Week 11 Blog

Sree Ram Boyapati	a1775690@student.adelaide.edu.au
Course	Comp Sci 7092 - Mobile and Wireless Systems
Group	CovidGuard-F
Supervisor	Zach Wang

What have I done this week?

This week I have primarily worked on Consent API for the healthcare server and waiting for my teammate to work on UI flow the user will take.

Workings of the consent API

I have created an API for granting consent and revoking the consent. I have attached screenshots of the failure scenarios and documented it.
https://github.cs.adelaide.edu.au/2020-Mobile-and-Wireless-Systems/CovidGuard-F/tree/master/API%20Screenshots%20%26%20Documentation/Consent%20API

General Flow of the Consent API:

1. The User gives consent to the healthcare API that verification server API can request for the health care information.
2. If the User grants the permission using an OTP. Consent is verified and the verification server can check for the health status of the user before accepting the keys to be uploaded.
3. The user can revoke the consent given to the server to upload more keys and the verification server will no longer connect the upload keys.

Test Results have been mocked using set-diagnosis-status API so that we can mark the user as recovered and positive using an API call and trigger the flow.

SSL Pinning

As an added security measure, I have pinned the SSL certificates of the servers which are wildcart certificates shared by all the app engine URLs. The certificate is issued to *.appspot.com. This certificate expires on 29th December of 2021.

SSL Pinning can be used to validate the certificates against hardcoded SHA 256 hashes of the trusted certificates, the developers (i.e us) have hardcoded. Using SSL Pinning, We can enforce the exact servers that the app will listen to.

PR:
https://github.cs.adelaide.edu.au/2020-Mobile-and-Wireless-Systems/CovidGuard-F/pull/74

Flow Droid (UnSuccessful)

Ref: https://github.com/secure-software-engineering/FlowDroid

PR:
https://github.cs.adelaide.edu.au/2020-Mobile-and-Wireless-Systems/CovidGuard-F/pull/76

Command:

```
java -cp lib/slf4j-api-1.7.9.jar -cp lib/slf4j-log4j12-1.7.9.jar
-jar  soot-infoflow-cmd-jar-with-dependencies.jar  -a
../../client/CovidGuard/app/build/outputs/apk/debug/app-debug.apk -p
~/Android/Sdk/platforms  -s sinks.txt  -al 5 -o output.txt
```

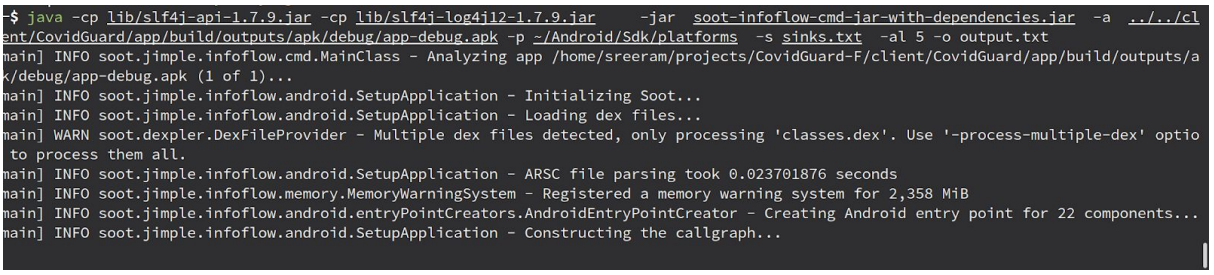

Figure 1: FlowDroid output

The flowdroid is stuck even after reducing the call depth. Still experimenting for the experimentation.

Next Week I would like to work on -

1. SQLite Encryption.
2. Adding features like revoke/add consent to the UI.

Paper Review #1

Abdelrahman Abdou and P. C. Van Oorschot. 2017. Server Location Verification (SLV) and Server Location Pinning: Augmenting TLS Authentication. ACM Trans. Priv. Secur. 21, 1, Article 1 (January 2018), 26 pages. DOI:https://doi.org/10.1145/3139294

Why have I chosen this paper?

Privacy laws of a country expect that the data of the user stays within the jurisdiction of the country. If the data is being exposed outside the country, Recovering it and ensuring the privacy laws apply will get very tough. Pinning the server to a location and pinning the certificates that the server can use. We can make sure clients only make calls to authorized servers within a country's jurisdiction

Conclusion of the Research

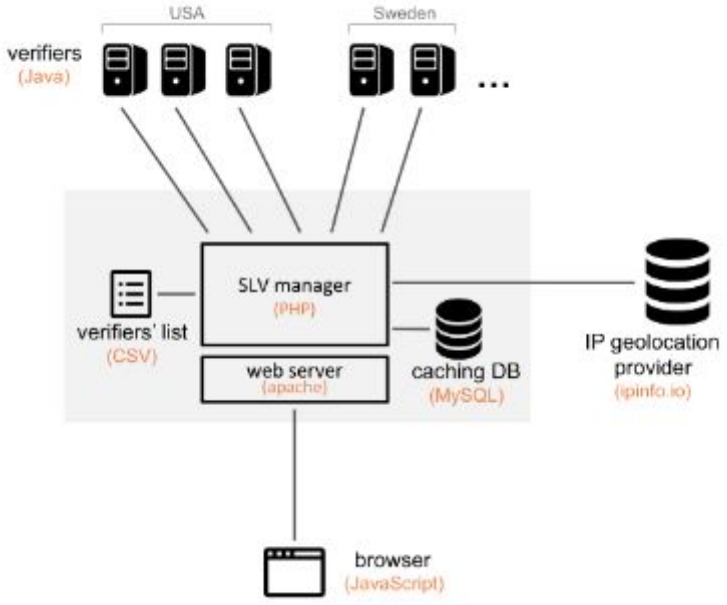

Figure 2: System Architecture proposed in the paper

The algorithm gets the IP from the geolocation provider and verifies the location based on the round trip time of the verifiers. If the verifiers return the same location, then it is verified. The verified location is then compared with the pinned locations the developers have given.
The algorithms for verification and validation of location are given in Algorithm: 1 and Algorithm: 2 of the paper. If the domain is not associated with the previous pins and location is verified, it adds the pin to the cache.

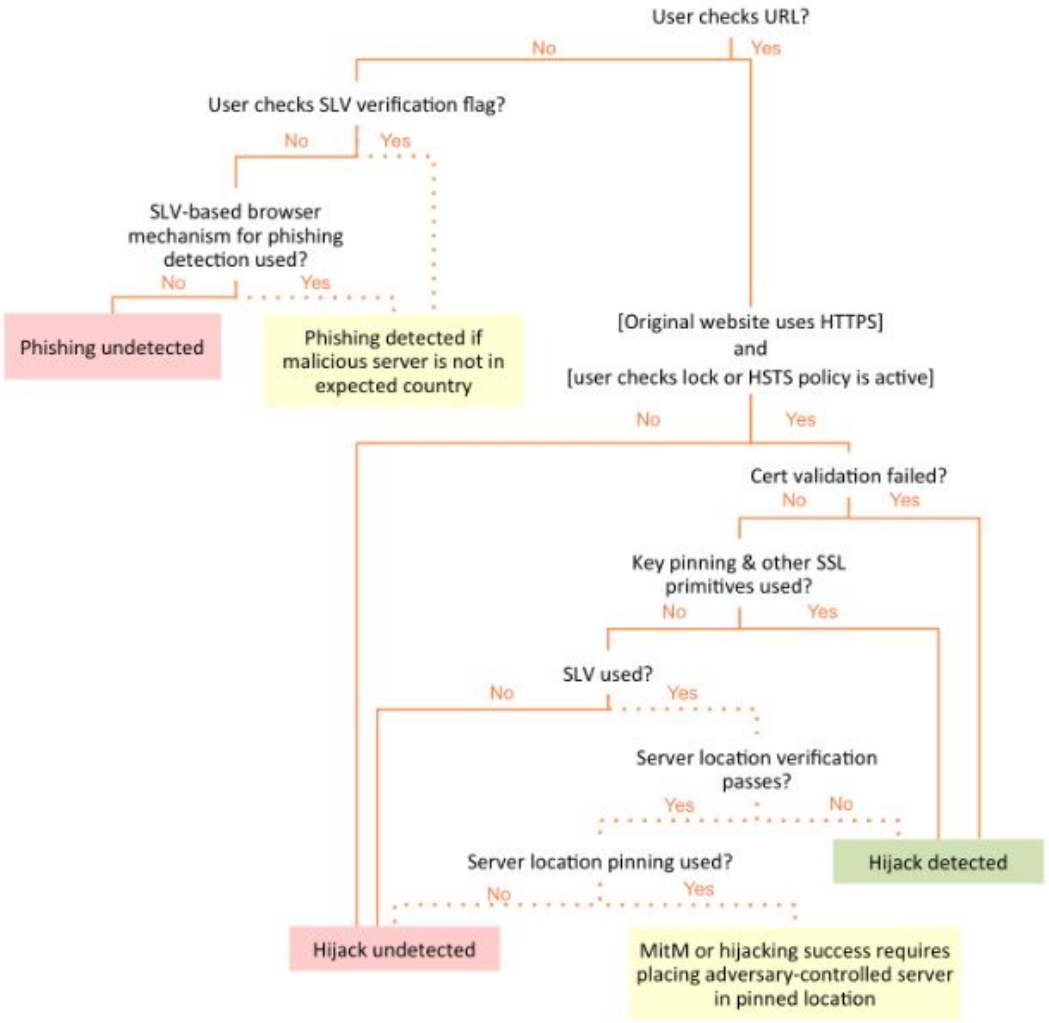

Figure 4. Decision tree for detection of traffic hijacking attacks. As explained in Section 3, from the server's perspective, phishing is a class of traffic hijacking. Dashed lines indicate attacks detected only by the new mechanisms presented herein.

Figure 4: Decision tree given in the paper for SLV verification

Outcomes that I can use.

Pros:

1. Using verifiers as the authors mentioned can circumvent attacks when the load balancer with the certificate might be in the region but not the server itself.
2. Successfully gives location pinning within a certain radius and validating it against known server locations.
3. The approach is feasible if we can use a local geoip2 database and the load average of the servers is low for the verifiers to return positively.
4. Cache of pinned locations with suitable cache eviction ratio can help in locating the server.

Cons:

1. This approach is not practically feasible. You need N number of verifiers and more the verifiers, better the accuracy of the location verification. However, these add up to egress costs and increase the time in SSL handshake.
2. Caching may not be allowed by IP Address providers. Google Maps API has such rules which can terminate your contract. Maxmind allows you to use the database in a local setting for 288 USD per month. https://www.maxmind.com/en/geoip2-databases
3. IP Address providers themselves have very poor accuracy at city level and higher as the radius increases. This may not be applicable to a lot of small countries in europe or south east asia. You are essentially wasting a lot of time because of the failures from verifiers. Ref: https://www.maxmind.com/en/geoip2-city-accuracy-comparison?country=Australia&resolution=city&cellular=all
4. RTT of the verfiers is not constant due to load average on the server. This can lead to a lot more incorrect readings and the servers may not be happy if there is no caching allowed by the IP Database and blacklist the verifiers.
5. Location is pinned by the first server that reacts to the setup. The attacker is not stealing your info by providing a server in the desired location when it is a new ip or gathering information on which servers are verifiers.
6. True Assertions of the evaluation is very high and I find it very suspicious.

The paper is really good but at scale and capacity planning, the approach may have very high server costs and poor accuracy. The paper referenced a few good papers which I find interesting.

1. GeoPKI: Converting Spatial Trust into Certificate Trust - they operate outside the hotpath of request-response and provide low latency.
2. Geolocalization of proxied services and its application to fast-flux hiddenservers. - It tries to solve the problem of malicious attackers leveraging load balancers in one regions and servers in another.
3. The papers notes that the properties of SLV are similar to “Perspectives: Improving SSH-style Host Authentication with Multi-Path Probing”

Australia

Exact City

Cellular and Broadband IPs

Exclude IPv4 IPs

Cellular IPs

Broadband IPs

Submit

Last updated: 2020-07-06

	Correctly Resolved	Incorrectly Resolved	Unresolved
GeoLite2 City	8%	89%	3%
GeoIP2 City	10%	88%	1%
GeoIP2 Precision City Service	12%	87%	1%

Figure 5: Accuracy of GeoIP based on city level.

Australia

250 km

Cellular and Broadband IPs

Exclude IPv4 IPs

Cellular IPs

Broadband IPs

Submit

Last updated: 2020-07-06

	Correctly Resolved	Incorrectly Resolved	Unresolved
GeoLite2 City	84%	13%	3%
GeoIP2 City	86%	13%	1%
GeoIP2 Precision City Service	86%	13%	1%

Figure 6: Accuracy at 250KM