Wireshark and Lua

June 16, 2010

Gordon "Fyodor" Lyon & Gerald Combs

Nmap & Wireshark

SHARKFEST **'10**

Stanford University

June 14-17, 2010

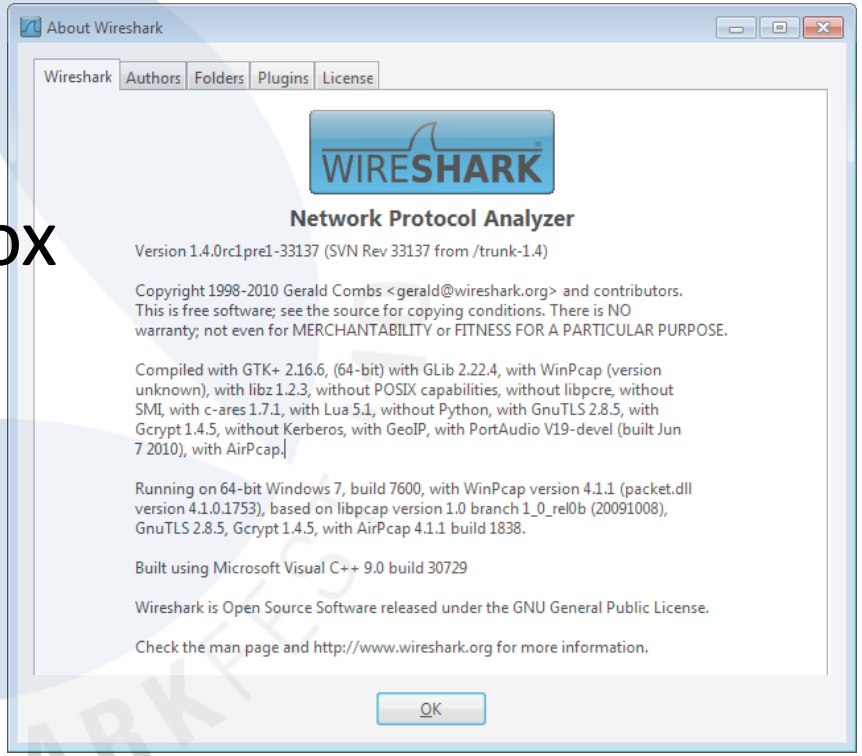
Lua?

- Lightweight scripting language
- Compares well to other languages
- Not C
- Easy prototyping, sharing, and deployment

Lua in Wireshark

- Disabled by default
- 1.0 and later
- Check your "About" box

  …or "-v"



Functionality

- Process packets
- Write files
- Run scripts
- Extend Wireshark
- Interact with the user (sort of)

Script Types

- Dissector
  - Called multiple times
  - Output to packet list and tree
- Postdissector
  - Called after frame is dissected
- Tap (listener)
  - Called via filter
  - Own output (window or text)

API Highlights

- *Field* Designate a field for extraction
- *FieldInfo* Instances returned from *Field*
- *Proto* Custom protocol definition
- *ProtoField* Custom protocol fields
- *Listener* Tap Instance
- *Dissector* Protocol dissector

Lots more in the API Reference

How Scripts Are Run

- Automatically:
  1. *init.lua* in the global configuration directory
  2. *init.lua* in the personal configuration directory*
  3. **.lua* in your plugins directory
- By hand:
  - Using "-X lua_script:*my_script.lua*"

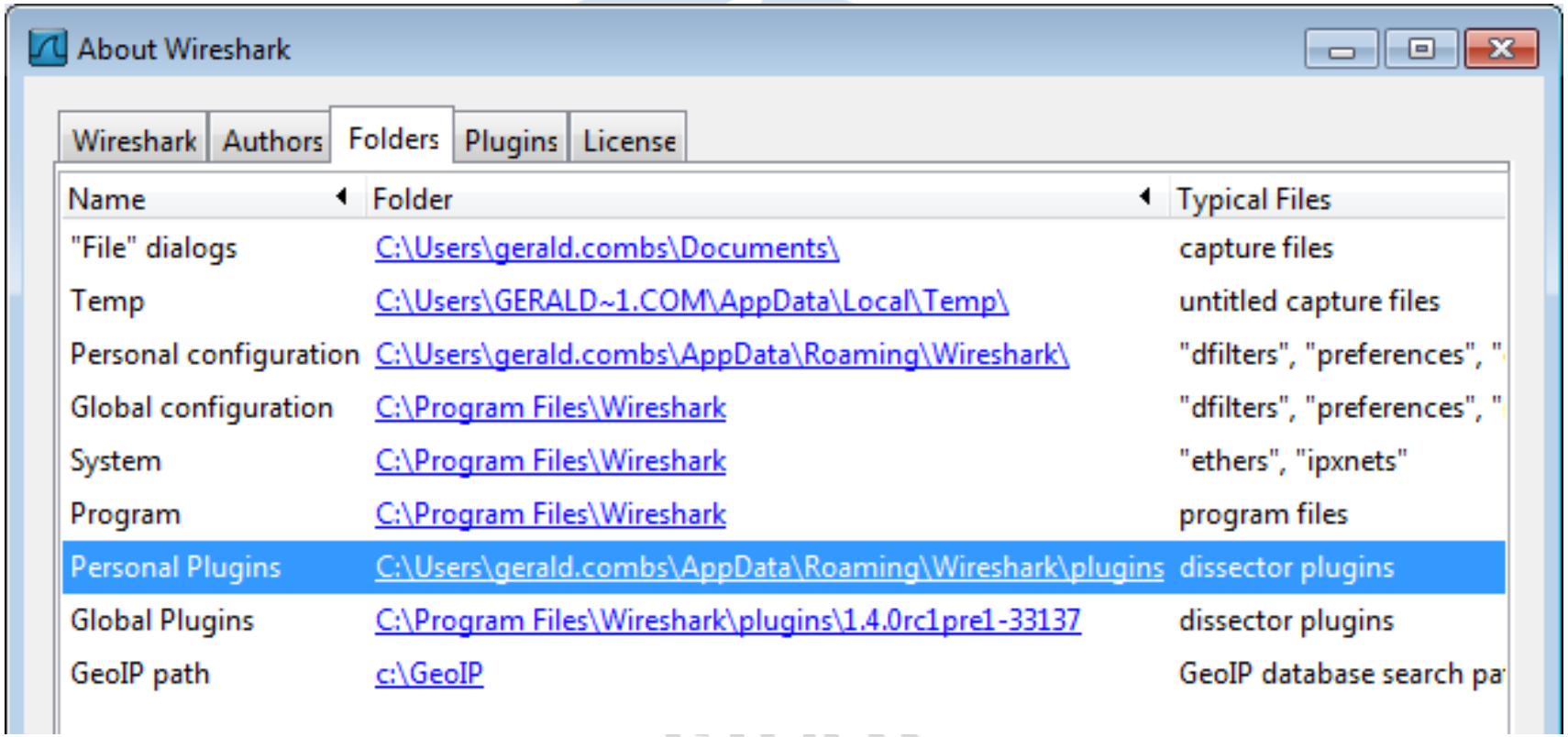
…but you have to enable it first

1. Find your personal init.lua
2. Fix it:

```
-- Change this...
disable_lua = true; do return end;
-- ...to this.
disable_lua = false; -- do return end;
```

3. Place scripts in "plugins" (or use -X)

Where are these directories?



Counting Dissections

1. Use "frame.number" as a table index
2. Create our protocol and fields

```
-- We need to extract the frame number
frame_num_f = Field.new("frame.number")

-- Declare our "protocol"
di_proto = Proto("dissection_info", "Dissection information")

-- Create the fields for our "protocol"
di_count_F = ProtoField.int32("dissection_info.count",
    "Number of times this frame has been dissected",
    base.DEC)

-- Add the field to the "protocol"
di_proto.fields = { di_count_F }

-- Track our dissection counts
dcounts = {}
```

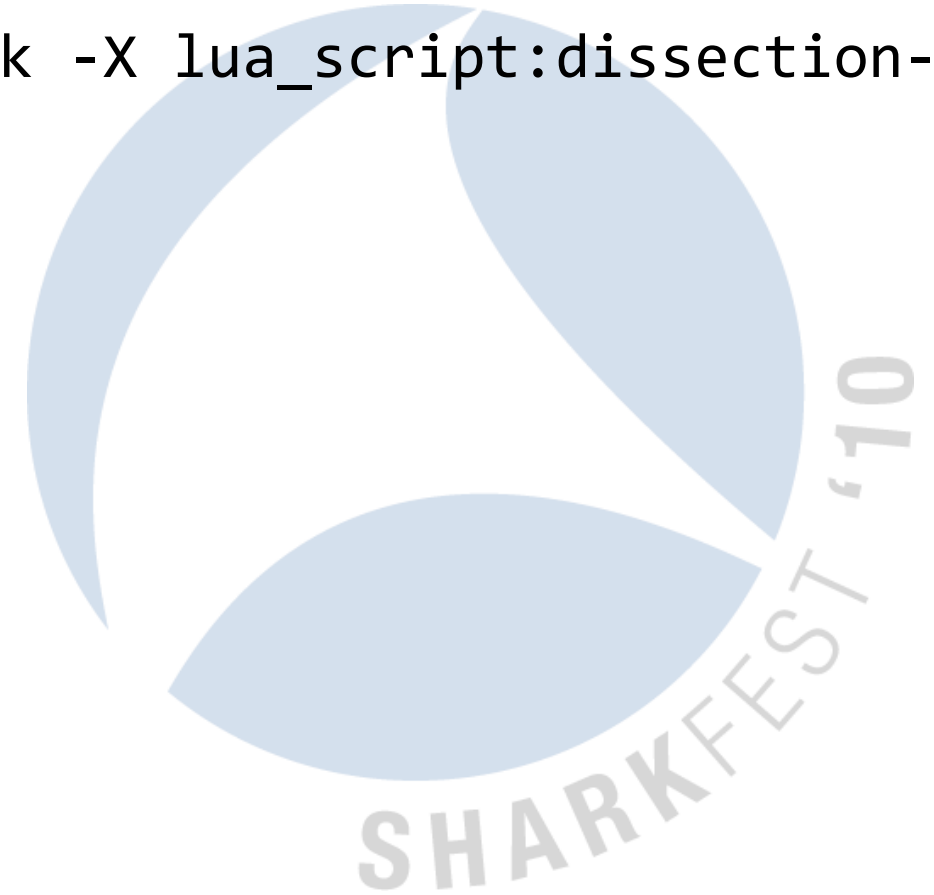
Displaying Data

3. Add the count to the tree

```
-- Create our postdissector
function di_proto.dissector(buffer, pinfo, tree)
    -- Obtain the current values the protocol fields
    local frame_num = frame_num_f()

    if (frame_num) then
        local subtree = tree:add("Dissection Count")
        local fnv = frame_num.value

        if (dcounts[fnv] == nil) then
            dcounts[fnv] = 0
        end
        dcounts[fnv] = dcounts[fnv] + 1
        subtree:add(di_count_F, dcounts[fnv])
    end
end

-- Register our protocol as a postdissector
register_postdissector(di_proto)
```

Running the Script

```
$ wireshark -X lua_script:dissection-count.lua
```



Traceroute Distance Tap

1. Find hops in a traceroute run
2. Geolocate addresses
3. Do math

How Does Traceroute Work?

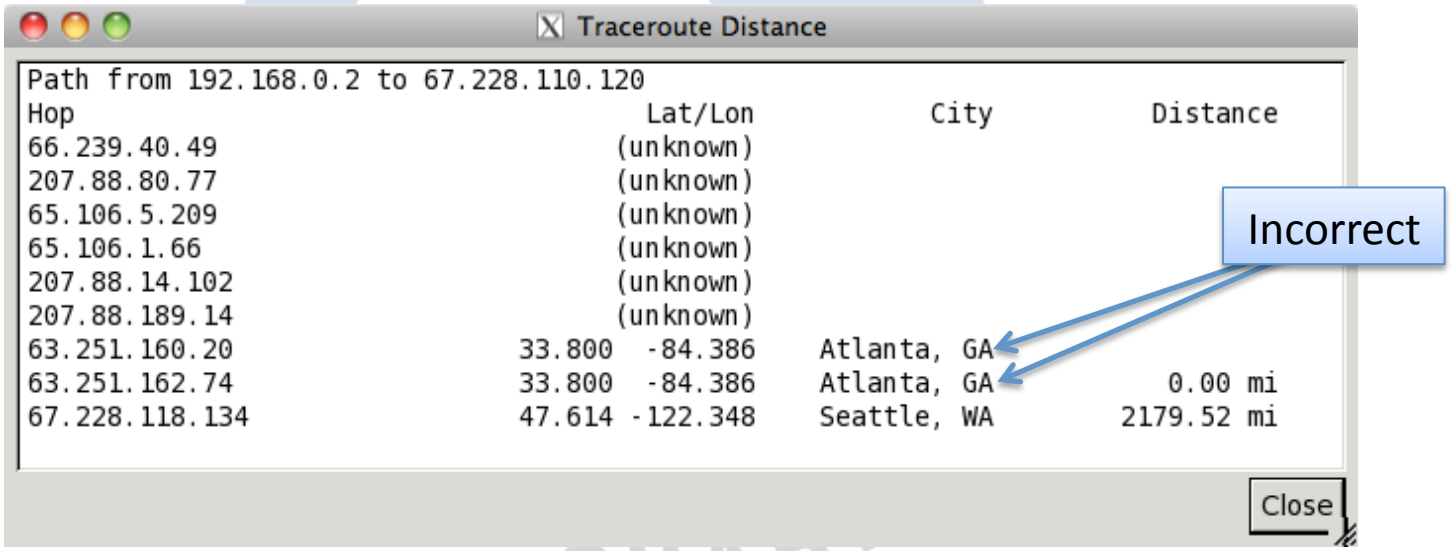
1. Send out IP datagrams with low TTLs
2. Look for ICMP TTL exceeded

Our Script

- Request tap
  - Low IP TTLs
- Response tap
  - ICMP TTL exceeded
- Matches [Source, Destination, IP ID] between requests and responses

TD Attempt #1

- Use GeoIP!
- Oops.



TD Attempt #2

- Map hostnames to coordinates

"Some people, when confronted with a problem, think *'I know, I'll use regular expressions'.* Now they have two problems." – Jamie Zawinski

Exercises for the Reader

- Add IPv6 support
- Better name → coordinate mapping
- Merge GeoIP and lookup tables
- Draw a map (Google Maps or OpenStreetMap)

Tips

- Use -X
- You can extract into a table

```
ip_src_f = Field.new("ip.src")
src_addrs = {ip_src_f()}
```

Further Information

http://www.lua.org/

http://lua-users.org/

http://www.cacetech.com/sharkfest.09/DT06_Bjorlykke_Lua%20Scripting%20in%20Wireshark.pdf

http://wiki.wireshark.org/Lua

http://www.wireshark.org/docs/wsug_html_chunked/wsluarm.html

wireshark-dev@wireshark.org