Active Protection of Trusted Security Services

Alec Yasinsac
Department of Computer Science
Florida State University
Tallahassee, FL 32306-4530
850.644.6407
Yasinsac@cs.fsu.edu

Abstract

Secure electronic communication relies on the application of cryptography. Recently, there has been an explosion in the growth of Public Key Infrastructure technology, where centralized or partially centralized services provide addresses and keys for participants desiring to establish secure channels with one another. We consider the traffic to and from these servers to be the foundation of the "critical infrastructure" for secure communication. These trusted services are worthy targets for intruders since successful intrusion would have wide-ranging impact. In this paper, we offer a method to detect anomalies in this traffic based on techniques proven in intrusion detection research and products and suggest proper autonomic responses to the anomalies that we can detect.

Section 1. Introduction.

Secure electronic communication relies on the application of cryptography. Recently, there has been an explosion in the growth of Public Key Infrastructure technology, where centralized or partially centralized services provide addresses and keys for participants desiring to establish secure channels with one another. These trusted services are worthy targets for intruders since successful intrusion would have wide-ranging impact. If a central trusted service can be compromised, it might be possible to use that service as an Oracle, to compromise communications between any two participants using that service or to masquerade as any participant with virtually no trace.

The security of the information provided by these services is dependent on security protocols. Extensive work has been done to test [MCF87] and verify [KMM93] security protocols, and significant progress has been made in these areas. Nonetheless, no method provides complete, or even measurable, confidence in security protocols. In fact, based on the nature of security protocols and their environment, it may be impossible to accurately predict their performance through formal analysis or automated testing. In [KSW98] the authors show how attacks can be constructed through interaction of two simultaneously executing protocols, even though both protocols are "secure" when run independently.

The work on verification of security protocols has been largely theoretical, since cryptographic systems are few in comparison to the overall scope of the Internet. Conversely, Intrusion Detection System (IDS) research has been highly successful in meeting practical, low assurance security requirements in an Internet environment. IDS techniques focus on characteristics of "normal" network traffic, and "normal" user behavior as identified through network and host

activities. Historical data is tracked and modeled by statistical measures providing a baseline to compare new activities against. Two examples of the data considered by IDSs are:

1. The commands a user routinely issues to a host and

2. The type of traffic generated by an application on the network (identified by the Internet Protocol (IP) port).

More recently, efforts have been made to extend intrusion detection techniques to a slightly different environment. In [JOU00], Jou, Gong, et al. show how network protocol traffic can be analyzed to protect the network routing infrastructure.

In the same way, we believe that established intrusion detection techniques apply to active analysis of the environment that surrounds trusted services in high assurance systems. In this paper, we provide a framework of how IDS technology can be applied to the security protocol environment, and extend this work by categorizing attacks in order to determine appropriate responses to detected anomalies and intrusions. While there has been intense research in security protocol analysis and verification and equal emphasis on research in intrusion detection, the two fields have not been considered together. We propose to combine these technologies to create a tool that will automatically detect attacks on trusted security services through identification of anomalies in the security protocol traffic. This tool will also characterize the potential intrusion attempt in order to suggest a proper action to take in response to the noted anomaly. It could be employed as a stand-alone monitor-response tool, or integrated into a network defense tool set such as P-Best [LP99].

The cumulative result is that this research will provide a mechanism for active defense and response to attack for security services. The mechanism will ensure reliable, effective performance of critical security services and will prevent sophisticated attackers from utilizing or masquerading as these security services.

Section 2. Security Protocol Verification

Security protocols are intended to provide secure services. Most often, these services entail establishing a secure channel between two communicating principals. Unfortunately, security protocols are subject to flaws that are not easy to detect. If the protocols underlying the secure channel are flawed, then the security objectives of the participants are undermined, possibly at great financial, physical, or other risk to the participants. [NS78] first highlighted vulnerabilities of security protocols, and a mountain of research has been conducted aimed at ensuring the effectiveness of these security essentials.

Such attempts have generally fallen into one of five categories:

1. Testing tools designed to reduce the search space of possible errors
2. Epistemic logics
3. Algebraic-based formal methods designed to reduce the search space for bad states
4. Proof systems based on a formal semantic
5. Model Checking

Section 3. The Operating Model

3.1. Definitions.

We consider the vulnerabilities to centralized security services to occur primarily through protocol sessions. Terminology applicable to secure channels is often confused with that associated with normal network traffic. Here, we present a few definitions to distinguish our discussion of messages, sessions, etc.

"Principals" are participants in a protocol session, distinguishable by a unique identifier. Our focus is on special principals that we refer to as "security servers", "trusted servers", and at times "Key Distribution Centers". The common characteristic of each of these three terms is that they represent principals that provide a security service to other principals, and are, thus, integral components of the security infrastructure, with significant impact on the security for those principals that they serve.

The messages that we are interested in are only those used in a protocol session. Anonymous messages, network overhead, and other traffic not associated with security protocols for trusted services are not messages in our sense. To distinguish the traffic of interest to us, we define a "message" as a tuple of at least four elements:

1. The identifier of a principal representing the source (originator) of the message
2. The identifier of a principal representing the destination (recipient) of the message
3. The message payload, which may comprise one or more data elements to be conveyed from the source to the destination.
4. A protocol identifier

A "protocol" is a fixed sequence of messages predefined to principals that either originate or receive the message(s). Each principal may recognize and utilize multiple protocols. A principal "recognizes" a protocol if the protocol is stored in the principal's private memory. Principal Alice utilizes protocol P if one protocol session exists or existed where Alice is either an originator or recipient of a message with the protocol identification field of P.

A "protocol session" is an instantiation of a protocol. Thus, it is a set of messages that correspond to the form of a protocol, where the generic source and destination identifiers are replaced with actual identifiers and an actual payload replaces the generic payload. Notice that every message that meets our definition is a message of a protocol session. We recognize that there will be many non-protocol transmissions on any network, but for our purposes, we ignore transmissions that are not messages by our definition.

3.2. Traces.

It is normal practice to specify protocols as an execution trace [YW93] of actions between principals, with each principal "taking turns". These protocols are listed as though the messages are executed sequentially, on a single processor, when they are intended for concurrent execution, in a distributed environment. In practice, any principal may be executing multiple protocols concurrently. In this case, a "trace" is the set of all messages executed by a principal. A trace may be thought of as an "interleaving" of protocol steps as described in [SYV94] and [KG91], meaning the execution of the steps of two different protocol sessions are intermixed.

It is important to distinguish between the symbolic execution of a protocol and an actual execution. For existing methods of protocol verification, symbolic protocol execution is examined. That is, protocols are encoded with generic values and readable, symbolic identifiers to facilitate reasoning about the results of the protocols. It is this symbolic version of the protocol that is recognized (as defined above) by principals. Conversely, when protocols are executing, they contain actual data that are not routinely readable to a human. For example, random numbers are largely unrecognizable to the human eye. Fortunately, protocols are executed on computers that can recognize random numbers and other protocol components, and can match the actual messages to their symbolic counterparts, even when executing multiple, complex protocols in a highly concurrent environment.

An actual protocol trace is the accumulation of actual messages sent and received by a principal in the order that they were sent and received. For a protocol session running alone, the trace is simply the listing of the messages in the protocol. If multiple protocols (or multiple instances of the same protocol) are executing concurrently, the trace will be extensively interleaved. This is a common occurrence in networking; where any large host may be concurrently executing requests from many different sessions for logon, file access, computations, etc. It is this type of trace that present intrusion detection technology targets.

3.3. The Man-in-the-Middle.

The Man in the Middle model has been around for a long time. In [DY83] and later [AT91], the authors formally define such an environment, where every message in the communication system must pass through a powerful intruder. We target this model and the extensions from [YAS96], because it is a powerful model, and it is easy to understand. Some of the characteristics of this model are that:

- All messages to/from every principal (including the KDC) are to/from a single party (i.e. the intruder)
- Principals operate in their own "address space". The only ways that personal memory changes is by receiving new information through the network, or by performing computation on information that is already stored. This personal memory represents the local state of the principal.
- The only knowledge that a principal can acquire about other principals is through the network.

3.4. Challenges of Man in the Middle.

It is clear that assuming such a powerful adversary presents challenges to our ability to detect and respond to intrusions. Among the consequences of our choosing the model are the following specifics.

a. The Intruder knows much more than we do. Unless we make assumptions about the underlying communications [e.g. that we use a broadcast medium] or inject a distributed information gathering mechanism such as roving agents, we cannot see all messages. In fact, we assume to see only a very limited subset of the communications that may be used to generate an attack against us, to the extent that a KDC can rarely expect to see all (or even most) messages exchanged in a protocol. On the other hand, the intruder is assumed to see (and have immediate access) to ALL messages on the network.

b. We are limited by realistic resource assumptions. In order to provide useful information about intrusions, even more critical for response, we must consider response time as a limiting factor for our methods. Conversely, because we assume the intruder is "very powerful", they may conduct resource intense activities, such as gathering information over a long period of time and conducting fast searches over extensive databases in real time.

Section 4. Profiles.

Intrusion has been a fertile research area since the mid-1980's and continues to be a topic of intense research [DS99], [VK99], [LP99]. A principal technique used in intrusion detection has been profiling, detailed in Denning's seminal paper [DEN86]. We will show how monitoring activity to detect and respond to attacks mirrors the environment that Denning addresses.

The problem is deeper than simply detecting all attacks, since we could meet this challenge by simply signaling "possible attack" for every protocol action taken. Of course this would provide no useful functionality, since we could not take effective action based on that feedback. With that in mind, we recognize two measures of our effectiveness as false negatives and false positives.

1. Do we detect all attacks (false negative)? If not, what percent of attacks will we detect? We call this metric: "percentage of attacks detected". While we can empirically analyze our system utilizing this metric, it is only useful in a laboratory environment, since in an actual environment we cannot know how many attacks that we do not detect.
2. Do we detect as attacks activities that are not attacks (false positives). If so, what percent of activities will produce false alarms? This metric is termed: "ratio of false alarms to activities". This important metric can be analyzed in a laboratory and in production use. We must have the goal of keeping this metric as low as possible to ensure that appropriate actions are taken when attacks are detected.

Profiling essentially means recording observed activity of a principal over time and producing a data structure that reflects "normal" activity of that principal. This data structure is called a profile. The fields in the profile contain data that models the activity in some predefined way. We will select models that allow us to accomplish the two goals that we just laid out, of detecting a high percentage of attacks and of producing a low percentage of false alarms.

4.1. [1]Behavior as an Attack Indicator.

Intrusion detection is focused on the behavior of communicating principals. The assumption is that, while it may be disguised, a principal's behavior will reflect their intentions. There are two fundamental behaviors that are used to identify potential intrusions:

---

[1] Throughout this paper, we use the terms "behavior" and "activity" almost interchangeably. Behavior is most often used to reflect the observed activity, as well as some intention that preceded the action. Most often, behavior refers to activities that are abnormal rather to that which is inherently dangerous.

1. That which has been previously shown to result in compromise
2. That which significantly deviates from the norm

4.1.1. Signatures.

In the first category we include characteristics of known attacks on cryptographic protocols as well as intuitively dangerous behavior. These attacks may be characterized by sequences of activity traces, similar to methods for virus scanning and for network intrusion detection [CDEKS96]. The pattern of these sequences produces a "signature" for the known attack. Traces that match these signatures are always suspect, and in some cases may be enough evidence to affect a protective or damage control response in and of themselves, with no corroboration necessary. An example, given in the same reference is that any program that sets UID during execution should be flagged as a high risk. Another example of an activity pattern that is always suspect given by Denning [DEN86] is a high rate of password failures by any user.

Since the famous attack on the Needham and Schroeder protocol in [DS81], uncounted attacks have been documented on contrived and production protocols. Syverson produced a taxonomy of protocol attacks [SYV94] that may allow abstract construction of signatures for intrusion detection through protocols in much the way that Spafford's taxonomy [KS95] of network intrusions provides a framework for identifying signatures for network attacks in IDIOT.

Examples of dangerous behavior in a security protocol environment include:

- Simultaneous triangular sessions (A->B, B->C & C->A).
- Sequential triangular sessions (A->B, B->C & C->A)
- Request to encrypt with a public key followed by a request to sign with the same key.
- Simultaneous Group Protocols
- Failed protocol sessions
- Suspended or partially completed protocol sessions
- Repetitive use of one cryptographic key

4.1.2. Profiling for Abnormal Behavior.

The second category of behavior that we are interested in is anomalous activities. If we assume that intrusions are not routinely accomplished, then it is reasonable to infer that abnormal behavior is more likely to be an intrusion than is normal behavior. Denning states it this way in [DEN86]:

> The model [for the use of profiling] is based on the hypothesis that exploitation of a system's vulnerabilities involves abnormal use of the system; therefore, security violations could be detected from abnormal patterns of system usage.

If we accept this premise, then we can reduce our problem of detecting intrusions to one of detecting abnormal behavior. First, we must categorize behavior in order to be able to distinguish that which is normal from abnormal. For a shared computer environment, Denning categorizes behavior based on activities on objects, where the objects are resources ("…files,

programs, messages, records, terminals, printers, and user- or program-created structures."[DEN86]).

Our view of intrusions is based more exclusively on activities; specifically, activities carried out through security protocols. We expect that, after sufficient data has been gathered to reduce the impact skew and after usage has stabilized after initial system usage, normal behavior will be recorded. Thus, we can characterize the behavior of each principal (Alice) based on measurable criteria such as:

4.1.2.1. Which protocols has Alice utilized by originating the first message? A legitimate principal (or an intruder that has compromised a legitimate principal) that initiates a protocol that they do not normally use may indicate an attempt to generate data that may allow an attack.

4.1.2.2. Which protocols has Alice utilized as recipient of the first message? A legitimate principal (Alice) that receives an unusual request for service may indicate that the originator is making an attempt to utilize Alice as an oracle.

4.1.2.3. How frequently does Alice utilize each protocol as originator of the first message? An increase in frequency of use of a protocol could reflect an attempt to generate a value in a data field necessary to construct an attack.

4.1.2.4. How frequently does Alice utilize each protocol as recipient of the first message? A legitimate principal (Alice) that receives an unusual number of requests for service may indicate that the originator is making an attempt to utilize Alice as an oracle.

4.1.2.5. What other principals are normal recipients for each protocol that Alice utilizes and where she originates the first message? A sudden change in the targets of requests for service by Alice may indicate that Alice has been compromised and is now being used to gather information for an attack.

4.1.2.6. What other principals are normal originators for each protocol that Alice utilizes and where she is a recipient of the first message. A sudden change in the sources of requests for service from Alice may indicate that another principal has been compromised and is now being used to gather information for an attack.

4.1.2.7. In what order does Alice utilize protocols where she is the first message originator? Multiple sudden changes in the order of requests for service from Alice may indicate that Alice is making an attempt to gather information for an attack.

4.1.2.8. In what order does Alice utilize protocols where she is the first message recipient? Multiple sudden changes in the order of requests for service to Alice may indicate that another principal is making an attempt to gather information for an attack.

4.1.2.9. How often does Alice exercise an encryption followed by signature? While signing an encrypted message is considered a vulnerable activity, a principal may, after careful consideration, conduct certain ordered activities without concern. A change in this ordering pattern may indicate that an attack is ongoing.

4.2. Trace profiles.

In our earlier definition of traces, we referred to the tendency of protocol analysis to focus on the symbolic execution. We again consider the symbolic execution of protocols, not as derived from a preconceived or contrived execution for test purposes, but the symbolic representation of messages executed in a production protocol environment. Rather than constructing the interleavings from the protocols, we reconstruct the symbolic trace from the execution trace of the protocols as they occur in the system.

Symbolic trace information may be extracted from state data maintained by the host representing each principal as the protocols are executed, or a sophisticated listener monitoring communications on the network may infer it. Since we are concerned with activities that correspond to a trusted principal, we can assume that the necessary state information will be available to translate actual messages into symbolic form in real time.[2]

Because we can recover the symbolic representation of protocols as they execute, we can construct profiles of protocol usage base on their symbolic characteristics. For example, we can record the symbolic representation of every protocol that executes on the monitored computer and record statistical information about the sessions, and about each message. We can determine which protocol that the monitored principal participates in. We can determine who the monitored principal communicates with and can gather statistics regarding the time and sequencing of application of these protocols. These statistics can be translated into the model information discussed in the next paragraph.

4.3. Statistical Models. The metrics described in paragraph 4.1 can be represented in statistical models that Denning describes [DEN86, pp122-3]. For example, the metrics described in paragraphs 4.2.3.4 and 5 can be analyzed using the operational model, mean, and standard deviation as given by Denning in par 5.2.1 and 2 and by the multivariate model from Denning's 5.2.1.3. The Markov Process Model as given in Denning's 5.2.4 can measure the metrics we describe in paragraphs 4.2.3.1, 2, 5, and 6. The Time Series Model Denning presents in paragraph 5.2.5 are applicable to the events we describe in paragraphs 4.2.3.3, 4.2.3.4, 4.2.3.7, 4.2.3.8, and 4.2.3.9.

Such modeling will serve to improve both the percentage of attacks detected and the ration of alarms to activities, similar to the results of intrusion detection systems.

A sample profile for measuring Trent's activity may include a three dimensional array, where one dimension represents each protocol that Trent recognizes, another represents each other principal that Trent communicates with, and the third distinguishes whether Trent was the originator or recipient of the first message.

[2] As an additional note, we consider evaluation of actual protocol messages and the explicit program actions that result to be an uncharted research arena, with roots in classic intrusion detection methodology. We leave that discussion for another time.

Section 5. Categorization of attacks and potential responses.

5.1. Taxonomies of attacks. We now turn from our focus from detecting attacks on trusted principals to categorizing attacks for the purpose of formulating appropriate responses. We have already pointed out taxonomies for intrusions into computer systems [KS95] and for attacks against protocols [SYV94]. The latter is of particular to us.

Syverson partitions attacks against protocols into two major categories of external and internal attacks. These are further decomposed into categories and sub-categories. These categories and responses follow:

- Interleaving attacks (including replays) requiring contemporaneous execution of more than one protocol. An appropriate response to detection of contemporaneous execution of two protocols that are vulnerable to such an attack would be to suspend or cancel one session or the other.
- Replay attacks that need not require contemporaneous execution of more than one protocol. The proper response in this case would be dependent on the state of the principals involved when the attack is detected. If the attack is detected during execution of the reference session, keys may be updated, certifications revoked, and existing sessions may be aborted. If the attack is detected during the attack session, the attack session would be aborted.
- Message deflection attacks. If message deflection is detected, there are two responses required. First, the principal that was the intended destination for the deflected message must be notified and damage control actions taken. Second, the principal that received the message should be notified and the protocol session, if it is still active, aborted.
- Message reflection attacks. The impact of message reflection is centralized to one principal that is the originator and recipient of the message. Again, the proper response depends on the timing of the detection. If the attack session is still under way, it should be aborted. If the attack session has ended, the victim should be notified of the details of the attack and should initiate local damage control activities.

5.2. Other categorizations of attacks.

The above taxonomy provides a comprehensive view of protocol vulnerabilities from the perspective of interleavings of messages. We take another perspective of these vulnerabilities to consider the intent of attackers and discuss responses related to these intentions. Once an attack is detected, at least three goals must be considered when constructing a response:

1. Assess and correct the damage of the compromise.
2. Prevent further compromise.
3. Catch and prosecute the perpetrator.

In the following discussion, we consider Alice and Bob to be uncompromised principals, Trent is a principal that provides trusted services, and Mallory is a malicious attacker.

- Compromise secrecy. This is the canonical attack. Alice and Trent need to share information privately. Mallory wants to know the information and constructs an

attack that will divulge the message meaning to her. Terminating the session, changing session or key exchange keys, identifying and gathering appropriate log files may be appropriate responses.

- Compromise integrity. Mallory may attack the system in order to provide inaccurate information to Alice or Trent. When such an attack is detected, data from the attacked session should be validated as well as conducting audits from previous sessions.

- Compromise nonrepudiation. If Mallory can sign messages as if she were Alice, then she can incorrectly attribute actions or information to Alice. Detection of such an attack should result in correction of any signatures generated during the attack session and should initiate review of records of previous transitions involving signature by Alice. Depending on the nature of the attack, long term key change may be in order.

- Compromise availability. Mallory may desire to prevent Alice from receiving one or more messages while preventing Alice from recognizing that the message(s) have been delayed or destroyed undelivered. The appropriate response to a denial of service attack is to restore the secure channel and notify other principals of the loss of service so that any lost transmissions may be recreated.

- Attempt to masquerade as Alice to Bob. If Mallory can convince Bob that she is Alice, she can compromise secrecy, integrity, and nonrepudiation between Alice and Bob. Response to detection of such an attack is dependent on its success. If the masquerade has been successful, affected participants should be notified and long-term keys changed. At a higher level, the nature of the attack should be evaluated and the security vulnerability removed. Participants should be notified of the vulnerability until it is resolved.

- Attempt to masquerade as Alice to Trent. If Mallory can convince Trent that she is Alice, she can compromise secrecy, integrity, and nonrepudiation between Alice and all other principals.

- Attempt to masquerade as Trent. If Mallory can convince all other principals that she is Trent, then Mallory can compromise secrecy, integrity, and nonrepudiation between all principals. Because of the widespread ramifications, these are the most dangerous masquerade attacks. Response to an attempt to masquerade as a trusted service must first focus on controlling the damage.

- Attempt to use Alice as an oracle. If Mallory can devise a general method of utilizing Alice as an oracle, then Mallory can masquerade as Alice to any other principal, compromising secrecy and nonrepudiation.

- Attempt to use Trent as an oracle. If Mallory can devise a general method of utilizing Trent as an oracle, then Mallory can masquerade as Trent.

Section 6. Conclusion.

We present a method to provide active defense for distributed security services. We have shown how proven intrusion detection technology combined with knowledge gained by formal analysis of security protocols can be applied to this problem. Our method involves addressing behavior relative to protocol activation and use rather than considering activities against objects, as is conducted in classic intrusion detection. We have categorized protocol-based attacks and have proposed general responses to instances of detected attacks based on our categories and on existing taxonomies.

Until recently, the requirement for trusted services essentially resided with the federal government and a few large corporations, where key exchange was most often carried out by courier, with the key material stored on paper tape or diskette. Present technology demands extension of the protection provided by cryptography. This necessitates extension of key distribution and, thus, authentication services. These centralized services are attractive targets for sophisticated intruders. The method we prescribe offers to protect this vital link to our security infrastructure.

Bibliography

[AK97] R. Anderson and M. Kuhn, "Low Cost Attacks on Tamper Resistant Devices, Security Protocols", 5th International Workshop, Paris, France, April 7-9, 1997, Proceedings, Springer-Verlag, LNCS 1361, pp. 125-136.

[AT91] Martin Abadi and Mark R. Tuttle, "A Semantics for a Logic of Authentication", Tenth Annual ACM Symp on Princ of Dist Computing, Montreal, Canada, August, 1991

[CDEKS96] Crosbie, M.; Dole, B.; Ellis, T.; Krsul, I.; Spafford, E, "IDIOT - Users Guide", Technical Report TR-96-050, Purdue University, COAST Laboratory, Sept. 1996

[DEN86] Dorothy E. Denning, "An Intrusion-Detection Model", From 1986 IEEE Computer Society Symposium on Research in Security and Privacy, pp 118-31

[DS81] D. E. Denning and G. M. Sacco, "Timestamps in key distribution protocols," Communications of the ACM, vol. 24, no. 8, Aug 1981, pp. 533-536

[DS99] Daniels and Spafford, "Identification of Host Audit Data to Detect Attacks on Low-level IP", Journal of Computer Security, Volume 7, Issue 1, 1999

[DY83] Dolev, D., and Yao, A.C. "On the security of public key protocols". IEEE Trans. Inf. Theory IT-29, 2(Mar. 1983), pp. 198-208. Also Stan-CS-81-854, May 1981, Stanford U.

[JOU00] Y. Jou, F. Gong, C. Sargor, X. Wu, S. Wu, H. Chang, and F. Wang, "Design and Implementation of a Scalable Intrusion Detection System for the Protection of Network Infrastructure", DARPA Information Survivability Conference and Exposition 2000, Jan 25-27, 2000, Vol. 2, pp 69-83

[KEM89] R. A. Kemmerer, "Using Formal Methods to Analyze Encryption Protocols," IEEE Journal on Selected Areas in Communications, vol. 7, mo. 4, pp. 448-457, May 1989

[KG91] Rajeshekar Kailar and Virgil D. Gligor, "On Belief Evolution in Authentication Protocols", In Proceedings of the Computer Security Foundations Workshop IV, PP 103-16, IEEE Computer Society Press, Los Alamitos, CA, 1991

[KS95] Sandeep Kumar and Eugene Spafford, "A Taxonomy of Common Computer Security Vulnerabilities Based on their Method of Detection", Technical Report, Purdue University, 1995

[KS96] Sandeep Kumar and Eugene Spafford, "A Taxonomy of Common Computer Security Vulnerabilities Based on their Method of Detection", Technical Report, Purdue University, 1995

[KSW98] J. Kelsey, B. Schneier, and D. Wagner, "Protocol Interactions and the Chosen Protocol Attack", Security Protocols, 5th, International Workshop April 1997, Proceedings, Springer-Verlag, 1998, pp. 91-104, http://www.counterpane.com/chosen_protocol.html

[KMM93] R. Kemmerer, C. Meadows, and J. Millen, ""Three Systems for Cryptographic Protocol Analysis", The Journal of Cryptology, Vol. 7, no. 2, 1993

[LP99] Ulf Lindqvist and Phillip A Porras, "Detecting Computer and Network Misuse Through the Production-Based Expert System Toolset (P-BEST)", 1999 IEEE Computer Society Symposium on Security and Privacy, pp 146-61

[MEAD95] Catherine Meadows, "Formal Verification of Cryptographic Protocols: A Survey," Advances in Cryptology - Asiacrypt '94, LNSC 917, Springer-Verlag, 1995, pp. 133-150, http://www.itd.nrl.navy.mil/ITD/5540/publications/1995/1995meadows-asiacrypt94.ps

[MEAD99] Catherine Meadows, "Analysis of the Internet Key Exchange Protocol using the NRL Protocol Analyzer", 1999 IEEE Computer Society Symposium on Security and Privacy, pp 216-34, http://www.itd.nrl.navy.mil/ITD/5540/publications/CHACS/1999/1999meadows-IEEE99.pdf

[MEAD99b] Catherine Meadows, "A Formal Framework and Evaluation Method for Network Denial of Service", 12th IEEE Computer Security Foundations Workshop, Jun 28-30, 1999, Mordano, Italy

[NS78] Roger M. Needham, Michael D. Schroeder, "Using Encryption for Authentication in Large Networks of Computers", Communications of the ACM December 1978 vol. 21 #12, pp. 993-999

[PAUL99] Lawrence C. Paulson, "Proving Security Protocols Correct"' IEEE Symposium on Logic in Computer Science, Trento, Italy (1999), pp 370-81, http://www.cl.cam.ac.uk/users/lcp/papers/Auth/lics.pdf

[SONG99] Dawn Xiaodong Song, "Athena: A New Efficient Automatic Checker for Security Protocol Analysis", 12th IEEE Computer Security Foundations Workshop, Jun 28-30, 99, Mordano, Italy

[SYV94] Paul Syverson, ``A Taxonomy of Replay Attacks," Proceedings of the Computer Security Foundations Workshop VII, Franconia NH, 1994 IEEE CS Press (Los Alamitos, 1994), http://www.itd.nrl.navy.mil/ITD/5540/publications/CHACS/1994/1994syverson-foundations.ps

[VK99] Vigna and Kemmerer, "NetSTAT: A Network-based Intrusion Detection System "Journal of Computer Security", Volume 7, Issue 1, 1999

[YAS96] Alec Yasinsac, "Evaluating Cryptographic Protocols:, Ph.D. Dissertation, University of Virginia, January 1996

[YW93] Yasinsac, Alec; Wulf, William A, "Evaluating Cryptographic Protocols", University of Virginia Technical Report, CS-93-66, December 22, 1993, ftp://ftp.cs.virginia.edu/pub/techreports/CS-93-66.ps.Z