SWINBURNE UNIVERSITY OF TECHNOLOGY

Covert Channels in the IP Time To Live TTL Field

Sebastian Zander, Grenville Armitage, Philip Branch

{szander,garmitage,pbranch}@swin.edu.au

http://caia.swin.edu.au

ATNAC 2006

Outline

- What are covert channels?
- What is the IP Time to Live (TTL) field?
- Covert channel encoding in IP TTLs
- ‘Natural’ TTL variation in Internet
- Countermeasures: detection and elimination
- Conclusions and future work

Covert Channels Motivation

- Encryption protects communication **only from being read** by third parties
- Covert channels **aim to hide the existence** of communication (information hiding)
- Often covert channels use means of communication not intended for communication (stealth over capacity)
- Introduced as mechanism to leak information between different processes on one computer
- Huge amount of network traffic makes Internet ideal for ‘high-capacity’ covert communication

Covert Channels Applications

- Government agencies, criminals, terrorists etc. hiding communications
- Hackers ex-filtrating data or controlling systems
- Users circumventing censorship, encryption laws
- Spreading of computer viruses, worms
- Attacking anonymisation techniques
- Authentication (‘port knocking’)

The Prisoner Problem

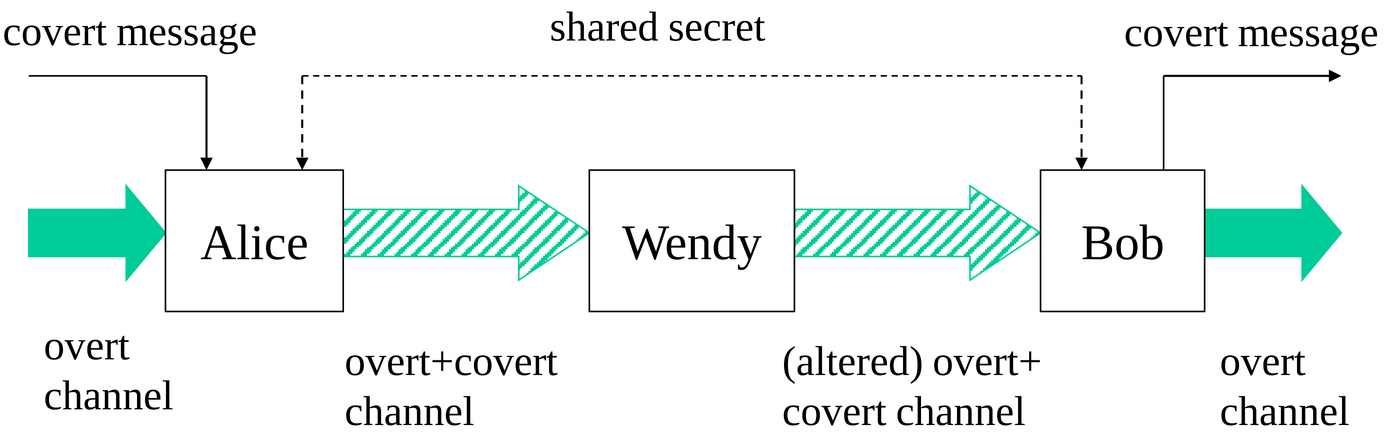
Wendy

Alice

Escape Plan

Bob

The Prisoner Problem cont'd



covert message

shared secret

covert message

Alice

Wendy

Bob

overt channel

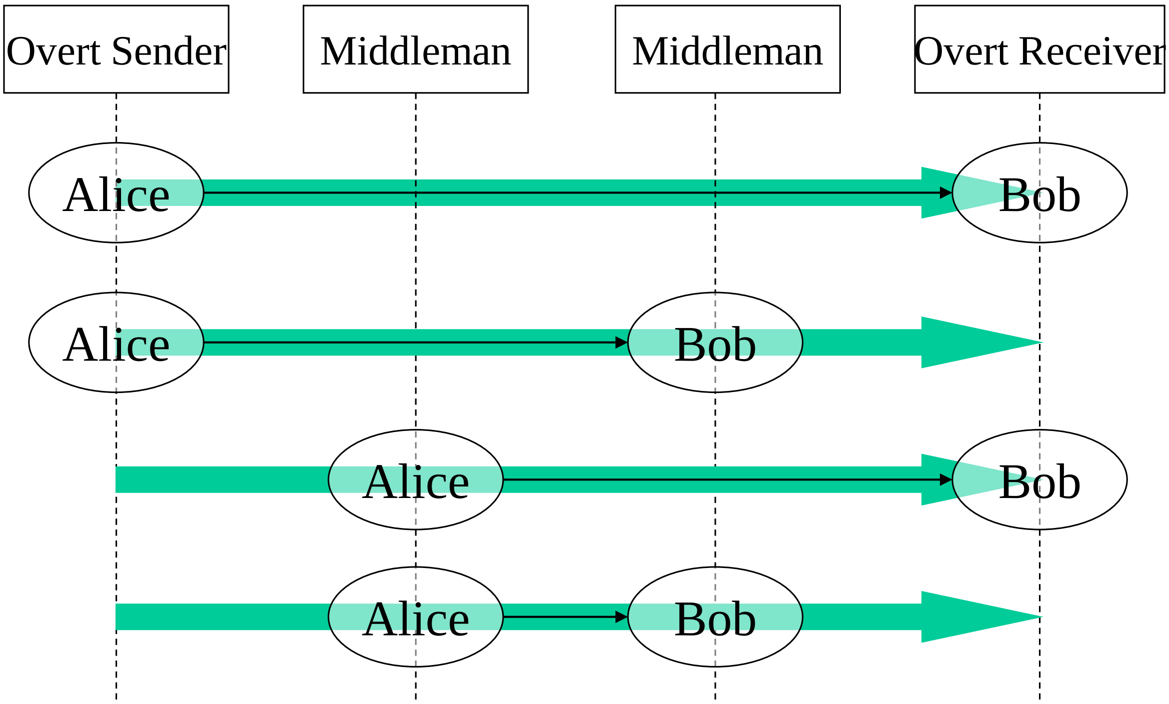
overt+covert channel

(altered) overt+ covert channel

overt channel

Alice sends covert information to Bob.
Wendy can be passive, active, malicious.

Communication Scenarios



IP Covert Channels

- Type of Service field [Handel96]
- Don’t Fragment flag [Kundur03]
- IP Identification field [Rowland97], [Ahsan02], [Cauich05]
- Fragment Offset field [Cauich05]
- Time to Live [Qu04]
- Modulate source/destination address and packet length fields [Girling87]

IP Time To Live Field

- TTL limits lifetime of IP packet in network
- Sender sets initial TTL value
- Each network element decrements TTL value
- Packet with TTL=0 is discarded

<table>
<tr><th>0</th><th></th><th>8</th><th>16</th><th>24</th><th>31</th></tr>
<tr><td>Ver</td><td>HLen</td><td>Type of Service</td><td colspan="3">Total Length</td></tr>
<tr><td colspan="3">Identification</td><td>Flags</td><td colspan="2">Fragment Offset</td></tr>
<tr><td colspan="2">TTL</td><td>Protocol</td><td colspan="3">Checksum</td></tr>
<tr><td colspan="6">Source Address</td></tr>
<tr><td colspan="6">Destination Address</td></tr>
</table>

TTL Covert Channel

- Naïve approach: Encode covert data directly in TTLs
  - □ Initial TTL values? Routing loops?
  - □ Bob needs to know (or guess) path length
  - □ **Abnormal** TTL distribution looks very suspicious to Wendy
- Real-world constraints
  - □ Initial TTL values: **64**, **128**, **255** (Windows, Linux, FreeBSD)
  - □ Path length in Internet typically **less than 32 hops**
  - □ If TTL changes in flows mostly only **2 distinct TTL** values **differing by 1** (our empirical findings)

TTL Covert Channel cont’d

- ■ Encode 1-bit as ‘high TTL’ (TTL of overt traffic)
- ■ Encode 0-bit as ‘low TTL’ (high TTL minus 1)
  - □ Bob **does not** need to know path length
  - □ Bob needs to see both zeros and ones before decoding
- ■ No negative side-effects on IP protocol
  - □ No TTL increase ⇒ no risk of looping packets
  - □ Very small decrease ⇒ given typical initial TTL and Internet path length risk of TTL=0 drops negligible
- ■ Encrypt covert information before sending

TTL Covert Channel cont’d

- ■ Channel capacity is 1 bit per packet (if no noise)
- ■ TTL channel **is noisy**
  - □ ‘Natural’ TTL changes ⇒ only in few flows (our empirical findings); Alice and Bob can probe channel before sending
  - □ TCP takes care of packet reordering/loss, but UDP does not ⇒ retransmission and/or error correction required
  - □ More elaborate channel model and error handling is work in progress

‘Natural’ TTL Variation

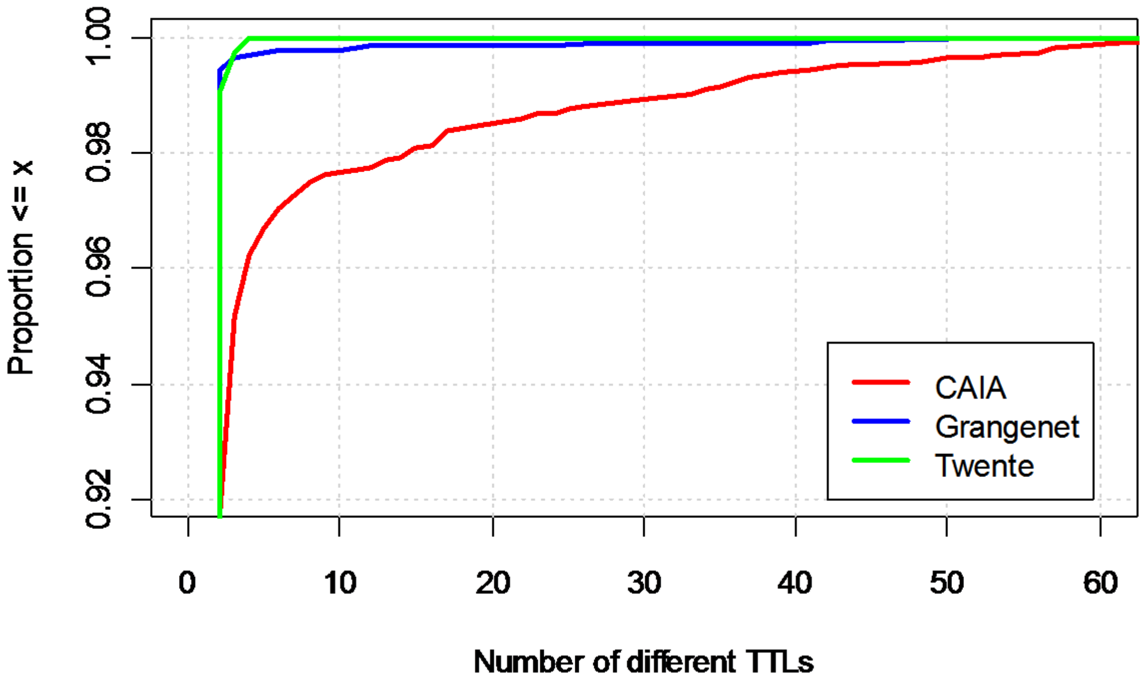
- Characteristics of ‘natural’ TTL variation occurring in Internet caused by effects such as path changes?
- Datasets
  - □ Public game/web servers (CAIA, Grangenet)
  - □ 1Gbit/s aggregated ADSL uplink (Twente)
- Group packet into unidirectional flows according to source/destination IP addresses and ports
- Only consider flows with ≥4 packets and ≥1 packet per second on average

‘Natural’ TTL Variation

- Flow has TTL change if at least two different TTLs
- Number of flows and volume in GB with and without TTL changes

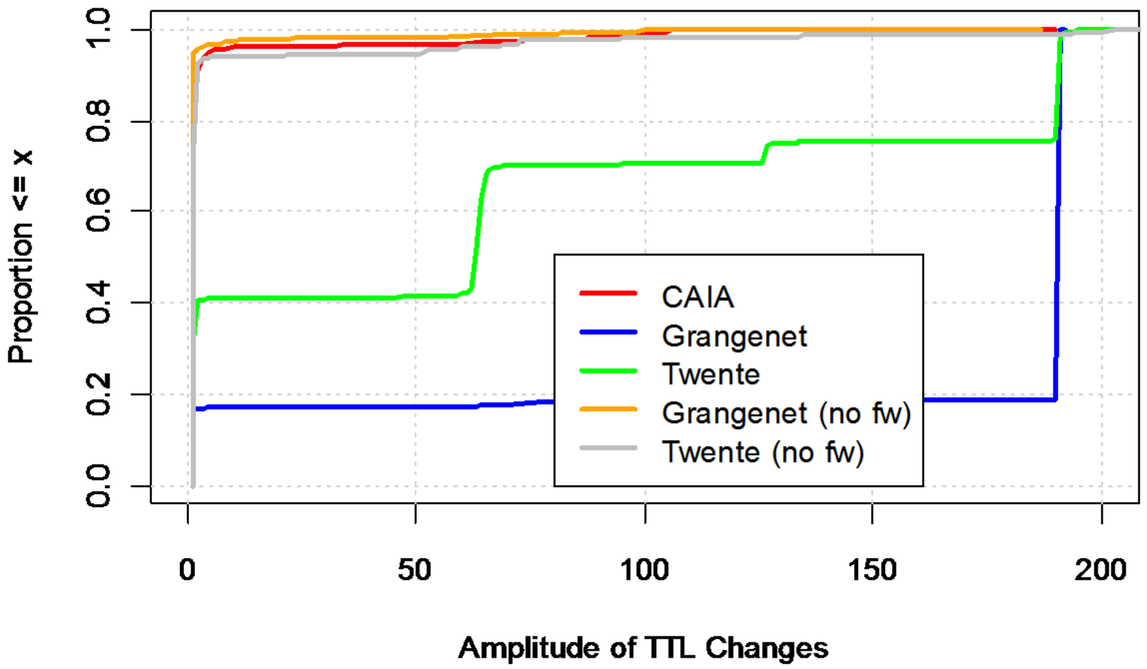
Dataset	Flows w/o TTL change	Flows with TTL change	Volume w/o TTL change	Volume with TTL change
CAIA	128,617	2766 (**2.1**%)	114.5 GB	6.0 GB (**5.0**%)
Grangenet	282,898	8582 (**2.9**%)	28.1 GB	0.9 GB (**3.1**%)
Twente	1,354,585	24,603 (**1.8**%)	62.0 GB	1.8 GB (**2.8**%)

‘Natural’ TTL Variation - Levels



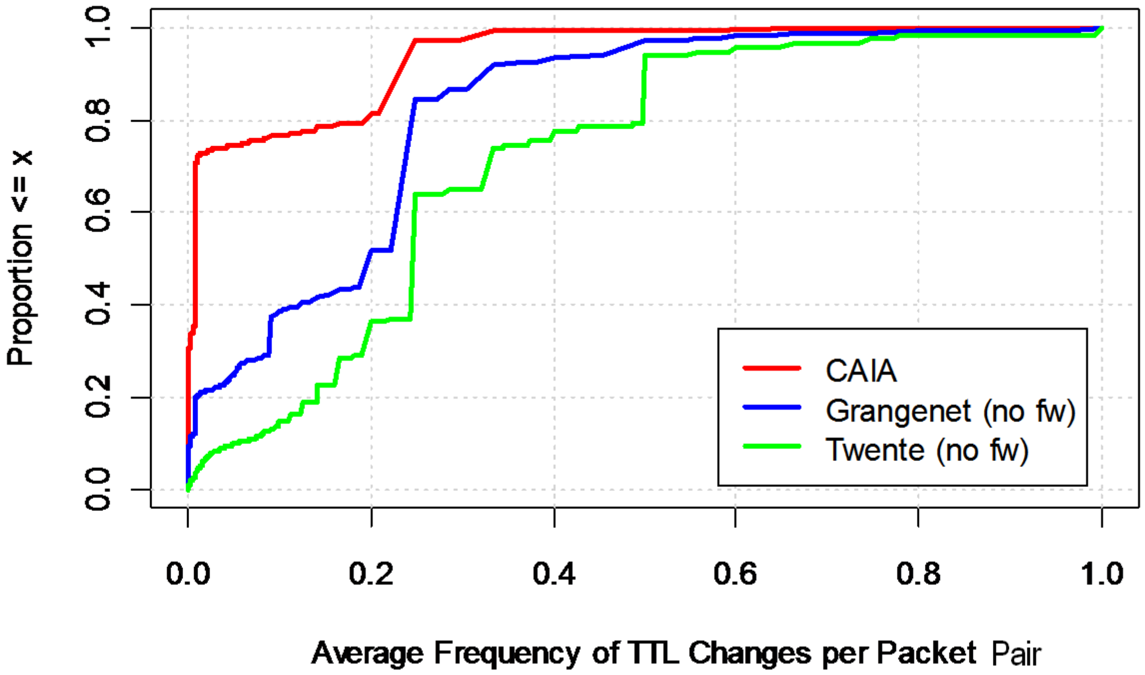
Number of distinct TTL values per flow

‘Natural’ TTL Variation - Amplitude



amplitude = $TTL_{max} - TTL_{min}$

‘Natural’ TTL Variation



frequency = # TTL changes / (packets – 1)

Countermeasures

- ■ Elimination (active warden)
  - □ Wendy sets all TTLs of packet flow to same value
  - □ If Wendy can intercept only subset of packets elimination is impossible but additional ‘noise’ reduces capacity
- ■ Detection (passive warden)
  - □ TTL covert channel looks similar to ‘natural’ TTL variation (amplitude, number of TTLs)
  - □ High change frequency uncommon but Alice can slow down
  - □ More detailed analysis reveals channel but computational effort could be high for large traffic volume

Conclusions

- Analysis of ‘natural’ TTL variation in Internet Flows: TTL changes only for small percentage of flows, but too common to be suspicious
- Proposed covert channel in IP TTL field that looks similar to ‘natural’ TTL variation
- IPv6 compatible (Hop Limit)
- Capacity depends on overt channel
- Capacity likely up to few 100bit/s for flows with hundreds packets/s; use of multiple flows possible

Future Work

- Extend TTL analysis towards more traces and more in-depth study of TTL change patterns
- Determine channel capacity in presence of noise (packet loss/reordering, ‘natural’ TTL variation)
- Improve channel encoding and error handling
- Implementation
- Evaluate efficiency of detection methods

The End

Questions?

Countermeasures

- General measures
  - □ Eliminate use of the covert channel
  - □ Limit capacity of the covert channel
  - □ Audit covert channel
  - □ Document covert channel
- Elimination/detection of TTL covert channel is harder than for most previously proposed covert channels in IP header fields but possible