An Overview of Port-Based Network Access Control

M. Mesic
Communication Networks Department
InfoNET projekt d.o.o.
Zelinska 4, HR-10000 Zagreb, Croatia
Phone: (+385-98)463 784, E-mail: milan.mesic@infonet.hr

M. Golub
Department of Electronics, Microelectronics, Computer and Intelligent Systems
Faculty of Electrical Engineering and Computing, University of Zagreb
Unska 3, HR-10000 Zagreb, Croatia
Phone: (+385-1)6129 967, E-mail: marin.golub@fer.hr

Abstract - This paper represents a technical overview of a current state in Port-Based Network Access Control (PBNAC) for security policy enforcement at network access layer. It describes architecture and security protocols used by PBNAC. Architecture components, such as Supplicant, Authenticator and Authentication Server are discussed. The paper analyzes security protocols, specially, Extensible Authentication Protocol (EAP) variants such as EAP over LANs (EAPOL) and EAP Transport Layer Security (EAP-TLS) and additionally Remote Authentication Dial In User Service (RADIUS). The paper also shows current limitations and future improvements of PBNAC systems.

I. INTRODUCTION

Defining under which conditions two or more network entities are allowed to communicate is key element of security policies. Five categories of security incidents that generate greatest losses are viruses, unauthorized access, theft of proprietary information, denial of service and insider network abuse [1]. Weak security policy enforcement at network access layer is direct cause of three out of these five categories. In addition of directly allowing unauthorized access, theft of proprietary information and insider network abuse, it can also be treated as indirect possible cause for virus spreading, and in some cases even denial of service.

To implement security policies at network access layer, Port-Based Network Access Control (PBNAC) can be used. PBNAC uses physical port characteristics to allow network connectivity only to authenticated and authorized devices and users. Access to systems protected by PBNAC is controlled by authentication process that determines whether client accessing protected network entity is authorized to use its protected services. Client requesting a service claims its identity, which should be verified by means of authentication. By referencing the configured policies for authenticated client through process of authorization, service can be granted or denied. Although PBNAC has by now reached stage where it resolves most of problems addressed in [10] and other initial security analysis, it is still a very dynamic field.

With existence of publicly available network access layer equipment, increasing popularity of wireless networks in form of IEEE 802.11 wireless local area networks (WLANs) and emerging IEEE 802.16 WirelessMAN, need for adequate network access protection is growing. Some of security flaws in 802.11 data link protocols can be found in [9]. Examples of wired network infrastructure that needs access control include corporate network equipment with ports for connecting corporate servers and workstations. Without proper access control, employees could connect their unauthorized end systems, such as private notebooks, to corporate network. Most corporate ports are open and according to [1], security events perpetrated by insiders are about as often as by outsiders. Wireless networks without proper access control allow whole range of passive and active attacks. It is often practically impossible to dimension wireless network range to both prevent its publical availability and keep its required service levels to authorized organization members.

II. ARCHITECTURE OF PBNAC SYSTEMS

Devices considered in this article connect to other devices through points of attachment. Point of attachment to network is considered to be one logical network entity to which can connect only one other point of attachment. In case of shared media networks, like shared Ethernet LAN segment, or wireless LAN, for each network association, one point of attachment per device is created. This means that PBNAC considers point-to-point connections, on shared media networks usually enforced by means of suitable encryption.

Each point of attachment has two parallel logical entities associated with it. They are controlled port and uncontrolled port. Uncontrolled port allows exchange of network packets between network access entities regardless of the authorization state of the point of attachment. Uncontrolled port is used for PBNAC authentication and authorization purposes, thus only PBNAC protocol messages can pass through it.

Controlled port allows exchange of network packets only if the current state of the point of attachment is authorized. Controlled port is used for general network traffic. To avoid frequent use of term point of attachment through this article, just term port is used instead. Terms controlled port and uncontrolled port are used for controlled and uncontrolled entities associated with point of attachment.

Each port adopts none, one or both of two possible roles: authenticator and supplicant. Authenticator port wishes to

enforce authentication before allowing access to services that are accessible via that port. Supplicant port wishes to access the services offered by the authenticator port. One port can adopt both authenticator and supplicant role, allowing two-way authentication. One final component of PBNAC architecture is authentication server. Authentication server processes authentication credentials that authenticator receives from supplicant and indicates to authenticator whether the supplicant is authorized to connect to authenticator port. Each PBNAC component is now discussed.

A. Supplicant

Supplicant is most easily and accurately described through its state machine. PBNAC components state machines are important for analyzing their operation and limitations. Without considering special case and timeout related states, supplicant can be in following states: disconnected, connecting, authenticating and authenticated. Supplicant transitions through its states as function of following inputs: Extensible Authentication Protocol (EAP) messages from authenticator, higher layer input, user actions and time (time is used for various timeouts, and is not considered in this article).

Supplicant is in disconnected state when port is inoperable, user explicitly logs off, or system initializes. From this state supplicant transitions to connecting when port becomes operable. In connecting state supplicant attempts to connect to authenticator. If no response is received, PBNAC unaware authenticator is assumed, and supplicant transitions to authenticated state. If EAP-fail or EAP-success is already received and accepted from higher layer logic, supplicant transitions to authenticating state. In authenticating state supplicant has received EAP-request message from authenticator and higher layer logic decides to respond to authenticator and transition to authenticated state is made.

B. Authenticator

Without considering special case and timeout related states, authenticator has same states as supplicant does. Difference is in meaning of some of its states and in transitions between them as described here. Disconnected state is entered when supplicant explicitly logs off. In connecting state authenticator is ready to establish communication with a supplicant. If higher layer logic is ready to send EAP-request message, authenticator transitions to authenticating state. In authenticating state, authentication procedure is started.

In authenticating state, if excessive timeouts occur, transition to disconnected state is made. In authenticating state, if authentication server returns reject message, transition to connecting state is made. In authenticating state, if authentication server returns accept message, transition to authenticated state is made. In authenticated state authenticator has successfully authenticated the supplicant.

C. Authentication server

Although authenticator can authorize supplicants locally and allow them access to its services, authentication server

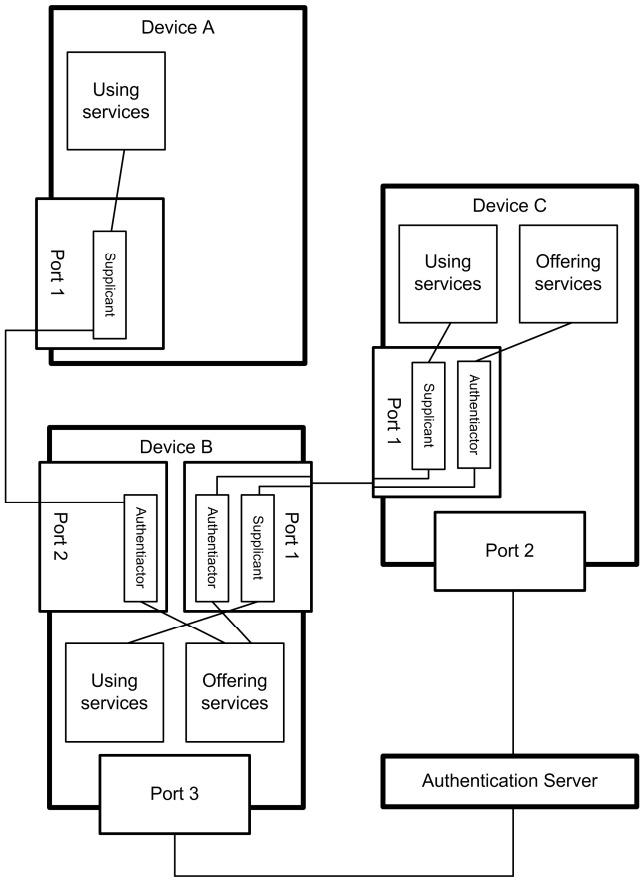

Figure 1 - Architecture of PBNAC system

is usually used. This allows greater scalability by allowing adding authenticators and supplicants without the need to configure policies on each of the authenticators. Secondly, when authentication server is used, authenticator doesn't need to understand every possible authentication protocol used by PBNAC system and can act as a conduit for relaying authentication protocol packets between supplicant and authentication server.

Authentication server usually communicates with authenticator through some higher layer protocol. This allows locating authentication server outside of the network segment where communication between authenticator and supplicant happens. For authentication server to be accessible to authenticator and vice versa, communication between them usually isn't controlled by PBNAC. When controlled ports are used for communication between authenticator and authentication server, port must be in authorized state.

III. PBNAC PROTOCOLS

Authentication in PBNAC systems consists of exchanging protocol messages between supplicant and authentication server. Extensible Authentication Protocol (EAP) is used for this communication. In this framework, authenticator can mostly be considered as a relay that accepts messages from supplicant, forwards them to authentication server, and vice versa. On receiving authentication process result messages, authenticator

grants or denies access to supplicant by opening or closing its controlled port.

Between supplicant and authenticator, EAP packets are encapsulated in EAP over LANs (EAPOL) protocol packets. Between authenticator and authentication server, EAP packets are encapsulated in Remote Authentication Dial In User Service (RADIUS) protocol packets. Each of mentioned protocols is now discussed.

A. EAP

EAP is an embedding protocol that can transport different authentication mechanisms. It typically runs directly over data link layers, and doesn't require IP. EAP provides great flexibility by allowing implementations of different authentication methods on authentication server, with authenticator that can act as a pass-through for methods implemented on authentication server. Different authentication methods can use EAP as a universal layer for transporting its authentication messages through different network devices, because of its universal implementation on those network devices. Only EAP framework needs to be implemented, with possibility that authentication server makes actual authentication decisions. Minimal requirements for authenticator are relaying EAP messages and enforcing received authentication decisions on its ports.

Authentication process starts with authenticator sending a Request packet to supplicant. There are different types of Request packets, for requesting different information from client. Possible types are Identity request, MD5-challenge, etc. Supplicant replies with Respond packet of same type as authenticator request packet. After initial Request – Respond pair of packets, authenticator may request more information from client by sending additional Request packets. This process continues this way as long as authenticator requires more information from supplicant to make its authentication decision based on authentication method implemented in EAP framework. This conversation continues until authenticator can make its authentication decision by sending either EAP Failure packet when supplicant can't be authenticated or EAP Success packet when successful authentication has occurred.

All EAP implementations must support initial authentication mechanisms as defined in [3]. These mechanisms are MD5-Challenge, One Time Password (OTP) and Generic Token Card (GTC). MD5-Challenge is analogous to the PPP CHAP protocol with MD5 as the specified algorithm for challenging supplicants. Similarly, One Time Password uses OTP challenge for authentication. GTC sends displayable message in its Request packets, and Response packets contain information read by a user from the token card device and entered as ASCII text.

In addition to initial set of authentication methods, many different open standard and proprietary EAP implementations are developed. Most often implemented are EAP-MSCHAPv2 for challenge-response based authentication, EAP-TLS for cryptographic based authentication with PKI certificates, and PEAP and EAP-TTLS for tunneling based authentication with tunnel protected EAP communication.

TABLE I
INITIAL SET OF EAP MESSAGE TYPES

Code	Message	Type	Type name
1	Request	1	Identity Request
		2	Notification
		4	MD5-Challenge
		5	One Time Password (OTP)
		6	Generic Token Card (GTC)
		254	Expanded Types
		255	Experimental use
2	Response	1	Identity Response
		2	Notification
		3	Nak
		4	MD5-Challenge Response
		5	One Time Password (OTP)
		6	Generic Token Card (GTC)
		254	Expanded Types
		255	Experimental use
3	Success	-	-
4	Failure	-	-

EAP authentication methods are subject to different security threats, and should include methods for mitigating those threats where required. When used on wireless LAN networks and over the Internet, but also on links, media and devices with possibility of attacker gaining access to authentication traffic, user identities should be protected when required. Although Identity Request and Response messages are included in initial set of EAP messages, they are optional, and actual identity exchange can be realized over protected channel established according to specific method. To avoid man in the middle attacks where a rogue authenticator forwards authentication messages between supplicant and legitimate server, mutual authentication should be used, and also cryptographic binding between the tunneling protocol and tunneled authentication method. Similar methods can also be used when supplicant connects to untrusted network with possibility of connecting to a rogue device. To protect against modification of authentication packets, integrity and replay protection are recommended. This is especially important for result indication packets, which are without adequate integrity and replay protection easily spoofed. Password authentication algorithms such as EAP-MD5 and similar are vulnerable to dictionary attacks, and when avoiding of this attacks is required, dictionary attack resistant methods are preferred.

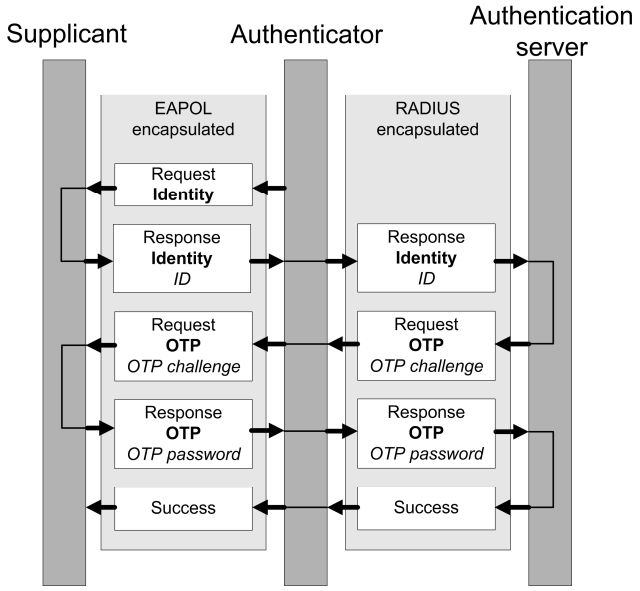

Figure 2 – Possible EAP communication

B. EAPOL

EAP over LANs (EAPOL) is encapsulation that carries EAP packets between supplicant and authenticator. EAPOL packet begins with Ethernet type and protocol version fields according to standard assigned numbers. Following these fields, EAPOL specific information is included. There are five different EAPOL packet types. In addition to EAPOL EAP packet that actually caries encapsulated EAP authentication, there are additional EAPOL packet types that carry specific signaling and keying information. To initiate EAP authentication and to terminate authenticate session, supplicant uses EAPOL-Start and EAPOL-Logoff packets respectively. To send specific SNMP traps as Alerting Standards Forum (ASF) alert, EAPOL-Encapsulated-ASF-Alert is used. Finally, EAPOL-Key packet type allows transmission of key information between the authenticator and the supplicant. Initially, as [2] defines, two key descriptor types are used: RC4 and IEEE 802.11.

Most security problems that can be related to EAPOL can be, and usually are solved by using adequate authentication method in EAP layer. However there are some implementation considerations inherently related to EAPOL.

When there is a possibility that more than one supplicant can access one authenticator's port, protection should be provided for avoiding case where unauthenticated supplicant uses open port based on connectivity granted to another authenticated supplicant. This can usually be achieved with cryptographic separation of each association between supplicant and authenticator.

Supplicant may try to send EAP messages with multicast and broadcast destination address, which could interfere with authentications occurring on other ports or segments. To prevent this, EAP messages with destination address other than authenticator port to which supplicant is connecting should be discarded, and non routable.

C. RADIUS

Remote Authentication Dial In User Service (RADIUS) is an authentication, authorization and accounting protocol that can be used in EAP authentication framework through its support for EAP as defined in [4] to forward EAP packets to and from the authentication server with implemented RADIUS server. To achieve effective PBNAC implementation, guidelines from [6] should be followed. RADIUS messages are called attributes and are comprised of variable length Type-Length-Value 3-tuples. EAP messages are encapsulated within EAP-Message RADIUS attribute, allowing flexibility of avoiding need of implementing separate RADIUS attribute for each authentication method.

On receiving EAP Response message from supplicant, authenticator may authenticate supplicant locally, or act as a pass-through and encapsulate EAP Response message into EAP-Message attribute of Access-Request RADIUS packet. Authentication conversation can continue this way with EAP messages traveling between authenticator and authentication server encapsulated within EAP-Message attribute of RADIUS Access-Request, Access-Challenge, Access-Accept and Access-Reject packets. In addition to EAP-Message attribute, Message-Authenticator attribute must also be used to provide authentication and integrity protection of RADIUS packets. For this purpose HMAC-MD5 algorithm is used.

To mitigate most of security vulnerabilities associated with use of RADIUS for encapsulating EAP messages, IPsec should be used. In original RADIUS specification [5] a shared secret was defined for hiding attributes, and for authentication computation. This method is not sufficient for security vulnerabilities to which RADIUS is subject to. IPsec covers all functions of shared secret along to many other security issues. IPsec resolves privacy issues, spoofing and hijacking, dictionary attacks (to which shared secret is especially vulnerable), known plaintext attacks and replay attacks. Man in the middle attacks can't be completely mitigated, as within RADIUS, security can only be provided on a hop-by hop basis, even when IPsec is used. To protect against man in the middle attacks, specific EAP methods should provide their own per-packet protection and authentication mechanisms for end-to-end protection.

D. EAP-TLS

EAP Transport Level Security (TLS) defined in [7] is often implemented and robust EAP authentication method that provides mutual authentication, integrity-protected ciphersuite negotiation and key exchange between two endpoints. TLS protocol defined in [8] is composed of two layers: TLS record protocol and TLS handshake protocol. TLS record protocol provides private and reliable encapsulation for TLS handshake protocol that allows supplicant and authentication server to authenticate each other and to negotiate an encryption algorithm and cryptographic keys for application protocol. TLS handshake protocol allows mutual authentication by using asymmetric cryptography. Shared secret negotiation is unavailable to eavesdroppers, and secure from man in the middle attacks. When supplicant and authentication server first start communicating, they agree on a protocol version,

select cryptographic algorithms, optionally authenticate each other, and use public-key encryption techniques to generate shared secrets. Both sides should be implemented in a way that handshake protocol never selects algorithms or key sizes that are not compliant with adequate security policies related to devices in considered system.

Actual goals of TLS handshake protocol are achieved starting with hello messages that establish security enhancement capabilities and key exchange between supplicant and authentication server. Following this hello messages certificates are exchanged (mutually or one-way) and with cipher specification messages ciphersuite is negotiated and agreed.

When using EAP-TLS authentication method for authenticating supplicants that initially don't have network connectivity, and get network connectivity only after successful authentication, problem of certificate authority (CA) certificate revocation lists reachability can arise. In this case, certificate revocation lists should be checked after connecting to the network.

IV. PBNAC IMPLEMENTATION

We have implemented PBNAC system for authenticating members of Microsoft Windows domain for one of our enterprise customers. System was implemented in a way to allow high availability of authentication services, with all critical components working in hot standby mode. Cisco Catalyst switches were used as authenticators, authenticating connected Microsoft Windows domain computers and users through 802.1x EAP encapsulation framework. Catalyst switches communicated with Cisco Secure Access Control Servers (ACS) used as authentication servers, through RADIUS encapsulated EAP messages. Two ACSs were used to achieve high availability. We added additional scalability element with Microsoft Active Directory domain controllers used as integrated identity stores. ACSs and domain controllers communicated through Lightweight Directory Access Protocol (LDAP). As EAP-TLS authentication protocol was used in our PBNAC framework, Certification Authority (CA) was needed. Microsoft CA configured as enterprise root CA was used for this purpose.

On Microsoft Windows domain additional authentication phase was needed. By blocking network access prior to port authentication, 802.1x breaks the machine-based group policy model. What is needed is the ability of a Windows workstation to authenticate itself, under its own identity, independent of the requirement for an interactive user session. This is achieved with machine authentication, used at boot time by Windows operating systems to authenticate and communicate with Windows domain controllers in order to pull down machine group policies. After establishing communication with domain controller and pulling down machine group policies, user can log on to domain, and finally through second phase PBNAC authentication, authenticate himself as user.

Microsoft Active Directory and CA inherently deal with machine certificates in a way that doesn't allow using EAP-TLS for machine authentication. User certificates are written to Active Directory while enrolling, allowing EAP-TLS to be easily employed for user authentication.

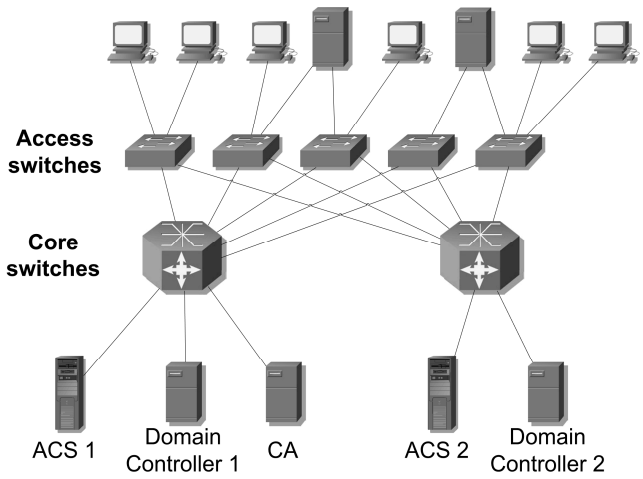

Figure 3 – Network of implemented PBNAC elements

Machine certificates are just deployed on supplicants, in their system store, without saving them to Active Directory. This doesn't allow ACS to use EAP-TLS methods for authenticating machines. To mitigate this problem, special certificate templates were designed, and used to add additional capabilities not inherent to machine certificates. This allowed writing machine certificates to Active Directory, and ACS could use them as elements of identity store for machine authentication, together with inherently possible EAP-TLS user authentication.

V. CONCLUSION

This paper presents an overview of current state of Port Based Network Access Control and analyzes it from security and implementation perspectives. Although cryptographic assumptions used to implement PBNAC exist for many years, and provide powerful tool for achieving ultimate network access control, there are still open problems. Security incidents continue to generate great losses, with network access control flaws still among categories on top of the list.

Cryptographic assumptions are highly unlikely cause for breaking network access control. Increasing complexity of protected and protection systems, with many components linked through many interfaces open possibility of breaking complete network access control system by breaking its inadequately implemented part. This makes complete PBNAC systems often as weak as its weakest component or weakest link between components.

In addition to implementation security, with standardization of PBNAC systems as its important part, there are still possible improvements. To provide greater flexibility, supplicants may be selectively granted access to particular network resources according to a more detailed security policy, this way extending simple grant/deny authentication decision. Network ports are also very convenient for controlling other security policy elements. In addition to authentication based authorization decisions, security policy compliance of supplicants can also be enforced through PBNAC systems.

Our implementation used five different categories of components, with greatest diameter of four different protocol hops. Described problems indicate that non-standardization still exist in field of PBNAC systems. Although most interfacing protocols are well defined and standardized, while combining them in larger chains as in our implementation, certain incompatibilities may arise.

As we have shown, to achieve certain functionalities of PBNAC, considerable customization efforts are required. Although this can be tolerated for more exotic and rarely implemented authentication methods, for general purpose authentication methods, system wide perspective of PBNAC should be considered, allowing easier implementations with fewer points of possible implementation failures that could lead to security flaws. Further work should focus on interoperability and functionality issues. Complete PBNAC authentication framework should be considered as one system containing many elements. Interfaces between these elements are well standardized, but functionality of every element should be considered from system wide perspective with PBNAC in focus.

REFERENCES

[1] L. A. Gordon, M. P. Loeb, W. Lucyshyn, and R. Richardson, *2005 CSI/FBI Computer Crime and Security Survey*, 10th annual, CSI, 2005.

[2] LAN/MAN Standards Committee of the IEEE Computer Society, *IEEE Std 802.1X™- 2004*, IEEE, New York, 2005.

[3] Network Working Group of The Internet Society, *Request for Comments: 3748, Extensible Authentication Protocol (EAP)*, The Internet Society, 2004.

[4] Network Working Group of The Internet Society, *Request for Comments: 3579, RADIUS (Remote Authentication Dial In User Service) Support For Extensible Authentication Protocol (EAP)* , The Internet Society, 2003.

[5] Network Working Group of The Internet Society, *Request for Comments: 2865, Remote Authentication Dial In User Service (RADIUS)* , The Internet Society, 2000.

[6] Network Working Group of The Internet Society, *Request for Comments: 3580, IEEE 802.1X Remote Authentication Dial In User Service (RADIUS) Usage Guidelines*, The Internet Society, 2003.

[7] Network Working Group of The Internet Society, *Request for Comments: 2716, PPP EAP TLS Authentication Protocol*, The Internet Society, 1999.

[8] Network Working Group of The Internet Society, *Request for Comments: 2246, The TLS Protocol Version 1.0*, The Internet Society, 1999.

[9] N. Cam-Winget, R. Housley, D. Wagner, J. Walker, *Security Flaws in 802.11 Data Link Protocols*, Communications of the ACM, 2003.

[10] A. Mishra, W. A. Arbaugh, *An Initial Security Analysis of the IEEE 802.1X Standard*, 2002.