Firewall Management

A Guide to Usage and Configuration of Firewalls in Windows and Ubuntu

Ananth Jillepalli & Risab Manandhar

August 15, 2016
Version 2.1

University of Idaho

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.

Contents

1 Objectives of this Tutorial >

1. Understand a little bit of the history and capabilities of Firewalls.
2. Understand how to manage firewalls through graphical user interfaces (GUI) on Windows and Ubuntu OS.
3. Understand the basics of rule-writing to manage firewalls through command line interfaces (CLI) on Windows and Ubuntu operating systems (OS).
4. Apply the gained knowledge on some challenges (in managing firewalls through both GUI and CLI).

This tutorial is not a complete user's guide to Firewall management. In contrast, this tutorial covers just the basics of Firewalls. In the ensuing tutorial, we will briefly explain, through either activities or challenges, the following specific things 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 capabilities of Firewalls, we must also look at a bit of its' history/background so that we can better get to know the foundational development of Firewalls. Firewalls capabilities are not just restricted to the ones discussed in this tutorial. Firewall is a complete suit with a host of other functionalities.
2. Fundamentals of Firewall management through Graphical User Interfaces (GUI), including granting of preferential network access permissions to applications and setting default global permissions, are all part of the content in this tutorial.
3. Knowledge of using Command Line Interface (CLI) based writing of rules for defining application's network access privileges and further advanced settings can be gained from this tutorial.
4. Finally, the skills learnt in the tutorial can be put to test by the user through undertaking challenges presented in the tutorial, clearing whom, will provide a strengthen the user's understanding of basic Firewall management.

2 Required Background <>

We assume that the reader of this tutorial has an extent of background knowledge in the following areas:

1. Working experience on usage of computers and software applications, like web browsers, and virtualization apps.
2. Basic overall idea of computer networks and Internet.
3. Fundamentals of networking mechanisms like domain setup, packet dropping, etc.,
4. An overall idea on general issues like data privacy, computer/network security, etc.,

Due to restrictions on time and manpower resources, we are not able to make the ensuing tutorial to be completely self-contained from the perspective of a user. As such, the tutorial is best used when the user already has 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 on using computers, installing and using common software applications (particularly web browsers, and virtualization software platforms). The tutorial does not explain how to navigate within the operating system's graphical user interface (GUI). Similarly, the tutorial also does not explain how to browse the Internet, how to install software applications, and how to use/navigate software applications.
2. An overall idea on the working os computer networks and Internet. The tutorial expects a user to understand common computer networking terms and their technical meanings. For example, “packets”, “streams”, ”protocols”, and “accept / drop packets” etc.,
3. Fundamental or very basic idea on networking mechanisms and computer networks. The tutorial expects a user to understand technical concepts like OSI model of networks, setting up a domain, delegating a domain controller, and assigned clients to the domain, and basic network attacks like man-in-the-middle attack etc.,
4. Also bit of exposure to logic notations and elementary programming skills would be very helpful in understanding the firewall rule writing. An overall brief idea on general computer-related issues like “data privacy”, “computer security”, “network security”, “application permissions”, “network access privileges” etc., will help a lot.

3 Hardware and Software Requirements <>

We recommend having at least the following hardware and software specifications for smooth execution of this tutorial's activities and challenges:

1. A computer which can at least boot 2 virtual machines (VMs) smoothly, with no noticeable lag and delay
2. A functional virtualization software platform. For example, VMWare or VirtualBox.
3. One vanilla Microsoft Windows Server 2012 R2 VM(*) and two vanilla Ubuntu 16.04 LTS VMs(**).

The activities and challenges of this tutorial occur in both Windows and Ubuntu operating systems. As such, it is imperative that the user of this tutorial has a machine which is powerful enough to boot at least 2 virtual machines smoothly, at a time. For the sake of consistency, more specific notation and VM requirements are given as below:

(*) The Windows Firewall portion of the current tutorial can be followed with any Windows operating system which has Windows Firewall installed on it. If the user choses to be consistent with the tutorial's navigation, we will be navigating Windows Firewall of Windows Server 2012 R2 VM.

(**)
Ubuntu VMs setup advise: Out of the two Ubuntu 16.04 LTS VMs, one should have the package `gufw` installed. Let's call this VM as "**VM-gufw**". The second one should have a `LAMP` stack installed, such that it can host some random website on *localhost*. Let's call this VM as "**VM-LAMP**".

Note: Please go to part **2** of the **Appendix** section for a guide on how to install LAMP stack on Ubuntu 16.04 and also for resources and website links to access some of the required software for the tutorial.

4 Problem Statement: 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 is different than just logging the communications or monitoring the traffic.

3. The process is convoluted at many stages due to the intricacies of user-involvement, a suitable balance between protection, permissions, and allowances.

1. Traffic controlling has long been an issue since the early 1980s when the internet was a fairly new technology when compared with today. Early network traffic controlling was done 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 operation, it is not necessary that controlling always involves and is limited to logging and monitoring.

3. Traffic controlling itself has been a target of many convoluted issues because of the problems arising from a mapping between users and software efficiency. A suitable balance between protection, permissions, and allowances must be forged because where protection and permissions are high, allowances are low - which is not good because then users start finding alternatives and work-around methods to bypass protections, thus, invalidating them. When allowances are high, protection and permissions are low, giving a higher scope of attack and vulnerabilities.

4. As such, the then existing method of direct router configurations does not catch up to the evolving and increasing demands in requiring a more balanced controlling mechanism rather than just plainly deny or allow something. To solve this problem, much research was invested into it from NASA, UC Berkeley, UC San Diego, Stanford and Lawrence Livermore.

5 Solution: Firewalls <>

1. Three generations of Firewalls:

   (a) 1st Generation: (Network Layer) "Packet" Filters

   (b) 2nd Generation: (Transport Layer) "Stateful" Filters

   (c) 3rd Generation: (Application Layer) "Protocol" Filters

   (d) In all the three generations above, the specifications are the highest level operable and not an exclusive level of operation.

(a) **First Generation:** The first published research on *firewalls* started appearing around 1988, when Digital Equipment Corporation (DEC) developed filter systems known as packet filtering firewalls. 'Firewall' as an entity when first introduced, was operating at network layer, in the form of packet filters, targeting network addresses and ports of the packet. From that information, the firewall module determines if the packet should be allowed or blocked.

(b) **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's capabilities, in this second generation, firewalls were able to operate up to transport layer of the OSI model. Stateful inspection of packets is the distinguishing feature in this generation, where a packet is determined if it belongs to new connection, an existing connection or does not belong to any connections.

(c) **Third Generation:** An application suite named as *FireWall ToolKit* (FWTK), developed by Wei Xu, Peter Churchyard, and Marcus Ranum, laid the foundations for this generation's firewalls. First application layer firewall was an extension of FWTK, enhanced by Wei Xu. The most significant advantage of application layer firewall is the fact that various protocols are now comprehensible by firewalls and can be included in rule-making. The protocols include, but are not restricted to: File Transfer Protocol, Domain Name System, and HyperText Transfer Protocol.

(d) All of the three generations above are not representative of exclusive entitlements. That is, an application firewall does not just work at the application layer, but it means that it can work at any layer up to application layer. That is, the capabilities of the firewalls are increasing as the generations go on increasing.

6 Tutorial's Approach <>

1. Current day classification because of hybrid nature:
   (a) Software based firewalls
   (b) Firmware based firewalls
2. Though their location of operation is different, their core functionality is similar.
3. For this tutorial, our approach will be towards software based firewalls.

1. Current day classification of firewalls are of hybrid nature because most firewalls have all the capabilities of operating until application layer and a few have additional integrated capabilities like proxies and network address translation. Because of this hybrid nature in performance of firewalls, the modern classification is consolidated into two levels. They are:
   (a) **Software-based firewalls:** These type of firewalls operate at software level and their maximum extent of control reach extends to the networking interface of the system, in particular. Popular examples of this kind of firewall are: Windows Firewall, Comodo Internet Security, ZoneAlarm, Norton 360, and PeerBlock. Software-based firewalls are not able to exercise significant control over traffic at network device level, like in routers.
   (b) **Firmware-based firewalls:** These type of firewalls operate at firmware device level and their maximum extent of control resides in the operational network devices in which they are configured for. Popular examples of this kind of firewall are: Cisco's ASA firewall, Juniper Network's ScreenOS, Dell's SonicWall, and Untangle's Zeroshell. Hardware-based firewalls are not able to exercise significant control beyond their own device range, particularly not at an operating system level, like in software-based firewalls.
2. Though the presence and location of operation for firewalls is different in the above two types of firewalls, they have a similar core functionality, which is to control traffic. for this tutorial, we will be approaching from a software-based firewall perspective.

7 Related Articles: Firewalls in 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]
4. Cisco's ASA firewalls compromised using malformed UDP packets. (11th Feb. 2016) [4] [PCWorld]

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].

4. **Cisco's ASA Bypass:** Cisco Systems' Adaptive Security Alliance (ASA) firewalls were confronted (and subsequently patched) with a critical bypass vulnerability which had the potential to allow remote attackers to over the firewalls, which are configured as virtual private network servers by simply sending malformed UDP (User Datagram Protocol) network packets to the firewalls [4].

   NOTE: All of the bypass vulnerabilities discussed below have been patched and can no longer be functionally reproduced in a similar fashion.

8 Outline of Tutorial <>

1. Background on firewalls.
2. Firewalls in Windows.
3. Hands-on activity [Windows Firewall (GUI and CLI)].
4. Challenges I, II.
5. Firewalls in Linux.
6. Hands-on activity [Ubuntu Linux (GUFW and IPtalbes)].
7. Challenges III, IV.

9 Background: Firewalls <>

1. Control of network traffic to and from a computing device.
2. They are first line of network protection into networks.
   - Different from IPS/IDS.
3. Control which program can/cannot access network.
4. Networks have incoming and outgoing communications.
   - There are separate rules for both incoming and outgoing traffic.

1. The primary and the most significant aspect of Firewalls is to control network traffic to and from a computing device and / or the ability of controlling the communication media going outside and coming inside of a system. Any network security system satisfying the above responsibility can be called as 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 of such firewalls can be similar to Intrusion Prevention/Detection, and User Identity Integration. However, the features integrated in the firewalls are not so advanced that they can be classified as complete instances 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 a state of computing system. Controlling activities naturally involve monitoring and logging of reports, but security systems which just do monitoring and logging are not usually considered as 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.

10 Background: Firewall Limitations <>

1. Micro-management is required for efficient functionality.
2. Over-protective settings can cripple some applications. If configured for alerts, they can desensitize users to warnings.
3. Firewall can be compromised/shut down by other compromised programs.
4. Firewalls are not a one-stop solution against network attacks and although a part of it, they are not considered "defense in depth" by 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 very lenient firewall settings are not efficient for practical use because in the obvious case of very lenient configurations, attack scope increases exponentially and in the case of over-protective settings, applications 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 or alternatives 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 that originating from inter-network communication.

4. Firewalls provide a fair degree of security protection, but they are not one-stop solution against network attacks like the Denial of Service attack, the Masquerader attack, and etc., Though they are a part of what constitutes "Defense in depth" approach, they are not alone sufficient. All said and done, it is still better to have a firewall deployed with optimal configurations, rather than having no firewall and exposing system to vulnerabilities.

11 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 through CLI is also available.

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. In modern times, commercially successful anti-virus/malware suites have their own private firewall integrated as a part of the suites.

2. For most requirements and everyday configurations, Microsoft's Windows Firewall has intermediary-level features which are not too advanced and not too basic. Windows firewall has easy-to-use graphical user interface and can be configured mostly through GUI, without even having the need to divulge into CLI commands. However, if needed and preferred, Windows Firewall can be setup and/or configured through CLI as well.

3. Usually, Windows Firewall contains three profiles, one for each: Domain Network, Private Network, and Public Network. The configurations and settings can be localized to each profile rather than one configuration for all three locales.

12 Questions: Firewall Background

<>

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

Q2: What is the difference between a network firewall and an application firewall?

Q3: Can firewall safeguard against Man-in-the-middle attack?

Q4: “Defense in depth” approach technique is mainly about firewall deployment and management.

Q1: Over-protective configurations can cause major hindrances to some applications, effectively crippling them.

Q2: Network firewalls are able to control traffic related to ports, IP addresses and network requests. Application firewalls can do all that and in addition, can configure the same with respect toapplications and services.

Q3: No, firewalls cannot do that. Although firewalls can control incoming and outgoing traffic, it cannot detect changes in network communication content itself.

Q4: “Defense in depth” approach is not about just firewalls. There are many layers of protection such an approach and firewalls are commonly the first line of protection for a network.

13 Walkthrough: Windows Firewall GUI <>

The following is for both incoming and outgoing communications:

1. Enable/disable firewall.
2. Allow/disallow a program/service's network traffic using firewall.
3. Allow/block a port's network traffic using firewall.
4. Rule Management.

Please refer to step 3 of the Hardware & Software Requirements slide to see the required VM setup for this part of the tutorial.

1. **To Enable/Disable Windows Firewall (WF):**
   (a) Go to *Control Panel*.
   (b) Navigate to *System and Security*. (Change "View by:" selection in the top-right of the frame to *Category* if unable to spot System and Security.)
   (c) Onwards to *Windows Firewall*.
   (d) Click on *Turn Windows Firewall on or off* in the left pane.
   (e) You can choose whether to turn Windows Firewall on or off for any of the three profiles available.
   (f) In *Windows Firewall* screen, on the left pane, *Restore Defaults* will reset the entire WF configuration to factory default.
2. **To Allow/disallow a program/service through WF:**
   (a) In *Windows Firewall*, click on *Allow an app or feature through Windows Firewall* in the left pane. In the resultant screen, click on *Change Settings*.
   (b) The list of service/program being displayed here are *Allowed*, which means, it is a white-list.
   (c) To allow a program/service, click on *Allow another app...* and browse to the location of program/service executable and select it. In this dialogue box, click on *Network Types* to select what profile is the program/service going to be allowed in. Finally, click on *Add*.

(d) To disallow a program/service through WF, select the specification in *Allowed apps and features* list and click on *Remove.*

3. **To Allow/block a port through WF:**

   (a) In *Windows Firewall*, click on *Advanced Settings* in the left pane.

   (b) In the resultant screen, decide if you want to allow/block the port for inbound connections or outbound connections from the system.

   (c) For both, the process is similar, just the location of creating rule varies (which can be selected from left pane). *New Rule* option is available in right pane, once either *inbound rules* or *outbound rules* are selected.

   (d) In the *New Rule Wizard*, in 'Rule Type' category, select *Port.* Next, select either TCP or UDP and specify the range of ports or a single port which is to be allowed/blocked. **NOTE:** Allowing all remote ports is NEVER recommended. Blocking can be done at user's discretion.

   (e) At this stage, the decision of either *allowing* or *blocking* or *allowing only if the connection is secure* decision can be made.

   (f) Select the profile for which this rule is to be applied.

   (g) Give the rule a name and a proper description (naming and describing helps keep track of the rationale behind creating the rule, to avoid confusion) and *Finish.*

4. **To Manage Rules:**

   (a) In *Windows Firewall*, click on *Advanced Settings* in the left pane. In this window, either *Inbound* rules or *Outbound* rules can be managed (copy-paste/cut-paste from one to another section or just simply delete.)

   (b) Creation of a new rule is the same as mentioned above. (We created rules for ports above. Rules for programs/services or predefined entities can be selected at selection screen as described in Step 4 of above seven steps.)

   (c) To export or import, in the *Windows Firewall with Advanced Security* window, click on *Windows Firewall with Advanced Security on Local Computer*. Importing and Exporting options are found on the right pane.

14 Walkthrough: Windows Firewall CLI <>

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

1. Enable/disable firewall.
2. Allow/disallow a program/service's network traffic using firewall.
3. Allow/block a port's network traffic using firewall.
4. Rule Management.

1. **To Enable/Disable Windows Firewall (WF):**

   (a) Go to search bar. Search for *cmd*. Right-click on *cmd* and select "Run as Administrator". (**Very Important**. Cannot work without Admin rights.)

   (b) At the *System32* prompt, type in `netsh advfirewall set allprofiles state on` to enable firewall for all three profiles.

   (c) At the *System32* prompt, type in `netsh advfirewall set allprofiles state off` to disable firewall for all three profiles.

   (d) Alternative values for `allprofiles` are: `currentprofile | domainprofile | privateprofile | publicprofile`

   (e) To reset the entire WF configuration to factory default, `netsh advfirewall reset` command can be used.

2. **To Allow/disallow a program/service/port through WF:**

   (a) At the *System32* prompt, type in `netsh advfirewall firewall add rule name="AllowProgramMyApp" dir=in action=allow program="C:PATH \ filename.extension" profile=any enable=yes` allow a program for all profiles.

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

   (c) For port, at the *System32* prompt, type in `netsh advfirewall firewall add rule name="AllowPort666" dir=in action=allow protocol=TCP localport=666 profile=any enable=yes` allow a program for all profiles.

   (d) More alternative values can be found out by: `netsh advfirewall firewall add rule ?`

   (e) To disallow a program/service/port, use `action=block`. When in direct conflict, bypass
   >block >allow.

   (f) Existing rules can be scanned using command `netsh advfirewall firewall show rule name=all profile=any`, but that is infeasible due to the huge data dump and only two filters.

3. **To Manage Rules:**

   (a) Using `netsh advfirewall` CLI shell commands, rules can only be added or deleted. Creation or addition of rules has been shown in the above section.

   (b) Deleting a rule can be done by the command: `netsh advfirewall firewall delete rule name=rule name program="C:PATH \ filename.extension"`

   (c) All rules can be deleted by: `netsh advfirewall firewall delete rule all`

   (d) Export - `netsh advfirewall export "PATH \ filename.wfw"`

   (e) Import - `netsh advfirewall import "PATH \ filename.wfw"`

15 Challenge I - Windows Firewall GUI <>

Using GUI: Block ICMP over all IP addresses.

1. Block both ICMPv4 and ICMPv6 protocol types.
2. Allocate blocking for all three profiles.
3. Block both incoming and outgoing communications.

Using CLI: Amongst the rules created above, disable rules belonging to both ICMPv4 and ICMPv6 protocol types, for incoming communications.

16 Challenge II - Windows Firewall CLI <>

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

1. For any/all protocol types.
2. Allocate blocking for all three profiles.
3. Block both incoming and outgoing communications.

And in addition, create an exception for an ip-address or an entire subnet of ip-addresses. For example, say, 192.168.1.100/24 subnet.

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 Firewalls in Linux <>

1. Linux firewalls are relatively command-line oriented.
2. For the most part, `gufw` and `IPtables` [previously `IPchains`] are good enough.
3. `gufw` (Uncomplicated FireWall (GUI)) is an easy-to-use front-end for `ufw`, which is an easier CLI version of `IPtables`.
4. `IPtables` is a relatively very advanced firewall with many integrated tools like user-space administration, packet filtering, protocol filters and network filter.

1. Most Linux firewalls have a highly advanced and flexible CLI command set.
2. Nonetheless, for most requirements and everyday configurations, Ubuntu has `gufw` [Uncomplicated FireWall (GUI)], which is not too advanced and not too basic. `gufw` is a front-end for `ufw` and is easy-to-use graphical user interface, which can be configured mostly through GUI, without even having the need to divulge into CLI commands. However, if needed and preferred, `IPtables` can be used directly, which is entirely CLI based, active in Linux kernel by default and for which `ufw` acts as an easier derivation.
3. Usually, `gufw` contains three profiles, one for each: Home Network, Public Network, and Office Network. The configurations and settings can be localized to each profile rather than one configuration for all three locales. Similarly, IPtables has something called “chains” which can be used to configure different threads of rules.

18 Activity: Uncomplicated FireWall (GUI) <>

1. Basic Functionality.
   - Set preferences, manage profiles, Enable/disable, and control traffic.
2. Rule Management.
   - Adding, editing and removing rules.
3. Dynamic Rules.
   - Creating/editing rules through dynamic port-listening reports.

Please refer to step 3 of the Hardware & Software Requirements slide to see the required VM setup for this part of the tutorial.

1. **Basic Functionality:**

   (a) Go to `Edit` in the upper left part of the window. Navigate to `Preferences`. In the pop-up box, change `Logging` to *Full*. Check `Logging Gufw activity` and `Show confirm dialog box for deleting rules`. Adjust the *Listening Report's* `Refresh Interval` to 1".

   (b) Coming to *Profiles*, there are three profiles by default: Office, Public, and Home. Any profile except the current one can be deleted. 255 profiles can be created. To create profiles, use the + symbol and to delete, use the − symbol. Renaming of newly created profiles can be done through double-clicking the name of profile. A maximum of 15 characters can be used to create a name for any profile.

   (c) Enabling and disabling Gufw is done based on per profile in Gufw. Select a profile and toggle the `Status` button `ON` and `OFF` to Enable/Disable the firewall.

   (d) Controlling global traffic can be done for per profile basis as well. Select a profile and use values in `Incoming` and `Outgoing` variables from the drop-down list to control the traffic. `Allow` option allows all the respective communications, `Deny` option drops all the concerned communication and `Reject` option drops all the respective communication and in addition, sends a message notifying the other side that their communication attempts have been rejected.

2. **Rule Management:**

   (a) Under the *Rules* section, + symbol can be used to create customized rules for firewall, instead of global control. – symbol can be used to delete any existing rule, which is shown in the rules registry. More than 356 rules can be created through Gufw. In *Policy* field, there is a new option available, called as `Limit`. The functionality of Limit is: system will deny connections if an IP address has attempted to initiate 6 or more connections in the last 30 seconds.

   (b) In the *Add a Firewall Rule* window, there are three tabs. First is the `Preconfigured` tab, which allows for easy, suggested, and effortless firewall configuration for a large number of applications. There are two filters, `Category` and `Subcategory` to help filter down the huge list of applications.

   (c) Second is the `Simple` tab, where in a firewall rule can be specified for any intended port or service, which is running on the system.

   (d) Third is the `Advanced` tab, which can be used to specify firewall rules at a completely custom level, with many options. The new options are: `Interface` tab, which gives an opportunity to design a rule for just one of the interfaces attached to a system. Logging flexibility, though it is always recommended to `Log All`. And more importantly, a custom range of ports with a custom range of IP addresses can be used to design highly specific firewall rules.

3. **Dynamic Rules**

   (a) A `Listening Report` is presented in the Gufw interface, just above the logs. The report displays the ports which are being used at the moment or which have been initiated, but not utilized. This feature is not very efficient in preventing attacks as the listening report generates information about active ports. But, at the same time, this feature is very helpful for thwarting continuous and repeated exploit attempts by attackers.

   (b) To add a firewall rule dynamically from `Listening Report`, just select on a port in the report to create a rule for and click on the + button. The rule will be initiated in the `Advanced` tab and most details will be filled in for you.

19 Activity: IPtables <>

1. Basic Functionality.
   - Enable/disable, Check Status, and Logging
2. Chain management.
   - Adding, editing and removing chains.
3. Rule management.
   - Adding, editing and removing rules.

1. **Basic Functionality:**

   (a) IPtables is enabled by default in Linux, at kernel level. To disable it in the classic sense, one needs a root permission. However, flushing out all the rules from IPtalbes chains effectively makes it disabled. `sudo iptables -F` is the command for flushing.

   (b) `sudo iptables -L` command gives the ability to check status as on what rules are in effect. Rules are segregated based on chains. Nonetheless, there exists a common pattern for rules status display; which is: `target`, `prot`, `opt`, `source`, `destination`, and `notes`.

   (c) Since it is not wise and efficient to log all traffic coming in and going out, for more optimal applicability, it is suggested to log only the packets which are dropped. A command for that would be `sudo iptables -I INPUT -j LOG --log-prefix "iptables denied:  " --log-level 7`, which saves logs to (/var/log/syslog). There are seven different log levels. They are:

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

2. **Chain Management:**

   (a) IPtables has three different chains by default. They are: *INPUT*, *FORWARD*, and *OUTPUT*. To create more chains, we can use the command `iptables -N chain-name`.

   (b) Sometimes there might be a situation where a chain has rules in it, but none of the rules match the traffic. In that case, a default policy can be alloted to any chain. The command `iptables -P chain-name ACCEPT || DROP`. The default policy can be to either accept or drop the packets if the traffic matches no rules.

   (c) To flush a chain of all rules, use the command `iptables -F chain-name`. To list the rules present in a chain, use the command `iptables -L chain-name`. To rename a chain, use the command `iptables -E old-chain-name new-chain-name` To delete a chain, the command `iptables -X chain-name` can be used. However, root privileges are required if anyone wants to delete one of the three default chains.

3. **Rule Management:**

   (a) To add rules in a chain, a simple way is to use the following command syntax: `sudo iptables -A chain-name -p tcp || udp -j ACCEPT || DROP || REJECT || LOG` . -A option specifies the instruction to "Append" a rule to the end of a chain. -p option specifies the instruction about which protocol we are targeting. -j option specifies the instruction about what to do (where to jump) when the rule is processed.

   (b) Instead of using the option -A to append the rule to end of a chain, we can use the option -I to insert a rule into some position of a chain. `sudo iptables -I chain-name 4` would insert the rule into chain and make it 4th rule in the chain list.

   (c) At some point, if we want to replace an already existing rule in the chain, we can use the option -R and specify the position of the rule at which we should carry out the replacement. `sudo iptables -R chain-name 4` would replace the existing rule at 4th place in the chain list, with new rule.

   (d) To delete a rule from any chain, the option -D can be utilized. 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. Else, if we do not specify the rule number, then the first rule matching the given options and values will be deleted. `sudo iptables -D chain-name 4` will delete the rule at 4th position. `sudo iptables -D chain-name -p udp -j ACCEPT` will delete the first rule which matches the criteria (protocol udp, should be accepting packets).

20 Challenge III - **gufw** <>

From “VM-gufw” machine, using `gufw`, perform the following tasks:

1. Allow Skype Telephony services.
2. Block and log all communications for everything related to steam and send a message to the person attempting communication, conveying the block.
3. Add a rule to block the `localhost` web server present in “VM-LAMP” machine. Test if your rule works and after it works, retract (delete) your rule.

21 Challenge IV - IPtables <>

From "VM-gufw" machine, using IPtables, perform the following tasks:

1. Create a new chain with the name "test". Rename it to "labtest". Then, delete the chain.
2. Write two rules so as to make the ping utility unresponsive. Test your rules and after they work, retract them.
3. Write two rules so as to make just the web server available in "VM-LAMP" machine inaccessible (both for HTTP and HTTPS). Test your rules and after they work, retract them.

HINTS:

2. Two rules are to be added, one should be for "INPUT" chain and another should be for "OUTPUT" chain.
3. Again, two rules are to be added, one should be for "INPUT" chain and another should be for "OUT" chain. Protocol can be mentioned using `-p`, destination and source can be mentioned using `-d` and `-s`.

22 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`.

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.

Details:

1. 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.
2. Windows Firewall provides notifications only when in critically unsecure state. As such, `Prohibit Notifications` should be disabled.
3. Dropped packets sometimes hold essential value in conducting forensic analysis. As such, it is always recommended to perform `logging of dropped packets`.
4. 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.

23 Conclusion <>

1. Firewall is relatively difficult to optimally maintain and configure than it is to install and setup.
2. There are many types of firewalls and many software packages are available to choose from.
3. Though various in number and diverse in appearance, the core functionality remains similar across every firewall.
4. Through hands-on walk-through activities and a few challenges, we have learnt how to perform basic management for firewalls in Windows and Linux.

1. 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 organization's needs.
2. However, there are many software packages of firewalls to choose from. Some of them offer easy to use graphic user interfaces: like `gufw`, some are very sophisticated to use like: `IPtables`, and some are flexible to accommodate both "ease of use" and "sophisticated" ideologies, leaving decision to users: like `Windows Firewall`.
3. Despite this variance in functionality on user's interface, the core functionality of every firewall remains the same: to control either incoming or outgoing traffic, or both. Thus, no matter what firewall is used/deployed in a system, the basic functionality of controlling communications will be at users' disposal.
4. Through the medium of this tutorials' walk-throughs, activities, and challenges, we have seen how Windows Firewall works (both with GUI and CLI). In addition, we have also observed the working and functionality of firewalls in Ubuntu, with `gufw` and IPtables respectively.

   Thus, this tutorial serves as a basic orientation course to firewalls in both Windows and Ubuntu, both through graphical and command line interfaces.

24 Appendix: Solutions and Change-log

<

1. Solutions to the challenges
   (a) Challenge I
   (b) Challenge II
   (c) Challenge III
   (d) Challenge IV
   (e) Bonus Challenge
2. Tutorial-Related Resources
3. Change-Log

1. **Solutions to the Challenges:**

   (a) **Challenge I:**

      i. In "Inbound Rules" section, click on "New Rule" at the right pane.
      ii. In resultant wizard, Custom Rule type. All Programs.
      iii. Protocol type: ICMPv4. Scope: Any IP address.
      iv. Action: Block the connection. Profile: All three.
      v. Give a RULENAME and description. Then, finish.
      vi. Repeat the same by changing Protocol type to ICMPv6.
      vii. 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.

   (b) **Challenge II:**

      i. In "Inbound Rules" section, click on "New Rule" at the right pane.
      ii. In resultant wizard, Custom Rule type.
      iii. Program: The program path is: `C-Windows-System32-mstsc.exe`.
      iv. Services: *Customize*. Apply to all services beginning with the name "Remote Desktop Services".
      v. Any protocol type – Any IP addresses – Block the connection-All Profiles

vi. Give a name and description. Then, finish.

vii. Repeat the same for “Outbound Rules”.

To create an exception for 192.168.1.100/24 subnet,

i. In “Inbound Rules” section, right click on the rule created in PART-1 of the challenge.

ii. Go to `General` tab. Change *Action* from `Block the connection` to *Allow the connection*.

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

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

v. Apply the modifications and `OK` to change the rule.

vi. Repeat the same for “Outbound Rules”.

(c) **Challenge III:**

In `gufw` program, use the option + to create a new rule in the “Rules” section. In “Preconfigured” tab:

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.

Use the option + to create a new rule in the “Rules” section. In “Advanced” tab:

i. Policy: “Deny”.

ii. Direction: Both. Interface: All Interfaces.

iii. Log all. Protocol: Both.

iv. To: <IP address of “VM-LAMP” machine>; Port: 80. Add.

(d) **Challenge IV:**

i. `sudo iptables -N test`,
`sudo iptables -E test labtest`, and
`sudo iptables -X labtest`

ii. `sudo iptables -A OUTPUT -s <YOUR_MACHINE_IP_ADDRESS> -j DROP`, and
`sudo iptables -A INPUT -d <YOUR_MACHINE_IP_ADDRESS> -j DROP`

iii. First command:

```
sudo iptables -A OUTPUT -d <LAMP_MACHINE_IP_ADDRESS> -p tcp
--destination-port 80 -j DROP,
```

Second command:

```
sudo iptables -A OUTPUT -d <LAMP_MACHINE_IP_ADDRESS> -p tcp
--destination-port 443 -j DROP,
```

Third command:

```
sudo iptables -A INPUT -s <LAMP_MACHINE_IP_ADDRESS> -p tcp
--destination-port 80 -j DROP, and
```

Fourth command:

```
sudo iptables -A INPUT -s <LAMP_MACHINE_IP_ADDRESS> -p tcp
--destination-port 443 -j DROP
```

(e) **Bonus Challenge:**

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

2. **Tutorial-Related Resources:**

   A free virtualization software platform - VirtualBox - CLICK ME.

   Microsoft Windows Server 2012 R2 (Free 180-day evaluation copy) - CLICK ME

   Ubuntu 16.04 LTS - Canonical Ltd. - CLICK ME

   Graphical Uncomplicated FireWall [GUFW] - Gufw Project - CLICK ME

   Guide to Installing LAMP on Ubuntu 16.04 LTS - DigitalOcean - CLICK ME

3. **Change-Log:**

Firewall Management tutorial			
Ver.	**Date**	**Authors**	**Changes**
v1	Apr. 23rd 2016	Ananth Jillepalli	First draft of tutorial.
v2	July 13th 2016	Ananth Jillepalli	Major content additions and remodeled the structure.
v2.1	August 15, 2016	Ananth Jillepalli	Added the 'Appendix' section and other minor enhancements.

References

[1] Bojan Zdrnja, "Windows Firewall Bypass Vulnerability using NetBIOS Name Spoofing", last accessed 16th March 2016, https://isc.sans.edu/diary/Windows+Firewall+Bypass+Vulnerability+and+NetBIOS+NS/13156, 8th May 2012.

[2] CVE Details, "Iptables Security Vulnerabilities", last accessed 16th March 2016, https://www.cvedetails.com/vulnerability-list/vendor_id-959/product_id-1656/Netfilter-Core-Team-Iptables.html, 18th February 2014.

[3] Lucian Constantin, "Juniper's VPN backdoor: Buggy Code with a Dose of Shady NSA Crypto", last accessed 16th March 2016, http://www.pcworld.com/article/3017803/security/the-juniper-vpn-backdoor-buggy-code-with-a-dose-of-shady-nsa-crypto.html, 22nd December 2015.

[4] Lucian Constantin, "Critical VPN Key Exchange Flaw Exposes Cisco Security Appliances to Remote Hacking", last accessed 16th March 2016, http://www.pcworld.com/article/3032497/critical-vpn-key-exchange-flaw-exposes-cisco-security-appliances-to-remote-hacking.html, 11th February 2016.

[5] Ramesh Natarajan, "25 Most Frequently Linux IPtables Rules Examples", last accessed 29th March 2016, http://www.thegeekstuff.com/2011/06/iptables-rules-examples/, 14th June 2011.

[6] Ubunutu Community Wiki, "Gufw", last accessed 29th March 2016, https://help.ubuntu.com/community/Gufw, 27th July 2014.

[7] Ubuntu Community Wiki, "IPtables", last accessed 29th March 2016, https://help.ubuntu.com/community/IptablesHowTo?action=show&redirect=Iptables, 19th August 2015.

[8] Microsoft Support, "How to Use The *Netsh Advfirewall Firewall* Context", last accessed 29th March 2016, https://support.microsoft.com/en-us/kb/947709, 22nd April 2012.