Select Caller Verification Software

Antonios John Bokas
Beacom College of Computer and Cyber Sciences
Dakota State University
Madison, South Dakota
tony.bokas@trojans.dsu.edu

***Abstract*—Vishing is an unresolved cybersecurity threat that targets customers with social engineering techniques. Existing countermeasures for phishing, spam, and scams appear to be increasingly effective, but countermeasures against vishing, especially live interaction vishing, are largely absent. In this article, I discuss the existing literature about vishing, document a diminutive, but unique, case study, and propose an open-source software design that allows customers to authenticate callers through a two-factor authentication scheme.**

Keywords—Vishing, caller identification (caller ID), two-factor authentication, one-time password (OTP)

I. Introduction

It is time for organizations to take self-authentication matters into their owns hands. When a customer calls a typical institution or bank, the institution scrutinizes his identity for up to a minute before it authenticates and assists him. For example, a representative asks the customer to provide his name, address, phone number, partial social security number, or telephone password. But when a representative calls a customer, he typically does not provide any credentials to the customer to authenticate himself. Ergo, the organization protects itself—but not the customer—from impersonators.

This oversight has inadvertently fostered a cybersecurity threat called vishing. *Vishing*, which is a blend of the words “voice” and “phishing,” is when an impersonator calls a customer, often through Voice over Internet Protocol (VoIP), and fraudulently identifies himself as a trusted party in order to extract vital information from the customer [1, 2]. Unlike a live, social-engineered vishing attack, *phishing* is primarily conducted through mass email and text messages [3]. Thus, the objective of a broad phishing campaign is to catch “any fish,” whereas the objective of a targeted vishing attack is to catch a “particular fish.”

To address this problem, I performed a literature review, conducted a rudimentary case study, and outlined software that counters vishing attacks by placing authentication power in the hands of the target. In section II, I describe the essentials of my literature review. In section III, I analyze an existing case study as well as my novel case study. In section IV, I propose my software design. In section V, I discuss the implications and limitations of my software design. And in section VI, I summarize my findings and suggest a course for future research.

II. Literature Review

A. Research Method

Before my formal literature review, I consulted textbooks, commercial literature, and gray literature for fundamental information about vishing. After I conducted my case study, I generated the following search phrase to collect pertinent articles about vishing attacks: (“vishing” OR “phone scams” AND financial AND cybersecurity AND compromised). I later modified the search phrase as necessary to broaden my results. In addition to technical papers, I reviewed business journals for cybersecurity-related topics in order to comprehend the issue from multiple perspectives. I also corresponded with a journal author that develops commercial identity and reputation management products. This helped me focus my research and algorithm design on realistic anti-vishing applications.

B. Findings

For over a decade, social engineers have ensnared their victims with vishing typically by one of three mechanisms: Interactive Voice Response (IVR), live interaction, or a combination of IVR and live interaction [3]. According to a multivocal literature review by computer scientists Mohammed Hijji and Gulzar Alam, vishing was the fifth most-commonly used social engineering technique during the COVID-19 pandemic through 2020 [4]. Per their analysis, social engineers used vishing 9.2% of the time and phishing (which was the most used technique) 35.3% of the time. While their review thoroughly classifies social engineering techniques, it does not describe the effectiveness of each technique. However, they do attest that a social engineer must possess significant knowledge about a victim in order to successfully trick him during an attack. For example, they explain that a social engineer collects information about a victim through online searches, evaluates the victim’s security system, and analyzes the victim’s personality and behavior before he launches an attack.

Despite the prevalence of vishing, countermeasures for it lag behind those for automated threats such as Short Message Service (SMS) phishing. In 2018, researchers from the Information Processing and Communications Laboratory at Télecom Paris created a machine-learning program in Python that vectorized SMS messages and correctly identified phishing messages at a rate of 86.17% to 90.65% [5]. They claimed their program could be adapted to counter vishing frauds. However, they did not delineate which type of vishing frauds it could counter. According to my investigation, their style of program would most likely be effective against IVR and hybrid IVR-live interaction attacks but would not be effective against purely live interaction attacks. Apart from program mechanics, the reason why programs typically cannot thwart a live interaction attack is due to its dynamic nature. A live interaction attack typically utilizes two tactics: 1) a mimicked, or *spoofed*, phone number; and 2) a dialogical stratagem. Popular countermeasures do not consistently detect spoofed numbers nor determine if callers have malicious intent based on their phraseology. This is the feature of live interaction vishing that defeats adaptable machine-learning programs.

Another reason why phishing, spam, and scam messages are easier to counter than vishing calls, according to Eric Burger, the former Chief Technology Officer for the U.S. Federal Communications Commission, and technologist Jim McEachern, is that filters for plaintext communications such as email scan such messages before they deliver them to the recipient [6]. However, under an emerging telephony framework that was approved by the U.S. Congress in 2019, phone call filters will rely on phone carriers to determine a caller's trustworthiness. Burger and McEachern explain that the anti-spoofing framework, which is laboriously named Secure Telephone Identity Revisited/Signature-based Handling of Asserted information using toKENs (STIR/SHAKEN), requires phone carriers to issue digital certificates to authenticated callers in order to establish an attestation score for each inbound call. Phone carriers will thus work to authenticate and maintain a public key for every caller. The benefits of this system are manifold and include the ability for phone carriers to trace a fraudulent call back to its source and improve the accuracy of caller identification (caller ID).

However, the STIR/SHAKEN framework has some shortcomings that should urge organizations to still implement their own vishing countermeasures.

1) STIR/SHAKEN is not a distributed system (it will be operated by a few major phone carriers).
2) It does not block fraudulent calls (it only identifies authenticated callers; it does not stop unauthenticated callers).
3) It is only being implemented in the U.S. and Canada (many scammers call North American residents from international locations [6]).

Despite the advent of STIR/SHAKEN, other researchers have proposed novel vishing countermeasures. The following experimental program, designed by Sumitra Biswal, a software engineer and artificial intelligence researcher, indeed analyzed a caller's phraseology to determine his intent [7]. Her program was called the Real-Time Intelligent Vishing Prediction and Awareness Model (RIVPAM). RIVPAM used pattern identification, contextual sentiment analysis (through natural language processing), and prediction training to determine if a live caller was a potential threat and then warned the call recipient in real-time if it detected fraud. Her test program successfully identified 65% of 112 vishing audio samples. Older studies proposed models that hearken to STIR/SHAKEN.

For example, as far back as 2008, researchers from Bell Labs (which was owned by Alcatel-Lucent but later purchased by Nokia in 2016) proposed the utilization of local trademark offices as certification authorities for commercial phone numbers [8]. They used a three-part administration model, called RealName, that managed, issued, and authenticated caller certificates for a SoftPhone call (a desktop-based VoIP call) via Session Initiation Protocol (SIP) in 0.3 to 2.5 seconds. They even integrated a "prove it" button that a call recipient could push to request certificate authentication during the phone call. The researchers conceded that their model was limited by the ability of a certification authority to maintain its registry, verify trademark owners, and connect to other, remote registries. It is unclear whether they conducted additional tests with their model. However, RealName still seems ahead of its time.

Considering its massive scale, even the lauded STIR/SHAKEN framework will suffer from deficiencies that are similar to those experienced by RealName. In fact, the propositions by Biswal and Bell Labs seem to undermine an assertion by Burger and McEachern that the "next best thing" to "[disclosing] the content of a call before it's connected" is to "track calls from the point where they physically enter the network . . . and then establish a caller's reputation" [6]. Biswal and Bell Labs developed programs that veritably authenticated the identity of a caller and even evaluated his intentions.

III. Case Studies

A. *Existing Case Study*

During my literature review, I searched for studies that cataloged vishing attacks. For example, in 2010, Frederico Maggi, a researcher from Milan Polytechnic, developed the PhonePhishing.info repository, in which he collected approximately 360 user-submitted vishing reports [9]. He analyzed the caller ID, subject, date and time, and national origin of each call and identified the most-frequently used phone numbers, prefixes, and words. For example, the top prefixes were 800, 866, and 877. The top three words in an IVR attack were "number," "account," and "credit." And the top three words in a live interaction attack were "number," "credit" and "person." Although it is somewhat dated, his case study confirmed that historical vishing attacks: 1) tried to impersonate businesses; and 2) were financially motivated. During the initial stages of my algorithm design, this case study supported my plan to design software that focused mainly on financial attacks.

B. *Novel Case Study*

Apart from a review of existing case studies, I also used personal contacts to obtain participants for my own survey. Given my limited resources and potential biases, my case study merely served as a tacit catalog of vishing tactics. I used it to refine my initial algorithm and outline further areas of exploration. My case study also provided recent and colloquial accounts of vishing from the perspective of the targets. Their viewpoints helped ground my research in practical application.

Each participant was selected irrespective of their gender, ethnicity, income, and other socioeconomic factors and had been previously targeted in a vishing attack. However, a language barrier inhibited communication between myself and two qualified, potential participants. Thus, I surveyed a total of five participants. Each participant answered a 10-question survey (given in the Appendix), which I developed during my literature review. Three participants answered the survey questions, which I paraphrased and transcribed, via phone call. Two participants answered the survey questions electronically.[1]

Due to the dearth of participants and the experimental nature of the survey questions, I applied intelligence techniques during my analysis. Based on my experience as an intelligence analyst, I find that these techniques help analysts properly evaluate data

[1] I included my own vishing experience in the case study.

significance. Thus, after the surveys, I reviewed the responses of each participant and summarized them into a textual model as described by former intelligence officer Robert M. Clark [10]. A *textual model* quickly conveys intelligence to decision-makers. I then adapted another technique called a *threat capability statement* to summarize the results of each survey question (see Table I) [11].

TABLE I. VISHING CASE STUDY RESULTS

Question	Topic	Analytic Statement
1	Qualification	All of the participants had been targeted by a vishing attack.
2	Pretext	The pretext of most calls was financial, some pretexts were social, and few were legal or technical.
3	Spoofing	Most attackers used a variation of spoofing and some did not.
4	Caller type	Most participants interacted with a bot first and some interacted with a real person first.
5	Call result	Most attacks were unsuccessful and few participants were compromised.
6	Reaction	Most participants just hung up, some tried to learn about the attack, and few took preventive measures.
7	Past security	Most participants were certain they had not been compromised before and some were unsure if they had been compromised.
8	Social media	Most participants did not use social media and some did use social media but posted no sensitive information.
9	Prior robocalls	Most participants frequently received unwanted calls prior to the attack and few received no unwanted calls.
10	Organizational response	Most targets did not get help from an organization after the attack and few did get help from an organization.
Scale (high to low) = All, Most, Some, Few, None		

My case study reveals that a typical vishing attack: 1) is generic; 2) is financially motivated; 3) utilizes spoofing, IVR, and live interaction; 4) is unsuccessful; 5) is tolerated by call recipients; and 6) is somewhat sophisticated. In other words, social engineers rely on a lack of countermeasures, an aggressive campaign, and passive targets for a few chances at success. It is notable that one participant downloaded an app called Robokiller, which eliminated approximately 98% of her unwanted calls. Another participant did not use social media, yet the social engineer knew his bank and partial social security number prior to the vishing attack. One participant was burglarized 17 years prior, in which burglars stole sensitive documents from her home. She was unsure if the theft inspired later vishing attacks.

C. Analysis

Together, these case studies solidified my assumption that vishing is a major, unresolved cybersecurity threat. They also revealed that an anti-vishing program may not need to counter all types of unwanted calls, such as spam and scams. Machine-learning applications such as Robokiller appear to be very effective against spam [6, 12]. Thus, if the most dangerous vishing attacks involve live interaction and impersonation of organizations that manage credit cards, accounts, finances, etc., an algorithm could be specially designed to help such organizations counter only those types of attacks and yet still be worthwhile.

IV. SOFTWARE DESIGN

To place the power of authentication directly into the hands of organizations and their customers, I propose a Select Caller Verification Software (SCVS) design (given in the Appendix). It is similar to RIVPAM and a distant cousin of RealName. Administratively, SCVS is managed by a sponsor, such as a bank, institution, or other organization, and incorporated into its existing applications. It is not intended as a stand-alone application. Technically, SCVS has two primary jobs:

1) To identify and block known scammers through the use of a blacklist;
2) To verify callers that identify as the sponsor with two-factor authentication (2FA) and end the call if they do not provide legitimate credentials.

The SCVS architecture consists of three main elements:

1) The main and authentication servers;
2) The customer-client, which refers to a customer's mobile app;
3) The caller-client, which refers to a VoIP or Direct Inward Dialing workstation of an authorized in-house or third-party agent.

Direct Inward Dialing (DID) refers to the traditional "landline" phone numbers of an organization [13]. DID contrasts with VoIP, which uses internet protocol to initiate and transmit phone calls. Both types of calls can route through an organizational *private branch exchange* (PBX), which is a system that relays analog and digital signals to phone carriers for retransmission to the call recipient [13, 14]. However, VoIP has now surpassed DID as the prevalent methodology used in enterprise call centers. Thus, SCVS must be compatible with various types of caller phones and workstations.

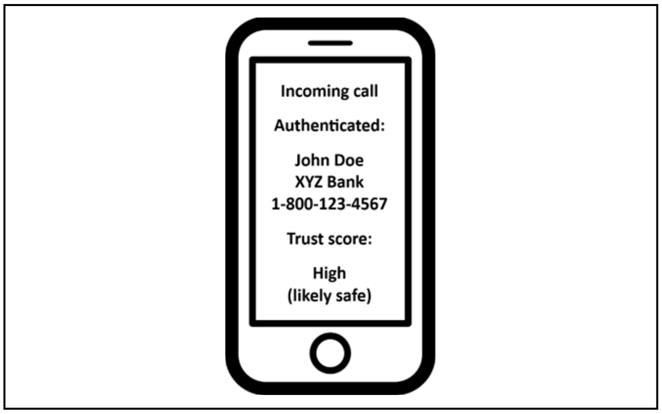
A. Customer-Client Side

On the customer-client side, the customer first sets an 8-character password in accordance with section 5.1.1.1 of NIST Special Publication 800-63B [15]. Then, in accordance with section 5.1.5.1, the customer-client key is seeded with a nonce as well as the customer password to generate a 32-bit one-time password (OTP). The OTP is then sent to the authentication server. The customer-client then enters a standby mode to wait for an inbound call and software updates and new blacklistings from the main server. Standby mode also resets and resends the OTP to the authentication server every 24 hours. Once an inbound call is detected, the customer-client exits standby mode and enters a screening mode.

In screening mode, the customer-client first evaluates the caller ID of an inbound call. As explained by the Bell Labs researchers in [8], a caller ID actually consists of two distinct elements: 1) a display-name; and 2) a phone number. Thus, the customer-client parses the caller ID name and number to determine if the caller is blacklisted. If so, the call is declined. If not, it determines if the call may be from the sponsor. If the call is not possibly from the sponsor, the call is allowed to pass to the customer without further evaluation (in the form of a ring, vibration, etc.). If the call is potentially from the sponsor, the customer-client recalls the last forthcoming caller ID (which, as I will later describe, was sent to it by the authentication server) and verifies if the inbound caller ID matches the expected caller ID. If the inbound caller ID is incorrect, the caller is untrusted and the call is declined. If the caller ID is correct, the customer-client answers the call for the customer.

At this point, the customer is still unaware of the inbound call, because the customer-client has not yet notified him of the transmission. Regardless, the customer-client then initiates an IVR prompt (given in the Appendix) that requests the OTP from the caller. After the prompt, the customer-client: 1) records the incoming audio signal for a number of seconds; and 2) determines if the caller responded. If the caller responded, the customer-client: 1) confirms the caller's response with another IVR prompt and recording cycle; 2) processes the response with voice recognition software; and 3) determines if the OTP is correct.

If the OTP is correct, the customer-client displays the verification results and a trust score on-screen and allows the call to pass to the customer (in the form of a ring, vibration, etc.). An example notification is, "Caller authenticated: John Doe, XYZ Bank, 1-800-123-4567, Trust score: High (likely safe)" (see Figure 1).



Incoming call

Authenticated:

John Doe
XYZ Bank
1-800-123-4567

Trust score:

High
(likely safe)

Fig. 1. Example notification after authentication.

If the caller: 1) does not verbally respond to the IVR; or 2) provides an incorrect OTP, the caller is untrusted and the call is ended. In sum, the detailed 2FA procedure, as shown in Figure 2, is as follows:

1) While in standby mode, the customer-client sends an OTP to the server;
2) The server then;
   a. receives the caller ID of a trusted caller-client;
   b. sends the caller-client's caller ID to the customer-client;
   c. sends the OTP to the caller-client;
3) The caller then;
   a. calls the customer;
   b. authenticates himself to the customer-client;
   c. connects to the customer.

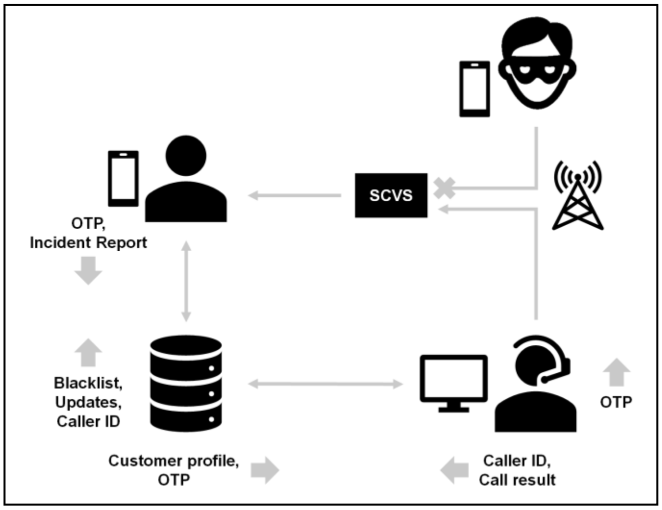


Fig. 2. Visualized 2FA procedure.

In the case of an untrusted caller determination, the customer-client generates an incident report and immediately sends a new OTP to the authentication server. The report contains all relevant data about the untrusted call and is relayed to the main server. Cybersecurity technicians from the sponsor organization may then review the report and take relevant action such as adjust local settings, update the blacklist, etc. The customer may also view the incident report and add commentary to it by checking a local application log. Regardless of a trusted or untrusted determination, the call result and customer-client activity are logged and available to the customer. The sponsor must determine the legal and appropriate privacy policies for SCVS upon its adoption and obtain consent from the customer when necessary.

In essence, the verified caller-client caller ID is a "what you are" authenticator and the OTP is a "what you know" authenticator. This mechanism is effective, because a social engineer: 1) cannot send a caller ID to the customer-client via the authentication server; and 2) does not possess the OTP. Furthermore, in addition to the notification illustrated in Figure 1 (which depicts a trusted caller scenario), the customer is also notified if an inbound call is declined or ended due to an untrusted determination.

These positive and negative confirmations allow the customer to intelligently evaluate inbound calls. Notice that the customer-client does not label an authenticated caller as completely "trusted." This is a subtle, but important, feature that allows the customer to decide how much he should trust an authenticated caller. Cybersecurity is not permanent, countermeasures are eventually defeated, and protections may then be weaponized by malicious actors. Thus, SCVS provides a hitherto nonexistent baseline protection to the customer until the next engineer discovers a flaw in its design.

B. *Server Side*

In addition to the customer-client side, I also mapped the server side of SCVS. The main feature that the server side will provide is the distribution of the customer-client's OTP to authorized in-house and third-party agents. A business must be able to utilize distributed call centers and agents to contact customers for legitimate purposes. However, in order to do so, these agents will need workstation software that will automatically send their caller ID to the authentication server for relay to the customer-client. They will also need access to the OTP via some type of customer profile provided by the sponsor. Thus, SCVS must be a wholistic security software.

A chief complaint of Molly Weis, a specialist in digital media and brand identification, about STIR/SHAKEN is that it may still label legitimate enterprise callers as spam due to its lack of true authentication [16]. As telecommunications expert Devi Koilada explains, STIR/SHAKEN relies on a certification authority and a verification service to validate an outbound call [13]. After a call travels through the telephony network and arrives at the final phone carrier, the verification service flags the call with a gradated attestation score that may fall short of complete identity authentication [6]. For example, a legitimate VoIP call on behalf of a financial lender that bypasses the lender's PBX and transmits directly to a phone carrier may be given a B- or C- attestation score under the STIR/SHAKEN framework instead of an A-attestation, which is the best score. With fewer hardware requirements and via existing applications, SCVS will potentially achieve a better authentication result than STIR/SHAKEN and can be internally managed by the sponsor.

On the server side, the sponsor must first manually establish a customer profile. Once the customer profile exists, an authorized caller-client may request it from the main server. SCVS does not contain a procedure to authorize caller-clients, as this procedure will most likely be different for every organization and depend on its policies and requirements. However, once the server receives a request from an authorized caller-client (i.e., agent, representative, etc.), the server:

1) logs the caller-client's caller ID name and number;
2) sends this caller ID to the intended customer-client;
3) determines if the caller-client is an in-house or third-party agent;
4) sends the caller-client the appropriate version of the customer's profile (which contains the OTP);
5) waits for a response from the caller-client after the caller calls the customer and attempts authentication.

Once the caller is authenticated (or denied) by the customer-client, the caller-client sends the result of the authentication attempt to the server, which logs the data. (The format and content of a third-party customer profile will be determined by the sponsor organization based on its policies and guidelines; however, regardless of its design, the ability of a third-party caller-client to access the OTP for the intended customer is integral to the SCVS design.) Once this process is complete, the main server enters a customer management mode.

While in customer management mode, the main server checks for new customer profiles, software updates, and blacklistings, of which the latter two are sent to the customer-client at a regular interval. If no updates are available, the main server waits for an incident report and the authentication server waits for a new OTP from the customer-client. For the sake of simplicity, I combined the functions of the main server and authentication server into one flowchart (given in the Appendix).

C. *Design Notes*

Ideally, SCVS would be programmed to be compatible with other cybersecurity programs on a customer's smartphone. The advantage of this is that SCVS would not need to address a multitude of unwanted call types and could strictly authenticate live callers that identify as the sponsor or its affiliates. The sponsor would also have to introduce a blacklist repository to its information operation. The sources for such a repository are beyond the scope of the SCVS design.

V. Discussion

SCVS will produce multiple benefits for both customers and organizations. Some possible benefits include:

1) Increased customer confidence in calls from the sponsor organization;
2) Increased customer cooperation with legitimate third-party marketing calls from the sponsor organization's affiliates;
3) Increased customer confidence in their own overall cybersecurity.

Another positive aspect of SCVS is that, unlike a paid app such as Robokiller, a customer with an existing banking or similar mobile app would not need to actively purchase and download SCVS. In a study about cybersecurity awareness, researchers from Penn State University noted that people typically have less cybersecurity knowledge than they assert, lack cybersecurity training, and reject cybersecurity measures if the benefits of the measures do not outweigh their costs [17]. While they did note that training substantially increased peoples' receptivity to cybersecurity measures, a sponsor that utilizes software such as SCVS will be taking effective preemptive measures to improve the security of its customers regardless of their receptivity level.

With these benefits in mind, I acknowledge that advents such as STIR/SHAKEN, Robokiller, and other architectures and applications are integral to a secure telephony ecosystem. STIR/SHAKEN will address many continental scams that use basic techniques to target victims. Meanwhile, third-party applications like Robokiller will address more sophisticated attacks. However, a software like SCVS will truly authenticate a caller. Furthermore, unlike STIR/SHAKEN and Robokiller, SCVS is intended as a distributed, open-source software, which means it will be installed, managed, and customized by the technicians of a sponsor organization. Therefore, it theoretically has a smaller attack surface than STIR/SHAKEN architecture.

Another benefit of SCVS is that it does not require a brand new approach. It will utilize the encryption and transfer protocols that are already used by secure mobile apps. It is otherwise a reversal of the phone-based 2FA models described by [18] and [19]. Instead of using a customer's phone as a token, the caller-client workstation functions as a token whose identifier is transmitted to the customer-client as the caller ID. In order for a social engineer to spoof the caller ID of a legitimate caller, he would have to know the caller's exact name and number strings at the time of the call. The social engineer would have to penetrate the sponsor's network and identify outbound calls in near-real-time in order to accomplish this feat.

Despite its pomp, SCVS does have some setbacks. First, it is only compatible with smartphones that have internet connectivity. This means that simple phones, which are not app-friendly, cannot download the required mobile app of the sponsor, and thus, the customer-client version of SCVS. Second, it may decline legitimate calls from the sponsor if the customer-client does not receive the forthcoming caller-client caller ID from the authentication server in time. For example, if the customer loses internet connectivity before an inbound call, the call may still transmit but the caller-client caller ID may not. Local interference could also impede transmission of vital data that are necessary to authenticate the caller. However, customers may prefer heightened security over convenience in relation to calls from a banking or other financial organization. In total, SCVS is an experimental software design that could spoil a significant portion of vishing attacks with familiar methods.

VI. Conclusion

Vishing is a social engineering attack that exploits human psychology to extract pivotal data from a target [3]. It often takes the form of an automated, live, or hybrid (automated and live) dialog. Unlike email and SMS phishing, vishing is a relatively unresolved threat with few countermeasures. Furthermore, there are few quantitative studies about the effectiveness of vishing. Researchers should calculate the success rate of vishing versus other social engineering techniques in order to determine its true socioeconomic threat. From an intelligence perspective, I hypothesize that a well-constructed vishing attack is more dangerous than a multitude of phishing attacks. Regardless, as cybersecurity professionals, we must devise ingenious methods that diminish the success rate of vishing.

A profound telephony framework named STIR/SHAKEN is being implemented in the U.S. and Canada to reduce spam, phone scams, and spoofing. However, such a system will still have deficiencies and does not replace the responsibility of organizations and their customers to authenticate each other. Therefore, I propose an open-source software design called SCVS, which utilizes an OTP and caller ID to create a 2FA scheme for agents that call customers. If implemented, SCVS may spoil a significant number of vishing attacks.

Furthermore, if SCVS is utilized alongside other frameworks and applications, it could function as one layer of a defense-in-depth strategy. The legality of an application answering a call on behalf of a customer must also be investigated and confirmed. Thus, the next task at hand is the creation of an SCVS prototype, a quantitative study of the prototype, additional legal research, and a detailed study of consumer preferences. These actions will validate or invalidate the tenets of the SCVS algorithm, identify the practicalities of its design, and take SCVS one step closer to realization.

References

[1] O. Salem, A. Hossain, and M. Kamala, “Awareness Program and AI based Tool to Reduce Risk of Phishing Attacks,” 10th IEEE International Conference on Computer and Information Technology, Jul. 2010, pp. 1418-1423, doi: 10.1109/CIT.2010.254.

[2] M. Kolhar, A. Alameen, and M. Gulam, “Performance evaluation of framework of VoIP/SIP server under virtualization environment along with the most common security threats,” Neural Computing and Applications, vol. 30, no. 9, Feb. 2017, pp. 2873-2881, doi: 10.1007/s00521-017-2886-y.

[3] P. Y. Leonov, A. V. Vorobyev, A. A. Ezhova, O. S. Kotelyanets, A. K. Zavalishina, and N. V. Morozov, “The Main Social Engineering Techniques Aimed at Hacking Information Systems,” Ural Symposium on Biomedical Engineering, Radioelectronics and Information Technology (USBEREIT), May 2021, pp. 471-473, doi: 10.1109/USBEREIT51232.2021.9455031.

[4] M. Hijji and G. Alam, “A Multivocal Literature Review on Growing Social Engineering Based Cyber-Attacks/Threats During the COVID-19 Pandemic: Challenges and Prospective Solutions,” in IEEE Access, vol. 9, pp. 7152-7169, Jan. 2021, doi: 10.1109/ACCESS.2020.3048839.

[5] B. E. Boukari, A. Ravi, and M. Msahli, “Machine Learning Detection for SMiShing Frauds,” IEEE 18th Annual Consumer Communications & Networking Conference (CCNC), Jan. 2021, pp. 1-2, doi: 10.1109/CCNC49032.2021.9369640.

[6] J. McEachern and E. Burger, “How to shut down robocallers: The STIR/SHAKEN protocol will stop scammers from exploiting a caller ID loophole,” in IEEE Spectrum, vol. 56, no. 12, pp. 46-52, Dec. 2019, doi: 10.1109/MSPEC.2019.8913833.

[7] S. Biswal, “Real-Time Intelligent Vishing Prediction and Awareness Model (RIVPAM),” International Conference on Cyber Situational Awareness, Data Analytics and Assessment (CyberSA), Jun. 2021, pp. 1-2, doi: 10.1109/CyberSA52016.2021.9478240.

[8] S. -L. Chen, S. Chow, C. Gustave, and D. Vinokurov, “Prototyping a New Identity Authentication Framework for IP Telephony,” Second International Conference on Emerging Security Information, Systems and Technologies, Aug. 2008, pp. 47-52, doi: 10.1109/SECURWARE.2008.31.

[9] F. Maggi, “Are the Con Artists Back? A Preliminary Analysis of Modern Phone Frauds,” 10th IEEE International Conference on Computer and Information Technology, Jul. 2010, pp. 824-831, doi: 10.1109/CIT.2010.156.

[10] R. M. Clark, “Intelligence Analysis: A Target-Centric Approach,” 5th ed., CQ Press, Sage Publications, 2017, pp. 81-82.

[11] U.S. Army, “Intelligence Preparation of the Battlefield,” Headquarters, Department of the Army, Mar. 2019, pp. 5-17-5-20.

[12] “Robokiller Report Reveals 54 Billion Spam Calls Reached Consumers in 2020,” The Public Record; Palm Desert, vol. 48, no. 17, Mar. 2021, pp. 5-6.

[13] D. V. S. R. K. Koilada, “Strategic Spam Call Control and Fraud Management: Transforming Global Communications,” in IEEE Engineering Management Review, vol. 47, no. 3, pp. 65-71, Sep. 2019, doi: 10.1109/EMR.2019.2924635.

[14] Y. Deng and Y. Deng, “Design and Implementation of Distributed Call-Center Based on Soft-Switch,” Journal of Physics: Conference Series 1883 012084, 2021, doi: 10.1088/1742-6596/1883/1/012084.

[15] P. A. Grassi, J. L. Fenton, E. M. Newton, R. A. Perlner, A. R. Regenscheid, W. E. Burr, and J. P. Richer, “NIST Special Publication 800-63B,” National Institute of Standards and Technology, Jun. 2017, p. 13.

[16] M. Weis, “Exploring STIR/SHAKEN,” Collector, vol. 84, no. 11, Jun. 2019, pp. 42-43.

[17] E. M. Raineri and J. Resig, “Evaluating Self-Efficacy Pertaining to Cybersecurity for Small Businesses,” The Journal of Applied Business and Economics, vol. 22, no. 12, 2020, pp. 13-23, doi: 10.33423/jabe.v22i12.3876.

[18] S. Kaman, K. Swetha, S. Akram, and G. Varaprasad, “Remote User Authentication Using a Voice Authentication System,” Information Security Journal: A Global Perspective, vol. 22, no. 3, May 2013, pp. 117–125, doi: 10.1080/19393555.2013.801539.

[19] W. A. Hammood, R. Abdullah, O. A. Hammood, S. M. Asmara, M. A. Al-Sharafi, and A. M. Hasan, “A Review of User Authentication Model for Online Banking System Based on Mobile IMEI Number,” IOP Conference Series: Materials Science and Engineering 769 012061, 2020, doi:10.1088/1757-899X/769/1/012061.