DESIGNING LAWFUL INTERCEPTION SYSTEM

Libor Polčák
Doctoral Degree Programme (1), FIT BUT
E-mail: xpolca03@stud.fit.vutbr.cz

Supervised by: Miroslav Švéda
E-mail: sveda@fit.vutbr.cz

Abstract: This paper focuses on issues of lawful interception. Few recommendations how to design a lawful interception system were written since the lawful interception legislative has been adopted. This paper lists several challenges which are not covered by the aforementioned documents. The challenges contain identification of the communication of a suspect, support of new protocols, detection evasion and others. An original architecture of the Intercept Related Information Internal Interception Function that deals with the identified challenges is proposed and described. The presented architecture utilizes both application logs and network traffic captured to detect the communication of a suspect.

Keywords: Lawful interception, Protocol recognition, User identification, Detection evasion

1 INTRODUCTION

In the beginning, the Internet was formed by military and academic researchers. Later on, more and more people were connecting their computers and LANs. Protection against possible intrusion and even espionage became necessity [1]. Not surprisingly, several law enforcement agencies (LEAs) are interested in network traffic inspection and criminal users monitoring.

In the European Union, the network and content providers are required to cooperate with LEAs and perform surveillance of suspicious criminals. The lawful interception (LI) was authorised by the European Council Resolution in 1995 [2]. Network operators and content providers have to cooperate with LEAs and provide signaling data of monitored subject's communication and the whole content of the communication.

Even though, the LI act is valid since the middle 1990s, there are still some unanswered questions how to effectively detect all communication of the monitored users. This paper aims to identify current challenges in recognition of traffic of a suspect as well as to propose solutions if it is possible. To my best knowledge there is no such overview in up-to-date literature concerning the topic. The paper is organised as follows. Section 2 describes documents about LI and related work. Section 3 brings the review of identified challenges. Section 4 presents the architecture of a key part of the LI system that addresses the identified challenges. Section 5 concludes the paper.

2 RELATED WORK

There are both American [3] and European [4] LI recommendations. European Telecommunications Standards Institute (ETSI) introduced a generic LI architecture for IP services and identified several issues to deal with ([4, 5, 6] and related documents). However, some important aspects of such systems are out-of-scope of ETSI documents. Additionally, new protocols have emerged since the documents were written. For example, an IP address is used as important identifier to recognize the tracked person. Using the Internet protocol version 4 (IPv4) it is common to have the same address

assigned for many sessions. The assignment is typically controlled by provider. The address space of the Internet Protocol version 6 (IPv6) is huge and the addresses that users use for communications are typically no longer directly controlled by providers. The IPv6 identification is not covered by the current recommendations [3, 4].

Cisco developers also proposed a basic architecture of a LI system [7]. However, they do not discuss the problems of designing LI system listed in this paper. Karpagavinayagam et al. [8] proposed a more detailed architecture of VoIP interception system. The system is designed for voice communication and does not deal with other applications.

LI is not the only field that benefits from the application layer protocol recognition. The other fields include network intrusion detection systems [9, 10] and protocol verification [11]. The detection can be done both by exploring application layer data and by statistical analysis of packet properties such as packet size, timing etc. [12]. Since LI system should detect suspect's traffic also by utilising application layer identification, content-aware protocol recognition is feasible.

3 MOTIVATION AND CHALLENGES

This section specifies identified challenges in the design of the LI system that are not covered by the ETSI documents. The aim is to provide knowledge required to design a LI system capable to detect as many current network protocols as possible. The LI system should be able to 1) discover IPv6 address of the target of an interception, 2) detect variety of network protocols, 3) monitor the traffic in a network and search for identifiers triggering the interception, and 4) detect an IPv6 traffic carried over an IPv4 network.

Within the IPv4 Internet computers typically use only one IP address configured either statically or dynamically. The Internet provider usually assigns one IP address per customer. When native IPv6 is available, a block of addresses (up to 2^{80}) is assigned to the end customer while a whole LAN might be connected. When a single computer connects to an IPv6 network, stateless configuration might be used. In this case the network part of the IPv6 address is sent to the computer and the computer itself generates the interface ID. A LI system should be aware that a suspect can be using more than one IP address at the time and that the addresses could change frequently. For example, if IPv6 privacy extension is activated (recent Windows have privacy extension turned on by default) the IPv6 address changes regularly. When a LI is applied directly in a suspect's LAN, the addresses of the interception target can be learned by observing the ICMPv6 neighbour discovery requests and replies. Alternatively, the LI system could periodically examine neighbour cache.

ETSI documents describe LI of Voice over IP (SIP, H.323) and e-mail (SMTP, POP3, IMAP4). Nevertheless, users also communicate using other protocols. For example instant messaging and web browsing are not covered in ETSI documents. The intercepting system should be general and support variety of application protocols. Frequently, a new protocol becomes to be used widely and proprietary protocols change (e.g. protocols for instant messaging). The protocol support of the LI system should be extensible and modifiable.

Although the intercepted traffic would only be a fraction of the traffic in an operator's or content provider's network, it is usually necessary to 1) classify the network traffic according to used application protocol and 2) search for identifiers triggering an interception. Although ETSI [5] does not require network operators to detect application level identifiers of communication of their customers with third parties, an interception is ordinary triggered by information contained in the application layer [4]. Application protocol identification by transport ports is not reliable [12]; thus, LI system should inspect packet content to determine application protocol. High speed application protocol recognition is time consuming task [10]. Therefore, the process should be hardware accelerated. The content provider could also utilise logs that are created by applications providing the service.

A suspect can also use tunneling mechanism to hide his or her traffic. In such scenario, an IP packet is encapsulated into another IP packet (e.g. ISATAP) or is transported within a UDP datagram inside another IP packet (e.g. Teredo). Nowadays, an IPv6 packet is typically transported within an IPv4 packet, since the tunneling mechanisms allow IPv6 connectivity when native connectivity is not available. The tunneling mechanism is automatically active in recent Windows versions and the user may not to be aware of its existence. A peer-to-peer application uses Teredo tunneling method to obtain a globally accessible IP address. Thus, the application is able to accept IPv6 connections established by the peers even though direct IPv4 connection is not possible. Therefore, up-to-date LI system should not ignore tunneled traffic.

Ambiguous data carried through the network are another issue. Bhargavan et al. [11] performed a study of behaviour of mail servers receiving unusual requests and request not completely covered in the protocol specification. They found that the behaviour might depend on the server-side software product and its version. Similarly, an attacker could send unusual content of the network or transport headers to evade LI detection. For example, different TCP segments having the same sequence number and different TTLs might be sent to deceive the system [9]. When it is not clear how the receiver would interpret the traffic, network intrusion detection system (NIDS) may normalize the traffic (i.e. to change the content of a packet to remove the ambiguous information) [13]. A LI system must not change the traffic; thus, the normalization cannot be performed. Since the attacker can also exploit a bug in the software that the system cannot know, such attacks appear fatal for the LI system and might be used to evade detection.

4 PROPOSED ARCHITECTURE

ETSI compose LI system of five functions [6]. This section proposes architecture of the *Intercept Related Information Internal Interception Function* (IRI-IIF). IRI-IIF is the key function to address the challenges defined in section 3. The goal of the IRI-IIF is to create intercept related information (IRI). IRIs contain signaling information about specific communication (e.g. successful and unsuccessful login attempts, logout etc.).

Theoretically, IRI-IIF is able to construct IRIs 1) from the application, which communication is monitored, 2) from the system log or 3) by parsing network communication. It is unlikely that applications would provide IRI information directly [14]. Since the logs contain information already processed by the communicating application, analysing system logs is easier than obtaining the information from the network communication. In order to avoid polling, Syslog capability to send log entries via network can be exploited for the task. Should the LEA ask network provider to intercept application protocols or should the IRI contain information that cannot be obtained from application logs, the network traffic would have to be parsed.

IRI-IIF architecture is depicted in the figure 1. Administrative function sends network identifiers of the traffic to be intercepted through INI1a interface. IRI-IIF forwards the created IRI through INI2 interface. If the content of a communication is intercepted, the network and transport layer identifiers of eavesdropped traffic are sent through CCTI interface. According to the network topology of the intercepting organisation and the services offered, IRI-IIF might be scattered over multiple network devices.

IRI-IIF utilise both preprocessed data (1) and a copy of network traffic (2). Preprocessed data are supplied from two sources: a) Syslog provides data as soon as possible and b) content of ARP and neighbour cache is polled regularly. Network data may be obtained using a) tap devices or b) SPAN ports of active network devices.

Logical (network layer) addresses used by computers identified by physical (data link layer) addresses are stored inside *L2–L3 Address Table* (3). The table is constructed on-the-fly and every computer

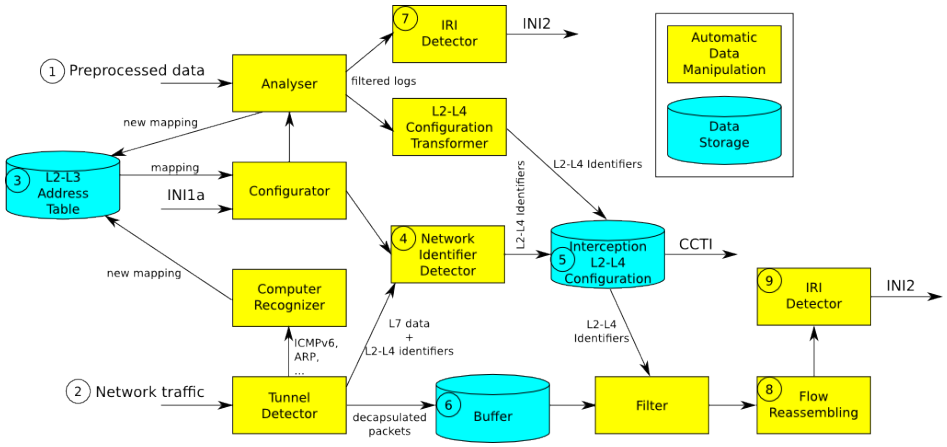

Figure 1: Intercept Related Information Internal Interception Function architecture

is monitored even if no interception is performed. Therefore, if a new interception arrives and the intercepted computer is already online, it is possible to capture data immediately.

Network identifier detection (NID, 4) monitors all communications on the link and filters the traffic according to identifiers in packet headers. If a communication of a specific application is intercepted its flows are identified by IP addresses, transport layer protocol and ports. When all traffic of a suspect is intercepted, the traffic is identified only by an IP address. In order to classify application layer protocols, fast payload processing and pattern matching has to be performed at this stage.

Interception Configuration (5) is based both on preprocessed data analysis and on information recovered from network traffic. A buffer (6) is employed to match delay caused by a) log data processing and 2) NID. IRI records are created from log entries (7) and network traffic (9). Data sources of both IRI Detectors contain only data relevant to interceptions in progress.

To address the problem of unstable and asymmetric routing, it is required to gather data from more network probes and perform *Flow Reassembling* (8). Provided that all paths from the user to the content provider are covered by IRI-IIF devices, all exchanged data would be collected and the stream could be reassembled.

IRI Detector (9) is an application protocol parser. Firstly, it detects elementary network events in unidirectional flow. In the second stage, the elementary events from both directions are correlated and IRI records are created. Languages specialized in description of packet headers and message sequences specification have been already developed (e.g. [9, 10, 11, 15]). During the future research I plan to compare these languages and choose the most suitable for the task. The reason is that specialised languages simplifies support of a new communication protocol.

5 CONCLUSION

Emerging protocols brings new issues to LI system design. This paper portrays that LI system has to deal with user identification, diversity and variability of application protocols, detection evasion techniques and software bugs in network software. The proposed IRI-IIF learns user identification by observing neighbour cache and ARP table and parsing of IPv6 traffic. Specialized languages will be used to describe structure of packet format of application protocols; thus, addition or modification of a protocol will be simplified. IRI-IIF is not able to detect attacks, however, these attacks will be detected by Mediation function. We plan to implement the proposed architecture during the future research.

ACKNOWLEDGEMENT

This work is part of the project VG20102015022 supported by Ministry of the Interior of the Czech republic. I would like to thank project team members for their inspiring notes about the LI. This work was partially supported by the the research plan MSM0021630528. I would like to thank Petr Matoušek for his help during the preparation of the paper.

REFERENCES

[1] Clifford Stoll. Stalking the Wily Hacker. *Commun. ACM*, 31:484–497, May 1988.

[2] The Council of the European Union. COUNCIL RESOLUTION of 17 January 1995 on the lawful interception of telecommunications (96/C 329/01). *Official Journal C 329*, pages 1–6, November 1996.

[3] Congress of the United States of America. *Communications Assistance for Law enforement Act of 1994 (CALEA)*.

[4] European Telecommunications Standards Institute. *ETSI TR 101 943: Telecommunications security; Lawful Interception (LI); Concepts of Interception in a generic Network Architecture*.

[5] European Telecommunications Standards Institute. *ETSI TR 101 944: Telecommunications security; Lawful Interception (LI); Issues on IP Interception*.

[6] European Telecommunications Standards Institute. *ETSI TR 102 528: Lawful Interception (LI); Interception domain Architecture for IP networks*.

[7] Fred Baker, Bill Foster, and Chip Sharp. *RFC 3924 Cisco Architecture for Lawful Intercept in IP Networks*, October 2004. http://tools.ietf.org/html/rfc3924.

[8] Balamurugan Karpagavinayagam, Radu State, and Olivier Festor. Monitoring Architecture for Lawful Interception in VoIP Networks. In *Internet Monitoring and Protection*, July 2007.

[9] Vern Paxson. Bro: a system for detecting network intruders in real-time. *Computer Networks*, 31(23-24):2435 – 2463, 1999.

[10] Ruoming Pang, Vern Paxson, Robin Sommer, and Larry Peterson. binpac: A yacc for Writing Application Protocol Parsers. In *Proceedings of the 6th ACM SIGCOMM conference on Internet measurement*, IMC '06, pages 289–300, New York, NY, USA, 2006. ACM.

[11] Karthikeyan Bhargavan and Carl A. Gunter. Network event recognition. *Formal Methods in System Design*, 27:213–251, 2005.

[12] Arthur Callado, Carlos Kamienski, Géza Szabó, Balázs Péter Gerö, Judith Kelner, Stênio Fernandes, and Djamel Sadok. A survey on internet traffic identification. *Communications Surveys Tutorials, IEEE*, 11(3):37–52, 2009.

[13] Mark Handley, Vern Paxson, and Christian Kreibich. Network Intrusion Detection: Evasion, Traffic Normalization, and End-to-End Protocol Semantics. In *Proceedings of the 10th conference on USENIX Security Symposium - Volume 10*, SSYM'01, 2001.

[14] Internet Architecture Board, Internet Engineering Steering Group. *RFC 2804 IETF Policy on Wiretapping*, May 2000. http://tools.ietf.org/html/rfc2804.

[15] Nikita Borisov, David J. Brumley, and Helen J. Wang. A Generic Application-Level Protocol Analyzer and its Language. In *In 14th Annual Network and Distributed System Security Symposium*, 2007.