International Journal of Advances in Computer Networks and its Security – IJCNS
Volume 4 : Issue 1 *[ISSN 2250 – 3757]*
Publication Date : 09 Januarv 2014

A Traffic Signature Sensitive to Client Machines

Kazumasa Oida
Department of Computer Science and Engineering
Fukuoka Institute of Technology
3-30-1 Wajiro-Higashi, Higashi-ku, Fukuoka, 811-0295 Japan
E-mail: oida@fit.ac.jp

***Abstract*—A robust authentication strategy requires a user to provide not only a password but also something the user has. Generally, "what you have" authentication methods require additional hardware to strengthen security. However, such devices are not widely used today. This paper proposes another "what you have" authentication approach, in which a user machine is identified by analyzing video traffic flowing from an authentication server to the user machine. This scheme is inexpensive and resistant to machine and credential theft. Experimental results demonstrate that the traffic signature used in this paper is responsive to a small difference between two user machines.**

***Keywords*—traffic signature, user authentication, two-factor authentication, video stream**

I. Introduction

User authentication is mostly based on passwords. Users who place high value on their accounts might be more willing to spend time configuring a robust authentication strategy. One modern trend is to combine multiple authentication factors to strengthen security [1]. Two-factor authentication requires a prover to provide two distinct factors to a verifier, where there are three distinct authenticating factors: something you have (for example, your house keys), something you know (for example, your password), and something you are (for example, your fingerprints) [2].

The most common client hardware is a standard desktop PC, which is an easy platform to attack. Therefore, an additional hardware device is used for "what you have" authentication. Such devices are smart cards, trusted platform modules (TPMs), etc. However, they are not widely used today since they are complex, may lead to a loss of privacy, reduce control of the computer, or need to be protected against device theft [1]. This paper proposes another "what you have" authentication approach, which overcomes the above-mentioned shortcomings. This is mainly because it does not require additional hardware.

Our approach is based on video traffic analysis. It requires that a server delivers a video stream and a client records packet arrival times to calculate a signature. A video stream is used to obtain a signature that is unique to a 3-tuple (client, server, path), where the path indicates a communication path from the server to the client. The advantages of this approach are that it does not require users to have additional hardware and that whatever cyber-attackers have/know, a server can reject requests from the attackers unless they access the server from a particular place.

This approach is promising in that end hosts can be identified by analyzing video traffic with a high probability [3] and that HTTP streaming is inexpensive and easy to use today [4]. A broad deployment of HTTP streaming solutions already exists. HTTP-based delivery avoids NAT and firewall traversal issues and the ability to use existing HTTP servers instead of specialized servers allows reuse of the existing infrastructure, thereby provides better scalability and cost effectiveness [4].

This paper is organized as follows. Section 2 explains previous work on stream identification. Section 3 shows how to calculate a traffic signature from collected video packets. Section 4 demonstrates how clearly the traffic signature varies when a client PC is replaced with a slightly different one. Section 5 outlines the way how to incorporate our traffic analysis approach into user authentication. Finally, Section 6 presents the conclusions.

II. Related Work

Traffic classification associates traffic flows with applications that generated them based on various parameters (e.g., port number, protocol, or flow statistics) [5,6]. Successful classification results heavily rely on recent advances in machine learning techniques. Stream identification, on the other hand, identifies individual video streams based on sampled statistics derived from video traffic. We have been considering methods for identifying TCP/HTTP-based video streams, by making use of existing classification algorithms [3,7].

We used decay rates, which are statistics defined in the next section, as a signature. An experiment, in which a single client accesses 100 online TV sites, showed that the naive Bayes algorithm [8] correctly classified 94.5% of the TV sites [3]. Therefore, each video delivery server can be associated with a unique signature with a high probability. Since servers send video packets, they directly affect statistical features of video traffic. Meanwhile, clients also affect the features [3], but the effect may be small since HTTP is a stateless protocol. This paper discusses the way how to obtain traffic signatures that are sufficiently sensitive to client machines.

III. Traffic Signature

This section defines decay rate $\beta(m)$ [3]. We focus on the statistic because it enables us to observe traffic variability caused by activities of various communication components (e.g., protocols, computers, routers, etc.) over various time scales m. The following describes how to calculate $\beta(m)$. Let $\{X_k\}$ be a time series, where X_k denotes the number of arriving packets during the k-th time interval of length δ. The m aggregated series $\{X_k^{(m)}\}$ are obtained by dividing $\{X_k\}$ into blocks of length m and averaging the series over each block as

$$X_\ell^{(m)} = \frac{1}{m} \sum_{i=\ell m-m+1}^{\ell m} X_i, \quad \ell = 1,2,\dots,\lfloor N/m \rfloor, \tag{1}$$

where m is a positive integer, N is the size of series $\{X_k\}$, and $\lfloor x \rfloor$ is the largest integer that does not exceed x. The variance of $\{X_k^{(m)}\}$ is given by

$$V^{(m)} = \frac{1}{\lfloor N/m \rfloor - 1} \sum_{k=1}^{\lfloor N/m \rfloor} (X_k^{(m)} - \bar{X})^2, \tag{2}$$

where $\bar{X} = \frac{1}{N}\sum_{i=1}^{N} X_i$. Hereafter, we assume that aggregation level $m(\geq 1)$ is a real number. The decay rate at m_i is defined as

$$\beta(m_i) = \log_{10} V^{(\lfloor m_{i+1} \rfloor)} - \log_{10} V^{(\lfloor m_i \rfloor)}. \tag{3}$$

For simplicity, $\beta(m_i)$ is described as β_i. We obtain 20 decay rates $(\beta_1, \beta_2, \dots, \beta_{20})$ per time series $\{X_k\}$, and for all i, $(\Delta \equiv) \log_{10}(m_{i+1}) - \log_{10}(m_i)$ is a positive constant. All decay rates in this paper are computed with parameter values in Table I.

TABLE I. PARAMETER VALUES USED IN THIS PAPER.

Symbol	Value
interval δ (s)	10^{-5}
size of $\{X_i\}$ N	6×10^6
size of $\{\beta_i\}$ M	20
interval Δ	$\frac{\log(N/50)}{M+1}$
level m_i	$10^{i\Delta}$

IV. Data Analysis

A. Observation Point Sensitivity

Fig. 1 explains how video packets are captured. In the figure, PC 2 accesses an online news channel (France 24) with Internet Explorer Flash Player Add-on. The server delivers online news at a constant rate (448 Kbps) using the TCP protocol. In Fig. 1, by using a port-mirroring hub, all packets destined for PC 2 are also delivered to PC 1. The two PCs collect the same packets from the server separately with WinDump [9].

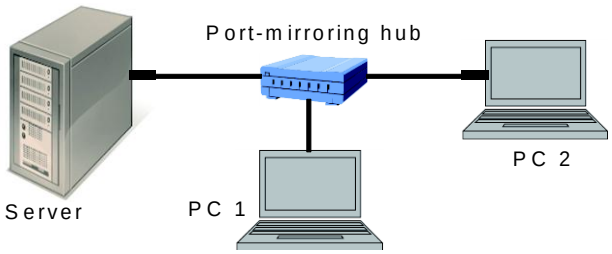

Figure 1. Two PCs capture all video packets transmitted by the server.

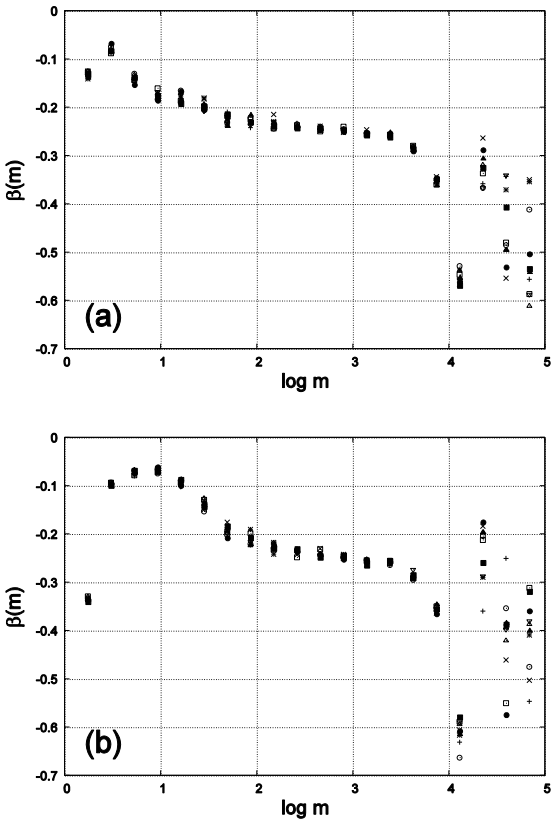

Figure 2. Decay rates measured at (a) PC 1 and (b) PC 2.

Fig. 2 shows decay rates observed at two points. To see variability of β_i, ten samples of $\{\beta_i\}_{1\leq i\leq 20}$ are obtained for each observation point. As shown in the figure, decay rates $\{\beta_i\}_{17\leq i\leq 20}$ tend to fluctuate largely. This is mainly because the number of samples to calculate variances is small. We focus only on stable rates $\{\beta_i\}_{1\leq i\leq 16}$. Although two PCs receive the same packets at almost the same time from the port-mirroring hub, decay rates in Figs. 2a and 2b are clearly different at levels m satisfying $\log(m) < 2$. From Table I, $\log(m) = 2$ indicates the time scale of one millisecond since $\delta \times 10^2 = 10^{-3}$ s. Switching hubs never affect variances at this large time scale.

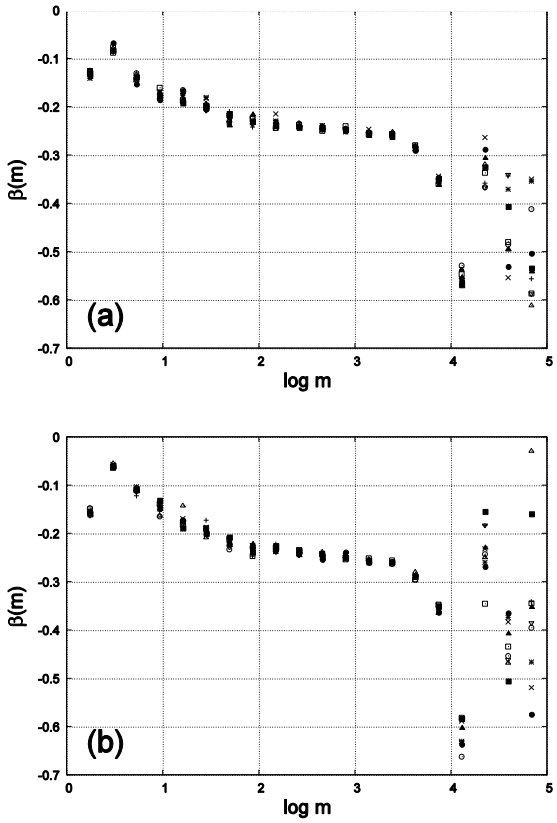

Figure 3. Decay rates measured at PC 1 (a) when PC 2 does not collect packets or (b) when PC 2 does.

The difference occurs because PC 1 only collects packets, while PC 2 performs both packet collection and packet processing. On PC 2, the two jobs are executed simultaneously on each packet arrival and packets tend to arrive in batches when the TCP protocol is used. Accordingly, the two jobs interfere with each other and this interference makes timestamps of packet arrival more incorrect. Thus, decay rates in Fig. 2 differ only at small time scales. Since a degree of the interference is influenced by various mechanisms (e.g., protocols, I/O controllers, device drivers, job scheduling, etc.), it may be difficult to predict decay rates in Fig. 2b even if Fig. 2a is given.

The interference between packet collection and processing affects not only precision in packet arrival times but also the packet processing rate at PC 2. The processing rate slightly fluctuates if the collection task is executed on PC 2. Fig. 3 demonstrates that decay rates measured at PC 1 change over very small levels m depending on whether PC 2 collects packets or not. Note that PC 2 affects decay rates at least through the way it sends TCP acks.

To sum up, if a client captures packets, decay rates measured at another place change slightly (Fig. 3) and they are clearly different from those measured at clients (Fig. 2). Since the difference is due to many factors, it may be difficult for a cyber-attacker at PC 1 to derive Fig. 2b from Fig. 2a.

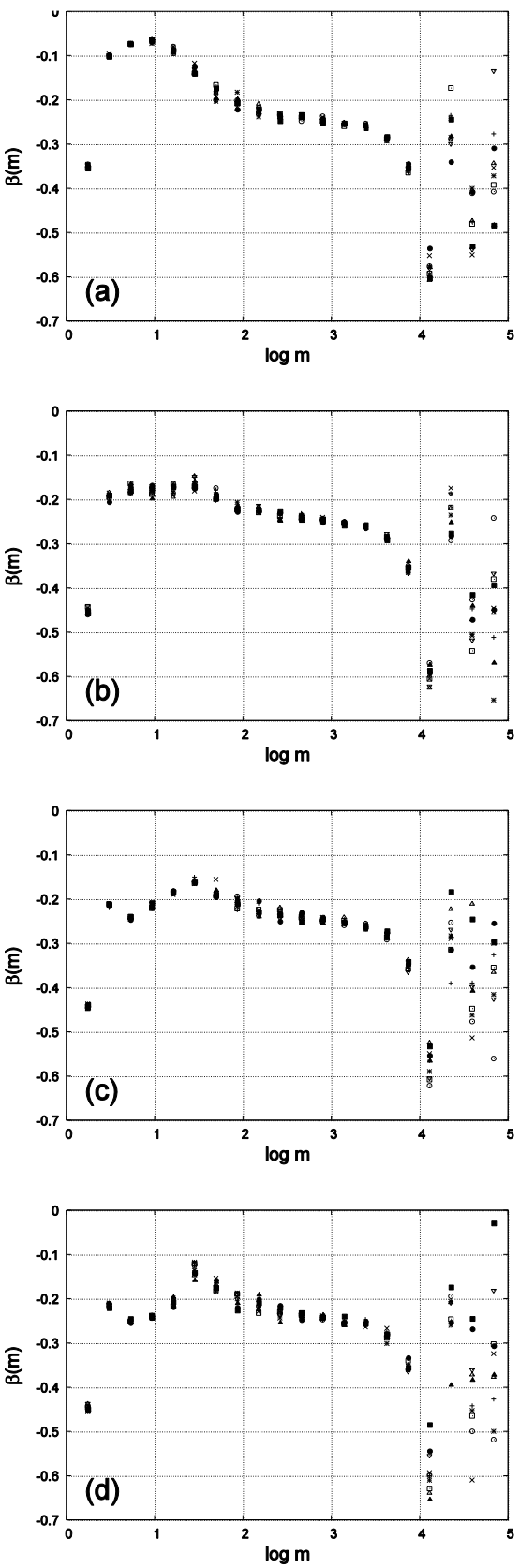

Figure 4. Decay rates measured at client PCs (a)-(d) in Table II.

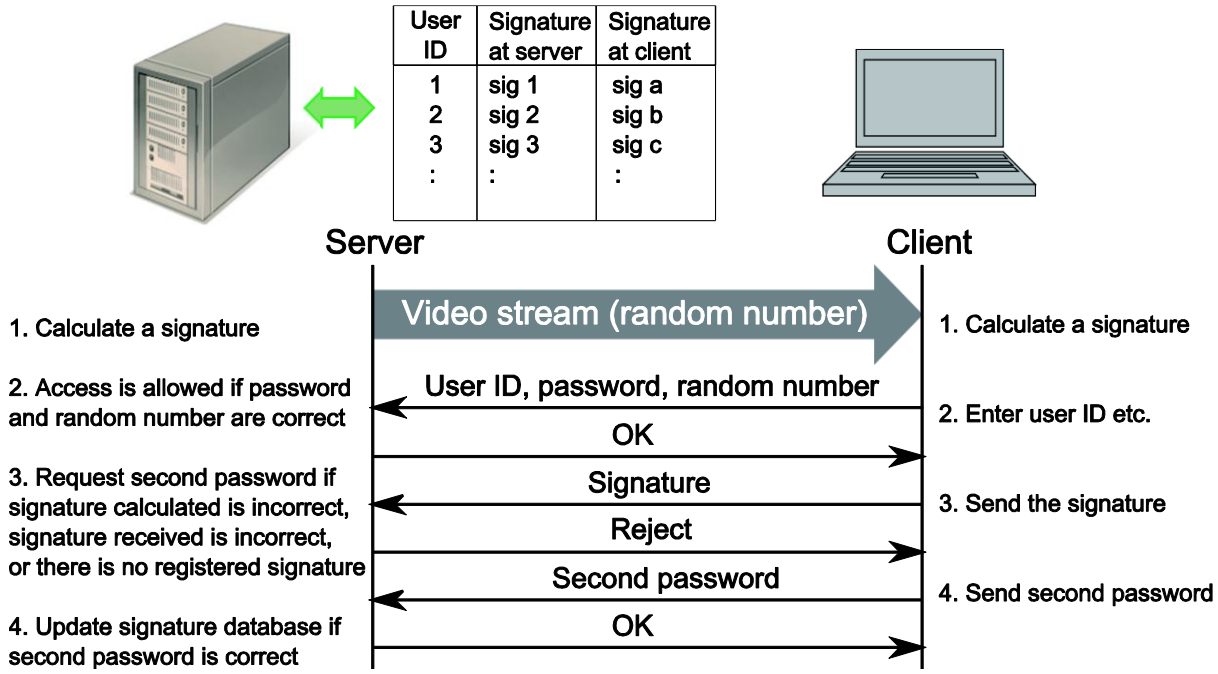

Figure 5. Outline of two-factor authentication.

B. *Client PC Dependence*

Let us clarify how much decay rates depend on client machines. We use four client PCs in Table II. They are all Windows machines produced by the same company, but their software/hardware components are more or less different. All clients receive the same online news (France 24) from the same server along the same communication path with the same streaming parameters (e.g., data rates). Fig. 4 shows decay rates obtained at the four client PCs. Decay rates are sufficiently different between any two PCs. To obtain decay rates in Fig. 4 through simulation or through setting up another client-server systems, attackers will need detailed technical specifications of the server, client, and communication path.

TABLE II. INFORMATION ON FOUR DELL PCS.

PC	Model	Date of purchase	Windows
(a)	XPS420	Jun. 2008	Vista, 32 bit
(b)	XPS435T	Jul. 2010	7, 64 bit
(c)	XPS9100	Jul. 2011	7, 64 bit
(d)	Inspiron ONE2320	Jan. 2012	7, 64 bit

V. Two-factor Authentication

A. *Sequence*

Decay rates of video traffic vary largely by altering some client components. Based on this result, the traffic analysis approach is applied to user authentication. Hereafter, the term signature is used to indicate decay rates $\{\beta_i\}$. Fig. 5 outlines the authentication process over encrypted channels. The following describes the process.

1. The server transmits a video stream, in which a random number that can be recognized by watching the video is included. The client and server calculate a signature separately. (For correct authentication, the server should always transmit a stream in the same software and hardware environment, whereas the video content does not have to be the same).
2. The client sends the user ID, password, and random number. (The random number is used to link the video stream to the user ID. This authentication process may be performed according to the one-time password scheme [10].)
3. The client sends the signature measured at the client. The server requests the second password (i.e., the second authentication starts) if one of the following two cases occurs. (1) The signature calculated at the server does not agree with the signature registered in the database ("signature at server" in Fig. 5) or the signature calculated at the client does not agree with the signature in the database ("signature at client"). (2) There is no entry for the user in the database. (The first case occurs when the user accesses the server by using a different PC or ISP. The second case occurs only if the user accesses the server for the first time. The signature verification can be performed, for example, by using the unsupervised learning algorithm in [7]).
4. If the second password is correct, two signatures measured at the server and client are added to the database. (For achieving correct identification, old signatures should be removed if they are not used from now on).

B. Discussion

In the authentication process, two signatures are obtained at different observation points. Signatures calculated at clients can be used to differentiate client machines (see Fig. 4). However, they may have lost some information related to servers or communication paths. Therefore, the authentication process double-checks two signatures to identify the client, server, and path environments more precisely. Moreover, by means of the double check, a server can request an illegitimate user to access the server from the place that is used by a legitimate user.

Fig. 5 illustrates the case where an authentication server delivers a video stream. In practice, video streams are transmitted by dedicated servers and multiple dedicated servers may be assigned to a user in case of server/network failures. In this case, the database has multiple signatures for the user. Similarly, there may be client machines used only for authentication.

In our approach, a video stream is just a signal to observe a response from the communication system, so we do not need sophisticated video delivery systems. For example, video streams may flow at low constant rates (e.g., less than 100 Kbps) and may not be adaptive to changes in network conditions. Furthermore, user interfaces (e.g., PAUSE and RESUME) to control streams may not be provided. The video content can be used for various purposes (e.g., advertisement).

The traffic-based authentication in this paper has the same targets of challenge as biometric-based authentication, where biometric information (e.g., fingerprint or iris) is required to be (1) reproducibly captured repeatedly, (2) sufficiently different between any two users, and (3) hard to be faked [11]. The traffic-based approach, however, has some advantages. First, video delivery can be optimized such that servers and clients cooperatively achieve targets (1) and (2). Second, biometric authentication becomes useless once biometric information is stolen or secretly obtained (e.g., from fingerprints on a water glass), whereas the traffic-based authentication requires not only knowledge (i.e., passwords and signatures) but also an action (i.e., accessing the server from a particular place).

There are various schemes that make this authentication approach more resistant to cyber-attacks. For example, the client also keeps a signature database to prevent phishing attacks. Session-specific SSL/TLS protocol information can be embedded in the random number to prevent man-in-the-middle (MITM) attacks [11]. It may be necessary to improve the secondary authentication mechanism so that its presence does not make accounts less secure [12].

VI. Conclusions

This paper proposed a novel two-factor user authentication scheme, which was based on stream identification techniques, where client-server pairs are identified through analyzing video traffic. Experimental results showed that signatures derived from video traffic were sufficiently different between any two slightly different client machines when the signatures were measured at client machines. The advantages of this authentication scheme are as follows.

- This scheme does not require users to have additional hardware.
- This scheme is inexpensive and easily achievable by using existing HTTP-based progressive download solutions.
- This scheme is resistant to machine and credential theft in the following sense: an illegitimate user needs not only machines and credentials but also the place used by a legitimate user for server access.

References

[1] M. Jakobsson, R. Chow, and J. Molina, “Authentication - Are we doing well enough?,” IEEE Security & Privacy, vol.10, no.1, pp.19-21, Jan.-Feb. 2012.

[2] D. DeFigueiredo, “The case for mobile two-factor authentication,” IEEE Security & Privacy, vol.9, no.5, pp.81-85, Sept.-Oct. 2011.

[3] K. Oida and K. Yamashita, “Video traffic attributes for end host identification,” Int. J. of Computer and Communication Engineering, vol.1, no.4, pp.396-401, 2012.

[4] O. Oyman, S. Singh, “Quality of experience for HTTP adaptive streaming services,” IEEE Communications Magazine, vol.50, no.4, pp.20-27, April 2012.

[5] A. Callado, C. Kamienski, G. Szabo, B. Gero, J. Kelner, S. Fernandes, D. Sadok, “A Survey on Internet traffic identification,” IEEE Communications Surveys & Tutorials, vol.11, no.3, pp.37-52, 3rd Quarter 2009.

[6] A. Dainotti, A. Pescape, K.C. Claffy, “Issues and future directions in traffic classification,” IEEE Network, vol.26, no.1, pp.35-40, Jan.-Feb. 2012.

[7] K. Oida and N. Nakayama, “Video stream identification for traffic engineering,” Int J. of Future Computer and Communication, vol.2, no.4, pp.275-280, 2013.

[8] G.H. John and P. Langley, “Estimating continuous distributions in Bayesian classifiers,” the Eleventh Conf. on Uncertainty in Artificial Intelligence, pp.338-345, Morgan Kaufmann, San Mateo, 1995.

[9] WinDump is available at http://www.winpcap.org/windump/

[10] IETF RFC 2289, “A one-time password system,” February 1998.

[11] T. Weigold, T. Kramp, M. Baentsch, “Remote Client Authentication,” IEEE Security & Privacy, vol.6, no.4, pp.36-43, July-Aug. 2008.

[12] S. Schechter, “When the Password Doesn’t Work: Secondary Authentication for Websites,” IEEE Security & Privacy, vol.9, no.2, pp.43-49, March-April 2011.