# INTEGRATING SYHUNT WITH AZURE DEVOPS

The information in this document applies to **version 6.9.6** of Syhunt Hybrid.

Since version 6.9.6, both Syhunt Community and Hybrid are able to connect to Azure DevOps Services, Azure DevOps Server or Team Foundation Server (TFS) and scan the source code of projects for application security vulnerabilities and weaknesses.

The new feature is available through Syhunt's GUI and CLI for Windows and allows to scan on-premise and cloud-based TFS and GIT projects. By default, Syhunt is compatible with today's Azure DevOps Services, Azure DevOps Server 2020 and 2019, and TFS 2018 down to 2015. If needed, as explained below, you can configure Syhunt to connect to older TFS versions such as 2013, 2012 and 2010.

In addition to the TFS support, Syhunt offers PowerShell integration which allows Syhunt to be easily integrated into DevOps environments.

**The TFS connection support in Syhunt Community and Hybrid relies on the export tool developed by Microsoft MVP Neno Loje and freely available from his website. The tool was included in the setup of Syhunt with permission of the developer.**

## SUPPORTED VERSIONS

| | |
|---|---|
| ☁ Azure DevOps Services 2021 | Cloud |
| Azure DevOps Server 2020 / 2019 | On-premises |
| Team Foundation Server 2018 / 2015 | On-premises |
| Team Foundation Server 2013 / 2012 / 2010 ✱ | On-premises |

✱ To scan projects hosted in TFS 2013 - 2010 servers only: requires Microsoft Visual Studio or Microsoft Team Explorer to be installed on the same machine.
✱ To scan GIT repositories instead of just TFS repositories, you will need to download and install Git for Windows on the same machine.

# SCANNING PROJECTS

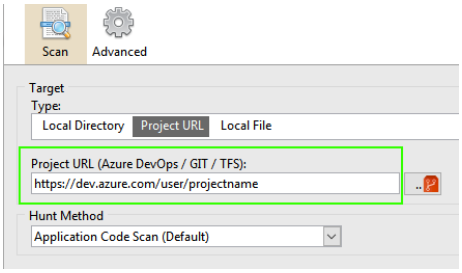
The following are a few examples of valid Azure DevOps / TFS project targets in Syhunt:

| | |
|---|---|
| Azure DevOps Services - TFVC repository | `https://dev.azure.com/user/projectname` |
| Azure DevOps Services - GIT repository | `https://dev.azure.com/user/projectname/_git/projectname` |
| Azure DevOps Server | `https://myserver/tfs/projectname` |
| Azure DevOps Services / Server / TFS | `collection:https://dev.azure.com/user$/projectname` |

In the last example, Syhunt will connect to the project collection at https://dev.azure.com/user and scan the server path $/projectname.

# SCANNING VIA USER INTERFACE

From the Syhunt Hybrid or Community GUI, as shown in the Syhunt Code quick start guide and the image below, after clicking the New scan button, selecting Project URL, and entering a valid target URL, you can start a scan against a Azure DevOps or TFS/TFVC project, or a GIT-based Azure DevOps repository. Through the Scan Scheduler, you can also schedule code scans against specific project URLs.




Your Azure DevOps credentials will be asked before connecting, unless you already authenticated to it before.

By default, Syhunt is compatible with today's Azure DevOps Services, Azure DevOps Server 2020 and 2019, and TFS 2018 down to 2015. Optionally, an older version, such as 2003 to 2010 can be set through the Advanced tab, but as explained before, this requires Microsoft Visual Studio or Microsoft Team Explorer to be installed on the same machine for it to work.

# SCANNING VIA COMMAND-LINE INTERFACE

Alternatively, from the Syhunt Hybrid or Community CLI, as shown below, you can scan a DevOps or TFS project.

```
scancode https://dev.azure.com/user/projectname
scancode https://myserver/tfs/projectname
scancode collection:https://dev.azure.com/user$/projectname
```

Your Azure DevOps credentials will be asked before connecting, unless you already authenticated to it before.

By default, Syhunt is compatible with today's Azure DevOps Services, Azure DevOps Server 2020 and 2019, and TFS 2018 down to 2015. Optionally, as shown below, you can specify an older version of the TFS server through the -tfsv parameter.

```
scancode https://myserver/tfs/projectname -tfsv:2013
scancode https://myserver/tfs/projectname -tfsv:2012
scancode https://myserver/tfs/projectname -tfsv:2010
```

More command-line usage examples and information can be found in the Syhunt CLI guide.

## SCRIPTING INTEGRATION

Today, scripting/pipeline integration can be implemented through:

- **PowerShell**, as documented in the PowerShell integration guide
- **CLI**, as documented above and in the CLI guide.