Sicherheitstest
Nachtest Bürgerbeteiligungs-App

Stadt Tübingen

Abschlussbericht

Version 2.0

[redacted] SySS GmbH
[redacted]

31. Oktober 2018

Projekt

Kunde:	Stadt Tübingen Am Markt 1 72070 Tübingen
Beauftragtes Unternehmen:	SySS GmbH Schaffhausenstraße 77 72072 Tübingen
Projektverantwortlich:	[redacted] SySS GmbH [redacted]
Projektbeteiligte:	[redacted]
Version des Berichts:	2.0
Status des Berichts:	Entwurf ⇒ interne QS beendet ⇒ Finalisiert
Vertraulichkeit:	Aus Vertraulichkeitsgründen wurden einzelne Elemente durch die folgende Zeichenkette ersetzt: #

Dieser Bericht wurde inhaltlich von [redacted] qualitätsgesichert.

Dieser Bericht wurde sprachlich von [redacted] qualitätsgesichert.

Dieses Dokument ist elektronisch erstellt und ohne Unterschrift gültig.

Inhaltsverzeichnis

1 Executive Summary

1.1 Projektziele

Im Rahmen eines Penetrationstests soll die SySS GmbH die Android- und iOS-Apps sowie die Webapplikation für die geplanten Bürgerbefragungen auf Sicherheitsschwachstellen hin untersuchen.

Die SySS GmbH wird die Risiken aller gefundenen Schwachstellen objektiv bewerten und einen Katalog mit Vorschlägen zu deren Behebung oder Abschwächung aufstellen.

1.2 Zusammenfassung und strategische Maßnahmen

Die SySS GmbH führte den Sicherheitstest für die Stadt Tübingen vom 24. bis 28. August und vom 10. bis 13. September 2018 mit einem Umfang von insgesamt acht Personentagen durch.

Während des Penetrationstests gelang es der SySS GmbH, mehrere Schwachstellen mittleren Risikos bei der Webanwendung und einige Schwachstellen, welche mit einer niedrigen Kritikalität bewertet wurden, in den beiden Apps festzustellen.

Webanwendung

Bei den beiden als mittleres Risiko bewerteten Schwachstellen handelt es sich um fehlende Prüfmechanismen, die es einem Angreifer ermöglichen, Aktionen im Kontext eines Benutzers auf der Webanwendung durchzuführen, wenn dieser beispielsweise eine andere, entsprechend präparierte Seite besucht. Außerdem werden sensible Informationen über die Adresszeile des Browsers übertragen, was dazu führt, dass diese beispielsweise in Logdateien oder dem Browserverlauf Spuren hinterlassen. Dieser Umstand wurde mit einem niedrigen Risiko bewertet.

Apps

Die beiden Apps sind sehr ähnlich, weshalb die gefundenen Schwachstellen auf beide Apps gleichermaßen zutreffen. Bei keiner der beiden Apps besteht ein Mechanismus, der erkennt, ob der Benutzer des Gerätes dort administrative Rechte besitzt – was wiederum zahlreiche Sicherheitsmechanismen des jeweiligen Gerätes aushebelt. Um die Anwendung weiter zu härten, wird außerdem empfohlen, das für die Datenübertragung verwendete TLS-Zertifikat oder dessen Signatur mit den beiden Apps auszuliefern, um die Analyse der Kommunikation mit dem Back-End zu erschweren.

Insgesamt ist das zum Testzeitpunkt vorgefundene Sicherheitsniveau der beiden getesteten Apps als hoch zu bewerten, das der Webanwendung als leicht verbesserungswürdig. Die beiden als mittleres Risiko bewerteten Schwachstellen der Webanwendung sollten zeitnah behoben werden, die Risiken mit niedrigem Risiko sollten anschließend ebenfalls angegangen werden, um das Sicherheitsniveau weiter zu verbessern.

1.3 Nachtest am 23. Oktober 2018

Die SySS GmbH führte am 23. Oktober 2018 einen Nachtest durch. Dabei konnte festgestellt werden, dass die meisten Funde behoben wurden.

Im Nachtest konnte auf unterschiedlichen iOS-Geräten kein Mechanismus festgestellt werden, der erkennt, dass ein Benutzer administrative Rechte auf dem Gerät besitzt.

Allerdings konnte festgestellt werden, dass die Webseite vor der Anmeldung Informationen über Abstimmungen sendet, mit deren Hilfe unauthentifizierte Benutzer Abstimmungsergebnisse von beendeten Abstimmungen einsehen können. Da Abstimmungsergebnisse nach Auffassung der SySS GmbH ohnehin öffentlich sind beziehungsweise sein sollten, ist hier nicht von einem Sicherheitsrisiko auszugehen. Nichtsdestotrotz sollte dieses Verhalten geprüft, diskutiert und ggf. angepasst werden.

2 Technische Zusammenfassung

Alle Feststellungen sind in nachfolgender tabellarischer Übersicht zusammengefasst.

Risiko	Feststellung	Empfehlung	Referenz
M2.1 **(behoben)**	**Cross-Site Request Forgery:** Die Webapplikation ist anfällig für Cross-Site Request Forgery	Schutzmaßnahmen überprüfen; Gegenmaßnahmen wie Synchronizer Token implementieren	5.1 Seite 13
M2.2 **(behoben)**	**HTML5 Cross-Origin Resource Sharing:** Zugriff auf Ressourcen von beliebigen Domains aus ist erlaubt	Nur vertrauenswürdige Domains im Header `Access-Control-Allow-Origin` zulassen oder CORS unterbinden	5.2 Seite 14
L1.1 **(behoben)**	**TLS-Konfiguration:** Veraltete TLS-Protokollversion 1.0 wird unterstützt	Sofern möglich, ausschließlich TLSv1.2 oder höher einsetzen	4.1 Seite 11
L1.2	**Veraltete Software:** Installierte Software ist zum Teil veraltet	Umstand prüfen; Software stets auf aktuellem Stand halten	4.2 Seite 11
L2.1 **(behoben)**	**Sensible Informationen sind Teil der URL:** Authentifikationsinformationen werden im Query String übertragen	Authentifikationsinformationen nicht im Query String übertragen; sensible Informationen ausschließlich im Body eines POST-Requests übertragen	5.3 Seite 15
L3.1	**Jailbreak/Rooting Detection:** Applikation Vivi kann auf Gerät mit Jailbreak/Rooting installiert und ausgeführt werden	Anwendungsspezifische Überprüfungen bezüglich eines durchgeführten Jailbreak/ Rooting des Gerätes implementieren	6.2 Seite 19
L3.2 **(behoben)**	**Certificate Pinning:** Certificate Pinning ist nicht implementiert	Certificate Pinning implementieren, um die Verbindungssicherheit zu erhöhen	6.3 Seite 20
I1.1	**Informationspreisgabe:** Informationen über installierte Software werden zur Verfügung gestellt	Preisgabe von derartigen Informationen generell unterbinden beziehungsweise weitgehend einschränken	4.3 Seite 12
A2.1	**Öffentliche Umfragedaten:** Unauthentifizierte Benutzer können Beschreibungen und Ergebnisse der Befragungen einsehen	Umstand prüfen und ggf. anpassen	5.4 Seite 16

Anmerkung:
Sicherheitslücken werden farblich hervorgehoben, wobei Risiken von der SySS GmbH wie folgt definiert werden:

Hohe Risiken	Unautorisierte Manipulation oder Einsichtnahme von Daten. Die Werthaltigkeit wird dabei nicht berücksichtigt.
Mittlere Risiken	Sicherheitslücken, die in Kombination mit weiteren – auch menschlichen – Komponenten zu einem Sicherheitsvorfall führen können.
Niedrige Risiken	Schwächen, die keine Änderungen durch nicht authentifizierte Dritte ermöglichen, wie nur unter Laborbedingungen zurückrechenbare Verschlüsselungsmethoden oder die Unterstützung von Debugging-Optionen (zum Beispiel die HTTP-Methoden TRACE und TRACK).
Information Leaks	Informationen beispielsweise über eingesetzte Software, deren Vorhandensein zwar kein direktes Risiko darstellt, möglichen Angreifern aber zu detaillierte Daten liefern.
Anomalien	Damit werden unübliche Konfigurationen gekennzeichnet (beispielsweise ein DNS-Service auf UDP-Port 3000). Ein derartiges Verhalten beschreibt kein Sicherheitsproblem.
Risiko	Während der Projektlaufzeit oder seit dem letzten Test behobene Sicherheitsschwächen werden nicht farblich markiert, sind aber aus Gründen der Vollständigkeit mit aufgelistet.

2.1 Legende des Nachtests

Die folgenden Abschnitte enthalten die ursprünglichen Formulierungen der im Rahmen der initialen Sicherheitsüberprüfung ermittelten Auffälligkeiten. Die Ergebnisse der aktuellen Nachprüfung befinden sich in grau hinterlegten Absätzen.

Zur Kennzeichnung der überprüften Ergebnisse wird die folgende Syntax genutzt:

Ein grüner Haken zeigt an, dass ein Problem behoben wurde.

Schwachstellen mit diesem Symbol sind nicht behoben worden.

Schwachstellen mit diesem Symbol sind nur teilweise behoben worden.

Eine Box ohne Zeichen oder mit einem Informationszeichen enthält generelle Informationen bezüglich des Nachtests.

3 Allgemeine Informationen

Dieses Kapitel dokumentiert allgemeine Informationen zum durchgeführten Penetrationstest.

3.1 Testgegenstand

Die unter der Domain `vivitest-app.aaronprojects.de` (`160.44.193.135`) erreichbare Abstimmungsplattform sowie das zugehörige Management-Interface `vivitest-mi.aaronprojects.de/vivitest-client.aaronprojects.de` (`160.44.205.106/160.44.199.54`) und die beiden mobilen Bürgerbeteiligungs-Apps (iOS und Android) bilden den durch die Stadt Tübingen festgelegten Testgegenstand dieser Untersuchung. Weitere eventuell angebundene oder angesprochene Endpunkte beziehungsweise Services sind von der Sicherheitsanalyse ausgenommen.

3.2 Bereitgestellte Daten

Zur Nutzung des Dienstes wurden der SySS GmbH die unten stehenden Zugangsdaten mitgeteilt. Zudem wurden noch zwei gleichwertige lokale Clients zum Erzeugen von Endbenutzern sowie zum Signieren von über die Managementoberfläche erstellten Umfragen bereitgestellt.

Bereitgestellte und für den Test genutzte Zugangsdaten:

- `syss-u1:########`
- `syss-u2:########`

Bereitgestellte und für den Test genutzte Clients, einschließlich der SHA-256-Checksumme:

- `syss1-linux` – `98fc8855eef3f21479aad4f79a9043b6acc349a73ff620657cf817d7228a2e28`
- `syss2-linux` – `4d1722614cff374f4eecff5cf4757c2f20d8a0e1a043e493b83ebea66a3f9c35`

3.3 Quell-IP-Adressen

Die Untersuchungen wurden von den unter Tabelle 3.1 gelisteten IP-Adressen der SySS GmbH aus durchgeführt.

IP	Info
37.24.4.134	SySS GmbH
80.151.155.239	SySS GmbH
87.190.8.192/29	SySS GmbH
185.142.184.0/22	SySS GmbH
212.71.209.0/24	SySS GmbH
2001:14f8:203:1::/64	SySS GmbH
2a07:2f00:1337::/48	SySS GmbH

Tabelle 3.1: IP-Adressen der externen SySS-Systeme

3.4 Systemhoheit

Die zu testenden Systeme stehen nicht unter der Hoheit der Stadt Tübingen, es lagen jedoch für den Testzeitraum Genehmigungen der Deutschen Telekom Individual Solutions & Products GmbH sowie Amazon Web Services Inc. vor. Grundsätzlich gilt, dass Systeme Dritter aus dem Test ausgeschlossen und wegen fehlender Rechtsgrundlage nicht getestet werden, wenn keine Testfreigabe vorliegt.

3.5 Bereinigung von Testdaten

Um Interferenzen mit anderen Teilnehmern zu einem späteren Zeitpunkt auf der während des Sicherheitstests genutzten Umgebung auszuschließen, empfiehlt die SySS GmbH, alle während des Sicherheitstests angefallenen Daten nach Abschluss der Tests zu beseitigen und die Systeme auf einen definierten Ausgangszustand (vor dem Sicherheitstest) zu bringen.

4 Ergebnisse der Infrastrukturanalyse

Im Rahmen des durchgeführten Tests wurden die Systeme anhand eines Portscans auf erreichbare Dienste hin überprüft. Im Testzeitraum konnten mehrere aktive TCP-Dienste identifiziert werden (siehe Auflistung 4.1). Extern erreichbare UDP-Dienste konnten nicht eruiert werden.

```
Nmap scan report for vivitest-app.aaronprojects.de (160.44.193.135)
rDNS record for 160.44.193.135: ecs-160-44-193-135.reverse.open-telekom-cloud.com
Not shown: 65532 filtered ports
PORT    STATE SERVICE  VERSION
22/tcp  open  ssh      OpenSSH 7.4 (protocol 2.0)
80/tcp  open  http     nginx
443/tcp open  ssl/http nginx

Nmap scan report for vivitest-mi.aaronprojects.de (160.44.205.106)
rDNS record for 160.44.205.106: ecs-160-44-205-106.reverse.open-telekom-cloud.com
Not shown: 65532 filtered ports
PORT    STATE SERVICE  VERSION
22/tcp  open  ssh      OpenSSH 7.4 (protocol 2.0)
80/tcp  open  http     nginx
443/tcp open  ssl/http nginx

Nmap scan report for vivitest-client.aaronprojects.de (160.44.199.54)
rDNS record for 160.44.199.54: ecs-160-44-199-54.reverse.open-telekom-cloud.com
Not shown: 65532 filtered ports
PORT    STATE SERVICE  VERSION
22/tcp  open  ssh      OpenSSH 7.4 (protocol 2.0)
80/tcp  open  http     nginx
443/tcp open  ssl/http nginx
```

Auflistung 4.1: Ergebnisse des Portscans

```
Nmap scan report for vivitest-app.aaronprojects.de (160.44.193.135)
Host is up (0.011s latency).
rDNS record for 160.44.193.135: ecs-160-44-193-135.reverse.open-telekom-cloud.com
Not shown: 65531 filtered ports
PORT      STATE  SERVICE  VERSION
22/tcp    closed ssh
80/tcp    open   http     nginx
443/tcp   open   ssl/http nginx
31334/tcp open   ssh      OpenSSH 7.4 (protocol 2.0)

Nmap scan report for vivitest-mi.aaronprojects.de (160.44.205.106)
Host is up (0.012s latency).
rDNS record for 160.44.205.106: ecs-160-44-205-106.reverse.open-telekom-cloud.com
Not shown: 65531 filtered ports
PORT      STATE  SERVICE  VERSION
22/tcp    closed ssh
80/tcp    open   http     nginx
443/tcp   open   ssl/http nginx
31334/tcp open   ssh      OpenSSH 7.4 (protocol 2.0)
```

```
Nmap scan report for vivitest-client.aaronprojects.de (160.44.199.54)
Host is up (0.012s latency).
rDNS record for 160.44.199.54: ecs-160-44-199-54.reverse.open-telekom-cloud.com
Not shown: 65531 filtered ports
PORT      STATE  SERVICE  VERSION
22/tcp    closed ssh
80/tcp    open   http     nginx
443/tcp   open   ssl/http nginx
31334/tcp open   ssh      OpenSSH 7.4 (protocol 2.0)
```

Auflistung 4.2: Ergebnisse des Portscans während des Nachtests

4.1 Unterstützung von TLSv1.0

L1.1

Bei der Überprüfung der TLS-Konfiguration der drei Dienste auf Port 443/TCP stellte die SySS GmbH fest, dass diese TLS in der Protokollversion 1.0 unterstützen. Zum aktuellen Testzeitpunkt ist diese Version als nicht mehr ausreichend kryptografisch sicher zu erachten und sollte demnach nicht mehr angeboten werden.

Die SySS GmbH rät, TLS in der Protokollversion 1.0 zu deaktivieren. Zur weiteren zukünftigen Härtung des Dienstes könnte gegebenenfalls auch über die Deaktivierung der Unterstützung von TLSv1.1 nachgedacht werden. Dabei sollte jedoch bedacht werden, dass durch diese Maßnahme ältere Endgeräte von der Nutzung der Webseiten ausgeschlossen werden könnten.

Während des Nachtests war nur TLSv1.2 aktiviert.

4.2 Veraltete Software

L1.2

Die bereitgestellten Informationen über die eingesetzten Softwareversionen (siehe Auflistung 4.1) legen nahe, dass unter Umständen nicht alle Softwareprodukte auf dem aktuellen Stand sind. Dies muss aufgrund eines möglichen Backporting jedoch nicht den Tatsachen entsprechen und sollte jeweils manuell geprüft werden.

Die SySS GmbH empfiehlt, die eingesetzte Software auf fehlende Sicherheitspatches und ausgelaufene Unterstützung zu prüfen und gegebenenfalls die entsprechenden Produkte auf ihre jeweilig aktuell unterstützte Version zu aktualisieren.

Außerdem empfiehlt sich die Einführung eines geeigneten Patchmanagements, um die Systeme auch in Zukunft stets auf neuestem Stand zu halten.

OpenSSH wurde auch zum Zeitpunkt des Nachtests noch in Version 7.4 (siehe Auflistung 4.2, rot markiert) betrieben. Laut Kevin Fischer von Aaron Projects wird auf den Systemen Centos 7.5.1804 eingesetzt. In dieser Distribution ist OpenSSH 7.4p1 die aktuelle Version, welche auch vom Hersteller mit Sicherheistpatches versorgt wird. Lediglich der Port wurde von 22 auf 31334 geändert. Soll der SSH-Dienst nicht aus dem Internet erreichbar sein, wird empfohlen, auf IP-Whitelisting für diesen Port zurückzugreifen.

4.3 Informationspreisgabe

I1.1

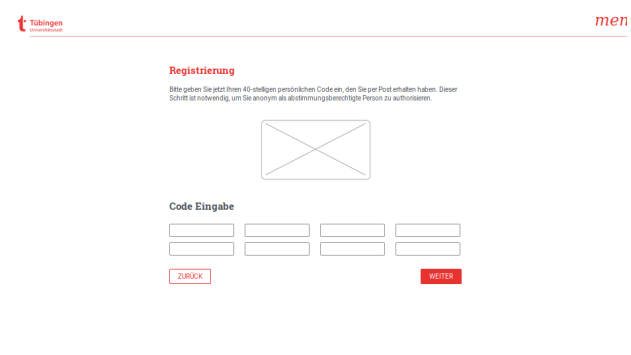
Die SSH-Dienste auf Port 22/TCP (siehe Auflistung 4.1) stellen über den ausgelieferten Banner Informationen über die installierte Software zur Verfügung. Diese Angaben können in Summe von einem Angreifer beispielsweise dafür genutzt werden, um gezielt Informationen über Schwachstellen oder mögliche Angriffsvektoren zu ermitteln und anschließend weitere zielgerichtete Angriffe auszuführen.

Die SySS GmbH rät, um die Informationsbasis für einen Angreifer so gering wie möglich zu halten, weder Software- noch Versionsinformationen preiszugeben.

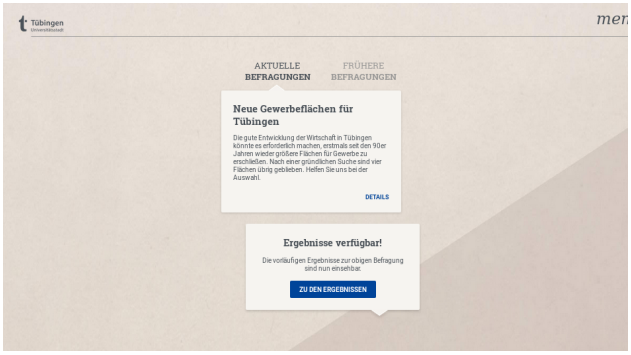
Die Version von OpenSSH wird immer noch preisgegeben.

5 Ergebnisse der Webapplikationsanalyse

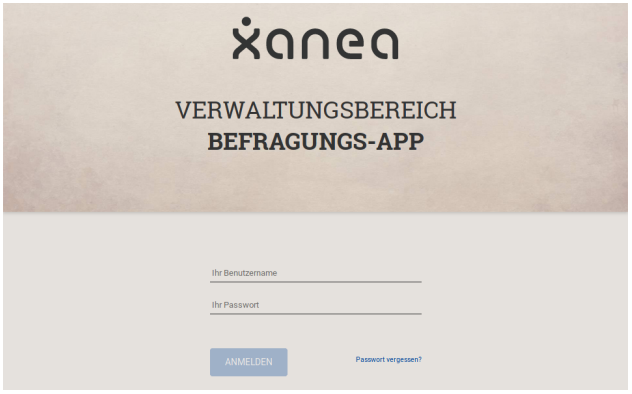
Dieses Kapitel dokumentiert die im Rahmen der Analyse der Webapplikationen `vivitest-app.aaronprojects.de` und `vivitest-mi.aaronprojects.de` (siehe Abbildung 5.1) identifizierten Sicherheitsschwachstellen.



(a) `vivitest-app.aaronprojects.de`



(b) `vivitest-app.aaronprojects.de`



(c) `vivitest-mi.aaronprojects.de`

(d) `vivitest-mi.aaronprojects.de`

Abbildung 5.1: Webapplikationen

5.1 Cross-Site Request Forgery

M2.1

Die Webapplikation ist anfällig für Cross-Site Request Forgery. Bei einem Cross-Site Request Forgery (XSRF/CSRF)-Angriff bringt ein Angreifer sein Opfer dazu, unbemerkt eine bestimmte, vorbereitete HTTP-Anfrage an die Webapplikation zu senden. Die HTTP-Anfrage kann häufig von einer externen Seite ausgelöst werden, etwa von einem Beitrag in einem Forum oder Wiki. Besucht das Opfer diese Seite, während es in der Webapplikation eingeloggt ist, so wird die Aktion im Kontext des Opfers ausgelöst. Die zur Identifikation des Benutzers nötigen Session-Parameter werden vom Browser automatisch mitgesendet.

Beispielsweise konnte die SySS GmbH eine Proof of Concept (PoC)-Seite erstellen (siehe Auflistung 5.1), bei deren Besuch durch einen Abstimmungsverantwortlichen die veröffentlichten Abstimmungsergebnisse für eine beliebige Umfrage, für welche man den `auth_client`-Wert kennt,

zurückgezogen werden. Der `auth_client`-Wert ist dabei jedem berechtigten Teilnehmer der Umfrage bekannt.

Die SySS GmbH empfiehlt, die Implementierung beziehungsweise Aktivierung der vom Webshop angeboten Schutzmaßnahmen zu prüfen und gegebenenfalls zu überarbeiten.

```
<html><body><script>history.pushState('', '', '/')</script>
<form action="https://vivitest-mi.aaronprojects.de/v1/revoke_results?auth_client=u5MLGK5Uyt
    PfSfkaouekWTCUJba7rv6caVFid4nh8ccP2CIhOeLWY4eqOqv1C9kW" method="POST" enctype="multipa
    rt/form-data">
<input type="hidden" name="election" value="60" />
<input type="submit" value="Submit request" /></form></body></html>
```

Auflistung 5.1: Cross-Site Request Forgery – PoC

Die SySS GmbH empfiehlt, zur Vermeidung von Cross-Site Request Forgery ein Anti-XSRF-Token einzuführen. Weitere Informationen zu diesem Thema können Anhang B.1 entnommen werden.

Während des Nachtests wurden im Management-Interface Bearer-Token statt Cookies zur Authentifizierung verwendet. Das Management-Interface war damit nicht mehr anfällig für Cross-Site Request Forgery. Es ist zu beachten, dass durch das veränderte Authentifizierungskonzept potenziell neue Schwachstellen entstanden sein könnten. Im Rahmen des Nachtests konnte die SySS GmbH die Wirksamkeit der neuen Authentifizierung nur rudimentär überprüfen und bei dieser Überprüfung keine neuen Schwachstellen identifizieren. Es ist essenziell, dass die eingeführten Bearer-Token bei jedem Zugriff verifiziert werden.
Im Zuge des Nachtests wurden Anfälligkeiten für Cross-Site Request Forgery auch auf der Abstimmungsseite gesucht, jedoch nicht gefunden.

5.2 Cross-Origin Resource Sharing

M2.2

HTML5 Cross-Site Origin Ressource Sharing (CORS) ist ein Standard in HTML5, der domänenübergreifende Anfragen steuert und dem Browser mitteilt, ob diese ausgeführt werden dürfen. Dafür wurden die drei folgenden neuen HTTP-Header eingeführt:

- `Origin`
- `Access-Control-Allow-Origin`
- `Access-Control-Allow-Credentials`

Der Header `Origin` wird vom Client an den Webserver übertragen und beschreibt, dass JavaScript-Code von einer bestimmten Domain auf Ressourcen des Webservers zugreifen möchte. Der Webserver überträgt nun in seiner Antwort den Header `Access-Control-Allow-Origin`, der beschreibt, welchen Domains es erlaubt ist, auf Ressourcen zuzugreifen. Stimmen diese überein, gibt der Browser die Anfrage frei und schickt sie ab. Übermittelt der Webserver zusätzlich den Header `Access-Control-Allow-Credentials` mit dem Wert `true`, kann der Browser darüber hinaus alle Cookies der Domain an die Anfrage anhängen.

Diese Funktionalität kann missbraucht werden, indem ein Angreifer einen Benutzer auf eine bösartige Seite lockt. Diese bösartige Seite enthält JavaScript-Code, der eine Anfrage gegen einen verwundbaren Webserver startet. Der Browser fügt automatisch den Header `Origin` mit der Domain der bösartigen Seite an. Der verwundbare Webserver übermittelt in seiner Antwort, dass ein Zugriff auf Ressourcen durch den bösartigen Webserver gestattet ist (durch den Header `Access-Control-Allow-Origin: *`). Dies signalisiert dem Browser, dass die durch den

JavaScript-Code der bösartigen Seite gestartete Anfrage durchgeführt werden darf. Der bösartige JavaScript-Code kann so Anfragen im Kontext des Benutzers gegen den verwundbaren Webserver durchführen und erhält zusätzlich die Antworten. Er kann sein Opfer als Proxy verwenden, solange dieser die bösartige Seite geöffnet hat und der JavaScript-Code aktiv ist. Da das Opfer als Proxy fungiert, erhält ein Angreifer so auch Zugriff auf interne Webapplikationen, die sonst nicht über das Internet erreichbar sind.

Die unter Auflistung 5.2 dargestellte, gekürzte Serverantwort zeigt, dass der Webserver einen Zugriff durch eine beliebige Domain gewährt. Die SySS GmbH empfiehlt, konsistent sicherzustellen, dass Ressourcen, bei denen ein Cross-Domain-Zugriff erlaubt ist, keine sensiblen Daten enthalten, da ein Angreifer Einsicht in diese erhalten kann. Weiterhin wird empfohlen, über einen Whitelist-basierten Mechanismus nur vertrauenswürdige Domains zuzulassen und die offenbar vorhandene Richtlinie konsistent auf allen Systemen zu forcieren. In keinem Fall sollte der Webserver jeden Wert für `Origin` akzeptieren oder `Access-Control-Allow-Origin: *` im Header zurückgeben.

```
HEAD / HTTP/1.1
Host: vivitest-app.aaronprojects.de
Origin: https://syss.de

HTTP/1.1 200 OK
Server: nginx
[<< gekürzt >>]
Strict-Transport-Security: max-age=31536000
X-Frame-Options: SAMEORIGIN
X-Content-Type-Options: nosniff
X-XSS-Protection: 1; mode=block
Access-Control-Allow-Origin: https://syss.de
Access-Control-Allow-Credentials: true
Access-Control-Allow-Methods: GET, POST, PUT, PATCH, DELETE, OPTIONS
Access-Control-Allow-Headers: Cache-Control,Content-Type,DNT,If-Modified-Since,Range,User-Age
    nt,X-Requested-With
Content-Length: 570
```

Auflistung 5.2: Cross-Origin Resource Sharing

Der Server antwortet nicht mehr mit `Access-Control-Allow`-Headern.

5.3 Sensible Informationen sind Teil der URL

L2.1

Die Webapplikation überträgt sensible Informationen als Teil der URL. Konkret wird hier der Hash zur Authentifikation im Query String an den Server übertragen. Dies stellt trotz der konsequenten Nutzung von TLS aus den nachfolgenden Gründen ein Sicherheitsrisiko dar. Unbedachte Nutzer, die beispielsweise die URL an einen Kollegen weitergeben, teilen diesem auch unbeabsichtigt ihre Authentifikationsinformationen mit. Weiter enthält der Browserverlauf diese Daten oder die URL wird durch weitere Plugins von Drittanbietern ausgewertet. Auch in Webserverlogs und bei Proxyservern können diese Informationen in bestimmten Konstellationen auftreten.

```
GET /v1/check_hash?auth_client=u5MLGK5UytPfSfkaouekWTCUJba7rv6caVFid4nh8ccP2CIhOeLWY4eqOqv1C9↩
    kW&auth_hash=0E469B096FD3FF0CE905575B387E9FB8702E6840 HTTP/1.1
Host: vivitest-app.aaronprojects.de
[<< gekürzt >>]
```

Auflistung 5.3: Sensible Informationen werden über die URL übertragen

Die SySS GmbH rät folglich, sensible Informationen ausschließlich im Body eines POST-Requests zu übertragen, um oben genannten Szenarien vorzubeugen.

Die initiale Überprüfung des Zugangscodes wird nun durch einen POST-Request durchgeführt. Auch bei der Stimmenabgabe wird der Code nunmehr als POST-Parameter übergeben. Weitere Übermittlungen des Zugangscodes oder anderer schützenswerter Daten wurden im Laufe des Nachtests nicht ausfindig gemacht.

5.4 Öffentliche Umfragedaten

A2.1

Beim ersten Aufruf der Bürgerseite wird momentan nur die Beschreibung der aktuellen Umfrage angezeigt. Erst nach Eingabe eines Zugangscodes gelangen die Bürger dann zur Übersicht, in der sie weitere Umfragen und freigeschaltete Ergebnisse einsehen können. Mit Kenntnis der jeweiligen Adressen lassen sich sowohl die Beschreibungen als auch die Ergebnisse von freigeschalteten Umfragen auch ohne Authentifizierung abrufen – beispielsweise unter `https://vivitest-app.aaronprojects.de/v1/elections?auth_client=u5MLGK5UytPfSfkaouekWTCUJba7rv6caVFid4nh8ccP2CIhOeLWY4eqOqv1C9kW&lang=de` und `https://vivitest-app.aaronprojects.de/v1/results?auth_client=u5MLGK5UytPfSfkaouekWTCUJba7rv6caVFid4nh8ccP2CIhOeLWY4eqOqv1C9kW&election=58`. Der hier aufgeführte `auth_client`-Wert kann auch von nicht authentifizierten Nutzern ermittelt werden, da er bereits vor der Anmeldung an den Browser übermittelt wird. Laut Verena Zaiser von Aaron Projects ist dieses Verhalten erwünscht bzw. sind Beschreibungen und Ergebnisse als öffentlich zu bewerten. Diese Feststellung wird daher lediglich als Anomalie aufgenommen. Nichtsdestotrotz wird empfohlen, dieses Verhalten zu prüfen, zu diskutieren und ggf. anzupassen.

6 Analyse der mobilen Applikationen

Dieses Kapitel beschäftigt sich mit der Analyse der mobilen Applikation Vivi für iOS und Android. Alle in diesem Kapitel erwähnten Sicherheitslücken gelten für iOS und Android gleichermaßen, sofern nicht explizit einer Plattform zugeschrieben. Die mobile Applikation erlaubt Bürgern der Stadt Tübingen an Abstimmungen der Stadt teilzunehmen. In Abbildung 6.1 sind einige Screenshots der Applikation abgebildet:

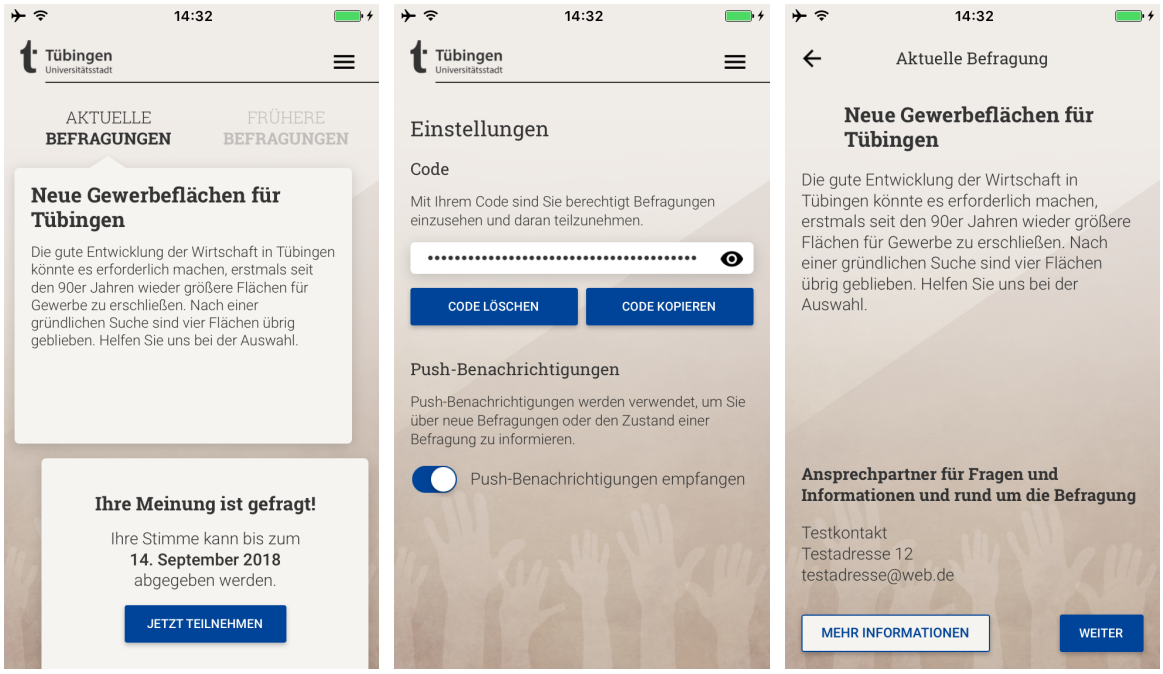


Abbildung 6.1: Screenshots der Applikation Vivi

Die SySS GmbH führte die Analyse der mobilen Applikation Vivi auf den in Tabelle 6.1 aufgeführten Testgeräten aus. Die Geräte wurden vor ihrer Verwendung mit einem Jailbreak (iOS) oder Rooting (Android) modifiziert, um eine Analyse aller auf dem Gerät gespeicherten Daten durchführen zu können.

Gerät	Betriebssystem	Bemerkung
Apple iPhone SE	iOS 10.2	Jailbreak vorhanden
Google Nexus 5x	Android 8.1.0	Rooting vorhanden

Tabelle 6.1: Verwendete Testgeräte

Für den Test wurden der SySS GmbH mehrere QR-Codes zur Verfügung gestellt, die den Hash-Wert für die Legitimation der Abstimmung enthalten. In Abbildung 6.2 auf der nächsten Seite ist exemplarisch ein Hash abgebildet.

Abbildung 6.2: Hash-Wert für die Authentifizierung

Die nachfolgenden Abschnitte beschreiben Sicherheitsrisiken, die innerhalb der Applikation identifiziert werden konnten.

Für den Nachtest wurden ein iPhone 6s mit iOS 10.1.1, ein iPad mit iOS 11.3.1 sowie ein Android-Gerät der Marke OnePlus One mit LineageOS und Android-Version 7.1.2 eingesetzt. Alle Geräte waren mit einem Jailbreak bzw. Rooting freigeschaltet.

6.1 Testaufbau

Um die bidirektionale Datenkommunikation zwischen der mobilen Applikation auf den Testgeräten und den entsprechenden Webservices analysieren zu können, betrieb die SySS GmbH die Testgeräte in einem eigenen 802.11-basierten Funknetzwerk. Auf diese Weise konnte sämtlicher Datenverkehr über ein entsprechendes System der SySS GmbH umgeleitet werden, was sich bei Nutzung eines Mobilfunknetzes mittels GSM beziehungsweise UMTS weitaus schwieriger gestaltet.

6.2 Jailbreak/Rooting Detection

L3.1

Die SySS GmbH konnte während des Testzeitraums die Applikation auf einem Smartphone verwenden, das mithilfe eines Jailbreak (iOS) oder Rooting (Android) modifiziert wurde. Mit administrativem Zugriff auf ein Gerät ist es zudem möglich, auf die lokal gespeicherten Daten der Anwendung (vgl. Kapitel 6.4) zuzugreifen und diese zu analysieren.

Die SySS GmbH empfiehlt, bei der Ausführung der Anwendung zu überprüfen, ob es sich um ein entsprechend modifiziertes Gerät handelt. In diesem Fall sollte dies dem Benutzer kenntlich gemacht werden, um auf die Gefahren eines durch einen Jailbreak modifizierten Gerätes hinzuweisen. Es ist jedoch anzumerken, dass entsprechende clientseitige Schutzmaßnahmen für eine derartige Erkennung auf einem Gerät mit administrativem Zugriff durch den Angreifer immer umgangen werden können und somit keinen hundertprozentigen Schutz bieten. Die Hürde für einen Angreifer bezüglich der Laufzeitanalyse einer Anwendung wird durch solche Schutzmaßnahmen jedoch erhöht.

Auf dem im Nachtest verwendeten iOS-Geräten (iOS 11.3.1 und iOS 10.1.1) wurde keine Jailbreak Detection festgestellt. Die App konnte trotz vorhandenem Jailbreak weiter verwendet werden, ohne dass eine Warnung erschien.

Hingegen wurde in der Android-Version die in Abbildung 6.3 dargestellte Nachricht angezeigt. Die App ließ sich jedoch weiterhin nutzen. Die SySS GmbH konnte auf einem weiteren Gerät mit einer eigenen Rooting-Variante die Rooting Detection umgehen. Trotzdem wird die vorhandene Rooting Detection auf Android als ausreichend bewertet.

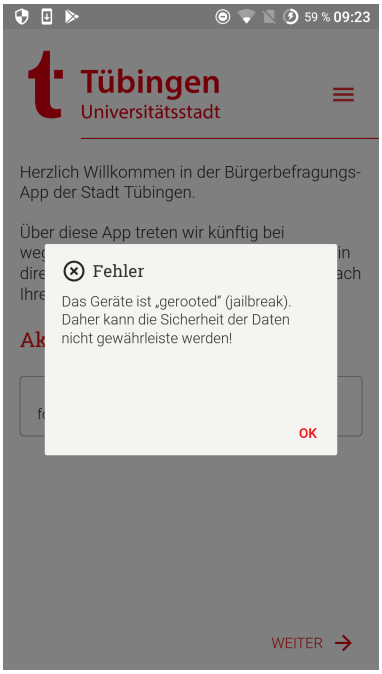


Abbildung 6.3: Benachrichtigung über festgestelltes Rooting

6.3 Fehlendes Certificate Pinning

L3.2

HTTP-Anfragen gegen das Back-End wurden durch den SSL/TLS-fähigen Proxy Burp Suite Professional abgefangen und im Anschluss weitergeleitet.

Ein entsprechendes Stammzertifikat des Webproxy, ohne das die Applikation einen SSL-Verbindungsaufbau nicht zuließ, wurde auf dem Testgerät über einen internen Webserver installiert. Ein Rooting oder Jailbreak des Endgerätes ist für die Installation des Stammzertifikats nicht notwendig. Anschließend konnte die Kommunikation zwischen der Applikation und dem Back-End im Klartext eingesehen und manipuliert werden.

Da erst hierdurch netzwerkbasierte Angriffe gegen die geprüfte Anwendung möglich werden, wird empfohlen zu prüfen, ob eine zusätzliche Gültigkeitsprüfung des serverseitigen SSL/TLS-Zertifikats auch in der Anwendung selbst erfolgen kann, ohne hierfür auf Betriebssystemfunktionen zurückzugreifen. Dabei wird beispielsweise das Serverzertifikat oder dessen Fingerprint mit der App ausgeliefert (Certificate Pinning) und beim Kommunikationsaufbau entsprechend validiert.

Hintergrund dieser Empfehlung ist nicht zuletzt die Erfahrung der SySS GmbH, dass innerhalb geschlossener Systeme die Möglichkeit nicht spezifikationsgetreuer Anfragen oftmals nur ungenügend berücksichtigt wird und sich hierauf basierende Angriffsvektoren ergeben können.

Während des Nachtests konnte festgestellt werden, dass auf beiden Plattformen ein Certificate Pinning umgesetzt wurde.

6.4 Datenspeicherung

Die beiden folgenden Abschnitte beschreiben die Speicherung des Hash auf dem Gerät.

6.4.1 Datenspeicherung unter Android

Unter der Plattform Android wird der Hash in der Datenbank `/data/data/com.vivi/databases/RKStorage` gespeichert, wie in Abbildung 6.4 auf der nächsten Seite zu sehen ist.

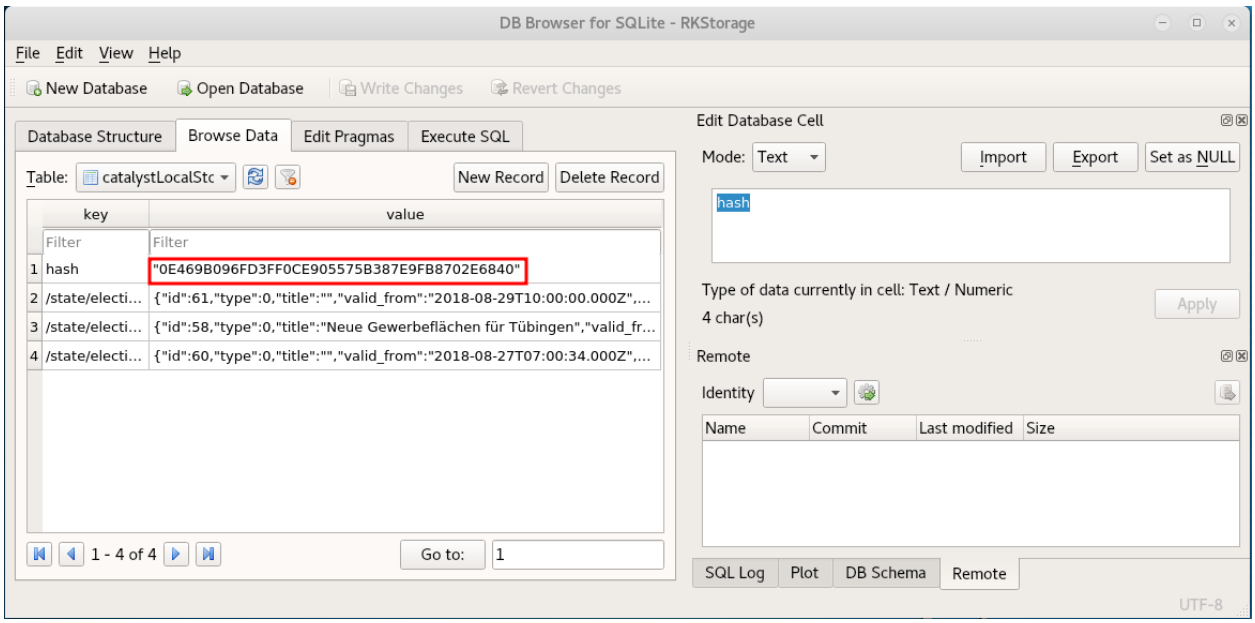


Abbildung 6.4: Der Benutzer-Hash wird in der Datenbank `RKStorage` im Programmverzeichnis der App gespeichert

Die SySS GmbH empfiehlt, diesen Authentifizierungsstring nur verschlüsselt abzulegen und es durch Obfuskierung des Programmcodes möglichst zu erschweren, den Klartext wiederherzustellen. Dies bietet zwar keinen hundertprozentigen Schutz, erschwert jedoch Angriffe stark.

6.4.2 Datenspeicherung unter iOS

Unter der Plattform iOS wird der Hash in der Datei `/var/mobile/Containers/Data/Application/<ID>/Documents/RCTAsyncLocalStorage_V1/manifest.json` gespeichert. Der folgende Auszug zeigt dieses Verhalten.

```
{
"/state/election/0/58": null,
"pushNotifications.usePushNotifications": "true",
"hash": "\"0E469B096FD3FF0CE905575B387E9FB8702E6840\"",
"/state/election/0/60": "{\"id\":60,\"type\":0,\"title\":\"\",\"valid_from\":\"2018-08-27T
    07:00:34.000Z\",\"valid_to\":\"2018-08-27T08:00:37.000Z\",\"short_description\":\"fooba
    r\",\"long_description\":\"<p>foobar</p>\",\"options\":[{\"id\":108,\"position\":1,\"val
    ues\":[\"1\"],\"title\":\"\",\"short_description\":\"foobar\",\"long_description
    \":\"\",\"values_description\":[\"\"],\"rating_text\":\"\"}],\"contact_email\":\"vivites
    t@m4.sy.gs\",\"contact_address\":\"test\",\"contact_name\":\"test\"}",
"pushNotifications.subscriptionARN": "\"arn:aws:sns:eu-central-1:506840112337:vivi_syss:389
    bbaa7-d224-4d7a-a3f6-70f139f29ac0\"",
"pushNotifications.applicationEndpointARN": "\"arn:aws:sns:eu-central-1:506840112337:endpoin
    t/APNS_SANDBOX/vivi_dev/bb21ed98-ec54-36d9-b46b-914e6ad5bfd1\""
}
```

Die SySS GmbH empfiehlt, diese sensiblen Daten verschlüsselt in der Keychain abzulegen, um zu gewährleisten, dass auch bei einem gestohlenen Gerät die Daten nicht im Klartext ausgelesen werden können.

A Hinweise zur Durchführung von Penetrationstests

Die SySS GmbH orientiert sich bei der Durchführung von Sicherheitsüberprüfungen an der Studie „Durchführungskonzept für Penetrationstests“[1] des Bundesamtes für Sicherheit in der Informationstechnik (BSI).

Sowohl der genaue Ablauf eines Tests als auch die Auswahl der Werkzeuge liegt in der Verantwortung des durchführenden Consultants. Dieser passt auf der Basis seiner Erfahrung sowohl den Ablauf an den Testgegenstand und insbesondere an die Testtiefe an und wählt die optimalen Werkzeuge aus.

Die Entscheidung, welche Werkzeuge genau eingesetzt werden, basiert zum einen auf dem zu erwartenden Erkenntnisgewinn und zum anderen auf der Testtiefe. Nicht jedes Werkzeug ist für jede Software einsatztauglich. Der Einsatz eines jeden Werkzeuges hat eine bestimmte Mindestlaufzeit. Überschreitet diese den geplanten Testzeitraum jedoch erheblich, muss auf den Einsatz verzichtet werden.

Als automatisierter Schwachstellenscanner kann zum einen das in Westeuropa recht unbekannte Produkt MaxPatrol des russischen Herstellers Positive Technologies zum Einsatz kommen. Zum anderen finden die Schwachstellenscanner Nessus des Herstellers Tenable und SAINT des Herstellers SAINT Corporation regelmäßig Verwendung. Bei Webapplikationen wird darüber hinaus auf die Dienste der Burp Suite Professional und weiterer HTTP-Proxy-Lösungen zurückgegriffen. Die Ergebnisse der Schwachstellenscanner werden bei einer anschließenden Analyse ausgewertet und verifiziert.

Bei internen Tests kommen in der Regel umfangreiche Exploit-Frameworks wie Metasploit, CORE IMPACT oder Immunity Canvas zum Einsatz. Während der Tests wird zudem eine Vielzahl weiterer, teils selbst geschriebener Werkzeuge eingesetzt.

Das Herzstück eines jeden Penetrationstests der SySS GmbH bilden jedoch die manuellen Tests. Bei der Überprüfung einer Applikation, eines Systems oder eines ganzen Netzes auf Schwachstellen hin profitieren die Mitarbeiter der SySS GmbH von ihrem großen Erfahrungsschatz. Zudem lernen sie durch interne Weiterbildungsmaßnahmen stets die neuesten Angriffstechniken und setzen diese dann auch ein.

Außerdem werden auch eigene kreative Techniken und Vorgehensweisen entwickelt, um die Überwindung bestehender Sicherheitsvorkehrungen oder die Möglichkeit zum Missbrauch einer technischen Schwäche zu demonstrieren und dem Kunden damit das Gefahrenpotenzial eines identifizierten Risikos zu verdeutlichen.

1 https://www.bsi.bund.de/DE/Publikationen/Studien/Pentest/index_htm.html

B Ergänzende Erklärungen zu Webapplikationen

Im Folgenden werden für Webapplikationen typische Sicherheitslücken sowie das für das Verständnis notwendige Basiswissen aufgeführt.

B.1 Cross-Site Request Forgery

Bei einem Angriff durch Cross-Site Request Forgery (kurz XSRF oder auch CSRF) bringt ein Angreifer sein Opfer dazu, unbemerkt eine vorbereitete HTTP-Anfrage an eine Webanwendung zu senden. Diese Anfrage bewirkt, dass dort eine vom Angreifer bestimmte Aktion im Kontext des Opfers aufgerufen wird. Der Angreifer nutzt dabei die Tatsache, dass einer Webanwendung ein vorhersagbares URL-Schema zugrunde liegt und dass den Anfragen eines authentifizierten Benutzers von der Webanwendung vertraut wird.

Ein einfaches Beispiel für Cross-Site Request Forgery bietet dieser Link auf die Abmelden-Funktion einer Webanwendung:

```
http://www.example.com/user.php?action=logout
```

Wird diese URL vom Webbrowser eines angemeldeten Benutzers aufgerufen, so wird der Benutzer ohne eigenes Zutun von der Anwendung abgemeldet.

Einem Angreifer bieten sich verschiedene Angriffsvektoren, um das Opfer zum Aufruf der URL zu bewegen:

- Unterschieben der URL: Der Angreifer übermittelt dem Opfer die URL per E-Mail oder legt sie als direkten Link auf einer vom Opfer frequentierten Webseite ab, die nicht unbedingt Teil der eigentlichen Webanwendung sein muss. Diese Methode nutzt die Gutgläubigkeit des Opfers aus. Die URL kann vom Angreifer weiter verschleiert werden (etwa durch einen Kurz-URL-Dienst[2]), genauso wie die E-Mail an das Opfer gefälscht werden kann.
- Persistent Cross-Site Scripting: Zunächst wird die manipulierte URL vom Angreifer in HTML-Code verpackt und dann an die Webanwendung übermittelt. Diese speichert den Code und liefert ihn bei späteren Anfragen von Benutzern mit aus. Die manipulierte Anfrage bettet der Angreifer etwa in einen `img`-Tag ein. Ein `img`-Tag weist den Webbrowser an, eine Bilddatei nachzuladen, was normalerweise automatisch geschieht. Sobald der Webbrowser des Opfers nun den präparierten Tag auswertet, wird allerdings keine Bilddatei nachgeladen, sondern die Anfrage des Angreifers abgesetzt. Die Webanwendung verarbeitet die Anfrage nun so, als wäre er vom Opfer autorisiert. Das folgende HTML-Code-Fragment zeigt, wie ein Link auf eine Funktion der Benutzerverwaltung die Rechte eines Benutzers, der vom Angreifer kontrolliert wird, erweitern kann:

  ```
  <img src="http://example.com/admin/setrole.php?user=hacker&newrole=admin">
  ```

2 Siehe hierzu auch `http://de.wikipedia.org/wiki/Kurz-URL-Dienst`

Diesen Link postet der Angreifer zum Beispiel in einem Forum oder einem Bewerbermanagement-System. Anstelle des `img`-Tags kann der Angriff auch über ein verstecktes `iframe`-Element oder durch eingebetteten JavaScript-Code geschehen.

Alle diese Methoden setzen voraus, dass sich das Opfer bereits bei der betroffenen Webanwendung angemeldet hat, seine Zugangsdaten in einem Cookie gespeichert werden beziehungsweise eine Session existiert, oder dass das Opfer der Aufforderung nachkommt, sich gegenüber der Webanwendung zu authentisieren.

B.1.1 Absicherung

Einige Maßnahmen zur Unterbindung von Cross-Site Request Forgery-Angriffen werden zwar empfohlen, bieten allerdings keinen hinreichenden Schutz. Durch sie wird die Hürde für den Angreifer jedoch höher gelegt.

Einfache, aber unzulängliche Gegenmaßnahmen:

- Nur HTTP-POST-Anfragen für ändernde Aktionen akzeptieren: Eine POST-Anfrage kann einfach abgesetzt werden, muss aber erst von JavaScript-Code oder dem bewussten Klick des Opfers erzeugt werden.
- Prüfung des HTTP-Referrers: Gefälschte Anfragen, die mittels Täuschung des Benutzers auf einer externen Webseite ausgelöst wurden, können am HTTP-Referrer erkannt und gefiltert werden. Allerdings sollte sich die Webanwendung nicht darauf als alleinige Schutzmaßnahme verlassen. Der Referrer kann beispielsweise durch Browser-Plugins beliebig verändert werden. Zudem kann ein HTTP-Referrer über Flash oder XHR[3] ebenfalls manipuliert werden.

Wird die Benutzerauthentisierung durch ein Cookie angezeigt, so sollte dieses nur eine begrenzte Lebenszeit haben, nach der ein Benutzer wieder ausgeloggt wird. Viele Webanwendungen bieten ihren Nutzern die Möglichkeit, dauerhaft angemeldet zu bleiben. Diese Komfortfunktion vergrößert aber auch die Angriffsfläche.

Den größten Schutz vor Cross-Site Request Forgery stellen sogenannte Anti-XSRF-Tokens dar. Jede Transaktion der Webapplikation wird dabei mit einem geheimen Token abgesichert, welches danach seine Gültigkeit verliert und neu generiert wird.

Das Token ist in einem Hidden Field auf der Seite eingebunden oder wird als Parameter an die URL des Aufrufs angehängt. Wichtig ist, dass dieses Token für jeden Benutzer einzigartig und verschieden von seiner Session-ID ist.

Die nächste Anfrage des Benutzers muss nun das aktuell gültige Token enthalten. Serverseitig ist dies zu überprüfen. Sollte kein oder ein falsches Token übermittelt werden, so ist davon auszugehen, dass ein Angriff auf eine bestimmte Sitzung aktuell stattfindet.

Bei einem Zustandswechsel der Webanwendung wird für den Benutzer wieder ein neues Token generiert.

B.1.2 Einschränkung des Bedienkomforts

Durch Anti-XSRF-Tokens leidet aber möglicherweise die Nutzbarkeit der Anwendung. Beispielsweise kann die „Zurück“-Schaltfläche des Browsers nicht mehr genutzt werden, um gemachte

3 Siehe hierzu auch `http://en.wikipedia.org/wiki/XMLHttpRequest`

Angaben in vorherigen Schritten zu ändern. Dabei werden Adressen mit abgelaufenen Tokens erneut aufgerufen, deren Prüfung dann fehlschlägt. Mit modernen Programmiertechniken kann diese Einschränkung aber umgangen werden.