ANALYSIS OF SECURITY PROTOCOLS FOR AUTHENTICATION IN DISTRIBUTED SYSTEMS

H. A. Eneh
School of Computing Science, Middlesex University, White Hart Lane, London N17 8HR

O. Gemikonakli
School of Computing Science, Middlesex University, White Hart Lane, London N17 8HR

ABSTRACT
Research works on the analysis and the development of authentication protocols frequently adopt the linear software development approach and are based on certain non-extensible assumptions (Tobler and Hutchison 2004), (Grob 2003), (Harbitter and Menasce 2002). This paper is part of an ongoing work regarding the development of a pre-emptive security mechanism for networks and distributed systems. We present in this paper a cyclic analytical method of protocol development for authentication solutions, which can be used by both designers and security administrators. Our analytical approach can be extended to a variety of protocols for either closed or open networks. The method lends itself to the analysis and therefore to the development and/or adoption of authentication protocols that can render computing resources unassailable to attackers. The motivation for this approach stems from the fact that despite the multiplicity of highly published protocols for authentication, an amazing number of attacks continue to evolve against authentication solutions, and thereby continue to hinder the placement of trust on networked computing resources.

KEYWORDS
Authentication protocols, distributed systems, protocol analysis, system development

1. INTRODUCTION
Several authentication utilities have been developed over the years to deal with application-level authentication and digital signatures. Authentication has been described severally in literature (Needham and Schroeder 1978), (Burrows et al 1990), (Lowe 1997) and is acceptably defined as the reliable means of identity verification. Computer networks and distributed systems rely on the paradigm of authentication for the purposes of attack prevention. Currently, cyber crimes, attack on computing resources, which hinder the placement of trust on computing elements, continue to grow exponentially in form and complexity. In order that the development of attacks will not spiral out of control, research and industry are beginning to grow attention in the direction of automatic generation of suitable security mechanisms to deal with the rapidly evolving attacks. These measures include the recent developments in automatic protocol generators (Tobler and Hutchison 2004) as well as the developments in automatic protocol verification (Meadows 1992), (Cervesato et al 1999), (Rubin and Honeyman 1993), (Heintze and Tygar 1996), (Harbitter and Menasce 2002) to mention but a few. So far, emphasis is on the development of protocols matching their exact specification. Basically, the protocols are to serve the purposes of secure communications within networks and distributed systems.

We adopt, in our project, the notion of access control model used by (Lampson et al 1992) to describe the elements of computer security. The elements are the Principals, Requests, Reference monitor, and objects, otherwise known as resources. A functional relationship exists within these elements such that a principal initiates a request to a resource, irrespective of how this request is made, the reference monitor, guard for the
objects, examines each request and decides whether or not to grant it. To be effective, the reference monitor deserves a reliable means of identifying the source of the request, authentication, and the determination of the access rule, authorisation.

A distributed system is characterised by a number of factors such as autonomy, size, heterogeneity, and fault tolerance (Lampson 1992), (Stallings 2002). Autonomy is the implication that a request may arise from a remote principal such that the distance between the principal and object is occupied by several internetwork nodes that may not be equally trusted and may also belong to different administrative domains. The size basically is a function of the number and complexity of the clients and servers. The distributed system can be larger than a centralised one with distributed servers in a rather open environment than a closed network environment. Heterogeneity is fundamental to the variety of security provisions of different domains that constitute the channel between a principal and the object. With fault-tolerance, a distributed system is expected to continue delivering services, requirement of availability, even if a component becomes inaccessible. These characteristics of distributed systems thus present a requirement that a user operating a client system proves his identity for each service desired and a superfluous requirement that servers prove their identity to clients. This scenario is the motivation for authentication mechanisms such as Kerberos and SAML whose standards are undergoing refinements to achieve acceptability (Neuman and Ts’o 1994), (OASIS 2004), (Mishra 2002), (Korman and Rubin 2000).

In this paper, we present a cyclic method of authentication protocol analysis and design which can be used by designers and security administrators alike for analysing such authentication protocol standards and specifications. Owing to the limitations of space, this paper is succinct but we hope to render a greater detail in our technical report due to ready soon. The balance of this paper is organised in sequence as follows; related work, a method for analysis of protocols, discussions, further work, conclusion, and references.

2. RELATED WORK

Most of the existing works on the analysis and design of authentication protocols follow the basic principle of software design. (Tobler and Hutchison 2004) attempted to automatically generate protocol implementations right from their exact specifications as a means of facilitating protocol analysis. Such analysis includes those comparing protocols’ performance and scalability testing. However, (Tobler and Hutchison 2004) limited their emphasis on implementation and its verification. As a matter of basic principle, the development processes are;

a) Requirements: this is the process of ascertaining and classifying needs, which include those of authentication, confidentiality, integrity etc.

b) Design and implementation: the process of designing a protocol that meets the stated requirements.

c) Protocol specification analysis: inference logics like the BAN or the GNY are used for the analysis of the protocols’ specifications to determine if the desired security properties hold.

d) Implementation: this deals with generating code and the corresponding executable program that conforms to the specification.

e) Implementation and verification: this is the method of proving that the generated code is a refinement of the security protocol’s specification.

(Tobler and Hutchison 2004) followed the above steps in what we describe as a linear method. Though this approach works well for software systems, we reckon that security protocols deserve more than simplified linear method. (Harbitter and Menasce 2002) used closed networks to compare the performance of variants of Kerberos, using a public key infrastructure or adding a proxy server. We are inclined to agree with the assertions of (Lowe 1997) and (GroB 2004) pertaining respectively to the need for precise authentication definition and further analysis of even highly published protocols. (Lowe 1997) outlined the significance of adopting appropriate definition for authentication prior to use in order to avoid misplacement of trust in certain authentication protocols. (Lowe 1997) asserted that the strength of the authentication protocols range from fundamentally satisfying the requirement of aliveness to a more secured assurance of full agreement with recentness. Due to the feasibility of attacks such as connection hijacking/replay, man-in-the-middle, and HTTP referrer attack (GroB 2004) stated that further analysis of the SAML protocol is still very necessary despite the fact that SAML single sign-on protocol is well designed and carefully described.
3. A METHOD FOR ANALYSIS OF PROTOCOLS

Despite the multiplicity of authentication approaches and proposals for improving proactive security of networks and distributed systems, threats of penetration and other forms of attacks have continued to evolve, increasing in number and complexity. The need for proactive authentication mechanisms coupled with the need for prudent deployment of authentication standards is the motivation for the analytical approach presented here. The underlying philosophy is towards the adoption of a method that can assure that resulting authentication protocols are correct and consistent with their specifications and additionally, to guarantee freshness, completeness, flexibility, and user friendliness of authentication solutions.

As indicated in figure 1, the method consists of some of the fundamental phases of system development (Tobler and Hutchison 2004) but in this case, they comprise of more complex stages that are arranged in a cyclic pattern. The motive is to assure automatic continuity of the development cycle such that a fresh cycle is triggered as soon as variation(s) appear in the specifications. Specification variation occurs whenever there is a perceived failure in a protocol or there exists a change, different from some norm, in the internal state of authenticating principals as they participate in the protocol.

Figure 1 shows that the main phases are; specification, refinement, design, protocol specification analysis, implementation, and implementation verification. The epicenter of this particular procedure being introduced here is the attack and attack analysis as the main component of the protocol specification analysis phase. As this design also deals with the automatic generation of correct protocols, concerns are also raised for the automatic generation of attacks to test the strength of resulting protocols. Notably, attacks can be internal, resulting from within the system or external, external penetration, industry changes in standards, or other system wide changes. The main phases are briefly described as follows;

a) Specification: this phase may be regarded as the initial phase. Specification phase deals concurrently with requirement analysis (authentication, confidentiality, integrity etc) and term-writing of the specification for appropriate security protocol to meet the requirements of the network or distributed system concerned vis-a-vis specification of trust and trust levels.
b) Refinement: this phase deals with fine-tuning of the specification to remove ambiguities such as granting and denying a transaction, a protocol run, at the same time, and other statements without precise definitions.

c) Design: or otherwise the design with specification, this phase deals with the actual design or adoption, as the case may be, of the protocol that matches the specification.

d) Protocol specification analysis: this phase is the core of this analytical method. It deals with all forms of automatic or manual discovery of attacks against the designed protocol for the authentication mechanism to be deployed. The attack analysis can employ either or a combination of state space techniques (Lowe 1997) and knowledge and belief logics (Mao and Boyd 1993). The moment an attack is discovered, the implementation and implementation verification stages follow.

e) Implementation: this phase deals with the automatic generation of code matching resulting protocol’s specification which is compiled to produce the binary form of the protocol.

f) Implementation verification: this phase is used for confirmation purposes. It is used to ensure that all the desirable properties inherent in the specification hold. If a deviation from the norm is observed, a fresh specification is written and the rest of the cycle continues else, the resulting protocol is adopted as the working specification.

4. DISCUSSIONS

The method presented here is an improvement on other existing development methods found in literature. The method is adaptable by designers or administrators comparing the effectiveness of existing authentication solutions and standards. The designers can employ the cycles for development of authentication protocols from the scratch while security administrators can use the method for the analysis of solutions.

Some concerns about this approach rapidly emerge. At the moment, these include issues of performance trade-offs, ease of integration, and re-cycling periods. The performance issues are those of the additional processing requirement of the attack constructions and variations, and those of scheduling the entire processes. It is expected that the recent developments in processor speeds should be able to bring the processing times, delay, to a bearable minimum. Concerning integration, systems interfaces are to be designed for efficient and seamless integration of this solution into existing authentication platforms. The system re-cycle time, which supposedly deals with the time for restarting the entire procedure, should be left as a matter of discretion of the designers or administrators who should use the nature of risks and the frequency of threats to determine the suitable frequency for the cycles.

5. FURTHER WORK

Further work includes determination of the attack analysis method that is most suitable for each type of authentication mechanism. An empirical analysis of the existing tools conforming to state space analysis and knowledge and belief logics will be conducted against the background of the latest variants of Kerberos and web based protocols such as SAML to ascertain correctness and appropriateness. A guide to the development of the necessary interfaces is to be rendered.

6. CONCLUSION

We have presented in this paper a method for continuous analysis, design, and verification of authentication protocols that can serve both open and closed networks and distributed systems. Owing to the fact that even the highly secured networks and computing resources remain vulnerable due to evolution of attacks; it becomes highly necessary to adopt methods that can render the positions of networks and computing resources unassailable to attackers. Our cyclic analytical method offers an improvement to the method used
Our method involves development cycles with special attribute of continuity in such a manner that additionally provides for freshness, correctness, and flexibility of protocols with respect to their specifications. Though we have made mention of trade-offs (in section 4), we strongly believe that the method, which guarantees seamless continuity of developmental phases of authentication protocols, will contribute immensely to improving strength of resulting mechanisms. Emphasis is particularly on attack analysis as it is believed that in-house discovery and treatment of attacks will definitely enable authentication mechanisms to be pre-emptive in addition to better security.

REFERENCES


Hodges, J. and Wason, T., Liberty architecture overview, [http://www.projectliberty.org/specs/](http://www.projectliberty.org/specs/)


