Packet Sniffing: What it's Used for, its Vulnerabilities, and How to Uncover Sniffers

Mathurshan Vimalesvaran
Tufts University

Abstract

Packets are the base of all data sent on the internet, yet they are often used insecurely. Tampering with live packets and the process it takes in order to alter packets traveling along the network are getting easier. Current exploits that attackers use are easily attainable by even novice attackers. There is a range of different packet scanning techniques used for sniffing. The purpose of this paper is to explain the nature and of packets and expose the vulnerabilities that attackers exploit. This paper also covers practices set in place to protect against packet manipulation and sniffing, and detection of sniffers on a network.

I. Introduction

Packet sniffing is commonly described as monitoring packets as they go across a network. Packet sniffers are typically software based, but can be hardware pieces installed directly along the network. Sniffers can go beyond network hosts that are seen in local area networks (LAN) that only handle data that is sent specifically to them.

Most sniffers are difficult to detect as they work passively and just collect data. Some, however, can be active and are, therefore, more likely to be detected.

Sniffers can be used lawfully within a network by system administrators in order to monitor and troubleshoot the traffic on their own network. For example, if a computer is having problems communication with another computer, an administrator can view the packet from one machine to the other machine and determine the cause of the issue.

The security risk appears when an adversary uses a sniffing tool to collect plaintext sensitive material such as passwords. Sniffers are, for most organizations, an internal threat [7].

II. To the Community

I chose to write on this topic, because packets are the underlying mechanism in how machines communicate and therefore it is important to understand how they work and the vulnerabilities related to them.

There was a lot of work done in securing data sent over the wire and there is a lot to learn in this area. It is vital to know how our data is being handled and interpreted in order to be more conscious of what could go wrong. There are a lot of people and businesses still utilizing unsecure methods of sending information and this could lead to a lot of leaked personal data.

III. Findings

Overview of packets

Machines connected to a network send data to other machines in a format called packets. Packets are essentially the pieces of the data that needs to be transferred.

When a person moves to a new home, they must bring their belongings from the old home to the new home. Packets can be thought of as the moving trucks that carry the items. Once a truck gets to the destination, the items can be taken out and used accordingly. Just like some people may need more than 1 truck, some data may need more than 1 packet. Larger data is broken up into multiple packets and the receiver waits until all packets arrive and then puts the data together.

Any time a computer tries to send data to another computer, it creates packets of the information. Every website visited, email sent, and instant message sent utilizes packets as its most basic transport.

In order for machines to understand a packet, there are set protocols on their format. There are usually 3 parts to each packet: the header, the payload, and the trailer/footer [10]. Each packet is set up so that it has its original (sender) address, destination address, and data regarding the packet itself within its header. The payload portion is the actual data being sent. This can be text for an email or even html for a website. The trailer is usually an end of packet signal that also contains an error checking method that allows the receiving machine to verify the packet is valid. This error check is not used to check against malicious packets, but rather packets that have had issues along the network and likely have incorrect data.

This protocol is how packets sent from one machine can reach any location within the network and receiving machines know how to use the data.

Unswitched Network

Since all machines on an unswitched LAN are attached to the same hub, packets on an unswitched network are sent to all machines on that local network. When packets are received by the hub, the packets are broadcasted to the entire LAN.

However, machines won't pay attention to a packet if it is not specifically sent to them. There is a way to change this. By altering the state of the machine into "promiscuous mode", the Network interface controller (NIC) in the machine can look into packets that pass through it even if they are destined for another machine. This is the easiest way to sniff a wire. Switched networks, different from unswitched hubs, became faster and cheaper over time. So most unswitched networks have since been converted to a switched network [4].

Switched Network

Switched networks are different in that each machine is attached to a switch instead of the entire hub. The switch keeps track of the machines using information such as the unique MAC address of each machine. When packets arrive that are meant for a specific MAC address, the switch sends it over the wire to the specific machine. If a packet arrives for a machine that the switch does not have saved, it will

send out an ARP request to get the names of the machines in the network. Then, when a match is found, it will cache that info, and send that machine the packet. [1] This enables traffic to be sent directly to the destination. Sniffers, therefore, cannot gain intermediate access to the traffic by sniffing the wire.

Packet Sniffing

Packet sniffing can be described in 3 steps:

1. "Packet sniffer collects raw binary data from the wire. Typically, this is done by switching the selected network interface into promiscuous mode

2. Captured binary data is converted into a readable form.

3. Analysis of the captured and converted data. The packet sniffer takes the captured network data, verifies its protocol based on the information extracted, and begins its analysis of that protocol's specific features." [2]

There are two main types of sniffers: commercial and underground applications. Commercial applications are often used legally to monitor network traffic. They can help sense and locate bottlenecks and various low level network issues. Underground applications have a more negative stigma that comes from attackers using these types of sniffers for malicious purposes. Attackers will often illegally sniff networks to gain access to private information.

Switched networks are, unfortunately, still susceptible to sniffing attacks. There are 3 main types of switched network attacks:

ARP Cache Poisoning:

[1] ARP poisoning can most easily be described as an adversary, A, pretending to be a different machine, D, so the switch sends the traffic for D to A. Then after getting the traffic for machine D, adversary A reads and/or makes a copy of that traffic, and then forwards the traffic to the correct machine D. This way, machine D is not aware of any loss of data and the switch believes it's doing its job while the adversary is able to stealthily collect packets.

ARP cache poisoning attacks can be implemented quite easily with a few tools, namely arpspoof (from dsniff suite), wireshark, and scapy. This paper will not go into the details of any mentioned tools and other tools exist not mentioned exist that can be used instead of the listed, but the steps following will mention these 3 tools.

First, make sure you're connected to the network and obtain the IP address of the machine to sniff. This can be done by using arp:

```
# arp -a
```

This will list the IPs on the network. Logging into administrator settings on the router will also provide this information.

The next step is to pass along the traffic so the victim does not experience an interruption in internet service:

```
echo 1 > /proc/sys/net/ipv4/ip_forward
```

We then redirect traffic sent from the router to the attacker's machine to your machine. For example, assume gateway is 192.168.1.1 and victim is 192.168.1.10. This can be done easily with arpspoof:

```
# arpspoof –i eth0 –t 192.168.1.1 192.168.1.10
```

This tells the router that you are the victim's machine so the router will send you data that is meant for the victim.

Now, in order to get all the traffic from the machine, we run another arpspoof command to redirect traffic sent from the victim to the router:

```
# arpspoof –i eth0 –t 192.168.1.10 192.168.1.1
```

This tells the victim that you are the router so the victim sends their requests to your machine.

The final step is to utilize a sniffer to view the packets sent to and from the victim. This can be done in two ways:

```
# wireshark
```

Then filter the results by using:

```
ip.addr == <ip address of victim>
```

OR

```
# scapy
# pkts = sniff(filter="host <ip address of victim>",
        count=<num of pkts to sniff>)
```

View the overview of sniffed contents by running:

```
# pkts.summary()
```

This type of attack is possible due to the fact that the ARP is stateless and does not retain any information about the specific machine during each request so it doesn't recognize that two different machines are saying they are the same machine.

CAM Table/MAC Address Flooding:

[1] "CAM table is a table that stores information like MAC addresses and switch port along with their Virtual LAN information." By flooding the switch with MAC addresses, the switch will be overwhelmed and start to broadcast network traffic like an unswitched network. This opens it up for easy sniffing.

Dsniff's macof tool can be used to accomplish this using the simple command:

```
# macof
```

At this point, sniffing can occur using the previously mentioned promiscuous mode.

When attacked, not all switches will behave by reverting to an unswitched network. Many can have protection against this type of situation.

Switch Port Stealing:

[1] Port stealing allows a sniffer to gain access to the traffic destined for a specific machine. The sniffer essentially sits between the destination machine and switch and is able to view the packets as they go from the switch to the destination machine.

The attacker floods the switch with ARP packets that have the source MAC address as that of the target host and the destination MAC address as that of the attacker. This is different than the flooding process used in CAM Table Flooding. “Since the destination MAC address of each flooding packet is the attackers MAC address, the switch will not forward these packets to other ports, meaning they will not be seen by other hosts on the network.” [13] Since the target is sending genuine packets, the attack must be fast enough so that the target does not overwrite the attacker’s table entry.

This attack can be done using Ettercap either through its GUI or through its CLI:

ettercap –G

OR

ettercap –T –M port

Packet Injection and Spoofing

Although not directly related to sniffing, another issue worth mentioning is packet spoofing. When packets are sent, they have the IP address of the host, but this address is not verified by the IP protocol [6]. This allows adversaries to alter packets and send them to users. An attacker can then pretend to be someone else, hide their true location, or hijack network traffic [6]. Sniffers can utilize this vulnerability to alter content in packets such as links and images. So not only can sniffers gain access to your data, they can alter what you see and do. This is why it’s important to defend against these types of attacks.

IV. Defenses

One obvious hardware fix is to change to a switched network which has a private line from a computer to the switch so no machine will have access to another’s packets while it’s sent over the wire. Although this isn’t a perfect solution, it is better than using an obsolete unswitched network.

Users can protect themselves from sniffers in a variety of ways. The best defense to packet sniffing and alteration is, first and foremost, interacting with reliable sites. A lot of sites do not use modern encryption methods to protect their users’ sensitive data. Most sniffers will have a hard time getting significant data if it’s encrypted.

Encryption is also a great way to protect data. Adversaries can still get access to the packets, but they will only see nonsensical data and will have difficult time deciphering it.

Safe Guards

Link layer encryption and end-to-end encryption are good precautions to take against sniffers. Link layer encryption occurs from node to node along the network, where sniffers would typically be listening. End to end encryption occurs at the end hosts where the sending host encrypts the packets and the receiving host decrypts the packets.

Best practice application level encryptions are SSL and TLS that keep data secure at the user interface level of applications. These are very commonly used and sites that do not utilize these precautions should be

avoided. Sites that are using some sort of encryption will have “HTTPS” instead of “HTTP” and most modern browsers will display a green padlock next to the site URL if your data is safe.

A strong level of protection when using public internet is a virtual private network (VPN). [5] This allows a user to send all traffic to a protected server that will encrypt all data and handle a user’s requests. A correctly set up VPN gives a user strong protection and provides security that is not guaranteed in a public internet.

Sniffer Detection

There are a couple of methods used to determine if a sniffer is being used on the network.

One is to generate packets that do not have valid addresses and send them out. Since sniffers usually accept all packets, if a machine is to accept that packet, it is using a sniffer. [3]

Commercial software, such as AntiSniff, also exists to help non-intrusively detect sniffers. [3] These are often utilized by system administrators to protect against attackers.

V. Conclusion

Data is sent across the internet in the form of packets. Packet sniffing can be used for the benefit of a network or for malicious purposes. It can monitor and analyze traffic and help with network research. It can also be used by adversaries in order to steal plaintext data or watch a user’s actions.

Software exists to help detect sniffers on a network. Business systems often set these in place in order to keep data safe. Without using modern defenses and best practices, data sent across the network can be easily seen by attackers. It’s important to verify that sites you access are utilizing the safe guards available, namely encryption, and avoid the sites that are not.

VI. References

[1] Rupam, Atul Verma, Dr, and Ankita Singh. "An Approach to Detect Packets Using Packet Sniffing." *International Journal of Computer Science & Engineering Survey* (2013): n. pag. Web.

[2] Asrodia, Pallavi, and Hemlata Patel. "Analysis of Various Packet Sniffing Tools for Network Monitoring and Analysis." *International Journal of Electrical, Electronics and Computer Engineering* (2012): n. pag. Web.

[3] Ansari, S., S.g. Rajeev, and H.s. Chandrashekar. "Packet Sniffing: A Brief Introduction." *IEEE Potentials* 21.5 (2002): 17-19. Web.

[4] Kishore, Aseem. "Router vs Switch vs Hub." *Help Desk Geek*. N.p., 9 Nov. 2009. Web.

[5] "4 Ways to Avoid Packet Sniffing and Data Theft." *Hacker Not Cracker*. N.p., n.d. Web. Dec. 2015.

[6] Templeton, S.j., and K.e. Levitt. "Detecting Spoofed Packets." *Proceedings DARPA Information Survivability Conference and Exposition* (2003): n. pag. Web.

[7] King, Tom. "Packet Sniffing In a Switched Environment." *SANS Institute*

InfoSec Reading Room (2006): n. pag. Web.

[8] Mzrak, A.t., S. Savage, and K. Marzullo. "Detecting Malicious Packet Losses." *IEEE Trans. Parallel Distrib. Syst. IEEE Transactions on Parallel and Distributed Systems* 20.2 (2009): 191-206. Web.

[9] Chomsiri, Thawatchai. "Sniffing Packets on LAN without ARP Spoofing." *2008 Third International Conference on Convergence and Hybrid Information Technology* (2008): n. pag. Web.

[10] "What Is a Packet?" *HowStuffWorks*. N.p., 30 Nov. 2000. Web. 28 Dec. 2015.

[11] "A Quick Intro to Sniffers: Wireshark/Ethereal, ARPSpoof, Ettercap, ARP Poisoning and Other Niceties." *A Quick Intro to Sniffers: Wireshark/Ethereal, ARPSpoof, Ettercap, ARP Poisoning and Other Niceties.* IronGeek, 1 Feb. 2005. Web. 28 Dec. 2015.

[12] Sankar, Ravi. "MAC Flooding with MACOF & Some Major Countermeasures." *Kali Linux Tutorials*. Kali Linux Tutorials, 22 Sept. 2015. Web. 2 Jan. 2016.

[13] Spangler, Ryan. *Packet Sniffing on Layer 2 Switched Local Area Networks* (2003): n. pag. *Http://www.packetwatch.net/*. Web. 2 Jan. 2016.

[14] "Layer 2 Security Features on Cisco Catalyst Layer 3 Fixed Configuration Switches Configuration Example." *Cisco*. Cisco, 17 Jan. 2007. Web. 3 Jan. 2016.