Msbuild on LOLBAS

Archived: 2026-04-05 18:14:03 UTC

.. /Msbuild.exe

Used to compile and execute code

Paths:

- C:\Windows\Microsoft.NET\Framework\v2.0.50727\Msbuild.exe
- C:\Windows\Microsoft.NET\Framework64\v2.0.50727\Msbuild.exe
- C:\Windows\Microsoft.NET\Framework\v3.5\Msbuild.exe
- C:\Windows\Microsoft.NET\Framework64\v3.5\Msbuild.exe
- C:\Windows\Microsoft.NET\Framework\v4.0.30319\Msbuild.exe
- C:\Windows\Microsoft.NET\Framework64\v4.0.30319\Msbuild.exe
- C:\Program Files (x86)\MSBuild\14.0\bin\MSBuild.exe

Resources:

- https://github.com/redcanaryco/atomic-red-team/blob/master/atomics/T1127/T1127.md
- https://github.com/Cn33liz/MSBuildShell
- https://pentestlab.blog/2017/05/29/applocker-bypass-msbuild/
- https://oddvar.moe/2017/12/13/applocker-case-study-how-insecure-is-it-really-part-1/
- https://gist.github.com/bohops/4ffc43a281e87d108875f07614324191
- https://github.com/LOLBAS-Project/LOLBAS/issues/165
- https://docs.microsoft.com/en-us/visualstudio/msbuild/msbuild-response-files
- https://www.daveaglick.com/posts/msbuild-loggers-and-logging-events

Acknowledgements:

- Casey Smith (@subtee)
- Cn33liz (@Cneelis)
- Jimmy (@bohops)

Detections:

- Sigma: file_event_win_shell_write_susp_directory.yml
- Sigma: proc_creation_win_msbuild_susp_parent_process.yml
- Sigma: net_connection_win_silenttrinity_stager_msbuild_activity.yml
- Splunk: suspicious_msbuild_spawn.yml
- Splunk: suspicious_msbuild_rename.yml
- Splunk: msbuild_suspicious_spawned_by_script_process.yml

- Elastic: defense_evasion_msbuild_beacon_sequence.toml
- Elastic: defense_evasion_msbuild_making_network_connections.toml
- Elastic: defense_evasion_execution_msbuild_started_by_script.toml
- Elastic: defense_evasion_execution_msbuild_started_by_office_app.toml
- Elastic: defense_evasion_execution_msbuild_started_renamed.toml
- BlockRule: https://docs.microsoft.com/en-us/windows/security/threat-protection/windows-defender-application-control/microsoft-recommended-block-rules
- IOC: Msbuild.exe should not normally be executed on workstations

AWL bypass

1. Build and execute a C# project stored in the target XML file.

   ```
   msbuild.exe file.xml
   ```

   Use case

   Compile and run code

   Privileges required

   User

   Operating systems

   Windows vista, Windows 7, Windows 8, Windows 8.1, Windows 10, Windows 11

   ATT&CK® technique

   T1127.001: MSBuild

   Tags

   Execute: CSharp

Execute

1. Build and execute a C# project stored in the target csproj file.

   ```
   msbuild.exe file.csproj
   ```

   Use case

   Compile and run code

   Privileges required

   User

   Operating systems

   Windows vista, Windows 7, Windows 8, Windows 8.1, Windows 10, Windows 11

   ATT&CK® technique

T1127.001: MSBuild

Tags

Execute: CSharp

2. Executes generated Logger DLL file with TargetLogger export.

```
msbuild.exe /logger:TargetLogger,C:\Windows\Temp\file.dll;MyParameters,Foo
```

Use case

Execute DLL

Privileges required

User

Operating systems

Windows vista, Windows 7, Windows 8, Windows 8.1, Windows 10, Windows 11

ATT&CK® technique

T1127.001: MSBuild

Tags

Execute: DLL

3. Execute JScript/VBScript code through XML/XSL Transformation. Requires Visual Studio MSBuild v14.0+.

```
msbuild.exe file.proj
```

Use case

Execute project file that contains XslTransformation tag parameters

Privileges required

User

Operating systems

Windows vista, Windows 7, Windows 8, Windows 8.1, Windows 10, Windows 11

ATT&CK® technique

T1127.001: MSBuild

Tags

Execute: XSL

4. By putting any valid msbuild.exe command-line options in an RSP file and calling it as above will interpret the options as if they were passed on the command line.

```
msbuild.exe @file.rsp
```

Use case

Bypass command-line based detections

Privileges required

User

Operating systems

Windows vista, Windows 7, Windows 8, Windows 8.1, Windows 10, Windows 11

ATT&CK® technique

T1036: Masquerading

Tags

Execute: CMD
