



Your Number Is Up!

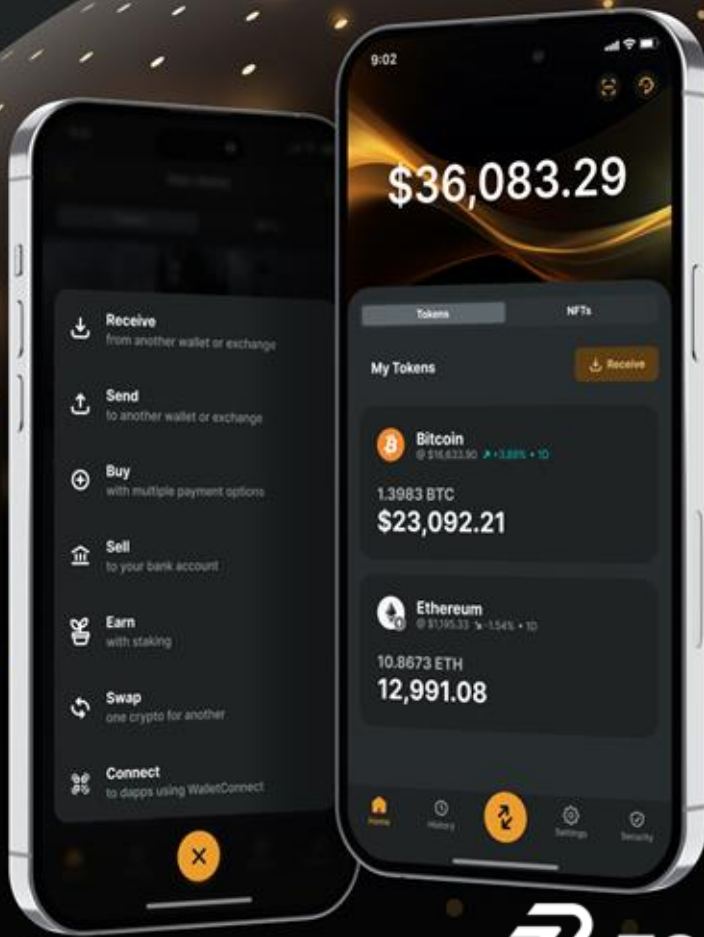
When 3.5 Billion Strangers Can Exploit Your WhatsApp Devices

👋 Hi, I'm Tal Be'ery

- Co-Founder, CTO @ Zengo (Acquired by eToro)
- 20+ years cyber security
- 10th time Black Hat Speaker
- [@talbeerysec](https://twitter.com/talbeerysec)



eToro



zengo

Agenda

- **WhatsApp security intro**
- **WhatsApp: Focusing on rogue clients' attacks**
- **WhatsApp End-to-End Encryption (E2EE) + privacy leaks**
- **New research tool!**
- **Attacks + Demos: reconnaissance, fingerprinting, device pinpointing**
- **Solutions**
- **Q&A**

WhatsApp Security

Intro

WHATSAPP IS THE IM GORILLA

A Giant in Instant Messaging

▲ 201 bm
▲ 126%

▼ 150%

OVER 3 BILLION

MONTHLY ACTIVE USERS WORLDWIDE



100 MILLION

MONTHLY ACTIVE USERS IN THE US



RANKED MOST USED

MOBILE MESSENGER APP IN THE WORLD



**MORE THAN
100 BILLION**
MESSAGES SENT
EACH DAY



**MORE THAN
7 BILLION**
VOICE MESSAGES
SENT DAILY





I rob banks because that's where the
money is.

— *Willie Sutton* —

AZ QUOTES



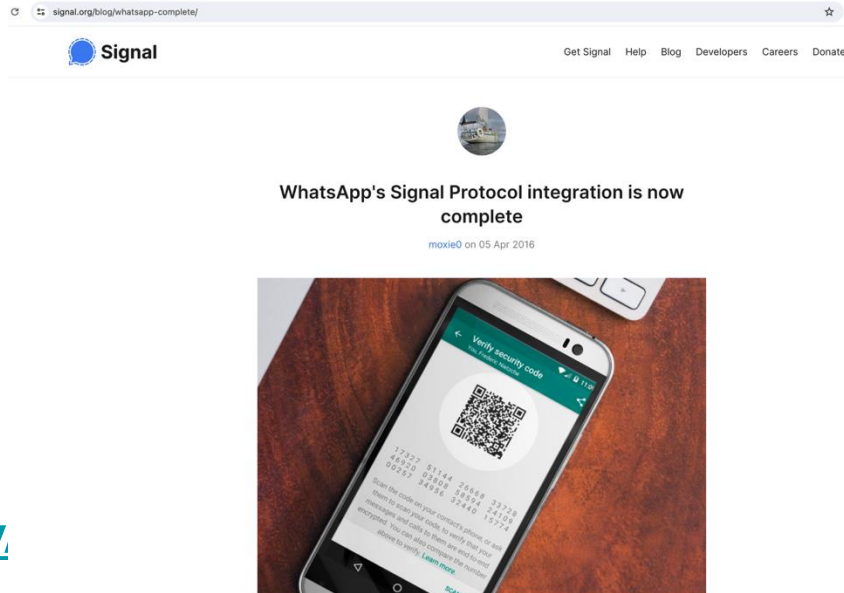
I hack WhatsApp because that's
where the data is.

— *Willie Sutton* —



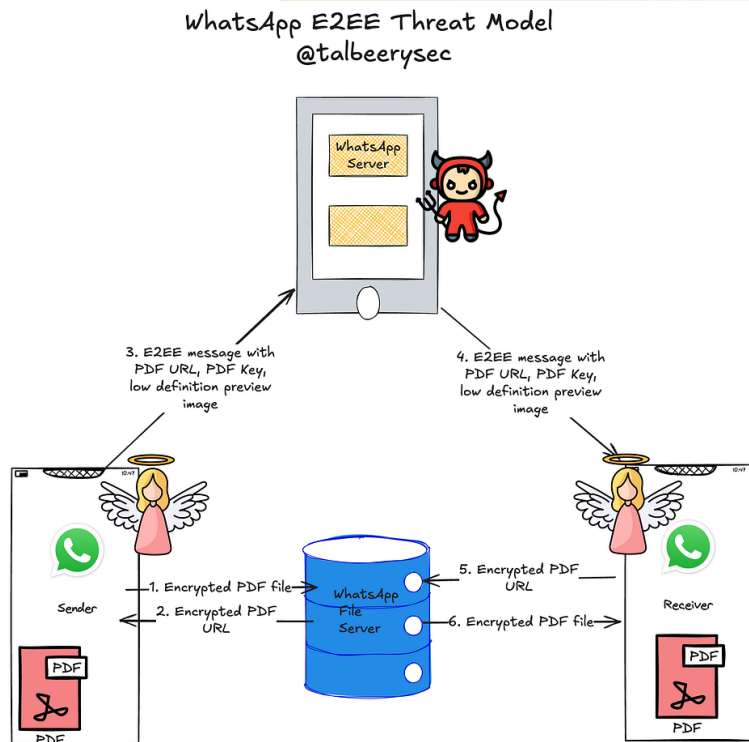
WhatsApp Signal E2EE

- Introduced in 2016
- Based on Signal Protocol
- <https://signal.org/blog/whatsapp-complete/>
- <https://www.whatsapp.com/security/WhatsApp-Security-Whitepaper.pdf>



WhatsApp Threat Model: Rogue Server

- No data on server
 - Metadata remains
- Prevents mass surveillance
- Server is a less attractive target now



CANNOT TAKE DATA FROM SERVER

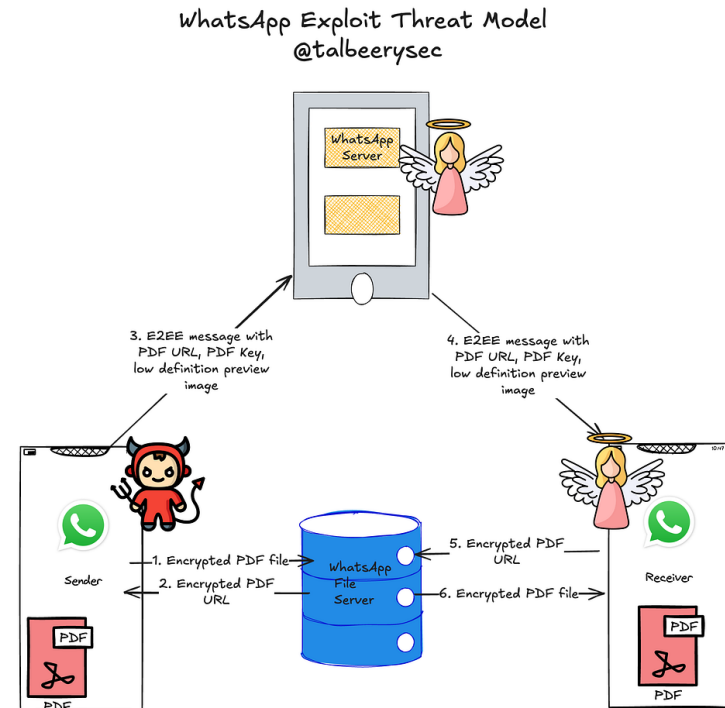
IF DATA NOT ON SERVER

WhatsApp Security

Rogue clients' attacks

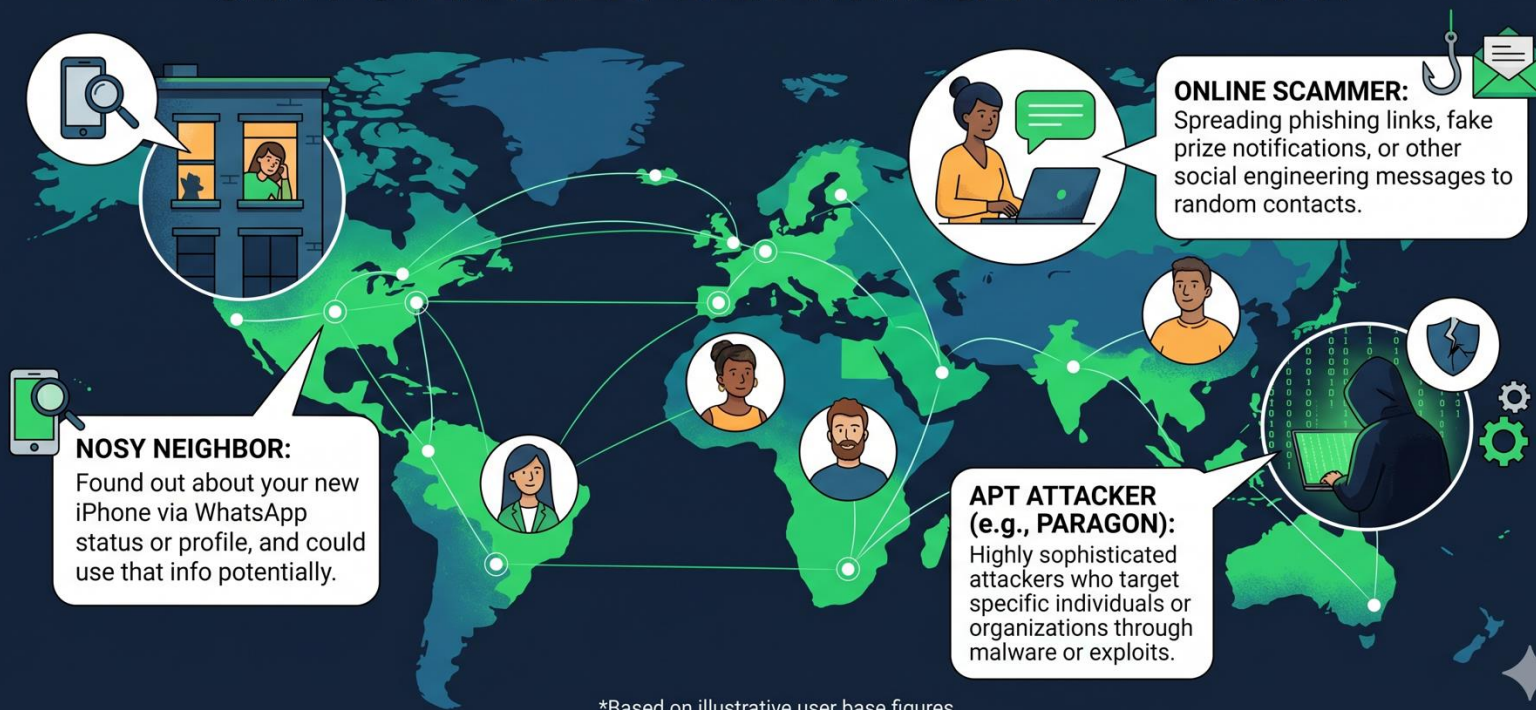
WhatsApp Threat Model: Rogue Client

- **Server is a less attractive target now**
 - **Client is a more attractive target**
- **Server is a “dumb pipe”**
 - **Cannot see malicious traffic**
 - **Cannot terminate sessions:****Client is directly “exposed”**



WhatsApp Threat Model: Rogue Client

ANYONE OF **WHATSAPP'S 3.5B*** USER BASE
CAN POTENTIALLY BE A WHATSAPP ATTACKER



*Based on illustrative user base figures.

SPAM and Scams

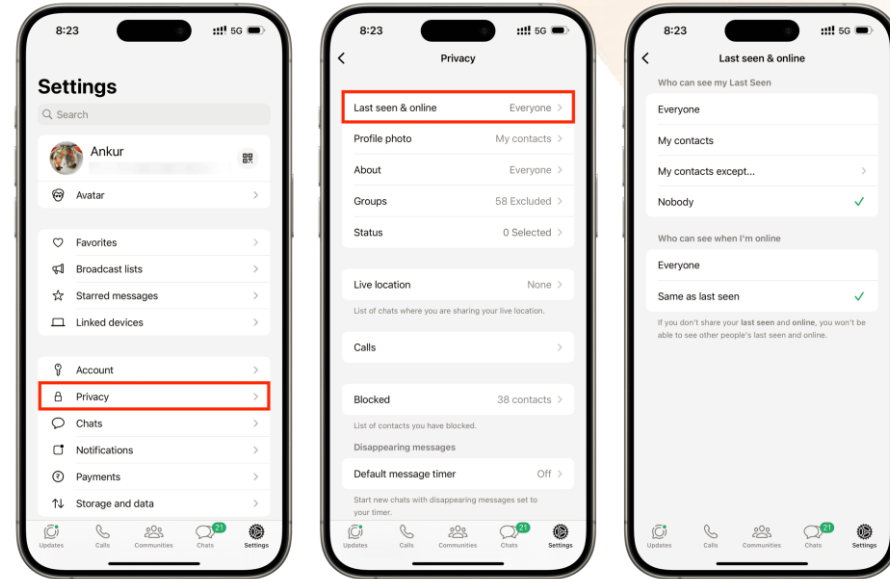
- **Rogue messages are not necessarily highly technical**
- **Same as email spam and scams**
- **However, unlike Gmail SPAM filters, WhatsApp server filters cannot see content, only metadata**



Source: norton.com

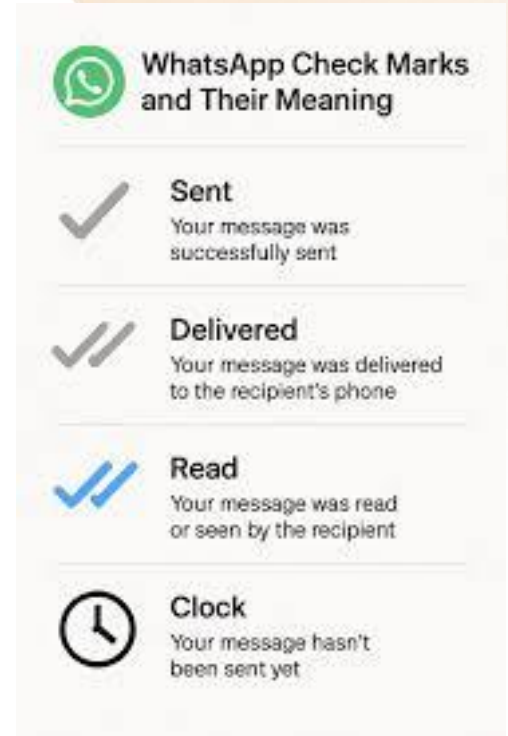
Stalking - 1

- Continuously monitoring “last seen” and “online status”
- Can be controlled by user



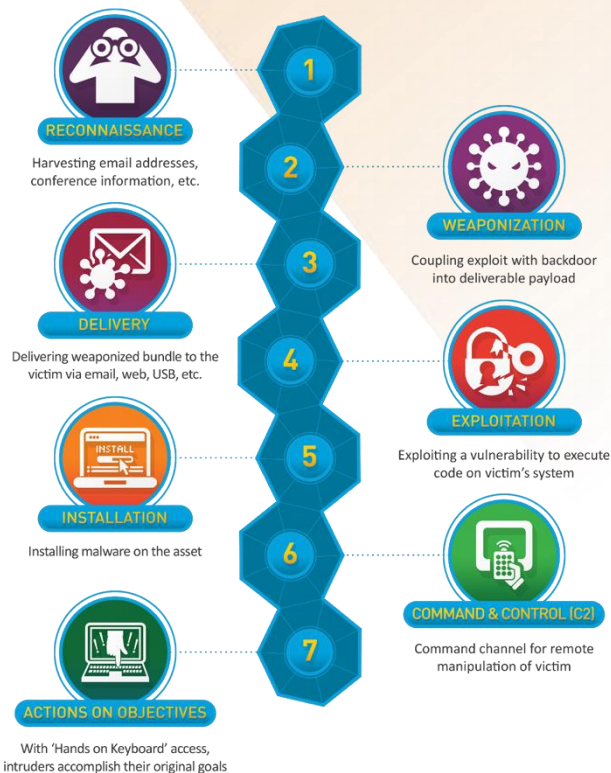
Stalking - 2

- **“Double check mark” = delivered**
- **Indicates the app is online**
- **Cannot be controlled by user**



APTs' WhatsApp kill-chain

- **Recon: Get the devices OS**
- **Weaponization : Coupling OS vuln with WhatsApp vuln to render file from a non-contact**
- **Delivery: sending pinpointed exploit only to the specific device**



Android WhatsApp 0-click

PARAGON'S ANDROID SPYWARE - WHATSAPP ZERO-CLICK FLOW

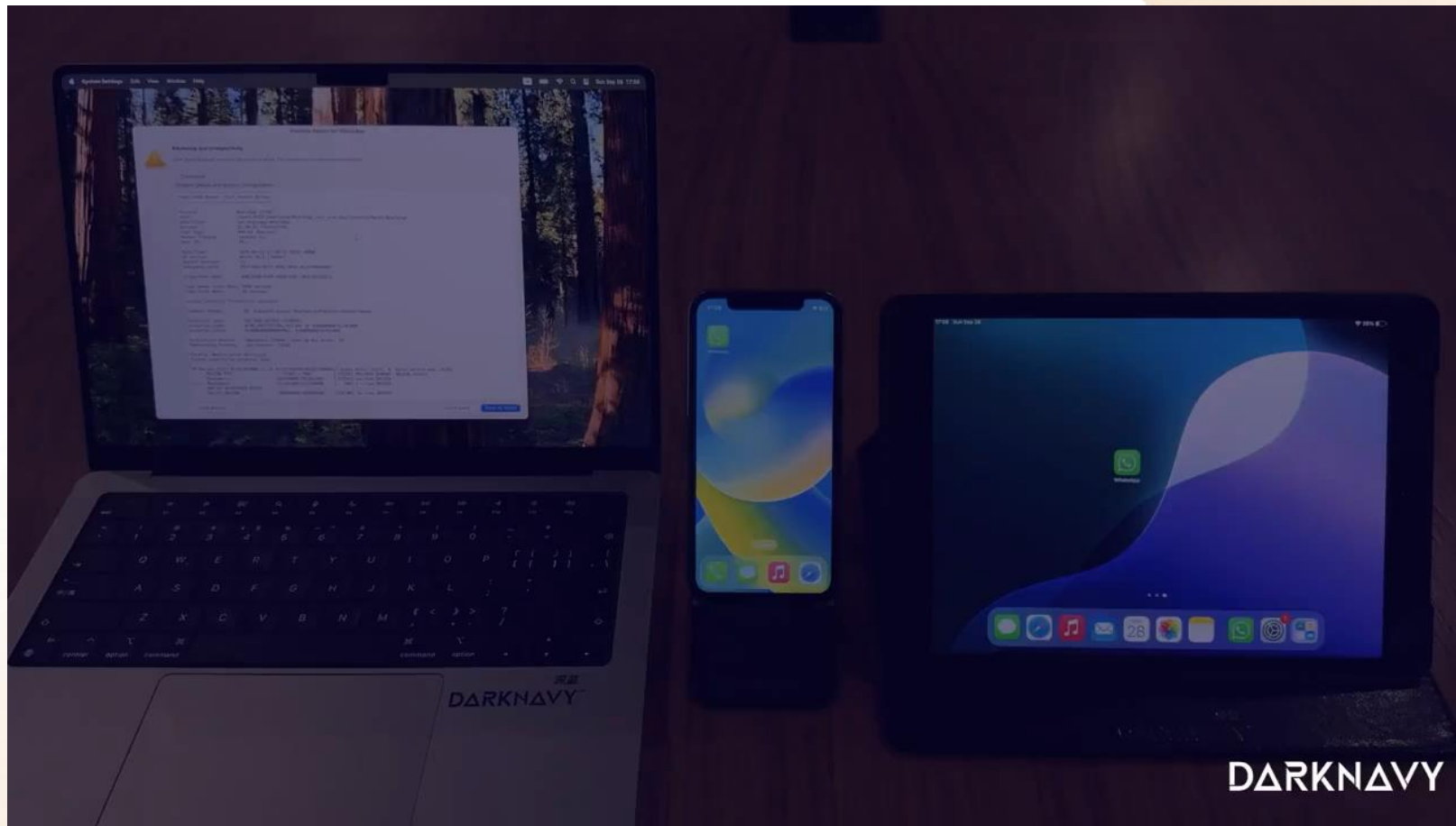


- 1 Attacker adds victim to WhatsApp group in a specific way
- 2 Attacker sends PDF to group
- 3 Victim's device automatically parses PDF, exploiting vulnerability
- 4 Graphite implant loaded into WhatsApp
- 5 Spyware escapes the Android sandbox and propagates to other apps on the device

iOS WhatsApp 0-click

- **OS specific 0-day RCE in some render-able format:**
 - **iOS DNG parsing (CVE-2025-43300)**
- **WhatsApp vuln to auto render it for preview, as usually files from unknown contacts are not auto-previewed**
 - **Whatsapp linked device sync messages (CVE-2025-55177)**
- **Compared to Gmail filters, no server-side scanning**

RCE Demo

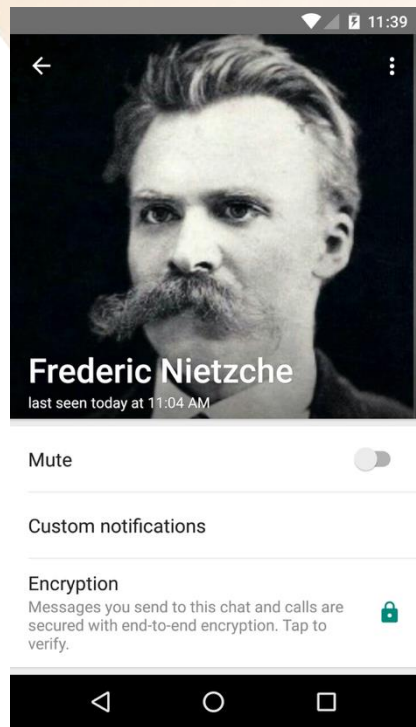
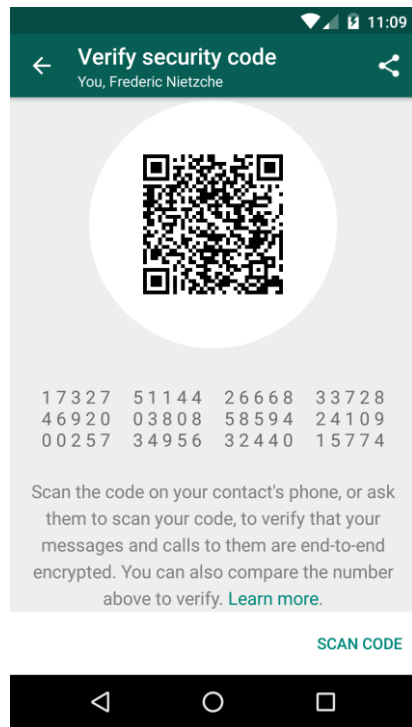


WhatsApp

E2EE deeper dive + privacy leaks

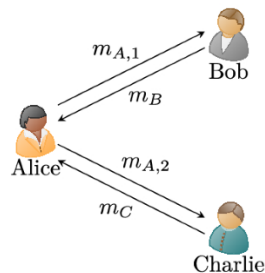
Signal E2EE in high level

- Each user generates a private key
- Reports public key to server
- Server shares public key with sender
- Sender & Receiver use X3DH to securely create a shared secret
- Messages can be E2EE!

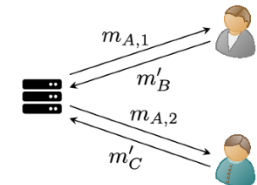
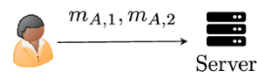


Extended Triple Diffie-Hellman (X3DH)

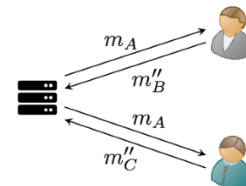
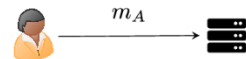
- Sending is immediate, no handshake overhead
- Receiver can be offline
- Additional “prekeys bundles” are uploaded to server in advance



(a) AKE protocol



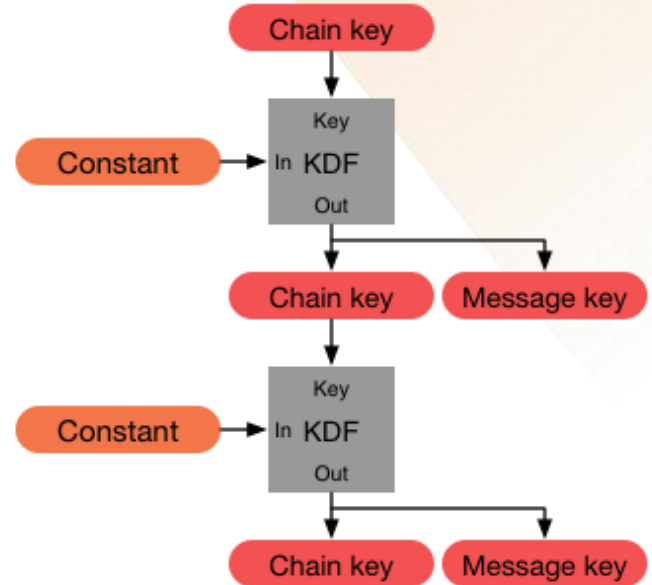
(b) Signal handshake as secure as AKE protocol but with large overhead (not used by the Signal app)



(c) Signal handshake slightly weaker compared to AKE protocol but with small overhead (used by the Signal app)

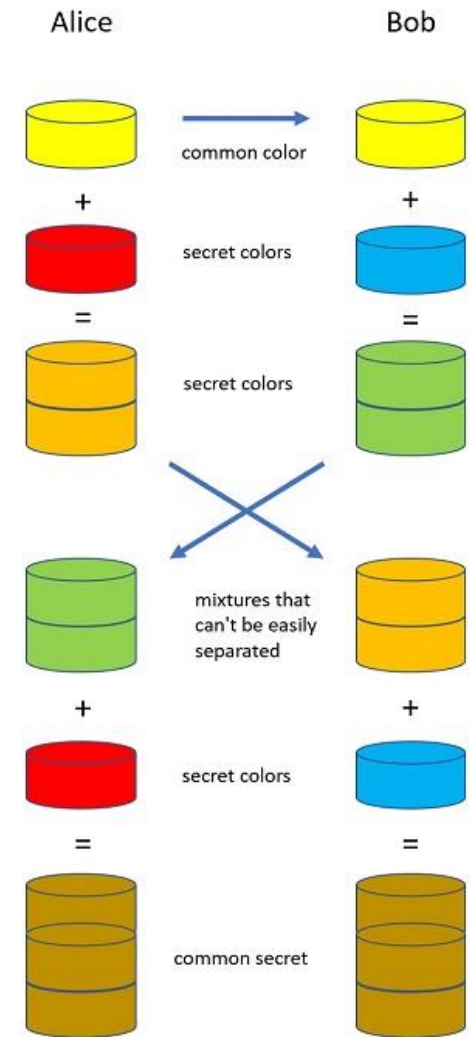
Signal E2EE: Symmetric Ratchet

- **Forward Secrecy (PFS)**
- **Attackers cannot go backwards**



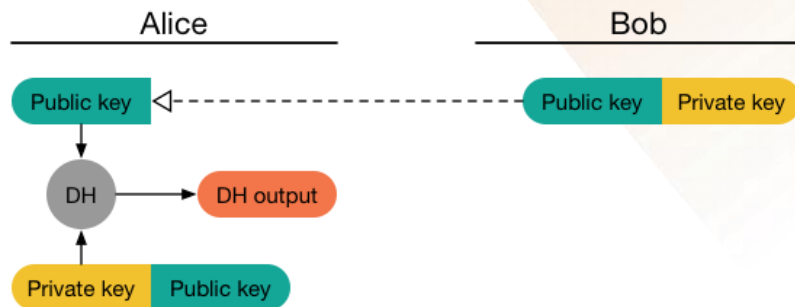
DH key exchange reminder

- Secure key exchange in the presence of an eavesdropper
- $A*(B*C) = B*(A*C)$



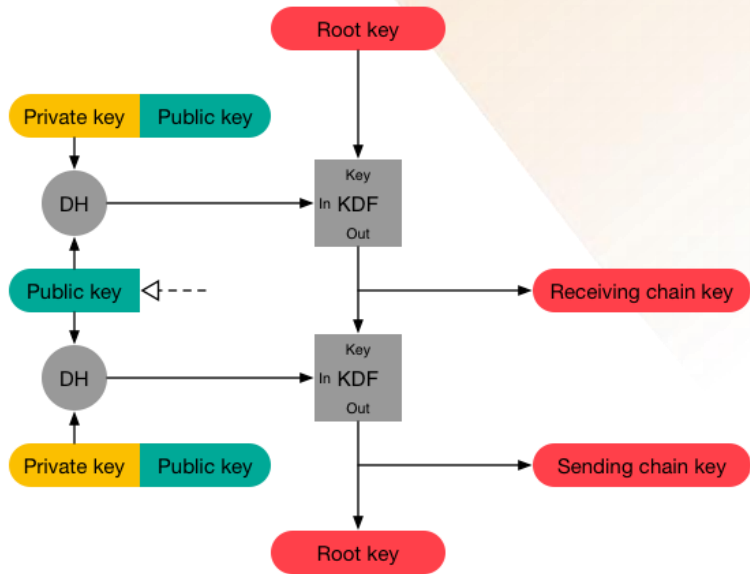
Signal E2EE: DH Ratchet

- **Post Compromise Secrecy (PCS)**
- **Attackers cannot keep track of messages once malware is removed**



Signal E2EE: Double Ratchet

- To provide **Forward Secrecy (PFS)** and **Post Compromise Secrecy (PCS)** the encryption key is updated
- Even if private key is compromised the system can heal itself



Levchin Prize 2017

Moxie Marlinspike and Trevor Perrin

For the development and wide deployment of the Signal protocol.

Moxie Marlinspike and Trevor Perrin are awarded the 2017 Levchin Prize for their development of the Signal protocol used to encrypt messages in communication systems. This protocol has been implemented into WhatsApp, Facebook Messenger, and Google Allo, encrypting the conversations of more than a billion people worldwide. The Signal protocol uses triple Diffie-Hellman, and a double ratchet mechanism to provide strong forward-secrecy. It is this massive deployment of an elegant encryption protocol for which Moxie and Trevor are awarded the Levchin prize.

CISA: Use E2EE!



Mobile Communications Best Practice Guidance

Introduction

The Cybersecurity and Infrastructure Security Agency (CISA) has identified cyber espionage activity by [People's Republic of China \(PRC\) government-affiliated threat actors targeting commercial telecommunications infrastructure](#). This activity enabled the theft of customer call records and the compromise of private communications for a limited number of highly targeted individuals. While applicable to all audiences, this guidance specifically addresses “highly targeted” individuals who are in senior government or senior political positions and likely to possess information of interest to these threat actors. CISA is releasing this best practice guidance to promote protections for mobile communications from exploitation by PRC-affiliated and other malicious cyber threat actors.

Best Practices

CISA strongly urges highly targeted individuals to **immediately review and apply** the best practices below to protect mobile communications. Highly targeted individuals should assume that all communications between mobile devices—including government and personal devices—and internet services are at risk of interception or manipulation. While no single solution eliminates all risks, implementing these best practices significantly enhances protection of sensitive communications against government-affiliated and other malicious cyber actors. Organizations may already have these best practices in place, such as secure communication platforms¹ and multifactor authentication (MFA) policies. In cases where organizations do not, apply the following best practices to your mobile devices.

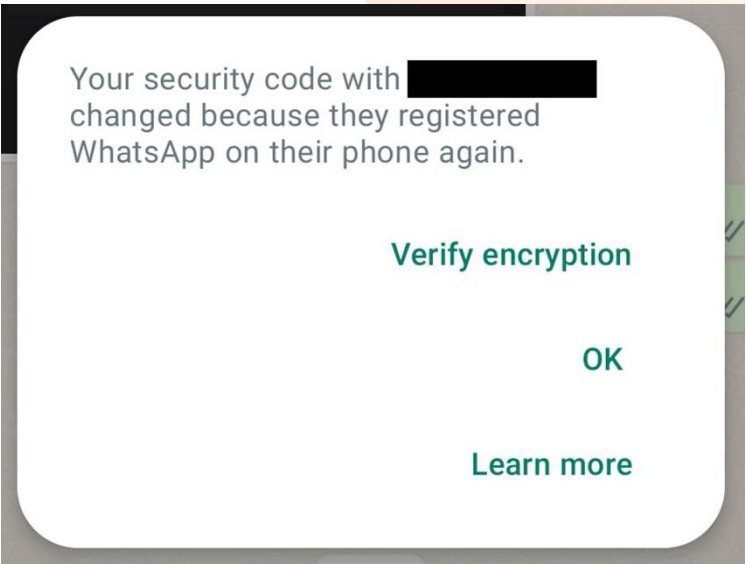
General Recommendations

Apply these best practices to your devices and online accounts.

1. Use only end-to-end encrypted communications.
 - ☐ Adopt a free messaging application for secure communications that guarantees end-to-end encryption, such as Signal or similar apps. CISA recommends an end-to-end encrypted messaging app that is compatible with both iPhone and Android operating systems, allowing for text message interoperability across platforms. Such apps may also offer clients for MacOS, Windows, and Linux,

A small privacy leak

- **E2EE created a new identifier to user device: The public key / security code**
- **When users change devices, they often change their public key / security code**
- **Other users learn about it**
- **Probably a small privacy price to pay to get E2EE privacy**

A screenshot of a WhatsApp notification. The notification is a white rounded rectangle with a thin grey border. It contains text about a security code change, three green buttons, and a redacted area. The background of the slide has a large, abstract orange shape in the top right corner.

Your security code with [REDACTED]
changed because they registered
WhatsApp on their phone again.

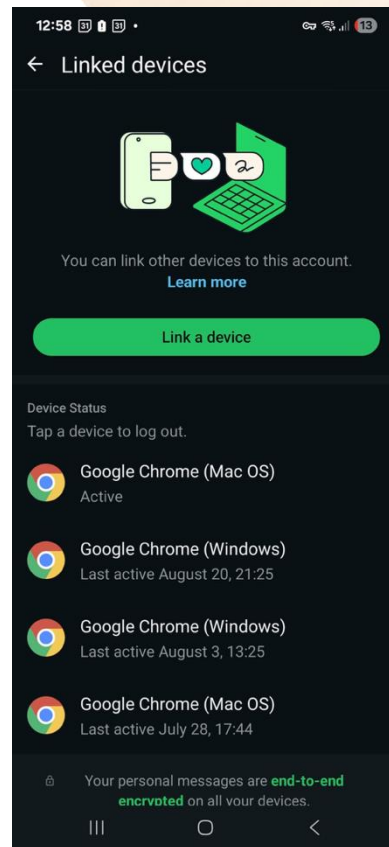
[Verify encryption](#)

[OK](#)

[Learn more](#)

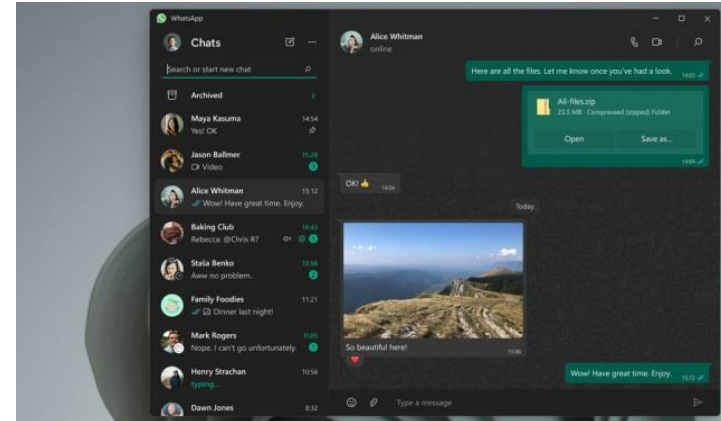
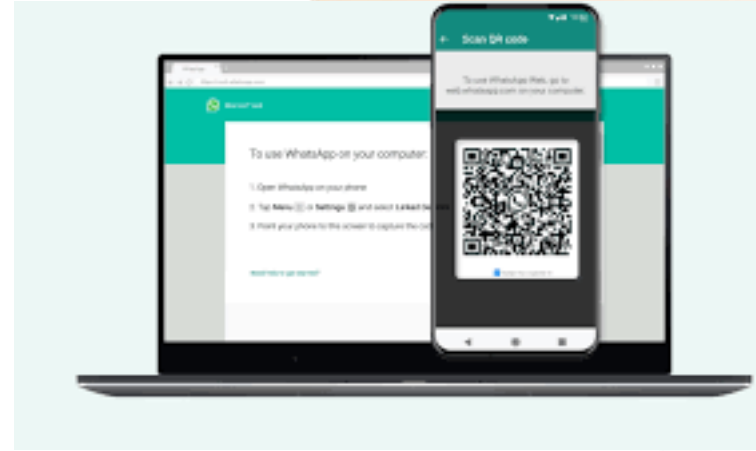
How to E2EE with Multi-devices?

- Signal Sesame : Multiple sessions
- Each device generates Key bundles
- Sender creates a session with:
 - Each receiver device
 - All other sender devices
- Sender sends message on all session



Mo' devices, Mo' problems

- Remember our small privacy problem?
- Users became aware of the device
- It got bigger!
- Now users are aware of other users' devices
 - Number of receiver device
 - Changes to them



WhatsApp with Privacy?

- <https://www.usenix.org/conference/woot24/presentation/beery>



WhatsApp with privacy? Privacy issues with IM E2EE in the Multi-device setting

Tal A. Be'ery, *Zengo*

<https://www.usenix.org/conference/woot24/presentation/beery>

**This paper is included in the Proceedings of the
18th USENIX WOOT Conference on Offensive Technologies.**

August 12-13, 2024 • Philadelphia, PA, USA

ISBN 978-1-939133-43-4

WhatsApp reaction: WONT FIX



Thursday, January 11, 2024 at 7:01 PM

Our reply

Hi Tal,

Thank you for reaching out. We have investigated your submission with the relevant product/engineering teams.

In WhatsApp's Security Whitepaper (<https://www.whatsapp.com/security/WhatsApp-Security-Whitepaper.pdf>) we state: "The client uses client-fanout for all the exchanged messages, which means each message is encrypted for each device with the corresponding pairwise session." Thus, the sender's device needs to be aware of the identities of all of the receiver's devices. This architecture is essential to support end-to-end encrypted delivery over multiple companion devices.

It's also important to keep in mind that key changes can happen for a variety of reasons on WhatsApp. It could happen if a user decides to re-install the WhatsApp application, for example. You can read more about this on our Help Center: <https://faq.whatsapp.com/1524220618005378>.

An attacker observing the fact that the victim's security code has changed won't necessarily be able to reliably deduce that the victim has, for example, changed devices or bought a new phone. Furthermore, key change notifications are a fundamental part of the privacy protections offered by WhatsApp: E2EE key change notifications let users know that they might not be talking with the same person as they were before. It's why we've made it easier for users to verify their secure connection with whom they are messaging with: <https://blog.whatsapp.com/new-security-features-account-protect-device-verification-automatic-security-codes>

Although this issue does not qualify as a part of our bounty program, we appreciate your report. We will follow up with you on any security bugs or with any further questions we may have.

Thanks,

Philip

Meta Security

WhatsApp

A new research tool

Looking under the hood

- The client does not let us see all that E2EE action
- We need a client that does!
- Based on Baileys open-source project
- <https://github.com/WhiskeySockets/Baileys>
- New client is added as another companion device





SAY HELLO TO MY LITTLE FRIEND

A CENTURY OF
SCARFACE

Author of The Exorcist Legacy: 50 Years of Fear

NAT SEGALOFF

FOREWORD BY STEVEN BAUER,
ON HIS STAR-MAKING ROLE AS MANNY RIBERA



WhatsApp Bailey's Device Toolkit

Baileys Web UI

Dashboard
Overview and status

Send Messages
Send messages and reactions

Chats
View and manage chats

Contacts
Contact management

Devices
Device management

Prekey Bundles
View device prekey bundles

Presence
Presence control

Developer
Advanced tools

Settings
Client configuration

Logout

WhatsApp Web Interface

OPEN REFRESH

Dashboard

Welcome to the Baileys WhatsApp Web Interface. Monitor your connection status and manage your WhatsApp operations.

WhatsApp Connection

Connected

Unknown@s.whatsapp.net

Baileys v6.7.21

Last updated: 4/17/2026, 12:09:42 PM

Quick Stats

50

Total Chats

1934

Total Contacts

3

Unread Chats

Recent Chats

WhatsApp State	1
User Account	0
Baileys Version	1
Last Update	1
WhatsApp	0

System Status

WhatsApp State
open

User Account
Unknown

Baileys Version
6.7.21

Last Update
4/17/2026, 12:09:42 PM

Authenticated

github.com/talbeerysec/WhatsApp-baileys-device-toolkit

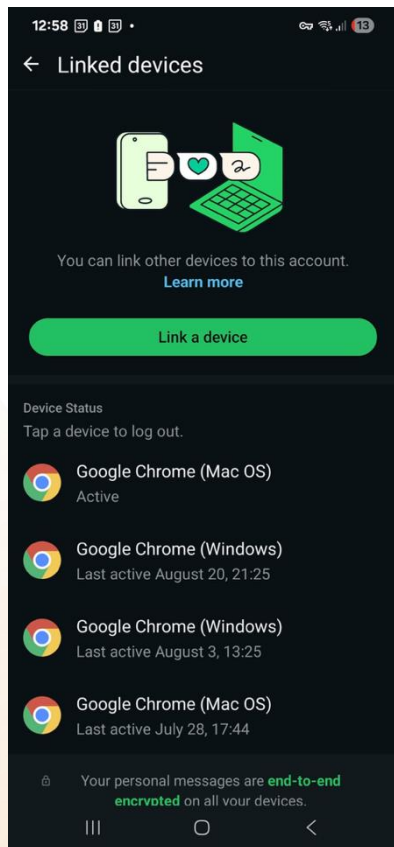
Say hello to my eldest son



WhatsApp Client Attacks

Device enumeration + pinpointed sending

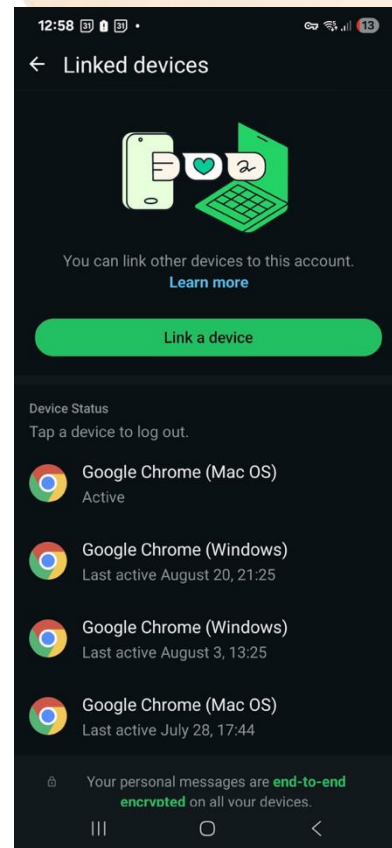
Device Enumeration



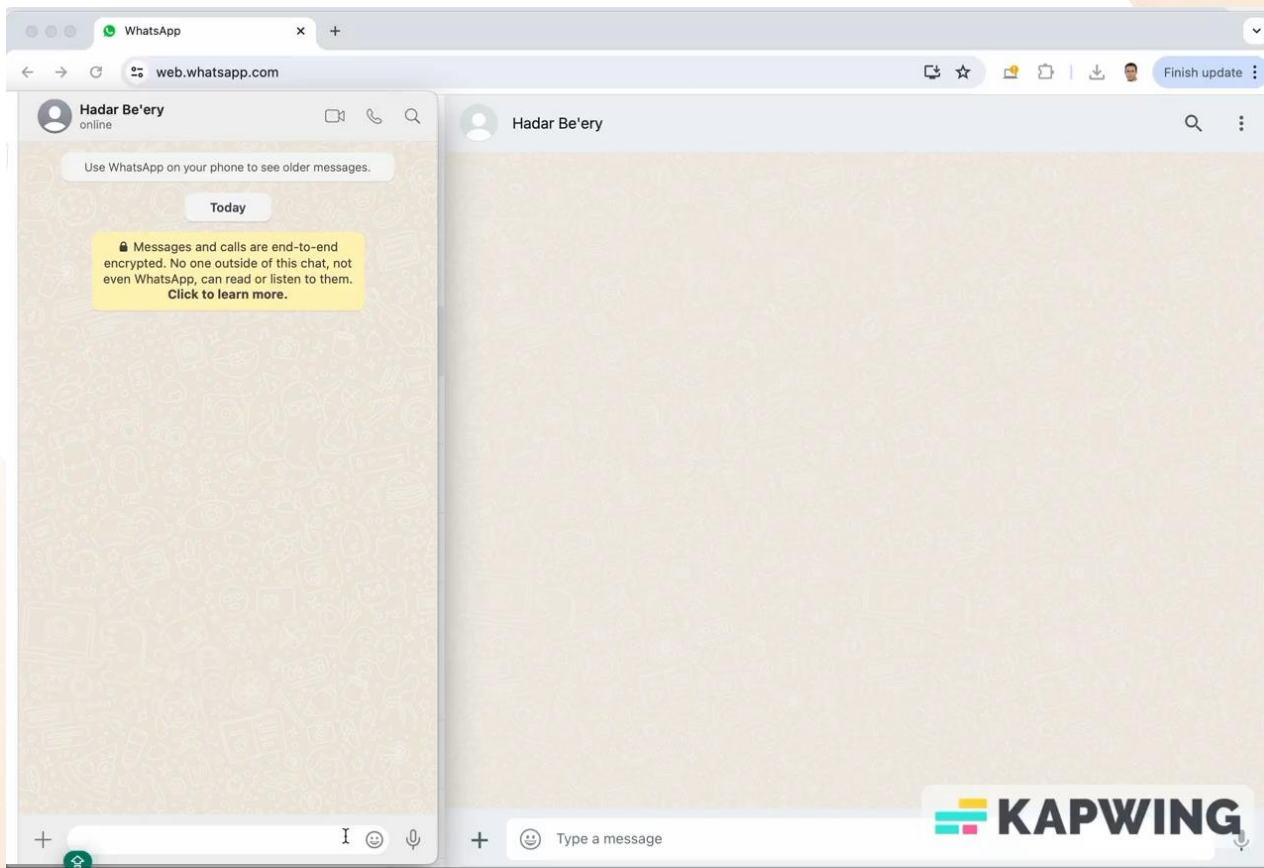
WhatsApp Web Interface	
Device	Type
Device 0	Primary
Device 79	Secondary
Device 80	Secondary
Device 81	Secondary
Device 82	Secondary

How to E2EE with Multi-devices?

- Signal Sesame : Multiple sessions
- Each device generates a Key
- Sender creates a session with:
 - Each receiver device
 - All other sender devices
- Sender sends message on all sessions
 - **what if it would send a different message?**

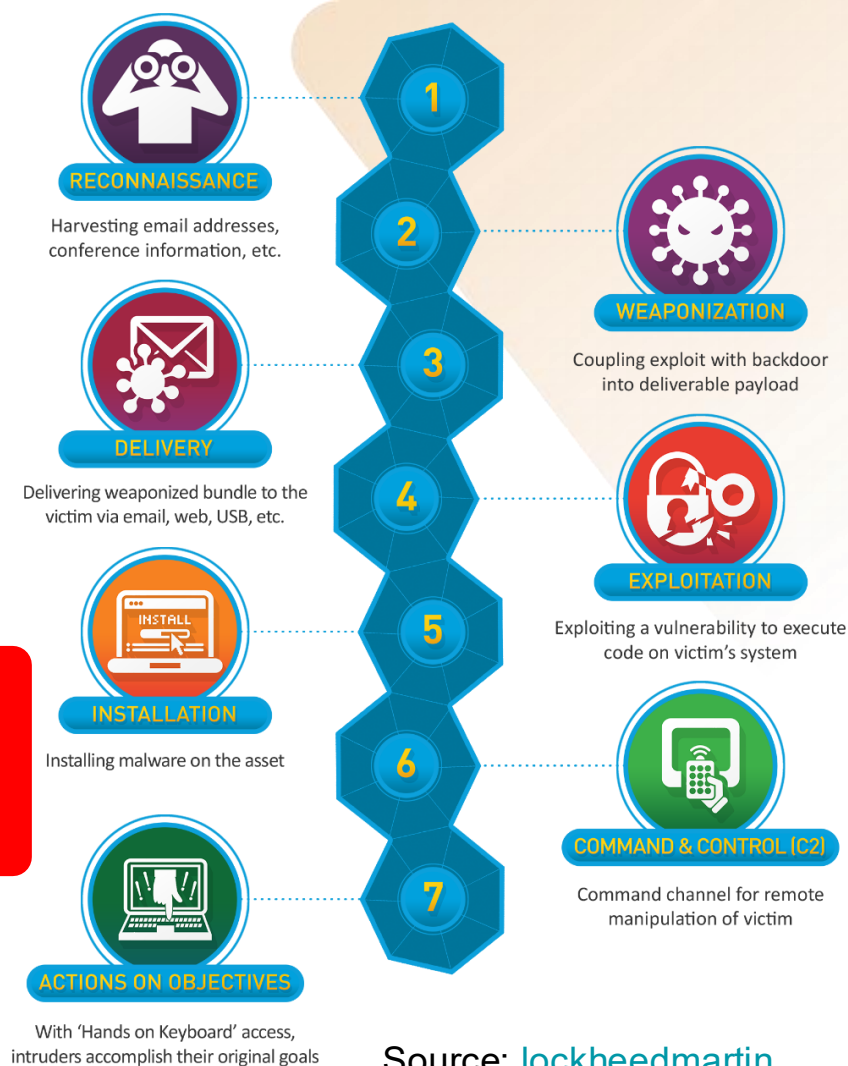


Pinpointed sending exploit Demo



WhatsApp kill-chain

- **Recon: Get the devices OS**
- **Weaponization : Coupling OS vuln with WhatsApp vuln to render file from a non-contact**
- **Delivery: sending pinpointed exploit only to the specific device**



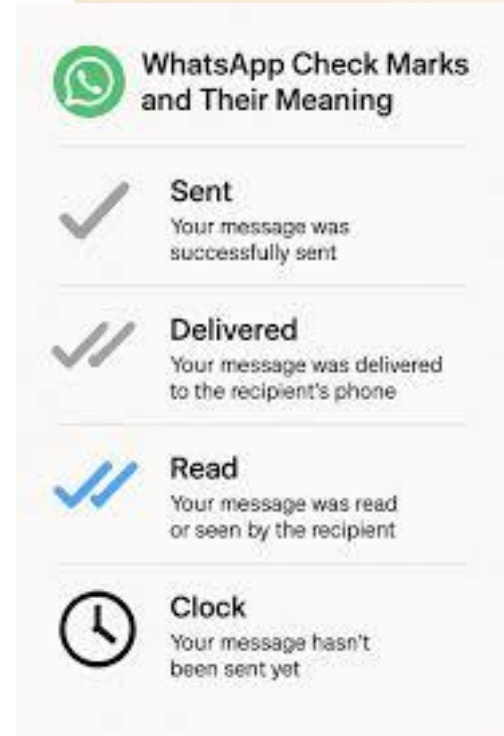
Source: [lockheedmartin](https://www.lockheedmartin.com)

WhatsApp Client Attacks

Online Status Reconnaissance

Stalking

- **“Double check” mark delivered**
- **Indicates the app is online**
- **Cannot be controlled by user**
- **What if we could send “transparent” messages?**



“Careless Whisper”

- Gegenhuber et al 2025 <https://arxiv.org/pdf/2411.11194v4>

Careless Whisper: Exploiting Silent Delivery Receipts to Monitor Users on Mobile Instant Messengers

Gabriel K. Gegenhuber^{1,2}, Maximilian Günther¹, Markus Maier¹,
Aljosha Judmayer¹, Florian Holzbauer¹, Philipp É. Frenzel³, and Johanna Ullrich¹

¹University of Vienna, Faculty of Computer Science, ²UniVie Doctoral School Computer Science, ³SBA Research

“Careless Whisper”: Online status

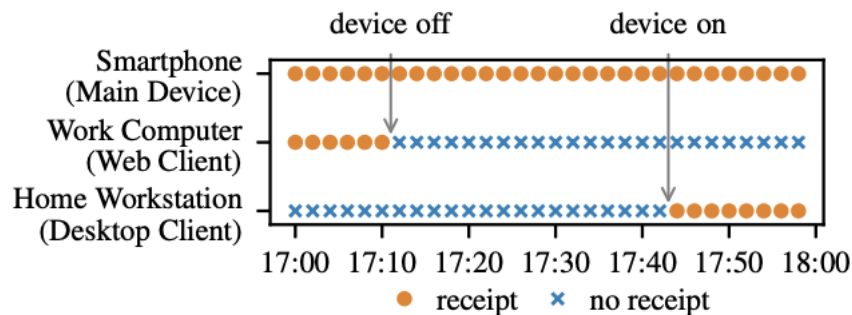


Fig. 3: A device's online status can be consistently and stealthily monitored with second-based granularity, possibly leaking the user's location and daily routines.

“Careless Whisper”: Screen on status

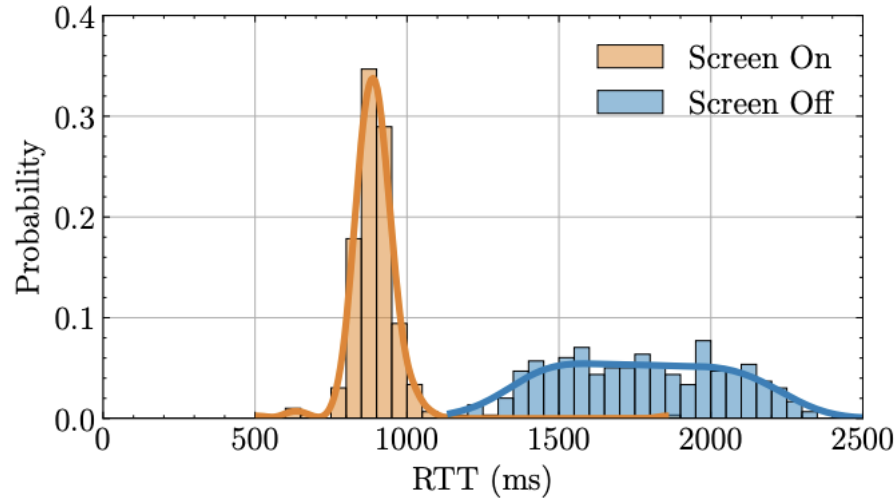
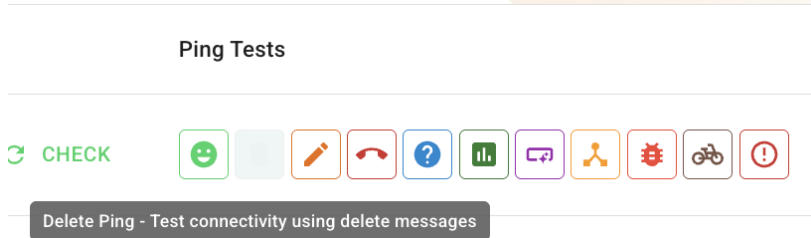


Fig. 1: Round-trip times (RTT) of delivery receipts, which are ≤ 1 second for *Screen On* states and > 1 second and above for *Screen Off* states measured on an iPhone using WhatsApp with a sampling rate of 1 Hz.

“Careless Whisper”: In our tool

#	Type	Mechanism
1	Reaction	Reaction to a non-existent message
2	Delete	Delete/revoke message (REVOKE protocol, type=0)
3	Edit	Edit message (MESSAGE_EDIT protocol, type=14)
4	Call-Reject	Call rejection (CALL_LOG_MESSAGE, type=22)
5	Unknown	Non-existent protocol type (type=101)
6	Poll Response	Vote on a non-existent poll
7	Button Response	Button response with a fake ID
8	Device Coordination	Multi-device sync message
9	App State	App state fatal exception
10	Peer Data	P2P data operation request
11	Malformed	Invalid proto field (<code>conversation1</code>)



WhatsApp

Malicious user attacks: Fingerprinting

Exploiting Sesame for APTs exploits

- **E2EE Sesame allows attackers can pinpoint the exploit to a specific device and not worry about other OS implications**
- **But how will attackers know which is which?**
- **Where to pin the point?**



Fingerprinting

- **WhatsApp clients are (re-)implemented in 3 major programming languages**
 - **Java (Kotlin) for Android**
 - **Swift for iOS: iPhone, Mac, iPad**
 - **Javascript: Web, Windows**
- **Due to E2EE the message is terminated by the client, not the server, thus allowing fingerprinting**

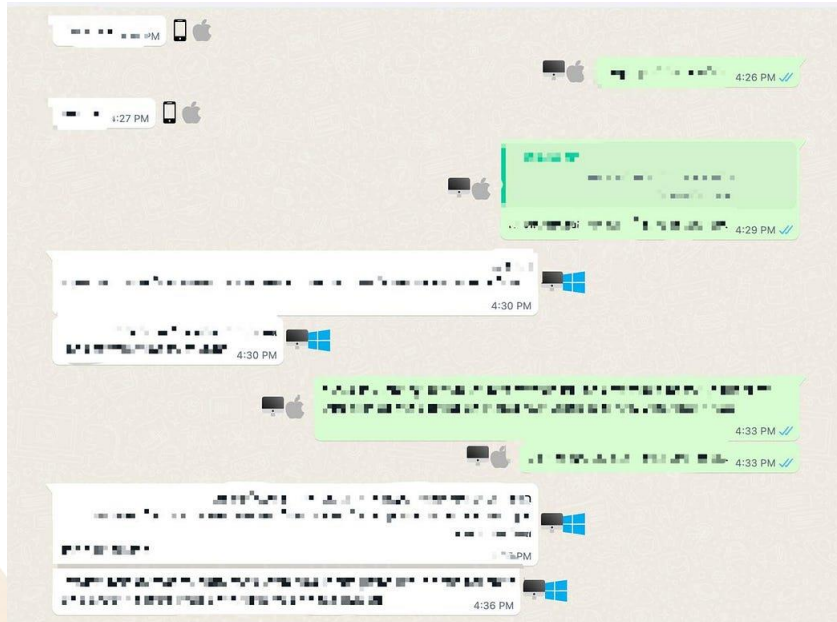


Fingerprinting: user sending

- When user sends a Message ID is assigned
- Message ID format is device specific
- medium.com/@TalBeerySec/

platform	length	prefix
Web	22	3EB0
Android	32	no
iPhone	20	3A, 5E

Fingerprinting: user sending demo



Fingerprinting: With silent pings

- **Different devices have different responses to silent pings**
- **Specifically, primary devices**
- **iOS acks a delete inexistant message, Android does not**
- **Web acks a reject call message with no call, Desktop does not**

Fingerprinting: With silent pings demo

WhatsApp Web Interface

OPEN

REFRESH

Device Management

Discover devices for users, check their online status, and identify device types through [REDACTED]

Profile All: Automatically tests all devices with [REDACTED]

@talbeerysec

Device Discovery

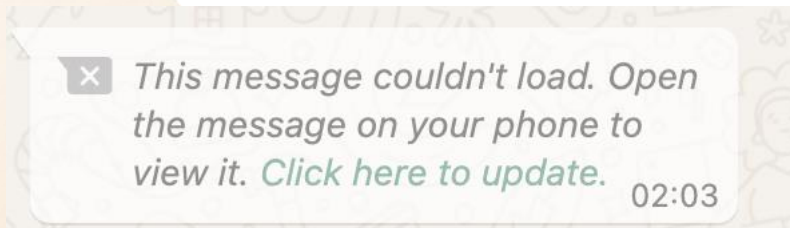
User ([REDACTED])

GET DEVICES

PROFILE ALL

Meta's reaction

- **A small bounty, some fixes promised**
- **What we saw**
 - **Some pings does not work on some platforms**
 - **Web now shows an indication when receiving unexpected messages**



Friday, January 2, 2026 at 7:36 PM

Our reply

Hi Tal,

Thanks for your report and apologies for the delayed response. We really appreciate your ongoing research and your patience.

We wanted to share our assessment on two issues included in your submission:

First, online status inference: this issue was already reported to us by another research team (the University of Vienna), who collaborated with us on testing our mitigations as part of their study, and we'll continue to harden this product surface.

Second, OS inference: We remain focused on protecting users against many different attack vectors while still ensuring we can smoothly run a service that over 3 billion people use around the world. While our goal is to continue to harden our app against emerging risks, we also know that device fingerprinting can be done across many technologies and platforms (including messaging apps), and through a variety of ways. In fact, operating systems make it trivial to infer – you can do so by typing in a number on iMessage to see if someone is on iOS or Android, for example, before any message is sent. With that, while OS inference is typically out of scope of our program because the practical impact is limited given that attackers already have numerous ways to infer device types, our engineering work to standardize how our app works across platforms is always ongoing. Your report helped us address a case of disparity between how the iOS and Android client handled invalid messages, as part of our continuous effort to harden the surface.

We've decided to award a bounty of \$500 for your finding that led us to both fix the issue related to OS disparity in handling invalid messages and to improve our bug bounty triage process in this area.

As for CVE, we generally do not assign CVEs for server-side issues (online status inference) and the client-side bug (OS disparity in handling of invalid messages) does not meet our severity threshold.

Thank you again for your submission and we look forward to seeing your future reports.

Thanks,

Meta Bug Bounty

OpenClaw detection

- A popular AI agent
- WhatsApp interface (using Baileys)
 - Identifies itself as Web
 - But now “real” web implementation does not answer bad reaction pings!



OpenClaw detection demo

Baileys Web UI

Dashboard

Overview and status

Send Messages

Send messages and reactions

Chats

View and manage chats

Contacts

Contact management

Devices

Device management

Prekey Bundles

View device prekey bundles

Presence

Presence control

Developer

Advanced tools

Settings

Client configuration

Logout

WhatsApp Web Interface

OPEN REFRESH

Device Management

Discover devices for users, check their online status, and identify device types through ping fingerprinting.

Profile All: Automatically tests all devices with edit pings, then fingerprints online devices (delete ping for primary device OS detection, call-reject ping for secondary device type detection).

Device Discovery

GET DEVICES

PROFILE ALL

“Prekey pogo”

- Gegenhuber et al 2025 <https://arxiv.org/pdf/2504.07323>

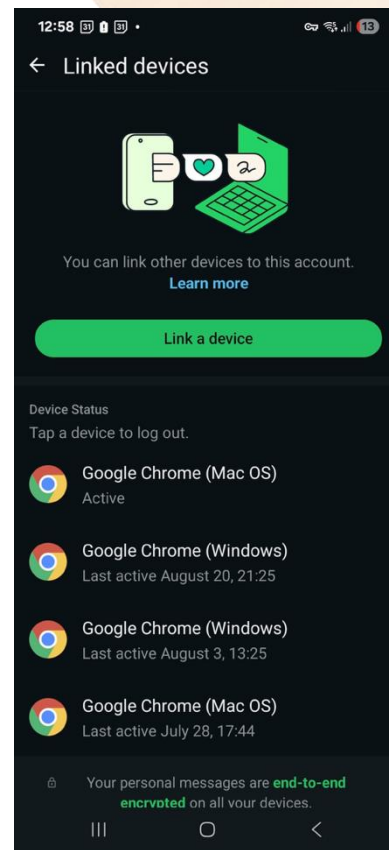


Prekey Pogo: Investigating Security and Privacy Issues in WhatsApp’s Handshake Mechanism

Gabriel K. Gegenhuber^{1,2}, Philipp É. Frenzel³, Maximilian Günther¹, and Aljosha Judmayer¹

How to E2EE with Multi-devices?

- Signal Sesame : Multiple sessions
- Each device generates Key bundles
- Each bundles include 3 Keys + IDs



Fingerprinting With prekeys ID

Client Implementation	Initialization Values for <i>key IDs</i>			Prekey Batch Size		Refill Trigger
	Registration	Signed PK	One-Time PK	Initial	Refill	
Android	<i>R</i>	0	<i>R</i>	812	812	10
iPhone	<i>R</i>	<i>R</i>	1	812	812	10
WhatsApp Web ^a	<i>R</i> & 0x3FFF	1	1	200	812	10
Desktop App macOS	<i>R</i>	<i>R</i>	1	200	812	10
Desktop App Windows	<i>R</i>	1	1	50	812	10

R Random number. ^a Verified on Firefox, Chrome, Safari.

Table 4: Different initialization values, prekey batch sizes, and incrementing ID patterns used across various implementations enable device fingerprinting (e.g., OS, device age). Beyond being a privacy risk, this could be exploited by an attacker during the reconnaissance phase to tailor further attacks.

Fingerprinting with prekey IDs: updated

Client Implementation	Initialization Values for <i>key IDs</i>			Prekey Batch Size		Refill Trigger
	Registration	Signed PK	One-Time PK	Initial	Refill	
Android	R	~~R~~	R	812	812	10
iPhone	R	R	1	812	812	10
WhatsApp Web ^a	R & 0x3FFF	1	1	200	812	10

Fingerprinting with prekey IDs: Demo

Devices for 972[REDACTED]

Device	Type	Device Type (Passive)
Device 0 972 [REDACTED]	Primary	 Android high confidence
Device 68 972 [REDACTED]	Secondary	 Windows Desktop high confidence
Device 70 972 [REDACTED]	Secondary	 Web high confidence
Device 73 972 [REDACTED]	Secondary	 Web high confidence

Attacks Summary

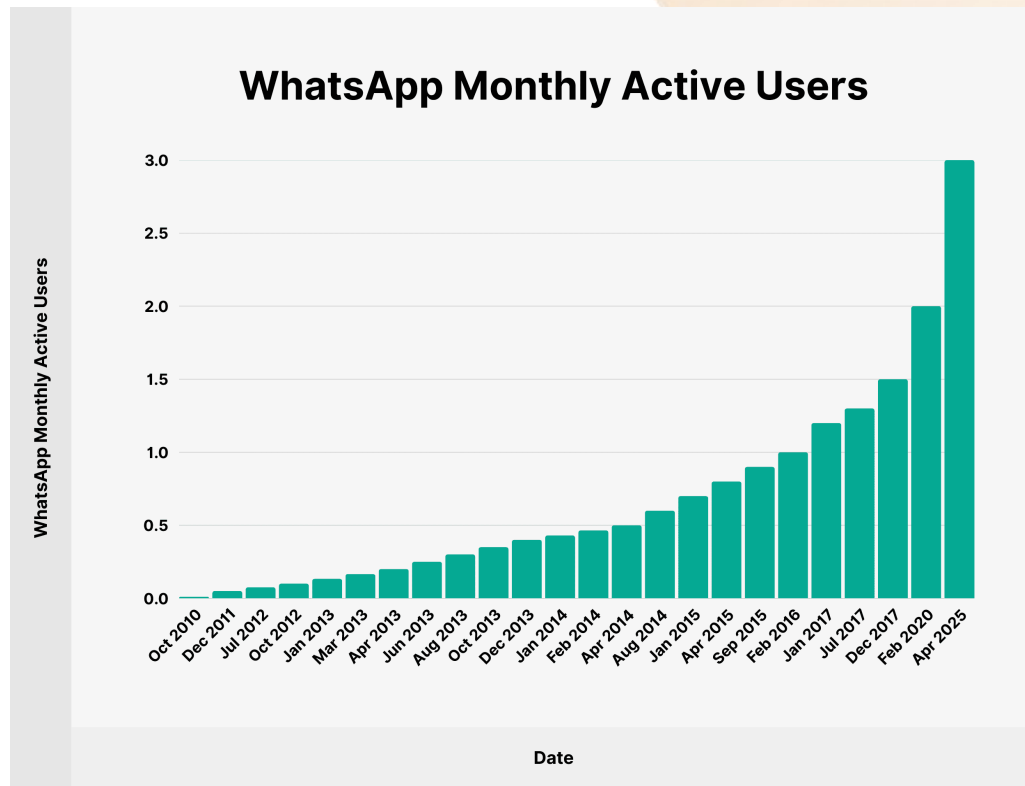
- **E2EE reduces the server's attack surface**
- **E2EE increases the client's attack surface and reduces server's capabilities to detect and mitigate**
- **Per device reconnaissance: online status, OS fingerprinting**
- **Per device sending: e.g. for pinpointing malware**
- **Enables APTs kill chain: OS Recon, pinpointed malware delivery**

WhatsApp

Solutions

WhatsApp: Then and Now

- 2010: 10M MAU
- 2025: 3B MAU
- US > 100M in 2025

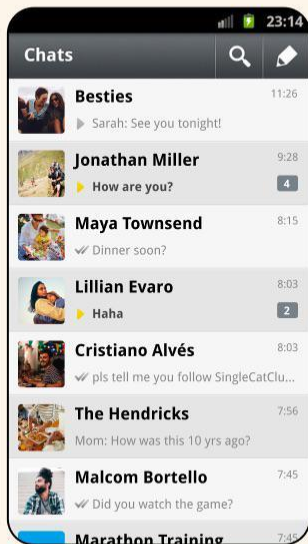


WhatsApp: Then and Now

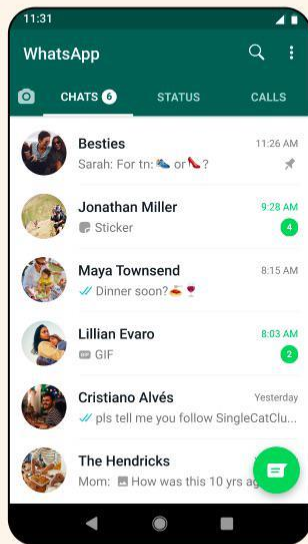
 Design over the years



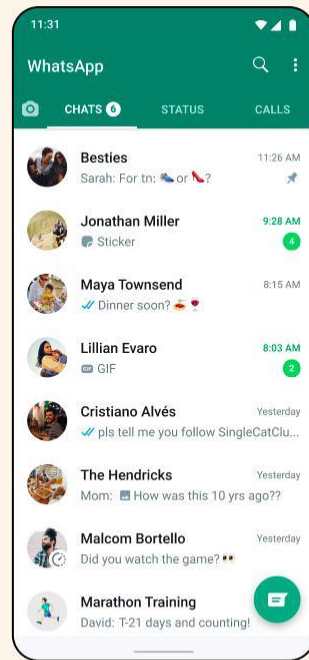
2011



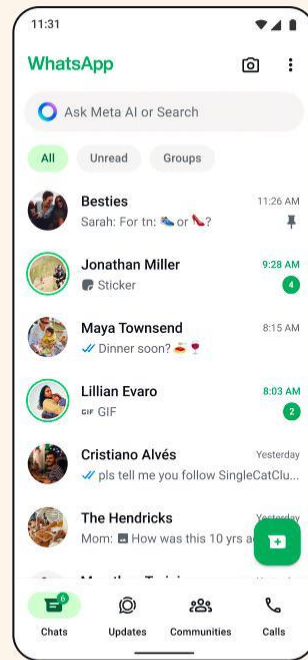
2012



2020



2021

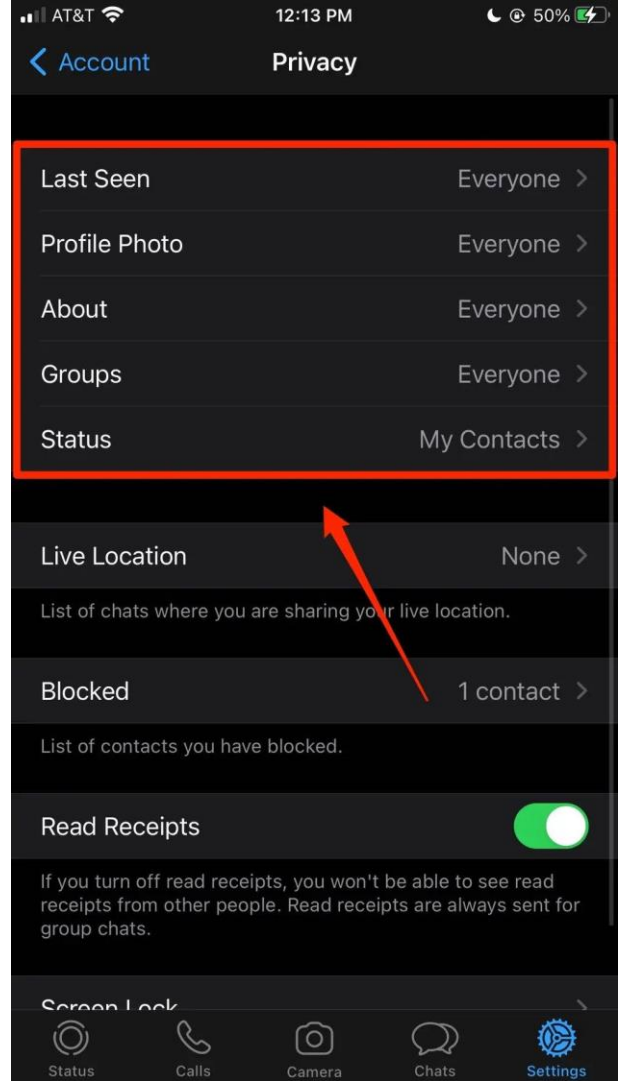


2024

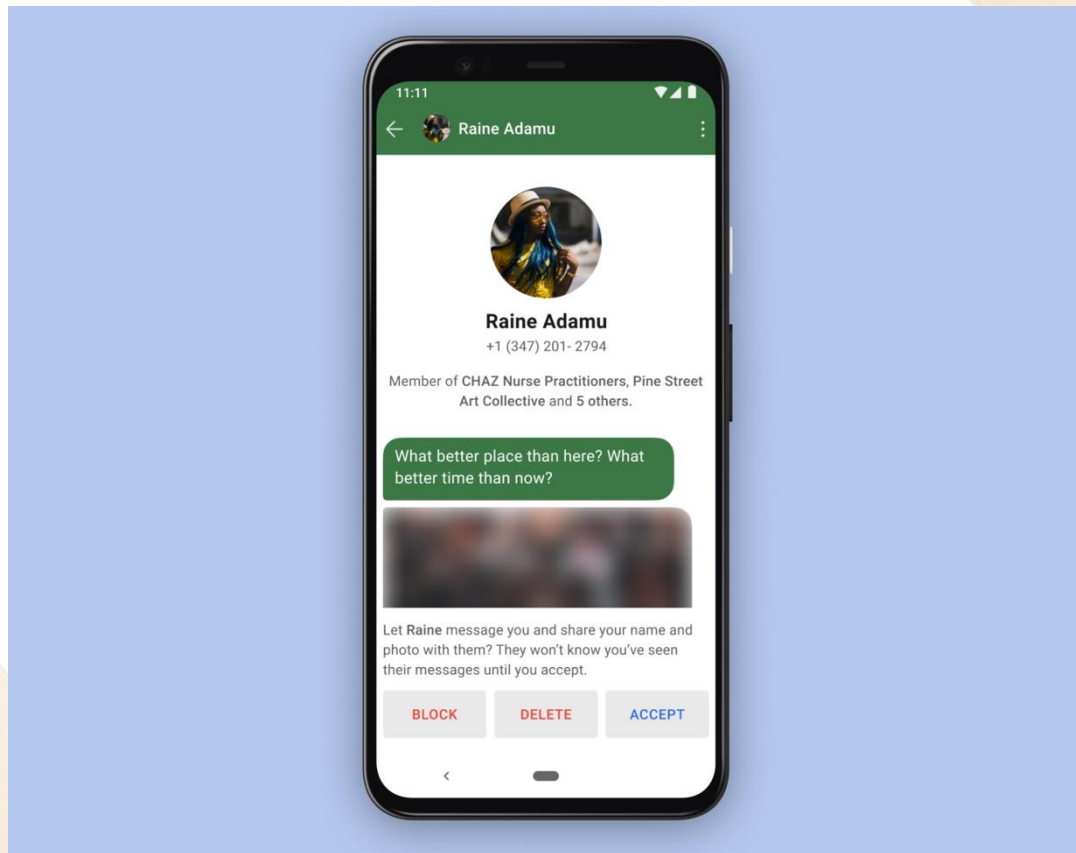
Source: backlinko.com

Introduce a Lockdown mode

- **If introduced: Limits users' exposure to contacts only**
- **Dramatically reducing the attack surface**
 - **3.5B >> 1K**
 - **Your contacts are usually less likely to attack you**



Introduce Message Requests (a la Signal)



Source: signal.org/blog/message-requests/

Replace Sesame: The ambassador protocol

- **Multiple devices, yet a single sender – receiver session**
 - **Privacy:** counterparty only aware of a single device
 - **Integrity:** single session
- **Adding an additional “ambassador” key pair to represent user**
- **Using the established E2EE network to send to other devices**
 - **Ambassador key distribution, Messages with opaque server fanout**
- See [TalBeerySec/the-ambassador-protocol](https://talbeerysec.com/the-ambassador-protocol)

WhatsApp

Summing up

Summing up

- **Trade-off: WhatsApp E2EE protects the server but exposes the client**
- **Multiple client based attacks + research tool: fingerprinting, online status reconnaissance and device pinpointing**
- **No way to currently protect as user**
- **Suggested solution WhatsApp can implement solutions to mitigate: Lockdown mode. Message Requests, Sesame improvements**



Questions?



Scan QR code
to Download the App



www.zengo.com



twitter.com/zengo



github.com/Zengo-X



zengo.com/contact