The Banana Phone Project:

A Technical Proposal for the Solution of Illegal Robocalls

Alexander E. Ruiz

ABSTRACT

The current problem of illegal robocalls is one of intelligent software attacking archaic receivers. The telephone line present in houses and businesses today is a terminal in the strictest sense; it only passes information, with no checks or filtering of any kind.

Furthermore, with the given state of IP-based telephony software, all incoming caller information data can, and most times is, falsified by the perpetrating robocall originators. Therefore, any attempt to track or “blacklist” suspect caller data for later reference and potential blocking is completely futile.

Given these facts, the ideal system for preventing robocalls will be a “whitelist” oriented system, with human validation of all potential callers prior to full call completion.
This system should also have a method for bypassing the human validation method completely in order to send legitimate robocalls through unhampered.

The Banana Phone Project has taken all of these considerations into account and in this document presents a solution to the problem of illegal robocalls with complete solvency based on the given criteria by FTC.

GENERAL DESCRIPTION

The key logical components of the Banana Phone Project (“the System”) are:

- The Human Validation Process, that tests all incoming callers unknown to the System for human presence;

- The Local User Database, which keeps track of all data relevant to the System, including caller IDs that have human validated previously (local whitelist records), globally distributed whitelist records, and incidents of attempted and successfully passed robocalls;

- The Global Database, which is responsible for receiving robocall incidents reported by System users, delegating distributed whitelist rules to specific groups of users of the System, and managing the authentication codes pertinent to non-internet connected users.

HUMAN VALIDATION PROCESS

Human validation is present in the System throughout the entire duration of any call being received. Once a call is received on the phone line, the caller information data is first parsed to ensure that it looks like a valid phone number (ex. not blank, not repeating of a single digit over and over, etc.). The System then takes the incoming caller identification (CID) data and attempts to match it against all active whitelist records on the Local User Database. If a match is found, the call is then passed to the live phone line, and the user's attached phone rings as normal. If the CID is found to be invalid or a match is not found on any active whitelist records, the call is then pushed to the Trap Line for human validation.

THE TRAP LINE

Once a call is received on the Trap Line, a prompt plays explaining that the phone line is equipped with a filter for automated telemarketer calls. The caller is then asked to listen to a sound clip and enter a four-digit number sequence that is embedded within the audio using their phone's keypad.

This four-digit random number sequence is an authentication code that is randomly generated with every phone call received on the Trap Line. This sequence is then said aloud and mixed over audio to prevent machines from attempting to listen and pull the numbers for falsifying a human caller. The caller has fifteen seconds from the end of the sound clip to attempt to enter the correct number sequence. Upon successful entry, the call is passed to the live phone line. If the caller enters the wrong input or the call times out, an incident is recorded on the Local User Database, and the call is hung up. It is to be noted here that should a human caller enter invalid input or make a mistake on the Trap Line, all they have to do is simply try to place the phone call again. The System does not blacklist any number that fails the human validation test whatsoever. It only reports incidents of failed validations.

USER CALL FLAGGING

Should a robocall somehow get by the human validation process and pass to the live line, the user is able to flag the call as a special incident by entering the "call flagging key sequence" on their phone's keypad (in our testing implementation, *44). Once this key sequence is entered, the call is immediately hung up on and an incident record is created on the Local User Database with a special code for notifying the Global Database of a breach in current caller validation methods. With that stated, no robocall has beaten the Trap Line to date.

AUTOMATIC WHITELISTING

When a valid unknown CID has successfully traversed the Trap Line with the call completing normally (not flagged by the user), the System will automatically add the CID to the Local User Database with a local whitelist record by default. For high call traffic users or businesses, this feature may be turned off at any time. Local whitelist records are kept on the System indefinitely, and allow the CIDs present on it to pass through to the live line uncontested should that number call again. Locally whitelisted CID are not shared with any other users of the System, via Internet connection or otherwise.

SYSTEMIC FLAWS IN THE WHITELIST-ORIENTED APPROACH

As with other areas of communication security, we must assume that these robocall attacks will attempt to improve and get around these newly implemented security measures before (if?) they stop entirely. We must consider not only the current problems, but also potential future attacks as well. In keeping a whitelist-based system, the largest issue that threatens the overall reliability of the given solution is one of "known good numbers."

Given that the System only allows numbers to pass through to the live line if they are present on the user's whitelist, there is a potential for demand of validated numbers within a given area code/prefix among these illegal robocalling parties to continue to harass consumers. With that said, there is enough data entropy present in each individual user's whitelist to warrant phone line brute-forcing impractical for robocaller's needs. However, the System must also be able to whitelist numbers that leverage legitimate robocalls for vital services, such as medical reminders, utility bills, and the like. Consider the implications if robocalls attempt to spoof the phone numbers of local hospitals, libraries, municipal offices, etc. This inherent whitelist security issue has been addressed with absolute solvency in the Banana Phone project with a process called Automatic Dialing Authentication.

AUTOMATIC DIALING AUTHENTICATION

The process for passing legitimate robocalls within the System is consists of two methods; one for users with internet connection, and one for users maintaining only a phone line.

THE KEEP-ALIVE WHITELIST METHOD

The Keep-Alive Whitelist method consists of pre-authenticating potential incoming calls for bypassing of the Trap Line by means of a remote delegation authority that passes whitelist rules through an internet connection. Contained within these rules is the caller identification data for said incoming call and a timestamp associated with the rule. This timestamp denotes the period for which the given rule is to remain valid when distributed to the Local User Database of a System. Timestamps can be anywhere from 5 minutes to several weeks, depending on the needs of the particular requester of the whitelist rule. The Global Database is responsible for handling the distribution of these global whitelist records to the appropriate users of the System. The Local User Database imports these global whitelist records and checks these global records for timestamp validity over the course of System use. If any global whitelist rules are found to be expired, they are marked as inactive and are no longer referenced when the System does a whitelist check on unknown CIDs.

BLIND DIAL CODE AUTHENTICATION

The second method provides a way to bypass the Trap Line during the actual human validation process itself. The System generates a random eight-digit “Blind Dial Code” that is stored on the Local User Database. The method is called “blind” because the System has no prior knowledge of the call before it hits the phone. When a call is received on the Trap Line, the caller can enter the four-digit authentication code that is played over audio or they can enter the eight-digit Blind Dial code if they know it. The Blind Dial code is regenerated and updated to the Global Database over an arbitrary time interval (for testing purposes, it is 2 weeks). The updating process for the Blind Dial Code is done entirely over the existing telephone line, with the System calling into the Global Database via a VoIP line. The Blind Dial Code serves dual purposes, acting as a non-networked alternative to the Keep-Alive Whitelist method and as a redundant means of authentication should a System lose internet connectivity.

EMERGENCY SERVICES CONSIDERATIONS

Emergency services must be taken into account for this solution to be complete. The System watches for outbound dialing of 911 and/or other emergency numbers. Upon detection, it places a local keep-alive whitelist rule in the Local User Database to allow any call to pass through unfiltered for a user-set duration of time. Considering the

immediacy of emergency calls and their services, a valid time frame of an emergency keep-alive rule could be anywhere from thirty minutes to several hours depending on the specific needs of the user.

SYSTEM IMPROVEMENT

Incident logging is implemented on the System for learning how these robocalls attack users. Incident records are accumulated on the Local User Database and exported to the Global Database for analysis every so often at an arbitrary time. In our testing, the time frame for incident exporting is the end of every week. A typical incident record will include:

- start/end time of the call
- date
- any CID information passed (including the text description)
- any DTMF signal input detected if the call hits the Trap Line
- The condition of the incident generation - call timed out, user-flagged, incorrect input on Trap Line, etc

With collection and analysis of these incident records and updatable software via an ethernet connection, the solution has the ability to improve if and when these robocall attacks choose to variate their methods.

IMPLEMENTATION

The software for the Banana Phone project is built from open source pieces and is hardware agnostic in design. Because of this, performance requirements for both home and enterprise can be scaled to the needs of the user by increasing hardware capacity alone. The Banana Phone landline prototype has been implemented and is currently in distributed beta testing on a minimalist hardware set up for maximum cost efficiency. Built from off-the-shelf parts, the hardware includes:

- an Analogue Telephone Adapter (ATA) - for allowing landlines to hook into VoIP-enabled systems

- a micro-embedded, ARM-based communications server - for telephony processing, call logic handling, and database integration

The System also requires a network switch for allowing the ATA and the server to send traffic between each other to facilitate the call. The solution, as of this writing, can tie into any existing landline or VoIP system and start filtering robocalls immediately. Double solvency is achieved for both landlines and VoIP servers with the same solution due to the fact that the unit sits between the Telco provider line and the user's phone with being a miniature VoIP server itself. The cost per unit in the prototyping stage (not mass production) is under $120. The System is also end-point distributed, being connected to the user's phone line after the Telco line ends and before it ties into the user's home phone.
This way, there not more infrastructure that needs to be added to any existing Telco internally. It is a consumer-based solution entirely.

SOLVENCY

In our current testing, the Banana Phone project has yet to be beaten out by any robocall methods to date. Test users of the solution are finding that their business productivity, once plagued by robocalls, is now back to normal. The difference in phone lines being bothered by robocalls is immediately apparent once the System is installed. Users are very satisfied with the current functionality that the Banana Phone project offers, and the project is still the beginning beta testing stages.

JUDGING CRITERIA FOR THE COMPETITION

Does it work? (weighted at 50%)

Q: How successful is the proposed solution likely to be in blocking illegal robocalls?
A: The Banana Phone project maintains 100% solvency in blocking unwanted robocalls using the Human Validation Process coupled with Automatic Dialing Authentication outlined in the design document. All incoming robocalls are stopped when they hit the Trap Line.

Q: Will it block wanted calls?
A: Leveraging the Keep-Alive Whitelist method and Blind Dial Codes, the System does not block any legitimate robocalls attempting to be made. In our testings, the Keep-Alive Whitelist works completely in successfully passing calls through the Trap Line. Blind Dialing Authentication works as well, with the same success rate. Emulating emergency call back scenarios, the Banana Phone performs as claimed, allowing all calls to pass through for a pre-set period of time as required by individual users.

Q: How many consumer phones can be protected? What types of phones? Mobile phones? Traditional wired lines? VoIP land lines?
A: The Banana Phone project is able to be installed on both VoIP and land lines as of this writing and will start building its whitelists and filtering calls immediately.

Q: What evidence do you already have to support your idea? Running code? Experiments? Peer-reviewed publications?
A: The evidence we have to support our data is a working proof-of-concept device with running software that portable and hardware agnostic. We also have logged data of attempted and failed robocalls hitting our filtering system.

Q: How easy might it be for robocallers to adapt and counter your scheme? How flexible is your scheme to adapt to new calling techniques? How have you validated these points? Remember that the real test of a security system is not whether or not you can break it; it's whether or not other people can.
A: As of current testing, there is no robocall method that has been able to beat the Banana Phone Project as it stands today. Proper steps have been taken to ensure that machine listening efforts will be useless. Future developments include implementations of a second method of human validation, with asking the caller to answer very simple arithmetic questions in order to validate human presence (Ex. 23 + 1, 654 - 1, etc). These questions will be variable length answers, between one four digits, and the System will randomly choose between one of the two validation with every phone call received. If robocallers wish to beat our scheme of human validation, they will have to have a human contest the call at some point in the call process. And if that scenario ever takes, the war on robocallers will be over. With that said, the incident logging coupled with the tracking of global whitelist records have enough granularity in data gathering to give us enough information to see how these robocallers are penetrating the system, and then to patch with updates.

Is it easy to use? (weighted at 25%)

Q: How difficult would it be for a consumer to learn to use your solution?
A: The System works in exactly the same function and capacity as a normal phone line. The one difference being that the user must remember to enter the special call flagging sequence on their keypad if they ever receive a robocall that is passed to the live line. in our testing, users have no issue with this small learning adjustment.

Q: How efficient would it be to use your solution, from a consumer's perspective?
A: The Banana Phone Project solution is very efficient to use from a consumer's perspective, as it requires little to no effort to learn how to operate. It works exactlt he

same as a normal phone. The System was designed to be as transparent to the end user as possible, with users of System only needing to remember the call fagging key sequence. Legitimate human callers will be hassled at a minimum amount until they validate human presence once on the System. After this takes place, the Banana Phone is transparent in usage to both the calling party and the user of the System.

Q: Are there mistakes consumers might make in using your solution, and how severe would they be?
A: The only mistake that a consumer is able to make on the System currently is to fail to human validate when they are contested on the trapline. With that said, all the user has to do is place the call again and human validate. The System will whitelist the caller as normal. The Banana Phone project was designed this way on purpose to be very forgiving of caller input error.

Q: How satisfying would it be to use your solution?
A: The test users in the beta group for the Banana Phone project are extremely satisfied with the results they are getting from the System. The results are immediately apparent in most cases, with robocall stopping as soon as the System is installed.

Q: Would your solution be accessible to people with disabilities?
A: The great part about the System is that is very much an “in-between” device. It only serves as a filter between the Telco provider and the actual device handset, so any and all currently existing phone handsets that cater to people with special needs can be hooked into it with no loss of functionality on either the handset or the System itself.

Can it be rolled out? (weighted at 25%)

Q: What has to be changed for your idea to work? Can it function in today’s marketplace? (E.g., Does it require changes to all phone switches world-wide, and require active cooperation by all of the world’s phone companies and VoIP gateways, or can it work with limited adoption?) Solutions that are deployable at once will be more heavily weighted, as will solutions that give immediate benefits with even small-scale deployment.
A: No existing infrastructure needs to be changed for the System to be rolled out. It is end-point distributed by design and can work with very limited adoption. The Banana Phone is already in active beta testing and is receiving overwhelmingly positive results. The prototype unit is cost viable for consumers now, and that cost can only go down if it hits mass distribution. Furthermore, the software can be implemented on existing VoIP devices and servers. In a practical instance, cable and satellite providers who also give

voice services to their customers could readily implement the solution and have a new service available for their existing consumer base. The Banana Phone project is already functioning in today's marketplace and has immediate benefit in small scale distribution, as shown by the current beta testing results.

Q: Is deployment economically realistic?
A: To put it bluntly, yes. It is absolutely realistic. This project has been designed, implemented and tested by a single person with the intent of creating a complete working solution that is software-based, hardware agnostic and can be rolled out with minimal wait time on commercial polishing. To be honest, I don't know how it couldn't be rolled out.

Q: How rapidly can your idea be put into production?
A: The project is currently under active development and should be ready for commercialized distribution