HTTP Security Primer

Dominick Baier
http://leastprivilege.com
@leastprivilege

pluralsight
hardcore dev and IT training

Agenda

- Transport security
- X.509 Certificates
- Setting up TLS endpoints

- HTTP authentication framework

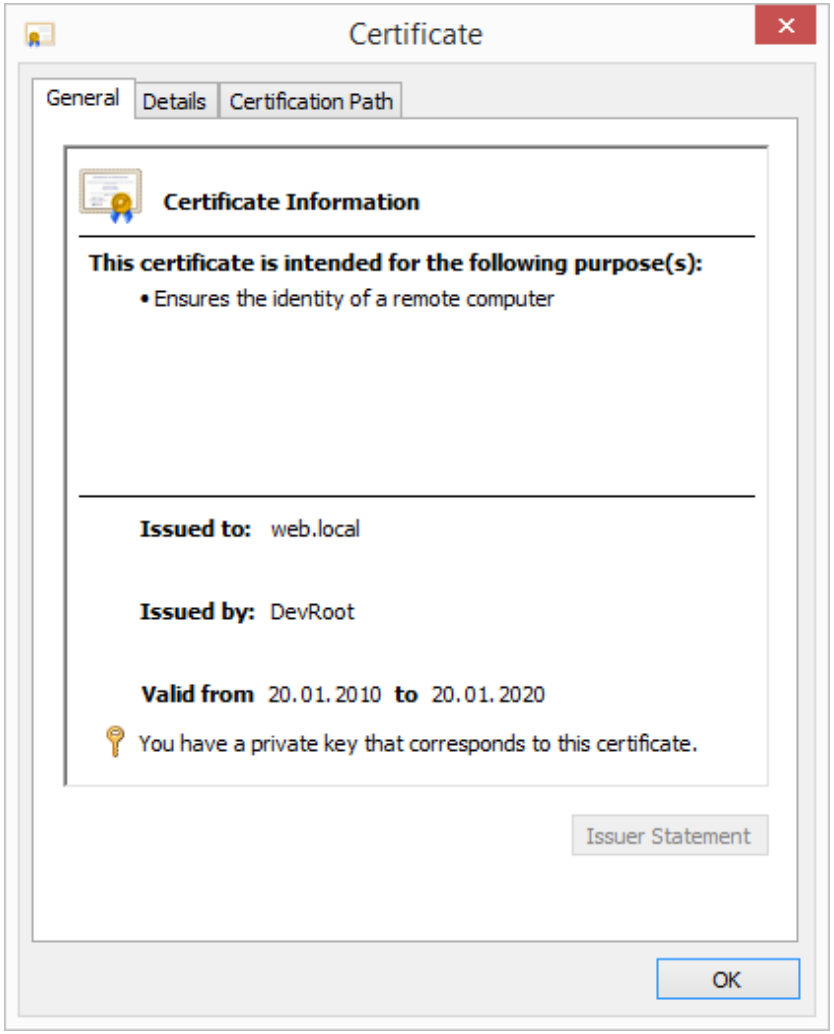
- APIs & Tools
- Resources

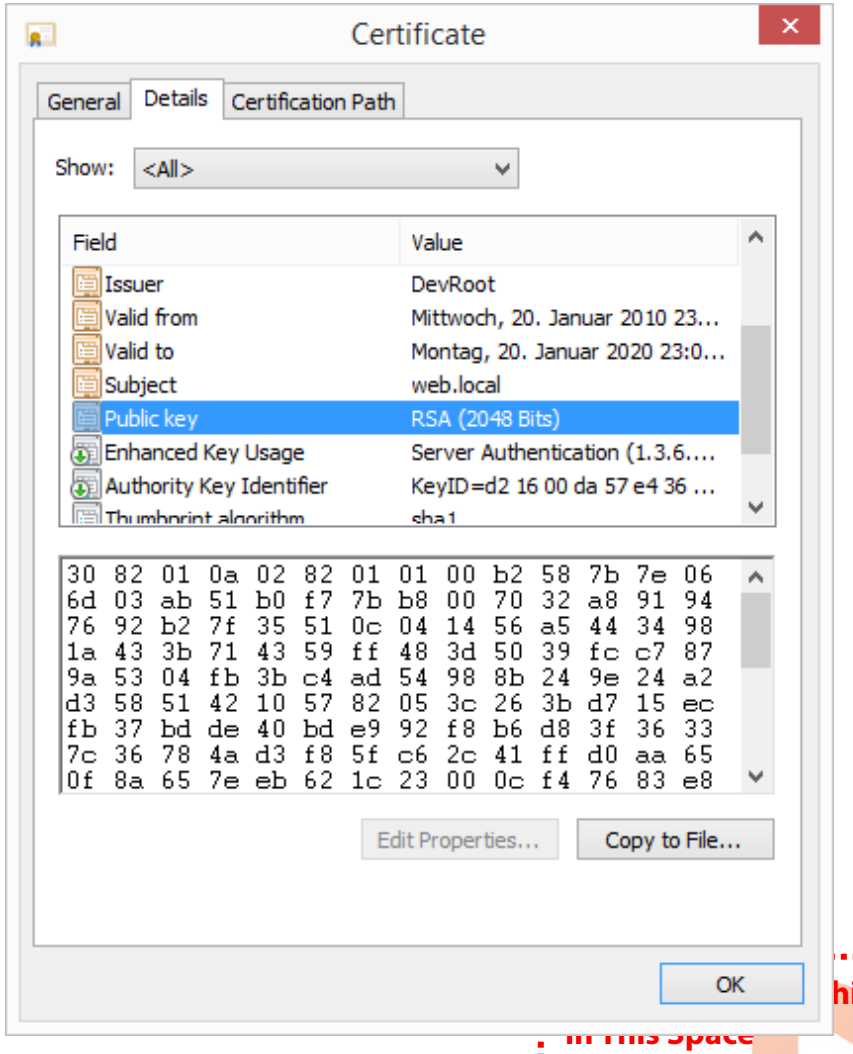
Transport security

- **HTTPS == HTTP over TLS**
  - RFC 1818

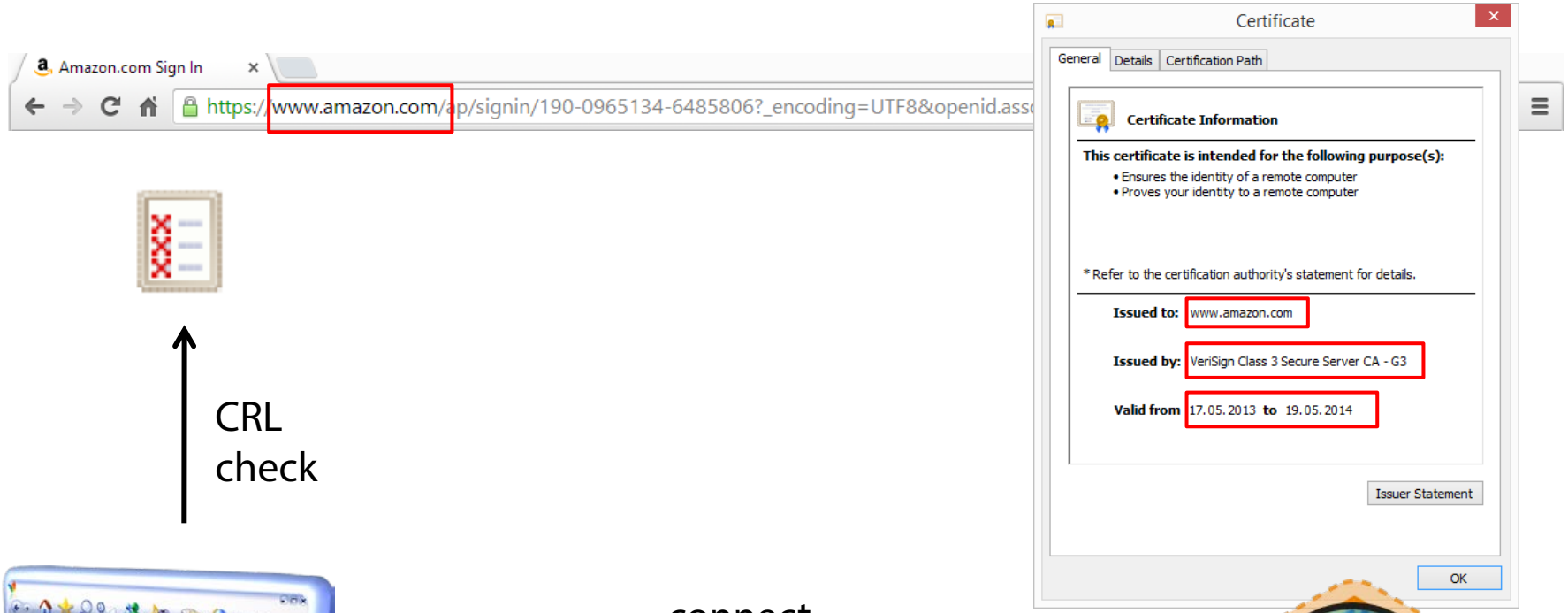
- **Tunnels unprotected HTTP and adds**
  - server authentication
  - integrity protection
  - replay protection
  - confidentiality

X.509 Certificates



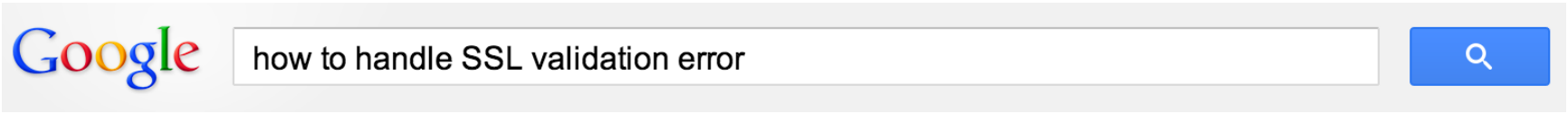


Simplified SSL handshake



http://www.moserware.com/2009/06/first-few-milliseconds-of-https.html

Developers & SSL



SSL Certificate **Validation Error** in .Net « Akbar's Blog
blog.syedgakbar.com/.../**ssl**-certificate-**validation-error**-in-net/
Jul 17, 2012 – This callback method is used to **validate** the certificate in an **SSL** conversation // Changed the **handle** to ignore the **SSL** Certificate **errors** n the ...

SSL Function Return Codes
publib.boulder.ibm.com/infocenter/.../**sssl**2msg1000885.htm
The environment or **SSL handle** specified on a System **SSL** function call is not ... Certificate **validation error**. ... An error is detected while validating a certificate.

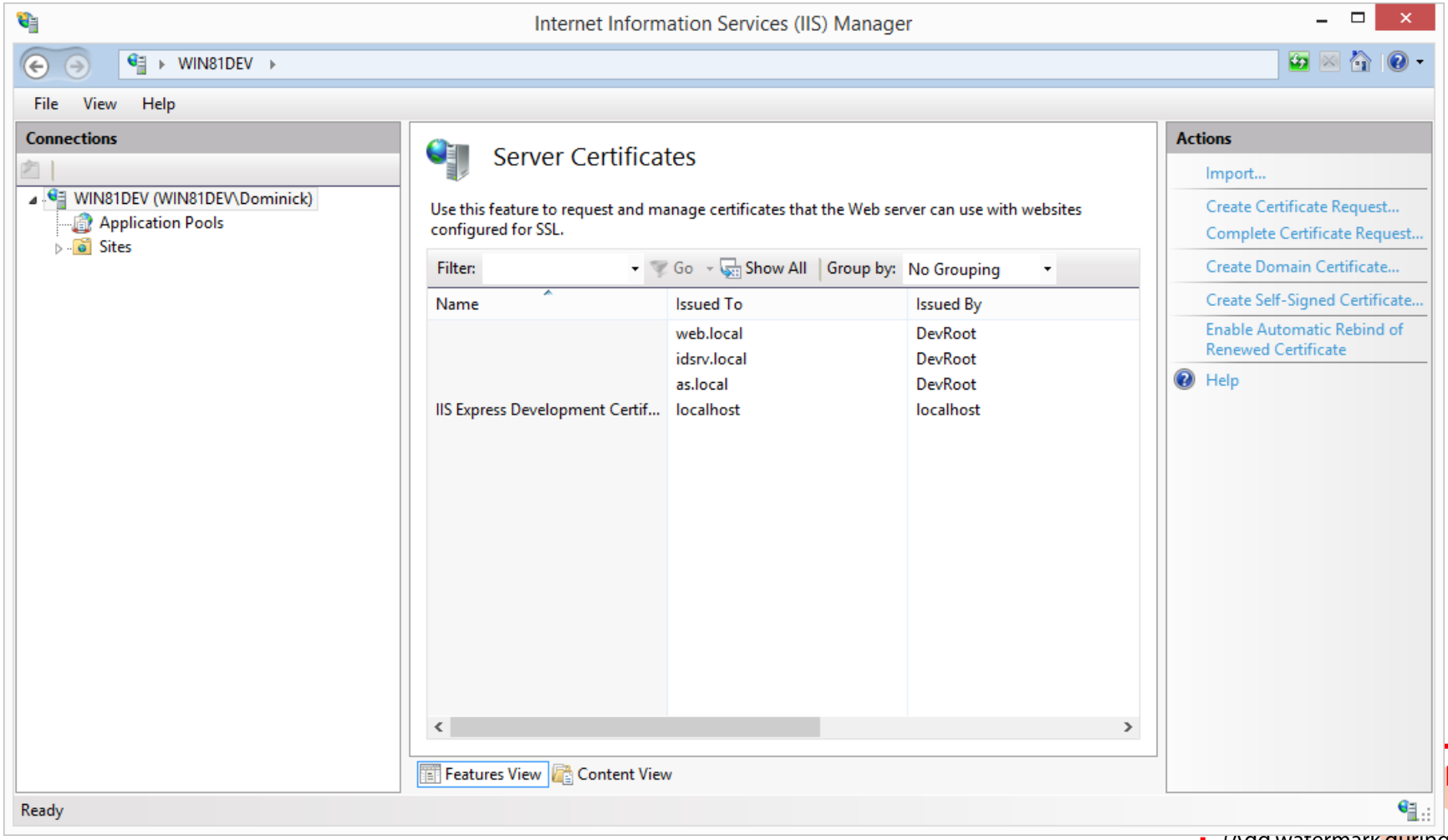
Ignoring **SSL validation** in Java - Stack Overflow
stackoverflow.com/questions/.../ignoring-**ssl-validation**-in-java
2 answers - 20 Nov 2012
Foreword: I DO know that skipping **SSL validation** is really ugly. In this ... ClientStateReceivedServerHello.**handle**(Unknown Source) at ... catch (KeyManagementException e) { log.**error** ("No **SSL** algorithm support: " + e.

How to handle invalid **SSL** certificates with Apache - Stack Overflow
stackoverflow.com/.../**how-to-handle**-invalid-**ssl**-certificates-wi...
9 answers - 1 Dec 2009
... at sun.security.validator.Validator.**validate**(Validator.java:235) at sun.security.**ssl**. ... When I go to mms.nw.ru, I get a **error** screen in Chrome.

Where to get certificates from?

- **Buy**
  - Verisign etc…
- **Corporate PKI**
  - Windows Certificate Services
- **Create yourself**
  - makecert.exe
  - OpenSSL

Creating/requesting certificates with IIS



Creating a root certificate

makecert.exe

-r	// self signed
-n "CN=DevRoot"	// name
-pe	// exportable
-sv DevRoot.pvk	// name of private key file
-a sha1	// hashing algorithm
-len 2048	// key length
-b 01/21/2010	// valid from
-e 01/21/2030	// valid to
-cy authority	// certificate type
DevRoot.cer	// name of certificate file

Creating an SSL certificate

makecert.exe

-iv DevRoot.pvk	// file name of root priv key
-ic DevRoot.cer	// file name of root cert
-n "CN=web.local"	// name
-pe	// mark as exportable
-sv web.local.pvk	// name of private key file
-a sha1	// hashing algorithm
-len 2048	// key length
-b 01/21/2010	// valid from
-e 01/21/2020	// valid to
-sky exchange	// certificate type
web.local.cer	// name of certificate file
-eku 1.3.6.1.5.5.7.3.1	// extended key usage

Setting up SSL

- **Establish trust**
  - Windows certificate store

- **Bind SSL certificate to port / host name**
  - IIS
  - netsh.exe
  - httpconfig.exe

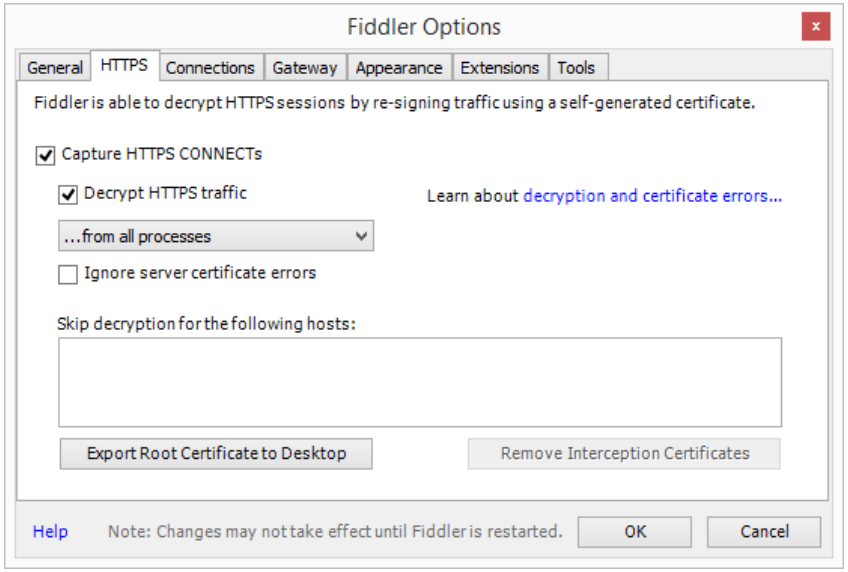
Using code to validate certificates

```
private bool ValidateUsingValidator(X509Certificate2 cert)
{
    var validator = X509CertificateValidator.ChainTrust;

    try
    {
        validator.Validate(cert);
        return true;
    }
    catch (SecurityTokenValidationException)
    {
        return false;
    }
}
```

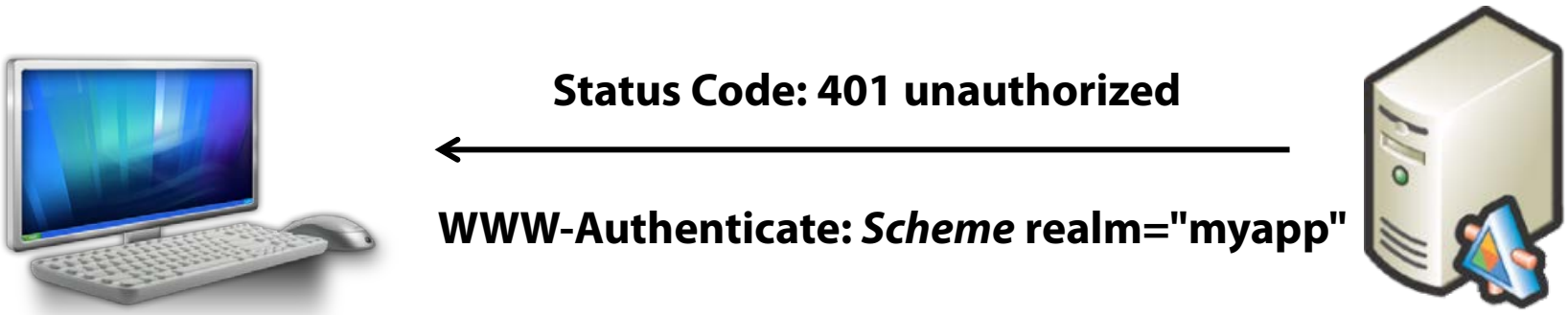
Useful tools

- **Fiddler**
  - HTTP proxy
  - mainly for debugging purposes
  - Can "sniff" HTTPS connections



HTTP Authentication Framework

- **Whenever authentication is required**
  - Status code of 401 indicates *unauthorized*
  - *WWW-Authenticate* response header indicates preferred authentication method



Authentication for HTTP-based services

- **Credentials transmitted (typically) via *Authorization* header**
  - e.g. Basic authentication, access tokens…
  - sometimes other means (query string, cookie…)

GET /service/resource

Authorization: *scheme* credential

Summary

- **HTTP has no transport security on its own**
  - SSL/TLS layer protects data on the wire
- **Every developer should understand how SSL/TLS works**
  - at least the simple rules
    - common name has to match DNS name
    - expiration
    - trusted root
  - don‘t disable SSL validation
- **HTTP authentication is simple**
  - challenge via status code 401 / WWW-authenticate header
  - send credentials via Authorization header

Resources

- **Thinktecture.IdentityModel**
  - https://github.com/thinktecture/Thinktecture.IdentityModel
- **HttpConfig**
  - http://www.stevestechspot.com/ABetterHttpcfg.aspx
- **Netsh documentation**
  - http://msdn.microsoft.com/en-us/library/windows/desktop/cc307236(v=vs.85).aspx
- **Fiddler**
  - http://www.telerik.com/download/fiddler

Resources II

- **PluralSight:**
  - HTTP Fundamentals - Scott Allen
  - Introduction to IIS Certificates - Paul Lemmers
  - IIS for Developers - Steven Evans