SYHUNT CODE: GETTING STARTED

The information in this document applies to **version 6.9.1** of Syhunt Code.

HOW TO PERFORM A CODE SCAN

Syhunt's whitebox scan (source code scan) can uncover multiple classes of application vulnerabilities and also identify key areas of the code that need review. Its static source code analysis functionality can detect over 40 vulnerability types, including the 2019 CWE Top 25 Most Dangerous Software Errors and the OWASP mobile top 10 security risks. Initially only PHP was supported. As of today, multiple web and mobile programming languages are supported.

SUPPORTED LANGUAGES (WEB)

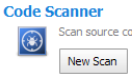
</> ASP Classic (VBScript & JavaScript)
</> ASP.Net (C# & VB.Net)
</> Java (JEE / JSP)
</> JavaScript (Client and Server-Side, Node.js, Angular, AngularJS, Express.js & Koa.js)
</> Lua (ngx_lua, mod_lua, CGILua & Lua Pages)
</> Perl
</> PHP
</> Python (CGI, Django, mod_python & WSGI)
</> Ruby (Rails & ERB)
</> TypeScript (Angular)

SUPPORTED LANGUAGES (MOBILE)

</> Java (Android)
</> Swift (iOS)
</> Objective-C, C & C++ (iOS)
</> JavaScript (including Node.js, Angular, AngularJS, Express.js & Koa.js)

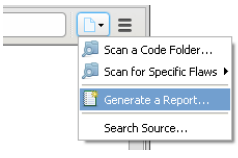
Follow along with this guide to learn how to perform a source code scan and generate a vulnerability report.

1. Launch Syhunt Hybrid and click the Syhunt Code icon or New Scan button in the welcome page.



2. Select a source code directory, source file, APK file or repository to scan.
3. Select a scan method. We recommend the Application Code Scan (Default) method, which scans for all vulnerabilities using the recommended settings - the different methods are explained in the Hunt Methods document.
4. Press the **OK** button to start the scan.

In the end of the scan, you can click Generate a Report to save the results as a HTML report or any other prefered format.



HOW TO PERFORM A CODE SCAN VIA COMMAND-LINE

1. Go to the directory Syhunt is installed using the command prompt.
2. Use the following command-line:

```
scancode [target] -hm:[a huntmethod]] -gr
```

```
# Examples:
scancode git://sub.domain.com/repo.git -gr
scancode https://github.com/user/repo.git -rb:master -gr
scancode c:\source\www\ -gr
scancode c:\source\www\file.php -gr
scancode c:\mobile\myapp.apk -gr
scancode "c:\source code\www\" -gr
```

Syhunt scancode tool reports are automatically generated and saved if the -gr parameter is provided. You can also open the session by launching Syhunt and using the ≡ Menu -> Past Sessions option.

The following parameters can be provided when calling the scancode tool, all of which are **optional**:

Parameter	Description	Default Value
sn:[name]	A session name that must be unique. If omitted, an unique ID will be generated and assigned	auto generated ID
hm:[name]	the Hunt Method to be used during the scan. If omitted, the default method will be used	appscan
rb:[branch]	Sets a repository branch	master
gr	Generates a report file after scanning	
gx	Generates an export file after scanning	
or	Opens report after generation	
er	Emails report after generation	
etrk:[trackername]	Email preferences to be used when emailing report	
esbj:[subject]	Email subject to be used when emailing report	Syhunt Code Report
rout:[filename]	Sets the report output filename and report format	Report_[session name].html
rtpl:[name]	Sets the report template	Standard

xout: [filename]	Sets the export output filename and report format	Export_[session name].xml
xout2: [filename]	Sets a second export output filename and report format	Export_[session name].xml
pfcond: [condition]	Sets a pass/fail condition to be reported	
nv	Turn off verbose. Error and basic info still gets printed	
inc:[mode]	Sets the incremental scan mode	targetpref
inctag:[name]	Optionally stores the incremental scan data within a tag	
refurl:[url]	Sets an URL associated with the current source code for reference purposes only	
noifa	Disables input filtering analysis	
about	Displays information on the current version of Syhunt	
help (or /?)	Displays the list of available parameters	

ADVANCED FEATURES

PREVENTING A CODE VULNERABILITY FROM BEING REPORTED

You can create rules that prevent specific vulnerabilities to be reported:

1. Go to the Code Preferences screen (≡ -> Preferences -> Code Preferences).
2. Go to the Advanced tab and click the **Vulnerabilities...** button
3. Click the plus button and add using the input dialog a new rule. Examples:

- `path=*,name=XSS` would prevent any vulnerability with XSS in the title from being reported
- `path=/demo/*,name=XSS` would prevent any vulnerability with a path starting with /demo/ and XSS in the title from being reported
- `path=*,"name=Web Technology Disclosure"` would prevent any vulnerability with Web Technology Disclosure in the title from being reported

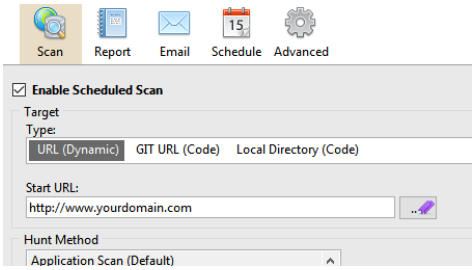
The following parameters can be used as part of a rule:

- **path** (required) - a wildcard text (which can contain the special characters ? and *) that will be matched against the affected path
- **name** - a text that will be matched against the vulnerability title
- **params** - a param name that will be matched against the affected param(s). If multiple params are provided, they must be separated by comma.
- **risk** - a risk that will be matched against the vulnerability risk (can be low, medium, high or info)
- **lines** - a number or numbers that will be matched against the affected source code line(s). If multiple lines are provided, they must be separated by comma.
- **cve** - a CVE ID that will be matched against the vulnerability's CVE references
- **cwe** - a CWE number that will be matched against the vulnerability's CWE references

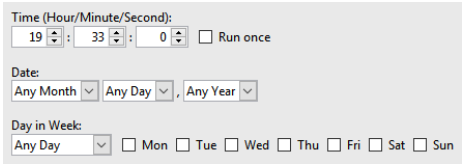
HOW TO SCHEDULE A SCAN

Adding and configuring a scheduled scan is an easy task:

1. Click the Scheduled Scans icon in the launcher toolbar. The Scheduled Scans screen will open.
2. Click the Add Scheduled Scan icon in the Scheduled Scans screen toolbar.
3. Enter a reference name for the new scheduled scan (like MyScan) and hit **OK**. A preferences dialog window will open.
4. In the Scan tab, enter the scan target details and select the desired scan method and options.



5. In the Report tab, enter the desired report generation options.
6. In the Schedule tab, enter the desired event plan.



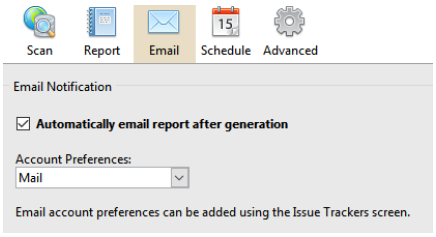
7. Click the **OK** button when you're done.

SENDING REPORTS VIA EMAIL

Firstly, you have to add an Email tracker:

1. Click the Issue Trackers icon in the launcher toolbar. The Issue Trackers screen will open.
2. Click the Add Tracker icon in the Issue Trackers screen toolbar and choose the Add tracker: Email menu option.

3. Enter a reference name for the new tracker (like Mail) and hit **OK**. A preferences dialog window will open.
4. Enter Sender/Recipient email addresses.
5. Enter the SMTP Authentication host and credentials and click the **OK** button.
6. Click the Scheduled Scans icon in the launcher toolbar. The Scheduled Scans screen will open.
7. Right-click the scheduled scan and click the Edit Schedule Preferences option. A preferences dialog window will open.
8. Go to the Email tab and check the **Automatically email report after generation** option.



9. Select the account preferences.
10. Click the **OK** button when you're done.

REVIEWING RESULTS FROM SCHEDULED SCANS

At any time you can see the results of past and current scans and generate a report. Just launch the Syhunt Hybrid application and click the Past Sessions icon in the launcher toolbar.

WORKING WITH THIRD-PARTY LAUNCHERS AND SCHEDULERS

See this document on how to start Syhunt from within third-party task schedulers, Jenkins and other launchers

SYSTEM REQUIREMENTS

Syhunt Hybrid (including its Community Edition) can be installed on 64-bit Windows or 64-bit Linux, but it is able to analyze applications designed for any target platform, including Android, Apple iOS and MacOS, BSD, Linux, Windows, Solaris and Unix, independently of the platform it is executed from.

1. 4GB of available RAM (8GB recommended)
2. 1GB of free disk space*
3. Internet Connection (recommended for code scans and dynamic scans and some features)
4. One of the following compatible 64-bit operating systems:
    1. Windows 7, 8 or 10, or Windows Server 2008 to 2019
    2. Ubuntu Server or Desktop 18 or higher
    3. CentOS 7 or 8 (Minimal or Everything)
    4. Any unofficially supported Linux distribution such as the ones listed below.

5. (Optional) GIT on Linux or GIT for Windows (optional for GIT repository scans)
6. Java or Java Headless installed on Linux OS
7. If native binary is not available for your specific Linux distribution yet, Wine64 Stable (3, 4 or 5) is required to be installed.
8. (Optional) Java 8 or higher (optional for Android APK file scan)

* This does not include the space required to save scan session data, which varies depending on the website or source code being analyzed and the scan frequency.

COMPATIBLE LINUX DISTRIBUTIONS

Officially Supported:

✔ Ubuntu Server/Desktop 18.10 and later
✔ CentOS 7.7 and later (Minimal or Everything)

Unofficially (Successfully Tested):

✔ Kali Linux 2019 and later
✔ Parrot OS 4.1, 4.7 and later
✔ Debian 9.11 and later
✔ Linux Mint 19.2 and later
✔ OpenSUSE Leap 15.1 and later
✔ Fedora 32
✔ MX Linux 19.1 and later
✔ KDE Neon 2020.03 and later
✔ Deepin 15.9
✔ Manjaro 19
✔ Arch Linux 2019 and later

Unsupported:

⊘ Elementary OS 5.1 (Successfully Tested), 5.0 (Unsupported)
⊘ CentOS 6.1 (Successfully Tested)
⊘ Solus 4.1 (Unstable)