

FRIDA 101



Ementa

Introdução a RASP

FRIDA 101

Scripting 101

Contornando Detecções contra...

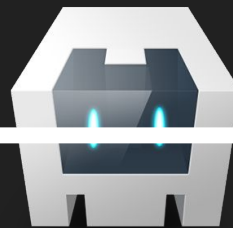
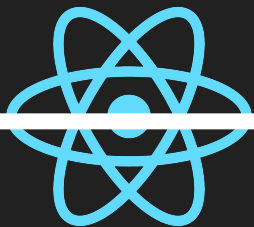
Root

Emulador

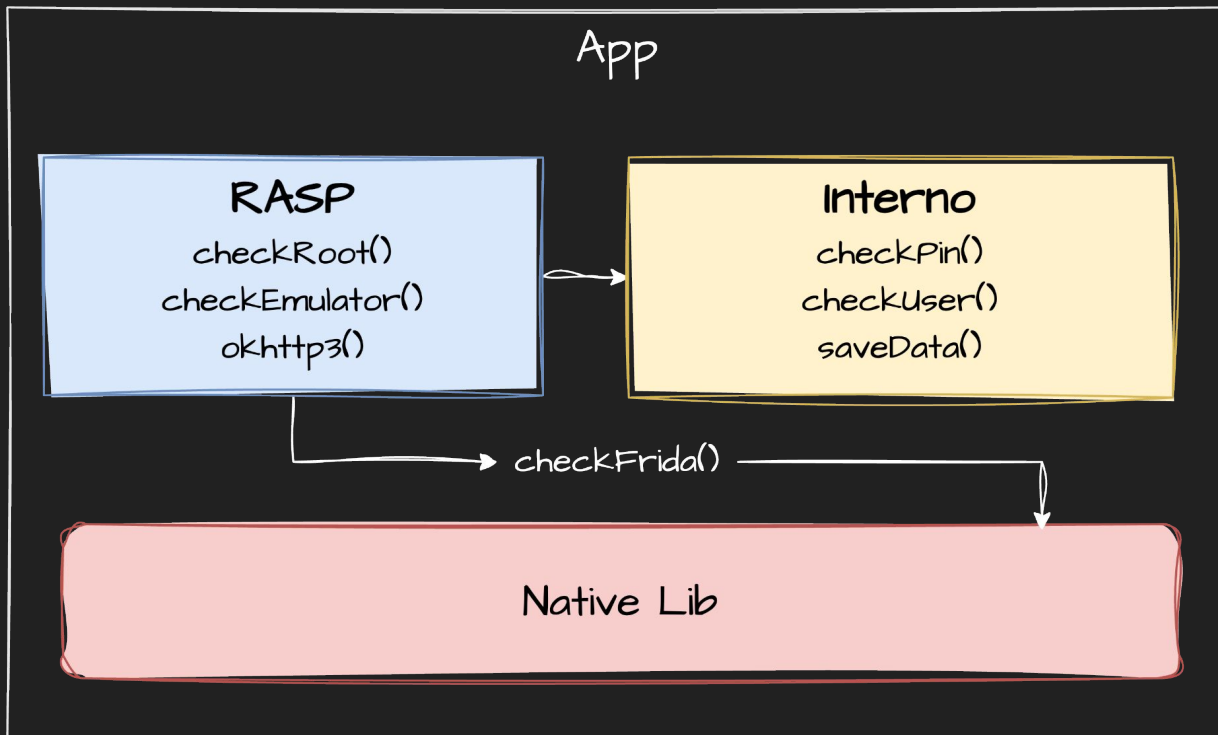
Frida

SSL Pinning

Nota



Introdução a RASP



Introdução a RASP

RootBeer



Trusted by 5 thousand+ apps

A tasty root checker library and sample app. We've scoured the internets for different methods of answering that age old question...

GantMan / jail-monkey Public

Notifications Fork 151 Star 629

<> Code Issues 38 Pull requests 17 Actions Projects Security Insights

master Go to file <> Code About

kmeye v2.8.3 ✓ 04e31e0 · 2 months ago

.github Update java version last year

.yarn/releases add yarn version into package.json 3 months ago

A React Native library for identifying if a phone is rooted or mocking locations

react android react-native jailbreak trust mock-locations

Android RASP

CI failing kotlin 1.8.20 minAndroidSDK 24 targetAndroidSDK 34 gradle 8.2.0 Maven

An open-source RASP (Runtime Application Self-Protection) solution for protecting Android apps against being run on vulnerable devices.

Note

Android RASP is still in development, meaning that some breaking changes are likely to be introduced in future releases. See [Versioning](#) section for more information.

Introdução a RASP

Proteção de alto nível contra as principais ameaças

- Engenharia reversa
- Adulteração de aplicativos (Tampering)
- Injeção de script
- Ataques a APIs
- Roubo de chaves e dados confidenciais
- Tunneling
- Keylogging
- Clonagem
- DDoS
- Roubo de IP
- Roubo de credenciais
- Sequestro de nós e rotas
- Ataques de sobreposição
- Desacoplamento de servidor

DexGuard

With DexGuard, developers achieve the highest level of protection in the easiest possible way for Android apps.



Multi-layered & polymorphic protection

DexGuard's comprehensive static and dynamic analysis protections are achieved through layered obfuscation and encryption techniques complemented by automated RASP checks and built-in malware defenses. DexGuard obfuscates all checks and the multi-layered approach includes Android NDK - C/C++ native libraries.



Productivity and visibility

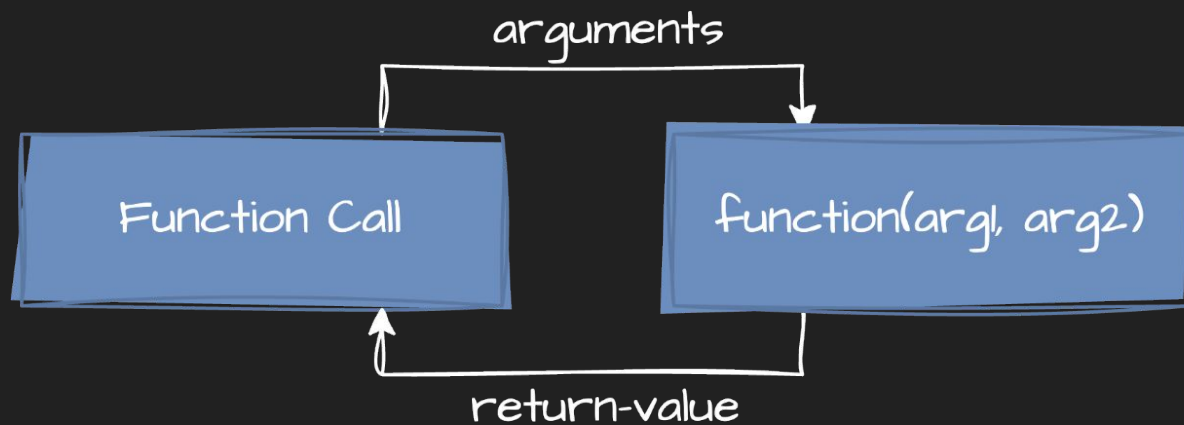
Achieve seamless implementation with a guided configuration that simplifies setup, ensuring the highest level of protection without compromising app stability or performance. Attain actionable security insights, enhancing collaboration between security and dev teams with build history visibility and protection reports for maximum app security.



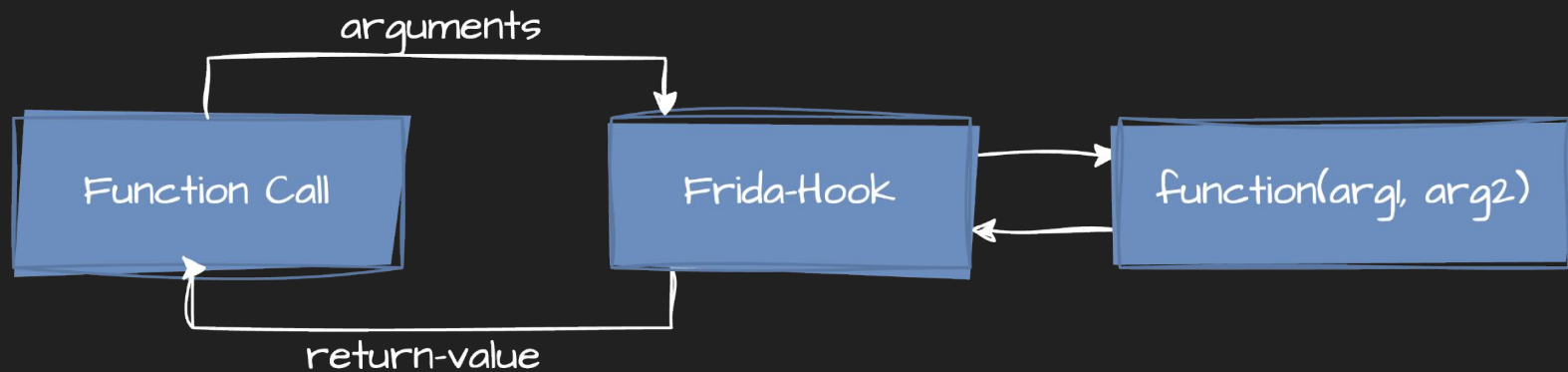
Backwards compatible with ProGuard & R8

When upgrading from ProGuard (or R8) to DexGuard, you can re-use your existing optimization configuration file. All you need to do is account for DexGuard's additional functionality, including its RASP and obfuscation capabilities.

FRIDA 101



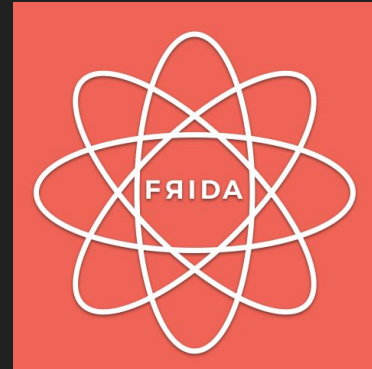
FRIDA-HOOK



- Ler / Modificar os argumentos da função
- Ler / Modificar o valor de retorno da função
- Chamar a função normalmente
- Chamar uma função diferente

Ferramentas

- Android Studio + Emulador + Platform Tools
- JADX-GUI
- Frida + Frida-Tools
- Burp Suite
- E muita paciência :)



Scripting IOI

- `Java.perform()` recebe uma função anônima que será executada dentro do contexto da JVM.
- Sem isso, pode ocorrer erro ao acessar classes Java.



```
Java.perform(function() {  
    console.log("Frida script rodando!");  
});
```

Scripting 101

- A função `Java.use()` permite obter uma referência para uma classe Java carregada na aplicação.



```
Java.perform(function() {  
    var MainActivity = Java.use("com.exemplo.MainActivity");  
    console.log("Classe carregada:", MainActivity);  
});
```

Scripting 101


- Se um método tem várias versões (overloads), usamos `overload()` para escolher qual queremos modificar.



```
Java.perform(function() {  
    var Classe = Java.use("com.exemplo.MainActivity");  
  
    Classe.login.overload("java.lang.String", "java.lang.String");  
});
```

Scripting 101

- Usamos `implementation` para substituir métodos de uma classe, ou para apenas fazer um debug.



```
Java.perform(function() {  
    var Crypto = Java.use("com.exemplo.Crypto");  
  
    Crypto.encrypt.implementation = function(data) {  
        console.log("Interceptado encrypt:", data);  
        return this.encrypt(data);  
    };  
});
```


DEMO