Verifiable Labels for Digital Services: A New Approach to Phishing Detection

Maël Gassmann
School of Engineering and Computer Science
Bern University of Applied Sciences
Biel/Bienne, Switzerland
email: mael.gassmann@bfh.ch

Annett Laube
School of Engineering and Computer Science
Bern University of Applied Sciences
Biel/Bienne, Switzerland
email: annett.laube@bfh.ch

***Abstract*—Users often feel unsafe and unsecure when using digital services. For normal users lacking a technical background, it is difficult to recognize a website's legitimacy. This makes them vulnerable to cyberthreats such as phishing attacks. In order to solve this issue, many organizations use corporate designs or logos to guide users through their websites. However, these files can be easily copied. More technical means are also advertised as solutions, like trusted Transport Layer Security (TLS) certificates with Extended Validation (EV) certificates, but they are too complicated for non-technical users and barely change the outcome. Right now, users lack a way to easily verify that they are using the intended digital service. Verifiable Labels uses cryptographic identifiers—e.g., from the TLS Public Key Infrastructure (PKI)—to bind an entity's label to its identifiable key pair, is a potential solution. Instead of trying to provide automated trust, Verifiable Labels acknowledge the presence of ill-intentioned entities. In order to differentiate them from trustworthy actors, cryptographic tools are used to define metrics, which allow a user client to form easily understandable recommendations and analyze a certain actor's reputation, thus allowing users to naturally develop an opinion and make an educated guess as to whether an entity is trustworthy or not. The end goal would be that most websites asking for some level of trust use Verifiable Labels. This not only has the potential to directly impact Internet users, but also to act as a guiding light for security companies. Since all participating websites would be listed with their reputation metrics, it becomes easier to identify high-risk websites and perform pertinent in-depth analysis in order to take action against phishers faster.**

***Index Terms*—Trust; Anti-Phishing; Digital Label; Reputation.**

I. Introduction

This article presents an extended and refined version of the conference paper titled "Verifiable Labels for Digital Services: A Practical Approach", which was presented during DIGITAL 2023, Advances on Societal Digital Transformation [1].

Nowadays, if website owners want to try and certify an accordance to a label, one sole option is at their disposal: The usage of copyable and thus untrustworthy digital representations, such as pictures or electronic documents, e.g., 'Digital Trust Label' [2]. Without having to make any distinction between true and false claims, it can already be deduced that it has as much value as a self-proclamation and is at least hard and inconvenient, if not impossible, to verify. This is leading naïve Internet users to give their trust to services unworthy of any. Moreover, it is far from affecting only a limited number of people, as since 2020, phishing attacks have become by far the most common type of attacks performed by cybercriminals [3]; 41% of security incidents begin with the initial access gained by a phishing attack [4]; approximately 1.385 million new phishing web pages are set up each month [5]; and overall, phishing is in the top three cybersecurity threat trends [6].

The real problem is there; a verifiable label would truly add value to anybody's Internet experience by directly reducing the impact of phishing. Verifiable labels strive to establish a distributed framework for the development of labels in general and enhance user-friendliness. Additionally, if the concept was successfully adopted, it could act as a guiding light for the existing security ecosystem.

The rest of the paper is structured as follows: Section II analyzes the current state of Internet related technologies; Section III defines the concept of verifiable labels, its underlying infrastructure and protocols; Section IV approaches the concept from a security point of view; Section V explains how the concept was adapted to a working prototype; finally, the work is concluded in Section VI.

II. State of the Art

A. TLS Certificates

Based on Public Key Infrastructure (PKI) to establish chains of trust and using X.509 certificates to bind web-servers to key pairs and domain names, Transport Layer Security (TLS) certificates are nowadays widely used to encrypt communications on the Internet [7]–[9]. These so-called chains of trust are all built upon an entrusted third party—a root of trust—that certifies the trustworthiness of other entities, which in turn are sometimes allowed to do the same. Such entrusted third parties are called Certificate Authorities (CA), as shown in Figure 1.

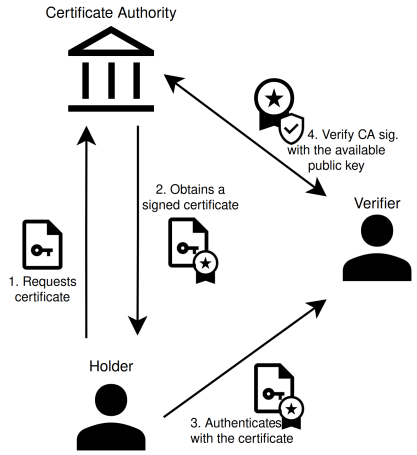

Figure 1. Minimalistic representation of a PKI.

Furthermore, the X.509 certificate itself can contain a variety of different claims. For instance, one way to bind a certificate to a server is to include its specific domain inside. In the case of TLS certificates, there are three major types of X.509 certificates that are used [10].

a) Domain Validated (DV) Certificate: These are the most basic types of certificates. The CA will only verify that the applicant has control over the requested domain name; this is typically done through email validation. More recently, the Automatic Certificate Management Environment (ACME) protocol allowed CAs to issue DV certificates without any intervention from their side [11]. When the ACME protocol is used, the certificate can be obtained free of charge.

b) Organization Validated (OV) Certificate: Not only is the domain ownership verified, but also the legal existence and physical location of the applicant. Automation is, of course, out of the question. Such a certificate can be obtained for a range of 200 to +1000 USD per year [12].

c) Extended Validation (EV) Certificate: EV certificates undergo the most rigorous validation process; this includes all steps taken for OV certificates, including legal status, operational existence, and telephone verification [13]. The price range goes from 400 to +1700 USD per year [12].

OV and EV certificates were advertised as a way to prevent the customers' users from being prone to phishing, as the web browser, recognizing an EV certificate, used to display a green indicator containing the entity's legal name. Thus, users who knew of that distinction would change their behavior according to the level of certification displayed. However, studies showed that user behavior did not alter [14], and polls [15] showed that the padlock's meaning was not understood correctly. Worse even, security researchers were able to prove that some EV certificates could be gotten with colliding organization names, which could be quite misleading as the domain would be hidden by the legal name in some browsers. That is why, in September 2019, most browsers stopped displaying any direct visual distinctions between DV, OV, or EV certificates, which invalidates the main selling point of these products [16].

Moreover, because CAs are private companies, the regulations are not always followed with the same rigor, as not all validation processes can be automated. A PKI infrastructure is always very sensitive to mistakes, and the verification process has proven to not be enough [17]. However, one thing is sure: TLS certificates do a good job of binding a domain name to its corresponding server, which holds the key pair. Especially with the help of the ACME protocol.

B. Decentralized Identifiers and Key Event Receipt Infrastructure

In opposition to the traditional central authoritative system that CAs and DNS represent, Decentralized IDentifiers (DID) and Key Event Receipt Infrastructure (KERI), both open standards in active development, are part of a broader movement that strives towards decentralized identity.

A Key Event Receipt Infrastructure is a secure and decentralized key management system [18]. It provides mechanisms for proving the Root of Trust for self-certifying identifiers and their associated key states. While TLS certificates bind themselves to a domain name by using trusted third parties, the cryptographic identifiers KERI generates are bound in the strongest manner to their key-pair by using one-way cryptographic functions. This means that the authenticity and integrity of identifiers are verifiable through cryptographic proof. It has great potential and could very well replace the current administrative centralized infrastructure of TLS certificates.

1. **DID** (for self-description)
2. **Set of public keys** (for verification)
3. **Set of auth methods** (for authentication)
4. **Set of service endpoints** (for interaction)
5. **Timestamp** (for audit history)
6. **Signature** (for integrity)

Figure 2. The standard elements of a DID document [20].

A DID resolves to a DID document—typically hosted on a decentralized network or infrastructure, e.g., a blockchain or a distributed ledger—which contains a set of public keys, authentication methods, service endpoints, a time-stamp to keep an audit history, and a signature for its integrity [19].

KERI has already standardized a way to link a KERI identifier to a DID. It leverages KERI's strong cryptographic controls to create decentralized identifiers.

A Verifiable Credential (VC) is a claim created from the key pair of a DID (the issuer) and is issued to a holder's wallet by using a holder proof. This holder proof varies greatly between implementations, and efforts are being made to standardize it. Self-Sovereign Identity (SSI) solutions strive to provide a way to assert, present, and verify claims in a decentralized manner [21].

A verifiable label solution would be quite straight-forward to implement with such technologies. The big issue with them is that not everything is yet standardized. For instance, once a DNS name is linked to a DID, the browsers will not recognize it as trustworthy. The truth is, it is not yet used in practice. For a verifiable label to be used, it needs to work with the current Internet cryptographic tools. It however highlights the need for a solution that adapts to any type of cryptographic identifier.

C. Users awareness

The first thing to identify before designing a solution is what level of awareness users have whilst navigating the Internet and the way cryptographic proofs are naturally understood due to current visual designs. Consider these three types of users.

1) The unaware user
2) The user with no technical background
3) The technically aware user

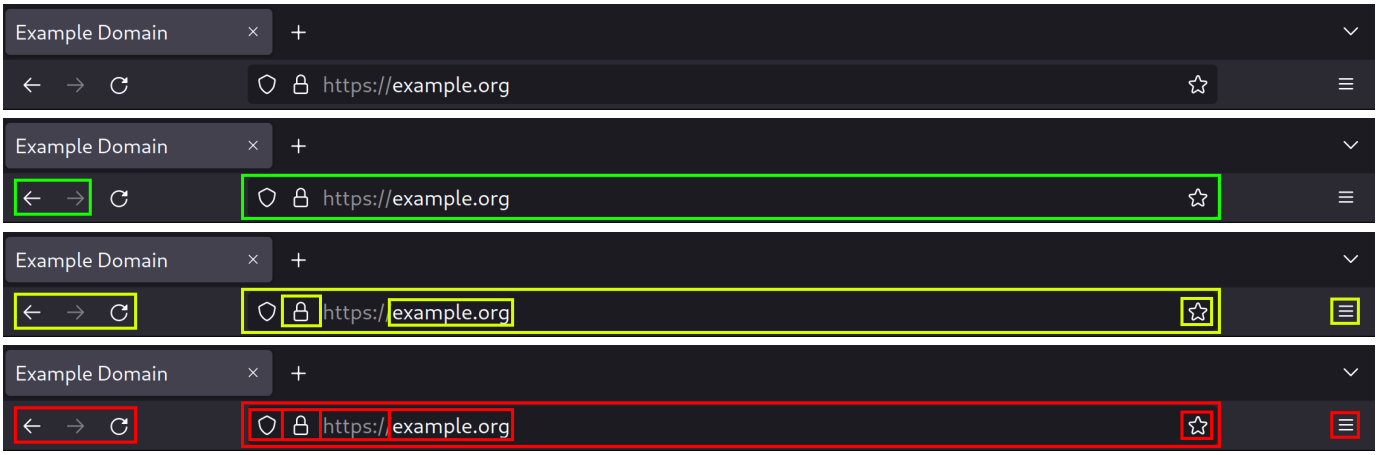

Figure 3. Firefox Browser UI, with each awareness levels depicted from lowest to highest.

These categories were defined after a small survey was conducted. The test consisted in asking users what was their understanding of the purpose of every icon, button or text present on the browser user interface (UI) of Figure 3.

Quick description of the UI:
The back and front arrow navigate through the history, the circle arrow can refresh the page, the shield allows to visualise cookies and trackers present on a website as well as setting protection rules against them, the lock let the user know whether the connection is private or not, `https://` is the protocol, *example.org* is the hostname, the star allows one to add the current URL to their favorites, and the three stacked lines icon is the menu of the browser.

1) Completely unaware users: These users are either very new to the concept of web browsing or rarely use it, they know how to type-in terms, not yet URLs, in the search bar and end up on a page where they can pick a website to visit. They know how to navigate their current session's history with the arrows and, most of the time, they somehow know how to input their credit card's information in any websites. Typically, such users will learn enough to reach the second level of awareness by practicing web browsing. A new solution should ensure that users can intuitively gain awareness as well.

2) Users with no technical background: Most users that often navigate the Internet get to such an understanding of the browser naturally. Through curiosity, they learn how the browser's functionalities work and understand the menus. They know how to add a website to their favorites and have a very basic understanding of a URL. The lock is understood as a 'safe' or 'unsafe' indicator, but there is no comprehension as to why one website would be safer than another. This is a problem, as this indicator does not actually differentiate a spoofer from an authentic website. All it does is indicating whether the connection to the web server is private or public.

3) Technically aware users: Here, users must have wrapped their head around the technical background of the Internet. They know about PKI and SSL/TLS certificates, the HTTPS protocol and how it differs from HTTP and the existence of cookies and trackers. The only way to get there is to study, which is why so many people stagnate in the previous level. But even then, sophisticated spoofing attacks could still succeed in a moment of inattention.

The problem is that SSL/TLS or even HTTPS understanding always remains unknown to the average users, and that the only information they could potentially grasp is a boolean indicator, which is misunderstood and also often misexplained, as 'safe' or 'unsafe'. The plain truth is, no one can ever be a 100% sure that they are visiting the correct website, even with extensive technical verification of the certificate.

If Verifiable Labels were to offer a UI showing green check marks or red warnings beside labels of a website, there would be no major improvements. This is because, although a failed cryptographic proof clearly signals a problem, the successful verification of such a proof does not warranty that no issues are present.

III. Concept

A. Different Perspective

The root of the verifiable label concept lies in a shift of perspective on what trust is and how it can be made identifiable to an end-user. As TLS EV certificates proved, a seemingly good concept will still need to be understood by anyone who uses the Internet in order to have any impact, especially by those who do not have any technical background. First, one must understand how trust is perceived as a concept alone; for this, a philosophical definition of trust is adequate.

'Trust is important, but it is also dangerous. It is important because it allows us to depend on others—for love, for advice, for help with our plumbing, or what have you—especially when we know that no outside force compels them to give us these things. But trust ***also involves the risk*** *that people we trust will not pull through for us, for if there were some guarantee they would pull through, then we would have no need to trust them.* ***Trust is therefore dangerous****. What we risk while trusting is the loss of valuable things that we entrust to others, . . . '* [22]

That is, when a person *decides* to place their trust in someone else, they know about the risks—risks that can be clearly identified as they are based on facts.

Instead of distributing a trust people have to blindly believe in, verifiable labels proposes the idea of providing simple facts about Internet entities so that anyone with no technical background can, in a reasonable time, learn how to navigate the Internet with the ability to discern entities that deserve their trust. To take a risk is, after all, an individual decision, and users must be able to make the assessment themselves and not have to entrust it to a third-party organization that does not have their interests at heart.

To do this, cryptography is paramount, as it is the sole option available to make any virtual information a tangible fact. The system must be implemented on top of the currently widely used Internet cryptographic technologies (e.g., TLS certificates) in order to have any chance of success, while also striving to be flexible and pushing towards more decentralized technologies (e.g., blockchains) because they provide a better and stronger infrastructure.

B. Definitions

1) VERIFIABLE LABEL
   A verifiable label is a data structure that is bound to two domain names; the holder's and the issuer's. This is done by signing the label with both domain name linked cryptographic identifiers (e.g., TLS certificate). This ensures that the label is bound to the server that holds the cryptographic identifiers. Therefore, ensuring the authenticity of both the holder and issuer, and that it cannot be copied. It also warrants the integrity of the content. It contains the following fields:
   a) Domain of holder
   b) Label name
   c) Domain of issuer
   d) Signature of the holder's cryptographic identifier
   e) Signature of the issuer's cryptographic identifier
2) ISSUER RECORDS
   An issuer record is another data structure similar to a verifiable label, it defines the identity of an issuer and its label while also fulfilling a variety of other roles. Since its validity is warranted by an external entity's—i.e., the verifiable enforcer's—signature, each time a holder is added or removed from the record, a new one has to be requested if an immediate update is deemed necessary. On top of that, the system will ensure that it has to be renewed once its time-stamp is too old. All previous records are to be kept online by the issuer, for each of them contains valuable behavioral data when combined with the rest. To ensure that all are present, each of them is bound by the previous record's signature of the verifiable enforcer. It contains the following fields:
   a) Label name
   b) Domain of issuer
   c) List of holders [(Domain of holder; Status; Issuer signature)]
   d) Previous verifiable enforcer's signature
   e) Timestamp
   f) Signature of the issuer's cryptographic identifier
   g) Signature of the verifiable enforcer's cryptographic identifier
3) VERIFYING USERS
   Simple users that visit a website. If a valid label is detected, the user will be able to see it, list facts that concern it, and develop an idea of this label's reputation.
4) LABEL-WORTHY WEB-ENTITY
   Such an entity can request a label from its corresponding issuer. If an issuance occurs, it officially becomes a holder and can display its digital label on their website, which is visible and verifiable by anyone. If copied to an alternative server and a different TLS certificate is used, validation will always fail.
5) ISSUER
   The entity that can verify and decide, of its own accord, who is worthy of being labeled. It will keep a record of who has been issued its label and can confirm it with the list of holders contained in its latest record. This gives the issuer the power to revoke a label by just removing the entry from the list.
6) VERIFIABLE ENFORCER
   The backbone of the concept is here; this entity can be understood both as a centralized, transparent authority, or as a set of rules enforced by a consensus. As it is the only piece of infrastructure that would require financing if implemented centrally, a decentralized implementation would be preferable, i.e., as a smart-contract [23]. It will follow specific automated guidelines, all of which are reproducible and thus verifiable. It will provide time-stamps from a trusted source—either a time-stamp authority or a blockchain time-stamping service [24]—and distribute them with a signature to pre-existing issuers on demand. As all issuers need an unexpired latest record, they will have to issue renewal requests in any case. Each request, building upon one another, starts to create a reputation. The purposes of the guidelines are to make sure that every issuer plays by the same rules by recording each of their behaviors in their own records, as well as to aim at enforcing duplicate label prevention when a new label requests its first signed record. The only data necessary for it to operate is all of the issuer's domains, which is easy enough for it to keep track of. This list will be publicly available but will have no other purpose than allowing exploration.

C. Protocol

a) Issuance of a label: Figure 4 depicts it. A website must create a verifiable label and sign it with its TLS certificate. This ensures that the draft label is bound to the domain name and also comes from the stated owner. The incomplete digital label can be sent to the issuer; no channel is specified. If the issuer decides to accept the request, it will sign it with its own TLS certificate, add the new signature to the now complete verifiable label, and send it back. Finally, the issuer save a copy of the signature and requester's domain in the

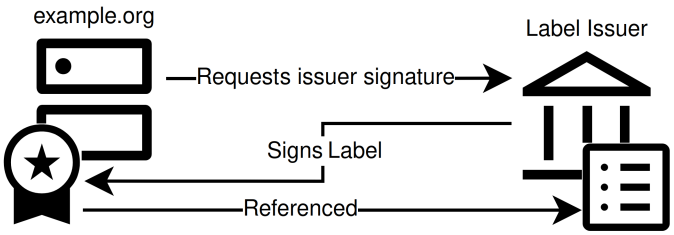

Figure 4. Verifiable Label Issuance

list of its new draft record. In order to make a valid issuer record out of this draft, the issuer has to request a new time-stamp and signature from the verifiable enforcer, as explained in Figure 5.

b) Issuer Record Validity: As stated before, an issuer's trustworthiness is defined by its own reputation. This reputation is built with time and the help of a trusted time-stamp source. The verifiable enforcer's role is to issue new signatures—necessary for the issuer's label to be considered cryptographically valid—and time-stamps to all pre-existing requesting issuers that are on the brink of expiration. As it does so, each request will always be examined in a replicable manner that warrants the verifiable enforcer's verifiability. First, it will verify the cryptographic signature of the new incomplete record and its previous records, which must be kept online. Once authenticity has been verified, each holder's verifiable label is retrieved from the list contained in the draft record, and each of their status field will be set to a boolean value that reflects whether or not said website is online with its verifiable label. Finally, the draft is signed and then sent back. However, if the issuer is new, i.e., does not possess a previous signature, the verifiable enforcer will take a look at the requested label name, domain name, and all fields that might be prone to confusing a human being. If it is considered

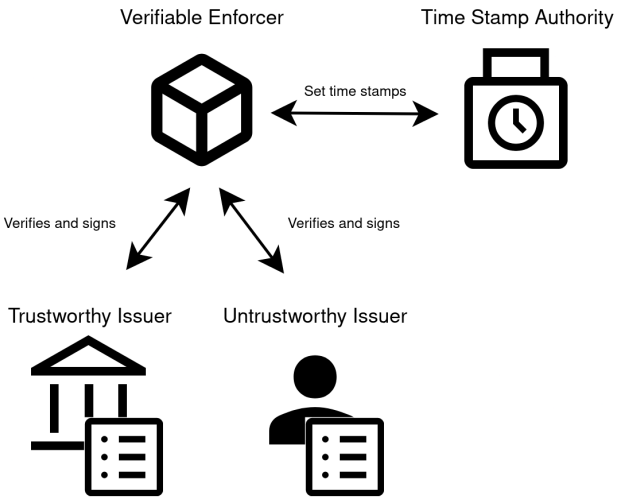

Figure 5. Verifiable Enforcer

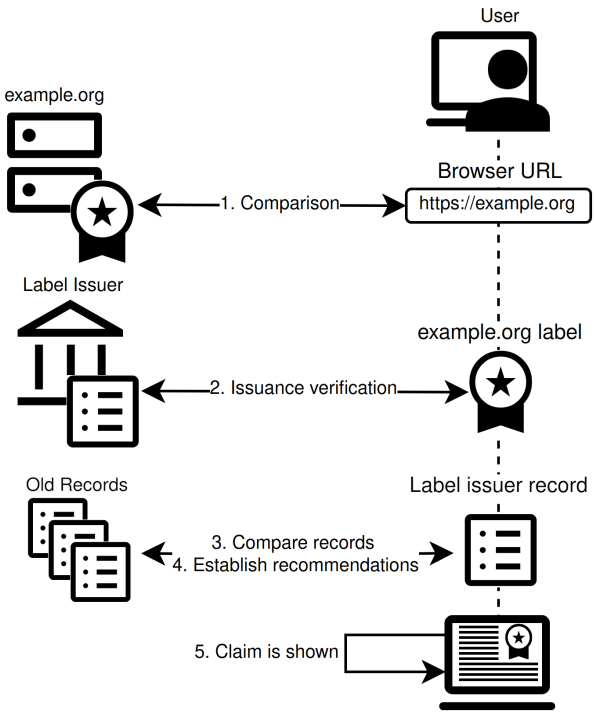

Figure 6. Verifiable Label Validation

not to be confusing as well as not a duplicate of any existing labels, the web entity will receive its first time-stamp and signature, making it an issuer.

c) Validation and interpretation client: As a user with the verifiable label validation and interpretation client installed navigates the Internet, the client will try to detect if a digital label is present on the currently visited website. If this proves to be the case, the validation process will begin, as shown in Figure 6. The first step consists of verifying the label's link with the domain and TLS certificate, that is, making sure the signature is correct and that the domain corresponds to the browser URL. On success, the next step will be to cross-check the label with the listed issuer record. The record and digital label are to be compared, signatures are to be verified for the same reason as before, and the domain of the holder must be found in the list. At last, if everything succeeds again, the client will go through the label's previous' records to derive a reputation.

D. Reputation System

Accurate protection is possible if we assume that a majority of web entities have adopted this digital label system. Only then would websites with bad or no label start to stand out, especially because they require trust, e.g., when they ask for credit card information or propose services.

IV. Security Considerations

Phishing attacks have been studied for years now. Some recent scientific papers as well as older ones [25]–[27] have described how a typical attack is performed, as shown in Figure 7. Most of the security industry focuses on stopping

the attack at step 2. The techniques used can be on the client side, like native browser blacklists and malware detection tools that analyze URL and page content or search engine rankings;

Whereas on the server side, the user would have to check for an EV certificate, i.e., if he knows about it and if the website still pays for one. Furthermore, security companies use a network of systems to track and update blacklists [27].

Thus, the anti-phishing ecosystem can be described as reactive, and a requirement for a reactive system to be effective is at least one low latency feedback source. However, the sources used by the current ecosystem seem to always be a few steps too far behind to provide effective measures at the right time.

Under the assumption that Verifiable Labels was put in place successfully, most business websites (holders) would have at least been issued one label to assert their online identity. Since every issuer of label is being accounted for, as is each of its holders' websites, it ultimately creates a list of links ready to be scanned. As a result, the set of websites to be analyzed is reduced to this list, which contains all websites that require the trust of their users, for that is where phishers want to be.

Therefore, when phishers eventually change their strategy to try and fulfill this new requirement, they will face a bigger risk of being discovered, as security companies will be able to actively search for them in a defined subset.

In an effort to already foresee and impede an attacker's effort to try and infiltrate the Verifiable Label system, further measures were imagined and are described in the sections below.

A. Issuer Minimal Requirements

Since phishers leverage the existing trust a user gives to an existing company by impersonating it, the following measures will try to hinder the creation of duplicate labeling companies.

Before a labeling organization can start issuing labels, as stated in the protocol subsection III-C0b, its new label will undergo a replicable analysis conducted by the verifiable enforcer:

1) With the given signature and the corresponding TLS certificate, the domain of the issuer is authenticated.
2) The domain's first prefix, as well as the label name, have to be unique. Moreover, the remaining suffix of the domain has to be part of a whitelist.
3) It is yet undecided whether a supervised machine learning clustering algorithm or a more traditional algorithm should strive to eliminate new labels with names and domains that resemble other well-known issuers or are considered confusing.

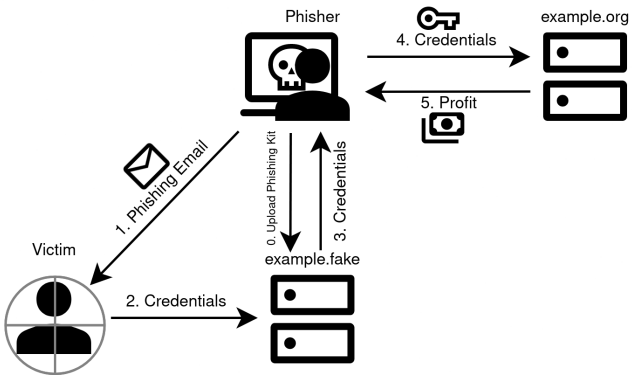

Figure 7. Typical Phishing Attack

As stated above, this process is reproducible and thus verifiable as well. It is to be applied uniformly in an automated manner that allows others to come to the same conclusion by reproducing the analysis. This should already impact the way impersonations are engineered when trying to create a fake label, thus reducing the potential risk.

B. Monitoring & Pattern recognition

To assume that a new issuer is trustworthy once it passes the initial registration verifications, as older systems showed, is wrong. Phishers will adapt. And so should the infrastructure supposed to safeguard the Internet trust. Verifiable Labels proposes a new approach to phishing detection: to provide recent data that characterizes the behavior of issuers, including their holders, and leverage it to evaluate the risk of trusting each entity.

What is evaluated is not trust, because, as demonstrated before, the end-user should always remain in control and choose for himself with the help of facts, i.e., the metrics in our case. However, a client-side risk-based evaluation could help further. This way, a user not only gets additional knowledge about the behavior of the label of each website, but also an illustrated meaning. Figure 8 clarifies each actor's domain of action and/or responsibilities in the Verifiable Labels ecosystem.

Please note that the weights applied to the following metrics are only meant to serve a temporary purpose while historical data is not available. A machine learning algorithm could be much more effective at identifying such weights, as an ever-growing historical dataset would allow it to adapt to the new trends and immediately fight back.

a) Age: The age of a specific label is quite different from a domain's age; research shows that 53.3% of phishing attacks used domains older than a year, it is generally assumed that this is because phishers were able to use compromised infrastructure in their campaigns [27] . Since a label is strongly bound to the cryptographic key of the server, the attacker also has to compromise the cryptographic identifier to take over a label issuer, making it harder, or close to impossible if the cryptographic identifier is well managed with KERI [18]. Hence, a new label would have a bigger risk of being fake than an old one.

Consequently, age will impact the risk the most when, e.g., a label is younger than a year. In this manner, all new labels would stand out.

b) Number of Holders: The second metric is the number of holders an issuer has. It can be counted by any client from the list of holders contained in the latest issuer's record. A low amount does not make sense and is easy for a phisher

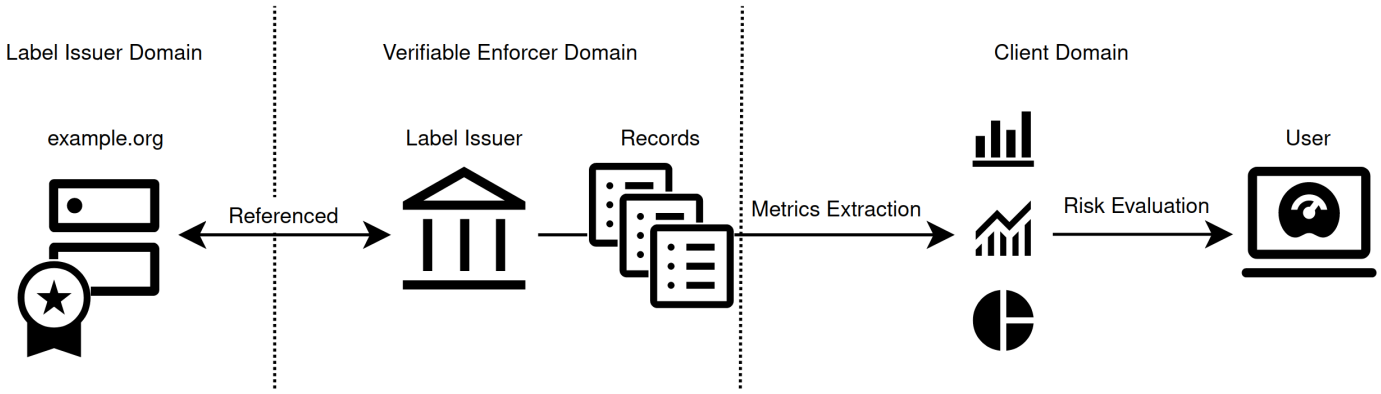

Figure 8. Delimitation of Each Actor's Domain

to achieve, while on the other hand, having many holders is natural and is quite hard for a phisher to obtain and maintain. It should pressure the phishers to group, which increases their infrastructure costs but also potentially attracts unwanted attention.

c) Number of Dead Holders: Holders that could not be reached by the verifiable enforcer during issuance were listed in the issuer record. It will penalize the issuer's risk evaluation heavily if any is present. In this manner, issuers are incentivized to make sure that they not only have a good number of holders, but also that the holders are qualitative. Therefore, making sure that the list of holders is up-to-date and that phishers are on their own.

d) Attrition Rate Analysis: Inspired by the concept of churn rate analysis, but with identifiers and a different purpose.

A client will identify all holders that were removed by going back through the old records of a label. All of these deleted holders will have their lifetime calculated; a long average lifetime is expected; otherwise, it would become clear that a portion of holders are being constantly replaced in a devious attempt to evade detection. The number of removed holders is also expected to be very low in comparison to the total number of holders.

Needless to say, a client will give much weight to both values in the risk evaluation.

e) Surge Detection: A surge is when a large number of holders are either added or removed. This mechanism is very useful to forbid and punish sudden increases in a number of holders, or a sudden evasion technique that consists in removing all phishers. The impact on the evaluation is to be proportional to the spike's intensity, making it temporarily stand out. However, with time, the burden will lessen.

f) User Feedback: Should all of the measures still fail to identify a malevolent entity, clients will be equipped with a reporting functionality that send all parameters that are used in the context of its visit—i.e., the user-agent, the holder's website, and a reason—in order to avoid cloaking mechanisms on the targeted website. Each time a report is sent, the user has to prove that they are human by filling out a captcha. Such feedback reports will be stored and made publicly available by the verifiable enforcer, and will not have any weight in the evaluation of risk. However, it should provide insights as to what subset of websites might be worth scanning with the reported parameters.

C. Integration with Existing Security Infrastructures

Hypothetically, if a web crawler was ordered to visit and evaluate the set of websites corresponding to all web-entities that use Verifiable Labels, it could filter them into an even smaller subset corresponding to all issuers, including their holders, with a medium to bad reputation.

This could greatly benefit the existing security infrastructure by pointing to the websites that require urgent measures, thereby rapidly winning more crucial battles against phishers. Which, in turn, will effectively lower the latency of feedback central to the reactive security ecosystem.

V. Implementation

A prototype has been implemented following a minimal working system approach. Furthermore, since different underlying technologies exist, extensibility is a top priority.

A. Verifiable Enforcer

Starting from the very root of the system, this implementation of verifiable enforcer uses a library that implements an RFC 3161 [28] client interface to interact with an external TSA to provide the time-stamps. A TLS certificate was used to sign issuer records. This server software consists of a simple HTTP API with two paths: the POST method on '/sign' and the GET method on '/get_records'. Meaning, it also acts as a publicly readable storage. All of this has been implemented in the most minimalistic way, with abstract interfaces of 'Storage', 'API', and 'Signer'. That is where flexibility is; the logical part of what makes the verifiable enforcer is detached from all other components that could find better long-term alternatives (e.g., more resource-efficient or different time-stamp sources such as a blockchain).

B. Issuer Client

The simple command-line client has persistent storage and saves all valid given arguments. If provided with a valid request, it will add a domain to its record and generate a valid digital label (.vlcert), which can be sent back to the holder through any channel. It can issue a signing request to the verifiable enforcer on demand. And, if successful, it will save the verifiable label issuer record (.vlicert). All vlicert have to be exposed on the label domain's web server root.

C. Holder Client

This simple command-line client with no persistent storage can only be used to generate a verifiable label without the issuer signature. It has to be manually sent to the issuer. Once a valid digital label (.vlcert) is in the holder's possession, it has to be exposed on its domain's web-server root as 'cert.vlcert'. This prototype thus only allows for one label per holder.

D. Browser Extension Analyzer

A browser extension was a mandatory component of the client, as the active URL has to be accessed to perform the first cryptographic tests. However, the specific environment did not provide any way to download a TLS certificate for a specified domain, which blocked further development. More research showed that by using the native messaging interface, the browser extension can communicate data to an underlying program. Using this method, a cryptographic verifier was developed. It sends back the necessary data to perform a reputation analysis and is then displayed in a panel.

VI. Conclusion

We proposed a system to implement a reliable reputation-based digital label system for websites to replace the now fragile and ineffective automated trust provided by the current security infrastructure.

Each website can request labels from self-declared label issuers. Each issuer, whether trustworthy or not, has its label activities monitored and stored in its own issuer records. This is ensured through the verifiable enforcer, whose timestamps and signatures are necessary for an issuer to be recognized as such. All records of each issuer are always kept online, allowing client software to extract pertinent metrics from them and evaluate the overall reputation of a label through a risk analysis. This risk analysis should allow humans to develop a sense of trustworthiness without having to understand Internet-related technologies.

It was argued that verifiable labels would mostly be used by websites that require the trust of their user base (e.g., webshops require credit card information). And since phishers leverage the pre-existing trust between a user and a web entity, it would only be a matter of time before they tried to infiltrate the reputation system. The metrics provided in every issuer record are thus not only useful to evaluate the risk for a user, but they would also be very pertinent to identifying what subset of websites should be on the watch list of the current security industry.

Furthermore, a minimalistic prototype was implemented. It is flexible, and, even if simplistic, it already implements all the necessary cryptographic tools.

Future work could investigate the following directions:

- Extend the prototype to support fully decentralized infrastructures.
- Conduct a field study of a live setup and user experience.
- Prove the effectiveness of the reputation metrics.
- Provide a comprehensive User Interface (UI) for computers and phones.

References

[1] M. Gassmann and A. Laube, "Verifiable labels for digital services: A practical approach," in *DIGITAL 2023, Advances on Societal Digital Transformation*. Wilmington, DE, USA: International Academy, Research, and Industry Association (IARIA), 2023, pp. 8–12, [retrieved: 03, 2024]. [Online]. Available: https://www.thinkmind.org/index.php?view=article&articleid=digital_2023_1_20_20012

[2] Swiss Digital Initiative. (2024) The digital trust label. [retrieved: 05, 2024]. [Online]. Available: https://digitaltrust-label.swiss/

[3] FBI Internet Crime Complaint Center, "Internet crime report 2023," Federal Bureau of Investigation, Tech. Rep., 2023, [retrieved: 05, 2024]. [Online]. Available: https://www.ic3.gov/Media/PDF/AnnualReport/2023_IC3Report.pdf

[4] IBM Security, "X-Force Threat Intelligence Index 2023," International Business Machines Corporation, Tech. Rep., 2023, [retrieved: 08, 2023]. [Online]. Available: https://www.ibm.com/reports/threat-intelligence

[5] M. Swindells. (2023) How many phishing emails are sent daily in 2023? 11+ statistics. [retrieved: 08, 2023]. [Online]. Available: https://earthweb.com/how-many-phishing-emails-are-sent-daily/

[6] Cisco Umbrella, "Cybersecurity threat trends: phishing, crypto top the list," Cisco, Tech. Rep., 2021, [retrieved: 08, 2023]. [Online]. Available: https://umbrella.cisco.com/info/2021-cyber-security-threat-trends-phishing-crypto-top-the-list

[7] E. Rescorla and T. Dierks, "The Transport Layer Security (TLS) Protocol Version 1.2," RFC 5246, Aug. 2008, [retrieved: 08, 2023]. [Online]. Available: https://www.rfc-editor.org/info/rfc5246

[8] E. Rescorla, "The Transport Layer Security (TLS) Protocol Version 1.3," RFC 8446, Aug. 2018, [retrieved: 08, 2023]. [Online]. Available: https://www.rfc-editor.org/info/rfc8446

[9] R. Housley, T. Polk, D. W. S. Ford, and D. Solo, "Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile," RFC 3280, May 2002, [retrieved: 08, 2023]. [Online]. Available: https://www.rfc-editor.org/info/rfc3280

[10] CA/Browser Forum, "Baseline requirements for the issuance and management of publicly-trusted tls server certificates, version 2.0.2," January 2024, [retrieved: 01, 2024]. [Online]. Available: https://cabforum.org/uploads/CA-Browser-Forum-TLS-BRs-v2.0.2.pdf

[11] R. Barnes, J. Hoffman-Andrews, D. McCarney, and J. Kasten, "Automatic Certificate Management Environment (ACME)," RFC 8555, Mar. 2019, [retrieved: 08, 2023]. [Online]. Available: https://www.rfc-editor.org/info/rfc8555

[12] DigiCert. (2023) Compare tls/ssl certificates. [retrieved: 08, 2023]. [Online]. Available: https://www.digicert.com/tls-ssl/compare-certificates

[13] CA/Browser Forum. (2022) Ev ssl certificate guidelines. [retrieved: 08, 2023]. [Online]. Available: https://cabforum.org/extended-validation/

[14] C. Thompson, M. Shelton, E. Stark, M. Walker, E. Schechter, and A. P. Felt, "The web's identity crisis: understanding the effectiveness of website identity indicators," in *28th USENIX Security Symposium (USENIX Security 19)*, 2019, pp. 1715–1732.

[15] PhishLabs. (2017) A quarter of phishing attacks are now hosted on https domains: Why? [retrieved: 03, 2024]. [Online]. Available: https://www.phishlabs.com/blog/quarter-phishing-attacks-hosted-https-domains

[16] Chromium Docs. (2019) Ev ui moving to page info. [retrieved: 08, 2023]. [Online]. Available: https://chromium.googlesource.com/chromium/src/+/HEAD/docs/security/ev-to-page-info.md

[17] C. Cimpanu. (2017) Extended validation (ev) certificates abused to create insanely believable phishing sites. [retrieved: 08, 2023]. [Online]. Available: https://www.bleepingcomputer.com/news/security/extended-validation-ev-certificates-abused-to-create-insanely-believable-phishing-sites

[18] S. M. Smith, "Key event receipt infrastructure (KERI)," *CoRR*, vol. abs/1907.02143, 2019, [retrieved: 11, 2023]. [Online]. Available: http://arxiv.org/abs/1907.02143

[19] D. Reed, M. Sabadello, A. Guy, and M. Sporny, "Decentralized identifiers (DIDs) v1.0," W3C, W3C Recommendation, Jul. 2022, https://www.w3.org/TR/2022/REC-did-core-20220719/.

[20] D. Reed. (2018) Decentralized identifiers (dids): The fundamental building block of self-sovereign identity (ssi). [retrieved: 11, 2023]. [Online]. Available: https://www.slideshare.net/SSIMeetup/decentralized-identifiers-dids-the-fundamental-building-block-of-selfsovereign-identity-ssi

[21] D. Longley, D. Burnett, K. D. Hartog, M. Sporny, B. Zundel, and G. Noble, "Verifiable credentials data model v1.1," W3C, W3C Recommendation, Mar. 2022, https://www.w3.org/TR/2022/REC-vc-data-model-20220303/.

[22] C. McLeod, "Trust," in *The Stanford Encyclopedia of Philosophy*, Fall 2021 ed., E. N. Zalta and U. Nodelman, Eds. Metaphysics Research Lab, Stanford University, 2021, [retrieved: 08, 2023].

[23] V. Buterin, "A next-generation smart contract & decentralized application platform," https://ethereum.org/en/whitepaper/, Ethereum Foundation, Tech. Rep., 2014, [retrieved: 05, 2024].

[24] L. Meng and L. Chen, "A blockchain-based long-term time-stamping scheme," in *Computer Security – ESORICS 2022*, V. Atluri, R. Di Pietro, C. D. Jensen, and W. Meng, Eds. Cham: Springer International Publishing, 2022, pp. 3–24, [retrieved: 01, 2024].

[25] D. Watson, T. Holz, and S. Mueller, "Know your enemy: Phishing - background on phishing attacks," https://honeynet.onofri.org/papers/phishing/details/phishing-background.html, The Honeynet Project & Research Alliance, Tech. Rep., 2005, [retrieved: 11, 2023].

[26] P. J. Nero, B. Wardman, H. Copes, and G. Warner, "Phishing: Crime that pays," in *2011 eCrime Researchers Summit*, 2011, pp. 1–10, [retrieved: 11, 2023].

[27] A. Oest, Y. Safei, A. Doupé, G.-J. Ahn, B. Wardman, and G. Warner, "Inside a phisher's mind: Understanding the anti-phishing ecosystem through phishing kit analysis," in *2018 APWG Symposium on Electronic Crime Research (eCrime)*, 2018, pp. 1–12, [retrieved: 11, 2023].

[28] R. Zuccherato, P. Cain, D. C. Adams, and D. Pinkas, "Internet X.509 Public Key Infrastructure Time-Stamp Protocol (TSP)," RFC 3161, Aug. 2001, [retrieved: 08, 2023]. [Online]. Available: https://www.rfc-editor.org/info/rfc3161