

- >  T1037.001
- >  T1037.002
- >  T1037.004
- >  T1037.005
- >  T1039
- >  T1040

Atomic Test #1 - Make and modify binary from C source

Make, change owner, and change file attributes on a C source code file

Supported Platforms: macOS, Linux

auto_generated_guid: 896dfe97-ae43-4101-8e96-9a7996555d80

Inputs:

| Name | Description | Type | Default Value |
|---------|-----------------|------|---|
| payload | hello.c payload | Path | PathToAtomicsFolder/T1548.001/src/hello.c |

Attack Commands: Run with  **sh** ! Elevation Required (e.g. root or admin)

```
cp #{payload} /tmp/hello.c
sudo chown root /tmp/hello.c
sudo make /tmp/hello
sudo chown root /tmp/hello
sudo chmod u+s /tmp/hello
/tmp/hello
```

Cleanup Commands:

```
sudo rm /tmp/hello
sudo rm /tmp/hello.c
```

Atomic Test #2 - Set a SetUID flag on file

This test sets the SetUID flag on a file in Linux and macOS.

Supported Platforms: macOS, Linux

auto_generated_guid: 759055b3-3885-4582-a8ec-c00c9d64dd79

Inputs:

| Name | Description | Type | Default Value |
|----------------|---------------------------------|------|-----------------|
| file_to_setuid | Path of file to set SetUID flag | Path | /tmp/evilBinary |

Attack Commands: Run with  **sh** ! Elevation Required (e.g. root or admin)

```
sudo touch #{file_to_setuid}
sudo chown root #{file_to_setuid}
sudo chmod u+s #{file_to_setuid}
```

Cleanup Commands:

```
sudo rm #{file_to_setuid}
```

Atomic Test #3 - Set a SetGID flag on file

This test sets the SetGID flag on a file in Linux and macOS.

Supported Platforms: macOS, Linux

auto_generated_guid: db55f666-7cba-46c6-9fe6-205a05c3242c

Inputs:

| Name | Description | Type | Default Value |
|----------------|---------------------------------|------|-----------------|
| file_to_setuid | Path of file to set SetGID flag | Path | /tmp/evilBinary |

Attack Commands: Run with `sh` ! Elevation Required (e.g. root or admin)

```
sudo touch #{file_to_setuid}
sudo chown root #{file_to_setuid}
sudo chmod g+s #{file_to_setuid}
```

Cleanup Commands:

```
sudo rm #{file_to_setuid}
```

Atomic Test #4 - Make and modify capabilities of a binary

Make and modify [capabilities](#) of a C source code file. The binary doesn't have to modify the UID, but the binary is given the capability to arbitrarily modify it at any time with `setuid(0)` . Without being owned by root, the binary can set the UID to 0.

Supported Platforms: Linux

auto_generated_guid: db53959c-207d-4000-9e7a-cd8eb417e072

Inputs:

| Name | Description | Type | Default Value |
|---------|---------------|------|---|
| payload | cap.c payload | Path | PathToAtomicsFolder/T1548.001/src/cap.c |

Attack Commands: Run with `sh` ! Elevation Required (e.g. root or admin)

```
cp #{payload} /tmp/cap.c
make /tmp/cap
sudo setcap cap_setuid=ep /tmp/cap
/tmp/cap
```

Cleanup Commands:

```
rm /tmp/cap
rm /tmp/cap.c
```

Atomic Test #5 - Provide the SetUID capability to a file

This test gives a file the capability to set UID without using flags.

Supported Platforms: Linux

auto_generated_guid: 1ac3272f-9bcf-443a-9888-4b1d3de785c1

Inputs:

| Name | Description | Type | Default Value |
|----------------|---|------|-----------------|
| file_to_setcap | Path of file to provide the SetUID capability | Path | /tmp/evilBinary |

Attack Commands: Run with `sh` ! Elevation Required (e.g. root or admin)

```
touch #{file_to_setcap}
sudo setcap cap_setuid=ep #{file_to_setcap}
```

Cleanup Commands:

```
rm #{file_to_setcap}
```