Tracing Back Attacks Against Encrypted Protocols

Tarik Taleb
talebtarik@ieee.org

Zubair Md. Fadlullah
zubair@it.ecei.tohoku.ac.jp

Kazuo Hashimoto

Yoshiaki Nemoto

Nei Kato

Graduate School of Information Sciences, Tohoku University, Sendai, Japan.

ABSTRACT

Attacks against encrypted protocols have become increasingly popular and sophisticated. Such attacks are often undetectable by the traditional Intrusion Detection Systems (IDSs). Additionally, the encrypted attack-traffic makes tracing the source of the attack substantially more difficult. In this paper, we address these issues and devise a mechanism to trace back attackers against encrypted protocols. In our efforts to combat attacks against cryptographic protocols, we have integrated a traceback mechanism at the monitoring stubs (MSs), which were introduced in one of our previous works. While we previously focused on strategically placing monitoring stubs to detect attacks against encrypted protocols, in this work we aim at equipping MSs with a traceback feature. In our approach, when a given MS detects an attack, it starts tracing back to the root of the attack. The traceback mechanism relies on monitoring the extracted features at different MSs, i.e., in different points of the target network. At each MS, the monitored features over time provide a pattern which is compared or correlated with the monitored patterns at the neighboring MSs. A high correlation value in the patterns observed by two adjacent MSs indicates that the attack traffic propagated through the network elements covered by these MSs. Based on these correlation values and a prior knowledge of the network topology, the system can then construct a path back to the attacking hosts. The effectiveness of the proposed traceback scheme is verified by simulations.

Categories and Subject Descriptors: C.2 [Communication Networks]: IWCMC 2007 Computer and Network Security Symposium.

General Terms: Security.

Keywords: Encryption, Intrusion Detection System (IDS), Traceback.

IWCMC'07, August 12–16, 2007, Honolulu, Hawaii, USA.

1. INTRODUCTION

In the last decade or so, network-based intrusions have become a major threat to the internet community. Intruders often conceal their identities by using "address spoofing" or employing "stepping stones" [1]. Such dubious techniques are difficult to detect and to counter. To make things worse, tracing the source of an attack becomes even more difficult, particularly if the attack-traffic is encrypted.

In the field of network security, traceback is a hot topic. An accurate detection of attackers against network resources by tracing back to the attack hosts, or at least to the network of the attack-origin, and taking appropriate actions against them, would discourage further attacks. Therefore, attack detection and prevention schemes are often coupled with traceback techniques. The contemporary research has been limited to detecting and tracing back attacks against application level protocols, which do not employ encryption. However, due to the increasing attacks against encrypted protocols, such as Secured Shell (SSH) and Secured Socket Layer (SSL) protocols, it has become an imperative to deal with such attacks. The conventional Intrusion Detection Systems (IDSs) are often unable to detect these attacks, let alone tracking them back, simply because they heavily depend on inspecting the contents of the packets and fail to do the same in case of encrypted packets. Our previous work investigated these issues and introduced a novel approach to detect attacks against encrypted protocols [2]. The approach consisted of network sniffing agents called monitoring stubs (MSs). By inspecting the unencrypted portions of the packets, MSs monitor specific features of a given encrypted protocol and use them to compare with a normal network profile learnt beforehand. A significant deviation from the normal profile, or an anomaly, is deemed as an attack. As an extension to our previous work [2], this paper focuses on tracing back the possible attackers after an attack has been detected by the MS adjacent to the victim. By exchanging and correlating the feature-patterns over time, the collaborating monitoring stubs attempt to determine the path back to the source of the attack.

The remainder of this paper is organized as follows. Section 2 provides an overview on previous research work related to various traceback schemes. Section 3 presents a brief description of the previous work and then describes the proposed traceback mechanism employing monitoring stubs. The section also presents the scope of attacks traceable by the proposed method. The performance of the proposed scheme is evaluated in Section 4. Section 5 concludes the paper.

2. RELATED WORK

Every attack against a network host commences with the launch of attack-packets from an attacking host, which serves as the entry point into the computer network across which the attack occurs. In a network-based attack, the perpetrator usually generates a series of packets that are destined to the victim host. Over the years, a wide library of work has been dedicated to tracing back the origins of attacks.

The traceback approach proposed in [3] uses a variation of the Time-efficient Stream Loss-tolerant Authentication protocol to generate a code, based on the IP-addresses of the routing devices, that sequentially handles packets. Using a map of the IP-addresses of all upstream routers, the victim of an attack can efficiently reconstruct the route of a packet through up to 32 devices [4]. These works focus on identifying the route within the network-packets without increasing the packet size, which has challenged traceback researchers over the recent years.

Other traceback approaches require routers to produce extra packets for every packet that is encountered by these routers [5]. The victim host collects these extra packets along with the original packets. These additional packets provide authenticated identification of the originating routing devices. The obvious shortcoming of this approach consists in the increase in network traffic due to extra packets generated for each original network-packet, which is to be traced. To alleviate this problem, [5] proposes an extra "trace-packet" to be generated on a probabilistic basis (e.g., approximately one per 20,000 packets). A large number of attack-packets (for instance, packets involved in TCP SYN-flood) continuing for a considerable period of time can be effectively traced in this way. However, in case of attacks involving a low number of packets, the system will fail to trigger enough trace-packets to reconstruct a route back to the attacker.

Recent traceback schemes adopting similar concept, employ different packet marking techniques [6] such as Probabilistic Packet Marking (PPM), Deterministic Packet Marking (DPM), ITrace, and logging techniques such as Source Path Isolation Engine (SPIE). Most of these schemes require the IP-header information. This requirement poses difficulty in tracing back an attacker which sends encrypted packets. In order to use these traditional traceback methods at a network monitor level, the monitoring agents may need to decrypt the headers of the attack packets. However, decrypting packets at intermediate monitors may raise privacy issues and incur additional overheads. Therefore, when cryptographic protocols are in use, the commonly known traceback approaches will not be effective.

One of the fundamental traceback problems, when it comes to encrypted connections, is to trace a stream of attack-packets through a number of "stepping stones". To find an answer to this problem, stream-matching approaches, based on either packet-contents or inter-packet timing, have evolved. An example of the content-based stream-matching schemes is proposed in [7]. The scheme is called "Thumbprinting". Although it shows good performance in tracing back attacks against non-encrypted protocols, encryption renders this method ineffective.

An alternate approach to this stream matching technique is achieved through correlation methods based on the inter-packet delay (IPD) for tracing back attacks against encrypted connections [8] [9]. By correlating IPD of different connections across the network, this approach identifies whether the inspected packets belong to the same connections. Additionally, a polynomial upper bound on the number of packets needed to confidently detect and identify encrypted stepping stone streams has been found by [1].

In conventional trace-back schemes, the contents of the packets are conveniently inspected to determine the source and destination for a given protocol. The traffic-volume, especially during a Denial of Service (DoS) attack, questions the feasibility of inspecting the contents of each and every packet. The advent of spoofing IP-addresses challenges this naive method. Mansfield *et al.* [10] have investigated these issues and introduced a novel technique based on packet-flow monitoring. This technique relies on flow patterns to be traced across networks to trace back intruders. Flow-patterns are profiled by considering the studies that reveal that the normal network usage patterns, in general, vary from those under an attack scenario. Such studies have included investigations of the characteristics of the TCP-RESET connections and Internet Control Message Protocol (ICMP) destination/port unreachable packets, in a campus-based network. Mansfield *et al.* have then looked for differentiating the features of the traffic profiles, in normal situation and during an intrusion. For instance, in case of TCP-SYN packet-based DoS attacks, TCP-SYN connections are monitored by RMON-devices or probes placed at different points of the network, which look for the presence of similar flows at other probes. Thus, the traceback is performed by correlating traffic-patterns observed at various probes placed in the network. The traffic-flow correlation has been reasonably successful, which leads to the inference that traffic-flows may be traced from one link to another across a network. However, this approach does not investigate whether it can be extended to tracing back attacks against encrypted protocols.

Recently some efforts have been made to thwart attacks against encrypted protocol, such as "Protomon" [11]. However, detecting such attacks alone will not suffice. It is also required to dig deep and track back the attackers. Unless traceback systems are employed, IDSs such as Protomon will act as a mere damage-controlling entity. Our preceding work [2] addresses the shortcomings of Protomon and presents a more effective detection method by using a dynamic thresholding scheme to detect anomaly and distributing unique monitoring stubs (MSs) over the network topology. To expand this work further, we envision incorporating traceback features in MSs. For this purpose, we adopt an approach, based on correlation schemes, similar to that proposed in [10] for tracing back attackers. In this approach, although it may not be possible to hunt down the actual attacker based on IP due to encrypted headers, it can trace back to the network or domain from which the attacker launched the attack. From thereon, the authorities of that network can hunt down the attack host. This is beyond the scope of this work and is left as future research work.

3. PROPOSED TRACEBACK SCHEME

3.1 Traceable Attacks against Encrypted Protocols

The attacks against cryptographic protocols rely upon the design and implementation of the target protocols. For instance, the OpenSSL implementation of SSL is susceptible to specific remote timing attacks [12], Man-In-The-Middle (MITM) attacks [13], buffer overflow attacks [14] [15], and Version Rollback attacks. In the remote timing attack, SSL renegotiation attack, and password attack (also called the dictionary attack or brute force attack) against SSH, there is a high interaction (in terms of transmitted messages) between the attacker and the cryptographic protocol server. On the contrary, in case of a buffer overflow attack, the interaction between the client and the server is not high. According to this observation, in our preceding work [2], we broadly categorized the different attacks into two types, namely highly interactive attacks and low interactive attacks. It is sufficient to define a strict protocol state change directly at the server to detect a low interactive attack [2]. Our goal is to detect and to trace back the highly interactive attacks.

3.2 Considered Network Topology

The network topology in Fig.1 consists of servers providing both encrypted and non-encrypted services. Users from the Internet (including untrusted networks) can connect to any one of the servers. To avoid additional computational loads on the servers, we choose not to implement the detection and traceback schemes at the servers. Rather, these are implemented aside of network elements such as routers. Each of these entities, which is able to detect and to trace back attacks, is called a monitoring stub (MS). Rather than adhering a MS to each server as in [11], MSs are strategically distributed over the entire network. The functionality of a MS is described in the following subsection.

3.3 Functionality of a Monitoring Stub

MSs are packet sniffing entities implemented close to a router. For application level protocols, it is possible to sniff the network packet headers as well as the payloads, and inspect and analyze them at a later instant. However, in case of encrypted protocols, a MS uses the TCP DUMP utility to monitor the TCP headers which are not encrypted. For example, detection of a failed SSH session by a MS requires the system to know how the SSH protocol works in the transport layer level. A client first attempts to establish a connection to the server by sending a SYN packet. The server acknowledges this by sending an ACK packet and a SYN packet of its own. If the client successfully logs onto the server and wants to quit, the client will first initiate the FIN packet. On the contrary, if the server initiates the FIN packet first, it implies that the server is closing the connection either due to an invalid "login" attempt or a time out. If a monitoring stub detects that the server is the first originator of the FIN packet soon after the connection attempt, it deems that event as a "failed session". A MS performs a number of functions: learning normal profiles, monitoring, generating alerts, and tracing back the attacker. The learning, monitoring and alert-generating phases have been depicted in [2]. The way the MSs trace back an attacker, by collaborating with one another, represents the focus of the research work outlined in this paper and will be described in the remainder of this section.

3.4 Proposed Traceback Scheme

The proposed traceback scheme utilizes the extracted features from the encrypted protocol behavior (e.g., failed session rates). Using the proposed scheme, in case of cryptographic protocols like SSH or SSL, the MSs monitor failed sessions over time and correlate these patterns, to identify the path of the attack traffic. Every MS stores, in its database [2], the information regarding failed sessions, which this particular MS observes for both incoming and outgoing traffic. The database of a MS also stores the list of collaborating MSs. After an attack against the encrypted protocol is detected, by the system proposed in [2], the MS nearest to the victim server switches to the "Trace-back mode". From thereon, the MS compares its monitored failed-session pattern with those of its neighboring MSs. For example, MS_a requests MS_b to compare the monitored failed sessions in the direction as shown in Fig.2. MS_b correlates the features of its outbound traffic (S_{out,MS_b}) with that of each incoming traffic of various inbound links [$d_{in,1}$, $d_{in,2}$, ... , $d_{in,n}$] into the confluence point R [16]. The strong correlation coefficients (in the vicinity of one) indicate the possible paths to attackers' subnetworks. In the same fashion, MS_b, which is now in charge of the traceback operation, may request adjacent MSs along those paths to traceback further.

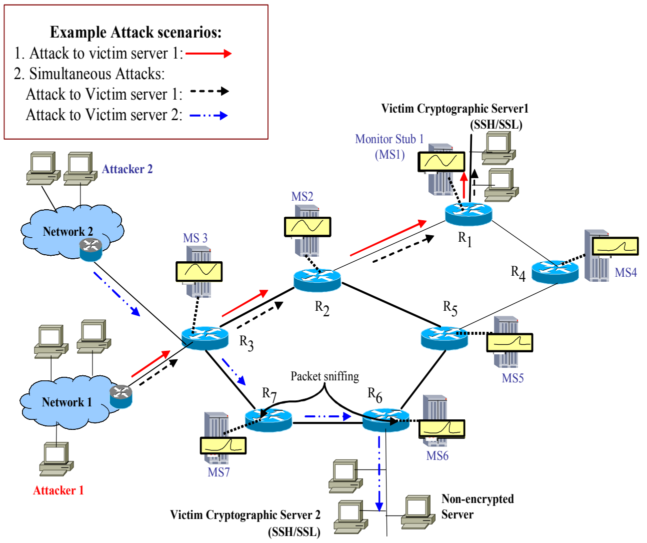


Figure 1: Attack scenarios in an example network architecture.

The failed sessions are monitored by the monitoring stubs in time-slots, λ. The failed sessions are monitored, by each MS, in a window, N, which consists of an integral number of these time-slots. Thus the attack pattern is defined by the length of the time-slot (λ), the size of the 'window' (N), and the monitored features in each slot in the window. These

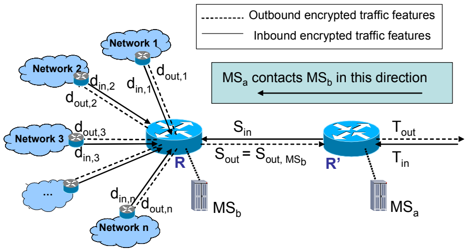
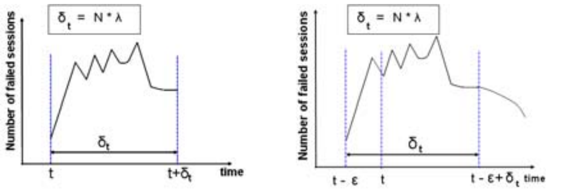


Figure 2: MS_b compares the outbound traffic feature (S_{out}) with the features of each inbound flow at the confluence point, R.



(a) # failed sessions over time seen at MS_a.

(b) # failed sessions over time seen at MS_b.

Figure 3: No. of failed sessions seen at two adjacent MSs: MS_a & MS_b.

metrics are then used to define the vector, V, as shown below:

$$(\; V = \lambda, \, F_1, \, F_2, \, F_3, \, \dots \, , \, F_l, \, \dots \, , \, F_N \;) \tag{1}$$

where F_l indicates the failed session rate, which was monitored during the l-th time slot.

Now let us consider the monitoring stubs, MS_a and one of its neighbors, MS_b. Let the one-way propagation delay between MS_a and MS_b be ξ. At MS_a, the monitoring begins at time t, and continues till $[t+(N^*\lambda)]$ as shown in Fig.3(a). Thus the vector, V_a can be constructed at MS_a by the following expression:

$$(\; V_a = \lambda, \, F_{a_1}, \, F_{a_2}, \, F_{a_3}, \, \dots \, , \, F_{a_l}, \, \dots, \, F_{a_N} \;) \tag{2}$$

At MS_b, the monitoring of F_l commenced at time = $(t$-$\xi)$ as shown in Fig.3(b). The monitoring at MS_b continues till $[(t$-$\xi)+(N^*\lambda)]$. Thus, V_b is constructed at MS_b as follows:

$$(\; V_b = \lambda, \, F_{b_1}, \, F_{b_2}, \, F_{b_3}, \, \dots \, , \, F_{b_l}, \, \dots, \, F_{b_N} \;) \tag{3}$$

The correlation coefficient, denoted by $r_{a,b}(V_a, V_b)$ or simply $r(V_a, V_b)$, between the target vectors, V_a and V_b, is obtained by the following equation [10]:

$$r(V_a, V_b) = \tfrac{1}{N\sigma_a\sigma_b} \textstyle\sum_{l=1}^{N} [V_a(l) - \dot{V_a}][V_b(l) - \dot{V_b}] \tag{4}$$

where $\dot{V_a}$ and σ_a indicate the mean and standard deviation of the monitored features of the vector V_a, respectively.

$$\dot{V_a} = \frac{\sum_{l=1}^{N} F_{a_l}}{N} \tag{5}$$

$$\sigma_a^2 = \frac{\sum_{l=1}^{N} (F_{a_l} - \dot{V_a})^2}{N} \tag{6}$$

The value of $r(V_a, V_b)$ ranges between {-1, 1}. If $r(V_a, V_b)$ yields a value of one, it implies a perfect match between the two patterns represented by the vectors V_a and V_b. If the correlation coefficient value is close to one (say, 0.8 or above), the vectors, V_a and V_b, are said to be highly correlated. If $r(V_a, V_b)$ is in the vicinity of zero or slightly negative, the vectors under comparison are not considered to be similar. A negative value of the correlation coefficient means that the vectors, which are being compared, are totally opposite of each other.

By applying this to the instance in Fig.2, an example set, τ, with k number of strong correlations between the failed sessions of the inbound flows and the outgoing flow (S_{out}) observed by MS_b may be found as follows:

$$\tau = [r(d_{in,1}, S_{out}), r(d_{in,2}, S_{out}), ..., r(d_{in,k}, S_{out})] \tag{7}$$

Table 1: Propagation delays between collaborating MSs.

MS_a	MS_b	*Propagation Delay (ms)*
MS_1	MS_2, MS_4	$\xi_{1,2}$=60, $\xi_{1,4}$=30
MS_2	MS_3, MS_1, MS_5	$\xi_{2,3}$=30, $\xi_{2,1}$=60, $\xi_{2,5}$=30
MS_3	MS_2, MS_7	$\xi_{3,2}$=30, $\xi_{3,7}$=30
MS_4	MS_1, MS_5	$\xi_{4,1}$=30, $\xi_{4,5}$=30
MS_5	MS_2, MS_4, MS_6	$\xi_{5,2}$=30, $\xi_{5,4}$=30, $\xi_{5,6}$=15
MS_6	MS_5, MS_7	$\xi_{6,5}$=15, $\xi_{6,7}$=30
MS_7	MS_3, MS_6	$\xi_{7,3}$=30, $\xi_{7,6}$=30

4. PERFORMANCE EVALUATION

4.1 Simulation Setup

Three simulation scenarios are envisioned. For the first two scenarios, Network Simulator (NS-2) [17] is used to design the initially considered topology as shown in Fig.1. The topological information of the monitoring stubs and the propagation delays, $\xi_{a,b}$ between two adjacent MSs, MS_a and MS_b, are given in Table 1. The simulation parameters are listed in Table 2. In [2], the attack aggressiveness of a SSH password attack is defined as a measure of the strength of the attack in terms of the number of failed login attempts in contrast with that of the valid ones. In the first scenario, a password attack against the victim SSH server-1 is simulated as illustrated in Fig.1. The behavior of the correlation coefficient values under various attack aggressivenesses for this attack is also investigated. In the second scenario, attacks are simultaneously launched against both the victim servers from $Network_1$ and $Network_2$, respectively (Fig.1). Simulations were run five times and the average values are used as results. In the final scenario, the simulation topology is extended to investigate the effectiveness of the proposed traceback scheme under the influence of confluence points.

4.2 Results and Analysis

I. Scenario 1

As illustrated in Fig.1, an attack with attack aggressiveness of 0.45 is launched against the cryptographic server-1 using the simulation topology. As explained earlier, there are seven MSs distributed over this network. Let T_{R_i,R_j} and T_{N_i,R_j} denote the features of traffic directed from Router R_i to router R_j and from $Network_i$ to router R_j, respectively. The attack packets traverse the routers which are monitored by MS_1, MS_2, and MS_3. MS_1, which is closest to the victim server, detects the attack, generates an alert and switches to traceback mode. At first, it compares S_{out,MS_1} with both T_{R_4,R_1} and T_{R_2,R_1} and the resultant correlation coefficients are 0.015 and 0.964, respectively. Therefore, MS_1 exchanges information with MS_2. S_{out,MS_2} is compared with both $T_{R3,R2}$ and $T_{R5,R2}$, resulting in correlation coefficient values equal to 0.972 and 0.012, respectively. Consequently when MS_2 contacts MS_3, S_{out,MS_3} is compared with each of $[T_{N_2,R_3}, \, T_{R_7,R_3}, \, T_{N_1,R_3}]$. The first two comparisons yield correlation coefficients of 0.015 and 0.019, respectively. The last case results in a strong value of correlation coefficient (0.866). Since MS_3 has no other neighboring MS left to exchange its information with, the traceback operation ends at MS_3. Thus the reverse path to the network, which originated the attack is constructed

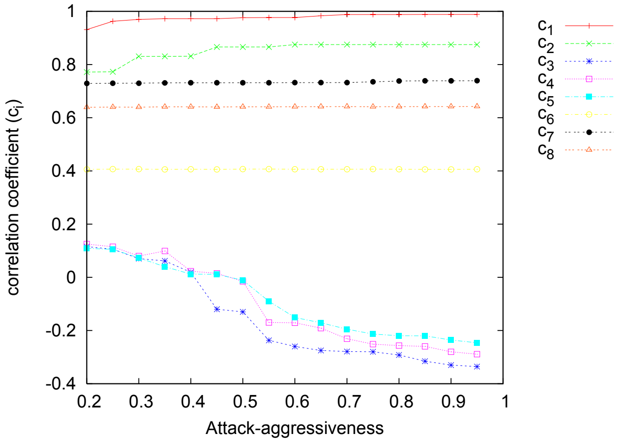


Figure 4: Comparison of correlation coefficients for different attack aggressivenesses.

as: $\{MS_1, MS_2, MS_3, Network_1\}$. MS_3 is, indeed, the monitoring stub that is closest to the attacker's network. In this fashion, the system can reconstruct the path back to the network from which the attack had originated.

For this attack, the correlation coefficients of few pairs of outgoing and incoming links observed by various MSs are listed in Table 3. For ease of description, these correlation coefficients have been indicated as c_1, c_2, and so forth. Furthermore, the correlation coefficient values of these pairs for various attack aggressivenesses ranging from 0.2 to 0.95 are plotted in the graph shown in Fig.4. Both c_1 and c_2 indicate high correlation values, in the vicinity of one, for various attack aggressivenesses. On the other hand, c_3, c_4, and c_5 have correlation values close to zero for initial attack aggressivenesses. For the higher values of attack aggressivenesses, c_3, c_4, and c_5 become negative, indicating further differences between the monitored patterns at the corresponding MSs. On the contrary, c_6, c_7, and c_8 have high correlation values and remain more or less the same with varying attack aggressiveness. This is due to the fact that attack traffic did not traverse through the corresponding MSs.

II. Scenario 2

This scenario investigates the performance of the proposed approach in case of simultaneous attacks from the attackers to more than one victim. The attack aggressivenesses are chosen to be 0.45. As MS_1 detects an attack against victim server-1, it compares S_{out,MS_1} with both T_{R_2,R_1} and T_{R_4,R_1}. The corresponding correlation coefficients are 0.924 and (-0.015), respectively. Consequently, MS_4 is not chosen as the MS in charge for tracing back the attack. As MS_2 is contacted by MS_1, S_{out,MS_2} is compared with T_{R_3,R_2} and T_{R_5,R_2}. The comparisons result in the correlation coefficients of 0.934 and 0.0193, respectively. MS_2 then contacts MS_3 causing S_{out,MS_3} to be compared with each of $[T_{N_1,R_3}, T_{N_2,R_3}, T_{R_7,R_3}]$. The corresponding correlation coefficients are found to be 0.956, 0.0210, and 0.0303, respectively. This result leads to the $Network_1$ which initiated the attack to the victim server-1.

Table 2: Simulation parameters for Scenarios 1 & 2.

Simulation Parameters	*Values*
Number of monitoring stubs	7
Dummy Encrypted Protocol	SSH (over TCP)
Background Traffic	CBR, FTP, Telnet
Simulation Time for Trace back	100s
Time slot, λ	1s
Monitoring Window, N	10
no. of times N was monitored	10

Table 3: Few correlation coefficient values observed by collaborating MSs (for an attack against server-1 with attack aggressiveness of 0.45).

Viewing MS	*Outbound flow* (for V_a)	*Outbound flow direction*	*Inbound flow* (for V_b)	$r(V_a, V_b)$
MS_2	S_{out,MS_2}	R_1	T_{R_3,R_2}	c_1=0.972
MS_3	S_{out,MS_3}	R_2	T_{N_1,R_3}	c_2=0.866
MS_4	S_{out,MS_4}	R_1	T_{R_5,R_4}	c_3=-0.12
MS_5	S_{out,MS_5}	R_4	T_{R_6,R_5}	c_4=0.015
MS_7	S_{out,MS_7}	R_3	T_{R_6,R_7}	c_5=0.011
MS_5	S_{out,MS_5}	R_2	T_{R_6,R_5}	c_6=0.406
MS_6	S_{out,MS_6}	R_5	T_{R_7,R_6}	c_7=0.731
MS_7	S_{out,MS_7}	R_6	T_{R_3,R_7}	c_8=0.640

Table 4: Simulation parameters for Scenario 3.

Simulation Parameters	*Values*
BRITE Topology Type	1 Level: Router only
Number of nodes	100
BRITE Model	Waxman
Node Placement Type	Random
Growth Type	Incremental
Bandwidth Distribution	Heavy-Tailed
n (number of incoming flows to considered confluence point)	20
Simulation Time	100s
Time slot, λ	1s
Monitoring Window, N	10
no. of times N was monitored	10

On the other hand, MS_6, after detecting an attack against victim server-2, compares S_{out,MS_6} with T_{R_5,R_6}. The resultant correlation coefficient (0.0145) is too small to consider MS_5 as the next MS for carrying on with the traceback operation. Meanwhile, S_{out,MS_6} is also correlated with T_{R_7,R_6} which yields a strong correlation coefficient in the vicinity of one (0.924). Consequently, MS_7 is the MS that takes charge of the traceback and compares S_{out,MS_7} with T_{R_3,R_7}. This comparison produces a strong correlation coefficient (0.935). As a result, MS_7 contacts with MS_3. This prompts MS_3 to compare S_{out,MS_3} with each of $[T_{N_1,R_3}, T_{N_2,R_3}, T_{R_2,R_3}]$. The resultant correlation coefficients are 0.0161, 0.971, and 0.024, respectively. Thus, $Network_2$, which is responsible for carrying out the attack, is discovered.

III. Scenario 3

Previously conducted research work such as [16] raise issues regarding the confluence points, where large volumes of attack-traffic from disparate sources along with legitimate traffic converge. In order to verify the applicability of our proposed scheme under the influence of confluence points, we set up a simulation environment consisting of one hundred nodes using the BRITE [18] topology generator for NS-2 with the simulation parameters shown in Table 4.

To highlight our interest about the confluence point, where both normal and malicious traffic merge, we again refer to Fig.2. At the confluence point R, the number of incoming

Table 5: N_{FN} and C_{FN} for various values of k.

k	N_{FN}	C_{FN}		k	N_{FN}	C_{FN}
1	0	-		9	0	-
2	0	-		10	2	0.57, 0.65
3	0	-		11	2	0.63, 0.65
4	0	-		12	4	0.59, 0.64, 0.64, 0.65
5	0	-		13	4	0.51, 0.52, 0.60, 0.64
6	0	-		14	4	0.52, 0.55, 0.60, 0.61
7	0	-		15	5	0.58, 0.62, 0.63, 0.64, 0.65
8	0	-				

flows is $(n + 1)$. The attack features observed by MS_b for each of these flows are represented by $[d_{in,1}, d_{in,2}, ..., d_{in,n}]$ and S_{in}. S_{out} is contributed by all other flows except S_{in} as follows:

$$S_{out} = \sum_{i=1}^{n} d_{in,i} \quad (8)$$

If the actual number of attack flows is k, then S_{out} may be rewritten as follows:

$$S_{out} = \sum_{i=1}^{k} d_{in,i} + \sum_{j=(k+1)}^{n} d_{in,j} \quad (9)$$

In order to attempt to evaluate the performance of the proposed scheme at the advent of combined traffic at the confluence point, we choose a considerably large confluence point with $n = 20$ from the previously described simulation topology. Attack flows originating from k number of disparate source nodes, with simulated features of remote timing attacks, merge at this confluence point and target the victim node, which is four hops away from the considered junction point. Normal traffic from $(n - k)$ nodes also flow via the confluence point based on on-off distributions during the simulation time, which contribute to none or quite low rates of attack-features. By applying the proposed technique, we then attempt to traceback the attackers by varying the value of k from one to fifteen for this confluence point. The performance of the scheme is evaluated in terms of the number of inbound attack flows, which are missed by the scheme due to moderate correlation coefficient values. N_{FN}, the number of false negatives encountered in the scenarios with various values of k is shown in Table 5. As seen from these results, for the values of k from one up to nine, the proposed scheme identifies all possible attack-flows. For higher values of k (from ten and above) the moderate correlation-coefficient values (indicated as C_{FN} in Table 5) in the vicinity of 0.50 present us with the problem in deciding whether to trace the corresponding inbound flows at the confluence point as attacks or not. The convergence of high volumes of attack features at the confluence point contributes to lower correlation coefficients between these actual attack flows and S_{out}. For instance, the average correlation coefficient value between the attack flows and S_{out} for $k = 15$ was 0.728, the highest value being 0.883. In contrast to this, the average correlation coefficient value between the attack flows and S_{out} for values of k from one to nine is computed to be 0.927, which is indeed an indication of strong correlations in those cases. Consequently, the MSs (e.g., MS_b in Fig.2) require to consider tracing back those flows at the confluence point which lead to doubtful correlation coefficient values in the vicinity of 0.50.

5. CONCLUDING REMARKS

In this paper we have addressed the problem of tracing back attackers against encrypted protocols based on a correlation scheme which compares the “failed-session” patterns over time at different monitoring stubs. We have performed simulations and demonstrated the effectiveness of the proposed traceback technique. The simulation results show that the proposed scheme effectively discovers the path back to the attacking host’s network. This work will further facilitate the job of the IDS, which we previously devised to combat attacks against cryptographic protocols [2].

6. REFERENCES

[1] A. Blum, D. Song, and S. Venkataraman, “Detection of Interactive Stepping Stones: Algorithms and Confidence Bounds,” in *Proc. Recent Advance in Intrusion Detection (RAID)*, Sophia Antipolis, French Riviera, France, Sep. 2004.

[2] Z. M. Fadlullah, T. Taleb, N. Ansari, K. Hashimoto, Y. Miyake, Y. Nemoto, and N. Kato, “Combating Attacks Against Encrypted Protocols,” in *Proc. ICC 2007*, Glasgow, Scotland, Jun. 2007.

[3] D. Song and A. Perrig, “Advanced and Authenticated Marking Schemes for IP Traceback,” in *Proc. IEEE INFOCOM*, Anchorage, Alaska, USA, Apr. 2001.

[4] W. Lee and K. Park, “On the Effectiveness of Probabilistic Packet Marking for IP Traceback under Denial of Service Attack,” *in Proc. IEEE INFOCOM*, Anchorage, Alaska, USA, Apr. 2001.

[5] S. Bellovin, “ICMP Traceback Messages,” *IETF Internet Draft*, Mar. 2000.

[6] Z. Gao and N. Ansari, “Tracing Cyber Attacks from the Practical Perspective,” *IEEE Commun. Mag.*, Vol. 43, No. 5, May 2005, pp. 123-131.

[7] S. Savage, D. Wetherall, A. R. Karlin, and T. Anderson, “Practical Network Support for IP Traceback,” *in Proc. SIGCOMM*, Stockholm, Sweden, Sep. 2000.

[8] X. Wang, D. S. Reeves, and S. F. Wu, “Inter-Packet Delay Based Correlation for Tracing Encrypted Connections Through Stepping Stones,” *in Proc.* 7^{th} *European Symposium on Research in Computer Security (ESORICS)*, Zurich, Switzerland, Oct. 2002.

[9] S. C. Lee and C. Shields, “Tracing the Source of Network Attack: A Technical, Legal and Societal Problem,” *in Proc. 2001 IEEE Workshop on Information Assurance and Security*, New York, USA, Jun. 2001.

[10] G. Mansfield, K. Ohta, Y. Takei, N. Kato, and Y. Nemoto, “Towards trapping wily intruders in the large,” *Computer Networks*, Vol. 34, No. 4, Oct. 2000, pp. 659-670.

[11] S. P. Joglekar and S. R. Tate, “ProtoMon: Embedded Monitors for Cryptographic Protocol Intrusion Detection and Prevention,” *J. Universal Computer Science (JUCS)*, Vol. 11, No. 1, Jan. 2005, pp. 83-103.

[12] D. Brumley and D. Boneh, “Remote Timing Attacks are Practical,” in *Proc.* 12^{th} *USENIX Security Symposium*, Washington DC, USA, Aug. 2003.

[13] R. Oppliger, R. Hauser, and D. Basin, “SSL/TLS session-aware user authentication - OR how to effectively thwart the man-in-the-middle,” *Computer Communications*, Vol. 29, No. 12, Aug. 2006, pp. 2238-2246.

[14] Available at CERT, “OpenSSL servers contain a buffer overflow during the SSL2 handshake process,” *CERT Vulnerability Note #102795*, Jul. 2002.

[15] Available at CERT, “OpenSSL servers contain a remotely exploitable buffer overflow vulnerability during the SSL3 handshake process,” *CERT Vulnerability Note #561275*, Jul. 2002.

[16] K. Sakaguchi, K. Ohta, Y. Waizumi, N. Kato, and Y. Nemoto, “Tracing DDoS Attacks by Comparing Traffic patterns Based on Quadratic Programming Methods,” *The Transactions of IEICE B (Japanese Edition)*, Vol. J85-B, No. 8, 2002, pp. 1295-1303.

[17] Network Simulator-ns(version 2). http://isl.edu/nsnam/ns

[18] Brite. http://www.cs.bu.edu/brite/