Proceedings of the Asia-Pacific Advanced Network ***2012*** *v. 34, p. 27-34.*
http://dx.doi.org/10.7125/APAN.34.4
ISSN 2227-3026

Using abnormal TTL values to detect malicious IP packets

Ryo Yamada and Shigeki Goto
Department of Computer Science and Engineering, Waseda University, 3-4-1 Okubo, Shinjuku, Tokyo 169-8555, Japan

E-Mails: {ryo, goto}@goto.info.waseda.ac.jp
*Tel.: +81-3-5286-3182; Fax: +81-3-5286-3182

Abstract: In general, an IP packet passes through less than 30 routers before it reaches a destination host. According to our observations, some IP packets have an abnormal time-to-live (TTL) value that is decreased by more than 30 increments from the initial TTL. These packets are likely to be generated by special software. We assume that IP packets with strange TTL values are malicious. This study investigates this conjecture through several experiments, and the results show that malicious packets can be discriminated from legitimate ones by observing only TTL values.

Keywords: network security; malicious packets; IP header; TTL; hop count

1. Introduction

The amount of malicious traffic is continuously increasing. Some studies report that 75% of e-mail traffic is occupied by spam messages [1]. These messages have *links* to online-dating sites, which try to swindle users, or stepping stone sites for downloading malicious software (*malware*). A typical malware program is *botnet* software, which is composed of malware-infected computers controlled by a *herder* that delivers distributed denial-of-service (DDoS) attacks or performs host scanning. The number of infected botnet hosts exceeded 350,000 at the end of 2011 [2]. Therefore, it is important for the internet users to be able to discriminate malicious packets from legitimate ones while the users' machines are connected to the Internet.

This paper proposes a new method for detecting *malicious* packets by observing only the time-to-live (TTL) value in the internet protocol (IP) header. This method is based on a simple concept. Usually, an IP packet passes through less than 30 routers before it reaches a destination

host. However, our observation reveals that some IP packets have an extraordinary TTL value that is decreased by more than 30 increments from the initial TTL value. These packets are likely to be generated by special software. We assume that IP packets with abnormal TTL values are malicious. This paper investigates this conjecture through several experiments with packets captured from a working network in our university. The results show that it is possible to discriminate malicious packets from legitimate ones by observing only TTL values.

So far, several methods have been proposed for detecting malicious packets. Some methods are based on machine learning, with numerous signature files or deep packet inspections. [3, 4] In contrast, our method requires no complex process for discrimination; it is a simple method that can be easily combined with other methods, if necessary. Other methods based on signature database for detecting intrusions are also popular. However, the signature database needs to be updated frequently for detecting latest attacks, while our method does not need any signature database and it is not necessary to be updated regularly.

The rest of this paper is organized as follows. Section 2 explains the proposed new method. Section 3 outlines the evaluation procedure and the dataset. Section 4 shows our evaluation results. Section 5 presents the conclusion and a brief explanation of the future plan.

2. Detecting malicious packets using TTL values

2.1. Calculating the hop count with the TTL value

A computer sets the TTL at the *initial* value when it sends an IP packet. The initial value of a TTL is specific to the operating system (OS) of the host machine, a protocol, and the network socket API. Table 1 illustrates the initial TTL values used by popular OSs [5]. For example, the initial value of TTL for TCP, UDP, and ICMP packets sent from Windows XP is 128 by default, while for FreeBSD, it is 64.

It is possible to estimate the number of routers (*hop count)* along a path from the sender host to the destination host. The estimation is based on the following facts.

1. The TTL initial values of popular OSs are well separated [5]. It is easy to judge the initial value. It is not necessary to identify the OS. We only need the initial TTL value.

2. The maximum count of routers along a path (hop count) is around 30 [6, 7].

If a host receives an IP packet with TTL equal to value t, then the initial value of the TTL, t_0 , can be assumed to be the minimum value that is larger than t in Table 1. The hop count can be calculated for the received packet as follows: (*hop count*) = $t_0 - t$. For example, when a host receives a packet with a TTL value of 120 (t = 120), the minimum number in Table 1 that is

larger than t is 128 (t_0 = 128). Therefore, the hop count is 8 (128 − 120 = 8).

Table 1. Initial TTL values of popular operating systems.

OS	Protocol	Initial TTL
Linux 2.4 kernel	ICMP	255
BSDI BSD/OS 3.1 and 4.0	ICMP	255
Windows Server 2008	TCP, UDP, ICMP	128
Windows7	TCP, UDP, ICMP	128
Windows XP	TCP, UDP, ICMP	128
Linux RedHat 9	TCP, ICMP	64
FreeBSD5	ICMP	64
MacOS X (10.5.6)	TCP, UDP, ICMP	64
AIX	TCP	60

2.2. Normal and abnormal TTL values

We captured the IP packets coming into our university campus network from December 10, 2011, to December 13, 2011. Figure 1 shows the distribution of the number of IP addresses from a unique source for each TTL value. Note that the vertical axis is marked with a logarithmic scale. A majority of the captured packets have a TTL value (t) between the initial TTL value (t_0) set by popular OSs and a TTL value that is 30 increments less than the initial value ($t > t_0 - 30$). However, some packets have a TTL value that is more than the 30 increments less than the initial value ($t_0 - 30 > t$). This paper classifies TTL values into *normal* TTL and *abnormal* TTL according to the value of the TTL (t) in the IP header of the captured packet.

Normal TTL: if $30 < t_G \leqq 64$, or $98 < t_G \leqq 128$, or $225 < t_G \leqq 255$.

Abnormal TTL: if $1 < t_G \leqq 30$, or $64 < t_G \leqq 98$, or $128 < t_G \leqq 225$.

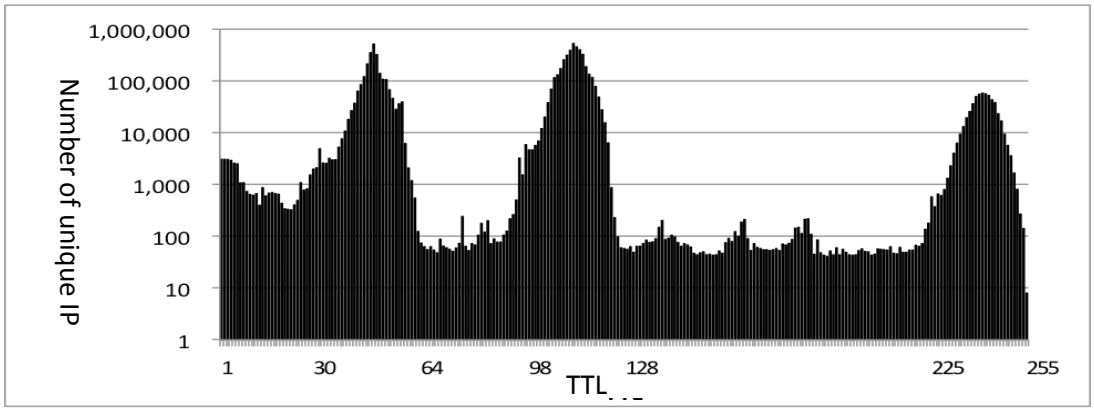

Figure 1. Number of unique IP addresses for each TTL value.

Some old OSs set a low initial TTL value such as 30 or 32. Despite this fact, we classify TTL values less than 30 as abnormal because they are produced by old OSs and they are not commonly used now [8].

2.3. Proposed method

When a packet with abnormal TTL is observed, there are two possible reasons why TTL is abnormal. First, the packet actually came through more than 30 routers. However, a packet rarely hops more than 30 routers, as mentioned earlier [4, 5].

Second, a sender modifies the initial TTL value. We consider that packets with an abnormal TTL were sent with a malicious intent. On the basis of this assumption, we propose a new method for distinguishing malicious packets from legitimate ones.

3. Evaluation

3.1. Outline of experiments

First, IP packets were captured at the gateway router of our campus network. Then, we applied the new method to classify packets into a normal set (P_NOR) and an abnormal set (P_ABN). Our conjecture is that the abnormal packets (P_ABN) are malicious. To verify the conjecture, three existing methods were applied to P_NOR and P_ABN to measure the maliciousness of the packets. If P_ABN receives a higher *maliciousness* score than P_NOR, the new method is able to significantly discriminate malicious packets.

Following are the three existing methods that were used to measure the maliciousness of the packets. (1) Port numbers: well-known port numbers are more likely to be used in malicious connection attempts. (2) Full kernel malware [9]: this method of *fingerprinting* detects malicious packets that are sent from hosts infected by full kernel malware. (3) Snort IDS: a popular intrusion detection system (IDS), *Snort*, is applied to P_ABN and P_NOR to generate alert messages when it finds some incident.

3.2. Dataset: Captured packets

The packet data were captured at the gateway router of our university network, which has three incoming links: two academic networks and one commercial network. We captured the incoming packets from outside of the campus (inbound) because the TTL values were meaningful for the incoming packets. The following chart shows the periods for capturing the data.

Experiment 1:

- Normal TTL packets: 18:30 (Jan 11, 2012) to 20:30 (Jan 13, 2011)

- Abnormal TTL packets: 18:20 (Jan 11, 2012) to 01:10 (Apr 27, 2012)

Experiment 2:

- Normal TTL packets: 18:30 (Jan 11, 2011) to 20:30 (Jan 13, 2012)
- Abnormal TTL packets: 18:20 (Jan 11, 2011) to 01:10 (Apr 27, 2012)

Experiment 3:

- Normal TTL packets: 14:55 (Nov 6, 2011) to 17:10 (Nov 6, 2011)
- Abnormal TTL packets: 14:55 (Nov 6, 2011) to 17:00 (Nov 15, 2011)

4. Results

4.1. Experiment 1: Distribution of destination port numbers

Table 2 shows the statistics of the destination port numbers specified for normal TTLs (P_NOR) and abnormal TTLs (P_ABN). The table shows only the top 10 port numbers.

For normal TTLs (P_NOR), packets bound to port 445 occupy only 7.69% of the total packets, while for abnormal TTLs (P_ABN), packets bound to port 445 comprise 36.30%, which is five times higher than that for P_NOR. Port number 445, which is known to be highly vulnerable, is used by *direct hosting server message block* (SMB) through TCP/IP. Port number 3389 is also indicated frequently in abnormal TTLs (P_ABN), and is used for Windows Remote Desktop as well as for a *port scan* to issue warning reports [14]. The experimental results indicated that most of the abnormal TTL packets (P_ABN) were malicious.

Table 2. Distribution of destination port numbers.

Normal TTL		Abnormal TTL	
Destination port	Rate of occurrence	Destination port	Rate of occurrence
445	7.69%	445	36.30%
80	3.27%	3389	10.73%
8080	0.76%	80	5.60%
1433	0.72%	22	2.54%
8284	0.50%	443	1.68%
3389	0.41%	25	1.09%
443	0.38%	1433	0.78%
23	0.32%	16000	0.67%
6881	0.22%	8443	0.58%
3128	0.21%	21	0.54%

4.2. Experiment 2: Full kernel malware and TCP fingerprinting

TCP fingerprinting is a method for identifying the OS that sent a packet. It inspects the parameters in the TCP packet header [7] for detecting certain differences among the parameters in the TCP header issued by various implementations of OSs. Full-kernel malware is a malicious program that is produced by a customized (full kernel) operating system. Kisamori et al. [9] successfully applied a TCP fingerprinting method to the captured data using honey pot (i.e., CCC, Cyber Clean Center, Dataset 2008 and 2009 [10]). They collected the fingerprint signatures that they judged to be "unknown." The fingerprint "unknown" indicates a signature that is not a known signature from popular OSs, but is produced by a customized operating system. Kisamori et al. identified 43 unique "unknown" signatures that appear frequently in the attack data at CCC honey pot. The "unknown" signatures are called malware workshop signatures (MWS signatures).

We used p0f software for fingerprinting [11]. P0f is an open-source fingerprinting tool used by Kisamori et al. [9], which extracts the parameters from libpcap-formatted TCP packets to identify which OS produced the packets.

We applied p0f to our dataset. The results are classified into three categories: known signatures that are produced by popular OSs, unknown signatures that do not match any OS signatures in the list, and MWS signatures that are likely to be produced by full-kernel malware [9]. The results of TCP fingerprinting are shown in Figs. 2 and 3. More than a half of the normal TTL packets (P_NOR) are known signatures. On the other hand, about 70% of abnormal TTL packets (P_ABN) are composed of unknown and MWS signatures. These results show that abnormal TTL packets (P_ABN) are mostly sent by hosts with full-kernel malware.

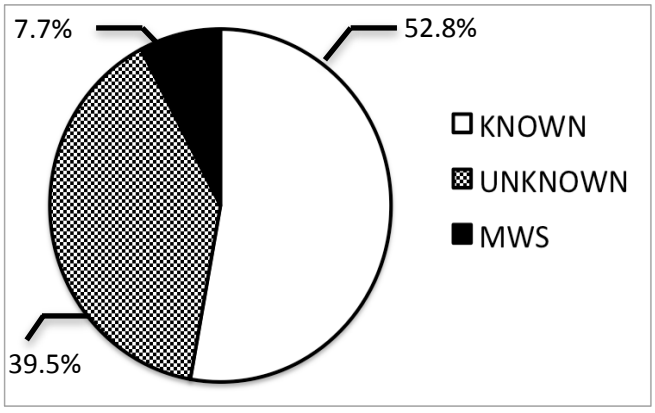

Figure 2. TCP fingerprinting for P_NOR, normal TTL packets.

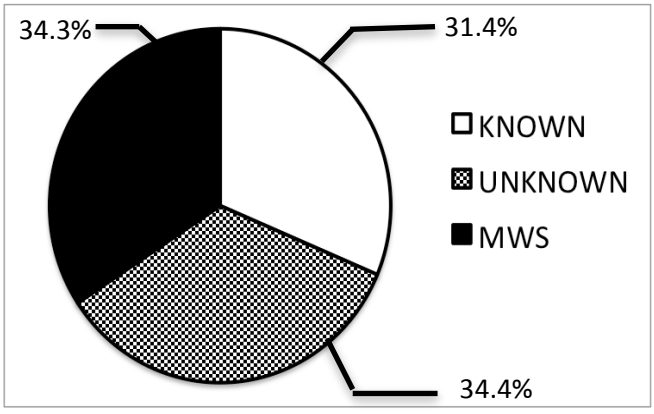

Figure 3. TCP fingerprinting for P_ABN, abnormal TTL packets.

4.3. Experiment 3: IDS snort and alerts

We applied Snort, which is a famous IDS [12], to our dataset and compared the number of alerts issued by Snort for P_NOR and P_ABN. Table 3 lists the number of alerts issued by Snort. P_ABN caused 50 times more alerts than P_NOR. The two datasets were almost equal in size. The results imply that P_ABN included a large number of malicious packets.

Table 3. Number of alerts issued by Snort.

	Number of Packets	Number of Alerts
Normal TTL	67,849,218	34,440
Abnormal TTL	69,169,306	1,658,923

5. Conclusion and Future work

This paper proposes a new method for detecting malicious packets, which is based on the fact that most IP packets have TTL values between the initial value (t_0) and a value that has been reduced by 30 increments ($t_0 - 30$). They are produced by popular OSs and classified as *normal*. If a packet has a TTL value less than $t_0 - 30$, it is classified as an *abnormal* TTL. Our proposed method is realized by filtering abnormal TTL packets.

We verified our new method using three existing methods that have been used to detect malicious packets. The results show that a dataset of abnormal TTL packets (P_ABN) includes a significantly higher rate of malicious packets than that of normal TTL packets (P_NOR). Therefore, we conclude that it is possible to discriminate a malicious packet by observing only the value of TTL.

It is emphasized that the proposed new method does not need any complex process of machine learning, a huge amount of signature files with attacking patterns, or a deep packet inspection. It is a simple tool suitable for real-time packet filtering and works well even during heavy traffic. Moreover, our method does not require updating database for discriminating malicious packets.

In this paper, the threshold between a normal TTL and abnormal TTL is a hop count of 30. Although it is fixed, based on the earlier observations of working network traffic, it should be considered more carefully. For further precise discrimination, more statistical investigation is required.

It would be also interesting to conduct more comparative studies with existing methods. This paper covers only three experiments. We are also investigating a method in which we can send an ICMP echo request packet to the source IP address of an abnormal TTL packet. Then, the ICMP reply packet indicates a TTL value. We can then compare the estimated TTL value and

actual hop count indicated by ICMP.

References

1. SECURELIST, “Spamreport:March2012,” May 2012. http://www.securelist.com/en/analysis/204792226/Spam_report_March_2012.
2. McAfee Labs, “McAfee Threats Report: Fourth Quarter 2011,” 2012.
3. S. Seufert and D. O’Brien, “Machine learning for automaticat defence against distributed denial of service attacks,” Proceedings of the IEEE International Conference on Communications 2007 (ICC ’07), pp.1217–1222, Glasgow, UK, June 2007.
4. K. Hwang, Y. Chen, and H. Liu, “Defending distributed systems against malicious intrusions and network anomalies,” Proceedings of the 19th IEEE International Parallel and Distributed Processing Symposium (IPDPS ’05), Denver, CO, America, April 2005.
5. A. Sebastian, “Default time to live (TTL) values,” Dec 2009. http://www.binbert.com/blog/2009/12/default-time-to-live-ttl-values/.
6. B. Cheswick, H. Burch, and S. Branigan, “Mapping and visualizing the internet,” Proceedings of USENIX Annual Technical Conference 2000, pp.1–12, San Diego, CA, America, June 2000.
7. K. Claffy, T. E. Monk, and D. McRobb, “Internet tomography,” Nature Web Matters, Jan 1999. http://www.nature.com/ nature/webmatters/tomog/tomog.html.
8. Quantcast, “Windows Versions Share, Quantcast Corporation,” February 2010. http://www.quantcast.com/inside-quantcast/quantcast/2010/02/windows-versions-share.html.
9. K. Kisamori, A. Shimoda, T. Mori, and S. Goto, “Analysis of malicious traffic based on TCP fingerprinting,” IPSJ Journal, vol.52, no.6, pp.2009–2018, June 2011.
10. M. Hatada, “Dataset for anti-malware research and research achievements shared at the workshop,” pp.1–8, Oct 2009.
11. M. Zalewski, “the new p0f,” Sep 2006. http://lcamtuf.coredump.cx/p0f3/.
12. Sourcefire, “Snort.” http://www.snort.org/.
13. R. Yamada, K. Tobe, S. Goto, “Discrimination malicious packets using TTL in the IP header,” IEICE-IN2011-176, vol.111, no.469, pp.235–240, Mar 2012.
14. Microsoft Malware Protection Center, “New worm targeting weak passwords on Remote Desktop connections (port 3389),” Aug 2011. http://blogs.technet.com/b/mmpc/archive/2011/08/28/new-worm-targeting-weak-passwords-on-remote-desktop-connections-port-3389.aspx.