June 8, 2015

Dear CA Customer:

The purpose of this Advisory is to inform you of a potential limitation that has been recently identified with CA Advanced Authentication. Please read the information provided below and follow the instructions in order to avoid being impacted by this problem.

PRODUCT(S) AFFECTED: CA Advanced Authentication RELEASE: All

PROBLEM DESCRIPTION:

# Browsers Restricting to Run Plugins.   The browser industry is leaning towards providing sandboxed environments, and any method which violates the sandbox is discouraged. Supporting this strategy is the announcement of Google Chrome which dropped their support for Plugins in version 42.

The following are the changes that are being implemented on all the browsers.

**Internet Explorer**

Internet Explorer does not support plugins and Applets in metro mode. In Desktop mode Plugins and Applets are supported.

**Chrome**

Chrome has dropped support for plugins and Applets starting in version 42.

**Firefox**

Firefox supports plugins from version 34.0.5 and above, but has plans to drop the plugins in the future.

CA Advanced Authentication clients use the Browser plugins to store the credentials into the file storage, which offers better persistent storage and device locking capabilities. Because of these browser changes the CA Advanced Authentication clients (both AuthID and DDNA clients) that are browser based will be affected.

Different types of storage, available in CA Advanced Authentication Clients in increasing order of persistence, are given below:

1. Cookies (Keys will be available in all browsers)
2. userData (Applicable only in Internet Explorer)
3. LocalStorage (Keys will be Available in all browsers)
4. Disk/File (Available only when plugin is enabled)

# **Browser Storage Support Matrix**

Here the fallback is from higher persistence to lower and the numbering, corresponding to persistence in the below table, is given in the reverse order, i.e. 1 means higher persistence storage and 2 means lower persistence storage.

For example, in Internet Explorer, which supports plugin, if the plugin is enabled, Hard Disk is taken as the persistence. If the Plugin is disabled then it falls back to User Data and if user data is disabled then it falls back to cookie.

Note: Hard disk is only possible when Plugin is supported and enabled.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Browser | Hard Disk  (implies plugin enabled) | Local Storage | User Data | Cookie |
| IE | 1 | Not Supported | 2 | 3 |
| Chrome < 42 | 1 | 2 | Not Supported | 3 |
| Chrome >=42 | Not Supported | 2 | Not Supported | 2 |
| Firefox > 34.0.5 | 1 | 2 | Not Supported | 3 |
|  |  |  |  |  |

# **Impact to clients when Browser Plugin support is taken away**

When an end user upgrades to a browser that no longer supports the plugin the following is the behavior depicted by clients.

**Desktop Mobile OTP client**

Desktop Mobile OTP client 2.3 and below has a dependency on the Authminder Plugin, and it always stores the account information in file storage. Starting in 2.3.1, the dependency is removed and the persistence falls back as mentioned in the table above.

Problem 1:

If a desktop mobile OTP client with a version less than 2.3.1 is used, and the browser is upgraded to a version which drops support for plugin, then the Desktop Mobile OTP client will not work, and will display an error saying “ Plugin Not Available - The Desktop Client requires the CA Technologies AuthMinder plugin to be enabled.”

Problem 2:

If a desktop mobile OTP client using a version 2.3.1 or above is used and the storage is in file storage, i.e the browser was using Authminder plugin and the browser upgrades to a version which drops support for plugins, the client will no longer have access to the keys.dat files. Therefore all the accounts provisioned onto the desktop mobile OTP client will not be seen and the OTP cannot be generated.

Solution:

Use any browser which supports Plugins.

For example, if a user has upgraded a Chrome browser to version 42, they can choose to use Internet Explorer with plugin enabled to continue accessing the accounts and generating the OTPs.

To use any browser that does not support Plugins.

In the case where a browser switch is not possible, install/upgrade the Desktop mobile OTP client, version 2.3.1 or higher, and perform a re-provision through AFM or any other Middleware which is administrating the Credential lifecycle for the users. This will re-provision the accounts onto the browser sandboxes (localStorage or userData) depending on the browser type, and is accessible only to that particular browser

Note: When the Persistent storage is not file storage, then the user should continue to use the same browser and should not switch browsers to provision accounts or generate OTPs.

**ANC (Arcot Native Client)**

ANC works only with Internet Explorer Browser on ActiveX technology and will continue to work in the same manner until Internet Explorer removes ActiveX support.

**CA AuthID Applet Client**

Since Applet depends on JRE plugins, when plugin support is taken away, Advanced Authentication Applet client will no more work. Starting with Chrome version 42, JRE plugins are not present in the browser. CA recommends that its customers use the CA AuthID JavaScript Client solution when the Applet is deprecated.

**CA AuthID Javascript Client**

If the plugin was enabled on the browser the CA AuthID Javascript Client AuthID gets downloaded into keys.dat which is stored into file storage, and all browsers can access AID and the user can authenticate from any browser. Hence the plugin enabled browser based JavaScript Client supports cross browser authentication and the user does not need to roam their credential to each browser. Now, with the new browsers, (eg. Chrome version 42) as plugin support is removed, the JavaScript client will no longer be able to access keys.dat file and hence the user has to roam their credentials (by Roaming Flow) and download AID again. As the file storage persistence isn’t available, the downloaded AID will get stored into browser storage (localStorage, userData or cookies etc.). Also, cross browser authentication will not

work and the user will have to roam their credentials onto each browser to authenticate themselves.

**RM DDNA Javascript Client**

If the plugin was enabled on the browser, the Device ID gets downloaded into keys.dat which is stored into file storage, and all browsers can access the Device ID and it can be sent to RM server for Risk Evaluation. Now, the new browsers (ex. Chrome version 42) with the plugin support removed, the DDNA client will no longer be able to access keys.dat file and hence the Device ID cannot be retrieved from keys.dat file to send to RM. Depending on the configuration in RM Server there are chances to take the user for the Secondary Authentication flow and on successful authentication the Device ID will now get stored into browser Storage (localStorage, userData or cookies). When the browser storage is used and if the user switches to a different browser, he will not be able to access the keys.dat as it is stored in the previous browser’s storage, hence he will be taken to a roaming flow and secondary authentication will be done again for the first time and the DeviceID will also be stored in the current browser.

# **Impact in Device Locking feature when Plugin support is taken away in the browser**

**Device Locking – Technical Overview**

The Device Locking feature enables an ArcotID to be locked to a specific machine, so that the ArcotID is not usable if it is copied to another machine.

The feature works by camouflaging (protecting) an ArcotID using two values.

1. The User PIN selected by the user when the ArcotID is issued. Note, in some implementations this value is hardcoded, which effectively makes the ArcotID a device credential rather than a user credential.
2. A Machine PIN, which is derived from unique machine-specific information, obtained from the hardware characteristics of the client machine.

In the implementation, the ArcotID is cryptographically camouflaged with two repeated applications of the algorithm, once with the User PIN and once with the Machine PIN. We do this as two steps to facilitate provisioning of the credential to multiple devices (roaming).

When an end user upgrades to a browser which no longer supports the plugin or when the user disables the plugin, the following is the behavior depicted by the respective clients with respective to device locking.

**PKI Javascript Client**

When user tries to download AID on a browser which doesn’t support plugin then AID will get downloaded and get stored onto either of following:

1. Local Storage
2. User Data (Internet Explorer only)
3. Cookies

Because the plugin is not available, the machine PIN (which is derived from machine hardware) cannot be retrieved and AID will not get device locked. If the user tries to copy AID from one machine to another (in browser) and tries authenticating themselves, authentication will succeed (with valid password).

**Desktop Mobile OTP client**

When user tries to provision a OTP account on a browser which doesn’t support plugin then the OTP Account will get stored onto either of following:

1. Local Storage
2. User Data (Internet Explorer only)
3. Cookies

Because the plugin is not available, the machine PIN (which is derived from machine hardware) cannot be retrieved and the OTP Account will not get device locked. If the user tries to copy the OTP Account from one machine to another (in browser) and accesses the Desktop Client account, the account will be available and with a valid PIN the user can generate a valid OTP and get authenticated successfully.

**PKI Applet Client**

Since the Applet depends on JRE plugins, when plugin support is taken away, the Advanced Authentication Applet client will no longer work. . Starting with Chrome version 42, JRE plugins are not present in the browser. CA recommends to its customers that they use the CA AuthID JavaScript Client solution when the Applet is deprecated.

# **Impact to PKI Applet Client in Safari Browser (version 7 and above)**

**PKI Applet Client**

Since Safari 7, java applets cannot access local file systems because of security reasons. This prevents downloading the ArcotId to the client's hard disk.

Please refer the following links for further information:

a)<https://support.apple.com/en-us/HT202819>

b)<http://apple.stackexchange.com/questions/108709/running-java-in-unsafe-mode>

This is a technical limitation in Safari. This can be solved by the following steps:

1. While visiting the website for the first time, user may see this message: "Do you want to trust the website name to use the Internet plug-in name plug-in?” Click "Trust" to allow the website to use the Java plug-in and display its content.
2. Choose Safari > Preferences…
3. Click the Security tab.
4. Click Manage Website Settings to see plug-in details for a particular website.
5. Java plug-ins installed on your computer appear on the left side of the plug-ins sheet. Select to configure its website settings.
6. Websites that are currently open in Safari appear on the right side of the plug-ins sheet. Websites that users have already configured by clicking "Trust" or "Cancel" also appear here.
7. Beside the website name, expand the dropdown and click on "Run in unsafe mode" >Click "Trust”. Make sure the dropdowns’ present value is "Always Allow”. This will allow the applet to access the local file system.

If you have any questions about this Advisory, please contact CA Support.  
   
Thank you,

CA Support Team