Ronald Eikenberg, Jörg Wirtgen

Surf-Versicherung für Android

Jelly Bean und älter trotz Schwachstellen sicher nutzen

Wer eine ältere Android-Version als 4.4 nutzt, muss mit Sicherheitslücken im Browser leben, die Google nicht schließen will – und selbst die existierenden Sicherheits-Patches kommen nicht auf allen Geräten an. Bevor die Lücken von Cyber-Ganoven ausgenutzt werden, sollten Sie daher auf Ihrem Smartphone und Tablet ein paar Vorsichtsmaßnahmen ergreifen.

Auf fast der Hälfte der aktuell genutzten Android-Geräte läuft noch Jelly Bean (Android 4.1 bis 4.3), auf über 10 Prozent eine noch ältere Version – und in vielen davon klaffen Sicherheitslöcher. Zwar hat Google bislang stets passende Patches entwickelt, doch die landen nur bei den Herstellern, die es dann oft versäumen, sie den Nutzern in Firmware-Updates zur Verfügung zu stellen. Ob und welche Lücken im eigenen Gerät klaffen, kann man nur mit großem Aufwand herausfinden.

Künftig spitzt sich die Lage für Nutzer alter Android-Versionen weiter zu: Google ließ durchblicken, künftig keine Sicherheits-Patches für den bis Android 4.3 genutzten Browser mehr zu entwickeln. Damit stirbt die letzte Hoffnung, dass neu entdeckte Lücken jemals geschlossen werden. Googles Entscheidung betrifft nicht nur den Browser, sondern das gesamte System: Seine sogenannte WebView-Komponente kommt in unzähligen Apps zum Einsatz, die dadurch ebenfalls angreifbar sind. Browser und WebView kann man bis Jelly Bean nur per Firmware-Update auf den aktuellen Stand bringen.

Mit dem Android-Check auf heise security finden Sie heraus, wie sicher Ihr Android-Gerät beim Surfen ist (siehe c't-Link).

Smartphone als Wanze

Würde es ein Angreifer darauf anlegen, ein Android-Gerät zu kompromittieren, hätte er reichlich Möglichkeiten. So bringt etwa das frei verfügbare Pentesting-Tool Metasploit inzwischen elf Module mit, die verschiedene Android-Lücken ausnutzen. Je älter die Android-Version, desto größer die Auswahl. Mit Metasploit haben wir mit überschaubarem Aufwand eine Webseite gebaut, die ein ansurfendes Jelly-Bean-Smartphone in eine Abhörwanze verwandelt – ohne dass der Nutzer etwas davon mitbekommt. Nach der Infektion konnten wir über das Netz unter anderem Kamera und Mikrofon anzapfen, die GPS-Koordinaten abrufen und auf Dateien zugreifen. Schuld ist eine Lücke in einer Browser-Schnittstelle, durch die Webseiten beliebige Java-Befehle auf dem Android-Gerät ausführen können.

Durch eine Lücke jüngeren Datums in Android bis 4.3 kann ein Angreifer eine essenzielle Schutzfunktion des Browsers austricksen, die sogenannte Same-Origin-Policy. Sie bewirkt, dass eine Website nicht auf Inhalte einer anderen Website zugreifen darf. Lädt etwa eine böswillige Seite in einem IFrame das Webmail-Interface von web.de, bekommt sie normalerweise keinen Zugriff auf den Inhalt des IFrames mit dem Mail-Posteingang. Bei den betroffenen Android-Versionen kann ein Angreifer jedoch die Same-Origin-Policy mit wenigen Zeilen JavaScript-Code umgehen und so auf persönliche Daten seines Opfers zugreifen. Auch das konnten wir nachvollziehen. Diese Angriffsform bezeichnet man als Universal-Cross-Site-Scripting (UXSS).

Der Sicherheitsexperte Tod Beardsley von Rapid7 hat den Angriff kürzlich auf die Spitze getrieben, indem er UXSS mit einer Lücke in Googles Play Store kombinierte. Seine Demoseite steuerte die Web-Ausgabe von Google Play fern. Sie lud zuerst die Produktseite einer beliebigen App und klickte anschließend auf den Kaufen-Button. Kurze Zeit später wurde die App auf dem Gerät des Webseitenbesuchers installiert und gestartet – vollautomatisch und ganz ohne Nutzerinteraktion. Google hat die verwundbaren Browser-Versionen kurz darauf aus der Web-Version des Stores ausgesperrt. Damit wird zwar die automatische App-Installation verhindert, die Wurzel des Übels – die UXSS-Lücke – existiert jedoch weiterhin.

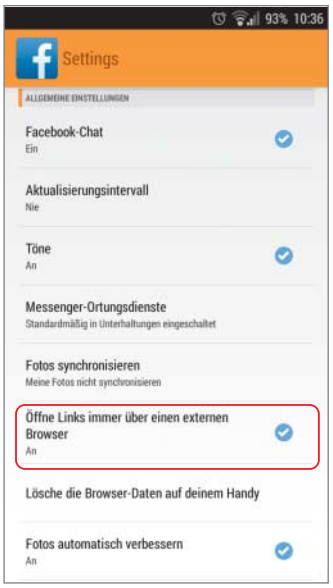
Immerhin war es nur möglich, Apps aus dem Play Store zu laden – die weitaus gefährlicheren Apps aus fremden Quellen lassen sich weiterhin nicht ohne Bestätigung des Nutzers installieren.

So schützen Sie sich

Die einzige wasserdichte Lösung ist der Umstieg auf Android 4.4 oder 5. Dort kommt eine WebView-Komponente auf Chrome-Basis zum Einsatz, die weniger Fehler aufweist und Updates automatisch über den Play Store bekommt – unabhängig vom Betriebssystem. Wenn der Gerätehersteller eine Aktualisierung auf Android 4.4 anbietet, sollten Sie die also aufspielen und sind fertig; weiter brauchen Sie nicht zu lesen. Sie fangen sich damit allerdings Einschränkungen beim Zugriff auf die SD-Karte ein [1].

Lässt Sie der Hersteller bei 4.3 oder älter hängen, können Sie überprüfen, ob er zumindest für den oben beschrieben UXSS-Bug einen Patch in die Firmware integriert hat. Wir haben dazu eine harmlose Testseite entwickelt (siehe c't-Link am Ende des Artikels), welche die UXSS-Lücke tatsächlich ausnutzt – gelingt das, ist die Wahrscheinlichkeit groß, dass der Hersteller auch andere Patches vernachlässigt hat. Sie sollten also vom Schlimmsten ausgehen.

Gibt es kein offizielles Update auf 4.4 oder neuer, kann ein CustomROM mit Android 4.4 eine Lösung sein; also ein selbst aufgespieltes Alternativ-Android. Für viele Geräte gibt es CustomROMs, eine gute Anlaufstelle ist CyanogenMod. Das ist aber nichts, was man mal eben schnell einspielt, sondern eine grundlegende Entscheidung. Selbst wenn alles glattgeht, müssen Sie Ihr Gerät dazu rooten sowie alle Apps und Einstellungen neu installieren. Sie verlieren meist auch die Hersteller-Garantie und müssen auf jene Apps des Herstellers verzichten,



Apps wie Facebook oder RSS-Reader nutzen einen eigenen, wahrscheinlich ebenfalls verwundbaren Browser. Einige lassen sich so einstellen, dass sie stattdessen den externen Browser aufrufen.

Maßnahmen bei Android bis 4.3

- Installieren Sie Firefox, Chrome oder Dolphin Browser.
- Stellen Sie browsende Apps wie Facebook so ein, dass sie den externen Browser nutzen. Alternativ: Nutzen Sie solche Dienste im Browser statt per App.
- Verzichten Sie auf Apps mit Werbebannern.
- Surfen Sie in öffentlichen WLANs per VPN oder bleiben Sie stattdessen im Mobilfunknetz.

die er nicht auch im Play Store anbietet. Beides ist bei älteren Geräten kein großer Verlust. [2]

Browsen ohne WebView

Wenn Sie kein 4.4 bekommen, müssen Sie mit den Lücken leben. Da es bisher noch keine echten Angriffe gibt, sondern nur Proof-of-Concepts, bleibt das ein geringes Risiko. Einige Vorsichtsmaßnahmen sind aber angebracht. Installieren Sie zuerst einen Browser mit eigenem Renderer, zum Beispiel Chrome, Firefox oder Dolphin Browser. Damit sind Sie nicht mehr angreifbar, solange Sie diesen Browser benutzen.

Viele beliebte Browser-Apps wie CM Browser, Maxthon oder Mercury haben keine eigene Browser-Engine, sondern rufen die WebView-Komponente auf. Sie bieten auf verwundbaren Geräten daher keine Abhilfe. Überprüfen Sie im Zweifelsfall über unsere Testseite, welche Engine zum Einsatz kommt. Die Seite zeigt einen Text wie „AppleWebKit/534.40“ an. Rufen Sie unseren Test zuerst mit dem Standard-Browser und dann der Alternative auf, um festzustellen, ob Letztere eine andere oder zumindest neuere Engine einsetzt.

Tippen Sie in einer Mail auf einen Link, wählen Sie Chrome und bestätigen Sie dieses Fenster mit „Immer“. Dann haben Sie den unverwundbaren Browser als Standard eingestellt.

Nach der Installation des Browsers sollten Sie ihn aus einer anderen App aufrufen, am einfachsten durch Antippen eines Links in einer Mail. Dann fragt Android nach, welcher Browser standardmäßig aufgerufen wird – wählen Sie dort den neuen aus und bestätigen Sie „immer“. Fehlt die Frage, hatten Sie vielleicht schon mal einen anderen installiert. Das Zurücksetzen so einer Default-Zuordnung löst Android recht umständlich: Suchen Sie in Einstellungen/Apps den startenden, als Standard eingestellten Browser (zu finden im Reiter „Alle“), tippen Sie darauf und dann auf den Knopf „Standardeinstellung zurücksetzen“ unter „Standardmäßig starten“. Danach kommt die Auswahlbox beim nächsten Antippen eines Links wieder.

Einige Apps bieten einen internen Browser zum Anzeigen von Websites und nutzen dazu vermutlich den kaputten WebView. Dazu gehören beispielsweise die Facebook-App und einige News- und RSS-Reader. Um darüber nicht angreifbar zu sein: Versuchen Sie, die App so einzustellen, dass sie einen externen Browser aufruft. Bei Facebook geht das über die „App-Einstellungen“ in dem unübersichtlich langen Menü, das sich beim Tippen des grauen Menüknopfs öffnet.

Hat die App keine solche Einstellung, sollten Sie mit ihr keine externen Links mehr aufrufen. Vielleicht können Sie sogar ganz auf die App verzichten und den entsprechenden Dienst nur noch per Mobil-Browser nutzen.

WebView woanders

Die meisten Apps, die Werbebanner einblenden, dürften dafür ebenfalls den WebView nutzen. Angriffe auf die Server, die solche Banner verbreiten, finden tatsächlich statt, bisher zielen sie auf Windows-Lücken. Auf diese Weise wurden auch schon renommierte Seiten mit böswilligen Werbebannern unterminiert. Es reicht also nicht, dem Seitenanbieter oder – im Fall von Android – dem App-Anbieter zu trauen, denn der Angriff findet möglicherweise außerhalb seines Einflussbereichs statt. Die einfachste Lösung ist, auf werbefinanzierte Apps zu verzichten oder die Bezahlversion zu kaufen.

Der WebView kommt auch zu vielen anderen Gelegenheiten in Apps zum Einsatz. Beispielsweise zeigen einige Apps ihre Versionshistorie oder Nutzungsbedingungen an, indem sie eine Website vom Anbieter aufrufen; andere laden als Hauptzweck Webinhalte nach. Auf den ersten Blick ist das ungefährlich, doch zwei Szenarien sind denkbar: Erstens könnte der Server des Anbieters gehackt werden und Schadcode ausliefern. Zweitens können Angreifer den Datenverkehr in öffentlichen WLANs manipulieren.

Gegen das Mithorchen hilft eine Maßnahme, die bei Benutzung kritischer Apps in öffentlichen WLANs sowieso eine Überlegung wert ist: Gehen Sie nur per VPN online. Richten Sie sich also entweder zu Hause einen VPN-Zugang ein (nicht ganz trivial), fragen Sie die IT-Abteilung Ihres Arbeitgebers oder schauen Sie sich bei kommerziellen VPN-Anbietern um [3]. Eine weitere Lösung wäre, die potenziell kritischen WLANs zu vermeiden und per Mobilfunk ins Internet zu gehen. Gegen gehackte Anbieterserver hilft beides nicht.

Rooten hilft wenig

Reparieren lässt sich die WebView-Lücke nach unserem Wissen nicht. Selbst auf gerooteten Geräten ist uns keine Lösung bekannt, WebView zu fixen oder notfalls ganz lahmzulegen.

Nach dem Rooten kann man jedoch Adblocker wie das Xposed-Modul MinMinGuard oder AdAway installieren, die vor korrumpierten Werbebannern schützen können. Das Xposed-Modul Xprivacy widmet sich zwar unter „Anzeigen (mittels Browser)“ dem WebView, verhindert aber hauptsächlich die Übertragung privater Daten wie dem UserAgent-String. Das Abschalten oder Absichern von WebView gelingt damit nicht. Somit erzeugen diese Tools oder auch Xposed-Firewalls ein eher trügerisches Gefühl der Sicherheit, ohne das Problem wirklich zu lösen. (jow@ct.de)

Wenn das Auswahlfenster nicht erscheint, ist schon ein Browser als Standard ausgewählt – möglicherweise der verwundbare. Das können Sie in den App-Infos überprüfen und zurückstellen.

Literatur

[1] Jörg Wirtgen, Beschreiben verboten, Einschränkungen beim Zugriff auf SD-Karten unter Android 4.4 und 5.0, c’t 3/15, S. 150

[2] Hannes Czerulla, Freiheit für Android, Rooting für Android-Smartphones und -Tablets, c’t 2/15, S. 82

[3] Urs Mansmann, Sonne, Strand und Internet, Im Urlaub sicher und günstig online gehen, c’t 15/14, S. 76

c't Testen Sie Ihren Browser: ct.de/yhue