

Automated Mobile Application Security Testing with Mobile Security Framework



Ajin Abraham

About Me

- Security Consultant @ 
- Security Engineering @ 
 - Next Gen Runtime Application Self Protection (RASP)
- Author of OWASP Xenotix XSS Exploit Framework, Mobile Security Framework.
- Teach Security via <https://opsecx.com>
- Blog about Security: <http://opensecurity.in>



The Takeaways

- ⦿ A FREE and Open Source Security Tool for Mobile App Security Assessment.
- ⦿ Mobile App Pentesters/Mobile Malware Analysts - How to make your job easier with MobSF.
- ⦿ Developers – Build secure mobile Apps identifying vulnerabilities at all stages of development. (SDLC Integration)
- ⦿ Web Pentesters – REST API Fuzzer capable of detecting vulnerabilities like SSRF, XXE, IDOR etc.

Agenda

❶ What is MobSF?

❷ MobSF Architecture

- ❶ Static Analyzer
- ❷ Dynamic Analyzer
- ❸ Web API Fuzzer

❸ Static Analysis

- ❶ Static Analysis & some Statistics
- ❷ Top Indian Bank Mobile Apps
- ❸ Top Indian Wallet Mobile Apps
- ❹ Observations

❹ Dynamic Analysis

- ❶ Dynamic SSL Testing
- ❷ Exported Activity Tester
- ❸ Challenges in Dynamic Analysis
- ❹ Dynamic Analysis on Custom VM/
Rooted Android Device.

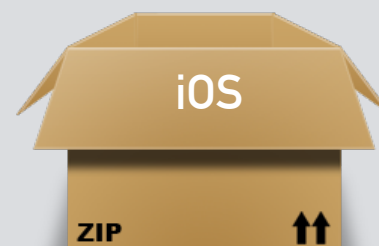
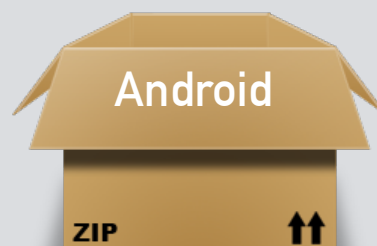
❺ Web API Fuzzer

- ❶ Vulnerabilities API Fuzzer detects.
- ❷ Explains the API Fuzzer Logic.

❻ Conclusion

What is MobSF?

Mobile Security Framework is an open source mobile application (Android/iOS) automated pentesting framework capable of performing end to end security testing of mobile Apps.



Hosted in your environment. Your application and data is never send to the cloud.



MobSF Architecture

Static Analyzer



Demo

Static Analysis & Report Generation

(Diva)

Static Analysis & Some Statistics

- ⦿ Static Analysis on Top Financial Apps - Criteria
 - ⦿ SSL bypass in Native Code
 - ⦿ SSL bypass in WebView
 - ⦿ Remote Web View Debugging
 - ⦿ Hardcoded Secrets









Top Indian Bank Apps Analyzed

Bank App	Native SSL Bypass	WebView SSL Bypass	Remote Webview Debug	Hardcoded Secrets	Root Detection
	Yes	No	No	No	No
	No	No	No	No	No
	No	Yes	Yes	No	No
	No	No	No	Yes	No
	Yes	Yes	Yes	Yes	No
	Yes	No	Yes	No	Yes
	No	No	No	Yes	No
	Yes	No	Yes	No	No
	No	No	No	No	Yes
	Yes	No	No	No	No
	No	No	No	No	No
	No	No	No	No	Yes

Face palm



Top Indian Wallet Apps Analyzed

Wallet App	Native SSL Bypass	Webview SSL Bypass	Remote Webview Debug	Hardcoded Secrets	Root Detection
	Yes	Yes	Yes	No	Yes
	Yes	Yes	No	No	No
	No	Yes	No	No	No
	No	Yes	Yes	No	No
	No	Yes	Yes	No	Yes
	No	No	Yes	No	Yes
	Yes	No	No	No	No
	No	Yes	No	No	Yes

Observations

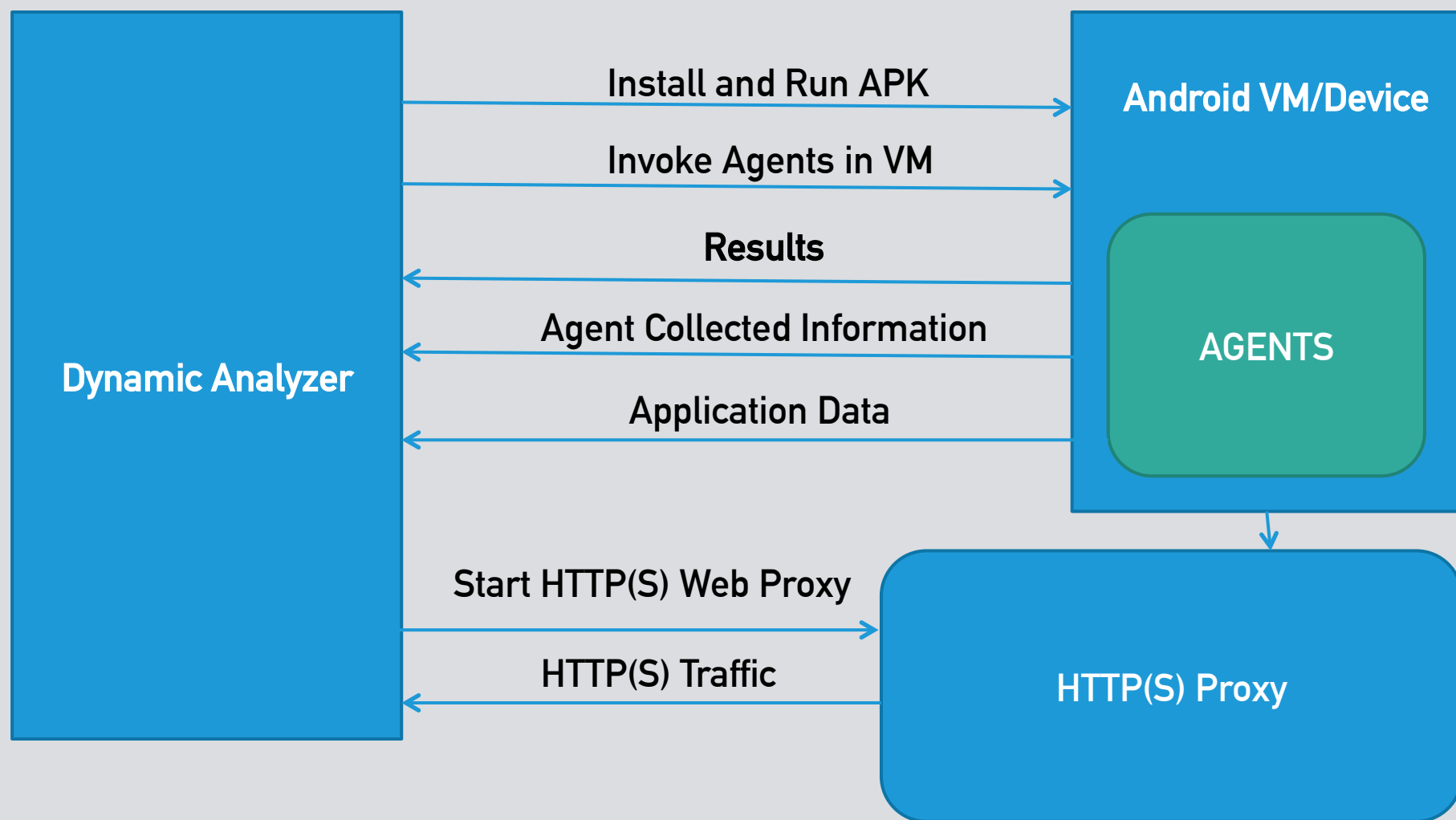
- 🌀 State of Mobile App Security, Not evolved as Web Security.
- 🌀 Most common issue is SSL Bypass in (Both Native Code and WebView)
- 🌀 SSL Error bypassed in WebViews are really really bad.


```
public void onReceivedSslError(Webview webView, SslErrorHandler sslErrorHandler, SslError sslError) {
    try {
        sslErrorHandler.proceed();
        return;
    }
}
```


Dynamic Analyzer



Dynamic Analyzer - Architecture



DEMO

(LOCX)

Dynamic SSL Testing

- 🌀 Dynamically verify if SSL connections are securely implemented.
- 🌀 Disable JustTrustMe and Remove MobSF Root CA.
- 🌀 If you can still access the decrypted HTTPS Web Traffic then that means the app is bypassing SSL errors.

Exported Activity Tester

- Android Exported Activities.

```
<activity
  android:name=".ExportedActivity"
  android:label="ExportedActivity"
  android:exported="true">
</activity>
<activity
  android:name=".ImplicitlyExportedActivity"
  android:label="ImplicitlyExportedActivity" >
  <intent-filter>
    <action android:name="opensecurity.vulnapp.INTENT"/>
  </intent-filter>
</activity>
```

DEMO

Challenges in Dynamic Analysis

- ⦿ Some Android Apps are built with security in mind.
 - ⦿ Anti VM Detection
 - ⦿ Anti Root Detection
 - ⦿ Anti MITM with Certificate Pinning.
 - ⦿ Some Apps / Malwares have sophisticated methods to detect Virtual Machines.

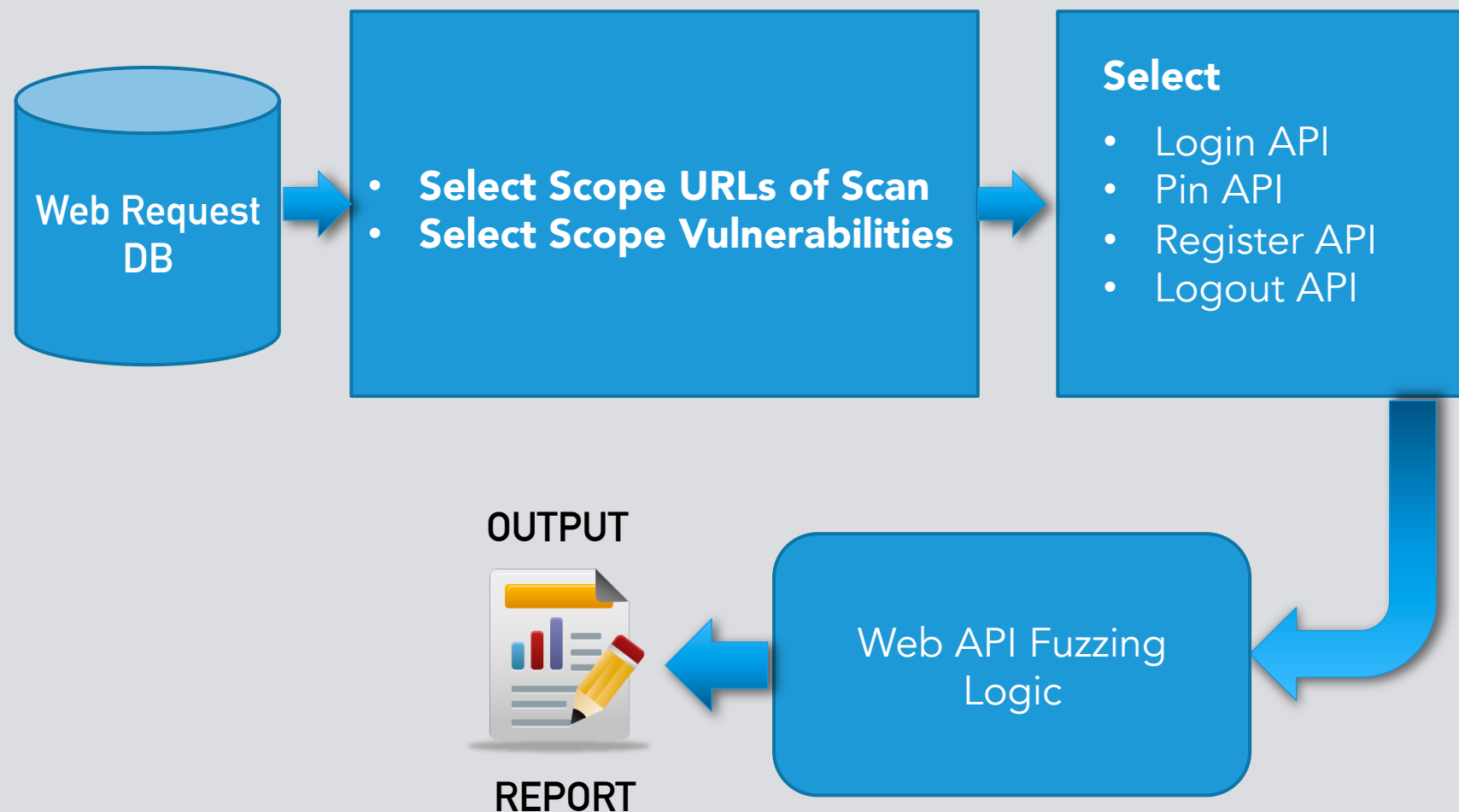
How to deal with these Challenges

- ⊗ API overriding with Xposed Framework
 - ⊗ Anti VM Detection Bypass → Android Blue Pill
 - ⊗ Anti Root Detection Bypass → RootCloak
 - ⊗ Anti MITM Certificate Pinning Bypass → JustTrustMe
- ⊗ APK smali Patching.
- ⊗ For sophisticated apps and malware, Use a real device for dynamic analysis.

Dynamic Analysis on Device

- ⚙ MobSFy Script – Convert your VM/ Device to support MobSF Dynamic Analysis
- ⚙ Documentation here:
<https://github.com/ajinabraham/Mobile-Security-Framework-MobSF/wiki/2.-Configure-MobSF-Dynamic-Analysis-Environment-in-your-Android-Device-or-VM>
- ⚙ DEMO : Weak Crypto
 - ⚙ Java - String hashCode() Method
 - ⚙ $s[0]*31^{(n-1)} + s[1]*31^{(n-2)} + \dots + s[n-1]$

Web API Fuzzer



Fuzzing REST APIs

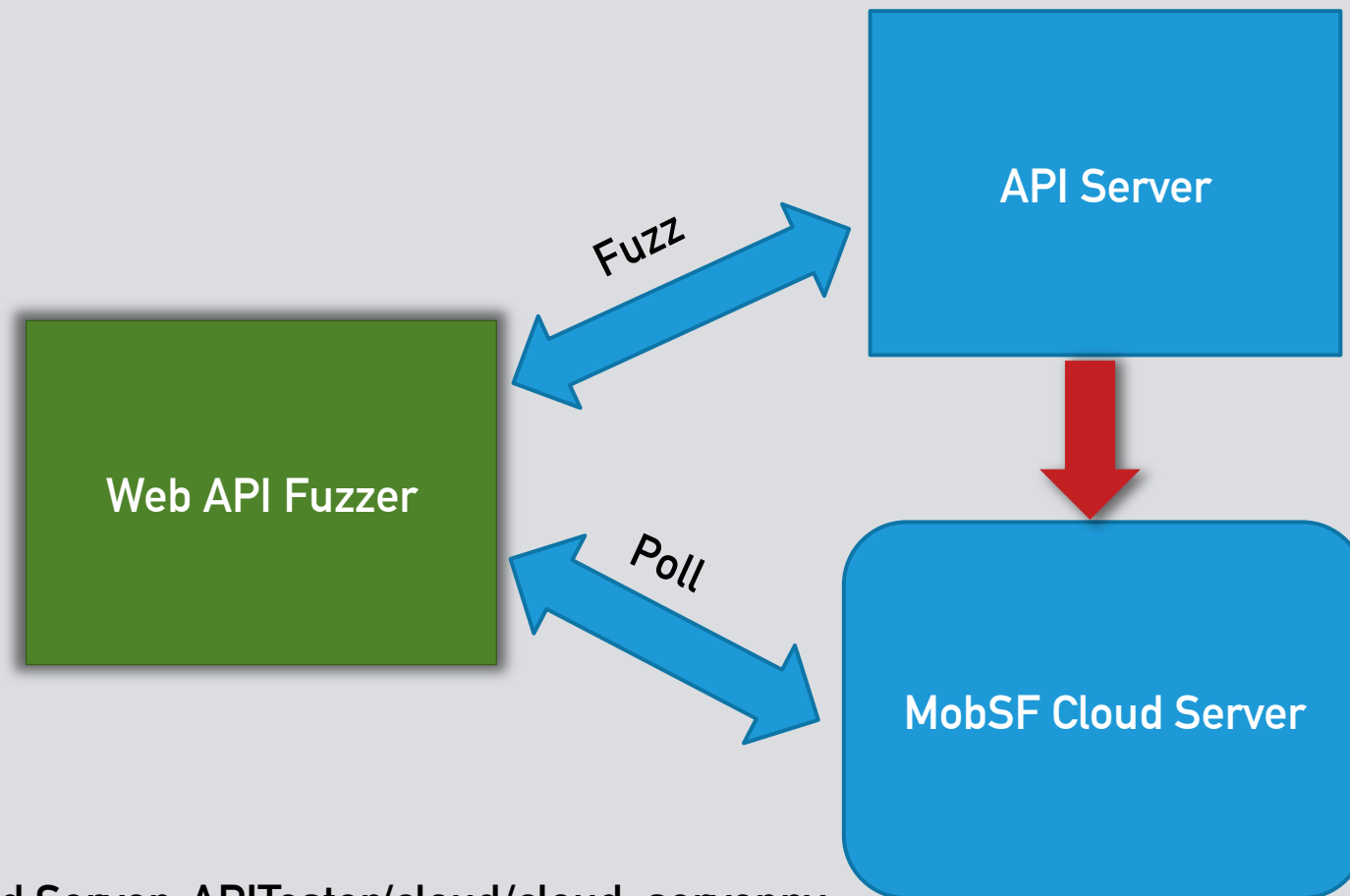
- ❶ Why most web scanners suck at API Testing?
- ❷ We have knowledge about the application and generic API routes (Login, Logout, Register).
- ❸ So we use more of Whitebox approach than Blackbox approach.
- ❹ Detects vulnerabilities like IDOR, SSRF and XXE.

What We Detect

- ⦿ **XXE**
- ⦿ **SSRF**
- ⦿ **IDOR**
- ⦿ **Directory Traversal or Path Traversal**
- ⦿ **Logical and Session Related**
- ⦿ **API Rate Limiting**

How we Detect

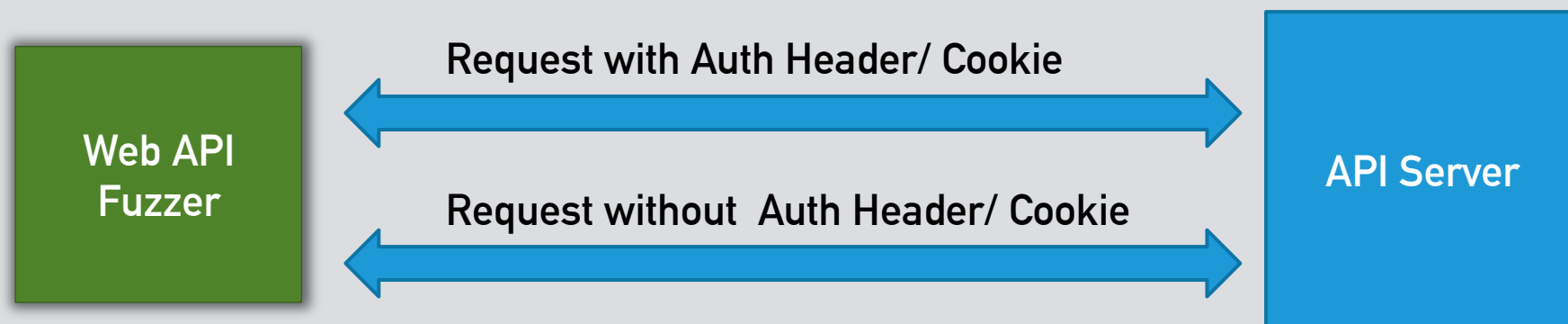
SSRF & XXE



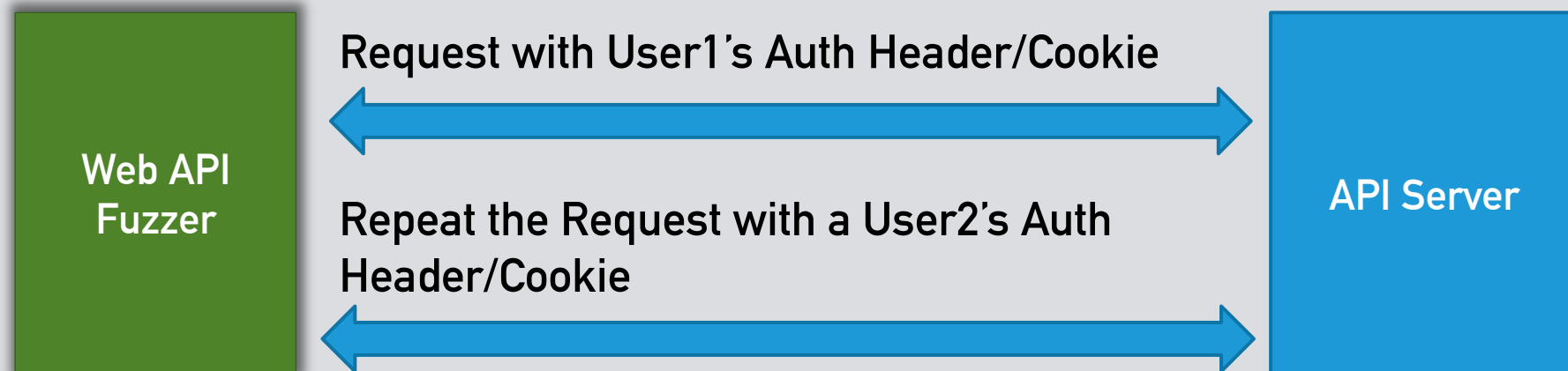
Cloud Server: APITester/cloud/cloud_server.py

Insecure Direct Object Reference (IDOR)

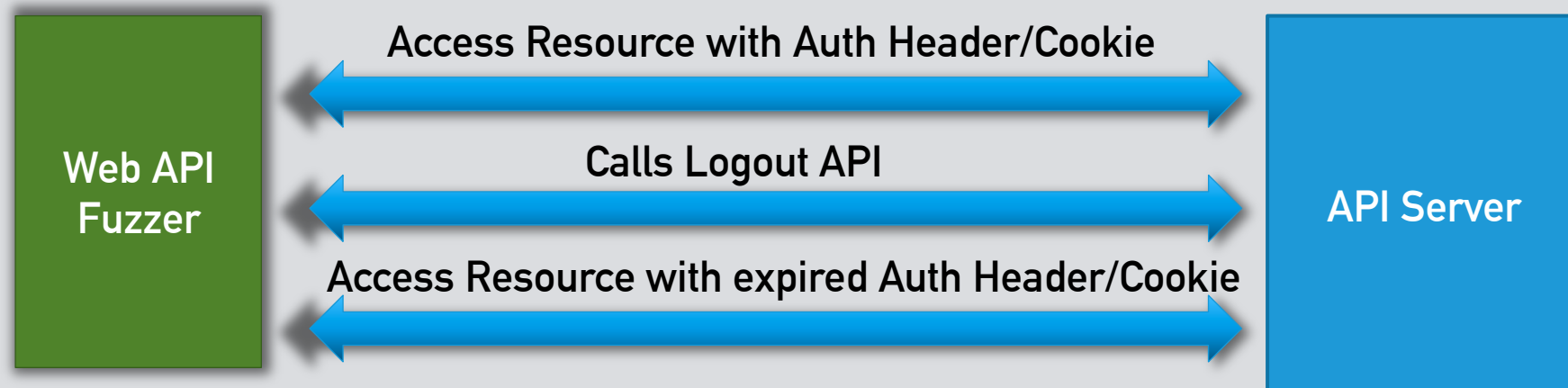
Without Credentials.



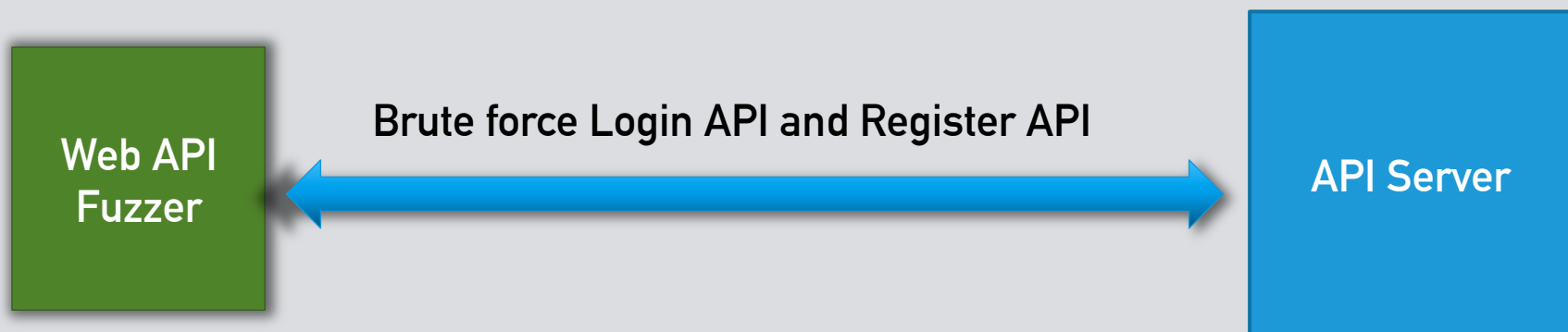
With multiple user credentials (needs two login attempts)



Session Related Checks



Rate Limiter



Other Checks

- Security Headers and Info Gathering
- Directory/ Path Traversal

DEMO

What's Coming Soon?

- 🌀 Windows App Security Analyzer.
- 🌀 iOS App Dynamic Analysis.
- 🌀 API Fuzzer to support detection of SQLi and RCE.
- 🌀 Export Proxy logs to BurpSuite/IronWASP/ZAP

Stakeholders

🎯 Looks like people are interested!

👁 Unwatch ▾ 32 ★ Star 115 🍴 Fork 42

👁 Unwatch ▾ 83 ★ Star 533 🍴 Fork 235

🎯 Bugs opened and Closed

<input type="checkbox"/>	🔔 9 Open	✓ 40 Closed	Author ▾	Labels ▾	Milestones ▾	A
<input type="checkbox"/>	🔔	IOException at /StaticAnalyzer_iOS/	bug			
		#50 opened 8 days ago by jairoseranosp				
<input type="checkbox"/>	🔔	[Suggestion - Enhancement] Save&Show reports	question			
		#49 opened 13 days ago by tachencko				
<input type="checkbox"/>	🔔	Error while "Extracting Strings from APK"	bug			
		#48 opened 15 days ago by orenyomtov				
<input type="checkbox"/>	🔔	In the case of strong binary, Failure message should be more clear and self exploratory. [Enc]	bug			
		#47 opened 16 days ago by bugwrangler				

<input type="checkbox"/>	🔔 13 Open	✓ 89 Closed	Author ▾
<input type="checkbox"/>	🔔	Dynamic Analysis	enhancement
		#105 opened 6 days ago by ajinabraham	
<input type="checkbox"/>	🔔	Suggestions	enhancement
		#102 opened 16 days ago by adithyanaresh	
<input type="checkbox"/>	🔔	Enhancement: Malware check for all URL / Domain / IP	enhancement
		#100 opened 21 days ago by anantshri	

Useful Links



Source:

<https://github.com/ajinabraham/Mobile-Security-Framework>



Issues:

<https://github.com/ajinabraham/Mobile-Security-Framework/issues>



Documentation:

<https://github.com/ajinabraham/Mobile-Security-Framework-MobSF/wiki>



Video Course:

<https://opsecx.com/index.php/product/automated-mobile-application-security-assessment-with-mobsf/>

QA

Thanks & Credits

- Sachinraj Shetty
- Kamaiah Nadavala
- Bharadwaj Machiraju
- Yashin Mehboobe
- Anto Joseph
- Tim Brown
- Thomas Abraham
- Graphics/Image Owners

@ajinabraham

ajin25@gmail.com