Marcher Malware: Android Baking Trojan Targets Austria | Proofpoint US

By November 03, 2017 Proofpoint Staff

Published: 2017-11-03 · Archived: 2026-04-05 20:06:57 UTC

Overview

Credential phishing, banking Trojans, and credit card phishing schemes are common threats that we regularly observe both at scale and in more targeted attacks. However, Proofpoint researchers have recently observed phishing attacks that incorporate all of these elements in a single, multistep scheme involving the Marcher Android banking Trojan targeting customers of large Austrian banks. Attacks involving Marcher have become increasingly sophisticated, with documented cases involving multiple attack vectors and a variety of targeted financial services and communication platforms [1][2]. In this case, a threat actor has been targeting customers of Bank Austria, Raiffeisen Meine Bank, and Sparkasse since at least January 2017.

The attacks described here begin with a banking credential phishing scheme, followed by an attempt to trick the victim into installing Marcher, and finally with attempts to steal credit card information by the banking Trojan itself.

Analysis

Marcher is frequently distributed via SMS, but in this case, victims are presented with a link in an email. Oftentimes, the emailed link is a bit.ly shortened link, used to potentially evade detection. The link leads to a phishing page that asks for banking login credentials or an account number and PIN. Figure 1 shows one such landing page using stolen branding from Bank Austria.

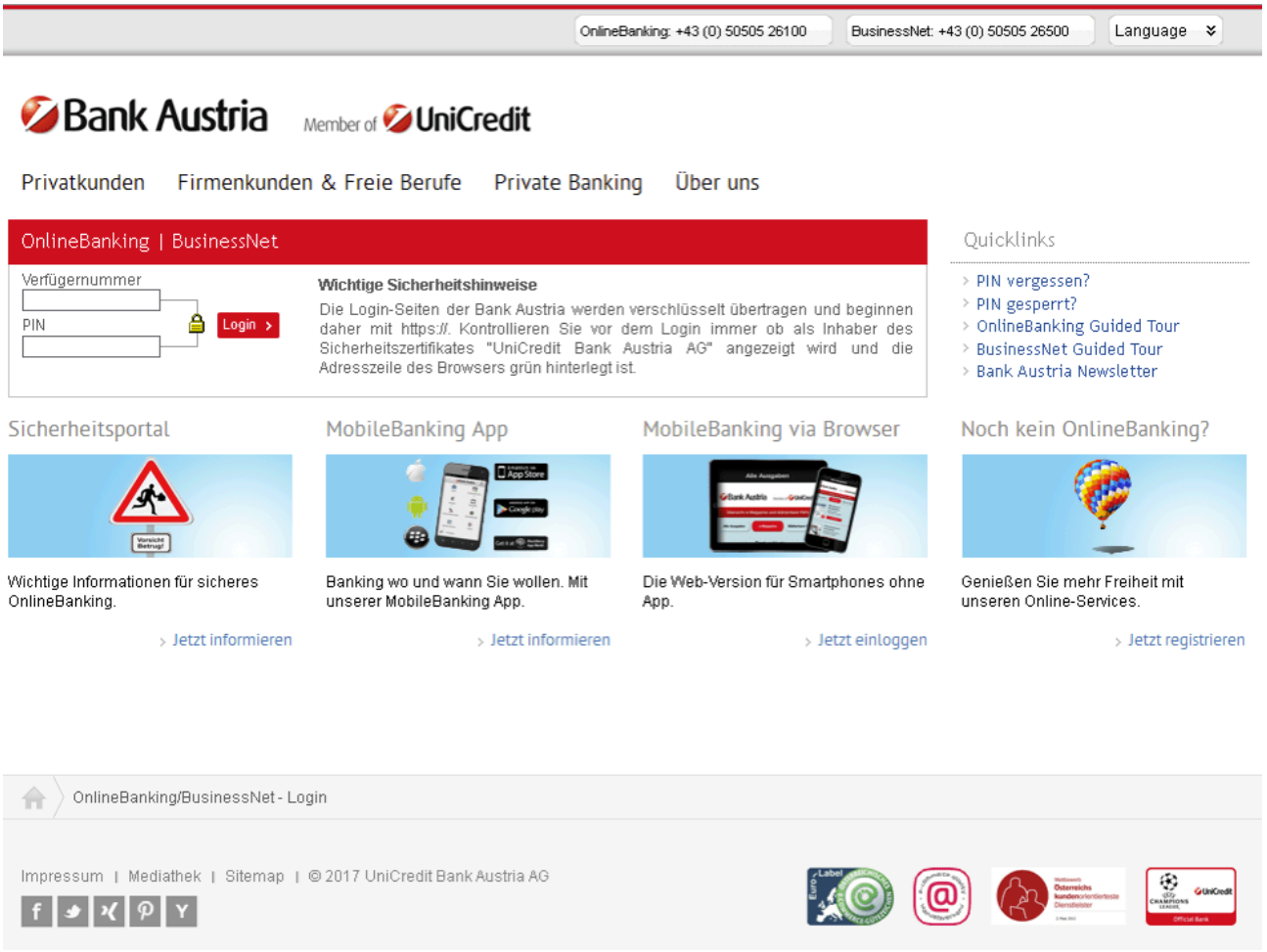

Figure 1: Landing page for phishing scheme asking for the victim's signatory number and PIN using stolen branding from Bank Austria

Because the actor delivered phishing links using the bit.ly URL shortener, we can access delivery statistics for this particular campaign. The link resolves to a URL designed to appear legitimate, with a canonical domain of sicher97140[.]info including the "bankaustria" brand.

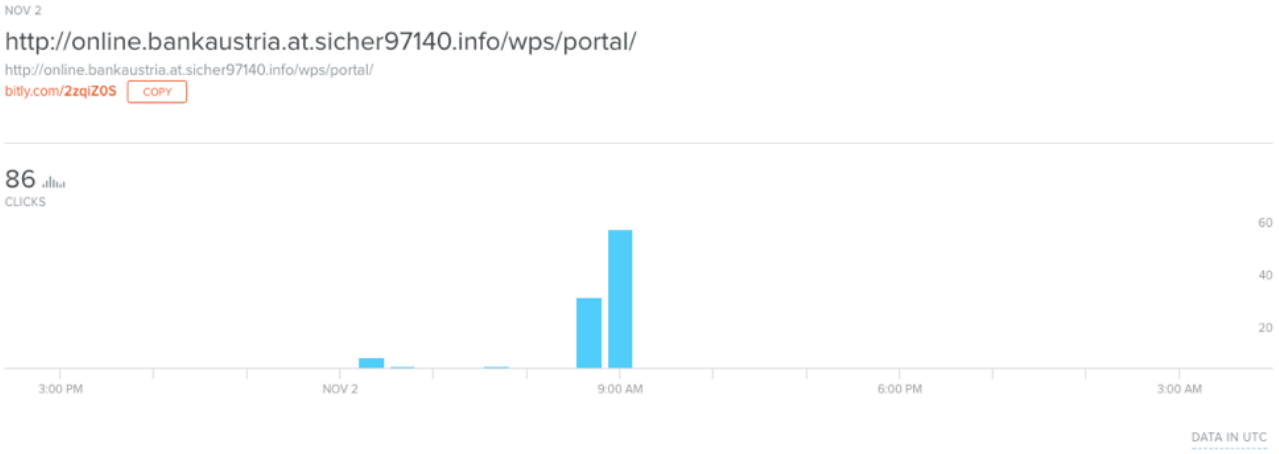

Figure 2: Bit.ly statistics for a phishing landing page targeting Bank Austria customers

The actor appears to have recently begun using ".top" top-level domains (TLDs) for their phishing landing pages and have implemented a consistent naming structure as shown below. Earlier this year, the actor used ".pw" TLDs while the Bank Austria scheme highlighted above used ".info". Some recent campaigns against other bank customers also used ".gdn" TLDs.

Other attacks on Bank Austria customers that we observed resolved to the following .top domains:

- Oct 23, 2017 hxxp://online.bankaustria.at.id8817062[.]top/
- Oct 23, 2017 hxxp://online.bankaustria.at.id8817461[.]top/
- Oct 23, 2017 hxxp://online.bankaustria.at.id8817465[.]top/
- Oct 23, 2017 hxxp://online.bankaustria.at.id8817466[.]top/
- Oct 23, 2017 hxxp://online.bankaustria.at.id8817469[.]top/
- Oct 17, 2017 hxxp://online.bankaustria.at.id58712[.]top/
- Oct 17, 2017 hxxp://online.bankaustria.at.id58717[.]top/
- Oct 17, 2017 hxxp://online.bankaustria.at.id58729[.]top/
- Oct 17, 2017 hxxp://online.bankaustria.at.id58729[.]top/
- Oct 17, 2017 hxxp://online.bankaustria.at.id87721[.]top/
- Oct 17, 2017 hxxp://online.bankaustria.at.id87726[.]top/

These permutations of TLDs and canonical domains incorporating the legitimate domain expected by the targeted banking customers exemplifies recent trends in social engineering by threat actors. Just as threat actors may use stolen branding in their email lures to trick potential victims, they reproduce a legitimate domain name in a fraudulent domain that is not controlled by the bank.

Once the victim enters their account information on the landing page, the phishing attack then requests that the user log in with their email address and phone number.

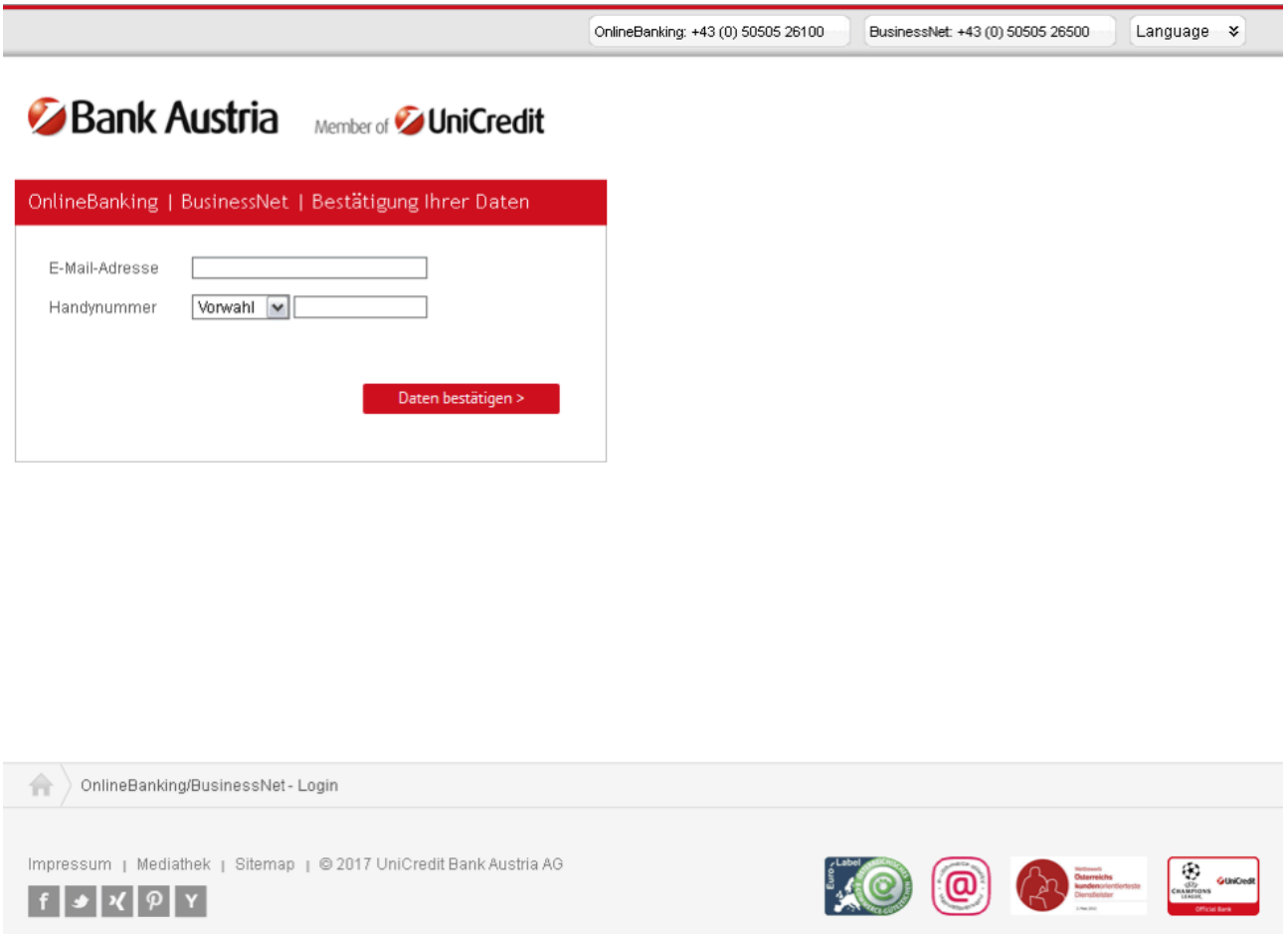

Figure 3: Step two of the credential phish asking for the victim's email address and phone number

Having stolen the victim's account and personal information, the scammer introduces a social engineering scheme, informing users that they currently do not have the "Bank Austria Security App" installed on their smartphone and must download it to proceed. Figure 4 shows the download prompt for this fake app; an English translation follows.

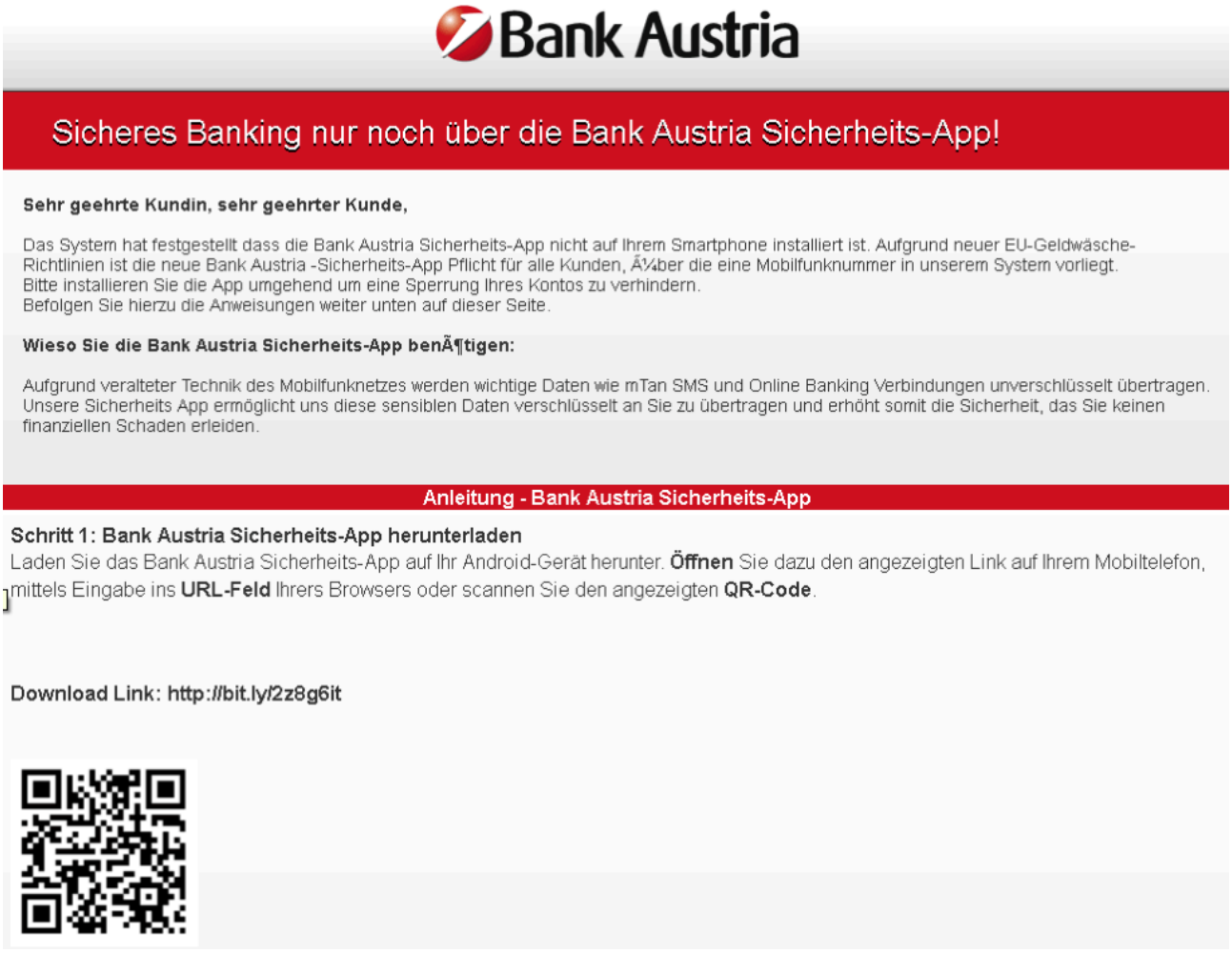

Figure 4: Alert prompting the victim to download an Android banking app (English translation below), with stolen branding and fraudulent copy

******Translation******

Dear Customer,

The system has detected that the Bank Austria Security App is not installed on your smartphone. Due to new EU money laundering guidelines, the new Bank Austria security app is mandatory for all customers who have a mobile phone number in our system.

Please install the app immediately to avoid blocking your account.

Follow the instructions at the bottom of this page.

Why you need the Bank Austria Security App:

Due to outdated technology of the mobile network important data such as mTan SMS and online banking connections are transmitted unencrypted.

Our security app allows us to transmit this sensitive data encrypted to you, thus increasing the security that you will not suffer any financial loss.

Step 1: Download Bank Austria Security App

Download the Bank Austria security app to your Android device. To do this, open the displayed link on your mobile phone by typing in the URL field of your browser or scan the displayed QR code.

******End translation******

The phishing template then presents additional instructions for installing the fake security application (Figure 5):

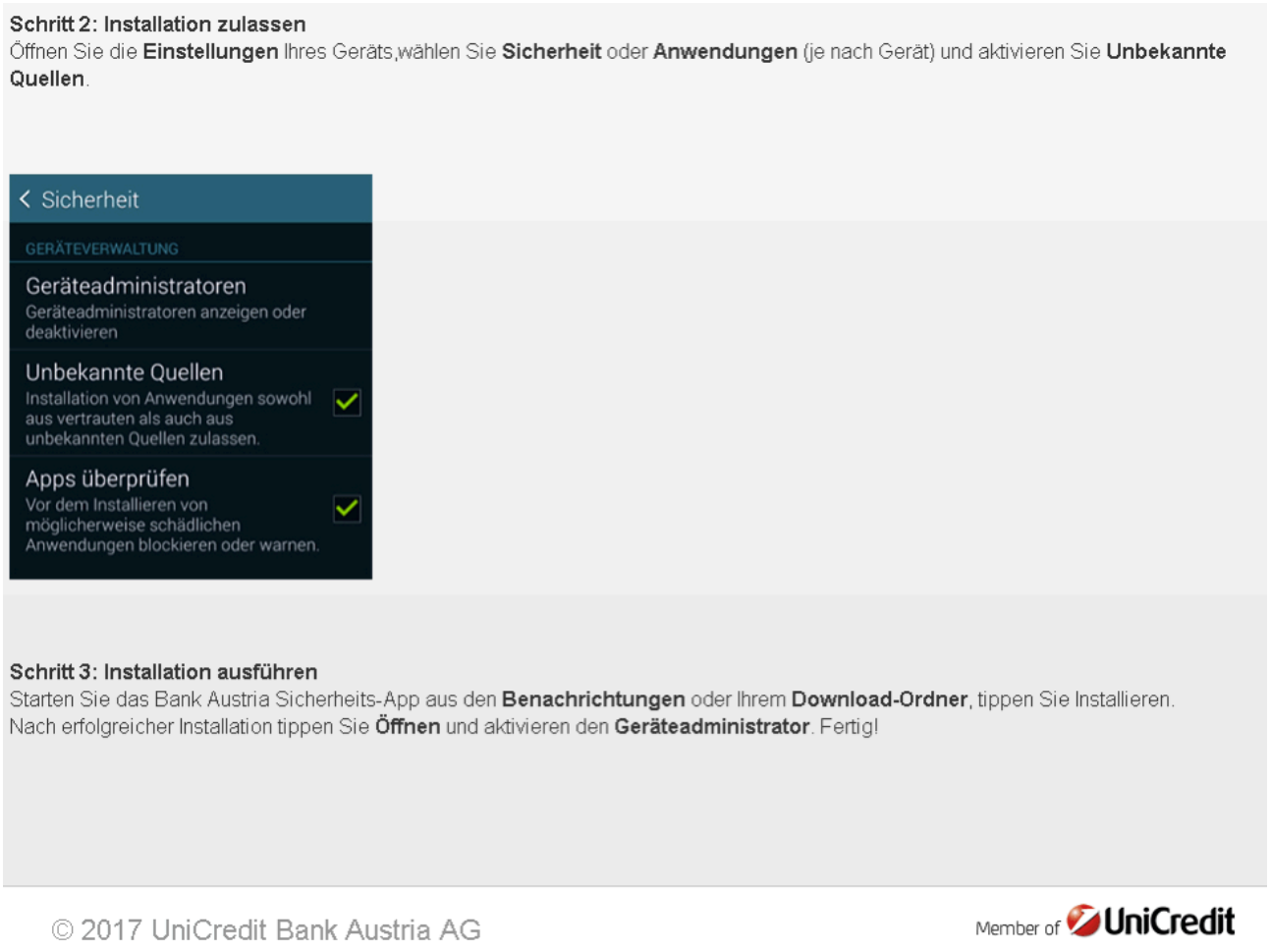

Figure 5: Additional instructions telling the victim to give the app the requested permissions (English translation below), with stolen branding and fraudulent copy

******Translation******

Step 2: Allow installation

Open your device's settings, select Security or Applications (depending on the device), and check Unknown sources.

Step 3: Run installation

Start the Bank Austria security app from the notifications or your download folder, tap Install.

After successful installation, tap Open and enable the device administrator. Finished!

******End translation******

Referring again to bit.ly, we can see click statistics for this campaign (Figure 6).

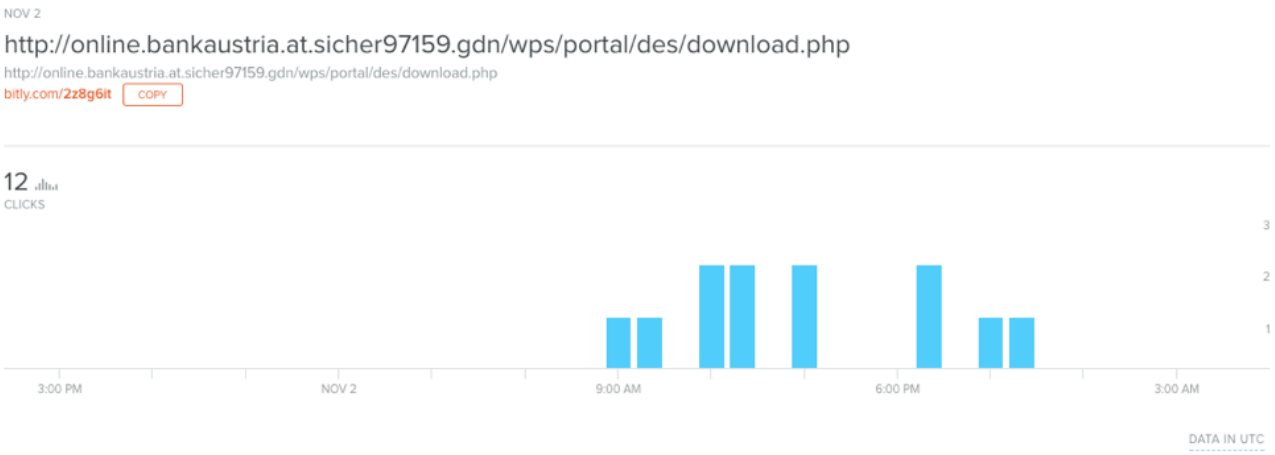

Figure 6: bit.ly statistics for the fake Bank Austria Android app download link

From this small sample, we see that 7% of visitors clicked through to download the application, which is actually a version of the Marcher banking Trojan named “BankAustria.apk”, continuing the fraudulent use of the bank’s branding to fool potential victims.

This sample is similar to those presented in other recent Marcher analyses [1][2].

This particular application is signed with a fake certificate:

Owner:
CN=Unknown, OU=Unknown, O=Unknown, L=Unknown, ST=Unknown, C=Unknown

Issuer
CN=Unknown, OU=Unknown, O=Unknown, L=Unknown, ST=Unknown, C=Unknown
Serial: 1c9157d7

Validity:
11/02/2017 00:16:46
03/20/2045 00:16:46

MD5 Hash: A8:55:46:32:15:A9:D5:95:A9:91:C2:91:77:5D:30:F6
SHA1 Hash: 32:17:E9:7E:06:FE:5D:84:BE:7C:14:0C:C6:2B:12:85:E7:03:9A:5F

The app requests extensive permissions during installation that enable a range of activities supported by the malware. Those permission shown in bold below are the most problematic:

- **Allows an application to write to external storage.**
- **Allows an application to read from external storage.**

- Allows an application to use SIP service.
- Allows an application to collect battery statistics
- **Allows an app to access precise location.**
- **Allows an application to receive SMS messages.**
- **Allows an application to send SMS messages.**
- **Allows an application to read SMS messages.**
- **Allows an application to write SMS messages.**
- Allows an application to initiate a phone call without going through the Dialer user interface for the user to confirm the call.
- Allows applications to access information about networks.
- **Allows applications to open network sockets.**
- **Allows an application to read the user's contacts data.**
- **Allows an application to read or write the system settings.**
- **Allows an application to force the device to lock**
- Allows applications to access information about Wi-Fi networks.
- Allows applications to change Wi-Fi connectivity state.
- Allows applications to change network connectivity state.

Analysis of the malware shows that it uses the common string obfuscation of character replacement (Figure 7):

```
public static final String ce = C0486n.a2119a("m**llR**e**llR**t**llR**h**llR**o**llR**d**llR**");
public static final String cf = C0486n.a2119a("s**llR**e**llR**n**llR**d**llR**_**llR**c**llR**a**llR**r**llR**d**llR**_**llR**n**llR**u**llR**m**]
public static final String cg = C0486n.a2119a("n**llR**u**llR**m**llR**b**llR**e**llR**r**llR**");
public static final String ch = C0486n.a2119a("m**llR**o**llR**n**llR**t**llR**h**llR**");
public static final String ci = C0486n.a2119a("y**llR**e**llR**a**llR**r**llR**");
public static final String cj = C0486n.a2119a("c**llR**v**llR**c**llR**");
public static final String ck = C0486n.a2119a("c**llR**o**llR**m**llR**.**llR**p**llR**a**llR**y**llR**p**llR**a**llR**l**llR**.**llR**a**llR**n**]
public static final String cl = C0486n.a2119a("c**llR**o**llR**m**llR**.**llR**a**llR**n**llR**d**llR**r**llR**o**llR**i**llR**d**llR**.**llR**v**]
public static final String cm = C0486n.a2119a("O**llR**K**llR**");
public static final String cn = C0486n.a2119a("c**llR**o**llR**m**llR**m**llR**a**llR**n**llR**d**llR**");
public static final String co = C0486n.a2119a("p**llR**a**llR**r**llR**a**llR**m**llR**s**llR**");
public static final String cp = C0486n.a2119a("t**llR**i**llR**m**llR**e**llR**s**llR**t**llR**a**llR**m**llR**p**llR**");
public static final String cq = C0486n.a2119a("c**llR**o**llR**m**llR**.**llR**d**llR**y**llR**n**llR**a**llR**m**llR**.**llR**");
public static final String cr = C0486n.a2119a("r**llR**u**llR**n**llR**");
public static final String cs = C0486n.a2119a("m**llR**a**llR**i**llR**n**llR**");
public static final String ct = C0486n.a2119a("I**llR**n**llR**t**llR**e**llR**n**llR**t**llR**F**llR**i**llR**l**llR**t**llR**e**llR**r**llR**");
public static final String cu = C0486n.a2119a("C**llR**o**llR**n**llR**t**llR**e**llR**x**llR**t**llR**");
```

Figure 7: Encoded Marcher Strings

```
public static final String ce = C0486n.m2119a("method");
public static final String cf = C0486n.m2119a("send_card_number");
public static final String cg = C0486n.m2119a("number");
public static final String ch = C0486n.m2119a("month");
public static final String ci = C0486n.m2119a("year");
public static final String cj = C0486n.m2119a("cvc");
public static final String ck = C0486n.m2119a("com.paypal.android.p2pmobile");
public static final String cl = C0486n.m2119a("com.android.vending");
public static final String cm = C0486n.m2119a("OK");
public static final String cn = C0486n.m2119a("command");
public static final String co = C0486n.m2119a("params");
public static final String cp = C0486n.m2119a("timestamp");
public static final String cq = C0486n.m2119a("com.dynam.");
public static final String cr = C0486n.m2119a("run");
public static final String cs = C0486n.m2119a("main");
public static final String ct = C0486n.m2119a("IntentFilter");
public static final String cu = C0486n.m2119a("Context");
```

Figure 8: Decoded Marcher Strings

As noted, the application requests extensive permissions during installation; Figure 9 shows the request to act as device administrator, a particular permission that should very rarely be granted to an app.

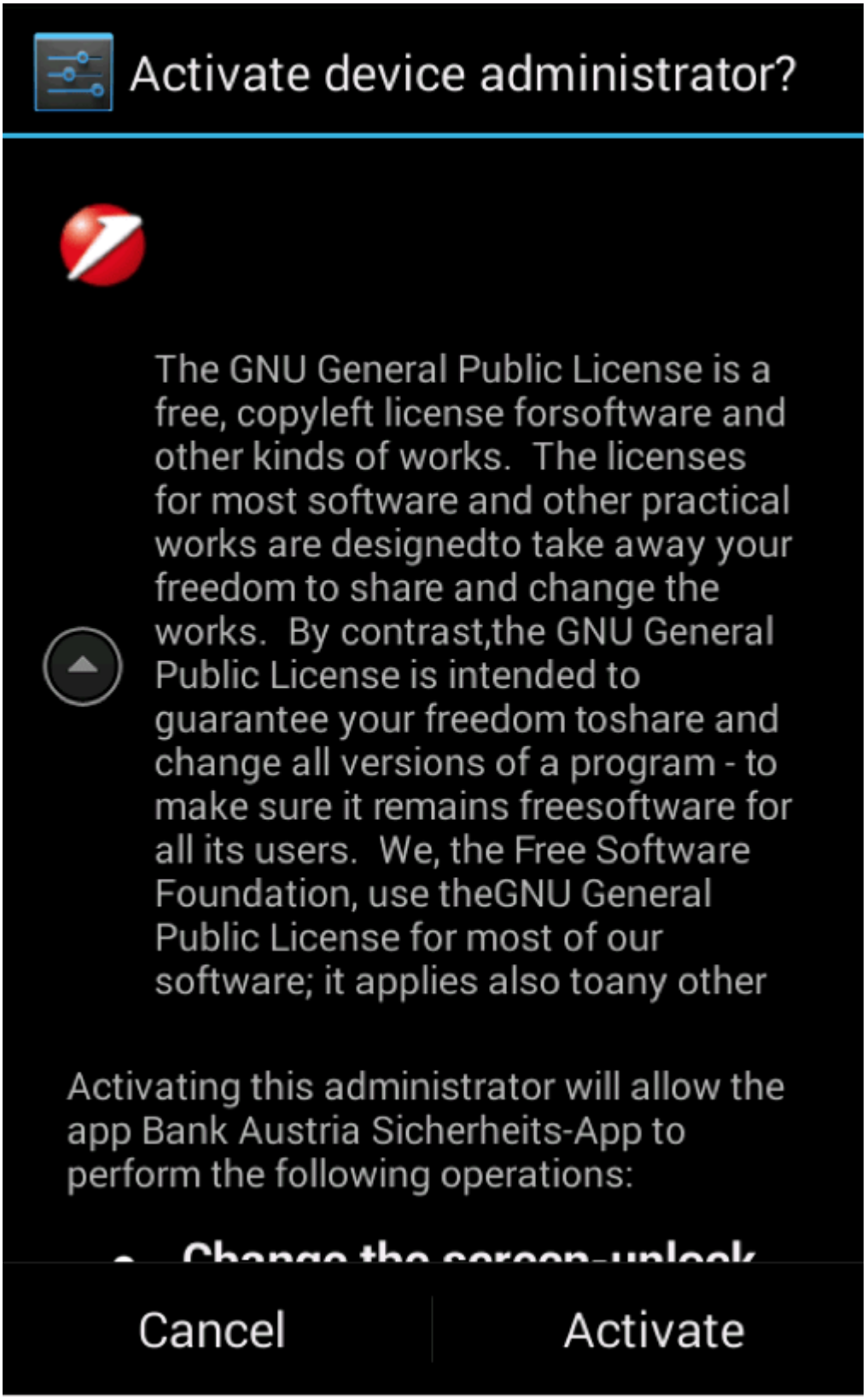

Figure 9: Prompt for application permissions upon installation

Figures 10 and 11 show the other permission screens for the app:

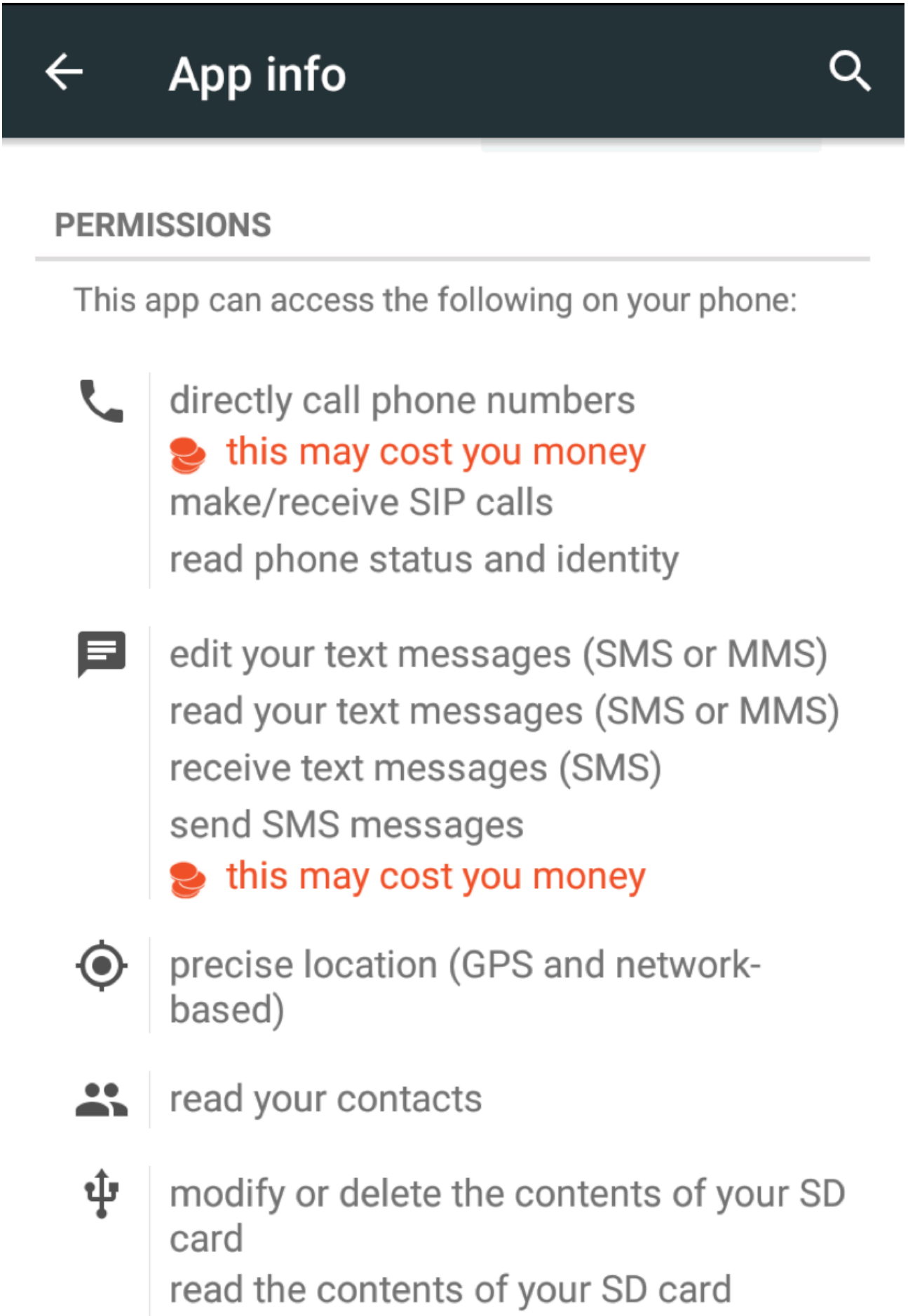
App info
PERMISSIONS
This app can access the following on your phone:
directly call phone numbers
this may cost you money
make/receive SIP calls
read phone status and identity
edit your text messages (SMS or MMS)
read your text messages (SMS or MMS)
receive text messages (SMS)
send SMS messages
this may cost you money
precise location (GPS and network-based)
read your contacts
modify or delete the contents of your SD card
read the contents of your SD card

Figure 10: Part 1 of the permission screen for the app

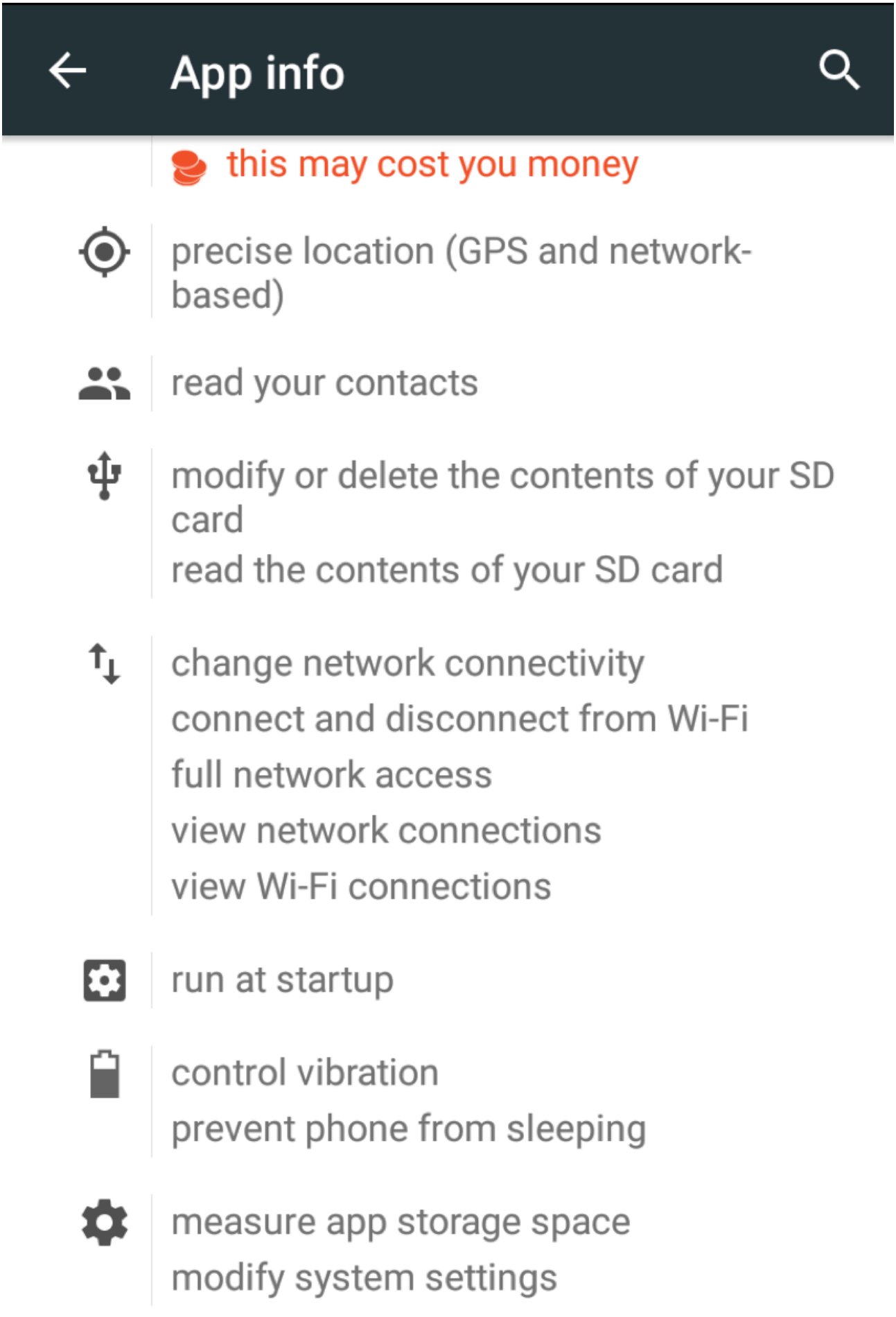
App info
this may cost you money
precise location (GPS and network-based)
read your contacts
modify or delete the contents of your SD card
read the contents of your SD card
change network connectivity
connect and disconnect from Wi-Fi
full network access
view network connections
view Wi-Fi connections
run at startup
control vibration
prevent phone from sleeping
measure app storage space
modify system settings

Figure 11: Part 2 of the permission screen for the app

Once installed the app will place a legitimate looking icon on the phone's home screen, again using branding stolen from the bank.

Figure 12: Fake Bank Austria Security application icon

In addition to operating as a banking Trojan, overlaying a legitimate banking app with an indistinguishable credential theft page, the malware also asks for credit card information from the user when they open applications such as the Google Play store.

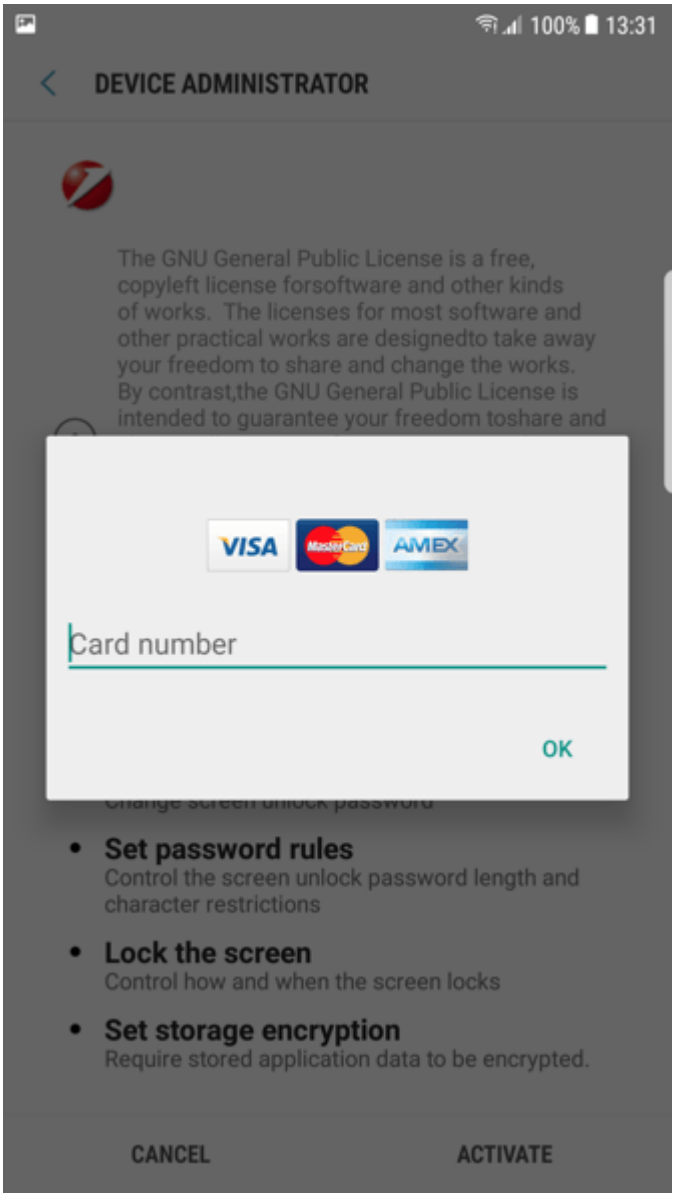

Figure 13: Popup asking for a credit card number

The application also supports stealing credit card verification information (Figures 14 and 15).

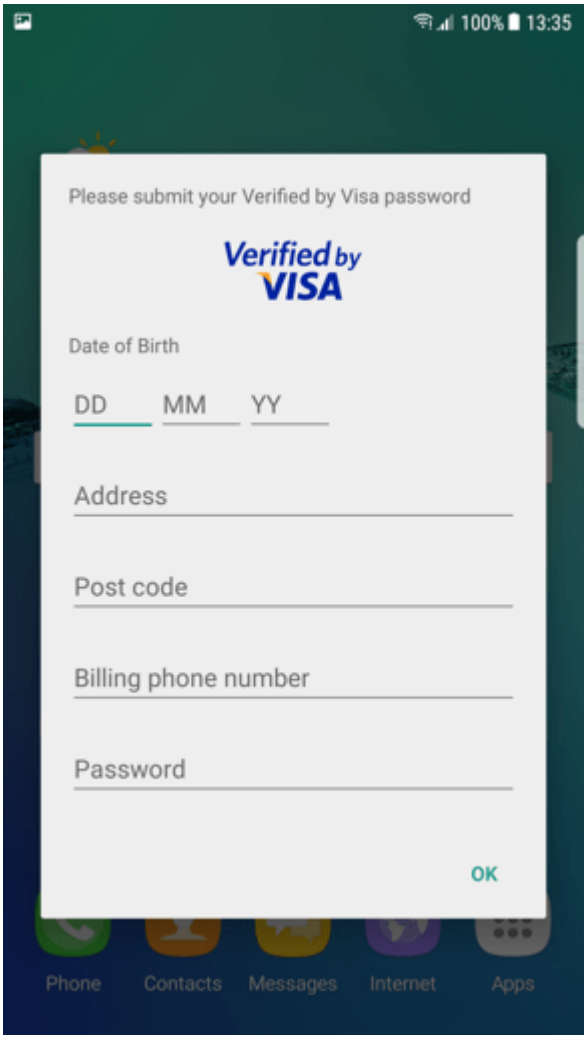

Figure 14: Information theft via fake credit card verification using stolen branding

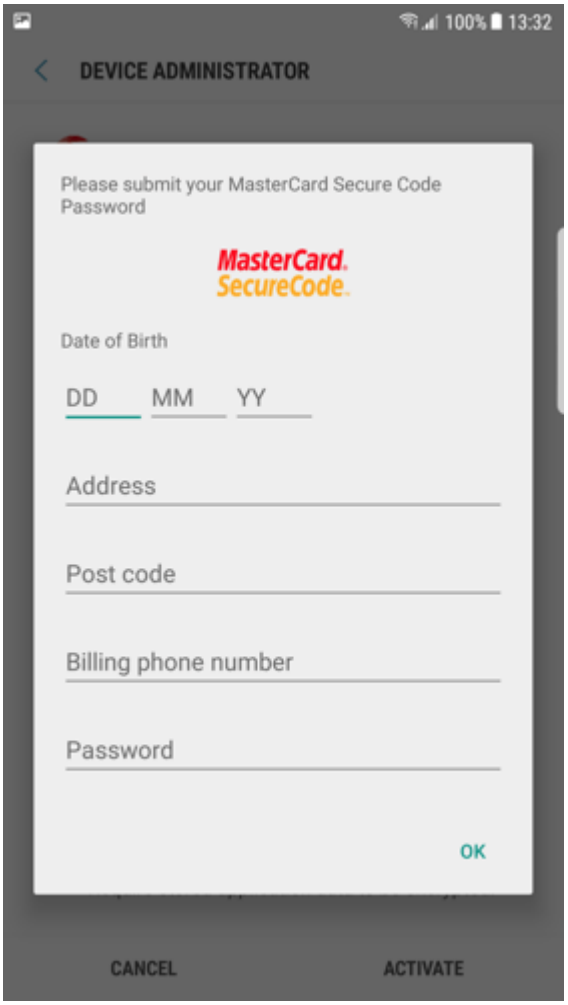

Figure 15: Information theft via fake credit card verification using stolen branding

Some of the campaigns appear to have a wider reach based on bit.ly statistics like this one from October 13, 2017:

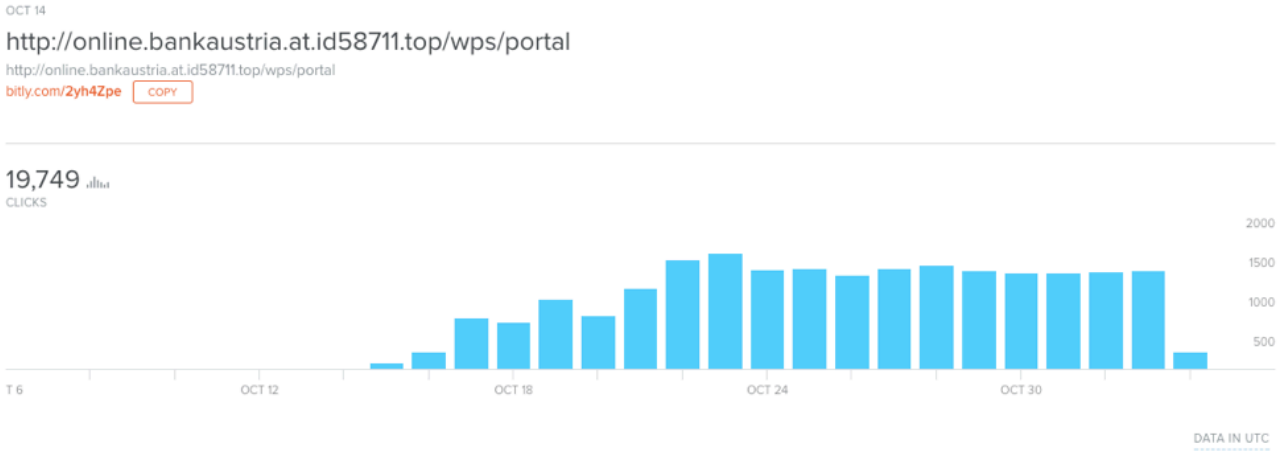

Figure 16: bit.ly statistics for an October 13, 2017 campaign

Over several days during the last three months, Proofpoint researchers observed campaigns using similar techniques targeting the banking customers of Raffeisen and Sparkasse. A review of the bit.ly statistics for these

campaigns shows that they were at least as effective in driving end-user clicks as the Bank Austria campaign analyzed above.

Conclusion

As our computing increasingly crosses multiple screens, we should expect to see threats extending across mobile and desktop environments. Moreover, as we use mobile devices to access the web and phishing templates extend to mobile environments, we should expect to see a greater variety of integrated threats like the scheme we detail here. As on the desktop, mobile users need to be wary of installing applications from outside of legitimate app stores and sources and be on the lookout for bogus banking sites that ask for more information than users would normally provide on legitimate sites. Unusual domains, the use of URL shorteners, and solicitations that do not come from verifiable sources are also red flags for potential phishing and malware.

References

[1] https://clientsidedetection.com/marcher.html

[2] https://www.mysonicwall.com/sonicalert/searchresults.aspx?ev=article&id=1047

Indicators of Compromise (IOCs)

IOC	IOC Type	Description
47.91.92[.]60	IP	Phish Landing
49.51.37[.]177	IP	Phish Landing
49.51.37[.]247	IP	Phish Landing
47.254.128[.]80	IP	Phish Landing
8dfc01cfed545651e3cf73437ab748dc	MD5	Marcher - Analyzed Sample
185.188.204[.]16	IP	Marcher C&C

ET and ETPRO Suricata/Snort Signatures

2024943 Raiffeisen Phishing Domain Nov 03 2017

2024944	Sparkasse Phishing Domain Nov 03 2017
2024946	BankAustria Phishing Domain Nov 03 2017
2024947	Successful Raiffeisen Phish Nov 03 2017
2024948	Successful Sparkasse Phish Nov 03 2017
2024949	Successful BankAustria Phish Nov 03 2017
2024950	Android Marcher Trojan Download - Raiffeisen Bank Targeting (set)
2024951	Android Marcher Trojan Download - Sparkasse Bank Targeting (set)
2024952	Android Marcher Trojan Download - BankAustria Targeting (set)
2024953	Android Marcher Trojan Download - Austrian Bank Targeting
2828513	Trojan-Banker.AndroidOS.Marcher Domain Request in SNI
2828514	Trojan-Banker.AndroidOS.Marcher Domain Request in SNI 2
2828515	Trojan-Banker.AndroidOS.Marcher Domain Request in SNI 3
2828516	Trojan-Banker.AndroidOS.Marcher Domain Request in SNI 4
2828517	Trojan-Banker.AndroidOS.Marcher Domain Request in SNI 5
2828518	Trojan-Banker.AndroidOS.Marcher Domain Request in SNI 6
2828519	Trojan-Banker.AndroidOS.Marcher Domain Request in SNI 7
2828520	Trojan-Banker.AndroidOS.Marcher Domain Request in SNI 8
2828521	Trojan-Banker.AndroidOS.Marcher Domain Request in SNI 9
2828524	Trojan-Banker.AndroidOS.Marcher.z DNS Lookup 1
2828525	Trojan-Banker.AndroidOS.Marcher.z DNS Lookup 2
2828526	Trojan-Banker.AndroidOS.Marcher.z DNS Lookup 3
2828527	Trojan-Banker.AndroidOS.Marcher.z DNS Lookup 4
2828528	Trojan-Banker.AndroidOS.Marcher.z DNS Lookup 5
2828529	Trojan-Banker.AndroidOS.Marcher.z DNS Lookup 6
2828530	Trojan-Banker.AndroidOS.Marcher.z DNS Lookup 7
2828531	Trojan-Banker.AndroidOS.Marcher.z DNS Lookup 8
