International Journal of Recent Technology and Engineering (IJRTE)
ISSN: 2277-3878 (Online), Volume-8 Issue-6, March 2020

Bypassing Wired Port Security

Vishnu V., Praveen K.

Abstract*: 802.1x is a part of the IEEE 802.1 group of Networking Protocols. It is mainly used to implement Port-based Network Access Control (PNAC) by providing an authenticating mechanism to connect to LAN or WAN. Existing attacks on the 802.1x are mainly focused on the older version of the protocol which does not provide encryption or enable authentication on a packet-by-packet basis. Later versions of the protocol includes MACsec to provide a two layer encryption to maintain the data integrity of the network packets. It also included support for devices like printers and VOiP phones which led to an easy attack vector. An attacker could easily spoof the MAC address to get into the corporate network. In this paper we go through the 802.1x protocol, the authentication mechanism of the protocol, the existing attacks on the protocol and a new attack to bypass Network Access Control enforced by the 802.1x. The proposed attack is an improvement on the NACKered script which was built upon the Alva Lease's Duckwall IV. We have added a couple of modules to run the responder script as well as an awareness script to keep it persistent. We end the paper by listing out the best practices that must be followed when setting up a corporate network with Network Access Control with 802.1x.

Keywords: 802.1 xs, EAP, Network Access Control, Port Security, RADIUS

I. INTRODUCTION

802.1x is an IEEE standard that provides a reliable solution to network access/admission control in enterprise networks through port-based authentication. Wired and wireless networks are commonly used to protect and isolate networks for servers, computers Network access devices peripherals and motive devices operated by users. Many of the network administrators are vulnerable to serious threats, which may lead to performance reduction due to total service failure by preserving the stable Wireless Local Area Network (WLAN) and insecure physical ports.

The research community has paid little attention to wired port security and has given the vendors most of the job to implement new protocols and services. The use of various proprietary and open source authentication schemes allows the implementers the flexibility to use technologies according to their context and to assess its effect on network performance on their own. When implementing authentication 802.1x in large networks, careful examination is required, in which even limited overhead traffic may contribute to severe network performance deterioration, Especially when large numbers of users are in the network.

In the last decade, wireless networking has advanced significantly and has been the basis of mobile computing. While wired networks are versatile and commonly used in all environments, wired LANs remain at the heart of every small-to-large infrastructure. It is a fact that the academic research community paid little attention to the architectural security in the wired Ethernet and left most of the work to the equipment manufacturers while giving more emphasis on the higher-level protocols and wireless networks. The academic research group has shown little commitment to the architectural security of wired Ethernet and has left most of the work to the vendors whilst putting greater emphasis on higher-level protocols and wireless Internet networks [1].

Nearly every network has physical wired Ethernet access ports, most of which lack network connection security. Therefore, network technicians are obsessed with using a Network Access Control (NAC) methodology often known by various equipment manufacturers as a network access control.

The principal aim of NAC being introduced is to verify the connecting entity before allowing it to engage in the network for authorized users/devices to access resources in compliance with the defined Access Policies. The required network access is allowed if the user has been correctly authenticated, or the port itself is disconnected from LAN, to prevent further connection to any part of the network.

II. THE 802.1X PROTOCOL

The 802.1x protocols involves the following components:

Supplicant System: The client-side software, installed on the user's computer/ device requesting network access. To start the authentication process, it uses Extensible Authentication Protocol via LAN (EAPoL). Today, the supplicant is an integral part of wired communication suites (802.3, 802.5) and wireless connection suites (802.11) on almost all operating systems.

Authenticator System: An 802.1x protocol-supporting device located in the middle of the supplicant network. The most frequent authenticator system examples are switches and Wireless Access Points. The authenticator uses a dual-port model for authentication of Controlled and Uncontrolled ports.

Manuscript received on February 10, 2020.
Revised Manuscript received on February 20, 2020.
Manuscript published on March 30, 2020.

* Correspondence Author
Vishnu V.*, TIFAC-CORE, Cyber Security, Amrita School of Engineering, Amrita Vishwa Vidyapeetham University, Coimbatore, India. Email: cb.en.p2cys18029@cb.students.amrita.edu
Praveen K., TIFAC-CORE, Cyber Security, Amrita School of Engineering, Amrita Vishwa Vidyapeetham University, Coimbatore, India. Email: k_praveen@cb.amrita.edu

The ports must not be confused with physical ports; these are simply two separate logical states of a physical port that can be managed by a Layer 2 switch. The Uncontrolled Port is always available and is used to enable the user to access the correct authentication services on the network by enabling EAPoL traffic only.

Authentication Server: Usually, a user authentication server and Authentication, Authorization and Accounting (AAA) protocol by maintaining user data such as user IDs and passwords, network access schedules and provider form of service allocation [2]. The certificate authority can also be combined into 802.1x and an additional user accounts like the Active Directory (Microsoft Active Directory) and the Lightweight Directory Access Protocol (LDAP) server can be used, rather than RADIUS credentials. The authenticator asks the user for his identity when a new device is connected to an 802.1x network. If the supplicant is not installed or running on the connecting device the switch port may simply deny network access. The connecting device passes the network credentials or the certificate through the uncontrolled authenticator port to the authentication server with EAPoL (Switch or Access Point). If the authentication is successful, the controlled port will open and traffic over the LAN is allowed. The controlled port remains unchecked and there is no clear LAN contact.

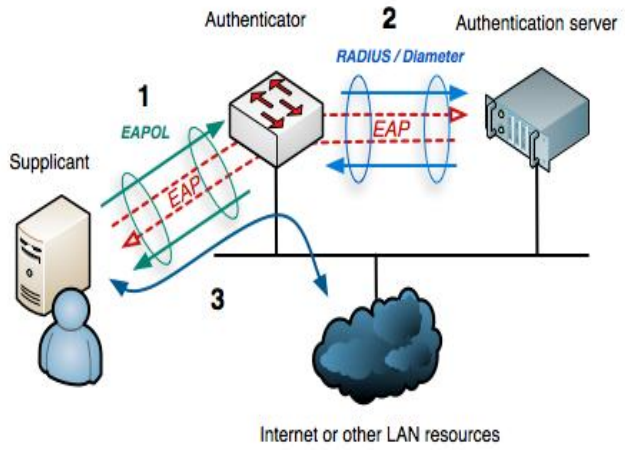

Fig. 1. Components of the 802.1x framework [1].

A. The 802.1x Authentication Process

The authentication process for 802.1x contains the following four processes [3].

Initialization: In this phase, the client or the supplicant connects to a port on the authenticator or a switch. Since this is a new connection, the switch opens the port which was initially disabled but only allows 802.1x traffic to be transmitted. This is called a restricted state, and, in this state, the open port is unauthorized.

Initiation: The 802.1x authentication process can be initiated by either the supplicant or the authenticator. The authenticator waits for the supplicant to send an EAPOL-Start frame. Once it receives it, the authenticator then requests the identity of the supplicant. The supplicant encapsulates this in a RADIUS ACCESS-REQUEST frame and then forwards it to the authentication server.

EAP Negotiation: Once the supplicant sends the encapsulated packet to the server, it responds with an EAP-Request frame encapsulated within a RADIUS ACCESS challenge. The authenticator forwards the EAP-Request frame to the supplicant by stripping the RADIUS ACCESS challenge. The supplicant then begins the EAP authentication using the usual EAP methods or responds with a Negative Acknowledgement which includes a list of acceptable methods.

Authentication: Once the EAP method is agreed upon by the supplicant and the authentication server, the authentication process begins. The specifics behind the authentication server depend on the EAP method chosen. Regardless of the EAP method chosen, the authentication process results in an EAP-Success or EAP failure message. If successful, the authenticator changes the port from an "unauthorized" state to "authorized".

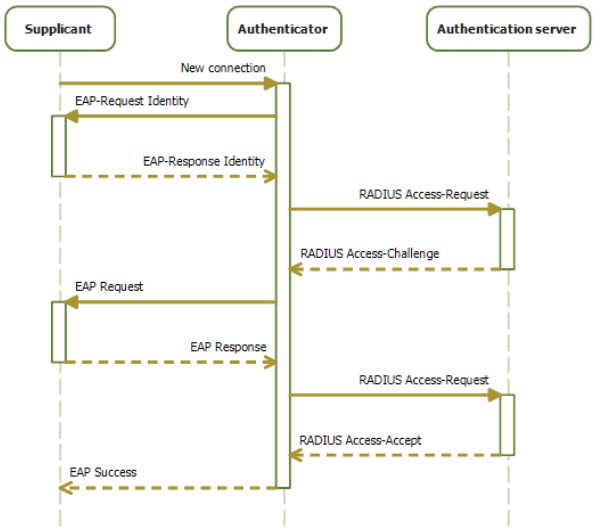

Fig. 2. 802.1x authentication [3].

B. Existing Attacks

Steve Riley's 802.1x-2004 bypass: In 2005, Microsoft researcher, Steve Riley found that 802.1x authenticates only when a new connection is established. Once a client is authenticated and the switch ports are opened, further connections by the same client are not authenticated. In such a situation, an attacker can perform a man-in-the-middle attack by disconnecting the authenticated clients from the port and connecting a hub instead. The attacker then connects the shadow system to the network and changes the mac and IP addresses of the shadow system to match the victim's computer. This attack was successful only on UDP as TCP packets would cause a race condition.

Marvin By Gremwell Security: "ABB" of Gremwell Security developed a tool called Marvin to bypass 802.1x. In this attack, a bridge is used instead of a hub. This attack introduces a rogue device directly between the supplicant and the switch. The tool diverts and re-injects network packets while preserving the original network addresses. The tool will bridge the traffic between the first and the second interface and inject traffic it receives from the third interface into the first two.

Alva Duckwall's 802.1x bypass: This is an improvement on the Marvin tool. It uses a transparent bridge to introduce the rogue device between the supplicant and the switch.

In this attack, packet injection is not necessary as network interaction is granted by using IPTABLES to source the NAT originating from the device. Traffic can be passed between the rouge device and the bridge by a hidden ssh service running on port 22 on the device. NACKered is a simple bash script based on Alva Lease's Duckwall IV activity to bypass Network Access Control 802.1x.

Valérian Legrand's Fenrir: This tool, created in 2017, works in a similar way to Duckwall's tool. It is written in Python instead of Java. Instead of using tools like iptables, ebtables and arptables, it makes use of a python module called scapy.

III. PROPOSED ATTACK

Our proposed system is built upon the NACKered script. We have also included a module from Laurent Gaffiie's Responder tool to gain access to login credentials while performing red teaming.

Access to a device that has already been authenticated is the fundamental requirement for the attack to succeed. This tool is used to connect to the network and inject network packets from different devices. This means placing the system of an attacker between the authenticated computer and the network switch we performed the attack using a Raspberry Pi and two network adapters.

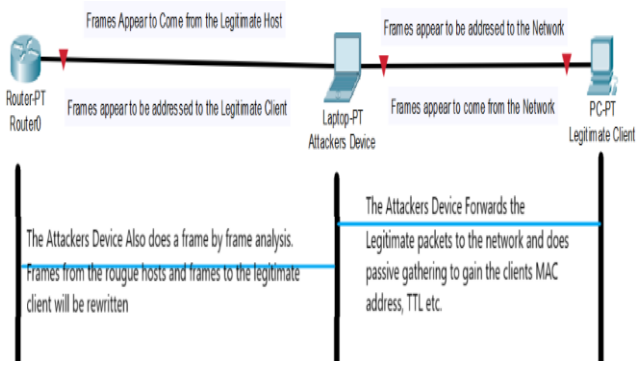

Fig. 2. Proposed System

The original NACKered script and our modified script are written on Debian based Linux distributions. The following software packages were used:

bridge-utils: The bridge-utils package contains a utility needed to create and manage bridge devices.

macchanger: Randomly creates and allocates a new MAC address to the network interface eth0.

arptables, iptables, and ebtables: Arptables are used to set up, maintain, and inspect the tables of ARP packet filter rules in the Linux kernel. Several different tables may be defined. Each table contains several built-in chains and may also contain user-defined chains.

mii-tool: This tool checks or sets the Media Independent Interface (MII) unit status of a network interface.

tcpdump: On a network interface matching the Boolean expression, Tcpdump prints a summary of the packet content.

br-netfilter (kernel module): Bridge filtering for a man-in-the-middle network node.

The bypass script is used along with a legitimate device as shown in the diagram below.

The legit client device is not connected to the network switch at first. Now on the attacker device, the bypass script is started. Bypass and attacker is one physical device. The figure of the attacker symbolizes the attacker's behavior against the NAC bypass. The first step is initial configurations: Unwanted services, such as the NetworkManager, are stopped, IPv6 deactivated and any DNS configurations initialized. The bridge is set up and deployed next. The kernel needs to be configured to forward EAPOL frames to ensure that bridging operates as desired. 802.1X authentication is not performed without this modification.

The network cables can be plugged in and the switch side can now be used as a passive forwarder after the setup is done. The bypass device transfers back and forth all network traffic from switch to client but cannot transmit packets. The client must now have the network switch authenticated and can successfully log into the network.

All network traffic passes the bridge and is therefore analyzable. This is done to capture Kerberos and SMB packets using tcpdump, as they are usually found multiple times over a Windows network, allowing network settings, such as the IP and the MAC address of the client to be accessed. This knowledge is used to set up the client-side of the bridge automatically. The network connection of the bypass remains to be blocked to ensure network packets from the attacker system are found and detected on the network.

If packets from the attacker are sent later onto the network, a MAC address will be overwritten by an ebtables law, which ensures that they appear like packets from the client Using IP law, the same process is applied with the outgoing TCP, UDP and ICMP systems having the same IP address as the client. Eventually, the attacker will connect to the network and act from his computer.

The awareness.sh script addresses changes in status and modifies the NAC bypass setup. All other measures must currently be performed manually using SSH on the device. Nevertheless, this SSH service can only be accessed via the victims ' network. The introduction of an extra management interface with a WLAN or cellular network adapter is one of the next steps.

A. Common EAP methods.

Extensible Authentication Protocol (EAP) is a widely used network and internet network authentication mechanism. It is specified in RFC 3748, rendering RFC 2284 redundant and updating it with RFC 5247 [4]. EAP is an authentication system for material and parameters developed by EAP processes to be transported and used. EAP is not a wire protocol but specifies interface information and formats only. That EAP protocol specifies a way of encapsulating messages within the messages of the client EAP.

Several ways of applying the EAP are available. The EAP methods are recognized for these different EAP implementations [3]. Some of the most popular EAP approaches used are:

EAP-MD5: The authentication of the EAP-MD5 process starts when an EAP-Request Identity is sent to the user by the authentication server. The supplicant responds with an EAP address identity that allows a randomly generated question string for the authentication server. This challenge string is sent from the authentication server to the supplicant as an MD5-Challenge Request.

The supplicant then puts the username, password and challenge string together into a single value, and sends the MD5 hash of that value to the MD5-ChallengeResponse server. When the MD5 challenge address has been received, the authentication server continues the query process: the authentication server connects the username, the password, and the challenge string into a unique value, which is inserted into the haze function of MD5. This second MD5 hash (created from the authentication server) is contrasted with the supplicant's MD5 request-response. The encryption effort works if the two hashes are the same. The encryption system is not authenticated when used alone. The encryption system is not authenticated when used alone. Otherwise, it will fail As Josh Wright and Brad Antoniewicz described in their presentation at Schmoocon 2008, the MD5 Challenge Request and MD5 Challenge-Response can be captured both by an assailant sniffing traffic between the applicant and the authenticator. The dictionary attack for the Plaintext Password is described by Wright and Antoniewicz.

EAP-PEAP: The method of authentication consists of two phases: external and internal authentication. External authentication first happens and begins when the supplicant requests authentication from the server through the authenticator. To prove his identity to the supplicant the authenticator will then respond with an x.509 certificate. If a supplicant accepts the server authentication certificate, outer authentication is efficient and a secure tunnel between the authentication server and the supplicant is created. Then we move through the secure tunnel to the inner authentication process. A secure tunnel was developed in order to protect the inner authentication process in response to weaknesses affecting unprotected EAP methods as EAP-MD5. Like EAP itself, many different protocols are available for use in the internal authentication process. MS-CHAPv2 is, however, the most widely used authentication protocol. It is an issue with this process. Whilst mutual authentication can be applied by means of inner authentication protocols such as MS-CHAPv2, the only way to verify the identity of the server is through the x.509 certificate. If the supplicant does not refuse invalid certificates, the supplicant (and thus in certain cases the user) will be responsible for rejecting the invalid certificates the authentication server gets. Note that the EAP is not just used for 802.1x wireless encryption, but also for WPA2 wireless authentication.

EAP-TLS: RFC 5216 implemented EAP-TLS in 2008 primarily to reduce the above security problems that plagued inadequate EAP approaches such as EAP-PEAP and EAP-TTLS. It uses shared certificate-based authentication to deter hackers from executing the sort of man-in-the-middle attacks which could be used to target weaker EAP implementations during the outer authentication process. The inconvenience of the installation of the client certificate on all the supplicant devices sadly reduced the technology's overall adoption rate.

PEAP: PEAP was initially created by Microsoft to provide an extra layer of security for password negotiations for 802.1x connections as an extension of the Extensible Authentication Protocol. This blends the functionality of TLS with standard EAP authentications by using a certificate from a server-side before the credentials issued by the supplicant are authenticated. PEAP messages (32 bits long) are transported using an inferior protocol such as 802.1x or PPP, from the supplicants to the authenticators, while RADIUS ensures PEAP communication from the authenticator to the authentication system.

2.2 RADIUS

IEEE 802.1x does not require central authentication, authorizing and accounting (AAA) implementation, but, keeping in mind the size of operation in corporate networks, the backend installation of AAA and most authenticators should operate as clients of RADIUS [5]. For authentication, accounting and authorization events, RADIUS uses a AAA model as a set and uses elements called the "Attributes" to display data. The authorization shall ensure that a point of contact is matched with a pre-defined set of rules to ensure the use of the requested resource such as the network, VLAN or any service is permitted. All relevant information on the authorization decisions and information on the activities of authorized sessions will be collected by the accounting components [6].

RADIUS attributes are categorized into three main classes

- Vendor-specific attributes: usually not interoperable with other vendors.
- Industry-specific attributes: interoperable in the same sector as other manufacturers.
- Internet-specific attributes: interoperable in several industries with other manufacturers, networks and technologies.

These attributes are used to authenticate accounts, allow users to execute different services or service changes, and account for network operations by logging.

B. MSCHAPv2

Microsoft Challenge Handshake Authentication Protocol v2 as specified in RFC2759 [7] represents a method of authentication used in the RFC 3748 EAP system. Authenticated and authorized users and devices can access the network locally (wired or wired) or via Virtual Private Network (VPN) remotely. MSCHAPv2 requires a shared authentication to authenticate client and server [8].

The following describes the successful authentication mechanism:

- An EAP session is established by the supplicant and authentication server
- Both negotiate to use the EAP method, MSCHAPv2 is chosen when the authentication method is configured as PEAP.
- These endpoints try to authenticate one another by exchanging messages from MSCHAPv2 contained in lower-level protocols such as 802.1x, PPP, EAP, PEAP or RADIUS.

IV. RECOMMENDATION

Keeping in view the authentication time of 802.1x (even in worst case it was under 1 second) it is highly recommended to have port-based authentication in all corporate networks despite of the additional administrative overhead

As 802.1x overhead is not persistent, i.e., additional packets are only exchanged at the time of session establishment and renewal, 802.1x authentication does not degrade the network performance.

Therefore, a suitable reconnection time could be set at the authenticator to ensure integrity of sessions depending upon the size of the network

Weak authentication methods should not be used (especially the ones with clear text passwords). PEAP provides the strongest encryption along with MS-CHAPv2 in the "certificate on the server only" environments.

The network should be segmented into multiple subnetworks or VLANs according to the logical grouping of the computers. The Tunnel-Pvt-GroupID can be used to allocate the authenticating client to an appropriate network segment.

Once 802.1x authentication is enabled for a network, all the devices attempting to connect to the designated ports must be authenticated by the RADIUS Server. In absence of a RADIUS server, no connection can be made. It is therefore highly desirable to have some fault tolerance mechanism for the server, that can easily be achieved by implementing multiple RADIUS servers on the network that will not only provide fault tolerance but also help balancing the load of incoming 802.1x connections.

V. RESULT AND DISCUSSION

For all businesses, cloud providers and other organizations security is a major challenge in today's business networks. Conventional network security systems such as Intrusion Detection Systems (IDS), Intrusion Prevention Systems (IPS)[10][11], anti-virus, anti-spyware, and firewall-integrated anti-malware generate a lot of false positives that make computer system management tedious. A big problem with building computer systems is that unauthorized access to network resources is limited, and computer systems and Network System implementation failures are avoided. The IEEE 802.1x standard is a unique method for protecting access to resources and is suitable for wired and wireless network applications. When we speak, 802.1x can resolve problems with edge network access and network administrators can quickly detect unauthorized access from the main authentication server.

The client and switch must support this standard in order to use the existing 802.1x. This model can use existing infrastructures such as the Windows domain environment, with accounts, passwords, groups stored in Active Directory, etc.

VI. CONCLUSION

In this paper we have successfully bypassed the 802.1x security protocol which is used in most corporate networks to implement Network Access Control. This bypass is possible only if we have access to a legitimate device. In this attack we first connect our device to the network and then run the bypass script. All the network packets originating from the attackers machine will have its MAC address overwritten by the ebtable rule. This will ensure that the packets will look like its coming from the legitimate system.

The awareness script in the background ensures persistence. If the client is disconnected from the network, it will look for the nearest system on the network and modifies the script accordingly.

We conclude the paper by recommending the best practices that can be followed to improve corporate network security.

REFERENCES

1. Simon, Dan, Bernard Aboba, and Tim Moore. "IEEE 802.11 security and 802.1 X." IEEE document 802.1 (2000): 1-00.
2. McMillan, Troy. (2018). Understanding 802.1x and AAA. 10.1002/9781119549505.ch9.
3. Ryan, Gabriel. (2018). Bypassing Port Security In 2018 – Defeating MACsec and 802.1x-2010.
4. Aboba, Bernard, Larry Blunk, John Vollbrecht, James Carlson, and Henrik Levkowetz. "Extensible authentication protocol (EAP)." (2004).
5. Congdon, Paul, Bernard Aboba, Andrew Smith, Glen Zorn, and John Roese. "IEEE 802.1 X Remote Authentication Dial In User Service (RADIUS) Usage Guidelines." RFC 3580 (2003): 1-30.
6. Aboba, B., Zorn, G., & Mitton, D. (2001). RADIUS and IPv6. Internet Engineering Task Force, Request for Comment, 3162, 1-10.
7. Zorn, G., 2000. Microsoft PPP CHAP extensions, version 2. RFC 2759, January.
8. Chughtai, Farrukh, Riaz UlAmin, Abdul Sattar Malik, and Nausheen Saeed. "Performance Analysis of Microsoft Network Policy Server and FreeRADIUS Authentication Systems in 802.1 x based Secured Wired Ethernet using PEAP." INTERNATIONAL ARAB JOURNAL OF INFORMATION TECHNOLOGY 16, no. 5 (2019): 862-870.
9. Riley, Steve. "Wireless security with 802.1 x and PEAP." MCS Trustworthy Computing Services (2003).
10. D. Nidhin, I. Praveen, and Praveen, K., "Role-Based Access Control for Encrypted Data Using Vector Decomposition", in Proceedings of the International Conference on Soft Computing Systems: ICSCS 2015, Volume 2, P. L. Suresh and Panigrahi, K. Bijaya, Eds. New Delhi: Springer India, 2016, pp. 123–131.
11. Mallissery, S., Praveen, K., & Sathar, S. (2011, November). Correlation of Alerts Using Prerequisites and Consequences for Intrusion Detection. In International Conference on Computational Intelligence and Information Technology (pp. 662-666). Springer, Berlin, Heidelberg.

AUTHORS PROFILE

Vishnu V. is currently pursuing M.Tech in the TIFAC-CORE in Cyber Security from Amrita Vishwa Vidyapeetham. Currently, he is working as a red-team intern at Ernst & Young LLP

Praveen k. obtained his PhD (Cryptography) from Amrita Vishwa Vidyapeetham. Currently, he is working as an Assistant professor in the TIFAC-CORE in Cyber Security, Amrita Vishwa Vidyapeetham University, Coimbatore.