Network Automation Methodology for detecting Rogue Switch

Vineet James

Department of Computer Information Systems
Grand Valley State University
Grand Rapids MI, USA
jamesvin@mail.gvsu.edu

***Abstract* —** The issue of detecting malicious switches on the network is still a concern even as networks continue to grow more complex. Even though Wired networks are considered more secure than wireless, the wireless rogue device problem has been solved. However, the wired rogue switch problem remains unsolved.

In this project, We apply core networking concepts and demonstrate a smart solution by combining the latest Automation techniques with highly effective software tool-sets available for detecting malicious systems connected to a rogue switch. This solution promises quick detection and requires Zero Downtime which could prove to be an ideal solution for enterprises having managed switch production networks.

We achieve this by continuously filtering and analyzing network traffic for any broadcast storms or new Address Resolution protocol packets using Packet Analyzers and then effectively tracing the malicious host connected to the rogue switch by deploying automation techniques. This technique also helps detecting rogue unmanaged switches ("plug and play" devices) having pre-loaded configuration.

I. Introduction

We begin with looking at two typical types of switches in the Network Infrastructure – Managed and Unmanaged.

The key difference between them lies in the fact that a managed switch can be configured and it can prioritize LAN traffic so that the most important information gets through[1].

An unmanaged switch on the other hand behaves like a "plug and play" device. It comes pre-configured and simply allows the devices to communicate with one another. They do not have an IP address and some also have no MAC address which make them very difficult to trace.

In this article, we initially put forwards the idea why this is an unsolved problem and what are the challenges that network administrators face to detect an unmanaged rogue switch on the network. Then we see some network concepts which help us get to the solution and we work on the solution in detail going through every step, collecting evidences to finally reach a conclusion.

This smart solution is implemented and demonstrated by combining the latest Automation techniques with highly effective software tool-sets available for effectively detecting and tracing malicious systems connected to a rogue switch.

II. Challenges to detect rogue switch

A. *Unmanaged switches are untraceable*

An unmanaged switch comes pre-configured and simply allows the devices to communicate with one another. These are Datalink Layer (referring to the OSI layer model) devices and do not have IP addresses and therefore ICMP tools such as traceroute cannot be used to trace the device. Moreover, some unmanaged switches also do not have MAC address which means there will be no entry of the switch on any CAM table of other switches. This makes the job of a network administrator who is trying to trace the rogue switch extremely difficult.

B. *Discovery Protocols are ineffective*

Switches being a Layer 2 device means we can use discovery protocols which are used to monitor directly connected neighbors and update connectivity status. Examining these neighbor-ship protocols would let us know the connected switch

The scope of this standard is to define a protocol and management elements, suitable for advertising information to stations attached to the same IEEE 802 LAN, for the purpose of populating physical topology and device discovery management information databases. The protocol facilitates the identification of stations connected by IEEE 802 LANs/MANs, their points of interconnection, and access points for management protocols[2].

However, such protocols such as LLDP (Link Layer Discovery Protocol), CDP (Cisco Discovery Protocol) will not be able to retrieve information about the connected rogue switch as most unmanaged switches do not support Discovery protocols, which means the managed switches CDP or LLDP updates won't be acknowledged by the directly connected rogue switch and would totally bypass it.

C. *Even STP BPDUs can't help detect rogue switches*

Bridge Protocol Data Units (BPDUs) are frames that contain information about the Spanning tree protocol (STP), which helps avoiding loops in the network. Managed Switches

send BPDUs for STP algorithms to function, the switches need to share information about themselves and their connections. What they share are bridge protocol data units (BPDUs) [3].

Typically, a BPDU guard is configured on the ports of managed switches to examine STP and disables the port which receives BPDU packets if it's not coming from a legitimate switch. However, Unmanaged switches do not have support STP and hence do not send BPDUs making this possibility of tracing it ineffective.

D. IP sweep tools cannot capture rogue switch location

Another possible way to detect a rogue switch is to somehow get some information about the hosts that are connected to that rogue switch. There is a possibility of running an IP sweep using tools in the entire LAN and examine the output for any anomalies in the network. These results can be utilized by the Network administrator to get clues on where to find the rogue switch and its actual physical connection on the network.

Today, we have such IP sweep tools available to us like NMAP (Network Mapper) which is a security scanner which can be used for this purpose. It is used to discover hosts and services on a computer network, thus building a "map" of the network. To accomplish its goal, these tools sends specially crafted packets to the target host(s) and then analyzes the responses[4].

Performing an IP sweep with such tools seems to be a good option, however, if you don't know what you are looking for it will be almost impossible to find. The problem with unmanaged switches is that there is no way to get information out of them as it has no management IP or supports SNMP. This makes finding the IP of the host connected to that rogue switch among a large pool of IP addresses impossible. The rogue switch remains entirely invisible to the mapping tool and does not show up in the topology map simulated by i**t.**

III. CONCEPTS LEADING TO THE PROPOSED SOLUTION

The solution that this article proposes is based upon certain networking concepts like ACD (IP Address Conflict Detection), ARP (Address Resolution Protocol) and Broadcast packets in addition to Wireshark tool which is a packet sniffer that we shall use to analyze frames on the LAN.

Before we actually work on the solution for detecting a rogue switch on the network, let's understand few Network concepts that we will be using that lead us to the solution.

A. ACD, ARP-Probe and ARP Announcement

ACD, also known as Address Conflict Detection is a utility of IPv4 which is used to avoid IP conflict in a LAN environment. This applies to all IEEE 802 Local Area Networks (LANs)[802],including Ethernet [802.3], Token-Ring [802.5], and IEEE 802.11 wireless LANs [802.11] [5].

The ACD utility in IPv4 uses 'ARP Probe' a term used to refer to an ARP Request packet, broadcasted on the local link. Before beginning to use an IPv4 address (whether received from manual configuration, DHCP, or some other means), a Network Interface Card in a host must test to see if the address is already in use, by broadcasting ARP probe packets.

An ARP Probe conveys both a question ("Is anyone using this address?") and an implied statement ("This is the address I hope to use.").

ARP Probe packets are broadcasts with an all-zero 'sender IP address'. The 'sender hardware address' MUST contain the hardware address of the interface sending the packet. The 'sender IP address' field MUST be set to all zeroes. The 'target hardware address' field is ignored and SHOULD be set to all zeroes. The 'target IP address' field MUST be set to the address being probed. ARP Probe header as illustrated in Fig1.1 below.

```
Address Resolution Protocol (request)
   Hardware type: Ethernet (1)
   Protocol type: IPv4 (0x0800)
   Hardware size: 6
   Protocol size: 4
   Opcode: request (1)
   Sender MAC address: Vmware_c0:00:01 (00:50:56:c0:00:01)
   Sender IP address: 0.0.0.0
   Target MAC address: 00:00:00_00:00:00 (00:00:00:00:00:00)
   Target IP address: 192.168.174.111
```

Fig 1.1 – ARP Probe

This process sends 3 ARP Probes, and if no one responds, the host officially claims the IP address with an ARP Announcement. Finally, the IP address is mapped to the physical address of the Source host. Fig1.2 below illustrates the ARP Announcement header when the host finally maps the physical address and tries to confirm if it has set a unique IP address on the LAN.

```
Address Resolution Protocol (request/gratuitous ARP)
   Hardware type: Ethernet (1)
   Protocol type: IPv4 (0x0800)
   Hardware size: 6
   Protocol size: 4
   Opcode: request (1)
   [Is gratuitous: True]
   Sender MAC address: 00:50:56:c0:00:01
   Sender IP address: 192.168.174.111
   Target MAC address: 00:00:00:00:00:00
   Target IP address: 192.168.174.111
```

Fig 1.2 – ARP Announcement

B. Switch behaviour when connected to a network

- The switch table is initially empty.
- For each incoming frame received on an interface, the switch stores in its table the MAC address in the frame's *source address field*, the interface from which the frame arrived. In this manner the switch records in its table the LAN segment on which the sender resides.
- There is no entry in the table for the Destination MAC. In this case, the switch forwards copies of the frame to the output buffers preceding *all* interfaces except for incoming interface. In other words, if there is no entry

for the destination address, the switch broadcasts the frame [6].

- As soon as the hosts tries to initiate some connection to other hosts, the switch tries to learn MAC addresses by use of broadcasts.

C. *SNMP for monitoring*

Simple Network Management Protocol (SNMP) is an Internet Standard protocol for collecting and organizing information about managed devices on IP networks and for modifying that information to change device behavior. Devices that typically support SNMP include cable modems, routers, switches, servers, workstations, printers, and more[7].

SNMP-managed network consists of two key components:

- Agent – software which runs on devices that we want to monitor. This can be configured in such a manner that it sends Trap signal to the SNMP Manager reporting any unknown behavior on it.
- Network management station (NMS) – software which runs on the manager. A network management station executes applications that monitor and control managed devices.

IV. THE SOLUTION

This section covers how we can collaborate all concepts defined and explained in the previous section and co-relate evidences gathered using those concepts to finally detect a Rogue switch in a network.

Let us assume a network design as shown in Fig 1.3, where we see 4 managed switches S1, S2, S3, S4 are connected, distributing the network into 3 departments. Here, assuming no Port-security features are configured on the managed switch, meaning that anyone can walk in and plug in an unmanaged switch in one of the ports. Here we consider a Rogue switch is plugged in port 5 of Switch S4, as illustrated in Fig 1.3 below.

We also observe from Fig 1.3 (circled and connection marked in 'red') that three hosts are connected to the rogue switch which are trying to consume the entire bandwidth of the LAN by downloading heavy files using Bit-Torrent from the Internet.

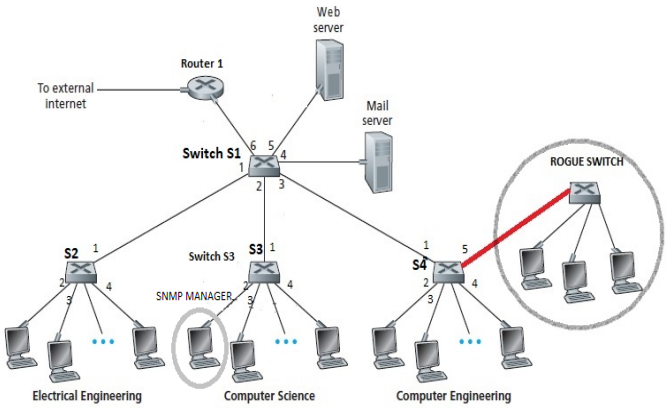

Fig 1.3 – Network Topology

In this design, the internal LAN is connected to the Internet via a Router R1 and the host connection go through the router to establish a connection with the outside world.

The following steps were implemented and demonstrated on a real network emulator under a testing environment setup. The implementation details will be discussed in the next section. GNS3 was used to setup the network topology, configured with real Cisco images. The solution was tested on a simple network as illustrated in fig 1.3 and also on a multi-VLAN complex network which also displayed the same results.

A. *Step 1 – Detect Brodcasts using SNMP*

As already discussed in the above section, we know that a switch's CAM table initially has zero entries when it is plugged into an existing network unless a host gets connected to any port of that switch and tries to communicate. As soon as a host is plugged into one of rogue switch's port, it will start broadcasting traffic to all ports as it has no MAC entries in its CAM table. This causes a broadcast storm on the network which is received by all switches in the network.

All legitimate switches in the network are configured as SNMP agents and pick up these broadcasts. The agents are configured to report Broadcast storms in the network and send Trap signals to the SNMP Manager (As in Fig 1.3) on the network.

These SNMP Trap messages will alert the Network Administrator. This broadcast storm indicates that a switch has been connected on the network, however, more evidence will be needed to find the physical location of the Rogue switch.

B. *Step 2 – Using Wireshark to examine ARP Probes*

We also discussed about ARP Probes and ARP announcements in the Section above. To trace the Rogue switch, we need to first trace the IP addresses of hosts that are connected to the rogue switch. We need to know exactly what we're looking for on this busy network.

We use Wireshark to sniff all ARP Probes around the time since we received an SNMP Trap for broadcasts in the network. Filtering ARP packets and looking for ARP Probes will give us the most recently configured IP addresses on the network, one of which can trace us back to the Rogue switch.

Fig 1.4 illustrates the Wireshark trace showing 3 ARP Probe packets followed by an ARP announcement which shows the host attains an IP of 192.168.174.111.

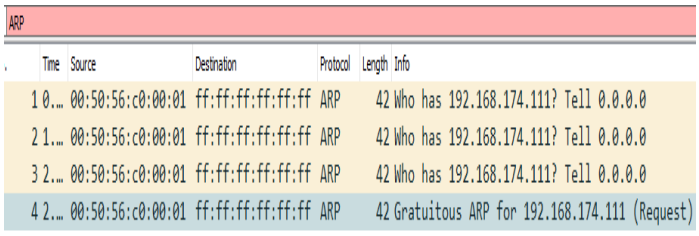

Fig 1.4 – Packet capture

Now, that we the host whose ARP probes corresponds to the same timestamp as the broadcast signal. We can use that IP to trace the Rogue switch.

C. *Step 3 - ARP table at the Router*

From the uplink router, it being a managed device the ARP entries can be seen for that specific IP as it has been downloading files from the Internet using Router R1.

Once we get the MAC address of the Host, we can trace the switch going downwards from the Router to the switches.in their forwarding databases. The MAC found on a switch link can be traced by following the link to the next switch until you find the access-port on which the MAC is listed. This is where the malicious rogue switch is connected.

V. IMPLEMENTATION DETAILS

GNS3 network emulator software and VM used for creating a virtual network topology using actual Cisco images.

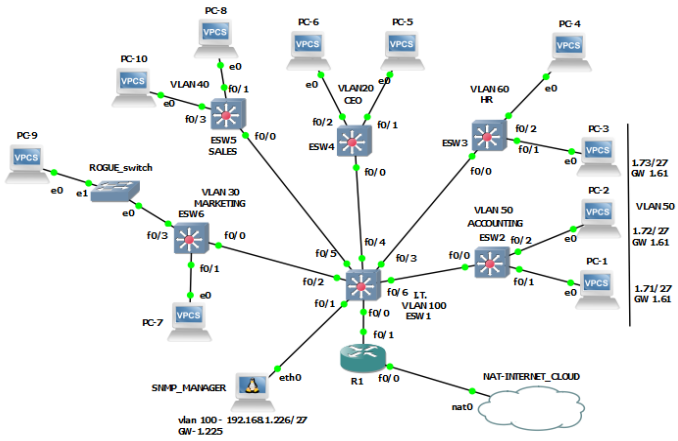

Fig 1.5 – GNS 3 – Multi-VLAN Network topology

WIRESHARK packet analyzer used for network monitoring and analyzing ARP and broadcasts. Monitoring of special ACD (Address Conflict Detection) packets proved vital in finding out the host IP connected to the rogue switch.

```
333 590.379079  Private_66:68…  Broadcast  ARP  64 Gratuitous ARP for 192.168.1.238
334 591.381455  Private_66:68…  Broadcast  ARP  64 Gratuitous ARP for 192.168.1.238
336 592.383194  Private_66:68…  Broadcast  ARP  64 Gratuitous ARP for 192.168.1.238
343 604.454846  Private_66:68…  Broadcast  ARP  64 Who has 192.168.1.225? Tell 192.1

> Frame 333: 64 bytes on wire (512 bits), 64 bytes captured (512 bits)
> Ethernet II, Src: Private_66:68:00 (00:50:79:66:68:00), Dst: Broadcast (ff:ff:ff:ff:ff:ff)
v Address Resolution Protocol (request/gratuitous ARP)
    Hardware type: Ethernet (1)
    Protocol type: IPv4 (0x0800)
    Hardware size: 6
    Protocol size: 4
    Opcode: request (1)
    [Is gratuitous: True]
    Sender MAC address: Private_66:68:00 (00:50:79:66:68:00)
    Sender IP address: 192.168.1.238
    Target MAC address: Broadcast (ff:ff:ff:ff:ff:ff)
    Target IP address: 192.168.1.238
```

Fig 1.6 – ACD packet capture

Python Scripts were implemented using the **Netmiko library** for configuring Cisco devices. Scripts were pushed on the **Docker cloud container** acting as the management system to automatically Trace the malicious system downwards from the uplink L3 device having its own ARP table using Python automation scripts.

Fig 1.7 – Technologies

VI. FUTURE WORK

This article successfully demonstrates how to detect a rogue switch on the wired network, however, there is a lot of future scope for enhancing this solution into a fully functional centralized management tool for effective network management and with advanced fault detection features.

Plans in place to continue building a fully functional centralized management tool with advanced GUI features.

GUI features would include smart network configuration, backups, management and fault detection capabilities of network equipment.

Full scale Integration with hardware networking equipment.

A Wireshark plug-in could be developed providing full integration with the platform.

VII. SOME MEASURES TO PROTECT YOUR NETWORK

Authentication Servers – These are servers which provide authentication, authorization and accounting in the network environment. Authentication servers in combination with 2-Factor or even 3-Factor authentication techniques should be used to secure the network of any unauthorized access. Shared secrets and Biometrics are used for this purpose.

Biometric identifiers are most commonly used as part of a multifactor authentication system, combined with a password (something a person knows) or a token (something a person has) [8].

Various biometric techniques and identifiers are being developed and tested, these include:

- fingerprint recognition;
- face recognition;
- voice recognition;
- keystroke recognition;
- handwriting recognition;
- finger and hand geometry;
- retinal scan;
- iris scan.

Port Security – Many devices that support MAC filtering do so on a device basis. Whitelisted MAC addresses are allowed through any port on the device and blacklisted MAC addresses are blocked on all ports. When port security is configured, the default settings are to allow only one MAC address per port, and to shut down the port if the allowed number of addresses is exceeded [9]. Configuring port-security restricts the number of MAC addresses that can be connected to a switch port which will then automatically block any unauthorized switch connection.

VIII. REFERENCES

[1] "Cisco Networking Academy's Introduction to Basic Switching Concepts and Configuration". Cisco Systems. 2014-03-31. Retrieved 2015-08-17

[2] "802.1AB-REV - Station and Media Access Control Connectivity Discovery". IEEE. Retrieved 2009-10-17

[3] Cisco (May 2007). "Configuring Spanning Tree Protocol". Retrieved 2014-06-10

[4] "The History and Future of Nmap". Nmap.org.

[5] Cheshire, S. (July 2008). "RFC 5227 - IPv4 Address Conflict Detection". Internet Engineering Task Force.

[6] Computer Networking: A Top-Down Approach, 6th Edition by James F. Kurose and Keith W. Ross, 2013, Pearson, ISBN13: 9780132856201.

[7] Douglas R. Mauro & Kevin J. Schmidt. (2001). Essential SNMP (1st ed.). Sebastopol, CA: O'Reilly & Associates.

[8] Federal Financial Institutions Examination Council (2008). "Authentication in an Internet Banking Environment" (PDF). Archived (PDF) from the original on 2010-05-05. Retrieved 2009-12-31.

[9] Automate the Boring Stuff with Python: Al Sweigart, April 14, 2015.

[10] GNS3 – official documentation, February 2, 2019

[11] "Configuring Port Security". Cisco. Retrieved 14 November 2015.