Implementation and evaluation of secure and scalable anomaly-based network intrusion detection

Bachelor Thesis Presentation

Aufgabensteller: Prof. Dr. Helmut Reiser

Betreuer: Dipl. Inf. Stefan Metzger

Datum: 12.12.2018

Philipp Mieden, 2018

Why network security monitoring?

Network environments have many vulnerable components

Hackers will always find a way in - “low hanging fruit” paradigm

Prepare for worst case - assist Incident Response and Forensics

Why network anomaly detection?

LUDWIG-MAXIMILIANS-UNIVERSITÄT MÜNCHEN

Number of signatures is exploding (Kaspersky: 330 000 unique new samples per day)

New threats cannot be detected by signatures

Existing malware can be obfuscated to be fully undetectable again

Problems?

LUDWIG-MAXIMILIANS-UNIVERSITÄT MÜNCHEN

"The monitor will be attacked."

–Vern Paxson, 1998

What about

memory safety

All existing frameworks are written in C or C++

Parsing network protocols is complex.

Mistakes happen a lot.

The network monitor could be disabled or compromised.

MITRE CVE results for Snort IDS

LUDWIG-MAXIMILIANS-UNIVERSITÄT MÜNCHEN

CVE-2016-1463	Cisco FireSIGHT System Software 5.3.0, 5.3.1, 5.4.0, 6.0, and 6.0.1 allows remote attackers to bypass Snort rules via crafted parameters in the header of an HTTP packet, aka Bug ID CSCuz20737.
CVE-2016-1417	Untrusted search path vulnerability in Snort 2.9.7.0-WIN32 allows remote attackers to execute arbitrary code and conduct DLL hijacking attacks via a Trojan horse tcapi.dll that is located in the same folder on a remote file share as a pcap file that is being processed.
CVE-2015-6427	Cisco FireSIGHT Management Center allows remote attackers to bypass the HTTP attack detection feature and avoid triggering Snort IDS rules via an SSL session that is mishandled after decryption, aka Bug ID CSCux53437.
CVE-2014-4695	Multiple open redirect vulnerabilities in the Snort package before 3.0.13 for pfSense through 2.1.4 allow remote attackers to redirect users to arbitrary web sites and conduct phishing attacks via (1) the referer parameter to snort_rules_flowbits.php or (2) the returl parameter to snort_select_alias.php.
CVE-2014-4693	Multiple cross-site scripting (XSS) vulnerabilities in the Snort package before 3.0.13 for pfSense through 2.1.4 allow remote attackers to inject arbitrary web script or HTML via (1) the eng parameter to snort_import_aliases.php or (2) unspecified variables to snort_select_alias.php.
CVE-2009-4211	The U.S. Defense Information Systems Agency (DISA) Security Readiness Review (SRR) script for the Solaris x86 platform executes files in arbitrary directories as root for filenames equal to (1) java, (2) openssl, (3) php, (4) snort, (5) tshark, (6) vncserver, or (7) wireshark, which allows local users to gain privileges via a Trojan horse program.
CVE-2009-3641	Snort before 2.8.5.1, when the -v option is enabled, allows remote attackers to cause a denial of service (application crash) via a crafted IPv6 packet that uses the (1) TCP or (2) ICMP protocol.
CVE-2007-0251	Integer underflow in the DecodeGRE function in src/decode.c in Snort 2.6.1.2 allows remote attackers to trigger dereferencing of certain memory locations via crafted GRE packets, which may cause corruption of log files or writing of sensitive information into log files.
CVE-2006-6931	Algorithmic complexity vulnerability in Snort before 2.6.1, during predicate evaluation in rule matching for certain rules, allows remote attackers to cause a denial of service (CPU consumption and detection outage) via crafted network traffic, aka a "backtracking attack."
CVE-2006-5276	Stack-based buffer overflow in the DCE/RPC preprocessor in Snort before 2.6.1.3, and 2.7 before beta 2; and Sourcefire Intrusion Sensor; allows remote attackers to execute arbitrary code via crafted SMB traffic.
CVE-2006-2769	The HTTP Inspect preprocessor (http_inspect) in Snort 2.4.0 through 2.4.4 allows remote attackers to bypass "uricontent" rules via a carriage return (\r) after the URL and before the HTTP declaration.

MITRE CVE results for Bro IDS

LUDWIG-
MAXIMILIANS-
UNIVERSITÄT
MÜNCHEN

Search Results

There are **6** CVE entries that match your search.

Name	Description
CVE-2018-17019	In Bro through 2.5.5, there is a DoS in IRC protocol names command parsing in analyzer/protocol/irc/IRC.cc.
CVE-2018-16807	In Bro through 2.5.5, there is a memory leak potentially leading to DoS in scripts/base/protocols/krb/main.bro in the Kerberos protocol parser.
CVE-2017-1000458	Bro before Bro v2.5.2 is vulnerable to an out of bounds write in the ContentLine analyzer allowing remote attackers to cause a denial of service (crash) and possibly other exploitation.
CVE-2015-1522	analyzer/protocol/dnp3/DNP3.cc in Bro before 2.3.2 does not reject certain non-zero values of a packet length, which allows remote attackers to cause a denial of service (buffer overflow or buffer over-read) via a crafted DNP3 packet.
CVE-2015-1521	analyzer/protocol/dnp3/DNP3.cc in Bro before 2.3.2 does not properly handle zero values of a packet length, which allows remote attackers to cause a denial of service (buffer overflow or buffer over-read if NDEBUG; otherwise assertion failure) via a crafted DNP3 packet.
CVE-2007-0186	Multiple cross-site scripting (XSS) vulnerabilities in F5 FirePass SSL VPN allow remote attackers to inject arbitrary web script or HTML via (1) the xcho parameter to my.logon.php3; the (2) topblue, (3) midblue, (4) wtopblue, and certain other Custom color parameters in a per action to vdesk/admincon/index.php; the (5) h321, (6) h311, (7) h312, and certain other Front Door custom text color parameters in a per action to vdesk/admincon/index.php; the (8) ua parameter in a bro action to vdesk/admincon/index.php; the (9) app_param and (10) app_name parameters to webyfiers.php; (11) double eval functions; (12) JavaScript contained in an <FP_DO_NOT_TOUCH> element; and (13) the vhost parameter to my.activation.php. NOTE: it is possible that this candidate overlaps CVE-2006-3550.

Bro issues not tracked in MITRE database

LMU LUDWIG-MAXIMILIANS-UNIVERSITÄT MÜNCHEN

Bro 2.5.5

Top

Bro 2.5.5 primarily addresses security issues.

- Fix array bounds checking in BinPAC: for arrays that are fields within a record, the bounds check was based on a pointer to the start of the record rather than the start of the array field, potentially resulting in a buffer over-read.
- Fix SMTP command string comparisons: the number of bytes compared was based on the user-supplied string length and can lead to incorrect matches. e.g. giving a command of "X" incorrectly matched "X-ANONYMOUSTLS" (and an empty commands match anything).

The following changes address potential vectors for Denial of Service reported by Christian Titze & Jan Grashöfer of Karlsruhe Institute of Technology:

- "Weird" events are now generally suppressed/sampled by default according to some tunable parameters:

  - Weird::sampling_whitelist
  - Weird::sampling_threshold
  - Weird::sampling_rate
  - Weird::sampling_duration

  Those options can be changed if one needs the previous behavior of a "net_weird", "flow_weird", or "conn_weird" event being raised for every single event. Otherwise, there is a new weird_stats.log which contains concise summaries of weird counts per type per time period and the original weird.log may not differ much either, except in the cases where a particular weird type exceeds the sampling threshold. These changes help improve performance issues resulting from excessive numbers of weird events.

!Problem?

To the rescue!

LUDWIG-MAXIMILIANS-UNIVERSITÄT MÜNCHEN

NETCAP

Traffic Analysis Framework

How does it work?

language for implementation: Golang (garbage collected runtime)

gopacket library for decoding packets

concurrent design: worker pool, each audit record written to a separate file

audit record generation as compressed protocol buffers

Why protocol buffers?

Type safe structured data - can represent complex nested structures

Goal: data accessibility -> generated data is available in almost any programming language

Huge landscape of frameworks for machine learning, in many interesting languages (R, Scala, Haskell…)

see: https://github.com/josephmisiti/awesome-machine-learning

What audit records are generated?

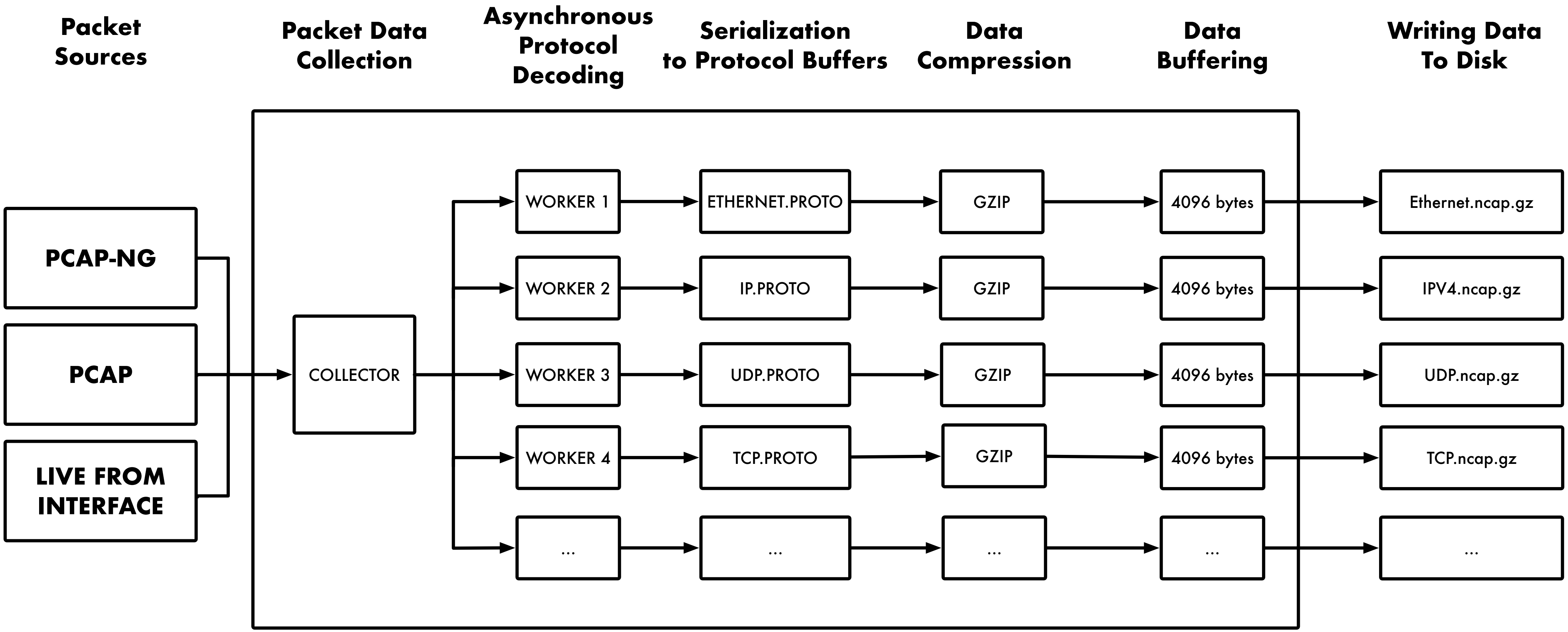
+ TLS (Client Hello Msg + Ja3)
+ LinkFlow
+ NetworkFlow
+ TransportFlow
+ HTTP
+ Flow (unidirectional)
+ Connection (bidirectional)

+ NTP
+ SIP
+ IGMP
+ LLC
+ IPv6HopByHop
+ SCTP
+ SNAP
+ LinkLayerDiscovery
+ ICMPv6NeighborAdvertisement
+ ICMPv6RouterAdvertisement
+ EthernetCTP
+ EthernetCTPReply
+ LinkLayerDiscoveryInfo

+ TCP
+ UDP
+ IPv4
+ IPv6
+ DHCPv4
+ DHCPv6
+ ICMPv4
+ ICMPv6
+ ICMPv6Echo
+ ICMPv6NeighborSolicitation
+ ICMPv6RouterSolicitation
+ DNS
+ ARP
+ Ethernet
+ Dot1Q
+ Dot11

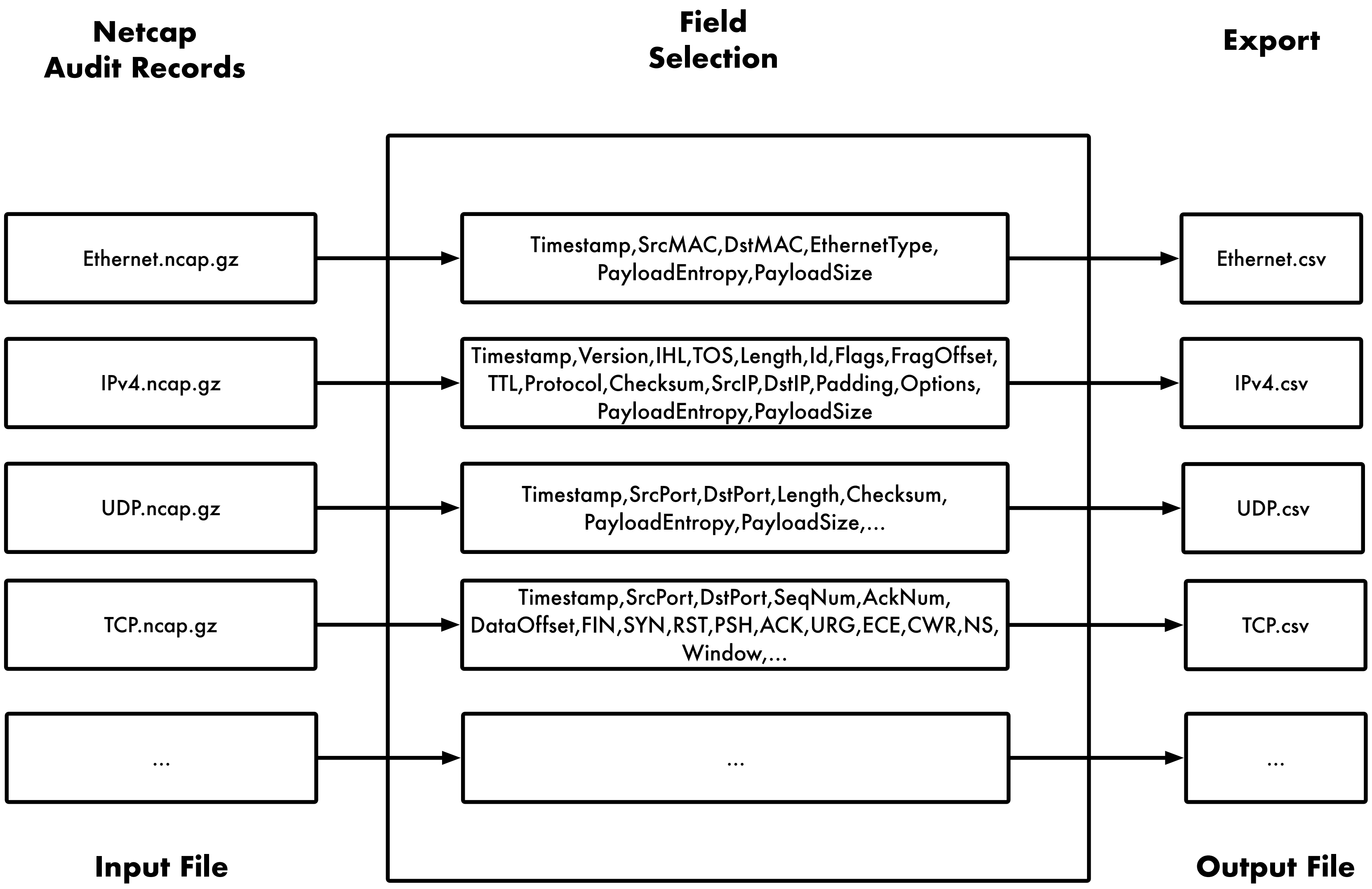
Internals

NETCAP in a nutshell



NETCAP filtering & csv export

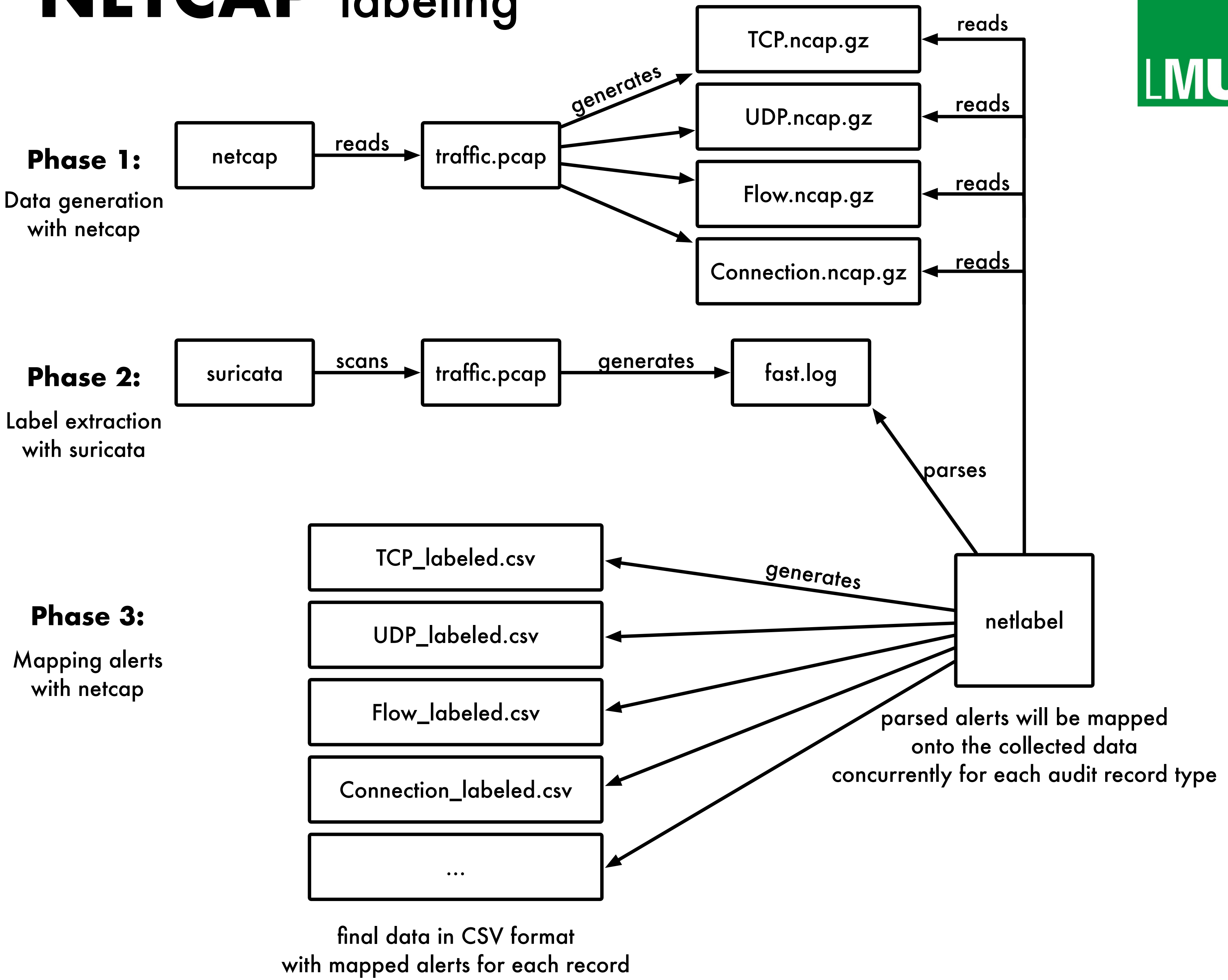
Field Selection



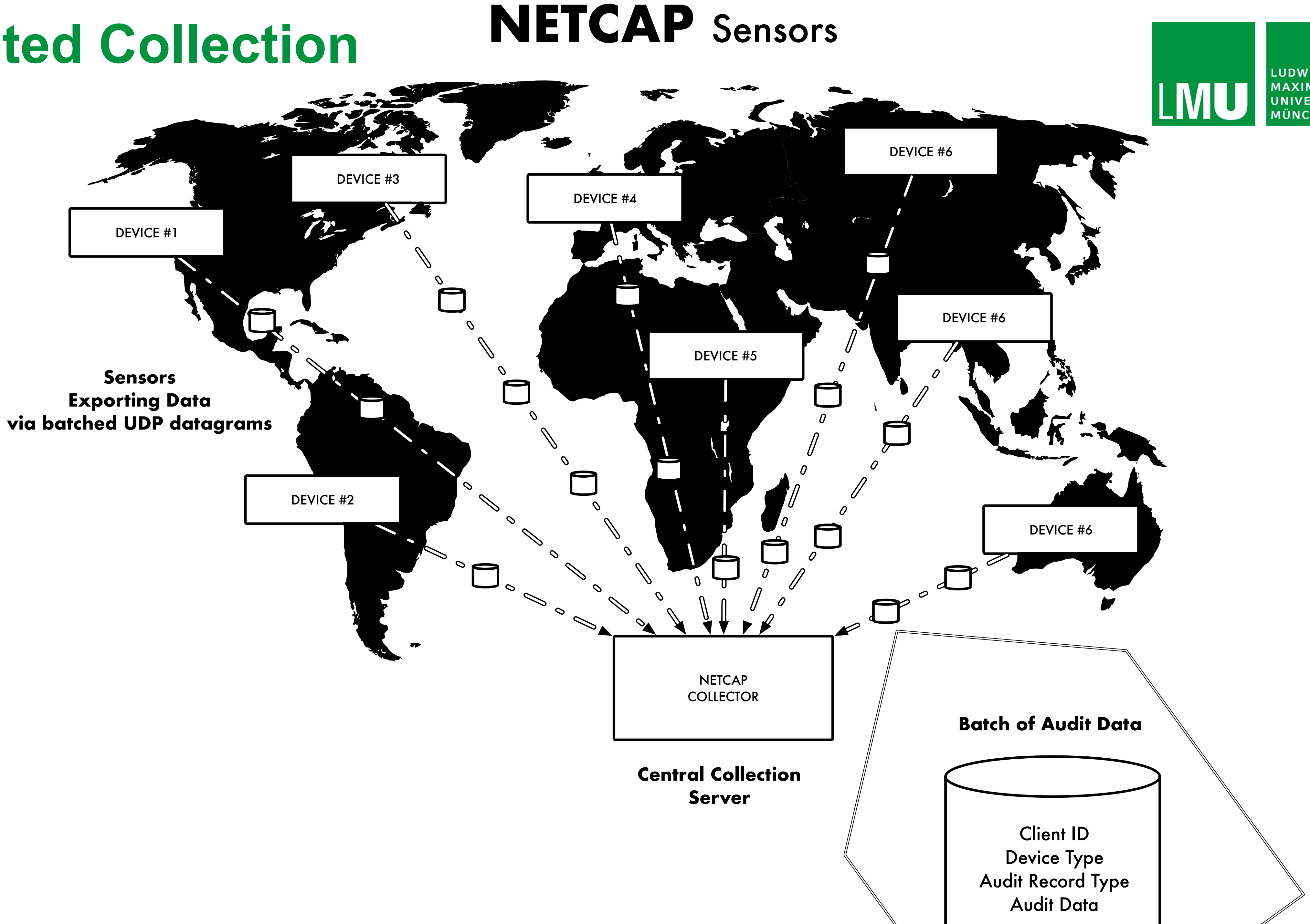

```
$ netcap -r TCP.ncap.gz    -select Timestamp,SrcPort,DstPort,SeqNum,Window,ACK,SYN,RST    > TCP.csv
```

Labeling

NETCAP labeling



Distributed Collection



Further use cases?

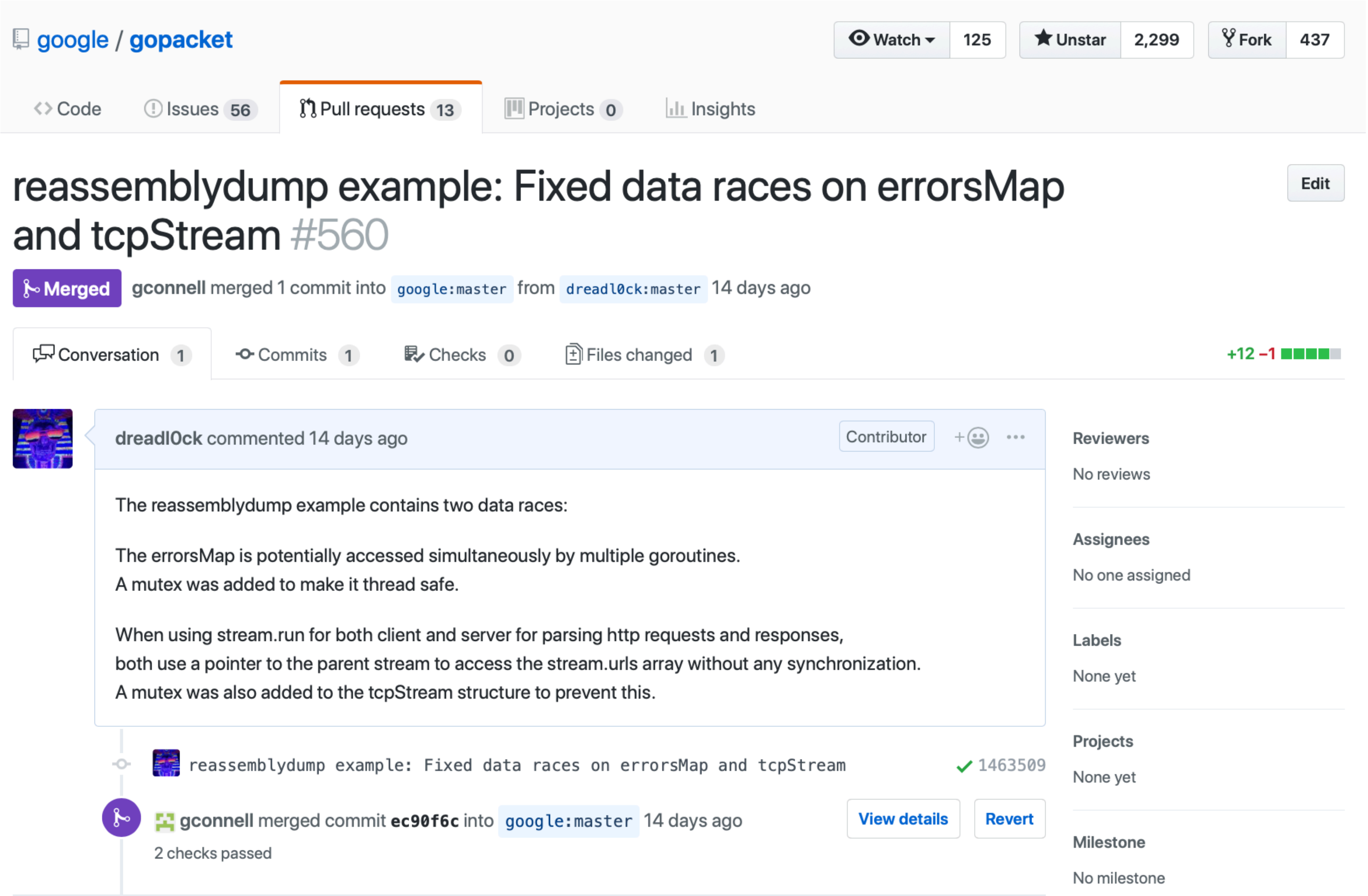
Monitor honeypots

Monitor medical devices

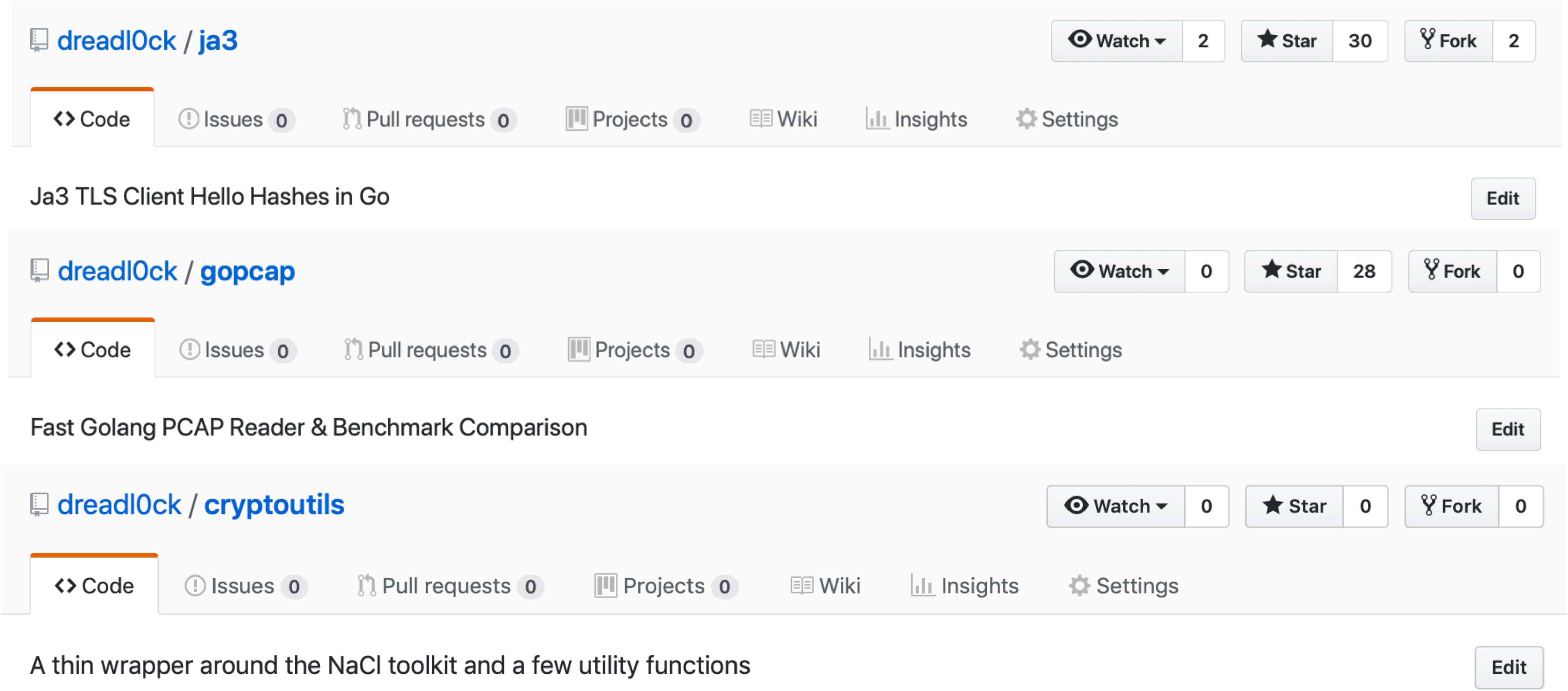
Forensic Analysis

Research! :) - GPLv3 license

Open source contributions during development



Open source contributions during development



PoC || GTFO

Classification of malicious behavior with Tensorflow and a Deep Neural Network

LMU LUDWIG-MAXIMILIANS-UNIVERSITÄT MÜNCHEN

UNB EST. 1785 UNIVERSITY OF NEW BRUNSWICK

Give to UNB Apply

Canadian Institute for Cybersecurity

About Research Members Datasets

Datasets

- IDS 2012 ›
- IDS 2017 ›
- NSL-KDD ›
- VPN-nonVPN ›
- Botnet ›
- Android Validation ›
- Android Botnet ›
- Tor-nonTor ›
- Dos Dataset ›
- Android-Adware ›
- Android-Malware2017 ›
- CSE-CIC-IDS2018 ›

Intrusion Detection Evaluation Dataset (CICIDS2017)

Intrusion Detection Systems (IDSs) and Intrusion Prevention Systems (IPSs) are the most important defense tools against the sophisticated and ever-growing network attacks. Due to the lack of reliable test and validation datasets, anomaly-based intrusion detection approaches are suffering from consistent and accurate performance evolutions.

Our evaluations of the existing eleven datasets since 1998 show that most are out of date and unreliable. Some of these datasets suffer from the lack of traffic diversity and volumes, some do not cover the variety of known attacks, while others anonymize packet payload data, which cannot reflect the current trends. Some are also lacking feature set and metadata.

CICIDS2017 dataset contains benign and the most up-to-date common attacks, which resembles the true real-world data (PCAPs). It also includes the results of the network traffic analysis using CICFlowMeter with labeled flows based on the time stamp, source and destination IPs, source and destination ports, protocols and attack (CSV files).

Generating realistic background traffic was our top priority in building this dataset. We have used our proposed B-Profile system (Sharafaldin, et al. 2016) to profile the abstract behaviour of human interactions and generates a naturalistic benign background traffic. For this dataset we built the abstract behaviour of 25 users based on the HTTP, HTTPS, FTP, SSH, and email protocols.

Dataset:

Up to date

Well documented

~50GB original PCAPs

Monday: Normal Traffic

Tuesday: Brute Force

Wednesday: DoS

Thursday: Web Attacks

Friday: Botnet Traffic

Experiment 1

Numeric values: zscore
Strings: dummy variables
Booleans: numeric (0 or 1)
Labels: attack class

LMU LUDWIG-MAXIMILIANS-UNIVERSITÄT MÜNCHEN

Tuesday-WorkingHours.pcap

File	Num Records	Size	Labels	Exec Time	Validation Score
Connection_labeled.csv	284166	51 MB	2119	50s	0.9919065381096488
DNS_labeled.csv	700447	144 MB	33	3h 27m	0.9999428946692174
Ethernet_labeled.csv	11551954	880 MB	3556	6m 49s	0.9996693201846267
Flow_labeled.csv	647294	112 MB	4852	4m 2s	0.9904835470415573
HTTP_labeled.csv	45852	14 MB	5609	21s	0.9637554585152839
IPv4_labeled.csv	11469736	1.0 GB	3555	-	-
NTP_labeled.csv	15507	2.1 MB	18	2s	0.9987113402061856
TCP_labeled.csv	10710230	1.5 GB	3504	-	-
TransportFlow_labeled.csv	91861	5.8 MB	11	8s	0.9999059354717336
UDP_labeled.csv	787015	44 MB	51	27s	0.9999491753703845

Table 6.7: Classification results experiment 1 Tuesday-WorkingHours.pcap

Experiment 2

Numeric values: standard score
Strings: index
Booleans: numeric (0 or 1)
Labels: attack class

Tuesday-WorkingHours.pcap

File	Num Records	Size	Labels	Exec Time	Validation Score
Connection_labeled.csv	284166	51M	1807	49s	0.993637566509951\8
DNS_labeled.csv	700447	144M	38	1m 57s	0.9999600255836265
Ethernet_labeled.csv	11551954	880M	3549	33m 41s	0.9996984060534857
Flow_labeled.csv	647294	112M	3340	1m 51s	0.9946855843385406
HTTP_labeled.csv	45800	14M	4214	38s	0.9275109170305676
NTP_labeled.csv	15507	3.0M	11	4s	0.9989682744389993
NetworkFlow_labeled.csv	29094	4.0M	1291	15s	0.9573824580698378
TCP_labeled.csv	10710230	1.5G	3450	29m 59s	0.9996728362186739
TransportFlow_labeled.csv	212613	20M	1721	44s	0.99153403318659
UDP_labeled.csv	787015	43M	44	2m 2s	0.9999440926232758

Table 6.11: Classification results experiment 2 Tuesday-WorkingHours.pcap

Experiment 3

Numeric values: standard score
Strings: index
Booleans: numeric (0 or 1)
Labels: attack description
Dropped lines with NaNs

LMU LUDWIG-MAXIMILIANS-UNIVERSITÄT MÜNCHEN

Tuesday-WorkingHours.pcap

File	Num Records	Size	Labels	Exec Time	Validation Score
Connection_labeled.csv	284166	51M	1803	1m 21s	0.9929900622167168
DNS_labeled.csv	700447	145M	38	2m 48s	0.999965736214537
Ethernet_labeled.csv	11551954	880M	3547	43m 36s	0.9996973672683657
Flow_labeled.csv	647294	111M	3334	1m 57s	0.9947412003163931
HTTP_labeled.csv	45795	14M	4206	38s	0.9336186566512359
NTP_labeled.csv	15507	2.1M	16	6s	0.9987103430487491
NetworkFlow_labeled.csv	29094	4.0M	1288	6s	0.9586197415452296
TCP_labeled.csv	10710230	1.5G	3444	29m 26s	0.9996728362186739
TransportFlow_labeled.csv	212613	20M	1715	34s	0.9918350453399556
UDP_labeled.csv	787015	43M	49	2m 1s	0.9999339276456896

Table 6.23: Classification results experiment 3 Tuesday-WorkingHours.pcap

Experiment 4

Numeric values: zscore
Strings: index
Booleans: numeric (0 or 1)
Labels: attack description
Dropped SrcIP, DstIP fields

LMU LUDWIG-MAXIMILIANS-UNIVERSITÄT MÜNCHEN

Tuesday-WorkingHours.pcap

File	Num Records	Size	Labels	Exec Time	Validation Score
Connection_labeled.csv	284166	51M	1803	1m 31s	0.9936516426902395
DNS_labeled.csv	700447	134M	36	2m 9s	0.999965736214537
Ethernet_labeled.csv	11551954	870M	3550	34m 58s	0.9996987523151923
Flow_labeled.csv	647294	111M	3334	2m 24s	0.9947720980818667
HTTP_labeled.csv	45823	13M	4206	31s	0.9397695530726257
NTP_labeled.csv	15507	2.1M	17	6s	0.9987103430487491
NetworkFlow_labeled.csv	29094	4.0M	1288	17s	0.9558702227110256
TCP_labeled.csv	10710230	1.5G	3444	32m 53s	0.9996720892694014
TransportFlow_labeled.csv	212613	20M	1715	58s	0.9916657260036874
UDP_labeled.csv	787015	43M	50	2m 7s	0.9999288451568964

Table 6.27: Classification results experiment 4 Tuesday-WorkingHours.pcap

Experiment 5

Numeric values: zscore
Strings: index
Booleans: numeric (0 or 1)
Labels: attack class
Ignored labels of class “Generic Protocol Command Decode”

LMU LUDWIG-MAXIMILIANS-UNIVERSITÄT MÜNCHEN

Tuesday-WorkingHours.pcap

File	Num Records	Size	Labels	Exec Time	Validation Score
Connection_labeled.csv	284166	51M	67	53s	0.9998170096562596
DNS_labeled.csv	700447	133M	33	2m 0s	0.999965736214537
Ethernet_labeled.csv	11551954	868M	121	31m 1s	0.9999885733636797
Flow_labeled.csv	647294	111M	49	1m 47s	0.9999258453628633
HTTP_labeled.csv	45800	13M	1461	16s	0.9939737991266375
NTP_labeled.csv	15507	2.1M	13	5s	0.9989682744389993
NetworkFlow_labeled.csv	29094	3.3M	33	5s	0.9989001924663184
TCP_labeled.csv	10710230	1.5G	75	28m 47s	0.9999925305072757
TransportFlow_labeled.csv	212613	19M	59	33s	0.9996613613274636
UDP_labeled.csv	787015	43M	46	2m 1s	0.9999390101344826

Table 6.31: Classification results experiment 5 Tuesday-WorkingHours.pcap

Experiment 6

Numeric values: standard score
Strings: index
Booleans: numeric (0 or 1)
Labels: attack description
Collecting Labels + Ignoring "GPCD" errors

File	Num Records	Size	Labels	Exec Time	Validation Score
Connection_labeled.csv	284166	51M	68	50s	0.9996340193125194
DNS_labeled.csv	700447	144M	33	1m 57s	0.999965736214537
Ethernet_labeled.csv	11551954	880M	121	29m 39s	0.999989265887093
Flow_labeled.csv	647294	112M	49	1m 52s	0.9999320249159581
HTTP_labeled.csv	45790	13M	1461	17s	0.9951956673654787
NTP_labeled.csv	15507	3.0M	13	4s	0.9987103430487491
NetworkFlow_labeled.csv	29094	4.0M	33	5s	0.9984877646411878
TCP_labeled.csv	10710230	1.5G	75	27m 26s	0.9999925305072757
TransportFlow_labeled.csv	212613	20M	59	46s	0.9996613613274636
UDP_labeled.csv	787015	43M	46	1m 57s	0.9999440926232758

Table 6.35: Classification results experiment 6 Tuesday-WorkingHours.pcap

Experiments takeaways

Encoding strategies are vital for performance

High detection accuracy can be achieved with only a handful of extracted features

Different approaches to labeling can be used to increase value for analysts

High accuracy for protocol specific approach

Conclusion

Use **memory-safe** programming languages for the development of **critical software infrastructure**

Developers need to focus on creating **solid logic**, rather than solid memory management.

Golang || Rust

Contact: philipp.mieden@protonmail.ch | twitter.com/dreadcode | github.com/dreadl0ck