

cwe_data.py

```
import os
import xml.etree.ElementTree as ET
from .cwe_scraper import get_cwe_data, get_cwe_dict as get_scraped_cwe_dict,
refresh_cwe_cache
from .cwe_map import CWE_TITLES

#Path where the MITRE CWE catalog XML should be (inside data folder)
#Using absolute path so we don't have issues with imports
DOWNLOAD_PATH = os.path.abspath
(
    os.path.join(os.path.dirname(__file__), "../data/CWE_Catalog.xml")
)

def parse_cwe_catalog():
    """
    Load CWE data from the XML file first.
    If that fails or the file doesn't exist, falls back to scraped data.
    """
    if not os.path.exists(DOWNLOAD_PATH):
        print(f"[CWE] XML not found at {DOWNLOAD_PATH}. Using scraper
instead.")
        return get_scraped_cwe_dict()

    try:
        # Load the XML tree and grab the root element
        tree = ET.parse(DOWNLOAD_PATH)
        root = tree.getroot()
    except Exception as e:
        # XML exists but won't parse → scrape instead
        print("ERROR: Couldn't parse CWE XML:", e)
        return get_scraped_cwe_dict()

    #Holds all CWE entries with their details
    nodes = {}

    #Some MITRE XML files use namespaces, so let's detect that
    ns = {'ns': root.tag[root.tag.find("{") + 1:root.tag.find("}")] if
root.tag.startswith("{") else {}
    weakness_xpath = "../ns:Weakness" if ns else "../Weakness"
```

```

#Loop through every Weakness entry in the XML
for weakness in root.findall(weakness_xpath, ns):

    #Grab the CWE ID and name
    cwe_id = weakness.attrib.get("ID", "").strip()
    name = weakness.attrib.get("Name", "").strip()

    #Pull the description (if it exists)
    desc_elem = weakness.find("ns:Description" if ns else "Description",
ns)
    description = desc_elem.text.strip() if desc_elem is not None and
desc_elem.text else ""

    #Grab any listed mitigations
    mitigations = []
    mitig_xpath = "ns:Potential_Mitigations/ns:Potential_Mitigation" if ns
else "Potential_Mitigations/Potential_Mitigation"
    for mitig in weakness.findall(mitig_xpath, ns):
        mitig_desc = mitig.find("ns:Description" if ns else "Description",
ns)
        if mitig_desc is not None and mitig_desc.text:
            mitigations.append(mitig_desc.text.strip())

    #Grab demonstrative examples (if any)
    examples = []
    ex_xpath = "ns:Demonstrative_Examples/ns:Demonstrative_Example" if ns
else "Demonstrative_Examples/Demonstrative_Example"
    for demo in weakness.findall(ex_xpath, ns):
        example_desc = demo.find("ns:Description" if ns else
"Description", ns)
        if example_desc is not None and example_desc.text:
            examples.append(example_desc.text.strip())

    #Also record any related weaknesses
    relationships = []
    rel_xpath = "ns:Related_Weaknesses/ns:Related_Weakness" if ns else
"Related_Weaknesses/Related_Weakness"
    for rel in weakness.findall(rel_xpath, ns):
        rel_id = rel.attrib.get("CWE_ID", "")
        rel_type = rel.attrib.get("Nature", "")
        relationships.append((rel_id, rel_type))

    #Only save it if we got an ID + name
    if cwe_id and name:
        nodes[f"CWE-{cwe_id}"] = {
            "id": f"CWE-{cwe_id}",

```

```

        "name": name,
        "description": description,
        "mitigations": mitigations,
        "examples": examples,
        "relationships": relationships,
    }

    #If for some reason the XML gave us barely anything, mix in scraper data
    too

    if len(nodes) < 50:
        print("[CWE] XML parsing gave too few results, adding scraper data to
fill in gaps")
        scraped_data = get_scraped_cwe_dict()
        #Add anything the XML missed
        for cwe_id, cwe_data in scraped_data.items():
            if cwe_id not in nodes:
                nodes[cwe_id] = cwe_data

    return nodes

def get_cwe_dict():
    """
    Get the full CWE dictionary.
    Try XML, if not available or empty, use scraper.
    Always returns something.
    """
    try:
        data = parse_cwe_catalog()
        if len(data) > 0:
            return data
    except Exception as e:
        print("Failed to load CWE catalog from XML:", str(e))

    #XML didn't work? Grab scraper data
    return get_scraped_cwe_dict()

def get_single_cwe(cwe_id):
    """
    Get one specific CWE entry.
    Uses scraper behind the scenes if needed.
    """
    return get_cwe_data(cwe_id)

```

```
def warm_cwe_cache():  
    """  
    Pre-loads the CWE cache so first-time lookups are faster.  
    Doesn't matter if it fails, just logs an error.  
    """  
    try:  
        refresh_cwe_cache()  
    except Exception as e:  
        print(f"Failed to warm CWE cache: {e}")
```