Practical Network Defense

Master's degree in Cybersecurity 2024-25

Forward proxy activity

Angelo Spognardi
spognardi@di.uniroma1.it
Dipartimento di Informatica
Sapienza Università di Roma

Squid activity: as a forward proxy

To do the activities

- We will use Kathará (formerly known as netkit)
  - A container-based framework for experimenting computer networking: http://www.kathara.org/
- A virtual machine is made ready for you
  - https://drive.google.com/file/d/1W6JQzWVyH5_LKLD20R6XH1ugPDP5LWP5/view?usp=sharing
- For not-Cybersecurity students, please have a look at the Network Infrastructure Lab material
  - http://stud.netgroup.uniroma2.it/~marcos/network_infrastructures/current/cyber/
    - Instructions are for netkit, we will use kathara

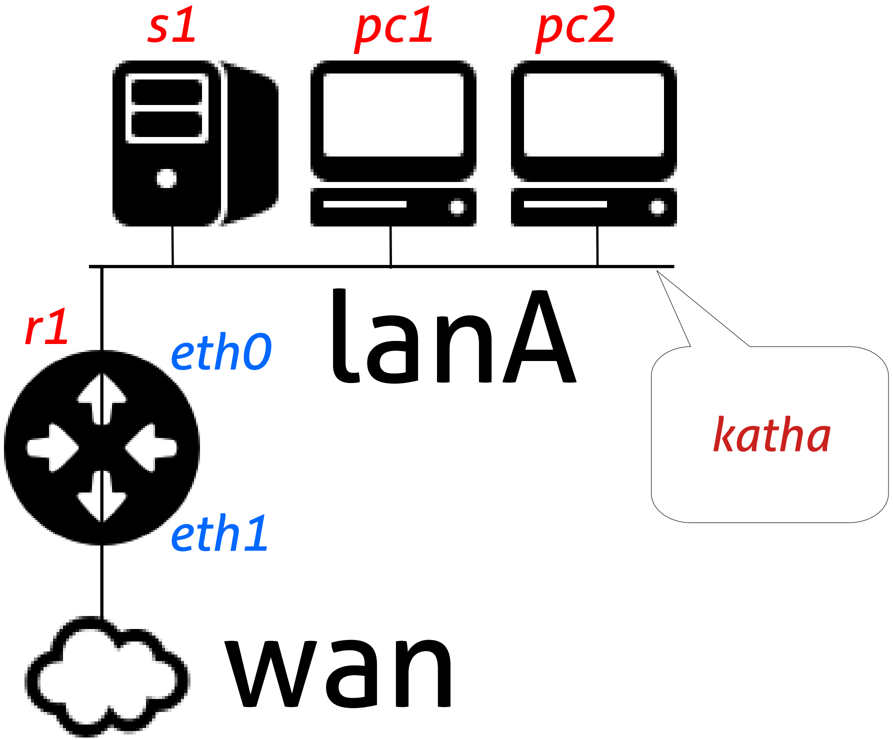
The kathara VM

- It should work in both Virtualbox and VMware
- It should work in Linux, Windows and MacOS
- There are some alias (shortcuts) prepared for you
  - Check with **`alias`**
- All the exercises can be found in the git repository:
  - https://github.com/vitome/pnd-labs.git
- You can move in the directory and run lstart
  - **NOTE**: launch docker first or the first lstart attempt can (…will…) fail

Lab activity: lab6/ex1

pnd-labs/lab6/ex1: squid proxy

- In this lab you have to incrementally build the squid configuration
  - You can join lanA using the `connect_lab` script
- Refer to this resource
  - Most of the activity can be solved looking at the above resource
- Firstly, in r1 enforce the policy that only the proxy can use http and https (and obviously DNS) with `iptables`
  - Verify that pc1 and pc2 cannot use internet
- Take a look at the simple squid configuration file at `/s1/etc/squid/squid.conf`



Activity 1

- Configure pc1 and pc2 to use the squid proxy
  - `export http_proxy=192.168.100.80:3128`
- To start (resp. stop) squid: `systemctl start squid`
- Verify you can connect with http to a website (that uses http!)
  - Ex: http://httpforever.com/, http://neverssl.com, http://http.badssl.com, http://detectportal.firefox.com/success.txt, http://google.com/generate_204
  - Check with wireshark what happens
- Modify the squid.conf so that pc2 can not use http
  - Check with wireshark what happens
- Modify again the squid.conf to use a file with blacklisted websites

Activity 2

- Configure squid so that it can also allow https
- `pc1$ export https_proxy=192.168.100.80:3128`
- To work, this requires the use of the CONNECT method
  - Reference https://wiki.squid-cache.org/SquidFaq/SecurityPitfalls#The_Safe_Ports_and_SSL_Ports_ACL
    - Extra details are provided in the original squid.conf file, found at `s1/etc/squid/squid.conf.bak`
    - Look for "SSL_Ports" and "Safe_ports"
  - When done, check with wireshark what happens

Activity 3

- Configure squid so that it requires the users to authenticate with username and password
- You can find more info about authentication methods on this resource:
  - http://www.squid-cache.org/Doc/man/
- You can use the ncsa method

Activity 4

- Configure squid to perform SSL Bump, in order to intercept the https traffic generated by the client pc1
- Reference:
  - https://wiki.squid-cache.org/Features/HTTPS
  - https://wiki.squid-cache.org/ConfigExamples/Intercept/SslBumpExplicit

Activity 5

- Configure squid and the topology to realize the configuration of a transparent proxy

That’s all for today

- **Questions?**
- See you next lecture!
- References:
  - Ari Luotonen, Kevin Altis, World-Wide Web Proxies, 1994
  - http://httpd.apache.org/docs/current/mod/mod_proxy.html
  - https://en.wikipedia.org/wiki/Proxy_server
  - Classical vs Transparent IP Proxies, RFC 1919
  - SOCKS 5, RFC 1928
  - HTTP 1.1, RFC 7230
  - Policy based routing and Linux advanced routing and traffic control
  - ICAP, RFC 3507
  - https://wiki.squid-cache.org/FrontPage