IOSR Journal of Computer Engineering (IOSR-JCE)
e-ISSN: 2278-0661, p- ISSN: 2278-8727Volume 16, Issue 1, Ver. VII (Feb. 2014), PP 01-04
www.iosrjournals.org

Cells breaks the Tor's Anonymity: Onion Router

Ms. S.Sivaranjani, Ms. R.Backiyalakshmi B.E, M.Tech.,
Department Of Computer Science and Engineering, PRIST University, Puducherry.
Assistant Professor, Department Of Computer Science and Engineering, PRIST University, Puducherry.

Abstract: *To hide the communication of users, the anonymity systems pack the application data into equal – sized cells. The size of IP packets in the Tor network can be very dynamic and the IP layer may be repack cells. A new cell-counting attack against Tor allows the attacker to confirm anonymous communication relationship among users very quickly. By varying the number of cells in the target traffic at the malicious exit onion router, the attacker can embed a secrete signal into variation of cell counter of the target traffic and it will be carried and arrive at the malicious entry onion router. Then an accomplice of the attacker will detect the signal based on received cells and confirm the communication among the users. There are several features of this attack. First, it is highly efficient and confirms very short communication session with only tens of cells. Second, this attack is effective and detection rate approaches 100% with a very low false positive rate. Third, it is possible to implement the attack in a way that appears to be very difficult for honest participants to detect.*
Keywords*: Anonymity, cell counting, mix networks, signal, Tor.*

I. Introduction

Anonymity has become a necessary and legitimate aim in many applications. Here the encryption alone cannot maintain the anonymity required by users. Generally speaking, mix techniques can be used for either message-based or flow-based anonymity applications. Research on flow-based anonymity applications has recently received great attention in order to preserve anonymity in low-latency applications, including Web browsing and peer-to-peer file sharing.

To degrade the anonymity service provided by anonymous communication systems, traffic analysis attacks have been used. The Existing traffic analysis attacks can be categorized into two types: passive traffic analysis and active watermarking techniques. The active watermarking technique has recently received much attention to improve the accuracy of attack. In this technique is to actively introduce special signals into the sender's outbound traffic with the intention of recognizing the embedded signal at the receiver's inbound traffic.

The core contribution of the paper is a new cell counting based attack against Tor network. This attack confirms anonymous communication relationship among users accurately and quickly and it is difficult to detect.

The attacker at the exit onion router detects the data transmitted to a destination and then determines whether the data is relay cell or control cell in Tor. After excluding control cells, manipulate the number of relay cells in the circuit queue and flushes out all cells in the circuit queue. This way the attacker can embed a signal into the variation of cell count during a short period in the target traffic. To recover the embedded signal, the attacker at the entry onion router detects and excludes the control cells, record the number of relay cells in the circuit queue and recover the embedded signal.

The main features of cell-counting based attack are: (1) This attack is highly efficient and can quickly confirm very short anonymous communication sessions with tens of cells. (2) It is effective and detection rate approaches 100 % with very low false positive rate. (3) It makes difficult for others to detect the presence of the embedded signal. The Time – hopping based signal embedding technique makes the attack even harder to detect.

II. System Architecure

There are two types of cells: Control cell and Relay cell. The CELL_CREATE or CELL_CREATED used for setting up a new circuit. CELL_DESTROY used for releasing a circuit. Relay cell is used to carry TCP stream data from client to bob. Some of the relay commands are: RELAY_COMMAND_BEGIN, RELAY_COMMAND_END, RELAY_COMMAND_ DATA, RELAY_COMMAND_SENDME, and RELAY_ COMMAND_DROP.

The Onion router (OR) maintains the TLS connection to other OR. Onion proxy (OP) uses source routing and chooses several ORs from cached directory. OP establishes circuit across the Tor network and negotiates a symmetric key with each OR, one hop at a time, as well as handle TCP stream from client application. The OR on other side of circuit connects to the requested destination and relay the data.

The OP will sets up TLS connection with OR1 using protocol, through this connection , OP sends CELL_CREATE cell and uses Diffie-Hellman (DH) handshake protocol to negotiate a base key $k1=g^{xy}$ with

OR1. Form this key; a forward symmetric key kf1 and backward key kb1 are produced. This way first hop circuit C1 is created. Similarly OP extends the circuit to second and third hop. After circuit is setup, OP sends a RELAY_COMMAND_BEGIN cell to the exit onion router and cell is encrypted as {{{Begin<IP, Port>}$_{kf3}$}$_{kf2}$}$_{kf1}$. While the cell traverses through circuit each time the layer of onion skin are removed one by one. At last the OR3 last skin is removed by decryption then it open a TCP stream to a port at the destination IP, which belongs to bob. The OR3 sets up a TCP connection with bob and sends a RELAY_COMMAND_CONNECTED cell back to Alice's OP. Then the client can download the file.

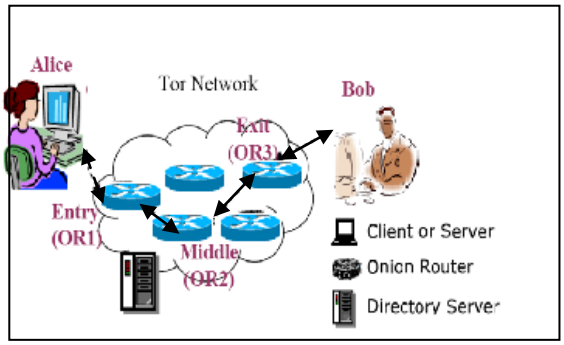

Fig. 1 Tor Network

III. Processing Cell At Onion Router

The TCP data is received by OR from port A and it is processed by TCP and TLS protocols. Then the processed data is delivered to the TLS buffer. The read event is called to read and process the data pending in the TLS buffer. This read event will pull the data from TLS buffer into the input buffer. Then the read event process cells from input buffer one by one. Each OR has routing table which maintains map from source connection and circuit ID to destination connection and circuit ID.

The transmission direction of the cell can be determined by the read event. To append the cell to the destination circuit the corresponding symmetric key is used to decrypt / encrypt the payload of the cell, replace the present circuit ID with destination circuit ID. The cell can be written directly for the destination connection if there is no data waiting in output buffer and the write event is added to the event queue. After calling the write event, the data is flushed to TLS buffer of destination. Then write event pull as many cells as possible from circuit to output buffer and add write event to event queue. The next write event carry on flushing data to output buffer and pull cells to output buffer else the cell queued in circuit queue can be delivered to network via port B by calling write event twice.

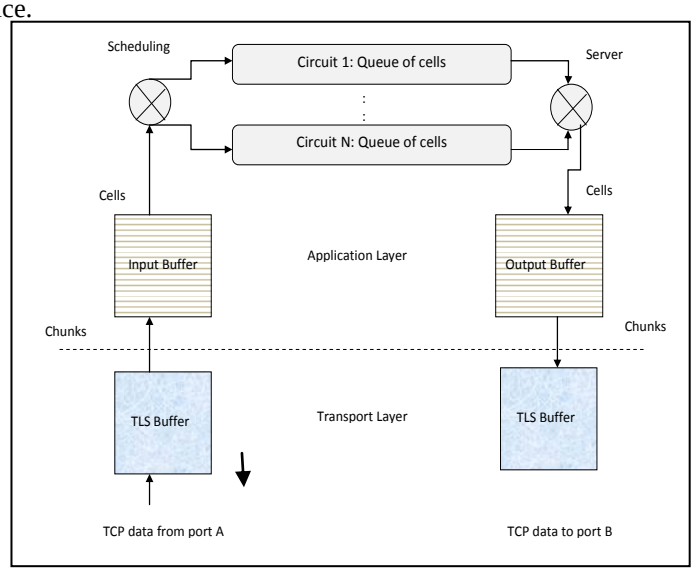

Fig. 2 Processing the cells at Onion router

IV. Cell – Counting Based Attack

The IP packets in Tor network is very dynamic and based on this the cell – counting based attack implemented.

(A) **Dynamic IP packets over Tor** : The application data will be packed into equal sized cells (512-B). While the packets transmitted over the Tor network it is dynamic. Because of this reason the size of packets from sender to receiver is random over time and large numbers of packets have varied in sizes. The main reason for this is the varied performance of OR cause cells not to be promptly processed and also if network is congested, cells will not delivered on time, because of this the cell will merge and non-MTU(Maximum Transmission Unit) sized packets will show up.

(B) Work-flow of Cell – Counting based attack:

Step 1: SELECTING THE TARGET :- The attacker log the information at the exit OR, including the server host IP address and port for a circuit and circuit ID and uses CELL – RELAY-DATA to transmit the data stream.

Step 2: ENCODING THE SIGNAL :- Until the write event is called the CELL – RELAY – DATA will be waited in the circuit queue. After the write event is called then the cells are flushed into output buffer. Hence the attacker can manipulate the number of cells flushed to the output buffer all together. This way the attacker can able to embed the secret signal. To encode bit 1, the attacker can flushes three cells from circuit queue and for bit 0, flushes one cell from circuit queue. Step 3: RECORDING PACKETS :- After the signal is embedded in the target traffic it will be transmitted to the entry OR along with target traffic. The attacker at the entry OR will record the received cells and related information and need to determine whether the received cells are CELL – RELAY – DATA cells.Step 4: RECOGNIZING THE EMBEDDED SIGNAL :- The attacker enters the phase of recognizing the embedded signal with the recorded cells. For this used the recovery mechanisms. Once the original signal is identified the attacker can link the communication between Alice and Bob.

There are two critical issues related to attack: (1) Encoding signals at exit OR: *Two cells are not enough to encode "1" bit.* Because if the attacker uses two cells to encode bit "1" then it will be easily distorted over network and also hard to recover. When the two cells arrive at the input buffer at the middle OR, the first cell will be pulled into circuit queue and then if the output buffer is empty, the first cell will be flushed into it. Then the second cell will be pulled to the circuit queue. Since the output buffer is not empty, the second cell stays in the circuit queue. When the write event is called, the first cell will be delivered to the network, while the second cell written to the output buffer and wait for the next write event. Consequently, two originally combined cells will be split into two separate cells at the middle router. So the attacker at the entry OR will observe two separate cells arriving at the circuit queue. This cells will be decoded as two "0" bits, leading the attacker to a wrong detection of the signal. To deal with this issue the attacker should choose at least three cells for carrying bit "1".

For transmitting cells, proper delay interval should be selected: If the delay interval among the cells is too large, users are not able to tolerate the slow traffic and to transmit the data will choose another circuit. When this condition happens the attack will fail. And if the delay interval is too small, then it will increase the chance that cells may combined at middle OR.

(2) *Decoding signals at the entry OR:* Distortion of signal: Anyway the combination and division of the cells will happen due to unpredictable network delay and congestion. This will cause the embedded signal to be distorted and the probability of recognizing the embedded signal will be reduced. Because of this distortion of the signal, a recovery mechanism can be used, that recognize the embedded signal.

The combination and division of cell can be categorized into four types: (1) Two types of the cell division for the unit of the signal and (2) Two types of the cell combination for different units of signal. To deal with these types of division and combination types of the cells the recovery algorithm can be used. If the number of cells recorded in the circuit queue is smaller than the number of the original signal are recovered as either two types of cell division for the unit of the signal. Suppose the number of cells recorded in the circuit queue is larger than the number of cells for carrying the signal; the recovered signal will be either two of the cell combination for different units of signal. When the signals are recovered in these types with $k \leq 2$, can consider that these signals are successfully identified otherwise cannot be identified.

(C) **Attack Delectability:** To improve the attack invisibility can adopt the time-hopping-based signal embedding technique, which can reduce the probability of interception and recognition. The principle of this technique is, there exit random intervals between signal bits. At the exit OR, the duration of those intervals are varied according to a pseudorandom control code which is known to only the attackers. To recover this signal, the attacker at the entry OR can use the same secret control code to position the signal bits and recover the whole signal. If the interval between the bits is large enough, the inserted signal bits appear sparse within the target traffic and it is difficult to determine whether groups of cells are caused by network dynamics or intention. Therefore the secret signal embedded into the target traffic is no different than the noise. And when a malicious entry node has confirmed the communication relationship, it can separate the group of cells by adding delay between the cells so that not even the client can observe the embedded signal. In this paper a signal is embedded into the target traffic, which implies a secrete sequence of groups of one and three cells. With the time-hopping technique, groups of one and three cells are separated by random intervals and it is hard to

differentiate them from those caused by network dynamics and since the embedded signal is very short and only known to attacker, can conclude that it is very difficult to distinguish traffic with embedded signals from normal traffic based on this very short secret sequence of cell groups.

V. Conclusion And Future Work

In this paper, we presented a cell-counting based attack against Tor network. This can confirm the anonymous communication among the user quickly and accurately and it is very difficult to detect. The attacker at the exit OR manipulates the transmission of cells from the target TCP stream and embeds a secret signal into the cell counter variation of the TCP stream. Then the attacker at the entry OR recognizes the embedded signal using developed recovery algorithms and links the communication relationship among the users. In this attack the detection rate is monotonously increasing function with the delay interval and decreasing function of the variance of one way transmission delay along a circuit. This attack could drastically and quickly degrade the anonymity service that Tor provides. Due to the fundamental design of the Tor network, defending against this attack remains a very challenging task that we will investigate in future work.

References

[1] W. Yu, X. Fu, S. Graham, D. Xuan, and W. Zhao, "DSSS-based flow marking technique for invisible traceback," in *Proc. IEEE S&P*,May 2007, pp. 18–32.

[2] N. B. Amir Houmansadr and N. Kiyavash, "RAINBOW: A robust and invisible non-blind watermark for network flows," in *Proc.16thNDSS*, Feb. 2009, pp. 1–13.

[3] V. Shmatikov and M.-H. Wang, "Timing analysis in low-latency MIX networks: Attacks and defenses," in *Proc. ESORICS*, 2006, pp. 18–31.

[4] V. Fusenig, E. Staab, U. Sorger, and T. Engel, "Slotted packet counting attacks on anonymity protocols," in *Proc. AISC*, 2009, pp. 53–60.

[5] X. Wang, S. Chen, and S. Jajodia, "Tracking anonymous peer-to-peer VoIP calls on the internet," in *Proc. 12th ACM CCS*, Nov. 2005, pp.81–91.

[6] K. Bauer, D. McCoy, D. Grunwald, T. Kohno, and D. Sicker, "Lowresource routing attacks against anonymous systems," Univ. Colorado Boulder, Boulder, CO, Tech. Rep., Aug. 2007.

[7] X. Fu, Z. Ling, J. Luo, W. Yu, W. Jia, and W. Zhao, "One cell is enough to break Tor's anonymity," in *Proc. Black Hat DC*,Feb.2009[Online]. Available: http://www.blackhat.com/presentations/bh-dc-09/Fu/ BlackHat-DC-09-Fu-Break-Tors-Anonymity.pdf

[8] R. Dingledine, N. Mathewson, and P. Syverson, "Tor: Anonymity online," 2008 [Online]. Available: http://tor.eff.org/index.html.en

[9] R. Dingledine and N. Mathewson, "Tor protocol specification,"2008 [Online]. Available: https://gitweb.torproject.org/torspec.git?a=blob_plain;hb=HEAD;f=tor-spec.txt

[10] J. Reardon, "Improving Tor using a TCP-over-DTLS tunnel," Master's thesis, University of Waterloo, Waterloo, ON, Canada, Sep. 2008.