The missing link in the chain? Android network analysis

Rowland Yu
Senior Threat Researcher II

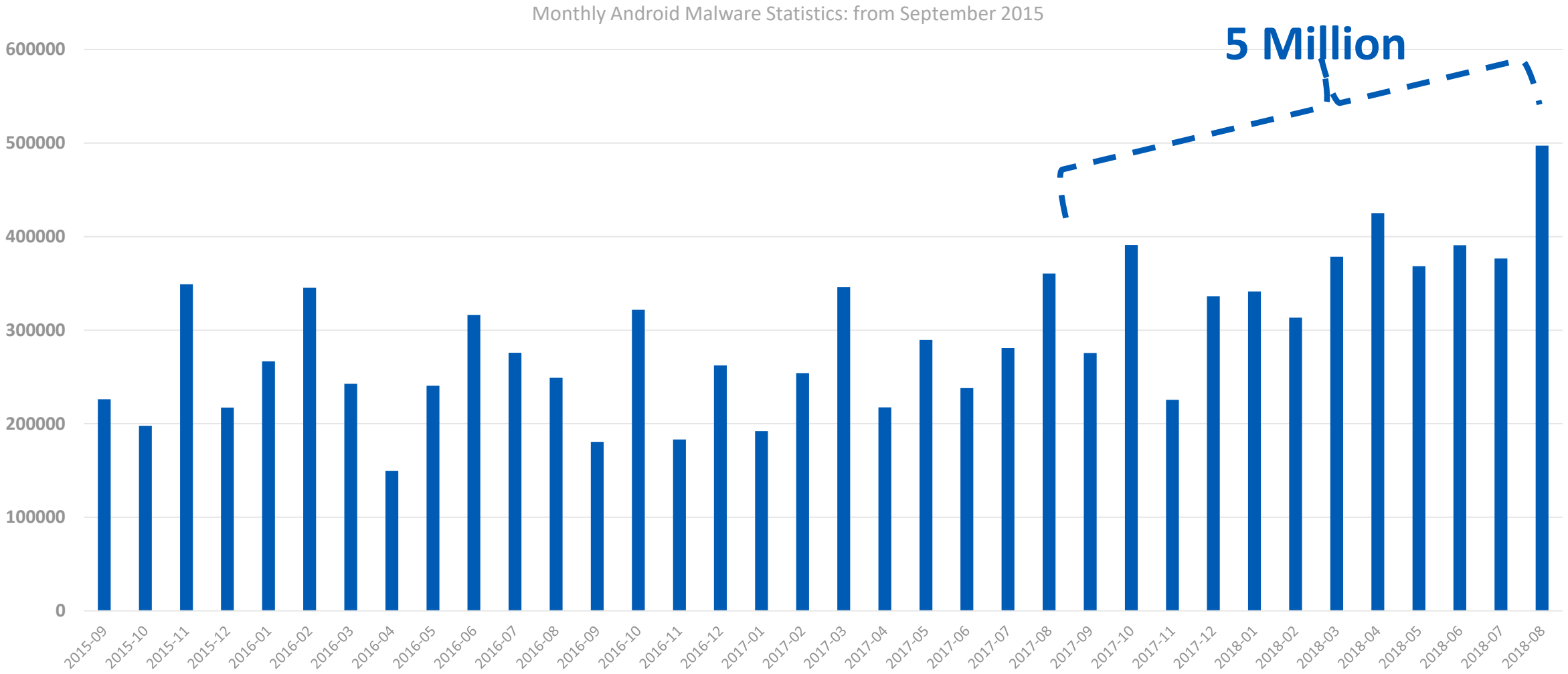
vb 2018 MONTREAL 3 – 5 October 2018

SOPHOS

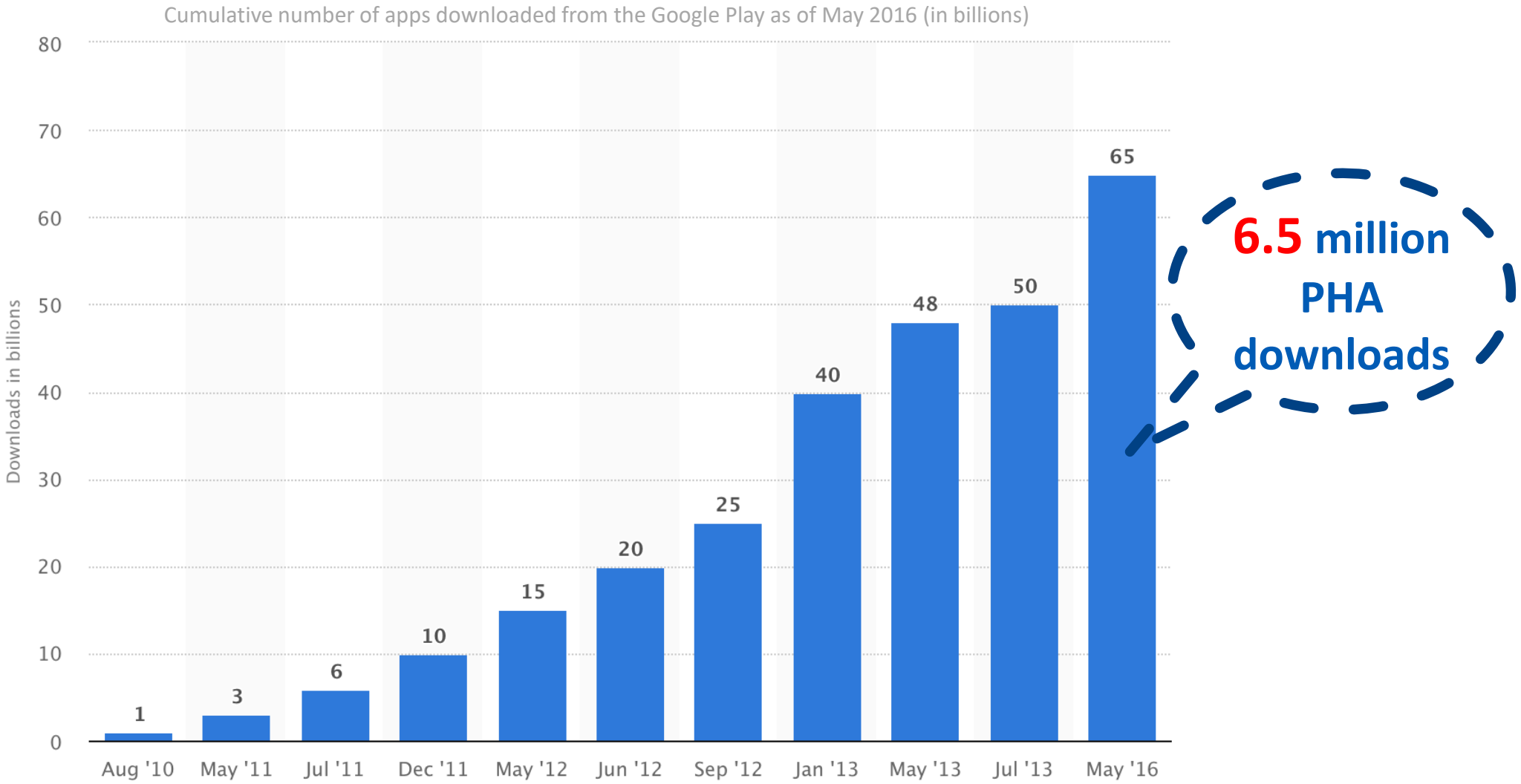
Facts

SOPHOS

Facts – Monthly Sample Doubled from 2015 to 2018



Facts – Google Play PHA: 1 out of every 10,000 downloads



Source : Android Developers Blog & Statista 2018

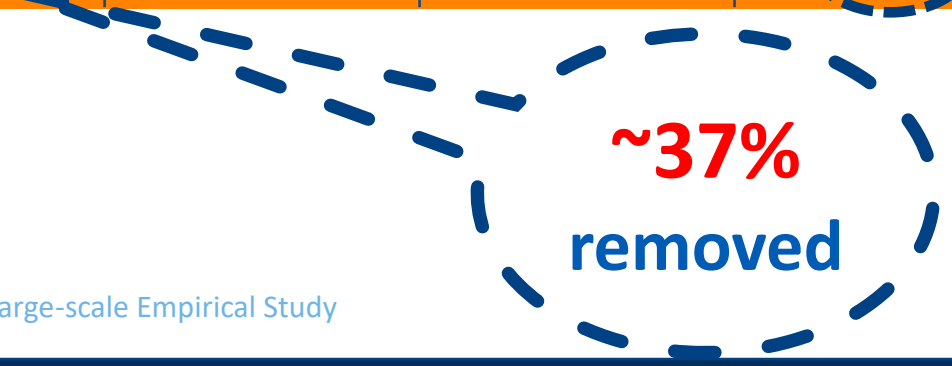
Facts – Possible PHAs Removed by Google Play

Details of the collected apps

	Apps	Free	Paid	Installs	Developers
Google Play 2015	1,502,180	1,278,078	224,103	89.9B	338,670
Google Play 2017	**2,144,733**	2012,893	131,840	193.5B	541,105

Details of the removed apps

	Apps	Free	Paid	Installs	Developers
Removed Apps (Step1)	795,374	684,835	110,539	14.7B	186,595
Removed Apps (Step2)	**791,138**	**681,241**	**109,897**	**14.2B**	184,852

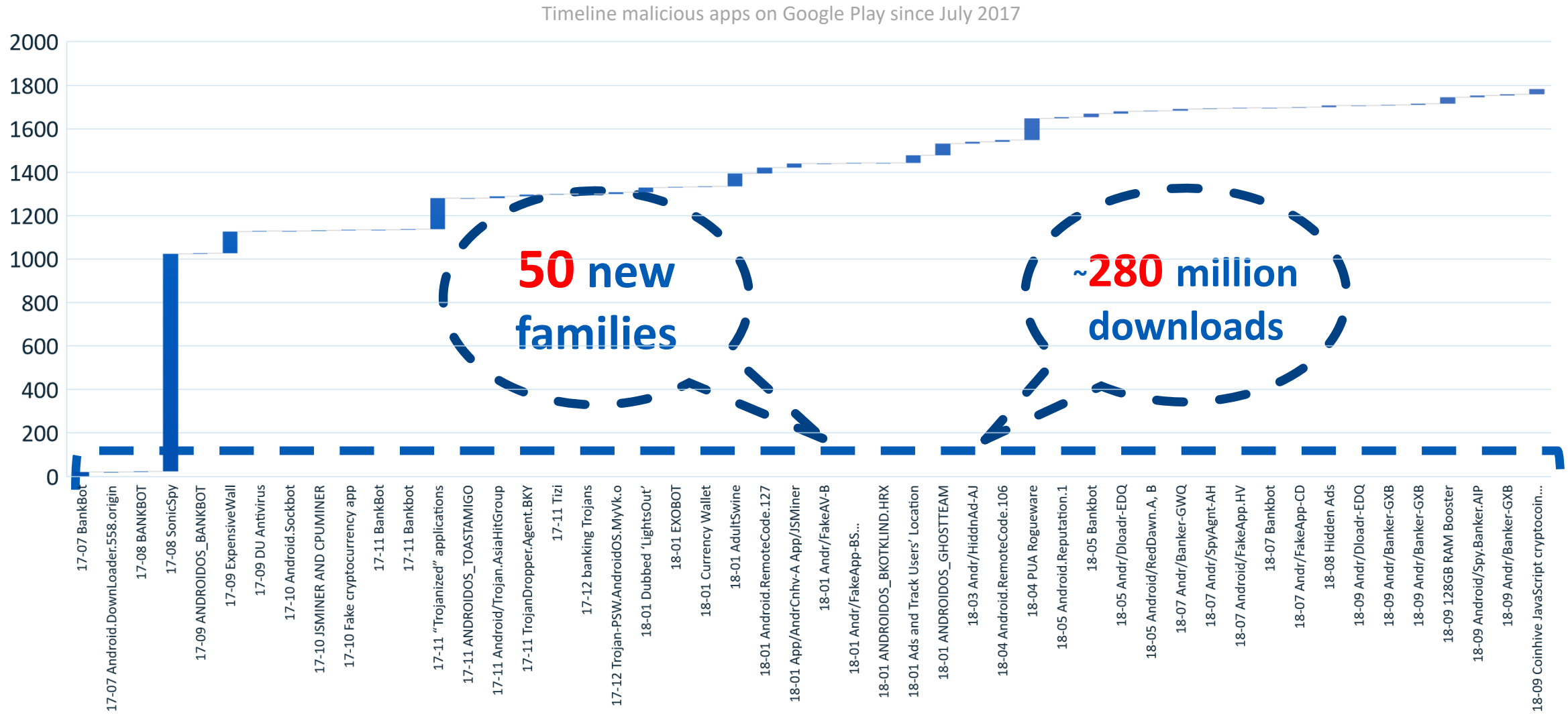


Source : Why are Android Apps Removed From Google Play? A Large-scale Empirical Study

Facts – Malware on Google Play Reported by Researchers

Family	Apps	Downloads
2017-07 BankBot	20	100
2017-07 Android.DownLoader.558.origin	1	1000000
2017-08 BANKBOT	2	3000
2017-08 SonicSpy	1000	100
2017-09 ANDROIDOS_BANKBOT	5	100
2017-09 ExpensiveWall	100	13503039
2017-09 DU Antivirus	1	30000000
2017-10 Android.Sockbot	1	100
2017-10 JSMINER AND CPUMINER	3	30000
2017-10 Fake cryptocurrency trading app	1	300
2017-11 BankBot	2	3000
2017-11 Bankbot	1	100

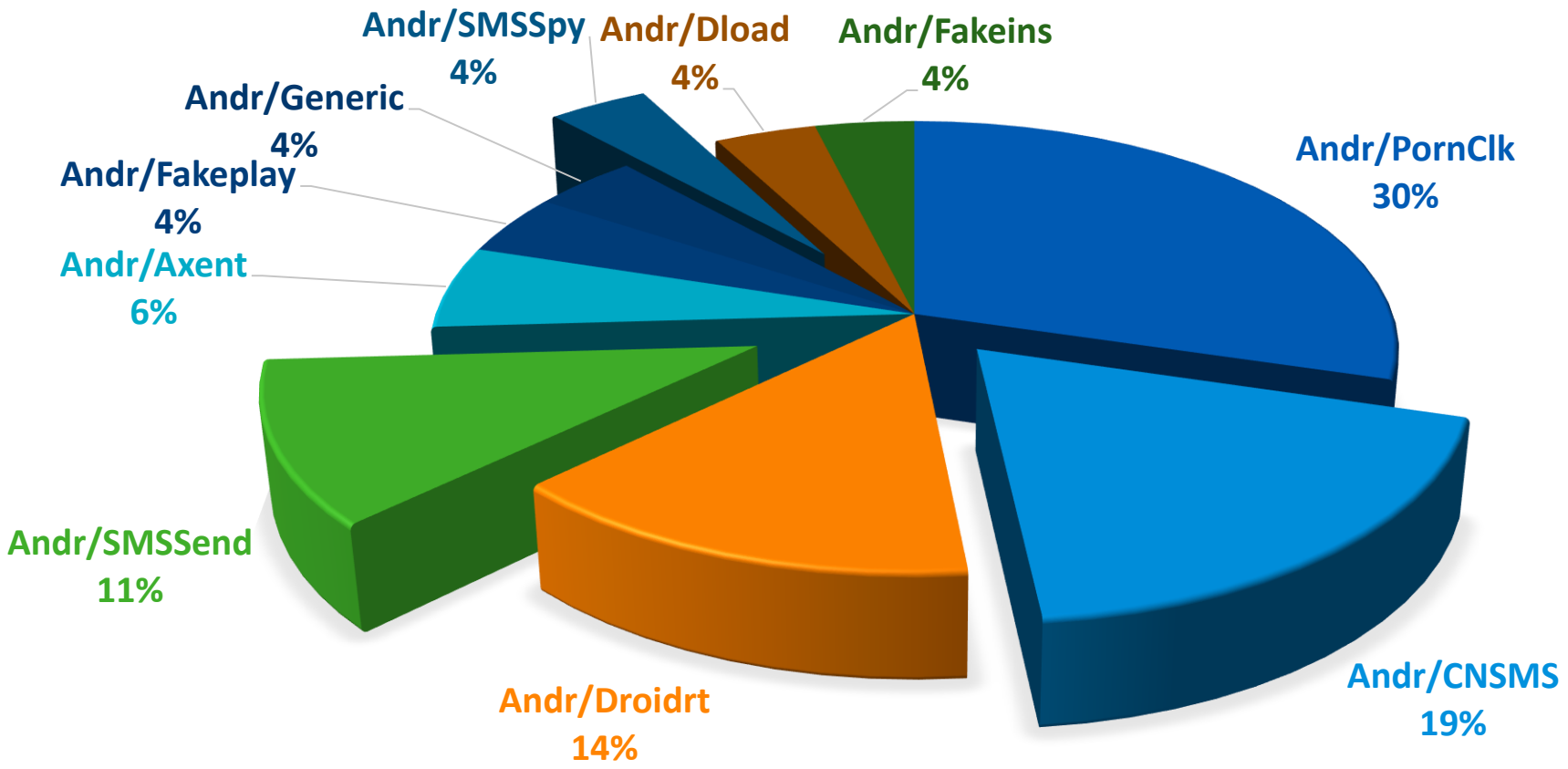
Facts – Malware on Google Play Reported by Researchers



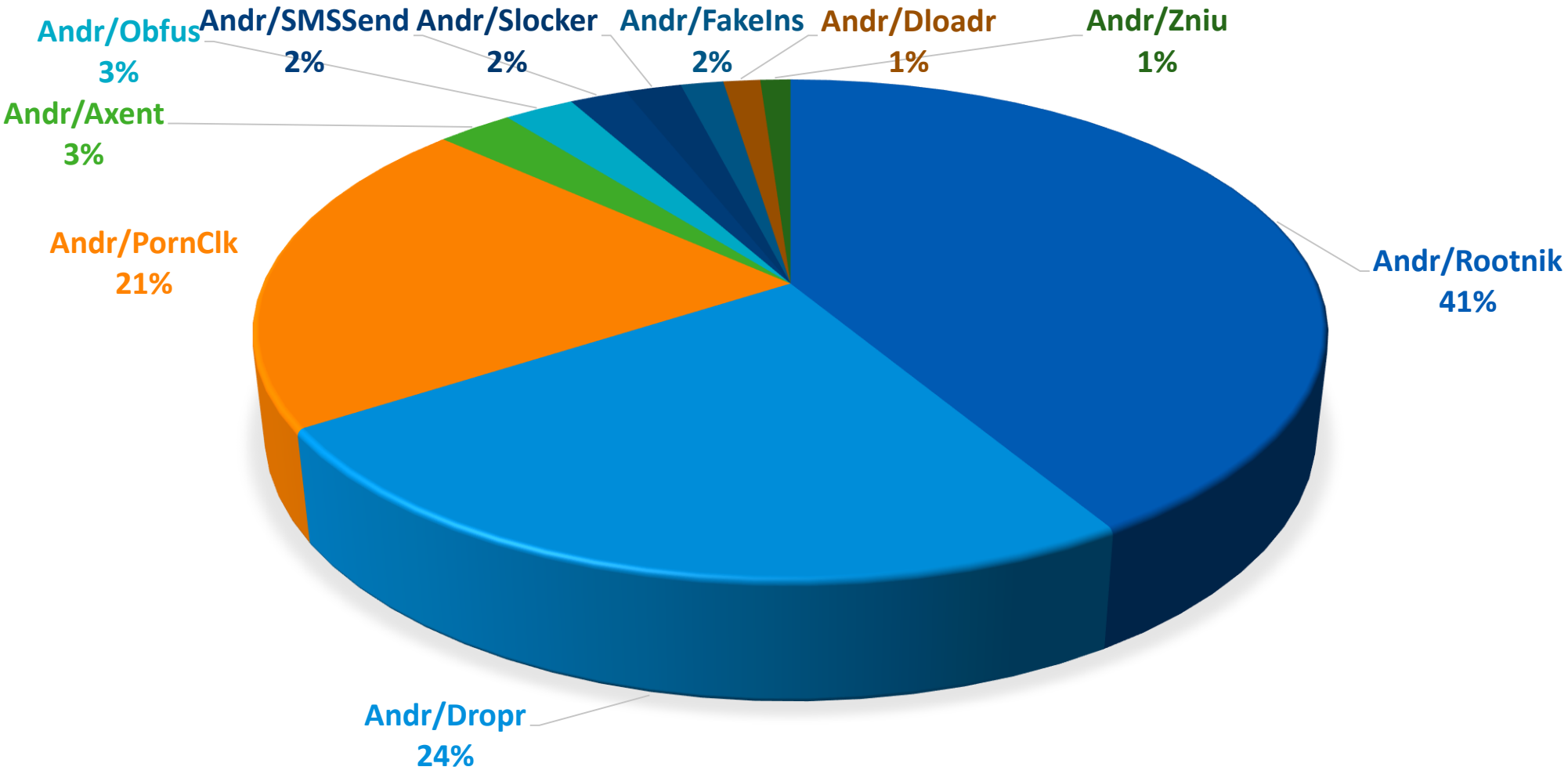
Evolution

SOPHOS

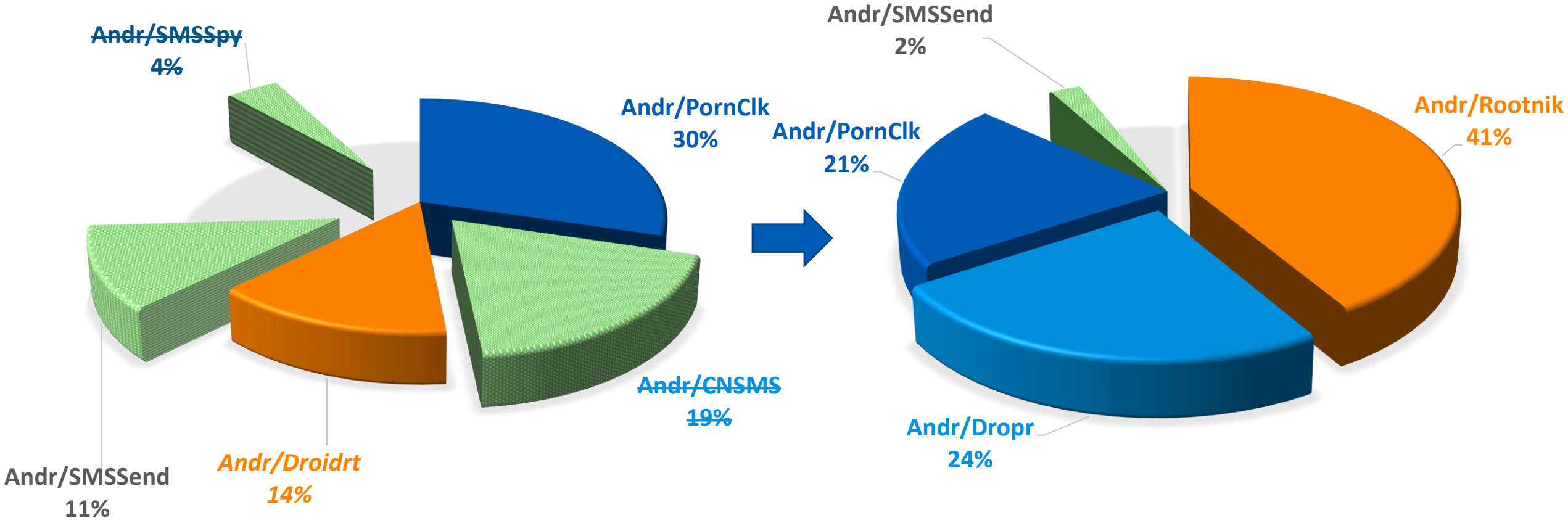
Top 10 Android Threats: 2016



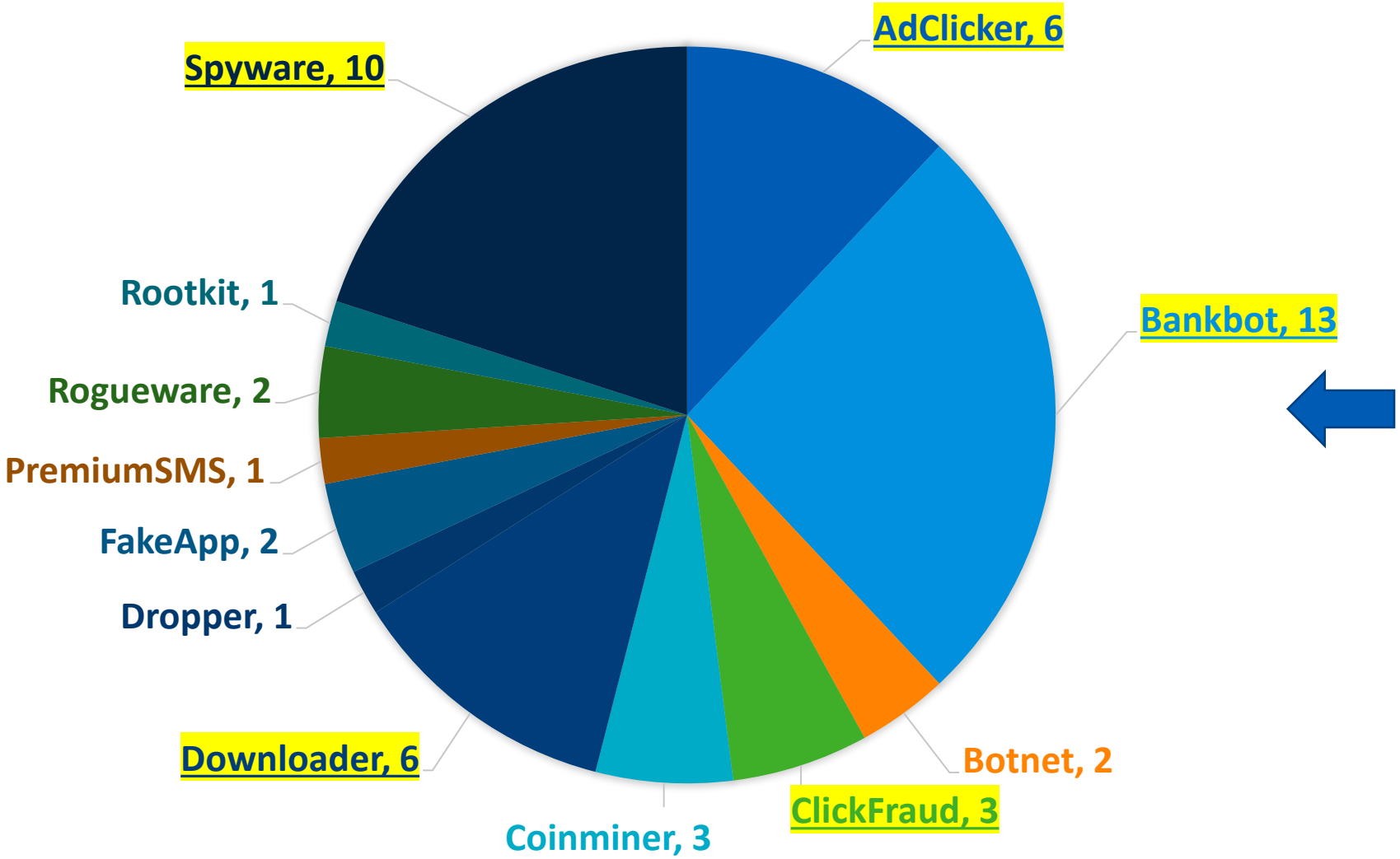
Top 10 Android Threats: September/2017 - August/2018



Top 10 Android Threats Evolution

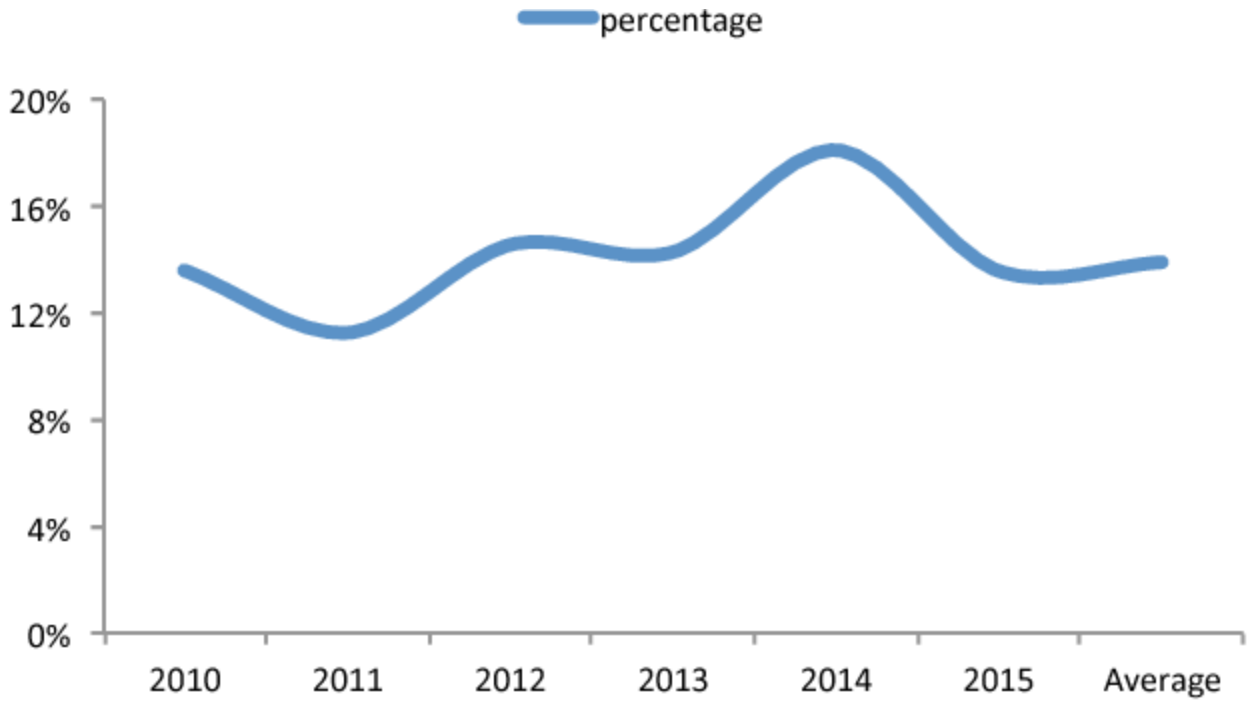


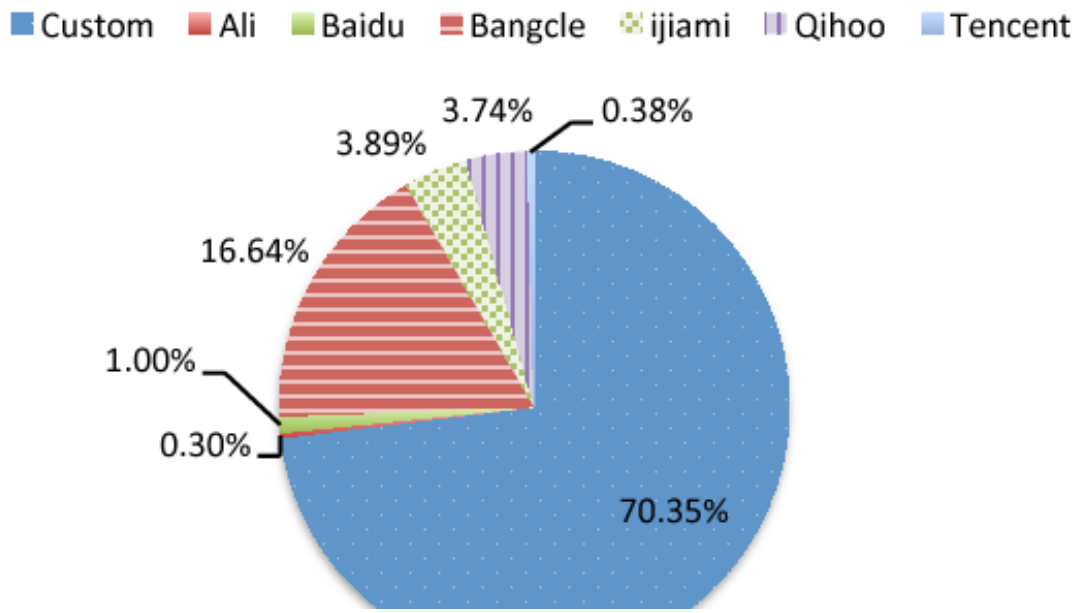
Google Play Malware Categories



Category	Timestamp	Num of Apps	Installs
AdClicker	2018-01	1	50 - 100
AdClicker	2018-01	60+	3,000,000 - 7,000,000
AdClicker	2018-01	27	4,500,000
AdClicker	2018-03	7	910,000
AdClicker	2018-04	8+	6,500,000
AdClicker	2018-08	8	50,000
Bankbot	2017-07	20+	N/A
Bankbot	2017-08	2	1,000 - 5,000
Bankbot	2017-09	5	N/A
Bankbot	2017-11	2	1,000 - 5,000
Bankbot	2017-11	1	N/A
Bankbot	2017-12	1	1,000 - 5,000
Bankbot	2018-01	23	1,585,000 - 7,565,000
Bankbot	2018-07	10	5000+
Bankbot	2018-07	1	N/A
Bankbot	2018-07	3	300+
Bankbot	2018-09	3	1500+
Bankbot	2018-09	5	5000+
Botnet	2017-10	1	N/A
Botnet	2018-01	3	1,000 - 5,000
ClickFraud	2017-09	100+	5,904,511 - 21,101,567
ClickFraud	2017-11	144	4,200,000 - 17,400,000
ClickFraud	2018-05	7	3,000+
Coinminer	2017-10	3	10,000 - 50,000
Coinminer	2018-01	19	120,000 - 570,000
Coinminer	2018-09	25	120,000
Downloader	2017-07	1	1,000,000
Downloader	2017-11	1+	100,000 - 500,000
Downloader	2017-11	6	N/A
Downloader	2018-05	15	400,000
Downloader	2018-05	10	1,200,000
Downloader	2018-09	1	500,000+
Dropper	2017-11	8	2,500 - 12,000
FakeApp	2018-01	1	10,000 - 50,000
FakeApp	2018-01	1	N/A
PremiumSMS	2018-01	1	1,000-5,000
Rogueware	2018-04	100+	100,000,000 - 250,000,000
Rogueware	2018-09	30+	500,000+
Rootkit	2017-11	3	N/A
Spyware	2017-08	1000+	N/A
Spyware	2017-09	1	10,000,000 - 50,000,000
Spyware	2017-10	1	100 - 500
Spyware	2017-12	7	10,000 - 100,000
Spyware	2018-01	36	10,000,000 - 50,000,000 +
Spyware	2018-01	53	50,000 - 250,000 +
Spyware	2018-05	3	N/A
Spyware	2018-07	2	200+
Spyware	2018-07	1	100+

Packers





Packer in malware: Yearly distribution vs Packer distribution.

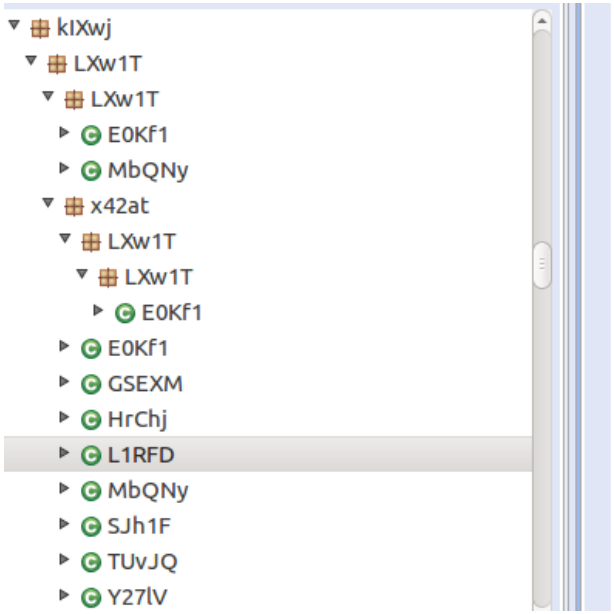
Source : Things You May Not Know About Android (Un)Packers: A Systematic Study based on Whole-System Emulation

Obfuscators

25% obfuscated on Google Play

Name	License	Obfuscation							Other				
		Package name	Class name	Method name	Field name	Overloading	Debug Data	Annotations	String Enc.	Class Enc.	Optimization	Minimization	Watermarking
Allatori[1,†]	$290	●	●	●	●	●	●	○	●	○	●	●	●
DashO[†]	On request	●	●	●	●	●	●	●	●	○	●	○	●
DexGuard[2,†]	On request	●	●	●	●	●	●	●	●	●	●	●	○
DexProtector	$800	●	●	●	●	○	○	○	●	●	○	○	○
GuardIT	On request	●	●	●	●	●	●	○	●	●	○	○	○
Jack[2,†]	Free	●	●	●	●	○	●	●	○	○	●	●	○
ProGuard[†]	Free	●	●	●	●	●	●	●	○	○	●	●	○
ReDex[2,†]	Free	●	●	●	●	●	●	●	○	○	●	●	○
yGuard[†]	Free	●	●	●	●	●	●	○	○	○	○	●	○

[1] Multiple obfuscation patterns, default can be detected
[2] Mirrors ProGuard's obfuscation with same configuration format
[†] Obfuscation features (partially) detected by OBFUSCAN




```
    return v0;
}

private static MbQNy Vhjf3(Context arg4) {
    MbQNy v2 = new MbQNy();
    v2.ywQui(Build.MANUFACTURER);
    v2.dz90j(Build$VERSION.RELEASE);
    v2.ywQui(Build$VERSION.SDK_INT);
    DisplayMetrics v1 = new DisplayMetrics();
    arg4.getSystemService("window").getDefaultDisplay().getMetrics(v
    v2.STsiU(v1.widthPixels);
    v2.dz90j(v1.heightPixels);
    Object v0 = arg4.getSystemService("phone");
    String v1_1 = ((TelephonyManager)v0).getDeviceId();
    if(com.kIXwj.LXw1T.x42at.MbQNy.ywQui(v1_1)) {
        v1_1 = "";
    }

    v2.RNvSP(v1_1);
    v1_1 = com.kIXwj.LXw1T.x42at.MbQNy.ywQui(arg4);
    if(com.kIXwj.LXw1T.x42at.MbQNy.ywQui(v1_1)) {
        v1_1 = "";
    }

    v2.STsiU(v1_1);
    v2.RNvSP(L1RFD.ywQui(arg4, ((TelephonyManager)v0), v1_1));
    v1_1 = ((TelephonyManager)v0).getSimSerialNumber();
    if(com.kIXwj.LXw1T.x42at.MbQNy.ywQui(v1_1)) {
```

Source : A Large Scale Investigation of Obfuscation Use in Google Play (2018)

“Modern Android malware takes full advantage of **the internet to execute remote tasks**. Dealing with packed or obfuscated Android apps by using existing well-known analysis tools like JEB, apktool and Radare2 **still remains a very challenging task**.”

State of the Art

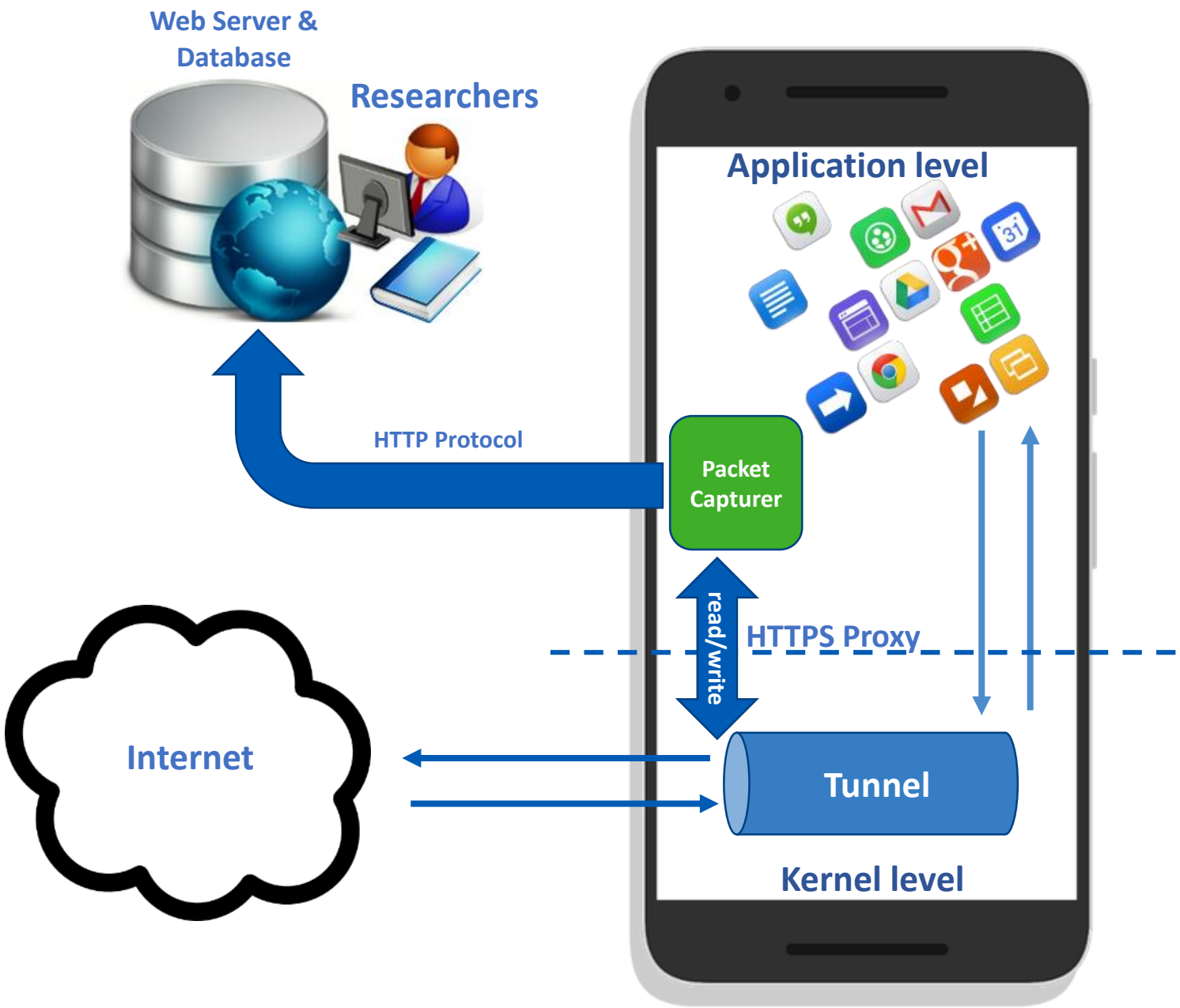
SOPHOS

Existing Packet Capture Tools

	External database	Forensics friendly	App based capture	SSL decryption	Open source	Exportable pcap/file	Android version
Android *tcpdump*	×	×	×	×	√	√	-
bitshark	×	×	×	×	×	√	2.3.3 and up
tPacketCapture	×	×	√	×	×	√	4.0 and up
SSL Packet Capture	×	×	√	√	×	√	4.0 and up
Wicap 2. Sniffer	×	×	×	×	×	√	4.0 and up
Root Packet Capture	×	×	×	×	√	√	-
CharlesProxy SSL Traffic Debug	√	√	×	√	×	-	4.0 and up

Architecture

SOPHOS


Web Server &
Database
Researchers
Application level
HTTP Protocol
Packet
Capturer
read/write
HTTPS Proxy
Internet
Tunnel
Kernel level

Application-Based VPN Service

- Android 4.0+
- Built-in VPN solution
- No root privileges

Application-Based VPN Service

- Declare a VPN in AnroidManifest.xml

```
<service Android:name=".MyVpnService"
   Android:permission="Android.permission.BIND_VPN_SERVICE">
   <intent-filter>
      <action Android:name="Android.net.VpnService" />
   </intent-filter>

</service>
```

- Prepare a VPN

```
Intent intent = VpnService.prepare(this);
```

- Supply those parameters to a VpnService.Builder and establish a VPN interface

```
Builder builder = new Builder();
builder.addAddress("10.8.0.1", 32);
builder.addRoute("0.0.0.0", 0);
builder.setSession("MyVpnService");
this.mParcelFileDescriptor = builder.establish();
```

Trusted CA (Certificate Authority) Certificate

- As long as the appropriate Certificate Authority (CA) certificate is installed in the trust credentials, many applications can't tell the difference between a connection to the original host and the intercepting proxy.

- Starting from Android 4.0, users can now easily add their own 'user' certificate.

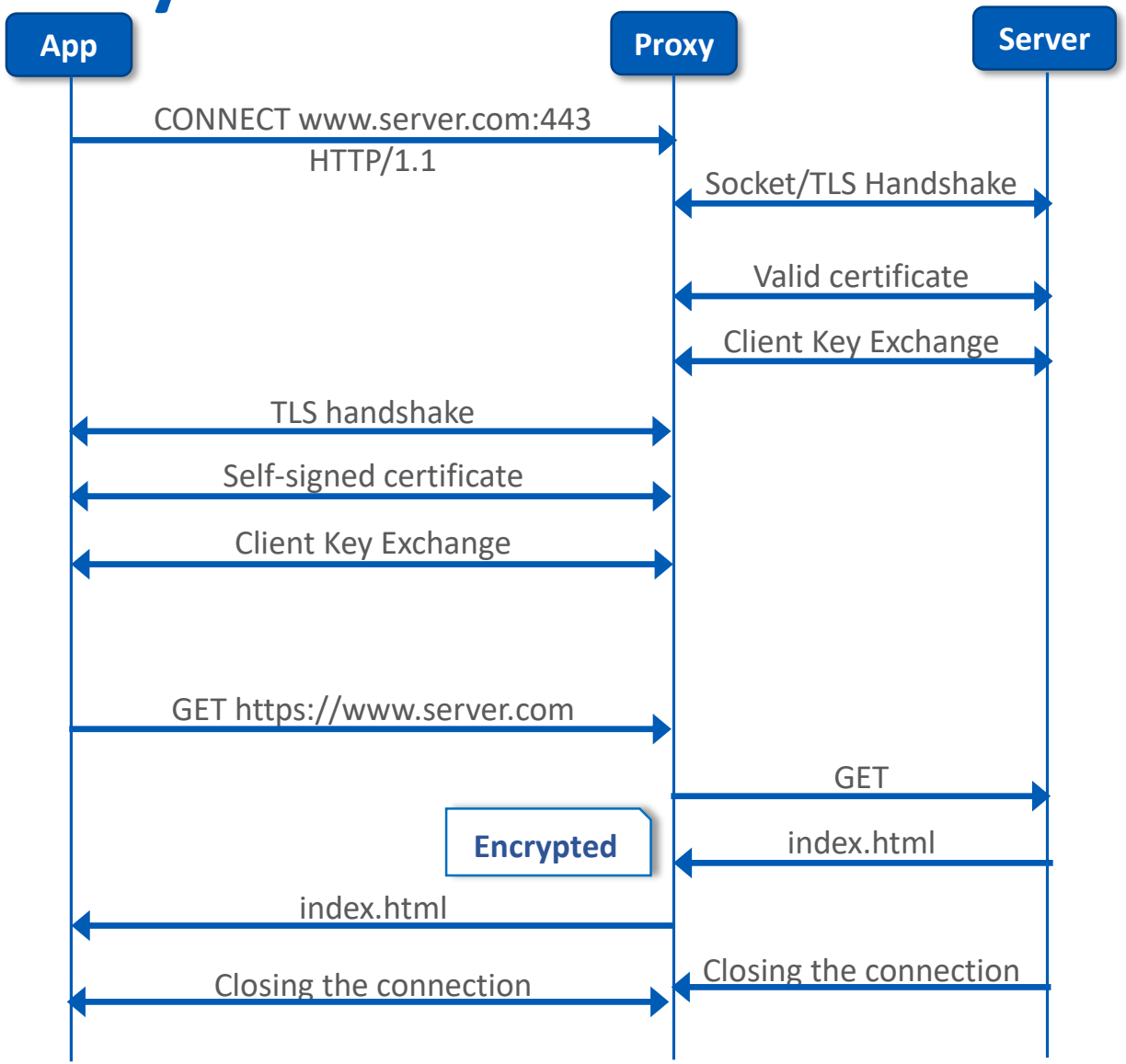
Trusted CA (Certificate Authority) Certificate

```
/** Generates a certificate for {@code hostName} containing {@code keyPair}'s
 * public key, signed by {@code keyPair}'s private key.
 */
@SuppressWarnings("deprecation") // use the old Bouncy Castle APIs to reduce dependencies.
public X509Certificate selfSignedCertificate(KeyPair keyPair)
      throws GeneralSecurityException {
   X509V3CertificateGenerator generator = new X509V3CertificateGenerator();
   X500Principal issuer = new X500Principal("CN=CA Certificate");
   X500Principal subject = new X500Principal("CN=CA Certificate");
   generator.setSerialNumber(new BigInteger(BigInteger.valueOf(System.currentTimeMillis())));
   generator.setIssuerDN(issuer);
   generator.setNotBefore(new Date(notBefore));
   generator.setNotAfter(new Date(notAfter));
   generator.setSubjectDN(subject);
   generator.setPublicKey(keyPair.getPublic());
   generator.setSignatureAlgorithm("SHA256WithRSAEncryption");
   return generator.generateX509Certificate(keyPair.getPrivate(), "BC");
}
/** This method will launch an intent to install the key chain */
Intent installIntent = KeyChain.createInstallIntent();
installIntent.putExtra("CERT", keystore.getCert());
installIntent.putExtra("name", "Capture CA Certificate");
startActivityForResult(installIntent, INSTALL_KEYCHAIN_CODE);
```

HTTPS MiTM Proxy

- A simple proxy is created to act as man-in-the-middle (MiTM) HTTPS communication.
- The proxy allows for monitoring, modifying, and (if necessary) replaying of HTTP / HTTPS traffic that passes through it.

HTTPS MiTM Proxy



Demo

SOPHOS

Conclusion

SOPHOS

- Strong evidences show that Android threats increasingly take advantage of network traffic to execute additional payloads.

- Both legitimate and malicious apps are leveraging packing and obfuscating mechanisms to protect themselves against reverse engineer.

- Existing packet capture tools, such as Android *tcpdump* and SSL Packet Capture, have been released for network analysis in the last few years, but have some disadvantages in different aspects.

	External database	Forensics friendly	App based capture	SSL decryption	Open source	Exportable pcap/file	Android version
New Packet Capturer	√	√	√	√	-	√	4.0 – 7.0

Limitations

- It is impossible for Android VPNService to capture all packets (like loopback packet data.
- Some apps such as Google Chrome may utilize secure techniques (i.e. Certificate Pinning) to avoid SSL proxy eavesdropping.
- The solution may not work very well in Emulator.
- From Android N onwards, it gets a little harder.

Q&A

Rowland.Yu@Sophos.com.au

SOPHOS

Security made simple.