Network Security

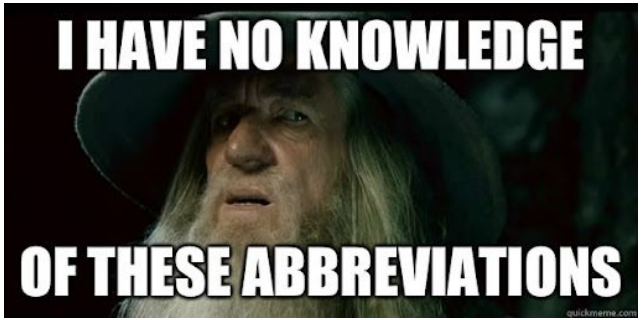
Introduction to Security (192.019)

Sebastian **Roth**

Security & **Privacy** Research Unit (192-06)
https://**secpriv.wien**

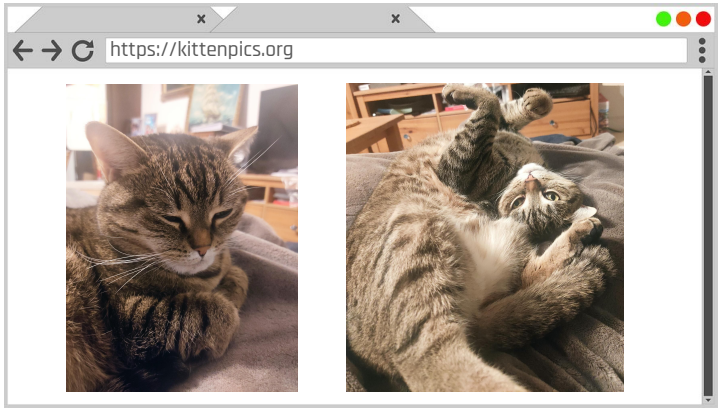
Overview

- Recap on Network Topology
- Threat Models / Attacks & Defenses
  - ARP Spoofing
  - BGP Hijacking
  - TCP Injection
  - DNS Poisoning
  - DOS / Amplification Attacks
  - Portscans
- General Defense Techniques
  - Firewalls
  - Tunneled Networks



Recap: What is a Network?

Networking Example: The Web

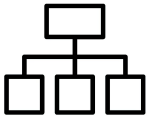


Networking Layers

Connected to a network

Who is my DNS Server?

What is the IP of kittenpics.org



Establish TCP connection

Upgrade to HTTPS

Network Topology

LAYER	EXAMPLE
Network Access Layer	Cable, WiFi, MAC, ARP
Network Layer	BGP, IP, etc.
Transport Layer	TCP, UDP, Ports, etc.
Application Layer	DNS, HTTP, VPN, etc.

Network Topology

LAYER	EXAMPLE
Network Access Layer	Cable, WiFi, MAC, **ARP**
Network Layer	**BGP**, IP, etc.
Transport Layer	**TCP**, UDP, **Ports**, etc.
Application Layer	**DNS**, HTTP, **VPN**, etc.

Threat Models in Networks

Network Topology

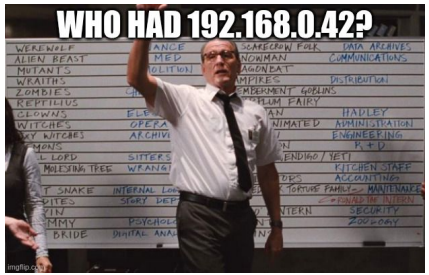
LAYER	EXAMPLES	ATTACKS
Network Access Layer	Cable, WiFi, MAC, **ARP**	Sniffing, ARP Spoofing
Network Layer	**BGP**, IP, etc.	BGP Issues, MITM, etc.
Transport Layer	**TCP**, UDP, **Ports**, etc.	TCP Spoofing, SYN Flooding, Port Scans etc.
Application Layer	**DNS**, HTTP, **VPN**, etc.	DNS Spoofing

Threats: ARP Spoofing

Address Resolution Protocol (ARP)

IPs can be handed out dynamic, so the Network Access Layer needs to keep a table up to date which network interface (MAC-Address) has which IP address.
This is taken care of by the **Address Resolution Protocol (ARP)**.

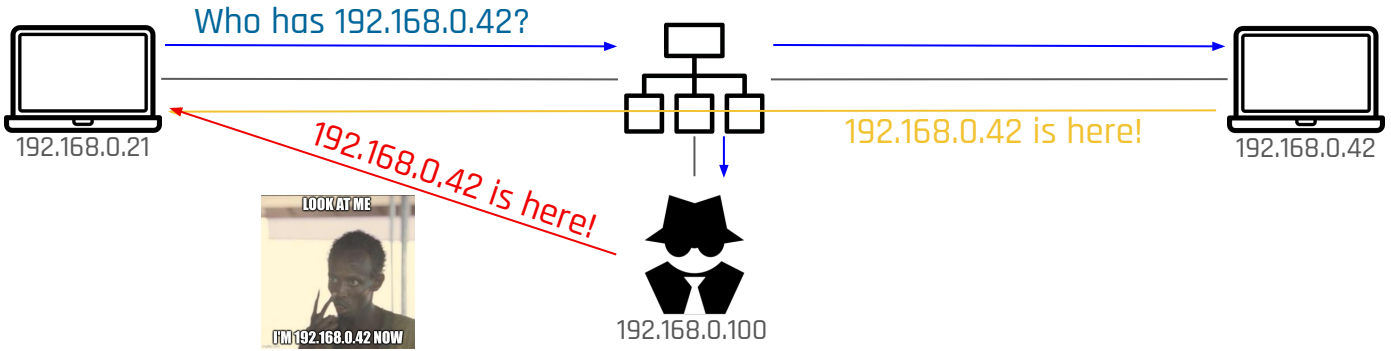
To find out the MAC address, a broadcast is sent out asking who has a certain IP. The device then responds by declaring its MAC address.




```
Who has 192.168.0.198? Tell 192.168.0.1
192.168.0.198 is at 08:00:27:40:05:2e
```

A screenshot from wireshark showing an ARP exchange

ARP Spoofing



ARP is **not** authenticated, so any device can respond to the request. If an attacker manages to convince a client and the router that the respective other device has the MAC address of the attacker, they can insert themselves in the path enabling Machine-in-the-middle (MITM) attacks. This impersonation is called **ARP spoofing**, however, the timing is critical as most implementations ignore unsolicited responses.

Mitigation of ARP Spoofing

ARP spoofing detected is easily …

- if single IP seemingly is connected to multiple ports, an intelligent switch can simply detect spoofing if for a single IP, multiple MAC addresses are announced

… but mitigation is hard.

- if two MACs pretend to own the same IP, which one do you trust?

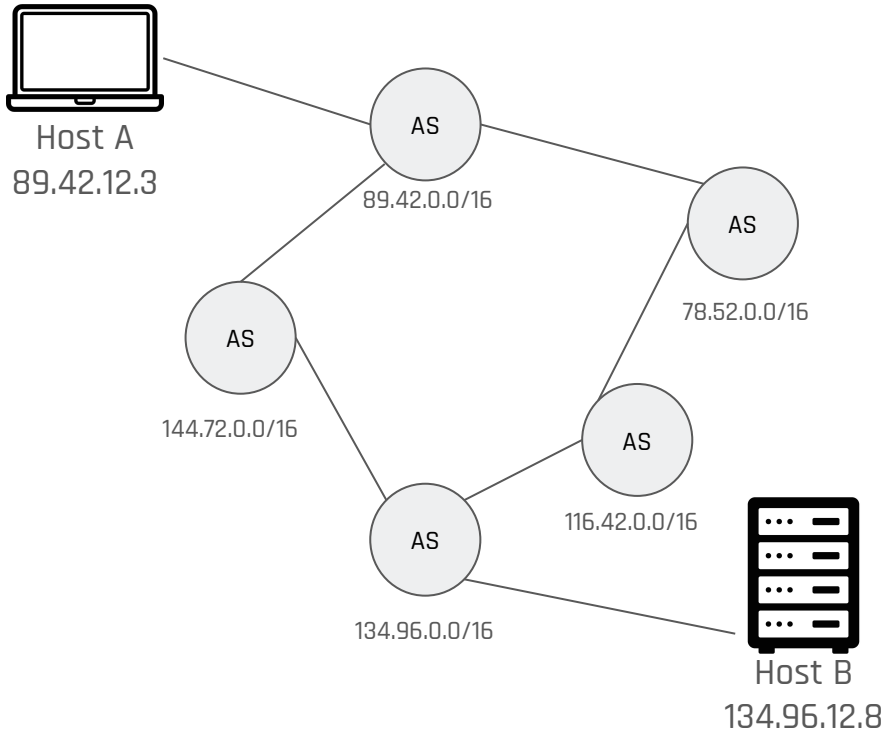
MAC	*IP*
BF:AC:4E:CB:12:96	192.168.0.42
…	…
08:DA:4F:5B:8D:FF	192.168.0.42

ARP Table

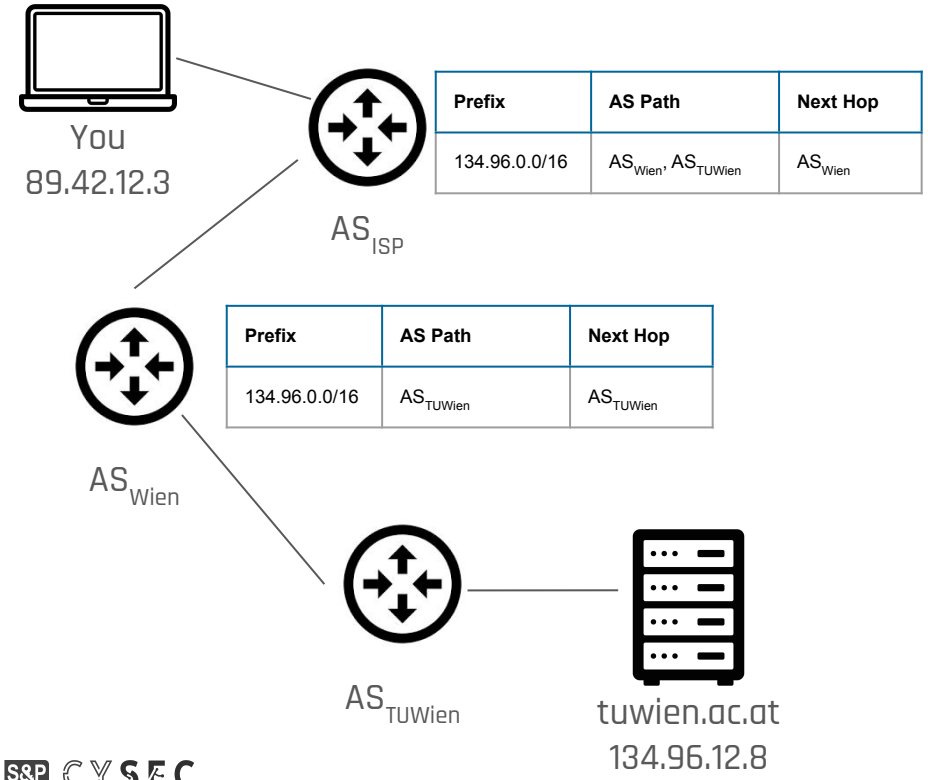
Threats: BGP Issues

Routing

- Usually there is a sequence of Autonomous Systems (AS) to traverse to connect two hosts
- Each AS is responsible for a prefix of an IP range.
- We want to have the shortest possible path to the communication partner.

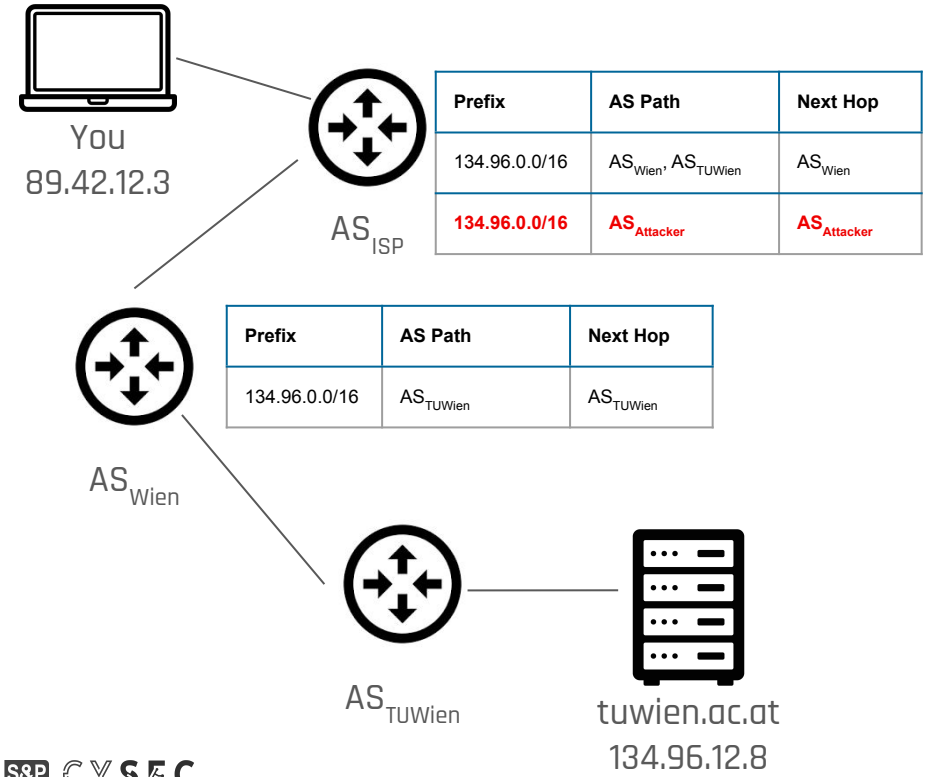


Routing via Border Gateway Protocol (BGP)



- Every AS has an BGP Routing Table
- Scales very well for large number of routers (Internet)
- Internal (iBGP) for routing within AS
- External (eBGP) routes between ASes
- Allows to model politics (we can ignore incoming routes)

BGP Hijacking



BGP always prefers the shortest path and/or more specific network ranges!

BGP Hijacking Example Incidents

December 24, 2004: TTNet in Turkey hijacks the Internet by sending 100k route advertisements

February 2008: Pakistan Telecom want to block Youtube and "accidentally" announced it globally

April 2010: Chinese AS of an ISP hijacks the Internet by stealing ~30k routes

December 2014: Traffic for UK based Atomic Weapons markers rerouted via Kiev

January 2017: Iranian BGP-based censorship for pornography detected

July 2018: Iran Telecommunication Company originated 10 prefixes of Telegram Messenger.

February 2022: Attackers hijacked BGP prefixes of a South Korean cryptocurrency platform, then issued a certificate on the domain to serve a malicious JS, stealing $1.9 million cryptocurrency.

BGP Hijacking Defences

- Route filtering
  - Route advertisements are checked against access control lists
  - Policies that have a specific address space has to be routed via X
- S-BGP (Secure BGP)
  - Add authentication and authorization capabilities
  - Public Key Infrastructure to authorize prefix ownership
  - Prevent route advertisements from modifications
  - Routing advertisements are encrypted
  - Has been around since 2000, rarely used in practice
- Route Origin Authorization (ROA)

BGP with Route Origin Authorization (ROA)

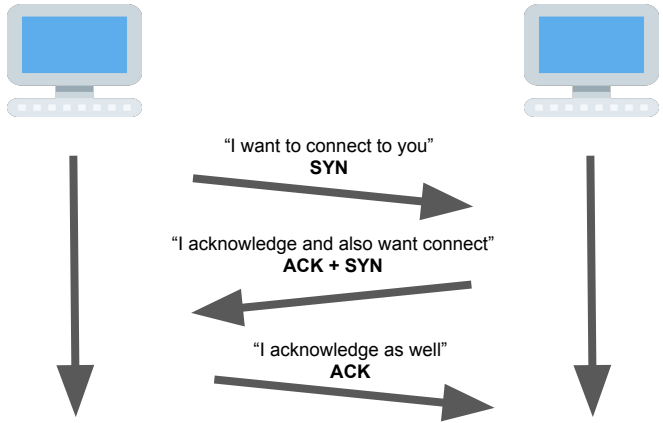
- Idea: use PKI to authorize some AS to announce a prefix
- High-level content of an ROA:
  - AS number: number of the originating AS
  - IP prefix to be announced
  - Single-Bit identifier to allow for more specific prefixes
    - e.g., AS 1 can announce 1.2/16, with more specific prefixes
    - AS 1 can also announce 1.2.3/24 or 1.2.0/17
  - Signature with known key
- Whenever a BGP announcement is received (from origin AS), it is checked against known ROAs for this AS, which protects against fake origin announcements.

Threats: TCP Spoofing & SYN Flooding

Transmission Control Protocol (TCP)

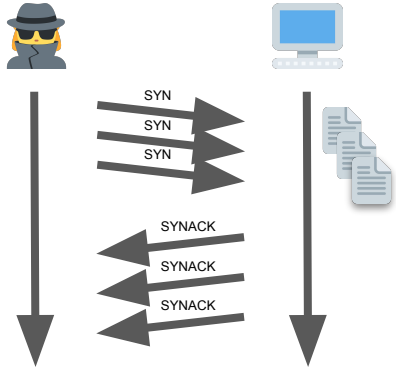
The **Transmission Control Protocol (TCP)** works on the transport layer and is:

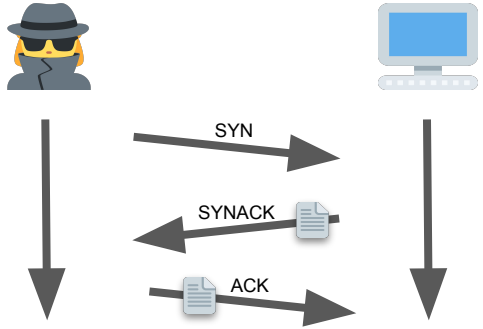
- *Connection-oriented:* Uses a three-way handshake to make sure that the states of the sender and receiver are synchronized.
- *Reliable transmission:* acknowledgement mechanism enforces arrival of data, and lost segments are retransmitted.
- *Point-to-point:* Two applications talking to each other



Attack: SYN Flooding

- Each connection needs resources to be managed. If an attacker sends enough packets, the other end won't be able to manage all connections, missing legitimate ones.





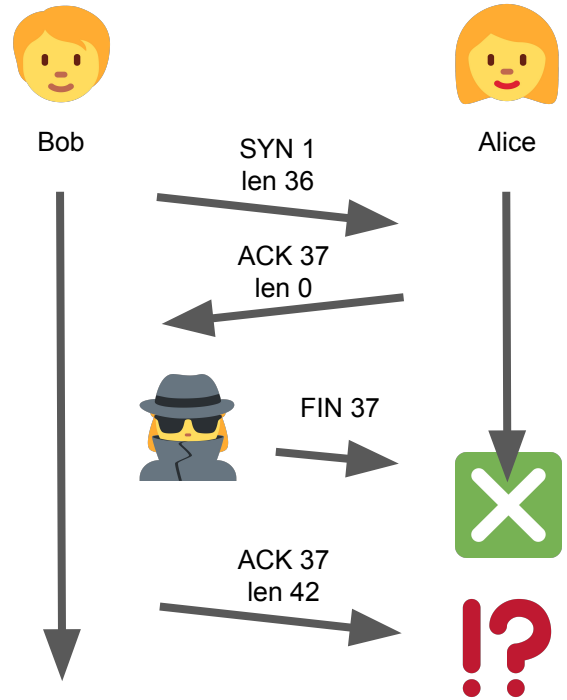
Defense:

SYN Cookies - use TCP sequence number to save state, verify cookie when ACK (with a cookie) is received, then establish the connection. Thus we can not have half open connections.

Attack: TCP Injection / Hijacking

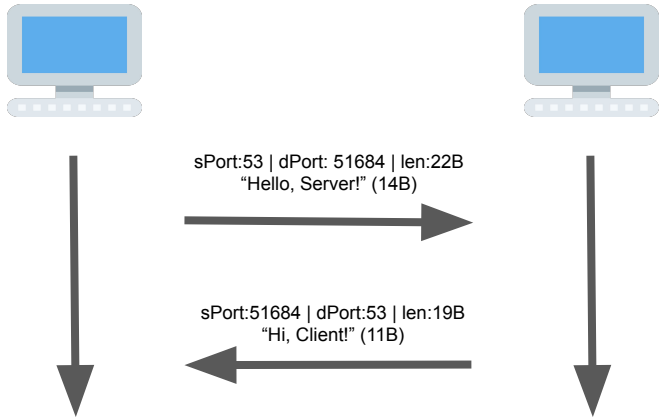
TCP gives assurances on flow and shared state, among others, but only if the rules are being followed. An attacker does not to abide by them and can exploit the assumptions.

- The packets are not authenticated, an attacker can inject packets *if they can predict the sequence number*
- In the simplest case an injected FIN message closes the connection
- Together with ARP spoofing this can lead to a complete TCP connection hijack, re-routing an established connection through the attacker



Excursion: User Datagram Protocol (UDP)

UDP is a minimalistic Transport Layer protocol which primary goal is to be fast. It uses a query-response pattern, where the response just swaps source/destination IPs and ports. Also, it is simple and stateless, such that packages have a low-overhead transmission of data. It has **no** identity validation (IP spoofing possible), and has **no** acknowledgement mechanism and thus unreliable data transmission. Notably, the loss of data should be tolerable (e.g., streaming, VoIP, ...).



Excursion: TCP vs. UDP

TCP (connection-oriented):

- Overhead
  - 3-way handshake
  - 20B header
- reliable transmission due to acknowledgements
- ordering preserved by using sequence numbers
- IP spoofing only works if attacker knows the sequence numbers

Example applications: HTTP, SMTP, IMAP

UDP (connection-less):

- Lightweight
  - query/response (nice for broadcasting)
  - 8B header
- prone to packet loss because arrival is never checked
- no ordering because no sequence numbers
- prone to IP spoofing

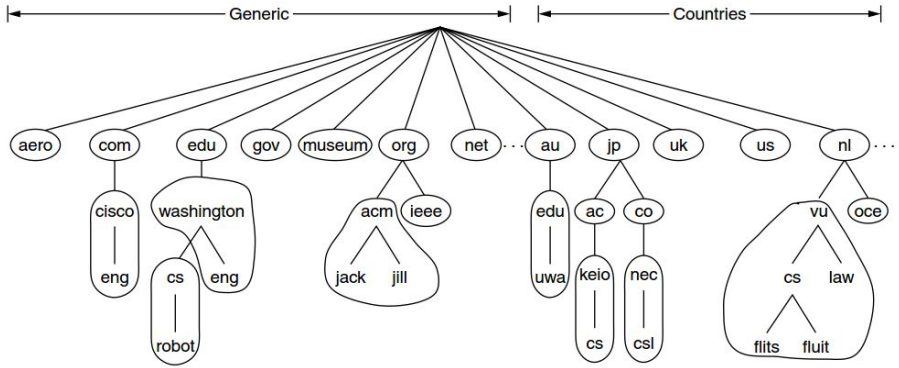
Example applications: VoIP, Gaming, simple network protocols like NTP/DNS

Threats: DNS Attacks

Domain Name System (DNS)

Humans can remember (Domain-)names easier than IP Addresses. The **Domain Name System (DNS)** is resolving domains to IP addresses. Queries work iteratively, asking the root-servers first, then following the delegations until an authoritative answer is obtained, or recursively, asking the DNS servers to forward the query.

DNS responses can contain various records for IPv4 (A), IPv6 (AAAA), mail servers (MX), etc.



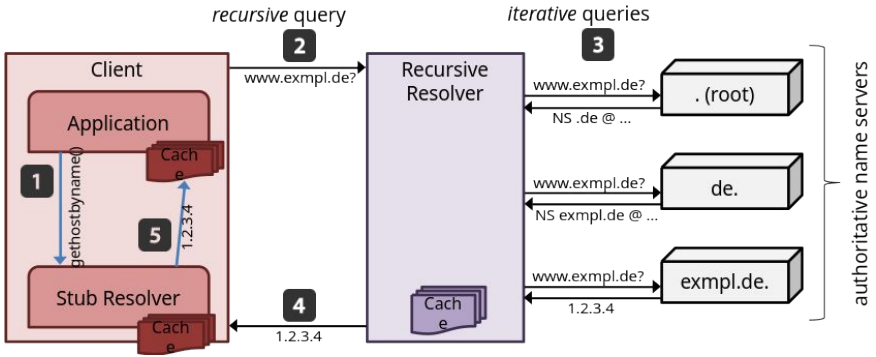
DNS Example

tuwien.ac.at

- **.** : root domain, administrates top-level domains (TLD).
  - Knows delegate NS for .at
- **.at**: TLD, administrates second-level domains (SLD).
  - TLD also referred to as effective TLD (eTLD) e.g., .gov.uk is an eTLD
    - Defined by the Public Suffix List (PSL)
  - Knows delegate NS for ac.at
- **.ac.at**: SLD or eTLD+1, administrates third-level domains
  - Knows who is the NS for tuwien.ac.at
- **tuwien.ac.at** is our fully-qualified domain name

DNS Resolver Types

- Stub Resolver
  - DNS Client
- Recursive Resolvers
  - Serve DNS clients (stub resolvers)
  - Resolve any domain
  - Iteratively query authoritative NSes
  - Restricted to authorized users
- Authoritative Nameserver
  - Serve recursive resolvers
  - Resolve domains in their zone only
  - Deny all recursive domain resolution
  - Cannot restrict user base



DNS Caching

To speed this up, answers and referrals (pointers to other Nameservers) are cached for a certain time, called **Time To Live (TTL)**.

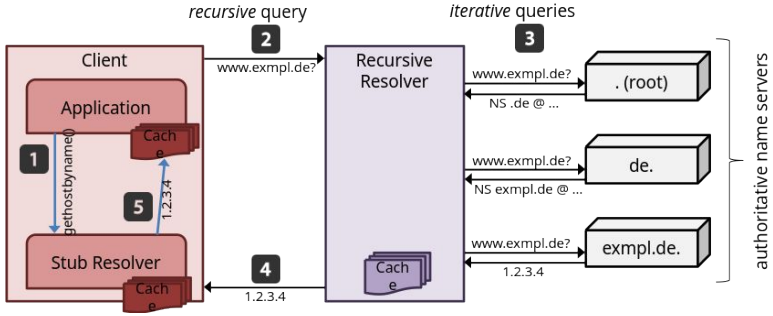
- Queries are answered from cache whenever possible
- Larger TTL decreases load, but incur update latencies

```
> dig secpriv.wien
...
;; Query time: 12 msec
;; SERVER: 128.131.4.3#53(128.131.4.3) (TCP)

> dig secpriv.wien # five seconds later
...
;; Query time: 0 msec
;; SERVER: 128.131.4.3#53(128.131.4.3) (TCP)
```

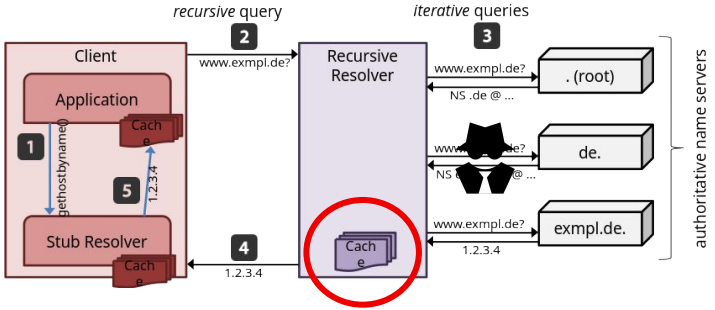
DNS Cache Poisoning

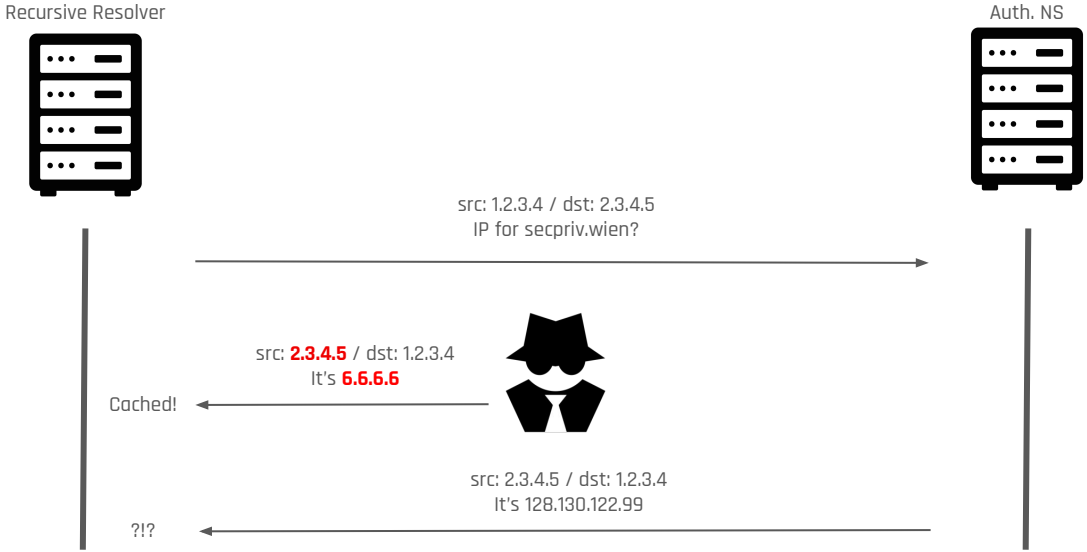
- **Attack goal:** inject attacker-specified Resolvers to cache
  - Once the cache is poisoned, attacker can control the traffic to the domains of all clients using that resolver
- Several "interesting" attacks
  - Redirect clients to malicious (e.g., phishing) web sites
  - Redirect mails to mail server under attacker control
  - Disable domains entirely (censorship!)
  - … and many more!
- So, how (and where) to poison a DNS cache?



DNS Cache Poisoning

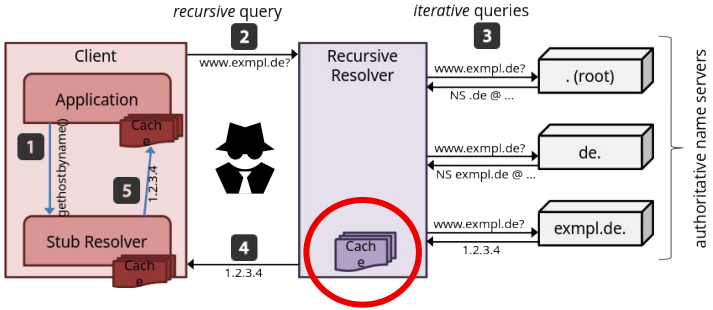
- **IP Spoofing by in-band attacker:**

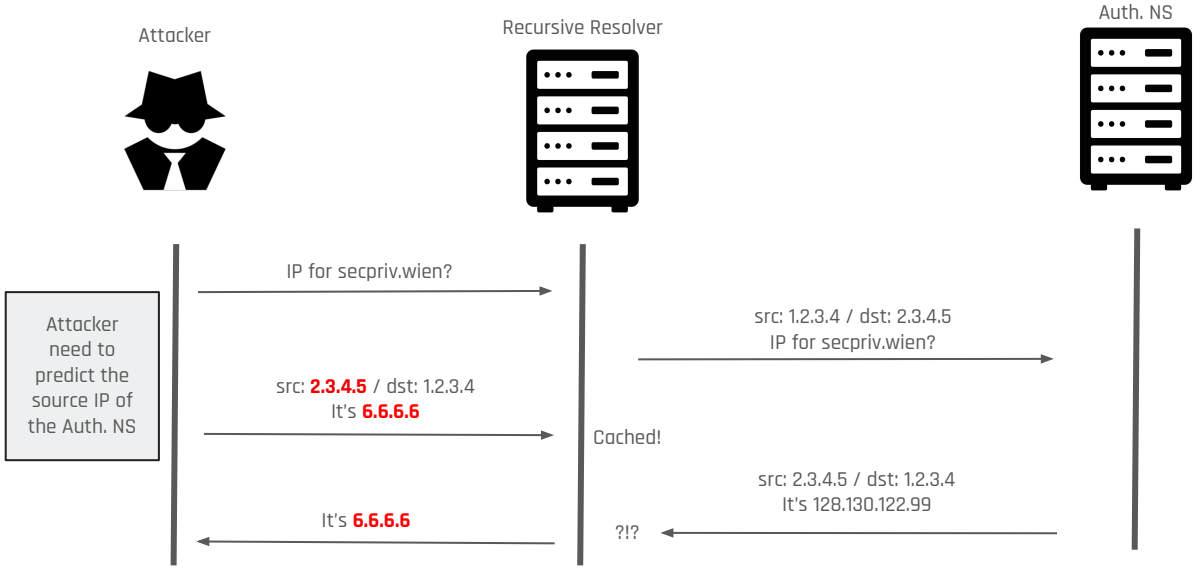




DNS Cache Poisoning

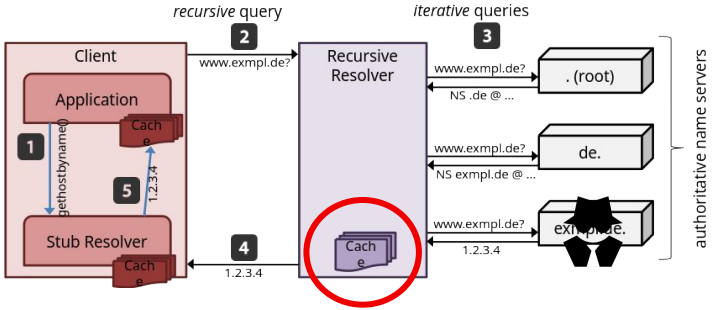
- **DNS Response Spoofing:**

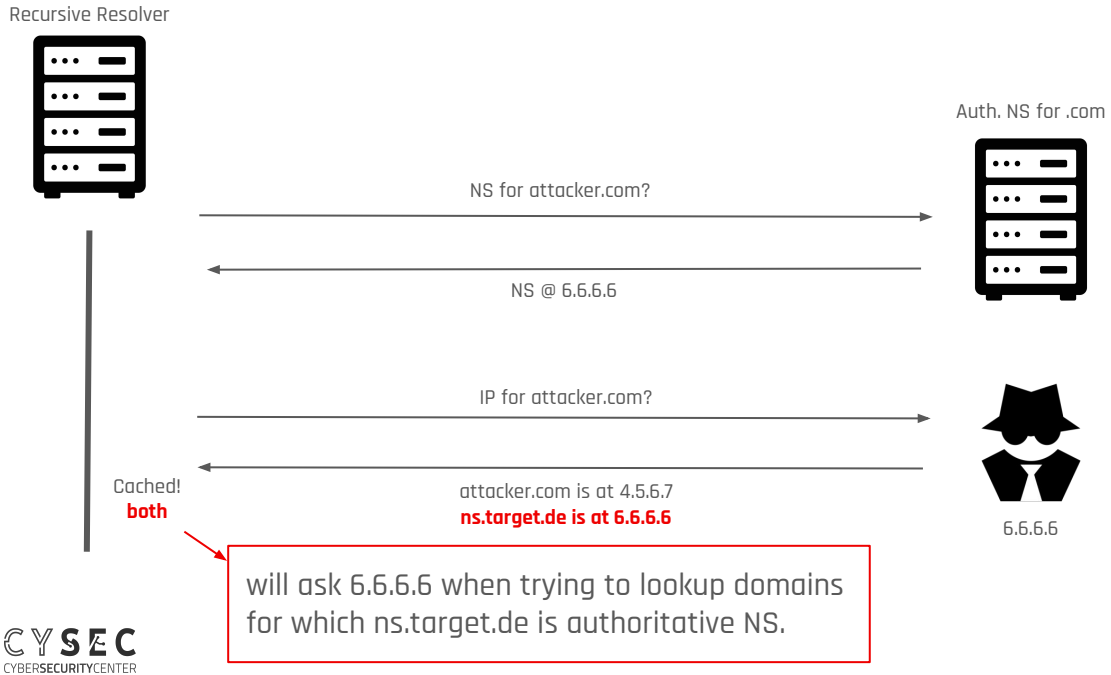




DNS Cache Poisoning

- **Out-of-bailiwick responses:**





Defences for DNS

- DNS Security Extensions (DNSSEC)
  - DNSSEC extends DNS and guarantees authenticity and integrity of the data. A DNS client can thus verify that the DNS zone data received is actually identical to the one authorized by the zone creator via signatures.
  - Confidentiality / Privacy is not provided with DNSSEC as the data is not transferred encrypted.

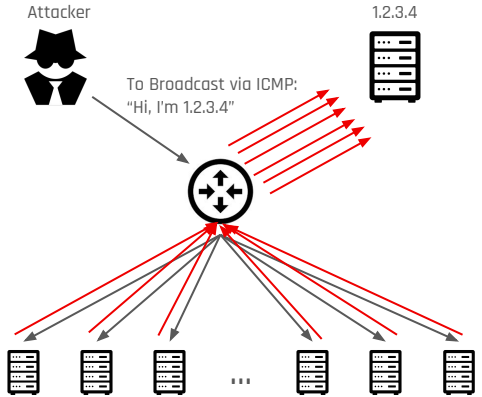
- DNS over TLS (DoT) / DNS over HTTPS (DoH)
  - TCP instead of UDP (additional overhead (also due to encryption))
  - DoT services run on dedicated port, DoH not (allows for deniability)

Threats: Amplification Attacks

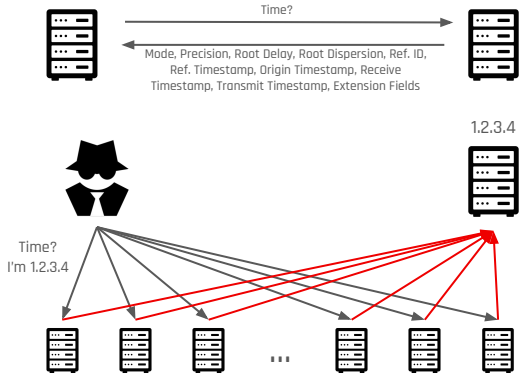
Amplification Attacks

We have seen *SYN Flooding (see TCP)*, but there are more examples:

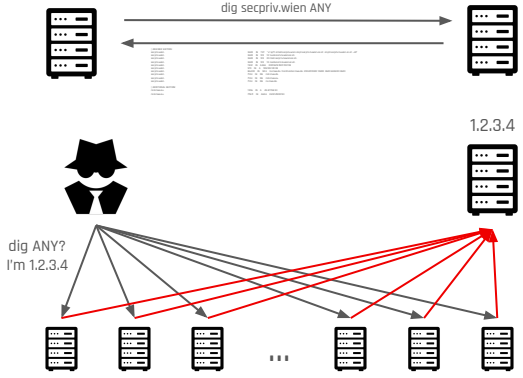
SMURF



NTP Reflection



DNS Amplification



Threats: Port Scanning

Ports

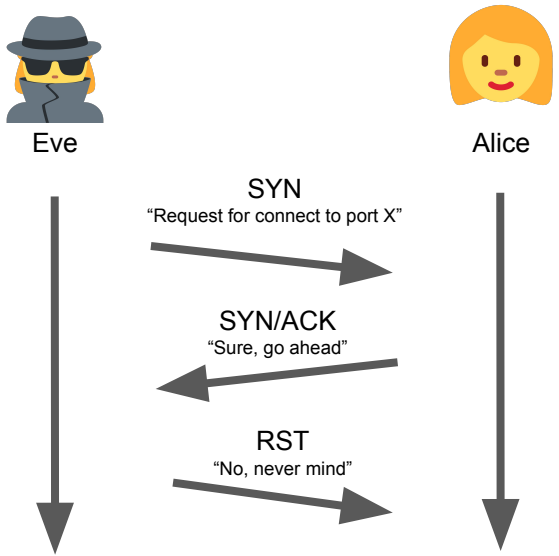
- With IP we have host-to-host communication, in practice we want process-to-process communication
- **Solution:** Ports bound to Processes
  - Packages sent to this port are passed to the process bound to it
  - Ports are included in addressing scheme (e.g., 192.168.0.42**:80**)
- A Port number is a 16-bit unsigned integer (0 - 65535), and has usually a well defined Service associated with it.

Port	Service
20	File Transfer Protocol (FTP) Data Transfer
22	Secure Shell (SSH) Secure Login
25	Simple Mail Transfer Protocol (SMTP)
53	Domain Name System (DNS) service
80	Hypertext Transfer Protocol (HTTP)
123	Network Time Protocol (NTP)
143	Internet Message Access Protocol (IMAP)
443	HTTPS (HTTP over TLS)

Port Scanning

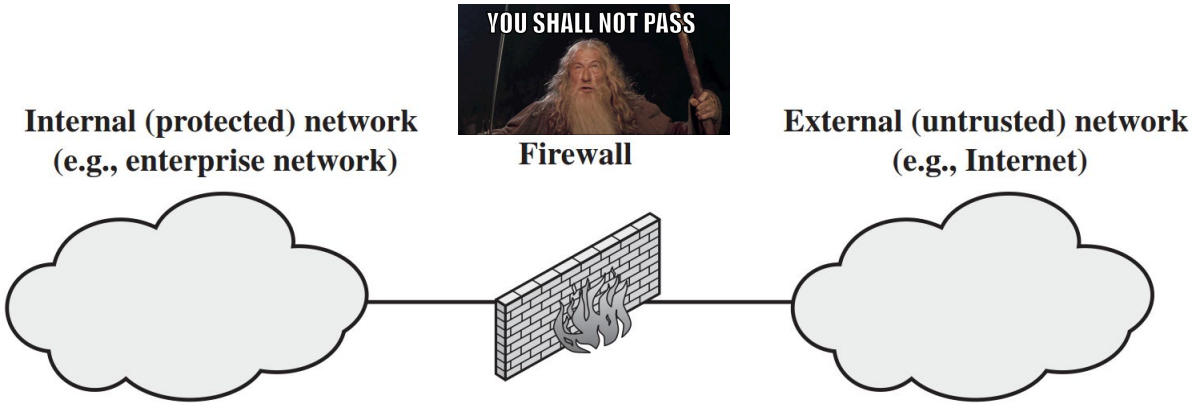
- We want to check which Ports are open at a specific host, thus we try to establish a TCP connection to each Port number and see if we if succeeds:

```
> nmap secpriv.wien
Starting Nmap 7.94 ( https://nmap.org )
...
PORT    STATE SERVICE
22/tcp  open  ssh
25/tcp  open  smtp
80/tcp  open  http
143/tcp open  imap
443/tcp open  https
```



Defence: Firewalls

Firewalls



- Protects an organization (inside) from Internet (outside)
- All traffic between inside and outside need to be checked by the firewall
- Only authorized traffic, as defined by policies, may pass
- Assumption: firewall cannot be compromised

Package Filter Firewalls

- Rules/policies regarding packet headers
  - Source and destination IP
  - IP protocol field (e.g., ICMP/TCP/UDP)
  - Source and destination transport-level address (TCP/UDP ports)
  - Example IPTables Rule: ("Block incoming connections from this IP")

```
iptables -A INPUT -s "$BLOCK_THIS_IP" -j DROP
```

  - *Pro:* Simple and fast
  - Con: Stateless decisions per packet & fail to identify "related" traffic.

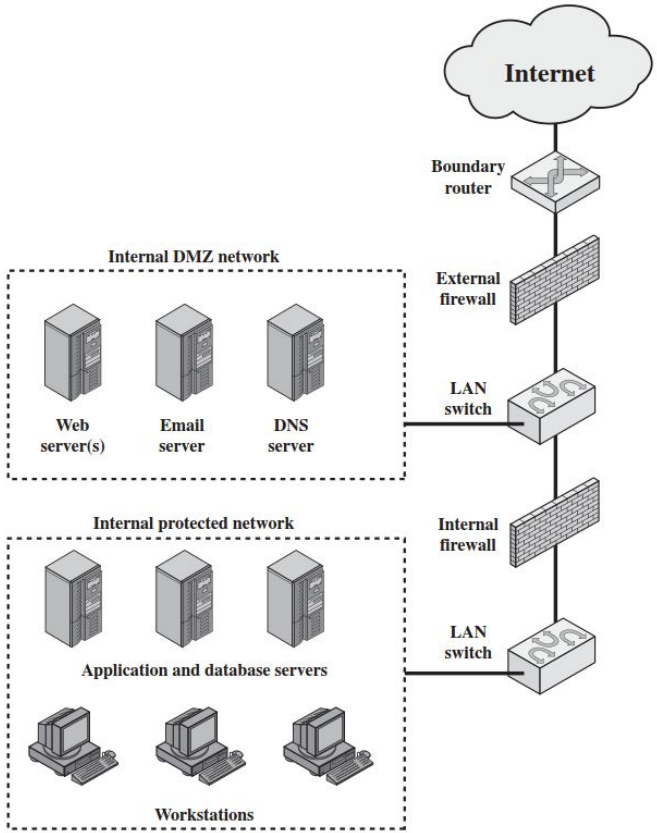
Stateful Inspection Firewalls

- **Solution:** Stateful Inspection Firewalls
  - Maintain directory of connections/streams (connection tracking)
  - Specify rules based on packets and/or connections
  - Three connection states
    - NEW: packets starts new TCP/UDP connection/stream
    - ESTABLISHED: packet is associated with an existing connection
    - RELATED: starts new connection associated with an existing connection.
  - Example IPTables Rule: ("allow all incoming HTTP traffic to port 80")

```
iptables -A INPUT -p tcp --dport 80 -m state --state NEW,ESTABLISHED -j ACCEPT
iptables -A OUTPUT -p tcp --sport 80 -m state --state ESTABLISHED -j ACCEPT
```

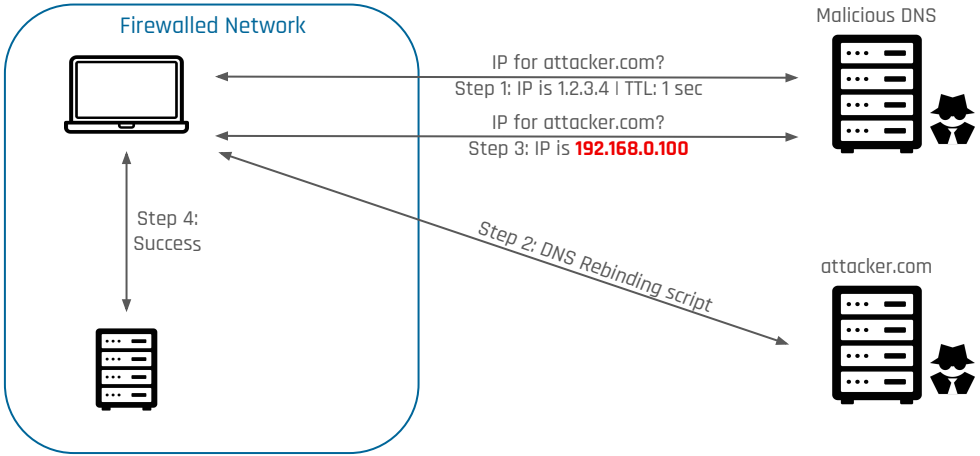
Demilitarized Zones (DMZ)

- Even in internal network, we have different protection levels
  - Web server must be reachable from the outside
  - Clients must not be reachable from the outside
  - Compromised Web server must not lead to access to clients
- Solution: Demilitarized Zones



Bypassing a Firewall

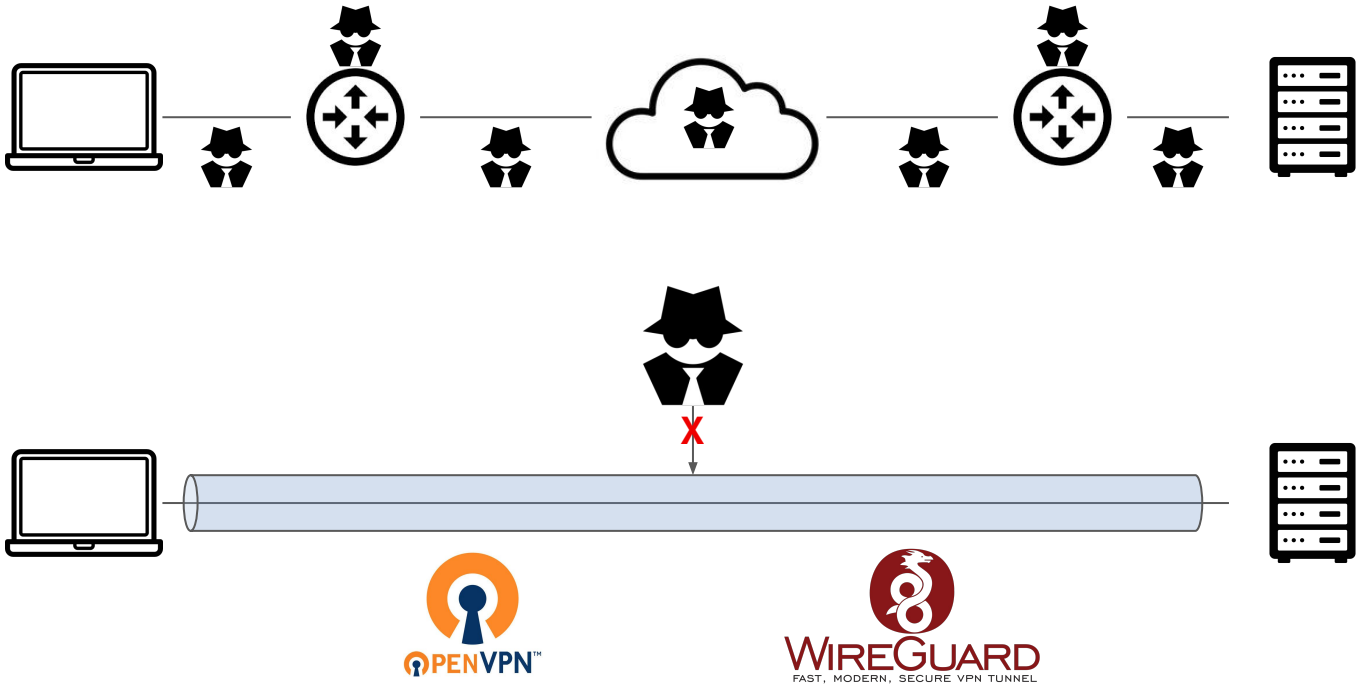
- DNS Rebinding:



- Server-Side-Request-Forgery (SSRF):
  - Details will be explained in the Web Security lecture
- etc.

Defence: Tunneled Networks

Tunneled Networks



Virtual Private Network (VPN)

- A Virtual Private Network (VPN) creates a encrypted tunnel between a device and a network, such that the device can access network resources even via insecure connections (public WiFis, Internet, etc.) though the secure tunnel.
- The VPN security model provides:
  - **Confidentiality:** If the network traffic is sniffed at the packet level, attackers only see encrypted traffic.
  - Sender **Authentication:** Only authorized users can access the VPN.
  - Message **Integrity:** The VPN detects and reject any instances of tampering with transmitted messages.

“Magic Properties” of VPNs

Investigating Influencer VPN Ads on YouTube

Omer Akgul, Richard Roberts, Moses Namara*, Dave Levin, Michelle L. Mazurek
University of Maryland, *Clemson University

Youtube Influencer Advertisements (from Akgul et al., S&P 2022):

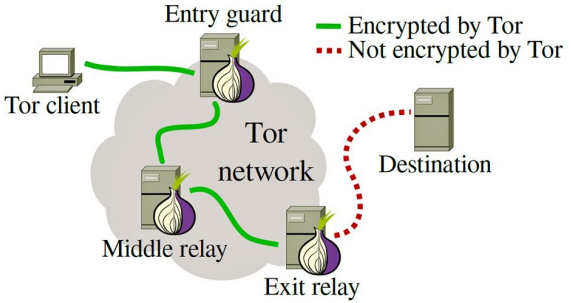
- “I promise if you go ahead and give NordVPN a try, you won’t ever have to worry about anything on the internet again”
- “If you’re not using a VPN when you are surfing the internet, you are playing with fire”
- “keep the ISP from out of your business, keep them from knowing what you’re looking at.”

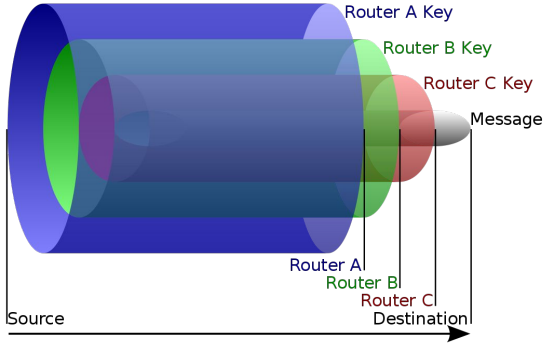
Excursion: Onion Network

The Onion Router Network directs Internet traffic via a worldwide volunteer overlay network for enabling anonymous communication.

To send a package first a random path of at least 3 relais is generated. Than the message is encrypted with those in the sending order. This ways the rails only know who sended them a package and what is the next hop:

- Entry Node knows you and a random node
- Exit node only knows the random node and destination





Combining TOR & VPN?

Investigating Security Folklore: A Case Study on the *Tor over VPN* Phenomenon

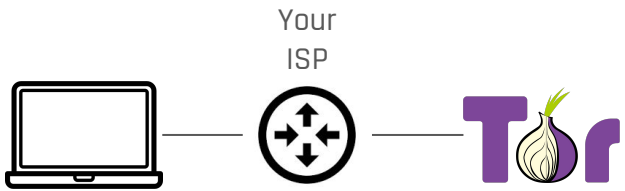
MATTHIAS FASSL, CISPA Helmholtz Center for Information Security, Germany
ALEXANDER PONTICELLO, CISPA Helmholtz Center for Information Security, Germany
ADRIAN DABROWSKI, CISPA Helmholtz Center for Information Security, Germany
KATHARINA KROMBHOLZ, CISPA Helmholtz Center for Information Security, Germany

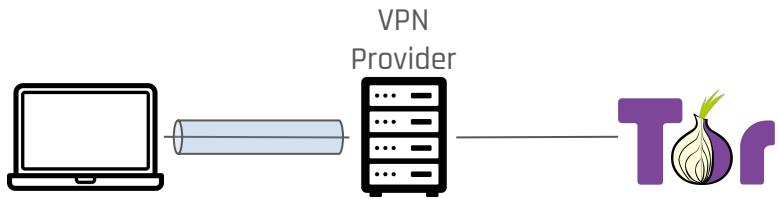
Participant Quote from Fassl et al., CSCW2 2023 [1]:

- "If I'm trying to be private about what I'm doing I might as well set up as many safeguards as I can."

Only meaningful use case: Circumventing censorship when access to the Tor is blocked

... however you could also use Tor bridges.





Thank You!
Q&A

Sebastian **Roth** <sebastian.roth@tuwien.ac.at>