International Journal of Scientific Engineering and Research (IJSER)
ISSN (Online): 2347-3878
Index Copernicus Value (2015): 62.86 | Impact Factor (2015): 3.791

Detection and Correction of Wireless Vulnerabilities using Firewall Algorithms

Ankit Soni, Sumit Roy

B.Tech student, Department of Computer Science, SRM University, India

Abstract: ***With recent technological advancements, every electronic device is wirelessly connected to the internet and the internet as we all know is a network of systems and devices connected together. As the saying goes "A chain is only as strong as its weakest link", a network is also as strong as its weakest node. People today are making innumerable efforts to their systems secure from various threats; however the vulnerabilities arise not just in our systems, but also in the devices with which we are connected to the internet. Thus no matter how secure out systems is, if our routers are not well secured our systems are always vulnerable to attacks and our personal files and data is always at risk. A person with malicious intent would not even require to gain access to your system or attack your system to know what data you have stored in your system, all he needs to do is to tap into the router you are connected to and exploit its vulnerabilities to gain access to every data packet that is being sent by your system or received by it, and you wouldn't even know. The aim of this project is to detect and identify such vulnerabilities and to prevent such attacks from happening by trying to understand how a router works, how a firewall works and the possibility of integrating both of them to stop an attacker from gaining access to the router or any wireless connecting device.***

Keywords: Wireless, Vulnerabilities, Firewall, Router

1. Router and Its Working

Router basically is a focused networking machine linked to multiple networks running software that allows the router to shift information between networks. A Router in an IP based network operates at the network layer. Basic purpose of a router is to provide interconnectivity between networks. It is an intra network connecting device whose sole task is to forward packet through the network for minimum cost. The minimum cost path which a router uses to forward packet is called the Optimal Path. There are many types of packets size used inside a router like IPv4, IPv6, etc. These are called Routed Protocol whose major aim is to find minimum cost path within the network.

The following are the basic working steps of a router:

1. Router powers on and initializes it's OS from the device's firmware.
2. The router initiates the setup file lastly saved to Non Volatile Random Access Memory and creates the routing protocols and network interface it'll run.
3. The network address and subnet for all the interfaces is added to the routers routing table..
4. Router consists of a basic constant default route so as to transmit each non-local data packet away from the network port which is connected to the ISP.
5. On receiving a web page request from a computer, a router checks the receiver Internet Protocol address beside its routing table.

2. A Brief Insight about Firewalls

A firewall are often either a computer code or hardware based mostly network security system. It uses rules to regulate, temperate the network traffic exchange. Firewall can be considered as a barricade between user and receiver network. Firewall uses a positive management model to regulate the traffic which implies solely the traffic matching the factors and following the predefined rules within the firewall policy is going to be allowed within the network, and every single different traffic is denied access. Before the idea of firewall the sole possible way of dominant traffic was by victimization Access management lists that were embedded within the routers.

2.1 Hardware Firewall

Hardware Firewalls measure are typically used in routers. It's the first defense mechanism using Packet Filtering. prior to the reaching of a web packet to a computer, the Firewall can scan packets while checking its supply. It checks whether or not header and IP address square measure are trust worthy. When checking, the packet reaches the pc. It blocks all links that contain malicious things supported by this Firewall setup in device. Hardware Firewall typically doesn't want lots of configuration. Rules square measure constitutional and predefined and support these inherent rules, Packet Filtering is finished.

With recent technological advancements it's not simply the standard Packet Filtering that is administered. Hardware Firewall has inherent Intrusion interference Systems, which previously was a completely different device. However currently they're enclosed. The biggest disadvantage is, it permits all outgoing traffic that is if malware enters the system and commenced sending information; it might be permitted till the end-user stops it, and set to prevent it.

2.2 Software Firewall

As the name suggests a software package firewall may be a piece of software package that forestalls your system from unauthorized access and attacks by cyber criminals. A firewall is often categorized into varied sorts consistent with their principle of operating.

Licensed Under Creative Commons Attribution CC BY

3. Vulnerabilities in Systems and Wireless Network Devices

3.1 The Problem Statement

With each passing day we are putting in more efforts to secure our computer systems than what we put in to secure out homes. The reason for this drastic change in the society is because every day more and more sensitive data is moving online from being offline. From our phone numbers to our personal bank information, everything is available online for the taking. Our money, our privacy and our entire life is electronically stored. This might even seem very trivial when we come to more sensitive things which are digitally and electronically stored. A country's defense system, nuclear launch systems, the data of all the bank customers and what not. This is the exact reason why we are putting in more and more efforts to secure our digital presence and data from unwanted people with malicious intents.

Now let us take a hypothetical example to better relate our problem and understand it. Consider a heavily fortified and secure house, the owner of the house is concerned for his safety and thus has put extensive safety measures to secure his house. However, he has to go out to work every day and the road right outside his house is full of notorious people waiting to rob him. And the road outside his house has absolutely no security. Now on the same lines let us consider the situation of our modern day network and its securities. We have been putting in extensive efforts to secure our systems from external attacks and to keep our data safe and secure, however what we all forget to realize is that no matter how secure and fortified our systems maybe, we are still connected to the same old insecure router leaving us vulnerable and open for attacks. As a matter of fact this is an even better situation for any attacker to access your personal information, he doesn't even have to bear the pain of bypassing your extensive security measures and hacking into your system, all he needs to do is just gain access to the router you are connected to, and he can gain access to all the information and data that moves in or out of your system, which might even include your personal banking information, your sensitive emails etc. The entire point of the above hypothetical situation was to only establish the common proverb "A chain is only as strong as its weakest link" and here the weakest link in a network is the router.

3.2 The Proposed Solution

The above mentioned problem has some very serious implications, but not enough attention or care has been given to this particular issue. As a matter of fact, there is very little research that has been done on this topic and none in depth. However we here propose a viable solution that could counter this problem with ease and efficiency and without much hardware or software cost.

Our proposed solution carries the idea of implementing a basic packet filtering firewall in a router's firmware, transforming the router from a vulnerability to be exploited to the first line of defense in securing our systems. In our proposed solution we have tried to develop algorithms for the various modules that could be implemented in a router to establish a firewall within it. Although a router has certain safety and security measure but they are puny and insignificant for any willing hacker and are not enough to stop him/her. However, a firewall is a dedicated towards just one task, to stop malicious programs or people from gaining access to your system and steal your private data. Firewalls use dedicated security algorithms to keep the unwanted programs or people at bay. For instance, Cisco has recent started working on the same concept, and has developed a proprietary algorithm called the Adaptive Security Algorithm, more commonly known as the ASA. This algorithm however is not a single algorithm, but a combination and collection of multiple security algorithms with each algorithm performing dedicated task. The Adaptive security algorithm uses and implements the stateful firewall inspection technique to secure the router.

Now, however easy this entire concept sounds, it actually is not. The first major issue that we might face while trying to implement a firewall in a router is not having enough computational strength and thus reducing the efficiency of the router. However the entire performance can be exponentially enhanced by taking a router with slightly better configurations. This entire process will not only help secure our systems but revolutionize a router by reducing its hardware costs by over 60%.

4. Methodology of the Problem Solution

The first and foremost challenge that we will face while trying to solve the above stated problem is to create the logic and algorithm for the successful capture of the data packets which are being transmitted in the network traffic and filtering them on the basis of the rules set defined. The approach that we are taking here is we will try to capture data packets by packet sniffing methods and then try to filter them on the basis of their destination port.

The common protocols used in the network layers are internet protocols, Internet control message protocols, Transmission control Protocols, Internet group management protocols.

Capturing packets is the process of gathering information that is shared on top of the network. Every time a network card gets an Ethernet frame, it verifies whether the MAC address of the destination matches that of its own. If the result is positive, it generates an interrupt request. The network card's driver is the routine that handles this entire work.

4.1 Basic firewall working

The firewall will "BLOCK or UNBLOCK" packets consistent with a group of rules. The principles area unit set by a user house configuration computer program. as an example,

"./mf –in –scrip ten.0.2.15 –srcnetmask 255.255.0.0" –destport eighty –proto TCP –action BLOCK

Licensed Under Creative Commons Attribution CC BY

4.2 Command Line Arguments parsing in glibc

Many UNIX system programs are command primarily based and typically the choices are sophisticated. Luckily, the GNU C library glibc provides some APIs to change the command line choice parsing.

Specifically, there're 2 strategies for parsing the commands, getopt and getopt_long. getopt() is employed to parse the one character choice, and getopt_long() works with each long choices and single-character choices. It's suggested to use "getopt_long()".

The "getopt_long" has the following prototype:

"int getopt_long (int argc, char *const *argv, const char *shortopts, const struct option *longopts, int *indexptr)"

argc and argv ar the argument count and argument vector, an equivalent similat to the two input parameters within the customary main model.

shortopts could be a string specifying the choice characters that ar valid as input choices. Associate in nursing choice character may be followed by a colon (':') to point it takes a needed argument. The character may be followed by 2 colons ('::') to point the argument is optional. If nothing follows the choice character, the choice doesn't take any arguments.

longopts describes the long choices to simply accept, that is Associate in Nursing array of struct choice. longopts should be terminated by a struct choice of all 0s.

Process to get opt_long match the input options

For short possibility, getopt_long returns the character code for the choice, and stores the option's argument (if it's one) in optarg.

For long possibility, getopt_long takes action supported flag and val fields of the choice matched.

If the flag is NULL, which means the choice isn't a flag. getopt_long returns val. unremarkably you'll use the short possibility's character code in val if long possibility is resembling the short option.

If the flag isn't NULL, which means it's a flag possibility. getopt_long can come back zero, and therefore the flag worth are set consequently.

For all choices, getopt_long stores the matched option's index in array longopts to *indexptr. you'll access the name of the choice by longopts[*indexptr].name.

getopt_long will place the argument worth in optarg if the choice has one. Otherwise, optarg is ready to NULL.

When getopt_long has no a lot of choices to handle, it returns –1. The index of next remaining argument in argv is keep in variable optind.

4.3 Filtering network options using Net filter

Net filter may be considered a set of hooks within LINUX kernel. It permits kernel modules to register request functions with the network stack so as to intercept and manipulate the network packet. When a network packet comes in, it's passed to the netfilter's 1st hook NF_INET_PRE_ROUTING. After that, the packet goes through the routing code, which decides wherever the packet is destined to, either another port in same network interface or another interface. It additionally would possibly drop the packet if it's unroutable.

4.4 Prototype of the Hook Function

Once a hook function has registered with any of the five netfilter hooks, it' will be called once a network packet goes through the hook within the network stack.

"unsigned int hook_func_in(unsigned int hooknum,
struct sk_buff *skb,
const struct net_device *in,
const struct net_device *out,
int (*okfn)(struct sk_buff *))

hooknum indicates one of the five main hook types (NF_INET_PRE_ROUTING, NF_INET_LOCAL_IN, NF_INET_FORWARD, NF_INET_LOCAL_OUT, NF_INET_POST_ROUTING)"

sbk is a pointer to the network packet buffer, which is defined in

"/lib/modules/$(uname -r)/build/include/linux/skbuff.h."

The detail about this packet buffer is covered in next section.

In and out are two pointers to the net_device structure, which are what Linux kernel uses to describe network interface, as defined in "/lib/modules/$(uname -r)/build/include/linux/netdevice.h." In the hook function, describes the network interface the packet passes through. Therefore, depending on the packets traversal, either in or out will be NULL.

okfn is a function pointer enables registering of a callback function triggered when all the functions registered with this hook returned NF_ACCEPT, thus "oking" the packets. When a registered function is called, it can do one of the five things and return the corresponding value, as defined in netfilter.h

NF_ACCEPT: let the packet pass.
NF_DROP: drop the packet.
NF_STOLEN: take the packet and don't let the packet pass.
NF_QUEUE: queue the packet, usually for userspace handling.
NF_REPEAT: call the hook again.

Licensed Under Creative Commons Attribution CC BY

5. Conclusion

With the above discussion and conceptualization, a new device can be build on the framework or the existing conventional routers, or we could simply embed the above discussed codes into the firmware of the router. With this, there might be a slight lag in the device performance; however that issue can be sorted by optimizing the code before embedding it into the router for better performance. This can be a revolutionizing concept in both the domestic and the commercial industry.

References

[1] Rosse, D' etal. Securely Deploying IEEE 802.11 WLANs. AusCRT Conference 2007
[2] Scarfon and Decoi, D. Wireless Network Security for IEEE802.11 b/g and Bluetooth. NIST Publication 801-57 Revision 1
[3] IEEE Standard 802.11, 1998 Edition. Also available at http://standard.ieee.org/getieee802.11/download/802.11-1998.pdf
[4] Security Threat Mitigation and Response - CS - MAS, Dalle Tesh/Grage Abelear, Cisco Press, June 28,2007
[5] Jonathen Postl. Transmission control protocol. RFC 974 (Standard), July1991. URL http://www.ietf.org/rfc/rfc0974.txt

Author Profile

Ankit Soni, Phone- +91-7417440464, Address- D1/3 SBI Colony, Char Imli, Bhopal 462016

Sumit Roy, Phone-+91-9811845348

Licensed Under Creative Commons Attribution CC BY