TK2100: Informasjonssikkerhet
Lesson 07: Network 2

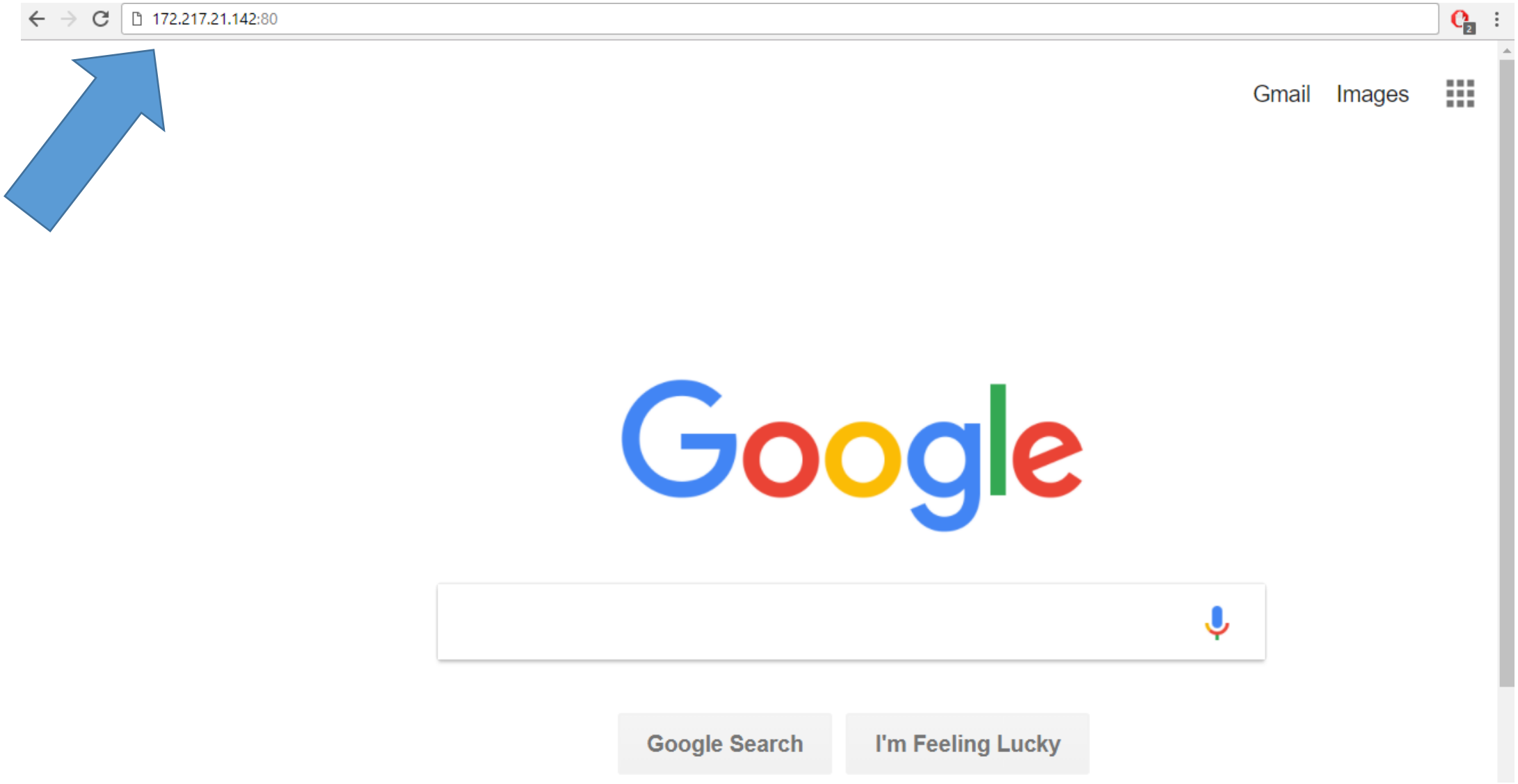
Dr. Andrea Arcuri
Westerdals Oslo ACT
University of Luxembourg

Goals

- Understand how DNS works, and why it is insecure
- Understand how to carry out a web site phishing attack

Domain Name System (DNS)

How many of you type IP addresses like **172.217.21.142** when browsing the internet?

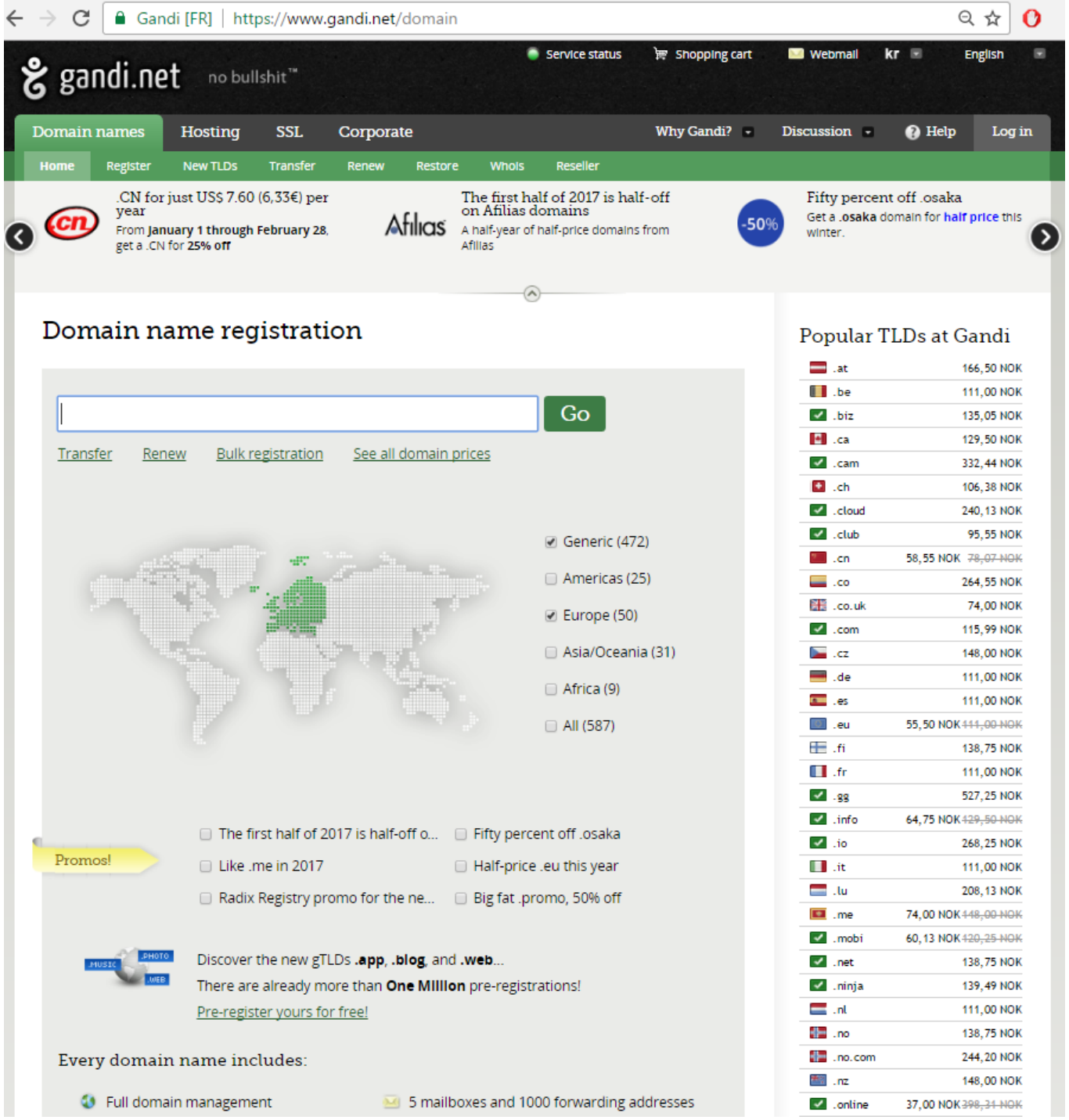


Hostnames

- Network/Transport protocols like TCP/IP have **NO** knowledge of *hostnames* like www.google.com or www.facebook.com
- But asking users to remember and manually type IP addresses like 172.217.21.142 is not viable
- Need something that can *translate/map* from hostnames to IP addresses
- Also extra benefit that you can change IP addresses without the need of clients to update (as they use the hostnames)

Domain Name System (DNS)

- *Application* Layer protocol used to find IP addresses from hostnames
- There exists DNS servers that can be contacted to do the hostname mapping
- *ICANN: Internet Corporation for Assigned Names and Numbers*
- You can pay a fee to register a hostname, and choose to map it to an IP address of your choice



Registering a domain mapped to an IP address of your choice is not particularly expensive: eg 139 NOK per year for a ".no" address

For example, I used it to register "arcuriandrea.org"

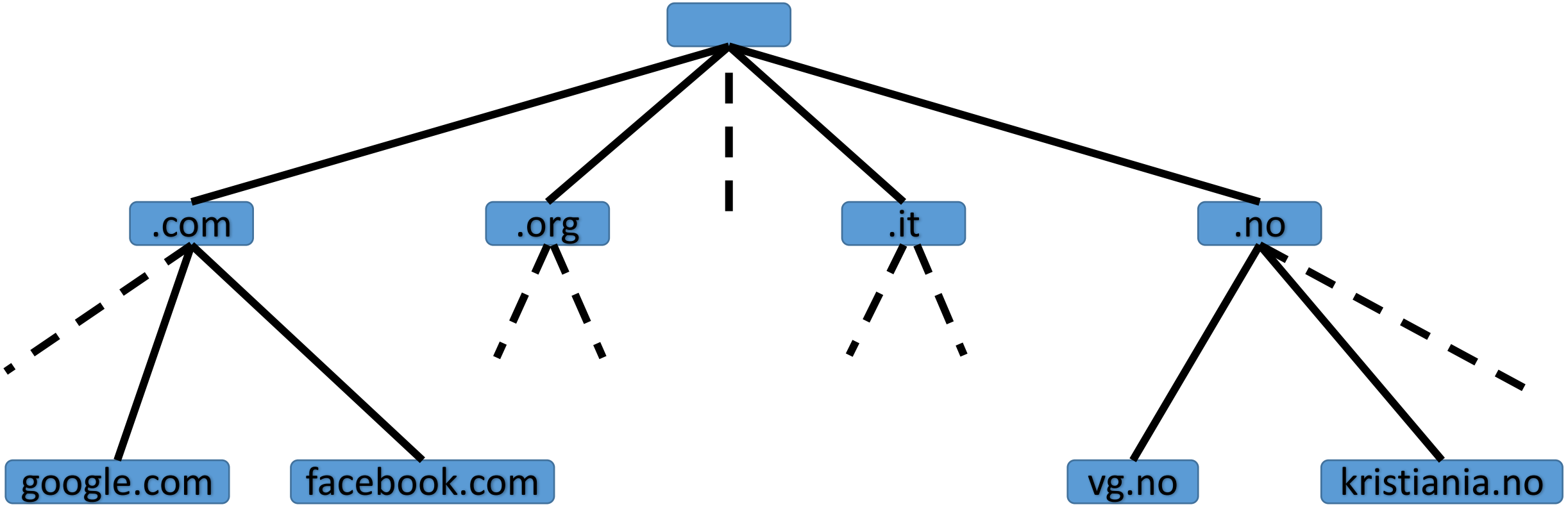
Hierarchy

- Domain names are organized in a hierarchy, which is read from *right* to *left*
- Consider the hostname “*kristiania.no*”
- “*no*”: is a top-level domain, representing Norway
  - countries have their own, like “*au*” for Australia and “*it*” for Italy
  - There also generic ones like “*com*” and “*org*”
- *“kristiania.no”*: a subdomain of “*no*”

What about “www”?

- WWW stands for *World Wide Web*
- “*www.kristiania.no*” is a subdomain of “*kristiania.no*”
  - they are 2 DIFFERENT hostnames
  - they can be mapped to same IP address, or different ones
    - although it is more common to use 301 HTTP permanent redirection (will see later in the course)
- Why?
  - Historical reasons, to make clear that “*www.foo.com*” was something you could type in this revolutionary new thing called internet browser… just using a “*.com*” was not clear enough when internet was first introduced in the 80s/90s
  - Might have different servers for different applications (eg email server) and want to use same main domain “*foo.com*”, where “*www.foo.com*” is a subdomain just for the HTML pages

Distributed Database

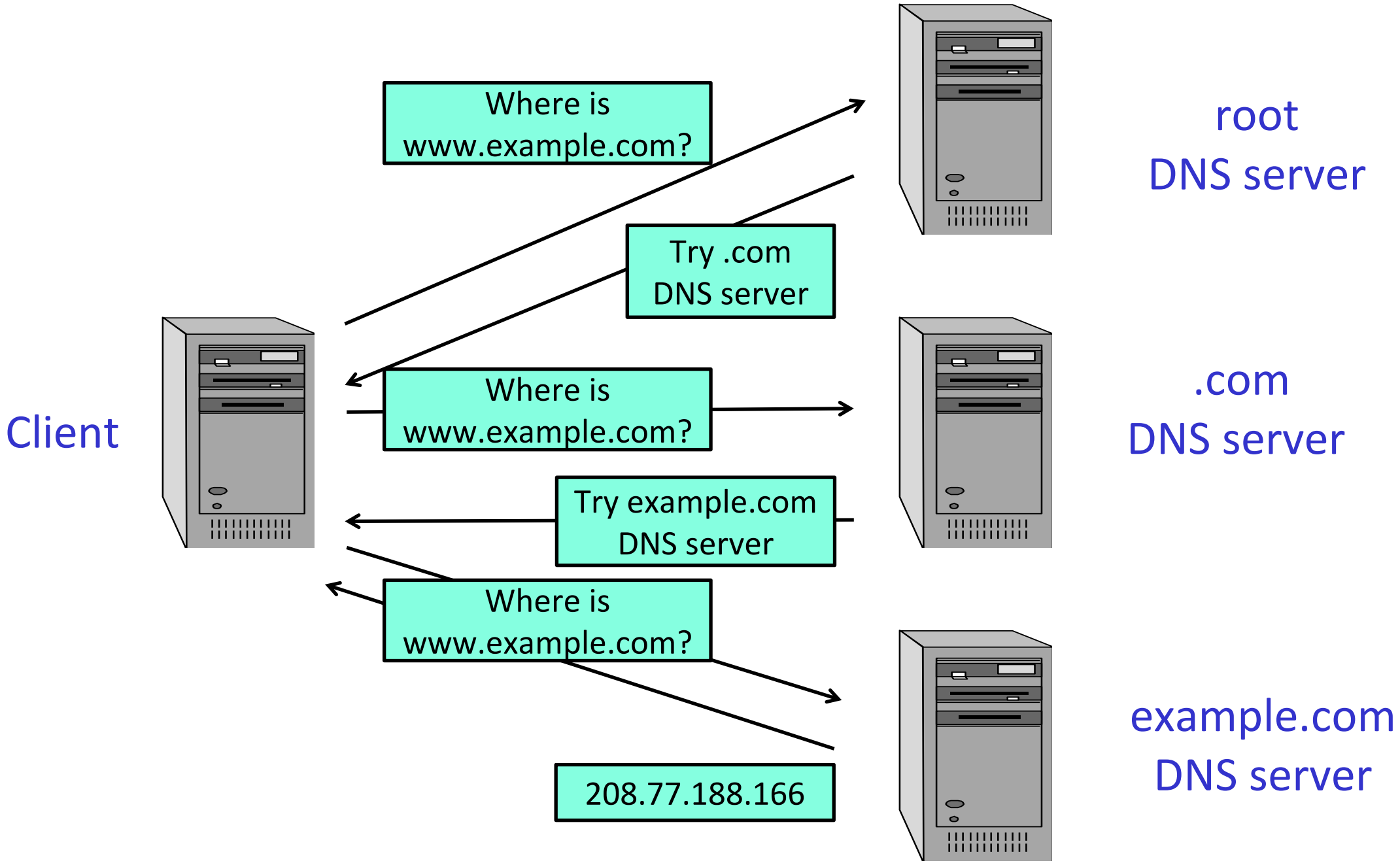


Many Servers

- The root DNS server has information of where to find (ie IP) the servers of the top-level domains (eg, "*.com*" and "*.no*")
- Top-level servers have info of where to find hosts in their subdomains
  - Eg, server for "*.no*" will have info for "*kristiania.no*" and "*google.no*", but not "*google.com*"
- National top-level domains are usually handled by government institutions in such countries
  - eg in Norway handled by Norid, own by the Norwegian Ministry of Education and Research

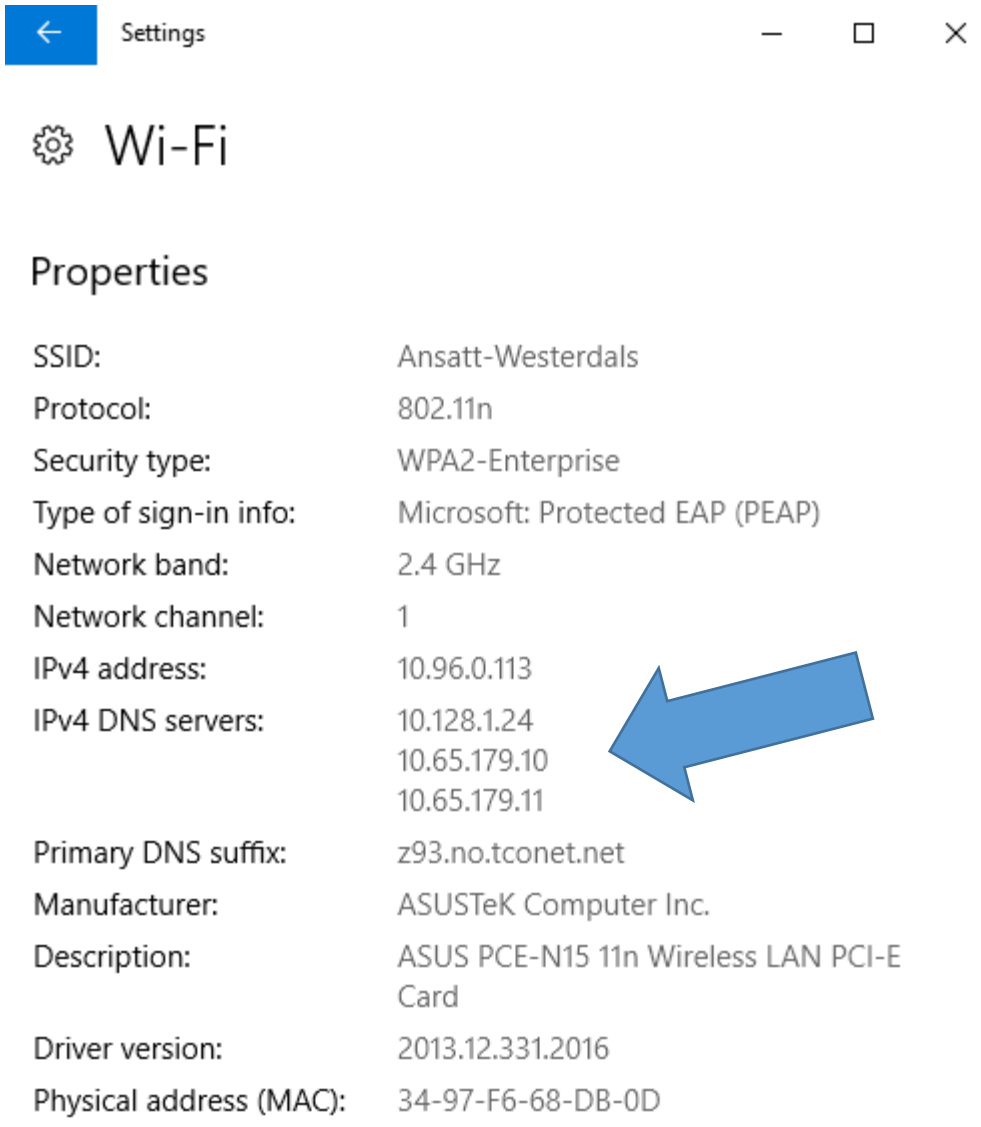
DNS Resolution

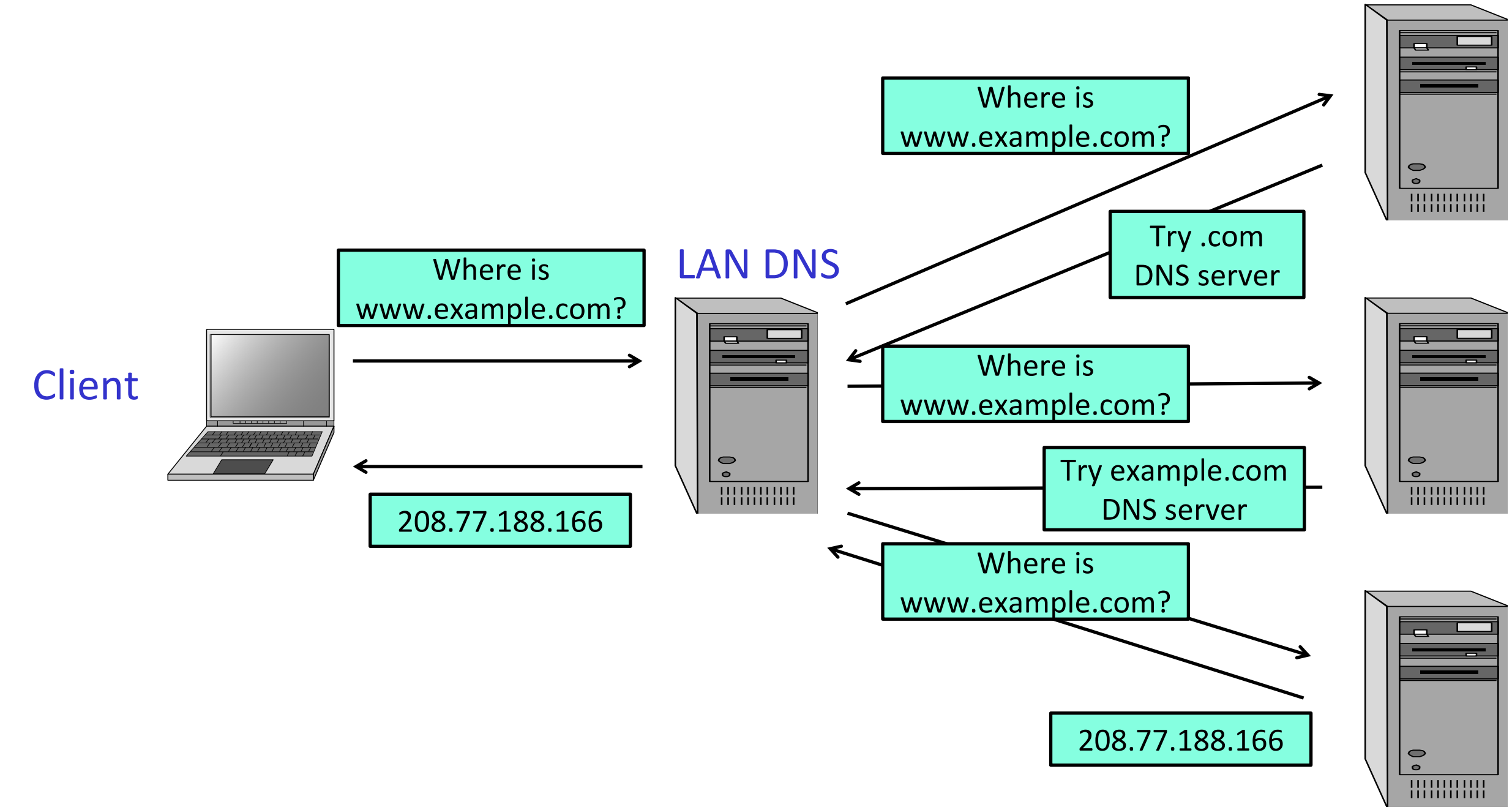
- Let’s say you need to find out IP address of “*www.kristiania.no*”
- First contact root server, which will answer by saying where to find server handling “*.no*”
- Then contact this latter one, which can give the IP or redirect to yet another DNS server in subdomain
- And so on recursively until finding a DNS server with the record for “*www.kristiania.no*”


Where is www.example.com?
Try .com DNS server
Where is www.example.com?
Try example.com DNS server
Where is www.example.com?
208.77.188.166
Client
root DNS server
.com DNS server
example.com DNS server

But how to find IP of DNS server?

- Could be hardcoded
- Or, in LANs, can be returned as part of DHCP messages when assigning IP address
- LANs will usually have their own DNS servers that query the main ones on the client behalf
  - Note the local IP 10.x.x.x addresses for the DNS servers (can be more than one)

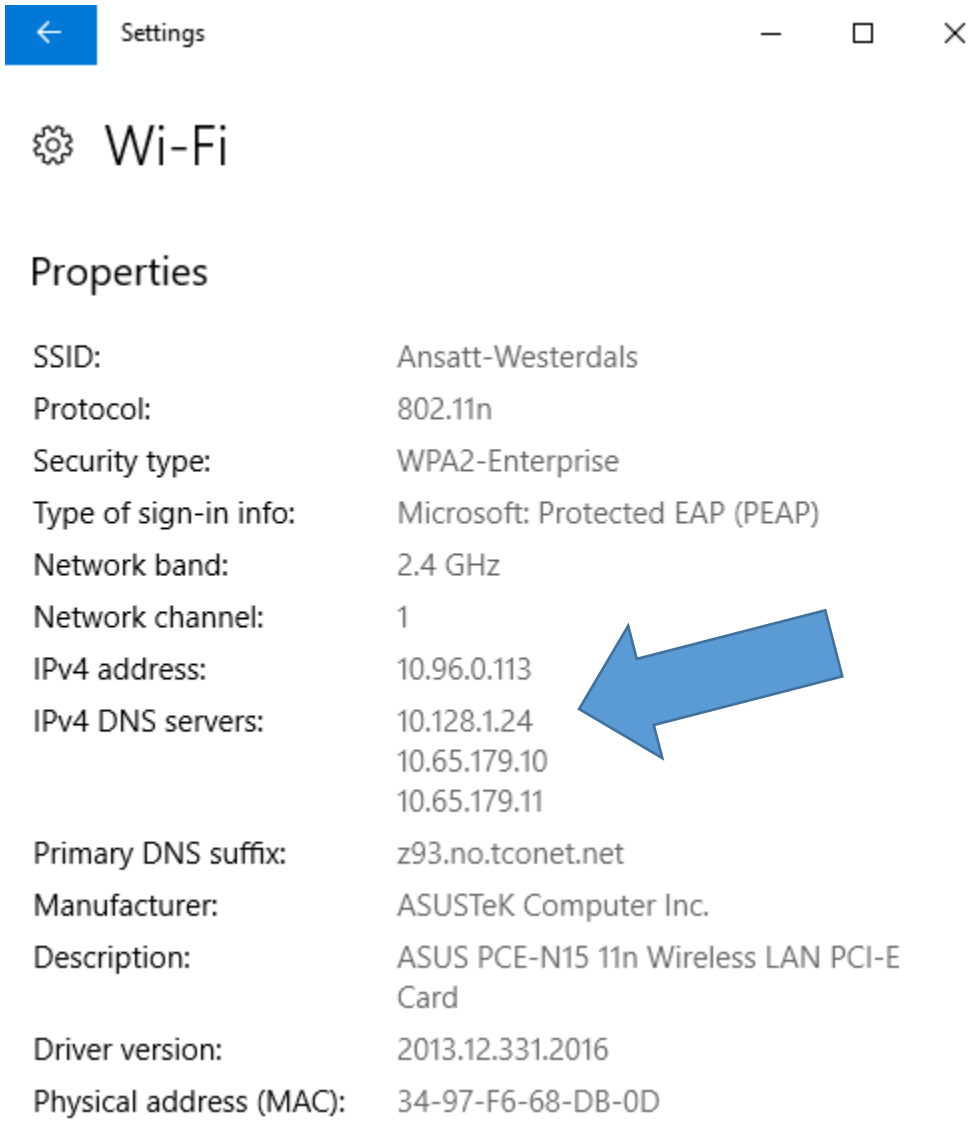



Client
Where is www.example.com?
LAN DNS
208.77.188.166
Where is www.example.com?
Try .com DNS server
Where is www.example.com?
Try example.com DNS server
Where is www.example.com?
208.77.188.166

Caching

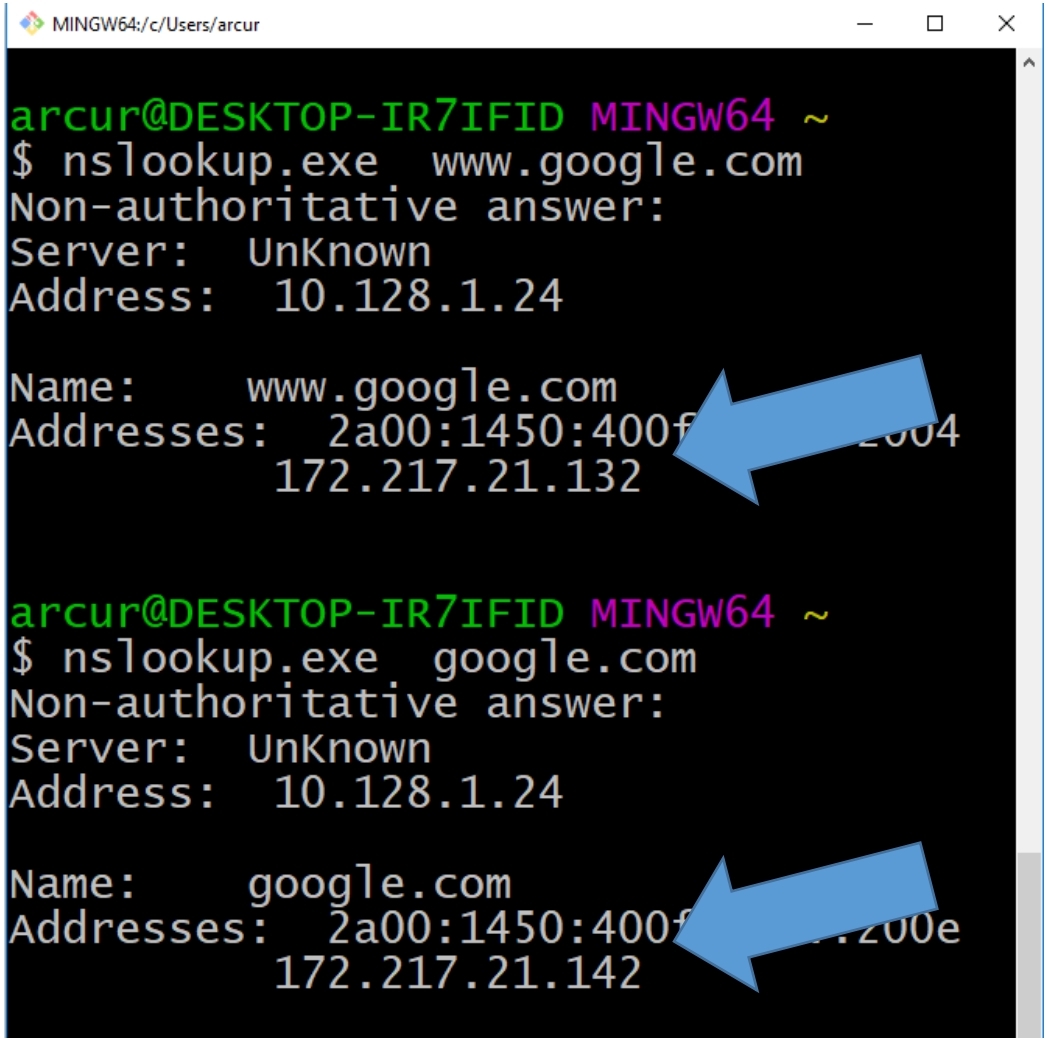
- Having to query the root DNS servers each time a hostname needs to be resolved would put a lot of stress on such servers
- There are *several billions* of machines connected to internet which would need to contact those DNS servers
- Using *caches* to avoid asking for hostnames that have already been resolved before
- Caches in different places: browser, OS, LAN DNS, etc

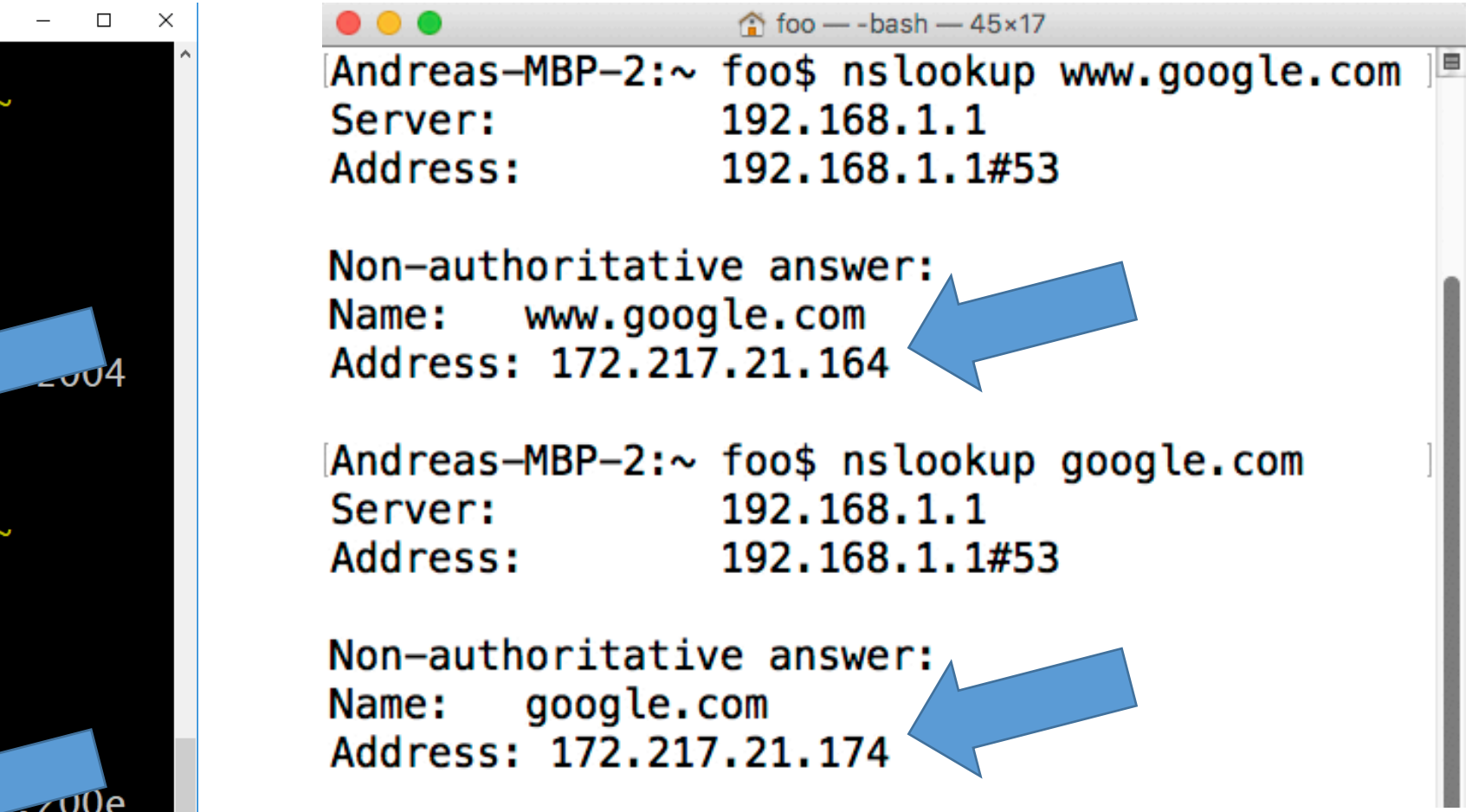
“nslookup”



- Note the IP of LAN DNS
- Getting both IPv4 and IPv6

- In this case, “*www.*” results in different IP address
- Also from different networks with different DNS caches, could get different IP addresses… eg Google does not have only a single machine serving “www.google.com” …





Time-to-Live (TTL)

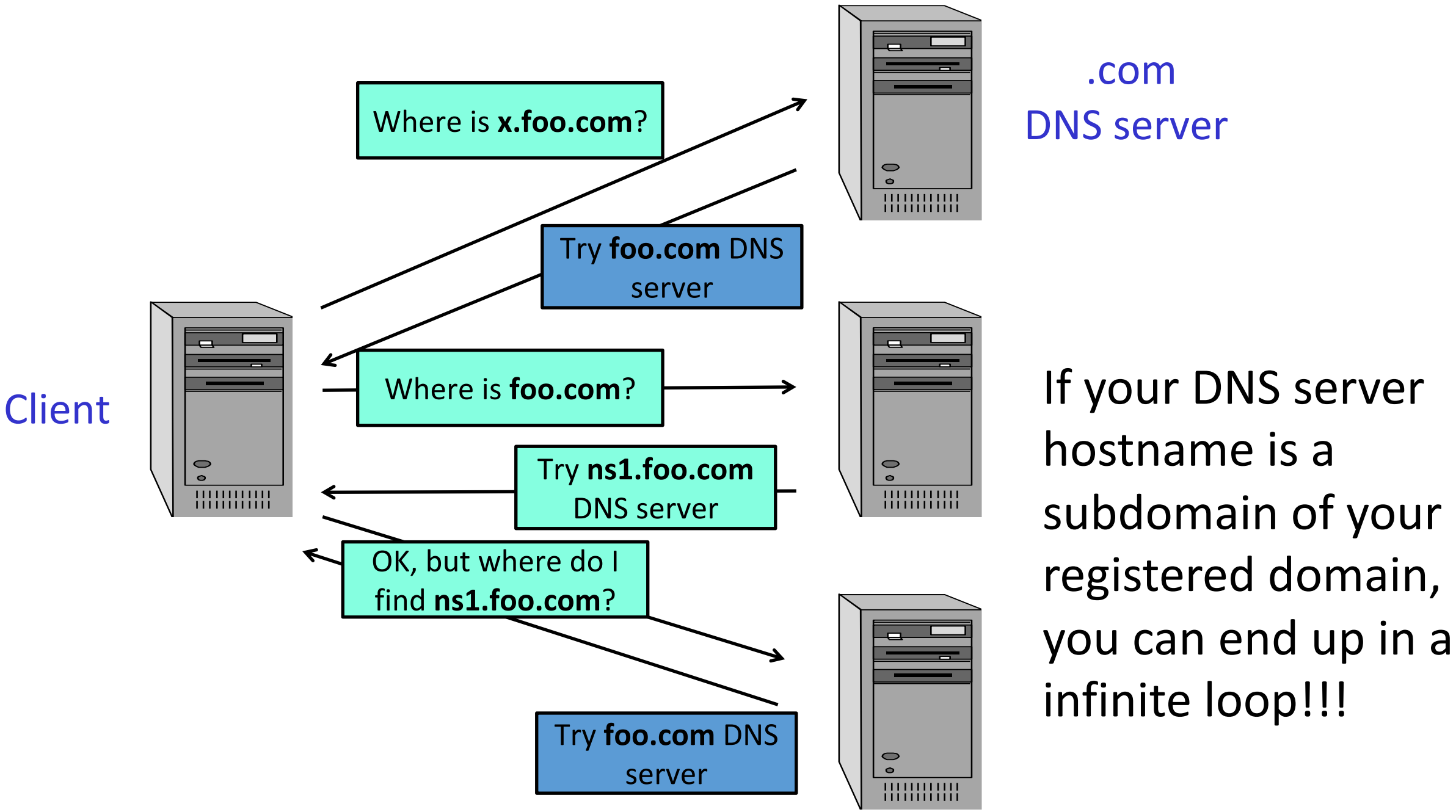
- Caching is critical for performance
- But mapping from hostnames to IP addresses can change
- In DNS protocol, each record has a TTL entry
  - eg, the mapping is only valid for 1 day
- When TTL expires, need to remove from cache, and ask DNS servers again
- Tradeoff between performance (large TTL) vs. getting more recent / correct mapping (small TTL)

Subdomain Resolution

- Assume you own "*foo.com*"
- You might want to have different subdomains for it, like:
  - *www.foo.com*, *email.foo.com*, *x.foo.com*, *y.foo.com*, etc.
- Example: you are a cloud provider, and want to give URLs for the web app you host:
  - *clientAppName.foo.com*, *anotherClientApp.foo.com*, etc.
- You could register all of these hostnames with the root "*.com*" DNS server, but expensive and does not scale (especially when customers choose their own app names)

Cont.

- Have your own DNS server under your registered “*foo.com*” domain
- For example, your own server DNS running at “*ns1.foo.com*”
- All your domains will be resolved by “*ns1.foo.com*” as the authoritative DNS for them
- Example: “*ns1.foo.com*” would have the IP mapping for “*clientAppName.foo.com*” and “*foo.com*”
- *Can you spot the problem here?*



If your DNS server hostname is a subdomain of your registered domain, you can end up in a infinite loop!!!

DNS Glue Record

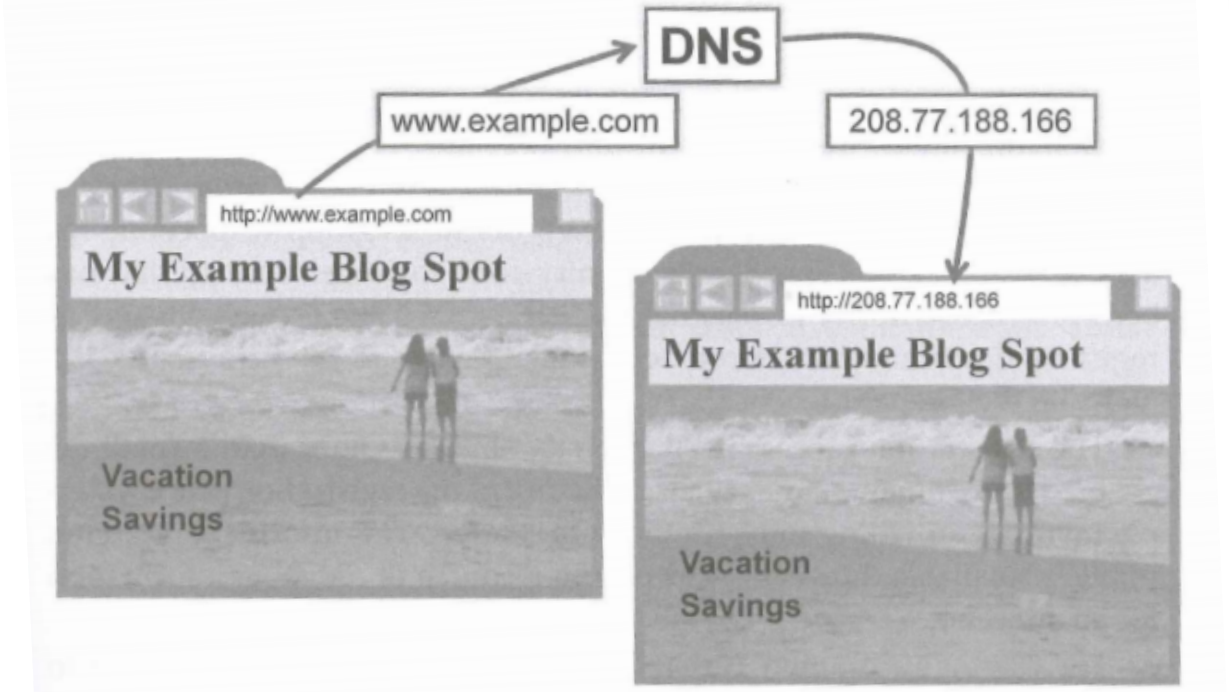
- To avoid infinite loop when subdomain “*ns1.foo.com*” is the authoritative DNS server for “*foo.com*”
- When querying *“.com*” DNS server for “*ns1.foo.com*” , return directly its IP address without redirecting to its parent “*foo.com*”, even though “*ns1.foo.com*” is not a direct subdomain of “*.com*”, and “*.com*” is not the authoritative DNS server for it

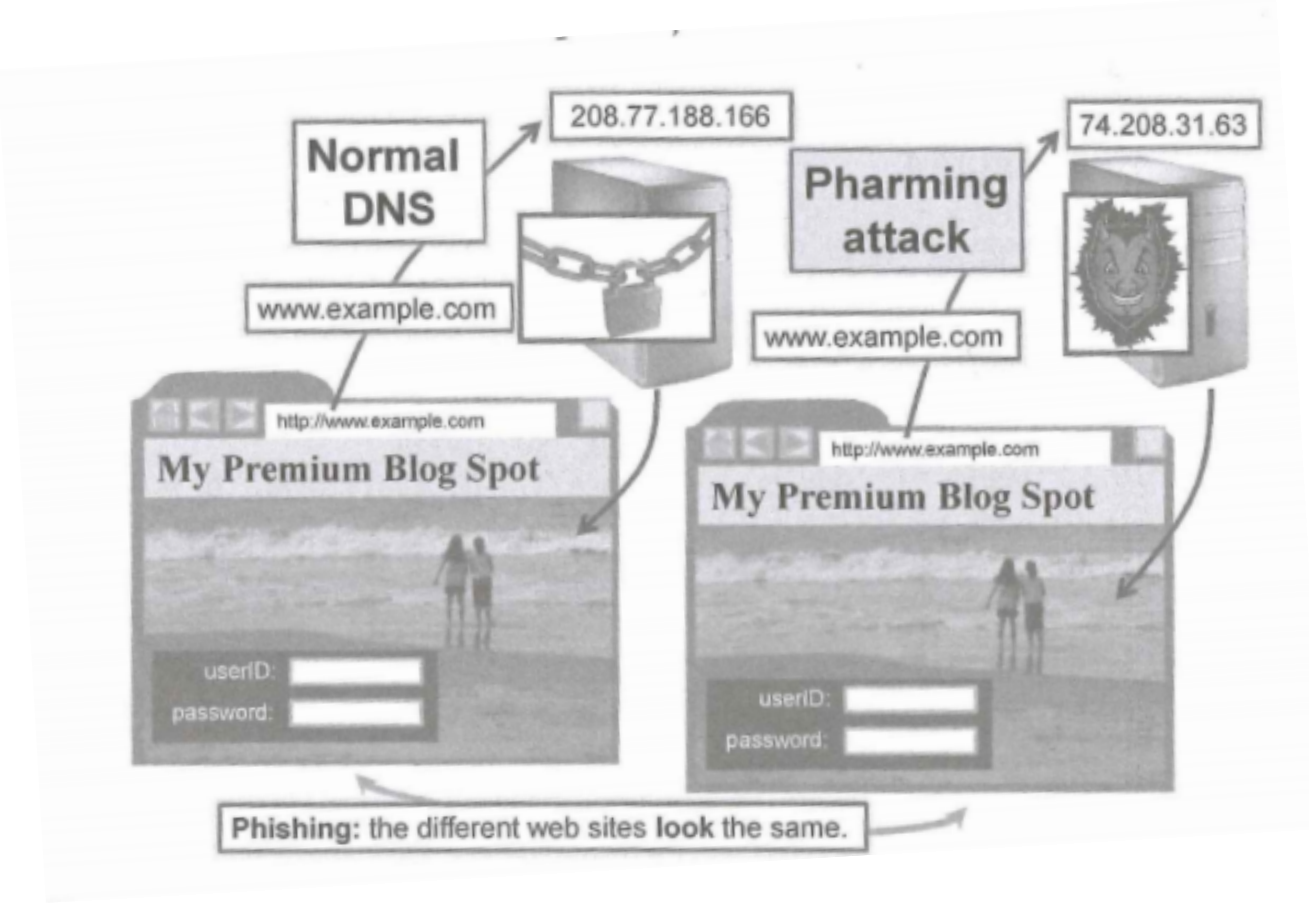
DNS Attacks

- If attacker can break the DNS protocol, could have disastrous consequences
- *Pharming*: alter DNS records to redirect to malicious server
  - User might think to communicate with real "*www.foo.com*", but instead is redirected to IP of a machine controlled by attacker

Pharming: Phishing

Trick user to type login/password in website that looks exactly the same



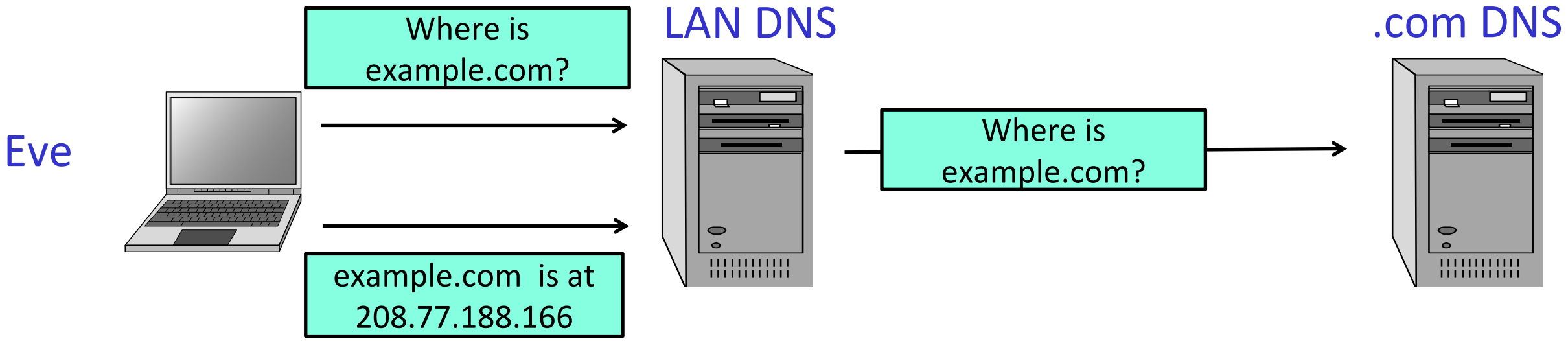


Pharming is not just Phishing

- OS update/patch servers
  - Provide malware that is installed like it was patch update
- Altering email servers
- Etc.

DNS Cache Poisoning

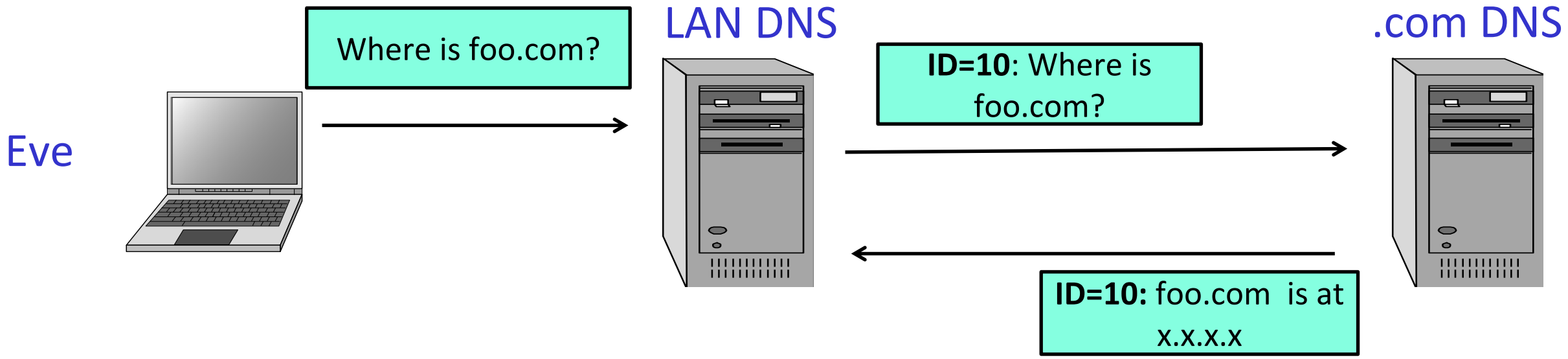
- Consider a target DNS server
  - eg the local one in a LAN
- A cache poisoning attack is to alter its DNS records in the cache, eg by pointing them to your malicious servers

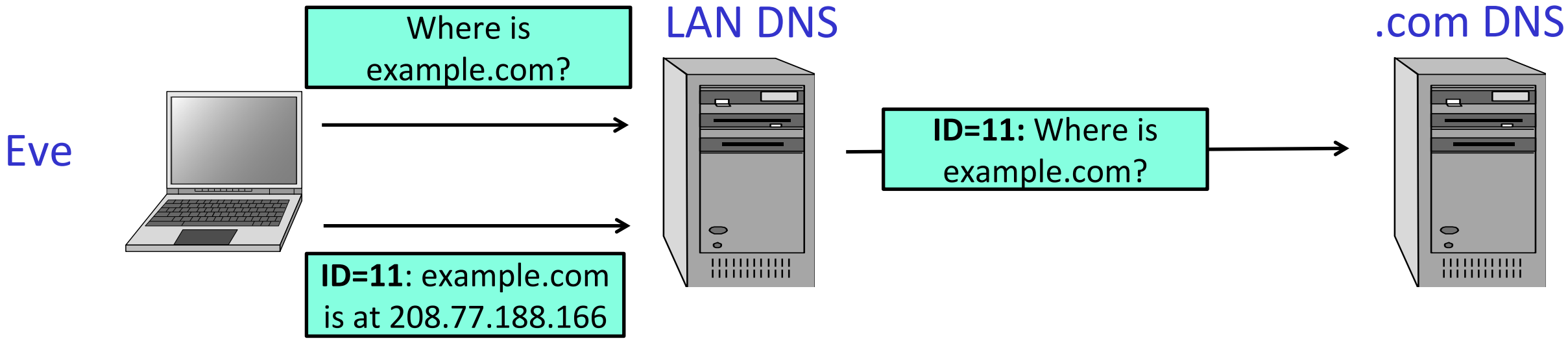


- Assume Eve wants to poison *example.com*
- Eve sends DNS request for *example.com*
- LAN DNS will send request to the authoritative DNS servers
- **BEFORE** those do answer, Eve sends a crafted DNS reply to her own request stating that *example.com* is at an IP address of her choice
- Eve needs to be *fast:* her message must arrive before "*.com*" replies

DNS Transaction Ids

- To avoid spoofed DNS replies, DNS requests/responses do have a transaction id
- When LAN DNS sends request to "*.com*" DNS, it will use an **id**, and expect the same id as response from "*.com*"
- If different id, the DNS replies are ignored
- Eve needs to predict which id the LAN DNS will use in its requests
- Until 2002, most DNS servers were just using an incremental id
- Eve just needed to intercept 1 request of LAN DNS, and will know the ids of all the following requests


Eve
Where is foo.com?
LAN DNS
ID=10: Where is foo.com?
.com DNS
ID=10: foo.com is at x.x.x.x


Eve
Where is example.com?
LAN DNS
ID=11: Where is example.com?
.com DNS
ID=11: example.com is at 208.77.188.166

Randomized Ids

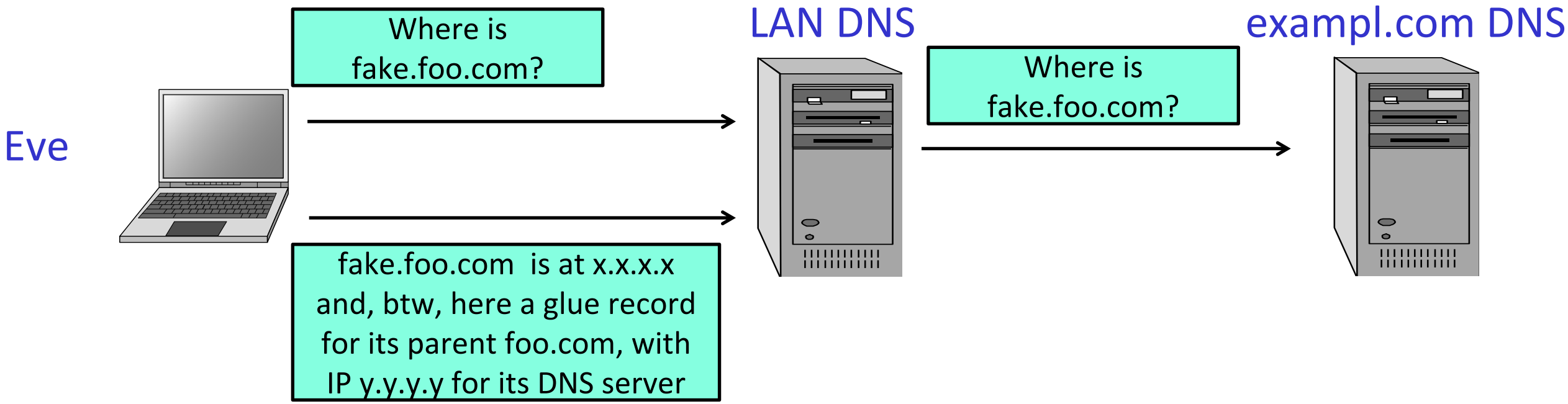
- Since 2002, DNS servers use randomized ids for the requests
- DNS ids are 16-bit numbers, so 65_536 values
- Eve can send more than malicious 1 reply, with different ids, hoping to get the right id
- Just need **213** spoofed replies to have a **50%** chances to get the right id
  - If interested, look at the Birthday Paradox to see the math behind it

Is DNS Cache Poisoning viable?

- The time-frame for Eve to operate is too short
- Still a chance that Eve guesses the right id, but quite low
- Once a DNS record is cached, its TTL value might be *days*
- In such cases, Eve would have to wait *days* before being able to attempt again
  - ie need to wait until TTL expires
- The attack is *feasible*, but not so *viable*

Subdomain DNS Cache Poisoning

- Type of viable attack found out in 2008
- Let's say Eve wants to poison "*foo.com*"
- Eve makes false requests for non-existent subdomain like "*idonotexists.foo.com*"
- LAN DNS server will send requests to the DNS of "*foo.com*", which will NOT reply (as subdomain does not exist)
  - DNS protocol does not send error messages when hostname does not exist
- Eve crafts her own replies for "*idonotexists.foo.com*", and does not need to worry to beat in time the DNS server of "*foo.com*", as that will never reply



- Poisoning the DNS entry of non-existent subdomain like "*idonotexists.foo.com*" achieves nothing in itself…
- … but, Eve can have a "glue" record with a poisoned IP address for "*foo.com*"

How to defend from this?

- LAN DNS are now usually not accessible from outside the LAN (it was not the case before 2008)
  - So, to do this attack, you need to be on the LAN
  - *And yes, trying this attack on the LAN DNS in school is a really, really bad idea… unless you are planning to get expelled*
  - But what about open WiFi in bars/restaurants/etc. ???
- Port randomization
  - Each time DNS sends requests, use different UDP port, and expect answer on same port
  - Eve not only needs to guess DNS transaction id, but also the used UDP port, so around 64k times more combinations
  - Make Eve's life harder, but attack is still viable

Client Side Poisoning

- Instead of just attacking a LAN DNS server, can attack specific users and the cache on their machines
- Can craft web pages requesting resources (eg images and CSS files) in non-existent subdomains
  - When browser loads the page, it will ask for the DNS resolution of these resources
- *Need a way to make user visit Eve's web application*
- Once Eve knows that the page has been accessed, can send DNS replies to that user's machine (the LAN DNS would not answer)

DNSSEC

- Current DNS can be broken, as *intrinsically insecure*
- Need to use a different protocol
- DNSSEC is a DNS extension with encryption
- But to work, both client and server need to support it
- Slowly starting to replace DNS

Social Engineering and Phishing

Open WiFi

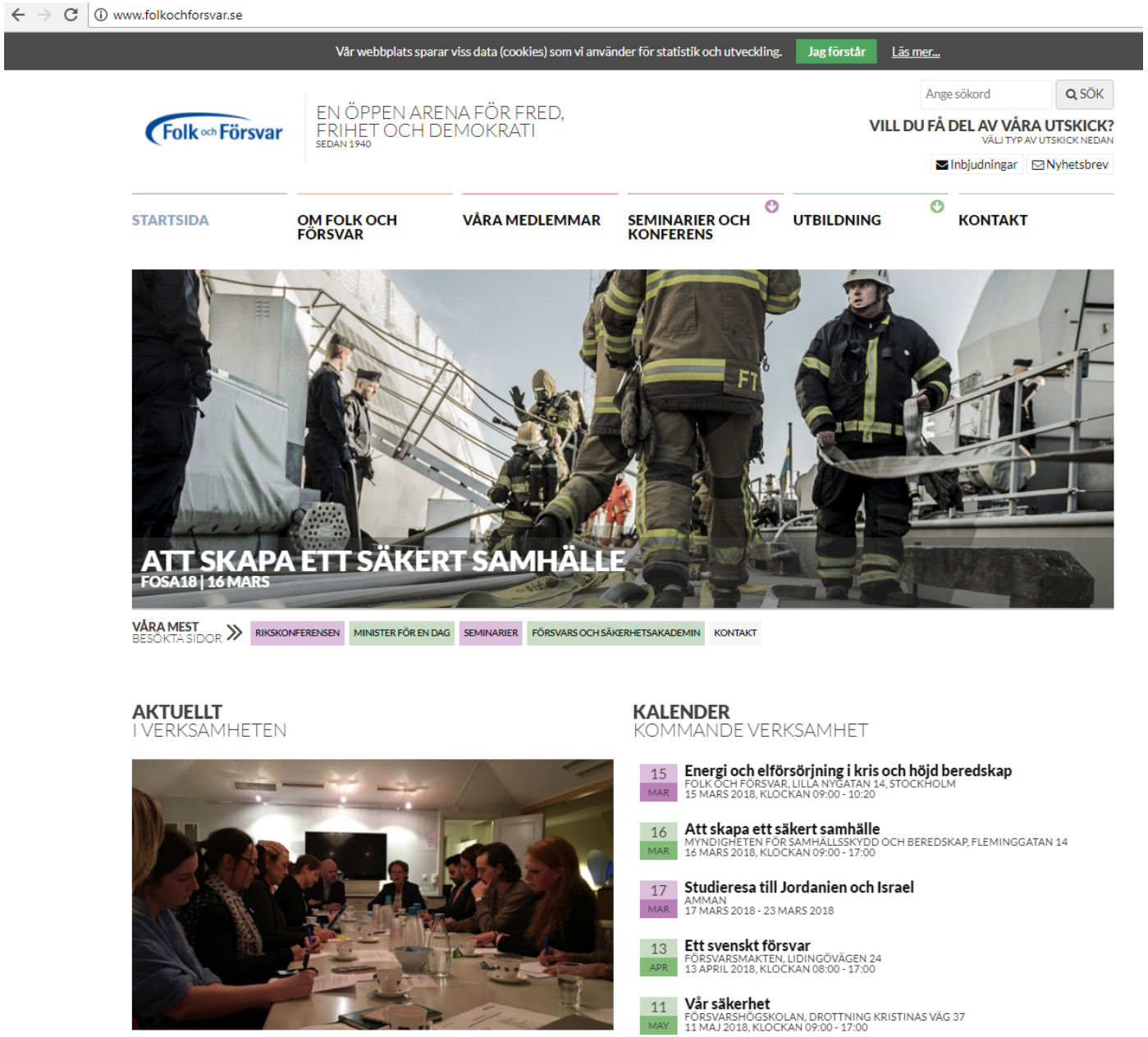
- Instead of poisoning an existing DNS server, have your own, and make users to connect to it
- But how to lure them?
- Just provide your own free WiFi service with a router, and give it an appealing name
- When users connect to your router, it is you that give them the IP address and DNS server info…

Is it legal?

- If you have illicit intent (eg, stealing info), then yes, most likely it is illegal
- *However, depending on the country, accessing a WiFi is illegal if the owner of the WiFi does not give permission*
- Eg, if someone connects to your WiFi, it might be *them* that are breaking the law
- Florida 2005: man faced felony charges for using a neighbor's internet connection
- Put in a different way: if you forget the door of your house open, that does not authorize people passing by to come in…

Let’s Talk About Sweden…

- …and let's talk of the *"Folk och Försvar"* ("People and Defense") conference, which is the *top security conference* in Sweden
- Attendees of *security experts*, *politicians* (including ministers), *journalists*, etc.



2015 Edition…

- Activists from Pirate Party did set up an open WiFi called "*Open Guest*"
- And of course most of the politicians and security "experts" used it…
- Why *extremely bad*? Accessing confidential emails from government servers using non-encrypted protocols…
- https://falkvinge.net/2015/01/14/hilarious-activists-turn-tables-on-political-surveillance-hawks-wiretaps-them-with-honeypot-open-wi-fi-at-security-conference/

Phishing

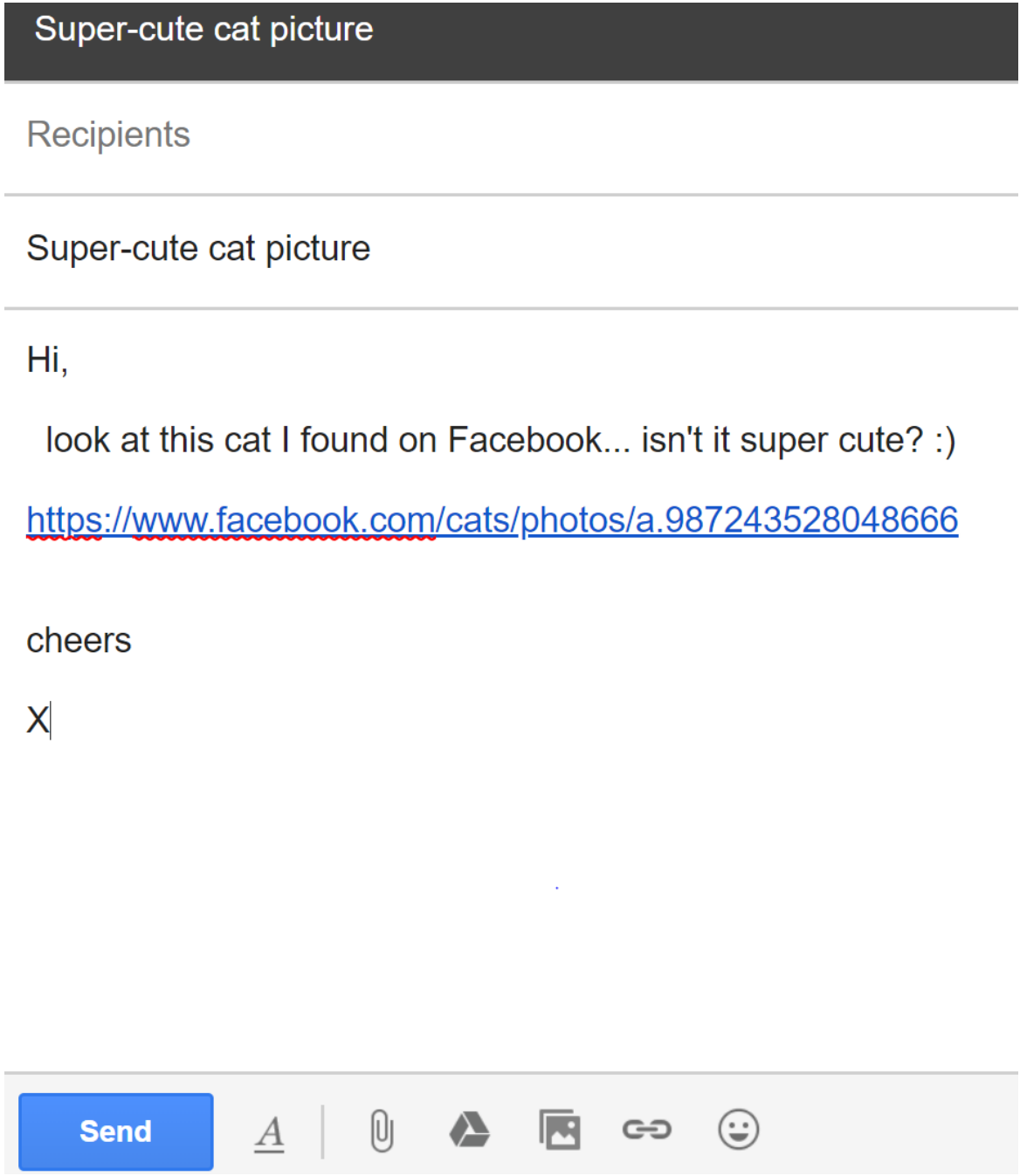
"Phishing is the attempt to obtain sensitive information such as usernames, passwords, and credit card details (and money), often for malicious reasons, by disguising as a trustworthy entity in an electronic communication." Wikipedia

Web Site Phishing

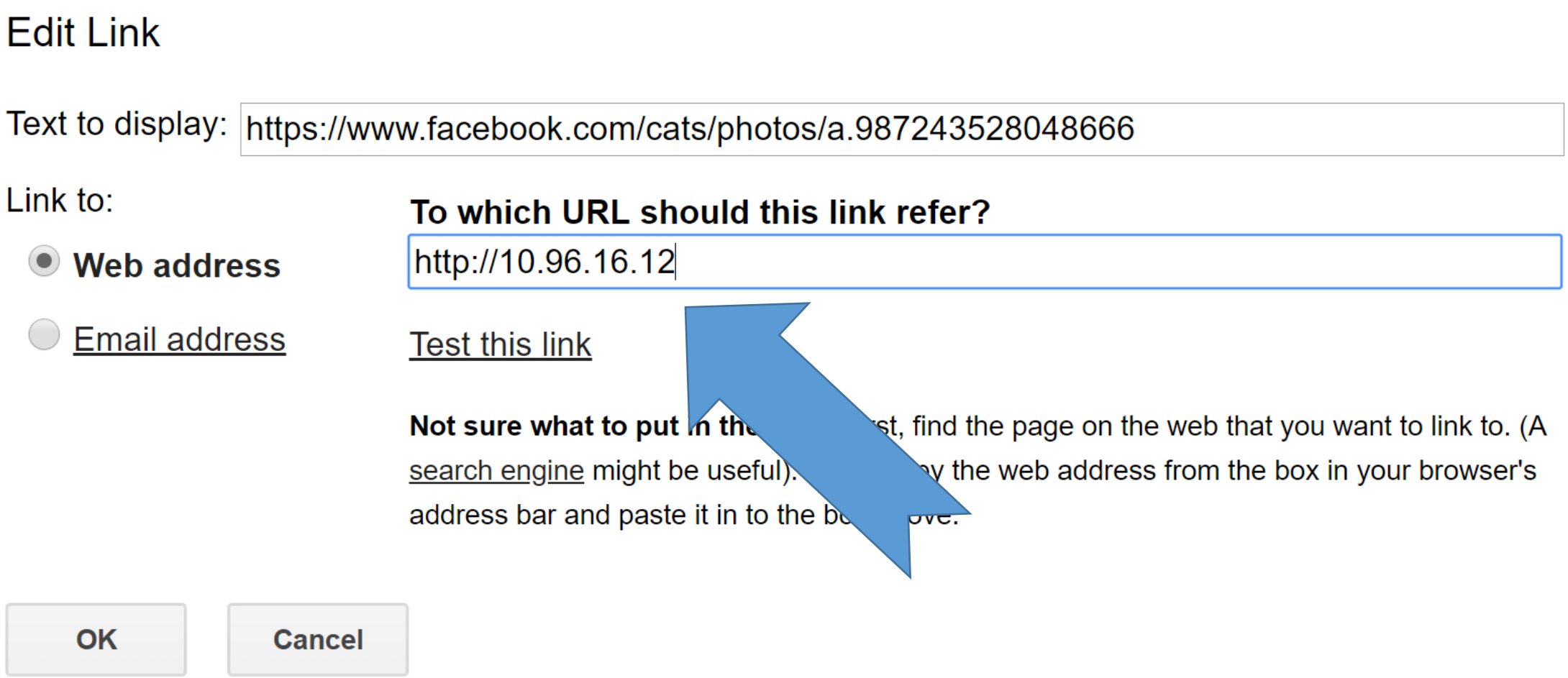
- Somehow you need to trick a user to visit one of your malicious servers
- You create a web page resembling the original target website (Facebook, Google, Amazon, their bank, etc)
- Log when they type login/password
- *But how to trick a user to visit your web site?*
- Can use DNS poisoning
- Can use web "*links*" in emails/chats/etc.

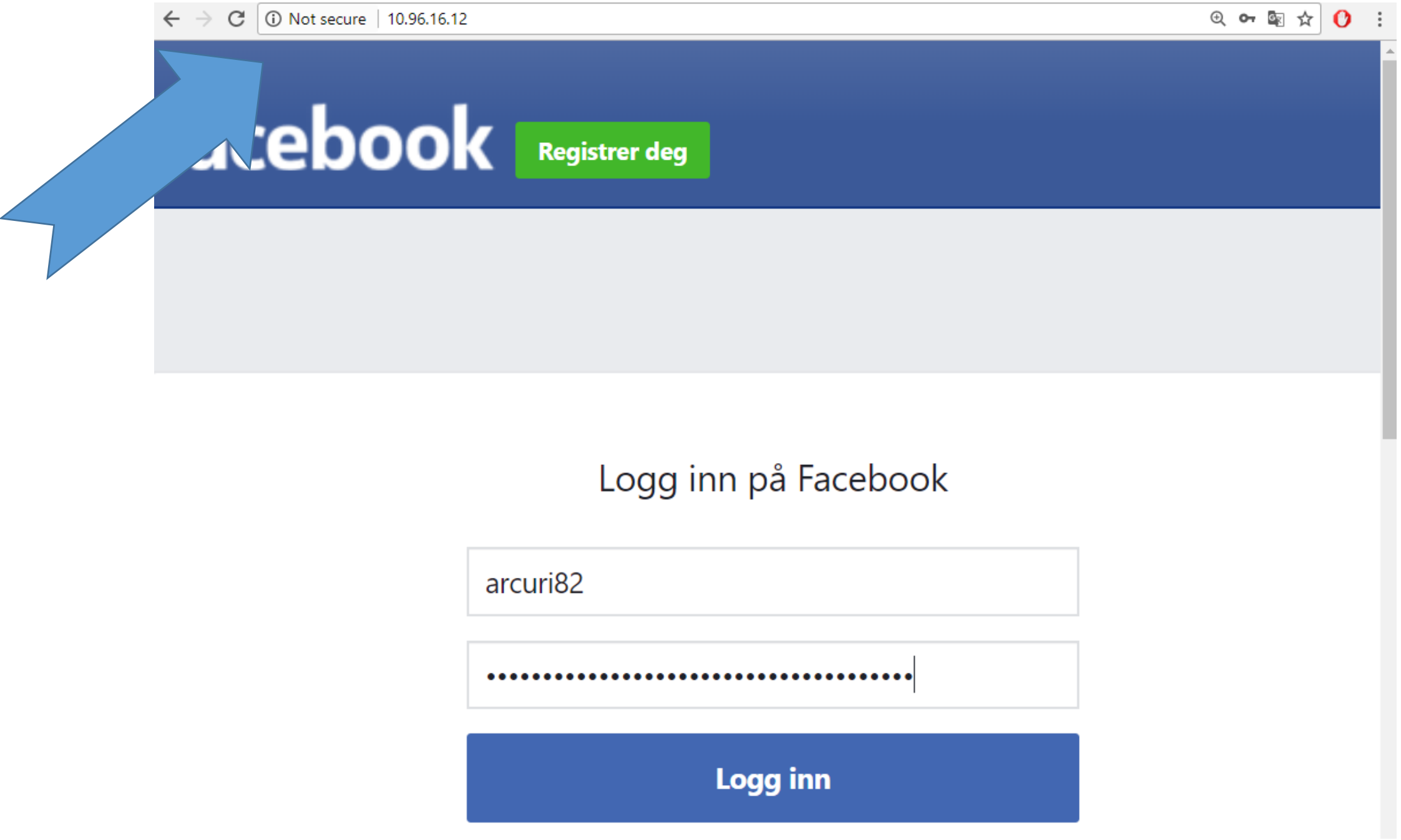
The ultimate phishing email???

Who would have the strength to restrain from compulsory clicking that link without thinking twice???

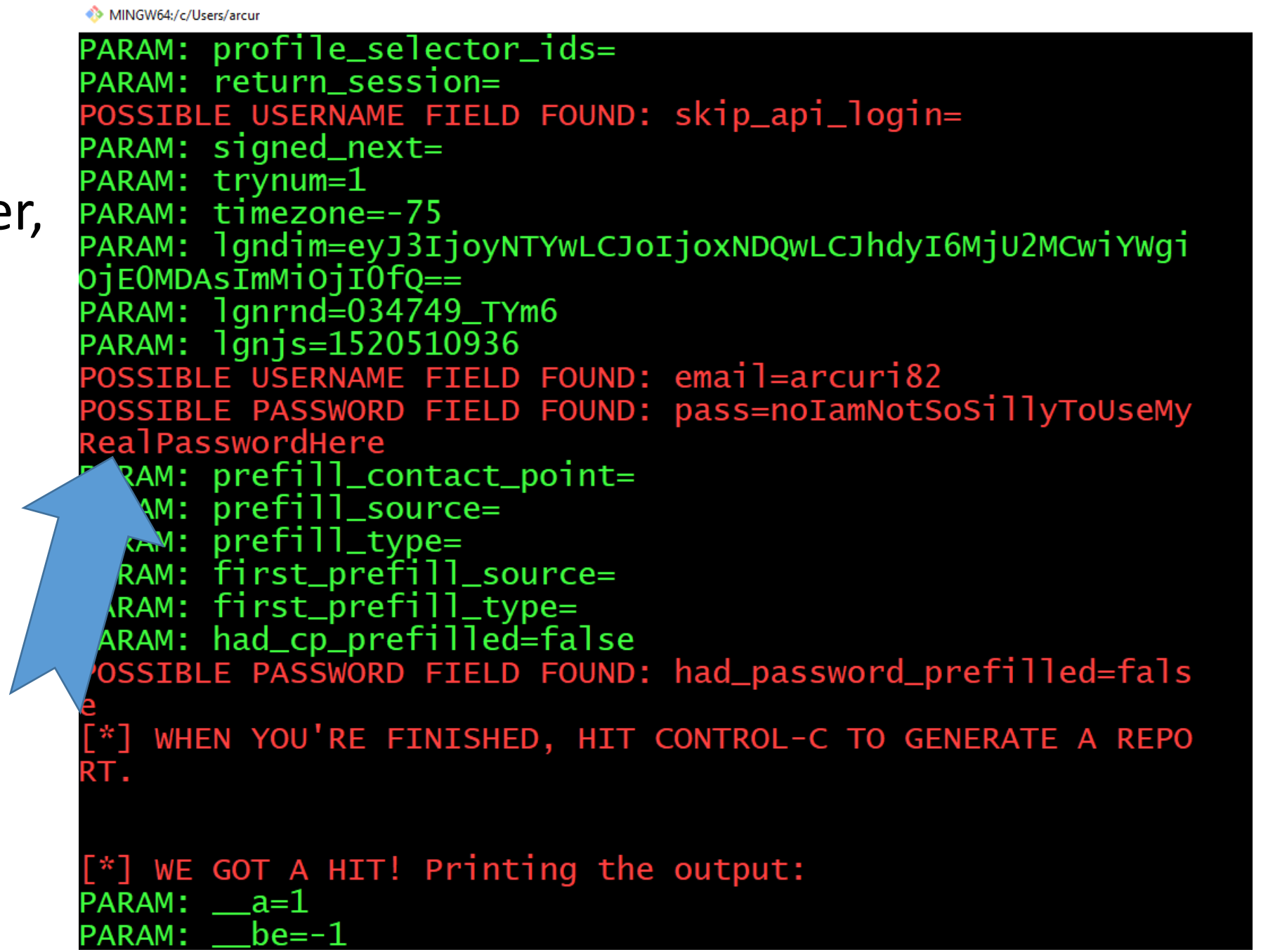


But the link is actually pointing to your evil web site… on which you are running a clone login page of Facebook…

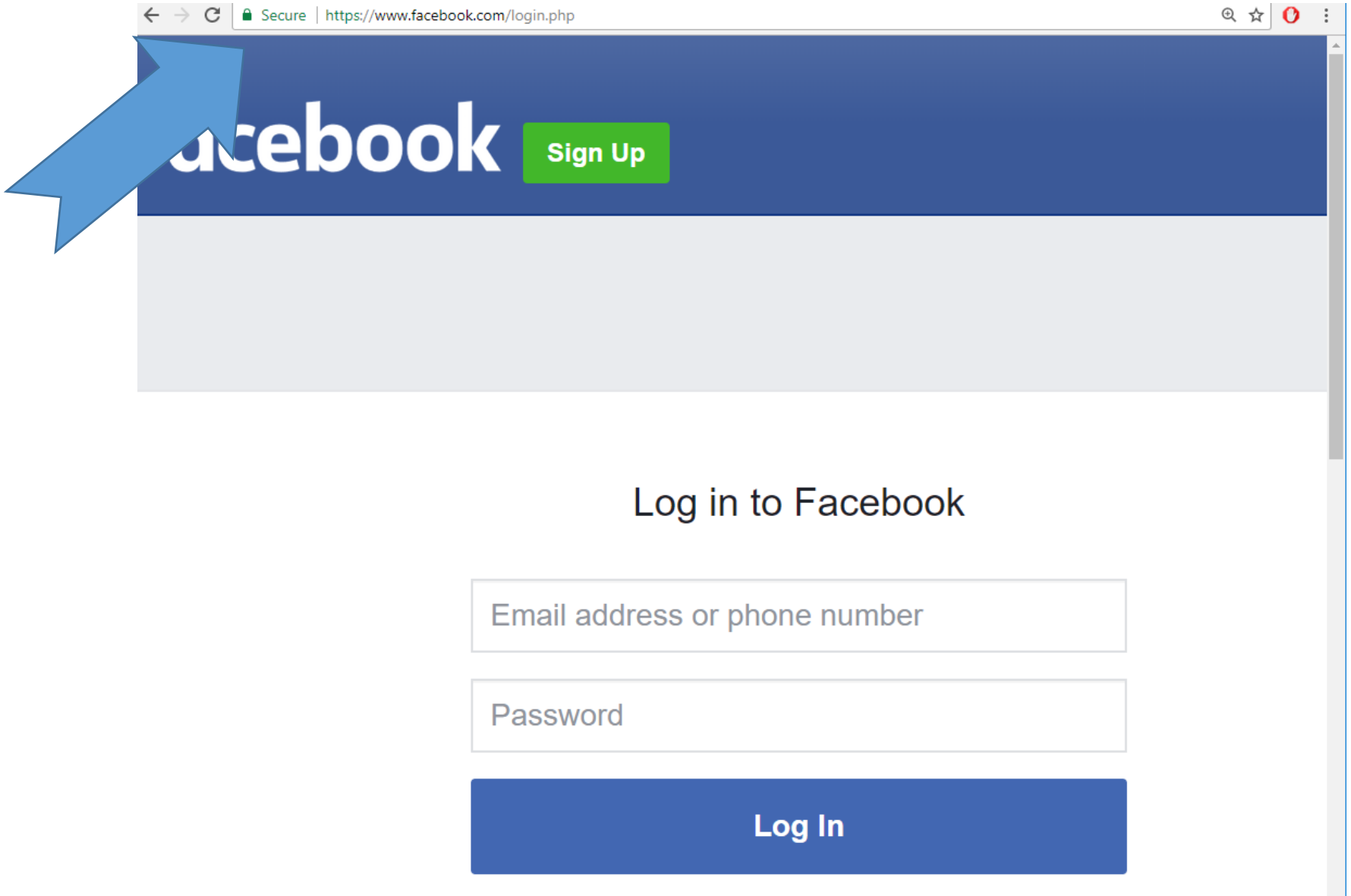



Not secure
10.96.16.12
cebook
Registrer deg
Logg inn på Facebook
arcuri82
Logg inn

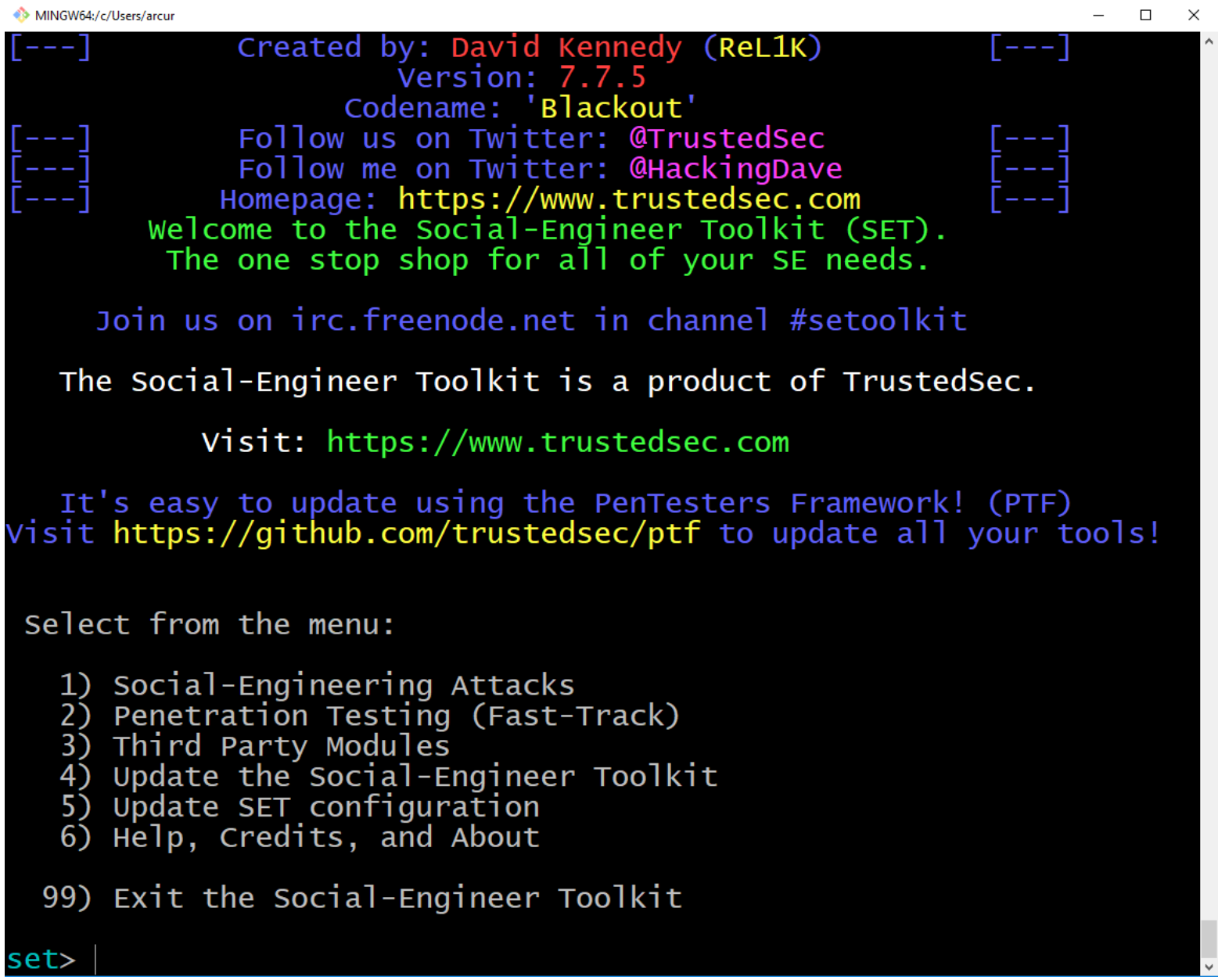
The POST in that HTML form does point to your server, where you can analyze if any sent data is a login/password



After first login attempt and password stolen, redirect to real website, so user would not suspect of what happened…



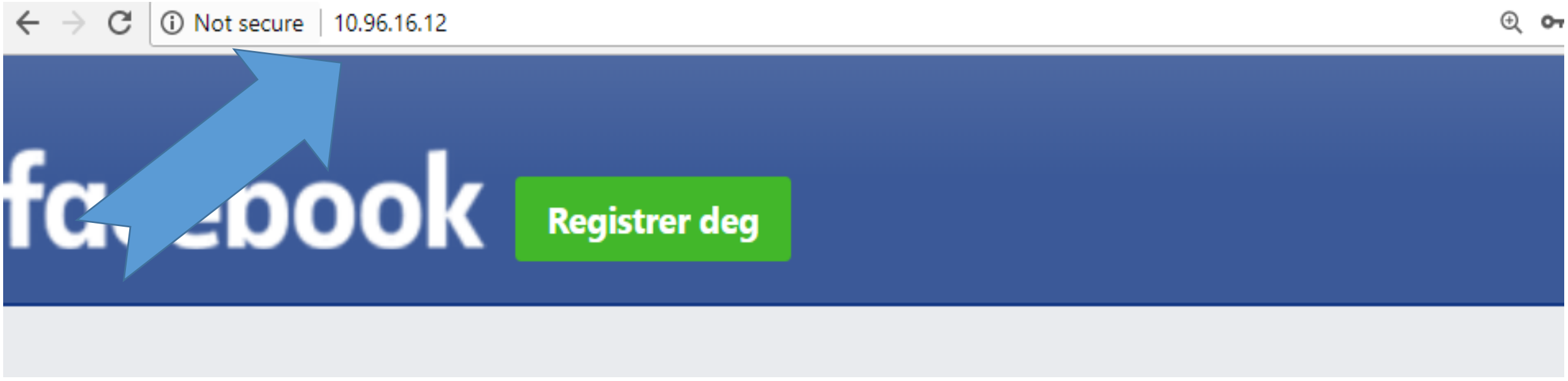
- But it sounds like a lot of work…
- Well, there are tools like *setoolkit* that can automate it
- Creating a phishing site just takes a couple of minutes…

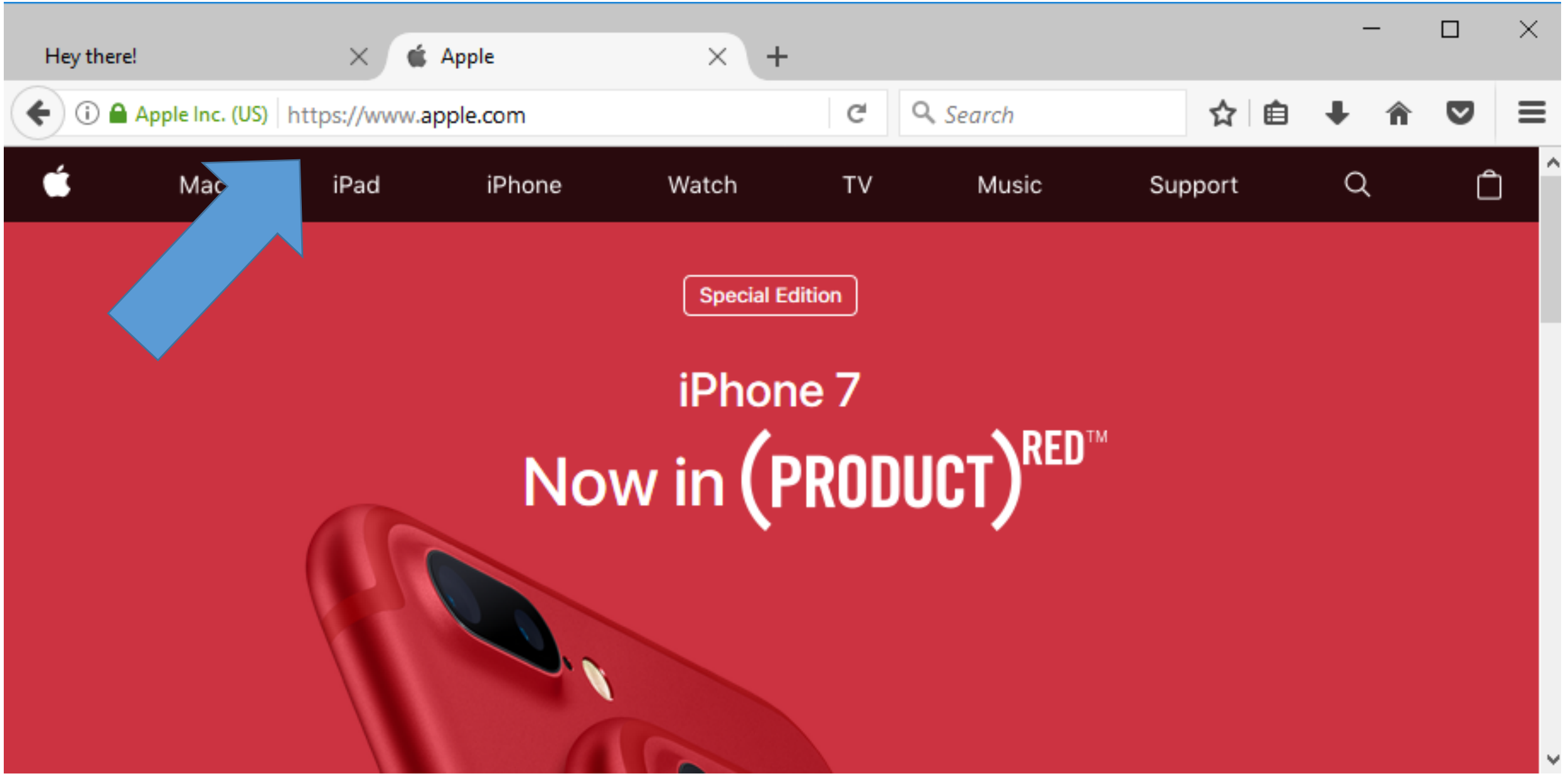


- **DO NOT DO IT**
- It is **ILLEGAL**, so target could call the police on you…
- But, if you still do it, and have the brilliant idea to do it to your *boyfriend/girlfriend* as a prank, the police will be the least of your problems…
- However, it will be the exercise for this week, between knowing students (but do not use your real passwords…)

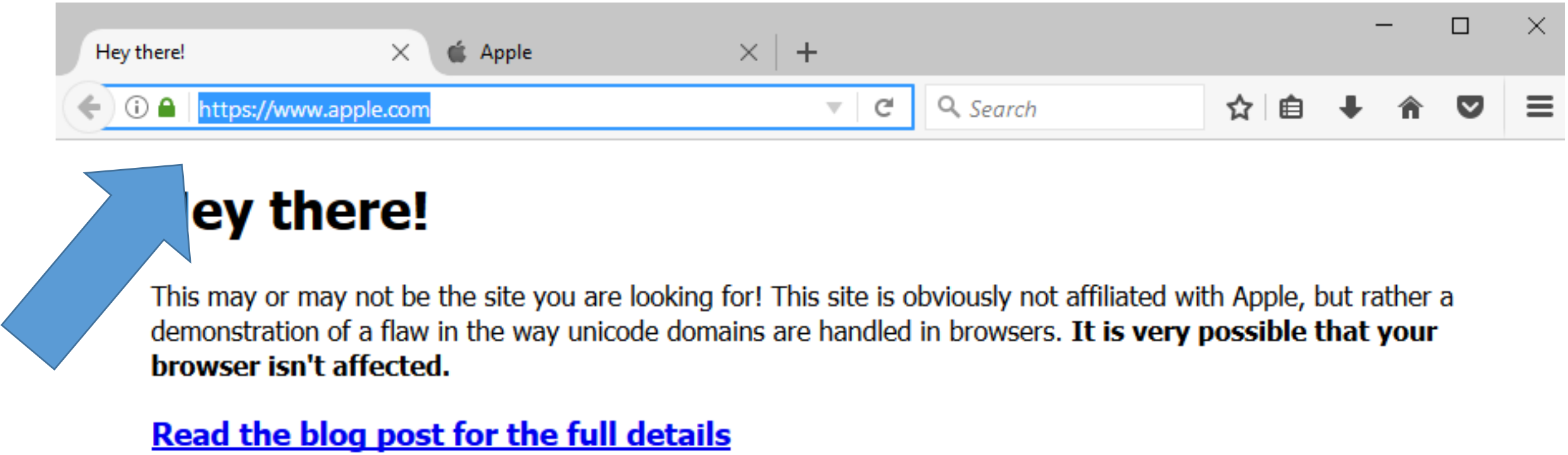
DEMO

- But would not the user see that the URL is weird???
- Most people will have no clue…
- To be more *stealthy*, you could register a hostname which is similar to the target
- What happen if you type "*facebok.com*" (missing an "o") in the browser?
- To avoid phishing attacks, when you buy a hostname, also needs to buy hostnames that are similar (eg typos)




Hey there!
Apple
Apple Inc. (US)
https://www.apple.com
Search
Mac
iPad
iPhone
Watch
TV
Music
Support
Special Edition
iPhone 7
Now in (PRODUCT)RED™

Still visiting www.apple.com, but this time the page looks slightly different… isn’it?



What the heck is going on???

- The web page is obviously not the real www.apple.com
- The actual page is www.xn--80ak6aa92e.com
- *Punycode*: way to express UNICODE symbols with subset of ASCII for URLs
- Important for displaying URLs with symbols from Chinese, Japanese, etc.
  - Eg, “xn--s7y.co” gets displayed in a browser URL bar with “短.co”
- Cyrillic (Russia) has many symbols that are similar to Latin ones, but different Unicode values, eg Cyrillic "a" (U+0430) is different from ASCII "a" (U+0061)
- “xn--80ak6aa92e” is representation of “apple” in punycode using only Cyrillic letters/fonts
- At time of writing this slide, issue fixed in Chrome but not Firefox…

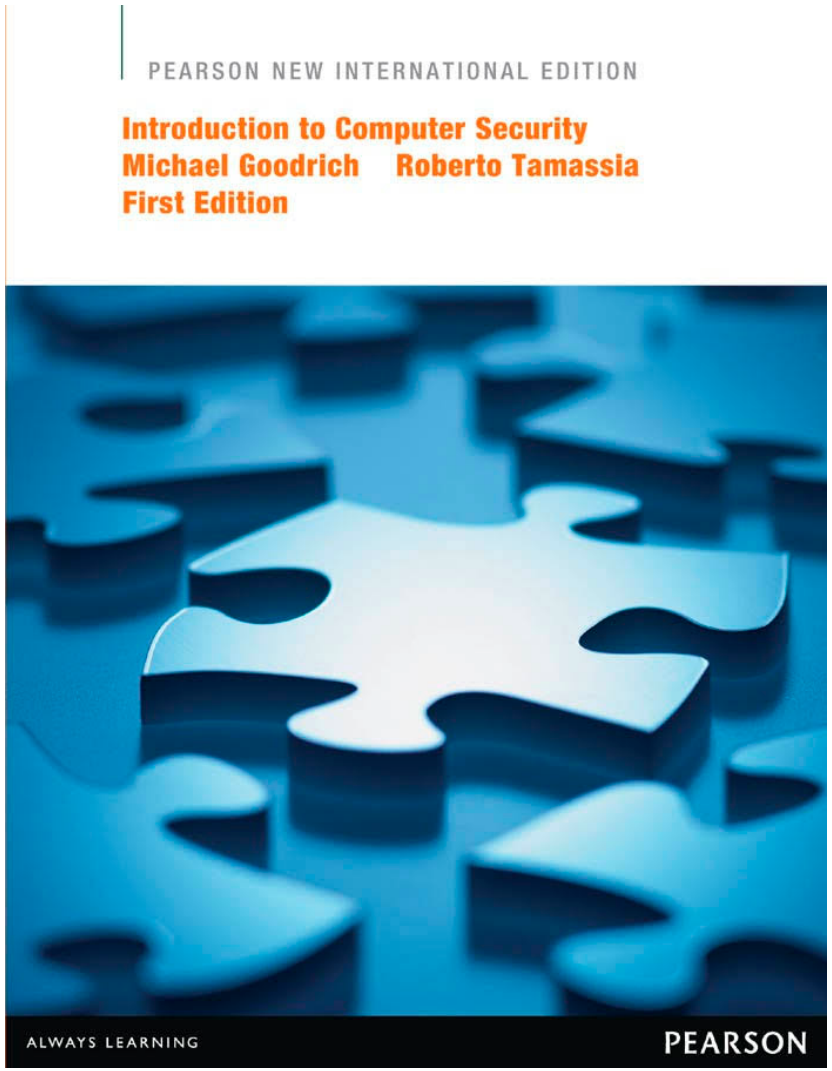
Letters of the Cyrillic alphabet (see also Cyrillic digraphs)

А A	Б Be	В Ve	Г Ge	Ґ Ge upturn	Д De	Ђ Dje	Ѓ Gje	Е Ye	Ё Yo	Є Yest	Ж Zhe
З Ze	З́ Zje	S Dze	И I	І Dotted I	Ї Yi	Й Short I	Ј Je	К Ka	Л El	Љ Lje	М Em
Н En	Њ Nje	О O	П Pe	Р Er	С Es	Ć Sje	Т Te	Ћ Tshe	Ќ Kje	У U	Ў Short U
Ф Ef	Х Kha	Ц Tse	Ч Che	Џ Dzhe	Ш Sha	Щ Shcha	Ъ Hard sign (Yer)	Ы Yery	Ь Soft sign (Yeri)	Э E	Ю Yu
Я Ya											

Important Cyrillic non-Slavic letters

Ӏ Palochka	Ә Cyrillic Schwa	Ғ Ayn	Ҙ Bashkir Dhe	Ҫ Bashkir The	Ҟ Bashkir Qa	Ӂ Zhje	Қ Ka with descender	Ң Ng	Ҥ En-ghe	Ө Barred O	Ү Straight U
Ұ Straight U with stroke	Һ Shha (He)	Ҳ Kha with descender	Ӑ A with breve								

For Next Week



- Book pages: 270-286, 349-350
- Note: when I tell you to **study** some specific pages in the book, it would be good if you also *read* the other pages in the same chapter at least once
- Exercises for Lesson 7 on GitHub repository