Why Secure Your Servers?

- That’s not the question for today.
- This webinar goes under the assumption that you’ve decided to do so and have the specific problem of securing server(s) that are internal (not on the Internet).

- Along the way, we will take a bit of a look at
  - How certificates work.
  - How computers find each other on a network (name resolution).

Who Says Your Server Is Who It Claims To Be?

- The aspect of verifying identity (you really ARE putting your password into the Bank of America website) is a matter of trust.
- Windows trusts a certain number of “root certification authorities”. This list is occasionally updated by Windows Updates.
- Windows also trusts any certificates signed by the private (secret) key of any of the certification authorities it trusts.
- Browsers verify the signed name in the certificate against the name you’ve entered in the browser.

Options for TLS on LAN/WAN

- Let's Encrypt is not an easy option
  - While LE is becoming increasingly popular and convenient on the Internet (and NetTalk 10 automates the process), it's not a practical option for internal servers.
  - LE requires a public Internet-facing server with port 80 available to receive the automatic authentication.
  - You might be able to build an ACME client to obtain a certificate, but the certificate would need to be re-obtained and manually re-installed on your internal server(s) every 90 days or less.

Options for TLS on LAN/WAN

1. Self-signed certificate. Tell users to click through the error. = (NO NO) * 10^3 * NO
2. Self-signed. Install certificate on each machine. (Marginally OK for a very small network.)
3. Buy a commercial certificate. Create custom zone on local DNS to enable all devices on LAN/WAN to connect to it without errors.
4. Create a Certificate Authority for your network. Trust it in Active Directory. All domain computers will automatically trust it. But non-domain computers (Chromeboxes, BYOD, phones, etc.) will not.

Options Are Good

- To the man whose toolbox contains only a hammer… every problem looks like a nail.

- We'll look at the latter two options today.

First, a Word About Encryption

- Simplest encryption uses a shared secret (a single key).
- You use the secret password or code to encrypt a message (maybe using something as simple as an XOR).
- The recipient uses the SAME password or code to decrypt the message.
- Advantage – simple and fast for computers to process.
- Disadvantage – too many people have the secret key.

Public Key Encryption – Two Keys

- I give EVERYBODY my encryption key. Go ahead, share it with Boris and Natasha!
- Because… what is encrypted with that key (my so-called PUBLIC KEY) cannot be decrypted with that key.
- It can only be decrypted with my PRIVATE KEY, which I keep SECRET.
- Mathematically complex, only a specific private key should mate/interact with its corresponding public key. They are a key pair.

Public Key Encryption

- OR… I can encrypt something with my private (secret) key and send it to you.
- But… doesn't that mean that Boris and Natasha can also intercept and decrypt it?
- Yes… but that's not the purpose of my encrypting it in this case – I'm not trying to keep the contents secret.
- In this case, because it is encrypted with my private key, the fact that you can decrypt it guarantees that it was I who sent it and it has not been altered. It is a signature that only I can make.

What Does a Certificate Provide?

- Makes your public key available to other users so they can
  - Send encrypted stuff to you (encrypted with your public key, which only you can decode)
  - OR decrypt something from you which guarantees that YOU sent it because it was encrypted with your private (secret) key -- (signature hash, etc.)
- Verifies that the server to which the user is connecting actually is what it says it is (the URL browsed-to matches what is embedded in the certificate that has been signed by an authority that the browser trusts.).

TLS (The ~~Artist~~ Protocol Formerly Known as SSL)

- Symmetric encryption is much faster than public key.
- So public key is just used to get things started.
- Client and server handshake – what ciphers to use, etc.
- Client computes a random “pre-master” secret, then uses the server’s public key to encrypt it.
- Server decrypts that “pre-master” using his private key.
- More hand-shaking ensues.
- Client and server generate “master secret” and then “session keys” based on the “pre-master” the client originally sent. They will use the session keys for fast symmetric encryption.

The Commercial Cert Approach

Certificate Trust

- Commercial certificates are signed by issuing Certificate Authorities that are trusted by Windows computers.
- You can see the list of trusted root and intermediate authorities in Internet Properties in Control Panel or in IE.
- These are updated from time to time by Windows Update. If you get complaints about a certificate (SSL or code-sign) being rejected, make sure the client is installing Windows updates regularly.

Can No Longer Buy Cert for Internal Server Names or Addresses

That option died in 2015.

https://cabforum.org/wp-content/uploads/Guidance-Deprecated-Internal-Names.pdf

Introduction

On November 22, 2011, the CA/Browser Forum adopted "Baseline Requirements for the Issuance and Management of Publicly-Trusted Certificates, Version 1.0" (hereafter referred to as the "BR 1.0") to take effect on July 1, 2012[1]. As part of these requirements, Section 9.2.1 indicates:

> As of the Effective Date of these Requirements, prior to the issuance of a Certificate with a Subject Alternative Name (SAN) extension or Subject Common Name field containing a Reserved IP Address or Internal Server Name, the CA SHALL notify the Applicant that the use of such Certificates has been deprecated by the CA / Browser Forum and that the practice will be eliminated by October 2016.

Definitions

For the purposes of certificate issuance pursuant to the BR 1.0, the following definitions are used:

- **Domain Name**: The label assigned to a node in the Domain Name System.
- **Fully-Qualified Domain Name (FQDN)**: A Domain Name that includes the labels of all superior nodes in the Internet Domain Name System.
- **Internal Server Name**: A Server Name (which may or may not include an Unregistered Domain Name) that is not resolvable using the public DNS.
- **Reserved IP Address**: An IPv4 or IPv6 address that the IANA has marked as reserved: http://www.iana.org/assignments/ipv4-address-space/ipv4-address-space.xml

  http://www.iana.org/assignments/ipv6-address-space/ipv6-address-space.xml

Background

The “Trick” to Using a Commercial Cert

- Use a URL that is valid on the INTERnet, i.e. reports.jftech.me
- Buy a commercial certificate for that URL.
- Configure your LAN/WAN so that its clients do not look in the INTERnet for that device, but instead look at your internal INTRAnet server. This will be done by means of your network’s internal DNS server(s) that are used by all computers on your network.

A Note About The Webinar Demo

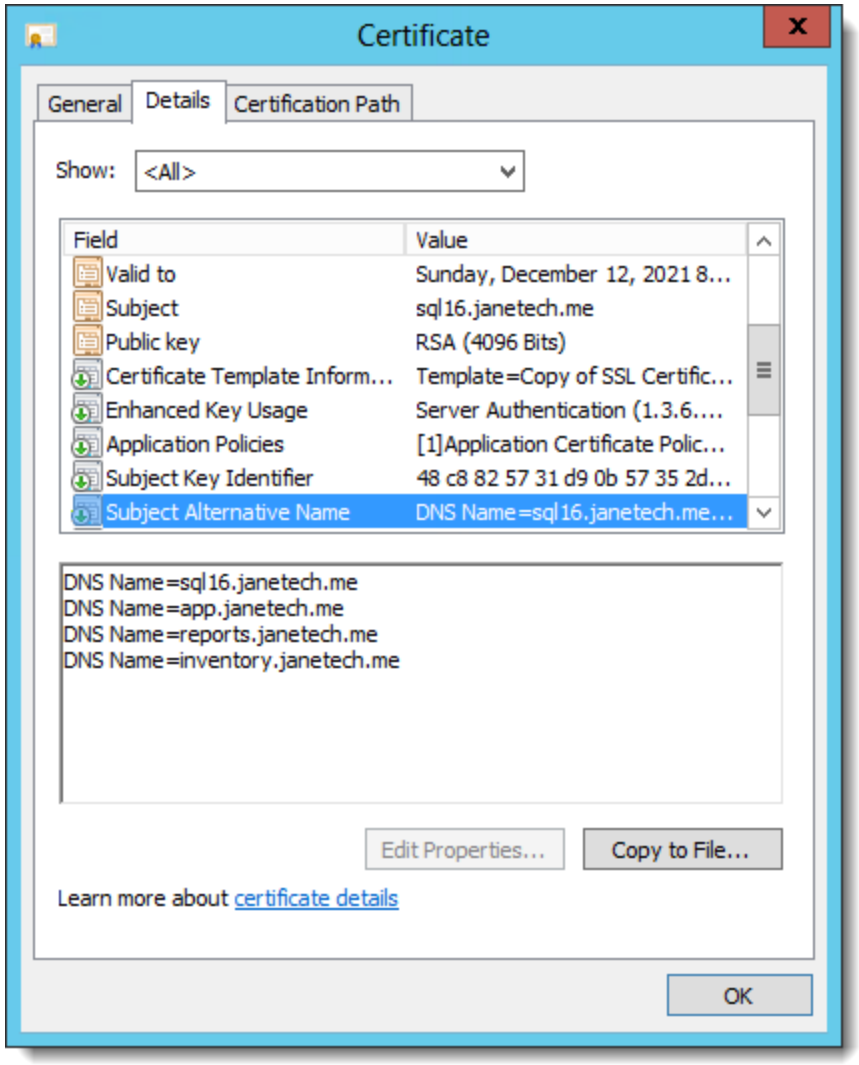
- I bought the www.janefleming.com certificate for this webinar – from the cheapest, low-hassle source I could find.
- You would normally NOT use a www.myrealcompanyname.com type of domain name on your internal network.
- You might carve out some non-www names with your real domain name, or choose another domain name for internal use.

Buying a Commercial Cert

- Think about what domain name to use – maybe buy one domain name just for this purpose (janetech.me) You will need an admin email address for this domain, but do not actually need to make a web server.
- If you know you need to provision a few servers, try to find a cert deal that lets you specify X number of Subject Alternative Names based on that domain name for a single price.
- Or if you need a lot of servers, buy a wildcard cert for that domain name.

Subject Alternative Names

Example of a cert valid for 4 websites



Buying a Commercial Cert – Your Private Key

- Do you know where your private key is?
- If certificate request was created “in Windows” (rather than with a tool like openssl)
  - The private key is stashed in Windows’ crypto store.
  - You need to retrieve the signed certificate on THAT computer to mate the certificate and key together. (Also true for code-signing certs.)
  - I’ll demonstrate this later today.

Name Resolution

- Your computer has a network interface card of some sort.
- Your NIC doesn't know anything about ClarionLive, Amazon, or any other names.
- For that matter, it doesn't know about IP addresses. It only knows its own MAC address.
- Somehow, packets have to get from the MAC address of a computer at Amazon to your NIC's MAC address.

Name Resolution

- Although MAC addresses can be spoofed, typically they're assigned by the manufacturer.
- This means there's no logic to where a particular one is located (one HP card might be in Rio and another one in Capetown). The MAC address tells you nothing.
- So "above" the MAC address (in the OSI and TCP models), is another layer with an IP address (OSI Layer 3).
- IP addresses are logical – by looking at two addresses and their subnet masks, you can determine whether they are on the same network or if they're separated by router(s).

Name Resolution

- On your home network, you can find the printer, other machines, etc. by “broadcasting”. “HALLO… whaz yer address, Printer??”
- Because someone in a Sony office in Tokyo doesn’t want to hear Jane searching for her printer, routers do not pass broadcast packets.
- So to find a machine that’s not available via broadcast on your network, you need to use something that CAN traverse routers (meaning IP – a Layer 3 protocol).

TCP/IP Name Resolution (Windows)

- Is this MY name?
- Is there anything in HOSTS file? (This makes HOSTS file useful for testing… also a target of viruses and malware.)
- Ask DNS. DNS packets primarily use UDP, not TCP. Because you have a DNS server address, packets can pass through routers (they are not broadcasts).
  - Your home DNS is provided by your ISP, typically through your router.
  - The type of networks we're talking about in this presentation maintain their own DNS servers. We will make use of this for using a commercial certificate.

TCP/IP Name Resolution (Windows)

- Packets pass from router to router as needed until they reach the segment where the addressee computer actually lives.
- Once packet reaches your actual network segment (based on the destination IP address), it needs to get your MAC address (which it finds using ARP). Amaze your girlfriend – show her your ARP cache:

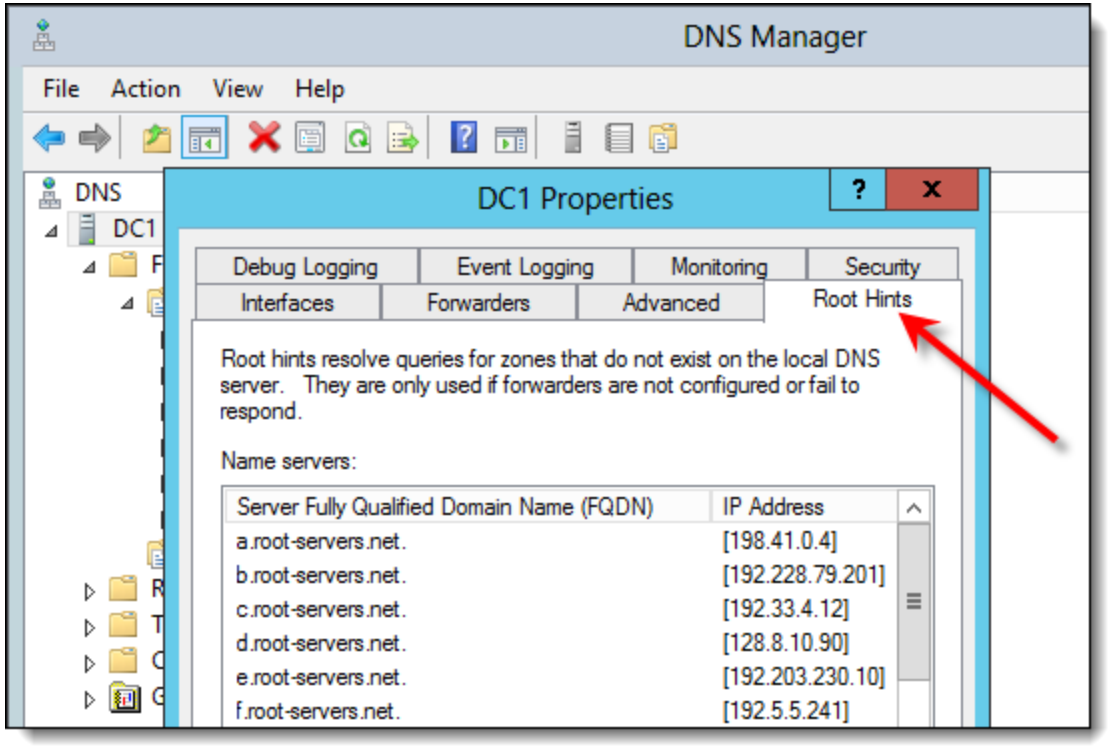
  arp –g

DNS Server Decision Sequence
when it receives a query

- Is this a query for a domain where *I* am authoritative? (We use this feature for the “DNS trick".)
- Otherwise
  - Use “root hints” – iteratively query root domain, then com domain, then appropriate next domain, etc., based on a chain of referrals.
  - OR use forwarder – ask another DNS server for a “yes/no” answer but not for a referral.

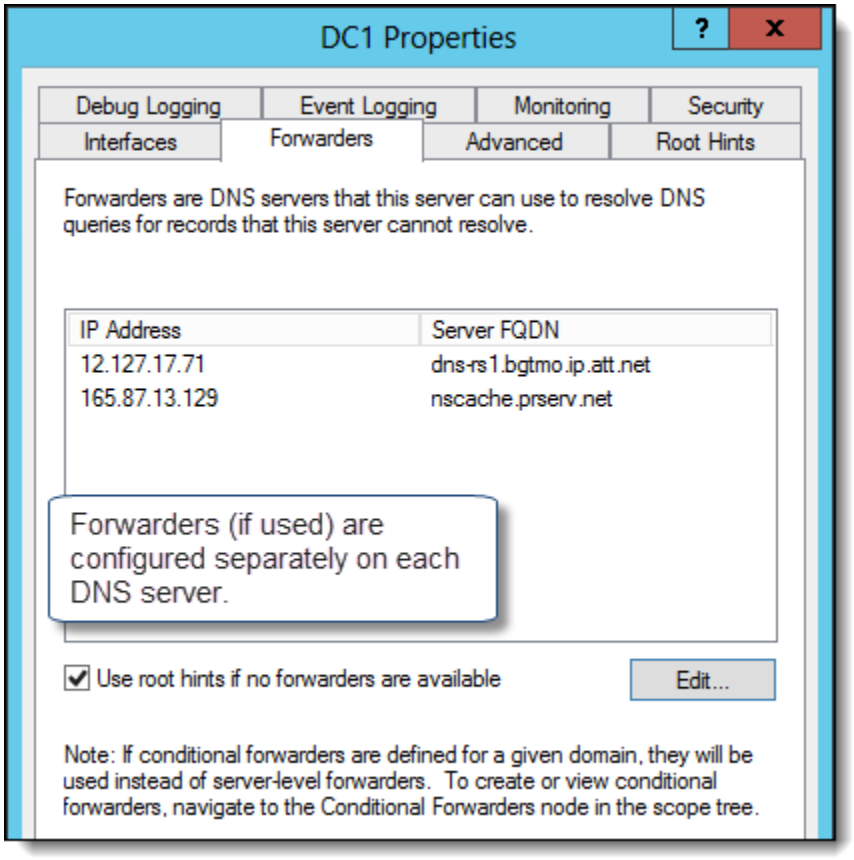
Root Hints

- Normally added automatically at installation



Forwarders

- Specified individually for each DNS server.



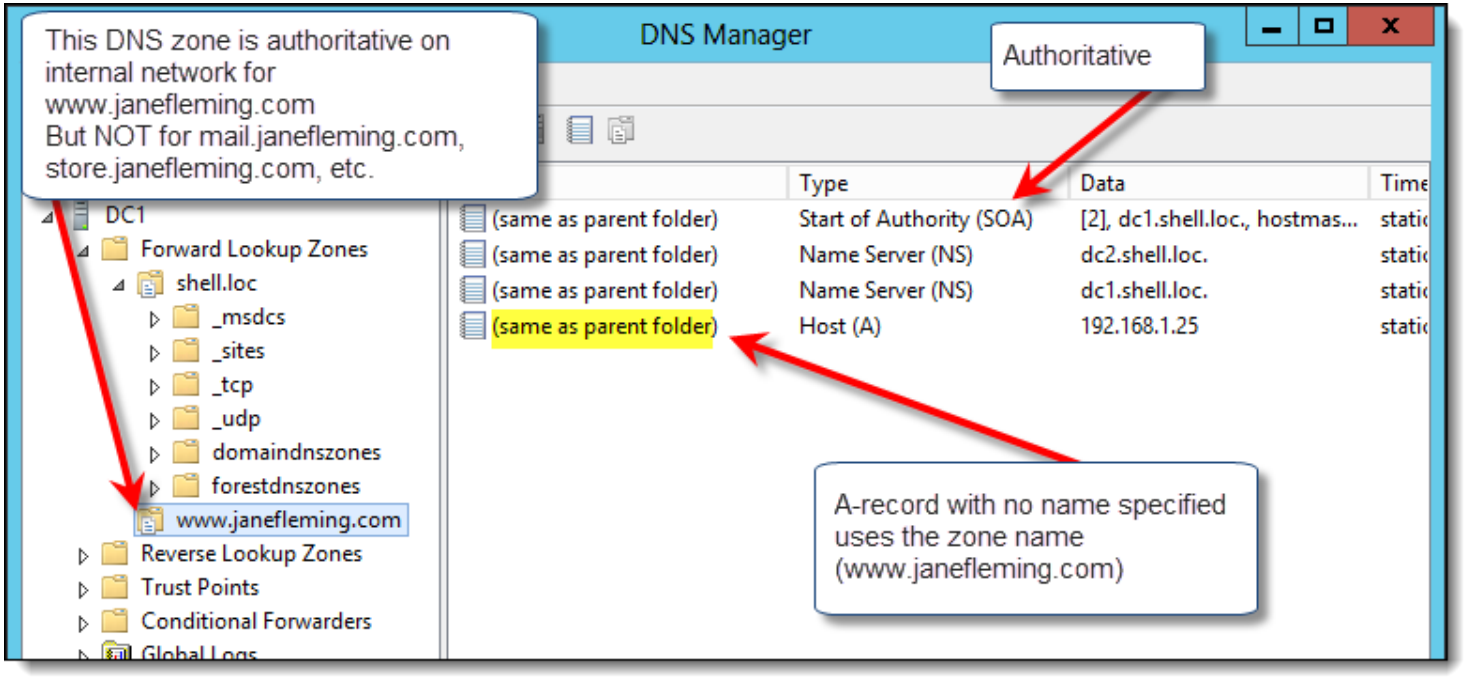
DNS Zone For Our Server

- We create a DNS zone limited to the exact URL for this server. (On my work network, we have dozens of these carve-outs.)
- Add a host (A-record) in the zone with the internal network IP address for this server.
- DO NOT specify a name for the A-record. Then it will use the domain name.

http://www.petenetlive.com/KB/Article/0000830

https://community.spiceworks.com/topic/566556-ssl-certificates-internal-use-and-external-use

Internal DNS Zone For Our Server



DNS / Security

- As an sidebar, today's demos should give you a wakeup about how DNS can be exploited/misused. (i.e., the kid at the next table at Starbucks who's running a WAP and his own DNS server on his laptop and hopes you'll connect to him so he can steer you to HIS "Bank Of America".)
- DNSSEC (secure) DNS is a "thing", but with the added overhead it brings, hasn't widely caught on.

The Enterprise CA Approach

(Be a big fish in your own little pond.)

What is a Windows Domain?

- Do NOT confuse this with the Internet meaning – as in your “domain name”. In Windows, this is a shared security realm.
- Simple network – peer-to-peer. You set permissions/passwords on stuff on each machine. No domain involved.
- Early Novell networks – you create user name and password for EACH SERVER on the network. And remember to change your password on each server each month. No domain involved.

What is a Windows Domain?

- Early Microsoft “domains” (Windows NT in the 90s) – designate one or more server computers as Domain Controllers
- DCs contain user names and password hashes
- Other computers in the network “join the domain” which means they will trust the users stored on the Domain Controller(s). (In computer properties, you will see the computer in a domain instead of in a “workgroup”.)
- User has one centralized logon/password for all network resources. Permissions can be set for each of those users on all resources in the network.

What is Active Directory (AD)?

- AD consists of a “forest”
- Each forest contains one or more “domains” that do the stuff that domains did in earlier versions of Windows NT (single logon/password for access to all domain resources, user and administrator groups, etc.).

What is Active Directory (AD)?

BUT WAIT, THERE'S MORE NOW!!!

- Microsoft re-architected its domain concept with Windows 2000, allowing for:
  - Much larger networks
  - Centralized control of every computer setting for users (wallpaper, which apps permitted in control panel, logon scripts, password policy, drive mappings, etc.) through something called Group Policy. Each GPO (Group Policy Object) can have ~5,000 settings. And they can cascade (a user may be subject to multiple GPOs.)
  - An additional layer of abstraction – organizational units (OUs) to which different Group Policies can be applied for different classes of users/computers. You can also delegate administration of an OU to somebody who is not a full domain admin. OUs are NOT the same as user groups (security groups).

What is Group Policy?

- Although Group Policy gives a lot of control over what users may or may not do, it is NOT the same as permissions (set on object such as files and folders) or rights (such as the right to add computers to the domain).
- Policies can be set that apply to all users, or to all computers, or to users or computers in a specific OU.

Why Do We Care about AD?

- It will provide a necessary component for one of the options for TLS on the LAN/WAN, because Group Policy can publish a certificate that all domain users and computers will automatically trust.

Enterprise CA

- Create a Certification Authority trusted by the entire Windows domain
- The cool kids do two levels
  - The root CA, which lives in a closet on a non-domain computer that is powered off and protected by hard men with tattoos and MP5s and dogs named Butch (gotta keep that private key private!)
  - One or more issuing CAs whose own certificates were signed by the root CA before he was turned off.

Enterprise CA

- The root CA's certificate is trusted in Active Directory domain Group Policy, therefore all computers joined to the domain automatically trust it… and trust any certificates signed by it or by the secondary issuing CA(s) that it has authorized. (But NOT BYOD and non-Windows devices.)
- Certificates signed by the issuing CA work directly/easily with SSRS, IIS, etc.
- To use with NT, use openssl to split an exported PFX into cert and non-encrypted key file.

From PFX to Cert/Key for NT Server

- Openssl pkcs12 –in MyDomain.pfx –nocerts –out MyDomainEncrypted.key
- Openssl rsa –in MyDomainEncrypted.key –out MyDomain.key
- Openssl pkcs12 –in MyDomain.pfx –clcerts -nokeys –out MyDomain.crt

Handy Tricks Department

NT and IIS both on port 80 on same server (different IP, of course)

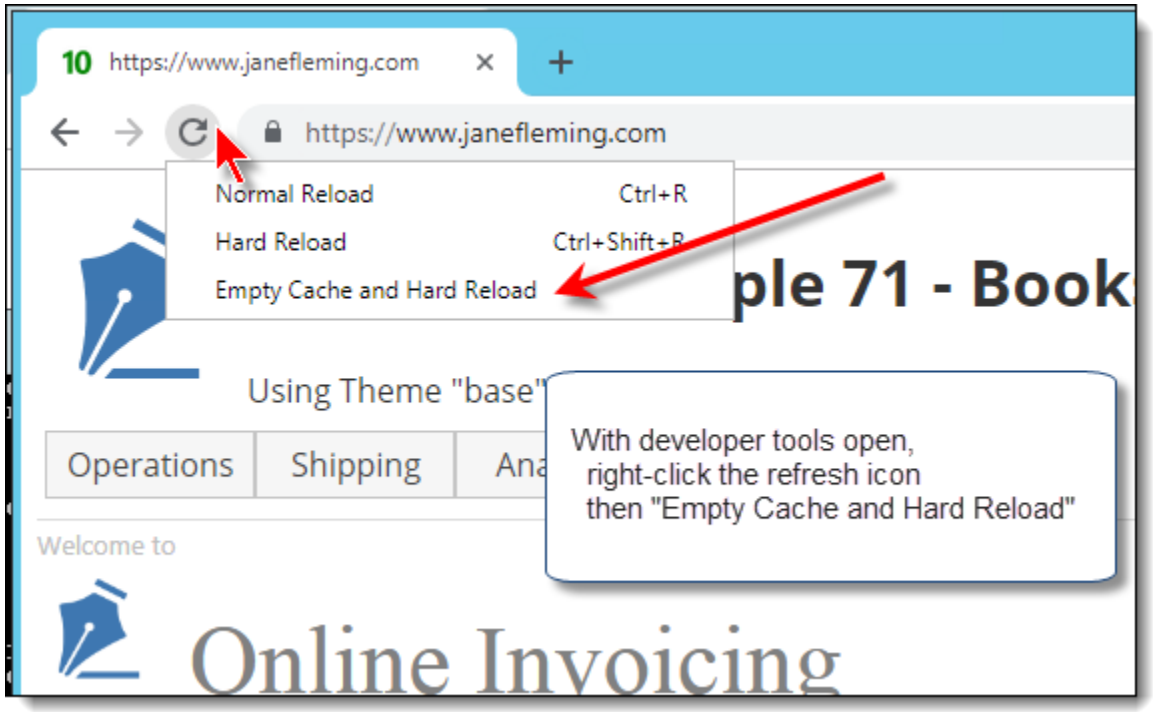
- Thanks to Johan on NT forum!!!
- Elevated prompt:
  - netsh
  - http
  - add iplisten ipaddress=[ip addr IIS should use]

  Otherwise, IIS listens on ALL addresses (even if not specified in bindings) and won't start if NT server is running.

  http://www.nettalkcentral.com/forum/index.php?topic=7206.msg29316;topicseen#msg29316

Hard Reset for Chrome Browser

- When you're doing development and Chrome won't stop caching stuff, open developer tools (F12), then right-click the Refresh icon.

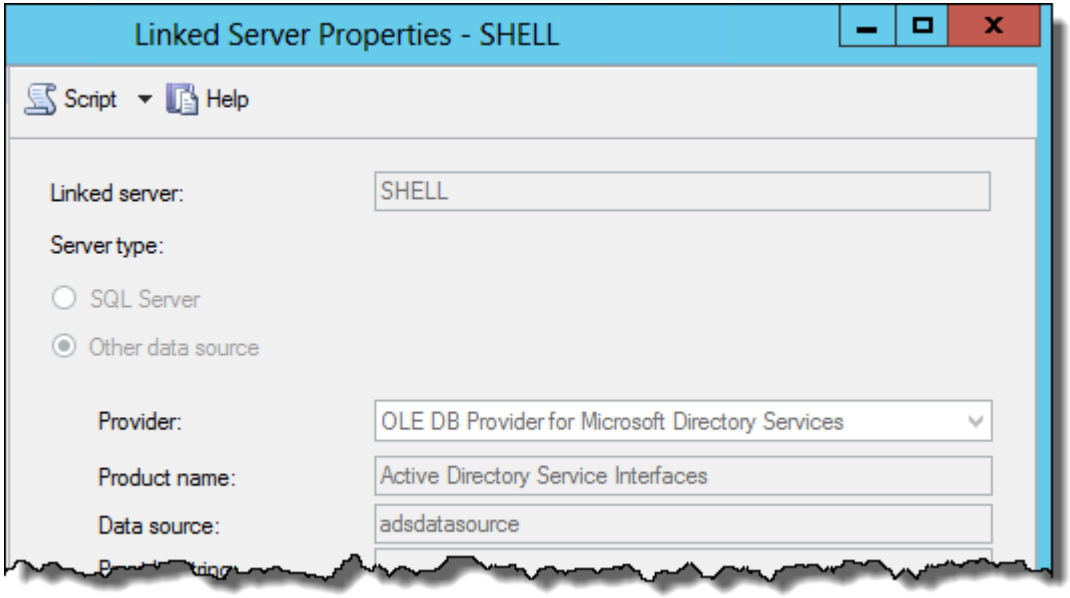


Fun with nslookup

- Command line tool nslookup shows whether a query is non-authoritative, etc.
- Set debug=on Then queries will show fun stuff like TTL, primary name server, etc.
- Manually specify a different server, then do queries and see what you see.

Fun with SQL and Active Directory

- Add your Active Directory as a Linked Server to a SQL Server.



- Learn to query it from SQL – a good example is to take a logged-in user and determine whether he is a member of an AD security group.

Fun with SSL

- Get the book from Feisty Duck. Hurt your head.