Individual Study Report

Dynamic Analysis of Medical Apps

Kanika Gupta, Nikhil Chopra

04/19/2017

➢ Problem Statement

In this study, we have done Dynamic analysis of Medical apps to find security vulnerabilities in them. We have concentrated on SSL implementation vulnerabilities which include Certificate Validation, Hostname Verification and Certificate Pinning.

➢ App Selection

We have analysed almost all the Medical Apps which include some sort of Doctor Patient Communication or personal health record storage or have more than a thousand users, for both Android and iOS. Total Apps analysed are 211 for Android and 195 for iOS.

➢ Tool Used

We have used Telerik Fiddler for the Security Testing of more than 400 apps, both Android and iOS apps combined. It has following features –

- Fiddler is a free web debugging proxy which logs all HTTP(s) traffic between computer and the Internet.
- It can decrypt HTTPS traffic, and display and modify requests using a man-in-the-middle decryption technique. Fiddler can be configured to decrypt all traffic, or only specific sessions.

- Fiddler script is customizable due to which we could customize the hostname in the server certificate provided by fiddler to the app.

- Fiddler can decrypt HTTPS packets, if Fiddler's root CA is installed as Trusted CA in the phone, by providing a fake certificate to the app with a similar hostname as expected by the app (which is collected by Fiddler using CONNECT request) on the fly.

➢ Analysis Technique

We have mainly analysed three basic validations in an app – Server Certificate validation, Hostname verification and Certificate Pinning.

- **Server Certificate Validation** is done by the client (app) when the server presents its certificate to the client for authentication. The client validates if the certificate is signed by an authorized CA which happens to be the root CA in the client's device.

- **Hostname Verification** is also done at the client side when App checks if certificate presented by the server is having server's name in subjectAltName field of type dnsName. *Common Name is deprecated.*
- If subjectAltName is empty, in that case Common Name is considered for verification.

- **Certificate Pinning** or Public Key Pinning Extension allows the admin of a server to "pin" a certificate authority's (CA) public key signature to a certificate, which is verified by the client (delivered via SSL extension). So, in Certificate Pinning you ignore the signature hierarchy, and say trust *this certificate only* or perhaps trust only certificates *signed by this certificate*.

For dynamic analysis, an app will fall in one of these 4 categories -

1. An app is validating both certificate chain (certificate is signed by Trusted Root CA) and hostname.
2. App is validating only certificate chain but not hostname.
3. App is validating only hostname but not certificate chain.
4. App is not validating either of them.

1. Check Certificate Validation (Given correct host name) - In this part we want to check if the app is checking the chain of trust for the certificate presented to it. So, we will be giving the app a certificate with nearly valid host name (which is collected by Fiddler using CONNECT request) and Fiddler's root certificate will NOT be added as trusted in the phone. Therefore, we can check if the app is validating the chain of trust when it is given the certificate with correct hostname which is Not signed by a trusted CA.

2. Check Hostname Verification (Given Trusted Certificate) - In this part we want to check if app is matching subjectAltName (or Common Name) of certificate with that of the server's name. So, we will be making Fiddler's certificate a Trusted one in the phone and provide any random hostname to the certificate which is signed by the trusted

Fiddler's CA and present it to the app. Therefore, we can check if the app is verifying host name also or it is just stopping all the validation after checking the chain of trusted CA's.

Real World Scenario *- An adversary can give any certificate (abc.com) to the app, signed by VeriSign (or any worldwide trusted authority) and perform MiTM.*

3. If an app fails for both of above scenarios, then we will check Category -4 i.e. app is not validating either of them. So, we will give any random hostname certificate to the app signed by our own CA which is not even in the list of trusted CAs in the phone and check whether the app allows us to Log in.

Note: In this study, we have checked the vulnerability of an app, in all these above-mentioned scenarios, against the Login functionality. For high profile and vulnerable apps, we have performed these experiments for almost all the functionalities.

➢ Methodology

1. Certificate Validation Analysis *[correct Hostname, wrong Root CA]* –

- **No Fiddler root** is installed as a trusted root CA in the phone.
- Fiddler will be sending the **fake server certificate with Correct Hostname** i.e. hostname of the actual server.
- Check if we can log in the app.
- If we can Login and HTTPS packets are decrypted revealing the username and password, then the app is Not checking the chain of trust for the certificate presented to it by the server.
- If we cannot Login, then the app is doing certificate validation correctly.

.

2. Hostname verification Analysis *[correct CA, wrong hostname]* –

- **Fiddler root is installed** as a trusted root CA in the phone
- Fiddler will be sending the **server certificate with Fake Hostname** such as some.fake.thing in the subjectAltName of the server certificate.
- Check if we can log in the app.
- If we can Login and HTTPS packets are decrypted revealing the username and password, then the app is just stopping all the verification by checking if the certificate is signed by a trusted CA. It is Not checking the actual certificate contents.
- If we cannot Login, then the app is checking certificate validation and hostname verification both.

3. Certificate Pinning Analysis *[correct CA, correct hostname (BOTH FAKE in a way)]* –

- **Fiddler root is installed** as a trusted root CA in the phone
- Fiddler will be sending the **server certificate with correct Hostname**
- We want to check if we can give illegitimate certificate [signed by self-signed proxy] generated by an illegitimate server to the app and whether app accepts it or not.
- If we can Login and HTTPS packets are decrypted revealing the username and password, then the app can accept any certificate which is signed by a trusted CA and has the server's hostname even if it is not from a legitimate source (like Fiddler).
- If we cannot Login, then the app will be accepting a certificate whose public key is hard coded in the app so that nobody can present a fake certificate to the app.

➢ Results

Android Apps Analysis –

- Total Android Apps Analysed – 211
- Certificate Validation Broken – 7
- Certification Validation Not Broken – 115
- Host name Verification Broken – 10
- Host name Verification Not Broken – 112
- Certificate Pinning – 13
- No Certificate Pinning – 109
- No HTTPS/No Login – 14
- All Information Remains on the Phone, No internet Usage – 74

iOS Apps Analysis –

- Total iOS Apps Analysed – 195
- Certificate Validation Broken – 7
- Certification Validation Not Broken – 111
- Host name Verification Broken – 12
- Host name Verification Not Broken – 106
- Certificate Pinning – 36
- No Certificate Pinning – 82
- No HTTPS – 2
- All Information Remains on the Phone, No internet Usage – 74

Results for each app can be found at this shared location –

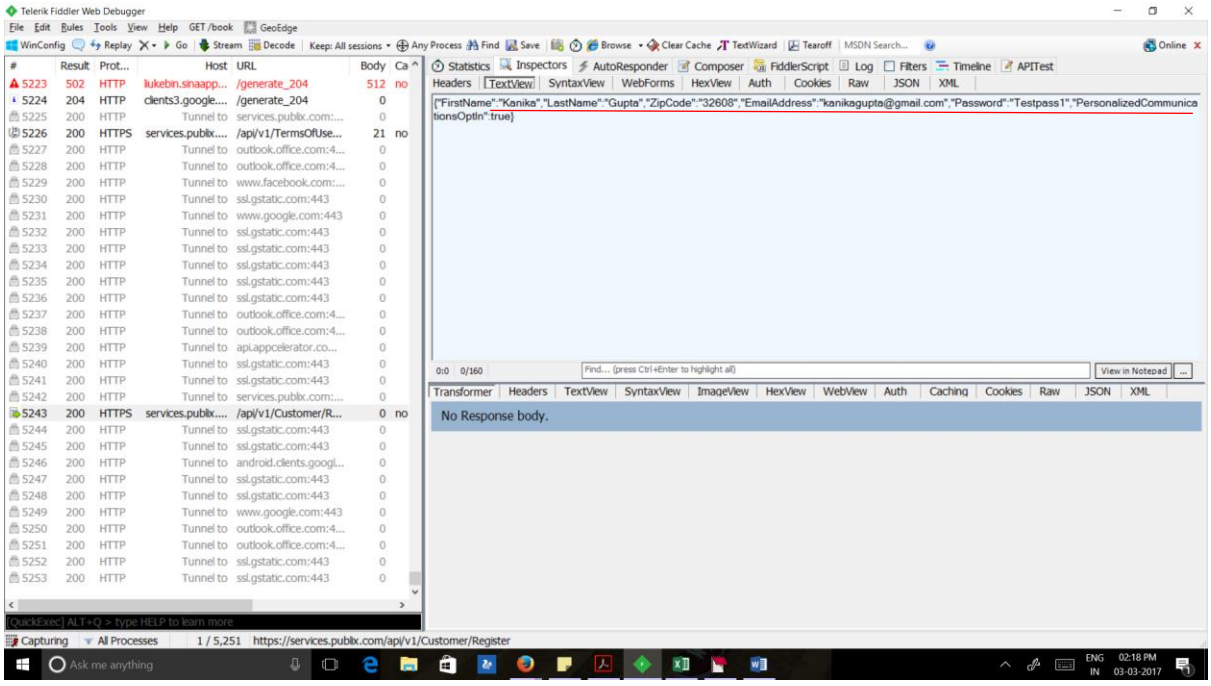
https://docs.google.com/spreadsheets/d/1TycD_2uh4MQeJ4hPjQmcDrte_Y9BQRMM8tAYJ7e2J4I/edit?usp=sharing

Appendix

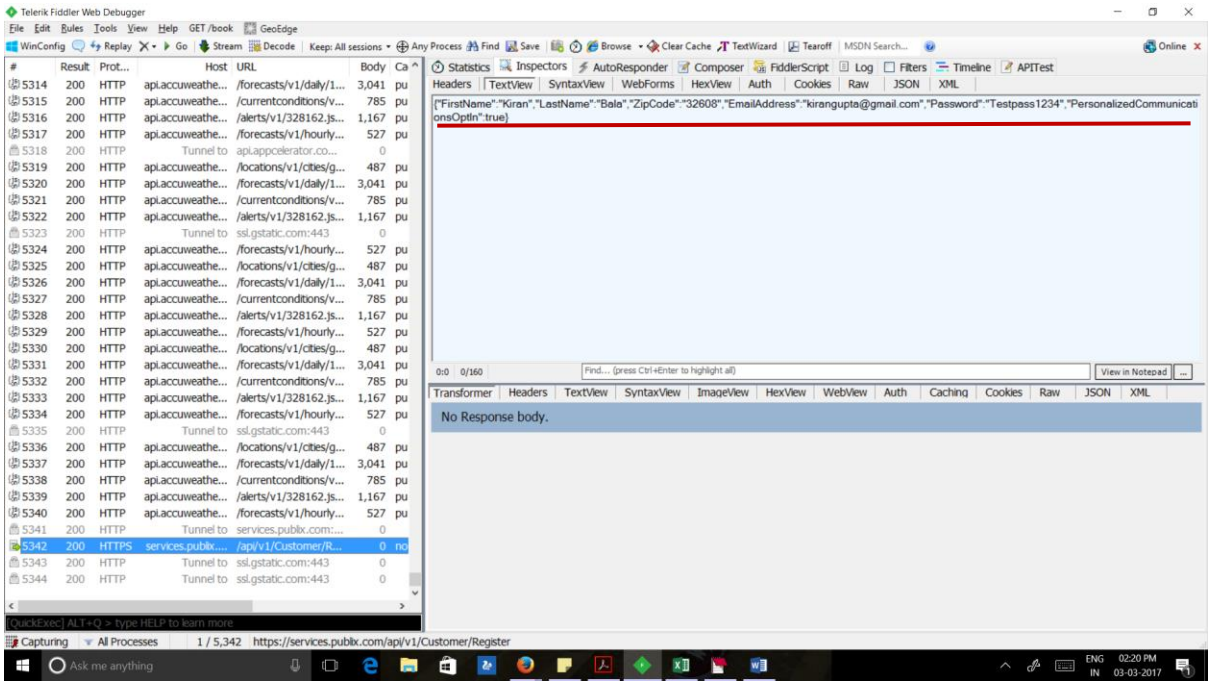
Detailed Analysis and screenshots of some apps are included below –

1. Publix App (both iOS and Android)

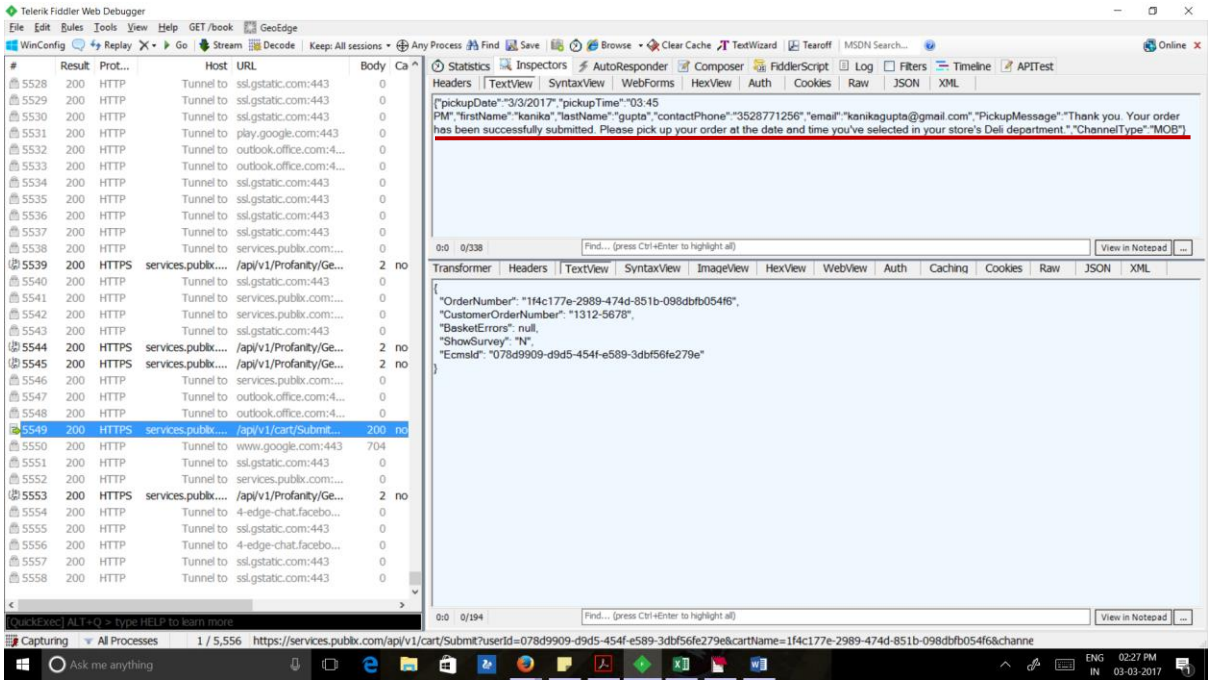
No Hostname verification – With trusted root CA installed and some.fake.thing hostname , the screenshot below shows all the user information including username and password when a user sign up for the app.



None of hostname and certificate chain verification - While signup, no trusted CA is installed and bad hostname is given, then also all the user information is visible.

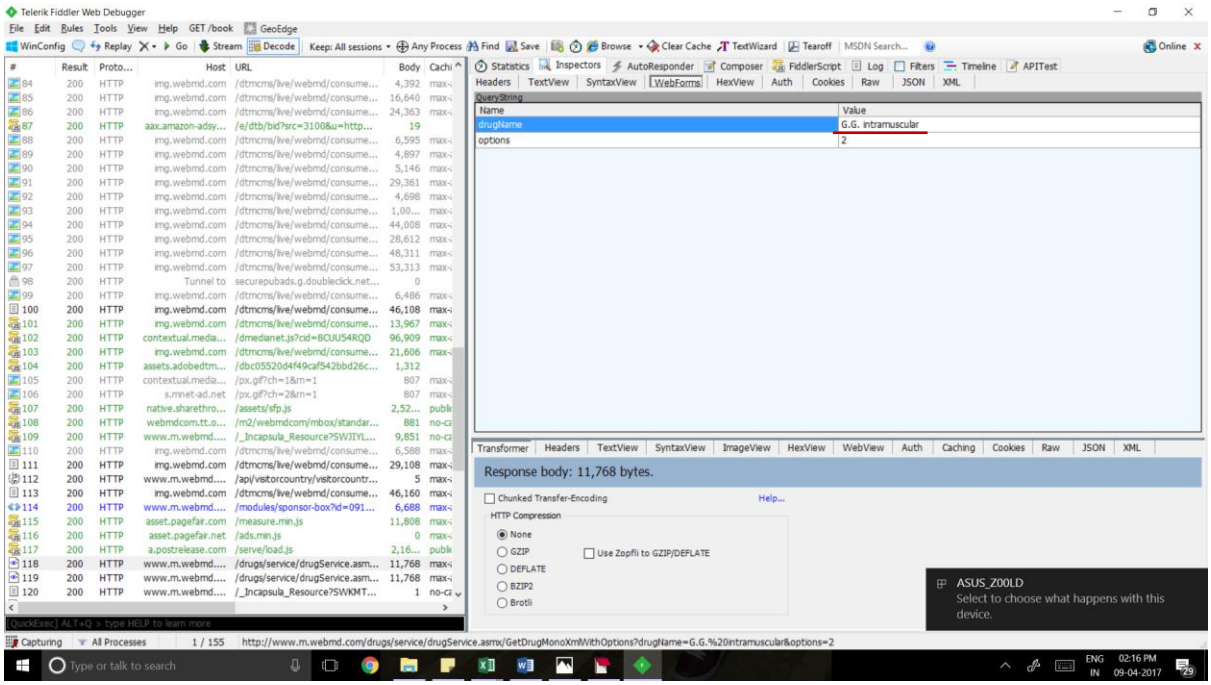


An order placed on Publix App with some.fake.thing hostname and no trusted root CA.

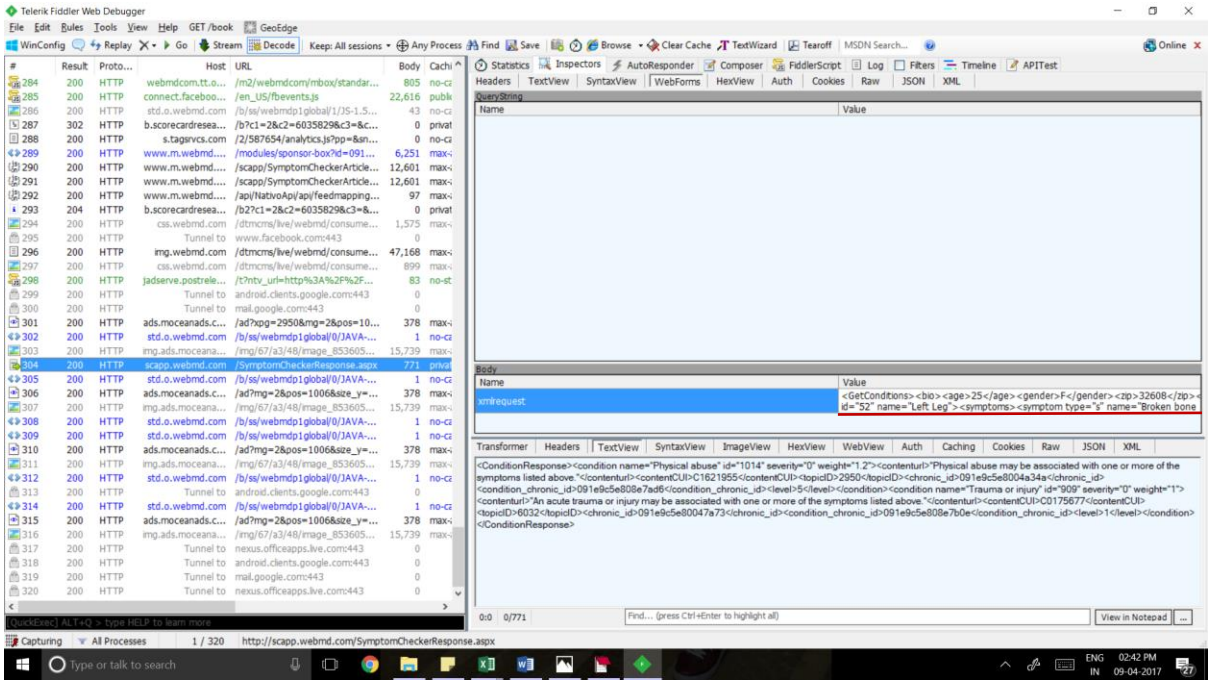


2. WebMD App (Android)

Bad Hostname Verification for drug searches - With trusted root CA installed and some.fake.thing hostname, the drug searched by the user is visible in clear text.

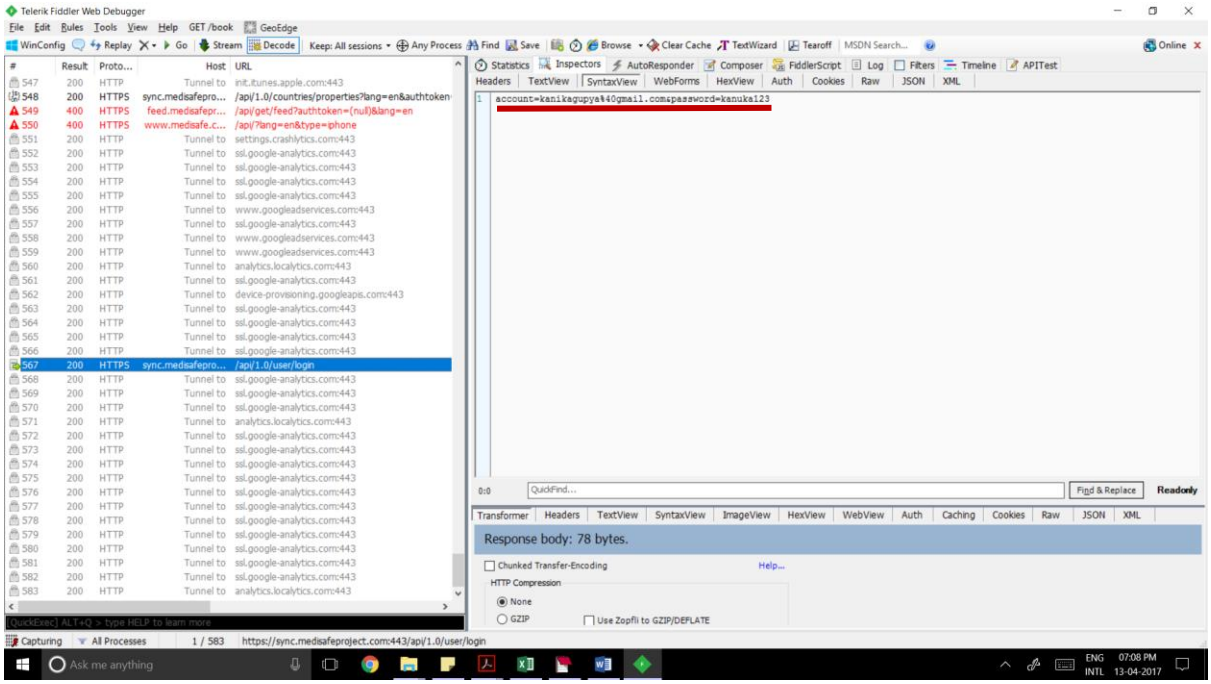


While checking symptoms in SYMPTOM CHECKER, all info is going in plain text.



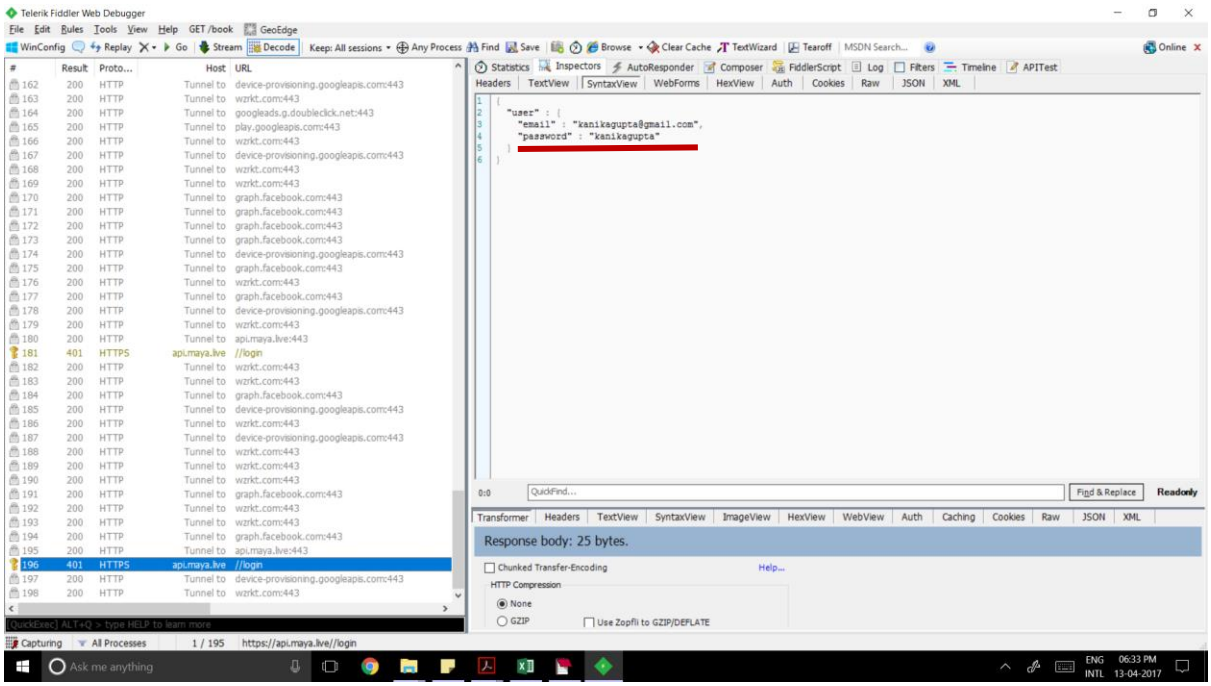
3. Medisafe App (iOS)

Hostname Verification broken – With trusted root CA and some.fake.thing hostname, username and password is visible in Medisafe App while Login.



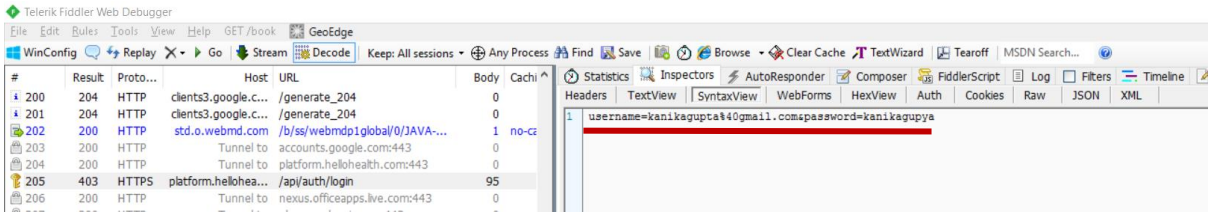
4. Maya period tracker App (iOS)

Certificate validation and Hostname Verification broken - By giving any certificate signed by any CA with some.fake.thing hostname, email and password is visible in clear text.

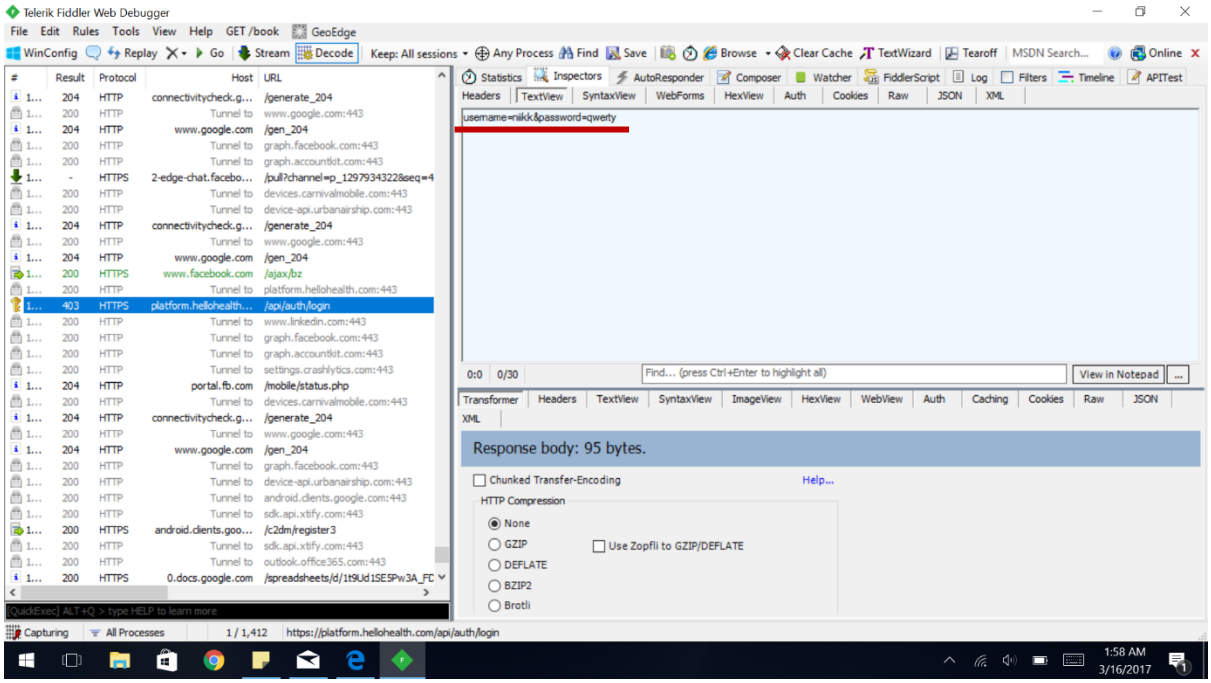


5. PortalConnect App (Android)

Bad Hostname Verification - By adding Fiddler's CA as a trusted CA in the phone and giving the certificate with some.fake.thing hostname, email and password is visible in clear text while Login.

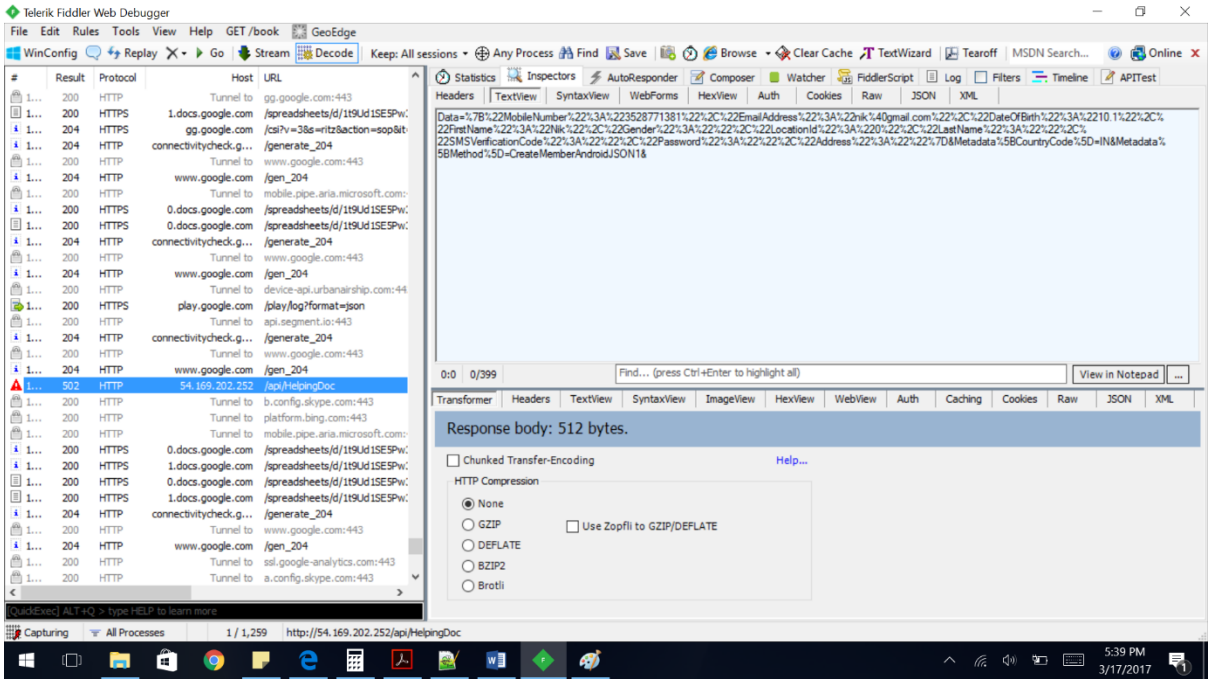


Certificate Validation Broken – By giving correct hostname without adding Fiddler's CA as a trusted root CA, email and password are visible in plain text while Login. So, any certificate with any hostname can be used for MiTM in this App.



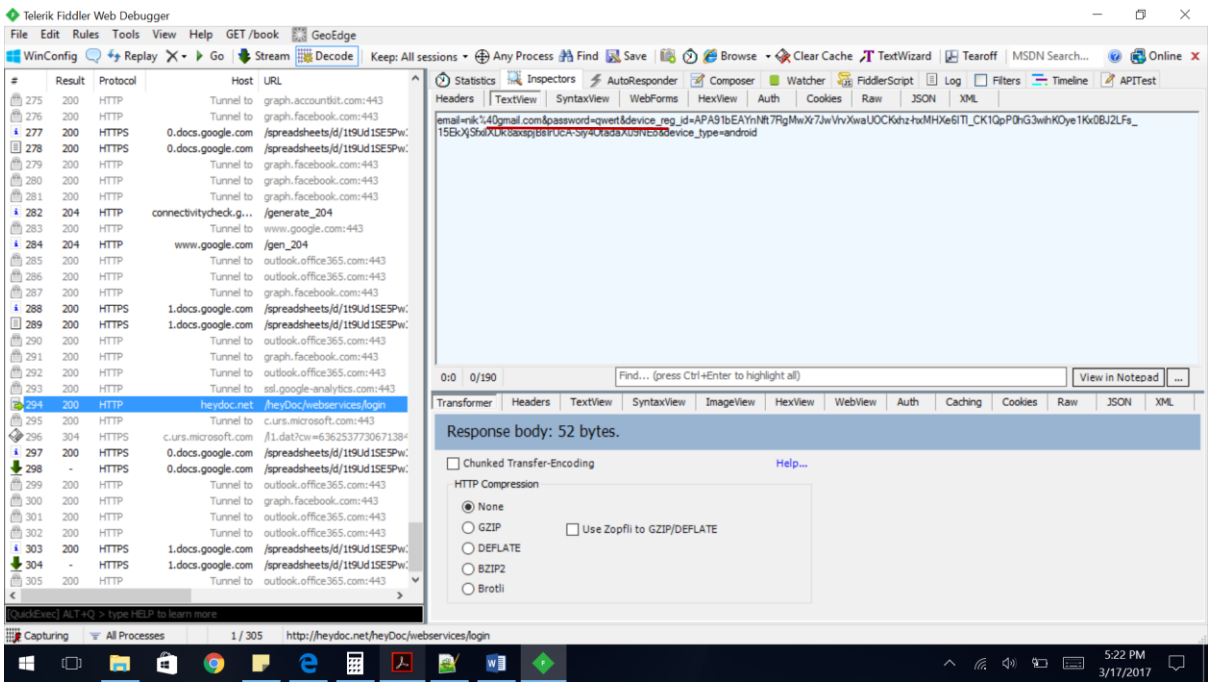
6. HelpingDoc App (Android)

Certificate Validation Broken – Root Chain of Trust is not validated by this app. By giving correct hostname without adding Fiddler's CA as a trusted root CA, email and password are visible in plain text while Login.



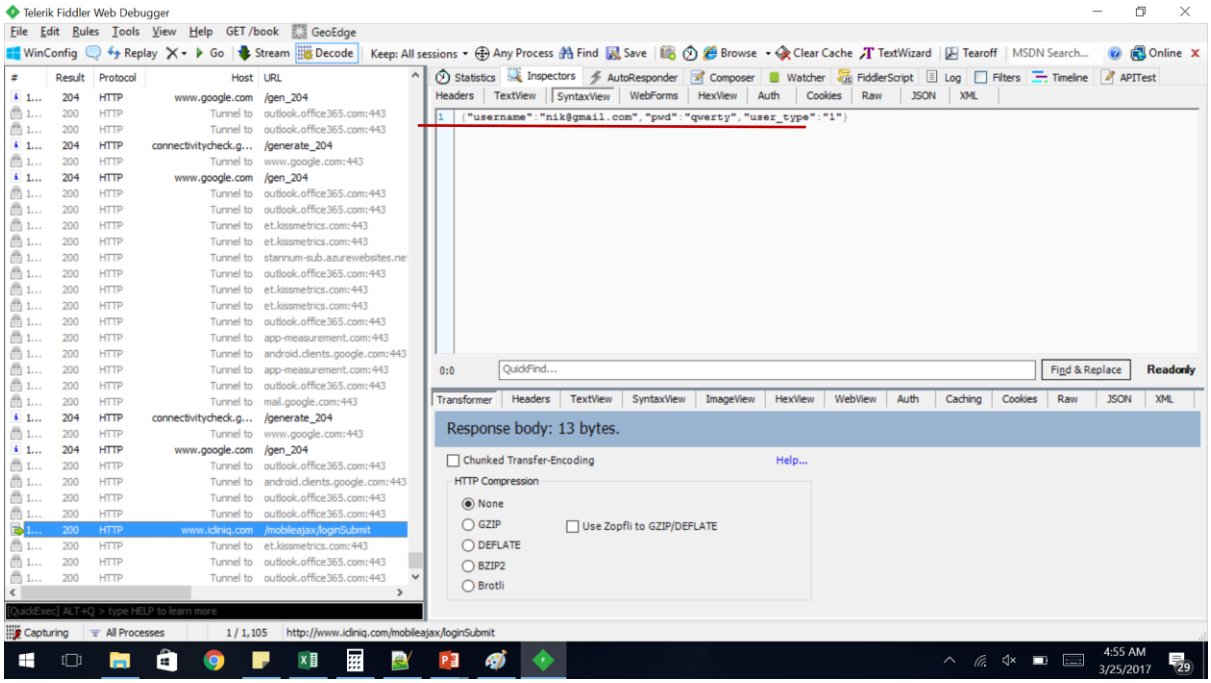
7. HeyDoc App (Android)

No HTTPS – In this app, there is no HTTPS connection while Login, so email and password are visible in clear text.



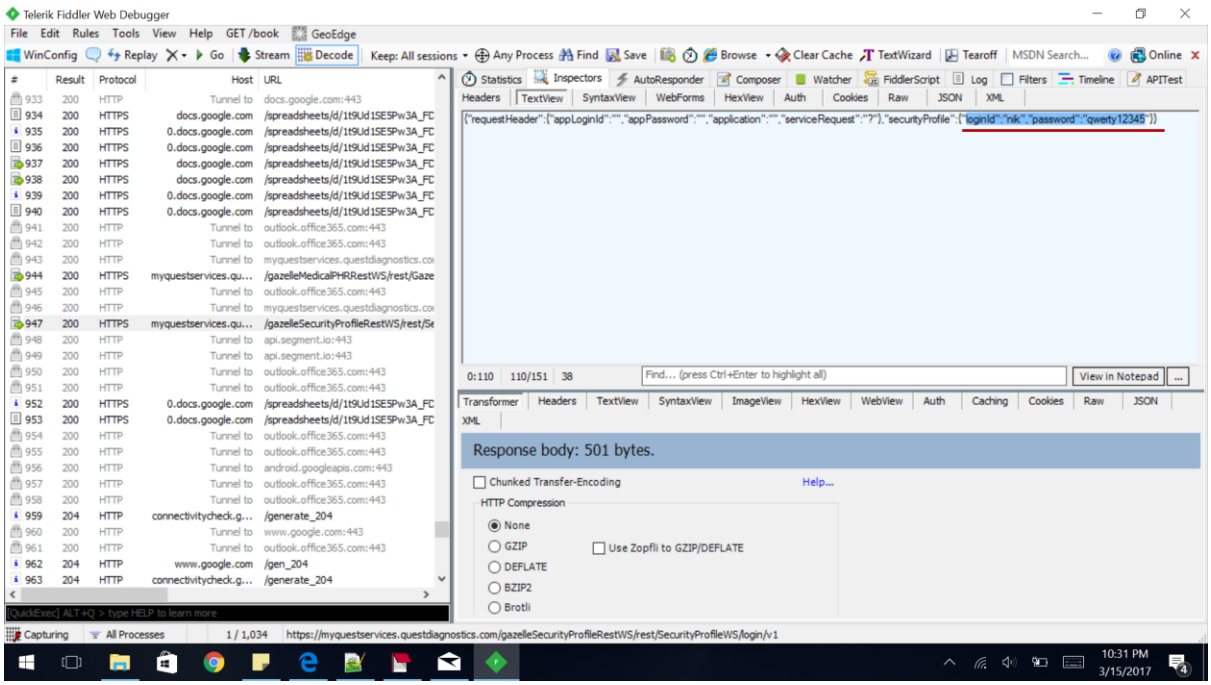
8. iCliniq App (Android)

Certificate Validation Broken – Root Chain of Trust is not validated by this app. By giving correct hostname without adding Fiddler's CA as a trusted root CA, email and password are visible in plain text while Login.



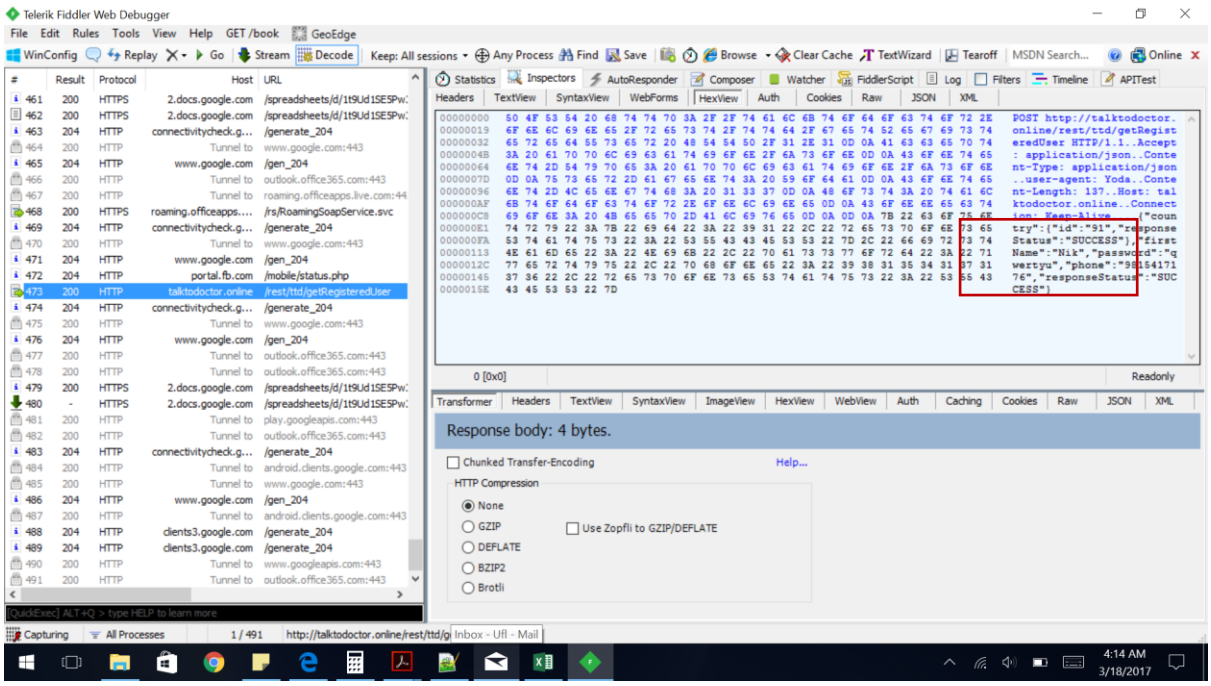
9. MyQuest App (Android)

Certificate Validation Broken – Root Chain of Trust is not validated by this app. By giving correct hostname without adding Fiddler's CA as a trusted root CA, email and password are visible in plain text while Login.



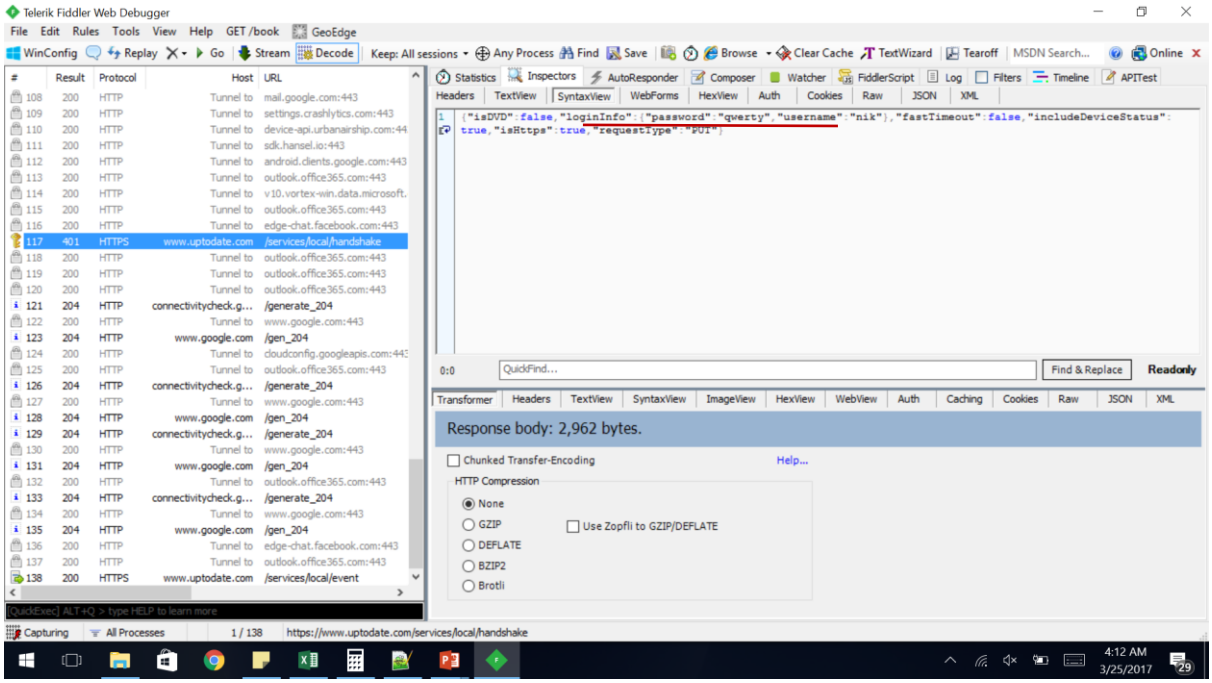
10. TalkToDoctor App (Android)

No HTTPS – In this app, there is no HTTPS connection made while Login, so email and password are visible in clear text.



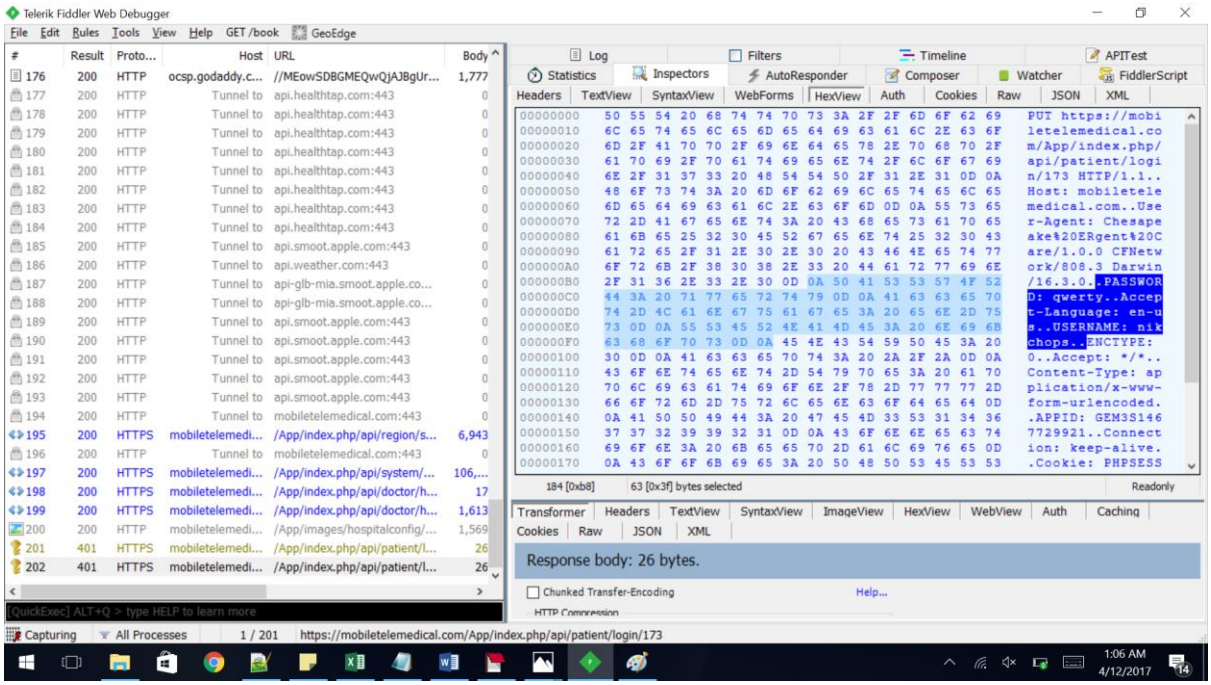
11. upToDate App (Android)

Certificate Validation Broken – Root Chain of Trust is not validated by this app. By giving correct hostname without adding Fiddler's CA as a trusted root CA, email and password are visible in plain text while Login.

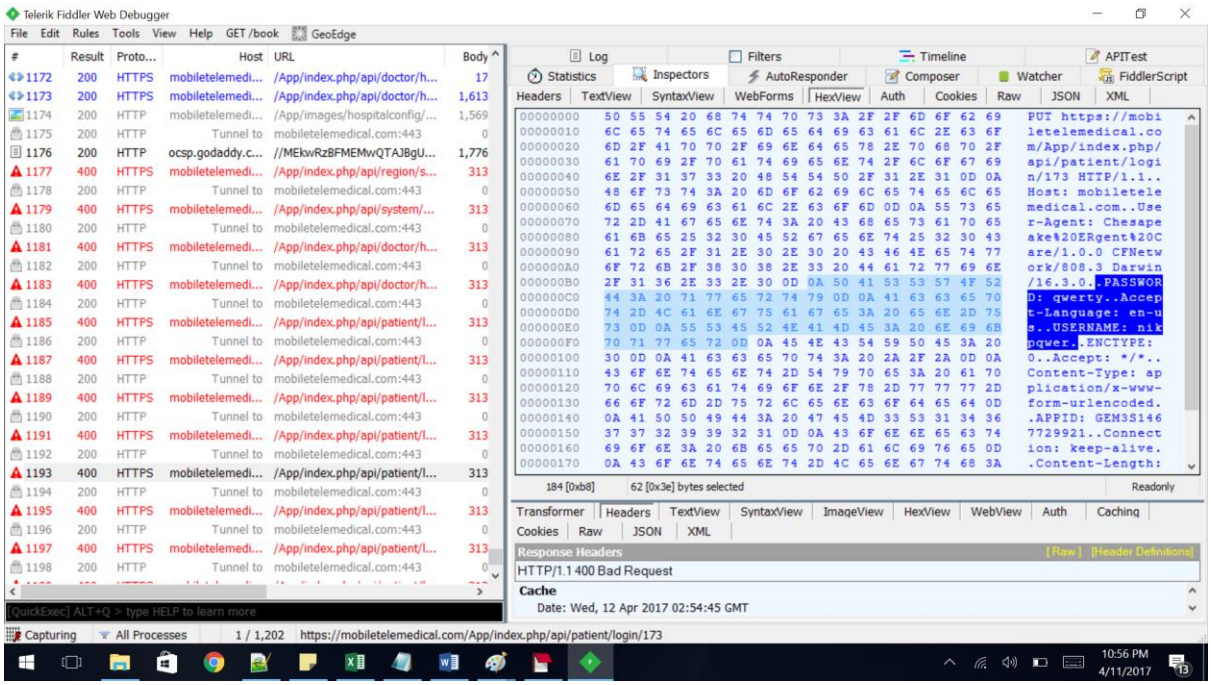


12. Chesapeake Ergent Care App (iOS)

Certificate Validation Broken – Root Chain of Trust is not validated by this app. By giving correct hostname without adding Fiddler's CA as a trusted root CA, email and password are visible in plain text while Login.

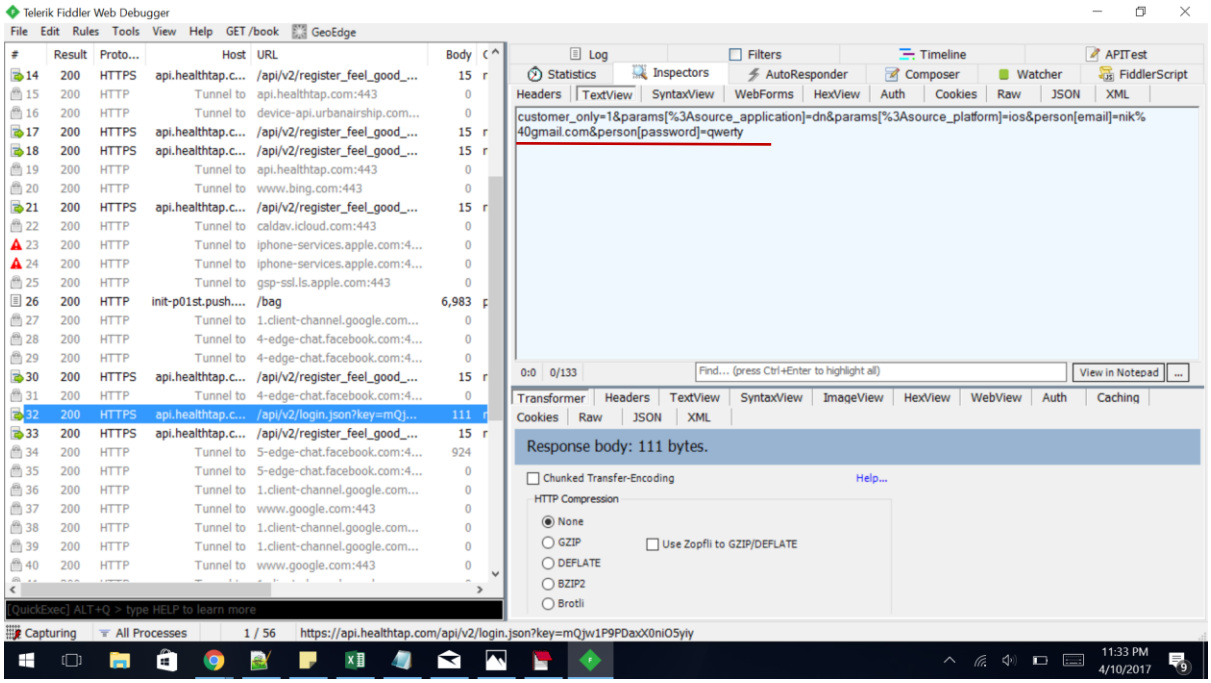


Bad Hostname Verification - By adding Fiddler's CA as a trusted CA in the phone and giving the certificate with some.fake.thing hostname, email and password is visible in clear text while Login. So, this app can be given any certificate with any hostname for MiTM.

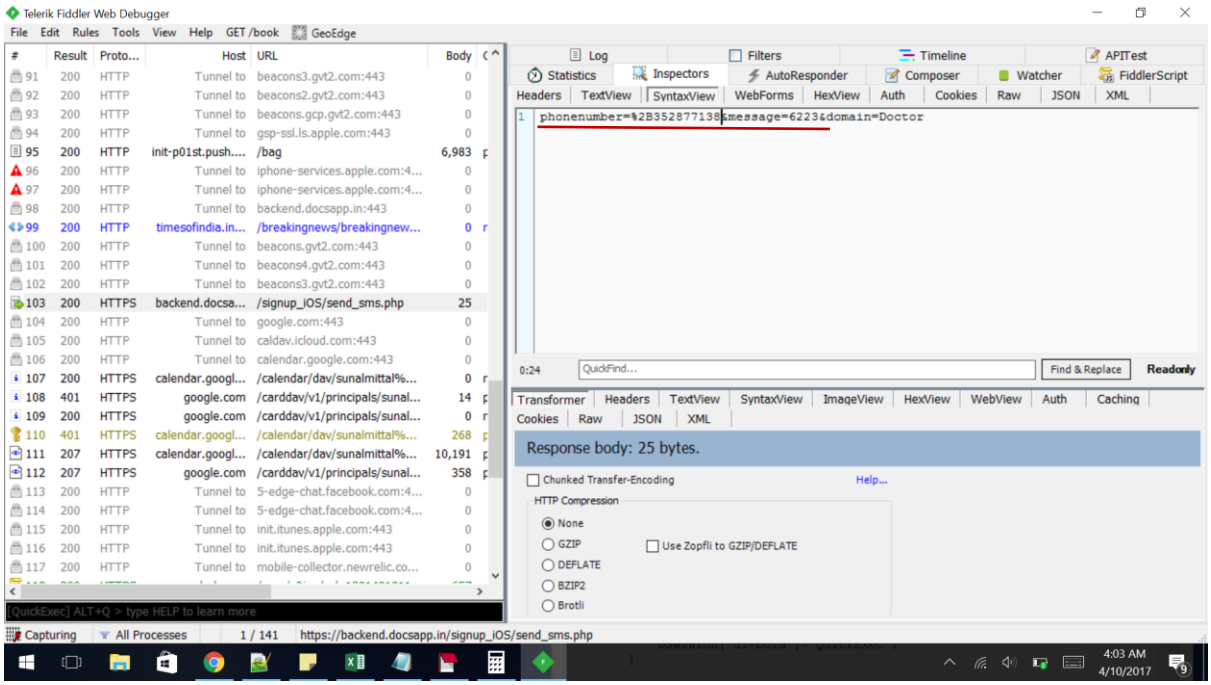


13. DocNow App (iOS)

Bad Hostname Verification - By adding Fiddler's CA as a trusted CA in the phone and giving the certificate with some.fake.thing hostname, email and password is visible in clear text while Login.

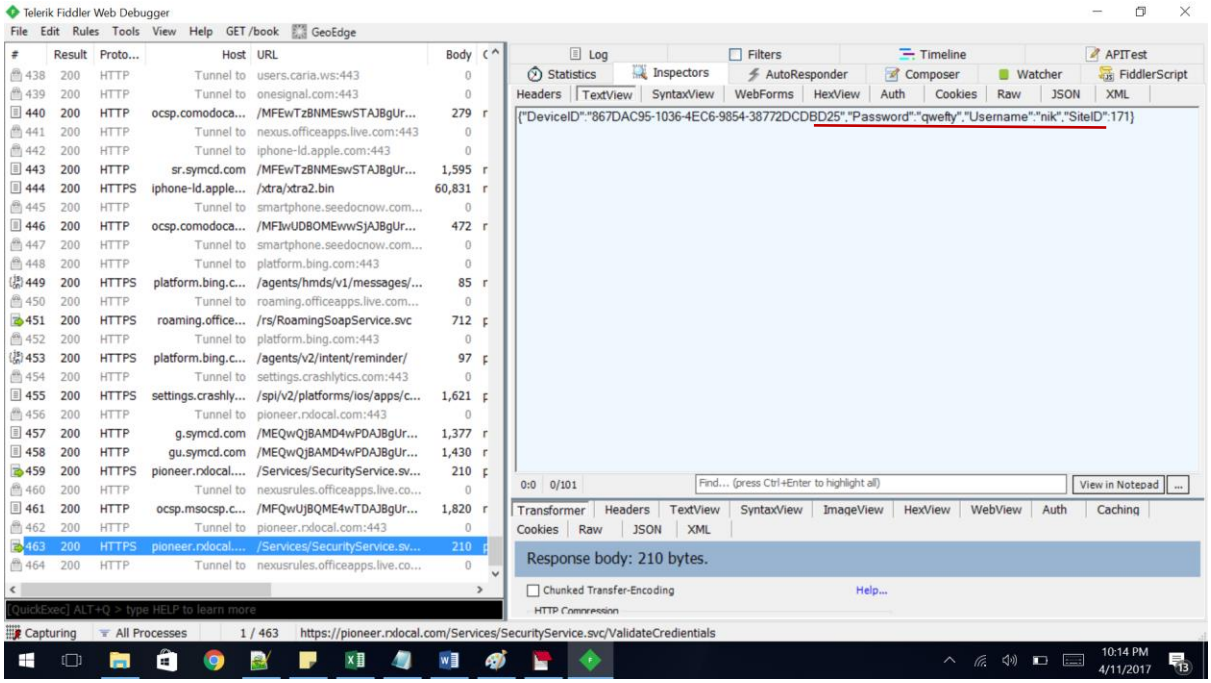


Phone Number and OTP (6223) sent are also visible in clear text in DocNow app.



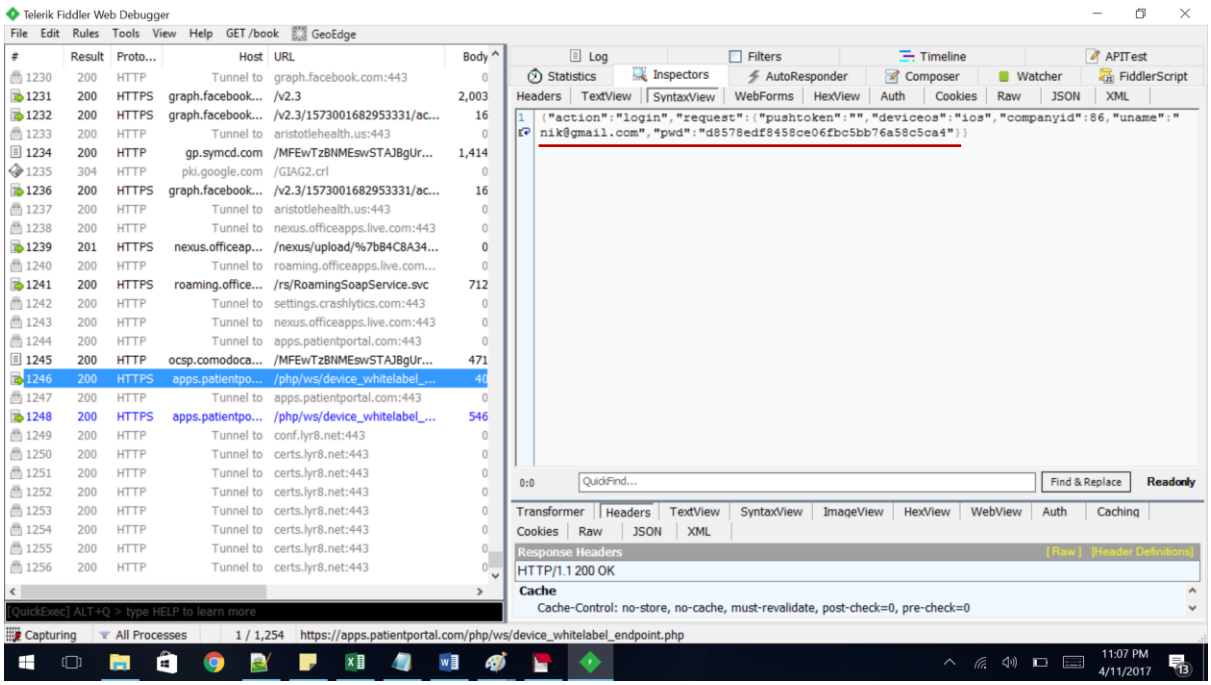
14. Houma Urgent Care Center App (iOS)

Bad Hostname Verification - By adding Fiddler's CA as a trusted CA in the phone and giving the certificate with some.fake.thing hostname, email and password is visible in clear text while Login.



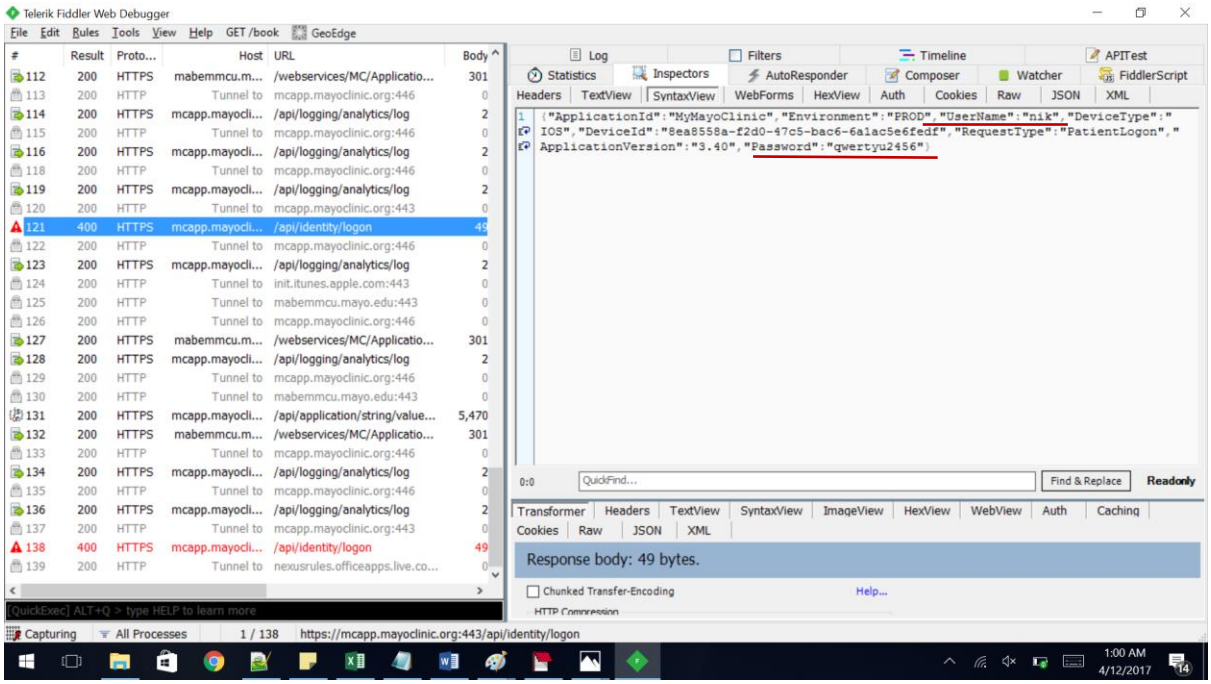
15. NextGen Patient-Portal powered by Vextar App (iOS)

Certificate validation and Hostname Verification broken - Root Chain of Trust is not validated by this app. By giving any certificate signed by any CA with some.fake.thing hostname, HTTPS packet was decrypted. Password is encrypted but username is visible in plain text.

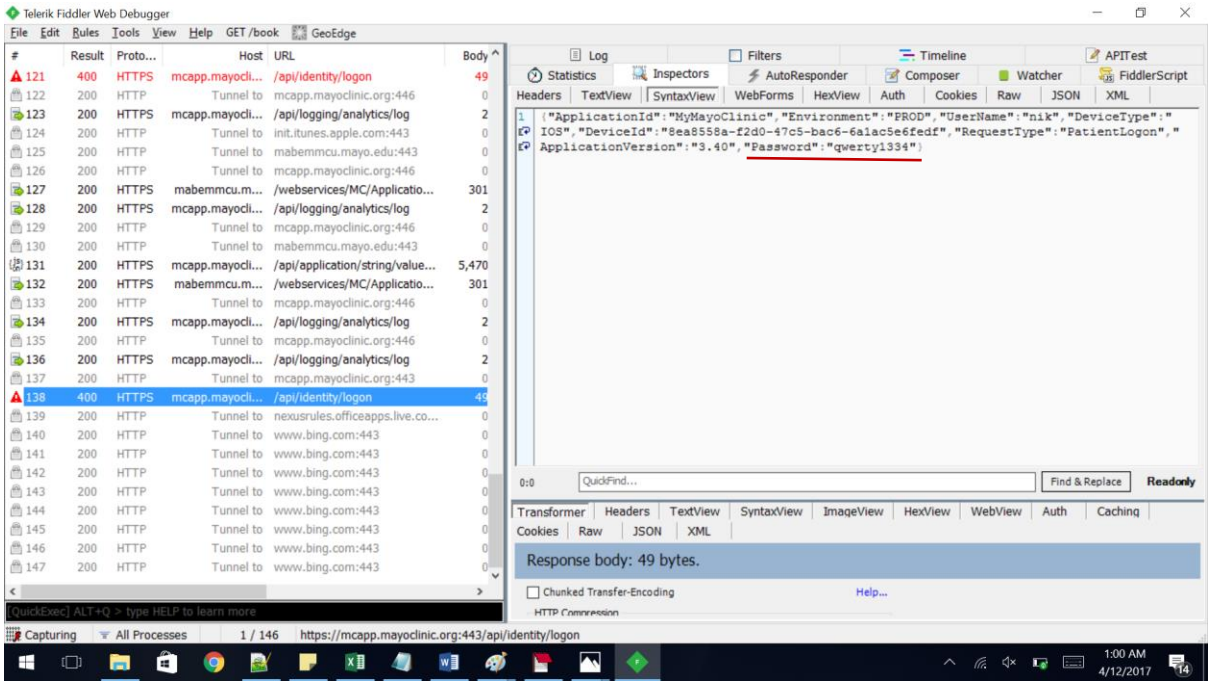


16. Mayo Clinic App (iOS)

Bad Hostname Verification - By adding Fiddler's CA as a trusted CA in the phone and giving the certificate with some.fake.thing hostname, email and password is visible in clear text while Login.



Certificate Validation Broken – Root Chain of Trust is not validated by this app. By giving correct hostname without adding Fiddler's CA as a trusted root CA, email and password are visible in plain text while Login. So, any certificate with any hostname can be used for MiTM in this App.



17. MyQuest App (iOS)

Certificate validation and Hostname Verification broken - Root Chain of Trust is not validated by this app. By giving any certificate signed by any CA with some.fake.thing hostname, email and password is visible in clear text.