CVE-2016-1519: GrandStream Android VoIP App TLS MitM Vulnerability

Dr. Georg Lukas, rt-solutions.de GmbH, 2016-03-16

Affected app: Grandstream Wave version 1.0.1.26 (and probably earlier)

Classification: CWE-295 Improper Certificate Validation

Summary

The Grandstream VoIP products deploy a remote provisioning mechanism that allows to automatically set configuration elements on app startup. By default, an insecure connection to `fm.grandstream.com` is used to obtain the provisioning profile (CVE-2016-1518). However, even if an HTTPS URL is configured, the certificate is not validated, allowing an active attacker to successfully impersonate the provisioning server with an invalid, mismatching or outdated certificate. Based on that, the attacker can elevate CVE-2016-1518 to redirect phone calls through a malicious server, or turn the phone into a bug. The changes are stored locally, so a single successful attack is sufficient to gain permanent control of the app.

Details

When accessing HTTP/HTTPS URLs, the Grandstream Wave app is using a custom HTTP connection manager in the `com.softphone.common` package. That class is deploying both a custom TrustManager and a custom HostnameVerifier (the depicted variable and class names have been changed as part of the deobfuscation):

```
private static TrustManager[] tm = new TrustManager[]{ new TM(null) };
private static HostnameVerifier hv = new HV();

...

SSLContext sc = SSLContext.getInstance("SSL");
sc.init(null, tm, new SecureRandom());
HttpsURLConnection.setDefaultSSLSocketFactory(sc.getSocketFactory());
HttpsURLConnection.setDefaultHostnameVerifier(hv);
```

The custom TrustManager skips all verification, essentially accepting any certificate for any connection:

```
class TM implements X509TrustManager {
    public void checkClientTrusted(X509Certificate[] c, String at) {}
    public void checkServerTrusted(X509Certificate[] c, String at) {}
    public X509Certificate[] getAcceptedIssuers() {
        return new X509Certificate[0];
    }
}
```

The custom HostnameVerifier does not verify if the server certificate corresponds to the hostname the connection was made to, allowing to present the client with a certificate issued for a different hostname:

```
class HV implements HostnameVerifier {
    public boolean verify(String hostname, SSLSession session) {
        /* skipped log output */
        return true;
    }
}
```

Each of these two issues individually allow an active MitM attacker (e.g. in a public WiFi network) to hijack the connection, either by presenting a self-signed certificate or by presenting a valid certificate issued to an attacker-controlled domain.

Impact

Even if the app is configured to use HTTPS for the provisioning server, an active attacker can trick it into obtaining a provisioning file from an attacker-controlled sserver, and thus to leverage CVE-2016-1518:

- Adding or replacing VoIP accounts / servers
- Turning the phone into a bug by setting a "silent" ringtone and enabling auto-answer
- Changing the Config Server Path URL to a server controlled by the attacker, allowing to push new configuration after the initial Man-in-the-Middle attack

Mitigation

On the Wave app, the only way to close the issue is by disabling remote provisioning. This can be achieved by entering an empty URL:

1. Open "GSWave"
2. Switch to the "Settings" tab
3. Open "Advanced Settings" menu
4. Set "Config Server Path" to an empty string

After disabling / securing the auto-provisioning, **all configuration elements** should be checked for prior manipulation, e.g. by exporting and inspecting the configuration file.

Timeline

- 2015-11-24 Discovery of the issue
- 2015-11-25 Requested CVE number
- 2015-12-01 Notification of vendor
- 2016-01-20 CVE number assigned
- 2016-03-16 Public disclosure

Contact

Please contact Dr. Georg Lukas with any further questions regarding this vulnerability.