Network Firewalls Part II

Network Firewall Configuration and Management Using pfSense

Gabe Gibler & Colton Hotchkiss

July 15, 2017
Version 1.5

University of Idaho

CS 439/CS 539: Applied Security Concepts

Summary

This tutorial focuses on pfSense configuration, weaknesses of default configurations and the mitigation of those weaknesses. Effectiveness of firewalls depends upon how well it is managed and not on how perfectly it is deployed. Therefore, in this tutorial, we will demonstrate management (usage and configuration) of the pfSense firewall.

Contents

1 Objectives of this Tutorial 1

2 Required Background 2

3 Hardware and Software Requirements 3

4 Network Layout 5

5 Firewalls Overview 6

6 Defense-in-depth: pfSense packages 7

7 Activity: Log in to pfSense GUI 8

8 Activity: pfSense Setup 9

9 Activity: Firewall Rules 10

10 Challenge 1: Configure a 2nd Internal pfSense 11

11 Challenge 1 (cont.): Verify the Setup 13

12 Challenge 2: Who's Leaking Sensitive Data? 14

13 Challenge 3: Block the Leak! 15

14 Methods of Mitigating the Leak 16

15 Mitigating the Leak: IDS/IPS 17

16 Snort 18

17 Snort: Defining Custom Rules 19

18 Challenge 4: Block the Leak Using Snort! 21

19 Conclusion 22

20 Appendix: Solutions and Change Log 23

1 Objectives of this Tutorial >

1. Review basic pfSense configuration:
   - Setup a firewall to manage a LAN, WAN;
   - Establish ingress and egress filtering rules.
2. Experiment with rules management and network monitoring:
   - Configure a restricted firewall and test its configuration;
   - Find issues with traffic originating from inside the network.

This tutorial is not a complete user's guide to pfSense management. For that you will want to consult the pfSense wiki. We will briefly explain, through walkthroughs, activities, and challenges, the above objectives.

2 Required Background <>

We assume some knowledge in the following areas:

1. Experience using computers and software applications, like web browsers, and virtualization apps;
2. Fundamentals of Internet and networking mechanisms like TCP/IP stack, TCP/UDP protocols and ports, etc.;
3. An introductory knowledge of data privacy, computer/network security, etc.;
4. Basic wireshark knowledge.

It is not the goal of this tutorial to be completely self-contained and self-explanatory. As such, the tutorial assumes certain background skills and knowledge. The following are some areas where we expect the users of this tutorial to have some previous skills/knowledge:

1. Practical experience using computers, and installing and using common software applications (particularly web browsers, and virtualization platforms). The tutorial does not always explain how to navigate within the operating system's graphical user interface (GUI) or how to execute commands from the command line. Some exposure to logic notations and elementary programming skills would be very helpful with writing firewall rules. Similarly, the tutorial does not explain how to browse the Internet, or how to install software applications.
2. Fundamental knowledge of networking mechanisms and computer networks. This tutorial expects a user to understand technical concepts like the ISO OSI model of networks, and common networking terms such as "packets", "ports", "protocols", "accept/drop" in relation to packets, "TCP" and "UDP", etc.
3. A broad understanding of general computer-related issues will help, such as "data privacy", "network access privileges", "application permissions", and different sorts of internet-based attacks.
4. Wireshark is software for monitoring, logging and filtering network traffic. It will be used for monitoring and examining network traffic in detail between the VMs. For an introduction to Wireshark, see the other Applied Security Concepts tutorials from the University of Idaho.

3 Hardware and Software Requirements <>

The tutorial was executed using the following environment:

1. A computer capable of hosting at least 3 virtual machines (VMs);
2. A virtualization software platform, e.g. VMWare or VirtualBox;
3. A) Two pfSense VMs, B) two Ubuntu VMs, C) one Kali linux VM, and D) one VyOS VM([D]).

The activities and challenges of this tutorial occur in multiple VMs. As such, it is imperative that the user of this tutorial has a machine powerful enough to boot at least 3 virtual machines smoothly at a time, since the pfSense and VyOS machines are not resource intensive. For the sake of consistency, specifics for each VM are given below:

A) We are using pfSense 2.3.3 64-bit. You can download the latest version here. The instructions to create this VM can be found here. We installed two packages **Snort** and **SquidGuard**, which can be added from within pfSense in the package manager.

B) We are using Ubuntu 16.04 64-bit. You can download the Ubuntu 16.04 ISO here. The instructions on how to create the VM in VMware can be found here, or for Virtualbox here. Netcat and Wireshark were installed on both machines.

C) We are using Kali Linux version 2016.2 64-bit. You can download the Kali 2016.2 ISO here. The instructions on how to create the VM in VMware can be found here, or for Virtualbox here.

D) We are using VyOS 1.1.7 64-bit. You can download it here in the form of an ISO or an OVA template that can be directly imported into VMware. For VMware installations using OVA see the VyOS wiki, and for Virtualbox see this.

In addition, a set of scripts are necessary to implement a communication function and its corresponding data leak for the challenges. We utilized netcat to make the connections, and cron jobs to schedule the tasks.

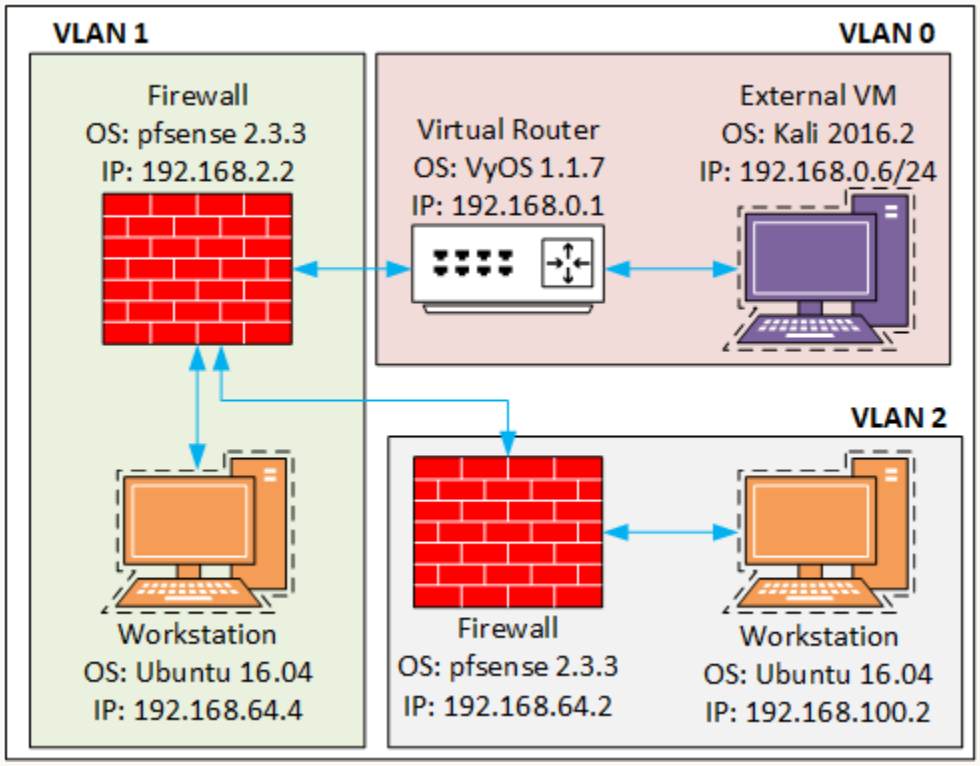
The first script, on Control, sends a small set of "status" signals along with timestamps from Control to Workstation. It sends the signals every 5 seconds. The script also intermittently sends, about every minute, a randomly generated string of the same length but different content from the status signals.

The second script, on Workstation, waits for the connection from Control. It saves the status signals to a log file, */home/seed/Documents/System_Status.log*, and the leaked data to another hidden file.

A third script, on Workstation, periodically transmits the contents of the hidden file to the Kali workstation in VLAN (Virtual Local Area Network) 0. It used any arbitrary port above 1000.

The final script is the script on the Kali workstation that awaits the outgoing connection from Workstation, and saves the received data to a file in Documents that can be monitored to see if communications are successfully exiting the company LAN.

4 Network Layout <>



5 Firewalls Overview <>

1. What is it?
   (a) Hardware and/or software based entity used to protect networks from unauthorized access.
2. What are the main features?
   (a) Stateful Packet Inspection;
   (b) Application Layer Awareness;
   (c) Network Address Translation (NAT);
   (d) Virtual Private Networking (VPN).

1. **What are they?**
   (a) Network firewalls are usually located at the boundary between the internal network and external networks (Perimeter Firewall) or between internal segments of networks (Interior Firewall).
2. **What are their features?**
   (a) Operating in the network layer the firewall examines a packet's header and footer and determines if the packet belongs to a valid session. Using this information the firewall decides if a packet should be forwarded to the internal network or rejected. [3]
   (b) Firewalls that are application aware are able to inspect the contents of packets and make decisions based on their knowledge of the pertinent protocols and programs.
   (c) Network Firewalls are able to change the network address of devices on either side of the firewall to hide the true addresses of devices. This can prevent devices on the outside of a network from being able to probe the true addresses of devices on a network. [2]
   (d) Network firewalls are able to create encrypted connections using VPNs between themselves. If a host on a network needs to communicate with a host on another network the firewall can establish a secure communication channel with the other host through its firewall.

6 Defense-in-depth: pfSense packages <>

A firewall by itself is not enough. There are over 40 packages available for pfSense to enhance its functionality, including:

1. Squid - Caching Proxy;
2. SquidGuard - URL Filter;
3. Darkstat - Network Traffic Monitor;
4. Snort - Intrusion Detection;
5. Country Block .

We will use Snort later in this tutorial.

1. Squid is a caching proxy for the Web that supports HTTP, HTTPS, FTP and others. [1]
2. SquidGuard is a URL redirector based on blacklists.
3. Snort is a rules-based intrusion detection system. It can be used in strictly monitoring mode, in logging mode, or in full intrusion detection mode, to analyze and regulate network traffic. It can do deep-packet inspection, meaning it can analyze the contents of network packets and act upon them using its rules and application awareness.

7 Activity: Log in to pfSense GUI <>

From the Ubuntu Workstation VM:

1. Open up a web browser;
2. Navigate to the pfSense interface, **192.168.64.1**;
3. Log in:
   User: **admin**;
   Password: **pfsense**;
4. What's the first thing that should've happened when setting up a firewall!?

2 Open a browser and navigate to the webConfigurator for pfSense. Its address is the address of the LAN interface.

If necessary open a console directly for the external pfSense, and find the address listed for the LAN interface.

2 `admin/pfsense` is the default password for pfSense routers.

Is it a good idea to change the default password of the built-in admin?

Yes! It is. Change the default password to something of your choosing and record it somewhere safe for the remainder of the tutorial.

8 Activity: pfSense Setup <>

Review configuration of pfSense for our "business" – this specific domain.

1 Review the configuration of its interfaces:

(a) WAN;

(b) LAN.

These are its physical connections to each network segment.

1. If you got here, then this is probably not news to you. But it's good to know where to check in on this information if something isn't working.

9 Activity: Firewall Rules <>

Review configuration of pfSense for our "business".

2 Review the current set of rules for each interface:

(a) WAN (ingress) Firewall Rules;

(b) LAN (egress) Firewall Rules.

These rules apply to either traffic coming into the network (ingress) or leaving the network (egress).

What services will need to be allowed for each interface to be able to do its intended function? What are the common functions of the people using this network? Should everything from the inside be allowed out? Are there downsides to blocking all by default?

10 Challenge 1: Configure a 2nd Internal pfSense <>

You are tasked with setup of a new internal router/firewall for a new fabrication control unit. "pfSense internal" will be used for this, and must be configured from scratch. The control unit has a monitoring protocol that sends on TCP port 5009. The firewall must only allow traffic in or out on this port.

Deliverables:

1. A fully configured instance of pfSense monitoring the connection between the new internal LAN segment and the existing LAN segment;
2. Its WAN must point to the existing internal network. Assign it 192.168.64.2/24;
3. Its LAN must point to a new internal network, for which it will be the gateway. Assign it 192.168.100.1/24;
4. Firewall rules configured to block all traffic except the monitoring transmissions on port 5009 (and the web management ports for pfSense, as well).

Duration: 45 min.

HINTS:

1. Configuration will begin with a direct console connection to the internal pfSense (not the web interface).

2. The format of the IP addresses indicates both the IP address of the interface and its network mask (the number of bits in the mask). For example, for 192.168.64.2/24, 192.168.64.2 is the IP address that needs to be assigned to the interface, 24 is the number of bits in the network mask for that network.

3. If the WAN points to the existing business network, which device must be its gateway – the gateway for the WAN?

4. How about the gateway for the LAN? Why don't you set up a gateway for the LAN interface?

5. This fabrication control unit is very sensitive, so you want to be very restrictive on the range of ports allowed for communication in or out. How do you restrict both incoming and outgoing ports in firewall rules?

6. Don't forget to specify to each interface to drop all other packets by default.

7. If it is not already so, don't forget to point the machines that will be operating on this new network to their gateway/router.

11 Challenge 1 (cont.): Verify the Setup <>

Verify that the firewall is properly configured by checking that you are receiving status updates in the file **System_Status.log**, located in `/home/seed/Documents` on Workstation (not Control).

Please, be careful. The system is finicky. Do not edit or save the file! It messes up the running process.

Deliverables:

1. The file **System_Status.log** should be receiving status updates, and a few select control sequences;
2. No other communications should succeed through the firewall.

Duration: 10 min.

HINTS:

1. If the process stops updating, you may need to restart the Workstation and/or Control machines.

12 Challenge 2: Who's Leaking Sensitive Data? <>

After the new network for the fabrication unit has been running awhile, it becomes apparent to the company that sensitive data is being leaked to competitors. The information has to be coming from the fabrication control unit somehow.

Determine, for your part, whether the data leak is happening through your network systems. If it is, what are its sources and destinations?

Deliverables:

1. Observation of unusual control packets and the route they are taking through your network.

Duration: 10-15 min.

HINTS:

1. What software is very helpful for monitoring traffic on a network?
2. You can look at the Wireshark Wiki for a list of helpful commands.
3. BTW: if you manage to locate actual files performing this data leak (if said files exist), for the sake of this tutorial, you're not allowed to interfere directly with them (should such files exist, which they don't). (So, P.S., don't go looking!)

13 Challenge 3: Block the Leak! <>

Having found the nature of the data leak, block it. Attempt to block it at the source, and if you can't, at least keep it from exiting your business' network.

Deliverables:

1. Firewall rules tightened to restrict all unnecessary communications.

Duration: 10-15 min.

HINTS:

1. What is the difficulty with attempting to block it at the source?
2. For a business with sensitive internal information and competition attempting to aggressively steal trade secrets, what can be improved or is perhaps necessary to tighten up security at the edge (external-facing) firewall?

14 Methods of Mitigating the Leak <>

The leak might be blocked any of several ways:

1. IDS/IPS rules (Snort, Suricata, etc.) on the internal or edge firewall;
2. Strict egress rules on the edge firewall.

Ultimately, the internal firewall can't be made any stricter with rules alone, because it is already blocking all traffic except port 5009 (or should be), and port 5009 is a required service. The exploit is piggy-backing on the open port.

15 Mitigating the Leak: IDS/IPS <>

There are still issues with data leaving the company. In just the same fashion as data left the control network, data can still piggy-back out of your network on necessary or otherwise commonly open ports.

Intrusion Detection Systems (IDS) / Intrusion Prevention Systems (IPS) are the tools better suited to handle this sort of situation. **Snort** is one example of such a tool.

16 Snort <>

1. Runs standalone or as an add-on package within pfSense;
2. Snort can run in either IDS or IPS modes – to simply detect or to act;
3. The Snort community provides a host of pre-defined rules, or, if you register, Snort lets you access other, more frequently monitored rules;
4. Can define custom rules for monitoring or blocking your network traffic.

HINTS:

1. To access Snort, in pfSense navigate to Services >Snort
2. Snort monitors each interface specifically – WAN or LAN – and can be enabled/disabled per interface.
3. Snort keeps rules updated, when connected to the internet.
4. Snort generates alerts when network traffic matches rules, and can block that traffic as well. These alerts can be monitored from the pfSense dashboard, or from Snort's interface.
5. Snort lets you manage lists of blocked and allowed hosts.
6. Snort lets you configure many settings pertaining to how and what it's monitoring on each interface.
7. To switch between operating as an IDS vs IPS for any given interface, Snort can be set to block hosts that generate alerts, or not.
8. Custom rules can be defined at Snort Interfaces >Edit the interface mapping for either WAN or LAN >WAN/LAN Rules

17 Snort: Defining Custom Rules <>

Snort allows you to specify protocols, source and destination IP addresses and ports, what contents in a packet to match, what action to take when matched, and to categorize a rule when matches show up in logs or alerts.

HINTS:

1. Navigate to Services >Snort >Snort Interfaces >Edit the interface mapping for WAN WAN Rules
2. Enter the following into the textbox:
   alert icmp any any ->$HOME_NET any (msg: "ICMP detected!"; sid:1000001; rev:1; classtype:icmp-event;)
3. Specifies *alert* rule action. Rule actions include: alert, log, pass, drop, reject, etc.
4. Specifies *icmp* for the protocol to match. Protocols include: icmp, tcp, udp, etc.
5. Then source IP address and port, a direction, and destination IP and port. Note the use of a predefined variable to indicate ranges of ports.
6. The parentheses enclose a variety of other parameters.

   - *msg* supplies the message shown in the alert
   - *sid* and *rev* provide unique identifiers and revision numbers for each rule (sid must be greater than 1 million; lower numbers are reserved)
   - *classtype* indicates a pre-defined set of categories to apply to matches

- *content* and *pcre* specify what to match in a packet's content. *content* looks for string matches anywhere in the packet. *pcre* defines regular expressions to apply against packet contents.
- **!** applies the logical *not* operation to *content* and *pcre* matches.
- There are other quantifiers that can be applied with *content* to specify where in a packet to look or how much of a packet to examine.

7. # comments out rules.

8. For more details:
   Snort Rules Workshop, InfoSec
   Snort Manual

9. Save the rule. Snort will validate the rule, and accept it if successful.

10. Open the *Alerts* tab under Snort, and see if your rule is matching any packets. You can switch the interface to see alerts for each.

11. As another example:
    alert tcp 192.168.64.2 5009 ->any any (content:"Time:"; msg:"Status updated!"; sid:5009001; rev:1; classtype:misc-activity;)

18 Challenge 4: Block the Leak Using Snort! <>

Now we have a tool to block the leak at the source. See if you can write a rule in Snort to detect the leak as it passes through the internal firewall.

Duration: 30 min.

HINTS:

1. Might *pcre* be an ideal choice to match the contents of our particular leak?
2. There are issues with Snort if you don't specify *classtype*.

19 Conclusion <>

1. Firewalls are fun! Firewalls are easy to install and access, but can be relatively difficult to configure for optimal functionality;
2. Through these walkthroughs and hands-on activities, we've learned some basic pfSense configuration;
3. Firewalls can't do everything, when it comes to network security. Intrusion detection systems (like Snort), network monitors (like Wireshark), and other tools are necessary for defense-in-depth!

20 Appendix: Solutions and Change Log <

1. Solutions to the challenges:
   (a) Challenge 1;
   (b) Challenge 2;
   (c) Challenge 3;
   (d) Challenge 4.
2. Change Log.

Solutions to the Challenges:

Challenge 1: By and large, we are iterating through the steps performed on "pfSense External" in the first part of the tutorial.

1. Begin by accessing "pfSense Internal" through a direct connection. In this situation, that means opening a console to the VM itself.
2. Configure the interfaces using menu item **1**. Assign the WAN to the lower-numbered interface. Assign the LAN to the higher-numbered interface.
3. Set IP addresses and interface options using menu item **2**. Don't configure the WAN by DHCP. Set the WAN's IP address: 192.168.64.2; network mask: 24; gateway: 192.168.164.1 (the address for the LAN interface on "pfSense External", because it is the gateway for the main company LAN). Skip options for IPv6. Set the LAN's IP address: 192.168.100.1; network mask: 24; turn off DHCP on this interface; keep the management access default of HTTPS.
4. From the "Control Workstation" VM (located on the newly delineated LAN), open a web browser. (Make sure the workstation is configured with an IP address on the 192.168.100.0/24 subnet and 192.168.100.1 as its gateway.)
5. Navigate to the LAN address for "pfSense Internal": https://192.168.100.1. Ignore any warnings about the security certificate and accept it.

6. Log into pfSense using the default credentials:
   username: admin
   password: pfsense

7. Change the default administrator password and record the new one in a safe place.

8. For the ingress (incoming, WAN) ruleset, all traffic should be blocked by default. It never hurts to be explicit and make the intent clear. Write rules for both IPv4 and IPv6 dropping packets for all protocols from any address to any address. Save and apply the changes.

9. For the egress (outgoing, LAN) ruleset, first create the rule that will explicitly allow only port 5009, both source and destination, from LAN to WAN. Then, ensure the default management rule from pfSense is in place. Finally, disable the default allow all for both IPv4 and IPv6. Save and apply the changes.

Challenge 2: You are the network or IT administrator whose responsibility is the network. So, once the mandate is passed down to find any evidence of the leaking data, you do what you can to monitor your domain and ensure it is airtight.

1. Use Wireshark to monitor the network from "Workstation". Select the *Any* option for capturing packets.

2. You might begin by checking that your firewall is properly filtering out all but port 5009. Filter the captured packets in Wireshark with the following command:

   ```
   tcp.port == 5009
   ```

   After observing for a while, you might refine your filter as follows, to cut out all the TCP synchronization packets and more clearly see the transmission contents:

   ```
   tcp.port == 5009 && tcp.flags.push == 1
   ```

   And then, even more helpfully:
   ```
   ip.src == 192.168.64.2 && tcp.flags.push == 1
   ```

3. After watching long enough, you should see amongst the regular control packets that you do expect, packets that don't match the pattern. They come less frequently and they're not in clear text, but they're there.

4. The last refinement to Wireshark's filter, to broaden your investigation to determine the full scope of what's happening, might be the following:

   ```
   (ip.src == 192.168.64.4 || ip.src == 192.168.64.2)
   && tcp.flags.push == 1
   ```

Challenge 3: There is no way to block the leak on "pfSense Internal" using firewall rules alone, because the leaked data is piggy-backing on the open port. pfSense is already doing all it can. Thus, to at least avoid sensitive data leaving the company, set strict egress rules on the edge firewall ("pfSense External"). (Make sure not to lock yourself out! But pfSense should help you with that.)

1. Navigate to the web management interface for "pfSense External".
2. Ensure that the WAN has a block all rule as its default action (last rule on the list; rules are checked from top to bottom). This should be the case as general best practice anyway.
3. On the LAN interface, create a rule to **block** traffic for protocol: **any**, from source: **any**, to destination: **any**. After saving, make sure it is the last rule on the LAN list.
4. Disable the default allow rules for IPv4 and IPv6.
5. Make sure the rules are active that allow traffic for ports 80 and 443, at least; and that the Anti-Lockout Rule is still in place.
6. Click apply changes.
7. To verify that this works, open a console to the Kali VM.
8. Open a command prompt, and monitor the received files to see if changes are happening. Execute the following command a few times over the course of a few minutes to see if the file is being added to:

   ```
   tail Downloads/received.txt
   ```

   Updates to that file should be blocked. Traffic heading out of the network should be a lot more tightly controlled, now that ports must be explicitly allowed.

Challenge 4:

1. To detect the target packets, we arrived at the following rule:
   ```
   alert tcp any any ->192.168.64.0/24 any
   (pcre:!"/((Time)|(F000)|(F888)|(FFFF))\w{28}/i"; msg:"Not allowed!";
   sid:1234567; rev:1; classtype:policy-violation;)
   ```
2. To drop the packets, switch **alert** to **drop**.

Change Log:

Ver.	Date	Authors	Changes
v1.0	Apr 18 2017	Gabe Gibler & Colton Hotchkiss	First draft of tutorial.
v1.1	Apr 21 2017	Gabe Gibler & Colton Hotchkiss	Added in material for day 2: about Snort, custom rules in Snort, and utilizing them to perform packet inspections and to attempt to catch reverse shells; capture-the-flag exercise.
v1.2	Apr 24 2017	Gabe Gibler	Removed extraneous slide. Removed capture-the-flag exercise. Added in answers to the challenges, and revised steps in the presentation to constrain solutions to the appendix.
v1.3	May 10 2017	Gabe Gibler & Colton Hotchkiss	Changed to CC-ByNCSA license. Made introductory sections consistent among tutorial editions. Removed challenge 5.
v1.4	May 12 2017	Gabe Gibler & Colton Hotchkiss	Making body styles consistent among tutorials.
v1.5	July 15th 2017	Ananth Jillepalli	Standardization (edits, consistency, TeX markup cleaning, and more).

References

[1] Alex Dawson and Adrian Chadd, "Squid: Optimising web delivery," 2017. [Online]. Available: http://www.squid-cache.org/

[2] University of Houston, "Firewalls, intrusion prevention and vpn," 2016. [Online]. Available: http://prtl.uhcl.edu/info-security/indepth/firewalls

[3] Zen Internet, "Stateful vs deep packet inspection," 2017. [Online]. Available: https://www.zen.co.uk/business/broadband/business-broadband/stateful-vs-deep-packet-inspection.aspx