

An Embarrassingly Simple Solution to the Problem of Protecting Browser Users

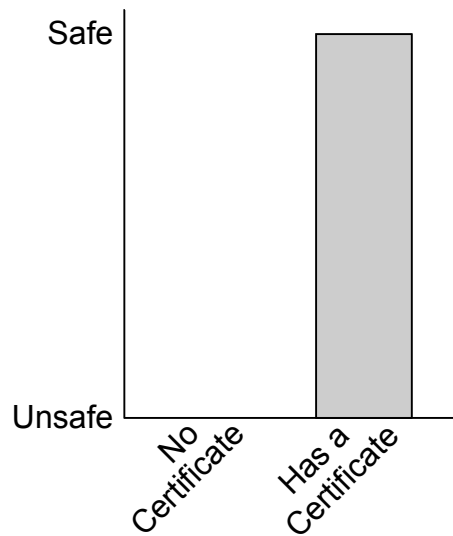
Peter Gutmann
University of Auckland

Browser Security

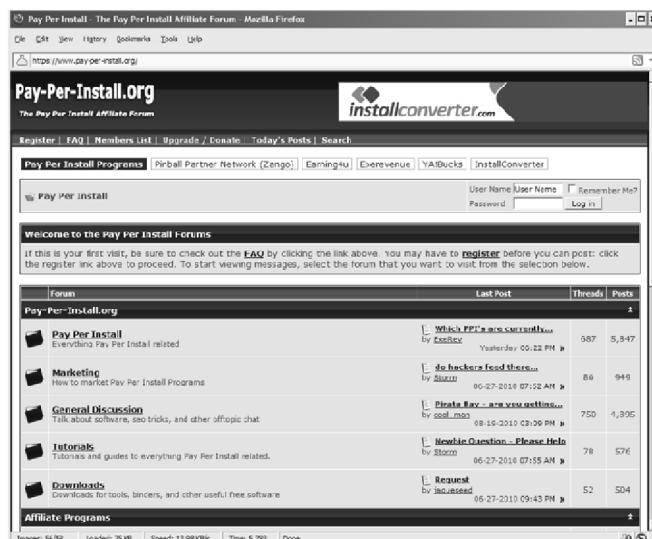
One of the most visible aspects of Internet security

- Everyone's familiar with it
- Experienced by a billion+ people
- Makes a good example to talk about

The Browser Security Value Proposition

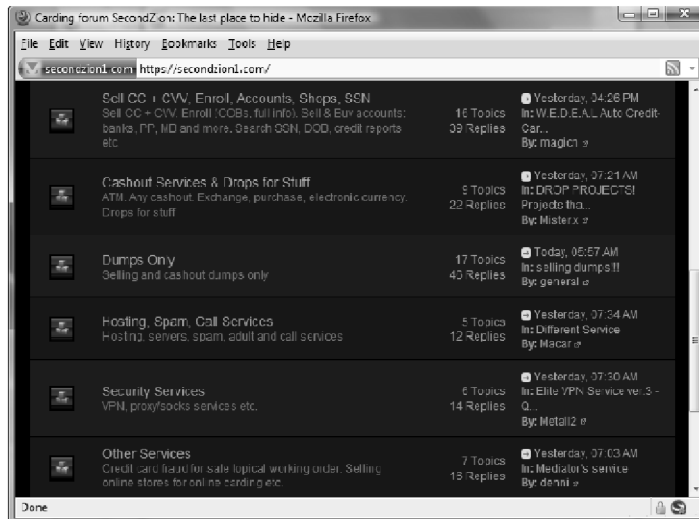


Browser Security: Examples



Obviously a safe, trustworthy site

Browser Security: Examples



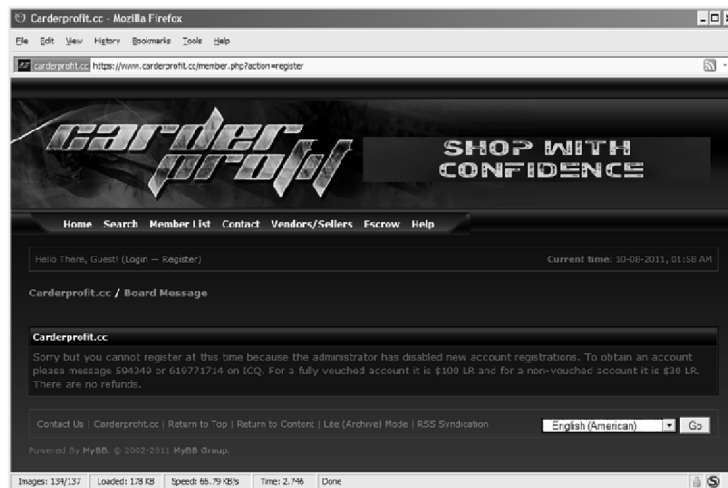
Another safe, trustworthy site

Browser Security: Examples



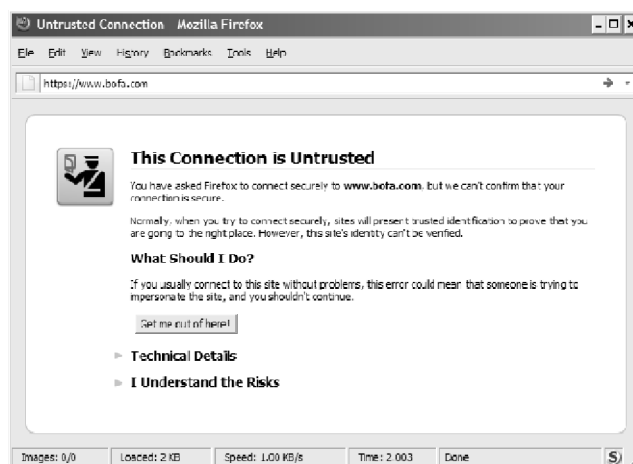
Yet another safe, trustworthy site

Browser Security: Examples



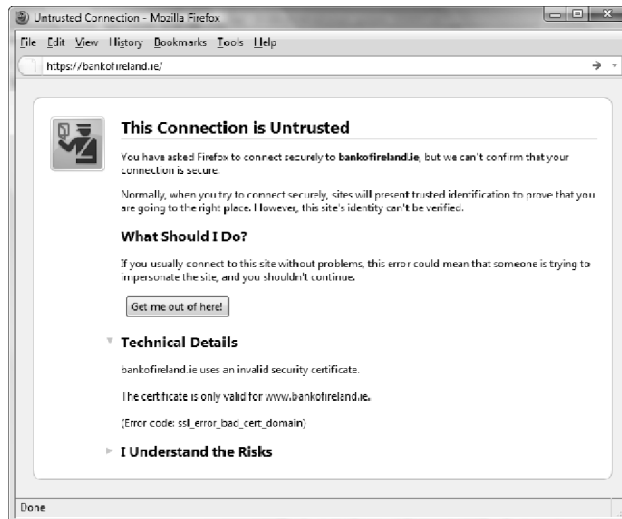
How many more would you like to see?

Browser Security: Examples



An untrustworthy site!

Browser Security: Examples



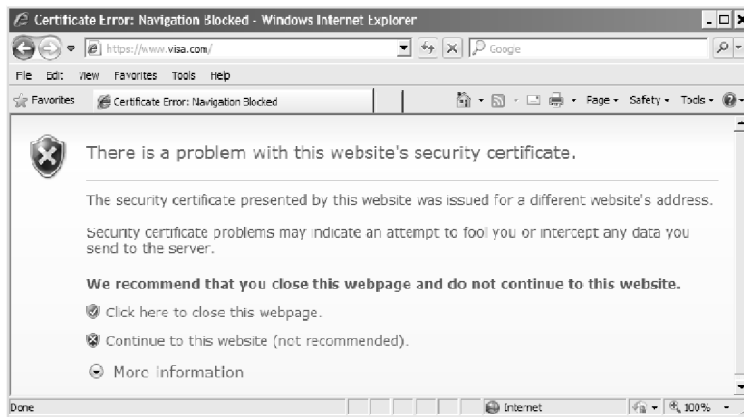
Another untrustworthy site

Browser Security: Examples



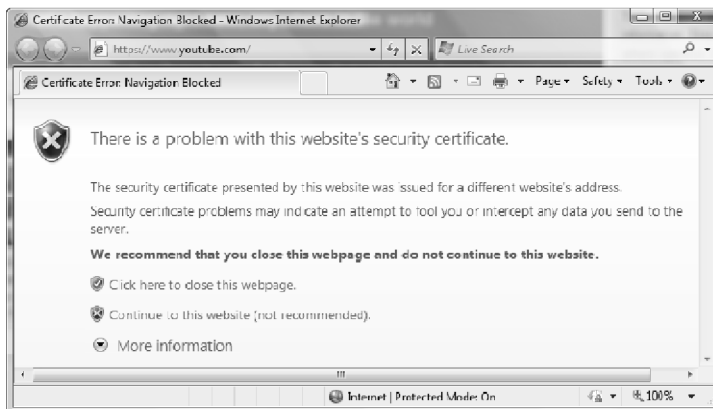
A few years ago...

Browser Security: Examples



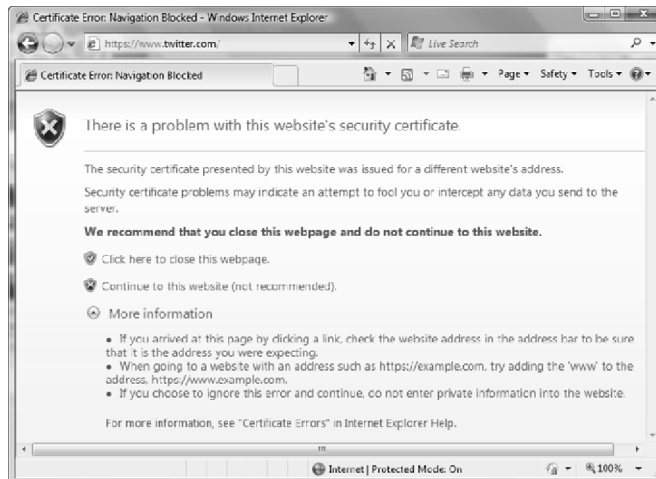
... at least they're consistent over time

Browser Security: Examples



Another dangerous site

Browser Security: Examples



Yet another dangerous site!

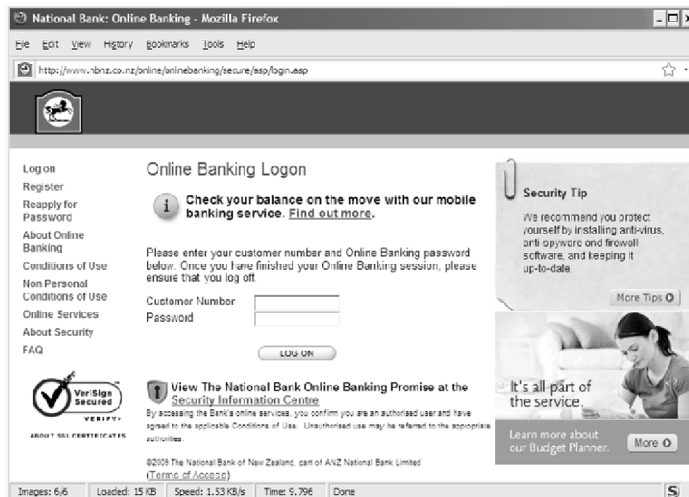
Browser Security

Gosh, the Internet sure is a dangerous place!

- It's a good thing that the browsers are so good at separating the good from the bad

But wait, there's more...

Browser Security, Revisited



What's wrong with this picture?

Browser Security, Revisited



It checks out OK

Browser Security, Revisited

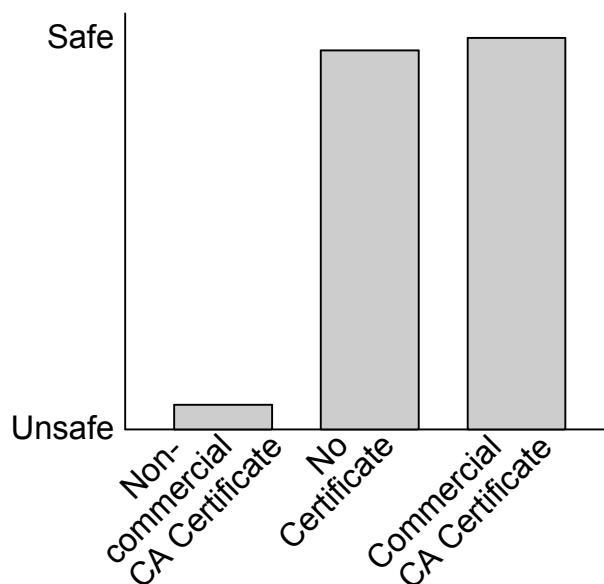
When this was tested in a real-world evaluation, 100% of users were fooled

When tested a second time against a roomful of hardcore geeks, it took multiple iterations of explanation to (try) and convince them that this was a real problem

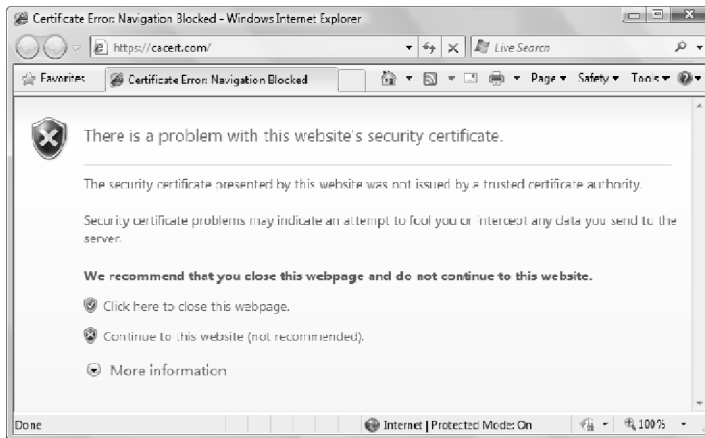
- “Well, it may work against the bank, but I run SSL on my own server and it won’t work there”



The Browser Security Value Prop. Revisited



The Browser Security Value Proposition



Youse gotta real nice lookin' web site here.
Be a shame iff'n customers was scared away...

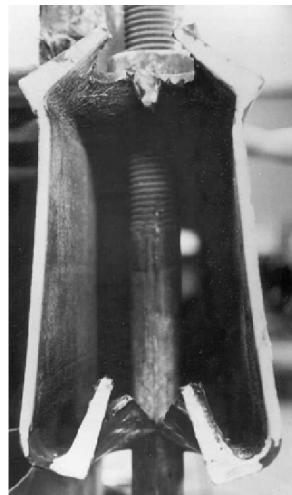
Risk Management / Sec.through Diversity

No diversification of defences

- Single point of failure

With diversified defences ...

- Individual defensive features may only make a small contribution to the overall strategy
- Combination of features makes an attacker's task much more challenging



Risk Management / Sec.through Diversity

Not the same as defence in depth



- Defence in depth is a set of layered defences that an attacker has to knock out one after the other

Risk Management / Sec.through Diversity

Diversification of defences has been a standard aspect of physical security since time immemorial

- Formalised in crime prevention through environmental design (CPTED)

Risk management means taking deliberate action to shift the odds in your favour — increasing the odds of good outcomes and reducing the odds of bad outcomes

— Dan Borge, Bankers Trust

Risk Management / Sec.through Diversity

Takes ideas from behavioural theory and community organisational theory

- Analyses the intended use of an area
- Examines how it can be arranged to minimise crime



Reduces the propensity of the physical environment to support criminal behaviour

Risk Management / Sec.through Diversity

CPTED includes the usual security measures ...

- Provide adequate lighting
- Minimise dense shrubbery
- ...



Fort ATF, Washington

Risk Management / Sec.through Diversity

... but also less-obvious ones

- Position windows for passive surveillance of public spaces
 - Based on the “eyes on the street” concept of urban activist Jane Jacobs
- Locate car parks in front of office buildings rather than around the side out of sight
 - For schools, put bike racks in front of classroom windows
- Use glazed-in balconies for residential buildings
 - Encourages the occupants to sit in them
 - Provides surveillance of the area outside the building

Risk Management / Sec.through Diversity

Spot (some of) the CPTED features



Risk Management / Sec.through Diversity

Put ATMs out on the street in full view of passers-by

- This is why they always seem to be positioned with no regard for user privacy
- They *almost* got this one right



Risk Management / Sec.through Diversity

Design residential houses for passive surveillance

- Orient houses in a cul-de-sac at 45 degrees to the entrance to the cul-de-sac
- Locate high-occupancy areas (traditionally kitchens) by the street entrance
 - Kitchen as panopticon predates CEPTED by centuries



Risk Management / Sec.through Diversity

Other measures are even less obvious...



Eliminate features that provide easy access to roofs

Use slippery (high-gloss) paint for columns and supports

Risk Management / Sec.through Diversity

Plant climbing plants along walls subject to graffiti



- Alternatively, use textured/patterned surfaces to the same effect

Risk Management / Sec.through Diversity

Use thorny plants to discourage people from entering areas



Risk Management / Sec.through Diversity

Paint areas around night-time lighting with white, reflective paint



Risk Management / Sec.through Diversity

These are pretty trivial measures

- The choice of *paint* used?

In combination they add up to considerable integrated defensive system

Risk Management / Sec.through Diversity

Security for the built environment isn't just about following fixed rules

- “We can secure schools by installing CCTV cameras”
- Kids who damage school property often do it to have their moment of fame
- Letting them know they could end up on TV is entirely the *wrong* motivation



Risk Management / Sec.through Diversity

What about providing adequate lighting?

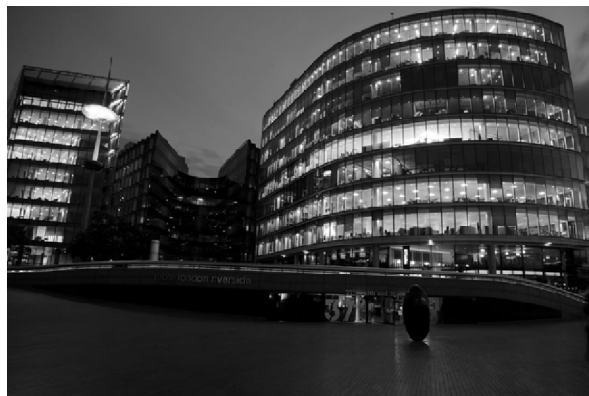
- Lighting up school sports grounds is an invitation for unauthorised use
- Provides a perfect excuse for trespassers to be on school grounds after dark
- Police are unlikely to prosecute a couple of teenagers for (apparently) wanting to kick a ball around.



Risk Management / Sec.through Diversity

Darkness can be a legitimate lighting strategy for CPTED

- Having buildings brightly lit helps intruders



- They don't have to provide their own lighting

Risk Management / Sec.through Diversity

Provide minimal lighting required by building codes

- Forces intruders to turn on the lights
 - These can be wired to an alarm
- Alternatively, they can use flashlights
- There are few things more attention-grabbing for security guards than a torch flashing around inside a dark building



Risk Management / Sec.through Diversity

Reverses the conventional thinking about using illumination for security

- Makes darkness a part of the security

Security through Diversity for the Web

The Internet has a great deal of diversifiable risk

- No need to rely on a single defence mechanism to try and cope with all risk
- Risks total failure when the sole mechanism fails

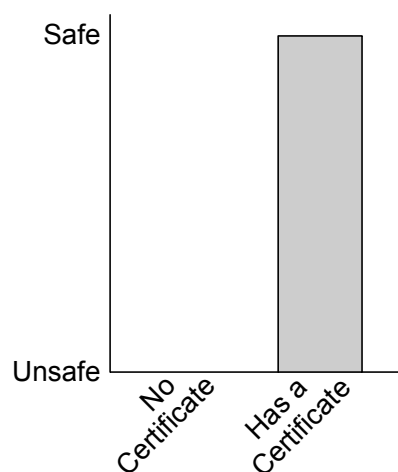
Diversify the mechanisms you use so that

- A failure of one of them can, at worst, somewhat increase the overall risk
- Won't lead to a total failure of security



Security through Diversity for the Web

Refresher: The current state of security through diversity on the web



Security through Diversity for the Web

Risk-based security assessment treats online risk as a sliding scale

- Not “good” or “bad” but “probably safe” down to “probably unsafe”

Already supported by web browsers for dealing with privacy settings

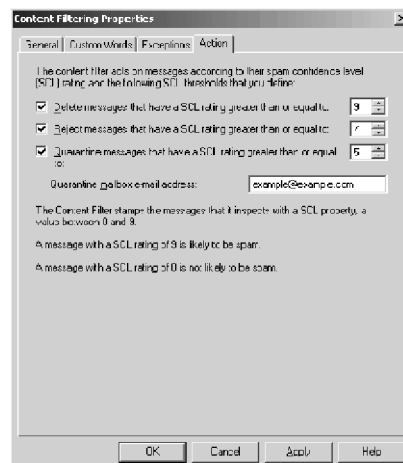
- Mostly based around specifying cookie-handling and script/plugin permissions in mind-numbing detail, ugh

Allow the user to choose their level of risk aversion

- *In consultation with interaction designers*

Security through Diversity for the Web

Mail software has done this for years



- (OK, the UI could do with a bit of work...)

Security through Diversity for the Web

Users could choose to be highly risk-averse ...

- Site has to pass some fairly stringent checks in order to be regarded as safe

... through to highly permissive

- Few checks are performed before the site is regarded as safe
- This is the current default for all browsers

Can be adapted over time as risk assessment measures mature

- Default setting can be moved from the current totally permissive level to more restrictive settings
- Users can choose to voluntarily apply more restrictive settings

Security through Diversity for the Web

Browser can choose to automatically apply changes in risk tolerance in a situation-specific manner

- Laptop normally connects to the Internet through a home wireless network
- Is currently connected through an unrecognised network
 - You're in an Internet café
 - Win7 already does something like this



Security through Diversity for the Web

Browser can increase the risk-aversion factor

- Extra checking is applied to sites before they're regarded as safe

Browser has far better situational awareness than the user

- Hijacked WiFi connection via a spoofed AP
- The user can't tell the difference, the computer can



Security through Diversity for the Web

Managing risk through diversification on the web works like CPTED

- There's no single silver bullet

There are lots of measures that make things a bit better

- Patterns

There are also many that we know make things worse

- Antipatterns

Security through Diversity for the Web

CPTED danger signs

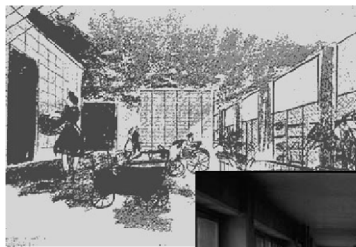
- High-rise multi-unit housing for low income-earners
 - “public housing” / “housing projects” / “housing estates”
- Families with a high proportion of young members
 - Think “youths” rather than “children”
- Shared communal entrances and living areas
- Elevators feeding double-hung corridors



Security through Diversity for the Web

This is a recipe for disaster

- We have entire books on design patterns for effective CPTED
- Extensive research and real-world experience have shown us what reduces crime and vandalism



Security through Diversity for the Web

Risk-based assessment isn't a silver bullet

- Can't give you a "definitely OK" or "definitely bad"
- Unfortunately conflicts with humans' innate zero-risk bias

Can give you a risk-based analysis of a site's trustworthiness

- Probably OK
- Probably dubious

Indications as to why it is or isn't OK

- A general measure of the overall level of risk or non-risk
- c.f. certificates: All a browser can say is "Someone paid a CA for this"

Key continuity

Is this the same key/certificate that we saw the last time we came here?

- SSH-style key continuity

Key continuity

Certificates get rolled over once a year when the CA's renewal fee is due

- Is the key in the new certificate the same as the old one?
 - Sites typically re-certify the same key year in, year out
 - The few CDNs that play games with cert changes can be handled through whitelisting
- Was the certificate issued by the same CA as the previous one?
- Does the combination of CA and site location make sense?
 - Did a CA in Brazil issue a certificate for a site in France?

Key continuity

Example: Comodogate

- Sudden change of CA for Google, Microsoft, Mozilla, Skype, and Yahoo is highly suspicious
 - (Not to mention their relocation to an ISP in Iran)
- Even the most trivial sanity check by browsers would have caught this

Example: Diginotar

- See “Comodogate”
- Browser vendors did nothing
- “Please sir, can I have some more?”

Key continuity

Simple presence of a certificate can still act as a risk-mitigation measure

- *Not* a boolean pass/fail measure
- Moves the the risk balance somewhat further towards “less risky”

Allows value to be assigned to different CAs and certificate types

- Standard vs. EV certificates
 - Not the ineffective blue vs. green colour bar
- Trustworthy vs. negligent CAs

Key continuity

This is a serious problem with the current boolean pass/fail mechanism

- Most diligent CA is treated no differently from the most negligent one
- No value in being careful, you get the padlock picture either way
- Incentivises the race to the bottom

Key continuity

Deals with the CA too-big-to-fail problem

- Once your root is in the browser, you'll never be removed no matter what you do
- (With a mere 700 server certificates, DigiNotar were small enough to fail)

Negligent CAs can be downgraded to a high-risk category

- Mirrors downgrading of risky financial ventures by ratings agencies
 - Until the skewing of the market during the 2008 financial crisis these had actually been a good indicator of a venture's credit risk

Key continuity

Current pass/fail measure rewards negligent CAs

- The less checking the CA does, the lower its operating costs

Risk-evaluation based system rewards diligent CAs

- Assigns them a lower risk rating than less diligent ones

Browsing History

History of interaction with a site over time

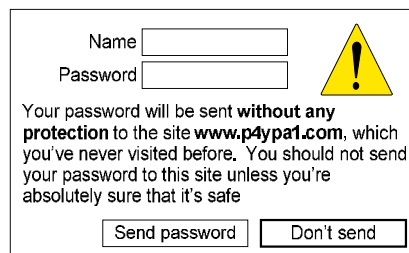


A screenshot of a web browser's password prompt. It features two input fields labeled 'Name' and 'Password'. To the right of these fields is a circular button with the text 'OK'. Below the input fields, a message states: 'Your password will be sent **securely encrypted** to the site **www.paypal.com**, which you've used 25 times in the past'. At the bottom of the dialog are two buttons: 'Send password' and 'Don't send'.

- Browser changes its behaviour depending on whether this is
 - First visit to a site
 - An infrequent visit
 - A frequent visit

Browsing History

User frequently visits `https://www.paypal.com`



A screenshot of a web browser's password prompt, similar to the one above but with a warning icon. It features two input fields labeled 'Name' and 'Password'. To the right of these fields is a yellow triangular warning icon with a black exclamation mark. Below the input fields, a message states: 'Your password will be sent **without any protection** to the site **www.p4ypa1.com**, which you've never visited before. You should not send your password to this site unless you're absolutely sure that it's safe'. At the bottom of the dialog are two buttons: 'Send password' and 'Don't send'.

- Is now visiting `https://www.p4ypa1.com` for the first time
- Browser increases the risk level associated with the site

Just this basic measure has been shown to significantly increase a user's ability to detect spoofed sites

Background Information Checks

Use hosting information for risk assessment

- Many larger and better-known sites are hosted/run by network operators with their own autonomous systems (ASes)



Background Information Checks

Check AS information for Société Générale

- Name is “SOCIETE GENERALE”
- Located in the EU
- Allocated in 1995
- Moves risk assessment towards “safe”

AS is in eastern Europe, or allocated yesterday

- Moves risk assessment towards “unsafe”

Background Information Checks

Check TTL for DNS records

- Very short TTLs are associated with fast-flux botnets
- Akamai, Google sometimes use short TTLs
 - Can be managed with whitelists for these providers

Check domain's registration date via a whois lookup

- Some registries already provide RESTful interfaces for this and other lookup types

Check whether A, MX, and/or NS records share the same IP prefix or AS

- Web site and mail server hosted in the same data centre
- Takedown-resistant “sites” are typically spread all over the place

Background Information Checks

Use stack fingerprinting to check the remote OS

- You're already communicating with the remote system's stack anyway

Phishing sites are frequently hosted on botnets via reverse proxies and other tricks

- E-commerce site running Windows 7 Home Premium
- Moves risk assessment towards “unsafe”

Site is hosted in an IP address block assigned to a DSL or cable modem provider

- Moves risk assessment towards “unsafe”

Site Content Checks

Use statistical classification techniques

- Uses text-mining feature extraction to get site-specific keywords
- Submit to Google as a search query
- A hit in the first few results means that it's probably OK
 - “Does my view of the site correspond to a third party's view?”
- Evaluated on 100 phishing and 100 legitimate pages
 - Detected 97% of phishing pages
 - 10% false positive rate

Site Content Checks

Use the Wayback Machine

- Was this site completely different six months ago?

Visit the site with user agent = Google/Googlebot vs. user agent = MSIE

- Some sites serve up benign content for Google's scanner, malicious content for MSIE to evade detection
- This is so prevalent that it's nearly destroyed the effectiveness of some web-site scanning initiatives

Check for odd characters in the URL for homonym attacks

- Generic .com site with a Cyrillic letter in the middle of the domain name

Site Content Checks

Use web page classification to try and detect signs of malicious sites

- Lots and lots of work done in this area
- Used by search providers to scan for malicious sites

Some features found to be effective in practice include...

- Number of elements like `div`, `iframe`, and `object` with small areas
 - Used to hide web bugs and carry out drive-by downloads and other malicious activities
- Number of elements with suspicious content like potential shellcode

...continues...

Site Content Checks

...continued...

- Number of overtly suspicious items like `object` tags containing classids of exploitable ActiveX controls
- Number of objects pulled in from other locations via `script`, `iframe`, `frame`, `embed`, `form`, or `object` tags
- Number of out-of-place elements
 - Typically caused by XSS injecting scripts or iframes into odd locations
- Presence of double documents
 - Multiple `html`, `head`, `title` or `body` elements
 - Side-effect of some types of web-site compromise

...continues...

Site Content Checks

...continued...

- Presence of malicious patterns
 - Things like meta refresh pointing to an exploit server
- Various known malicious patterns like `swfNode.php` and `pdfNode.php`
 - Commonly used by exploit toolkits
 - Will eventually be evaded by toolkit authors, but it doesn't hurt to check anyway

Site Content Checks

Other less obvious features have been discovered using Bayesian classifiers and support vector machines

- Can automatically extract new, non-obvious features from data sets

Presence of the string “members” in the URL

- Triggered by the presence of phishing and malware sites on free hosting services like `members.aol.com` and `members.tripod.com`

Owners of IP ranges for DNS records

- Registrars like RIPE are less likely to register malicious sites
- Registrars like GoDaddy are more likely to register malicious sites

Site Content Checks

Checking Javascript is particularly effective

- Obviously malicious code is easily detected
- Obfuscated code looks like no normal Javascript

Signs of malicious Javascript

- Keyword-to-word ratio
 - Number of keywords like `var`, `for`, `while`, and others is limited in comparison to operations like instantiation, arithmetic operations, and function calls in exploits
- Number of long strings and their entropy
 - Typically used to obscure payloads

...continues...

Site Content Checks

...continued...

- Presence of classes of function calls like `eval()`, `substring()`, and `fromCharCode()`
 - Widely used for de-obfuscation and decryption
- Length of strings passed to functions like `eval()`
 - Another de-obfuscation indicator
- Number of string assignments
 - Another characteristic of de-obfuscation/decryption procedures
- Number of bytes allocated through string operations like assignments, `concat()`, and `substring()`
 - Used for heap exploits and heap spraying

...continues...

Site Content Checks

...continued...

- Number of DOM-modifying functions
 - Operations like `document.write` and `document.createElement`
 - Typically used to instantiate vulnerable components or create page elements used to load scripts and exploit pages
- Values of attributes and parameters in method calls
 - Long strings are typically used for buffer-overflow attacks
- Number of event attachments like page load triggers
 - Used for drive-by downloads
- Presence of iframe-injection code or code to inject other objects or scripts into a page

Site Content Checks

What about an anti-virus style arms race?

- Not so easy for malicious web sites

Don't have complete control over the output of XSS, SQL injection, or similar injection attacks

- Produces malformed strings, repeated or out-of-place tags, and other danger-sign patterns
- Already used by tools like NoScript to try and mitigate attacks

Site Content Checks

Obfuscating Javascript isn't so hard

- Artefacts of the obfuscation are rather harder to disguise
- Results in page content that's nothing like any legitimate web site
- Recall that many of the previous triggers were for de-obfuscation artefacts

Attackers can try and evade detection like native-code malware authors

- Anti-detection mechanisms themselves increase their detectability

Site Content Checks

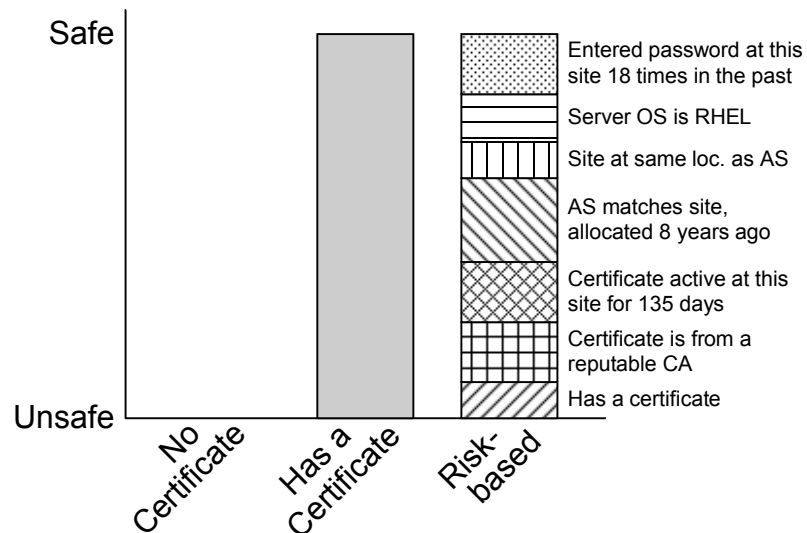
Some legitimate sites use obfuscated Javascript “to protect our IP”

- Translated: The boss fell for a scare marketing campaign

Malware vendors have reported some success in discouraging this

- “Giving your site the characteristics of a site run by the Russian mafia is not a good idea”
 - Note the careful lawyer-proof wording

Risk-Assessment Value Proposition



Risk-Assessment Value Proposition

Total of all of these measures is well over 100%

Illustrates the fault-tolerant nature of the assessment

- Even if some of the indicators aren't available you can still be reasonably confident that the site is safe to use

In reliability engineering this is called de-rating

- Run a device at less than its rated maximum level to provide a safety margin

Risk-Assessment Value Proposition

Only one mechanism, in only one browser, currently does anything like this

- SmartScreen filter in IE9
- Not to be confused with the basic site blacklist of the same name

Checks the “reputation” of a download

- “Reputation” is somewhat vaguely defined by Microsoft

Reduces warnings from the generic, useless “This type of file may harm your computer” to one that only occurs with files without a “reputation”

- According to Microsoft, 90% of downloads no longer display (pointless) warnings

Risk-Assessment Value Proposition

Mechanism takes effect immediately without the delay inherent in blacklists

- In one attack SmartScreen blocked 99% of downloads of malware from the moment that it was released
- Infection rates peaked 4-5 hours later
- 6-7 hours after *that*, the first blacklist-based blocking finally began to kick in

Unfortunately, no proper evaluation of its effectiveness

- Simply blocking all downloads would be 100% effective in blocking malware
- Still, better than the nothing that other browsers are doing

But What About...

But if we do this, our users will move to another browser!

- We haven't actually asked any of them, but we're absolutely certain they will

Actual surveys have shown that users expect vendors/ISPs to take steps to proactively protect them

- This follows expectations set by standard industry and consumer-protection legislation practice

But What About...

Problem has been evaluated in practice by ISPs implementing walled gardens

- *Increased* rather than decreased user confidence in and loyalty to the ISP
- “They care about my security (unlike everyone else)”

The evidence we have is that it increases, rather than decreases, user satisfaction in a product or service

But What About...

The browser vendors aren't interested, what about...

- Plugins
 - Require poking around in way too much of a browser's internals
 - For the users most at risk, plugins don't exist
- Local proxy on the user's PC
 - Need to know that the proxy exists, download, install, reconfigure the browser
- ISP-level proxies
 - Bad guys can't easily subvert these because they're indistinguishable (if done right) from end-user PCs

But What About...

This will affect performance!

- Use risk-assessment heuristics to help decide which sites need the extra checking
- The checks are overkill if applied to all sites

Apply standard performance-optimisation techniques for the few sites that do need the checks

- Run checks in background threads while the current pages is being loaded/rendered
- Perform opportunistic lookups for links on the current page
- Cache results from previous lookups
- ...

But What About...

“Is the site asking for user credentials (typically a password)?”

- Check for strings like `login`, `signin`, `secure`, `bank` and `account` in the page URL
- Check for situation-specific ones like `webscr` (PayPal) and `ebayisapi` (eBay)

Goal of phishing is to acquire credentials

- Any site that tries to obtain credentials should be subject to extra checking

But What About...

Check page contents for keywords or other indicators of a high-value site

- Impersonating a bank is more or less impossible without including lots of banking-related keywords on the site
 - Go to your bank’s home page to see an example of this
- Phishers try and disguise this using image text
 - A site that consists mostly of images and asks for a password is another danger sign

But What About...

Consult a database of common phishing targets to identify the need for extra checking

- The bad guys have been doing this for years
- Banking trojans contain built-in databases specifying attack behaviour for target sites
- Pull a target list from any banking trojan

Whats Good for the Goose...

Diversifying security mechanisms for risk management works for CAs as well

- In Comodogate, any one of a series of extremely rudimentary checks would have caught the problem
- Certificates for well-known, high-profile US sites requested from Iran
- Certificates issued by completely different CAs were replaced by Comodo ones
- EV certificates replaced by non-EV certificates

These anomalies should have triggered every alarm that it's possible to trigger

Whats Good for the Goose...

I'd like to swap my EV certificate...



Whats Good for the Goose...

... for a DV certificate



- Either there's a body in the boot/trunk of the Mercedes or a Picasso in the boot/trunk of the 2CV

Whats Good for the Goose...

Simple automated check for CAs

- Open an HTTPS connection to the target site
- Do they already have a certificate from another CA?
- Is the certificate nowhere near its expiry date?
- Are EV certs being replaced by non-EV ones?

Counter-check for a legitimate cert-replacement

- Is the certificate that's being replaced listed on a CRL?

Whats Good for the Goose...

Moves the risk level to “high-risk”

- Triggers manual review and a requirement for more comprehensive authentication

Email the domain holder for a chance to object

- Fully automated process
- c.f. mailing-list sign-up checks

UI Issues

How do you notify the user?



- I know, let's pop up a warning message!

UI Issues

For users a site's appearance will override all security indicators and browser-supplied phishing warnings

- "It looks like a bank, so it is a bank"

Lots and lots of research by anti-phishing folks into this effect

- No known security indicator or mechanism can override this
- "I can see my bank's site in the browser, the security warning must be a false positive, I'll disable it"

UI Issues

Turn a bug into a feature

- Change a site's appearance if it registers as being high-risk

Disable Javascript and plugins

- A wise precaution when visiting a high-risk site anyway
- Most major US bank sites don't function without Javascript
- Some simply display blank home pages

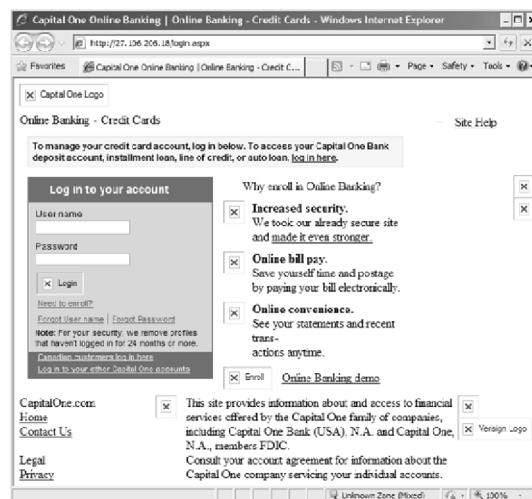
Block images

- Another good precaution given malformed-image attacks

Render the site in the default font

- No boldface headings, different fonts and font sizes

UI Issues



- Overrides the standard “it looks OK so it must be OK” test applied by users

UI Issues



- What the user would be expecting...

UI Issues

Now you've really got the user's attention

- Provide further information to explain why the site looks the way it does

Reverses the standard situation

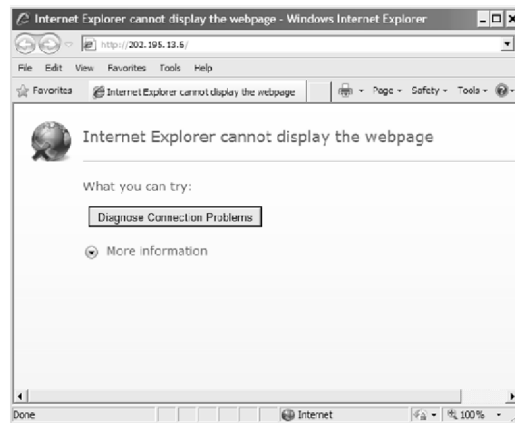
- User can see their goal and anything you do is an impediment to reaching it

Now the user can see that there's a problem

- Looking for an explanation

UI Issues

If the site registers high enough on the danger scale...



- Treat an inability to provide a safe connection as an inability to provide any connection

UI Issues

“We’ve detected that this site is a phishing site loaded with malware”

- But we’ll allow you to go there anyway after ignoring this message from the sponsor

The user wants to connect to a site to perform a secure, safe transaction

- If they’re being led to an obviously unsafe site then this should be treated identically to a standard network problem

(Obviously this is only for sites that read highly on the danger scale)

Conclusion

Browsers currently do *nothing* to protect users from malicious web sites

- Browser PKI is pure security theatre, indistinguishable from placebo by any measure that we know of
- Site blacklists don't work

Apply standard risk-management measures to protect users

- A concept built on centuries of real-world work
- Every measure presented here has been evaluated in real-world testing or real-world conditions

This is not rocket science

- It's so simple that it's almost embarrassing having to mention it