

Mittendrin aufgehackt

Raspberry Pi als Hacking-Werkzeug für SSL- und Man-in-the-Middle-Angriffe

Wissen Sie eigentlich, was Ihr PC und Smartphone an Daten ins Internet übertragen? Mit einem zum Hacking-Werkzeug umgebauten Raspberry Pi können Sie den Netzwerkverkehr analysieren – und sogar verschlüsselte Daten abfangen. Wir zeigen, wie Sie sich das Gerät leicht selbst bauen.

VON MIRKO DÖLLE

Wehe, wenn sie eingeschaltet: iPhone und iPad melden sich ganz selbstverständlich in Cupertino und fragen nach den neuesten Updates, Android-Geräte kontaktieren zuvorkommend ihre Hersteller, und Dutzende installierte Apps im Hintergrund sind auch äußerst kommunikationsfreudig, wenn sie einen Internetzugang entdeckt haben. Beim PC ist es kaum besser, nicht nur das Betriebssystem telefoniert auf der Suche nach Updates nach Hause, auch der Virens scanner lädt sich sofort die neuesten

Signaturen herunter und bekundet nebenbei seine Anwesenheit im Netz der Netze – welche Dienste sonst noch im Hintergrund Daten übertragen, wissen nur die Wenigsten. Bei vernetzten Geräten wie NAS, IP-Kamera oder intelligenter Heizungssteuerung hat man meist gar keinen Einblick, was wann wohin gesendet wird.

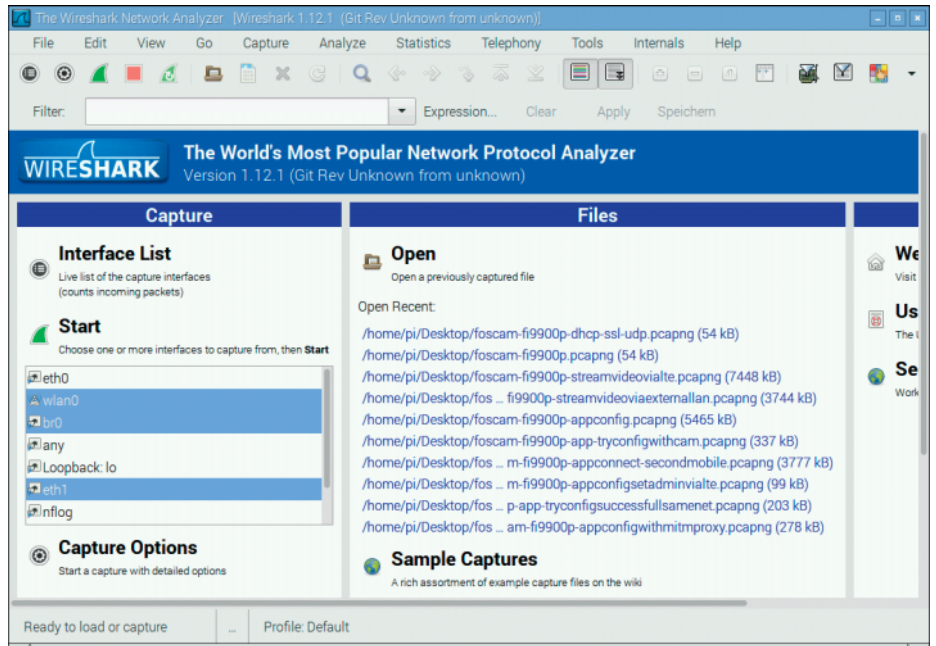
Solche Einblicke bringt ein zum Hacking-Werkzeug umgebauter Raspberry Pi, der sich als WLAN- und Ethernet-Router ausgibt und als „Raspi in the Middle“ den Datenverkehr belauscht – so-

gar, wenn dieser verschlüsselt ist. Dazu kommen der Netzwerk-Sniffer Wireshark und zwei spezielle Proxy-Dienste zum Einsatz: Der eine versucht, Links zu verschlüsselten Websites gegen unverschlüsselte URLs auszutauschen, der zweite greift die Verschlüsselung mit gefälschten Zertifikaten an und enthüllt so, was die Programme tatsächlich übertragen. Der Nachbau kostet gut 60 Euro.

Raspi als Router

Seine Internetverbindung erhält der Raspi über seine Ethernet-Buchse, etwa von Ihrem DSL-Router, und stellt sie als NAT-Router den Opfern zur Verfügung. Damit sich auch der Netzwerkverkehr von Smartphones und anderen drahtlosen Geräten mitschneiden lässt, empfiehlt sich der Raspberry Pi 3 – Sie können aber auch einen Raspi 2 mit einem WLAN-Adapter ausstatten. Ein USB-Ethernet-Adapter für gut 10 Euro erlaubt außerdem die Beobachtung kabelgebundener Netzwerkgeräte.

Als Betriebssystem verwenden wir Raspbian, dessen Einrichtung auf Seite 84 beschrieben ist. Die Netzwerkkonfiguration erfolgt jedoch nicht über den Network Manager, sondern manuell in der Datei /etc/network/interfaces. Hintergrund dafür ist, dass WLAN und USB-Ethernet-Adapter später gemeinsam genutzt werden sollen. Seine eigene Internetanbindung (WAN) erhält der Raspi über den internen Ethernet-Anschluss, die Konfiguration er-



Wireshark eignet sich gut als Netzwerk-Sniffer. Durch die Wahl des richtigen Interface – br0 für alle Opfer, eth1 für die per Ethernet angeschlossenen und wlan0 für die WLAN-Geräte – behält man leichter den Überblick.

folgt per DHCP – weshalb Sie folgende Zeilen an die Datei /etc/network/interfaces anfügen:

```
auto eth0
iface eth0 inet dhcp
```

Für die Opfergeräte, die per USB-Ethernet- oder WLAN-Adapter angebunden sind, soll der Raspi als Router inklusive NAT (Network Address Translation) fungieren und ihnen eine Internetverbindung bereitstellen. Damit wie bei einem herkömmlichen WLAN-Router auch die per

Ethernet angebundene Geräte jene aus dem WLAN kontaktieren können und sich außerdem im selben Subnetz befinden, fassen Sie den USB-Ethernet-Adapter eth1 und den WLAN-Adapter wlan0 zu einer Bridge zusammen. Dazu müssen Sie das Paket bridge-utils nachinstallieren, die Konfiguration der Bridge erfolgt in der Datei /etc/network/interfaces durch folgende Zeilen:

```
auto br0
iface br0 inet static
    bridge_ports eth1 wlan0
    address 192.168.250.1
    netmask 255.255.255.0
```

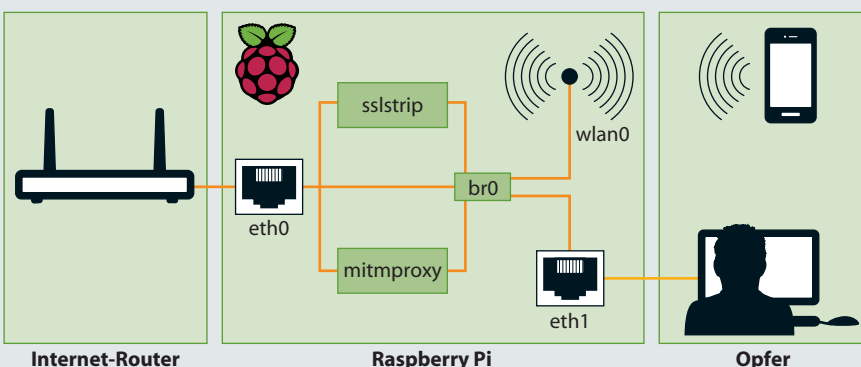
Das fügt die Schnittstellen eth1 und wlan0 zur Bridge br0 zusammen und konfiguriert die Bridge mit der statischen IP-Adresse 192.168.250.1. Damit die Schnittstellen eth1 und wlan0 unkonfiguriert bleiben und auch vom Network Manager in Ruhe gelassen werden, müssen Sie noch zwei weitere Zeilen ergänzen:

```
iface eth1 inet manual
iface wlan0 inet manual
```

Damit der Raspberry Pi als WLAN-Access-Point arbeitet, installieren Sie das Paket hostapd nach. Anschließend laden Sie sich über den c't-Link am Ende des Artikels die von uns vorbereitete Konfigurationsdatei hostapd.conf herunter, speichern sie in /etc/hostapd und passen die Einstellungen an – vor allem die Netz-

Raspi in the Middle

Über einen USB-Ethernet- und einen WLAN-Adapter, die zur Bridge zusammengefasst sind, arbeitet der Raspberry Pi wie ein herkömmlicher WLAN-Router inklusive NAT. Seine eigene Internetverbindung erhält er über die interne Ethernet-Schnittstelle.



werk- und Bridge-Einstellungen sowie die SSID und die Verschlüsselung:

```
interface=wlan0
bridge=br0
ssid=raspi-in-the-middle
wpa=1
wpa_passphrase=raspi-in-the-middle
wpa_key_mgmt=WPA-PSK
wpa_pairwise=CCMP
rsn_pairwise=CCMP
```

Das Beispiel zeigt die Konfiguration des WPA-verschlüsselten WLAN raspi-in-the-middle mit dem gleichnamigen Pre-Shared Key. Wir haben die SSID absichtlich so auffällig gewählt, damit sich niemand versehentlich mit dem WLAN verbindet. Verwenden Sie einen günstigen WLAN-Adapter mit Realtek-Chip, benötigen Sie den HostAP-Daemon des Herstellers, den Adafruit fertig übersetzt zum Download anbietet (c't-Link am Ende des Artikels). Kopieren Sie die Binärdatei aus dem Zip-Archiv einfach in das Verzeichnis

/usr/sbin. Außerdem müssen Sie den Namen des Treibers in der Konfigurationsdatei hostapd.conf korrigieren:

```
driver=rtl871xdrv
```

Für den Raspi 3 und alle anderen WLAN-Adapter können Sie den Standard-Treiber nl80211 und den vorinstallierten HostAP-Daemon benutzen. Damit der Dienst in Zukunft automatisch startet, tragen Sie noch den Pfad der Konfigurationsdatei in /etc/default/hostapd unter DAEMON_CONF ein.

DHCP für die Opfer

Wie für einen WLAN-Router üblich sollte der Raspi seinen Opfern auch einen DHCP- und DNS-Server bereitstellen. Diese Aufgabe übernimmt der Dienst dnsmasq aus dem gleichnamigen Paket. Damit der Dienst lediglich die USB-Adapter bedient, müssen Sie die Bridge als einziges Interface in der Datei /etc/dnsmasq.conf angeben. Außerdem benö-

tigt der Dienst noch den Adressbereich für die DHCP-Clients:

```
interface=br0
dhcp-range=192.168.250.50,192.168.250.150,24h
```

Schließlich müssen Sie noch dafür sorgen, dass der Raspi künftig die von den USB-Adaptoren empfangenen Pakete ins Internet weiterleitet. Dazu aktivieren Sie die Paketweiterleitung für IPv4 und IPv6 durch folgende Einträge in der Datei /etc/sysctl.conf:

```
net.ipv4.ip_forward=1
net.ipv6.conf.all.forwarding=1
```

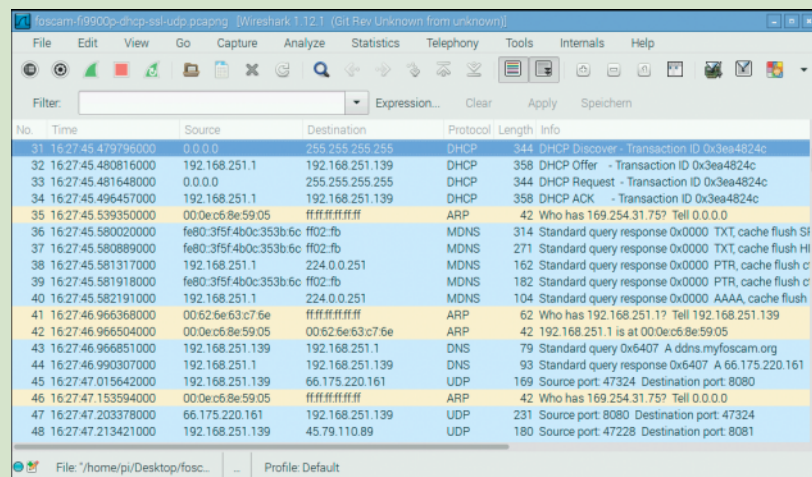
Nun fehlt noch die Adressumsetzung per NAT, damit die Antwortpakete aus dem Internet auch ihren Weg zu den Opfergeräten finden. Dafür ist die Firewall des Raspi zuständig, der notwendige Befehl lautet:

```
sudo iptables -t nat -A POSTROUTING \
-o eth0 -j MASQUERADE
```

Telefonitis diagnostiziert

Ob ein Gerät von sich aus Server im Internet kontaktiert, lässt sich mit Wireshark auf dem Raspi in the Middle leicht feststellen. Dazu starten Sie zuerst Wireshark und wählen als Device die Schnittstelle aus, an der Sie das Opfergerät anschließen wollen. Wir haben das mit einer IP-Kamera von Foscam [1] am Ethernet-Anschluss eth1 ausprobiert. Zuletzt schließen Sie das Kamera-Netzteil an und sehen in Wireshark von der ersten Sekunde an, was das Opfer tut.

Dabei ist es meist gar nicht nötig, in die Datenpakete (siehe Abbildung links) selbst hineinzusehen: Schon anhand der Kommunikationspartner und der Kurzinformation in der letzten Spalte von Wireshark sehen Sie, dass die Kamera unmittelbar nach Zuweisung der Adresse 192.168.251.139 per DHCP (Pakete 31 bis 34) den Nameserver unseres Raspi (192.168.251.1) nach der Adresse von Foscams DDNS-Dienst fragt (Paket 44). Die Antwort aus Paket 44, die IP-Adresse 66.175.220.161, benutzt die Kamera gleich darauf zum Aufbau einer UDP-Verbindung (Pakete 45 und 47). Zudem baut die Kamera mit Paket 48 eine Verbindung zur IP-Adresse 45.79.110.89 auf – mit unbekanntem Zweck. Das Log beweist auch, dass die Kamera die Adresse des zweiten Servers nicht beim Nameserver erfragt hat, sie also entweder fest programmiert oder zuvor von Foscams DDNS-Dienst übertragen worden sein muss.



Ein Blick auf das Kommunikationsprotokoll enthüllt, dass die IP-Kamera sofort zwei Server kontaktiert, nachdem sie per DHCP eine IP-Adresse erhalten hat.

Damit Sie diesen Befehl nicht nach jedem Neustart wieder neu eingeben müssen, installieren Sie das Paket `iptables-persistent` nach und lassen im automatisch gestarteten Konfigurationsdialog des Pakets die aktuelle Firewall-Konfiguration speichern. Haben Sie später weitere Regeln hinzugefügt, die Sie automatisch nach jedem Start wiederherstellen lassen wollen, genügt der Aufruf von

```
dpkg-reconfigure iptables-persistent
```

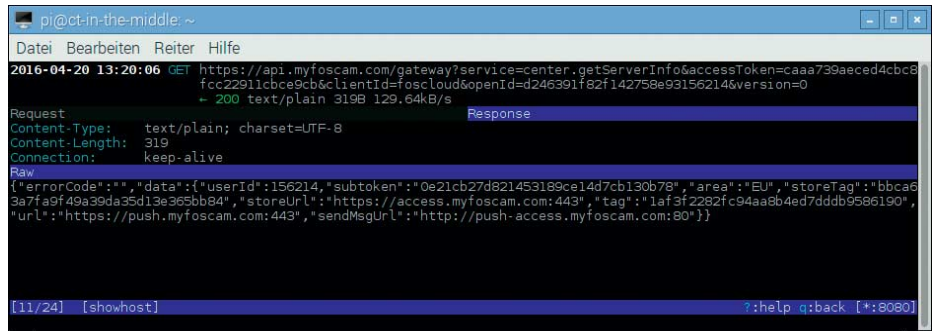
um die aktuellen Firewall-Regeln dauerhaft zu speichern. Damit ist die Konfiguration des Raspberry Pi als NAT-Router abgeschlossen und Sie sollten den Mini-Rechner neu starten, damit alle Dienste und Einstellungen neu geladen werden.

Abgehört

Sie können nun die Opfer-Geräte mit dem Raspberry Pi verbinden. Für die Beobachtung empfehlen wir Ihnen das grafische Sniffing-Tool Wireshark aus dem gleichnamigen Paket: Das Programm können Sie auch auf Ihrem PC unter Windows, Mac OS oder Linux einsetzen und dort bequem zuvor auf dem Raspberry Pi gespeicherte Logs analysieren. Ein Beispiel, wie Sie mit Wireshark in der Praxis solchen Plaudertaschen auf die Spur kommen, zeigt der Kasten links.

Beim Start von Wireshark wählen Sie entweder eine oder mehrere Schnittstellen aus, die Wireshark abhören soll, oder öffnen ein Log. Bei der Schnittstellenauswahl können Sie auf dem Raspi sowohl die Bridge `br0` als auch die physischen Geräte `eth1` oder `wlan0` auswählen. Das hilft, den Datenverkehr zu begrenzen: So wählen Sie am besten nur `wlan0`, wenn Sie etwa erst einmal die Kommunikation einer Smartphone-App untersuchen wollen. Geht es Ihnen um die Daten, die zwei per Ethernet und WLAN verbundene Opfer untereinander und mit dem Internet austauschen, ist `br0` eine gute Wahl. Auch die Daten, die der Raspi ins Internet überträgt, können Sie sich ansehen, indem Sie die Schnittstelle `eth0` überwachen lassen.

Das Wireshark-Fenster ist dreigeteilt: Oben werden Quelle, Ziel und Typ des empfangenen Pakets aufgelistet, darunter ist das Protokoll aufgeschlüsselt und unten steht schließlich der Inhalt des Pakets in hexadezimaler Darstellung. Für eine erste Analyse genügt das oberste Fenster,



```

pi@ct-in-the-middle: ~
Datei Bearbeiten Reiter Hilfe
2016-04-20 13:20:06 GET https://api.myfoscam.com/gateway?service=center.getServerInfo&accessToken=caaa739aeced4cbcbfcc22911cbce9cb6c&clientId=fosc&loud&openId=d246391f82f142758e931562146&version=0
  - 200 text/plain 3198 129.64KB/s
Request
Content-Type: text/plain; charset=UTF-8
Content-Length: 319
Connection: keep-alive
Response
Raw
{"errorCode":"","data":{"userId":156214,"subtoken":"0e21cb27d821453189ce14d7cb130b78","area":"EU","storeTag":"bbca63a7fa9f49a39da35d13a365bb84","storeUrl":"https://access.myfoscam.com:443","tag":"1af3f2282fc94aa8b4ed7ddb9586190","url":"https://push.myfoscam.com:443","sendMsgUrl":"http://push-access.myfoscam.com:80"}}
[11/24] [showhost] ?help q:back [*:8080]

```

Der Man-in-the-Middle-Proxy `mitmproxy` schiebt dem Opfer gefälschte Zertifikate unter und kann so den verschlüsselten Datenverkehr leicht entschlüsseln.

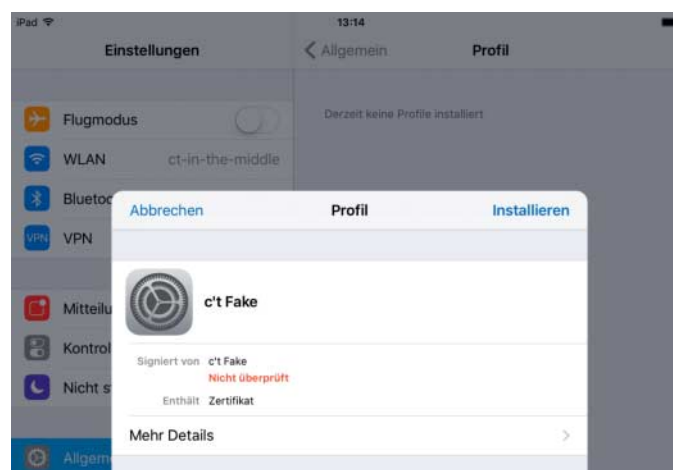
dort sehen Sie schnell, wer mit wem kommuniziert und welche Art Daten ausgetauscht werden – etwa eine DHCP-Anfrage. Über das Textfeld oberhalb des oberen Darstellungsbereichs können Sie Filterkriterien eingeben. Interessiert Sie etwa die Kommunikation, die von oder an das Gerät mit der IP-Adresse `192.168.250.139` geht, tragen Sie `ip.addr==192.168.250.139` als Suchkriterium ein. Um den Datenverkehr zwischen diesem und einem zweiten Gerät mit der IP-Adresse `192.168.250.103` zu beobachten, kombinieren Sie beide Suchkriterien zu `(ip.addr==192.168.250.139) && (ip.addr==192.168.250.103)`. Ein Klick auf die Schaltfläche „Expression“ öffnet ein Fenster mit allen verfügbaren Filterfunktionen und erleichtert Ihnen die Eingabe.

Der unterste Bereich des Fensters ist interessant, wenn Sie einen verdächtigen Datenaustausch beobachtet haben und nun genau wissen wollen, welche Daten dabei übertragen wurden. Das gelingt al-

erdings nur bei unverschlüsseltem Datenaustausch, SSL-verschlüsselte Pakete kann Wireshark nicht im Klartext anzeigen.

Verschlüsselung angreifen

Das Programm `sslstrip` aus dem gleichnamigen Paket ist ein erster Schritt, um die normalerweise verschlüsselte Kommunikation mitzuhören. Das Programm arbeitet als transparenter HTTP-zu-HTTPS-Proxy und ersetzt in über ihn abgefragten HTML-Seiten HTTPS-URLs durch HTTP-URLs und merkt sich, für welche Adressen diese Ersetzung stattgefunden hat. Lädt das Opfer eine von `sslstrip` modifizierte HTTP-URL, ruft `sslstrip` die Original-HTTPS-URL verschlüsselt ab und leitet das Ergebnis unverschlüsselt an das Opfer weiter. So kann der Server gar nicht bemerken, dass der Datenverkehr nur bis `sslstrip` verschlüsselt ist, das Opfer und somit auch Wireshark aber unverschlüsselte Daten erhalten. Der Kasten auf der nächsten Seite beschreibt, wie Sie damit zum



Damit Browser und Apps die von `mitmproxy` ausgestellten Fake-Zertifikate klaglos akzeptieren, importieren Sie das vom Programm generierte CA-Zertifikat auf dem Opfer. Unter iOS wird es als Profil abgelegt.

Beispiel an Zugangsdaten verschlüsselter Websites gelangen.

Um den Datenverkehr der Opfer durch `sslstrip` zu leiten, müssen Sie (mit Root-Rechten) eine Firewall-Regel hinzufügen, die für Port 80 bestimmte Anfragen auf den Standard-Port 10000 von `sslstrip` umleitet:

```
iptables -t nat -A PREROUTING -i br0 \
-p tcp --dport 80 -j REDIRECT \
--to-port 10000
```

Verwendet der HTTP-Server der Gegen-seite einen anderen Port, so müssen Sie dafür eine weitere Regel hinzufügen. Anschließend starten Sie `sslstrip` ohne wei-

tere Parameter, wozu Sie keine besonderen Rechte mehr benötigen. Das Programm läuft im Vordergrund und meldet seine Veränderungen an den HTML-Seiten auch auf der Konsole.

Lässt sich eine SSL-verschlüsselte Verbindung nicht vermeiden, etwa weil die HTTPS-URLs dynamisch mit JavaScript erzeugt werden oder das Opfer eine Adresse direkt ansteuert, können Sie versuchen, mit dem transparenten SSL-Proxy `mitmproxy` und gefälschten Zertifikaten trotzdem an den Inhalt zu gelangen. Während `sslstrip` die URLs in den HTML-Seiten manipuliert, sitzt `mitmproxy` (Man-in-the-Middle Proxy) im Datenstrom und verwendet für verschlüsselte Verbindungen gefälschte SSL-Zertifikate. Signiert sind die Fake-Zertifikate von einer eigenen CA (Certificate Authority), die `mitmproxy` beim ersten Start generiert.

Damit SSL-Anfragen über `mitmproxy` umgeleitet werden, müssen Sie wie bei `sslstrip` eine Firewall-Regel einrichten:

```
iptables -t nat -A PREROUTING -i br0 \
-p tcp --dport 443 -j REDIRECT \
--to-port 8080
```

Das Programm wird interaktiv im Terminal bedient und listet die URLs der Datenpakete auf, die von ihm mit gefälschten Zertifikaten übertragen wurden. Indem Sie ein Paket mit den Cursor-Tasten auswählen und dann Enter drücken, gelangen Sie in die Detailansicht, wo Sie die genaue Anfrage und, nachdem Sie mit der Tabulator-Taste umgeschaltet haben, die Antwort des Servers sehen können.

Bei aktuellen Browsern bewirkt der Austausch der Zertifikate, dass die Verbindung entweder gänzlich abgelehnt wird oder der Browser vor einem ungültigen Zertifikat warnt. Wird das Zertifikat vom Browser oder einer App ohne jegliche Warnung akzeptiert, ist dies ein konkreter Hinweis darauf, dass es das Programm mit der Sicherheit nicht so genau nimmt. Das gilt auch für manche Smartphone-App, der Kasten rechts zeigt, wie wir mit `mitmproxy` den Passwortverrat einer IP-Kamera-App unter iOS nachweisen konnten.

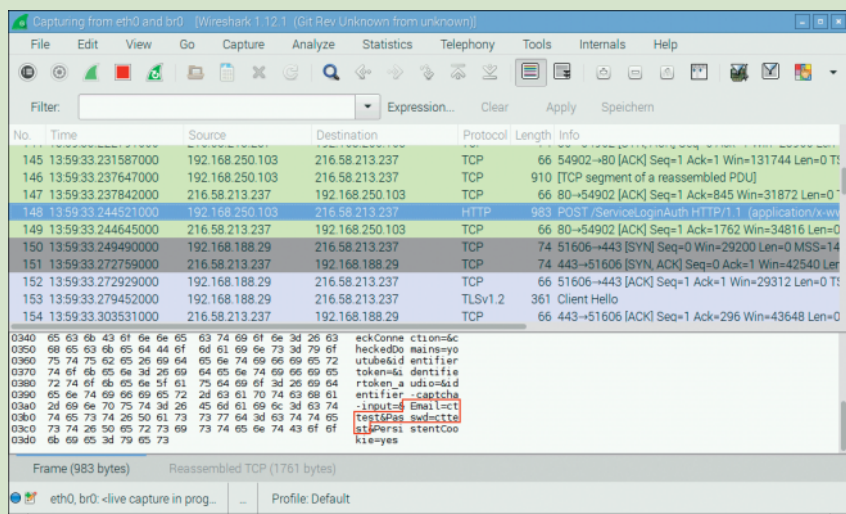
Um bei aufmerksamen Programmen trotzdem den Datenverkehr kontrollieren zu können, importieren Sie einfach die von `mitmproxy` generierten CA-Zertifikate auf dem Opfergerät. Damit wird `mitmproxy` für die Opfer zu einer vertrauens-

Zugangsdaten ausgespäht

Die Verschlüsselung von Websites lässt sich mit `sslstrip` aushebeln, etwa um an Zugangsdaten zu gelangen. Dazu starten Sie `sslstrip` auf dem Raspi im Terminal und richten die Firewall-Regel ein, die den Surfverkehr auf `sslstrip` umlenkt. Anschließend lauschen Sie mit Wireshark auf `br0` oder dem Device, über das das Opfer angebunden ist. Erst jetzt starten Sie den Browser des Opfers. Dabei nutzt `sslstrip` aus, dass selbst bei der Eingabe eines Domainnamens wie „google.de“ in der Adresszeile des Browsers zunächst eine HTTP-Verbindung aufgebaut wird.

Spätestens für den Anmeldevorgang wird der Benutzer auf eine verschlüsselte Seite weitergeleitet –

und hier greift `sslstrip` an: Der transparente Proxy ersetzt den HTTPS-Link auf die Anmeldeseite durch einen unverschlüsselten HTTP-Link, sodass der Browser weiterhin unverschlüsselte Kommunikation erwartet und auch akzeptiert. Klickt man nun auf den Anmeldelink, baut `sslstrip` als Proxy in der Mitte eine HTTPS-Verbindung zur Anmeldeseite auf – liefert das Ergebnis aber unverschlüsselt an den Browser zurück. Den unverschlüsselten Teil können Sie problemlos in Wireshark beobachten und so etwa Passwörter ausspähen. Das Opfer bemerkt diesen Angriff nur, wenn es darauf achtet, ob die Anmeldeseite wirklich per HTTPS im Browser abgerufen wird.



Der transparente Proxy `sslstrip` ersetzt HTTPS-Links durch HTTP-URLs. So können Sie in Wireshark die unverschlüsselte Kommunikation zwischen Opfer und `sslstrip` leicht mitlesen.

würdigen CA, deren Fake-Zertifikate andstandslos als echt akzeptiert werden.

Die dafür nötigen CA-Zertifikate finden Sie im Verzeichnis `.mitmproxy`, der Name beginnt mit `mitmproxy-ca-cert`, und sie haben die Endungen `.pem`, `.cer`, und `.p12`. Letztere ist das CA-Zertifikat im PKCS#12-Format, die beiden anderen enthalten das Zertifikat im PEM-Format. Am leichtesten ist es, die Zertifikate per Browser des Opfergeräts zu importieren. Dazu installieren Sie auf dem Raspi das Paket `apache2`. Im nächsten Schritt legen Sie im Apache-Dokumentenverzeichnis Hardlinks zu den von `mitmproxy` generierten Zertifikatdateien an:

```
sudo ln /root/.mitmproxy/mitmproxy- \
  ↪ca-cert.pem /var/www/html/ca-cert.pem
sudo ln /root/.mitmproxy/mitmproxy- \
  ↪ca-cert.cer /var/www/html/ca-cert.cer
sudo ln /root/.mitmproxy/mitmproxy- \
  ↪ca-cert.p12 /var/www/html/ca-cert.p12
```

Schließlich legen Sie im Document-Root des Webservers, dem Verzeichnis `/var/www/html`, eine neue Datei `index.html` an, die lediglich drei Links auf die drei Zertifikatsdateien erhält. Rufen Sie dann die URL `http://192.168.250.1` von einem der Opfergeräte auf, sehen Sie die Links auf die Zertifikate und können sie direkt herunterladen und einrichten.

Verweigert eine App oder eine Anwendung die Zusammenarbeit trotzdem, kann es daran liegen, dass sie eine eigene Liste vertrauenswürdiger CAs pflegt oder dass die Zertifikate der betreffenden Websites gepinnt sind, also nur noch von bestimmten CAs akzeptiert werden. Können Sie das CA-Zertifikat von `mitmproxy` dort nicht hinzufügen, bleibt Ihnen der Blick in die Daten leider verwehrt. In den meisten Fällen können Sie nach dem Import des CA-Zertifikats auch SSL-verschlüsselte Aufrufe in `mitmproxy` mitlesen und so he-

rausbekommen, ob ungewöhnliche Kommunikation stattfindet oder sensible Daten übertragen werden. Sollten Sie fündig werden, freuen wir uns über Hinweise.

Vergessen Sie aber nicht, nach Abschluss einer Untersuchung das Fake-Zertifikat und die WLAN-Konfiguration für das Opfernnetz wieder zu löschen: Sonst laufen Sie Gefahr, dass sich ein Mobilgerät unbemerkt mit dem Opfernnetz verbindet und der vermeintlich verschlüsselte private Datenverkehr versehentlich lesbar übertragen wird. (mid@ct.de) **ct**

Literatur

[1] Mirko Dölle, *Passwort-Petze, Passwortverrat und Firewall-Untertunnelung bei Foscam-Kameras – und wie man es unterbindet*, c't 4/16, S. 74

HostAP-Konfiguration und Daemon für Realtek-WLAN-Adapter: ct.de/ytak

Verräter enttarnt

Baut etwa eine Smartphone-App unmittelbar eine SSL-Verbindung zu einem Server auf, ohne einem Link aus einem HTML-Dokument zu folgen, kann `sslstrip` keine Links austauschen und ist somit wirkungslos. Hilfe verspricht `mitmproxy` auf dem Raspi, das nicht die Daten, sondern die Verschlüsselung selbst angreift. Wir haben dieses Verfahren in [1] bei der Foscam-App für IP-Kameras angewandt. Haben Sie die Firewall-Regel eingerichtet, die HTTPS-Verkehr auf `mitmproxy` umleitet, starten Sie das Programm im Terminal. Erst dann öffnen Sie die App auf dem Opfer – den entschlüsselten Datenverkehr können Sie dann im Terminal mitlesen.

`Mitmproxy` verwendet ein gefälschtes Zertifikat, um den Datenverkehr mit der App zu verschlüsseln. Das kann das Opfer aber leicht bemerken, indem es die Signatur des Zertifikats überprüft: Da es von keiner als vertrauenswürdig eingestuften Certificate Authority signiert wurde, ist es höchstwahrscheinlich gefälscht. Ein Browser würde an dieser Stelle warnen. Die Foscam-App hingegen störte die Fälschung nicht, sodass `mitmproxy` den Datenverkehr entschlüsseln und somit auch die Übermittlung von Benutzernamen und Passwort von Foscams Server mitschneiden konnte (siehe Abbildung links).

Damit kritischere Apps das gefälschte Zertifikat als echt betrachten, genügt es meist, das von `mitmproxy` generierte CA-Zertifikat auf dem Opfer zu installieren. Dann ist die Fälschung korrekt von einer vermeintlich vertrauenswürdigen CA unterzeichnet und damit nicht zu beanstanden.

Der Mitschnitt von `mitmproxy` beweist, dass die Foscam-App Benutzername und Passwort vom Server abrufen – wenn auch verschlüsselt. Das gefälschte SSL-Zertifikat störte die App dabei nicht.

```
pi@ct-in-the-middle: ~
Datei Bearbeiten Reiter Hilfe
2016-01-26 13:37:45 GET https://api.myfoscam.com/gateway?openId=d246391f82f142758e9
3156214&clientId=foscloud&accessToken=4457802d1594430590225
638e3569094&service=user_ipc_setting_v2_0.list&version=
+ 200 text/plain 436B 139.32kB/s
Request Response
Content-Type: text/plain; charset=UTF-8
Content-Length: 436
Connection: keep-alive
Raw
{"errorCode":"","data":[{"id":245095,"userId":156214,"macAddr":"00626E63C76E","ipcU
id":"3N9FZGEAG2RZA4FX111ABZZZ","productType":0,"deviceType":1,"deviceName":"FI9900P
","username":"ctadmin","password":"ca3be21fla029d8de1fc6571617f1f14","additionInfo
":{"httpsPort":"","macAddr":"","deviceName":"FI9900P","httpPort":"","mediaPort":"","hasusertag":2,"supportP2p":1,"supportStore":1,"s
upportRichMedia":1}]}
```