

Enhancing browser security by evaluation from public domain databases and business registries

Casper Skovmand Agesen



Lyngby 2014

Photonics-Compute-MSc-2013/2014

Technical University of Denmark

Department of Photonics Engineering
Building 343, DK-2800 Lyngby, Denmark
Phone +45 4525 6352, Fax +45 4593 6581
info@fotonik.dtu.dk
www.fotonik.dtu.dk

Department of Applied Mathematics and Computer Science
Building 303B, DK-2800 Lyngby, Denmark
Phone +45 4525 3031, Fax +45 4588 1399
reception@compute.dtu.dk
www.compute.dtu.dk

Contents

I.	Summary	1
II.	Related work.....	2
III.	Acknowledgements	3
1.	Introduction	4
2.	Why online security is so important.....	7
2.1.	In- and outgoing types of data, the human is the weakest link.....	8
2.2.	Phishing, a problem on a global scale.....	10
2.3.	Measures to prevent phishing and fraud.....	13
3.	Fundamental internet mechanisms are trusted implicitly.....	14
3.1.	DHCP.....	15
3.2.	DNS.....	16
3.3.	IP routing.....	17
3.4.	ARP.....	19
3.5.	Summary on ill placed trust in the basic internet functions.....	22
4.	Reputation, trust and identity in physical vs. digital domains	23
5.	The audience that has the need of educated guidance.....	26
5.1.	Personal experiences about the common user.....	26
5.2.	Users are not <i>stupid</i> but <i>unaware</i>	28
6.	Usability and security seldom go hand in hand.....	29
6.1.	Choosing the right design for the right task and audience	29
6.2.	The Windows UAC done wrong, an example from a workplace.....	31
6.3.	Designing a trade-off between usability and security.....	38
7.	The banks are only very rarely safety nets for online transactions	40
7.1.	Account to account transactions	40
7.2.	Credit card payments	41
7.3.	How the banks want to provide better online safety	42
7.4.	Summary on which role the banks are playing.....	46
8.	The digital certificates and padlocks	47
8.1.	How the digital trust schemes manifest themselves.....	50
8.2.	Extended validation	55
8.3.	Certificate revocation methods.....	58
9.	Various degrees of encryption	60
9.1.	Secure sockets layer (SSL).....	61
9.2.	Transport layer security (TLS)	62
9.3.	MD5 and SHA-1 are still being used, even when insecure.....	63
10.	Alternatives to traditional certificate PKI structure	64
11.	Creating a browser extension that does it right	66
11.1.	The roles of and limitations by using DK-Hostmaster and CVR	67
11.2.	The toolbars that did not achieve the desired effect.....	69
11.3.	Hope remains for making users rely on add-on programs	71
12.	From addressing the problem to not becoming the problem	73
13.	Conclusion.....	74
14.	References.....	75

I. Summary

The goal of the thesis is to provide a design for a browser plugin that can support the identification of Danish companies behind Danish websites and thus cement their validity and integrity by lookups in public databases and cross checking the data. This is needed because internet identities and their encryption methods are bought from companies that have to make a revenue and are not provided by the physical institutions that issue identities to its citizens. Therefore, money is a major instigator when it comes to digital trust schemes.

To help users see through phishing fraud is a major instigator for attempting to design such a plugin and there is a heavy emphasis on user studies with more or less successful attempts in trying to make them change default behaviour. The articles used are from 2004 and onwards and while technology has advanced, the basic issues of having to do with unaware users appear to stay the same. Alternative means of getting users to adapt security initiatives is therefore explored and elaborated.

There are also recent alternatives to current hierarchal certificate trust structure and it will solve distinct problems. In particular with having many independent top-level entities and their own chains of trust, in which they are allowed to trust themselves when nobody really should do that. It is also very easy to implement strictly on server side and is already live across many systems being used daily and will prevent pre-installed trust distribution with web browsers, however an actual break with the old methods have yet to come.

II. Related work

Developing supportive tools for complicated mechanisms has been around for quite some time and the conflict between making something secure and user friendly has been and still is a subject for much debate. There exists multiple user studies about if already implemented security schemes are working the way they should or if new initiatives fare any better.

Four articles have had an especially large significance on my thesis and they are, sorted after which year they were published, the following:

“Aligning security and usability” by Ka-Ping Yee from 2004. Yee identifies design problems in operating systems and suggests alternative ways to make and take. He lists 10 guidelines for secure interaction design that all are relevant to consider, also ten years later here in 2014. [8]

“Do security toolbars actually prevent phishing attacks?” by Min Wu from 2006. Wu and his two fellow researchers look if the various offers in security toolbars work as intended on two groups of subjects. They equip the first group with printed tutorials for the toolbars and they show promising results but the second group without tutorials do not change behaviour at all and remain easy targets to swindle. [37]

“Security usability principles for vulnerability analysis and risk assessment” by Audun Jøsang from 2007. Jøsang and his group of four other researchers list actions and principles regarding what is required of users and the inverse, where they identify the cause of the same principles’ vulnerabilities. [16]

“Exploring user reactions to new browser cues for extended validation certificates” by Jennifer Sobey in 2008. Sobey and her team of three other researchers look at if the initiative from 2007 that allows a browser’s address bar to turn green has any kind of informative effect on the users. They provide their own alternative interface as well and find that it achieves the better results. [23]

III. Acknowledgements

I would like to thank the following people:

Christian Damsgaard Jensen, associate professor.

My supervisor from DTU Compute whose ability to take a clumsy brainstorm and turn it into a project description made this thesis possible, after having spent months of getting insignificant results with another sparring partner.

Kristian Falk Sidelmann, network and security manager.

For always having had the time and interest to discuss current network, server and security topics at our former common workplace and for having played no small part in my choices of both educational and career paths.

Thomas Møller Nielsen, client adviser.

My contact from Arbejdernes Landsbank who shared his insight concerning the banking industry, a sector we are all familiar with to an either lesser or greater extent, yet their business aspects are seldom part of a portfolio for their clients.

John Schweitzer, CEO at DIFO and DK-Hostmaster.

For, a quite uncommon act in my opinion, calling me the day after I had sent him a request to set up a meeting and spent a good amount of time describing their processes and what role they played.

My family and friends.

For all the obvious reasons.

1. Introduction

The current method of providing internet browser security is not as secure or even valid as one may be led to believe. It is based on digital certificates, issued by companies that sign and thereby forward their trust into the integrity of a given domain name, ultimately meaning that a browser trusts a website because a third party company does.

The most popular browsers and applets counting Microsoft's Internet Explorer, Google's Chrome, Mozilla's Firefox, and Apple's Safari are already from installation knowledgeable of hundreds of different issuers of certificates with little to no afterthought on, if the list has become bloated or deprecated. And due to the global perspective they operate in, as a resident in Denmark you are also trusting issuers from Turkey, South Africa and Indonesia, even if you are never going to visit a website they are trusting.

The easiest explanation to this common implementation is due to sheer convenience. Supplying a browser with no pre-trusted certificates will require an amount of knowledge from every user that both very few possess and perhaps most importantly, are prepared and willing to spend in order to continue with their activities and may just choose another and more manageable browser instead. Thus, while the topics of security and trust are of high importance, the necessities of usability are even higher than that, if users are going to adopt and adapt new initiatives.

Because of that, I want to research the possibility of developing a method that can provide guidance to a user that is attempting to determine the authenticity of a given website. The feasibility of the developed method will have to be evaluated through the design of a plugin for a popular browser.

The first task of that will be to investigate web authentication in order to determine and describe the necessary steps needed for a website owner to achieve the padlock icon on a user's browser. The investigation of web site authentication will be used to examine how it can be compromised and if it is possible to locate and identify certain trends in the utilised methods.

To involve users is most likely going to be the largest challenge, as it has to provide maximum value with minimum involvement and so different methods of grading and presentations for the user will have to be researched and somehow tested. Design will be absolute key in ensuring its widespread usage, where the assessment categories are likely to be a check of whom the domain is registered to and if that company exist in the business registry along with the option of grading the provided encryption strength and method.

A common denominator unfortunately seems to be, that the work carried out by one group of researchers, which appears to yield some promising results, only gets to remain interesting for one or two years until new research arrives that disproves the initial research.

One persistent result often remains, being that it is impossibly difficult to expect users to look for security cues or other types of information by themselves, since there is too much “carrot” and not enough “stick” where nobody seems to be particularly interested in the carrot either.

Much of this is very likely due to human nature: To get the job done satisfactory but exert as little effort as possible. Two options show themselves in which way it can be solved: Either rely on the users to make the educated choice by having read and understood the underlying procedures for long-term decision-making. Alternatively, simply accept that this picture, no matter the size of effort, will rarely come true and the browser plugin eventually will have to provide so good value that it can be utilised properly without having to read any tutorials.

Due to the required amount of insight of human nature, I propose to work closely with an anthropologist, especially if users are to be somehow “tricked” into performing well and appreciate that result as if they reached it on their own. Because the returning hurdle for other toolbar inventors seems to be, that the users attach less and less significance to their presence as time passes when it does not give them meaningful feedback.

Therefore, in this thesis, I present my findings on how users are being led into fraudulent schemes and the initiatives taken to help prevent it. Along with comparisons of trust in the digital and physical domains, where trust in website certificates are both vastly different from physical evidence and even before they come into play, there is already other digital systems being automatically trusted implicitly.

Much effort has been laid into exploring the connection between users and inadequate usability, where good ideals can end up becoming more of a hindrance than actual help, where users fabricate their own ways around it.

Fraud is usually synonymous with loss of money, so the role our banks are having is also elaborated upon, where they always seem to find a way out for themselves.

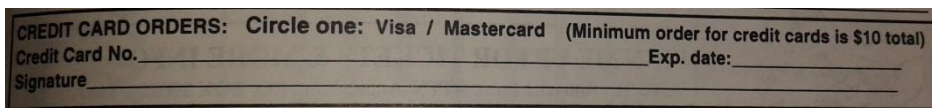
A major part of this thesis is centred on digital certificates and the authenticity they do (not) provide along with cryptographic services, where the issuers and leading software companies complacently cling to outdated and insecure standards.

Finally, I provide my own vision of how the structure of a plugin relying on public authority databases could look like, why I despite earlier failed toolbar attempts remain positive about its success and how I want to prevent it from only shifting immense power from certificate issuers to the public authorities.

2. Why online security is so important

When discussing trust between two human beings, it is implied that someone places a certain amount of good will in another person, meaning that they believe that the individual lives up to a mutual agreement. *Reputation* plays a significant role, in the sense that it can open or close a vast amount of doors, depending on *whom* you know and who knows *you*. It is a challenge to attempt to apply the same theory on human beings and the inner workings of the internet mechanisms, especially since the latter is of such a foreign character to many, where issues of trust between people are to be considered an everyday occurrence whereas behind a screen, it becomes a different matter.

Few will likely argue that being in control of one's own personal information is bad, but will at the same time place a distinction between losing a credit card somewhere and using its details over the internet. Ideally, there should be no difference between them since the name, card number, expiry date and the three security letters are the same whether read on a stolen piece of plastic, by eavesdropping on an unencrypted data stream or hacking a database. Yet there is a far greater fear linked with losing a credit card than actually using the details it shows on the front to use for payment.



CREDIT CARD ORDERS: Circle one: Visa / Mastercard (Minimum order for credit cards is \$10 total)
Credit Card No. _____ Exp. date: _____
Signature _____

Figure 1: A comic book order form from 1996, requesting card details sent openly via mail

Compare two examples where:

- A. Someone immediately checks his belongings for a credit card after waking up at home from a night out on the town as he might have dropped it.
- B. Him being less concerned with the security of a shady internet store where he made a cheap purchase, after having come back home while still under influence.

In both cases, there is a risk that the information on the credit card might have been compromised. Case A deals with someone either intercepting the details when the card is out or copying the details onto a credit card replica, for instance in a bar or restaurant.

In case B the card does not physically leave its owner like in case A, but the necessary information it holds to make purchases with, does.

The PIN code is only a small comfort since its utilisation is not universal because a signature on a receipt is often enough. However, by using the PIN, at least the card is always in the vicinity.

2.1. In- and outgoing types of data, the human is the weakest link

Since the 90s where internet access started to become widespread, there has been an expanding industry of security software and in particular personal antivirus. With e-mail “spam” becoming colloquial language, so did the awareness of having to protect one’s computer against digital attacks and there were some hard lessons learned following having been a victim of a virus attack. That also includes myself, who experienced his first virus delete the computer’s start-up process and format the hard disk, rendering it completely inoperable. The virus had come from a CD borrowed from a classmate, which he had burned himself. Being one of the first to get internet in 1996, he had got the virus that way and then passed it unknowingly onto others.

There are two different ways of becoming a victim of fraudulent schemes, either by infection or by submitting information. The first is the easiest to prevent as it mainly targets specific systems, where the second seeks to trick the user into handing over confidential data.

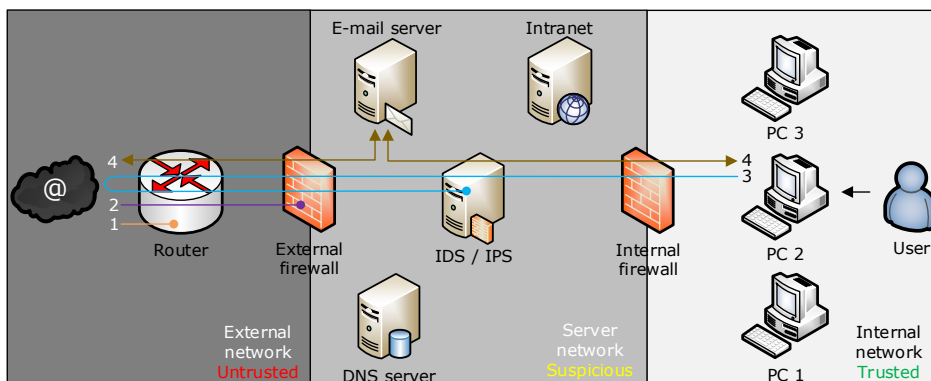


Figure 2: Unsuccessful and successful attacks on a company

Figure 2 is the depiction of how any given company without its own web services might have set up its internal network structure to prevent attacks. The first attack 1 is denied by the router. Attack 2 is prevented by the external firewall. Attack 3 is foiled when the user tries to download something from the internet and it is caught by the intrusion detection/prevention system (IDS/IPS). Attack 4 is a personal email attack, where, upon opening it, the email infects the computer and tries to infect the other PCs on the internal network.

What it means is that security systems are excellent at detecting system attacks but worse at combating attacks that has the user in mind. At the same time, humans are regrettably poor at detecting schemes devised by other humans (but still better than computers) which is precisely why phishing is targeting humans. Rachna Dhamija and J.D. Tygar call it “The limited human skills property” [1]:

Humans are not general-purpose computers. They are limited by their inherent skills and abilities.

An example is a staged penetration test by the American Homeland Security Department (HSD) in 2011, where staff dropped a number of “phone home” USB thumb drives on their parking lot in collaboration with a network security firm. Curious employees inserted 60% of those drives into HSD’s computers and if they bore the HSD logo, the number was as high as 90%. The network security firm’s CEO commented that [2]:

There is no device known to mankind that prevents people from being idiots.

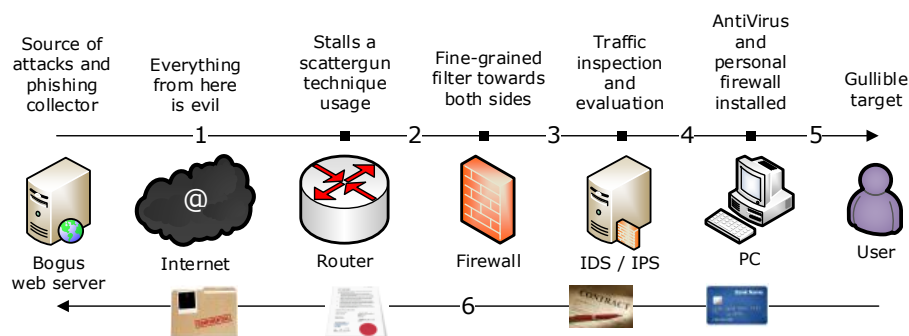


Figure 3: If you can reach the gullible target, then you hit the jackpot

Figure 3 depicts five stages of an attack that has the user as target. They all have to succeed before stage six happens, where the user sends personal and classified information back. The systems are not yet good enough at detecting what kind of information is transmitted, for instance checking for 16-digit credit card numbers and stop that kind of traffic. Encryption makes that even harder if not impossible. Credit card information, contracts, deeds and other important personal information gets treated no differently than using a web-based email service to send pictures of cats and a grocery list to other users on the internet. Especially because it all happens by ordinary web-traffic usage and if that is restricted, nobody in the company can use their PC to visit work-relevant websites.

The lesson to learn from this is that no matter the training and working environment, human curiosity sometimes leads to regress rather than progress. Ultimately, one must also come to realise that despite a plethora of security systems, humans are still remarkably easy to deceive. One might instead restrict users from downloading and running suspicious programs, but there has not yet been invented a method to prevent them from handing over their credit card numbers and social security details on any given website and press a button that says “Submit”.

2.2. Phishing, a problem on a global scale

According to a website called Word Spy, the word “phishing” turned up in January 1996 and had its first citation in the media in March 1997. It is explained by “creating a replica of an existing web page to fool a user into submitting personal, financial, or password data. [3]

An organisation called Anti-Phishing Working Group (APWG) is a collaboration of more than two thousand institutions worldwide and advises governmental institutions, trade groups and treaty commissions. Every three to six months they gather their findings in statistics reports that are available to the public. Some of those reports will be used accordingly here. [4]

APWG's key findings during first half of 2013 are [5]:

- Vulnerable hosting providers are contributing to phishing due insufficient awareness of suspicious traffic to and from their systems.
- China is a major victim of phishing because the middle class' newfound prosperity makes it a popular target for fraud.
- The number of targets has gone up which indicates that phishers are looking for new opportunities.
- Inattentive or indifferent domain name registrars and registries are being fooled by phishers.
- On average, the persistence of phishing attacks is climbing.

	1H2013	2H2012	1H2012	2H2011	1H2011	2H2010
Phishing domain names	53,685	89,748	64,204	50,298	79,753	42,624
Attacks	72,758	123,476	93,462	83,083	115,472	67,677
TLDs used	194	207	202	200	200	183
IP-based phish (unique IPs)	1,626	1,981	1,864	1,681	2,385	2,318
Maliciously registered domains	12,153	5,833	7,712	12,895	14,650	11,769
IDN domains	78	147	58	36	33	10
Number of targets	720	611	486	487	520	587

Table 1: APWG's six latest half-yearly statistics

As Table 1 shows, there is a decrease from the second half of 2012 and to first the half 2013 in both the amount of different domain names used, attacks therefrom and the number of top-level domains (.dk, .org, .com, .info, etc.). However, at the same time, there are much more maliciously registered domains and overall targets. The total amount of phishing domains minus the purposely maliciously registered ones equals 41,532 considered hacked or compromised. The increase in malicious ones are found to be from an uptick by Chinese phishers and while 194 top level domains (TLD) have been used, 159 were coming from three only, being .com, .tk and .info.

Attacks by Industry, 1H2013 - Excluding Shared Virtual Server Attacks

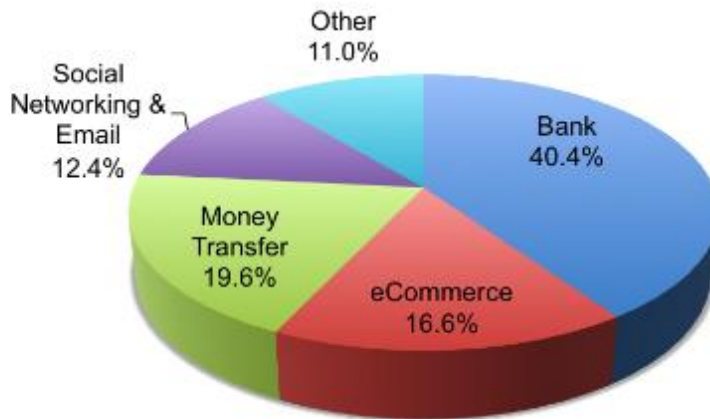


Figure 4: APWG's diagram on target distribution

Table 1's 720 targeted institutions in 2013 have been split up in Figure 4, where online money transfer PayPal.com has been the most targeted with just over 18% of the 72,758 and next in line is Chinese Taobao.com with 9%. The 80 most attacked targets were hit over 100 times each and out of the remaining 640 targets, half of those were hit up to three times each in first half of 2013.

One desirable piece of information not included in the APWG report is how much money is estimated to have been lost due to phishing. Instead, the security company RSA had by August 2012 some results from first half of 2012 where they estimate, that an amount of \$687 million was lost worldwide [6]. Even if the number of attacks have gone down from the first half of 2012 to the first half of 2013, it cannot be said for certain if that also means that the amount of money has gone down as well.

2.3. Measures to prevent phishing and fraud

There have been multiple ideas for helping people see through phishing attempts while maintaining their own integrity by having them choose a self-selected scheme they recognise and feel secure about using. The most prominent solution known today seems to be image recognition, meaning that if user sees an image he or she is familiar with, only then is it safe to assume that the website has not been tampered with. It is also important to take notice when designing new security initiatives, that users follow “the path of least resistance”. Rachna Dhamija and Lisa Dusseault write about that in correspondence with developing new systems having high usability [7]:

Ironically, attackers are experts in usability – they know how to exploit users’ lack of understanding and their tendencies to use shortcuts by developing social engineering attacks to steal identity information.

In 2005, Rachna Dhamija and J.D. Tygar proposed a solution that has the user select and remember a specific image, which will then occur every time the user wants to authenticate himself somewhere, see Figure 5. At the same time, Dhamija and Tygar also propose a change to the browser windows where the user has logged in successfully, being that its background changes complexion. Their key point is that users are better at remembering an image and notice the change of background than remembering passwords and checking website certificates. [1]



Figure 5: A trusted password window

3. Fundamental internet mechanisms are trusted implicitly

Some of the internet's mechanisms are so pivotal that they naturally require trust placed in them, but even there complications can happen. Examples of this are the translation of an address expressed with letters that humans understand, into a binary string consisting of zero's and one's, expressed by internet protocol (IP) numbers – the domain name system (DNS), the dynamic host configuration protocol (DHCP) and the address resolution protocol (ARP).

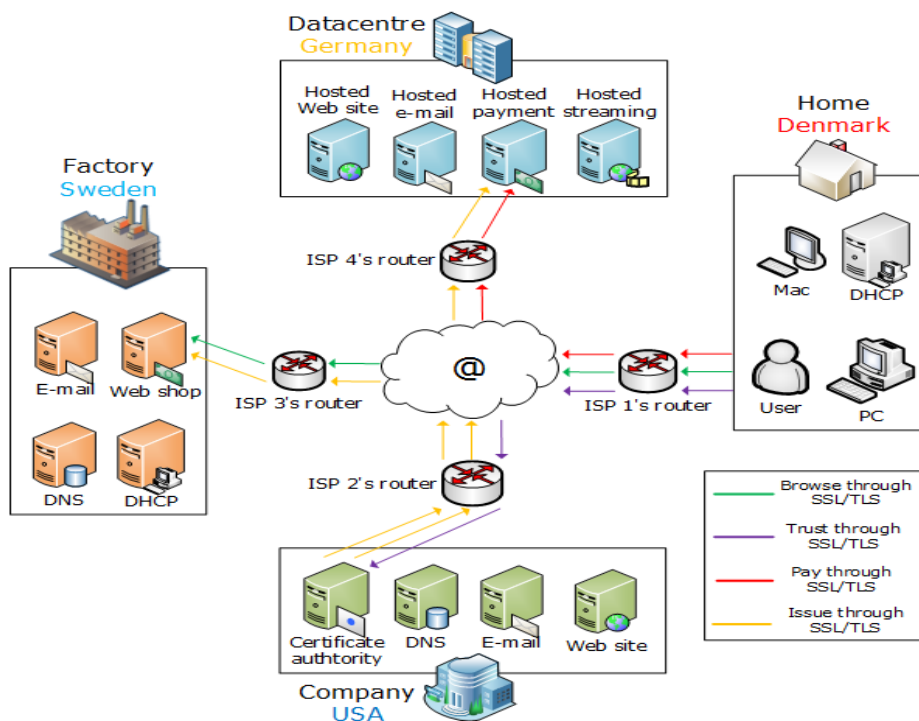


Figure 6: An interpretation of internet locations where trust is not a tangible subject

Figure 6 is the depiction of a web shop purchase through the usage of digital certificates and encryption (SSL/TLS). The user at home uses either his PC or Mac to visit the factory's own web shop through the green line. When payment is about to take place, the red line illustrates contacting a specific payment handler, which proceeds to withdraw money from the user's bank account and deposit them into the factory's bank account.

The certificate authority (CA) has certified both the factory's web shop and the payment handler located in the datacentre through the yellow lines. The internet browsers installed on both the user's PC and Mac trusts the CA through the violet line and through this, they derive trust in the web shop and payment handler. These data transfers require that the DHCP service, DNS service and IP routers are operating as intended or else they will not be possible.

3.1. DHCP

With the widespread use of DHCP that automatically configures all types of units such as internet clock radios, smart phones, laptops, printers and desktop computers to access the internet, it is imperative that this function does not return false results, leading visitors into the wrong hands.

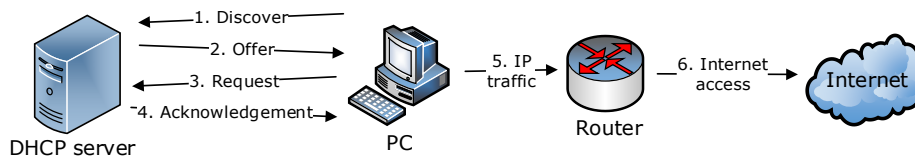


Figure 7: How DHCP works

In Figure 7, the PC does a broadcast onto its network interface in stage 1 and receives an offer from a neighbouring DHCP server in stage 2, containing IP address information. The PC accepts and returns a request for the offered IP address to the server in stage 3. The server acknowledges the request in stage 4 and returns a lease duration along with other requested configuration information. The PC now knows which IP address the router has and the PC can access the internet via stage 5 and 6. Often the router also acts as a DHCP server, so that a standalone server is not needed.

Stage 1 and 2 in Figure 7 are crucial in the sense that the PC does a broadcast onto the network it sits on and has no method to determine whether the info received from the server is truthful or not. If a malicious entity wanted to, they could insert their own DHCP server on the network and whichever server answered first in the discover stage, would control which network settings the PC will be operating with. This includes which “phone books” to look up in, also known as DNS.

3.2. DNS

Much like an ordinary phone book being used to look up names of people and get telephone numbers as a result, DNS is the easy way to connect to any given website. Because it is easier for a human to remember a name than a IPv4 number with between 4 and 12 digits, not to mention if it was to be translated into what the computers actually use, being a 32 digit number.

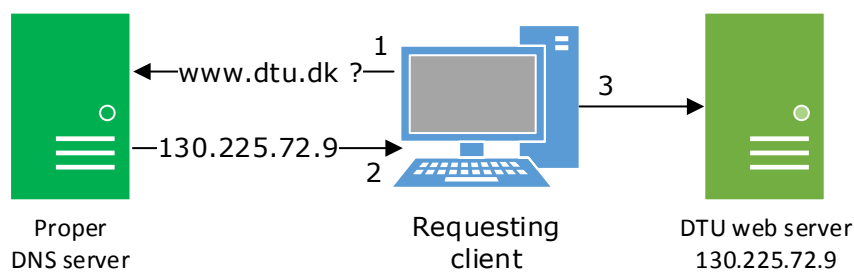


Figure 8: Good DNS

Additionally, domain names are more consistent than IP numbers are, meaning that you can own a website name that is not tied to a specific IP address. This makes switching between hosting providers easy, since a web address does not care if it lies at host A or B, as long as correct information is provided for its visitors. Figure 8 shows a properly working DNS request and answer session. There is seldom a system without errors though and DNS is not exempt from that either. Ensuring that the answers to requests both are up to date and not purposely falsified means almost everything to the everyday usage of internet services.

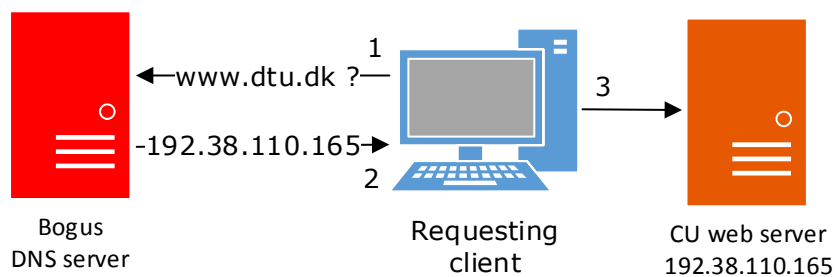


Figure 9: Bad DNS

Figure 9 shows a correct request but an incorrect answer, meaning that the visitor for the DTU website is led astray to the web server at the University of Copenhagen. The only thing to do about this is to either wait and see if the issue eventually fixes itself or try to use another DNS provider.

3.3. IP routing

Both DHCP and DNS are very important functions but they pale in comparison to what actually ties the internet together, being the internet protocol. IP is the standardised addressing scheme that every device has to make use of, in order to traverse from different peer points to other peer points.

IP is flatly structured, meaning there are no hierarchies in the sense that it is not in any way “easier” to reach a low number such as 1.2.3.4 than it is to reach one like 251.252.253.254. IP is also a service that does what it is told but not more than that, which is best explained in the phrase “I will do everything I can to deliver my payload but I make no guarantees for its arrival”. Therefore additional functions are needed as extensions to IP for data integrity checks, to reply whether data has been received or not and finally which local and remote port to “speak” to. This is carried out by the transmission control protocol (TCP) and user datagram protocol (UDP), but their finer details are not going to be explored here.

Addressing schemes over IP is handled using *routers*, which are simple but powerful computers located at branching points in networks. Less advanced and much cheaper routers are nowadays a common household item, no matter the type of chosen internet connection. Each router maintains a *routing table*, which it looks up in when it forwards traffic, called an *IP packet*, from one end and to another.

It is up to each router to keep knowledge of its adjacent routers, in order for it to pick the preferably shortest and least congested way to the destination, told by the packets it is currently handling. Sufficed to say, the individual router’s tables has to be as accurate as possible, so that packets are not being led the wrong way where they never reach their intended destination, ending up being discarded. Routers are the “I do not know who you are looking for, but I know someone else who can send you further along the right path” internet stewards.

I have already discussed how both DHCP and DNS work, but their places in the bigger picture come to greater justice when all three come together. Every arrow in Figure 10 means it is IP traffic.

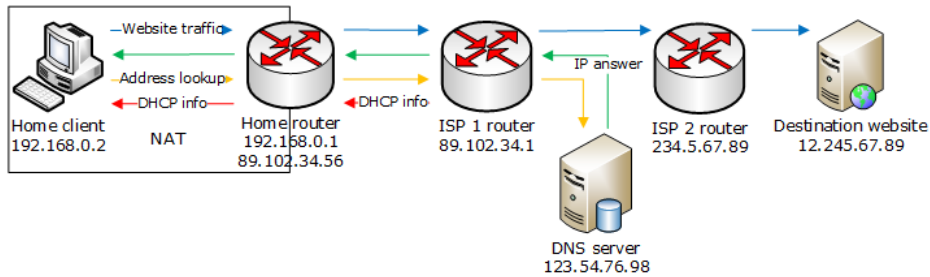


Figure 10: What typically happens behind the scene when visiting a website

In Figure 10, the ISP 1 router auto configures the customer's home router (HR) through DHCP. Now the HR knows whom to contact, when it does not know the requested destination itself and configures the home client (HC) through DHCP. The HC knows it has to make use of the HR to reach other computers not on its own network.

The HC wants to visit the website where it knows the address in letters but not IP number so it contacts the DNS server's IP, which is already known because of DHCP. The HR forwards the packets to ISP 1's router that knows the DNS server and forwards the request to it.

Assuming correct DNS lookup, a reply with the likewise correct IP number of the website is sent back to the HC. Finally, the HC can send a request to the requested website. First through the HR again, then through ISP 1's router, then through ISP 2's router that knows the website server and only then does it end up at its intended destination.

Figure 10 showed how properly functioning routers are taking care of traffic, so what is missing is to show what can happen when they are not.

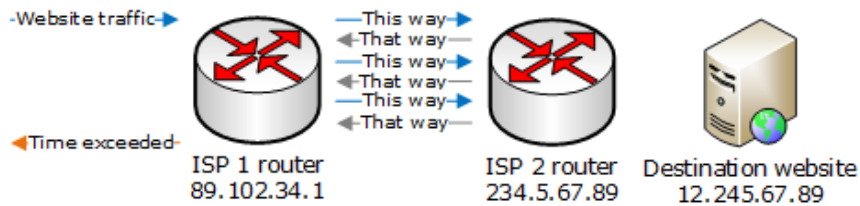


Figure 11: ISP 2's router is failing

In Figure 11, it should be assumed that everything right up until the website traffic begins is the same as in Figure 10. The difference compared to before, is that ISP 2's router believes that the website's address 12.245.67.89 lies past ISP 1's router and sends the traffic back, whereas ISP 1's router is determined it is past ISP 2's router and keeps sending it back that way again.

Although it means the webserver cannot be contacted and thus a potential loss of revenue for its owner, it would be even worse if the traffic loop would continue indefinitely and use up all resources in the router but luckily, that is not the case. IP has a built-in function called *time to live* (TTL) which determines how long every single packet may exist in the network. TTL is a value, which has a maximum of 255, is reduced by one in every router it passes through and is discarded when it reaches zero.

In Figure 11, the website traffic request arrives with a TTL value of seven so ISP 1's router discards it when it reaches zero and sends a reply back to the packet's originator, that the time has been exceeded.

3.4. ARP

Every network device has a physical address (PA) and that includes the various network interfaces on many devices, such as the antennae and network slots at the back and sides of laptops and stationary computers. This is needed in addition to IP because IP essentially is an end-to-end addressing scheme, where a packet knows from which address it originates and which address it wants to reach. On a small local network, the number of intermediate network devices is likely in the single digits so IP traffic between two adjacent computers might only pass

through one or two such devices. However if one were to communicate across the internet to, say, reach a website in Japan from somewhere in Denmark, the number of intermediate devices that the traffic has to pass through is much more likely in the double digits. Every network port that the IP traffic passes through along the way has its own unique number, so that while the IP packet knows its destination, it is being “hand-to-hand” carried from router to router by ARP.

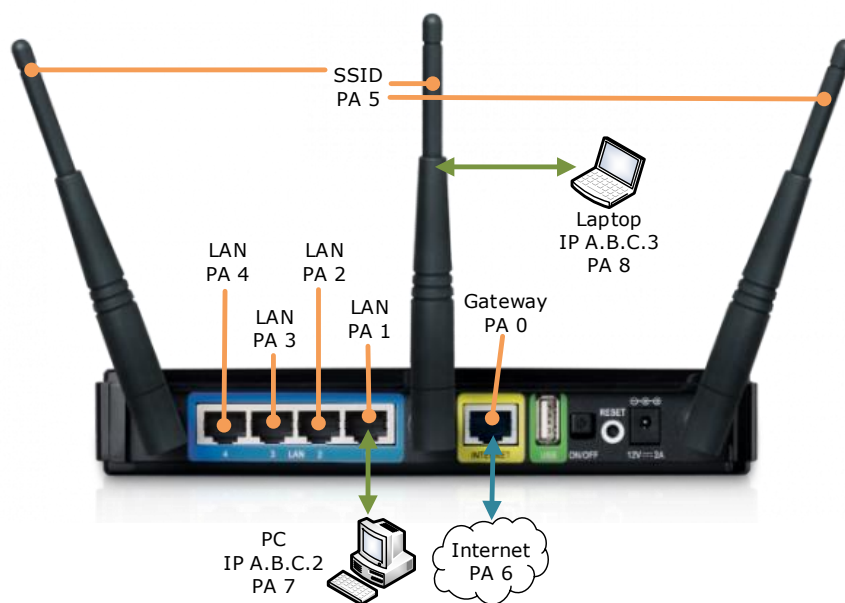


Figure 12: A home router with the various PAs

When the router in Figure 12 starts up it creates a list of its own unique PAs and when a device connects, it saves that particular device’s PA on its list, where it pairs it with its own corresponding PA so that it is now linked with the new device. It also binds the new device’s IP address to its PA with ARP and stores it in a cache, so the router know which PA to use in order to reach that exact IP address.

Example: The PC wants to exchange data with the laptop and by IP addressing it knows it wants to go from A.B.C.2 to A.B.C.3. The PC has stored its own PA 7 beforehand and knows it is connected to PA 1 on the router. It forwards the data to PA 7 that forwards it to PA 1, where the switching fabric in the router receives it and forwards it to the wireless PA 5 that finally forwards it to PA 8.

This essentially means that by IP addressing there is only one hop between the PC and the laptop but ARP wise, there are three. Once a link between two PAs has been established, a record of which adjacent PA to exchange data with in order to reach the same destination for every following batch of data headed the same way is kept for approximately five minutes. Even on a network as small as in the example, it is crucial that there are not two or more identical PAs since it would then result in traffic not going where it is supposed to go. With $16^{12} \approx 281$ trillion unique PAs and with them being distributed block wise to network manufacturers, it is fairly improbable that it should happen on its own, as it seldom happens that they are being reused.

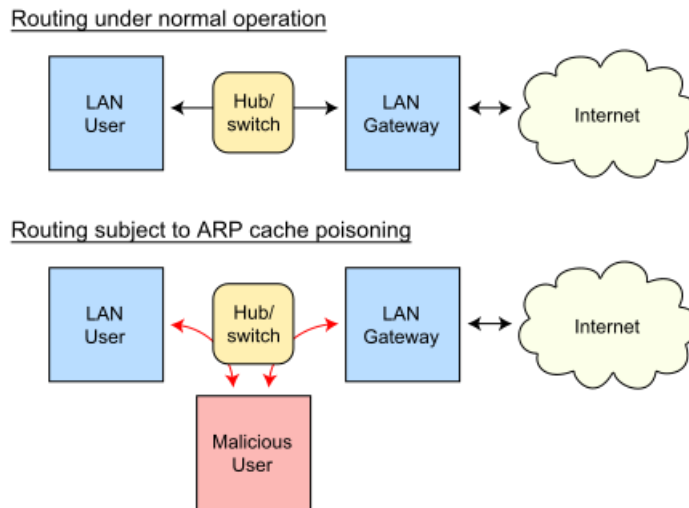


Figure 13: ARP spoofing/cache poisoning

The danger with ARP is to become a victim of spoofing where an attacker wants to intercept transmitted data. Here in Figure 13, the malicious user has successfully performed a man-in-the-middle attack by replying to ARP requests for both the LAN user and LAN gateway. This is possible due to ARP not in itself provides any protection against such attacks, although software does exist to detect and protect against it.

3.5. Summary on ill placed trust in the basic internet functions

Before there is DHCP and before there is DNS, there is IP routing and ARP. While it is possible not to make use of DHCP and manually configure one's own devices, it only takes a single mistyped number before there will be no connectivity. The same goes for DNS where it is also possible to navigate the internet without use of ordinary www addresses, but the amount of work associated with that is simply staggering. This is especially true when domain names are much easier to relocate onto different IP addresses than the other way around. Therefore, an IP-address used today might not point to the same place tomorrow if the domain has moved.

The points are, that there really is no way (or at least no easy way) around using the methods that are provided and keep the faith that the IP table and DNS administrators know what they are doing. Even when using them, there is no reason to have complete faith in them either. That is no problem for the ordinary user to abide by, since they are already completely unaware of the structures they rely on and are for the most part not required or interested in knowing about them either.

The problem arises when digital trust is being discussed and these topics are kept out of the loop, likely because it is assumed that they always work as intended. Perhaps due to the high amount of surveillance they continuously are under by their owners, the different internet providers. Nevertheless, they are still systems and systems do occasionally fail.

4. Reputation, trust and identity in physical vs. digital domains

The individual identity that people have and which makes them who they are is usually certified by the resident government in the form of a birth certificate, public health care statement, driver's license or a passport. These forms of proof are typically given a high amount of significance, because they are issued by institutions that in one way or another are products of the trust, which people in turn place in their governments. This makes them domestically and in some cases internationally valid for precisely determining the identity of their holder.

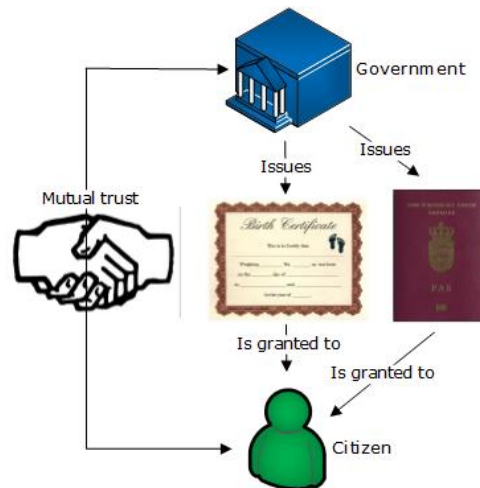


Figure 14: Physical trust and *receiving* an identity

Figure 14 depicts a government issuing a birth certificate and passport to one of its citizens. The governmental authorities place their trust and issue the physical evidences where the citizens trust the government to provide them with genuine identification.

Since the issuers can be both local and residential, the concept of a physical proof of identity leaving these institutions in a letter is not difficult to grasp for the average person. Even if someone does not trust or agree with their government's actions, possessing the monopoly on issuing proof of identity still makes government almost impossible to circumvent.

The state of affairs in the digital domain is very different from the physical. Perhaps most tellingly is there are no *people* authorities but only *system* authorities, where you do not trust a person but instead the software they are using. Moreover, unlike the physical world, where borders make up where one jurisdiction ends and another one begins, there are no effective borders on the internet. Save for a very few misguided couple of places such as North Korea and China, but at least it was not designed to be that way.

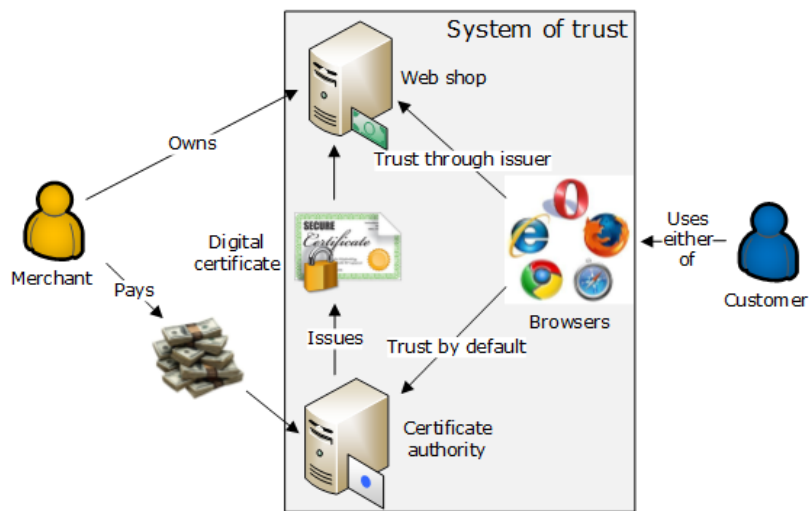


Figure 15: Digital trust and *buying* an identity

Figure 15 depicts the relationships between a merchant, a customer and the relevant systems in between when making a purchase. It is a further elaboration upon this aspect compared to Figure 6 that only took a top-down approach. A merchant has paid a random CA to issue a digital certificate to his web shop server and it is known in advance by all five browsers. By visiting the web shop and reading the certificate, the shop appears to be approved by the CA and is presented as a safe transaction to the customer.

When ordering an internet service from a provider, all they essentially do is to provide someone with an IP address for delivery and tracking purposes and the ability to receive and transmit bits over various forms for physical mediums. Essentially, the internet service providers (ISP) such as the traditional over telephone, cable and fibre, along with mobile 3G and LTE providers are called *bit carriers*. The products that are being sold is really just the capability of transmitting

and receiving the IP packets explained in chapter 3. Unlike the services provided from a physical government, the digital ones can come from all over the world, thus it is not tying anyone to operate in a national workspace.

While you cannot make use of a neighbouring country's ISPs unless they operate in the area in which you live (and under local national law), you can for the most part make use of the services they offer. The opposite scenario, where you for instance as a Dane want a Swedish passport without first having changed citizenship is not possible. This illustrates the distinction between nationalities on the physical plane, but not in the digital.

5. The audience that has the need of educated guidance

With the goal of supporting user choices regarding matters of browser security, it makes sense to determine both who they are and what their needs are. I base my project on experiences I got during a job I had between 2008 and 2009 while still being a student, where my task was to visit residents in Copenhagen on bicycle and solve computer related problems for ordinary people in their homes. The company was small, had only one other employee at the time of my own employment and at its peak there were about fourteen employed, as both driving supporters and accounts assistants.

5.1. Personal experiences about the common user

The most common misconception the company's customers had, was that a piece of antivirus or "internet security" software they had bought would always aid them directly or even take control of which websites they could visit and what they could and could not download. Often they had paid a larger amount of money for that software, only to find out that it still did not keep them from installing officious browser toolbars that originated from websites they had visited. It could also have come bundled with other software they had installed but not deselected during the installation, only going for the "Next" and "OK" buttons to speed up the process.



Figure 16: Reading and learning in advance is a show stopper for many

It certainly did not help the situation that a particular piece of software had often been recommended and sold to the client by the very company I worked at. Thus, it not only meant a false sense of security to the customers but also that they now had become the company's clients again and had to pay someone to come and undo what they had believed they were well protected against.

A turning point for one particular client came after my third visit with the same routine of stopping and deleting already running bogus programs, uninstalling various pieces of unneeded software and changing the browser start page back to

what it was before. The first advice I gave them was a rule of thumb: Always to click “No” instead of “Yes” when asked about something. I say rule of thumb because it is often very difficult for ordinary users to discern between websites wanting to install either updated software (because it requires knowledge about programs already actually installed on one’s computer) or harmful software.

The last advice I gave them was that the best means against unwanted software was sitting half a meter from the screen, meaning that a sceptical approach was the best defence they had available. After that, I did not hear from them again so I am letting myself believe that it had worked out well.

A common phrase is that “you do not need to be a mechanic to drive a car” and that is true, however with the evolution in the car industry, it should be “a mechanic and an electronics expert” since car computers have become such an integral part of modern motor industry. It goes to show, that even in an area that has been notorious for home-made solutions to problems where duct tape and cable ties have been the most prominent problem solvers, it has since become so advanced that there is often no way around an authorized service garage.

To the average users, a computer is a piece of electronic equipment that lets them go about their browsing, shopping, emailing, playing and social networking routines. Therefore, explanations of the lower layers of their functioning need not be common knowledge. Many also appear to be willing to pay to have some software take care of everything, and even if it cannot do it, the illusion remains to them.



Figure 17: Many are happy if they can leave all security decisions up to software

If users can be helped to not necessarily understand it but at least be made aware of potential pitfalls and then act accordingly, then I believe such a help will come a long way.

5.2. Users are not *stupid* but *unaware*

From my personal findings, it seems reasonably clear that ordinary computer users are not by definition stupid but merely lack knowledge to process the inputs properly that they are being presented with. They do not act against advice given to them but often openly welcome it, though they also have a hard time linking the same advice with similar situations. For instance, warning somebody against accepting installation of a browser extension or a bundled toolbar from a piece of software does not necessarily result in a natural wariness of opening email attachments.

On the same notion, it also became evident to me, that users with pre-installed antivirus software were less concerned about their online safety than those who knew that they did not already have it or had installed it by themselves. Often they were not even aware that it was already installed, as the programs rarely draw attention to themselves if there is nothing to report.

What I would like to highlight from that particular finding is that users, who have taken an active part in installing a piece of software with a certain function, are more aware of the hidden dangers that the software against which should be safeguarding them.

6. Usability and security seldom go hand in hand

One of the oldest conflicts between developers and users are restricting functions that are necessary on a security level but time consuming and seemingly superfluous for the user. It has typically been a choice between wanting to build easy to use software and then try to make it secure or the other way around where security comes first and usability second. Both approaches must therefore share an equal amount of attention in the design phase.

6.1. Choosing the right design for the right task and audience

Having established that design should take notice of both security and usability, the question of who dictates the design remains. Ka-Ping Yee, a PhD student in computer science from Berkeley, writes in an article from 2004 that all relevant parties are assumed to adhere to a mutually understood framework of acceptable behaviour. Yee's own example takes copying restrictions and pits music distributors and listeners on each side and designers in the middle, where the designers are faced with an impossible task if both distributors and listeners find each other's' claims unreasonable. Here, the source of the conflict is not usability related but stems from policies. [8]



Figure 18: It should not be the developer's assignment to sort out policies

In another article from 2005, Peter Gutmann and Ian Grigg, a researcher at the Department of Computer Science from the University of Auckland and a financial cryptographer respectively, write that the 1990s have been spent building and deploying security that was not needed by the average user and a decade later, nobody seems able to use it. [9]

Gutmann & Grigg are mainly critical of how the common denominator from security experts seems to be how awful it is when security seems added at the last minute and fail to recognise that the same thing has happened in reverse order when attempting to make use of secure functions in software. From their own findings with software that appeals to a large audience, it seems indicative that security comes second to usability in the sense that only when a given piece of software has gained a large enough audience by a good design rather than good security, only then is it time to improve its security measures. If good design attracts more customers than security features, it will also have gained a larger income incentive if its users are either willing to pay for services or if they can be served targeted advertising akin to Google, who by 2013Q3 have had an income of \$36.5 billion through advertising alone over a period of nine months. [10]

Their point is that it does not have to be a bad thing if software is designed to have the security added at a later stage, unlike the conventional approach, which is often a homebrewed combination of both aspects at once. What they mean is that good usability eventually pays for good security in the end.

Gutmann & Grigg also identify key processes in software development where the race to create the new Skype or YouTube allocates resources away from security, so it ends up with a term they call *layered*. Layered means building existing security upon an existing piece of software or the other way around, the same way that a security system is added to web browsing for money transactions or e-mail sent over a secure line. The trick to make it all happen seems to be by making sure to keep a familiar interface while having the changes taking place under the surface. Yee is in agreement with Gutmann & Grigg by underlining that neither security nor usability should be bolted to the other at the final stages and that the collaboration should be carried out in iteration. He posits that the two practices' conflict happen, when security restricts access to functions with undesirable results, where usability improves access to desirable functions.

There is also the problem of using security initiatives in a non-intrusive way. Pop-up boxes that express warnings are almost a certain way to teach users that security is obstructive and interrupts their usual workflow. It almost suggests utilising muscle memory to click a button to be rid of an obstruction rather than performing a thought out action. Yee draws up the following 10 guidelines for secure interactive design:

Guidelines for secure interaction design

These 10 design guidelines are based on the actor–ability framework. Readers might find them helpful in designing and evaluating user interfaces for secure systems.

General principles

- *Path of least resistance.* The most natural way to do a task should also be the safest.
- *Appropriate boundaries.* The interface should draw distinctions among objects and actions along boundaries that matter to the user.

Maintaining the actor–ability state

- *Explicit authorization.* A user's authority should only be granted to another actor through an explicit user action understood to imply granting.
- *Visibility.* The interface should let the user easily review any active authority relationships that could affect security decisions.

- *Revocability.* The interface should let the user easily revoke authority that the user has granted, whenever revocation is possible.
- *Expected ability.* The interface should not give the user the impression of having authority that the user does not actually have.

Communicating with the user

- *Trusted path.* The user's communication channel to any entity that manipulates authority on the user's behalf must be unspoofable and free of corruption.
- *Identifiability.* The interface should ensure that identical objects or actions appear identical and that distinct objects or actions appear different.
- *Expressiveness.* The interface should provide enough expressive power to let users easily express security policies that fit their goals.
- *Clarity.* The effect of any authority-manipulating user action should be clearly apparent to the user before the action takes effect.

Figure 19: Ka-Ping Yee's 10 guidelines for aligning usability and security

Yee finally concludes that practitioners of both usability and security have more in common than what appears to be obvious, despite the many obstacles both face on a daily basis.

6.2. The Windows UAC done wrong, an example from a workplace

Every user of the Windows operating systems since Vista has become acquainted with the pop-up dialogue box in Figure 22 asking for permission to run various programs and processes. It started as a great annoyance to many due to unfamiliarity with both the default restricted user environment and that some directories were deemed more critical than others. Microsoft calls it the user account control (UAC) and while it may not have had the envisioned security strengthening impact, it has at least done *something* to try to teach its users that some actions have consequences.

In a home environment the damage from bogus programs stays on a small scale with repercussions mainly limited to the residents, but apply the same unrestrained conduct to a larger scale environment and restrictions will generally have to come into place. There are various options in order to achieve this, most often by the use of a centrally managed system with a server maintaining a *domain*, so that inhibiting user rights becomes a very easy task to perform. The problem with this arises when the administrative presence and the software to be controlled do not follow the same evolution.

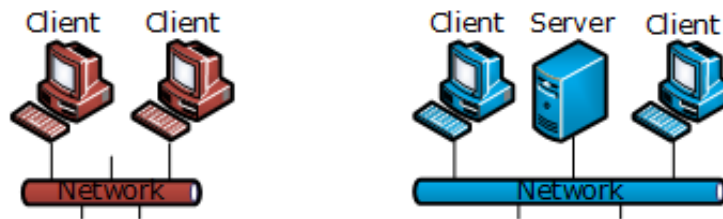


Figure 20: Unmanaged client/client and managed client/server setup

From the beginning of April 2010 until the end of March 2013, a friend of mine and I were responsible for everything IT-related at a school for children with special needs. Simultaneously I also held a job in what I call a “regular office environment” with everything it entails regarding the dos and don’ts in IT security. Already from this small presentation, it should seem to be two incomparable workplaces and that is indeed very much true, but at the same time, it puts the two aspects of usability and security usage in convenient black and white.

In order to take the edge off the upcoming comparison, I would also like to clarify the rather unorthodox working arrangements that were set up between the school and us. Despite the size of the school and all its satellite departments with around 150 employees and 250 students, my friend and I were only present on the largest site once or twice a week, because many administrative tasks could be handled with remote access. Each child had its own low-priced stationary computer and each class its own laptop for the teachers, so the management of all of them was not something that could be left up to carelessness. Upcoming computer repairs were emailed or put on a list to be taken care of during weekends and the day-to-day support was handled by the chief financial officer (CFO) to the best of her ability. Suffice to say this was far from an optimal setting but it kept working satisfying, until it eventually became too much work and they had to find a full-time employee solution.

To help set the tone, I have made a list of the security that is implemented in the office environment and it describes the extent to which the school set aside security in order to maintain usability. The left side of the table below shows how the IT department works at my other workplace and the right side shows the contrast of how the school chose their way of implementation.

Traditional security principles	Usability carried out by exceptions
The IT department provides on-site support within opening hours.	Regular visiting hours once per week and repairs during weekends.
Username and passwords for domain logins are strictly personal and dictate each user's corresponding rights.	Slipshod respect around user names, students were allowed to use a teacher's own login details or new students were told to use older student's details.
Wireless internet is provided "as-is" for telephones and tablets and is kept separate from company infrastructure due to increased chance of intrusion.	Wireless internet was seen as "business critical" since teachers refused to be tied down with a cable. Poor to no connectivity was often a direct result.
The CIO is inquired about desirable technical initiatives that make use of the internal network.	Surveillance and other initiatives were carried out by the school owners and repairperson, most often without knowledge of the network responsible.
Passwords are required changed every 30 to 60 days.	Passwords last infinitely for the students and they are printed out and put up on the walls in their booths.
"Bring your own device" (BYOD) for work purposes is not something employees are allowed to do, unless there is a plan for their uses and connectivity.	Several part-time employees brought their own PC or Mac and plugged them into the company network, devoid of security repercussions. Someone even brought their own wireless router that created an unprotected network upon the school's internal one.
The IT staff are the only ones with administrative privileges. Programs meant for office use adheres to restricted user environments.	The employees were allowed to install games, which are often forbidden in a restricted environment.

Table 2: Comparisons between good and bad practices of security and usability

It is important to make clear, that even though the conduct on the right side may seem like deliberately destructive behaviour, it could not be further from the truth. In its essence, it is simply a clash between highly specialised knowledge in two very different fields of work. The teachers and social educators are there to teach and help the students, who, by all means, deserve all the help and care they can

get. To them, the students' computers and their own laptops are simply tools for educational programs, games, and a means to print out schedules and invitations.

On that front, the overall need is simple but compared to the usual security/usability provided by a Windows domain network with a client/server setup, it requires some creative thinking about letting the usual options work in such a diverse environment.

Which career background the employees have in regards to complying with new work methods, such as having to use a username and password to log on any given computer, is very significant. Previously there had not been a need for that since every computer was not centrally managed and only very, very few even knew what a computer domain was or what it would look like once implemented.

This also meant that the employees had to be educated in its use, in order to use and pass that knowledge on when working with their students. Sukamol Srikwan and Markus Jakobsson put it the following way [11]:

A problem that exasperates the effort educating users of security is that is not sufficient to explain the problems to the target audience, but one must also change their behaviour. It is often ignored that there is a tremendous discrepancy between what typical users know and what they practice.

Even though their work is used to teach users about security aspects, it is still addressing the need of changing behaviours of those they want to help.

In Figure 21 below, the first vertical flowchart column "Home" shows how the process of installing new software in a Windows environment taking place at a home location. The default setting here is to give every computer user restricted access, meaning a dialogue box will show up every time a program requests access to areas where the current credentials are not sufficient. Such as the directory where Windows files and programs are located, because they are deemed crucial for the operating system. In reality, this is a situation where usability outweighs security since all that is needed is to select "Yes" instead of "No".

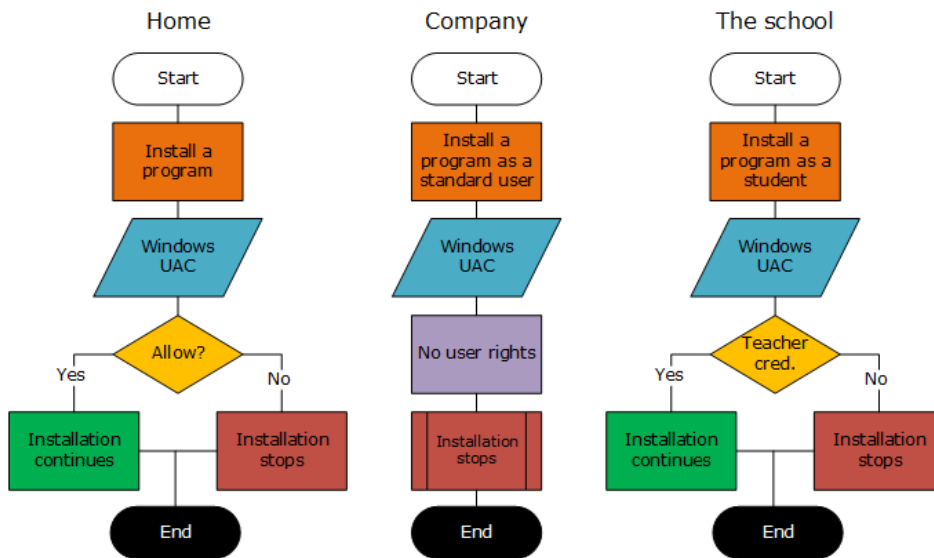


Figure 21: Three different showcases of usability vs. security in Windows

Even though the “No” button is highlighted as default in Figure 22, there is no incentive or advice against selecting “yes”. On top of that, it often shows itself based on actions already taken by the user, meaning that they have to confirm the action they wanted to carry out in the first place.

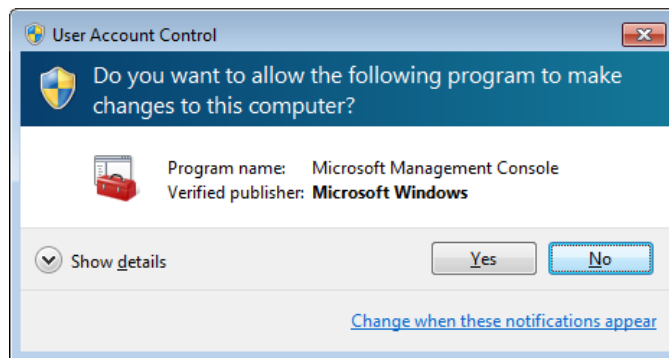


Figure 22: Win 7 UAC “home”

The only real sense of security surfaces when programs attempt to access restricted areas on their own, so the user is made aware of underlying changes to restricted areas. This would be all right, if not clicking “Yes” meant that most likely the dialogue box will not be bothersome again and the program is allowed to do as it pleases, even if its real intent was not one a kind one.



Figure 23: Win 7 UAC "company"

The second column in Figure 21 named “Company” shows the situation where security outweighs usability in the sense that nobody with ordinary user permissions are allowed to make changes to system directories. Effectively meaning, that only programs the IT department have sanctioned and very often pre-installed are made available.

On top of that, it can come with a feeling of dread to ask for the permission to get something unsanctioned installed, when it removes an employee from his or hers other doings and there has to be provided a good reason to require something not already on the list of available software. In particular if it is not something important for carrying out one’s normal daily operations.

The last column “The school” shows the decision process at the school. Here neither security nor usability outweigh one another but keep close to a somewhat standstill. This should be seen in the light that for the students, it functions like the company, where they are not allowed to install their own software. To the teachers it behaves as they are used to, provided they supply their own usernames and passwords.

In theory, this approach had the potential to work out just fine but the reality was that it did not, cf. Table 2. There were mainly two reasons for it not working out, being that the teachers did not have enough knowledge about the myriad of programs available to discern between what belonged on them and what did not. On top of that, even though each student could use whichever computer he or she wanted to, everyone mostly used the same that eventually helped him or her attain some kind of ownership, leading to installations of programs being used at home.

The other problem that surfaced was that there was not enough time for my friend and me to help the teachers with these questions, given that we were rarely present during teaching hours due to the nature of the special and often chaotic teaching environment.

In the end, the standard Windows model of usability and security that restricts groups of computer users from installing programs not meant for business or educational purposes, proved to be somewhat unsuccessful in this kind of environment. The students found out that they had “the people with passwords” always available and the eldest ones could easily trick them into giving permission to install all kinds of things they should not have. On the other hand, the teachers also quickly assumed the role of how they knew it worked on their private computers, that their own passwords were often the quickest way to solve an issue, instead of finding out whether it was something the student actually needed or not. It ultimately proves what Srikwan and Jakobsson say is true, that you need to change people’s behaviour and practices.

It was not all for naught, though, as my friend and I did experience a significant drop in the number of repairs needed. Previously a student’s computer had to be reinstalled every two months but after the restricted user environments came into place, it was either much less or not at all. A new student would just have to log on with his own username and password and a new, clean profile would be

created, ready to be used. Although it did not have major influence it was envisioned to, the Windows UAC did save us for a large amount of tedious and time-consuming work.

6.3. Designing a trade-off between usability and security

During all of chapter six, it has been established that there are strong opinions of both usability and security but seldom in correlation. Several authors of scientific articles advise against favouring one over the other but instead work them both into the design process as early as possible and to do it iteratively. An example of a design where both have not been upheld is, despite its good intentions, Windows' UAC pop-up box. It seems to apply the opposite of what Ka-Ping Yee suggests and ends up trivialising security in a way that clicking "Yes" has become the easiest and least obstructive choice, even if the actions it has a possibility to entail are not favourable.

Braz, Seffah and M'Raihi list the following for providing both aspects regarding multifunction teller machines but they can be applied universally [12]:

- 1) It is important to make sure that the users understand what they should do well enough to avoid making potentially high risk mistakes; this is especially important for security mechanisms, since if a secret is left unprotected, even for a moment, there is no way to ensure it has not been compromised.
- 2) Security is a secondary goal for many users, a necessary step on the way of achieving their primary goals such as checking their account balance or their email; therefore, developers should not optimistically assume that users are motivated to read manuals or look for security rules and regulations.
- 3) Security concepts might seem self-evident to the security developer but are very unintuitive to many users; developers therefore need to put extra effort into understanding the users' mental models and be sure to use concepts the users are familiar with.
- 4) Feedback is often used to prevent users from making mistakes, but this is difficult in security since the state of a security mechanism is often very complex and hard to explain to the user.

- 5) Security is only as strong as its weakest component. Therefore, users need guidance to attend all security aspects in their usage of the mechanism. [13]
- 6) Attackers can abuse a system that it is “too” usable but not very secure.

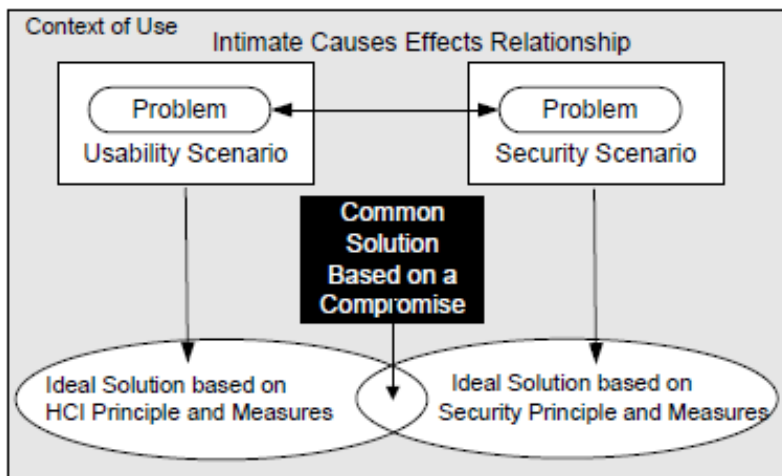


Figure 24: Braz, Seffah and M'Raihi's model of a compromise solution

The six listed statements above and Figure 19 support the findings in chapter 5.1 and 6.2.

To summarise, there does not appear to be an easy way about implementing usability and security in such a way that both cater to a person's needs. It is dedicated work from the beginning and, as Figure 24 depicts, a compromise must necessarily be struck to achieve the goals. If not, the mistaken “creativity” and carelessness from users are at times a frightening marvel to behold.

7. The banks are only very rarely safety nets for online transactions

My thesis supervisor and I had a brief discussion regarding the enormous success that online trading has seen the past decade. When it came to the subject of *why*, it was assumed that banks and their ability to recall payments has played a major role in providing security for their clients if anything bad should happen. If this had turned out to be true, it could very well have influenced this project, given that it would be safer for consumers with a credit card in their hands than meeting in person with someone and pay in cash. Spurred on by this, I set up a meeting with my own bank adviser to find out whether it is true or not. It turns out that the banks have much fewer tools to operate with in case of fraud than what we had thought.

The banks operate with two different schemes, one being account-to-account transactions and the other being credit cards. To the average consumer there seems to be little difference between them since they both include transferring money from account X to account Y but legally they are slightly different.

7.1. Account to account transactions



Figure 25: Account to account payment

Figure 25 depicts Client A using its own bank's internet service to transfer an amount of money to the foreign bank account of Client B.

- When performing a transfer between accounts, there is an option to cancel it before 5 o'clock in the morning where the transfers usually take place.
- The bank advisers have only the exact same tools at their disposal.
- When the money has been transferred, the chain of events has the banks talk to each other on behalf of their clients. If client B agrees that the money arrived in error and there is enough in the account, then the money can be transferred back directly.

- However if the money has been spent and there is no more left in the account, there is nothing to do about it.
- An important exception to this is if the money comes from the public sector, since they possess various methods to limit income in other ways. They will as good as always get what is theirs.
- Should any complications arise, it becomes a matter for the police. If it ends up as a case between civilians, the police will not do anything
- The banks in Denmark have good working relationships in these cases but that is also true with foreign banks. In this regard, there is little to no difference between sending money to someone of the same nationality and to someone with another.
- In the end, one will have to be rather unlucky to end up with a bank at the other end that is unwilling to cooperate.

7.2. Credit card payments

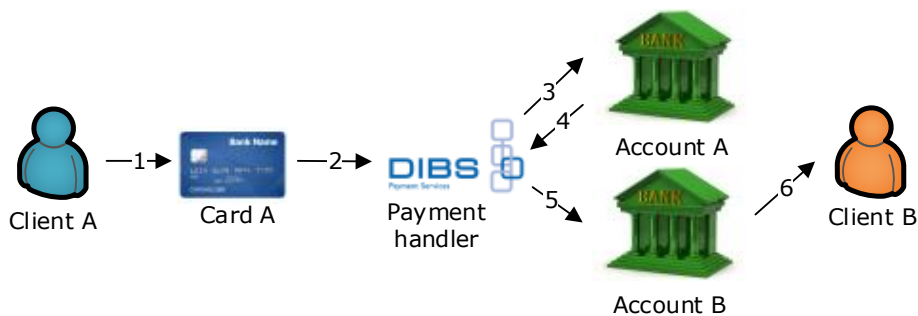


Figure 26: Credit card payment

Figure 26 depicts how Client A uses his credit card in an internet store. When the order has been placed, the store uses a payment handler, DIBS in this case. On behalf of Client A, the money is withdrawn from Client A's bank account and deposited into Client B's bank account, through the payment handler.

- Credit card usage is a matter between two individuals and therefore the banks are placing themselves outside influence.
- When the button to accept the payment has been pressed, then nothing can be done to stop it. Not even calling the bank and telling them to halt the payment is an option.

- If you are somehow fooled when using your credit card, then it is usually tough luck.
- In Denmark, it is only the administrator of the national credit card Dankort, Nets, that is able to halt or stall an on-going transaction. This mostly happens if it is deemed too suspicious or falls within other automatic categories of interest.
- Often Nets will cover the money lost, in cases where a customer has not received ordered goods but the store claims to have shipped it. Usually the burden of proof rests with the store and if it cannot prove it, the customers are getting their money transferred back.
- The most typical scenario, according to my own bank, is that customers have not read the terms and conditions of sale, meaning that there is no legal action to take against the shop. As long as it keeps within the Danish sale of goods act.
- My bank adviser has eleven years of experience and according to him; they record a negligible amount of cases where theft of identity is the primary reason for loss of money.

7.3. How the banks want to provide better online safety

The banking sector has a reputation within the networking industry of being very conservative regarding the adaptation of new initiatives in their structures, both physically and electronically. That is most likely because they cannot afford the risk of gambling with the systems or transaction schemes in which the digital money is stored and moved. However, even though they are not keen on being first movers, they are not entirely late adapters either. There already exists an additional layer of security around paying with a credit card, if the storeowners subscribe to the added layer of security. This is named the “Verified by VISA” or “MasterCard Secure Code”.



Figure 27: MCSC and VBV logos

What it means to the end user, is the compulsory addition of selecting a self-made password, which is used to affirm ownership of a given credit card.

The Danish banks are fortunately no slouches when it comes to adding new security measures, spurred on by the penetration of the national “NemID” system usage by the Danish population.

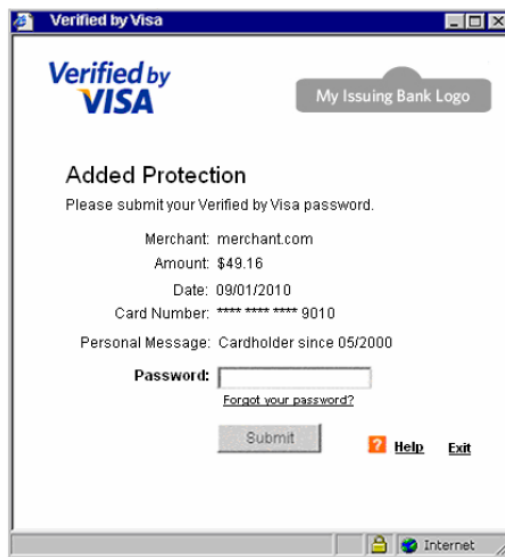


Figure 28: VBV pop-up box

As of November 1st 2013, all payments that would require the entry of a password are obsolete and instead an SMS with a randomly generated password for a one-time usage will be sent.

This requires the customer to have signed up to the new initiative with NemID as the proof of identity and to be in possession of a mobile telephone or other devices that are capable of receiving SMS. So in the case of either no signal coverage or if the SMS is somehow not received, then there can be no further transactions. [14, 15]

This setup assumes the following:

- All mobile phones are impossible to take control of by an attacker
- SMS sent to a phone passes through the mobile cellular network, independently of the internet.
- A user is able to transfer information contained within the SMS manually and without error from a mobile device to a client computer.
- Verifying the correct information transfer from phone to client allows the bank to assume genuine intent of transaction submission.
- It is difficult for an attacker to steal someone's mobile phone.

The above assumptions come from Audun Jøsang, a Norwegian professor from the Department of Informatics at the University of Oslo. In 2007, he was the main author behind an article called "Security usability principles for vulnerability analysis and risk assessment" that also took transaction authorization with SMS into consideration through a security perspective. [16]

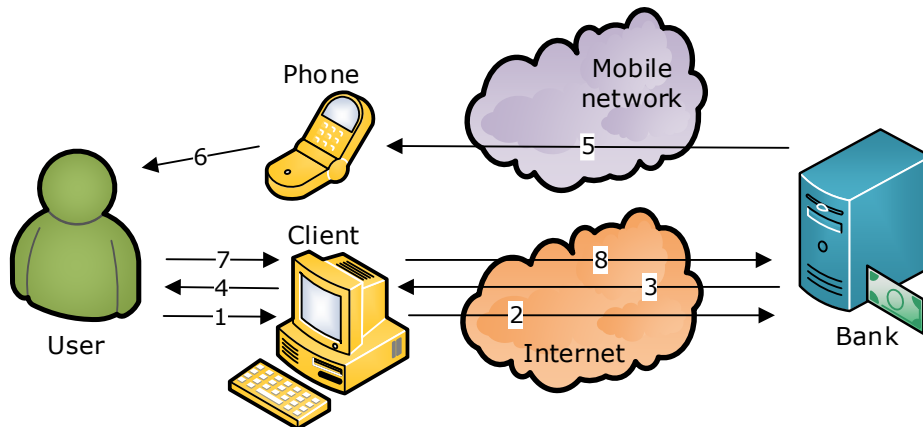


Figure 29: General transactions in 2007

Figure 29 depicts how Jøsang in 2007 and now the banks in 2013 have envisioned the increased security initiatives. The user uses his client computer to access the bank's home page through the internet (1&2). The bank responds and presents the user with a login page through the computer that is also transmitted over the internet (3&4). At the same time, it sends an SMS over the independent mobile network to the user (5&6). The user processes the information from (4&6) and returns the collective transaction information through (7&8).

The added security approach happens by utilising the separate client and phone along with the internet and mobile network, as it is assumed difficult for an attacker to be present at once on both channels. However, it still leaves the bank's system as the single point of failure.

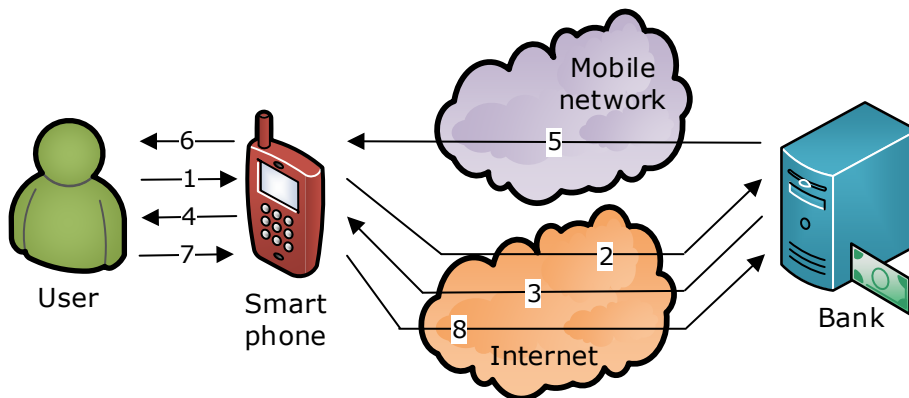


Figure 30: Transactions possible in 2014

Figure 30 is the depiction of the same type of functionality as in Figure 29. The main difference is that there are no longer a separate client and phone; they are both integrated in the smart phone. The user still connects to the bank through (1&2) but get two replies to the same device (3&5), processes them (4&6) and carries through with the login information (7&8). In the 2007 scenario, the bank's system was the single point of failure but now the phone has become one as well. If it is lost or stolen, an attacker will also be able to initiate the connection with the bank and receive the SMS.

Jøsang also argues in favour of a system that incorporates the recipient in the received SMS so that it will not just be a random number. However, studies from 2008 show that 21% of a group of participants failed to notice that the recipient's account number had been modified during transit. [17]

Another important factor to consider is that the phone in Figure 30 turns out to have become a single point of failure, meaning that it facilitates both the client and receiver of passwords. Since it makes very little sense to carry around two mobile phones where one is only to be used as an internet client and the other for password purposes, there is no longer a practical distinction between the two

systems. One could argue here that the steps taken would have made sense a decade ago but that it no longer provides the type of security envisioned. At the same time, the two carrier systems of a mobile network and the internet are not separate and independent of each other. This was the overall strength of the system in the 2007 model but in 2014 where phones and computers are integrated, this is no longer the case.

Regretfully it only strengthens the notion that banks are far from proactive in their efforts to protect their customers, when using a technology that became outdated when smartphones become common household items.

7.4. Summary on which role the banks are playing

Based on an interview with my bank adviser, it has become clear that the banks have not been playing a major part in establishing monetary security regarding online transaction they were thought to be. In fact, it appears as though they always have a way to acquit themselves if they can. In the wake of the financial crisis of 2008, though, it does not come as a big surprise.

What made a larger impression was that banks put great effort into determining the trustworthiness of their clients, in particular whether someone appears to have made sound judgements while using their credit cards, especially online. They view these cases as if they were doing someone a favour and if they get the slightest hint of irresponsibility, they are prone to leave them hanging.

One reason they do not see these cases very often is, according to my adviser's own personal opinion, that the Danish population has a comparatively high level of education, making them better at detecting frauds.

8. The digital certificates and padlocks

A digital certificate is best described as a type of electronic identification of either a person, an organisation, a piece of software or actually anything that has a unique identity. It contains various types of information including whom its issuer is, who or what it identifies and at the same time providing a public encryption key from the public key infrastructure (PKI). It also contains a serial number, expiration date and the likewise digital signature of the issuer of the certificate authority (CA) but the top hierarchal CA keeps and maintains what is called a *root* certificate. Many certificates conform to a standard developed by the international telecommunications union (ITU) called X.509.

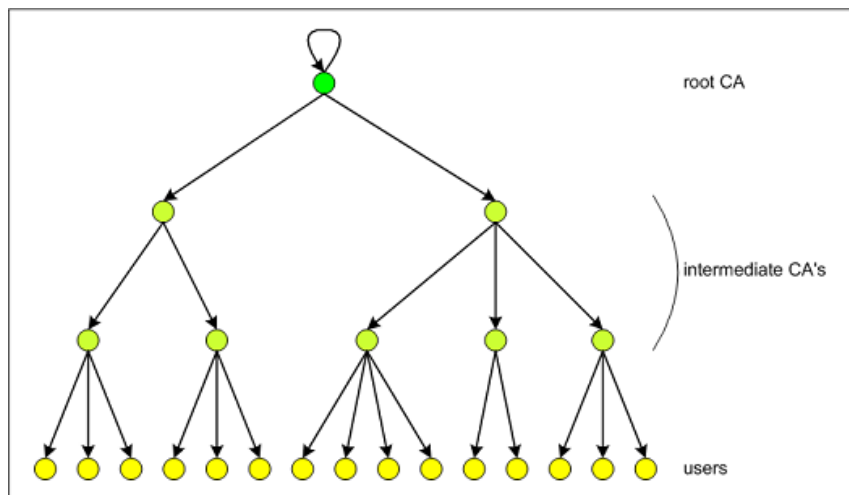


Figure 31: X.509 hierarchy structure [18]

Figure 31 shows the root CA creating a *self-signed* certificate, which is then used as reference point by the intermediate CAs meaning that they are trusted by the issuer of the root CA. The arrows between the intermediate CAs and users are only the depiction of the direction of trust, not that users are trusted by any CA. As such, when the various browsers trust the root CA, they also trust the corresponding intermediate CAs, which are trusted by the root CA. A root CA authorises the intermediates and they issue the certificates.

How the trust hierarchy is presented in a browser is shown in the next figure, along with the location of public key information etc.:

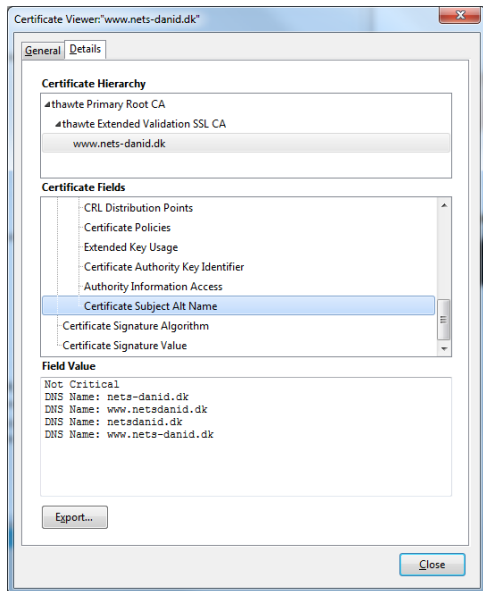


Figure 32: Certificate information in Firefox 26

Figure 32 depicts how Firefox 26.0 presents a digital certificate to the viewer. Here it is from the website of Nets-DanID, the company tasked with administrating the national NemID solution. In the “certificate hierarchy” box, it can be seen that the company “thawte” is the root CA of nets-danid.dk, meaning that only by trusting thawte’s root certificate is **nets-danid.dk** also trusted. Practically speaking, only by trusting a third party South African company is the process of obtaining the only means of interacting with Danish banks and tax authority possible. The “field value” box shows the DNS names, another issue of trusted systems.

A major issue with the certificates is that while the encryption it provides between two computers can be the best that money can buy, it makes no guarantees that the recipient company is not fraudulent in any way and that they will live up to their end of the bargain. The CA from which a certificate is bought is not sending a physical delegate to check if those behind a request have only good intentions in mind, as that operation would be immeasurable in scale and cost. This means that when somebody visits a website with a certificate installed, blind trust is put in the third party CA that is the entity who approved of the website being visited.

Suffice to say, this practice is fallible as even the CAs are not without their own share of problems. In September 2011, a Dutch root CA named DigiNotar discovered that it had had a breach in its security systems, which meant that outsiders had exploited the system that can issue certificates. The outsiders had therefore utilised the option to issue certificates to both any kind of website and to change properties of already installed certificates that had DigiNotar as their

root CA. They subsequently attempted to lure hijacked DNS visitors into false websites of Microsoft and Google, which then appeared to be trusted by a CA.

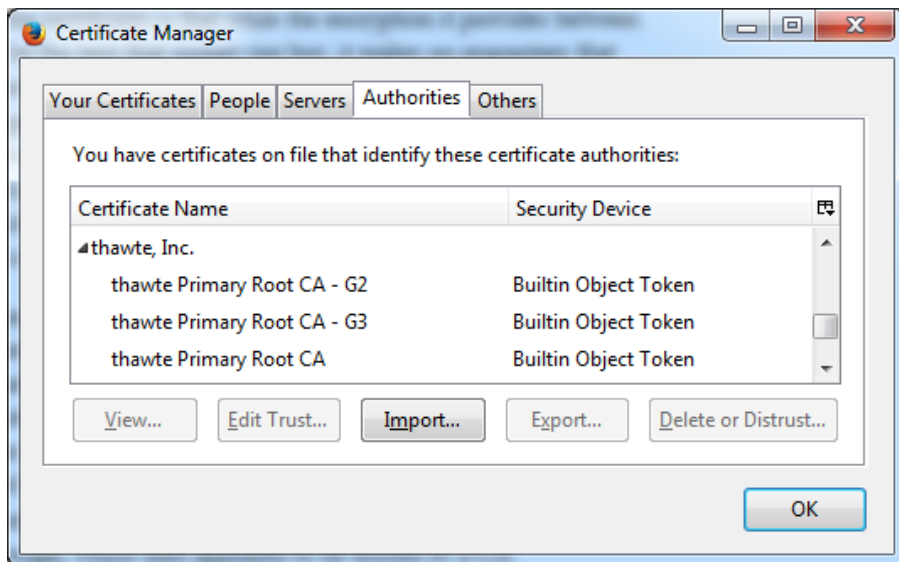


Figure 33: Pre-trusted CAs in Firefox 26.0, shown here the ones used in Figure 32

Another major point of interest is how browsers are “born” with a trust of root CAs and a few normal CAs, since it by no means is an indication of consideration but simply for easiness’ sake. Not including them would mean that someone visiting a website making use of a certificate would be told that their browser does not recognise the issuer and that they are recommended against proceeding any further.

The example with the Nets-DanID website used for acquisition of the NemID cardboard keypad will ask users to determine whether or not to trust something called “thawte, Inc” in order to proceed. While it really should be compared with choosing to proceed with driving a car if warned that the brakes and airbags are not functioning, many will undoubtedly select yes to trust it, as it usually lets them go about their business as usual.

It is again a case of believing that whatever seems to be wrong is the computer systems’ own fault and since it often turns out to be working anyway, the dangers and causes of it are dismissed too quickly.

8.1. How the digital trust schemes manifest themselves

A certificate contains a public key that when it is being read by a browser provides the first step in encrypting the data transmission. A browser reads the public key and encrypts the next messages and the server decrypts them back again, as shown here on Figure 34:

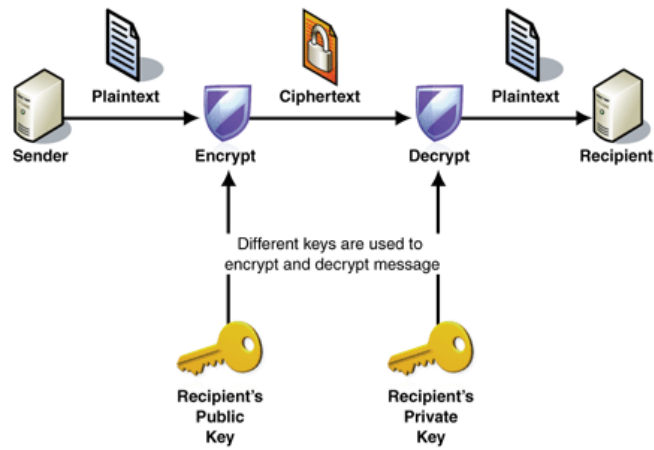


Figure 34: Public key index scheme [19]

The setup for browsers is very much alike, although never displayed directly to the users because of a no-need-no-tell reasoning, as seen here in Figure 35:



Figure 35: SSL setup between browser and server software [20]

However, a major issue arises when these intricate workings are displayed in a sub-optimal way. The display should not necessarily scare users but neither should it make light of informing people about potential dangers when becoming internet users.



Figure 36: An early IE 6 warning message

One of the very first examples of this was Internet Explorer 6's warning in Figure 36 that when submitting information to the internet, it might be readable for third parties. It is a very good example of bad design since its default proposed action is to not show the message again and highlight the "Yes" button. Granted, removing the check mark and/or pressing "No" either means that the box will show up again or the person using the browser will not be able to browse the internet.

There is no reason to explain the user that some information will always be sent due to the way that web traffic works, but the warning that was the first thing so many people met when internet access had begun to become common could have been improved in many ways. For instance, an in-browser animation that introduced first time users to using the internet and elaborated on the meaning of "sending information to the internet", which is not explained very thoroughly could have popped up. Reading it as it stands, it can mean filling out a form with one's personal details and pressing a button to submit it, but it also means typing in a simple web address and clicking a hyperlink. Personally, I believe that the vast majority only thought about the first example, leaving them unaware that it actually covers much, much more.

The same can easily be said about the awareness of certificate fraud where first time users were and still are not introduced to what a certificate actually means. The widespread way of showing them is in the first place not very user friendly since there seems to be an understanding that the word "certificate" signals that a digital certificate has to be presented as a physical copy would for users. The issue is that you still need to know they exist in order to find them and once that is done, knowing how to properly read and process their information becomes another obstacle.

Audun Jøsang describes how in March 2007, the Hawaii Federal Credit Union was the target of a phishing operation where the attacker had gone to great lengths to fool the targets. The Union’s website is www.hawaiiifcu.com but it applied to neither of the certificates in both Figure 37 and Figure 38 where the latter actually appears the more trustworthy of the two, due to its resemblance to the true address [16].

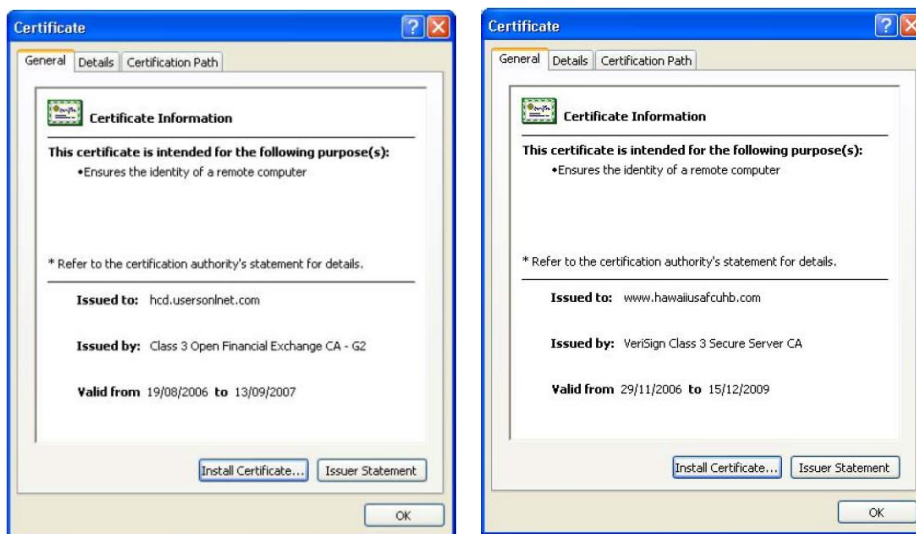


Figure 37: The bank’s certificate from 2007 Figure 38: Certificate for phishing from 2007

None of the genuine certificate’s information bears any kind of resemblance what so ever to belonging to a credit union located on Hawaii so if one was to investigate whether or not there was any fraud involved, it would not be obvious if there was. Clicking the “Issuer Statement” button at the bottom right of both information windows leads the user to a document consisting of 2,666 words but the required work involved with deciphering it would become intolerable and might as well have been obsolete.

The fraudulent scheme was orchestrated as a fake login page that imitated the original so that its users would submit their usual login information. In addition to the server hosting the fraudulent website, there was a number of other systems involved in making the fraud trustworthy:

- ✓ The domain name **hawaiiusafcuhb.com** had been bought legally from any one of the thousands of available domain name resellers but used for phishing, where after it was propagated between DNS servers for IP address translation.
- ✓ A server hosted on an IP address anywhere in the world, belonging to either the wirepullers' own internet provider or where the server was physically placed.
- ✓ A legally bought certificate from a company called VeriSign, which is also a proprietor of its own root certificate.

The systems themselves are not to blame but they are simply much too easy to take advantage of for illegitimate purposes. DNS entries were once in one single text file with web addresses and their corresponding IP numbers were located on a single web server, but due to future network load and security reasons it had to be decentralised. Instead, DNS servers now send and receive updates from each other automatically and reversing that process would be near impossible today. One always has the option to go the .dk/.com/etc. administrator and report a misuse of the name and it will likely be suspended. However since it is easy to register a new domain name that also sounds like the target in question, it is an ever-ongoing battle where the cost associated with obtaining a certificate is the largest hindrance, depending on how profitable the phishing scheme has already been. A one-year single-server certificate from thawte costs \$200 so there is a high probability that an amount as small as that has quickly paid for itself when setting up a new phishing scheme.

An IP address is very often obtained on a lease through an internet (= IP) service provider and rarely bought directly from the few institutions that have the authorisation to distribute them. The IP addresses within a small physical area can therefore differ by a large margin, even if the path to reach them stays fairly the same. In that sense, there is always an accurate index over who owns which IP addresses, but not for what they are being used. The owner *can* be found as IP addresses are not subject to the same amount of secrecy as domain names can be, although much still depends on their willingness to cooperate regarding pursuing the illegitimate actions, in particular if it happens to be in a place that are not keen on setting resources aside for it.

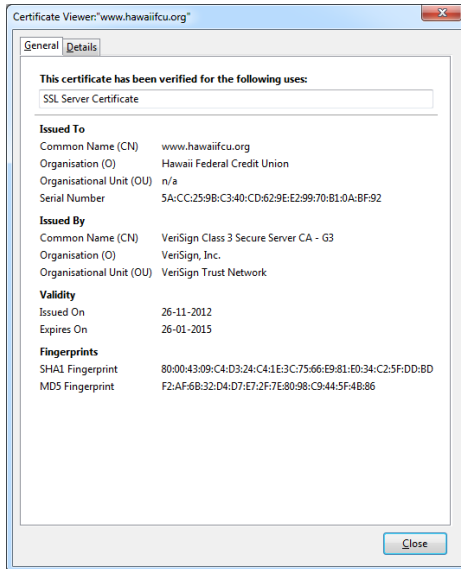


Figure 39: The bank's certificate from 2013

Finally there are the certificate authorities themselves who appear all too eager to sell customers their services, with focus on encryption strength, which governmental entity has authorised the various mechanisms and “the safety in having their company’s name appear side by side with the customer’s web address”, an attempt to relate physical and digital trust. Nothing can be found on either of VeriSign and thawte’s websites about what happens when one of their products is used for bogus purposes, which is not at all astonishing from a business point of view. Exposing

misuse or other fraudulent ways to exploit one’s product on the front pages is a bad marketing strategy.

It still does not remove focus from the fact that the danger is still present and real but appears to lack the attention it deserves from the regulatory institutions, namely the US’s national institute of standards and technology (NIST). In July 2012, the NIST released a paper that addresses how to prepare and respond to breaches in certification authorities and it lists four different schemes covering theft and impersonations but nothing regarding trusting an issuer who has a number of bad apples in its basket [21].

It almost seems like the CAs can do no wrong regarding whom they sell their certificates to and that past mistakes are conveniently forgotten when doing business enters the picture, Figure 39 is a prime example of that. It appears that when the Hawaii Federal Credit Union certificate’s validity stopped, they had either to renew it or choose another issuer. They appear to have chosen the last option. The choice has fallen upon VeriSign, the very same company who signed the certificate used by the fraudulent website in Figure 38, not a track record to be proud of but one must assume they instead provided the most value for money.

8.2. Extended validation

VeriSign and thawte both refer to extended validation (EV) as getting the “green address bar” in their product portfolio (which they just as well might since VeriSign acquired thawte in the year 2000). It is not wrong, as Figure 40 shows, but there is unsurprisingly more to the name than just that, even if the green address bar only applies one hundred percent to the users of Internet Explorer.

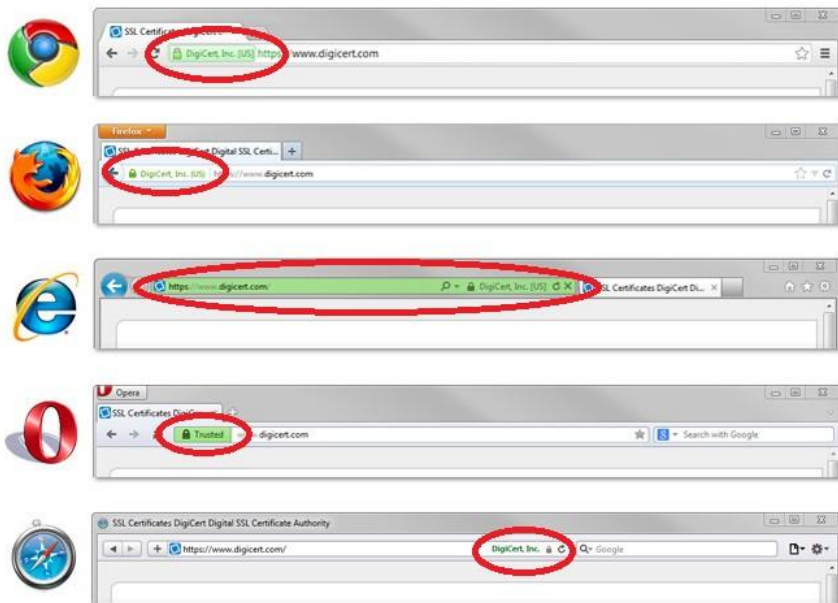


Figure 40: Padlocks and EV in Chrome, Firefox, Internet Explorer, Opera and Safari

EV is an initiative from a group of CAs in a common forum where the work started in June 2007 during which it was quickly adapted and finished by April 2008. It is still a part of the X.509 standard and requires a number of criteria to be fulfilled before it can be issued:

- Legal identity along with operational and physical presence of the website owner must be established.
- The applicant must either be the domain name owner or have exclusive rights over it.

- Legal documents for the certificate purchase are signed by an authorised officer and confirmation of identity and authority of acting owners of the website.

Once that is complete and the certificate has been issued, the browsers also have to support its usage. All of the five major browser brands have been supporting it for a long time, by changing the user interface around the actual website content (also referred to as the “browser chrome”) which includes changing the colour of the address bar. The remaining question is of course, does it actually help?

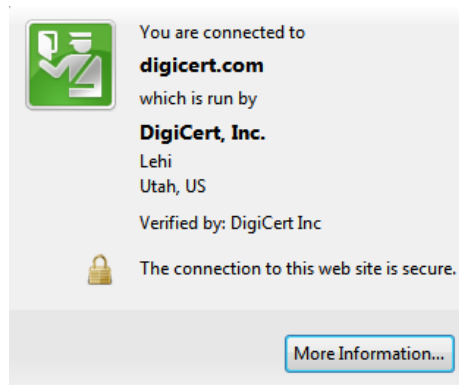


Figure 41: EV information in Firefox 26.0

In two studies from 2008 and 2009, Jennifer Sobey and Robert Biddle are asking the same question and perform a number of experiments with a group of users, in order to find whether EV has the desired effect or not. In the first article from 2008, they make use of eye tracking software to determine how their test subjects respond to changes in the browser chrome areas, such as the green address bar. Here they reach the conclusion that it does not help to add the extended validation to a certificate because users are not noticing the colour change. Instead, the users look for confidentiality statements on the web page itself, which is much easier to falsify, and that to make a real difference it requires better techniques for grabbing a user’s attention. This could include having the browser itself point to the new initiatives or through a pop-up window. [23]

The second article from 2009 asks the relevant question on browsers’ EV SSL interface usability, whether developers have thought through who their target users are, or if the users do not have proper information or background to perform weighted actions. They list unfamiliar technical terms, lengthy messages and misleading or confusing wording as the main causes for the various user interface failures to inform properly about what is taking place. This covers words such as “encryption”, “certificate” and “security” where they are far from being unequivocal for a user as the developers apparently tend to think. Sobey and

Biddle therefore suggest splitting up indicators for identity and confidentiality, as they are already separate concepts along with scrapping ambiguous terms such as “secure” and “certification authority”. Their solution is to replace the dialogue box in Figure 41 with their own design, as shown on Figure 42.



Figure 42: Sobey and Biddle's EV SSL certificate information

The three blue dots represent a score of 3 out of 3 possible, meaning that based on the information already embedded in the EV SSL certificate, the degree of trust in this website is the highest possible.

Their results were promising, as it had demonstrated improvements regarding who owns the website, what data safety measures they utilise and raised it when encryption is present and it increased the accuracy of security decisions. In their concluding remarks, Sobey and Biddle put their finger on the user disparities between different browser brands or new versions of the same browsers that often change the interface and messages, leading to unnecessary added confusion. [24]

8.3. Certificate revocation methods

In 2011 when DigiNotar was attacked and its certificate issuing mechanism compromised, then the browser developers were quick to update their products to blacklist that particular chain of trust. Including Microsoft (as responsible for Internet Explorer), that are known for only releasing security updates the second Tuesday in each month. Even they had to react firmly with a response that was ready within 24 hours, further emphasising the severity of the problem.

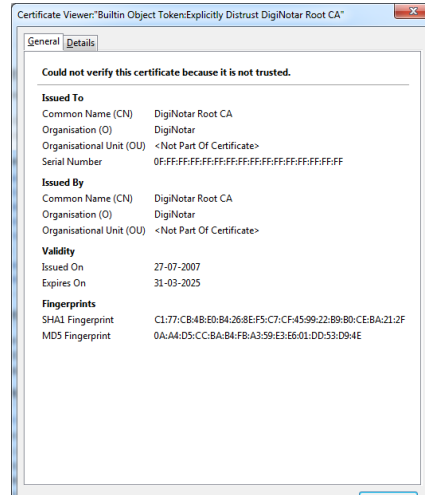


Figure 43: Firefox' DigiNotar distrust

Another option is to make use of a certificate revocation list (CRL), which is the CAs' own mechanism to distrust an already issued certificate. Any CA can initiate a revocation list and they are generally issued at set intervals, however the lists may also be issued right after a revocation has taken place. Unlike program updates resulting in distrust of a third party, a CRL would be impossible in DigiNotar's case, as it could not depend on the X.509 structure for distrusting itself.

Yet another downside with CRLs, apart from when it is the CA's own self-signed certificate that is distrusted, is that in order for revocation to happen, it has to be checked every time trust is going to be placed in any given certificate. If this fails, a distrusted certificate will be able to keep functioning as a trusted one, so for this PKI scheme to be effective, it must always have access to up-to-date CRLs.

Even if this requirement was met, an attack on a CA's internet connection that renders it unable to communicate will result in major issues if certificates cannot be reviewed accordingly. These are among the reasons that an alternative method was developed, being the online certificate status protocol (OCSP).

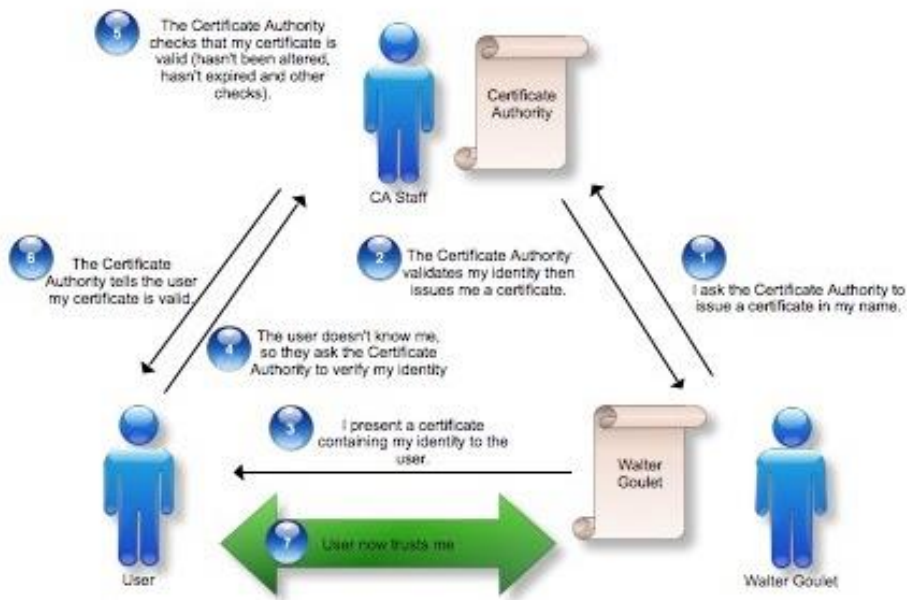


Figure 44: OCSP exchange [27]

OCSP is a way to integrate CRLs into current X.509 infrastructure, where it benefits over traditional CRL, is by utilising less data transmission and real-time and near real-time status checks for crucial operations.

It can even support more than one level of CA where it may be chained between other peer responders in a query, where these responders may then verify each other's OCSP responses against a root CA.

9. Various degrees of encryption

Encryption is the mechanism that ensures confidentiality of data between two or more entities and obfuscation for everyone else, meaning that only the involved parties are able to decipher the information based on a, for them, common decryption scheme. One aspect is confidentiality, however in telecommunication, there is no such thing as infallible data transmissions and that could mean that messages between two parties have been altered slightly along the way. This is why there is a need for a function, which can check if a message has been altered in transit and ensure the message's integrity, being the hash functions.

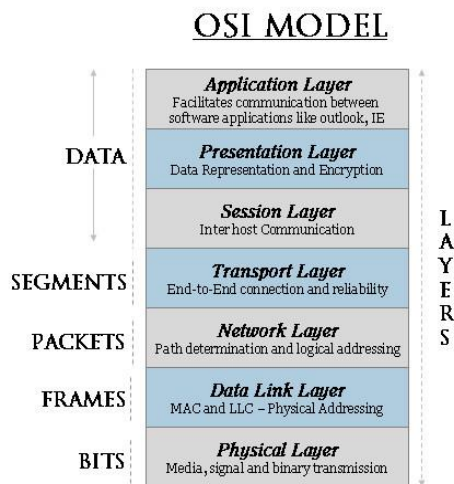


Figure 45: The 7-layer OSI model with layer 1 at the bottom and 7 at the top

One overshadowing issue with both encryption and hash functions is the low rate of change that has been applied throughout the years and the support for backwards compatibility. Somehow, a widespread wish against the quick adaptation of new technology updates and upgrades – I classify going from SSL 2.0 to 3.0 as an update and from SSL 3.0 to TLS 1.0 as an upgrade – seems to exist. At times even decades pass by without adopting and incorporating already proven technology along with phasing out deprecated mechanisms, either because

no one wants to be first mover or because they are in no hurry since existing technology already works satisfyingly and there is little incentive to spend money on new implementations. It is only at the end-to-end services this is required though, as the in-between network providers are not affected by this. This is due to the way data is encapsulated within each other from the top to bottom, as seen in Figure 45, to end up as streams of bits. An internet provider offering an IP addressing scheme need only concern itself with up to layer 3, the network layer, and is generally indifferent about what transpires in layer 4 to 7 where the software running on home computers and servers come in, such as a browser and a web shop.

Finally, there is the issue of fallback methods, where if the best method cannot be achieved between two entities, they will try to negotiate usage of other less-secure methods in a systematic way, until a connection is eventually achieved. This can have a severe potential if a system is allowed to fall so far back that they communicate in a way that is considered insecure and much software is guilty of this work method, where it will rather want to support *usability* and not *security*. One such application is Java from Oracle, as seen in Figure 46, and it is used on a wide range of devices, most likely counting hundreds of millions of installations worldwide. Its default setting is to disable what is considered insecure (red), enable the two most widespread adaptations used as of this writing (yellow), but for some reason also disable the two most recent and more secure updates (green).

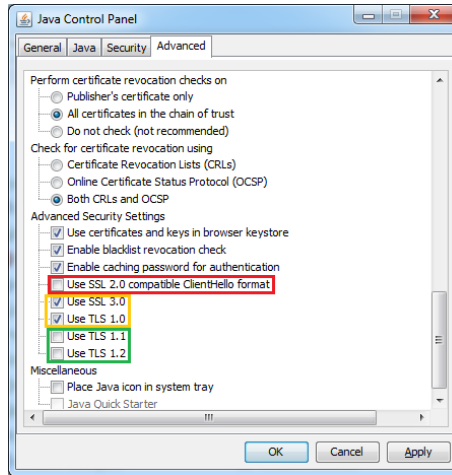


Figure 46: Java ver. 7 upd. 45 control panel

9.1. Secure sockets layer (SSL)

SSL was invented by Netscape Communications and works at the sixth layer, presentation, where it provides its services for the seventh layer, application, and the protocols that exist there, such as HTTP for website traffic. Two versions of SSL still exist while this is being written, 2.0 and 3.0 where the former “does not provide a sufficiently high level of security and has known deficiencies” and is due to [28]:

- The usage of message digest 5 (MD5) for message authentication, which is deemed out of date and insecure
- The initial handshake messages are not protected and could permit a man-in-the-middle (MITM) attack
- The same key is used for both message integrity and message encryption, which is undesirable if a weak key is negotiated
- The data sessions can be easily terminated by a MITM and so it cannot be determined if it was a legitimate end or not

There is evidence for SSL 2.0 from 1995 being very much outdated and insecure, yet there is still support for its usage in Java where it appears that they have not wanted to take the final step and remove it, no matter how many of their customers that may still be using it.

SSL 3.0 is also an old acquaintance from 1996 and should no longer be considered secure, yet there is a tremendous amount of websites still supporting it. It uses old and insecure data encryption keys provided by data encryption standard (DES), triple DES (3DES) and RC4 but also never authentication schemes such as RSA, named after its creators Ron Rivest, Adi Shamir and Leonard Adleman. It has the option to choose between MD5 and secure hash algorithm version 1 (SHA-1) for integrity check, where the latter is considered the better of the two. [29]

9.2. Transport layer security (TLS)

Compared to SSL, the current three different versions (1.0, 1.1 and 1.2) of TLS share an advantage, being that it does not generally downgrade to either SSL 3.0 if first it has been initiated with TLS although it can be forced to do so, at the cost of weakening security. However, in March 2011 the last backwards compatibility between TLS and SSL 2.0 was finally removed.

TLS 1.0 was defined in January 1999, 1.1 came to in April 2006 and only included a minor set of updates but with 1.2 in 2008 arrived an appreciated change of the MD5 and SHA-1 integrity checks, being the SHA-2 with increased cipher strength and support for the much more up-to-date advanced encryption standard (AES). [30]

Protocol	Support	Best protocol
SSL v2.0	302,886	-
SSL v3.0	607,249	3,249
TLS v1.0	604,242	603,404
TLS v1.1	838	827
TLS v1.2	11	11

Figure 47: 2010 website encryption

One major obstacle remains, which is that few websites appear to have taken TLS 1.1 and 1.2 to heart. In a data-farming survey from 2010, Ivan Ristic found that out of approximately 600,000 servers with a certificate installed; see Figure 47, just about half

still supported SSL 2.0 if asked to, although none of them advertised it as being the best they could offer. [31]

On the other hand, there is an astonishingly low representation of TLS 1.1 and 1.2, despite having been standards for 4 and 2 years respectively. One explanation to why that is, could be that it requires action on the webserver administrators' part to implement a change that for apparently many is not even cosmetic, as it does not visibly change anything for their users. Not even if it means a strengthened structure behind their own website and for their users where at long last by January 2014, the five major browser brands all finally support TLS 1.2.

9.3. MD5 and SHA-1 are still being used, even when insecure

Fingerprints

SHA1 Fingerprint 9D:A2:57:46:47:66:4C:49:DD:9F:6E:6C:69:A4:51:2A:E9:A9:2E:C7
 MD5 Fingerprint B3:D8:DB:8C:66:26:B7:27:26:0D:E3:FC:DB:CC:91:2B

Figure 48: Fingerprints from the certificate of **nets-danid.dk**

As explained previously, their function is to ensure that no matter the type of change in transit, the certificate as a whole makes the above fingerprints appear different from what they are here in Figure 48. Its physical world equivalent is likely the old wax seal on a document that is broken on use, but where the options of digital fraud are much more sophisticated and also untraceable. Therefore, in return, the protection must be sophisticated and thorough as well.

MD5 is already written off as insecure, yet it still exists in certificates. SHA-1 is better but with increased computational power, weaknesses have been found in that too. SHA-2 from 2010 and SHA-3 from 2012, where only theoretical attacks have been successful on SHA-2 and there are to date no known successful attacks on SHA-3, have come to take over SHA.1. [32]

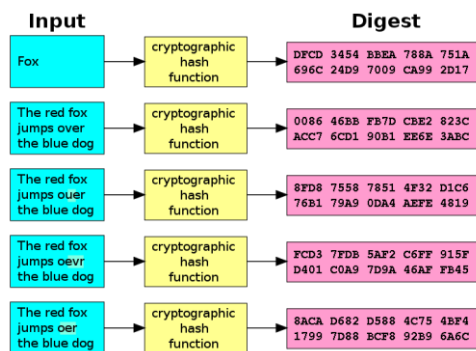


Figure 49: SHA-1 in action

Despite the obvious strengths of SHA-3, Microsoft has said in 2013 that they will stop letting Windows read SHA-1 certificates by 2017, forcing a replacement by minimum SHA-2, but where they might as well have gone for SHA-3. [33]

10. Alternatives to traditional certificate PKI structure

Audun Jøsang is the author of an article from 2011 in which he suggests an alternative solutions to the current PKI functions with CAs and pre-trusted root CAs implemented in browsers. He argues that since certificates already contain DNS names, then they should also be issued and distributed by the DNS system itself as well, through the usage of domain name system security extensions (DNSSEC).

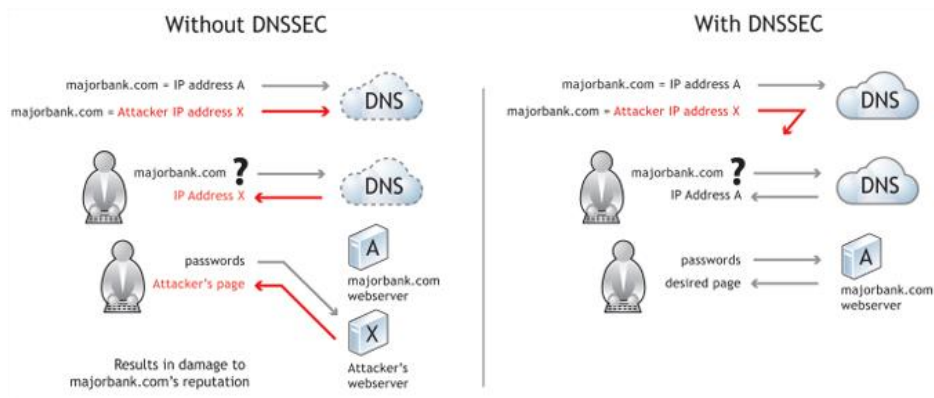


Figure 50: ICANN's presentation of DNSSEC [34]

DNSSEC differs from DNS by offering signed responses for, that the IP answers that a server transmits are the same it received from its own requests towards the upper hierarchies and they have not been maliciously altered during transit.

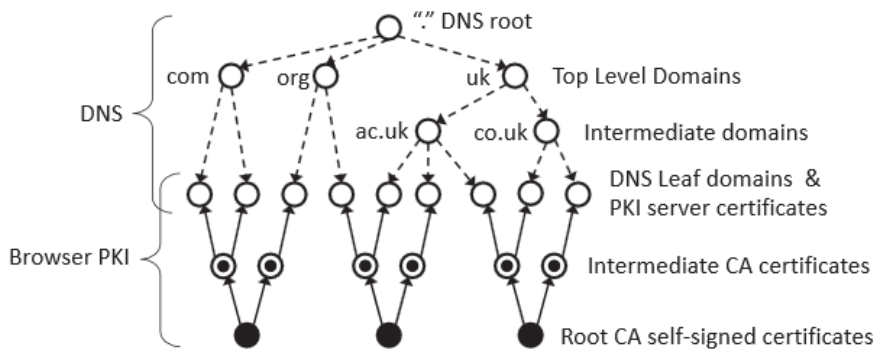


Figure 51: Jøsang's figure on where DNS and PKI X.509 overlap

DNSSEC can also protect information other than just the integrity of domain names, by binding the public keys on certificates to domain names and/or IP addresses. This is instead of distributing the public keys through an insecure channel as shown previously on Figure 35, only to secure the very same channel afterwards.

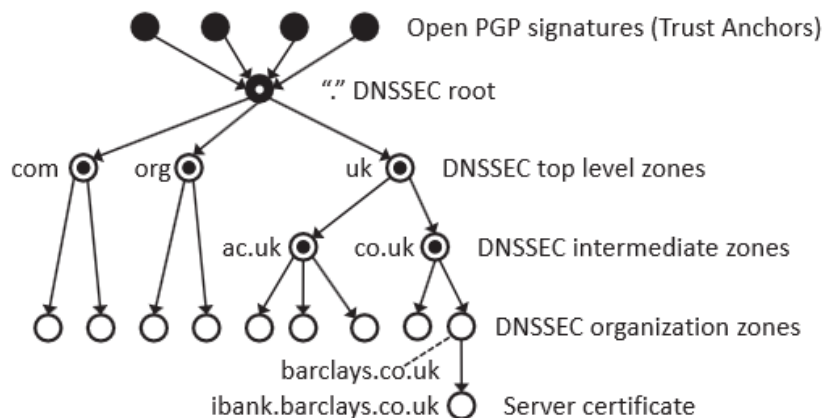


Figure 52: Jøsang's DNSSEC as certificate platform

Figure 52 is the depiction of a proposition where a British bank's website is trusted based on the DNS zone in which the particular server is located. There is still need for connections over TLS to the actual banking server at the bottom called **ibank.barclays.co.uk** but the certificate is signed by the DNS zone **barclays.co.uk** and not any given intermediate CA. The certificate is also able to be stored on the DNS server itself and thus be available for all who are accessing it. This will solve the problem of trusting the separate PKI structure, the server authentication strengthened and costs reduced when DNSSEC is deployed. [35]

The strength of such a system compared to traditional X.509 structure is:

- It will only be based on server-side administration, so that root certificates no longer have to be distributed with various browsers.
- Trusting someone's self-signed certificate is considered a security risk, but that is precisely what the root CAs themselves are asking for.
- DNSSEC is general and does not rely on a range of different CAs that have to earn money by certifying unreliable and outdated trust schemes.

11. Creating a browser extension that does it right

Looking at the previous initiatives highlighted in this thesis, there is not a shortage of well-meaning additions that aim to help its users make educated choices. There has been conducted numerous experiments with colour and image changes and different representations of the already existing security indicators – but with only a relatively small amount of success. With that, I mean that none of the initiatives has managed to create something groundbreaking, no matter who the test persons were or in which kind of environment the tests were carried out.

What they have in common is that they base an increased security awareness around schemes that, admittedly, offer confidentiality through encryption but the actual and real identity confirmation of the proprietor of a website can remain a mystery. The extended validation only requires that the company spokesperson is not a made-up character, is related to domain ownership and can sign a legal document on behalf of a company. It is a step in the right direction but it lacks vision and is still tied up on a company that aims to make a revenue.

The primary reason for the industry showing itself as it does today, I believe, comes from the 1990's where both security and usability in browsers were on a very low level. The information box from Internet Explorer 6 on Figure 36 at page 51 warns the first time users of something they in the first place do not understand. Secondly, that the pop-up box suggests to never mentioning it again says many things about in which state usability and security was. IE6 was also known as the most unsafe way to browse the internet due to its amenable way of running every malicious content one could throw at it. Of course, a new browser installation in 2014 does not even make the user aware that unencrypted web traffic is open for eavesdropping by a third party and the persons who should tell this to the younger generations cannot do it because they were not rigorous enough in getting to understand that themselves.

Another major issue has been the technical jargon that only persons related to computer science and encryption schemes understand, whereas the everyday user lacks the necessary knowledge to assess properly what the systems tell them. Terms such as “eavesdropping”, “encryption”, “digital certificate” and “hash function” are not very user friendly even if they make perfect sense to myself. It would be a completely different story if I asked my younger brother and parents.

Not all blame rests on the lack of education of users, because the industry itself bears an even greater amount of responsibility. Notorious lack of constant care in renewing encryption and authentication schemes rests solely on the parties that keep issuing and supporting outdated and insecure mechanisms such as SSL 2.0 and MD5, which by 2014 are respectively 19 and 23 years old. They should have been taken completely out of commission years ago. The same goes for Microsoft's statement that only by 2017 will they prevent their operating systems from accepting certificates that use the SHA-1 algorithm, which is also too long to keep supporting an old standard that was superseded by SHA-2 back in 2001 and now SHA-3 in 2012. One would think that five years from 2012 to 2017 was ample time to get SHA-3 implemented.

The recurring motif is that nobody wants to be the first to break with the complacency that dominates the security and trust industry and that is why I would like to offer a different perspective on the browser usability and security, along with providing identity checks by usage of public domain databases and business registries.

11.1. The roles of and limitations by using DK-Hostmaster and CVR

Taking company websites located in Denmark as the starting point there are two institutions that can supply more detailed background information to the public, as long as the companies have selected to do so and are not withholding the information. These are the administrator of the .dk domain, DK-Hostmaster (DKH), and The Central Business Register, (CVR), which contains primary data on all businesses in Denmark. This also means that any Danish company that does not have a .dk website or is located in a foreign country and still has a .dk website are not subjects, along with where certificates are not issued to a company.

However, there are a number of limitations to be aware of when requesting data from DKH. The first is that in order to get the best result, one should send the requests to standard TCP port 43 instead of using a screen scraper that performs the same actions a human would on their website. The second is only one connection per 256 of the same IP class D network hosts, which means that two persons sitting on public IP addresses 1.2.3.1 and 1.2.3.2 may not send a request at the same time but 1.2.3.1 and 1.2.4.1 are allowed to do so. The third is a one to two second delay between lookups, so the server does not get overloaded. [36]

Regarding the CVR, it is much more complicated to retrieve information compared to DKH, mainly because it does not answer to internet requests but sends its extracted data by email in a database file containing what is based in various search criteria. In addition to this, there are approximately 650,000 businesses in the register where only 50,000 can be extracted at a time. This means the same extraction process must be carried out at least 13 times for completeness sake and then has to be joined following that.

Since CVR does not support single requests in real time, this operation also has to be carried out frequently in order to not use a deprecated database. Exactly how often is up for debate, because on the one hand, the extracted data has to be as recent as possible but on the other hand, the amount of work needed to maintain it must also fall within acceptable boundaries.

One option is to let a program and an email scraper perform the action once every 24 hours and then host the database file on an internet server, thus making it available for public lookup by the extension, which could then cache either all or just some of it locally.

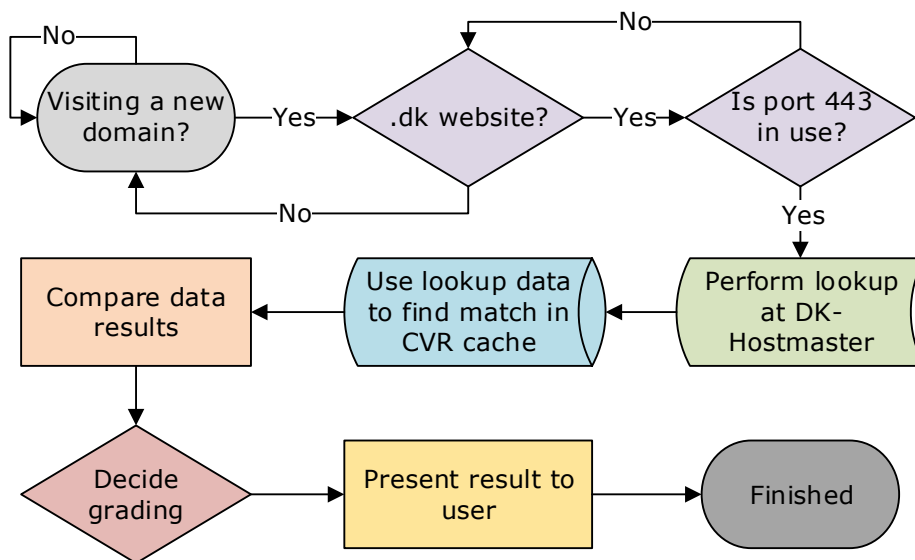


Figure 53: Extension workings flowchart

Figure 53 is my vision of how such an extension could work. It adheres by the DKH limitations since it only performs a lookup when it is both a .dk address and it uses secure HTTPS over TCP port 443. No matter how much or little any given company has chosen to share; there will always be some kind of data return that can be compared with the CVR database but the next challenge will be to compare it with CVR data in a quick and precise way.

The extension should therefore check if the current domain the browser is visiting stays the same or changes. If there are no changes, it continuously loops until a new domain is visited and once that happens, it should proceed to find out if its top level is .dk. Since it makes little sense to query DKH unless a certificate is in use, it should only happen when the remote TCP port is 443. If the three prerequisites are met, the lookup should be performed and the data returned should be passed on to a function that indexes the (for now) offline copy of the CVR database.

Depending on how much of a name match is found, it could be less than 25%, 25-75% and more than 75%, where the grading then can be decided accordingly. I will recommend a design like back on Figure 42 with the three blue dots that light up from one to three instead of one indicator that changes colour, in order to consider colour-blind persons. Achieving three dots should therefore provide the users with a high degree of certainty that the domain name has been registered by an actual and legit company, by the processing of information from two independent “real” Danish authorities.

11.2. The toolbars that did not achieve the desired effect

In an article from 2006, Min Wu, Robert Miller and Simson Garfinkel are researching if security toolbars have any effects on reducing the amount of users falling prey to phishing schemes and they identify and dissect five different toolbar products:

- SpooftStick displays the visited website’s real domain name in order for exposing phishing sites that otherwise obscure it.
- Netcraft displays information about the visited domain along with the date of its registration, where it is being hosted and its popularity based on visits by other users of the toolbar.

- TrustBar makes SSL connections more visible by displaying the website's logo and CA.
- Account Guard by eBay that displays a green icon if the site being visited belongs to eBay or PayPal, red icon if it matches a list of known phishing sites and grey icon for everything else.
- SpoofGuard, which does a calculation of a set of heuristics, derived from earlier phishing attacks. Then it translates them into a score of red, yellow and green regarding when it is likely a swindle, fifty-fifty and unlikely.

SpoofStick

You're on **paypal.com**

Netcraft Toolbar

Since: [Oct 2001](#) Rank: [41](#) [Site Report](#)  [US] [eBay, Inc](#)

TrustBar



eBay Account Guard



SpoofGuard

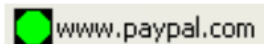


Figure 54: The five different toolbars in action

They also identify some general shortcomings of toolbars, being that:

- They are often very small and are placed in the browsers' peripheral area (the chrome) where it is likely to not receive the required amount of attention at the right times.
- They show security related information but security is only rarely the user's primary goal in web browsing and is likely not to care about it.
- If it is not one hundred percent accurate in its passing of judgement then the users are likely to learn to distrust it. Therefore, when it correctly identifies a fraud, the user is unlikely to believe in it.

Another important factor they highlight is the usual absence of user guides or tutorials for these toolbars and that users seldom read the documentation anyway.

Their pilot project showed that providing a printed a tutorial had a remarkable effect on the users' performance where only 7% of the phishing attacks in a group of five subjects were successful. However when a group of other test users did not receive a tutorial but had to click a link in the toolbar to read the documentation, nobody did and succumbed to 94% of the attacks.

Many of their test subjects also relied on being able to tell from the website content itself if the site was a swindle or not, in particular because the website takes up most of the space in a browser and is centrally placed. The earliest phishing websites were also subject to very bad grammar but the simulated attacks were rid of that and of a high quality, resulting in users disregarding the toolbars simply because of website design aesthetics. They conclude that the toolbars by themselves are unable to prevent users from being spoofed by phishing attacks of a high quality design and failed to keep attention on the security indicators and if they did, they did not adequately know how to interpret them. [37]

11.3. Hope remains for making users rely on add-on programs

While the findings of Wu, Miller and Garfinkel in the previous chapter certainly do not speak in favour of making another toolbar that will require its users first to read a printed documentation, I will allow myself to remain positive due to the popularity of two extensions for the Firefox browser, being Adblock Plus (ABP) and NoScript Security Suite (NSSS).

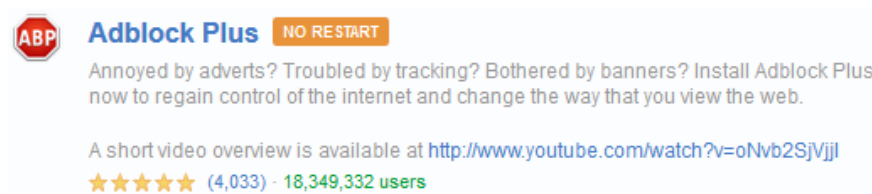


Figure 55: The most popular Firefox extension, ABP

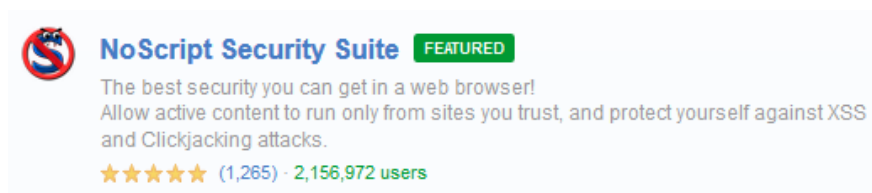


Figure 56: The fourth most popular Firefox extension, NSSS

Neither ABP nor NSSS are being advertised by the browser's installation program or start-up page and as such have to be found and installed on the users' own initiative. ABP has become so successful, that a number of website owners relying on advertising for providing free content have begun a campaign to make visitors with ABP installed either remove it or at least whitelist their website. NSSS cannot boast the same impressive 18.3 million users but 2.2 million that are mindful of website security issues is no small feat. A combined 20 million (although there may be overlaps) users as of January 2014 of two very small programs based on a single browser brand shows that it is possible to rely on users for installing useful programs only by word-of-mouth and I believe this strategy could be the way for my toolbar as well.

One factor that would help immensely in sanctioning the toolbar is if also the Danish banks would support the project. Albeit their direct influence in this project is very small, which also becomes known in chapter 7, they remain widely known physical institutions through which every Danish citizen has some sort of monetary relationship, whether they like it or not. Post the financial crisis in 2008 their reputation has suffered considerably but even so, if the bank handling their finances also acknowledge the usage of a toolbar to prevent fraud, the customers should be likely to trust the bank's statement.

The evaluation process will also have to be looked at, where there is a tradition for observing test subjects either with or without preceding knowledge about what they are about to embark upon. It is also of utmost necessity to abide by the results from Wu, Miller and Garfinkel's research where "forcing" new users to go through a tutorial and have the toolbar draw visible attention to itself on every .dk website that utilises HTTPS have to be implemented. High usability that follows Ka-Ping Yee's "path of least resistance" has shown itself closer to human nature on a computer than any security precautions have, so it is also a matter of changing the human nature.

Here I believe a fruitful addition to the test evaluation team could be an anthropologist who is much more likely to observe details that are less suited for direct security comparisons, but could indirectly prove a valuable asset when designing a security toolbar. Especially one that users will have to find meaningful to install and keep looking to for website evaluations by themselves.

12. From addressing the problem to not becoming the problem

If one is to apply self-criticism to a toolbar such as this, a major flaw is undoubtedly that there is a high chance for it to end up as yet another piece of software that people use, but do not know how it works and in the end do not really care about it either. This being the case for how CAs have brilliantly turned issuing of identity into big business and have grown utterly complacent and neglected to, if not evolve then at least adapt, more current technology and phase out the old more rapidly.

Assuming that it could gain a broad adaptation in the public, its displayed results shall require a rigorous accuracy in order to keep trustworthiness since losing that will ultimately mean the end of it. Trust is good but blind trust is not, whereas it defeats the purpose if the toolbar should somehow take over the CA role, then nothing will be learned and gained as its goal should only be to support a decision making process, not carry it out by itself.

This puts a significant amount of power in the databases maintained by DKH and CVR and requires that they are both always available and that editing one's own database entries can be swiftly carried out. Otherwise, there will come to exist a distrust to the public authorities and nobody will benefit from that.

13. Conclusion

Digital certificates that provide browser security are not providing the same kind of identity that people know and relate to from real world experiences and their inner workings are too often tied up on outdated cryptographic functions that cannot be considered secure any longer. A whole industry's complacency followed up with lacking security design choices are major reasons for this state of things, where nobody dares to deprecate old mechanisms and set deadlines for new adaptations too closely into the future. Computer users are often either unable to understand what security actions are required of them or simply do not possess sufficient knowledge to act accordingly and this is why it is very difficult to reliably know about and see through the paid identity schemes on which the certificate authorities thrive.

The proposed solution is to incorporate data from two Danish public authorities that users will either already know or better relate to, rather than being ensured by an unknown company that the visited website is genuine. It is a design process, fit for being taken the next step and developed into an actual product. Research on how users interact with security schemes along with personal experiences from work situations are the foundations of how it should be designed and what can be done to catch the hearts and minds of potential users. Further evaluation requires a group of test subjects but has to be carried out with more emphasis on human behaviour and how to reward any appropriate actions taken.

It is very unlikely that the way digital certificates are being used is going to change, especially because they are used on a global scale but that does not prevent helpful browser plugins from being created for national purposes. Nevertheless, with free browser choice and no reliable way to ensure wide adaptation of a plugin, users have to believe they are doing something actively to better be protected from phishing attacks and other fraudulent schemes, because it uses information from two authoritative national institutions. If not, then it is likely that it will end up becoming just another good idea that only a handful use and even fewer of those pay the necessary attention to.

However, post the incidents of 2013 where the American national security agency's (NSA) methods were brought to light, it should provide fertile soil for increased awareness that demands uncompromised information about identities and the data integrity between them, like this browser plugin can help provide.

14. References

- [1] Dhamija, R., & Tygar, J. D. (2005, July). The battle against phishing: Dynamic security skins. In *Proceedings of the 2005 symposium on Usable privacy and security* (pp. 77-88). ACM.
- [2] Found thumb drives: Another way employees are a security menace {2013-12-12}
http://gcn.com/blogs/pulse/2013/11/~/_link.aspx?_id=6F7ED59B05F645EB9E31937A968A338C&_z=z
- [3] Word Spy on “phishing” {2013-12-16}
<http://www.wordspy.com/words/phishing.asp>
- [4] Anti-Phishing Working Group – About {2013-12-16}
<http://apwg.eu/about-APWG/>
- [5] Global phishing survey: Trends and domain name use in 1H2013
http://docs.apwg.org/reports/APWG_GlobalPhishingSurvey_1H2013.pdf
- [6] Phishing in season: A look at online fraud in 2012 {2013-12-16}
<https://blogs.rsa.com/phishing-in-season-a-look-at-online-fraud-in-2012/>
- [7] Dhamija, R., & Dusseault, L. (2008). The seven flaws of identity management: Usability and security challenges. *Security & Privacy, IEEE*, 6(2), 24-29.
- [8] Yee, K. P. (2004). Aligning security and usability. *Security & Privacy, IEEE*, 2(5), 48-55.
- [9] Gutmann, P., & Grigg, I. (2005). Security usability. *Security & Privacy, IEEE*, 3(4), 56-58.
- [10] Google 2013 Q3 earnings over a 9 months period
http://investor.google.com/pdf/2013Q3_google_earnings_data.pdf
- [11] Srikwan, S., & Jakobsson, M. (2008). Using cartoons to teach internet security. *Cryptologia*, 32(2), 137-154.
- [12] Braz, C., Seffah, A., & M'Raihi, D. (2007). Designing a trade-off between usability and security: A metrics based-model. In *Human-Computer Interaction—INTERACT 2007* (pp. 114-126). Springer Berlin Heidelberg.
- [13] Cranor, L. F. (2007). *Security and usability: Designing secure systems that people can use*. O'reilly.
- [14] Nets on the new initiative (Danish) {2013-11-07}
<http://www.nets.eu/dk-da/Om/nyhedsbreve/cards-nyhedsbrev/Pages/Verified-by-Visa-og-MasterCard-Secure-Code.aspx>

- [15] Arbejdernes Landsbank's FAQ about the initiative (Danish) {2013-11-07}
https://www.al-bank.dk/media/documents/FAQ_3DSecure_2013.pdf
- [16] Jøsang, A., AlFayyadh, B., Grandison, T., AlZomai, M., & McNamara, J. (2007, December). Security usability principles for vulnerability analysis and risk assessment. In *Computer Security Applications Conference, 2007. ACSAC 2007. Twenty-Third Annual* (pp. 269-278). IEEE.
- [17] AlZomai, M., AlFayyadh, B., Jøsang, A., & McCullagh, A. (2008, January). An experimental investigation of the usability of transaction authorization in online bank security systems. In *Proceedings of the sixth Australasian conference on Information security- Volume 81* (pp. 65-73). Australian Computer Society, Inc.
- [18] MD5 considered harmful today {2013-12-23}
<http://www.win.tue.nl/hashclash/rogue-ca/>
- [19] Comparing suitable network security keys: Kerberos and PKI {2013-12-25}
<http://blagovision.org/comparing-suitable-network-security-keys-kerberos-and-pki/>
- [20] How to setup SSL on Arch Linux Apache or NGINX {2013-12-27}
<http://www.adminempire.com/how-to-setup-ssl-on-arch-linux-apache-or-nginx/>
- [21] Preparing for and responding to CA compromise and fraudulent certificate assurance
http://csrc.nist.gov/publications/nistbul/july-2012_itl-bulletin.pdf
- [22] Extended validation {2013-12-31}
http://en.wikipedia.org/wiki/Extended_validation
- [23] Sobey, J., Biddle, R., van Oorschot, P., & Patrick, A. (2008, October). Exploring user reactions to browser cues for extended validation certificates. In *European Symposium on Research in Computer Security*.
- [24] Biddle, R., van Oorschot, P. C., Patrick, A. S., Sobey, J., & Whalen, T. (2009, November). Browser interfaces and extended validation SSL certificates: an empirical study. In *Proceedings of the 2009 ACM workshop on Cloud computing security* (pp. 19-30). ACM.
- [25] Revocation list {2014-01-03}
http://en.wikipedia.org/wiki/Revocation_list
- [26] Online certificate status protocol {2014-01-04}
http://en.wikipedia.org/wiki/Online_Certificate_Status_Protocol
- [27] Summarising PKI certificate validation {2014-01-04}
<http://blog.securism.com/tag/ocsp/>

- [28] RFC 6176: Prohibiting secure sockets layer (SSL) version 2.0
<http://datatracker.ietf.org/doc/rfc6176/>
- [29] RFC 6101: The secure sockets layer (SSL) protocol version 3.0
<http://datatracker.ietf.org/doc/rfc6101/>
- [30] RFC 5246: The transport layer security (TLS) protocol version 1.2
<http://datatracker.ietf.org/doc/rfc5246/>
- [31] Ivan Ristic: Internet SSL survey 2010 – Black Hat USA 2010
http://blog.ivanristic.com/Qualys_SSL_Labs-State_of_SSL_2010-v1.6.pdf
- [32] Cryptographic hash function {2014-01-06}
http://en.wikipedia.org/wiki/Cryptographic_hash_function
- [33] Windows PKI blog – SHA1 deprecation policy {2014-01-06}
<http://blogs.technet.com/b/pki/archive/2013/11/12/sha1-deprecation-policy.aspx>
- [34] ICANN on DNSSEC {2014-01-08}
<http://www.icann.org/en/news/in-focus/dnssec>
- [35] Josang, A., & Dar, K. S. (2011). Server Certificates based on DNSSEC. In *Proceedings of NordSec*.
- [36] DK-Hostmaster’s tech notes – WhoIs service {2014-01-08}
<https://www.dk-hostmaster.dk/english/tech-notes/whois-service/>
- [37] Wu, M., Miller, R. C., & Garfinkel, S. L. (2006, April). Do security toolbars actually prevent phishing attacks? In *Proceedings of the SIGCHI conference on Human Factors in computing systems* (pp. 601-610). ACM.