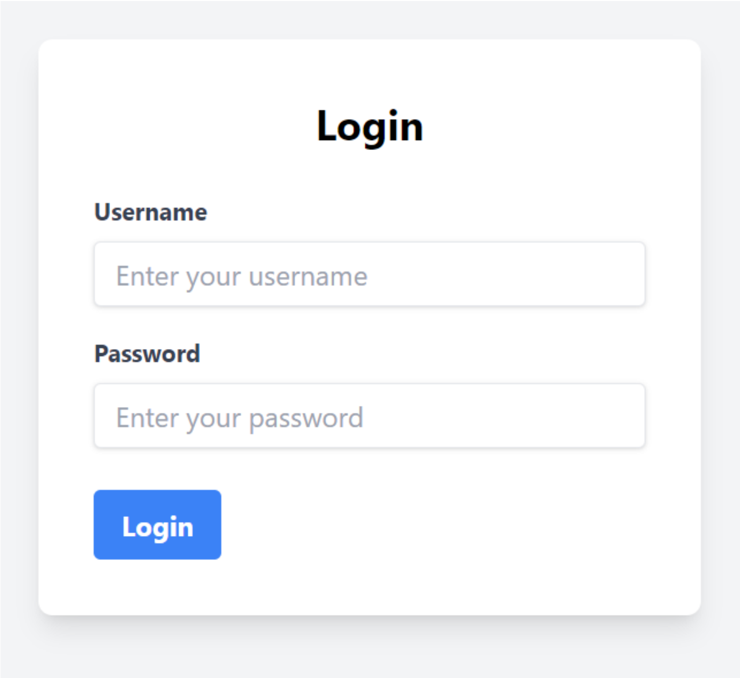
CVE CHECKER TOOL EXPLANATION

Login and Authentication

Use your LDAP credentials

Steps:
1. Username or LDAP email address
2. Password



After Successful Login User can see their profile details like Username, Email, Full name and Title.

Add New Project

User can add new project based on the tool selection.

There are total three tools available

1) **CVEHMI**: Using this tool, user can give input like manifest file which consists all the package name and corresponding package version. It generates reports which contains the list of CVE IDs (with other details) based on package name and vendor.

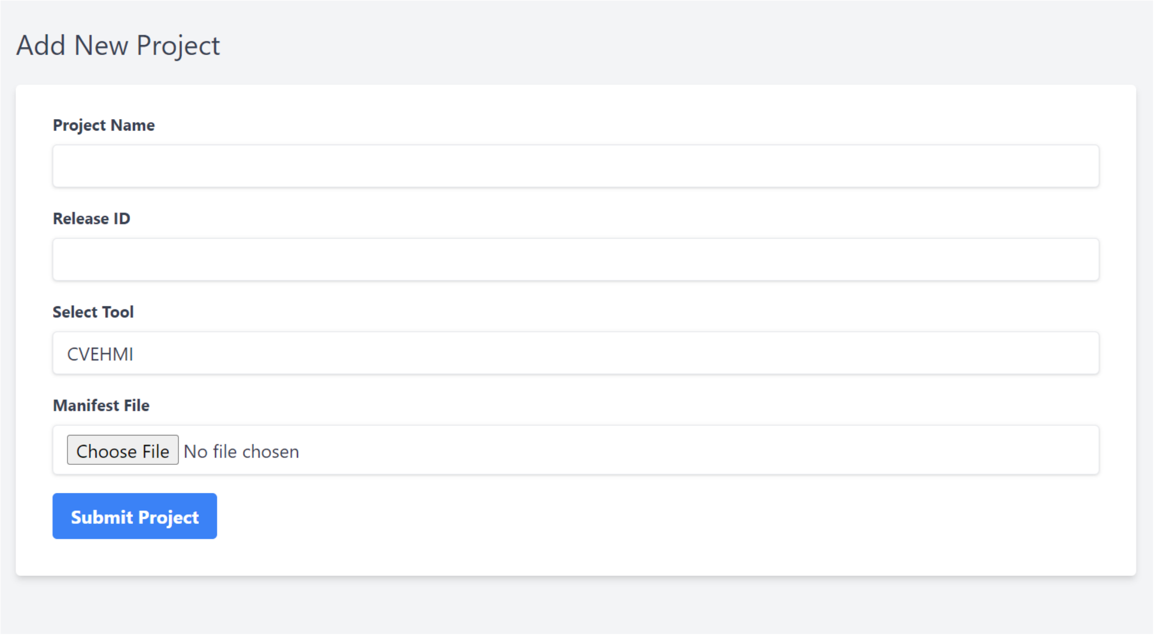
2) **PKCT**: Using this tool, user can give input like manifest file which consists all the package name and corresponding package version, Project kernel git link, Project kernel branch name, Stable branch name and build file. It generates reports which contains the list of CVE IDs (with other details) with status like FIXED, OPEN, UNUSED based on package name and vendor.

3) **Integrated**: This is same as CVEHMI , the only difference is it gives the status of kernel package name's CVE ID's.

A) CVEHMI Tool

To add project based on CVEHMI, file and field details which require are as follows:

1.Project Name (make sure it should be unique)
2. Release ID
3. Select Tool should be CVEHMI
4. Manifest file upload (make sure its file extension should be csv, xlsx, manifest)
5.Blacklist file (Optional): It contains the list CVE IDs (whose data is not included in report) in first column only.



Add New Project

Project Name

Release ID

Select Tool: CVEHMI

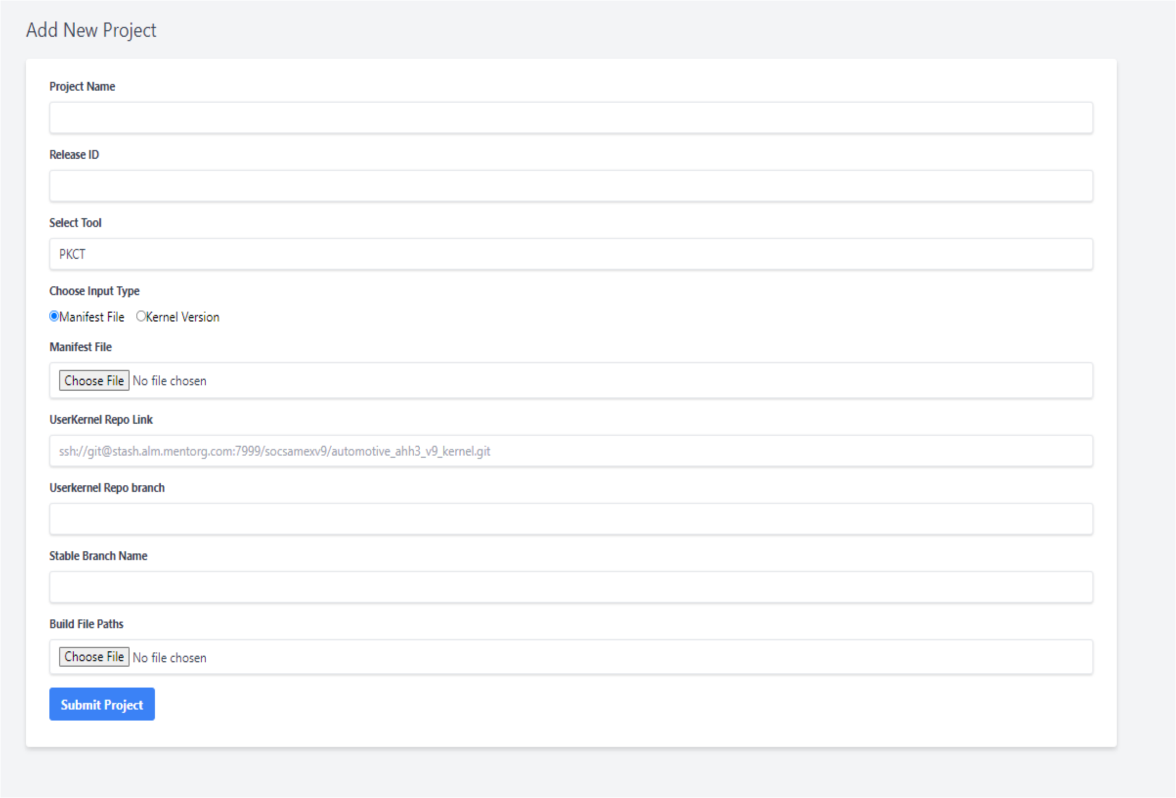
Manifest File: No file chosen

B) PKCT (Patch Kernel Checker Tool)

To add new Project Based on PKCT, file and field details which require are as follows:
1.Project Name (make sure it should be unique)
2. Release ID
3. Select Tool should be PKCT or Integrated
4. User has two options

- Using Package and kernel Version Number
- Using Manifest file (It should contain the package name in first column vendor in Second column and version number in third column.)

Note: Make sure this manifest file consists of 3 columns only otherwise it gives invalid manifest file
5. Project Repository link: remote repository link where the user kernel to be checked is available.
6. Project branch name: The branch of the user kernel repository which needs to be checked.
7. Kernel Stable branch name: Name for the dot kernel branch to be used for more precise analysis.
8. Build Folder File Paths: It contains all the compiled file name which exist in User Kernel. Upload a .txt file which contains the list of paths of every file in the build directory of the user kernel.
9. Blacklist File (Optional): It contains a list of CVE IDs for which the scan does not run.



Add New Project

Project Name

Release ID

Select Tool

PKCT

Choose Input Type

● Manifest File ○ Kernel Version

Manifest File

No file chosen

UserKernel Repo Link

ssh://git@stash.alm.mentorg.com:7999/socsamexv9/automotive_ahh3_v9_kernel.git

Userkernel Repo branch

Stable Branch Name

Build File Paths

No file chosen

• How to generate the above build folder file paths txt file:
In your kernel development folder, make a clean build.
Best method is

```
make mrproper
make <yourconfig>  make O=<object folder>
```

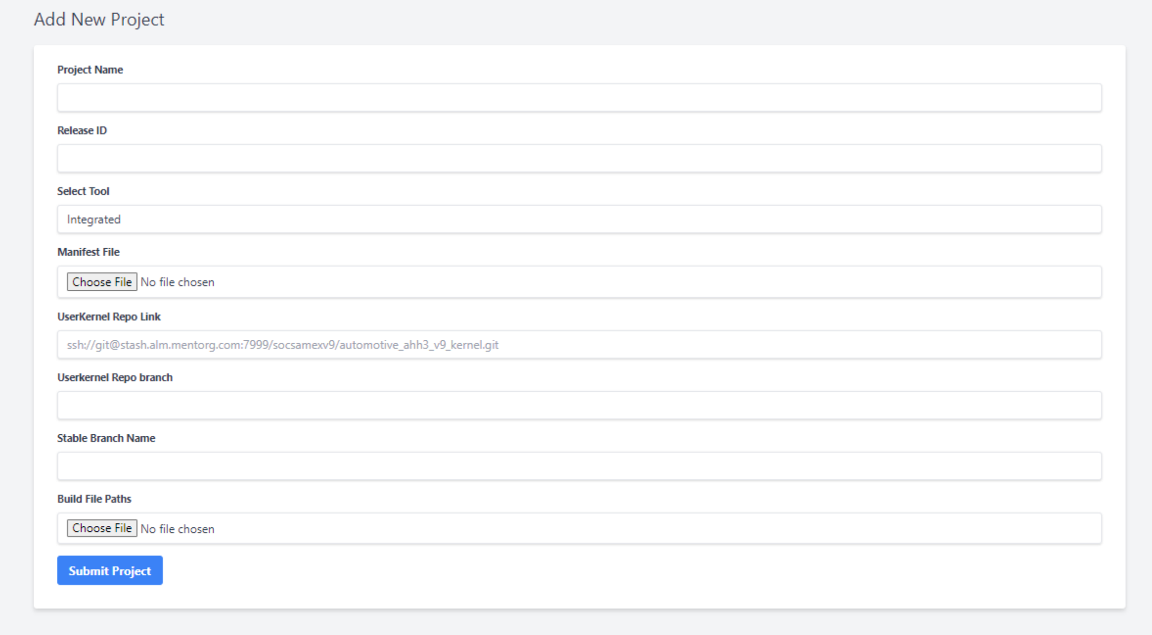
This will build the kernel in the directory specified by object folder.
Go to the folder where build files are. Run the following command to store the path of every file inside the build folder into a text file.

```
find ./ -type f -not -path '*/.*' > .txt
```

C) INTEGRATED

To add new Project Based on Integrated Tool, file and field details which require are as follows:

1.Project Name (make sure it should be unique)
2. Release ID
3. Selected Tool should be Integrated
4. Upload manifest file : The remote repository link where the user kernel to be checked is available.
5. Project Repository link: The remote repository link where the user kernel to be checked is available.
6. Project branch name : The branch of the user kernel repository which needs to be checked.
7. Kernel Stable branch name : : Name for the dot kernel branch to be used for more precise analysis.
8. Build Folder File Paths: It contains all the compiled file name which exist in User Kernel Upload a .txt file which contains the list of paths of every file in the build directory of the user kernel.



Add New Project

Project Name

Release ID

Select Tool
Integrated

Manifest File
No file chosen

UserKernel Repo Link
ssh://git@stash.alm.mentorg.com:7999/socsamexv9/automotive_ahh3_v9_kernel.git

Userkernel Repo branch

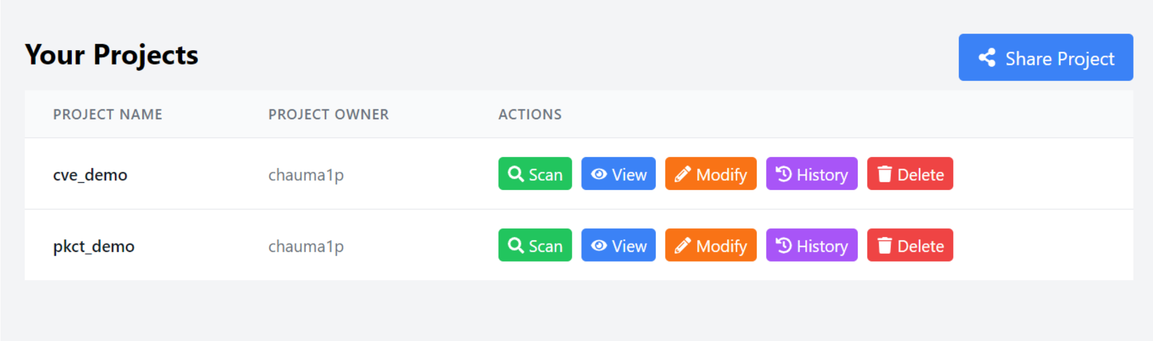
Stable Branch Name

Build File Paths
No file chosen

SCAN ADDED PROJECT

After adding the project, User can see all the added projects and user have multiple actions to do like Scan, View Results, Modify, Modify History and Delete Project.

Scan Project
User can scan multiple projects concurrently by running the scan in background

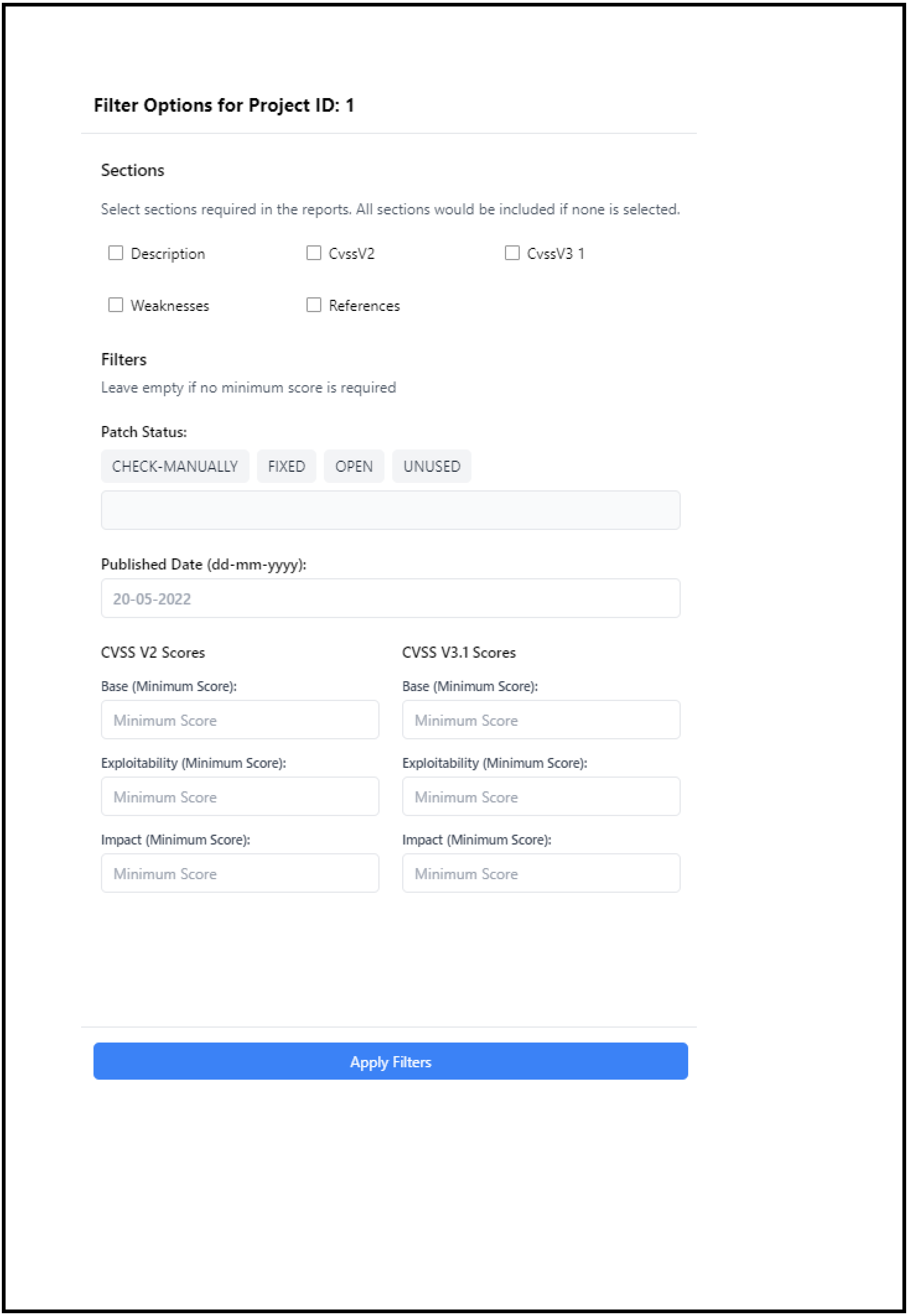


Your Projects

PROJECT NAME	PROJECT OWNER	ACTIONS
cve_demo	chauma1p	Scan View Modify History Delete
pkct_demo	chauma1p	Scan View Modify History Delete

There is filter option given to user to apply and get the filtered results After clicking on scan user can see filter modal.

When user selects the option and submit, scan get started



Filter Options for Project ID: 1

Sections

Select sections required in the reports. All sections would be included if none is selected.

☐ Description ☐ CvssV2 ☐ CvssV3 1

☐ Weaknesses ☐ References

Filters

Leave empty if no minimum score is required

Patch Status:

CHECK-MANUALLY FIXED OPEN UNUSED

Published Date (dd-mm-yyyy):

20-05-2022

CVSS V2 Scores

Base (Minimum Score):

Minimum Score

Exploitability (Minimum Score):

Minimum Score

Impact (Minimum Score):

Minimum Score

CVSS V3.1 Scores

Base (Minimum Score):

Minimum Score

Exploitability (Minimum Score):

Minimum Score

Impact (Minimum Score):

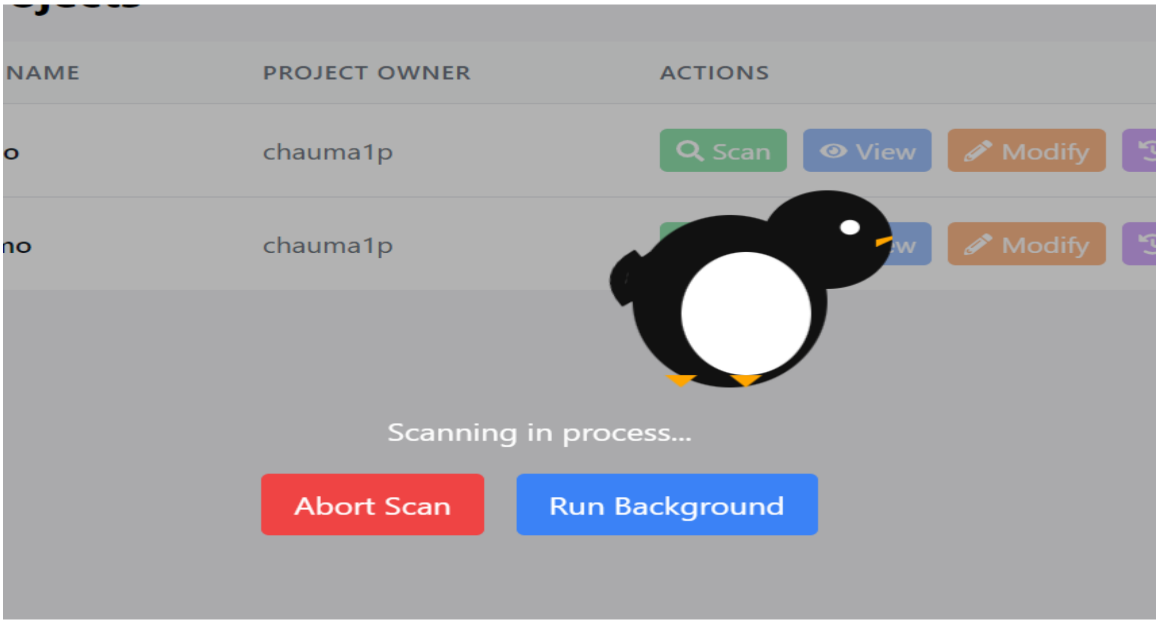
Minimum Score

SCAN PAGE FILTER PARAMETERS

Refer to the following list to know about the optional filter parameters that can be given below.

Filter Parameters	Parameters Definition	Tool Used
Description	It serve as a summary of the vulnerability and can include information such as the vulnerable product, impacts, attack vector, weakness or other relevant technical information	CVEHMI, PKCT and Integrated
Cvss V2 Base Score	This scoring scale helps prioritize vulnerabilities by assessing their exploitability and potential impact on the affected system, providing a standardized measure of their severity	CVEHMI, PKCT and Integrated
CvssV3 Base Score	This scoring scale helps prioritize vulnerabilities by assessing their exploitability and potential impact on the affected system, providing a standardized measure of their severity	CVEHMI, PKCT and Integrated
Weakness	It contains all the details like source, type and description	CVEHMI, PKCT and Integrated
References	These URLs are supplemental information relevant to the vulnerability, which include details that may not be present in the CVE Description. References are given resource tags such as third-party advisory, vendor advisory, technical paper, press/media,	CVEHMI, PKCT and Integrated
Patch Status	It shows the status of patch for a Particular kernel CVE ID. It could be FIXED, NEW,OPEN and UNUSED	PKCT and Integrated

Published Date	States the date when the actual checklist document was published,	CVEHMI, PKCT and Integrated
Cvss V2 Impact Score	The Impact Score measures the potential impact of the successful exploitation of the vulnerability on the confidentiality, integrity, and availability of the affected system	CVEHMI, PKCT and Integrated
Cvss V3 Impact Score	The Impact Score measures the potential impact of the successful exploitation of the vulnerability on the confidentiality, integrity, and availability of the affected system	CVEHMI, PKCT and Integrated
Cvss V2 Exploitability Score	The Exploitability Score is a subcomponent of the Base Score that reflects how easily an attacker can exploit the vulnerability. It considers factors like the access vector (local, adjacent network, network)	CVEHMI, PKCT and Integrated
Cvss V3 Exploitability Score	The Exploitability Score is a subcomponent of the Base Score that reflects how easily an attacker can exploit the vulnerability. It considers factors like the access vector (local, adjacent network, network)	CVEHMI, PKCT and Integrated

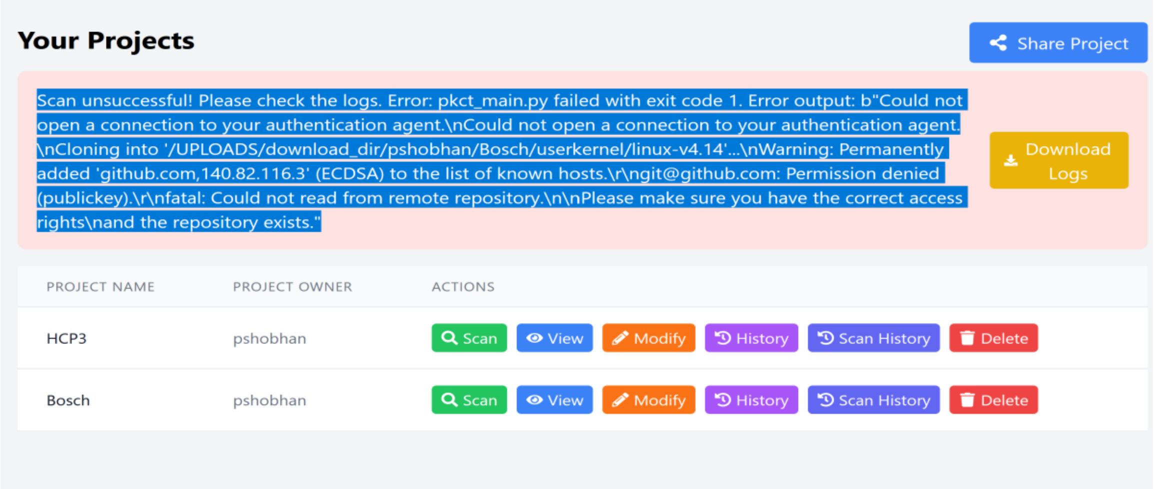


NAME	PROJECT OWNER	ACTIONS
o	chauma1p	Scan View Modify
no	chauma1p	w Modify

Scanning in process...

After Scanning completed the page redirects to Reports page where user can see All the scans which ran successfully in the past.

Note: If you scan project based on PKCT or Integrated , make sure you have valid permission to clone the Project kernel repository otherwise user can see
“Scan unsuccessful, Permission denied to clone the repository”

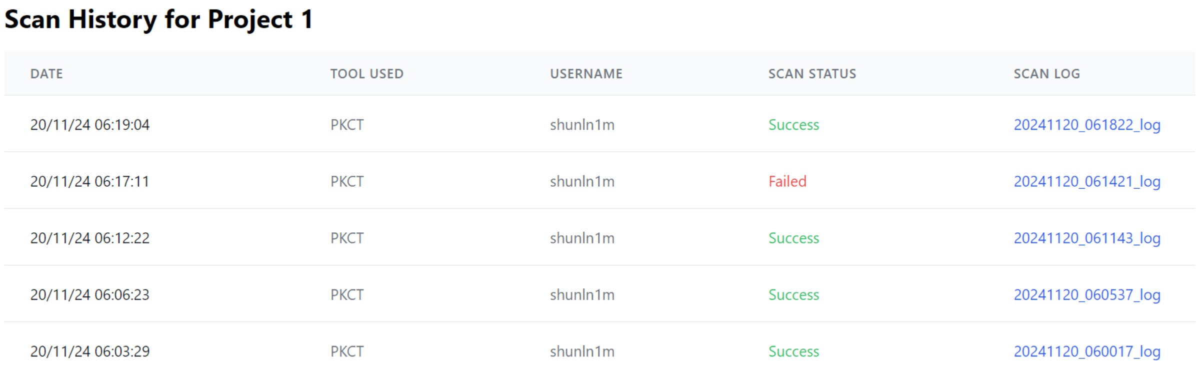


Your Projects

```
Scan unsuccessful! Please check the logs. Error: pkct_main.py failed with exit code 1. Error output: b"Could not open a connection to your authentication agent.\nCould not open a connection to your authentication agent.\nCloning into '/UPLOADS/download_dir/pshobhan/Bosch/userkernel/linux-v4.14'...\nWarning: Permanently added 'github.com,140.82.116.3' (ECDSA) to the list of known hosts.\r\ngit@github.com: Permission denied (publickey).\r\nfatal: Could not read from remote repository.\n\nPlease make sure you have the correct access rights\nand the repository exists."
```

PROJECT NAME	PROJECT OWNER	ACTIONS
HCP3	pshobhan	Scan View Modify History Scan History Delete
Bosch	pshobhan	Scan View Modify History Scan History Delete

Scan History View



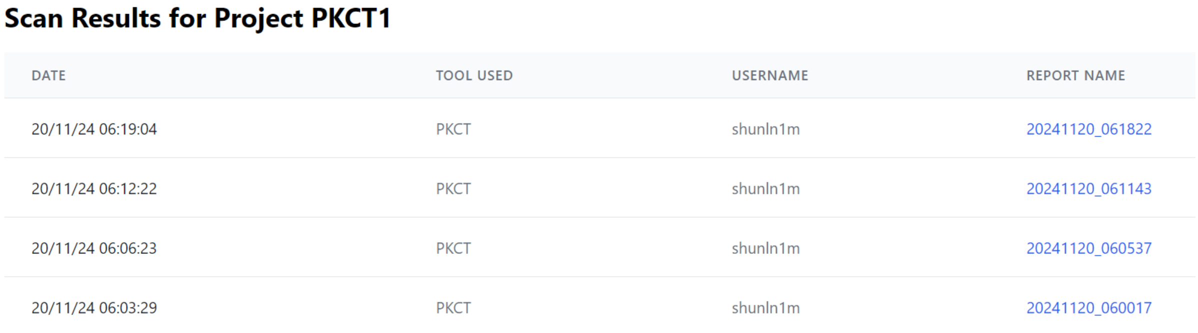
Scan History for Project 1

DATE	TOOL USED	USERNAME	SCAN STATUS	SCAN LOG
20/11/24 06:19:04	PKCT	shunln1m	Success	20241120_061822_log
20/11/24 06:17:11	PKCT	shunln1m	Failed	20241120_061421_log
20/11/24 06:12:22	PKCT	shunln1m	Success	20241120_061143_log
20/11/24 06:06:23	PKCT	shunln1m	Success	20241120_060537_log
20/11/24 06:03:29	PKCT	shunln1m	Success	20241120_060017_log

It contains the status of each scan with corresponding logs whether it fails or pass

Reports

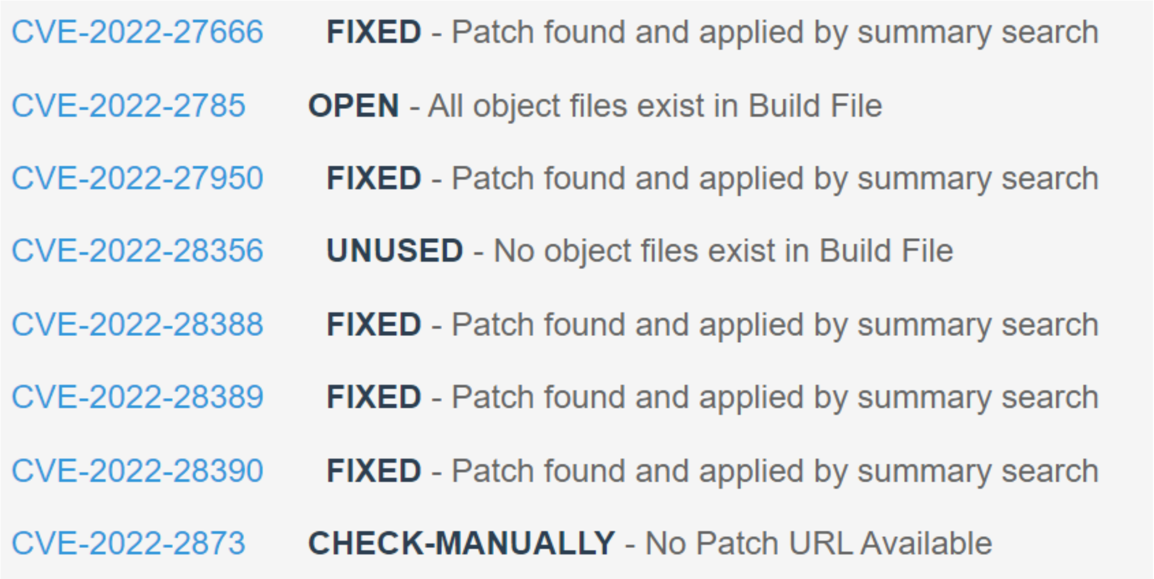
It contains the successful scan reports in xlsx, HTML and PDF extensions.



Scan Results for Project PKCT1

DATE	TOOL USED	USERNAME	REPORT NAME
20/11/24 06:19:04	PKCT	shunln1m	20241120_061822
20/11/24 06:12:22	PKCT	shunln1m	20241120_061143
20/11/24 06:06:23	PKCT	shunln1m	20241120_060537
20/11/24 06:03:29	PKCT	shunln1m	20241120_060017

Patch Status Description in PKCT and Integrated Reports



CVE-2022-27666	**FIXED** - Patch found and applied by summary search
CVE-2022-2785	**OPEN** - All object files exist in Build File
CVE-2022-27950	**FIXED** - Patch found and applied by summary search
CVE-2022-28356	**UNUSED** - No object files exist in Build File
CVE-2022-28388	**FIXED** - Patch found and applied by summary search
CVE-2022-28389	**FIXED** - Patch found and applied by summary search
CVE-2022-28390	**FIXED** - Patch found and applied by summary search
CVE-2022-2873	**CHECK-MANUALLY** - No Patch URL Available

The PKCT Tool classifies the CVE IDs into several classes as Patch Status.

1. **CHECK-MANUALLY :** This classification indicates that the patch verification of the CVE needs manual verification. Either the patch file URL is not available for the CVE or not all the object files exist in the build file.

2. **FIXED :** This classification indicates that the CVE has been identified as fixed for the user kernel.

a. Patch Found Using Commit ID: The patch was applied to the user kernel and verified by the presence of commit ID (which contains patch) in both Linux- stable and user kernel repositories.

b. Patch Found Using Summary Search: The patch was applied to the user kernel and verified by the presence of commit message in both Linux-stable and user kernel repositories.

c. Patch Found Using Code Matching: We identified the patch using code matching and verified its applicability to the user kernel source code. Patch verification for the CVE ID was performed by comparing the patch file's code in both the Linux stable repository and the user kernel repository.

3. **OPEN :** The kernel checked is vulnerable to this CVE. There is an upstream fix available, but has not been ported to this kernel. The classifications are as follows:

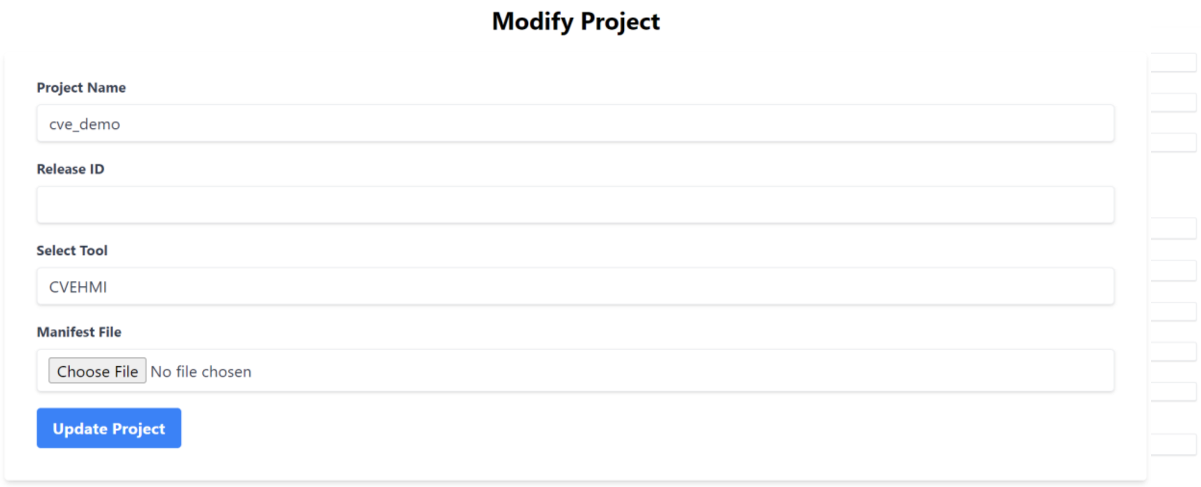
a. All objects files exist in Build File: The kernel configuration is affected by the CVE.

b. Patch found and checked by Code match: This means that the patch is available, and it can be cleanly applied in the user kernel.

4. **UNUSED :** The affected kernel function is not selected/activated in this kernel. Either deactivated through kernel config, related to different architecture or unused/unsupported driver.

Modify Project

Based on the permission of specific project, user can modify the project details and scan the Updated project



Modify Project

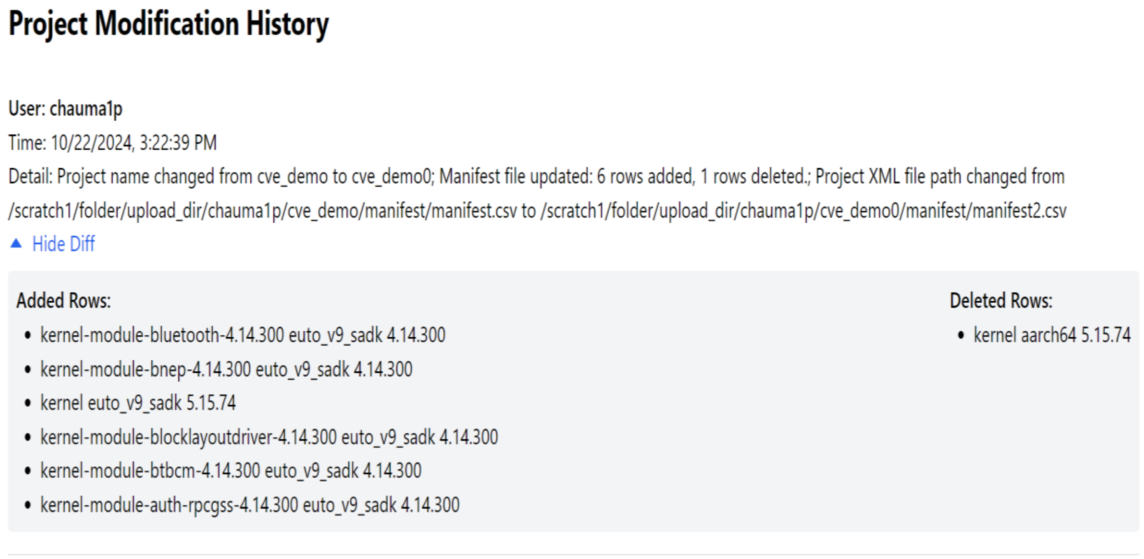
Project Name: cve_demo

Release ID:

Select Tool: CVEHMI

Manifest File: No file chosen

After modification user can see the modified project history by clicking history Button on the projects page.



Project Modification History

User: chauma1p

Time: 10/22/2024, 3:22:39 PM

Detail: Project name changed from cve_demo to cve_demo0; Manifest file updated: 6 rows added, 1 rows deleted.; Project XML file path changed from /scratch1/folder/upload_dir/chauma1p/cve_demo/manifest/manifest.csv to /scratch1/folder/upload_dir/chauma1p/cve_demo0/manifest/manifest2.csv

Added Rows:

- kernel-module-bluetooth-4.14.300 euto_v9_sadk 4.14.300
- kernel-module-bnep-4.14.300 euto_v9_sadk 4.14.300
- kernel euto_v9_sadk 5.15.74
- kernel-module-blocklayoutdriver-4.14.300 euto_v9_sadk 4.14.300
- kernel-module-btbcm-4.14.300 euto_v9_sadk 4.14.300
- kernel-module-auth-rpcgss-4.14.300 euto_v9_sadk 4.14.300

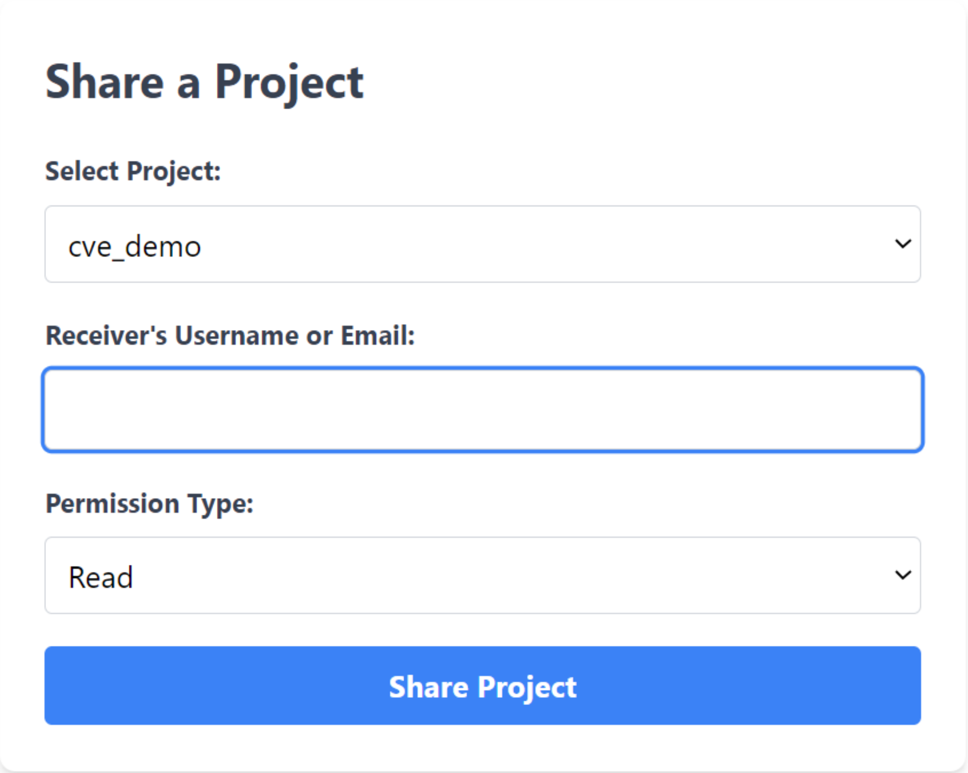
Deleted Rows:

- kernel aarch64 5.15.74

Share Project with Valid User

User can share the project with any other users and additionally give permission.

Like
1.**Read:** User can only scan the project and see the result
2.**Write :** User can modify the project as well
3.**Admin :** User can share the project as well as delete the project



Share a Project

Select Project: cve_demo

Receiver's Username or Email:

Permission Type: Read