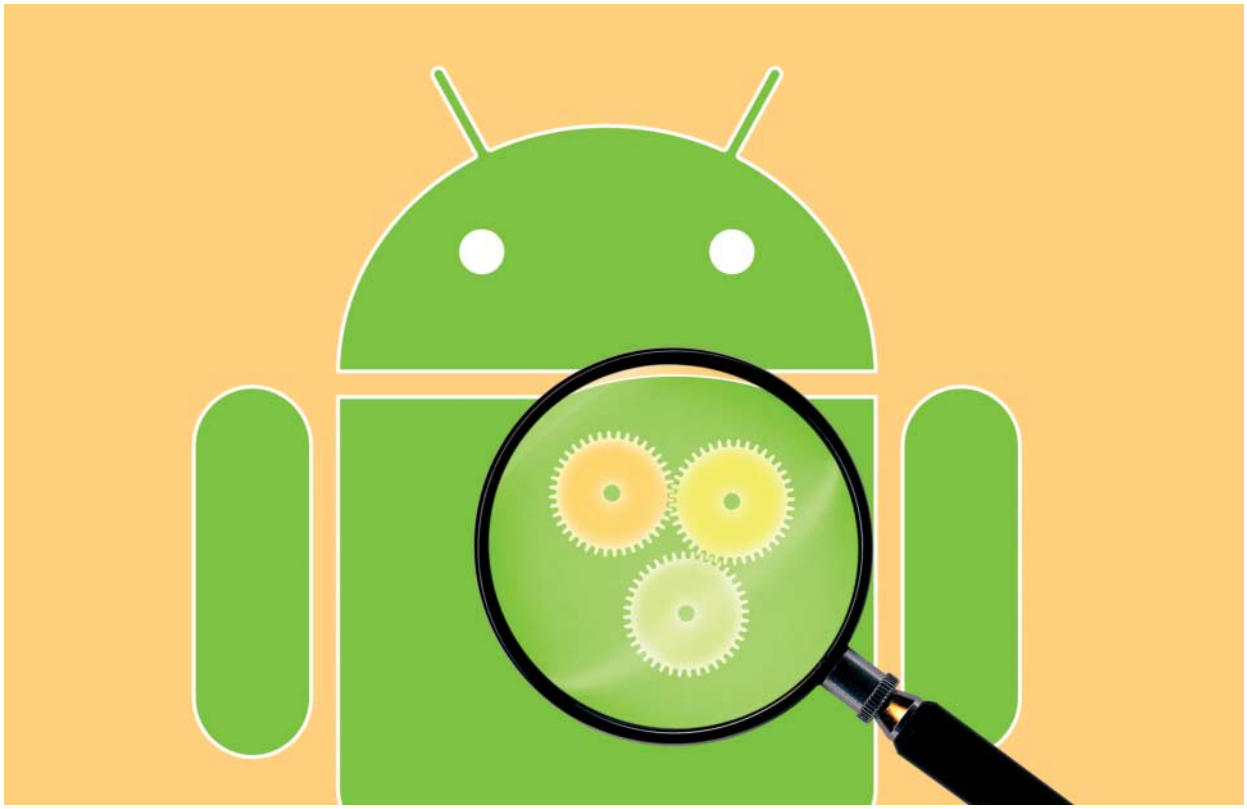


Android-Traffic unter der Lupe

Datenverkehr von Apps und System direkt mit dem Smartphone analysieren

Android-Apps fordern gerne Zugriffsrechte auf Kamera, Kontakte et cetera ein und greifen selbstverständlich aufs Internet zu – eine brisante Mischung. Was eine App daraus macht, erfährt man meist nicht – zieht sie gerade Neuigkeiten aus dem Netz oder schickt sie Ihr Adressbuch und Ihre Urlaubsfotos nach Hause? Mit der Analyse-App Packet Capture finden Sie es heraus.

Von Ronald Eikenberg

Mit dem Analyse-Tool Packet Capture kann man leicht überprüfen, welche Daten das Android-System und die installierten Apps mit dem Netz austauschen. Das Tool läuft direkt auf dem Smartphone oder Tablet und erfordert keine Root-Rechte. Es verschafft sogar Einblick in verschlüsselte SSL-Verbindungen. Nebenbei findet man heraus, ob eine App überhaupt verschlüsselt kommuniziert oder vertrauliche Daten wie Passwörter im Klartext durchs Netz schickt. Dies wäre insbesondere in öffentlichen Netzen wie Hotspots ein Sicherheitsrisiko, da jeder diese Daten mitlesen und manipulieren kann.

Damit die Verkehrsanalyse leicht von der Hand geht, nutzt Packet Capture zwei Tricks: Zum einen arbeitet es als lokaler VPN-Server, durch den sämtlicher Datenverkehr des Android-Geräts geschleust wird. Dort fließt nicht nur WLAN-Traffic vorbei, sondern auch Daten, die über das Mobilfunknetz übertragen werden. Zum anderen klinkt sich das Tool als Man-in-the-Middle (MitM) in SSL-Verbindungen ein, damit man sie im Klartext mitlesen kann. Normalerweise entsteht ein verschlüsselter SSL-Tunnel zwischen App und Ziel-Server; den Inhalt kann man nicht ohne Weiteres mitlesen. Während

einer Analyse gibt es Tunnel: einen zwischen der überwachten App und Packet Capture und einen zwischen Packet Capture und dem Ziel-Server. Dazwischen fließt der Datenverkehr im Klartext.

Packet Capture erstellt dynamisch passende Zertifikate für alle SSL-Verbindungen – etwa ein facebook.com-Zertifikat bei Nutzung der Facebook-App. Damit die Verbindung zwischen der überwachten App und dem MitM zustande kommen kann, muss die App dem CA-Zertifikat des MitM vertrauen. Das erreicht man, indem man das Packet-Capture-Zertifikat in den Speicher vertrauenswürdiger CA-Zertifikate des Systems importiert. Das ist nicht ganz ungefährlich: Das Analyse-Tool kann so sämtliche verschlüsselten Verbindungen und die darüber übertragenen Daten mitlesen. Wer auf Nummer sicher gehen möchte, setzt Tools wie Packet Capture nur auf eigens dafür abgestellten Testgeräten ein. Das Tool generiert das CA-Zertifikat individuell bei der Einrichtung. Würden alle Nutzer ein identisches Zertifikat nutzen, wäre dies ein erhebliches Sicherheitsproblem.

Analyse-App einrichten

Installieren Sie zunächst Packet Capture des Entwicklers Grey Shirts aus Google Play auf Ihrem Android-Gerät. Beim ersten Start begrüßt Sie der Assistent und bietet Ihnen im Schritt „SSL Decryption“ die Installation des SSL-Zertifikats an. Dies können Sie jederzeit nachholen – allerdings ist es ratsam, es gleich zu tun, da Ihnen ansonsten der Inhalt potenziell spannender SSL-Pakete entgeht. Drücken Sie auf „Install Certificate“ und geben Sie zur Autorisierung der Installation Ihre Android-PIN respektive Ihren Passcode ein. Haben Sie Ihr Gerät bisher weder durch PIN noch Passcode geschützt, müssen Sie es jetzt nachholen, um das Zertifikat installieren zu können. Den Dialog „Zertifikat benennen“ bestätigen Sie einfach mit OK.

Losgeschnüffelt

Packet Capture ist jetzt einsatzbereit. Klicken Sie auf den grünen Pfeil oben rechts, um die Aufzeichnung zu starten. Daraufhin erkundigt sich Android, ob Sie dem Aufbau der lokalen VPN-Verbindung zustimmen. Nachdem Sie mit OK bestätigt haben, beginnt Packet Capture mit der Aufzeichnung. In der Liste erscheint ein neuer Mitschnitt, der auf den Zeitpunkt des Aufnahmestarts datiert ist. In der zweiten Zeile des Eintrags steht die Anzahl der bisher aufgezeichneten Pakete. Klicken Sie auf die Sitzung, um die mitgeschnittenen Daten in Augenschein zu nehmen.

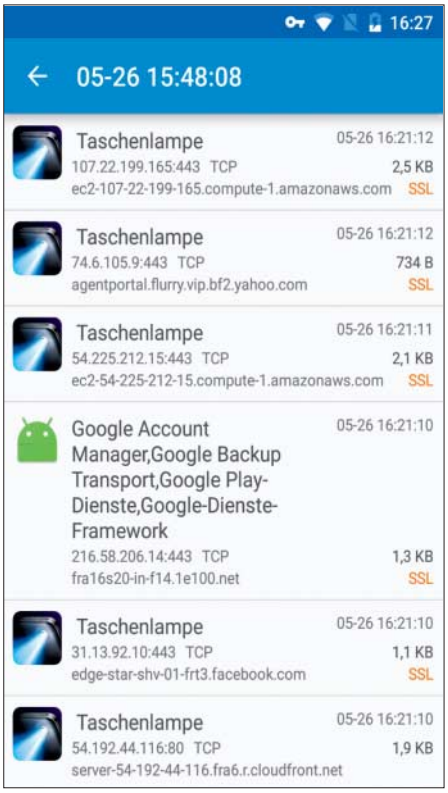
Es öffnet sich die Verbindungsliste, die auf einen Blick alle wichtigen Informationen zu den einzelnen Datenverbindungen anzeigt: Links das Icon der App, von der die Verbindung ausging, mittig der Name der App, darunter die IP-Adresse des Verbindungspartners und der Ziel-Port, das Protokoll (TCP oder UDP) und in der dritten Zeile schließlich der Hostname des Gegenübers. Rechts erscheinen der genaue Zeitpunkt, die Menge der übertragenen Daten und im Fall von verschlüsselten Verbindung die Kennzeichnung „SSL“.

Nach einem Klick auf eine Verbindung öffnet sich die Konversationsansicht, welche an die Wireshark-Funktion „Folge TCP-/UDP-Stream“ erinnert. Sie zeigt nicht einzelne Datenpakete, sondern gleich den ganzen Dialog mit dem Verbindungspartner: also sowohl die Anfrage als auch die Antwort darauf. Finden während einer Verbindung mehrere Anfragen statt, tauchen diese ebenfalls innerhalb dieser Konversation auf.

Daten dekodieren

Um den anfallenden Datenverkehr zu reduzieren, nutzen viele Systemkomponenten und Apps das Kompressionsverfahren gzip. Das erkennen Sie zum einen an dem HTTP-Header „Content-Encoding: gzip“ und zum anderen daran, dass unterhalb der Header lediglich eine scheinbar zufällige Kombination diverser Zeichen angezeigt wird. Packet Capture verwandelt die komprimierten Daten in Klartext, wenn Sie oben rechts auf die Lupe mit dem Schriftzug „HTTP“ drücken. Dadurch interpretiert die App die geöffnete Verbindung als HTTP.

Wer möchte, kann sich die Pakete auch als Hex-Dump anzeigen lassen – ent-



Viele Android-Apps stecken voller Überraschungen: Die erstbeste Taschenlampen-App etwa kommuniziert mit mehr Servern, als auf diesen Screenshot passen.

Das Update-Programm unseres Test-Smartphones hat versucht, eine APK-Installationsdatei über eine ungesicherte HTTP-Verbindung abzurufen. Das ist ein potenzielles Sicherheitsproblem.

weder die gesamte Verbindung oder einzelne Pakete. Die gesamte Verbindung lassen Sie sich in der Konversationsansicht über den Button mit den drei Punkten und HEX im Hexadezimalsystem darstellen, einzelne Pakete über die HTTP-Ansicht über den Knopf „TEXT“.

Exportieren

Sie beenden die Aufzeichnung über den roten Stop-Knopf im Hauptfenster von Packet Capture, das Sie über die Zurück-Taste Ihres Geräts erreichen. Wenn Sie die Aufzeichnung erneut starten, erstellt die Analyse-App einen neuen, separaten Mitschnitt. Die alten Aufzeichnungen bleiben dauerhaft erhalten, auch wenn Sie die App beenden.

Möchten Sie die Daten für eine tiefergehende Untersuchung teilen oder Ihre Erkenntnisse weitergeben, können Sie die Mitschnitte im Textformat exportieren. Konversationen exportieren Sie über das Menü der Konversationsansicht (der Button mit den drei Punkten) und „Save Upstream“, „Save Downstream“ oder „Save Both“. Analog dazu können Sie mit einzelnen Paketen in der HTTP-Ansicht verfahren.

SSL-Spezialitäten

Wenn Sie bereits zu den sieben Prozent gehören, die Android Nougat (7.0 oder 7.1) nutzen, dann müssen Sie unter Umständen mehr Aufwand betreiben, um den Inhalt verschlüsselter Verbindungen zu analysieren. Google hat bei Nougat viele Sicherheitsschrauben angezogen – und eine davon betrifft unmittelbar die Auswertung von SSL-Traffic mit Analyse-Proxies. Apps, die für Android 7 (SDK-Version 24) entwickelt wurden und auf ebendieser Version ausgeführt werden, ignorieren alle Zertifikate, die der Nutzer zur Liste vertrauenswürdiger CA-Zertifikate hinzugefügt hat – einschließlich der Zertifikate von Analyse-Tools. Durchgeleitete SSL-Verbindungen schlagen fehl und Packet Capture zeigt sie nur mit dem Zusatz „No data“ an.

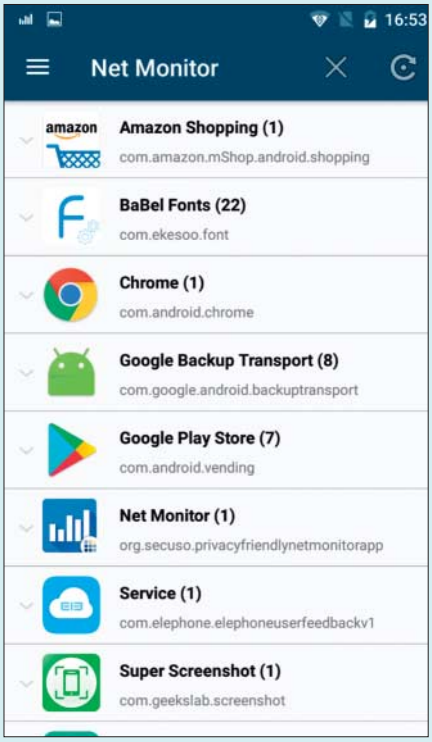
Zwar hat Google eine Hintertür offen gelassen, diese ist jedoch vor allem für Entwickler interessant: Apps vertrauen den nachinstallierten Zertifikaten weiterhin, wenn dies explizit in der APK-Installationsdatei (genauer gesagt in einer Datei namens network_security_config.xml) festgelegt wurde. Für den Entwickler der jeweiligen Apps ist diese Änderung ein Leichtes. Außenstehende müssen die APK-Datei zunächst auf Rechner übertragen, entpacken, modifizieren, wieder zusammenpacken und neu signieren. Anschließend wird die Datei zurück aufs Android-Gerät kopiert und neu installiert. Wer es probieren möchte, findet unter ct.de/y9fh eine ausführliche Anleitung und ein Bash-Skript, das die Änderungen automatisch durchführt. Der schnellste Weg zum Ziel ist in so einem Fall der Einsatz eines Android-Geräts, auf dem höchstens Android 6 läuft. Alternativ

Weitere nützliche Helferlein

Neben Packet Capture gibt es noch einige weitere Analyse-Tools, die auch ohne Root-Rechte einen wertvollen Einblick in den Datenverkehr gewähren. Einen ersten Überblick über die derzeit aktiven Netzwerkverbindungen liefert der quelloffene **Net Monitor** von der Secuso Research Group. Die App präsentiert die von dem Unix-Tool **netstat** bekannten Metainformationen auf einer ansprechenden Bedienoberfläche. Sie erfahren, welche Apps mit welchen Servern sprechen, können allerdings nicht in die Verbindungen hineinschauen. Net Monitor bietet gegenüber netstat einige Extras: So fragt die App etwa bei verschlüsselten Verbindungen den Analysedienst SSL Labs, wie sicher die Krypto-Konfiguration des Verbindungspartners ist.

Wer schon mal Datenverkehr auf einem Unix-System mitgeschnitten hat, der kennt **tcpdump**. Das Kommandozeilen-Tool schneidet Traffic auf dem Netzwerk-Interface mit. Es läuft auch auf Android, setzt allerdings root voraus. Darum kann man sich mit der kostenlosen App **tPacketCapture** herummogeln: Sie arbeitet als lokaler VPN-Server und zeichnet sämtlichen Datenverkehr im PCAP-Format auf. Die Mitschnitte können Sie anschließend auf einen Rechner kopieren und komfortabel mit **Wireshark** analysieren. Anders als Packet Capture arbeitet tPacketCapture komplett passiv – das bedeutet auch, dass es SSL nicht im Klartext anzeigt.

Etwas chaotisch wirkt die Analyse-App **SandroProxy**. Wer sich hineinfuchst, wird mit einem gewaltigen Funktionsumfang belohnt: Die App entschlüsselt SSL-Verbindungen, schneidet PCAP-Dateien mit und bietet sogar ein Webinterface, über das man den Traffic bei Bedarf auf einem Gerät mit großen Bildschirm untersuchen kann. Dort kann man die durchgeleiteten Datenpakete auch in räumlichen Ansichten beobachten, in denen die Datenschnipsel etwa auf eine Kugel projiziert werden – eine nette Spielerei. SandroProxy ist umfassend konfigurierbar und über Java-Code erweiterbar. Damit die Daten fließen, muss man in den Einstellungen der WLAN-Verbindung als Proxy „localhost“ auf Port 8008 einstellen.



Der Net Monitor liefert einen ersten Überblick über kommunizierende Apps und Verbindungsziele.

kann man auch einen Android-Emulator auf dem Rechner nutzen.

Unabhängig von der eingesetzten Android-Version kann das sogenannte Public Key Pinning die Analyse erschweren. Beim Pinning baut die App verschlüsselte Verbindungen nur dann auf, wenn das SSL-Zertifikat des Verbindungspartners von einem bestimmten Herausgeber (CA oder Sub-CA) signiert wurde. Manchmal werden sogar nur bestimmte SSL-Zertifikate akzeptiert. Die Information, welche Zertifikate zulässig sind, ist entweder im Code verankert oder wurde bei einer erfolgreichen Verbindung für einen bestimmten Zeitraum abgefragt. Erfahrungsgemäß setzen nur sehr wenige Apps das Pinning ein. Meist geht es dabei ums Geld: So pinnen etwa viele Banking-Apps, um ihre verschlüsselte Kommunikation zu schützen.

Möchten Sie den Traffic einer verdächtigen App untersuchen und stoßen dabei auf diese Schutzfunktion, haben Sie zwei Optionen: Entweder legen Sie die Pinning-Funktion im Code lahm, indem Sie die APK-Datei modifizieren, oder Sie rooten das Android-System und installieren ein Xposed-Modul wie SSL Unpinning, das versucht, Pinning für einzelne Apps abzuschalten (siehe c't-Link). Beides sollten Sie nur auf einem dafür abgestellten Testgerät oder im Emulator durchführen, da Sie damit eine wichtige Schutzfunktion aushebeln.

Tiefergehende Analyse

Analyse-Apps wie Packet Capture eignen sich hervorragend, um Datenschleudern mit geringem Aufwand auf frischer Tat zu ertappen – überall und jederzeit. Bei umfangreicheren Analysen kann es sich jedoch weiterhin lohnen, einen ausgewachsenen Analyse-Proxy auf dem Rechner zu installieren und dort den Traffic des Android-Geräts durchzuschleusen. Es ist komfortabler, einen langen Datenaustausch auf einem großen Bildschirm mit Tools wie Burp oder mitmproxy zu analysieren. So kann man die Datenpakete etwa vor der Übertragung an den Ziel-Server abfangen und in Ruhe einsehen, um vertrauliche Daten aufzuhalten. Einen Einstieg in Burp liefert [1] und in [2] erfahren Sie, wie Sie ein Analyse-System aus Raspberry Pi und mitmproxy bauen.

(rei@ct.de) c't

Literatur

[1] Achim Barczok, Ronald Eikenberg, David Wischnjak, Durchleuchtet, Schnüffel-Apps durch Analyse und Monitoring aufdecken, c't 9/15, S. 130

[2] Mirko Dölle, Mittendrin aufgehackt, Raspberry Pi als Hacking-Werkzeug für SSL- und Man-in-the-Middle-Angriffe, c't 10/16, S. 88

Analyse-Apps zum Download: ***ct.de/y9fh***