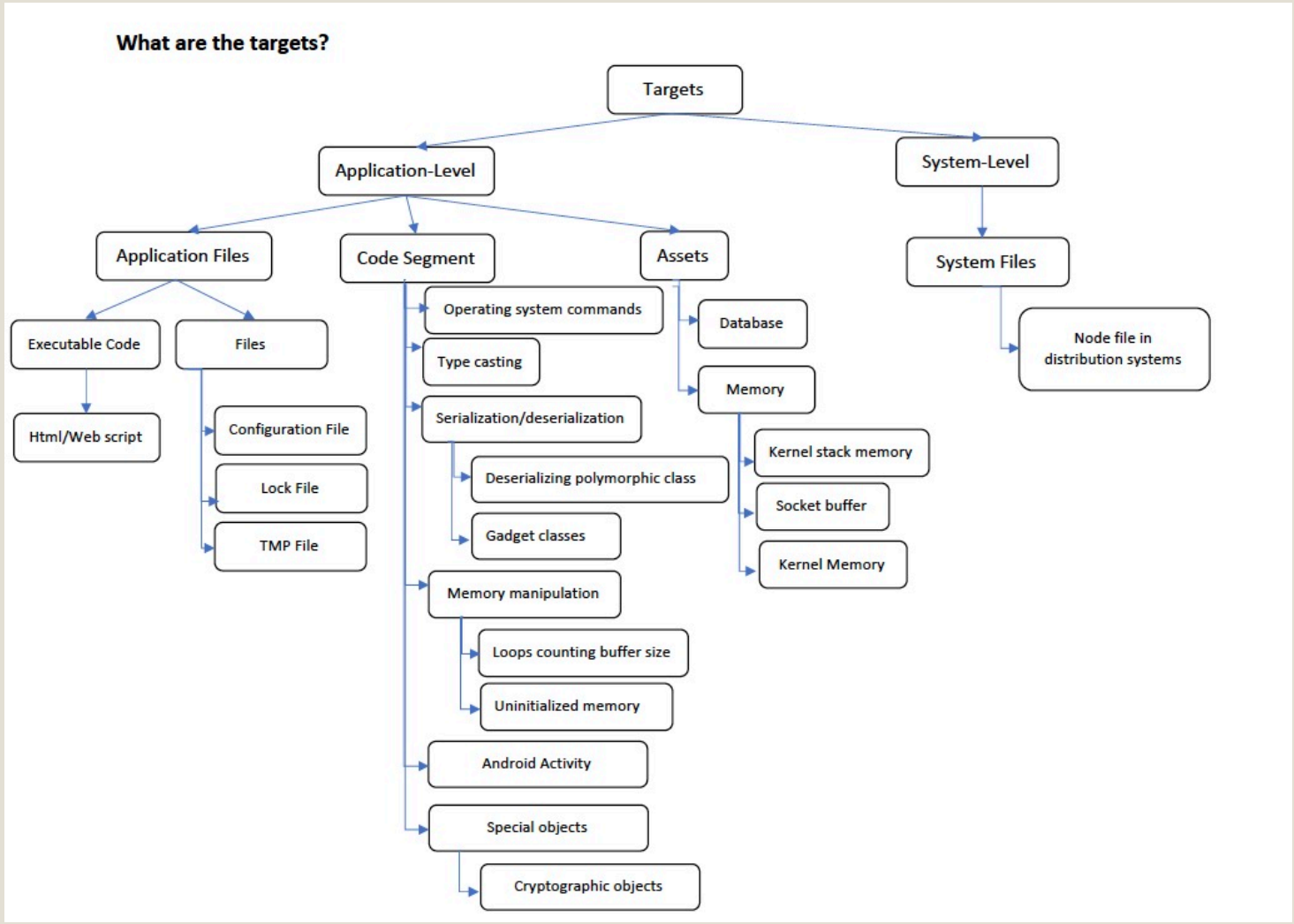
Memo#4: Input Data Types Codes

Description:
This memo shows types of input data that are identified during coding process.

Data Type	CVE_ID	Note
Tiff file	CVE-2020-6067	Out-of-bounds Write
XML	CVE-2020-6238	CWE-20Improper Input Validation
BLOB A binary large object (blob) is concentrated binary data that's compressed into an individual file inside a database. The large size of the file means they need special storage treatment. Blobs are binary, which means they are usually images, audio or other media.	CVE-2020-7248	Out-of-bounds Write
YAML It's basically a human-readable structured data format. It is less complex and ungainly than XML or JSON, but provides similar capabilities. It essentially allows you to provide powerful configuration settings, without having to learn a more complex code type like CSS, JavaScript, and PHP.	CVE-2020-1947	CWE-502-Deserialization of Untrusted Data
Android Parcel Android Parcel would be that of a message container for lightweight, high-performance Inter-process communication (IPC). .	CVE-2020-0017	CWE-200- Exposure of Sensitive Information to an Unauthorized Actor
JSON	CVE-2019-10749 CVE-2019-10748	CWE-89- Improper Neutralization of Special Elements used in an SQL Command ('SQL Injection')
	CVE-2018-7489	CWE-184- Incomplete List of Disallowed Inputs
	CVE-2018-18836	CWE-74- Improper Neutralization of Special Elements in Output Used by a Downstream Component ('Injection')
	CVE-2017-18349	CWE20-Improper Input Validation
	CVE-2017-17485	CWE-502- Deserialization of Untrusted Data
	CVE-2014-5017	CWE-89- Improper Neutralization of Special Elements used in an SQL Command ('SQL Injection')
	CVE-2014-3994	CWE-79- Improper Neutralization of Input During Web Page Generation ('Cross-site Scripting')
IPv4 packet	CVE-2020-1638	CWE20-Improper Input Validation

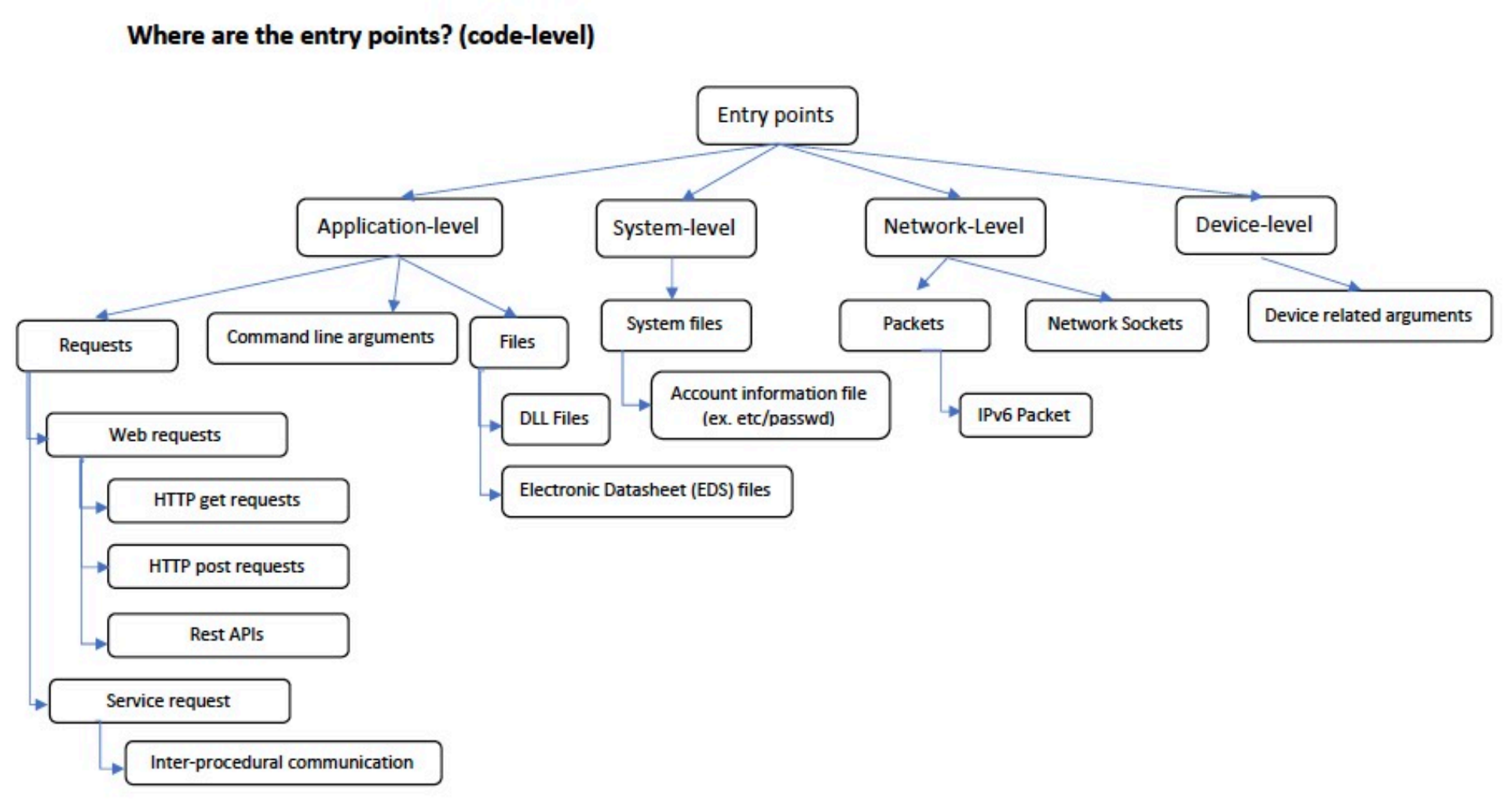
Memo#5: Initial Axial Codes and Categories for Entry Points

Description:
This memo shows the relationship between concepts emerged and preliminary categories for Entry Points.



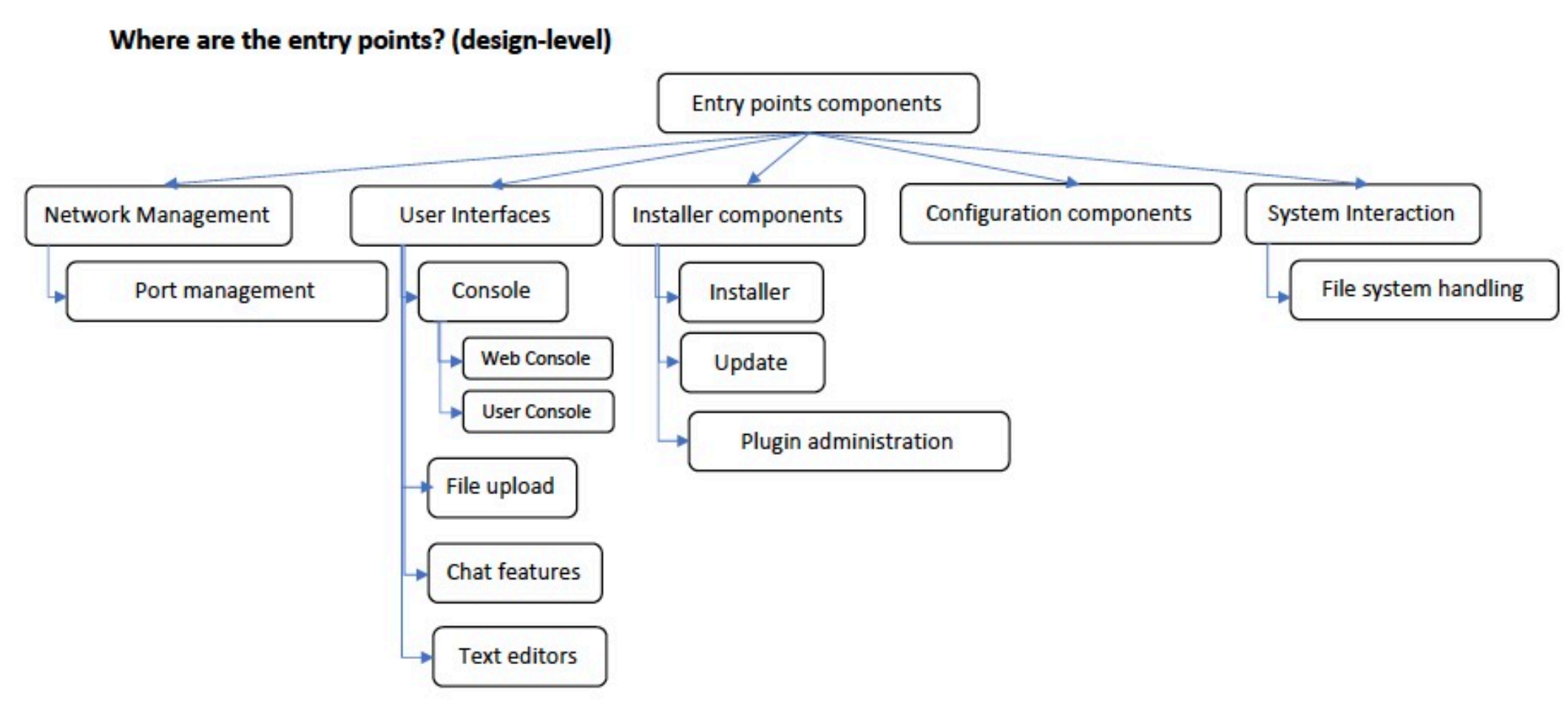
Memo#6: Initial Axial Codes and Categories for Targets

Description:
This memo shows the relationship between concepts emerged and preliminary categories for Targets.



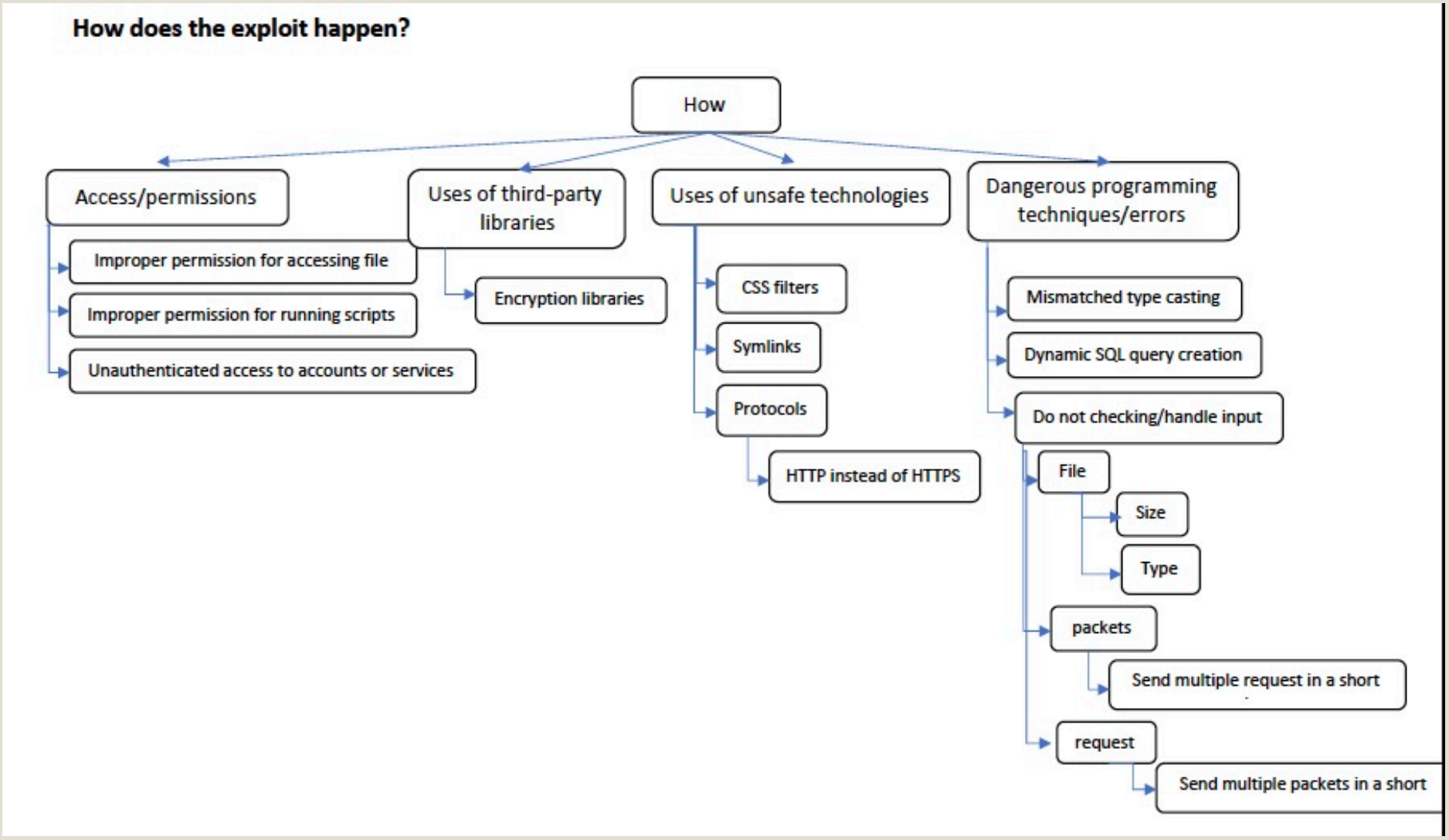
Memo#7: Initial Axial Codes and Categories for Targets (Design-Level)

Description:
This memo shows the relationship between concepts emerged and preliminary categories for Targets (design-level).



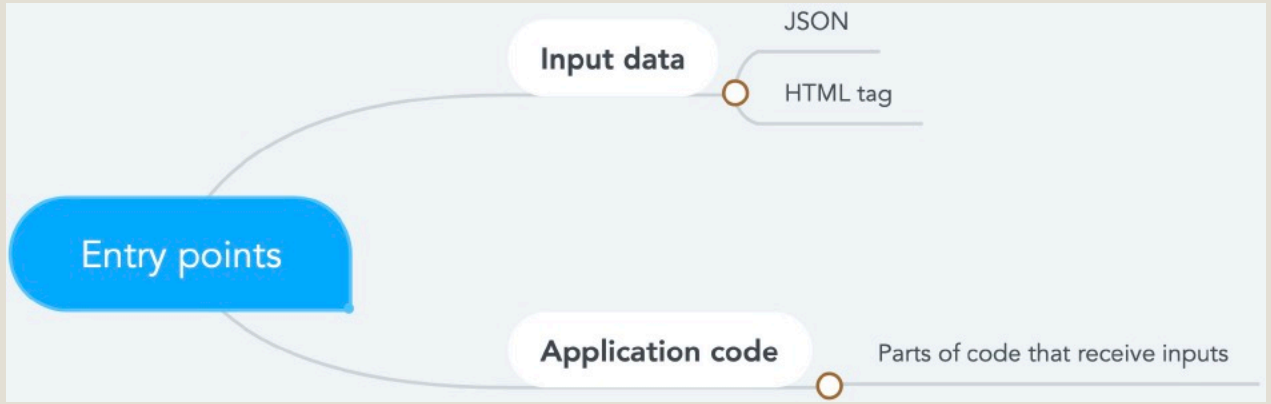
Memo#8: Initial Axial Codes and Categories for Mechanisms

Description:
This memo shows the relationship between concepts emerged and preliminary categories for Mechanisms (how).



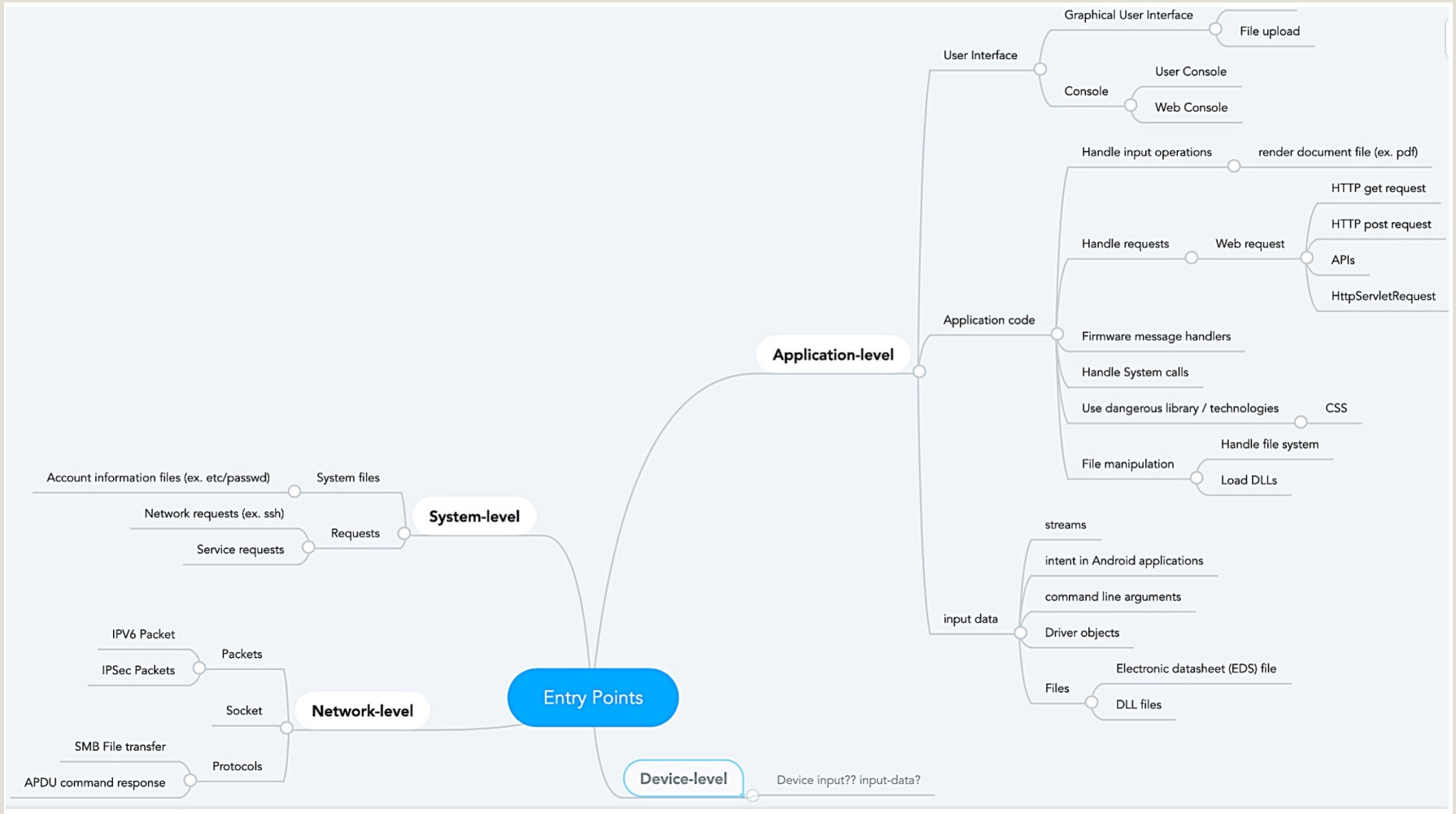
Memo#9: Mindmap Sample for Entry Points (Axial Codes)

Description:
This mindmap shows a sample of the initial relationships between open codes and categories emerged for Entry Points(where).



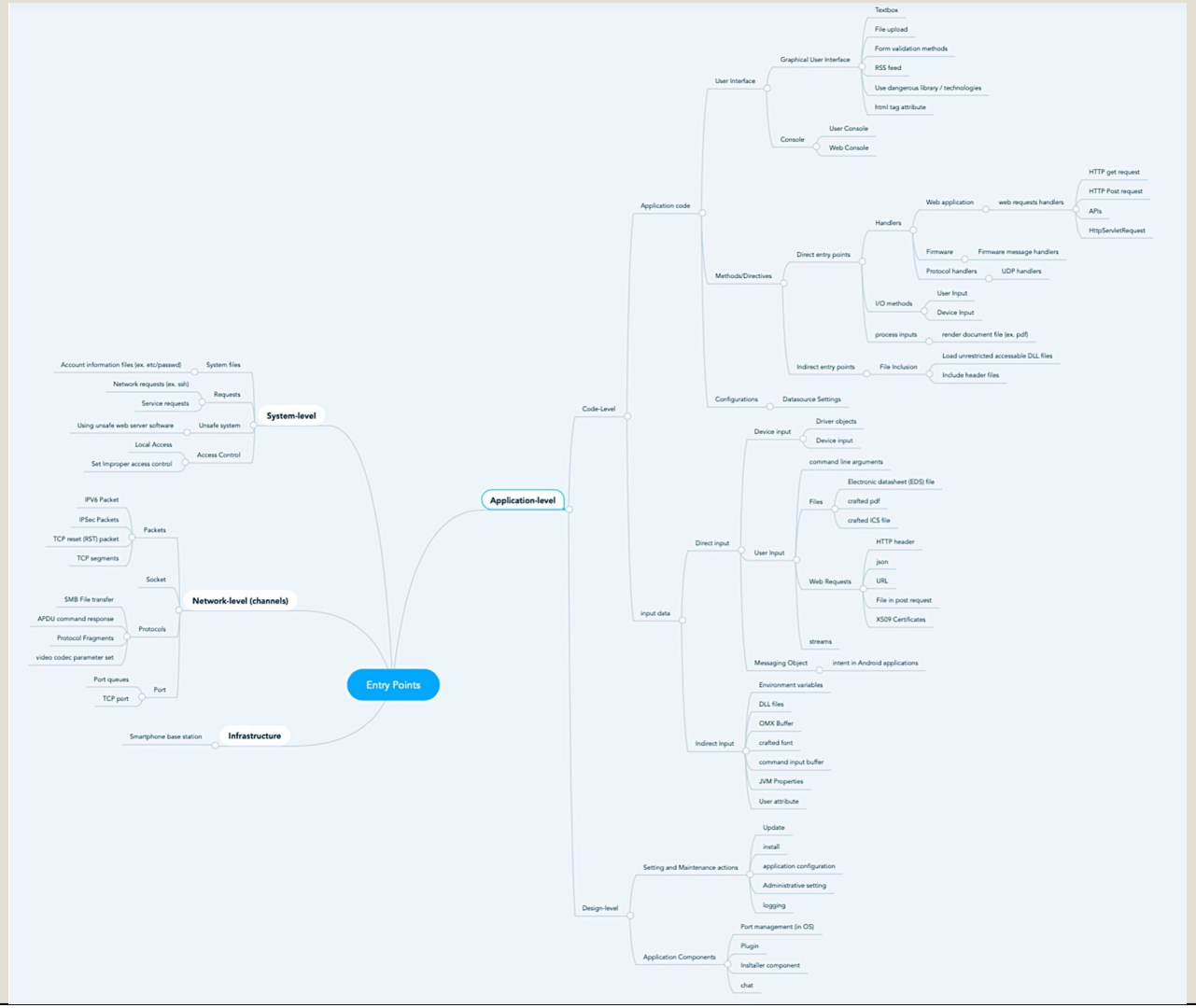
Memo#10: Mindmap 2 for Entry Points (Axial Codes)

Description:
This mindmap shows the relationship between open codes and categories (axial codes) emerged for Entry Points(where) at the middle stages of coding process.



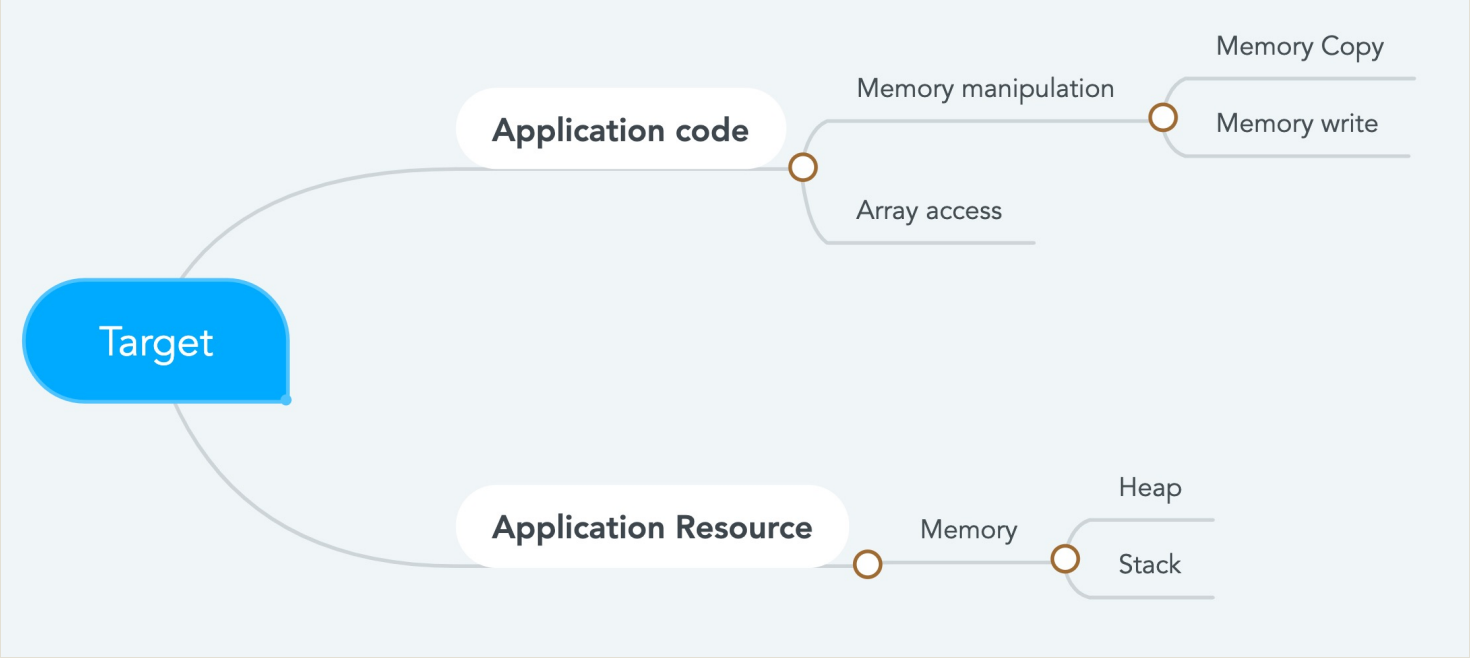
Memo#11: Mindmap 3 for Entry Points (Axial Codes)

Description:
This mindmap shows the relationship between open codes and categories (axial codes) emerged for Entry Points(what) at the middle stages of coding process.



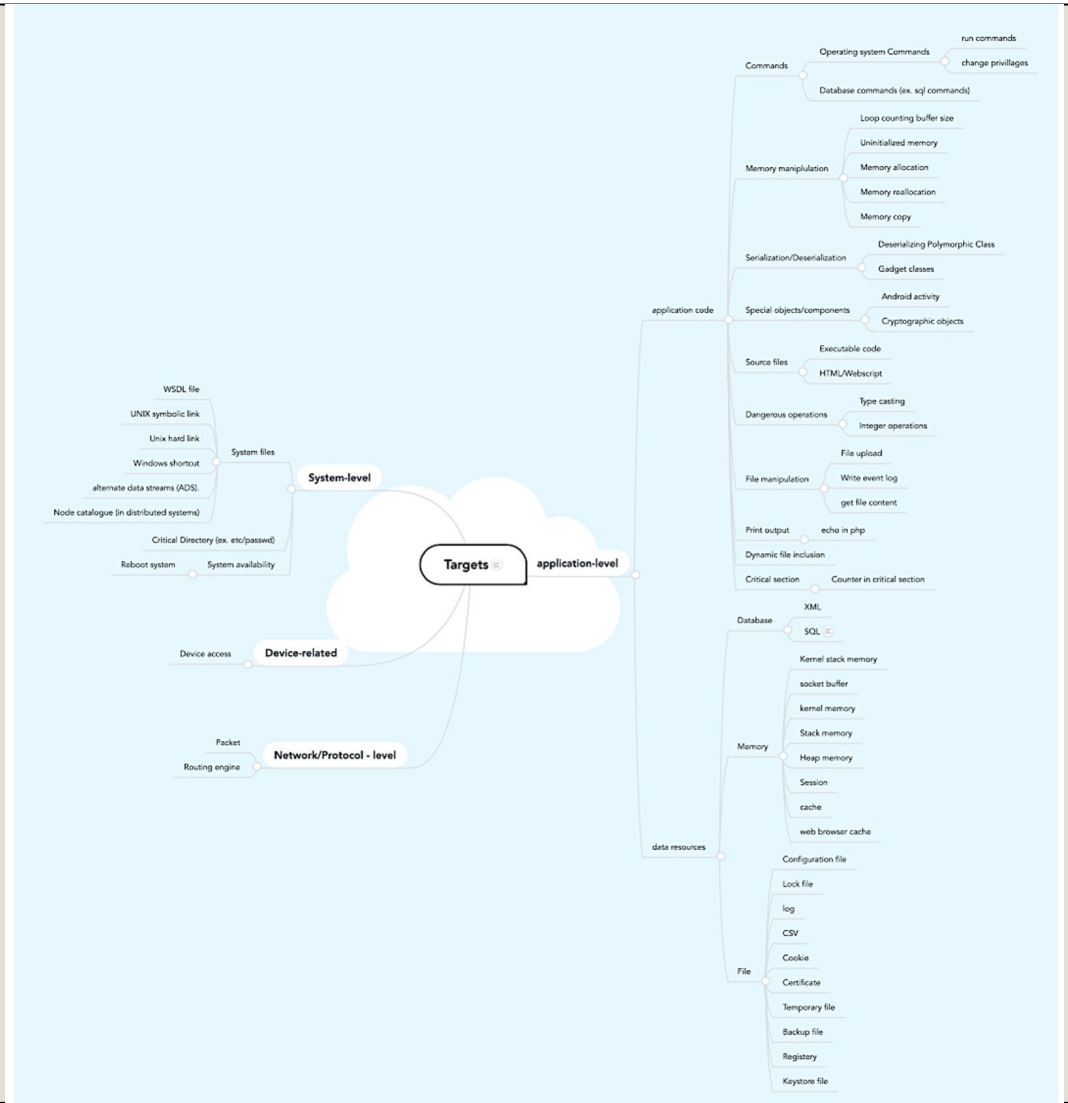
Memo#12: Mindmap Sample for Targets (Axial Codes)

Description:
This mindmap shows a sample of the initial relationships between open codes and categories emerged for Targets (what).



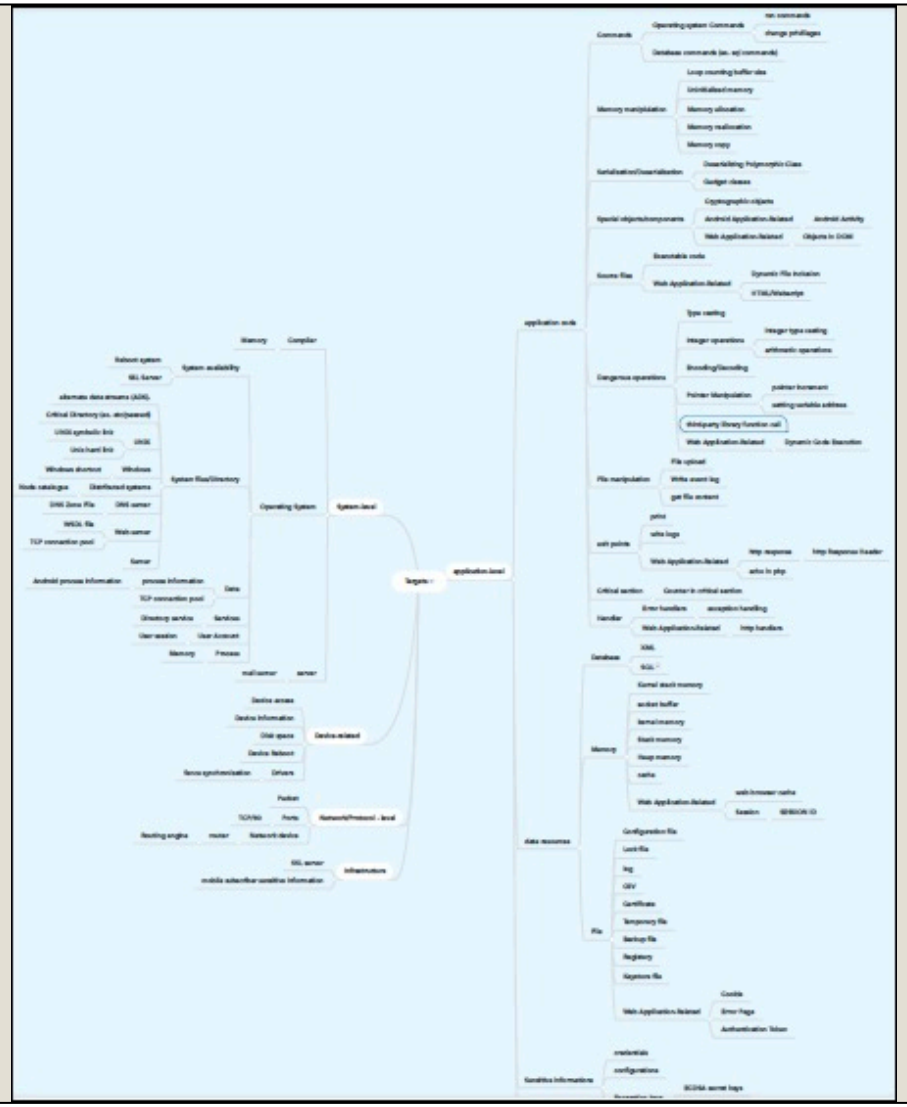
Memo#13: Mindmap 2 for Targets (Axial Codes)

Description:
This mindmap shows the relationship between open codes and categories (axial codes) emerged for Targets (what) at the middle stages of coding process.



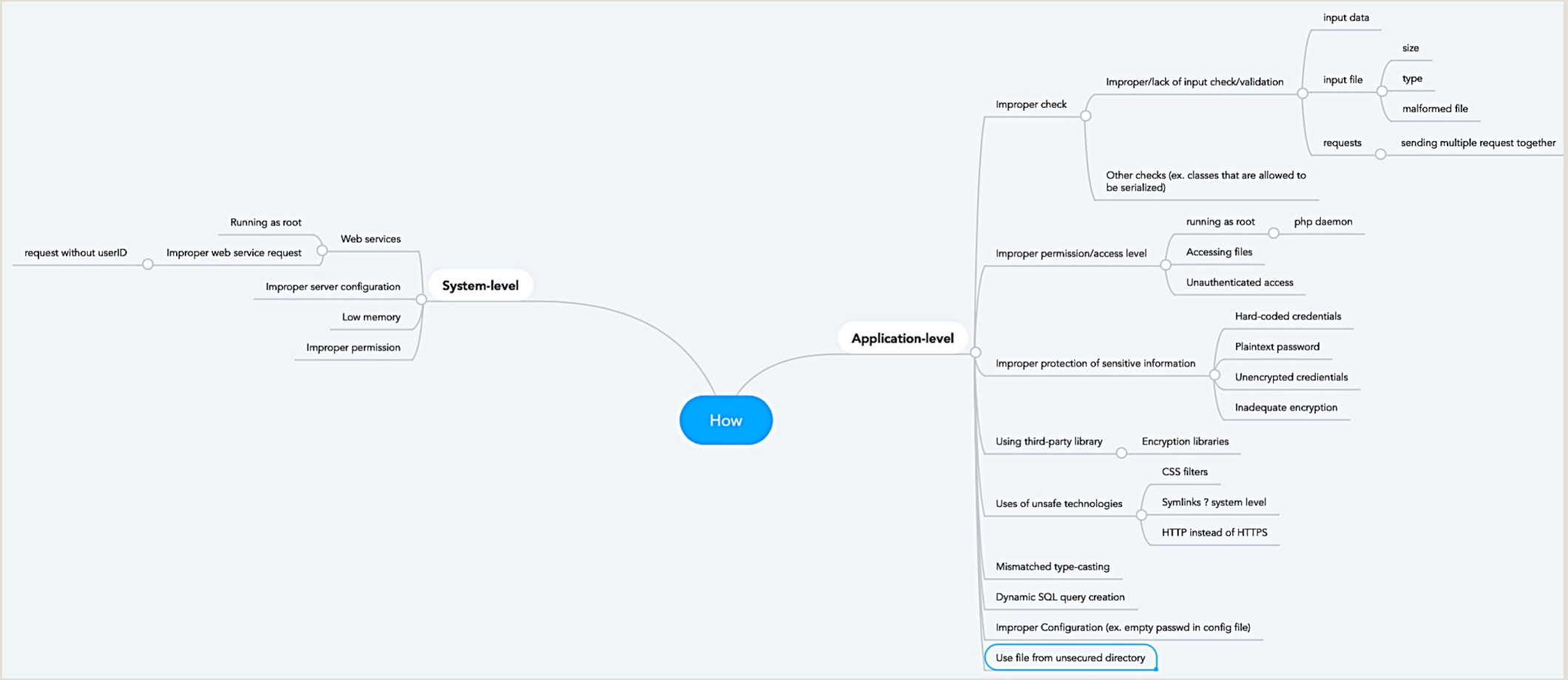
Memo#14: Mindmap 3 for Targets (Axial Codes)

Description:
This mindmap shows the relationship between open codes and categories (axial codes) emerged for Targets (what) at the middle stages of coding process.



Memo#15: Mindmap 1 for Mechanisms (Axial Codes)

Description:
This mindmap shows the relationship between open codes and categories (axial codes) emerged for Mechanisms (how) at the middle stages of coding process.



Memo#16: Mindmap 2 for Mechanisms (Axial Codes)

Description:
This mindmap shows the relationship between open codes and categories (axial codes) emerged for Mechanisms (how) at the middle stages of coding process.