

AppChecker

Руководство пользователя

Скачано с  техключи.рф

Содержание

1	О РУКОВОДСТВЕ	3
2	ОБЩАЯ ИНФОРМАЦИЯ	5
3	ВХОД В ПРОГРАММУ	6
4	ПРОЕКТЫ	7
5	ОТЧЕТ СТАТИЧЕСКОГО АНАЛИЗА	11
6	СКАНИРОВАНИЯ	15
7	УПРАВЛЕНИЕ ПРОЕКТОМ	17
8	ЖУРНАЛЫ	18
9	КОНСОЛЬНЫЙ КЛИЕНТ	19
10	ПОДДЕРЖКА	21
11	ГЛОССАРИЙ	23
12	О КОМПАНИИ	25

1 0 руководстве

Этот документ представляет собой Руководство Пользователя AppChecker.

Руководство предназначено для следующих целей:

- ❖ рассказать об отчетах статического анализа;
- ❖ научить работать с дефектами кода;
- ❖ помочь загрузить новые версии исходного кода и конфигурации проекта;
- ❖ показать, как запускать сканирование проекта повторно;
- ❖ научить сравнивать сканирования;
- ❖ рассказать о журналах проекта и сканирования.

Разделы документа

Этот документ содержит следующие разделы.

Общая информация ([стр. 5](#))

Раздел содержит инструкции по загрузке программы.

Вход в программу ([стр. 6](#))

Раздел содержит инструкции по входу в программу.

Проекты ([стр. 7](#))

Раздел содержит общую информацию о работе с проектами.

Отчет статического анализа ([стр. 11](#))

Раздел описывает содержание отчетов статического анализа и учит работать с дефектами кода, обнаруженными в ходе анализа.

Сканирования ([стр. 15](#))

Раздел содержит инструкции по запуску и сравнению сканирований проекта.

Управление проектами ([стр. 17](#))

Раздел содержит инструкции по загрузке новых версий исходного кода и конфигурации проекта.

Журналы ([стр. 18](#))

Раздел содержит описание журналов проекта и сканирования.

Консольный клиент ([стр. 19](#))

Раздел содержит информацию о работе консольного клиента AppChecker.

Поддержка ([стр. 21](#))

Раздел содержит информацию о технической поддержке AppChecker.

Глоссарий ([стр. 23](#))

Раздел содержит определение терминов, встречающихся в документе.

О компании ([стр. 25](#))

Раздел содержит информацию о АО «НПО «Эшелон».

2 Общая информация

AppChecker – статический анализатор кода, предназначенный для поиска дефектов в исходном коде приложений, разработанных на C/C ++, C#, Java, PHP.

Функции

- ❖ поиск дефектов кодирования с помощью постоянно обновляемой базы правил и технологии анализа потоков данных;
- ❖ поиск программных закладок.

Преимущества

- ❖ поиск свыше 100 типов дефектов кодирования;
- ❖ постоянное пополнение базы правил поиска дефектов;
- ❖ поддержка классификации CWE;
- ❖ применение технологии анализа потока данных;
- ❖ возможность проводить совместный аудит кода несколькими экспертами;
- ❖ гибкая конфигурация анализируемых проектов.

3 Вход в программу

Получите адрес веб-интерфейса, логин и пароль у администратора.

Введите адрес веб-интерфейса в адресной строке браузера.

Войдите в программу, используя свой логин и пароль.



Вход в систему

Логин

Пароль

AppChecker поддерживает назначение ролей. Роль определяет возможные действия пользователя. Это означает, что некоторые функции AppChecker могут быть недоступны.

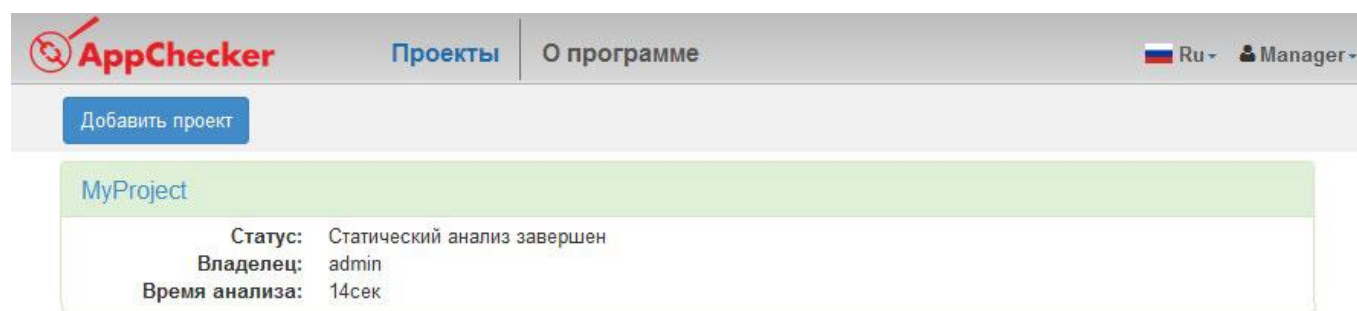
Обратитесь к администратору для изменения назначенных вам ролей.

Язык интерфейса показан в правом углу.

Нажмите «RU», чтобы сменить язык на английский.

4 Проекты

После входа в AppChecker на главной странице вы увидите список проектов и статус статического анализа. Выберите нужный проект из списка.



4.1 Добавление проекта

Для создания нового проекта нажмите «Добавить проект» в левом верхнем углу и введите имя проекта.

Имя проекта может содержать только латинские буквы, цифры, дефис, нижнее подчеркивание и точку.

Если функция создания проекта недоступна, обратитесь к администратору.

Укажите метод загрузки файлов проекта. Доступны следующие методы загрузки: загрузка архива или загрузка с помощью систем управления версиями. Нажмите «Добавить».

Добавление проекта

Название

Загрузите проект в AppChecker, нажав «Выберите файл». Загрузка проекта начнется автоматически.

Дождитесь окончания загрузки.

Загрузка исходных текстов проекта "Project1"

Для C/C++ проектов требуется дополнительная конфигурация.
[Перейти на страницу загрузки утилит и инструкции по конфигурированию.](#)

Загрузите zip-архив с исходными текстами либо архив, полученный в результате конфигурирования проекта утилитами.

[Выберите файл](#) или переместите файл на форму

[Запустить анализ](#) [Удалить проект](#)

Для проектов, написанных на C/ C ++, требуется дополнительная конфигурация для улучшения качества анализа. За более подробной информацией нажмите «Перейти на страницу загрузки утилит и инструкции по конфигурированию».

Запустите анализ проекта, нажав «Запустить анализ».

Статический анализ будет запущен.

Статический анализ проекта "Test1"

Выполнение статического анализа 🔄

4.2 Работа с проектом

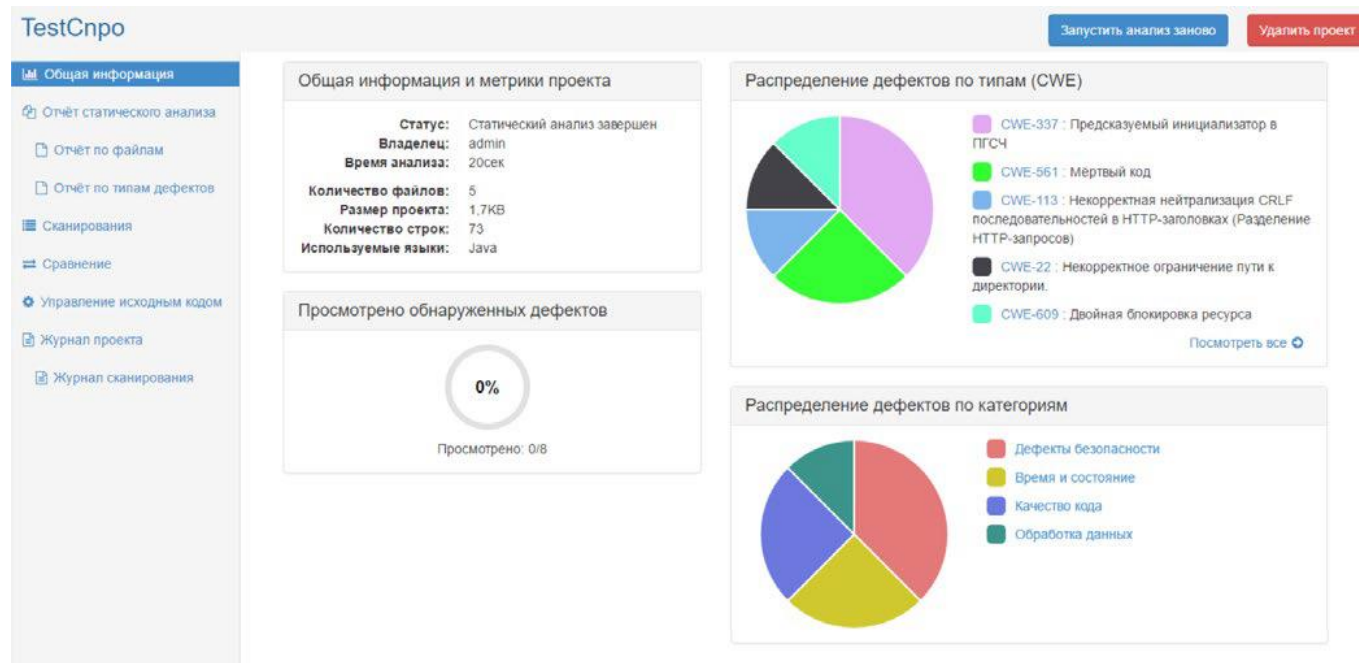
При просмотре главной страницы проекта видна общая информация и метрики проекта, распределения по CWE, по времени, по категориям дефектов и статистики просмотренных срабатываний. Все данные о проекте показываются по результатам последнего анализа.

Распределение CWE показывает количество срабатываний по конкретному дефекту к общему числу срабатываний. Нажмите «Подробнее», чтобы узнать количество срабатываний по определенному дефекту.

AppChecker позволяет многократно производить анализ проекта.

Распределение по времени показывает, сколько дефектов было обнаружено в каждом из сканирований. Подробнее про сканирования см. раздел [6](#).

Распределение по категориям показывает, какая категория дефектов встречается чаще всего.



The screenshot displays the TestCnpo web interface. On the left is a navigation sidebar with options like 'Общая информация', 'Отчёт статического анализа', 'Сканирования', and 'Управление исходным кодом'. The main content area is divided into several sections:

- Общая информация и метрики проекта:** A table with the following data:

Статус:	Статический анализ завершен
Владелец:	admin
Время анализа:	20сек
Количество файлов:	5
Размер проекта:	1,7КВ
Количество строк:	73
Используемые языки:	Java
- Просмотрено обнаруженных дефектов:** A circular progress indicator showing 0% with the text 'Просмотрено: 0/8' below it.
- Распределение дефектов по типам (CWE):** A pie chart with a legend listing:
 - CWE-337: Предсказуемый инициализатор в ПГСЧ
 - CWE-561: Мертвый код
 - CWE-113: Некорректная нейтрализация CRLF последовательностей в HTTP-заголовках (Разделение HTTP-запросов)
 - CWE-22: Некорректное ограничение пути к директории.
 - CWE-609: Двойная блокировка ресурса
- Распределение дефектов по категориям:** A pie chart with a legend listing:
 - Дефекты безопасности
 - Время и состояние
 - Качество кода
 - Обработка данных

Диаграммы распределения по CWE и по категориям дефектов позволяют переходить на страницу выбранного дефекта и категории. Для этого нажмите на название дефекта/категории дефектов или на часть круговой диаграммы с интересующим дефектом.

Для нового сканирования проекта нажмите «Запустить анализ заново» в правом верхнем углу страницы.

Для удаления проекта нажмите «Удалить проект» в правом верхнем углу экрана.

5 Отчет статического анализа

Отчет статического анализа показывает статус и категорию дефекта, его CWE-идентификатор и фрагмент кода, где дефект был обнаружен. Перейдите на страницу с конкретным дефектом, нажав на него в таблице.

Для сортировки по определенному столбцу нажмите на этот столбец.

Для формирования сводного отчета по статусу или по категории дефекта воспользуйтесь фильтрами рядом с надписями «Статус» и «Категория» соответственно.

Для поиска дефектов по CWE-идентификаторам выберите нужный CWE-идентификатор в ниспадающем списке рядом с надписью «CWE».

Для просмотра дефектов в конкретном файле выберите нужный файл в ниспадающем списке рядом с надписью «Фрагмент кода».

Отчёт статического анализа

Статус	Категория	CWE	Фрагмент кода
[все] x	[все] x	[все] x	[все] x
Новый	Обработка ошибок	CWE-390: Пустой блок обработки условий и ошибок	CWE-390_1556e428-38f4-48ba-ae1d-d79dc66ba03c/1.php: 6: } catch (Exception \$e) { 7: ... 8: }
Новый	Качество кода	CWE-676: Использование потенциально опасной функции	CWE-506_92436eb8-bd10-4053-ba78-33553edc924d/1.php: 3: eval(base64_decode('DQoJcQkLly9GB3IgzXhjZhwgZmlsZS=='));
Новый	Качество кода	CWE-676: Использование потенциально опасной функции	CWE-506_89d049e9-e357-44ec-82ea-da02851bcd6d/1.php: 10: eval(\$v);
Новый	Качество кода	CWE-676: Использование потенциально опасной функции	CWE-506_92436eb8-bd10-4053-ba78-33553edc924d/2.php: 3: eval(gzinflate(base64_decode(str_rot13('7'))));
Новый	Качество кода	CWE-676: Использование потенциально опасной функции	CWE-506_89d049e9-e357-44ec-82ea-da02851bcd6d/2.php: 10: eval(\$v);
Подтвержден	Качество кода	CWE-482: Сравнение вместо присваивания	CWE-482_58a60841-e380-4720-abc9-0400cbc6107c/1.php: 3: \$a == \$b;
Новый	Качество кода	CWE-482: Сравнение вместо присваивания	CWE-482_58a60841-e380-4720-abc9-0400cbc6107c/2.php: 5: \$a == \$b;
Новый	Качество кода	CWE-398: Признак плохого качества кода	CWE-398_05f62a7c-86e2-4a3a-b7ba-648a3abbd130/2.php: 3: if (\$a) { 4: \$b++; 5: fu(); 6: ...

5.1 Работа с конкретным дефектом

Перейдите на страницу с конкретным обнаруженным дефектом, нажав на нужную строчку в таблице. Если дефект состоит из нескольких частей, то можно переходить к ним, используя кнопку «Перейти к следующей части дефекта» слева от надписи «Исходный код файла».

CWE-79: Некорректная нейтрализация пользовательских данных, используемые для генерации web-страниц (XSS, межсайтовый скриптинг)

Программное обеспечение не нейтрализует (некорректно нейтрализует) пользовательские данные перед тем, как они будут использованы для генерации веб-страниц.

[Посмотреть подробнее на cwe.s3r.ru](#)

Статус: Комментарий:

Исходный код файла 3.php

```
1 <?php
2
3 function foo($b) {
4
5     echo $b;
6
7 }
```

Измените статус дефекта и прокомментируйте свое решение.

Выберите статус «Ложное срабатывание» или «Подтвержден» из ниспадающего списка, введите комментарий и нажмите «Применить».

По умолчанию, статусом всех дефектов является «Новый». Процент обработанных экспертом срабатываний показан на главной странице проекта.

Обратите внимание, что подавлять ложные срабатывания можно в исходном коде проекта. Более подробную информацию смотрите в пункте [5.2](#).

5.2 Подавление ложных срабатываний

В некоторых случаях анализатор находит дефект кода там, где его нет, и возникает ложное срабатывание.

Чтобы избавиться от ложных срабатываний, укажите в исходных текстах программы в месте перед срабатыванием специальный комментарий «`appchecker CWE-XXX`» или «`appchecker all`», - где XXX – это номер CWE, который надо подавить, all – для подавления всех предупреждений, вне зависимости от типа дефекта.

Для примера рассмотрим код:

```
gets (buf) ;
```

В этом случае анализатор выдаст предупреждение: «CWE – 242: использование опасных функций».

Измените код следующим образом, чтобы подавить ложное срабатывание дефекта CWE-242:

```
// appchecker CWE-242  
gets (buf) ;
```

Для подавления всех срабатываний введите:

```
// appchecker all  
gets (buf) ;
```

5.3 Отчет по файлам и отчет по типам дефектов

Отчеты по файлам и по типам дефектов отображают количество срабатываний.

Нажмите на конкретный файл или тип дефекта, чтобы перейти на «Свободный отчет» с соответствующим фильтром.

Отчёт по файлам

Файл	Количество срабатываний ↕
CWE-506_8c3629a6-d2bf-4f61-b94b-ecbf403049a3/1.php	2
CWE-510_9a235455-7de1-4523-b021-705c52997234/3.php	2
CWE-510_9a235455-7de1-4523-b021-705c52997234/5.php	2
CWE-482_58a60841-e380-4720-abc9-0400cbc6107c/2.php	2
CWE-476_47caff63-8213-4e28-a160-6446d8874eb5/11.php	2

Отчёт по типам дефектов

CWE	Количество срабатываний ↕
CWE-457: Использование неинициализированной переменной	9
CWE-480: Использование неверного оператора	9
CWE-510: Программная закладка	7

Отчёт статического анализа

Статус	Категория	CWE	Фрагмент кода
[все] X	[все] X	[все] X	CWE-457_79e06d3a-4818-4f7a-9211-8970ed2968f2/2.php X
Новый	Качество кода	CWE-457: Использование неинициализированной переменной	CWE-457_79e06d3a-4818-4f7a-9211-8970ed2968f2/2.php: 6: \$a = \$x;
⏪ ⏩ Стр. 1 из 5 ⏪ ⏩			Просмотр 1 - 1 из 84

6 Сканирования

AppChecker позволяет многократно анализировать один проект.

Нажмите «Запустить анализ» на главной странице проекта для повторного сканирования с текущим исходным кодом.

Нажмите «Управление исходным кодом» для изменения исходного кода.

Повторный анализ проекта

Запустить анализ

Анализ будет запущен с текущим исходным кодом. При необходимости Вы можете его изменить:
[Управление исходным кодом](#)

Все проведенные анализы над проектом представлены во вкладке «Сканирование».

Просмотрите результат конкретного сканирования, нажав «Посмотреть отчет».

Нажмите «Удалить» для удаления сканирования.

Сканирования

Время начала анализа	Время окончания анализа	Статус	Действие
15 апр 2016 г., 07:29:53	15 апр 2016 г., 07:30:10	Анализ успешно завершён	посмотреть отчёт удалить
15 апр 2016 г., 07:33:22	15 апр 2016 г., 07:33:37	Анализ успешно завершён	посмотреть отчёт удалить
15 апр 2016 г., 07:34:45	15 апр 2016 г., 07:34:59	Анализ успешно завершён	посмотреть отчёт удалить

Сравните сканирования между собой в разделе «Сравнения», выбрав два сканирования для сравнения.

Сравнение

Выберите два сканирования для сравнения:

15 апр 2016 г., 07:33:22 - 15 апр 2016 г., 07:33:37

15 апр 2016 г., 07:34:45 - 15 апр 2016 г., 07:34:59

Сравнить

Новые (11)

Пропавшие (11)

Оставшиеся (73)

Статус	Категория	CWE	Фрагмент кода
Новый	Качество кода	CWE-676: Использование потенциально опасной функции	CWE-506_92436eb8-bd10-4053-ba78-33553edc924d/1.php: 3: <code>eval(base64_decode('DQoJCQkJLy9Gb3IgzXhjZlWmgZmlsZS=='));</code>
Новый	Качество кода	CWE-676: Использование потенциально опасной функции	CWE-506_89d049e9-e357-44ec-82ea-da02851bcd6d/1.php: 10: <code>eval(\$v);</code>
Новый	Качество кода	CWE-676: Использование потенциально опасной функции	CWE-506_92436eb8-bd10-4053-ba78-33553edc924d/2.php: 3: <code>eval(gzinflate(base64_decode(str_rot13('7'))));</code>

7 Управление проектом

AppChecker позволяет загрузить новые версии исходного код.

Измените версию исходного кода, загрузив архив с проектом.

При использовании систем управления версией программа будет обновлять исходный код автоматически.

Управление исходным кодом

Исходные тексты загружены. Вы можете загрузить новый архив.
Внимание! Текущие исходные файлы будут удалены!

Выберите файл

8 Журналы

Журнал проекта предназначен для регистрации происходящих событий в проекте.

Журнал проекта

```
[25.08.16 17:47:59] TestCnpo User: admin, Project created
[25.08.16 17:48:14] TestCnpo User: admin, Uploading file (source archive)...
[25.08.16 17:48:14] TestCnpo User: admin, chunk:[0] chunks:[1]
[25.08.16 17:48:14] TestCnpo User: admin, Source unpacking...
[25.08.16 17:48:14] TestCnpo unpacking archive...
[25.08.16 17:48:32] TestCnpo User: admin, Running analyzer with parameters: java -Xmx6697766092 -jar appchecker.jar -n TestCnpo -s ../data/projects
[25.08.16 17:54:22] TestCnpo User: admin, Uploading file (source archive)...
[25.08.16 17:54:22] TestCnpo User: admin, chunk:[0] chunks:[1]
[25.08.16 17:54:22] TestCnpo User: admin, Source unpacking...
[25.08.16 17:54:22] TestCnpo unpacking archive...
[25.08.16 17:54:26] TestCnpo User: admin, Running analyzer with parameters: java -Xmx6697766092 -jar appchecker.jar -n TestCnpo -s ../data/projects
```

Журнал сканирования предназначен для регистрации происходящих событий для каждого конкретного сканирования. Для просмотра журнала выберите сканирование из выпадающего списка.

Журнал сканирования

#2 25 авг 2016 г., 17:54:26 - 25 авг 2016 г., 17:54:48 ▾

```
Starting project execution at 2016/08/25 17:54:26
JVM memory settings:
Total memory: 125829120
Free memory: 121122616
Maximum memory: 5954338816
Starting project execution at 2016/08/25 17:54:26
Launching source loader with arguments -r -e java,c,cc,cpp,cxx,h,hpp,php,php5,php4,inc,cs -i ../data/projects/TestCnpo/src/ -p ../data/projects/
Starting import...
Import successfully completed
Starting project execution at 2016/08/25 17:54:26
Launching java parser with arguments ../data/projects/TestCnpo/prj/ -tA at 2016/08/25 17:54:26
parsing started
parsing finished
Launching parser-cs with arguments ../data/projects/TestCnpo/prj/ -tA at 2016/08/25 17:54:29
Appchecker mode
C# parser started
C# parser finished
Launching parser-cpp-sa with arguments ../data/projects/TestCnpo/prj/ at 2016/08/25 17:54:29
Project doesn't have C/C++ files
Launching parser-php-sa with arguments ../data/projects/TestCnpo/prj/ at 2016/08/25 17:54:29
Project doesn't have PHP files
Starting project execution at 2016/08/25 17:54:29
Launching Appchecker with arguments -p ../data/projects/TestCnpo/prj/ -c configs -r cwe_base at 2016/08/25 17:54:29
Analyze started.
```

9 Консольный клиент

Консольный клиент `appchecker-rest-client.jar` расположен в директории сервиса. также консольный клиент доступен для скачивания во вкладке «О программе».

Запустите консольный клиент, используя следующую команду:

```
java -jar appchecker-rest-client.jar [параметры]
```

Все доступные параметры приведены в конце данного раздела.

Запуск анализа проекта

Запустите анализ проекта, задав следующие параметры:

```
-i, --input <входная_директория>
```

Параметром могут быть директория, файл с расширением TRE или zip-архив. Если в качестве параметра задана директория, то она будет упакована в zip-архив. Если в качестве параметра задан файл с расширением TRE, то в zip-архив будут упакованы те файлы, на которые он указывает.

```
-n, --name <имя_проекта>
```

Имя проекта указывается в кавычках. Имя проекта по умолчанию – Project.

```
-o, --output <выходная_директория>
```

Директория для отчетов о проведенном анализе.

Пример запуска анализа проекта с именем Project1 приведен ниже.

```
java -jar appchecker-rest-client.jar -i  
C:\Projects\Project1\source_code.zip -n "Project1" -o  
C:\Projects\Project1\Results
```

Удаление проекта

Удалите проект, указав следующие параметры:

-d, --delete

Удаление проекта.

-n, --name <имя_проекта>

Имя проекта указывается в кавычках. Имя проекта по умолчанию – Project.

Пример удаления проекта с именем Project1 приведен ниже.

```
java -jar appchecker-rest-client.jar -d -n "Project1"
```

Справка

Для получения справки введите следующий параметр:


-h, --help


Все доступные параметры консольного клиента приведены в таблице ниже.

Параметр	Описание
-i,--input <входная_директория>	Входная директория.
-n,--name <имя_проекта>	Имя проекта. По умолчанию – Project.
-o,--output <выходная_директория>	Выходная директория.
-d,--delete	Удаление проекта.
-h,--help	Вызов справки.
-p,--port <порт_сервиса>	Порт сервиса. По умолчанию – 11000.
-s,--server <адрес_сервиса>	IP или доменный адрес сервиса. По умолчанию - http://127.0.0.1
-t,--timeout <время_таймаута>	Время таймаута в секундах. По умолчанию – 30.
-u,--user <имя_пользователя>	Имя пользователя. По умолчанию – admin.
-pw,--password <пароль>	Порт сервиса. По умолчанию – 11000.

10 Поддержка

Вкладка «О программе» содержит информацию о версии AppChecker, информацию о лицензии, ссылку для загрузки консольного клиента AppChecker, ссылки для загрузки утилит конфигурации и ссылку на банк данных угроз безопасности информации и уязвимостей ПО (ФСТЭК).

 **AppChecker** | [Проекты](#) | [Администрирование](#) | **[О программе](#)**

 **AppChecker** версия 2.4.50 ([Проверить на наличие новой версии](#))
© 2017 АО "НПО "Эшелон"
Техническая поддержка: support@cnpo.ru
Сайт продукта: appchecker.cnpo.ru

Информация о лицензии

Номер лицензии: 12792
Владелец лицензии: Infosecurity
Срок действия лицензии до: 6 сентября 2018 г.
Максимальное число проектов: 1024
Доступные парсеры: C/C++, Java, PHP, C#

Утилиты, вспомогательные скрипты и инструкции

[Настройка парсера C/C++ \(PDF\)](#)
[Консольный клиент](#)
[Плагин для Jenkins/Hudson](#)
[Плагин для TeamCity](#)
[Утилита подготовки конфигурации srrconf_linux \(для проектов, собирающихся в среде Linux\)](#)
[Обёртка для gcc/clang \(gcc_conf.sh\)](#)
[Обёртка для g++/clang++ \(g++_conf.sh\)](#)
[Утилита подготовки конфигурации srrconf_monitor \(для проектов, собирающихся в среде Windows\)](#)
[Утилита подготовки конфигурации srrconf_source \(конфигурация без сборки\)](#)

Справочная информация

[Банк данных угроз безопасности информации и уязвимостей ПО \(ФСТЭК России\)](#)
[Common Weakness Enumeration \(CWE\)](#)

Для проверки актуальности версии нажмите на ссылку «Проверить на наличие новой версии».

Для получения технической поддержки отправьте электронное письмо на адрес support@cnpo.ru.

11 Глоссарий

Дефект кода

Ошибка в исходном тексте программы, которая может привести к уязвимости.

Категория дефектов кода

В зависимости от типа дефекты кода делятся на 8 категорий: дефекты безопасности, качество кода, окружения, обработки ошибок, дефекты API, обработка данных, программные закладки, время и состояние.

Конфигурация проекта

Настройка проекта для повышения качества анализа.

Ложное срабатывание

Ситуация, когда фрагмент кода ошибочно отмечается программой как содержащий дефект.

Программная закладка

Скрытно внедренный в систему код, позволяющий злоумышленнику осуществить несанкционированный доступ.

Роль

Набор разрешений, соответствующий тому, какие действия может выполнять пользователь или группа пользователей.

Система управления версиями

Программное обеспечение, позволяющее хранить несколько версий одного документа.

Статический анализ

Анализ программного обеспечения, производимый без реального выполнения исследуемых программ.

CWE

Common Weakness Enumeration. Представляет собой список дефектов безопасности программного обеспечения. CWE охватывает более 15000 дефектов безопасности и включает в себя детальную структуру классификации из различных научных источников и примеров.

CWE – совместимость

Позволяет продукту быть зарегистрированным в качестве официального «CWE-совместимого». CWE-совместимый продукт должен представлять пользователям возможность поиска дефектов безопасности в CWE-идентификаторам и получения связанных с ними CWE-идентификаторов.

12 0 компании

АО «НПО «Эшелон» специализируется на комплексном обеспечении информационной безопасности.

Основными направлениями деятельности являются:

- ❖ проектирование, внедрение и сопровождение комплексных систем обеспечения информационной безопасности;
- ❖ сертификация, аттестация и лицензирование в области защиты информации;
- ❖ аудит информационной безопасности организаций;
- ❖ проектирование и аудит систем управления информационной безопасностью;
- ❖ разработка и поставка оборудования и средств защиты информации, в том числе в защищенном исполнении;
- ❖ обеспечение технической поддержки и сопровождение поставляемых решений, систем и продуктов;
- ❖ испытания, экспертизы, исследования в области безопасности информации.

Более детальную информацию о компании вы сможете найти на сайте npo-echelon.ru.