T1027.004 - Obfuscated Files or Information: Compile After Delivery

Description from ATT&CK

> Adversaries may attempt to make payloads difficult to discover and analyze by delivering files to victims as uncompiled code. Text-based source code files may subvert analysis and scrutiny from protections targeting executables/binaries. These payloads will need to be compiled before execution; typically via native utilities such as csc.exe or GCC/MinGW.(Citation: ClearSky MuddyWater Nov 2018)
> Source code payloads may also be encrypted, encoded, and/or embedded within other files, such as those delivered as a Phishing. Payloads may also be delivered in formats unrecognizable and inherently benign to the native OS (ex: EXEs on macOS/Linux) before later being (re)compiled into a proper executable binary with a bundled compiler and execution framework.(Citation: TrendMicro WindowsAppMac)

Atomic Tests

- Atomic Test #1 - Compile After Delivery using csc.exe
- Atomic Test #2 - Dynamic C# Compile
- Atomic Test #3 - C compile
- Atomic Test #4 - CC compile
- Atomic Test #5 - Go compile

Atomic Test #1 - Compile After Delivery using csc.exe

Compile C# code using csc.exe binary used by .NET Upon execution an exe named T1027.004.exe will be placed in the temp folder

Supported Platforms: Windows

auto_generated_guid: ffcdbd6a-b0e8-487d-927a-09127fe9a206

Inputs:

Name	Description	Type	Default Value
output_file	Output compiled binary	path	C:\Windows\Temp\T1027.004.exe
input_file	C# code that launches calc.exe from a hidden cmd.exe Window	path	PathToAtomicsFolder\T1027.004\src\calc.cs

Attack Commands: Run with `command_prompt`!

```
C:\Windows\Microsoft.NET\Framework64\v4.0.30319\csc.exe /out:#{output_file} #{inpu
```

Cleanup Commands:

```
del #{output_file} >nul 2>&1
```

Dependencies: Run with `powershell`!

Description: C# file must exist on disk at specified location (#{input_file})

Check Prereq Commands:

```
if (Test-Path #{input_file}) {exit 0} else {exit 1}
```

Get Prereq Commands:

```
New-Item -Type Directory (split-path #{input_file}) -ErrorAction ignore | Out-Null
Invoke-WebRequest "https://github.com/redcanaryco/atomic-red-team/raw/master/atomi
```

Atomic Test #2 - Dynamic C# Compile

When C# is compiled dynamically, a .cmdline file will be created as a part of the process. Certain processes are not typically observed compiling C# code, but can do so without touching disk. This can be used to unpack a payload for execution. The exe file that will be executed is named as T1027.004_DynamicCompile.exe is contained in the 'bin' folder of this atomic, and the source code to the file is in the 'src' folder. Upon execution, the exe will print 'T1027.004 Dynamic Compile'.

Supported Platforms: Windows

auto_generated_guid: 453614d8-3ba6-4147-acc0-7ec4b3e1faef

Inputs:

Name	Description	Type	Default Value
input_file	exe program containing dynamically compiled C# code	path	PathToAtomicsFolder\T1027.004\bin\T1027.004_DynamicComp

Attack Commands: Run with `powershell`!

```
Invoke-Expression #{input_file}
```

Dependencies: Run with `powershell`!

Description: exe file must exist on disk at specified location (#{input_file})

Check Prereq Commands:

```
if (Test-Path #{input_file}) {exit 0} else {exit 1}
```

Get Prereq Commands:

```
Invoke-WebRequest https://github.com/redcanaryco/atomic-red-team/raw/master/atomic
```

Atomic Test #3 - C compile

Compile a c file with either gcc or clang on Linux or Macos.

Supported Platforms: Linux, macOS

auto_generated_guid: d0377aa6-850a-42b2-95f0-de558d80be57

Inputs:

Name	Description	Type	Default Value
input_file	source file	path	PathToAtomicsFolder/T1027.004/src/T1027-004-test.c

Attack Commands: Run with `bash`!

```
gcc #{input_file} && ./a.out
clang #{input_file} && ./a.out
```

Dependencies: Run with `sh`!

Description: the source file must exist on disk at specified location (#{input_file})

Check Prereq Commands:

```
if [ -e  #{input_file} ]; then exit 0; else exit 1; fi
```

Get Prereq Commands:

```
wget https://github.com/redcanaryco/atomic-red-team/raw/master/atomics/T1027.004/s
```

Atomic Test #4 - CC compile

Compile a c file with either gcc or clang on Linux or Macos.

Supported Platforms: Linux, macOS

auto_generated_guid: da97bb11-d6d0-4fc1-b445-e443d1346efe

Inputs:

Name	Description	Type	Default Value
input_file	source file	path	PathToAtomicsFolder/T1027.004/src/T1027-004-test.cc

Attack Commands: Run with `bash`!

```
g++ #{input_file} && ./a.out
clang++ #{input_file} && ./a.out
```

Dependencies: Run with `sh`!

Description: the source file must exist on disk at specified location (#{input_file})

Check Prereq Commands:

```
if [ -e  #{input_file} ]; then exit 0; else exit 1; fi
```

Get Prereq Commands:

```
wget https://github.com/redcanaryco/atomic-red-team/raw/master/atomics/T1027.004/s
```

Atomic Test #5 - Go compile

Compile a c file with either gcc or clang on Linux or Macos.

Supported Platforms: Linux, macOS

auto_generated_guid: 78bd3fa7-773c-449e-a978-dc1f1500bc52

Inputs:

Name	Description	Type	Default Value
input_file	source file	path	PathToAtomicsFolder/T1027.004/src/T1027-004-test.go

Attack Commands: Run with `bash`!

```
go run #{input_file}
```

Dependencies: Run with `sh`!

Description: the source file must exist on disk at specified location (#{input_file})

Check Prereq Commands:

```
if [ -e  #{input_file} ]; then exit 0; else exit 1; fi
```

Get Prereq Commands:

```
wget https://github.com/redcanaryco/atomic-red-team/raw/master/atomics/T1027.004/s
```