Amazon Fire Phone – Silent Certificate Install

25/06/2015

Software:	Amazon Fire Phone
Affected Versions:	Fire OS < 4.6.1
CVE Reference:	-
Author:	Bernard Wagner
Severity:	Medium
Vendor:	Amazon
Vendor Response:	Patched in Fire OS 4.6.1

Description

The CertInstaller package on the Amazon Fire Phone allows applications to install certificates without interaction with the user. Although the application's name is identical to the base Android package, the source code has been modified specifically for the Amazon Fire Phone. Successful exploitation of the vulnerability would allow an attacker to Man-in-the-Middle (MiTM) encrypted traffic. Although no user interaction is required, a notification is sent when a certificate has been installed.

Impact

If the vulnerability was to be successfully exploited, all encrypted traffic that does not make use of certificate pinning could be intercepted with a Man-in-The-Middle attack.

Cause

The package checks for an extra that, if set, results in the silent installation of a certificate.

Interim Workaround

Users are advised to only install applications from trusted sources and exclusively make use of trusted networks. Users that notice any notifications regarding "Certificate Installed" should immediately remove the certificate and uninstall any possibly malicious applications that were recently added.

Solution

Users should update to the latest version of Fire OS, as the issue has been addressed in Fire OS 4.6.1.

Technical Details

The **onCreate** method of the **CertInstaller** activity is given below:

```
protected void onCreate(Bundle savedStates) {
        super.onCreate(savedStates);
        this.mCredentials = createCredentialHelper(getIntent());
        if (this.mCredentials.installSilently() || UserHandle.myUserId() == 0) {
            this.mState = savedStates == null ? 1 : 2;
            if (this.mState != 1) {
                this.mCredentials.onRestoreStates(savedStates);
                this.mNextAction = (MyAction)
savedStates.getSerializable(NEXT_ACTION_KEY);
            } else if (!this.mCredentials.containsAnyRawData()) {
                showErrorDialogAndFinish(R.id.activity_text);
            } else if (!this.mCredentials.hasPkcs12KeyStore()) {
                MyAction action = new InstallOthersAction();
                if (needsKeyStoreAccess()) {
                    sendUnlockKeyStoreIntent();
                    this.mNextAction = action;
                } else {
                    action.run(this);
                }
            } else if (this.mCredentials.hasScepPassword()) {
                this.mNextAction = new
Pkcs12ExtractAction(this.mCredentials.getScepPassword());
                this.mNextAction.run(this);
            } else if (this.mCredentials.installSilently()) {
                Log.e(TAG,   "Installing   silently?   Pkcs   password   missing!
Aborting...");
                finish();
            } else {
                showDialog(STATE_RUNNING);
            }
        } else {
            showErrorDialogAndFinish(R.string.only_primary_user_allowed);
        }
    }
```

The following line is where the vulnerability exists:

```
        if (this.mCredentials.installSilently() || UserHandle.myUserId() == 0) {
```

The logic trace of `installSilently()` method is given below:

```
    CredentialHelper(Intent intent) {
        this.mBundle = new HashMap();
        this.mName = "";
        this.mCaCerts = new ArrayList();
        Bundle bundle = intent.getExtras();
        if (bundle != null) {
            String name = bundle.getString("name");
            bundle.remove("name");
            if (name != null) {
                this.mName = name;
            }
            String scepPassword = bundle.getString(EXTRA_SCEP_CLIENT_PASSWORD);
            bundle.remove(EXTRA_SCEP_CLIENT_PASSWORD);
            if (scepPassword != null) {
                this.mScepPassword = scepPassword;
            }
            this.mInstallSilently = bundle.getBoolean("q_install");
            bundle.remove("q_install");
            Log.d(TAG, "# extras: " + bundle.size());
            Iterator i$ = bundle.keySet().iterator();
            while (i$.hasNext()) {
                String key = (String) i$.next();
                byte[] bytes = bundle.getByteArray(key);
                Log.d(TAG, "      " + key + ": " + (bytes == null ? -1 :
bytes.length));
                this.mBundle.put(key, bytes);
            }
            parseCert(getData("CERT"));
        }
    }

    boolean installSilently() {
        return this.mInstallSilently;
    }
```

As can be seen from the above source code, if an Intent containing the extra **q_install (**with its value set to true) starts the activity, the certificate will be installed silently.

Detailed Timeline

Date	Summary
19/01/2015	Reported to Amazon
04/02/2015	Amazon confirms reception and validity
09/03/2015	MWR requests status and progress
13/03/2015	Amazon notifies MWR that implementation of fixes has commenced
27/03/2015	Amazon notifies MWR that testing of fixes have commenced
10/04/2015	Amazon notifies MWR that testing is almost complete
01/05/2015	FireOS 4.6.1 released
03/05/2015	Amazon and MWR coordinate public release of advisory
25/06/2015	Advisory published