AMANDA BARBOSA SOBRINHO
AMANDA@RESH.COM.BR

Como preparar seu ambiente para Pentest de Aplicações iOS

resh®
Cyber Defense

Pentest de Aplicações iOS: por quê?

Qualquer aplicação (web e mobile) precisa passar por um processo de Pentest antes de ser lançada;

- Evita vazamentos, distorções de regras de negócio e invasões em sistemas internos.

MacOS e iOS são sistemas de acesso mais limitado;

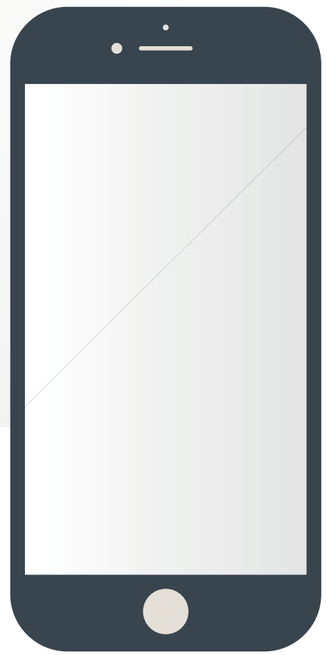
- Ideia de que são mais seguros do que os outros;
- Leva a uma potencial negligência com relação a segurança no desenvolvimento de aplicações.

Pentest de Aplicações iOS: por quê?

2

> Pentests em apps iOS reduzem a área de ataque destas aplicações.
> Encontra vulnerabilidades que podem ser empregadas por usuários mal-intencionados para subverter o funcionamento esperado do app.

Equipamentos Necessários



iPhone com no mínimo iOS 11.0 e Jailbreak;

É possível realizar o Pentest sem Jailbreak, mas o processo é mais moroso e menos assertivo.

MacOS Sierra (10.12.6) ou superior.

Adversidades na preparação para o Pentest

Incompatibilidade entre versões de Sistemas Operacionais e de hardware;

Incompatibilidade de versões de Jailbreak para iOS;

Incompatibilidade entre versões de iOS e ferramentas.

Compatibilidade entre versões de iOS e iPhones

iOS	iPhone
11	5S - X
12	5S - XR
13	6S/6S Plus - XR

Existência de Jailbreak para versões de iOS

iOS	Jailbreak?
11 - 11.4.1	Sim!
12.0 - 12.2	Sim!
12.3 - 12.3.2	Sim (mas é um pouco mais complicado)
12.4	Sim!
12.4.1 - 13.2.3	Sim (mas é um pouco mais complicado)

Problema: pode ser que o Jailbreak disponível seja Semi-Untethered.
Link: https://canijailbreak.com/

Passo-a-passo da preparação do ambiente

Ferramentas Básicas

Devem ser instaladas no iPhone via Cydia.
Serão utilizadas para transferência de arquivos e comunicação com o iPhone.

1. OpenSSH - lembrar de trocar a senha padrão (alpine) para os usuários “root” e “mobile”.
2. APG 0.6 Transational (apt-get)
3. Erica Utilities
4. Unzip
5. adv-cmds (ps)

1

Ferramenta 1
Xcode

Necessária para o processo de instalação dos apps.

Instalar por meio da loja no macOS.

Versões mais antigas:
https://developer.apple.com/download/more/?=xcode

2

Ferramenta 1
Xcode

Objetivo: obter o perfil de provisionamento que será usado para reassinar os apps a serem testados.

Apps enviados para teste são assinados com o perfil de provisionamento do desenvolvedor.

- UDID do iPhone em que o app será testado deveria ser cadastrado no momento de gerar o perfil de provisionamento.

Reassinar o app com seu perfil de provisionamento gratuito.

3

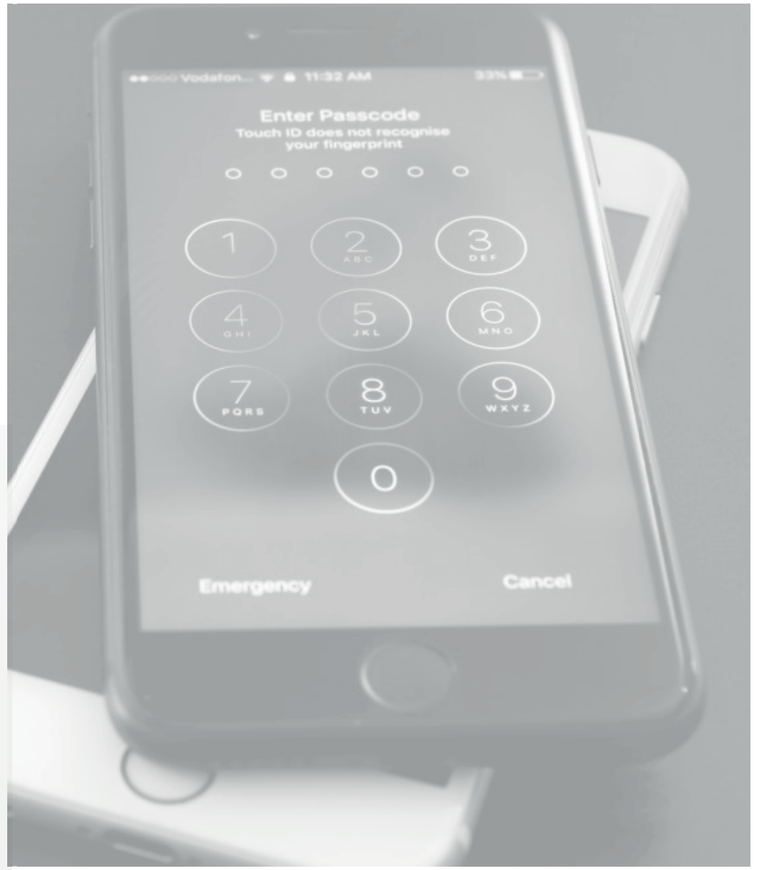
Ferramenta 1
Xcode

O Xcode permite exportar o perfil de provisionamento (Provisioning Profile) que será usado para reassinar o arquivo do app;

Para isso, é necessário criar um projeto “dummy” no Xcode com a sua Apple ID comum associada.
Arquivo de provisionamento do projeto “dummy” será copiado e empregado para reassinar o app.

Detalhes:
https://github.com/nowsecure/node-applesign/-wiki/Setting-up-your-resigning-environment

Ferramenta 2
Node Applesign



Módulo em Node-JS para reassinar arquivos ipa.

Usa o perfil de provisionamento exportado pelo Xcode.

Forma mais simples de reassinar um app que será testado (existem outras formas).

Ferramenta 2
Node Applesign

2

Instalação no macOS:

1. Instalar o Node e NPM a partir do instalador oficial.
2. npm install applesign

Utilização:

/bin/applesign.js --identity <hash_da_id> --mobileprovision ~/embedded.mobileprovision caminho/do/arquivo.ipa

Cria um novo arquivo ipa reassinado.

Antes de instalar o app no iPhone

Importar o certificado correspondente do perfil de provisionamento que será usado para reassinar os apps a serem testados.

Certificado precisa ser exportado via Xcode (arquivo .developerprofile);

Arquivo .p12 deve ser importado no iPhone e “confiado”.
https://github.com/nowsecure/node-applesign/wiki/-Setting-up-your-resigning-environment

Ferramenta 3
ios-deploy

- Faz a instalação do ipa no iPhone.
- Instalação no macOS:

npm install -g ios-deploy --unsafe-perm=true

(Node e NPM já foram instalados).

- Utilização:
ios-deploy -b caminho/do/ipa_reassinado.ipa

Ferramenta 4 - Frida

Permite interceptar o funcionamento do app e inserir funções JavaScript durante o funcionamento.

Útil para verificar o fluxo do app, e obter valores retornados por funções internas durante sua execução.

Necessária para que a ferramenta Objection funcione.

Ferramenta 4 - Frida

2

• Instalação no macOS (precisa do Python3):
pip install frida-tools

• Instalação no iPhone:
1. Repositório https://build.frida.re no Cydia;
2. Buscar “Frida” no Cydia e instalar.

Ferramenta 5
Objection

Permite obter diversas informações sobre o app de forma simples.

É a ferramenta com mais recursos e mais simples de usar.

Baseada no Frida.

Instalação no macOS (precisa do Python3):
pip3 install objection

Ferramenta 5
Objection

2

Utilização:

• Descobrir o nome/ID do app a ser analisado:
frida-ps -Ua -> lista todos os apps rodando no iPhone.

• Com o app em primeiro plano:
objection -g nome_do_app explore

• A interface do Objection será aberta, e os comandos a serem dados podem ser listados com TAB.

Ferramenta 5
Objection

03

Alguns comandos úteis:
env -> lista as pastas usadas pelo app;

sqlite connect arquivo.db -> conecta com uma base de dados do app;

ios cookies get -> mostra os cookies do app;

ios pasteboard monitor -> monitora o que é copiado/colado do/para o app.

Ferramenta 6
Burp

Proxy para interceptar as requisições que saem/ entram na aplicação;

Permite visualizar, editar e repetir as requisições.

Instalação no macOS (versão Community):

Baixar o instalador (arquivo dmg) no site oficial:
https://portswigger.net/burp/communitydownload

Ferramenta 6
Burp

2

Configurar o Burp para ouvir conexões de dispositivos externos:
Proxy -> Options -> Add
Bind to port: 1337 e All interfaces.

É preciso configurar a conexão do iPhone para passar pelo Proxy:
Na conexão Wi-fi: Info -> Configurar Proxy -> Manual -> Servidor: IP do macOS e porta 1337.

Ferramenta 6
Burp

3

Ainda no iPhone, é necessário instalar o Certificado do Burp para que seja possível visualizar o tráfego:

- Safari -> http://burp -> CA Certificate -> Instalar.
- Configurações -> Geral -> Sobre -> Certificados confiáveis -> marcar o PortSwigger CA como confiável.

Usar o app normalmente no iPhone.

Se ainda assim não for possível visualizar o tráfego no Burp, é porque o app usa SSL Pinning.

Ferramenta 7 - SSL Kill Switch 2

Desabilita o SSL Pinning dos apps instalados.
Instalação:

1. Baixar o arquivo .deb em:
https://github.com/nabla-c0d3/ssl-kill-switch2/releases

2. Copiar o arquivo .deb para o iPhone via scp.

3. dpkg -i arquivo_copiado.deb

4. killall -HUP SpringBoard (reinicia a interface gráfica).

5. Passará a existir um novo menu “SSL Kill Switch 2” em Ajustes.

Ferramenta 7 - SSL Kill Switch 2

2

Utilização:

1. Configurar o proxy do Burp no iPhone;

2. Ativar o SSL Kill Switch no menu;

3. Descobrir o número do processo da aplicação alvo:
   ps aux | grep -i <nome_do_app>

4. Matar o processo:
   kill -9 <número_do_processo>

5. Iniciar o app novamente.

Próximos passos

Definir uma metodologia de Pentest:

- OWASP Mobile Security Testing Guide e Checklist (https://github.com/OWASP/owasp-mstg);

- OWASP Mobile Top Ten (https://www.owasp.org/index.php/Mobile_Top_10_2016-Top_10).

Escrever um relatório com as vulnerabilidades:

- Deve conter as vulnerabilidades, sua criticidade e provas de conceito de como foram verificadas/exploradas.

(17) 3353-0833
amanda@**resh**.com.br
www.**resh**.com.br

resh®
Cyber Defense

/in/amandabsobrinho/

Amanda Barbosa Sobrinho
PGP KeyID: E9142CDE