

UniFi Protect | IP-Camera von Fremdherstellern einbinden

Inhaltsverzeichnis

- [1 Anforderungen](#)
- [2 Das Prinzip](#)
- [3 Zertifikat erstellen](#)
- [4 Adoption Token erstellen](#)
- [5 Den Proxy starten und die Kamera einbinden](#)

In diesem Artikel geht es darum eine Möglichkeit aufzuzeigen auch noch alte bzw. vorhandene IP-Kameras der Infrastruktur in UniFi Protect einzubinden.

Da ich aus dem Discord Channel darum gebeten wurde, habe ich nochmal eine deutsche Anleitung aus meiner eigenen Erfahrung dazu niedergeschrieben.

1 Anforderungen

- **Diese Anleitung setzt Grundkenntnisse in Docker sowie einen funktionierender Docker-Host voraus**
Der Proxy kann auch nativ laufen, siehe dazu die original Anleitung.
- **Die Fremdkamera muss RTSP unterstützen.**
Dies kann man z.B. testen, indem man mit VLC mal eine Url öffnet:
`rtsp://user:password@cameraip/`
- **Es wird ein einigermaßen Leistungsstarker PC benötigt.**
- **OpenSSL**
- **Linux-Shell oder unter Windows eine WSL Shell**
Alternativ müsst ihr die Befehle auf den Windows-Pfadsyntax anpassen.

2 Das Prinzip

Damit die Fremd-Kamera erkannt wird, müssen wir UniFi Protect vorgaukeln, dass es sich um eine UniFi Kamera handelt. Dies übernimmt der UniFi-Kamera-Proxy. Er gibt sich als UniFi Kamera aus, und übermittelt die Daten von einer beliebigen RTSP Quelle.

```
Fremd-Kamera --> UniFi Kamera Proxy --> UniFi Protect
```

Dieser Proxy muss permanent laufen, ist er aus, ist die Kamera für UniFi Protect auch aus.

Ich habe aktuell keine Erfahrungen wie viele Kameras auf einem Raspberry PI 4 laufen würden, ich vermute aber mal nicht all zu viele. Daher würde ich schon einen leistungsstärkeren Computer/Server für diese Aufgabe betreuen.

3 Zertifikat erstellen

Bevor es losgeht benötigen wir erstmal ein Fake-Zertifikat für die Kamera.

Dazu erstellen wir uns ein Verzeichnis in dem wir arbeiten. Ich habe z.B. /home/hefti/unifi-cam erstellt. Das könnte unter Windows aber auch C:\unifi-cam sein.

Wir öffnen eine Shell und wechseln in das Verzeichnis und führen darin folgende Befehle aus. Achtung: für Windows Pfade anpassen.

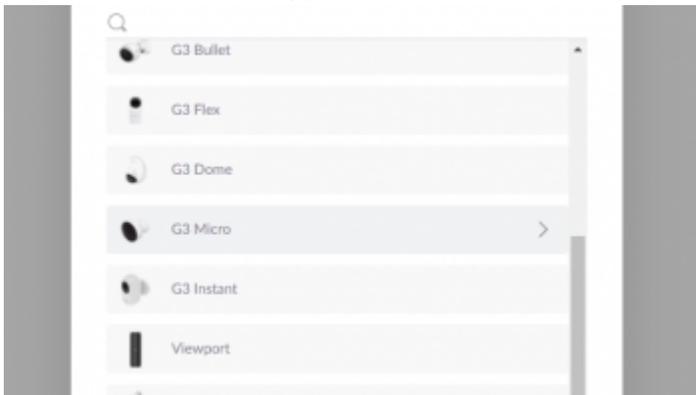
Code

```
openssl ecparam -out /tmp/private.key -name prime256v1 -genkey -noout  
openssl req -x509 -days 36500 -key /tmp/private.key -out /tmp/public.key  
cat /tmp/private.key /tmp/public.key > client.pem  
rm -f /tmp/private.key /tmp/public.key /tmp/server.csr
```

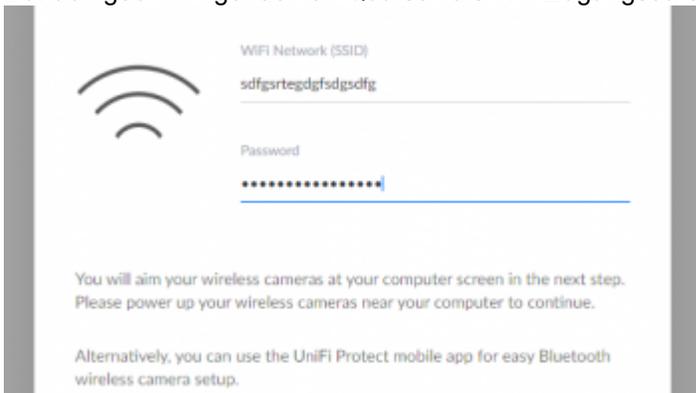
Jetzt sollte es im Verzeichnis eine client.pem geben die wir im späteren Schritt benötigen.

4 Adoption Token erstellen

1. Wir öffnen Unifi Protect, und fügen eine neue Kamera auf der Devices Seite hinzu.
<https://unifi/protect/devices> (Url kann je nach Installation variieren)
2. Als Kamera wählen wir G3 Micro



3. Auf der nächsten Dialog-Seite klicken wir auf Continue
4. Danach gebt ihr irgendeinen Quatsch als Wifi-Zugangsdaten ein



Und klickt anschließend auf Generate QR Code

Von dem QR Code macht ihr jetzt einen Screenshot. Unter Windows 10 habe ich das mittels [Windowstaste + Shift + S] gemacht und dann den QR-Code ausgeschnitten.

Snipping-Tool oder einfach die Drucken-Taste und dann in MsPaint.exe einfügen geht aber sicher auch.

Der QR Code muss jedenfalls als Bild gespeichert werden.

Danach geht auf folgende Webseite, ladet das soeben gespeicherte Bild hoch und schaut euch den kodierten Inhalt des QR Codes an.

[ZXing Decoder Online](#)

Im Abschnitt `Raw text` kopiert ihr jetzt die Zeichenkette der vorletzten Zeile.

z.B.

Code

```
Ro00npyC2nmjW3BLECXB8PSFJws2wCoA
```

Dies ist euer Adoption-Token.

Dieser ist nur zeitlich begrenzt gültig. Wenn ihr für den nächsten Schritt zulange benötigt, kann es sein, dass ihr nochmal einen neuen Token erstellen müsst.

5 Den Proxy starten und die Kamera einbinden

Bitte vergewissert euch nochmals, dass ihr die Kamera über VLC mit RTSP aufrufen könnt.

Jetzt starten wir unifi-camera-proxy als docker container.

Dazu müsst ihr

- den Pfad für die `client.pem` auf dem Host anpassen, in meinem Fall ist dies `/home/hefti/unifi-cam/client.pem`
- die Platzhalter inklusive der spitzen Klammern mit eueren Daten ersetzen
- und natürlich die richtige rtsp Url mitangeben.

Code

```
docker run --rm -v "/home/hefti/unifi-cam/client.pem:/client.pem" keshavdv/unifi-cam-proxy t
```

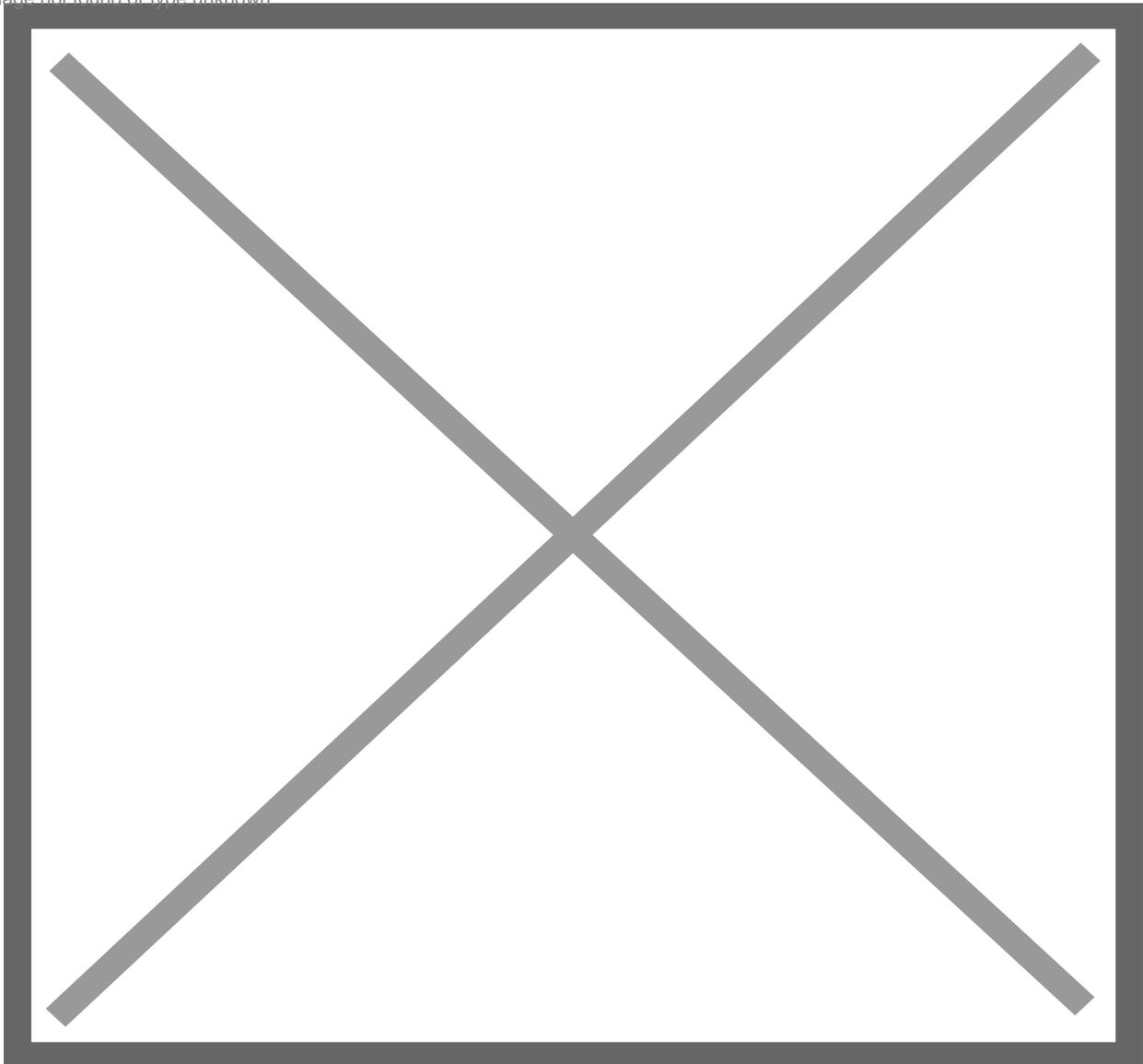
Ich habe hier bewusst den Parameter `--mac` mit angegeben, an letzter Stelle eine 01 für die 1. Kamera. Wenn ihr eine Zweite Kamera hinzufügen wollt, soltet ihr diese Zahl erhöhen auf z.b. 02 usw. für jede weitere

Kamera. Bei mir ist dies identisch mit der letzten Stelle der IPv4 Adresse der Kamera.
Die Kamera wird euch in Unifi-Protect dann auch mit den letzten beiden Stellen der MAC-Adresse angezeigt, also 0001.

Nachdem ihr den Befehl gestartet habt, sollte Docker den Proxy ausführen und eure Kamera in Unifi Protect auftauchen.

Hier der Link zur original Dokumentation / Anleitung:

Image not found or type unknown



[GitHub - keshavdv/unifi-cam-proxy: Enable non-Ubiquiti cameras to work with Unifi NVR](https://github.com/keshavdv/unifi-cam-proxy)

Enable non-Ubiquiti cameras to work with Unifi NVR - GitHub - keshavdv/unifi-cam-proxy: Enable non-Ubiquiti cameras to work with Unifi NVR
github.com

Disclaimer: Alle Anleitungen/Tutorials sind nach bestem Wissen und Gewissen verfasst, gehen immer von den definierten Software/Firmware-Versionen aus und sind auf das englische GUI ausgelegt.

Es gibt keine Garantien auf Erfolg. Im Falle eines Misserfolges hilft aber sicherlich die Community hier immer weiter.

Keiner der Autoren oder der Betreiber des Forums ist für die aus der Nutzung resultierenden Probleme/Herausforderungen verantwortlich.

Jegliche hier beschriebenen Schritte erfolgen ausnahmslos in eigener Verantwortung des Durchführenden. Eltern haften für ihre Kinder.

Auswählen: _____

Gültige Software-Version UniFi Protect 1.16.x UniFi Protect 1.10.x