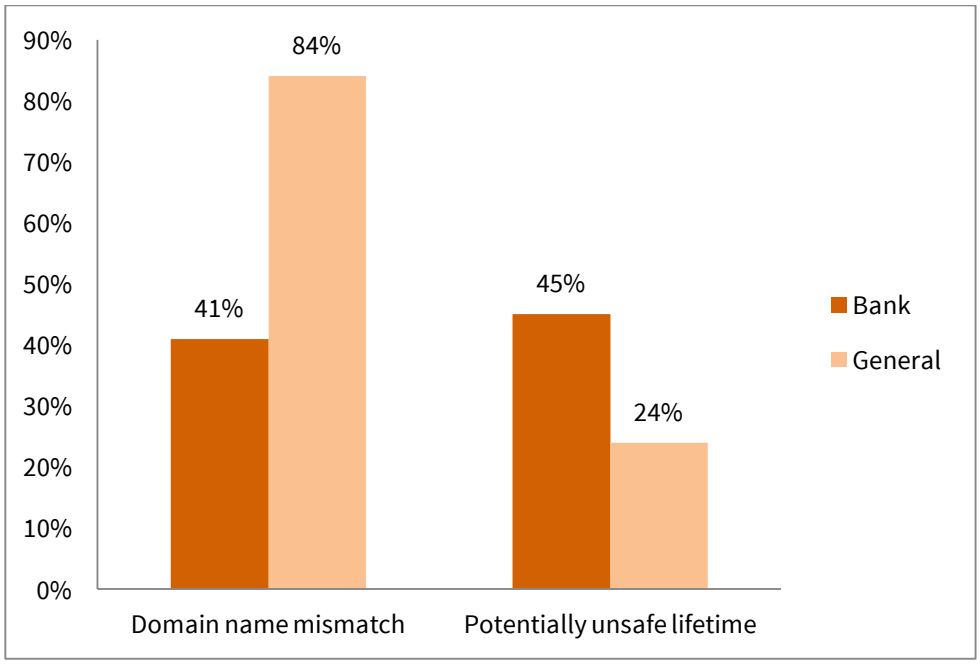
Dong Z, Kane K, Chen S, Camp L. The New Wildcats: High-Risk Banking From Worst-Case Certificate Practices Online. *Technology Science*. 2016041401. April 14, 2016. http://techscience.org/a/2016041401

The New Wildcats: High-Risk Banking From Worst-Case Certificate Practices Online

Zheng Dong, Kevin Kane, Siyu Chen, and L. Jean Camp

Highlights

- We implemented a large-scale examination of certificates used to authenticate and secure communications online by comparing the practices of 27,000 Federal Deposit Insurance Corporation (FDIC)-insured banks against the top 1 million most popular general websites
- We found only 23 percent of banks had official ranked domains, and 50 percent of those domains lacked certificates
- In general, more bank website certificates (45 percent) had very long validity times (a risky practice) than did general websites (24 percent)
- To address these vulnerabilities, we recommend that the public key infrastructure (PKI) follow technical best practices: use strong cryptography, provide clear revocation information, discourage wildcard certificates, and limit extended key usage (EKU) per certificate
- We also propose developing an official third-party certificate notarization authority that applies to banks and other important financial institutions to indicate to the user when a domain is officially operated by a federally insured depository institution



Banks have fewer domain-name mismatches (half as many as popular general interest sites), but are much more risk seeking when it comes to certificate lifetime.

Abstract

Phishing attacks against bank websites occur when imposters masquerade as official bank websites. The idea is to convince the victim that the imposter is actually from a known, familiar institution, in order to fool him or her into providing passwords and other personal information. A solution requires the ability to distinguish legitimate banking institutions from other sites. The current core security designed to thwart these attacks relies on certificates that cryptographically certify the connection between a website and a user. However, such certificates are often used incorrectly, and even when implemented properly, they have weaknesses that can be exploited for attack against online banking sites. We implemented a large-scale examination of certificates, downloading some 4 million certificates over two years using machines on three continents as a baseline for comparison against a second set of bank certificates from the Federal Deposit Insurance Corporation (FDIC)'s list of 27,000 federally insured depository institutions.

Results summary: We found that the use of certificates and the rest of the core authentication and transmission security infrastructure is weak for online banking, with a greater share of bank sites having at least one of the PKI vulnerabilities analyzed when compared against a group of popular general interest websites. As shown above, long-lived certificates (which exacerbate the risk of breach) are used 45 percent of the time by banks, but only 24 percent of the time by general websites. For FDIC-insured banks, only 50 percent have a certificate that reflects the bank or domain name, and only 23 percent of banks had official domains at all. Even when the banks have both domains and certificates, 41 percent of those do not

match. Since certificates are intended to verify the identity of an online entity, the lack of widespread available verification is problematic.

In response to these weaknesses, we present a set of technical best practices, and show how rarely these standards are met in practice. The failures we identify mean that banks are not correctly identified to their customers, and traffic between banks and customers is often insecure. We close with a specific regulatory and technology policy solution of creating an authenticated official banking website indicator that will reduce the vulnerability of banking websites to phishing and related attacks and, which would require a structural change neither in the certificates themselves nor in the larger public key infrastructure.

Introduction

Banking institutions are common targets of phishing attacks [1]. These attacks rely on tricking the victim into thinking they are connecting to a bank, when in fact they are connecting to an attacker.

A phishing attack usually begins with an email alert supposedly from a person's bank. The victim is instructed to respond to the alert by following a provided link and entering the requested information. The victim, trying to do the right thing to keep his or her information secure, may go to the site, which is controlled by the attacker, and enter his or her bank account information and password.

To prevent such deception, the connection between a website and a customer needs to be secured and authenticated. No third party should be able to insert themselves into the middle of that connection. No third party should be able to read the information transmitted between the bank and customer. No third party should be able to pass off their own site as the bank's official site.

A security infrastructure that confirms the identification of a website does exist: The existing public key infrastructure (PKI) authenticates a website's identification to users. In this paper, we examined the public key certificates of websites that correspond to the depository institutions or banks insured by the Federal Depository Insurance Corporation (FDIC [2]. We examined whether, in practice, PKI actually works for these important financial institutions.

It does not work.

However, the problem is one of policy, not technology. Therefore, we propose a policy solution.

Background

What is PKI?

PKI comprises a set of standards, the organizations that implement those standards, and the devices that use the resulting standardized documents. PKI for websites defines a hierarchy of issuers (i.e., those who can authenticate) and the structure of the certificates themselves (i.e., what data are authenticated). The existence of PKI enables consistent issuance of public key certificates.

Understanding certificates

Certificates are at the core of PKI. A certificate is a set of assertions, often about the identity of a website's owner that is cryptographically signed by a trusted third-party organization, which provides mathematically verifiable evidence of the assertions' validity.

The underlying mathematical structure of a certificate relies on public key cryptography, which uses a set of complementary mathematically based keys, one secret and one public, each of which can decrypt what the other encrypts. Information such as a digital signature is encrypted by the secret key and can be decrypted only with the public key. This means anyone can obtain the public key and confirm that the information was encrypted with the secret key. The secret key and public key are linked to an identifier, and that identifier corresponds to a Certificate Authority (CA), usually a trusted third-party organization, which issues the certificate and attests to certain facts by signing the certificate.

The certificates serve two main purposes.

The first is to confirm that a website is what it claims to be, as a form of identification. Therefore, domain names and the common name of the party responsible for the domain name are in a certificate. For example, if "IU.edu" is the domain name, then that domain name should be listed either as the subject's common name or in the subject alternative name extension. The owner of the website (in this case, Indiana University) should be listed as the subject's organization name in the certificate.

The second purpose is to enable encrypting communication between the domain name and anyone who communicates with that domain. In other words, a certificate should confirm to whom you are speaking and then prevent anyone else from listening in on the conversation. In technical terms, once verification of the presented certificate is complete, the public key encrypts a random pre-master secret, which in turn generates a master secret key and a session key. A secure communication channel is then established between the user's computer and the website. The (symmetric) session key now protects future communication against eavesdropping or modification by a third party.

While there has been considerable work on how users interact with certificate warnings and notifications from their browsers when a website has problems implementing certificates [3, 4], this study focuses on understanding how often these problematic implementations occur today on the web, especially for banking sites.

Potential vulnerabilities

Certificates are mathematically secure and elegant, but improperly implemented certificates can make users vulnerable to attackers.

Lack of authentication results in masquerade attacks, in which individuals trustingly give personal information to a website controlled by an attacker. Masquerade attacks include phishing, pharming, and man-in-the-middle attacks.

Phishing attacks trick the victim into entering information on a false site with an incorrect and possibly misleading domain name. Often, the false site looks extremely similar in design to the legitimate site. The absence of a certificate is common, but so is the use of misleading but apparently trustworthy certificates. For example, an attacker can obtain a legitimate certificate for an obfuscated domain name (e.g., amazon.com.payment.gerin.net) or by hosting the attack in the cloud (thereby leveraging the trusted cloud certificate).

Pharming is a more sophisticated attack, in which the attacker manipulates the victim's software to direct him or her to a website that is incorrect but nonetheless shows the correct domain name. This is done by changing the IP address from that of the actual website to that of the attacker's website in one of the victim's devices. In this case, only certificates can distinguish the two sites. The most common form of attack requires adding incorrect information to the victim's local device (e.g., a laptop or phone), but home routers are also quite vulnerable.

Man-in-the-middle (MITM) attacks occur when an attacker inserts him- or herself into the initial authentication. The attacker pretends to the website to be the user, and pretends to the user to be the website. This is detected by matching the certificate presented by the connected website to the domain requested by the user. Certificate warnings from browsers are often seen when connecting online through public networks at airports, hotels, or coffee shops and a network connection to the network provider's site interrupts the initial user sought-after website (e.g. showing starbucks.com first on a user's device after connecting to a Starbucks network). A malicious party can intercept the same way with MITM, without being visible to the user or the web server. The solution to this attack requires a functional, semantically meaningful PKI.

Even when certificates are implemented, certificates can fail in four ways.

1. The set of facts embedded in the signature is somehow incorrect, either because of changes over time or incorrect issuance.

2. The cryptography could be flawed [5].

3. The software that is supposed to confirm the authenticity of the certificate is flawed, and authenticates flawed or falsified certificates.

4. Individuals could perceive that the certificate means something quite different from the intended issuance and implications.

In our examination of bank certificates, we found problems of the first and second type. Other researchers have documented serious problems in terms of the third type [6, 7, 11]. The fourth type is a focus of ongoing research in industry, in the academy, and indeed in our research group.

Certificate structure and practices

Certificates issued for the World Wide Web follow the X.509 format, which contains the following fields:

1. Certificate version. This field indicates the format of other certificate fields.

2. Serial number. This is a unique certificate identifier assigned by the CA.

3. Signature algorithm. This is the algorithm used to generate and verify the digital signature.

4. Message authentication algorithm. This algorithm generates the message digest, which is the compressed form of the entire certificate information and what is technically verified in the certificate signature.

5. Issuer. This field contains information about the CA that issues the certificate. Common name, organization, physical address (city, state, country), and email are typically but not universally included.

6. Validity. The start and end dates of the certificate's validity period.

7. Subject. This contains information about the entity to which the certificate is issued. Typical components are common name, organization, physical address (city, state, country), and email.

8. Certificate extensions. Depending on the certificate version, several optional but important certificate fields may exist. For example, "basic constraints" can indicate whether a certificate can be used as an intermediate certificate, which can sign subordinate certificates. "Extended key usage" restricts the use of the public key to a list of specific purposes enumerated upon issuance.

The ability to sign subordinates is particularly important for security because any CA can issue a certificate to any website. Since the actual operations of CAs can vary significantly, it may be possible for an attacker to obtain a valid certificate from a less-diligent CA, which will receive identical trust from web browsers as any other certificate.

One approach to address this is the use of Extended Validation (EV) certificates to create more trustworthy certificates. EV certificates are explicit assertions by the CA that there was a higher-than-normal level of due diligence in the certificate issuance. For example, CAs typically do not perform strict verifications on the actual association of an entity requesting a certificate and the corresponding website, since there is no such standard practice in the industry.

However, EV certificates are not widely used. In addition, there is no research indicating that average users actually notice visual cues used to distinguish EV from non-EV certificates in browsers [4]. Thus, due to the significantly higher cost and difficulty of obtaining an EV certificate, the majority of websites still use non-EV certificates.

Finally, practices related to issuing certificates vary widely and change slowly. Reasons for this include the large number of CAs, each of which has its own operational processes and the burden of legacy requirements. Technical practices, expertise levels, and jurisdictional practices vary significantly across certificate authorities.

Therefore, while the existing technical structure of the certificates enables identification of financial institutions, it is the current marketplace dynamics that create disincentives to greater adoption.

Challenges with PKI

Four major categories of failures in PKI include (1) weak cryptography, (2) disorganized revocation, (3) inadequate information, and (4) flawed evaluation software.

1. Weak cryptography

   For those unfamiliar with basic cryptography, this simply means that there are stronger and weaker signatures. Weaker signatures have a greater risk of falsification, just as weakly designed banknotes are at higher risk for forgery.

   The current consensus among the cryptography community is that 1024-bit RSA keys offer insufficient security for the typical validity periods of end-entity X.509 certificates, as attacks against RSA have become increasingly sophisticated. Since 2011, the common recommendation has been for at least a 2048-bit key length for these certificates [8]. Yet in 2014, CAs continued to allow issuance of certificates with 1024-bit RSA keys for validity periods of at least one year. Some argue that this is due to legacy platforms whose software cannot use keys longer than 1024 bits and resource-constrained platforms that expend more processing time and battery power to do public key operations on longer keys. While these factors may constrain key length, they do not constrain certificate lifetime. Thus, there is no justification, particularly for high-value certificates, for the use of lifetimes longer than recommended for a given key length.

Another element of weak cryptograph is the use of hash algorithm. The hash algorithm MD5 was standard for certificate signatures before SHA-1, and it continues in use despite increasingly effective attacks. The Flame malware attack in 2012 took advantage of a collision in MD5 to create a fraudulent certificate [9]. Recognition of the increasingly severe weaknesses in MD5 helped generally eliminate its use in new issuance, but older certificates that use MD5 were still in use as of 2014. Certificates that downgrade from SHA1RSA to MD5RSA and from SHA256RSA to SHA1RSA continue to be observed, although the trend overall is positive; that is, entities that get new certificates may downgrade as well as upgrade.

In both cases, the use of weak cryptography is complicated by the use of long validity periods, sometimes 3, 5, 7 years or more for end-entity certificates. The validity period limits the possible exposure of a cryptography break by rendering a certificate useless by the time an attacker could brute-force the key. When the validity period exceeds this safe duration because of advances in crypto-analysis, these certificates become vulnerable but continue to be accepted.

2. Disorganized revocation

Two standards for revocation, certificate revocation lists CRLs and the Online Certificate Status Protocol (OCSP), are in common use. CRLs are lists of serial numbers of certificates that appear unreliable in terms of cryptography; reliable software can check the lists before accepting the cryptography. The advantage of a CRL is that updated lists can be downloaded periodically. One CRL file can include multiple revoked certificates, saving time when checking several certificates from the same CA simultaneously. The OCSP obtains a certificate's real-time revocation status from the server. It requires confirmation before use by the relying party. OCSP is more responsive to changes in certificate status, but CRLs are less affected by network delays or slow connections.

Practices amongst CAs vary, with some issuing certificates with CRL information, some with OCSP, some with both, and some with neither. Even if the CA implements best practices in its certificate issuance, this problem is further complicated by the irregular behavior of browsers and Web-application clients in checking revocation status. In 2014, Mozilla Firefox decided to use OCSP exclusively, meaning that all certificates with only CRL information in the certificate become effectively irrevocable to Firefox clients [10]. The use of CRL requires a substantial data download compared with the smaller traffic required for OCSP. Clients on constrained data connections, such as cellular connections, may use only OCSP, if they do any revocation checking at all. Apps and other non-browser web clients that use SSL frequently do no revocation checking at all, making it practically impossible to effectively revoke the certificates of servers to which they connect.

3. Inadequate information

   Failures to include appropriate or necessary fields to limit the use and valid applications of a certificate are a recurring problem. In the past, CAs issued certificates with poorly chosen Extended Key Usages (EKUs). The EKU is what restricts a certificate to use only for particular purposes, such as authenticating an SSL server, authenticating a client, signing code, or providing a trusted timestamp. The Flame malware attack also took advantage of an intermediate CA that had an unused but valid code-signing EKU, allowing rogue certificates issued from it to be used to sign code.

4. Flawed evaluation software

   The reality of certificate-checking is a source of serious and legitimate concern [11]. Both Apple [12] and Microsoft [13] have long-lived flaws in software that evaluates certificates. Apple's software practices were relatively more grounded in the use of open code, with the code available to all to review. Yet a significant certificate-authenticating error stood for months. In contrast, Microsoft had internal software engineering and formal code review, and errors in its code lasted even longer. While these are serious issues, they are beyond the scope of this work.

Methods

Approach

We document the current state of bank certificates. We compare these with general-purpose certificates (i.e., the top 1 million websites). We survey the various proposals for the certificate market writ large, including pinning and notaries. We identify how those fit and fail to fit the unique problem of banking certificates.

Having identified the systematic failures in certificates, we discuss the proposals in the technical community for addressing them. None of these resolve the problems that plague online banking. What is needed is a policy solution. We close with a policy proposal, including technical and implementation recommendations, to ensure certificates can be a valid basis for consumer trust.

Collecting certificates

Evaluating the state of certificates in the wild requires large-scale analysis of the certificates. We wanted to be able to answer two questions: First, what is the state of banking certificates? Second, are they more reliable than general-purpose certificates, that is, those used by the top million websites?

We also compare this collection to other efforts. The fundamental difference is that we have the only certificate compilation focused on financial-sector analysis. We also illustrate that our certificate compilation is at least as complete as other approaches, as we complement our daily scans with geographical diversity and multiple data sources.

The dataset we compiled used the PlanetLab research platform [14], allowing us to view certificates from different locations on the globe. Specifically, our scripts run on servers in the United States, both East and West coast time zones. We also ran the scripts on Asian and European servers through PlanetLab. Some certificates can be hosted on content-distributed networks, and thus will be the same from every vantage point. Other certificates are linked to a specific device, so that the same domain will result in different certificates when visited from different places.

We implemented each of the following search and compilation strategies daily from December 18, 2012, until March 2014:

1. **The top one million websites from the ranking of the previous day**. Our script obtains the website list from Alexa every morning and tries to connect to each website on the list via HTTPS. We download a certificate from the website if it is different from our previous observation.

2. **FDIC-insured bank official websites.** The FDIC maintains an official list of its member institutions. For each member, our script retrieves the name, physical address, and official web domain of the bank (if any). The script then removes invalid URLs (e.g., email addresses) from the list, and tries to download a certificate from each valid website on the official list.

   For each FDIC website without a certificate matching the listed domain name, we download the homepage of the website and search for HTTPS links. We download additional certificates by following these hyperlinks. We filter out popular and common links (e.g., “Like Us on Facebook”) from the observations.

Certificates from the two data sources enable us to conduct a thorough analysis on the current status of banking certificates in the United States. By August 20, 2014, we had observed 1.1 million distinct certificates from 3.8 million popular general websites. Note that our geographically distributed exploration results in a far broader view of the PKI than the average user would experience. One study of browser histories illustrated that for a specific individual, some 90% of all root certificates would not be encountered at all [15].

There are similar projects in collecting certificates. The Electronic Frontier Foundation (EFF) actively scans the IPv4 address space for certificates and continuously augments its TLS Observatory [16]. With the EFF browser extension, users can submit their observed certificates and receive warnings from the EFF server if there is a discrepancy between a certificate the user observes and the previously observed certificates stored in the

Observatory. Another centralized certificate notary is maintained by the International Computer Science Institute (ICSI) from live HTTPS traffic passively collected at its participating organizations [17]. Based on this dataset, Amann et al. performed a data analysis on the structural differences between benign certificates and rogue certificates observed in previous CA compromises [18]. Finally, similar to EFF, Durumeric et al., regularly scanned the entire IPv4 address space for certificates and made several recommendations for the PKI ecosystem [19].

None of these organizations made their datasets available for evaluation as a whole dataset. Our results are being made available via Protected Repository for the Defense of Infrastructure Against Cyber Threats (PREDICT) for reproduction or further investigation by others. They support individual queries. The ICSI dataset includes roughly one year of our own compilation that we made available to that project. The larger ICSI dataset was not available for our analysis. We also encountered challenges in accessing EFF's data. Thus, building a dataset for analysis of general certificates was necessary. In addition, while others have evaluated phishing sites and associated certificates, to our knowledge no other group has a dataset of banking certificates.

Analysis of banking certificates

We investigated two approaches for analyzing FDIC-insured bank certificates: direct observation and machine-learning classification. We started by making several direct observations of problems in the banking certificates collected. We then supplemented these insights with machine learning. This led to the discovery of distinct patterns in and between the categories of certificates and systematic differences between non-banking and banking certificates.

Machine learning

We examined the classification performance with three different machine-learning algorithms: J48, NBTree, and Random Forest.

1. J48 is a Java implementation of a traditional decision-tree algorithm, C4.5. This algorithm builds the decision tree based on information gains of each member in the feature set.

2. NBTree is a combination of the Naive Bayes regression and a decision tree.

3. Random Forest is an ensemble algorithm that builds several decision trees and makes the final decision based on a majority vote of all decision trees. For each tree in the forest, it uses only a subset of randomly selected features.

We used machine-learning models to classify certificates into two categories: FDIC-insured banks and general websites. The set of certificates available to be classified as "banks" was

less than one-quarter of all FDIC-insured entities. Table 1 below lists the classification performance. For each algorithm, we report the overall percentages of certificates correctly and incorrectly identified. For each category, we record the true positive and false positive rates, indicating percentages of the correct and incorrect instances for the particular category. As noted in the table, all three algorithms had an overall accuracy rate above 99.4%. The true positive rates for the bank category is 96% for all three algorithms and the false positive rates (i.e. the certificates categorized as banks but are actually in the general category) are as low as 0.1%. For the general website category, the true positive rates are as high as 99.9%, while only 3.7% of the bank certificates were ever misclassified as general. The correctness of classification can improve even further by combining the results of all three machine-learning algorithms.

		J48	NBTree	Random Forest
Overall	Percentage correct	99.48%	99.81%	99.83%
	Percentage incorrect	0.16%	0.19%	0.17%
As bank website	True positive rate	96.30%	96.80%	96.60%
	False positive rate	3.70%	3.20%	3.40%
As general website	True positive rate	99.90%	99.90%	99.90%
	False positive rate	0.10%	0.10%	0.10%

Table 1. Classification performance summary.

Results

Banks without valid websites or certificates

Many banks lacked domains, and thus appropriate certificates. Among all the 27,000 records in the official FDIC list, only 6,000 had valid domains. We tried to connect to every web domain on the list, but we could establish HTTPS connections with only 4,000 of them (Figure 1).

We found no domain or certificate for 20,000 banks. The lack of association of domain name and certificate is problematic for two reasons. First, it means that it would be feasible for an attacker to register a domain for a bank, obtain a certificate for the domain name, and have that be the sole certificate. Second, as banks close, merge, or simply change branding, it would be quite feasible for an attacker to obtain a domain name similar to an expired bank domain, and then obtain a certificate. As no certificate ever would have been issued previously for that domain, none of the proposed changes to the certificate architecture would address such an attack.

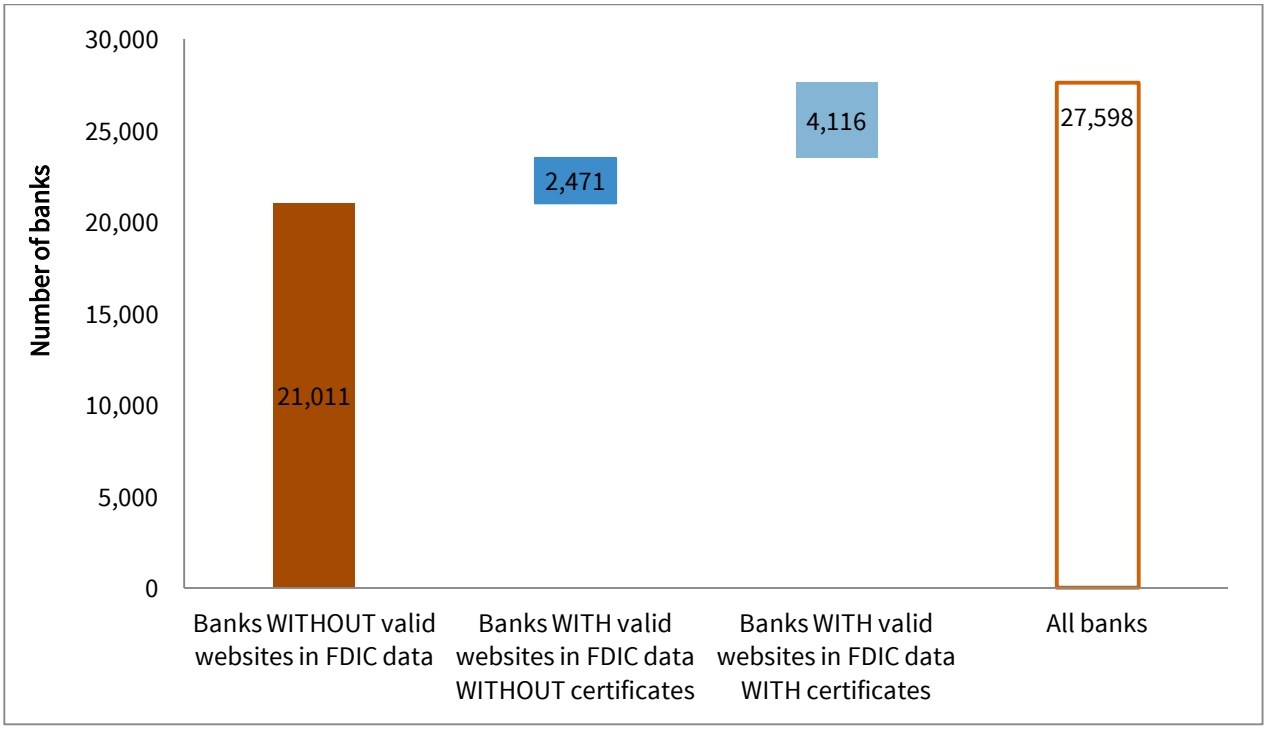


Figure 1. Breakdown of banks by having valid websites and by having certificates.

Issues seen in certificates

1. **Mismatch of the web domain and subject entries in the certificate.** A certificate can be used only in the specific web domains indicated in the subject's common name and alternative name extension fields. Among all downloaded certificates, we discovered 498 domain name mismatches in bank certificates for 41% of bank websites at some point in our data collection (Table 2). For comparison, 84% of general websites had mismatched certificates.

2. **Certificate sharing by multiple domains.** Some web domains shared the same mismatched certificate. This occurs when a single entity hosts online banking for multiple organizations. However, and to greater risk, many of the shared certificates were provided as part of the default server configuration, which has not been changed by their website administrators. As one extreme example, one certificate for sinkdns.org was observed in use by 51 different HTTPS bank domains. A certificate of webaccess1.com was used by 43 different banks. Certificates of the virtualization company Parallels were shared by 37 financial websites. In total, 5% of bank websites used shared certificates.

3. **Period of certificate validity.** With any key, cryptographic or physical, the longer it is unchanged, the more risk of it being subverted. Unlike physical keys, cryptographic keys cannot be tracked, making subversion undetectable. Software vulnerabilities,

such as web clients that misconfigure TLS, can expose keys to risk. This is exacerbated by weak encryptions keys, (5) below.

4. **Missing EKU.** The EKU limits the use of a certificate for the intended purpose. The most common use is to indicate that a certificate cannot be used to issue other certificates. A certificate that is subverted, but issued legitimately, can be used by to create new certificates under the control of an attacker.

5. **Weak encryption standard used.** Not all encryption standards are equally strong. There is no cost-based reason against using best cryptographic practices and obtaining a stronger key with a superior algorithm. Algorithms with well-known weaknesses continue to be issued, presumably for keys that have little commercial value. However, depository institutions are not in that category of customers.

Category	Attribute	% of bank websites	% of general websites	Potential vulnerability
Mismatch of domain and certificate's subject	Yes	41.00%	84.04%	Yes
	No	59.00%	15.96%	
Period of certificate validity	> 2 years	45.56%	24.01%	Yes
	1-2 years	41.90%	46.87%	
	< 1 year	12.54%	30.12%	
Missing EKUs on certificates	Yes	5.76%	20.67%	Yes
	No	94.24%	79.33%	
Encryption standard used	RSA-1024	1.34%	5.33%	Yes
	RSA-2048	95.71%	68.83%	
	SHA-1	49.87%	40.91%	
	SHA-2	49.98%	57.93%	
	Other standards	<1%	1.26%	

Table 2. Potential vulnerabilities uncovered in certificates from bank and general websites. The percentage of bank websites and percentage of general websites for each vulnerability reflects the share of sites with that attribute at some point during the data collection from December 2012 to March 2014.

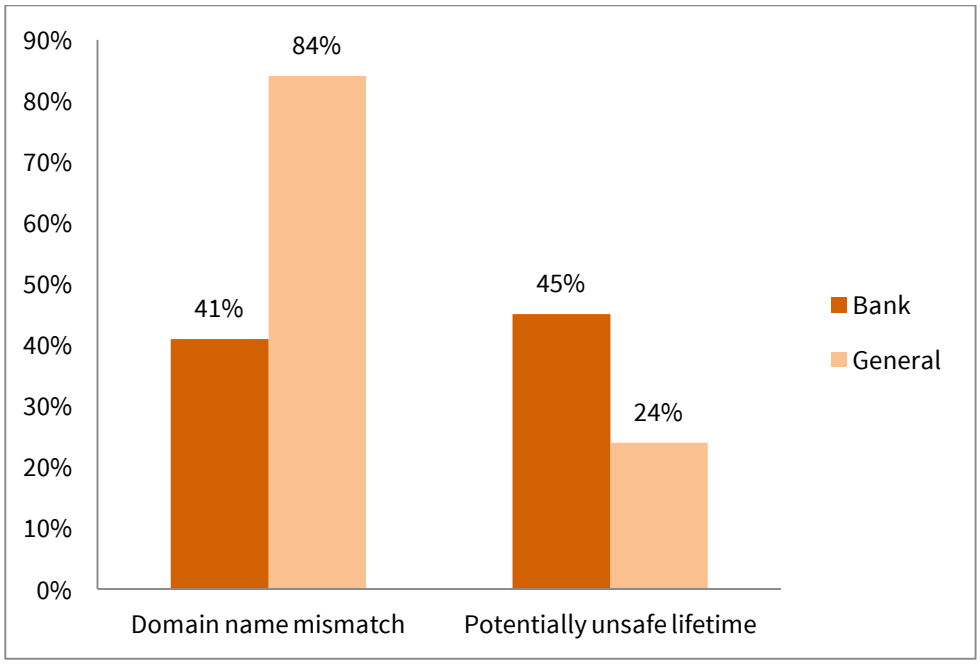


Figure 2. Banks have fewer domain name mismatches – half as many as popular general interest sites —but are much more risk-seeking when it comes to certificate lifetimes.

Vulnerability in the cloud

The issue of cloud computing also makes the lack of consistent, identifiable financial certificates problematic. Botnets provide a platform where there is no constraint on criminal activity. Cloud provider services are also misused by attackers, including attackers who engage in masquerade attacks such as phishing. In our research, 22 sites PhishTank identifies as phishing sites were hosted on Google Drive. In addition, there are reports of criminal use of Microsoft's Azure [20].

Discussion

We showed that there are significant problems with financial certificates. We propose how these might be at least mitigated.

Banks, citizens, customers, creators of web browsers, and other legitimate businesses all have a shared interest in having identifiable and secure bank websites. Creating a mechanism for distinguishing and recognizing banks encourages online banking and online trust.

The technical entities understand the requirements for certificates and the regulatory authorities understand the nature of systematic risk. A collaboration that consists of major cloud providers, banking regulators, cryptographic and interaction experts, browser

manufactures, and selected banks could feasibly create and support adoption of best practices suitable for depository institutions.

Recommended technical best practices

Preventing masquerade fraud against financial institutions requires differentiating legitimate financial sites as distinct from other sites. Rather than trying to identify every phishing site against every bank, a valid cryptographic mechanism could exist for identification of banks only. The simple model of "good versus bad" in PKI fails to provide adequate information. If this were combined with targeted password reuse identification or other mechanisms to flag input into websites, it could make masquerading as a bank far more difficult to masquerade as a bank [21]. Yet any such solution requires a reliable and correct implementation of PKI for banks.

Here we enumerate some basic best practices. None of these proposals are particularly innovative in and of themselves, but combined, they create a list of feasible requirements for high-value certificates, such as for the financial industry.

The X.509 standard itself sets a very low bar for what constitutes a valid certificate. As a result, industry consortiums mandate further requirements, and many of these are obligatory for inclusion in the trusted root certificate list of web browsers. Several issuance best practices can be added on top of these requirements. Although legacy requirements are chiefly why these best practices are not yet required, they should eventually become so.

1. Strong cryptography

   The first best practice is the use of strong cryptography. RSA remains the dominant public key algorithm for certificates, and the cryptographic community recommends at least 2048-bit keys for end-entity certificates. MD5 has been shown to be vulnerable, and new research is exposing vulnerabilities in SHA-1 as well [21]. Therefore, the SHA-2 family of hash algorithms should be employed as part of the signature algorithm as much as possible. The use of strong cryptography then can be augmented by applying reasonable validity periods to the certificates, such as one to three years for end-entity certificates. This limits exposure from any future attacks.

   Where possible, elliptic key algorithms should be considered instead of RSA, as support for these algorithms becomes increasingly deployed. Elliptic curve (EC) keys should have at least 256 bits of length for end-entity certificates. Because the EC standard is still under discussion in both the Internet Engineering Task Force (IETF) and World Wide Web Consortium (W3C), requiring it would be premature. The issues with the National Institute of Standards and Technology (NIST)'s dual EC deterministic random bit generator (DRBG), specifically the potential back door [23] and Bullrun decryption program, [24] reasonably resulted in decreased trust in this standard. While the challenges of operational risk can be handled in part at a national

level, cryptographic standards for browsers and interoperability cannot be based on untrusted curves. Thus, we recommend the use of RSA with a key size of 2048 bits, as market acceptance will not be problematic. Similarly, requiring SHA-2 is a reasonable and arguably necessary step. Maximum validity periods could be determined empirically. A single year would be ideal; however, two is not beyond the pale. The longest lifetime we saw in our compilation is 40 years (happily, not from a bank). Clearly, this is not reasonable.

2. Usable revocation information in certificates

   For cases where there is either compromise of a particular certificate or an attack against an entire class, CAs should include usable revocation information in every certificate. *Usable* means that every major browser and web app that supports any kind of revocation checking can use this revocation information. Although the particulars of revocation checking are beyond the scope of this document, there can be none at all if the CAs do not participate.

3. Discouraging wildcard certificates

   The purpose of the certificate is not only to enable a key exchange to occur, but also, to bind the server's identity to a particular principal, such as a person or a corporate entity, with the authority to use that domain. Wildcard certificates arose with the expectation that all servers under a particular domain name would belong to the same principal. Therefore, it was an acceptable optimization to use a single certificate for a larger set of server names, given that each individual certificate incurs a certain cost.

   The advent of multi-tenant environments turned this expectation on its head. Hosting providers that use load-balancing SSL terminators may deploy the same certificates with multiple domain names used by many different customers. For example, the hosting company godaddy.com may host the domain 123456789.com.
   However, because of the structure of an X.509 certificate, only a single subject name is present, namely that of the hosting company (godaddy.com). The registered owner of the domain exists as a point of contact, but the SSL certificate itself does not correctly identify the site's owner. Yet if the hosting provider allocates hostnames from its own domain name but uses a wildcard certificate, not even that information identifying the site's owner is available. For example, if the site is 123456789.godaddy.com, the certificate may not provide any information about Company 123456789. The use of a wildcard certificate in this case, while expedient, breaks a fundamental assumption of the certificate-based identity model. Therefore, for each site operated by a different entity, CAs should issue unique certificates as much as possible. In situations where this is not possible, such as the SSL terminator scenario mentioned previously, the

CAs should maintain records of attestations from the hosting provider that the domain owners authorize this use.

Wildcards should be discouraged, with a unified certificate issuer being an ideal practice for larger multi-domain entities. Wildcards should be prohibited in multi-tenant environments in the case of hosting services for a depositor entity. A federally insured bank with a domain name should reasonably be expected to have the corresponding certificate, even if that certificate is associated with the domain name as a second-level instead of first-level certificate. Multi-tenant environments can support a unified certificate issuer but may be unable to support domain-specific certificates.

4. Limit EKU per certificate

   Recall the Extended Key Usage (EKU) extension that indicates the purpose or valid use of a certification. A best practice is for CAs to issue separate certificates for separate purposes and not combine multiple unrelated EKUs in a single certificate. In practice, not all certificate chain engines check "transitive EKUs," where not only must the end-entity certificate possess a certain EKU, but all CAs along the path to the root must as well. However, it is still a best practice for a CA to segregate its intermediate CAs by intended purpose, such as server authentication or code signing. Further, it is best practice for a CA to embed EKUs in the certificates of those CAs as well, so that a compromised CA is still limited to its original purposes.

If certificates are a part of operational risk for an individual institution, then systematic weaknesses in the PKI protecting depository interactions are part of the systematic risk for the banking system. Thus, there should be at least a minimal standard. The best practices above are a solid starting point for depository institutions.

Inadequacies of relying only on technical changes

The situation currently has avoidable risks not addressed by any of the proposed technical best practices for improving the PKI. Consider primarily the lack of association between domain names and certificates for 21,011 banks (Figure 1). This lack of association, which leads to the potential for an attacker to create a masquerade site, would not be resolved by any of the current technical proposals.

Phishing is now a race that defenders cannot win. A phishing domain can be detected only after it is used in an attack. Thus, barring a change in policy, there will always be a window of opportunity for phishers. The attack site must further be labeled as malicious, then associated with a warning. Takedowns usually occur within a week or so [25]. The implication of this cycle is that there is no history to new phishing domains to analyze for blacklisting, so history-based proposals for solving the challenges in PKI would fail. Results from revocation mechanisms such as Certificate Revocation Lists (CRLs) and Online Certificate Status

Protocol (OCSP) have a lag between the time when a bogus certificate first appears and when it becomes blacklisted. In the extreme case when a CA is compromised, the CRL and OCSP may become untrusted altogether.

Whitelists such as the Electronic Frontier Foundation certificate observatory, which issues warnings when certificates are inconsistent with the observatory, are also vulnerable. New certificates are not flagged when they first appear. Thus, this common attack could in fact be exacerbated by the existence of the observatory if it were to become trusted.

Another disadvantage of using whitelists and revocation is that these approaches are inherently centralized. In contrast, the use of certificate notaries represents a distributed approach to validate certificates. The Perspectives Project offers a tool that relies on a comparison between the user-submitted certificate hashes and observations made by geographically distributed notary servers [26]. Convergence [27] is a Firefox browser extension that lets users control which data sources (e.g., notaries) to trust without disclosing their network addresses to the data source. However, an average online user may not be able to evaluate the trustworthiness of online notaries. It may make an attack much easier if an adversary runs a notary and can trick other users to trust it.

Certificate pinning associates each website with a small whitelist stored by the local browser. The list is updated upon first visit, as originally proposed in Tsow, Viecco, and Camp [28]. Google Chrome implemented this approach and protected several Google-owned domains against the use of rogue certificates. One weakness of this approach is the long tail in browsing, given the sheer scope of the problem of authenticating everyone.

Under DNS-based Authentication of Named Entities (DANE) [29], Domain Name System Security Extensions (DNSSECs) bind a domain name to its legitimate certificate. The requirements for DANE are universal adoption of both DANE and DNSSEC, on which it relies. We know of no research or evidence that points to a realistic expectation of the global, universal adoption of DNSSEC in the near term. That a specific domain under DANE can be associated with only one issuer solves only a very narrow and unusual class of attacks. It does not solve the problem of attackers with a legitimate domain name masquerading as a bank's official site, including through cloud misuse.

If certificate and domain name providers were capable of not issuing domain names to malware providers, botnet controllers, and other malicious parties, these threats would be a lesser issue. However, certificate and domain providers are not always so scrupulous, and thus are not appropriate gatekeepers. It has been documented that six CAs in recent year issues issued rogue certificates: Comodo [30], DigiNotar [31], DigiCert [32], TurkTrust [33], French Government CA [34], and India CCA [35]. Nor is this only a recent problem. Perhaps most famously, VeriSign issued two certificates in Microsoft's name in 2001 [36], for which Microsoft could only issue a security bulletin, as removing VeriSign as a trusted CA was clearly infeasible (MS01-017).

Finally, DANE's reliance on DNSSEC results in all the problems of DNSSEC being a component of certificate risk. The problems of DNSSEC are both well documented [37, 38, 39] and beyond the scope of this work.

Regulatory and technology policy recommendations

Our policy proposal solves problems that lead to attacks specifically against banks, and does so with no changes to the current technical standards or to the competition among certificate providers. We propose the use of only the best standards and the creation of a mandatory certificate extension for FDIC-insured entities. This could be used to validate a certificate regardless of where it is hosted. Without the ability to identify a remote entity as a bank, masquerade attacks on the financial system will continue. Having a signed extension by a single authority, one that is constant across all FDIC-insured entities, easily can be integrated with the current authentication practices in Firefox, Chrome, and Internet Explorer.

Advocating for identification of specific categories of sites is not new. The W3C Standard Web Security Context: User Interface Guidelines recommend "prior designation of high-value sites," [40], yet this has not been implemented. While the proposal is long-standing, a policy to implement it has been lacking.

The core of our proposal is that a federal entity, such as the U.S. Department of the Treasury or the FDIC itself, take two actions.

1. We propose the development of technical requirements for certificate issuance. Minimal requirements to control operational risk are not in any way a banking regulatory innovation, and specifying best practices in this domain is straightforward. Defining maximum lifetimes and minimal cryptographic strength and recommending extensions are a feasible, reasonable way forward.

2. We propose the cooperative development of a third-party certificate notarization authority that applies only to banks and possibly other important financial institutions. Notice that while this would not be a CA, it would provide cryptographic notarization of an extension for certificates provided by current CAs. Such a notarization could provide proof that a legitimate federally insured bank operated the specific domain name. Rather than having every domain name reseller attempt to prevent any misleading domain name, our proposal would distinguish legitimate banking sites from other sites.

Of course, this proposal could also provide value to cloud service providers, which are currently challenged in that every customer, including masquerading attackers, has an equal capability to use the infrastructure of the cloud. By distinguishing financial institutions from other institutions, our proposal has the potential to decrease the need for cloud providers to invest in providing certificates to every hosted site. By making this a second signature, rather

than a whitelist or a blacklist approach, citizens can use this method without reporting their banking or browsing habits to any third party.

We argue that coordinating and setting up this plan is feasible due to the small number of major browser providers and cloud providers. More-secure interactions serve all parties' interests. Furthermore, augmenting rather than replacing certificate authorities doesn't displace or decrease business. In fact, limiting the lifetime of certificates aligns with CA incentives.

Conclusion

A functioning public key infrastructure requires certificates that authenticate a website to a user before the person authenticates to the website. The current PKI is well established. Yet the challenge of online certification of banks is not solved. The lack of a solution enables tens of thousands of attacks on financial institutions every year. It also enables snooping, allowing eavesdroppers to observe the content of communications with financial institutions. Our policy proposal offers a way to answer the basic query "Is this a bank?", and further to support the confidentiality of connections to banks. Of course, answering that question enables the solution but does not solve the challenges of human factors.

The current policy of relying entirely on competition in the certificate authority market to set standards is inadequate. We illustrated that the current practice of purchasing certificates with neither best practices nor regulatory minimums badly fails consumers, particularly in the financial sector.

The lack of security is widespread. Certificates with incorrect names, incorrectly structured certificates, or cryptographically weak and shared certificates all plague online banking. We show the vast majority of banks (88%) apparently lack the expertise, support, or incentive to implement certificates correctly.

We conclude by arguing for a change in the regulation of certificates for the financial sector. We describe and recommend the adoption of commonly accepted best practices. We propose the creation of a readily identifiable official banking website indicator that requires neither a structural change in the certificates themselves nor in the larger public key infrastructure. Yet our proposal will address the failure of banks to authenticate or secure communications. With the recognition of the indicator in browsers and on cell phones, our proposal would leave phishers who target FDIC-insured institutions high and dry.

The adoption and widespread use of our proposed solutions would counter the concerns that public key certificates, while critical, are "signifying nothing" [36].

Finally, we believe that our proposal can be extended to other important consumer financial institutions beyond FDIC-insured banks. For example, researchers have shown that the tens of thousands of credit unions governed by the National Association of Federal Credit Unions

are generally less secure than service providers for online banking, with problems that include scripting weaknesses and certificate reuse [41].

References

1. Kaspersky Labs. Financial Cyberthreats in 2014. Kaspersky Lab Report. February 2015. https://securelist.com/files/2015/02/KSN_Financial_Threats_Report_2014_eng.pdf

2. Of course there are a range of organization types under the FDIC umbrella, but for readability we refer to these as banks for the rest of the paper.

3. Sunshine J, Egelman S, Almuhimedi H, Atri N, Cranor L. Crying wolf: An empirical study of SSL warning effectiveness. USENIX Security Symposium. August 10, 2009. http://dl.acm.org/citation.cfm?id=1855793

4. Biddle R, van Oorschot P, Patrick A, Sobey J, Whalen T. Browser interfaces and extended validation SSL certificates: an empirical study. Proceedings of the 2009 ACM workshop on Cloud computing security. November 13, 2009. http://dl.acm.org/citation.cfm?id=1655012

5. For example, and as discussed later, research shows that the hash functions MD5 is vulnerable to collision attacks.

6. Holz R, et al. TLS in the wild: An Internet-wide analysis of TLS-based protocols for electronic communication. arXiv preprint arXiv:1511.00341. 2015. http://arxiv.org/abs/1511.00341

7. Meyer C, Schwenk J. Lessons Learned From Previous SSL/TLS Attacks-A Brief Chronology Of Attacks And Weaknesses. IACR Cryptology ePrint Archive. 2013. https://eprint.iacr.org/2013/049

8. Barker E, Roginsky A. Transitions: Recommendation for Transitioning the Use of Cryptographic Algorithms and Key Lengths. National Institute of Standards and Technology. January 2011. http://csrc.nist.gov/publications/nistpubs/800-131A/sp800-131A.pdf

9. Microsoft. Microsoft security advisory 2718704: Unauthorized digital certificates could allow spoofing. June 13, 2012. http://technet.microsoft.com/en-us/security/advisory/2718704

10. Mutton P. Certificate revocation: Why browsers remain affected by Heartbleed. Netcraft. April 24, 2014. http://news.netcraft.com/archives/2014/04/24/certificate-revocation-why-browsers-remain-affected-by-heartbleed.html

11. Brubaker C, Jana S, Ray B, Khurshid S, Shmatikov V. Using frankencerts for automated adversarial testing of certificate validation in ssl/tls implementations. Proc. of SP '14'. IEEE Computer Society. May 18, 2014. http://dl.acm.org/citation.cfm?id=2650793

12. Apple. About the security content of ios 7.0.6. April 6, 2015. http://support.apple.com/en-us/HT202934

13. Saade T. The stuxnet sting. Microsoft Malware Protection Center, Threat Research & Response Blog. http://blogs.technet.com/b/mmpc/archive/2010/07/16/the-stuxnet-sting.aspx

14. PlanetLab. https://www.planet-lab.org/

15. Braun J, Rynkowski G. The potential of an individualized set of trusted CAs: Defending against CA failures in the web PKI. 2013 International Conference on Social Computing. IEEE. September 8, 2013. http://ieeexplore.ieee.org/xpl/login.jsp?tp=&arnumber=6693387&url=http%3A%2F%2Fieeexplore.ieee.org%2Fxpls%2Fabs_all.jsp%3Farnumber%3D6693387

16. Electronic Frontier Foundation. The EFF SSL Observatory. Accessed February 24, 2016. https://www.eff.org/observatory

17. Amann J, Vallentin M, Hall S, Sommer R. Extracting certificates from live traffic: A near real-time SSL notary service. Technical report TR-12-014. ICSI. November 2012. http://www.icir.org/johanna/papers/icsi12extractingcertificates.pdf

18. Amann B, Sommer R, Vallentin M, Hall S. No attack necessary: The surprising dynamics of SSL trust relationships. Proc. of ACSAC '13. December 9, 2013. http://dl.acm.org/citation.cfm?id=2523665&dl=ACM&coll=DL&CFID=721071513&CFTOKEN=34580039

19. Durumeric Z, Kasten J, Bailey M, Halderman J. Analysis of the HTTPS certificate ecosystem. Proc. of IMC '13. ACM. October 23, 2013. http://dl.acm.org/citation.cfm?id=2504755

20. Segura J. Cyber-criminals interested in microsoft azure atoo. Malwarebytes Labs. April 29, 2014. https://blog.malwarebytes.org/fraud-scam/2014/04/cyber-criminals-interested-in-microsoft-azure-too/

21. Such local data storage would require encryption, and simple comparison is one operation that can be made on encrypted data.

22. Stevens M, Karpman P, Peyrin T. Freestart collision for full SHA-1. IACR-EUROCRYPT-2016. February 22, 2016. https://eprint.iacr.org/2015/967

23. Shumow D, Ferguson N. The possibility of a back door in the NIST sp800-90 dual ec prngprng. Crypto 2007 Rump Session. August 19, 2007. http://rump2007.cr.yp.to/15-shumow.pdf

24. Perlroth N, Larson J, Shane S. Secret documents reveal N.S.A. campaign against encryption. New York Times. September 5, 2013. http://www.nytimes.com/interactive/2013/09/05/us/documents-reveal-nsa-campaign-against-encryption.html

25. Moore T, Clayton R. Examining the impact of website take-down on phishing. Proceedings of eCrime. October 4, 2007. http://dl.acm.org/citation.cfm?id=1299016

26. Wendlandt D, Andersen D, Perrig A. Perspectives: Improving SSH-style host authentication with multi-path probing. ATC334. USENIX 2008 Annual Technical Conference. June 22, 2008. http://dl.acm.org/citation.cfm?id=1404041

27. Marlinspike M. The convergence project. Accessed March 2, 2016. http://convergence.io

28. Tsow A, Viecco C, Camp J. Net Trust: PrivacyP-aware architecture for sharing web histories. IBM Systems Journal. August 2007. http://www.cs.indiana.edu/cgi-bin/techreports/TRNNN.cgi?trnum=TR651

29. Barnes R. DANE: Taking TLS authentication to the next level using DNSSEC, IETF Journal. October 2011. http://www.internetsociety.org/articles/dane-taking-tls-authentication-next-level-using-dnssec

30. Hallam-Baker P. Comodo SSL affiliate the recent RA compromise. Comodo Blog. March 23, 2011. https://blogs.comodo.com/uncategorized/the-recent-ra-compromise/

31. Fisher D. Diginotar says its ca infrastructure was compromised. Threat Post. The Kaspersky Lab Security News Service. August 30, 2011. https://threatpost.com/diginotar-says-its-ca-infrastructure-was-compromised-083011/75594/

32. DigiCert. 2nd clarification statement by Digicert SDN berhad. November 7, 2011. https://www.digicert.com.my/news/news_20111107.htm

33. From McDonald M, Cranor F. The cost of reading privacy policies. ISJLP 4, 543. Microsoft . Microsoft security advisory 2798897: Fraudulent digital certificates could allow spoofing. January 14, 2013. https://technet.microsoft.com/library/security/2798897

34. ANSSI. Revocation of an igc/a branch. 2013. http://www.ssi.gouv.fr/en/the-anssi/events/revocation-of-an-igc-a-branch-808.html

35. Microsoft. Microsoft security advisory 2982792: Improperly issued digital certificates could allow spoofing. July 17, 2014. https://technet.microsoft.com/en-us/library/security/2982792.aspx

36. Forno R, Feinbloom W. Inside risks: PKI: a question of trust and value. Communications of the ACM. June 1, 2001. http://dl.acm.org/citation.cfm?id=376184

37. Osterweil E, Ryan M, Massey D, Zhang L. Quantifying the operational status of the dnssec deployment. Proceedings of the 8th ACM SIGCOMM conference on Internet measurement. October 20, 2008. http://dl.acm.org/citation.cfm?id=1452548

38. Yang H, Osterweil E, Massey D, Lu S, Zhang L. Deploying cryptography in internet-scale systems: A case study on DNSSEC. Dependable and Secure Computing, IEEE Transactions. 2011. http://irl.cs.ucla.edu/~eoster/doc/todsc-paper.pdf

39. Lian W, Rescorla E, Shacham H, Savage S. Measuring the practical impact of DNSSEC deployment. USENIX Security. August 14, 2013. http://dl.acm.org/citation.cfm?id=2534816

40. Roessler T, Saldhana A. Web security context: User interface guidelines. World Wide Web Consortium LastCall WD-wsc-ui-20100309. August 12, 2010. https://www.w3.org/TR/wsc-ui/

41. Bisht P, Venkatakrishnan V. XSS-guard: Precise dynamic prevention of cross-site scripting attacks. Detection of Intrusions and Malware, and Vulnerability Assessment. 2008. http://link.springer.com/chapter/10.1007%2F978-3-540-70542-0_2 http://arxiv.org/pdf/1511.00341.pdf See [7] for previous attacks

---

Authors

Zheng Dong is a data scientist in Microsoft's Safety platform group, where his work focuses on protecting users against online phishing attacks. Zheng obtained his doctoral degree in 2015 from IUB; his work focused on innovatively designing and implementing several machine-learning mechanisms to detect phishing and malicious certificates. Before that, he earned a master's degree in computer science at Indiana University. Zheng is published in academic conferences and journals, and has served on the program committee for several security conferences and workshops.

Kevin Kane is a principal software engineer in Microsoft's security and cryptography team. His work currently focuses on security models for the Internet of Things and its emerging industry standards. Previously, he worked on anomalous X.509 certificate detection in the global PKI, applications of security processors in mobile devices, declarative access control policies, and the use of such policies in virtual machine monitors. He earned his master's and doctoral degrees from the University of Texas at Austin in 2005 and 2006, respectively.

Siyu Chen is a software developer at Neurocrypt, where he is responsible for building and maintaining the company's website and developing new products. As part of the IU security lab in graduate school, he was responsible for researching and collecting certificates, as well as managing their database. He has a master's degree in computer science from Indiana University.

L. Jean Camp is a professor at the School of Informatics and Computing at Indiana University Bloomington (IUB). She joined IUB from Harvard's Kennedy School after a year as senior member of the technical staff at Sandia National Laboratories. She has a doctorate from Carnegie Mellon and an MSEE from University of North Carolina-Charlotte. She is the author of two monographs and more than 160 over one additional works. She has made scores of presentations across six continents. Her professional service included a year as a Congressional Fellow of the IEEE under the aegis of the AAAS.

Research was sponsored by the Army Research Laboratory under Cooperative Agreement Number W911NF-13-2-0045 (ARL Cyber Security CRA). This material is based upon work supported, in part, by the DHS BAA 11-02-TTA 03-0107 Contract N66001-12-C-0137, Cisco Research Support Proposal 591000, Google Privacy Security Focused Research Program, and Microsoft. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the DHS, DoD, Google, Cisco, Microsoft, or Indiana University. The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the Army Research Laboratory, Department of Homeland Security, Google, Microsoft, NSF, Indiana University, or the U.S. government. The U.S. government is authorized to reproduce and distribute reprints for government purposes notwithstanding any copyright notation herein. These funding sources had no involvement in study design; in the collection, analysis, and interpretation of data; in the writing of the report; or in the decision to submit the article for publication.

Referring Editor: Latanya Sweeney

Citation

Dong Z, Kane K, Chen S, Camp L. The New Wildcats: High-Risk Banking From Worst-Case Certificate Practices Online. *Technology Science*. 2016041401. April 14, 2016. http://techscience.org/a/2016041401

Data

Protected Repository for the Defense of Infrastructure Against Cyber Threats. https://www.predict.org/Default.aspx?tabid=169