2009 International Conference on Availability, Reliability and Security

Towards Intrusion Detection for Encrypted Networks

Vik Tor Goh*, Jacob Zimmermann† and Mark Looi‡
Information Security Institute,
Queensland University of Technology, Australia.
v.goh@qut.edu.au, † j.zimmerm@isi.qut.edu.au, ‡m.looi@qut.edu.au

Abstract

Traditionally, network-based Intrusion Detection Systems (NIDS) monitor network traffic for signs of malicious activities. However, with the growing use of Virtual Private Networks (VPNs) that encrypt network traffic, the NIDS can no longer analyse the encrypted data. This essentially negates any protection offered by the NIDS. Although the encrypted traffic can be decrypted at a network gateway for analysis, this compromises on data confidentiality. In this paper, we propose a detection framework which allows a traditional NIDS to continue functioning, without compromising the confidentiality afforded by the VPN. Our approach uses Shamir's secret-sharing scheme and randomised network proxies to enable detection of malicious activities in encrypted channels. Additionally, this approach is able to detect any malicious attempts to forge network traffic with the intention of evading detection. Our experiments show that the probability of a successful evasion is low, at about 0.98% in the worst case. We implement our approach in a prototype and present some preliminary results. Overall, the proposed approach is able to consistently detect intrusions and does not introduce any additional false positives.

1. Introduction

Intrusion detection grew from the notion that computer misuse can be detected by analysing audit data in a computer system or network [1]. Computer log files, computer settings and network traffic usually form the structure of most audit data. Current intrusion detection systems analyse the audit data in either *anomaly-based* detection or *misuse-based* detection mode.

Regardless of the detection mode used, if the audit data is encrypted, the IDS will fail. There are two broad categories of IDS; the *host-based* IDS (HIDS) and *network-based* IDS (NIDS). We focus mainly on the effects of encrypted audit data on NIDS. This is motivated by the growing use of end-to-end (ETE) encrypted networks that obfuscates all network traffic including malicious traffic between any two endpoints in the network. Some commonly used ETE encryption protocols are Secure Sockets Layer (SSL) and various other Virtual Private Network (VPN) protocols.

To a certain extent, this problem can be solved with proper network design where the encrypted network is terminated at the NIDS for analysis before sending it along its route in decrypted form. However, this setup is not suitable when confidentiality must be maintained from the source right up to the destination. What is needed is a NIDS that can integrate and function well in ETE encrypted networks.

In this paper, we propose an approach that allows a NIDS to operate in ETE encrypted networks and does not compromise the network's confidentiality or integrity. For most NIDS, such integration has never been directly addressed. There is an implicit assumption that unencrypted network traffic is always available and it is up to the network administrator to ensure this.

2. Related Works

Besides modifying the network design to accommodate the NIDS, another is to use a man-in-the-middle approach. This is a type of active sniffing technique where the sniffer makes independent connections with two communicating peers and relays messages between them, with each of the peers believing that it is communicating directly with its counterpart.

Yamada et al. [2] noted that with the *NIDS-in-the-middle*, all encrypted network traffic can be decrypted with the private keys of both peers. One of the difficulties in using this approach is the need to have a secure key management system.

This work is supported in part by funds from (ISC)²

978-0-7695-3564-7/09 $25.00 © 2009 IEEE
DOI 10.1109/ARES.2009.76

For these reasons, most research work have focused on other approaches. To the best of our knowledge, there are three known approaches to this problem.

The first approach uses statistical traffic analysis techniques to detect intrusions. With such techniques, network packets but not its payload, are examined to infer information from patterns in the communication process. Specific patterns of network occurences often characterise an attack. Yamada et al. [2] and Piccitto et al. [3] used this technique to identify malicious activities in SSL and VPN traffic. This analysis is limited in scope due to the few traffic patterns that can actually be deduced purely by observing the network.

The second approach assumes that any attempts to misuse ETE encryption protocols are symptoms of attacks on an endpoint. To detect protocol misuse, works by Md. Fadlullah et al. [4], Joglekar and Tate [5] and Yasinsac and Childs [6] begin by defining an accurate specification of a legitimately working ETE encryption protocol. Any deviation from this specification is considered as an attack. In spite of that, the task of defining this specification is not trivial. Every possible legitimate state within the protocol must be modeled. Moreover, if an attack complies to the protocol exactly, a malicious payload can still be sent because this approach does not check the payload.

The third approach analyses the payloads of network packets. Also known as deep packet inspection, this approach is suitable for the detection traditional attacks such as malformed URL string or SQL injection attacks. The data and/or the header portion of network packets can be quickly matched against attack signatures, without relying on any complicated preprocessing as the previous two approaches.

Abimbola et al. [7] installed their specialised NIDS sensors in network endpoints where network traffic has already been decrypted and accessible. Instead of proposing a new detection algorithm, they feed the decrypted network traffic into a traditional NIDS like SNORT [8]. Although effective, it does not address the fact that the sensor residing on the target system can be defeated if the system is compromised.

We thus propose a framework that integrates a NIDS into an encrypted network, which is functionally equivalent to passive sniffing techniques but without the need for public-key infrastructure (PKI) and without compromising on confidentiality.

Although a HIDS could have been used to address the problem of detecting ETE attacks, it has the added overhead of being intrinsically complex. A HIDS has to monitor many distinct aspects of the host such as system calls and file statuses.

3. Detection Framework

3.1. Approach

A standard NIDS intercepts and analyses the network traffic between communicating parties, either using passive sniffing, or by acting as a relay in the case of a NIDS-in-the-middle configuration.

If the traffic is ETE encrypted, passive sniffing becomes infeasible unless the NIDS receives all decryption keys of the communicating parties. This design involves the use of a PKI which implies heavy key management overhead. We propose instead a protocol, based on a *secret-sharing* scheme, in which a copy of all network traffic is explicitly forwarded to the NIDS over standard channels. This is done while preserving the privacy and integrity of the communication process. Moreover, the protocol also makes it difficult for a malicious sender to evade detection.

To achieve this, we have a Central IDS (CIDS) that carries out traffic analysis and intrusion detection. The CIDS operates as a separate host in the encrypted network. We make no further assumptions about the type of detection technique used by this CIDS.

Next, we install IDS sensors in all endpoints of the same network [7]. An IDS sensor ensures that all network traffic that goes to the receiver also goes to the CIDS for analysis. Our approach is summarised by the following principle:

All traffic sent to a receiver by a sender must be replicated and forwarded also to the CIDS, without the possibility of the sender withholding traffic from the CIDS or forging fake traffic, and while maintaining the confidentiality and integrity of the ETE network.

To realise this principle on an ETE encrypted network, it is thus necessary to achieve three objectives:

1) Ensure that all traffic sent by the sender is forwarded to both the receiver and CIDS, without possibility of witholding traffic from the CIDS;
2) Prevent the sender from sending forged traffic to the CIDS; and
3) Do not compromise the privacy afforded by the ETE encrypted network.

To enforce these objectives, we use message forwarding proxies shown in Fig. 1 and Shamir's secret-sharing threshold scheme [9]. These proxies are regular network hosts and the protocol requires each host implement such relaying capabilities.

If a sender wishes to send a message to the receiver, the sender will first split the message into its corresponding shares using the secret-sharing scheme. Briefly, a (k, n) secret-sharing scheme represents a

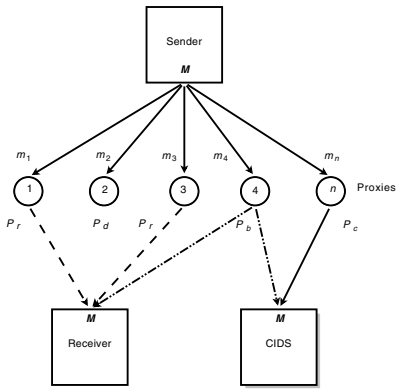

Figure 1. Proposed transmission scheme ($k = 2$)

secret S as n shares where $S = \{s_1, s_i, \ldots, s_n\}$. Knowledge of any k or more s_i recovers S while knowledge of any $k-1$ or less reveals nothing. In secret-sharing, confidentiality is thus an inherent feature and key management is not needed.

So, if we let the message to be sent be M, its shares are $\{m_1, m_i, \ldots, m_n\}$. The sender will then send one share m_i to one of the proxies. Each proxy in turn randomly chooses where to forward the share to. Specifically, it does one of the following actions:

- Forward to receiver with probability $\boldsymbol{P_r}$;
- Forward to CIDS with probability $\boldsymbol{P_c}$;
- Forward to both CIDS and receiver with probability $\boldsymbol{P_b}$; or
- Drop the message with probability $\boldsymbol{P_d}$.

The message M is recoverable by both the receiver and CIDS provided that they receive at least k shares each. We can summarise the entire process as follows:

Algorithm

1) Sender splits M into $\{m_1, m_i, \ldots, m_n\}$
2) Each m_i is sent to a proxy
3) Proxy p_i does one of the four predefined actions
4) Sender receives k or more m_i and recovers M
5) CIDS receives k or more m_i and recovers M

If we assume that in the initial state, all hosts (except possibly a single malicious sender) follow the protocol, the three objectives identified above are met as follows:

1) The sender cannot prevent the forwarding proxies from forwarding the traffic to the CIDS;
2) The randomised behaviour of the proxies prevents the sender from choosing a subset of proxies that would allow him to send forged traffic to the CIDS.
3) The shared secret scheme ensures that none of the relaying proxies is able to reassemble the message, unless k proxies (or $k-1$ proxies plus the sender) conspire.

We analyse these different cases formally in Sect. 3.3. Note that encryption could have been used to encrypt M prior to sending instead of using the secret-sharing scheme, but that would require every endpoint possess the decryption keys of every other endpoints, with all the implied complexity of key management.

3.2. Detection Process

Consider now the case of a sophisticated attacker. In an attempt to evade detection, the attacker sends two different messages; M to the target while M' is for the CIDS. This is done with the intention of misleading the CIDS with forged and incorrect traffic. Hence, if the CIDS can detect the presence of two distinct messages, it can detect an evasion attempt.

For the sake of clarity, let M be *malicious message* to the receiver and M' be *forged but harmless message* to the CIDS. Each of them is represented as $M = \{m_1, m_i, \ldots, m_\alpha\}$ and $M' = \{m'_1, m'_j, \ldots, m'_\beta\}$ where $\alpha + \beta \leq n$ and $\alpha, \beta \geq k$ for a (k, n) secret-sharing scheme.

Since the actions of the proxies are *a-priori* unpredictable, the attacker will not know beforehand which of the n proxies will forward to whom (CIDS, receiver or both). It is therefore impossible for the attacker to reliably select which proxies should receive shares of M and which should receive shares of M'. This is true if we assume that all the proxies are uncompromised and not collaborating with the attacker.

With this uncertainty, the receiver may receive k or more shares that resolve to one of the following three cases.

1) Receives only m_i shares and recovers M;
2) Receives m'_j shares and recovers M'; or
3) Receives a *mixture* of both m_i shares and m'_j shares. We call this case as corrupted and label it as C.

Referring to cases above, an evasion is successful if the CIDS receives M' (harmless message). However, if C is received, the receiver or CIDS knows that there has been an attempt to forge a fake message. The effects of receiving M (malicious message) are different for the receiver and CIDS. While M compromises the receiver, the CIDS will identify M as malicious by its detection engine.

Consistent with our assumption, we say that an attack is successful if and only if the CIDS receives M' (harmless message) and the receiver receives M (malicious message). This is reflected in Table 1.

Table 1. Truth table of detection outcomes where a *True Positive (TP)* is detection by CIDS and a *False Negative (FN)* is a successful attack

CIDS	Receiver		
	M	M'	C
M	TP	TP	TP
M'	**FN**	**TN**	TP
C	TP	TP	TP

As shown in Table 1, whenever the receiver or CIDS receives C, it is immediately known that there has been an attempt to evade detection. Notice that in our approach, there are no *false positives* outcomes. This detection method does not introduce any additional false positives on top of the false positives caused by the CIDS, which relies on traditional IDS.

There are actually two classes of true positives that can be detected by the approach. They are as follows,

- **Remote attack.** This type of attacks can generally be detected by IDS like SNORT. In any case, these attacks pose a traditional intrusion detection problem. This can be detected if the CIDS received M; and
- **Evasions.** Attacks that attempt to evade CIDS detection in our context, by trying to forge false reports to the CIDS. This can be detected if either the CIDS or receiver receives C.

When both the receiver and CIDS receive the forged but harmless message M', a true negative outcome is observed. A *true negative* (TN) outcome does not negatively affect the receiver or CIDS.

Notice also that in Table 1, there is only one false negative or successful attack outcome. We will analyse the probability of this occurrence below.

3.3. Analysis

Each proxy carries out a specific action with a certain probability as stated earlier in Sect. 3.1. The probabilities P_r, P_c, P_b and P_d are the same for all proxies and $P_r + P_c + P_b + P_d = 1$.

Consider α malicious shares given as $\{m_1, m_i, \ldots, m_\alpha\}$ that are exclusively meant for the receiver. The probability of the receiver receiving x shares of m_i only, where $0 \leq x \leq \alpha$ is,

$$P(x) = {}^{\alpha}C_x \, (P_r)^x \, (P_d)^{\alpha - x} \tag{1}$$

The notation ${}^{\sigma}C_{\varphi}$ is the binomial coefficient $\binom{\sigma}{\varphi}$. Eq. (1) considers the fact that m_i should not be sent to the CIDS to prevent mixtures from occurring at the CIDS. Similarly, Eq. (2) follows from Eq. (1) for β harmless shares of $\{m'_1, m'_j, \ldots, m'_\beta\}$ where $0 \leq y \leq \beta$.

$$P(y) = {}^{\beta}C_y \, (P_c)^y \, (P_d)^{\beta - y} \tag{2}$$

With secret-sharing, the minimum number of shares required to recover M and M' is k. Furthermore, if an attack is to be successful, the receiver must only receive m_i while the CIDS must only receive m'_j. Hence, the overall probability of getting a false negative (FN) outcome (successful attack) is,

$$P(\text{FN}) = \sum_{x=k}^{\alpha} \sum_{y=k}^{\beta} P(x) \cdot P(y) \tag{3}$$

In contrast, there are two classes of true positives which are the remote attacks and evasions. To determine their probabilities, we begin by defining the probability of the CIDS receiving u shares of m_i,

$$P(u) = {}^{\alpha}C_u \, (P_b + P_c)^u \, (P_d + P_r)^{\alpha - u} \tag{4}$$

where $0 \leq u \leq \alpha$. Likewise, the probability of the CIDS receiving v shares of m'_j where $0 \leq v \leq \beta$ is,

$$P(v) = {}^{\beta}C_v \, (P_b + P_c)^v \, (P_d + P_r)^{\beta - v} \tag{5}$$

The joint probability of the CIDS receiving only u shares of m_i and v shares of m'_j is thus given as,

$$P(u, v) = P(u) \cdot P(v) \tag{6}$$

We know that a remote attack is detected if the CIDS receives only malicious shares and none of the harmless shares. This results in the CIDS getting M. Since k is minimum number of shares required to recover M from m_i, we get

$$P(\text{Remote attacks}) = \sum_{u=k}^{\alpha} P(u, v = 0) \tag{7}$$

Evasions are detected if the CIDS receives a mixture of m_i and m'_j. Consequently, the probability of the CIDS detecting evasions is given as,

$$P(\text{Evasions}) = \sum_{u=1}^{k-1} \sum_{v=k-u}^{\beta} P(u, v) + \sum_{u=k}^{\alpha} \sum_{v=1}^{\beta} P(u, v) \tag{8}$$

Summing Eq. (7) and Eq. (8), the total probability of a true positive outcome (successful detection) on the CIDS is,

$$P(\text{TP}) = P(\text{Remote attacks}) + P(\text{Evasions}) \tag{9}$$

From Sect. 3.1, we have the conditions $\alpha + \beta \leq n$ and $\alpha, \beta \geq k$. If the attacker uses the minimum number of shares for both α and β which is k, the total number of shares is $2k$. Evasion thus becomes strictly impossible if $2k \geq n$.

4. Evaluation

A prototype that implements the proposed approach has been developed for a Linux operating system. An IPsec-based VPN with 12 endpoints and one CIDS running a standard SNORT installation is used as the experimental network in our tests. From the discussions presented in Sect. 3, we can reasonably expect the proposed approach to have the following operating properties.

- It complements and works with a standard NIDS;
- There should not be any negative impact on the detection ability of the NIDS; and
- It makes the task of evading detection very difficult if not impossible.

We begin our evaluation by verifying that the proposed approach does not introduce any additional false positives. We apply a dataset of synthetic traffic against SNORT in a network with and without our implementation. The dataset consists of both malicious and non-malicious traffic for a range of network protocols like HTTP, ICMP and FTP. Throughout the experiment, we observe no differences in the number of SNORT alerts when our implementation is in use, compared to a network without it. This experiment is not an IDS evaluation and therefore we make no further attempts to ensure its completeness.

4.1. Experiment and Results

We proceed to apply our proposed approach which is configured to use a $(n = 10, k = 3)$ secret-sharing scheme in the same network setup.

We assume the presence of a sophisticated attacker where the attacker attempts to evade CIDS detection by forging harmless network traffic. Harmless network traffic such as a "`GET /index.html HTTP/1.1`" request is forged to be sent to the CIDS, while malicious traffic such as "`GET /x90x90x90...`" is sent to the receiver. The sequence of `x90` (NOOP sledge) is often symptomatic of buffer overflow-based shellcode attacks. Accordingly, most misuse-based IDS use `x90` as a detection signature.

For this test, a *mixture* of shares (resulting in C) is obtained at both the receiver and CIDS. This outcome is considered as a positive detection. The test is repeated numerous time and all evasion attempts are consistently detected. In fact, the probability of successfully evading detection is small and is expressed as Eq. (3). We also separately verified Eq. (3) by simulating attacks. Fig. 2 presents a plot of Eq. (3) and the simulated results for varying values of k. We set $P_r = \frac{2}{6}$, $P_c = \frac{2}{6}$, $P_b = \frac{1}{6}$ and $P_d = \frac{1}{6}$.

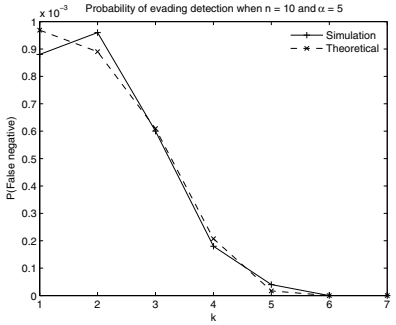

Figure 2. Probability of false negative with $n = 10$

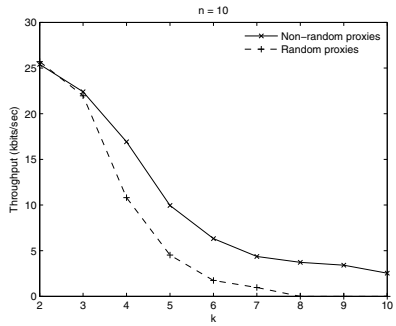

Figure 3. Throughput when k shares are required by receiver and CIDS for $n = 10$

Theoretically, any value of k between $2 \leq k \leq n$ can be used. As Fig. 2 shows, there is a downward trend in an attacker's ability to evade detection as larger values of k are used. While $k \leq \frac{n}{2}$, there exist a chance for an attacker to evade detection. However, at the cutoff value of $k > \frac{n}{2}$, an attacker's chance of evading detection is completely eliminated.

While using a larger k may seem like a good idea, it comes with a cost of increased network latency. Latency depends on k because a receiving party must have received at least k shares before the message can be recovered. Due to the unpredictable and independent nature of the proxies, there exist situations when less than k shares are forwarded to the receiving party.

Fig. 3 shows the throughput (kbits/sec) of our prototype. The **dashed line** is obtained when the proxies are randomly forwarding or dropping network packets. Latency increases with increasing values of k and we thus see a drop in the throughput. As a baseline for comparison, we also plot the throughput when all n proxies are set to forward their packets to both the receiver and CIDS (**solid line**). The average drop in throughput is calculated to be about 40%.

4.2. Discussions

One concern in using the proposed approach is the implied network overhead. If a sender sends t packets of data to a receiver with our proposed approach, the total network overhead is calculated and observed to be $2tn$ packets. Taking this into consideration, we believe that the approach is more suitable when used in a more selective manner. For instance, some traffic intensive applications like media and voice streaming can be deemed safe and do not require our approach.

Our approach thus far does not consider cases of multi-node conspiracy. A conspiracy in our context is a scenario where an attacker collaborates with already compromised proxies to further propagate its malicious activities without being detected by the CIDS. Rather than being unpredictable to the attacker, a conspiring proxy can actually dictate which share (malicious or harmless) be forwarded to whom (CIDS or receiver).

4.2.1. Other applications. Although the motivation for our work has mainly been the need for an effective technique to monitor ETE networks, we believe that it can be adapted for applications where nodes in a multi-node network cannot be trusted.

Such is the case with *mobile ad-hoc networks* (MANETs). A MANET is formed when a group of mobile nodes cooperatively communicate with each other without a pre-established infrastructure. According to Tseng et al. [10], a MANET is inherently trust-all-peers by design and therein lies its problem. A malicious node can corrupt other trusting nodes by forging incorrect data packets to evade detection. This problem bears similarities with our work, specifically the fact that not all nodes can be fully trusted.

5. Conclusion

In this paper, we propose a principle that allows a CIDS to analyse network traffic even when end-to-end encryption is used. This principle ensures that all network traffic are forwarded to both the CIDS as well as the intended receiver. The implementation of the principle is able to ensure that network traffic arrives at both the CIDS and receiver. It is also able to detect attempts to evade detection by forging network traffic. This is achieved without compromising the confidentiality of the transmissions.

We have demonstrated through the experiment and simulations that our approach has been able to consistently detect two types of attacks, specifically remote attacks and evasions. In particular, our approach makes it difficult for an attacker to successfully evade detection. The results have so far been promising, especially the fact that the approach does not cause additional false positives. We have also identified a number of limitations and are currently working to address them.

Traditional NIDS methodologies will no longer be feasible in fully encrypted network infrastructure and will have to be adapted. We believe that our approach is one step in that direction.

References

[1] J. P. Anderson, "Computer security threat monitoring and surveillance," James P. Anderson Co., Tech. Rep., 26 Feb. 1980.

[2] A. Yamada, Y. Miyake, K. Takemori, A. Studer, and A. Perrig, "Intrusion detection for encrypted web accesses," in *21st Intnl. Conf. on Advanced Information Networking and Applications Workshops*, Niagara Falls, Canada, May 2007, pp. 569–576.

[3] D. Piccitto, S. Burschka, and G. Urvoy-Keller, "Traffic mining in IP tunnels," Master's thesis, Eurecom Institute, Sophia-Antipolis, France, Sep. 2007.

[4] Z. Md. Fadlullah, T. Taleb, N. Ansari, K. Hashimoto, Y. Miyake, Y. Nemotoi, and N. Kato, "Combating against attacks on encrypted protocols," in *IEEE Intnl. Conf. on Communications*, Glasgow, Scotland, Jun. 2007, pp. 1211–1216.

[5] S. P. Joglekar and S. R. Tate, "Protomon: embedded monitors for cryptographic protocol intrusion detection and prevention," in *Intnl. Conf. on Information Technology: Coding and Computing*. Las Vegas, Nevada, USA, Apr. 2004, pp. 81–88.

[6] A. Yasinsac and J. Childs, "Analyzing internet security protocols," in *6th IEEE Intnl. Symposium on High Assurance Systems Engineering*. Boca Raton, Florida, USA, Oct. 2001, pp. 149–159.

[7] A. Abimbola, J. M. Munoz, and W. J. Buchanan, "Nethost-sensor: Investigating the capture of end-to-end encrypted intrusive data," *Computers & Security*, vol. 25, no. 6, pp. 445–451, Nov. 2006.

[8] M. Roesch, "Snort - Lightweight intrusion detection for networks," in *13th Large Installation System Administration Conf.*, Seattle, Washington, USA, Nov. 1999, pp. 229–238.

[9] A. Shamir, "How to share a secret," *Communications of the ACM*, vol. 22, no. 11, pp. 612–613, Nov. 1979.

[10] C. H. Tseng, S.-H. Wang, C. Ko, and K. Levitt, "DEMEM: Distributed evidence-driven message exchange intrusion detection model for MANET," in *9th Intnl. Symposium on Recent Advances In Intrusion Detection*, vol. 4219. Hamburg, Germany, Sept. 2006, pp. 249–271.