Security Vulnerability in Maconomy Products

A security vulnerability has been identified in Maconomy components that utilize MScript, the Maconomy scripting language. It is possible to obtain text from the first line of individual files on the web server by crafting URLs to access files outside of the component's configured search path. The severity of this issue is rated as Moderate.

Impacted Maconomy Components

Maconomy MScript, Maconomy Workspace Client, Touch for Maconomy:

All versions (if installed)

Maconomy Portal:

X1 Service Pack 28 and older

2.0 Service Pack 8 and older

2.1 Service Pack 5 and older

2.2

Note: Deltek Maconomy Cloud customers have the patches in place

Mitigation

To limit the security impact of this vulnerability we recommend configuring an error script on all Maconomy components.

Configuring an error script for Maconomy Components

Each Maconomy component has an associated configuration file (.I file). These configuration files are located with the components in the folder 'cgi-bin/Maconomy' below the web server root.

The configuration file name patterns for the affected components are:

MaconomyMScript.<language>.I

MaconomyWS.< language>.I

MaconomyTouch.<language>.I

The following line should be added in the same section as the 'SearchPath' setting:

```
ErrorScript = errorscript.ms
```

The error script file ('errorscript.ms') must be placed in the search path folder for each component, as specified by the setting 'SearchPath'. If more than one file path is specified the error script can be placed in any of the configured file locations. It is not necessary to restart any Maconomy services for these changes to take effect.

To test that the error script is installed and working correctly, access the component in a browser and add an invalid script path to the end of the URL. Note: replace the <TEXT> in the string below in order to match to your existing installation.

Example:

cgi-bin/Maconomy/MaconomyWS.<TEXT>[.exe]**/invalid/path**

If the error script is installed the resulting error should now be one of the following:

Runtime error. See log file for details [Incident ID: <incident-ID>]

500 - Internal Server Error

Reference

https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2019-12314

Script(s):

```
var IdChars = "abcdefghijklmnopqrstuvwxyz1234567890";
function getRandomIdChar()
{
  return IdChars[random(0, sizeof(IdChars) - 1)];
}

function getIncidentId()
{
  var incidentId = "";
  for (var i = 0; i < 10; ++i)
  {
    incidentId += getRandomIdChar();
  }
  return incidentId;
}

//////////
// MAIN //
//////////

// Generate incident ID and write error to log file.
var incidentId = getIncidentId();
file::print(io::log(), sprint("Incident ID: ^1\n", incidentId));

// Discard output from failing script and write a generic error message.
discardoutput();
httpheaderset("Status", "500");
```

```
print("Runtime error. See log file for details [Incident ID: ^1]", incidentId);
```