Vietnam's contact tracing app broadcasting a fixed ID

April 26, 2020

Prologue

Bluezone (Android, iOS) is a Bluetooth-based contact tracing app sponsored by the Government of Vietnam and developed by a coalition of local tech companies and the Ministry of Information and Communications.

The developers claim that the app is designed to alert people who may have come in contact with the virus while keeping their identity anonymous. The Android and iOS apps were released a few days ago, and according to the official tally they have attracted more than 150,000 users (last update April 30th 2020). The media quoted a top official touting that Bluezone is a breakthrough because it allows the government not to collect people's information and is able to solve the basic errors in similar apps from other countries.

Needless to say, Bluezone piqued my curiosity. I want to know it works. The developers pledge to open source the app, but they actually haven't released neither code nor documentation. I break software professionally, and decided to send them a note offering my service pro bono, if they could give me early access to the design or something. I also suggested that they should consider open solutions such as DP3T. They said thanks, but didn't send me nothing.

When the app was released Thursday, I downloaded and reverse engineered it. I found terrible vulnerabilities. I wrote a report and shared with the developers, urging them to use DP3T. I also published a summary without too many details on my blog to alert my community. The developers published a rebuttal basically calling me mistaken. They didn't send me a copy.

Below is my report which includes a rebuttal to their rebuttal.

--

The report

1/ First of all, Bluezone assigns each installation a fixed 6-character ID, and broadcasts it over and over again. I've spent a majority of my waking moments in the past two weeks studying secure and private contact tracing technologies and to the best of my knowledge, Bluezone is the only app doing this. Needless to say, this puts users at grave risks, as it leaks their location, movement, or social graph, etc. to whoever's watching.

In their rebuttal, the developers said they don't think changing the ID can improve user privacy because the Bluetooth MAC address is constant anyway. This is not true.

Bluetooth has two sub standards: Bluetooth Classic and Bluetooth Low Energy. A Bluetooth Classic device can be either discoverable or non-discoverable. When it is discoverable, it broadcasts its MAC address, device name and other information. But when it is non-discoverable, it doesn't broadcast anything. For privacy, neither Android nor iOS is discoverable by default. Phones are discoverable if and only if the Bluetooth System Setting app is running in the foreground, i.e., only when users want to pair with other devices.

Both Android and iOS also randomize the MAC address when broadcasting BLE advertisements. On Android, experiments by various groups show that restarting BLE advertising causes the OS to choose a new random MAC address. On iOS, experiments by a friend of mine and in this paper show that the OS automatically chooses a new random MAC address every hour or so.

I don't blame them for this confusion. I was confused myself when I thought Android and iOS expose APIs allowing apps to change the MAC address however they see fit.

2/ To add insult to injury, the fixed 6-character IDs are predictable.

This is how they are generated:

```
var generateUserId = function() {
    var t = '0123456789ABCDEFGHIJKLMNOPQRSTUVWXYZ',
    n = t.length,
    o = '';
    Math.seedrandom(new Date().getTime());
    for (var s = 0; s < 6; s++) o += t.charAt(Math.floor(Math.random() * n));
    return o;
 }
```

It uses this seedrandom library which always generates the same output when seeding with

the same value. In this case, the seed is the current timestamp. The app fell victim to one of the classic blunders in practical crypto software security. Pro tips: if you are to design a PRNG and let your users choose their own seed you may as well just return 4.

Anyways, this means all past and future IDs are predictable. The developers rebut that the timestamp is unpredictable because it has a millisecond precision. This is missing the forest for the trees. While it may be a bit difficult to guess the ID assigned to a particular person, there are many ways to exploit this weakness at the system level. A few examples from top of my head:

- An attacker can generate all possible IDs and broadcast them everywhere. He can also target a group of users, who he knows installed the app in some particular time frame.

- After generating an ID, the app attempts to register it with the server. It generates a FCM token and associates the ID with the token. I bet that the server uses this to send infected IDs and other notifications to the app. If the registration fails because the ID already exists, the app doesn't generate another ID and try again. Because I can predict all future IDs, I can preregister all of them. This is a denial of service attack, denying all future users to participate in this system.

3/ Acute readers probably have noticed that the IDs are too short! There are only 36^6 ~ 2^31 IDs. Because of the birthday paradox, if 65K people use the app, two of them will be assigned the same ID with high probability. When so, if a person is declared infected, the other is also too! At the current growth rate, I think the first collision will happen within 2 weeks.

The developers rebut that they've ensured that no collision is going to happen, regardless of the probability.

Because the app tries to register the ID with the server, it's possible to handle collisions, by checking whether the ID already exists on the server database. I don't think the Android app is doing that (I haven't looked at the iOS app).

The developers also claimed that their system can solve a problem that many other teams around the world can't. That is, they can prevent abusers from replaying IDs collected at a hospital. How? Users can optionally upload their observed IDs to the server. Wait, what? This is contradict to the "no data collection" claim. You don't want to upload observed IDs to the server.

4/ The app backups to Android's public storage (a.k.a external storage) the database of all observed IDs and the MAC address and the device name of observed Bluetooth classic

devices (yes, it also scans for them -- this is the source of the confusion in #1?). This means, information correlated to the user's location, movement and social graph is accessible to other apps installed on the phone that can read external storage. If the backup database is used by health authority to determine who to quarantine users can easily falsify exposure records.

In my initial report, I didn't know that they did this but I still flagged the external storage permission request because I don't think a contact tracing app needs access to photos, media and files to do its job. This is the principle of least privilege.

They rebut saying that requiring external storage access is not a security issue, and expressing surprise why a Google engineer doesn't know external storage is where Android stores photos, media and files. Well I flagged the permission request exactly because I *knew* external storage has sensitive information.

The app also requires access to fine-grained location information. Now, this is something I didn't know. I flagged this permission request, but it turns out that for privacy Android requires the ACCESS_FINE_LOCATION permission if an app wants to use Bluetooth.

--

Epilogue

The boss of the developers emailed asking me to help secure the app. I said I'd love to, if they move development to GitHub. This is what they've pledged since day one. The boss said they're going to upload today, but I haven't seen it yet. Stay tuned for more fun!

Update: fix some typos and remove unnecessary languages.

Update: fix more languages. The app now has 150K users. The good news is the growth rate has slowed down. My friends and I have filed several bugs on the app's GitHub page, but got zero response so far.

(Discussions on Hacker News)

English

Normal person said...

Thanks Thai, birthday paradox is new for me, so interesting.
You're doing great work, keep going.

2:36 AM

Unknown said...

Hi Thái,

You said "If the registration fails because the ID already exists, the app doesn't generate another ID and try again". If so there cannot be 2 users with the same ID registered in the server, as the 2nd one would have been unable to start the app at all?

Minh

2:54 AM

icemain said...

As I thought yesterday, you have captured the BZ genID algorithm.

5:47 AM

Hà said...

This comment has been removed by the author.

6:31 AM

Hà said...

Hi Thái,
Về vấn đề chống replay/relay attach ở mục 3, mình hình dung cách làm của BlueZone là tăng cường điều kiện xác định F1, cụ thể là chỉ khẳng định đúng F1 nếu thoả mãn cả 2 điều kiện sau:
1) Trên máy F1 có chứa một số token phát ra từ F0, và
2) Trên máy của F0 có chứa một số token phát ra từ F1
Như thế, nếu là relay/replay attach thì chỉ thoả mãn điều kiện 1 mà không thoả mãn điều kiện 2.
Mình có nhầm lẫn gì không nhỉ? :)

Ngoài ra, vì Android yêu cầu quyền truy cập vị trí khi dùng bluetooth, nên các nhà thiết kế Android cho là đã cho dùng bluetooth thì mặc nhiên người dùng lộ dữ liệu vị trí? Và nếu thế thì nên chăng BlueZone dùng luôn dữ liệu vị trí của người dùng?

Hà,

6:45 AM

Thai Duong said...

>You said "If the registration fails because the ID already exists, the app doesn't generate another ID and try again". If so there cannot be 2 users with the same ID registered in the server, as the 2nd one would have been unable to start the app at all?

When registration fails, the app just silently cruises along. It can still scan and broadcast IDs, but because it can't talk to the server it won't be able to do any matching to detect exposure.

11:30 AM

Thai Duong said...

Hà: đó không phải là cách họ nói. Nguyên văn lời của họ

>Cụ thể giải pháp của nhóm cho phép khi 1 Bluezoner giả định có phát hiện tiếp xúc F0 (nhận được thông tin qua broadcast), Bluezoner này có tùy chọn xác minh F0 mình đã tiếp xúc có đúng là F0 thật hay không, bằng cách gửi lịch sử tiếp xúc F0 của mình lên hệ thống để so sánh với lịch sử tiếp xúc của F0 đã được cơ quan Y tế cập nhật. Nếu không có sự tương đồng, Bluezoner không phải là F1.

Vả lại có đúng là họ làm như vậy thì đây là cách làm rất nguy hiểm, vì nó bắt buộc người dùng chia sẻ dữ liệu với máy chủ, đi ngược lại với cam kết "không thu thập dữ liệu".

11:45 AM

icemain said...

They open the source https://github.com/BluezoneGlobal/

6:01 PM

Trung Doan said...

Người Việt chắc ko cần đọc tiếng Anh vì mấy bữa nay cái nhau bằng tiếng Việt thấy hai chiến tuyến toàn dùng lời lẽ chả đẹp tí nào: một bên thì kiểu như ban ơn và chỉ đạo lũ ngu cần làm theo lời vàng ngọc của anh, đẳng cáp của anh nó khác, đừng lèm bèm, bên kia thì bảo là sai đầy ra, nhiều chỗ ko biết thì đừng tinh tướng, chưa biết ai ngu hơn ai. Cộng thêm các comment của các ủng hộ viên mỗi bên, kéo dài vài bữa nữa thiên hạ lại đc xem thêm kịch, bi hay hài còn chưa biết.
Có thể sửa vài lỗi ngữ pháp tiếng Anh, bỏ đi mấy phần ăn thua và đưa lên các diễn đàn, mạng xã hội để có nhiều thảo luận có ích hơn. Mới đọc đc cái này thấy không khí nó khác ở đây quá:
https://twitter.com/matthewrdev/status/1254336105203200000

9:02 PM

slither said...

Hi a Thái,

> "Vả lại có đúng là họ làm như vậy thì đây là cách làm rất nguy hiểm, vì nó **bắt buộc** người dùng chia sẻ dữ liệu với máy chủ, đi ngược lại với cam kết "không thu thập dữ liệu"."
> "Bluezoner này **có tùy chọn** xác minh F0 mình đã tiếp xúc có đúng là F0 thật hay không, bằng cách gửi lịch sử tiếp xúc F0 của mình lên hệ thống để so sánh với lịch sử tiếp xúc của F0 đã được cơ quan Y tế cập nhật."

Theo mô tả ở trên là BlueZone có tuỳ chọn là có muốn xác minh với cơ quan Y tế hay ko mà anh, đâu phải bắt buộc?

9:11 PM

Thai Duong said...

slither: ý từ bắt buộc ở đây là để kiểm tra theo cách mà Bluezone nói thì phải chia sẻ dữ liệu với máy chủ. Thật tế thì không cần làm vậy. Máy chủ có thể đẩy dữ liệu F0 xuống cho người dùng mà không cần người dùng phải chia sẻ dữ liệu gì cả.

10:11 PM

Thai Duong said...

Trung Doan: I posted to Hacker News and Twitter. It was on the front page of HN for several hours.

I reread all my posts and comments. I never attacked people, I only attacked their work. The difference is subtle, but not difficult to notice if one paid attention. When I said they made basic mistakes, I presented hard, verifiable evidences. Nothing personal, purely business. I hate their work because it'll put other people at risk, but I don't hate them. If I met the team, I'd be happy to have a beer with them (but I doubt they'd want to ;-)

11:35 PM

tengicungduoc said...

Chuyện không liên quan, hôm qua em đọc blogspot bình thường thì nay nhà mạng (CMC) đã chặn blogspot :)))

11:36 PM

baotd said...

> Chuyện không liên quan, hôm qua em đọc blogspot bình thường thì nay nhà mạng (CMC) đã chặn blogspot :)))

Trang này bị chặn miết, đóng mở thất thường

12:20 AM

mathe said...

Not sure if it's suspicious, but it looks like from China: https://github.com/BluezoneGlobal/bluezone-app/commit/31db572e1fc01dba62083bd3520d96fc2c4a595e#r38764608. But it's not important thingy.

12:50 AM

Zic said...

Các chiên giá thay vì chém gió tung giời
Adua acong
Thì PR (pull request) trên github đi nào

Cheers

4:18 AM

VOV said...

Great report!

seedrandom function is being commented out from the latest source code (maybe after this report)
https://github.com/BluezoneGlobal/bluezone-app/blob/master/app/Configuration.js#L50

Here is the new function for it: https://github.com/BluezoneGlobal/bluezone-app/blob/15305e20da6619a069b1e023dc305dcc8ab984bd/ios/Scanner/BluezonerIdGenerator.swift#L15
(The equivalent native-lib for android has not been published yet I think) ~ Hopefully you can have a look to give our community a comment/assessment..

Technically speaking, please keep your great finding

8:20 PM

Thai Duong said...

The current version on Google Play released on April 23 still uses the seedrandom function.

BluezonerIdGenerator.swift looks fine, but the version on Android is broken: https://github.com/BluezoneGlobal/react-native-bluetooth-scan/issues/2.

8:31 PM

Tester said...

From Whitehat forum:
> Nhóm phát triển cho biết họ dùng hàm sinh ngẫu nhiên theo millisecond. Cần đoán chính xác đến millisecond của 1 người thì chúng ta tự biết là có thể làm điều đó hay không.

Safe and Secure PRNG: am i a joke to you?

12:07 AM

Thai Duong said...

Ha! Someone found a problem in the Swift version too!
https://github.com/BluezoneGlobal/react-native-bluetooth-scan/issues/5

1:39 AM

CuongLM said...

@Thai Duong:

> It uses this seedrandom library which always generates the same output when seeding with the same value

I think it's ok to use seedrandom, but the way they use is wrong. According to seedrandom document https://github.com/davidbau/seedrandom#script-tag-usage

> // Calling seedrandom with no arguments creates an ARC4-based PRNG
// that is autoseeded using the current time, dom state, and other
// accumulated local entropy.
var prng = new Math.seedrandom();
console.log(prng()); // Reasonably unpredictable.

and:

> // Warning: if you call Math.seedrandom without $\neq w$, it replaces
// Math.random with the predictable new Math.seedrandom(...), as follows:
Math.seedrandom('hello.');
console.log(Math.random()); // Always 0.9282578795792454
console.log(Math.random()); // Always 0.3752569768646784

so the problem is not about seedrandom, but about how they use it.

8:30 PM

Thai Duong said...

CuongLM: https://github.com/davidbau/seedrandom/blob/released/seedrandom.js#L193 is how seedrandom is autoseeded. I think when use in React Native, it is only autoseeded with the current time.

Javascript code in React Native can't generate secure random numbers, but must rely on native code in Android or iOS. https://github.com/rh389/react-native-securerandom sounds pretty good, but I haven't reviewed it carefully.

9:29 PM

CuongLM said...

@Thai Duong: Yes, but my point is that if they use seedrandom correctly, mean using $\neq wMath.\,seedr\ \text{and}\ om();$ instead of globally seed it, then the output won't be predictable anymore (even you use the same seed multiple times).

7:58 AM

Thai Duong said...

I suspect that on React Native $\neq wMath.\,seedr\ \text{and}\ om();$ is the same as $\neq wMath.\,seedr\ \text{and}\ om(\times tamp);$.

3:45 PM

Jonathan Katz said...

How can I send you email about Tink?

1:16 PM

Thai Duong said...

My work email is thaidn@google.com

3:10 PM

Post a Comment

Chiếu bí iPhone

About

Email: thaidn@gmail.com. Tôi sinh ra ở Sài Gòn, lớn lên trên Internet. Tôi là kỹ sư an ninh mạng, từng được trao giải thưởng danh giá Pwnie Award. Hiện tại tôi làm ở Google, với vai trò kỹ sư cấp hơi cao. Tôi từng là kỹ sư trưởng Google Tink, thư viện phần mềm mã hóa đạt được hơn ba triệu download mỗi tháng. Trong thời gian vừa qua tôi thấy hơi chán, chưa biết làm gì tiếp theo, nhưng tôi vẫn đang kiên trì lãnh lương hàng tháng.

I'm a security researcher, best known for my SSL attack trilogy: BEAST, CRIME, and POODLE. I was half of the team that discovered the crypto vulnerability in ASP.NET affecting millions of websites. For this work I was a recipient of the prestigious Pwnie Award. Currently I'm a senior member of the worldwide product security team at Google. My team built Project Wycheproof and Tink. My mission is to help everyone on the Internet use cryptography correctly.

Popular Posts

BEAST

September 25, 2011

So we gave a talk and a live demo at

READ MORE

Nói chuyện tại Hội đồng Lý luận TW

October 17, 2022

Hồi cuối tháng 8/2022, nhân dịp đang ở Hà Nội,

READ MORE

Làm an toàn thông tin thì học gì?

May 02, 2012

1 Giới thiệu Tôi nhận được thư từ của nhiều bạn hỏi về việc nên học gì và như thế nào để có thể tìm

READ MORE

Library

Security Research

The POODLE attack
The CRIME attack
BEAST: Surprising Crypto Attack Against HTTPS
Cryptography in the Web: The Case of Cryptographic Design Flaws in ASP.NET
Practical Padding Oracle Attacks
Flickr's API Signature Forgery Vulnerability
Zombilizing The Web Browsers Via Flash Player 9
Giám sát an ninh mạng
Một phương pháp chống DDoS bằng xFlash
Lỗ hổng nghiêm trọng của SSL/TLS