

Digital Certificates Fails and Flounders

Jeff Krakenberg

jeffkraken


@abandonfreewifi

Hello, my name is Jeff.

I'm a bit of a cyber goblin, I work as a technical trainer, consultant, and researcher.
Also as an author, game designer, and Dungeons & Dragons nerd.


If you're online, you can find me as either "jeffkraken", "abandonfreewifi", or by name.

What does this talk cover?

1) Digital Certificates (Overview) 

2) Fails  + 

3) Flounders 

4) Digital Certificates (Wrap-up) 



What is a Digital Certificate?



A special type of file that identifies a resource, cryptographically.

There is a really common analogy out there that your digital certificate is like your driver's license. I'm not the biggest fan of that analogy.

I like to think of them as (see slide).

I like that term "resource" because a digital certificate can identify a bunch of stuff. Computers, users, servers, printers, access points, firewalls, et al.

You can even use a Digital Certificate to trick Active Directory into thinking you're a printer.

As an example, you can think of it like a "name tag".

You sign your name and then someone else (like GoDaddy.com) signs that you're correctly identifying yourself.

And yes, if you're not aware, the company that had those incredibly questionable ads through the 2010's is a player in the game of verifying website trust.

Why does that matter?



This matters because when you access a website: your browser is referring you to a server, that server is giving you the site. In accessing the site, you are trusting that the server is who they say they are... If not directly, mostly indirectly, by trusting the digital certificate.

If the certificate is incorrect, invalid, or malicious, you could be unknowingly connecting to a fra

Personally, I access a lot of remote machines.

Most of them are legitimate, and the configurations are saved.

Some of them, though, are VMs that only live for a few hours before they've finished their job.

Sometimes, I'm only PRETTY sure that "this" is my server.

Why does that matter?



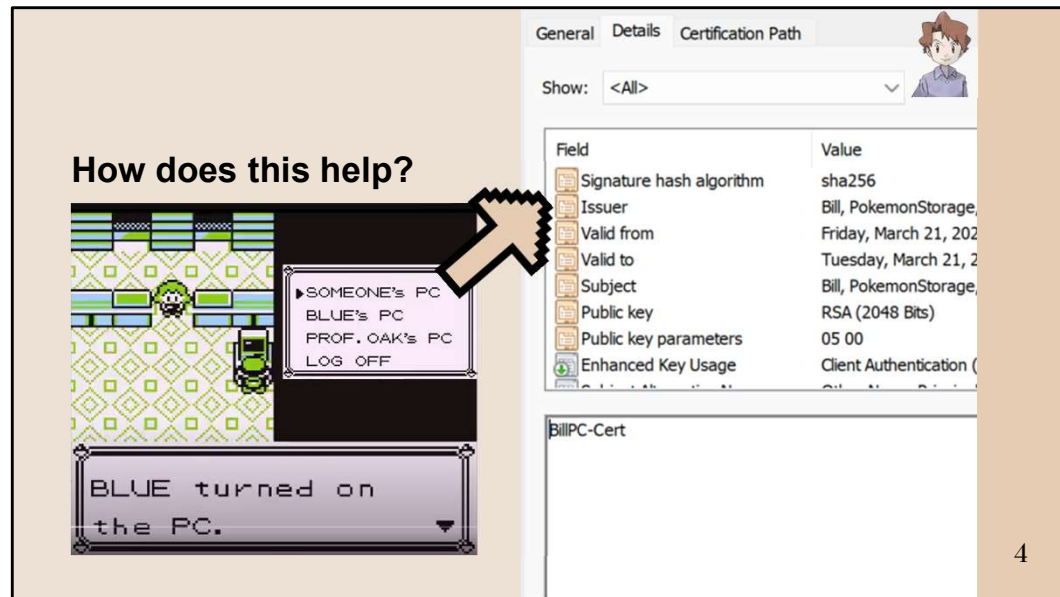
Digital
Certificate:
Mr. Pigeon

Jeff realizes their mistake and says:
"Oh, sorry, Mr. Pigeon."

A digital certificate is one way to quickly double-check ownership of the resource.

Your resources should have your certificates.

My resources should have my certificates.



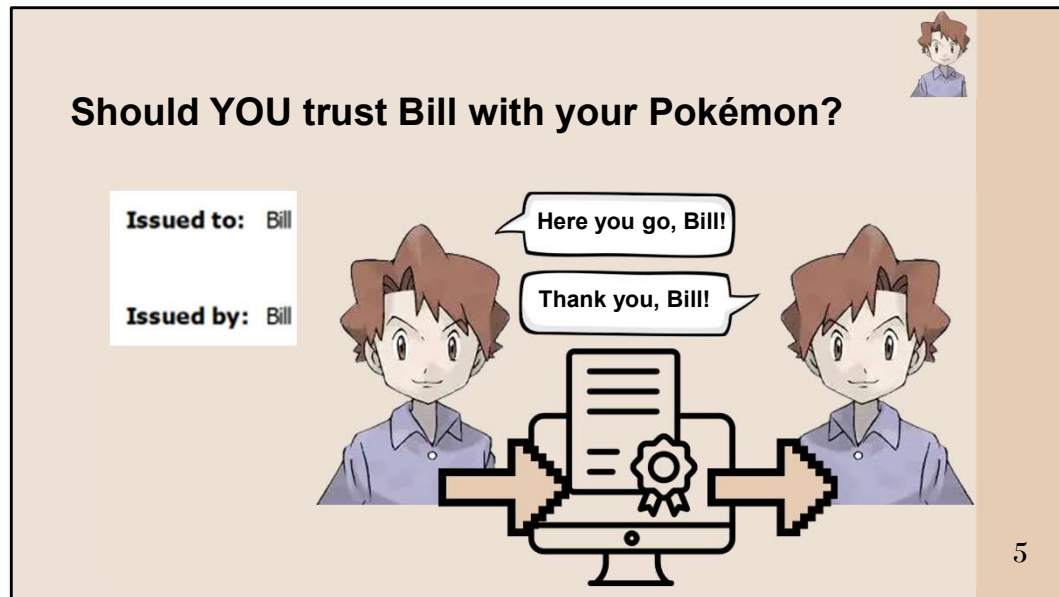
Pull from Pokemon knowledge:

When your inventory is running full and you need to send your excess Pokémon to storage, you might need to access SOMEONE's PC.

I don't know about y'all but I don't trust "SOMEONE" with my pocket monsters.

In this case: a digital certificate would help identify that SOMEONE as Bill, so that you know your Pokémon are... somehow... going to be stored in boxes in Bill's PC.

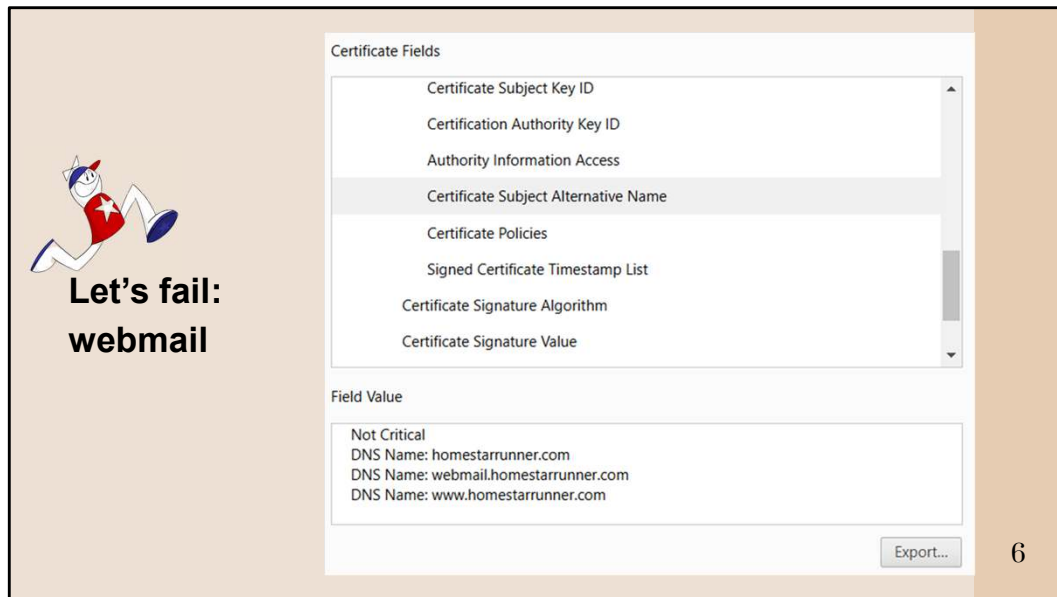
That way, later, you can log back in, get your bonus Pikachu from Bill's PC, and force evolve it into an angry electric cat-mouse.



Regardless of your thoughts on Bill and his...
habits...
he is a Pokémon expert.

Though, in this case, there isn't an overarching body that hands out certificates.
So Bill has to create the certificate for Bill's PC.

And Self-Signed Certificates must always be trusted manually...
that means that unless you script something semi-malicious on a redirect,
the user is going to get an "website insecure" or "website untrusted" warning
that they'll probably ignore anyway or assume its the end of the world*.

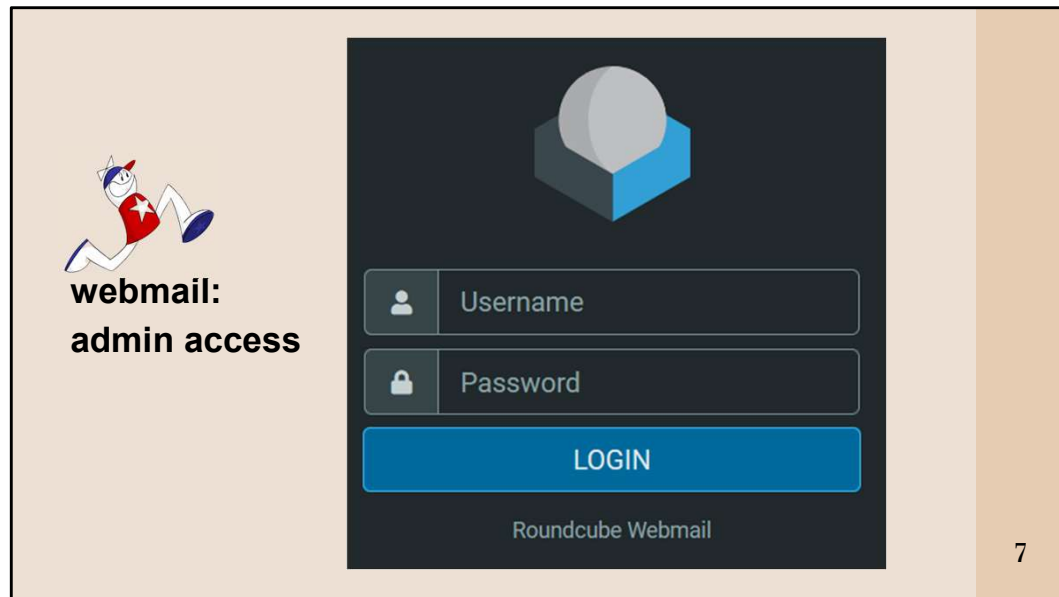


During a discussion on digital certificates,
I asked for an example of a school/workplace appropriate website
and I was given homestarrunner.com.

Upon opening up the cert, we can see that the SAN
(subject alternative name = way to record DNS naming info) has more than one entry.

It has the standard website, the www., but then it also has a webmail.

To clarify, I'm not saying that having your webmail
on the same cert/site as your page is a vulnerability...
however, when you can find it by cert-scanning and that leads you to...



A vulnerable log-in, now it's something that we should talk about.

(side note: this was reported back in 2023 and again in 2024)

After reporting and getting the okay-to-poke we found:
authentication via defaults
multiple internal mailboxes



**webmail:
sessions
+ libraries**

Content-Language: en
Set-Cookie: roundcube_sessid=qk8il2018ap6h0vga99goa5p33; path=/; secure; HttpOnly

Input Vector:

Description:
The identified library bootstrap, version 4.5.3 is vulnerable.

Other Info:
CVE-2024-6531

Solution:
Please upgrade to the latest version of bootstrap.

8

After some tooling (ZAP)
we found weak session cookies and a known vulnerable library.

I would consider this the creamy center of a double layer cake.
It wasn't the best find, but it was a good inclusion.



**webmail:
alt port
+
redirect**

```
SF-Port8080-TCP:V=7.94SVNI=7%D=2/20Time=67B7B61A%P=x86_64-pc-linux-gnu\r
SF:(GetRequest,37D,\"HTTP/1.1\\x20200\\x200K\\r\\naccept-ch:\\x20Sec-CH-UA,\\x20
SF:Sec-CH-UA-Platform,\\x20Sec-CH-UA-Platform-Version,\\x20Sec-CH-UA-Mobile\\
SF:r\\ncache-control:\\x20max-age=0,\\x20private,\\x20must-revalidate\\r\\nconne
SF:ction:\\x20close\\r\\ncontent-length:\\x20461\\r\\ncontent-type:\\x20text/html
SF:;\\x20charset=utf-8\\r\\ndate:\\x20Thu,\\x2020\\x20Feb\\x202025\\x2023:09:14\\x2
SF:0GMT\\r\\nserver:\\x20nginx\\r\\nset-cookie:\\x20sid=b3eb1774-efdf-11ef-b08d-
SF:d2054870da89;\\x20path=/;\\x20domain=.;\\x20expires=Wed,\\x2011\\x20Mar\\x20
SF:2093\\x202:23:21\\x20GMT;\\x20max-age=2147483647;\\x20HttpOnly\\r\\n\\r\\n<htm
SF:l><head><title>Loading\\.\\.\\.</title></head><body><script\\x20type='text/
SF:javascript'>window\\.location\\.replace\\('http://'?ch=16js=eyJhbGciOiJIU
SF:ZiI6IiwiaW50eC161kpXVCJ9YyByJHdWQ101JKb2tlbiIsImV4cCI6MTc0MDEwMDE1NCwiaW
SF:F0IjoxNzQwMDkyOTU0LCJpc0MiOiJKb2tlbiIsImpzIjoxLCJqdGkiOiIzMGowcWtsOWJSM
SF:GpzNTRsYXNmNTduMjc1LCJmYiojE3NDAwOTI5NTQsInRzIjoxNzQwMDkyOTU0NzA1MTg3
SF:fQ\\x20VMaqqV3Zz_U8TI7u\\x20KnKHHM0v7onrhWCiL3_t8yH946sid=b3eb1774-efdf-11ef-
SF:b08d-d2054870da89'\\x20</script></body></html>\">;
```

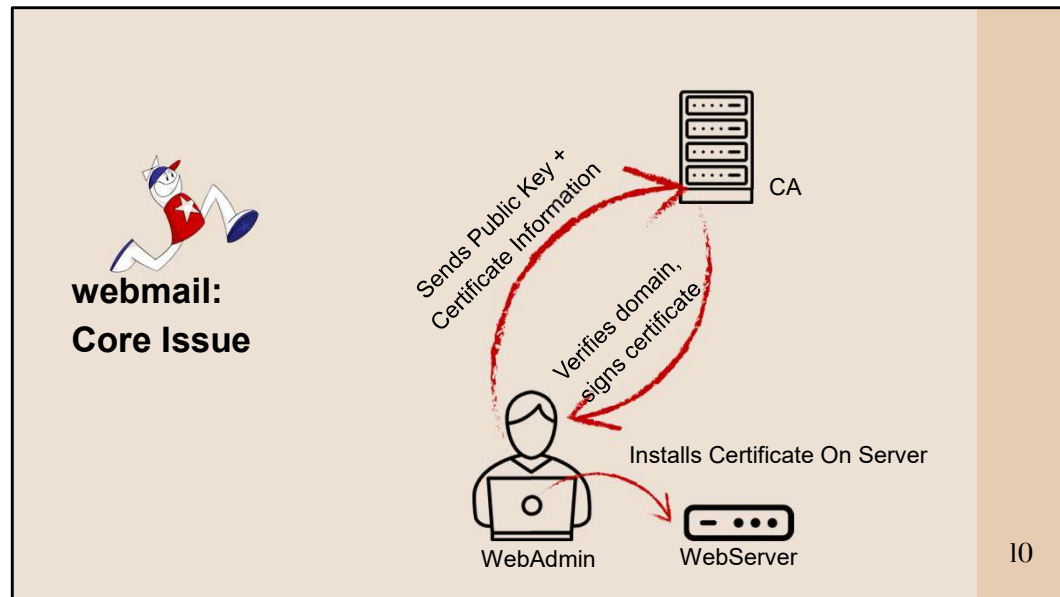
```
javascript'>window\\.location\\.replace\\('http://
```

9

The icing on the cake was a redirect.

Their port scan revealed a common alt http port
and looking at the fingerprint,
found a JavaScript redirection that uses
`window.location.replace()` method to redirect to another URL.

This could be maliciously altered as part of phishing or an Attacker-in-the-middle.



One issue here is a general misunderstanding of what a Certificate Authority (CA) does.

If you're requesting a certificate for your web-server, they're going to verify that you're the legitimate owner of the domain and that your info is formatted correctly.

They're not going to verify:

- to see if your exposing internal resources
- to see if your internal resources are vulnerable
- that you're a legitimate entity.

This exposes potential attack surfaces, especially if:

The webmail app is outdated or misconfigured. WHICH IT WAS

The login page is accessible to the public. WHICH IT IS

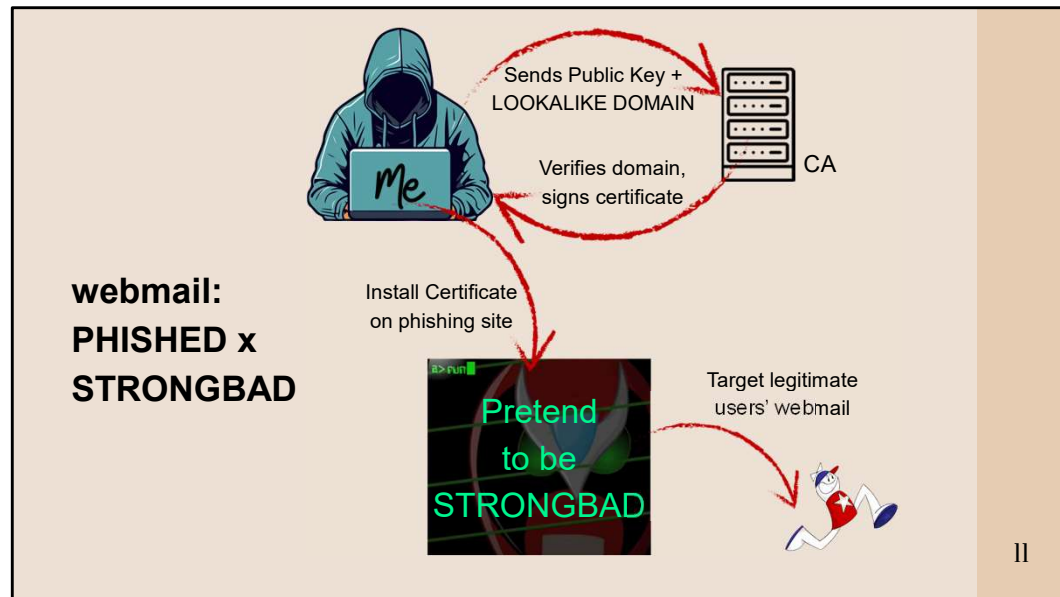
Brute force or phishing attacks could be attempted on email users. WHICH IT CAN.

The reason this comes up...

If you're working with a small site as the web admin,
you're going to send the certificate request in (or just click a box on your webhosting platform)
and then the CA is going to verify the domain and sign the certificate.

Then, you'll take the certificate from them and install it on the webserver.

The issue that comes up is that there is nothing here to verify that it IS good outside of just looking good.



webmail-homestarrunner.com instead of webmail.homestarrunner.com

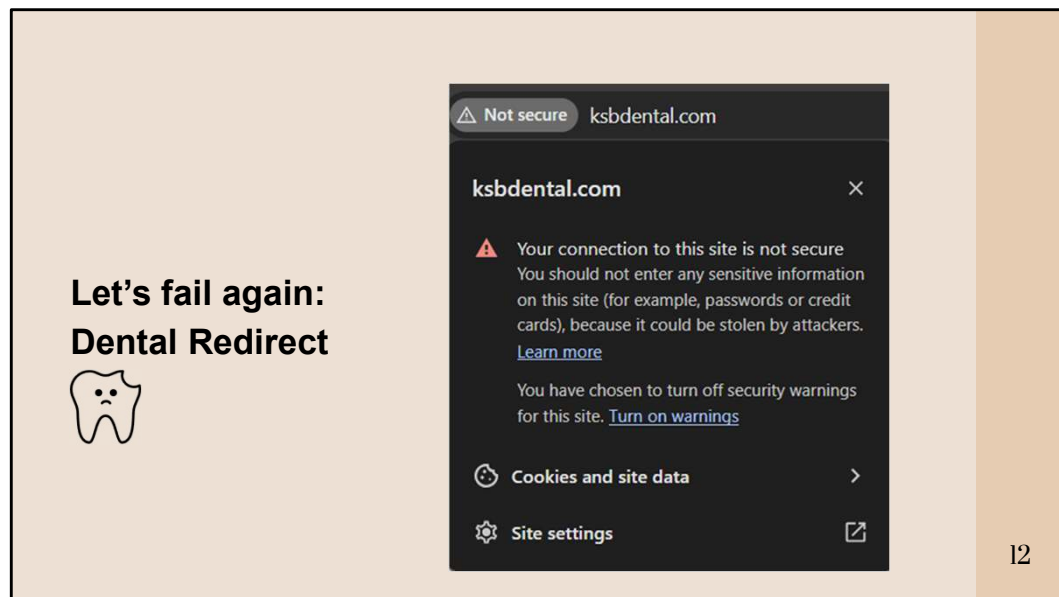
Let's Encrypt with spoofed info.

The CA is:

- Verifying the certificate request
- Verifying the domain registration

The CA is NOT:

- always verifying the legitimacy of the individual.

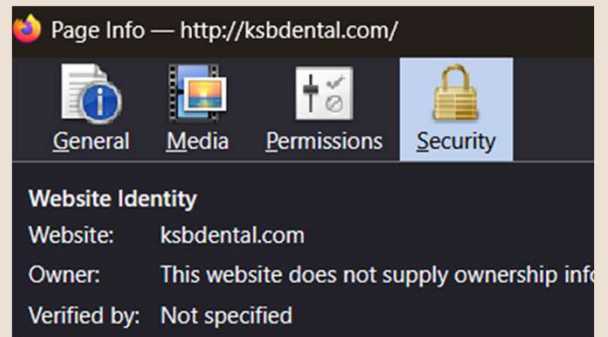


Software/support company. I've known two people that worked there, I told both to pass this along since my emails were getting "return to sender"-ed.

Initial concerns were with some forms on their site. Those forms are no longer "publicly accessible".

Reported in 2022 and 2025.

Dental Redirect: HTTP on Main?



The site was not downgrading or redirecting, it was just responding via HTTP, not HTTPS.


Dental Redirect:
If at first you
don't HTTPS, try
HTTPS again.



```
https://ksbdental.com/  
The certificate is not trusted because it is self-signed.  
HTTP Strict Transport Security: false  
HTTP Public Key Pinning: false  
Certificate chain:
```

If you force HTTPS, you will find a legitimate site
and you'll be greeted by a warning because
just like Bill's PC from earlier... this certificate is self-signed.

However, HSTS is off so it's not going to force users through HTTPS
allowing most folks to end up on the HTTP site instead.



Username:

Password:

Login Reset

```
<input type="hidden" value="/sslvpn_web_logon" name="action">
<input type="hidden" value="logon" name="fw_logon_type">
<input type="hidden" value="false"
  " id="accept">
***
<script language="JavaScript">
document.user_auth_form.fw_username.focus(); </script> == $0
<input name="lang" type="hidden" value="en-US">
```

top

ksbdental.com

auth_portal/Default

images


scripts

general.js

style

sslvpn_logon.shtml

```
216 // MEMBER management APIs (for address-like)
217
218 /*****
219 *
220 * Begin dynamicTableObj
221
222 function M_makeMember() {
223   return arguments;
224 }
225
226 function M_fillMember()
```

Dental Redirect:
What the
watchguard?


15

After accepting the trust,

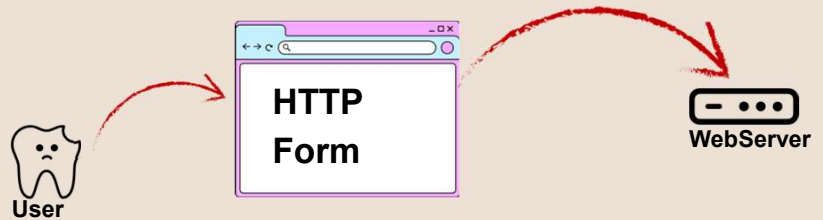
Redirected to /sslvpn_logon.shtml

insecure implementation of a mostly secure system WatchGuard.

I'm not here to talk about WatchGuard

but the original version found had several functions exposed without proper validation.

Dental Redirect: Core Issue



15

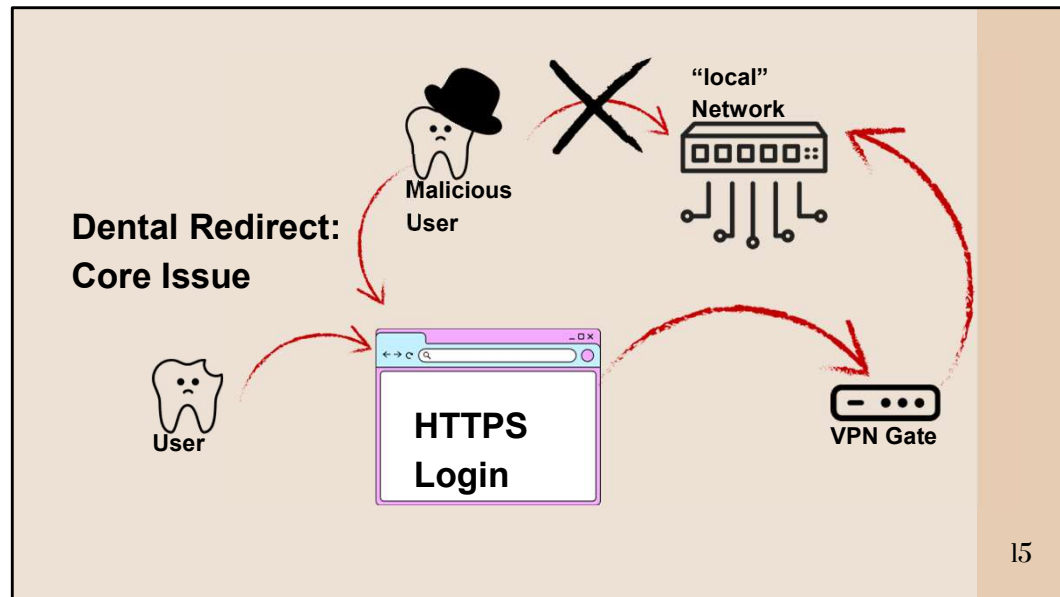
Core Issue:

Users were submitting data to forms
(most of these forms have been removed).

That data is sent in clear text to the server.

This is a old security issue that I'm always shocked to still run into.
Sites using http without the s.

Information sent in plain text included customer name, address,
contact information, comments/requests, and account info.



Core Issue:

Second issue here is when you consider unintentional or intentional malicious users.

Exposing your VPN login through your cert is similar to exposing your webmail site.

Not inherently an issue, but if you've got other things going on...
It will lead to an attacker gaining access to your "local" network.

Malicious user up at the top. As I always say,
we can tell he's malicious because he's got a hat on and he's unhappy.

The HTTPS site represents an unmanaged
or mismanaged attack vector for the malicious user in this case.

They can't access the network directly, but when they break through the login,

the VPN Gateway will put them right on that local network.

Another issue in this case, any "security focused" users would be redirected to the VPN login instead of the companies main page because of browser security settings.

Even though the server wasn't forcing https, the client could still be forcing it. Leaving them with the "wrong" page.

Let's flounder: Firewall Misconfig



Certificate Formats

+

Puppet Configurations

16

Now I don't have any images (grainy or appealing)
for this one since it was internal to a company that I worked for...
but I can talk about it because I know the person
who did the misconfiguration and I stripped the "identifying" details out.

P.S. It was me.

Firewall Misconfig: Cert Format



- | | |
|---------------------|-----------------------------------|
| • Linux Certs | • Windows Certs |
| • File Format: .crt | • File Format: .cer |
| • Fix: openssl: | • Fix: Powershell: |
| openssl x509 \ | \$cert = get-childitem "cert.crt" |
| -inform PEM \ | export-certificate ` |
| -in "cert.cer" \ | -cert \$cert ` |
| -out cert.crt | -filepath "cert.cer" |

17

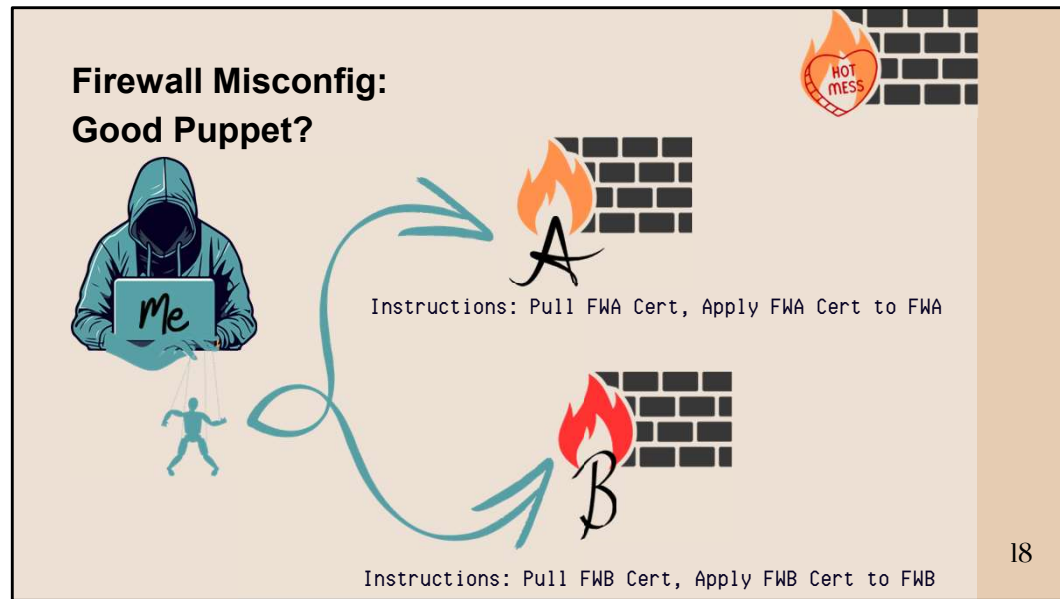
Managing systems and certs can be annoying in a mixed environment.
I dealt with both Windows and Linux systems.

Unfortunately, I was not the one to make the certs,
just the one to request and upload them.

So I would regularly get the wrong format.

There are a lot of work arounds on this one.
You can use OpenSSL or PowerShell to force a conversion.
Or there are plenty of other ways to get it to work.

Or you can do what our SOP said and "tell the OPs guy to redo the cert". (shrug)



This only really became an issue when we were trying
(and mostly failing to automate the system management with puppet).

We had several sets of firewalls and servers but for simplicity
I'm going to say that we had 2. Firewall A and Firewall B.

We wanted to write up a deployment that could quickly
pull and apply new certs at scale.

Firewall Misconfig: Bad Puppet!



```
file { ["/etc/ssl/certs/${Firewall_A}-cert.crt":  
source => "puppet:///modules/firewall/certs/${firewall::certificates[${Firewall_A}]]",  
mode => '0644', }  
if $::hostname == $Firewall_B {  
file { ["/etc/ssl/certs/${::hostname}-cert.crt":  
source => "puppet:///modules/firewall/certs/${firewall::certificates[${::hostname}]]",  
mode => '0644', } }
```

19

Someone failed to update the source when
doing a little copy/pasta from FirewallA to FirewallB.

These chunks of code do not need to be here.
It was left over from when the first Firewall was being deployed.

They tested it on one, it worked perfectly fine, so they moved on.

So we've got highlighted portions here that I like to refer to as:

"Redundant"

"Bad"

and "Kill it"

Firewall Misconfig: Key Point



EXPIRED



Current: 398 Days



March '26: 200 Days



March '29: 47 Days

As a quick update for those that haven't seen it yet,
there is a potential major shift on the horizon
with the cert lifespan shrinking to almost 10% of the current lifespan in four years.

This means that if you're not considering shorter certs and automation already, ooof.

I'm a fan of the shorter lifespans and I think that they
will solve a lot of the issues that we run into currently if adopted.
Companies definitely won't find a way to undermine that by
creating their own CA and then paying themselves for certs... Ahem... Google Trust Services... A

Firewall Misconfig: Key Point



Lifespans are shortening!

- ⚙ Automate
- 👤 Onboard
- ⏮ Alternate

What I would take away from this is that automation is coming for your certs, either that or you'll need to pay for a dedicated certificate-person or switch to something like DANE (and rely on DNS because that NEVER breaks lol).

How should we think of Digital Certificates?



- Driver's License? Close.
 - User provides information to the CA.
 - CA verifies and signs info into a certificate.
 - CA provides the User with the Digital Certificate.
- Land Deed? Closer.
 - Not "really" real.
 - Only trusted if you trust the signatory.

20

Earlier I mentioned: "Cryptographic proof of resource ownership" but that is too wordy and too technical.

You provide information to the CA, the CA then signs and returns that information as a file, and you display that file to prove ownership of the file.

While I can see the similarities to a Driver's License, I feel like it's more in common with something else.

Land Deed? Closer.

Mostly because I'm a fan of space fantasy westerns. Though, 'Land Deeds' aren't "really" real and they only mean something if you trust the person who signed it. So that is a little closer...

The best analogy for a Digital Certificate is...



- Something that no one ever really asks for.
 - unless everything else is going wrong.
- Something that proves you did THE simple thing.
- Something that is used for generic gatekeeping.

Any guesses???

- Bonus Point: Something easy to get online for a small fee or free.

A better analogy for a Digital Certificate is...

High School Diploma



22

I'll admit. As much as I hate the driver's license analogy, it is a pretty good one.

However, I haven't been able to find a case of a digital certificate being used to prove identity in a legal situation.

I have however seen entire industries dedicated to forging them, like how you can drop out and take a digital arts class at your local community center or library and fake your own High School Diploma.

DIGITAL CERTIFICATE The danger of High School Diploma “forgeries”?



Spoof attacks?



Third-Party Hosting?



Forced Trust?

23

What if they can just spoof that trust? Domain redirects and look-alike domains.

What if they can just register directly through a 3rd party?

Anyone else hate Shopify for this? That could be another whole talk.

<https://community.shopify.com/c/retail-and-point-of-sale/can-shopify-prevent-scams-through-their/td-p/1531032>

What if they can force the trust?

I saw an interesting attack recently where the hacker sent an .RDP file to the victim with a cert from LetsEncrypt attached to the resource.

It essentially forced the trust by having the user connect to the "certified" resource, which was just the attacker's machine.

<https://www.linkedin.com/feed/update/urn:li:activity:7319687538535800832/>

DIGITAL CERTIFICATE How to protect against High School Diploma “forgeries”?



Basics: OCSP + CRL



DNS Filter?



Automated Checks

24

The basics are in place and we can definitely debate the effectiveness of them. However properly configured OCSP (Online Certificate Status Protocol) and CRLs (Certificate Revocation Lists) can help.

However, both of these are reactionary, so they won't help protect against novel or unique attacks.

DNS Filters like AdGuard and Umbrella (Cisco) can be configured to block or alert when unknown domains come up.

There's also some tools like PhishTank and URLScan.io that can be used to check a domain's legitimacy that we will take a look at in a second. With these tools and whois, we can scripted a baby-API running like a fake neural network to "score" the certificate's likelihood. This has worked against spoofed or lookalike domains as well as the Forced Trust I mentioned on the last slide, but it doesn't scale easily.

PhishTank

```
def check_phishtank(domain):  
    payload = {'format': 'json', 'url': domain}  
    response = requests.post(PHISHTANK_API_URL, data=payload)  
    if response.status_code == 200:  
        data = response.json()  
        return data.get('results', {}).get('in_database', False)  
    return False
```

URLScan.io

```
def check_urlscan(domain):
    headers = {
        'API-Key': URLSCAN_API_KEY,
        'Content-Type': 'application/json'
    }
    data = {"url": domain}
    response = requests.post('https://urlscan.io/api/v1/scan/', headers=headers, json=data)

    if response.status_code == 200:
        scan_id = response.json().get('uuid')
        print(f"Submitted to URLScan.io: {scan_id}")
        result = requests.get(f"https://urlscan.io/api/v1/result/{scan_id}/")
        if result.status_code == 200:
            verdicts = result.json().get("verdicts", {})
            return verdicts
    return {}
```

Domain Age

```
def get_domain_age(domain):
    try:
        w = whois.whois(domain)
        creation_date = w.creation_date
        if isinstance(creation_date, list):
            creation_date = creation_date[0]
        if creation_date:
            age_days = (datetime.now() - creation_date).days
            return age_days
    except Exception as e:
        print(f"WHOIS lookup failed: {e}")
    return None
```

“Scoring”

```
if is_phish:
    score += 50
    reasons.append("PhishTank flagged this domain.")
if urlscan_data.get('overall', {}).get('malicious', False):
    score += 30
    reasons.append("URLScan.io considers the site suspicious.")
if domain_age is not None and domain_age < 30:
    score += 20
    reasons.append("Domain is very new (<30 days).")
```

CERTIFICATE TRANSPARENCY LOGS



What is a CT Log?



How can these help?



Example: CRT.SH

24

What CT logs are:

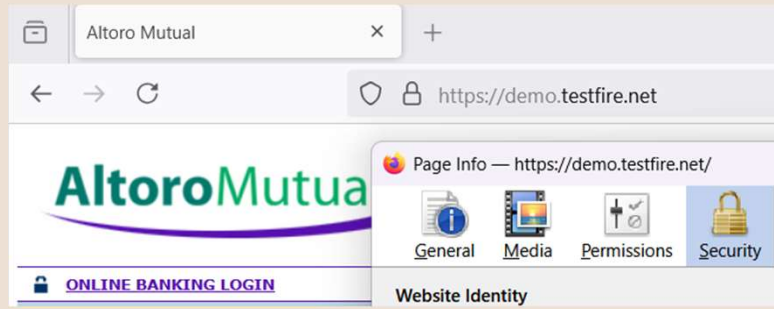
Public, auditable logs of all certificates issued by publicly trusted Certificate Authorities. CT logs are designed to detect and mitigate the kind of abuse that we've mentioned (lookalike/phishing).

Why they matter: They help detect mis-issued or rogue certificates early.

Real-world example: A malicious certificate for paypal.com spotted in a CT log before it could do damage.

Demo: Open demo.testfire.net, open certificate and get the fingerprint. Paste the fingerprint into the search of CRT.SH.

DEMO: CRT.SH - TESTFIRE



DEMO: CRT.SH - Certificate Fingerprint

Fingerprints

SHA-256

5F:30:FD:81:ED:CE:4D:9C:A2:B7:C1:6A:E3:B6:93:DA:38:27:36:00:03:92:79:BC:22:37:44:F7:12:3E:2A:4A

SHA-1

FF:A1:46:B7:99:91:C0:21:01:47:13:1D:8D:3E:A7:33:E0:38:F5:66

DEMO: CRT.SH - Certificate Search


crt.sh Certificate Search

Enter an **Identity** (Domain Name, Organization Name, etc),
a **Certificate Fingerprint** (SHA-1 or SHA-256) or a crt.sh ID:

Search

[Advanced...](#)

© Sectigo Limited 2015-2025. All rights reserved.



DEMO: CRT.SH - Certificate Search

Criteria

SHA-256(Certificate) = '5f30fd81edce4d9ca2b7c16ae3b693da38273600039279bc223744f7123e2a4a'

crt.sh ID

13272863561

Summary

Leaf certificate

Certificate Transparency

Log entries for this certificate:

Timestamp	Entry #	Log Operator	Log URL
2024-06-03 12:37:57 UTC	1257040	Let's Encrypt	https://oak.ct.letsencrypt.org/2025h2
2024-06-03 12:37:58 UTC	236022	Google	https://ct.googleapis.com/logs/eu1/xenon2025h2
2024-08-05 03:05:18 UTC	109488805	DigiCert	https://nessie2025.ct.digicert.com/log
2024-08-05 03:05:18 UTC	192374384	DigiCert	https://yeti2025.ct.digicert.com/log

Revocation

Report a problem with this certificate to the CA

Mechanism	Provider	Status	Revocation Date	Last Observed in CRL	Last Checked (Error)
OCSP	The CA	Check	?	n/a	?
CRL	The CA	Not Revoked	n/a	n/a	2025-04-30 19:30:02 UTC
CRLSet/Blocklist	Google	Not Revoked	n/a	n/a	n/a
disallowedcert.stl	Microsoft	Not Revoked	n/a	n/a	n/a
OneCRL	Mozilla	Not Revoked	n/a	n/a	n/a

Certificate Fingerprints

SHA-256

5F30FD81EDCE4D9CA2B7C16AE3B693DA38273600039279BC223744F7123E2A4A

SHA-1

FFA146B79991C

ASN.1

Certificate

Graph

Hierarchy

px

Certificate:

Data:

Version: 3 (Rx?)

Thank you for your time!



Scan to connect, get a
Digital “Digital
Certificate” Certificate,
or provide feedback.

Thank you for your time and massive thanks for letting me go off on this tangent.

If anyone is interested in connecting, I've got a QR code up that you can scan.
There's also a form on there if you want a silly
digital digital certificate certificate or if you want to provide feedback.

We do have some time for Q&A if anyone has any Q's that I can A.