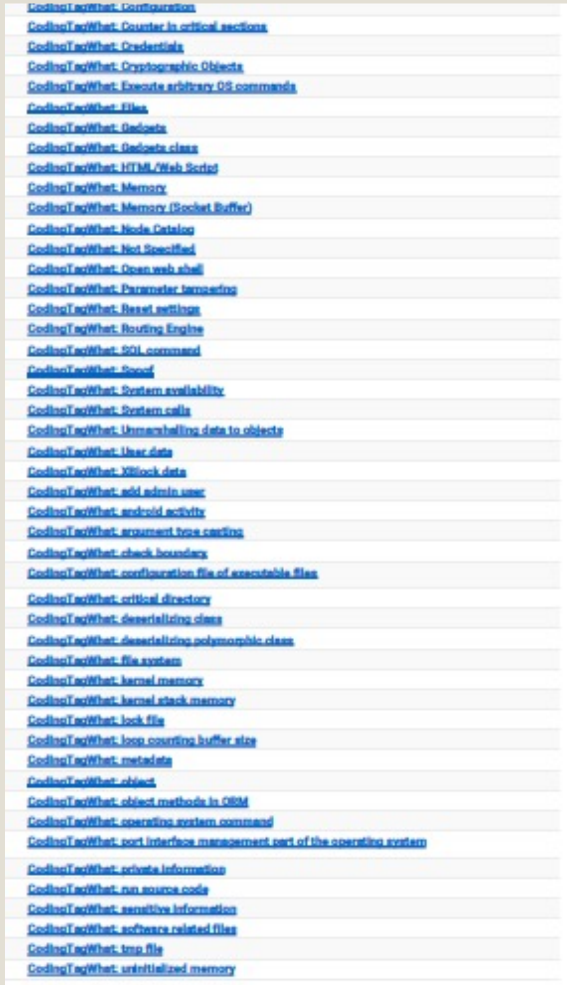
Analysis Notes (1)

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).



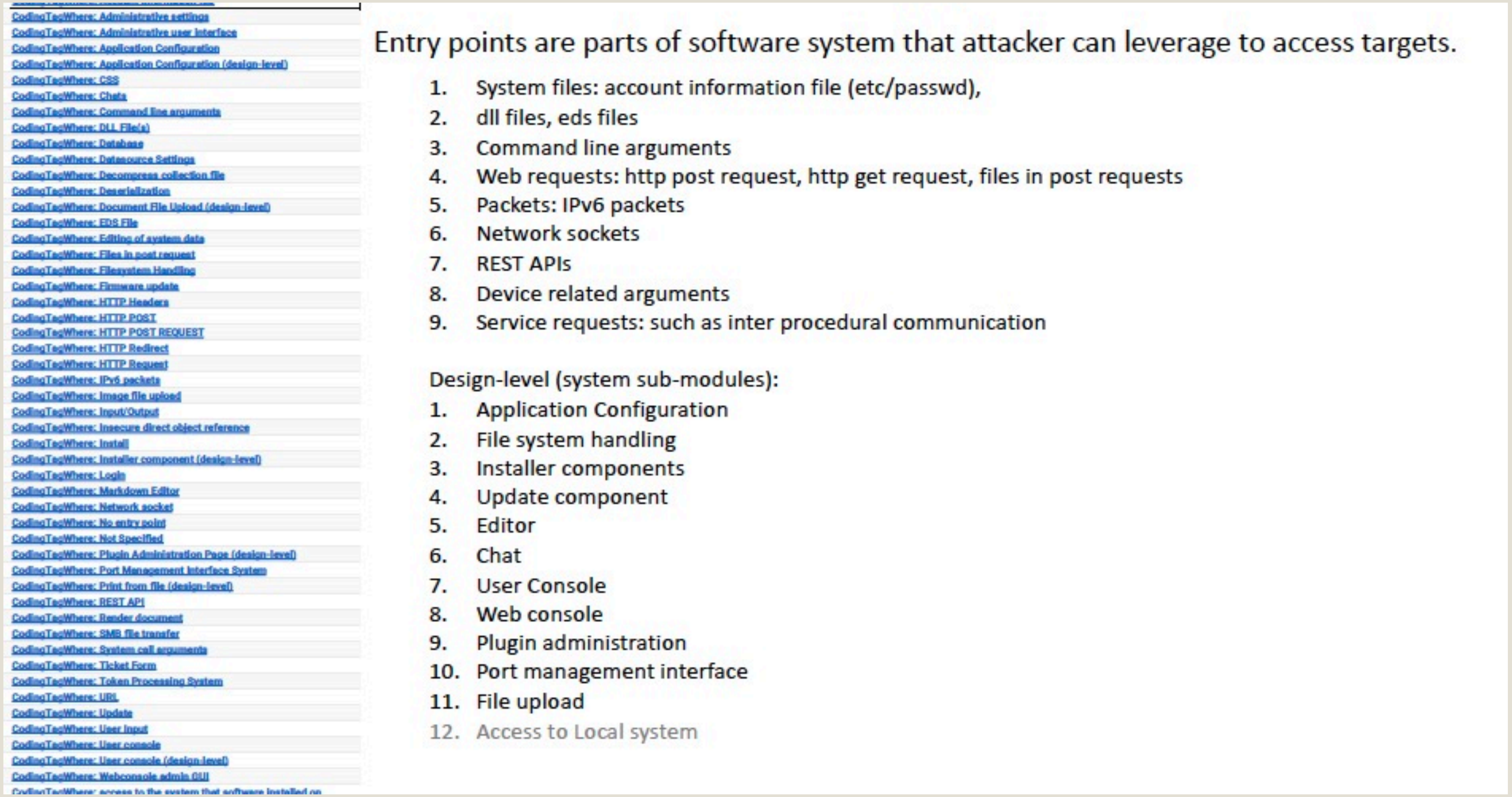
CodingTagWhat: Configuration
CodingTagWhat: Counter in critical sections
CodingTagWhat: Credentials
CodingTagWhat: Cryptographic Objects
CodingTagWhat: Execute arbitrary OS commands
CodingTagWhat: Files
CodingTagWhat: Gadgets
CodingTagWhat: Gadgets class
CodingTagWhat: HTML/Web Script
CodingTagWhat: Memory
CodingTagWhat: Memory (Socket Buffer)
CodingTagWhat: Node Catalog
CodingTagWhat: Not Specified
CodingTagWhat: Open web shell
CodingTagWhat: Parameter tampering
CodingTagWhat: Reset settings
CodingTagWhat: Routing Engine
CodingTagWhat: SQL command
CodingTagWhat: Spoof
CodingTagWhat: System availability
CodingTagWhat: System calls
CodingTagWhat: Unmarshalling data to objects
CodingTagWhat: User data
CodingTagWhat: XBlock data
CodingTagWhat: add admin user
CodingTagWhat: android activity
CodingTagWhat: argument type casting
CodingTagWhat: check boundary
CodingTagWhat: configuration file of executable files
CodingTagWhat: critical directory
CodingTagWhat: deserializing class
CodingTagWhat: deserializing polymorphic class
CodingTagWhat: file system
CodingTagWhat: kernel memory
CodingTagWhat: kernel stack memory
CodingTagWhat: lock file
CodingTagWhat: loop counting buffer size
CodingTagWhat: metadata
CodingTagWhat: object
CodingTagWhat: object methods in ORM
CodingTagWhat: operating system command
CodingTagWhat: port interface management part of the operating system
CodingTagWhat: private information
CodingTagWhat: run source code
CodingTagWhat: sensitive information
CodingTagWhat: software related files
CodingTagWhat: tmp file
CodingTagWhat: uninitialized memory

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).



CodingTagWhere: Administrative settings
CodingTagWhere: Administrative user interface
CodingTagWhere: Application Configuration
CodingTagWhere: Application Configuration (design-level)
CodingTagWhere: CSS
CodingTagWhere: Chats
CodingTagWhere: Command line arguments
CodingTagWhere: DLL File(s)
CodingTagWhere: Database
CodingTagWhere: Datasource Settings
CodingTagWhere: Decompress collection file
CodingTagWhere: Deserialization
CodingTagWhere: Document File Upload (design-level)
CodingTagWhere: EDS File
CodingTagWhere: Editing of system data
CodingTagWhere: Files in post request
CodingTagWhere: Filesystem Handling
CodingTagWhere: Firmware update
CodingTagWhere: HTTP Headers
CodingTagWhere: HTTP POST
CodingTagWhere: HTTP POST REQUEST
CodingTagWhere: HTTP Redirect
CodingTagWhere: HTTP Request
CodingTagWhere: IPv6 packets
CodingTagWhere: Image file upload
CodingTagWhere: Input/Output
CodingTagWhere: Insecure direct object reference
CodingTagWhere: Install
CodingTagWhere: Installer component (design-level)
CodingTagWhere: Login
CodingTagWhere: Markdown Editor
CodingTagWhere: Network socket
CodingTagWhere: No entry point
CodingTagWhere: Not Specified
CodingTagWhere: Plugin Administration Page (design-level)
CodingTagWhere: Port Management Interface System
CodingTagWhere: Print from file (design-level)
CodingTagWhere: REST API
CodingTagWhere: Render document
CodingTagWhere: SMB file transfer
CodingTagWhere: System call arguments
CodingTagWhere: Ticket Form
CodingTagWhere: Token Processing System
CodingTagWhere: URL
CodingTagWhere: Update
CodingTagWhere: User Input
CodingTagWhere: User console
CodingTagWhere: User console (design-level)
CodingTagWhere: Webconsole admin GUI
CodingTagWhere: access to the system that software installed on

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

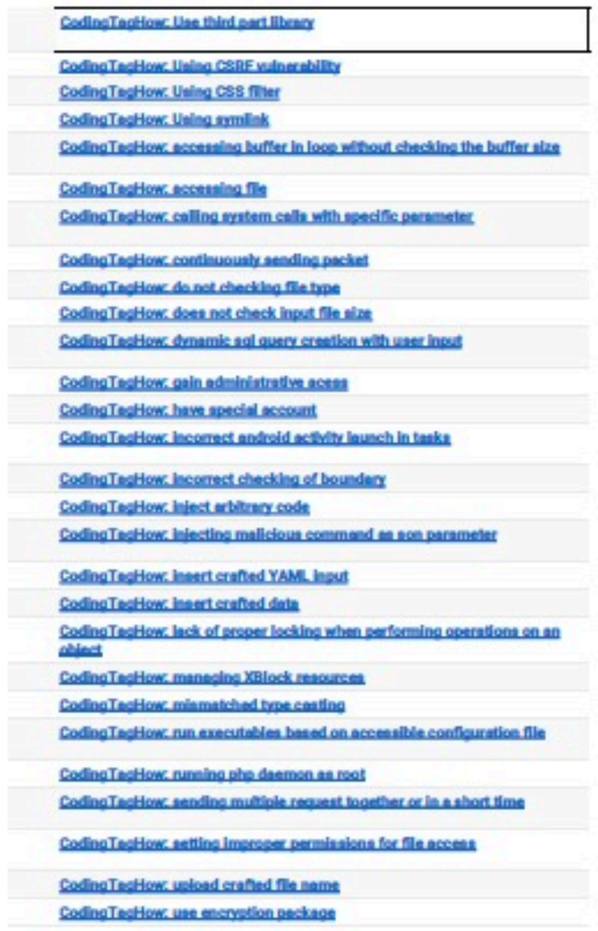
1. System files: account information file (etc/passwd),
2. dll files, eds files
3. Command line arguments
4. Web requests: http post request, http get request, files in post requests
5. Packets: IPv6 packets
6. Network sockets
7. REST APIs
8. Device related arguments
9. Service requests: such as inter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



CodingTagHow: Use third part library
CodingTagHow: Using CSRF vulnerability
CodingTagHow: Using CSS filter
CodingTagHow: Using symlink
CodingTagHow: accessing buffer in loop without checking the buffer size
CodingTagHow: accessing file
CodingTagHow: calling system calls with specific parameter
CodingTagHow: continuously sending packet
CodingTagHow: do not checking file type
CodingTagHow: does not check input file size
CodingTagHow: dynamic sql query creation with user input
CodingTagHow: gain administrative acess
CodingTagHow: have special account
CodingTagHow: incorrect android activity launch in tasks
CodingTagHow: incorrect checking of boundary
CodingTagHow: inject arbitrary code
CodingTagHow: injecting malicious command as son parameter
CodingTagHow: insert crafted YAML input
CodingTagHow: insert crafted data
CodingTagHow: lack of proper locking when performing operations on an object
CodingTagHow: managing XBlock resources
CodingTagHow: mismatched type casting
CodingTagHow: run executables based on accessible configuration file
CodingTagHow: running php daemon as root
CodingTagHow: sending multiple request together or in a short time
CodingTagHow: setting improper permissions for file access
CodingTagHow: upload crafted file name
CodingTagHow: use encryption package

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

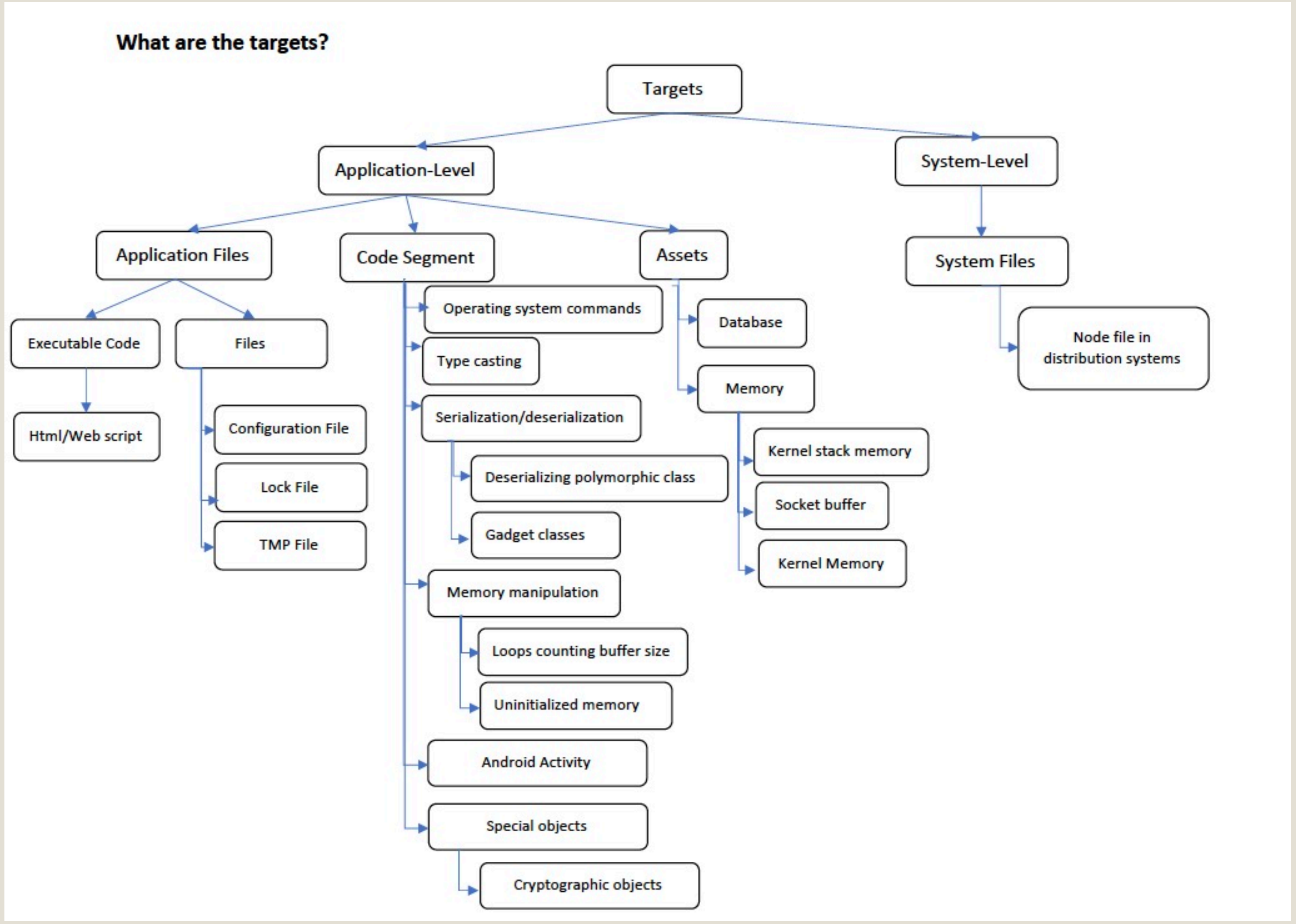
Description:

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

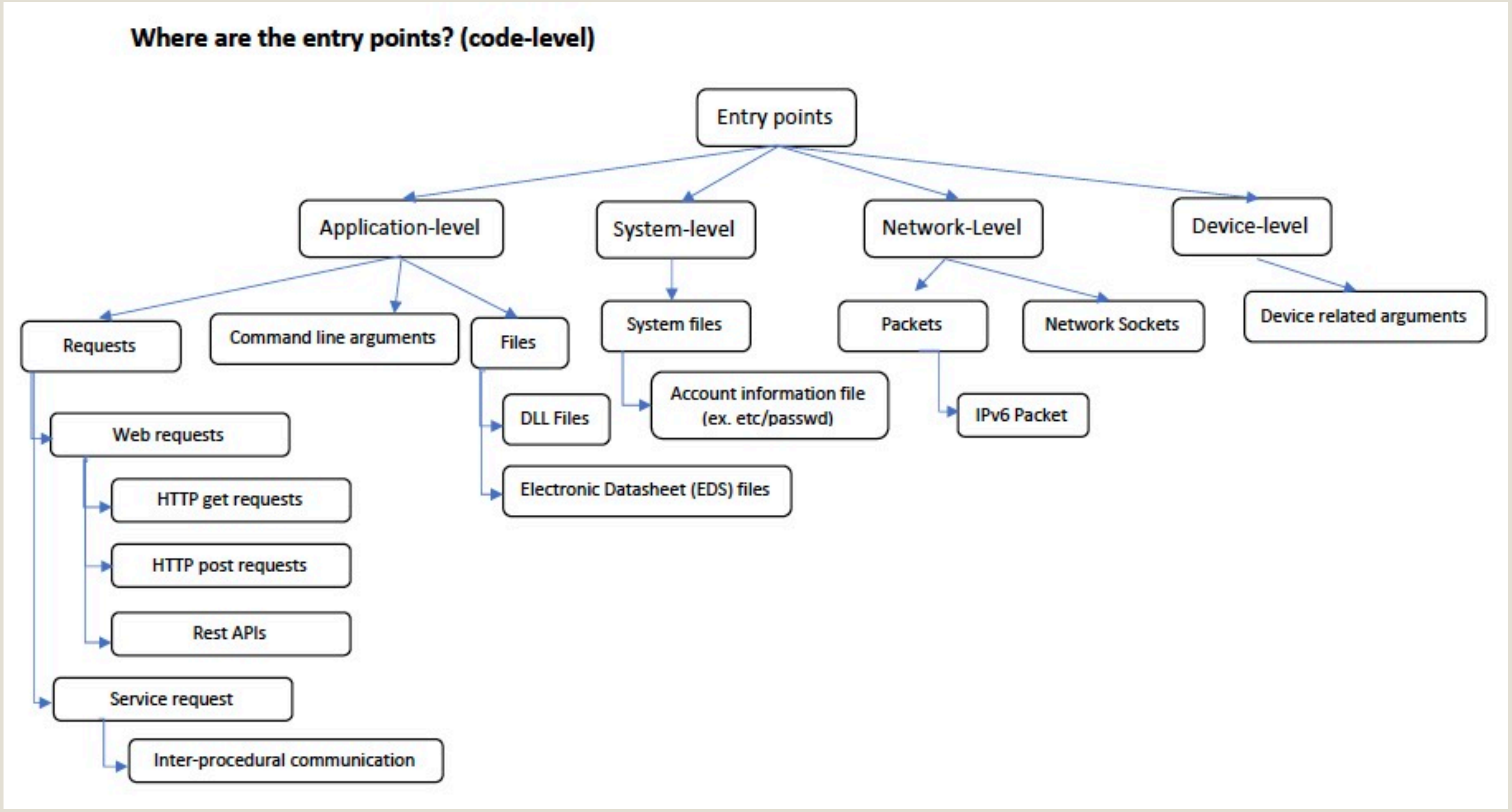
Memo#5: Initial Axial Codes and Categories for Entry Points

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



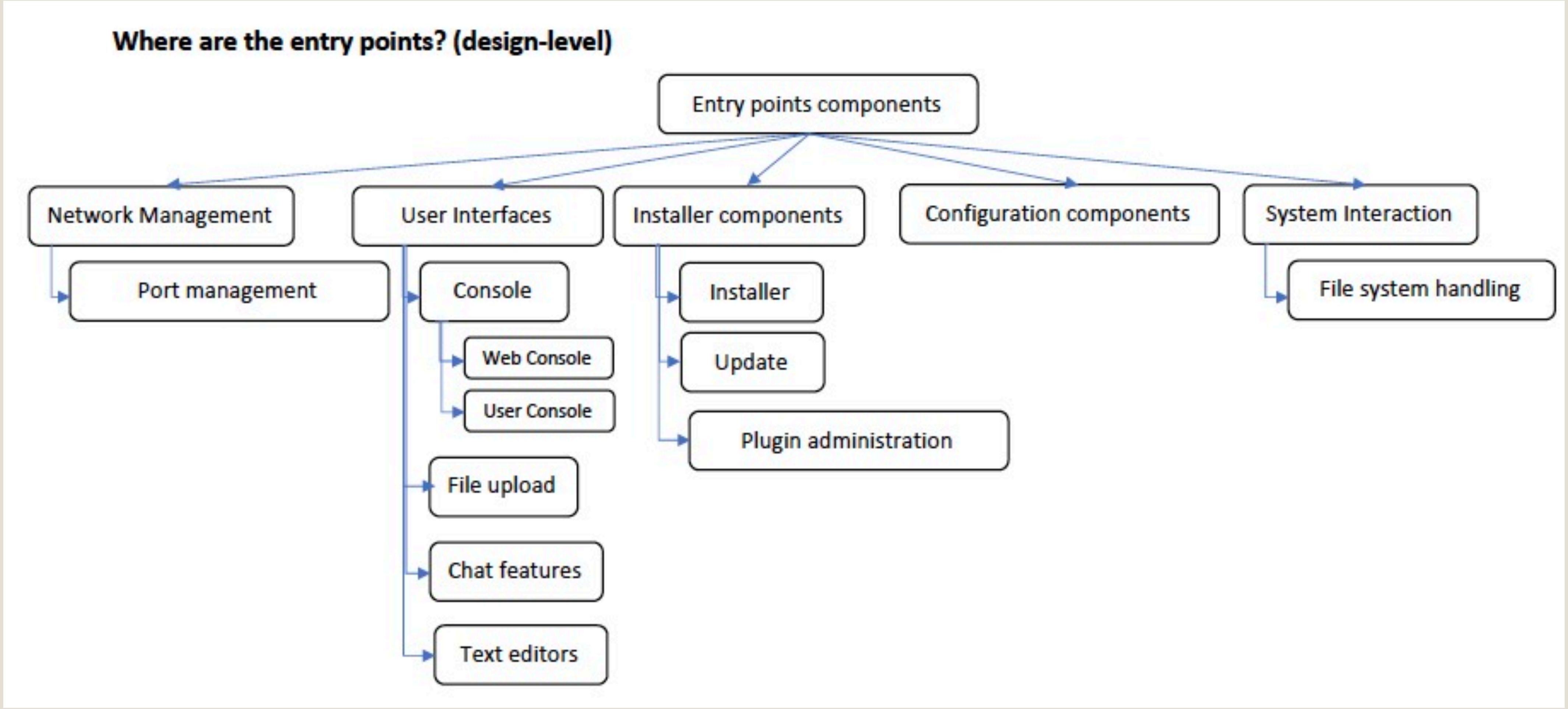
Memo#6: Initial Axial Codes and Categories for Targets

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



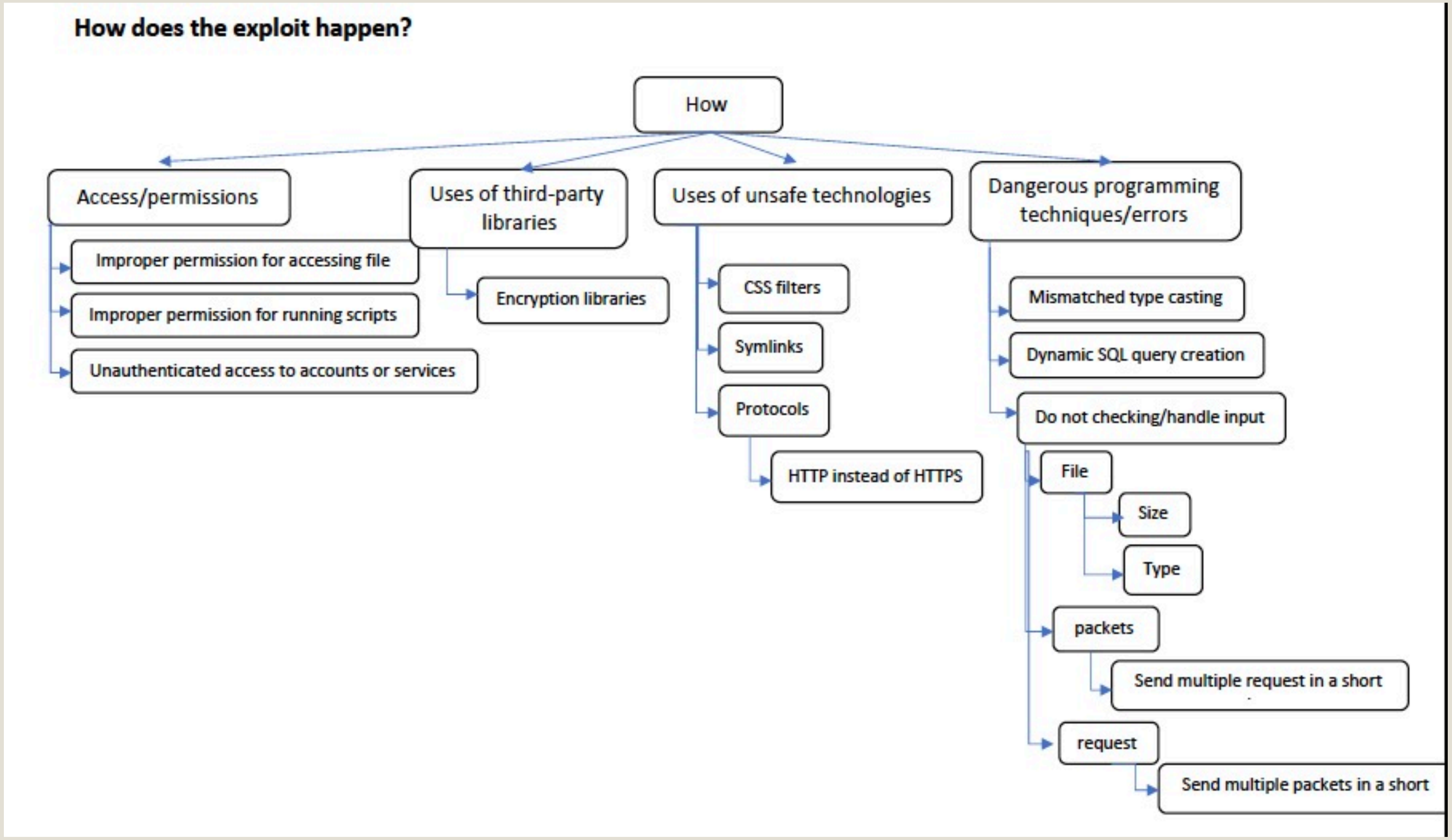
Memo#7: Initial Axial Codes and Categories for Targets (Design-Level)

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



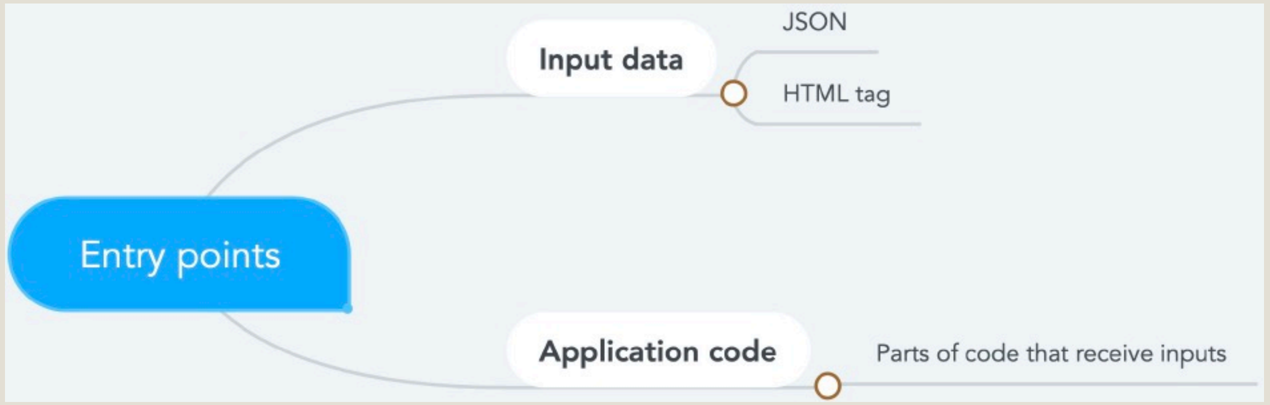
Memo#8: Initial Axial Codes and Categories for Mechanisms

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


How does the exploit happen?
How
Access/permissions
Improper permission for accessing file
Improper permission for running scripts
Unauthenticated access to accounts or services
Uses of third-party libraries
Encryption libraries
Uses of unsafe technologies
CSS filters
Symlinks
Protocols
HTTP instead of HTTPS
Dangerous programming techniques/errors
Mismatched type casting
Dynamic SQL query creation
Do not checking/handle input
File
Size
Type
packets
Send multiple request in a short
request
Send multiple packets in a short

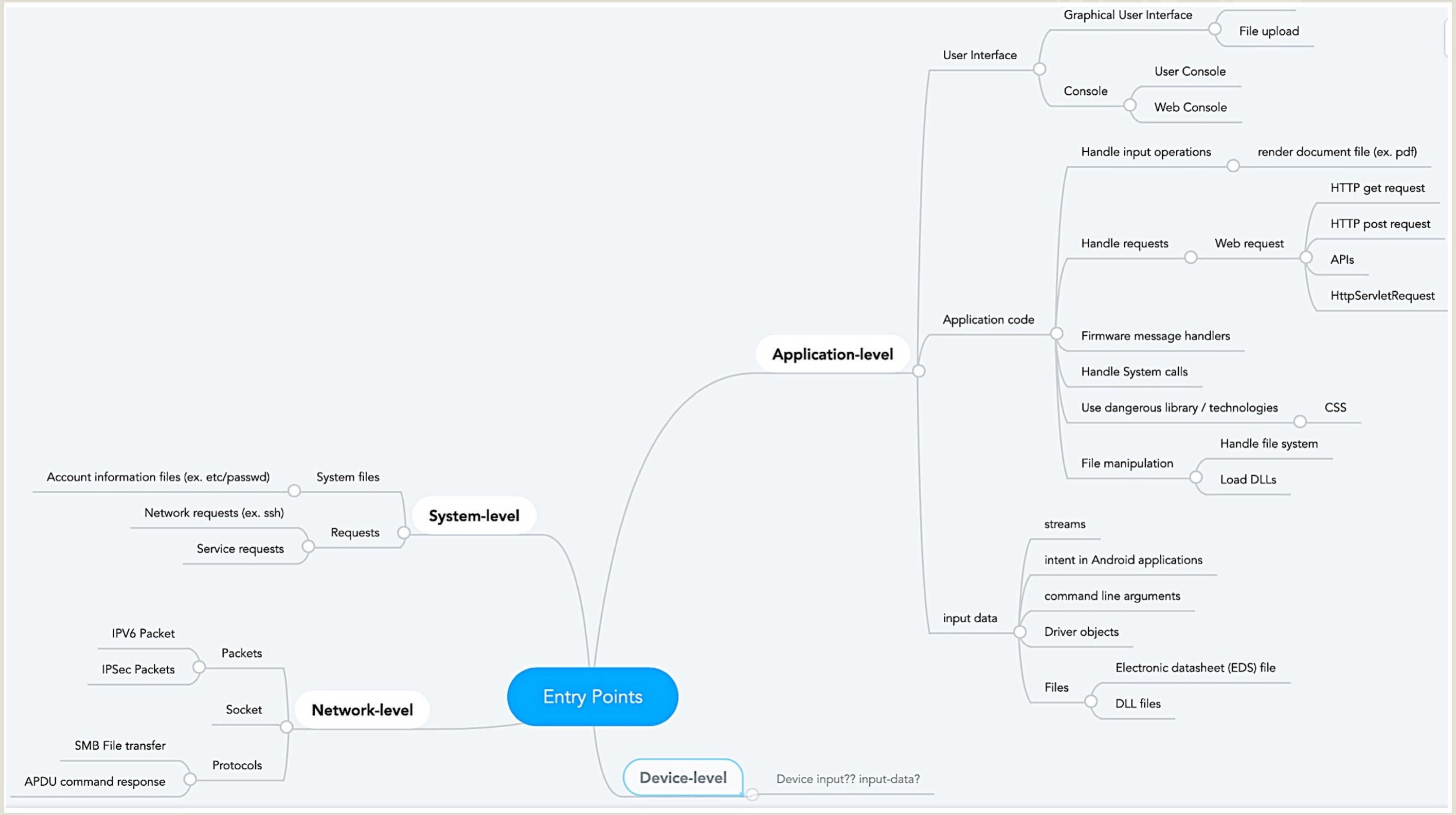
Memo#9: Mindmap Sample for Entry Points (Axial Codes)

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



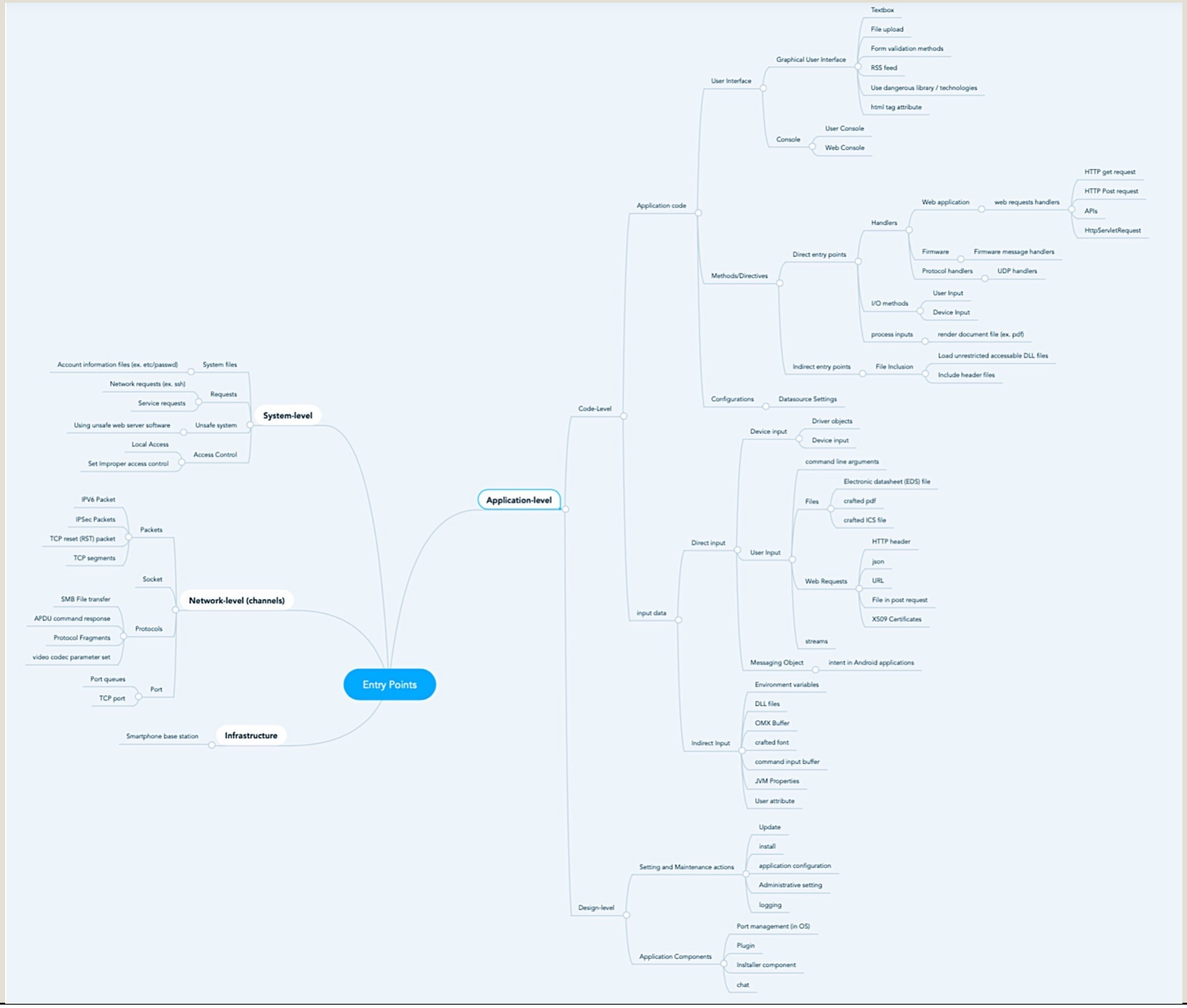
Memo#10: Mindmap 2 for Entry Points (Axial Codes)

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.



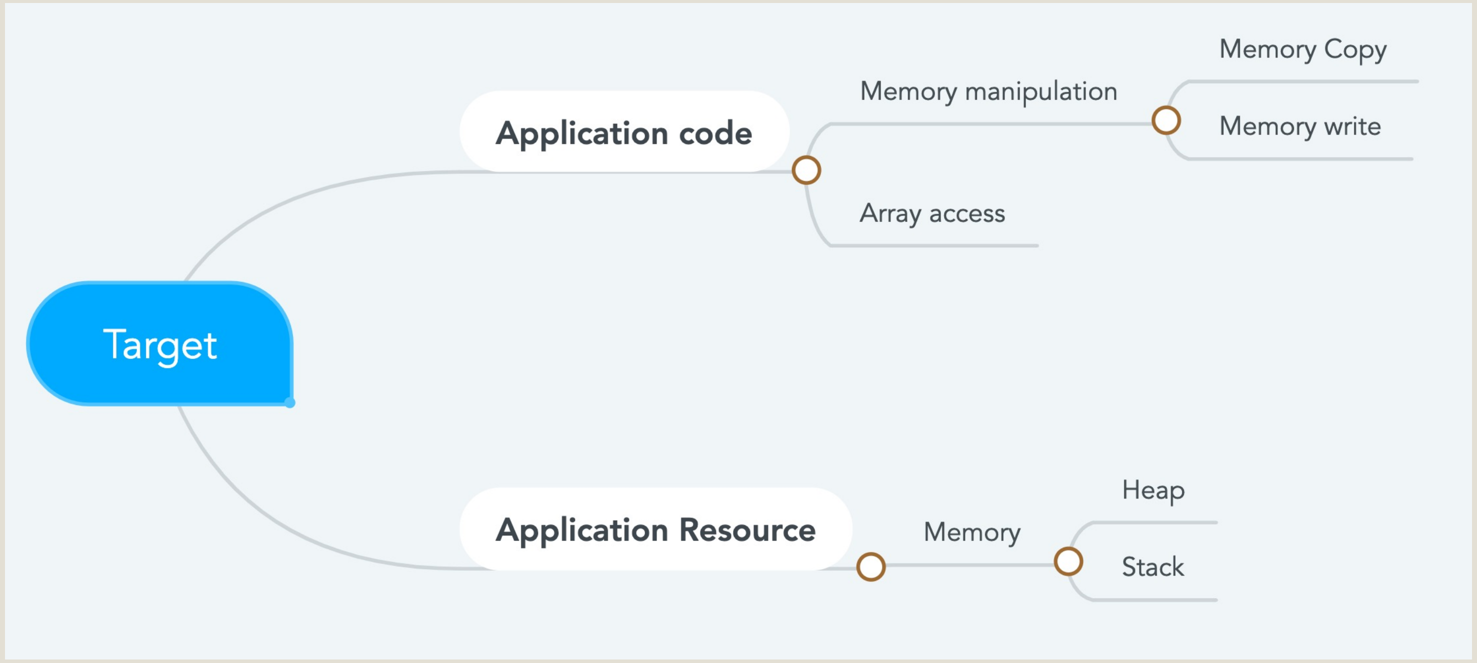
Memo#11: Mindmap 3 for Entry Points (Axial Codes)

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.



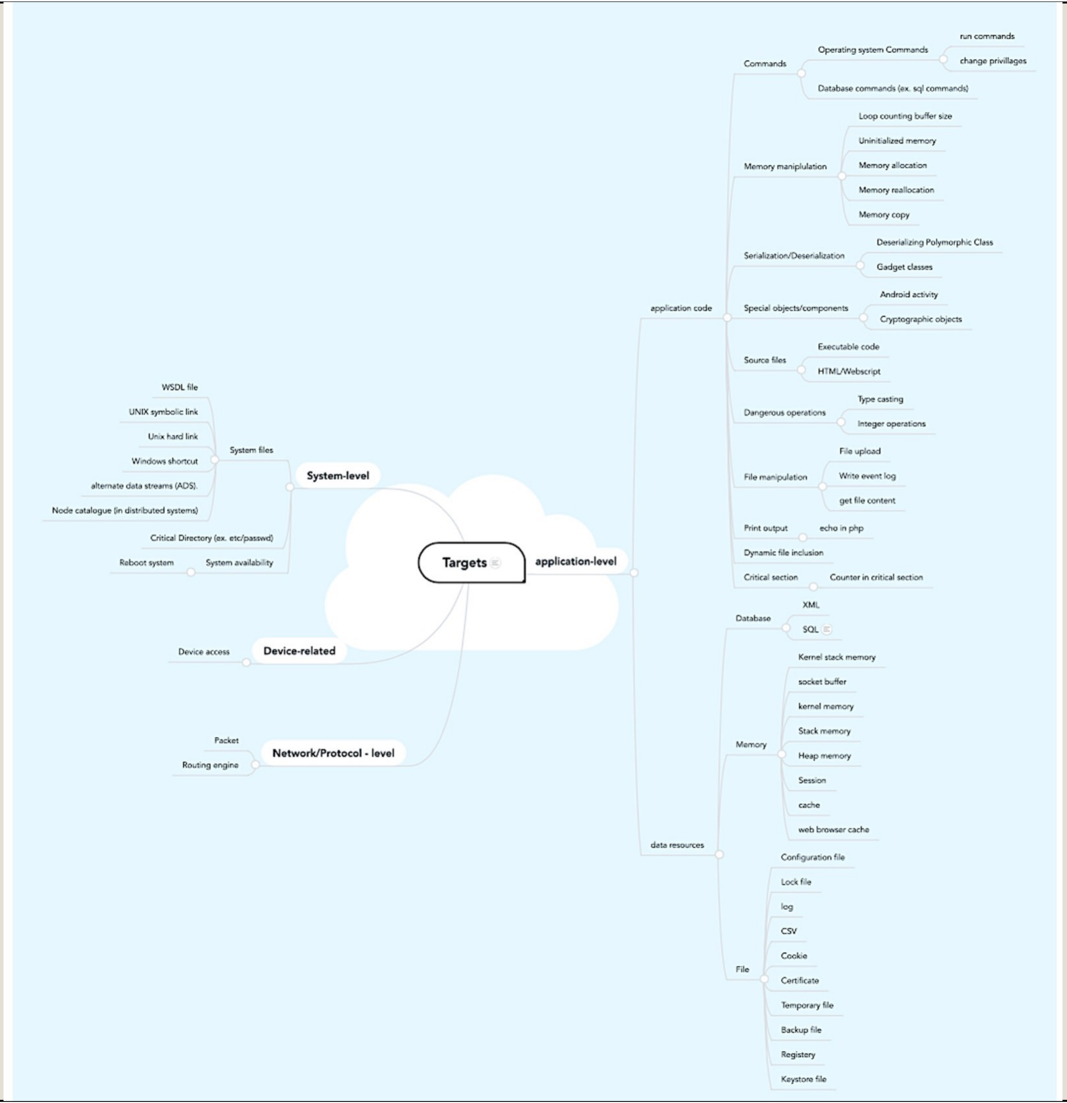
Memo#12: Mindmap Sample for Targets (Axial Codes)

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



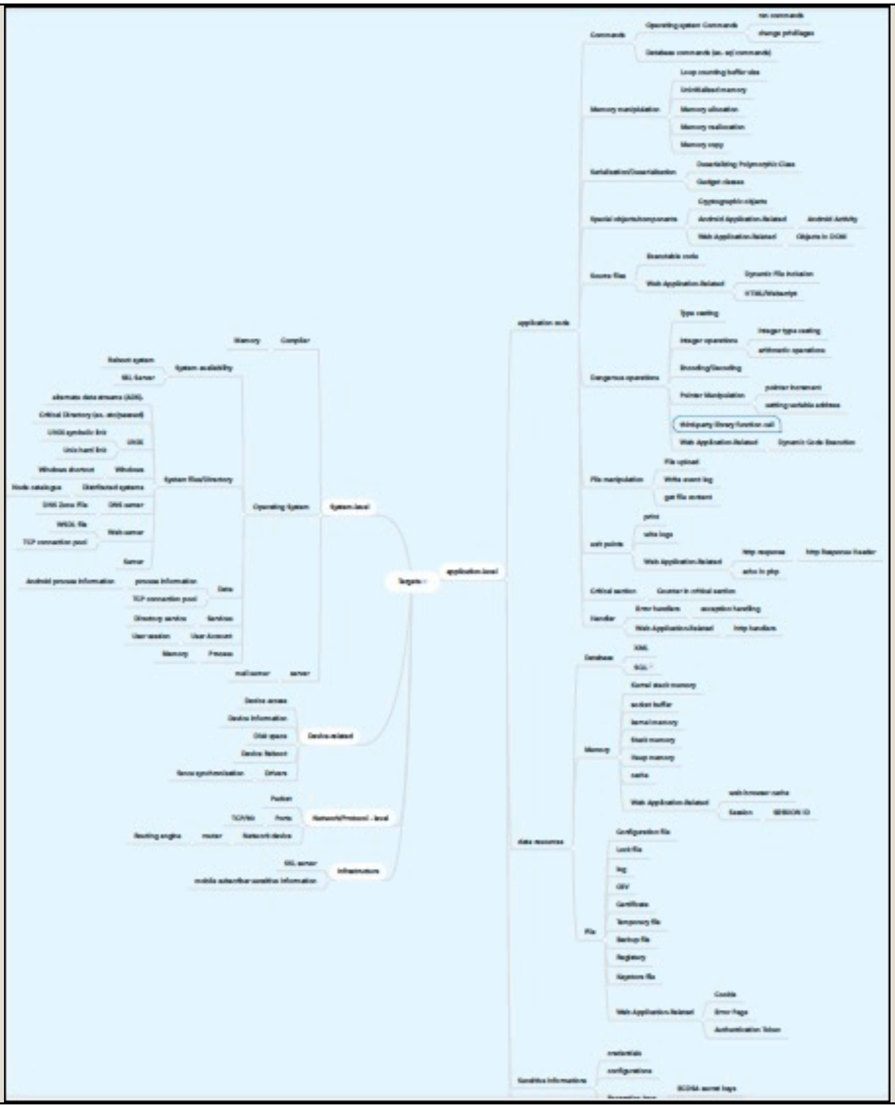
Memo#13: Mindmap 2 for Targets (Axial Codes)

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



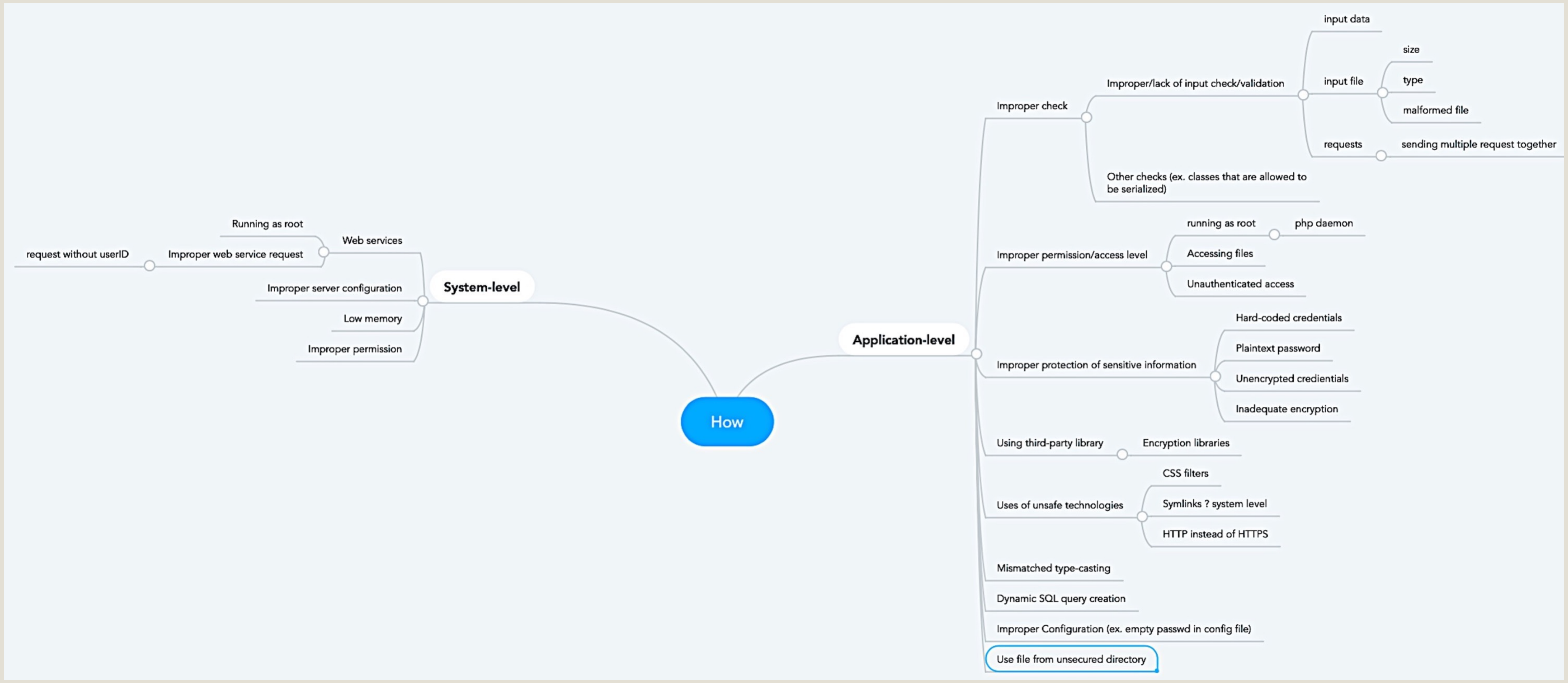
Memo#14: Mindmap 3 for Targets (Axial Codes)

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



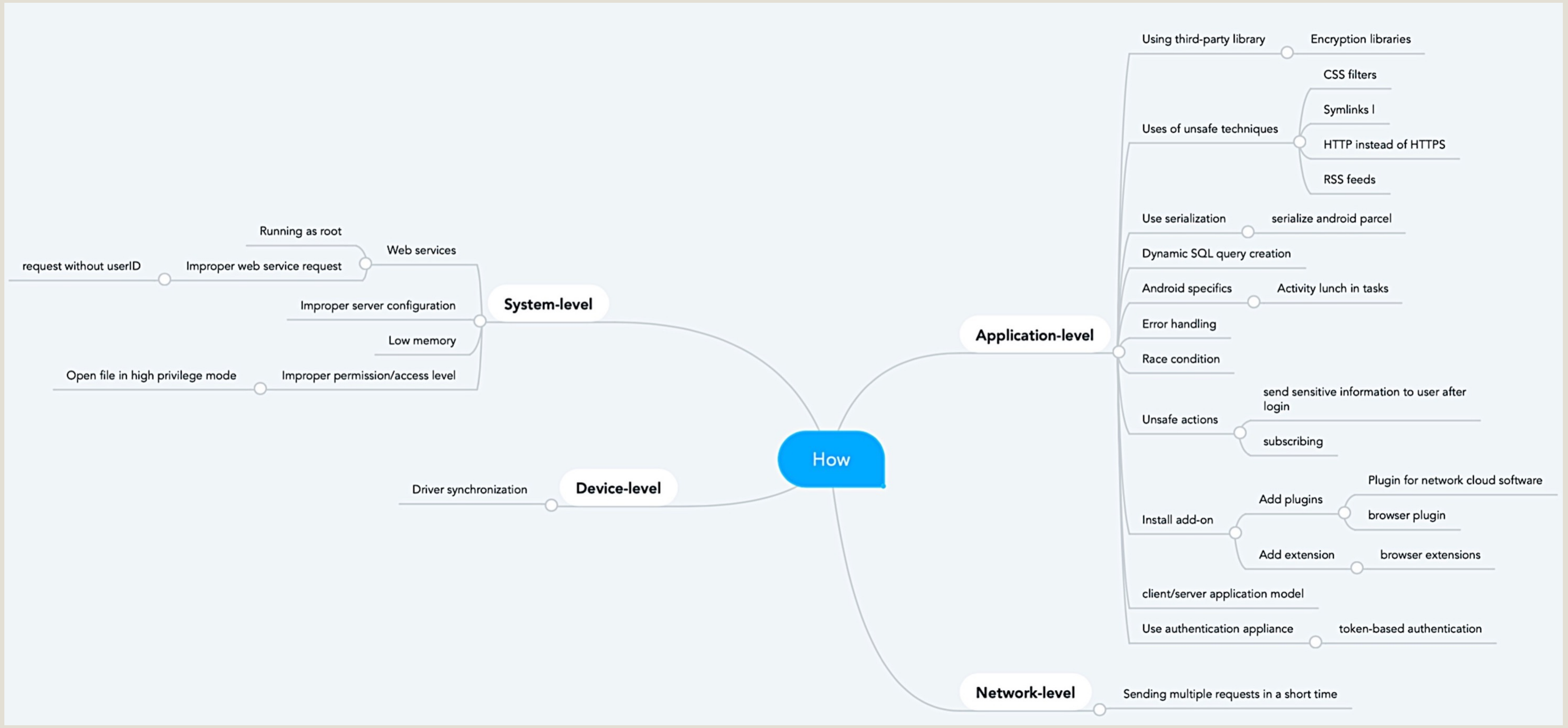
Memo#15: Mindmap 1 for Mechanisms (Axial Codes)

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


How
System-level
Web services
Running as root
Improper web service request
request without userID
Improper server configuration
Low memory
Improper permission
Application-level
Improper check
Improper/lack of input check/validation
input data
input file
size
type
malformed file
requests
sending multiple request together
Other checks (ex. classes that are allowed to be serialized)
Improper permission/access level
running as root
php daemon
Accessing files
Unauthenticated access
Improper protection of sensitive information
Hard-coded credentials
Plaintext password
Unencrypted credientials
Inadequate encryption
Using third-party library
Encryption libraries
Uses of unsafe technologies
CSS filters
Symlinks ? system level
HTTP instead of HTTPS
Mismatched type-casting
Dynamic SQL query creation
Improper Configuration (ex. empty passwd in config file)
Use file from unsecured directory

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.



]

Memo #17: Open Codes and Initial Concepts for Targets

Codings	Category
CodingTagWhat: Arbitrary file read and deletion	-
CodingTagWhat: Availability	-
CodingTagWhat: Configuration	Application file
CodingTagWhat: Counter in critical sections	Application code
CodingTagWhat: Credentials	Sensitive information
CodingTagWhat: Cryptographic Objects	Application code
CodingTagWhat: Execute arbitrary OS commands	Application code
CodingTagWhat: Exhaust disk space	System-level ??
CodingTagWhat: Files	System-level files
CodingTagWhat: Files on server	Server-level-files
CodingTagWhat: Functions	-
CodingTagWhat: Gadgets	Application code -serialization
CodingTagWhat: Gadgets class	Application code -serialization
CodingTagWhat: HTML/Web Script	Application code-output
CodingTagWhat: Memory	
CodingTagWhat: Memory (Socket Buffer)	
CodingTagWhat: Memory allocation	Application memory
CodingTagWhat: Node Catalog	System file-distributed systems
CodingTagWhat: Not Specified	
CodingTagWhat: Open web shell	System-level server
CodingTagWhat: Parameter tampering	Network-level (http parameter tampering)
CodingTagWhat: Reset settings	-
CodingTagWhat: Routing Engine	Network-level
CodingTagWhat: SQL command	Application code
CodingTagWhat: Spoof	-
CodingTagWhat: System availability	-
CodingTagWhat: System calls	Application code
CodingTagWhat: Unmarshalling data to objects	Application code
CodingTagWhat: User data	-

CodingTagWhat: XBlock data	-
CodingTagWhat: add admin user	Deisgn-level
CodingTagWhat: android activity	Application-code
CodingTagWhat: argument type casting	Application-code
CodingTagWhat: check boundary	Network-level (boundary in network-packet)
CodingTagWhat: configuration file of executable files	Application-data file
CodingTagWhat: critical directory	Server-level directory
CodingTagWhat: deserializing class	Application-code
CodingTagWhat: deserializing polymorphic class	Application-code
CodingTagWhat: device access	Device level
CodingTagWhat: dynamic code inclusion	Application-code
CodingTagWhat: enable escalation of privilege	-
CodingTagWhat: file system	System-level
CodingTagWhat: file upload	Application-code UI
CodingTagWhat: get file content	Application-code
CodingTagWhat: heap-based memory	MEMORY-LEVEL
CodingTagWhat: integer operations	Application-code
CodingTagWhat: kernel memory	MEMORY-LEVEL
CodingTagWhat: kernel stack memory	MEMORY-LEVEL
CodingTagWhat: lock file	Application-data file
CodingTagWhat: loop counting buffer size	Application-code
CodingTagWhat: memory copy	Application-code
CodingTagWhat: memory reallocation	Application-code
CodingTagWhat: metadata	-
CodingTagWhat: object	-
CodingTagWhat: object methods in ORM	-
CodingTagWhat: operating system command	Application-code
CodingTagWhat: port interface management part of the operating system	DESIGN-LEVEL
CodingTagWhat: print output	Application-code
CodingTagWhat: private information	-
CodingTagWhat: remotely execute code	-
CodingTagWhat: run source code	Application-code
CodingTagWhat: sensitive information	-
CodingTagWhat: software related files	-

CodingTagWhat: system reboot	SYSTEM REBOOT
CodingTagWhat: tmp file	Application-DATA FILE
CodingTagWhat: uninitialized memory	Application-code MEMORY
CodingTagWhat: write event log	Application-code

Memo#18: Open Codes and Initial Concepts for Entry Points

CodingTagWhere: APDU command response	Network-related protocol
CodingTagWhere: Account information file	System-level file
CodingTagWhere: Administrative settings	
CodingTagWhere: Administrative user interface	
CodingTagWhere: Application Configuration	Application-level file
CodingTagWhere: Application Configuration (design-level)	
CodingTagWhere: CSS	Application-level code
CodingTagWhere: Chats	
CodingTagWhere: Command line arguments	Input-data
CodingTagWhere: Create Repository	
CodingTagWhere: DLL File(s)	Application-level code
CodingTagWhere: Database	Application-level data resource
CodingTagWhere: Datasource Settings	Application-level code
CodingTagWhere: Decompress collection file	?
CodingTagWhere: Deserialization	Application-level code
CodingTagWhere: Document File Upload (design-level)	Application-level user inteface
CodingTagWhere: EDS File	Input-data file
CodingTagWhere: Editing of system data	Application-level code
CodingTagWhere: Files in post request	Input-data file
CodingTagWhere: Filesystem Handling	Application-level code file manipulation
CodingTagWhere: Firmware update	
CodingTagWhere: HTTP Headers	Application-level code handle requests
CodingTagWhere: HTTP POST	Application-level code handle requests
CodingTagWhere: HTTP POST REQUEST	Application-level code handle requests
CodingTagWhere: HTTP Redirect	Application-level code handle requests
CodingTagWhere: HTTP Request	Application-level code handle requests
CodingTagWhere: IPsec packet	Network-level packets
CodingTagWhere: IPv6 packets	Network-level packets
CodingTagWhere: Image file upload	Application-level code file manipulation

CodingTagWhere: Input/Output	Application-level code handle input data
CodingTagWhere: Insecure direct object reference	?
CodingTagWhere: Install	
CodingTagWhere: Installer component (design-level)	
CodingTagWhere: Login	
CodingTagWhere: Markdown Editor	
CodingTagWhere: Network socket	Network-level sockets
CodingTagWhere: No entry point	
CodingTagWhere: Not Specified	
CodingTagWhere: ORM Query Generator	-
CodingTagWhere: Plugin Administration Page (design-level)	
CodingTagWhere: Port Management Interface System	
CodingTagWhere: Print from file (design-level)	
CodingTagWhere: REST API	Application-level code handle requests
CodingTagWhere: Render document	?
CodingTagWhere: SMB file transfer	Network-related protocol
CodingTagWhere: SQL command	?
CodingTagWhere: System call arguments/driver objects	Input data
CodingTagWhere: Ticket Form	
CodingTagWhere: Token Processing System	
CodingTagWhere: URL	
CodingTagWhere: Update	
CodingTagWhere: User Input	
CodingTagWhere: User console	Application-level User interface console
CodingTagWhere: User console (design-level)	Application-level User interface console
CodingTagWhere: Webconsole admin GUI	Application-level User interface console
CodingTagWhere: access to the system that software installed on	-
CodingTagWhere: application web console	
CodingTagWhere: application web console (design-level)	
CodingTagWhere: device related arguments	
CodingTagWhere: file upload	Application-level code file manipulation
CodingTagWhere: firmware message handlers	Application-level code

CodingTagWhere: http get parameter	Application-level code handle requests
CodingTagWhere: input textbox	Application-level UI
CodingTagWhere: load dll	Application-level code file manipulation
CodingTagWhere: local access	Access-level local access
CodingTagWhere: network request	System-level request ssh
CodingTagWhere: obtain the ability to execute high-privileged code	
CodingTagWhere: request to system service	System-level request network
CodingTagWhere: system call	?

Memo#19: Open Codes and Initial Concepts for Mechanisms

CodingTagHow: Alter application files	-
CodingTagHow: Arguments not being validated.	Application-level Input validation
CodingTagHow: Bad request	-
CodingTagHow: Configuring database using unescaped shell arguments	Application-level Input validation
CodingTagHow: Cross-site Scripting (XSS)	-
CodingTagHow: Decode credentials	
CodingTagHow: File deletion based on user input	Application-level Input validation
CodingTagHow: Filename	-
CodingTagHow: HTTP Request Smuggling	-
CodingTagHow: Hard-coded Credentials	Improper sensitive information protection
CodingTagHow: Improper Configuration	Application-level
CodingTagHow: Improper Screen Resizing	
CodingTagHow: Improper Server Management Configuration	System-level
CodingTagHow: Improper check	Application-level
CodingTagHow: Improper permissions	Application-level
CodingTagHow: Inadequate encryption	Improper sensitive information protection
CodingTagHow: Injecting crafted json	-
CodingTagHow: Intercept HTTP request	-
CodingTagHow: Invoke functions via deserialization	
CodingTagHow: Lack of input validation/sanitization	Application-level Input validation
CodingTagHow: Lack of password validation	Application-level Input validation
CodingTagHow: Low memory	System-level
CodingTagHow: Malformed file	Application-level Input validation
CodingTagHow: Man-in-the-Middle	
CodingTagHow: Manipulating Transfer-Encoding and Content-Length in http headers	
CodingTagHow: Memory allocation with incorrect size	

CodingTagHow: Missing http response security headers	
CodingTagHow: Modification of account information file	
CodingTagHow: Modify file extension	
CodingTagHow: Not Specified	
CodingTagHow: Open file in high privilege mode	
CodingTagHow: Out of bounds memory read	
CodingTagHow: Plaintext password storage	Improper sensitive information protection
CodingTagHow: Race Condition	
CodingTagHow: Repeated requests	Network-level
CodingTagHow: Request service without user ID	System-level
CodingTagHow: Shell Command	-
CodingTagHow: Shell Command via HTTP request	-
CodingTagHow: System functions	-
CodingTagHow: Type confusion	Application-level improper type casting
CodingTagHow: Unauthenticated access	Improper sensitive information protection
CodingTagHow: Uncontrolled recursion	-
CodingTagHow: Unencrypted credentials	Improper sensitive information protection
CodingTagHow: Use file from unsecured directory	Application-level
CodingTagHow: Use of HTTP instead of HTTPS	Application-level unsafe technologies
CodingTagHow: Use third part library	Application-level
CodingTagHow: Using CSRF vulnerability	-
CodingTagHow: Using CSS filter	Application-level unsafe technologies
CodingTagHow: Using symlink	Application-level unsafe technologies
CodingTagHow: accessing buffer in loop without checking the buffer size	Application-level improper check
CodingTagHow: accessing file	-
CodingTagHow: calling system calls with specific parameter	Application-level unsafe technologies
CodingTagHow: continuously sending packet	Network-level
CodingTagHow: do not checking file type	Application-level Input validation
CodingTagHow: does not check input file size	Application-level Input validation
CodingTagHow: gain administrative acess	Sysm-level application-level improper permission
CodingTagHow: have special account	
CodingTagHow: improper input validation	

CodingTagHow: incorrect android activity launch in tasks	Application-level insecure programming practices
CodingTagHow: incorrect checking of boundary	Application-level improper check
CodingTagHow: incorrect type decleration	
CodingTagHow: inject arbitrary code	-
CodingTagHow: injecting malicious command as son parameter	-
CodingTagHow: insert crafted YAML input	Application-level Input validation
CodingTagHow: insert crafted data	-
CodingTagHow: lack of proper locking when performing operations on an object	?
CodingTagHow: managing XBlock resources	-
CodingTagHow: mismatched type casting	Application-level insecure programming practices
CodingTagHow: run executables based on accessible configuration file	
CodingTagHow: running php daemon as root	application-level improper permission
CodingTagHow: sending multiple request together or in a short time	
CodingTagHow: sending repeatedly malformed packets	Network-level
CodingTagHow: serializing android parcel	
CodingTagHow: setting improper permissions for file access	application-level improper permission
CodingTagHow: supply crafted smartcards	-
CodingTagHow: upload crafted file name	-
CodingTagHow: use encryption package	Application-level insecure programming practices
CodingTagHow: web service runs under the root user	System-level improper permission