Utilizing Switch Port Link State to Detect Rogue Switches

Travis Quitiquit and. Vijay Bhuse
Grand Valley State University, Allendale, MI, USA
quitiquit@mail.gvsu.edu
bhusevij@gvsu.edu

Abstract: There are many methods to detect rogue wireless access points, but the same case can not be said for rouge switches on a LAN network. Detecting these rogue switches is key to security of any organization. The introduction of a rogue unmanaged network switch has the potential to cripple a network. These types of switches pose a big risk because they are usually plug and play types of devices and can prove difficult to track. A switch becomes rouge when it is connected to a network without proper authorization. Rogue switches are a huge threat to the security and reliability of any network. An attacker could use a rogue switch to launch an attack or spy on network traffic information. Many organizations these days implement a “bring your own device” policy that can prove to be a daunting task to monitor for any network administrator. It is important that these rogue network switches are not introduced to a network, whether by accident or in a malicious attempt. The vulnerability that is introduced could comprise the confidentiality of network messages, degrade network performance, or even allow hackers or authorized users access to critical network infrastructure and data. In this paper we present a method that can help detect these rogue switches by monitoring the ethernet frames across the network and looking at the link state of the network switch ports. We will be using Wireshark, a Windows Computer, and a local switch setup to test methods for detecting a rogue switch. In our scenario we were able to provide some evidence of a method that could be used in conjunction with other rules and policies to detect rogue switches connected to a network. We were able to determine based on the port link state that there was another device, most likely a rogue switch between the good switch and the computer.

Keywords: rogue switch, link state, port monitoring, network security, LAN, network detection

1. Introduction

There are two main types of switches that can be introduced to a network, managed and unmanaged. A managed switch allows for greater control over the settings of the switch and can prioritize network traffic. These are typically not the rouge switches you think about since managed switches take more effort to configure to work on a network. On the other hand, the unmanaged switches are what pose the biggest risk to an organization. These are essentially plug and play devices with no settings to configure (Services, 2021). These unmanaged switches can pose a large risk to any network because they do not have an IP address, and most do not have MAC addresses for device tracing.

Using managed switches in your organization, you should be able to monitor all traffic on the network for MAC addresses among other things. You can then enforce rules and policies such as only one MAC address can be routed to each port on the managed switch. If a rogue switch was used to plug into an existing port, you would notice this because there would be more than one MAC addresses routed to a given port.

A switch becomes rouge when it is connected to a network without proper authorization. Rogue switches are a huge threat to the security and reliability of any network. An attacker could use a rogue switch to launch a cyber-attack or spy on network traffic information. Many organizations these days implement a bring your own device policy that can prove to be a daunting task to monitor for any network administrator. It is important that these rogue network switches are not introduced to a network, whether by accident or in a malicious attempt. The vulnerability that is introduced could comprise the confidentiality of network messages, degrade network performance, or even allow hackers or authorized users access to critical network infrastructure and data (Bhuse, 2019).

We live in an IP connected world and internet protocol has become the standard for all connected technology. The dilemma today is the greater connectivity the greater challenges if you’re an IT or security manager. Network growth and complexity is skyrocketing. Before the explosion of connected devices, networks usually grew with the number of connected users. These days networks must scale three to five times for each employee based on the number of IP devices we use to do our work. This means the challenge of managing and securing endpoints has also become much more difficult (Solarwinds, 2019).
When you consider all the connected devices on a network: PCs and laptops, VOIP phones and BYOD devices, tablets, smartphones. There is also another type of device lurking out there and those are rogue devices that have the potential to cause harm to your network. The larger the network, the greater the potential risk. All it takes is one rogue device to inflict chaos on a network and organization (Solarwinds, 2019).

The goal of security is to implement security controls at the network layer. The network layer represents the logical area of vulnerability and point of intrusion. Some goals of network security at this level should include identifying authorized devices, building network access controls, and monitoring network connected devices. The network switches represent the best touch point for implementing many of these goals, because network switches are the part of infrastructure which is closest to network cabling. Another good fact is that switches operate at OSI layers 1 and 2, which can detect activity on your network which is not detectable at higher levels (3-7) (Solarwinds, 2019).

Using some methods of security that are already in place today (Bhuse, 2019) along with the test methods of this paper we came up with an experiment to see if detecting rogue switches utilizing port link state is beneficial. Some setup and assumptions of the experiment are that the good switches are managed switches with a rule that each port can only have one MAC address. It is also assumed that the rogue switch is plugged into an enterprise network where we have access to the good network switches and to a computer that is connected to the rogue switch that we have access to get logs from.

2. Related work

The job of detecting a rogue network switch can be a difficult task because most unmanaged switches do not have IP addresses. They may also lack the full network protocol stack and finding the true location of an unmanaged switch can prove difficult even if it is detected (Bhuse, 2019). There are some solutions out there that help in detecting rogue switches. One method that has been investigated is leveraging the Dynamic Host Control Protocol (DHCP). In this method you can use the DHCP request to see if a computer requests an IP if the connection from the workstation to the network switch is severed and it doesn’t request an IP after reconnecting the network. Another useful technique that can be deployed today to help eliminate rogue network switches is implementing MAC address whitelisting. An organization could create a whitelist of valid MAC addresses of known switches, and the valid switches would know of its network’s valid MACs to communicate with. If a switch received a packet from a MAC address that is on the whitelist, it would forward that packet and add that MAC to its forwarding table. If a switch received a packet from an unknown MAC that was not on the whitelist, it could flag the packet as potentially originating from a rogue device (Bhuse, 2019).

2.1 Detecting rogue access points in a computer network

Keohane came up with a computer implemented method, data processing system, and computer program product for discovering an unauthorized router in a network. The process first obtains a physical address of a suspected router or destination device. A data packet is created which contains a destination MAC field, a destination internet protocol field, and a time to live field. The destination MAC field includes the physical address of the destination device, where the destination internet protocol field includes a fake internet protocol address, and where the time to live field includes a value indicating the data packet has exceeded a time limit. The data packet is sent to the destination device using the physical address in the destination MAC field. If a time exceeded message is received from the destination device, the destination device is determined to be enabled for routing (Keohane, 2011).

2.2 Rogue router hunter

Lin and Chen produced a method of detecting an unauthorized wireless access point in a network using a detector. A rogue access point detector receives an incoming data packet which is scanned for a time expiration value. The time expiration value may be a time to live value. It is determined whether the time expiration value is the same as a threshold time expiration value. If the time expiration value is not the same as the threshold value, it is determined whether the incoming data packet was routed through an authorized access point in the network. If it is determined that the packet is not being routed from an authorized access point, a security alert is sent to the network, which could be a device like an administrators’ computer (Lin, 2011).
3. Problem

The main question today is how do you detect unmanaged switches on a network? They don't have IP addresses or any other way to be contacted directly. The typical personal or desk switch may have four or five ports, a power plug, and LED status lights. There is no direct connection to them like a serial cable to interface with.

This perked my interest as a potentially interesting problem to solve. If you had the time and money, you could just brute force the problem by picking one of your own (managed) switches and tracing out from each port to the rogue system. If you encounter another device along the way, then you have your proof of a rogue switch.

This doesn’t seem feasible for large enterprises and doesn't seem to be time or cost effective. That sort of approach does not scale. It might be good if you suspect a certain person or know of an exact location of a breach, but it would be difficult in a large implementation.

You could also try to monitor the port for traffic for rules or policies such as more than one MAC address on a port. This also takes a fair amount of work, assuming your switch supports getting the traffic information on that port. If your switch supports port security and MAC filtering, you might be able to do that, but again, that requires maintenance, since network changes occur daily.

After pondering this one for a while, we came up with the idea of looking at port link state of the switch and computer. We are assuming that we have control of the switches. What this enables us to do is to enable and disable ports on the switch either manually or through software (if the switch has that capability). We would be able to temporarily turn off a port if we suspected something isn’t normal in the condition of the port or it isn’t meeting a security rule or policy.

4. Scenario setup

To setup and test our link state theories we used a combination of Windows Computers, Wireshark, a local good network switch, a rogue switch, and a router with DHCP (Netgear, 2021). By using a local switch, we could control and simulate the link state of a port and whether it was on or off. When we toggled the control of the port, we could see the difference in communication packets in Wireshark (Wireshark, 2021). In our scenario testing we made a couple assumptions: we had access to the good network switch, and we had access to a computer that was connected to the rogue network switch. There is a router, which is connected to the good switch, which is connected to a rogue switch, which has a computer connected to it (Figure 1).

![Figure 1: Scenario Setup](image)

5. Results

The scenario was started by changing the state of the downlink port from the good switch to rogue switch from on to off (Figure 2 and Figure 3), this would simulate the port looking just like it had been unplugged from the good switch or turned off through software. The computer we have control of and have access to is connected to the rogue switch where the uplink port was turned off. The computer did not lose its connection because it was still connected locally to the rogue switch. It still saw the active ethernet connection to the switch, even though the uplink on the rogue switch had been severed. The computer was able to log the change in network connectivity which resulted from the good network switch port being deactivated. The rogue switch is now disconnected from the rest of the production network from a logical perspective, but at the physical layer, the computer can still see activity. To find this fact we used the Windows Event Viewer (Figure 4).
Another useful tool to use during this port enable and disable scenario was Wireshark and Windows command using ARP. We were able to use ‘arp -a’ and ‘arp -d’ to refresh the arp table on the local PC. Showing in a normal state you can see the router in the arp table (Figure 5). This screenshot was taken with an active uplink connection on the rogue switch. We could see the router because we had an uplink connection to the good switch which was connected to the router.

When we severed the uplink to the rogue switch, we were able to use the Windows command and clear the arp table. When we ran the command arp -a we found a different result than we got earlier with ethernet connectivity to the rogue switch. You can now see that we’re no longer able to see the router in the arp table (Figure 6). This shows that even though our ethernet connectivity to the PC remains in place, we lose routing due to the uplink being disabled.

We were able to find similar findings by using Wireshark and searching for ARP packets. When the uplink on the rogue switch was enabled, we could communicate with the router at 192.168.0.1 (Figure 7). With the uplink disabled we saw a different result in the ARP packets. The computer was asking for router information but couldn’t find the information since the uplink was disabled (Figure 8).
Travis Quitiquit and Vijay Bhuse

Figure 6: ARP table with uplink disabled on rogue switch

<table>
<thead>
<tr>
<th>No.</th>
<th>Source</th>
<th>Destination</th>
<th>Protocol</th>
<th>Length</th>
<th>Info</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>192.168.0.100</td>
<td>192.168.0.99</td>
<td>ARP</td>
<td>8</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>192.168.0.200</td>
<td>192.168.0.255</td>
<td>ARP</td>
<td>8</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>192.168.0.255</td>
<td>192.168.0.200</td>
<td>ARP</td>
<td>8</td>
<td></td>
</tr>
</tbody>
</table>

Figure 7: Wireshark ARP packet with uplink enabled on rogue switch

Figure 8: Wireshark ARP packet with uplink disabled on rogue switch

The network connection shows from the logs and constant ping that the network connection to the computer was always up (Figure 9). The computer never lost its original DHCP reservation from when the uplink was enabled. In the scenario the only time the Windows Event Viewer showed a disconnect from the ethernet connection was when we physically unplugged the ethernet cable from the computer itself (Figure 10).

Figure 9: Ping was consistent even with uplink disabled

Figure 10: Computer with Ethernet Cable unplugged from rogue switch

Another useful command line tool was used to look at the difference in port and connection status of the link to the computer. Using `netstat -a` we can display active TCP connections, TCP connections with the listening state, as well as UDP ports that are being listened to. We ran netstat with an active connection to the router (Figure 11) and another test with the uplink disabled and unable to reach the router (Figure 12).

Figure 11: Active TCP connections

Figure 12: Netstat with uplink disabled
6. Conclusions

In our scenario we were able to provide some evidence of a method that could be used in conjunction with other rules and policies to detect rogue switches connected to a network. We were able to determine based on the port link state that there was another device, most likely a rogue switch between the good switch and the computer. The fact that the computer’s ethernet connection never dropped even though we severed the link from the good switch to a port suspected of having multiple MACs shows that there was a device on the other end keeping the ethernet connectivity of the computer alive. Pulling all the information together from the Windows Logs, Wireshark, and Windows command line tools we formed a good method of what to look for on a computer to see if there is a rogue switch installed somewhere on the network that may affect network connections and protocols.

Another method that could be used that is related to our findings is using the DHCP server, if one exists. You could use DHCP server logs to see if that machine starts asking for a new lease after the good network port is toggled on and off, essentially turning off the uplink on the rogue switch. If the computer doesn’t ask for a new
lease, it’s likely that the computer didn’t see the port state change, which means something blocked it from seeing that, most likely a rogue switch.

If an adversary was able to install a switch that was set to drop the port link states on its own ports when its uplink port also dropped off, that could prove to be another problem in of itself. Overall, we believe the findings of this scenario show another novel method that could be used to detect rogue switches on a network.

References


