State of Suricata

15 November 2017 -- Victor Julien, OISF

SURICON
PRAGUE
2017

SURICON
PRAGUE
2017

SURICON
PRAGUE
2017

Overview

- 10 years of Suricata (code)
- Past year
- Next year

November 2007

- Matt Jonkman and Will Metcalf and myself had been talking about doing a new IDS engine
- We thought it was too hard to get it funded
- At some point I just started coding something

10 years of Suricata (code)

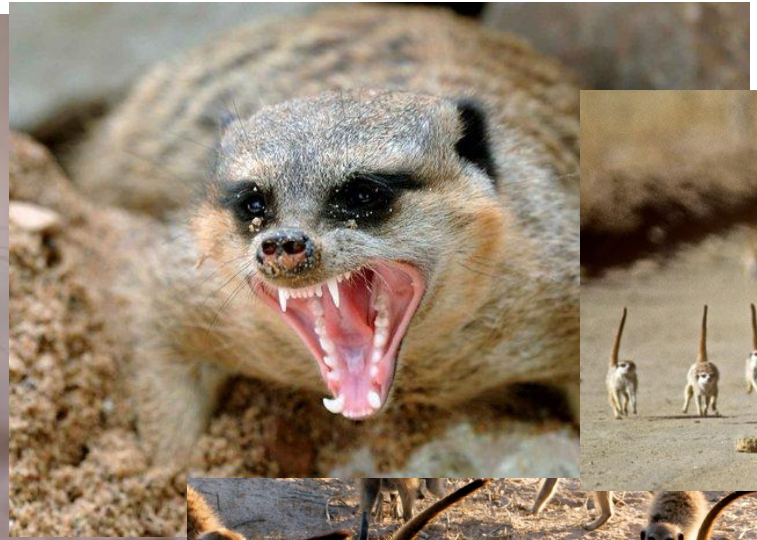
- Code started in Nov 2007
- Playground for multi-threading experimentation in C
- Netfilter NFQ based packet forwarder

What's in a name?

- VIPS: 2007- July 2009
- EIDPS: July 2009 - Dec 2009
- Suricata: Dec 2009 - now

10 years of Suricata

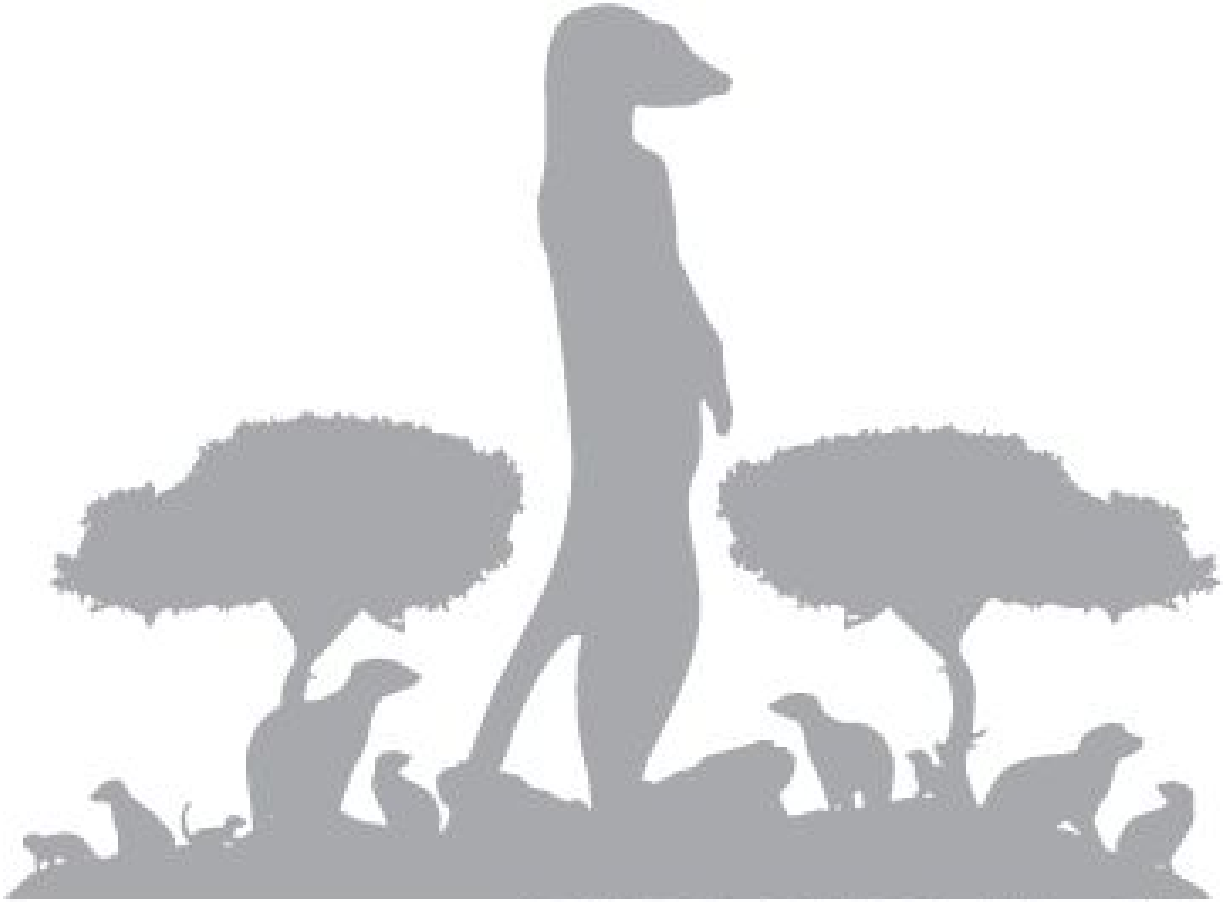
- Name “Suricata” came from the suggestion of using the Meerkat as mascot
- Latin Genus name: “Suricata”
- Late July 2009 first email conversation about “Suricata”
- December 2009 code starts using “Suricata”



SURICON
PRAGUE
2017

SURICON
PRAGUE
2017

SURICON
PRAGUE
2017



SURICATA

SURICON
PRAGUE
2017

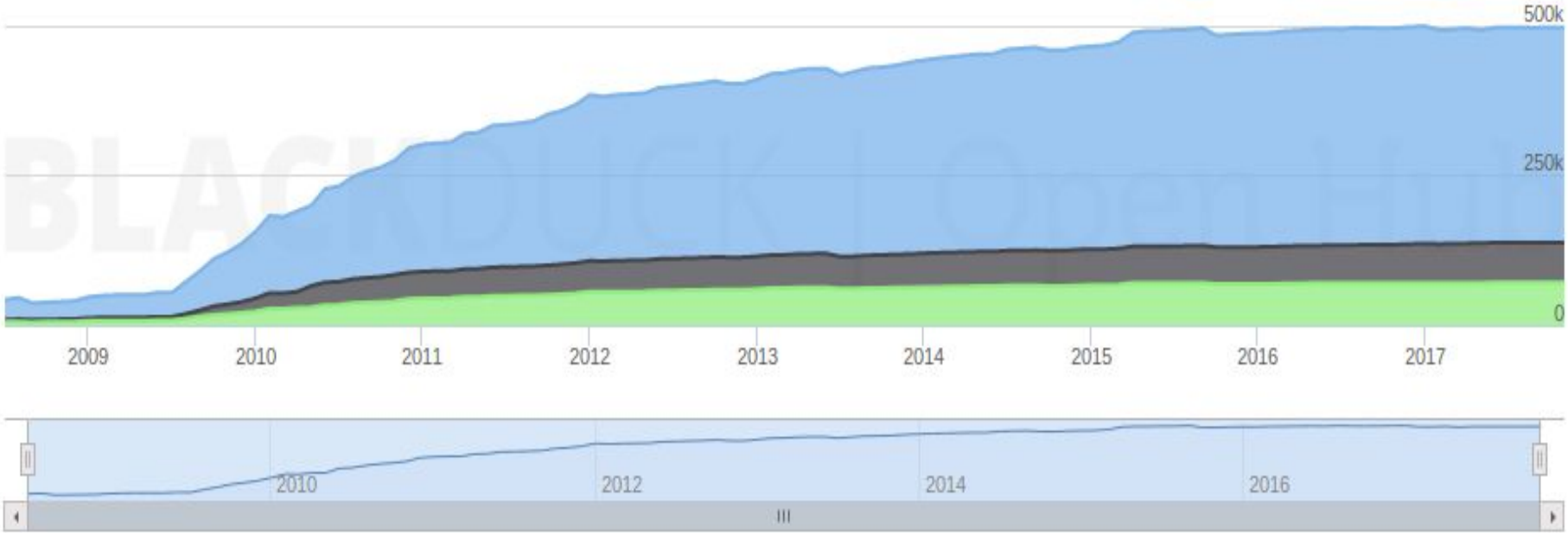
SURICON
PRAGUE
2017

SURICON
PRAGUE
2017

SURICON
PRAGUE
2017

Current Code

- ~360K LOC
- Mostly C, a bit of Rust (~1.5%)



SURICON
PRAGUE
2017

In a Nutshell, Suricata IDS/IPS...

... has had 8,293 commits made by 106 contributors
representing 358,360 lines of code

... is mostly written in C
with an average number of source code comments

... has a well established, mature codebase
maintained by a large development team
with decreasing Y-O-Y commits

... took an estimated 96 years of effort (COCOMO model)
starting with its first commit in July, 2008
ending with its most recent commit 11 days ago

Language	Code Lines	Comment Lines	Comment Ratio	Blank Lines	Total Lines	Total Percentage
C	344,284	64,307	15.7%	68,803	477,394	96.2%
Rust	5,904	865	12.8%	851	7,620	1.5%
Autoconf	2,448	13	0.5%	295	2,756	0.6%
Python	1,527	898	37.0%	462	2,887	0.6%
C++	1,484	539	26.6%	447	2,470	0.5%
Perl	857	99	10.4%	103	1,059	0.2%
Automake	815	11	1.3%	77	903	0.2%
shell script	800	97	10.8%	162	1,059	0.2%
Make	156	6	3.7%	30	192	0.0%
CUDA	58	29	33.3%	9	96	0.0%
Lua	27	1	3.6%	6	34	0.0%
Totals	358,360	66,865		71,245	496,470	

2017

or: what we did since SuriCon DC

SURICON
PRAGUE
2017

Suricata 3.2 (2016/12/01)

- TLS improvements
- Bypass functionality
- SCADA protocols
- User docs => sphinx (pdf, readthedocs)
- Performance improvements

SURICON
PRAGUE
2017

Suricata 3.2.x releases

- 3.2.1: 2017/02/15 typical point-one release, lots of fixes
- 3.2.2: 2017/06/07 fix ippair and vlan issues
- 3.2.3: 2017/07/13 DER/ASN1 fix
- 3.2.4: 2017/10/18 lots of smaller fixes, DoS fix
- EOL soon

Suricata 4.0 (2017/07/27)

- Detection capabilities extended for HTTP, TLS and more
- Further TLS improvements, incl STARTTLS
- Experimental Rust: NFS, DNS, NTP
- Extended EVE json log fields
- Rewritten TCP stream reassembly engine

Suricata 4.0.x release(s)

- 4.0.1: 2017/10/18 collection of minor fixes
- 4.0.2: ETA late November / early December

Github repo now OISF/suricata

- Was inliniac/suricata
- Looks better
- Helps team do reviews

Rust

- Based on Pierre's talk and work
- Rust is a safe & fast system programming language
- Experimental for now
- Will stay experimental in 4.1
- 1.6% of code, hope to double that by SuriCon 2018

Rust implementations

- NFSv2 & NFSv3, including logging & file extraction
- DNS
- NTP


OISF / suricata
Code
Pull requests 26
Suricata git repository maintained by
security
ids
ips
nsm
C 96.4%
Rust 1.6%

Contributors (by commit)

42	Mats Klepsland
8	Pierre Chifflier
6	Alexander Gozman
5	David Wharton
5	Sascha Steinbiss
4	Phil Young
4	fooinha
2	Jon Zeolla
2	Ray Ruvinskiy
2	jason taylor
1	Abbed
1	Derek
1	Giuseppe Longo
1	Julian
1	Peter Sanders
1	Sebastian Garcia
1	Selivanov Pavel
1	Travis Green
1	psanders240
1	qiangbei

We're hiring!

- Want to work on open source code in your pyjama's?
- Some travel
- Skills required:
  - C and/or Rust
  - Python helps
  - Community oriented
  - Able to communicate with a virtual/distributed team

EOL policy

- We’ve had an implicit EOL policy
- Now formalized
- We will send EOL announcements
- 4.0 is ‘stable’
- 3.2 is ‘oldstable’, EOL soon
- https://suricata-ids.org/about/eol-policy/
- Still no LTS, as no one stepped up to fund it (but you could!)

Support Status (in progress)

- Ongoing effort to document what parts of the engine are supported by OISF, which by community
- Currently ‘tier 1’, ‘tier 2’, community
- Tier 1: core features on most important platforms
  - E.g. AF_PACKET on Linux
- Tier 2: lesser used features and/or OS’
  - E.g. OpenBSD support

Support Status (in progress)

- Community: contributed things that have limited use cases or are just very new
- https://redmine.openinfosecfoundation.org/projects/suricata/wiki/Support_Status

Upcoming

SURICON
PRAGUE
2017

SURICON
PRAGUE
2017

suricata-update

- rule / intel updater designed especially for Suricata
- Meant to replace oinkmaster / pulledpork for Suricata users
- GPLv2 licensed, developed in Python
- Command line tool
- Maintained by Jason Ish
- https://github.com/OISF/suricata-update

4.1dev is about to open

- Lots of Pull Requests and branches waiting
- Lots more coming

SMB/SMB2/SMB3

- Implementation in Rust
- Reimplementation of current SMB1 and DCERPC parsing
- Add SMB2 and SMB3 support

SMB

- Funded by FoxIT to develop logging for SMB2+
- Adding SMB1 as well
- Detection
- File extraction
- Logging

```
{
   "id": 2,
   "dialect": "NT LM 0.12",
   "status": "STATUS_SUCCESS",
   "domain": "",
   "user": "",
   "host": "DESKTOP-V1FA0UQ",
   "tree_id": 2048,
   "named_pipe": "\\SCV\\IPC$",
   "type": "response"
}
```

```
{
  "id": 1,
  "dialect": "2.FF",
  "status": "STATUS_SUCCESS",
  "domain": "",
  "user": "",
  "host": "",
  "client_dialects": [
    "PC NETWORK PROGRAM 1.0",
    "LANMAN1.0",
    "Windows for Workgroups 3.1a",
    "LM1.2X002",
    "LANMAN2.1",
    "NT LM 0.12",
    "SMB 2.002",
    "SMB 2.???"
  ],
  "type": "response"
}
```

```
{
  "id": 3,
  "dialect": "3.11",
  "status": "STATUS_SUCCESS",
  "domain": "DESKTOP-2AEFM7G",
  "user": "Willi Wireshark",
  "host": "DESKTOP-2AEFM7G",
  "tree_id": 1,
  "named_pipe": "\\\\192.168.199.133\\IPC$",
  "type": "response"
}
```

```
{
  "id": 4,
  "dialect": "3.11",
  "status": "STATUS_SUCCESS",
  "domain": "DESKTOP-2AEFM7G",
  "user": "Willi Wireshark",
  "host": "DESKTOP-2AEFM7G",
  "dcerpc": {
    "request": "BIND",
    "response": "BINDACK",
    "interfaces": [
      {
        "uuid": "4b324fc8-1670-01d3-1278-5a47bf6ee188",
        "version": "3.0",
        "ack_result": 2,
        "ack_reason": 0
      },
      {
        "uuid": "4b324fc8-1670-01d3-1278-5a47bf6ee188",
        "version": "3.0",
        "ack_result": 0,
        "ack_reason": 0
      },
      {
        "uuid": "4b324fc8-1670-01d3-1278-5a47bf6ee188",
        "version": "3.0",
        "ack_result": 3,
        "ack_reason": 0
      }
    ]
  },
  "type": "response"
}
```

Detection

- “alert smb …” -> match smb1/2/3
- “alert smb … (file_data; content:”MZ”; depth:2; …)”
- “alert smb … (smb_named_pipe; content:"|5C 5C|192.168.199.133|5C|IPC|24|"; …)”
- “alert smb … (smb_share; content:"|5C 5C|ts412|5C|pcap"; …)”
- “alert smb … (filemagic:”PDF”; …)”

Transformation keywords

E.g.

file_data; compress_whitespace; content:”window.location=”;

Takes a ‘stickybuffer’ and transforms it.

Multiple transforms can be chained.

Transform: strip_whitespace

Remove whitespace from buffer.

E.g.

“A B C” => “ABC”

Transform: compress_whitespace

Normalize whitespace in buffer.

E.g.

“A B C” => “A B C”

Transform: to_sha256

Replace buffer with sha256 hash of the buffer content.

E.g.

“www.baddomain.com” =>
“1b9d9527933923ca7499c100a97715d142911a8267d41b1a649fee8905a46495”

Transformations

Planned:

- strip_nulls
- Others needed?
- Bring your ideas / whishes to the brainstorm session

Optimizations

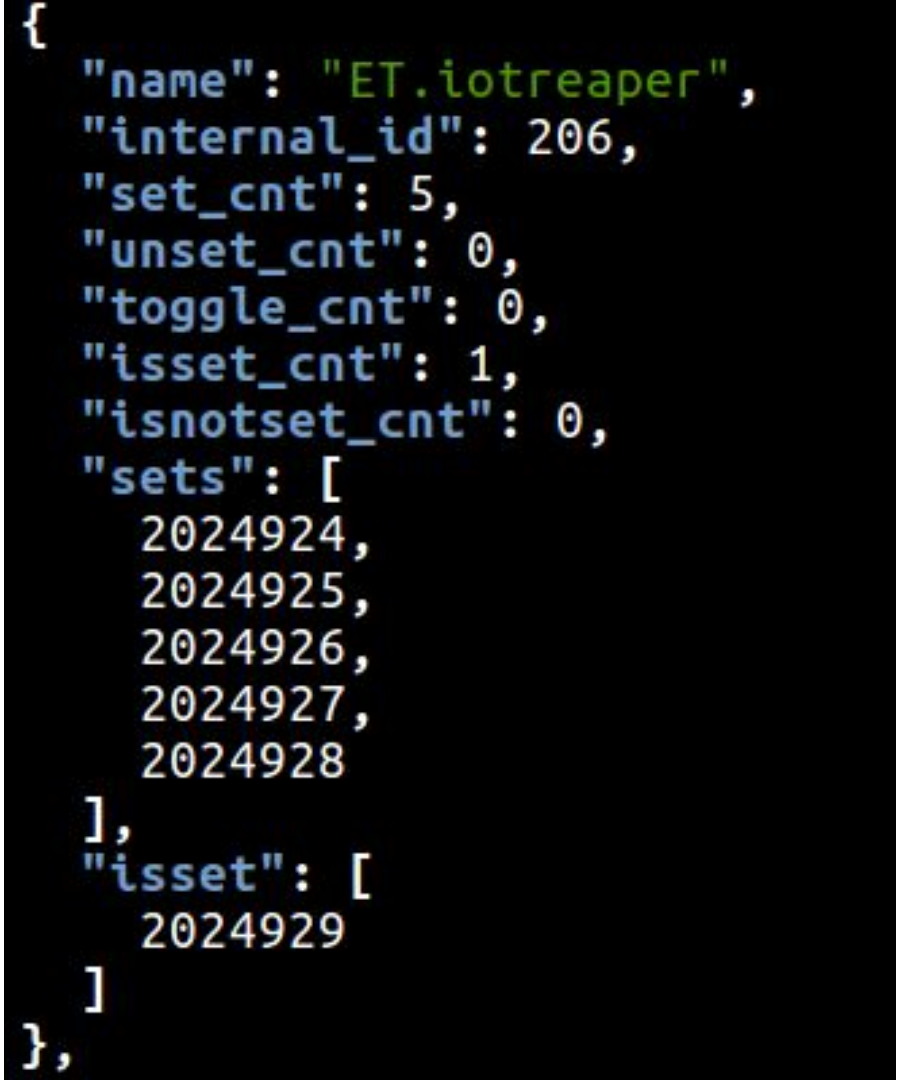
- Detection engine rewrite
- Reduce complexity
- Clean up
- Support transforms
- Making extending easier
- Solve some corner cases
- Improve performance
- Output more useful info about rules

```
{
  "id": 2009218,
  "gid": 1,
  "rev": 7,
  "msg": "ET SCAN Tomcat admin-blank login credentials",
  "app_proto": "http",
  "engines": [
    {
      "name": "http_header",
      "direction": "toserver",
      "is_mpm": true,
      "matches": [
        {
          "name": "content",
          "content": {
            "pattern": "\r\nAuthorization: Basic YWRtaW46\r\n",
            "nocase": false,
            "negated": false,
            "ends_with": false
          }
        }
      ]
    },
    {
      "name": "http_uri",
      "direction": "toserver",
      "is_mpm": false,
      "matches": [
        {
          "name": "content",
          "content": {
            "pattern": "/manager/html",
            "nocase": true,
            "negated": false,
            "ends_with": false
```

Flowbits

- Adding a flowbits analyzer
- Will warn when rule checks a bit that is never set
- Dumps flowbits to JSON for analysis

```
[13582] 13/11/2017 -- 16:26:49 - (detect-flowbits.c:465) <Warning> (DetectFlowbitsAnal
yze) -- [ERRCODE: SC_WARN_FLOWBIT(302)] - flowbit 'ETPRO.TinyNuke' is checked but not
set. Checked in 2024513 and 0 other sigs
```


{
"name": "ET.iotreaper",
"internal_id": 206,
"set_cnt": 5,
"unset_cnt": 0,
"toggle_cnt": 0,
"isset_cnt": 1,
"isnotset_cnt": 0,
"sets": [
2024924,
2024925,
2024926,
2024927,
2024928
],
"isset": [
2024929
]
},

More optimizations

- Improved the existing MPM implementations AC and its variants by taking depth/offset into account
- Added logic to automagically set depth/offset where possible
  - E.g. content:”abc”; depth:3; content:”defg”; within:4; distance:0;
- Apply ‘urilen’ to http_uri/http_raw_uri as depth
- Improves performance for both Hyperscan and built-in AC variants

XDP/eBPF

- Work by Eric Leblond with testing by Michal Purzynski
- More better flow bypass
- See their talk :)

Rate_filter “by_both”

- Rate_filter per ip-pair
- Developed by Ruslan Usmanov
- Hopefully path into full thresholding support for ip-pair

Rule Metadata logging

- Use rule ‘metadata’ field as a key/value pair list
- ET/ETpro has started using this
- Eric Leblond is working on this

PCAP improvements

- Danny Browning is adding improvements to PCAP file processing
- Commandline option to process a directory of pcaps
- Option to keep Suricata active so you can drop in new files
- Improvements to the Unix Socket mode, like adding support for directories there as well

Traffic ID ruleset

- New ruleset to be released soon
- Will be like ‘AppID’ / ‘OpenAppID’
- Classification: label flows with metadata
- Support bypass, so ‘bypass netflix traffic’
- Open source, license TBD. Likely GPL or BSD/MIT.

PF_RING Hardware Bypass

- “Flow bypass” -> skip processing of elephant flows that are uninteresting, e.g. Netflix
- Alfredo Cardigliano is adding “flow bypass” support to PF_RING
- PF_RING can offload this bypass to hardware if the hardware supports it
- Alfredo will explain it in his own talk

Suricata (Community) Council

- Idea to have regular & somewhat structured conversations with community members and contributors
- Likely schedule: talk every quarter by phone or chat, if possible in-person at SuriCon
- Open discussion and idea sharing between dev team and community
- More info soon

SURICON
PRAGUE
2017

Wrapping Up

SURICON
PRAGUE
2017

State of Suricata

- Overall state is good
- Community is healthy and growing
- OISF is doing well

Things are good, but...

- We want to grow the dev team
- We need funding for that
- So… We need your help

Join Us!

- We are looking for developers, QA help, doc writers, etc
- We have paid positions available both full time and part time
- Talk to me or any of the team at SuriCon or email me at victor@inliniac.net

SURICON
PRAGUE
2017

Wrapping up (really)

- 10 years of Suricata has been a lot of fun
- It was much harder than we imagined
- I made many friends
- Very grateful for your support
- I’m proud and all, but my mind is already on what I will be working on next week

SURICON
PRAGUE
2017

Let’s make the next decade “TREMENDOUS”