Firewall Management

Basic Usage and Configuration of Firewalls in Windows and Ubuntu

Ananth Jillepalli & Risab Manandhar (Versions 1 and 2)
Gabe Gibler & Colton Hotchkiss (Version 3)

July 29, 2017
Version 3.4

University of Idaho

CS 439/CS 539: Applied Security Concepts

Summary

A Firewall is a technological barrier which can be used in most computing devices to control the incoming and outgoing network traffic. The controlling is done primarily through use of rules, either pre-configured or user-specified. 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 firewalls on Windows and Ubuntu operating systems.

This work is licensed under a Creative Commons Attribution 4.0 International License.

Contents

1 Objectives of this Tutorial >

1. Understand a bit of the history and capabilities of firewalls;
2. Understand how to manage firewalls through graphical user interfaces (GUI) on Windows and Ubuntu;
3. Understand the basics of rule-writing to manage firewalls through command line interfaces (CLI);
4. Apply that knowledge by managing firewalls through both GUI and CLI in a series of activities and challenges.

This tutorial is not a complete user's guide to firewall management. It is a brief overview of a few firewalls in particular. We will briefly explain, through walkthroughs, activities, and challenges, the following specifics about firewalls:

1. Firewalls, as we know them today, are a product of a number of transformations from one generation to the next. To truly understand the current state of firewalls, it helps to understand a bit of their history and the course of their development. The capabilities of firewalls are not restricted to the aspects discussed in this tutorial. Most modern firewalls are complex suites of functionality.
2. This tutorial aims to provide a basic understanding of the elements available in a firewall's graphical user interface and how those apply to management of the firewall's functionality.
3. This tutorial also aims to instill a knowledge of the basic usage of the command line interface for writing rules that define network access and other advanced settings within the firewall.
4. Finally, the skills presented in this tutorial are put to the test by undertaking a set of challenges to provide direct experience of basic firewall management.

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.

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. If setting up the tutorial, additional knowledge is required to set up a domain, set up DNS, etc.
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.

3 Hardware and Software Requirements <>

The tutorial was executed using the following environment:

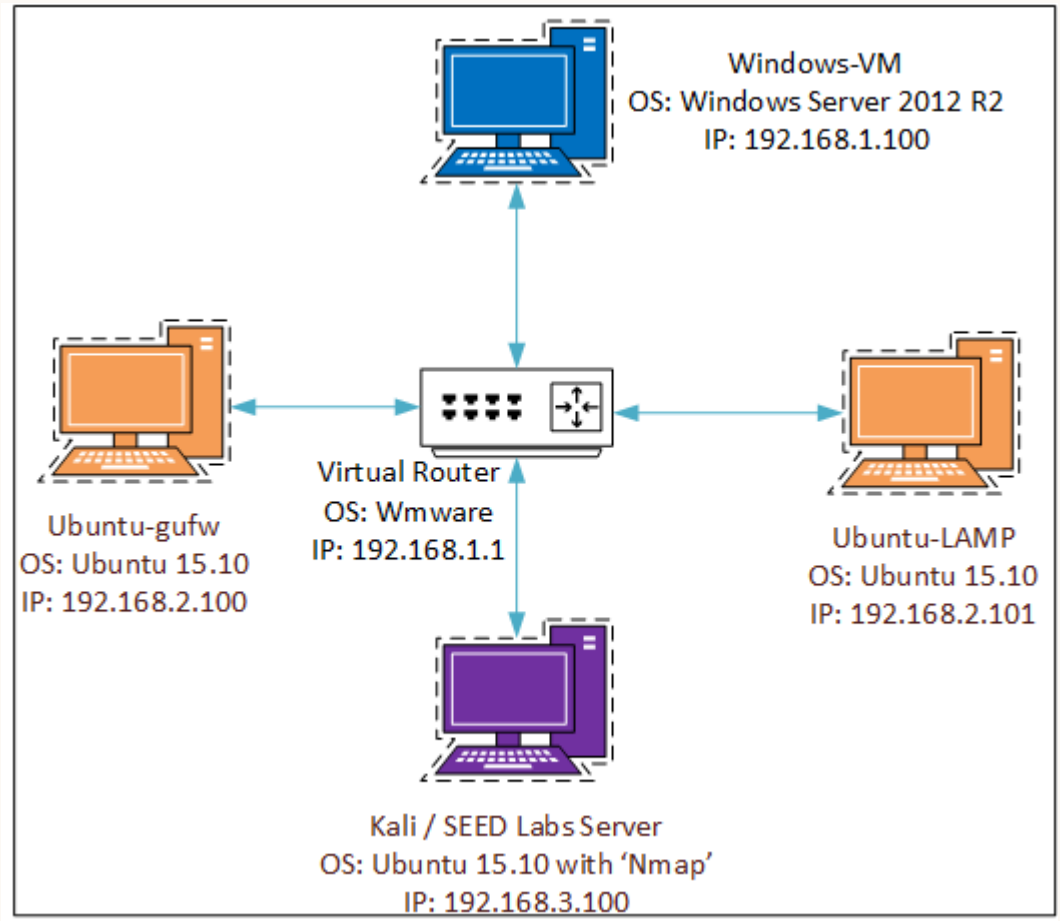
1. A computer capable of hosting at least 4 virtual machines (VMs);
2. A virtualization software platform, e.g. VMWare or VirtualBox;
3. A) One standard Microsoft Windows Server 2012 R2 VM, and B) three Ubuntu 15.10 LTS VMs.

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 4 virtual machines smoothly at a time. For the sake of consistency, specifics for each VM are given below:

A) We are using a standard Microsoft **Windows** Server 2012 R2 VM. You can download it here (free 180-day evaluation copy). However, for the Windows Firewall portion of this tutorial, any Windows OS from Windows 7/Windows Server 2008 onwards can be substituted and will be functionally equivalent.

B) We are using two Ubuntu 15.10 LTS VMs. One should have the package `gufw` installed. It can be downloaded from the Gufw Project. We call this VM **Ubuntu-gufw**. The other two have the `LAMP` stack installed. For a guide to installing LAMP on Ubuntu, read this DigitalOcean tutorial. This VM will host a single website with multiple references to it in the Hosts files of the Windows and "Ubuntu-gufw" VMs. We call it **Ubuntu-LAMP**. Finally, we also use another VM, which will host any number of websites referenced by the Hosts files of the Windows and "Ubuntu-gufw" VMs. In addition, it has Nmap installed. We call it either **Kali** or **SEED Labs server**, depending which role it serves in a challenge.

4 Network Layout <>



5 A Problem: Network Traffic Control <>

1. How can system administrators control network traffic and/or network communications originating from or destined to a network address?
2. It requires more than simply monitoring traffic or logging transactions.
3. The process can be complicated at many stages by business and user requirements. A suitable balance between protection, permissions, and allowances is required.

1. Controlling network traffic has been an issue since the early days of the internet in the 1980s. Early network traffic control was accomplished by direct router configuration and did not involve any additional software modules or embedded software programs.
2. Most confuse traffic controlling with logging the communications or monitoring them because of the overlap of some functionality between the three: controlling, logging, and monitoring. While the latter two might be a subset of the former, depending upon the context of the operation, controlling need not always involve logging and monitoring.
3. Network traffic control is an area of conflicting demands, because of the need for users to utilize software and access network and internet resources efficiently and for system administrators to protect important or sensitive resources. A suitable balance between protection, permissions, and allowances must be attained. If protection and permissions are too restrictive, per the concept of least privilege, users become unable to use business software and network resources efficiently. When permissions are broad and protection lax, a greater range of vulnerabilities are made available.

6 The Solution: Firewalls <>

1. Control of network traffic to and from a computing device;
2. They are the first line of network protection;
3. Different from Intrusion Prevention/Detection Systems(IPS/IDS);
4. Control program access to a network;
5. Networks have incoming and outgoing communications. There are often separate rules for both incoming and outgoing traffic.

1. The primary and most significant aspect of Firewalls is to control network traffic to and from a computing device, and the communication media going in and out of a system. Any network security system satisfying the above responsibility can be called a firewall.
2. Firewalls are usually the first line of network protection for any given network. Some firewalls, mostly third generation application layer operational level firewall systems, have the ability to functionally perform deep packet inspection. This feature is similar to Intrusion Prevention/Detection, and User Identity Integration. However, the features integrated in the firewalls are not as advanced and should not be used in place of Intrusion Detection/Prevention Systems.
3. Modern firewalls also have the ability to configure individual programs or services to be allowed or disallowed, to receive incoming traffic, or send outgoing data. Such a functionality allows for great levels of flexibility in configuring the state of a computing system. Controlling activities naturally involve monitoring and logging of reports, but security systems that only log an monitor are usually not considered firewalls.
4. In most firewalls, outgoing communications are allowed by default and incoming traffic is usually filtered. There's a flaw in this approach. If a system is compromised and it can communicate with server, information theft can occur unnoticed and unhampered. Also, if a system is infected and it can communicate with every other system, infectious files can easily propagate to other systems. Thus, outgoing communications should also be filtered at all times.

7 Classification of Firewalls <>

Types of Firewalls:

1. *Packet-Filtering* (Network Layer);
2. *Stateful Inspection* (Transport/Session Layer);
3. *Application Layer* (Application Layer).

Each item describes a type of functionality, but a firewall may implement multiple types of functionality.

1. **First Generation:** The first published research on "firewalls" appeared around 1988, when Digital Equipment Corporation (DEC) developed filter systems known as packet-filtering firewalls. When first introduced, firewalls operated at the network layer in the form of packet filters targeting network addresses and ports of the packet. From that information, the firewall module determined if the packet should be allowed or blocked.
2. **Second Generation:** Dave Presotto, Janardan Sharma, and Kshitij Nigam of AT&T Bell Laboratories developed the second generation of firewalls and christened them "circuit-level gateways". In addition to first generation capabilities, second generation firewalls were able to operate at the transport and session layer of the OSI model. Inspection of the "state" of packets is the distinguishing feature in this generation. It determines whether a packet belongs to a new connection, an existing connection, or a connection that does not exist at all.
3. **Third Generation:** An application suite named *FireWall ToolKit* (FWTK), developed by Wei Xu, Peter Churchyard, and Marcus Ranum, laid the foundations for the next generation of firewalls. The first application layer firewall was an extension of FWTK, enhanced by Wei Xu. The most significant advantage of application layer firewalls is the fact that various protocols are now comprehensible by firewalls and can be included in rule-making. Such protocols include: File Transfer Protocol (FTP), Domain Name System (DNS), and HyperText Transfer Protocol (HTTP).
4. Current day firewalls are generally hybrid in nature, because most firewalls have capabilities from packet filtering, stateful inspection, and the application layer.

8 Locality of Firewalls <>

1. Firewalls can be situated at different positions within a network and relative to a user:
   (a) *Endpoint* firewalls, more commonly called *software* firewalls;
   (b) *Gateway* firewalls, more commonly called *"hardware"* firewalls;
2. Though location of operation differs, the core functionality is largely all in software;
3. This tutorial utilizes software-based firewalls.

1. Firewalls can sit at the point-of-entry for a network, filtering traffic to and from everything on that network, or they can sit at each specific device's connection to a network, filtering traffic for that device alone:
   (a) **Software (or Endpoint) firewalls:** This type of firewall operates as software installed on the system it is protecting, and its control extends strictly to the networking interfaces of that system. Software-based firewalls depend heavily on OS support to insert themselves into the network stack. Popular examples of this kind of firewall are: Windows Firewall, Comodo Internet Security, ZoneAlarm, Norton 360, and PeerBlock.
   (b) **Hardware (or Gateway) firewalls:** These types of firewalls generally operate as firmware on devices dedicated to controlling traffic at entry/exit points for entire local networks. Hardware firewalls can not exercise much control at the application or session layer for any given system on the network. The firewall often co-exists with other network control functionality on its device. Popular examples of this kind of firewall are: Cisco's ASA firewall, Juniper Network's ScreenOS, Dell's SonicWall, and Untangle's Zeroshell.
2. Though the location of firewalls differ, they largely all perform the core of their functionality (controlling traffic) in software, rather than as actual circuits in hardware. For this reason, it is actually more appropriate to call "hardware" firewalls "firmware" firewalls. Most firewalls of the firmware sort are actually implementations running in a Linux OS on the dedicated "firewall" device.

9 Firewall Limitations <>

1. Micromanagement is required for efficient functionality;
2. Overprotective settings can cripple some applications. If configured to trigger alerts frequently, they can desensitize users to warnings;
3. Firewalls can be compromised or shut down by other compromised programs;
4. Firewalls are not a one-stop solution against network attacks. They are not "defense-in-depth" in & of themselves.

1. The primary limitation of firewalls is that, for efficient deployment, implementation, and subsequent maintenance, a higher degree of micro-management is required. Micro-management is the term which describes the requirement to specify configurations or policies, with high level of specificity and detail, to a software module or component. The micro-management often makes it tedious for configuring firewalls optimally and many times, system administrators and users just go with either over-zealous and over-protective settings or very lenient settings.
2. Both over-protective and lenient firewall settings are not efficient for practical use because in the case of lenient configurations, attack scope increases exponentially. In the case of over-protective settings, apps maybe crippled and if configured to produce alerts, there are many alerts being produced every second that the users of specified system will get desensitized to warnings and will find workaround to firewalls.
3. Like any other program, firewalls are not completely immune themselves to vulnerabilities and security flaws, as observed in *Firewalls in News* section. And also, firewalls can be compromised from the inside through another program/service/file which has been compromised through means other than inter-network communication.
4. Firewalls provide a fair degree of security protection, but they are not a one-stop solution against network attacks like the Denial of Service attack, the Masquerader attack, and others. Though they are a part of what constitutes a "defense-in-depth" approach, they are not sufficient by themselves. Still, it is better to have a firewall deployed with minimal configuration, than no firewall at all.

10 In the News <>

1. Windows Firewall can be bypassed using NBNS.
   (8th May 2012) [1] [SANS]
2. IPTables can be bypassed using "–syn rules".
   (20th June 2014) [2] [CVE]
3. Juniper's ScreenOS firewalls possess VPN backdoor.
   (22nd Dec. 2015) [3] [PCWorld]

NOTE: All of the bypass vulnerabilities discussed below have been patched and are no longer vulnerable on current versions.

1. **Windows Firewall Bypass:** NetBIOS and its weaknesses often pave way for easier medium of spoofing and especially, Name Spoofing. These kind of spoofing attacks were well known since 2005. In the world of Internet today, NetBIOS Name Spoofing have serious impacts on our security. An effective way of preventing this exploit is to not use LM/NT hashes in the Windows systems. In the Domain Controller tutorial, we have discussed on how to disable LM/NT hashes [1].

2. **IPTables Bypass:** Synchronization rules and Password Synchronization features in IPTables platform using the development switches L2B-05.03.07 and L2E, L2P, L3E, and L3P before 09.0.06 sets an SNMP string to the same string as the administrator password, which allows remote attackers to obtain sensitive information by sniffing the network [2].

3. **Juniper's ScreenOS Bypass:** From a mixed cause of likely third party malicious code modifications and Juniper's own cryptography failures, a vulnerability had arisen in Juniper's ScreenOS firewalls which had the potential to allow attackers to decrypt VPN traffic originating from ScreenOS device interfaces. Subsequently, Juniper released patches to address the issue and updated the firmware [3].

11 Quiz: Firewall Background <>

Q1: Can Firewalls hinder applications? If so, how does that happen?

Q2: What is the difference between a packet filter firewall and an application layer firewall?

Q3: "Defense in depth" is mainly about firewall deployment and management. T/F?

Q4: Why are "hardware" firewalls more properly called "firmware" firewalls?

12 Firewalls in Windows <>

1. A (very) wide range of private firewall solutions;
2. For the most part, Microsoft's Windows Firewall is good enough;
3. Windows Firewall is a GUI-based firewall and packet filter;
   (a) It is a network firewall and an application firewall;
4. Configuration is also available through command-line interface.

1. Microsoft's Windows operating system, due to its immense popularity, enjoys a wide range of options in many types of applications. Firewalls are not an exception and, as such, there are at least 17 advanced firewall applications in Windows at the time, in active development. Additionally, many commercially successful anti-virus/malware suites have their own integrated firewall.
2. For most requirements and everyday configurations, Windows Firewall has intermediate-level features – not too advanced and not too basic. Windows firewall has an easy-to-use graphical user interface and can mostly be configured using the GUI alone. However, Windows Firewall can be configured by CLI as well.
3. By default, Windows Firewall contains three profiles: Domain Network, Private Network, and Public Network. Settings can be customized per profile.

13 Activity: Windows Firewall GUI <>

The following is for both incoming & outgoing communications:

1. Enable/disable firewall.
2. Rule Management:

   Allow/disallow a program/service's network traffic;

   Allow/block a port's network traffic;

   Manage existing rules.

1. **To Enable/Disable Windows Firewall:**
   (a) Open *Control Panel*.
   (b) Go to "System and Security" > "Windows Firewall".
   (c) In the left pane, click on "Turn Windows Firewall on or off".
   (d) Windows Firewall is turned on or off per profile. (If your computer is not connected to a domain, then you will not see the Domain profile, as is the case with our VM.)
   (e) Back in the main **Windows Firewall** screen, on the left pane, **Restore Defaults** will reset the entire configuration to factory default.
2. **Allow/Disallow a Program/Service:**
   (a) In the left pane of the main *Windows Firewall* screen, click on "Allow an app or feature through Windows Firewall".
   (b) Click on "Change Settings".
   (c) The list of services/programs displayed here are **Allowed**, which means, it is a white-list.
   (d) To allow a program/service, click on "Allow another app..." and browse to the location of program/service executable and select it. Before adding this program, click on "Network Types" to select for which profile(s) this rule will be enabled. Finally, click "Add".

(e) The profiles for which a rule applies can also be changed using the checkboxes in the *Allowed apps and features* list.

(f) To disallow a program/service, select the rule in the *Allowed apps and features* list and click on "Remove".

3. **Allow/Block a Port:**

(a) In the left pane of the main *Windows Firewall* screen, click on "Advanced Settings".

(b) In the left pane of the new window, select "Inbound Rules" or "Outbound Rules" to create rules for the respective direction of traffic. Either way, the process is similar.

(c) In the right pane, click "New Rule...".

(d) In the *New Rule Wizard*, select *Port* for the type of rule. Click "Next".

(e) Select either *TCP* or *UDP* and specify the range of ports or a single port to allow or block. Click "Next".
NOTE: Allowing all remote ports is NEVER recommended. When blocking, it is at the user's discretion.

(f) Select the appropriate option to allow, block, or allow a connection only if it is secure. Once you've made your selection, click "Next".

(g) Select any combination of *Domain*, *Private*, or *Public* to choose for which profile(s) the rule will be enforced.

(h) Give the rule a name and a description. (Good comments help keep track of the rationale behind a rule when you or someone else come back much later to figure which rules are still valid or not.) Click "Finish" to create the rule.

4. **Manage Existing Rules:**

(a) In the *Advanced Settings* left pane, select either "Inbound Rules" or "Outbound Rules" again. Rules can be duplicated easily using copy/paste, or removed by pressing **Delete**.)

(b) Creation of a new rule is the same as above. (We created rules for ports, but rules for programs/services or predefined entities can be selected in step *d* above.)

(c) To export or import, in the left pane, click "Windows Firewall with Advanced Security". Importing and exporting options are found in the right pane ("Import Policy..." and "Export Policy...").

14 Activity: Windows Firewall CLI <>

CLI functionality of Windows Firewall can be accessed through Network Shell scripting [netsh].

1. Enable/disable firewall.
2. Rule Management:

   Allow/disallow a program/service traffic;

   Allow/Block a port's network traffic;

   Manage existing rules.

1. **To Enable/Disable Windows Firewall (WF):** [4]
   (a) From the home screen, navigate to the search bar or use **WIN + S**.
   (b) Search for **cmd**. Right-click on **cmd** and select **Run as Administrator**.
   NOTE : This will not work without Admin rights.
   (c) At the cmd prompt, enter the command:
   `netsh advfirewall set allprofiles state on`
   to enable firewall for all three profiles.
   (d) Now enter the command:
   `netsh advfirewall set allprofiles state off`
   to disable firewall for all three profiles.
   (e) Alternative values for **allprofiles** are:
   currentprofile — **domainprofile** — **privateprofile** — **publicprofile**
   (f) To reset the entire WF configuration to factory default, use the command:
   `netsh advfirewall reset`

2. **To Allow/disallow/modify a program/service/port in WF:** [4]
   (a) To allow a program for all profiles, enter the command:
   `netsh advfirewall firewall add rule name="AllowProgramMyApp"`

dir=in action=allow program="C:PATH \ filename.extension" profile=any enable=yes
NOTE: If bypass is selected and dir=in connecting computers will need to be verified in the rmtcomputergrp along with setting the authentication/encryption flag. If `dir=out` the authentication/encryption flag must be set.

(b) The alternative values are as follows: **(dir=in | out), (action=allow | block | bypass), (enable=yes | no), (profile=any | public | private | domain)**

(c) To allow a port(i.e. port 655) for all profiles, enter the command:
netsh advfirewall firewall add rule name="AllowPort666" dir=in action=allow protocol=TCP
localport=655 profile=any enable=yes

(d) To disallow a program/service/port, use **action=block** . To resolve conflicts, **action=bypass** can be used.

(e) Deleting a rule(ie. a rule you named Rule43) can be done with the command:
netsh advfirewall firewall delete rule name=Rule43

(f) Although not recommended, all rules can be deleted with the command:
netsh advfirewall firewall delete rule all

(g) To modify an existing rule(i.e. a rule named AllowMessenger) to require a new security type, use the command:
set rule name="AllowMessenger" new security=authenticate **NOTE:** The keyword "new" is placed before the parameter you are changing.

(h) Example commands can be found with:
netsh advfirewall firewall add rule ?

(i) Existing rules can be scanned using command:
netsh advfirewall firewall show rule name=all profile=any .
This list can be very large and you may just want to view a specific rule with the command:
netsh advfirewall firewall show rule name=RName
where "RName" is the name of the rule you would like to view.

(j) To Export your policy use the command:
netsh advfirewall export "PATH \ filename.wfw"

(k) To Import a policy use the command:
netsh advfirewall import "PATH \ filename.wfw"

15 Challenge 1: Windows Firewalls <>

Use the "Windows Server 2012" and Kali VMs to perform the following tasks:

1. Disable Windows Firewall;
2. From the Kali VM, scan the Windows Server machine to see how it appears from an outside perspective. What ports are open? What other info can be gathered?
3. Now Enable Windows Firewall and perform the scan again. What are the differences?

Deliverables:

1. Observations of the differences in scanning outcomes with the firewall enabled and disabled.

Duration: 8 min.

16 Challenge 2: Windows Firewalls <>

Using GUI, block Remote Desktop Connection for all IP addresses:

1. Block it for all three firewall profiles;
2. Block it both incoming and outgoing.

Create an exception for an IP address or an entire subnet of IP addresses. For example, the subnet 192.168.1.100/24.

Deliverables:

1. Remote Desktop should be blocked in all cases except for the specified IP addresses.

Duration: 20 min.

HINTS:
For effective implementation, both programs and services must be customized while defining rules. Source executable of *Remote Desktop Connection* can be found by examining the `Open file location` option of RDP properties. Rules can be modified when and as needed.

17 Challenge 3: Windows Firewalls <>

Use Windows Firewall to perform the following tasks:

1. Add a rule to block problematic typosquatter website(s) that are continual sources of malware on your network. Test that the bad websites are blocked, but that you can still navigate normally to other websites.

Deliverables:

1. Only the problematic websites should be blocked; all others should still be accessible.

Duration: 10 min.

HINTS:

2. The typosquatter websites are **facbook.com**, **gogle.com**, and **mikrosoft.com**. They are all hosted on the "Ubuntu-LAMP" server.

   The regular websites to test against are hosted on the SEED Labs server. They are **wtmobilestore.com**, **wtelectronicsstore.com**, **wtcamerastore.com** and **wtshoestore.com**.

   To find the IP address for a domain name, you can use the `nslookup` command.

18 Firewalls in Linux <>

1. Linux is relatively command-line oriented; subsequently so are its firewalls;
2. `gufw` and `iptables` (previously `ipchains`) are examples of software firewalls for Linux;
3. `gufw` (GUI for Uncomplicated FireWall) is an easy-to-use front-end for a command-line interface (CLI) firewall, `ufw`;
4. `iptables` is a fairly advanced firewall with many integrated tools like user-space administration, packet filtering, protocol filters, and network filters.

1. Most Linux firewalls have a highly advanced and flexible command set for command-line interface (CLI).
2. For everyday uses, Ubuntu has `gufw` (GUI Uncomplicated FireWall). `gufw` is a front-end for `ufw`. It has an easy-to-use graphical interface, and can generally be configured through the GUI. If command-line is more your style, `iptables` is available and active in Linux kernel by default. It is entirely CLI. `ufw` is also available. It is derived from `iptables` and may be an easier option.
3. `gufw` by default contains three profiles: Home Network, Public Network, and Office Network. Configurations and settings can be customized per profile.
4. `iptables`' methodology is called "chains". They are configured as linked chains of rules.

19 Activity: gufw <>

1. Basic Functionality:
   - Setting preferences, managing profiles, enabling/disabling profiles, and controlling traffic.
2. Rule Management:
   - Adding, editing and removing rules.
3. Dynamic Rule Generation:
   - Creating/editing rules dynamically from reports.

1. **Basic Functionality:**
   (a) Open `gufw`: **System Settings** > **System**, "Firewall Configuration".
   (b) In the top menu, go to **Edit** > **Preferences**.
   (c) In the dialog box, change "Logging" to *Full*.
   (d) Check the boxes for "Logging Gufw activity" and "Show confirm dialog box for deleting rules".
   (e) Adjust the Listening Report's "Refresh Interval" to *1"*.
   (f) In the **Profiles** section, there are three profiles by default: Office, Public and Home. Currently we have Home and Lab. Any profile except the current one can be deleted. A total of 255 profiles can be created. To create profiles, use the + symbol. To delete, use the - symbol. Profiles can be renamed by double-clicking their name. A profile's name may be a maximum of 15 characters.
   (g) Enabling or disabling *Gufw* is done per profile. Select a profile. Toggle the "Status" button *ON* or *OFF*, and the firewall will be enabled or disabled respectively while using that profile.
   (h) Controlling global traffic is done per profile, as well. Select a profile and set values in the "Incoming" and "Outgoing" drop-downs. *Allow* permits all communications for that channel. *Deny* drops all communication for that channel. *Reject* drops all communications as well, but also sends a message to the computer on

the other end of the connection that their communication attempts have been rejected.

2. **Rule Management:**

   On the main window, the **Rules** section can be used to create custom rules. Once again the + and − symbols are used to add and delete rules.

   In the "Add a Firewall Rule" window, there are three tabs.

   (a) The "Preconfigured" tab offers easy configuration for specific applications. The list of applications can be filtered by *Category* and *Subcategory*.

   (b) The "Simple" tab allows configuration of specific port and transport protocol combinations for incoming and outgoing traffic.

   (c) The "Advanced" tab contains options that enable the creation of completely customized rules. New options presented here are **Interface**, **Logging Flexibility**, and **Custom IP Addresses and Ports**. The **Interface** tab is for the creation of rules for specific interfaces available to the system. For **Logging Flexibility**, it is always recommended to *Log All*. Last but not least, a custom range of ports and IP addresses can be used to design highly specific firewall rules.

   The "Policy" field presents a new option, *Limit*. *Limit* will deny connections from an IP address if it attempts to connect more than 5 times within a 30 second time frame.

3. **Dynamic Rule Generation**

   (a) On the main window, **Report** lists ports that are actively listening. This feature is not very effective for assisting with attack prevention, since it only generates information about active ports. But, it is very helpful for thwarting continuous, repeated exploit attempts.

   (b) Firewall rules can be added dynamically from **Report**. Click a port in the list, then click the + button. This jumps to a new rule in the **Advanced** tab with many details filled in automatically for you, matching the rule you selected.

20 Activity: iptables <>

1. Basic Functionality:
   - Enabling/disabling, checking status, and logging
2. Chain Management:
   - Adding, editing and removing chains.
3. Rule Management:
   - Adding, editing and removing rules.

1. **Basic Functionality:**

   (a) To check what rules are in effect, try either of the following commands and their various options:

   `sudo iptables -L[ chain-name|TCP|UDP][ -v]`

   `sudo iptables -S[ chain-name|TCP|UDP]`

   Rules are grouped according to chain. The columns displayed for each rule: **target**, **prot**(ocol), **opt**(ions), **source**, **destination** and **notes**.

   (b) It is not necessary to log all traffic in and out of a firewall. Doing so can create very large log files. It is more practical to only log dropped packets. That can be set using the following command:

   `sudo iptables -I INPUT -j LOG --log-prefix "iptables denied: " --log-level 7`

   Logs are saved at */var/log/syslog*. There are seven levels of logging:

Level Number	Meaning
0	Emergency: system is unusable
1	Alert: action must be taken immediately
2	Critical: critical conditions
3	Error: error conditions
4	Warning: warning conditions
5	Notice: normal but significant condition
6	Informational: informational messages
7	Debug: debug-level messages

(c) To save all changes, so they remain the next time the `iptables` service is re-started, you must use the following command (on Ubuntu):

`sudo /sbin/iptables-save`

(d) `iptables` is enabled by default in Linux, and is integrated at the kernel level. There's no real way to turn it on or off. One can fully configure all settings as desired, and save the configuration file. Then, use a command to flush every rule, set the default chains to accept all traffic, configure any necessary routing or services for basic network functionality, and save that configuration file separately. Then, switch back and forth between the config files to not lose the original settings.

The flush command is:

`sudo iptables -F`

2. **Chain Management:**

(a) `iptables` has three chain types by default. They are: *INPUT*, *FORWARD*, and *OUTPUT*. The *INPUT* chain is for processing incoming connections. The *OUTPUT* chain is for processing connections originating from your own computer. The *FORWARD* chain is for processing connections that are only being forwarded and are not intended for this computer itself. This chain is not generally used.

To create more chains, use the command:

`iptables -N chain-name`

(b) When a request does not match any rules in a chain, `iptables` has default policies to handle it. The default policies can be set to either accept or drop.

```
iptables --policy INPUT ACCEPT|DROP
iptables --policy OUTPUT ACCEPT|DROP
iptables --policy FORWARD ACCEPT|DROP
```

(c) To list the rules present in a chain, use the command:

`iptables -L chain-name`

(d) To flush a chain of all rules, use the command:

`iptables -F chain-name`

(e) To rename a chain, use the command:

`iptables -E old-chain-name new-chain-name`

(f) To delete a chain, use the command:

iptables -X chain-name

Root privileges are required to delete one of the three default chains.

3. **Rule Management:**

(a) To add rules to a chain, use the following command syntax:

sudo iptables -A chain-name[-s {IP Address|hostname}][-d {IP Address|hostname}][-p tcp|udp] -j ACCEPT|DROP|REJECT |LOG|chain-name

`-A` specifies the instruction to append a rule to a chain.
`-s` specifies a source IP address or name to match.
`-d` specifies a destination IP address or name to match.
`-p` specifies a protocol to target.
`-j` specifies what to do when the rule is processed, including jumping to other chains for processing.
There are also `-sport` and `-dport` options to specify source and destination ports [5].

An interesting extra: `!` can be used as a "not" operator with -s and -d.

These and numerous other options are available for -A and the other commands that follow.

(b) Instead of `-A` to append the rule to the end of a chain, `-I` inserts a rule at a specific position within a chain. For example, to insert a rule at the 4th position in a chain:

sudo iptables -I chain-name 4

(c) `-R` specifies the position of an existing rule to replace. For example, to replace the 4th rule in a chain:

sudo iptables -R chain-name 4

(d) To delete a rule from any chain, use `-D`. There are two ways a rule can be deleted using this option. If we provide a rule position number, then the rule at that particular position will be deleted from the chain. If no rule number is provided, then the first rule matching the given options is deleted. The rul must be an exact match, though. For example, to delete the rule at the 4th position in a chain:

sudo iptables -D chain-name 4

To delete the first rule permitting UDP traffic:

sudo iptables -D chain-name -p udp -j ACCEPT

21 Challenge 4.a: Linux Firewalls <>

Use the "Ubuntu-GUFW" and Kali VMs to perform the following tasks:

1. Disable both firewalls;
2. From the Kali VM, scan the Ubuntu VM to see how it appears from an outside perspective. What ports are open? What other info can be gathered?

Deliverables:

1. Observation of what is visible from outside when the firewalls are disabled.

Duration: 10 min.

HINTS:

1. When disabling the firewalls, we need to disable both. But because `iptables` is configured at the kernel level, one does not simply disable *iptables*. Make it as passive as possible on all chains.

22 Challenge 4.b: Linux Firewalls <>

(Continued from the previous task.)

3 Enable `gufw`, leaving `iptables` as-is. Set "Incoming" to *Deny*, if not already so;

4 Repeat the scans from the Kali VM. What differences do you see?

5 Disable `gufw`, and set the default chains in `iptables` to `DROP`.

6 Scan again from the Kali VM. Any differences?

Deliverables:

1. Observation of what is visible from outside when the firewalls are enabled in turn individually.

Duration: 15 min.

23 Challenge 5: gufw <>

On the "Ubuntu-GUFW" VM, use `gufw` to perform the following tasks:

1. Block and log all traffic related to Steam and send a message to the person attempting communication, conveying the block;
2. Add a rule to block problematic typosquatter website(s) that are continual sources of malware on your network. Test that the bad websites are blocked, but that you can still navigate normally to other websites. Remove the rule.

Deliverables:

1. Any usage of Steam should be rejected (versus simply dropped);
2. Only the problematic websites should be blocked; all others should still be accessible.

Duration: 15 min.

HINTS:

1. The typosquatter websites are **facbook.com**, **gogle.com**, and **mikrosoft.com**. They are all hosted on the "Ubuntu-LAMP" server.

   The regular websites to test against are hosted on the SEED Labs server. They are **wtmobilestore.com**, **wtelectronicsstore.com**, **wtcamerastore.com** and **wtshoestore.com**.

   To find the IP address for a domain name, you can use the `nslookup` command.

24 Challenge 6: iptables <>

On the "Ubuntu-GUFW" VM, use `iptables` to perform the following tasks:

1. Write two rules to block pings. Test your rules to make sure they work. Then, remove them;
2. Add a rule to block problematic typosquatter website(s) that are continually sources of malware on your network. Test that the bad websites are blocked, but that you can still navigate normally to other websites. Remove the rule.

Deliverables:

1. Pings to "Ubuntu-GUFW" should be unresponsive;
2. Only the problematic websites should be blocked; all others should still be accessible.

Duration: 15 min.

HINTS:

1. Two rules are needed, one for the *INPUT* chain and the other for *OUTPUT*.
2. See the `gufw` challenge above for the websites to test against.
   `iptables` can block by either IP address or domain name.

25 Bonus Challenge: Windows Firewall GPO <>

Windows Firewall can be configured up to an extent through Microsoft's Group Policy Editor. For domain profile, do the following:

1. Disable `local program exceptions`;
2. Disable `prohibit notifications`;
3. Enable `logging dropped packets`;
4. Disable `local ports exceptions`.

Deliverables:

1. Client machines should be unable to specify any local program exceptions contrary to Windows Firewall policies set by Domain Controller.
2. Windows Firewall should no longer notify when it is in a critically insecure state.
3. Dropped packets should be logged and viewable in Windows Event Viewer.
4. Client machines should be unable to specify local exceptions for ports contrary to Windows Firewall policies set by Domain Controller.

Duration: 30-45 min.

HINT: Firewall settings are a component of network connections to be used as a part of an overall network for deploying administrative templates of computer configuration.

26 Conclusion <>

1. There are many types of firewalls to choose from. Despite the variety, their core functionality is similar;
2. Firewalls are easy to install and access, but can be relatively difficult to configure for optimal functionality;
3. Through these walkthroughs and hands-on activities, we've learned some basic management of firewalls in Windows and Linux.

1. There are many software firewalls to choose from. Some of them offer easy to use graphical user interfaces, like `gufw`, some are very sophisticated to use, like `iptables`, and some are flexible to accommodate both ease of use and sophisticated rule management, like `Windows Firewall`.

   Despite the variety, the core functionality of every firewall remains the same: to place the management of network traffic in the user's control.

2. Unlike service applications which are easier to maintain and configure, firewalls are relatively difficult in nature to maintain and configure for optimal functionality. Setup of a firewall is very trivial in most cases, but the ensuing configuration can take a long time depending on an organization's needs.

3. In this tutorial, we have seen how `Windows Firewall` works (both GUI and CLI), and we have seen examples of Ubuntu firewalls with `gufw` and `iptables`.

27 Appendix: Solutions and Change Log <

1. Answers to the Quiz;
2. Solutions to the Challenges:
   (a) Challenge 1;
   (b) Challenge 2;
   (c) Challenge 3;
   (d) Challenge 4;
   (e) Challenge 5;
   (f) Challenge 6;
   (g) Bonus Challenge;
3. Change Log.

Answers to the Quiz:

1. Over-protective configurations can cause major hindrances to some applications, effectively crippling them.

2. Packet filter firewalls are able to control traffic related to ports, IP addresses and network requests. Application layer firewalls can do all that and in addition, can configure rules with respect to specific applications and services.

3. "Defense in depth" approach is not about just firewalls. There are many layers of protection. Firewalls are often the first line of protection for a network.

4. They are more properly called "firmware" firewalls because most firewalls carry out their duties at the software level as applications in Linux, rather than as actual circuits in hardware.

Solutions to the Challenges:

1. **Challenge 1:**

   (a) In "Inbound Rules" section, click on "New Rule" at the right pane.

   (b) In resultant wizard, Custom Rule type. All Programs.

   (c) Protocol type: ICMPv4. Scope: Any IP address.

   (d) Action: Block the connection. Profile: All three.

   (e) Give a RULENAME and description. Then, finish.

   (f) Repeat the same by changing Protocol type to ICMPv6.

   (g) Repeat the same thing twice (once for ICMPv4 and another for ICMPv6) in "Outbound Rules".

   To disable rules using CLI, use the command: `netsh advfirewall firewall set rule name="RULENAME" new enable=no`, where RULENAME is the name of rule which one has assigned at the end of creating every rule.

2. **Challenge 2:**

   (a) In "Inbound Rules" section, click on "New Rule" at the right pane.

   (b) In resultant wizard, Custom Rule type.

   (c) Program: The program path is: `C-Windows-System32-mstsc.exe`.

   (d) Services: *Customize*. Apply to all services beginning with the name "Remote Desktop Services".

   (e) Any protocol type – Any IP addresses – Block the connection-All Profiles

   (f) Give a name and description. Then, finish.

   (g) Repeat the same for "Outbound Rules".

   To create an exception for 192.168.1.100/24 subnet,

   (a) In "Inbound Rules" section, right click on the rule created in PART-1 of the challenge.

   (b) Go to `General` tab. Change *Action* from `Block the connection` to *Allow the connection*.

   (c) Go to `Scope` tab. Do the following step for both `Local IP address` and `Remote IP address`

   (d) Change radio selection to `These IP addresses:`. Click on button `Add` and enter the subnet value `192.168.1.100/24`

   (e) Apply the modifications and `OK` to change the rule.

   (f) Repeat the same for "Outbound Rules".

3. **Challenge 3:**

   (a) To block the offending website(s), you will need an IP address. In command prompt execute the following command:
   nslookup websitename

   (b) In the Windows Firewall Advanced Settings:
   Create an inbound and outbound rule as follows:

      i. Select "New Rule"
      ii. Select "All Programs"
      iii. Protocol type Any and All Ports
      iv. Under "Which remote IP addresses does this rule apply to?"
      v. Add the IP address of the offending website and select "Next".
      vi. Select "Block the connection"
      vii. Select for all profiles.
      viii. Name the rule.

   (c) Now test to make sure your rule works

4. **Challenge 4:**

   (a) i. In gufw, switch the state of the current profile to *OFF*.
      ii. Open a terminal.
      iii. Execute the following 3 commands to make sure each of the default chains in iptables are set to ACCEPT:
      sudo iptables --policy INPUT ACCEPT
      sudo iptables --policy OUTPUT ACCEPT
      sudo iptables --policy FORWARD ACCEPT
      iv. You might execute the following commands to confirm each chain states "Chain {chain-name} (policy ACCEPT)":
      sudo iptables -L
      OR
      sudo iptables -L INPUT
      sudo iptables -L OUTPUT
      sudo iptables -L FORWARD

   (b) On the Kali VM:

      i. Open a terminal.
      ii. Initially, execute the following command to see if you can determine the IP address of the target machine ("Ubuntu-gufw"):
      nmap 192.168.1.0/24
      iii. Once you have the IP address, execute the following command:
      nmap -p "*" {IP Address} -sV -sS -O -T4

iv. What you see is that Nmap reports all ports as closed, and the OS could not be identified.

(c) Return to the "Ubuntu-GUFW":

i. In `gufw`, switch status to *ON*.

ii. Set "Incoming" to *DENY*, if it is not already set to that.

(d) Repeat step 2. Nmap reports that all ports are now filtered. The OS is still not recognizable.

(e) Return to the "Ubuntu-GUFW":

i. In `gufw`, switch the state of the current profile to *OFF*.

ii. In the terminal, execute the following commands:
`sudo iptables --policy INPUT DROP`
`sudo iptables --policy OUTPUT DROP`
`sudo iptables --policy FORWARD DROP`

(f) Repeat step 2. Nmap still reports that all ports are filtered, the OS indeterminate.

(g) Before continuing, set `iptables` back to an open state:
`sudo iptables --policy INPUT ACCEPT`
`sudo iptables --policy OUTPUT ACCEPT`
`sudo iptables --policy FORWARD ACCEPT`

5. **Challenge 5:**
Open `gufw`.

(a) In the **Rules** section, use the + button to create a new rule. On the **Preconfigured** tab:
Block and log all traffic related to Steam and send a message to the person attempting communication, conveying the block.

i. Policy: "Allow".

ii. Direction: Both. Category: All. Subcategory: Telephony.

iii. Application: Skype-Normal. Copy values and Jump to "Advanced" tab.

iv. Log all. To ports - 23390:23399. Add.

(b) i. To block the offending website(s), you will need an IP address. In the terminal, execute the following command:
`nslookup` {**`websitename.tld`**}

ii. In the **Rules** section, use the + button to create a new rule. On the **Advanced** tab:
Policy: *Deny*;
Direction: *Both*;
Interface: *All Interfaces*;
Log: *Log All*;
Protocol: *Both*.

iii. To, IP: {IP Address};

iv. Add the rule.

v. Test that you are no longer able to get to the offending website(s) in the browser. Try **gogle.com** and **facbook.com**

vi. Test that you can still reach other unrelated websites. Try **wtcamerastore.com**, **wtelectronicsstore.com**, and **wtshoestore.com**. This set of websites should not be blocked.

vii. You might experiment with the difference between setting Direction in the rule to *In* or *Out*. You should find that *In* does not block web requests (because they originate from your own computer).

viii. Viewing the **Rules** section, select the rule in the list of rules, and click – to delete it.

6. **Challenge 6:**
On the "Ubuntu-GUFW" VM, open a terminal:

(a) i. The following should work:
sudo iptables -A INPUT -p icmp -j DROP
More specifically:
sudo iptables -A INPUT -p icmp --icmp-type 8 -j DROP
Or perhaps more appropriately [6]:
sudo iptables -A OUTPUT -p icmp -j ACCEPT
sudo iptables -A INPUT -p icmp --icmp-type echo-reply -s 0/0 -j ACCEPT
sudo iptables -A INPUT -p icmp --icmp-type destination-unreachable -s 0/0 -j ACCEPT
sudo iptables -A INPUT -p icmp --icmp-type time-exceeded -s 0/0 -j ACCEPT
sudo iptables -A INPUT -p icmp -j DROP

ii. To delete the rules, repeat any of the commands you executed above and change `-A` to `-D`.

(b) i. To block the offending website(s), `iptables` enables you to use either IP address or domain name. Blocking by IP address might be a little easier to implement in our current case, but it is not necessarily appropriate in a real-world environment; whereas blocking by domain name is better targeted, but can be more tedious at the moment if entering all the offending websites:
In a terminal, execute the following sequence of commands:
sudo iptables -I INPUT 1 -s {IP Address} -j drop
sudo iptables -I OUTPUT 1 -d {IP Address} -j drop
Or, as the better option:
sudo iptables -I INPUT 1 -s gogle.com -j drop
sudo iptables -I OUTPUT 1 -d gogle.com -j drop
... etc.

ii. Test that you are no longer able to get to the offending website(s) in the browser. Try **gogle.com** and **facbook.com**

iii. Test that you can still reach other unrelated websites. Try **wtcamerastore.com**, **wtelectronicsstore.com**, and **wtshoestore.com**. This set of websites should not be blocked.

iv. To delete the rules:
sudo iptables -D INPUT 1
sudo iptables -D OUTPUT 1
etc.

7. **Bonus Challenge:**

`Computer Configuration` –> `Administrative Templates` –> `Network` –> `Network Connections` –> `Windows Firewall` –> `Domain Profile.`

Details:

(a) Disabling `Local program exceptions` will make it work such that the client machines cannot specify any local program exceptions to Windows Firewall policies set by Domain Controller.

(b) Windows Firewall provides notifications only when in a critically unsecure state. As such, `Prohibit Notifications` should be disabled.

(c) Dropped packets sometimes hold essential value in conducting forensic analysis. As such, it is always recommended to perform `logging of dropped packets`.

(d) Disabling `Local ports exceptions` will make it work such that the client machines cannot specify any local exceptions for ports in contrast to Windows Firewall policies set by Domain Controller.

Change Log:

Ver.	Date	Authors	Changes
v1.0	Apr 23 2016	Ananth Jillepalli	First draft of tutorial.
v2.0	July 13 2016	Ananth Jillepalli	Major content additions and remodeled the structure.
v3.0	Feb 3 2017	Colton Hotchkiss	Grammatical improvements to the Windows section.
v3.0	Feb 3 2017	Colton Hotchkiss	Added and modified rules in both Windows firewall Walkthroughs.
v3.0	Feb 3 2017	Colton Hotchkiss	Replaced challenge I and added challenge III.
v3.0	Feb 3 2017	Gabe Gibler	Grammatical improvements to the Linux section.
v3.0	Feb 4 2017	Gabe Gibler	Rearrangements to the order of presentation.
v3.0	Feb 4 2017	Gabe Gibler	Removed the Skype challenge from the Linux section. Added a challenge to the Linux section to see basic outcomes of enabling/disabling the firewalls. Added corresponding solutions to the appendix.
v3.0	Feb 4 2017	Gabe Gibler	Modified the default web page on the LAMP VM. Modified the challenge in the Linux section about blocking the LAMP sites. Modified the corresponding solutions in the appendix.
v3.1	Feb 8 2017	Gabe Gibler	Changed the content of some slides to better present the capabilities of firewalls and the distinction between software and hardware firewalls. Changed the background questions to remove questions pertaining to removed content, and to add questions and update existing questions to better reflect content changes.
v3.2	May 10 2017	Gabe Gibler & Colton Hotchkiss	Changed to CC-ByNCSA license. Made introductory sections consistent among tutorials. Added network diagram.
v3.3	May 12 2017	Gabe Gibler & Colton Hotchkiss	Making body styles consistent among tutorials.
v3.4	July 29th 2017	Ananth Jillepalli	Standardization (network layout diagram, edits, consistency, TeX markup cleaning, and more).

References

[1] Bojan Zdrnja, "Windows firewall bypass vulnerability using netbios name spoofing," May 2012. [Online]. Available: https://isc.sans.edu/diary/Windows+Firewall+Bypass+Vulnerability+and+NetBIOS+NS/13156

[2] CVE, "Vulnerability details : Cve-2012-2663," Feburary 2012. [Online]. Available: https://www.cvedetails.com/cve/CVE-2012-2663/

[3] Lucian Constantin, "Juniper's vpn backdoor: buggy code with a dose of shady nsa crypto," December 2015. [Online]. Available: http://www.pcworld.com/article/3017803/security/the-juniper-vpn-backdoor-buggy-code-with-a-dose-of-shady-nsa-crypto.html

[4] Microsoft, "Netsh advfirewall firewall commands," August 2009. [Online]. Available: https://technet.microsoft.com/en-us/library/dd734783(v=ws.10).aspx

[5] Oskar Andreasson, "Iptables tutorial 1.2.1: 10.2. implicit matches," February 2017. [Online]. Available: https://www.frozentux.net/iptables-tutorial/chunkyhtml/x2436.html

[6] StackExchange, Ask Ubuntu, "How can i block ping requests with iptables?" February 2017. [Online]. Available: http://askubuntu.com/questions/17548/how-can-i-block-ping-requests-with-iptables