I JUST WANTED TO…
GRAFUNA RED TEAM

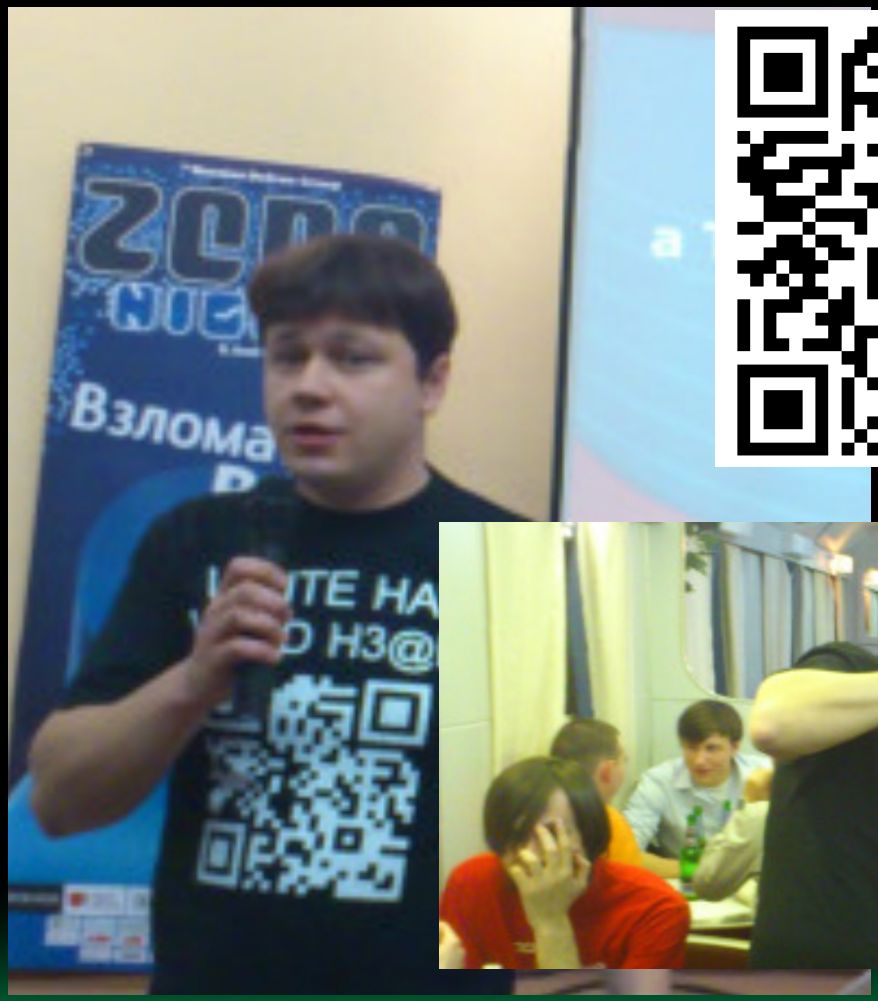
Sergey Gordeychik

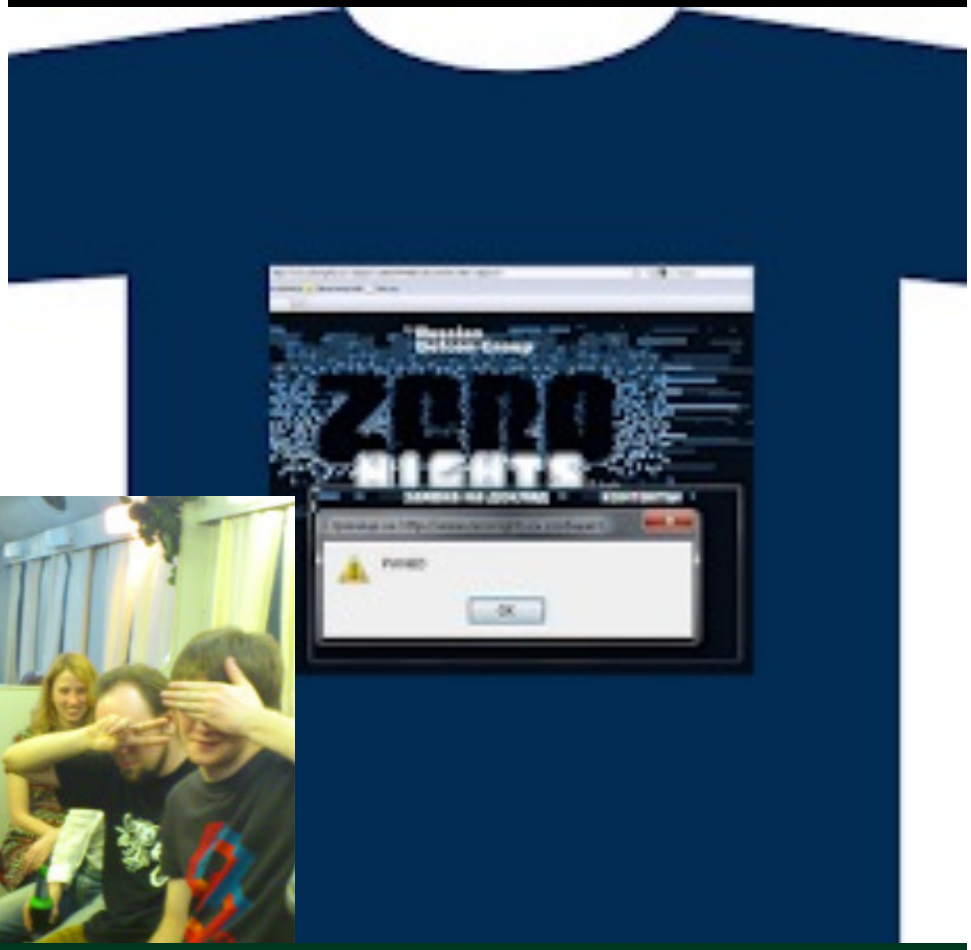
CYBEROK

ZERONIGHTS

whoami

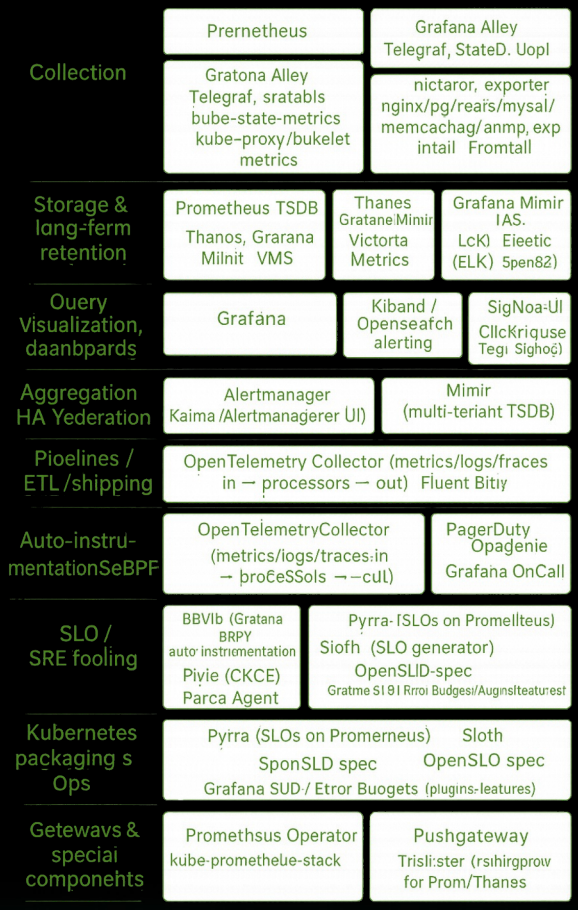
https://www.youtube.com/watch?v=WNtALw2fTYY





Cloud-native observability stack

- aka **PAG** aka **LGTM**
- an open-source toolkit for metrics, logs, traces, and profiling across modern apps and infra
- Kubernetes/microservices/dockers
- Tasks
  - Collection (emitters & agents)
  - Storage & long-term retention
  - Query, visualization, dashboards
  - Alerting & incident routing
  - Aggregation, HA, federation
  - Pipelines / ETL / shipping



Book of Observability: Genesis

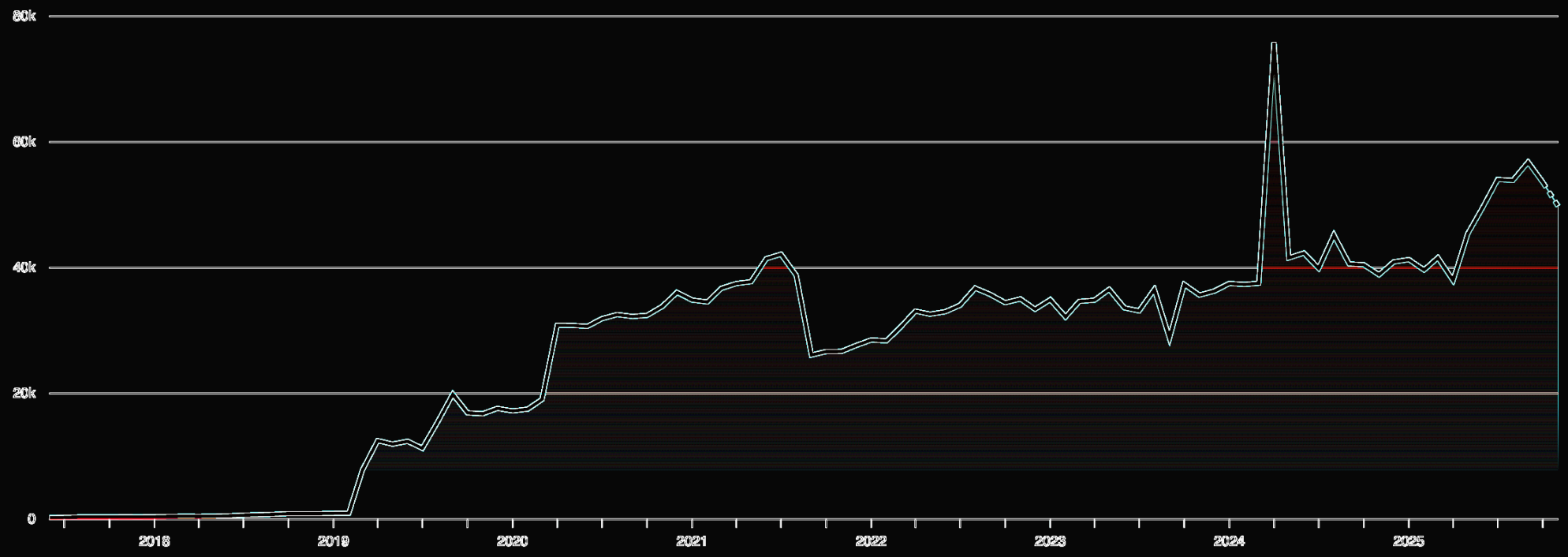
ZERO NIGHTS

- **2019** — *Zen mode*: dashboards glow, Alertmanager naps.
- **2020** — *Curiosity kicks in*: first naïve hacks, /metrics left wide open “for science”.
- **2021** — *Childhood diseases*: big research wave; Grafana plugin traversal (CVE-2021-43798).
- **2022** — *Exporters bite back*: Toolkit auth-cache bug (CVE-2022-46146); /debug/pprof season.
- **2023** — *Auth soap opera*: Azure AD OAuth facepalm (CVE-2023-3128); Alertmanager XSS cameo.
- **2024** — *Exposure at scale*: “hundreds of thousands” of open Prometheus—observability becomes an attack surface.
- **2025** — *Enterprise plot twist*: Grafana SCIM (CVE-2025-41115, 10/10); GitHub Actions scare; patch, patch, patch.

Prometheus

ZERO NIGHTS

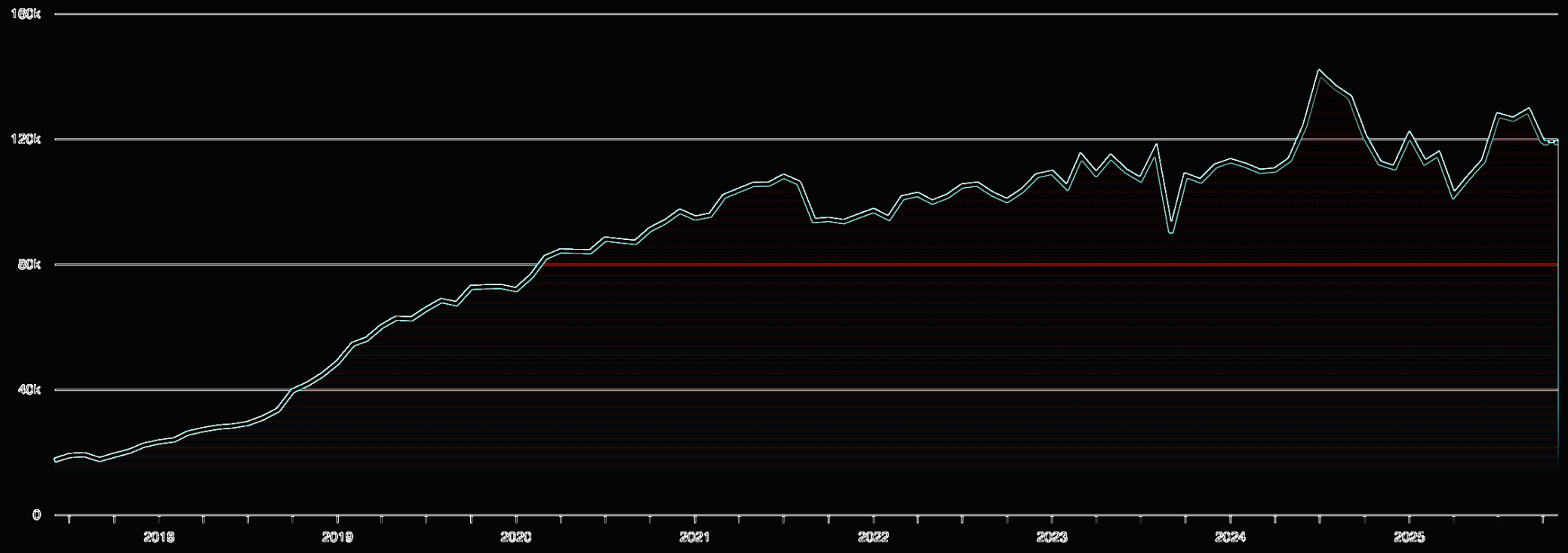
24 MONTHS AGO	12 MONTHS AGO	6 MONTHS AGO	3 MONTHS AGO	1 MONTH AGO	OCT 2025
37,360	40,488	37,974	54,042	56,814	53,584
↑ 43.43%	↑ 32.35%	↑ 41.11%	↓ 0.85%	↓ 6.03%	



Grafana

ZERO NIGHTS

24 MONTHS AGO	12 MONTHS AGO	6 MONTHS AGO	3 MONTHS AGO	1 MONTH AGO	OCT 2025
108,558	120,989	102,055	127,749	129,221	118,902
↑ 9.53%	↓ 1.76%	↑ 16.51%	↓ 7.44%	↓ 8.68%	



Мы пилим СКИПА ака “Русский Шодан” ака “Балалайка”

- Лучше чем Shodan [done]
- Лучше чем Fofa [loading…]

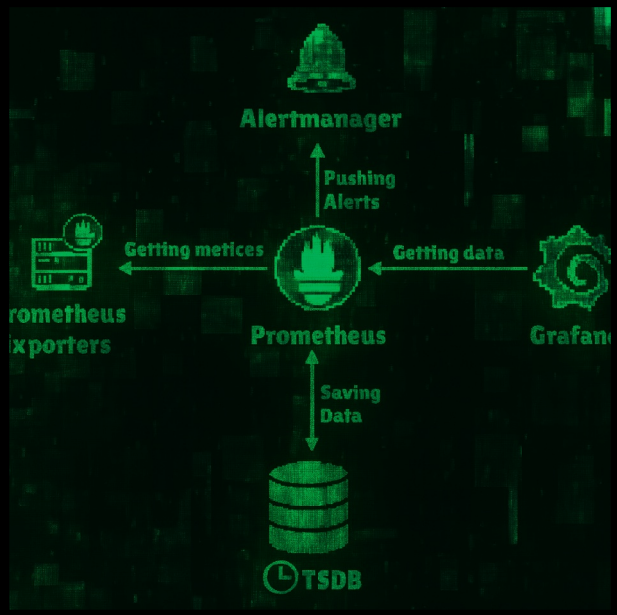
Каждый год на мы рассказываем о рунете:

- 47 миллионов IP
- 150 000 000 доменных имен
- 51 000 0000 “живых“ сервисов на IPv4/6

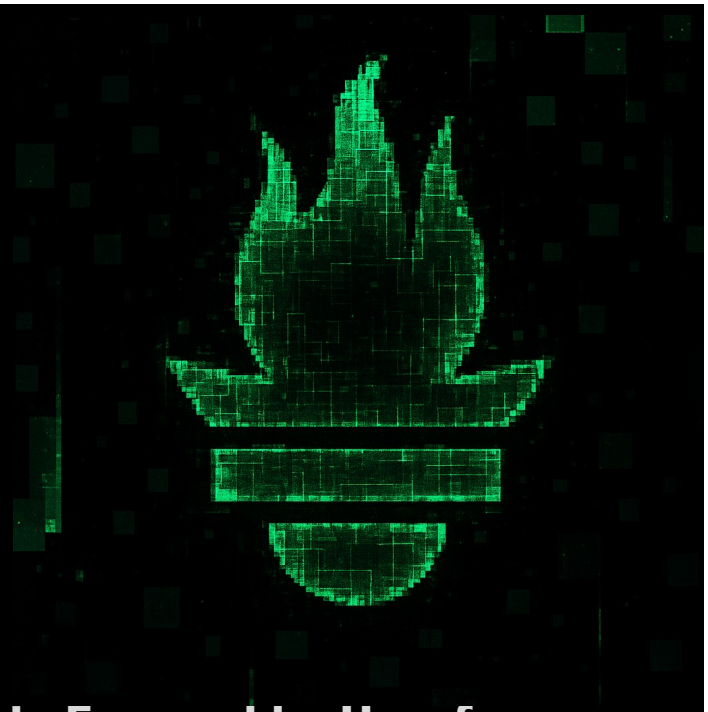
OK, LET'S HACK A BIT

Prometheus

- Prometheus is an open-source **monitoring and alerting toolkit**, widely adopted in cloud-native and DevOps ecosystems.
- **Scraping model:** Prometheus **pulls metrics** from targets via HTTP endpoints (/metrics).
- **PromQL:** slicing, aggregating, and analyzing metric data in real time.
  - **Prometheus Server** — scraping, querying, storage.
  - **Alertmanager** — grouping, routing, deduplication of alerts.
  - **Exporters** — adapters exposing metrics for systems without native Prometheus support (Node Exporter, Blackbox, PostgreSQL, Nginx, etc.).
  - **Grafana** — the most common visualization layer.



Light my fire



- **Known security issues / pitfalls**
  - No authentication or authorization "out of the box"
  - Endpoints and APIs often exposed to the internet
  - Leaks sensitive data via labels and metadata

- **Selected security research**
  - **"Don't Let Prometheus Steal Your Fire – Real World Secrets Exposed by Unsafe Defaults"** (JFrog)
  - **"How Attackers Use Exposed Prometheus Server to Exploit Kubernetes Clusters"** (Sysdig / KubeCon EU talk)
  - **"300,000+ Prometheus Servers and Exporters Exposed to DoS Attacks"** (Aqua Security)

Is Anybody Out There?

Exporters

updated:>"1.10.2025" and location.country:"RU" and not domain:"*" AND product.name:"prometheus node exporter"

Города

moscow	8,180
st petersburg	2,382
novotroitsk	174
lytkarino	165
ramenskoye	86
novosibirsk	85
irkutsk	53
yelabuga	49
biysk	38
nizhniy novgorod	35

Организации

jsc selectel	6,639
yandex.cloud llc	5,952

на 29 418 сервисах (уникальных IP-адресов: 29 299)

TCP / HTTP / 9100 [redacted]

Россия, (55.738602, 37.6068)

Prometheus Node Exporter

ASN

Наименование: JSC Selectel
Номер: 49505

Abuse

Наименование: JSC Selectel
Email: abuse@selectel.ru
Адрес: OOO "Network of data-centers "Selectel", Tsvetochnaya 21, 196006, Saint-Petersburg, RUSSIAN FEDERATION

Title: 82a0f88ef0578b248343c02… Body: 5b2l

Title: Node Exporter

Banner size: 136 bytes Response time: 4 s

```
HTTP/1.1 200 OK
Connection: close
Content-Length: 150
```

Prometheus Server

updated:>"1.10.2025" and location.country:"RU" and not domain:"*" AND (http.title:"Prometh

999+

Города

moscow	1,058
st petersburg	584
kazan'	34
korolyov	15
novosibirsk	15
cheboksary	8
perm	7
yekaterinburg	6
chelyabinsk	5
orenburg	5

на 3 432 сервисах (уникальных IP-адресов: 3 361)

TCP / HTTP / 9090 [redacted]

Россия, Moscow, Moscow, (55.748699, 37.618698)

Prometheus

ASN

Наименование: Yandex.Cloud LLC
Номер: 200350

Abuse

Banner: 8b9a395eb31

Banner size: 156 bytes

Special guests

- Native and app-specific exporters
- Pmth-style (mtail, Smokeping Prober)
- Thanos
- Prometheus Pushgateway
- Levelups
  - Redirects
  - IPv6 things
  - Domains
  - Ports
- …
- Mr AlertManger

nginx prometheus exporter	11,079
prometheus pushgateway	1,071
prometheus sql exporter	78
alertmanager – prometheus	50
aerospike prometheus exporter	35
prometheus java client	29

Порты	
9100	94,525
9113	10,792
9104	6,447
9902	4,772
3903	2,593
9091	937
9374	907
9101	575
9187	553
9200	523

node exporter	12,708
mtail on [::]:3903	2,555
postgres exporter	323
nginx prometheus exporter	257
mysqld exporter	255
prometheus pushgateway	243
windows exporter	165
apache exporter	101
smartctl_exporter	58

Not “Prometheus” but still /metrics

```
Not Found! The Prometheus Ruby Exporter only listens on /ping, /metrics and /send-metrics
```

Using the URLconf defined in `my_django_project.urls`, Django tried these URL patterns, in this order:

```
 1. admin/
 2. api/accounts/
 3. api/subscriptions/
 4. api/projects/
 5. api/chats/
 6. api/keywords/
 7. api/messages/
 8. swagger/ [name='schema-swagger-ui']
 9. swagger.json [name='schema-json']
10. swagger.yaml [name='schema-yaml']
11. metrics [name='prometheus-django-metrics']
```

The empty path didn’t match any of these.

What can we find there?

- /metrics, Cap!

- /debug
  - /pprof/* – Go pprof profiling endpoints if enabled :)
  - /vars - Go expvar variables, includes internal stats

- Looks very bad
  - DoS, Info Disclosure
- Still enabled by default
- Available on 60%+

Feature request - cmdline flag to disable/change url for debug pprof endpoints #2860

Closed

1234568909

discordianfish closed this as completed on Dec 18, 2023

discordianfish mentioned this on Dec 19, 2023

security vulnerability fix - add new cmdline flags to disable/change ... #2859

J. Random /metrics

- Internal stuff – everywhere
  - service addresses, hostnames, target IPs/ports, environment metadata
- Kuber stuff – enough
  - kube_node_info, node_dmi_info, kube_namespace
- PII & Personal data – 152 FZ protected
- Infrastructure 2 Infrastructure mapping
  - yep, EASM
- User/service names
- ~~Secrets~~

Server + Alertmanager

- **Useful API**
  - /api/v1/targets
  - /api/v1/status/flags
  - /api/v1/status/config
- Endpoints can tell you a story
  - /payment
  - /sms_in_queue
- username: qwerty. Let's try to guess password
- ~~Secrets~~
- **web.enable-admin-api = 'true'**
  - **/api/v1/admin/tsdb/snapshot** **Create a snapshot of the TSDB**
  - **/api/v1/admin/tsdb/delete_series** **Delete matching series from TSDB**
  - **Around 7% skydive for fun. The rest don't enable --web.enable-admin-api in Prometheus.**

Show me your /metrics, and I'll tell you who you are. Socrates

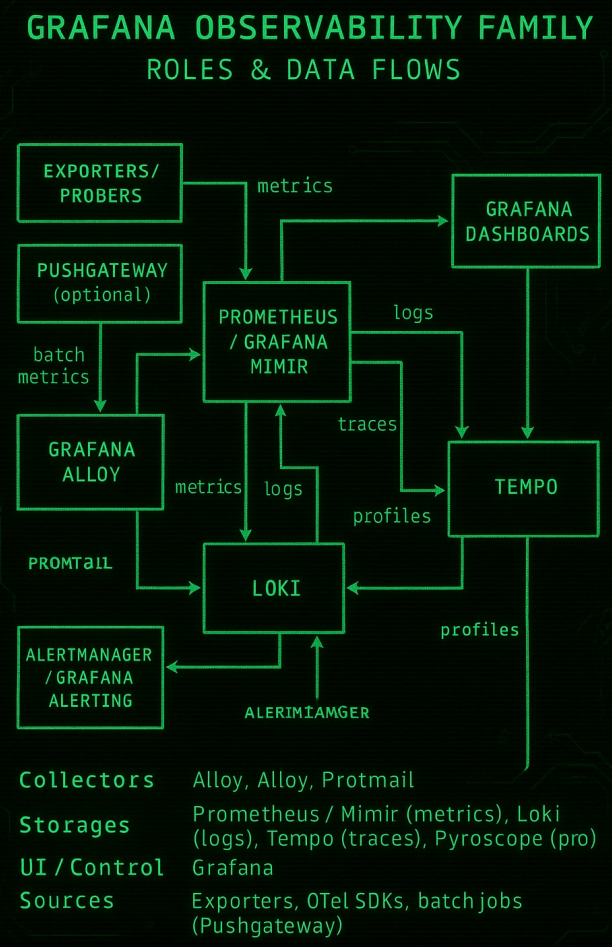
Push me pull me

- **Pushgateway**
  - Accepts any HTTP push (no auth, no rate limit)
  - POST or PUT on the /metrics/job/... path
  - Parses the body in Prometheus exposition format
  - Stores the pushed data in memory
  - Exposes it via /metrics for Prometheus to scrape
- Easy to
  - dos
  - spoof
  - delete
- Nice second chance vector

GRAFUN

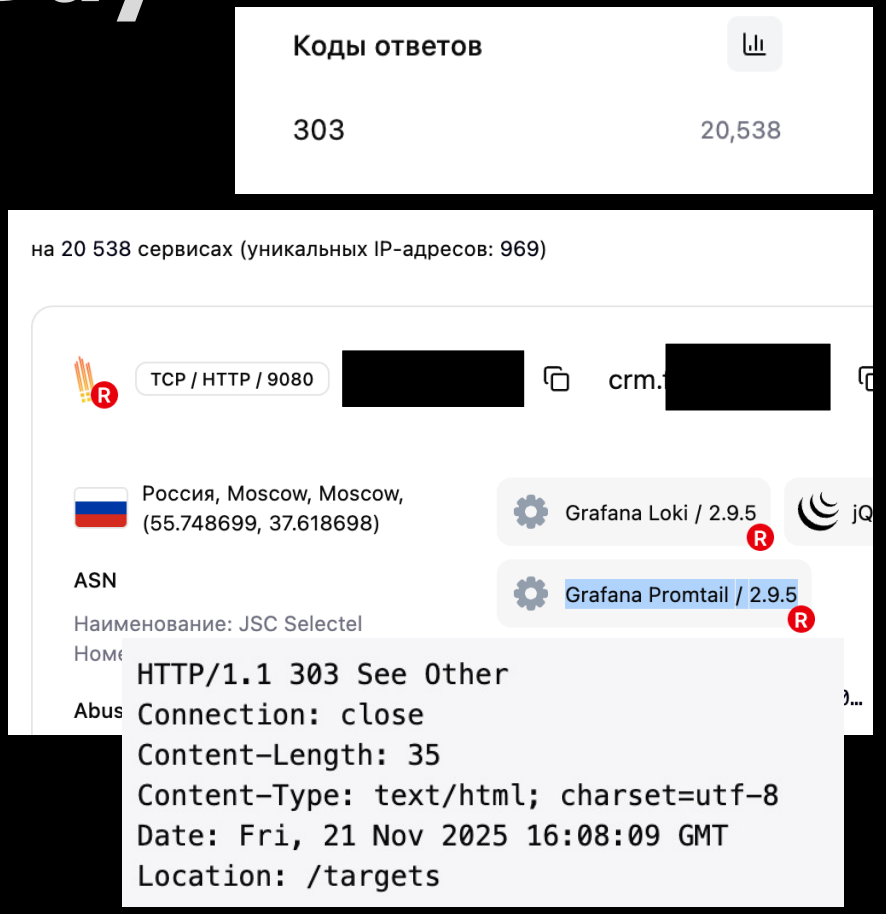
Grafana family

- Grafana Dashboards (UI)
  - My precious-s-s-s-s
- Grafana Alloy (ex Grafana Agent)
  - collector/forwarder
- Grafana Promtail
  - Log shipper for Loki
- Grafana Loki
  - A horizontally scalable log database
- Grafana Mimir
  - metrics storage & query



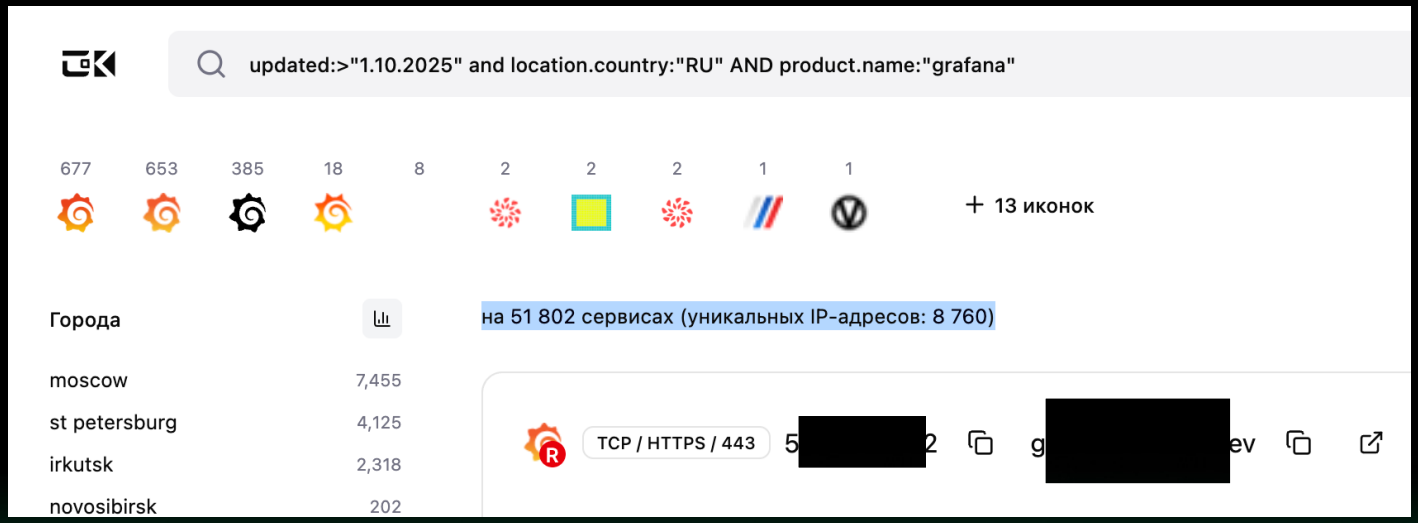
Same Metrics, Different Day

- **Grafana Alloy**
  - **/metrics, /-/ready, /-/healthy**
- **Grafana Promtail**
  - **/metrics, /config, /loki/api/v1/push, /promtail/api/v1/raw**
- **Grafana Loki**
  - **/metrics/ /ready, /log_level, /config, /services, /loki/api/v1/status/buildinfo**
- **No auth**



My precious-s-s-s-s

- Public – sometime
- Internal – a lot
- Weak/default creds
- Creds from /metrics
- Open sign-on, unath API



Access via API

- **Lucky man**
  - **Base check**
    - /api/health – anonymous access
    - /api/org – protected (should be)

```
{
  "id": 1,
  "name": "Main Org.",
  "address": { ... }
}
```

- **Users**: /api/org/users (roles), /api/users (site info)
- **Datasources**: /api/datasources (servers, endpoints, connection strings, users)
- **Plugins**: /api/plugins (id, type, version, signed/enabled)
- **Dashboards & Folders**: /api/search?limit=5000, /api/dashboards/uid/<uid>

Automate!

marcusolsson-csv-datasource	marcusolsson-csv-datasource	proxy	False	False	/etc/passwd
prometheus	prometheus	proxy	False	False	
prxoyds	yesoreyeram-infinity-datasource	proxy	False	False	http://detectportal.firefox.com/success.txt?ipv4

Health

```
{
  "database": "ok",
  "version": "11.3.0",
  "commit": "d9455ff7db73b694db7d412e49a68bec767f2b5a"
}
```

Users (1)

Notes: GET http://localhost:9300/grafana/api/users -> HTTP 403

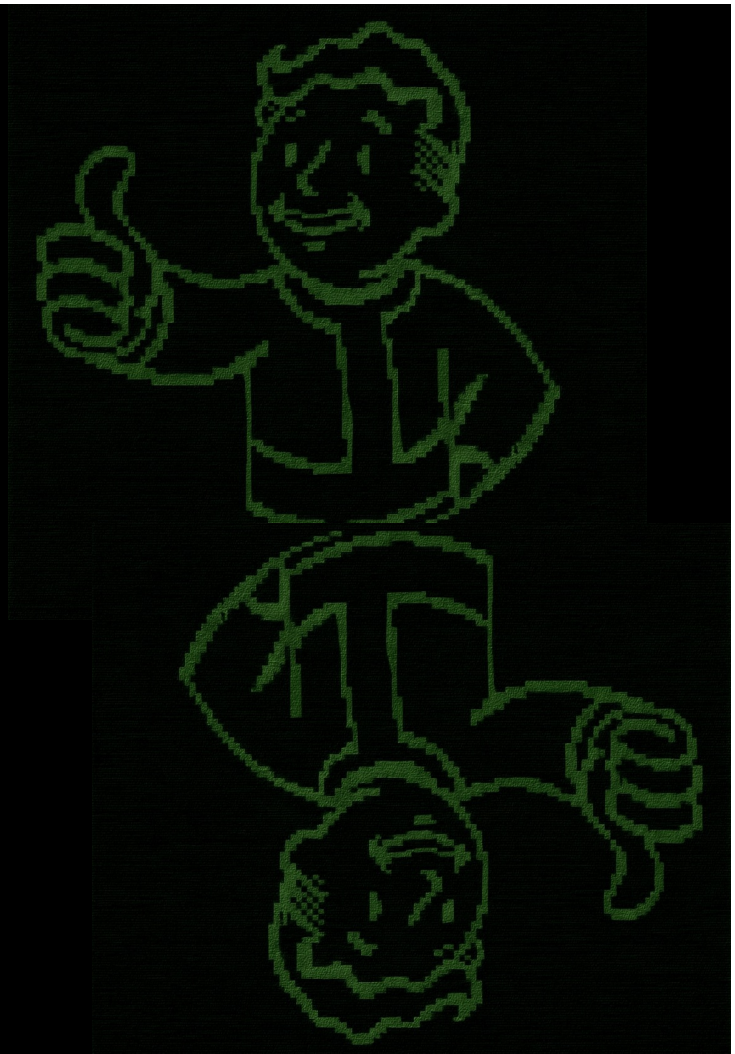
id	login	email	name	role	isAdmin	isDisabled	lastSeenAt
1	admin	admin@localhost		Admin	True		2025-11-25T17:44:55+03:0

w552bkc	PORT UDP	/gr
3pmg3kc	PROBES	/gr
h4feoa	RECON	/gr
456akgc	REST Search	/gr
n82cjka	RUTCP	/gr
n82cjka32	RUTCP (DEV)	/gr
n8rp4w0f	SERVICE	/gr
35e7swf	Session scan	/gr
nfjfggb	TECH SEARCH	/gr
gpxedcc	TERRA INCOGNITA	/gr
pca0aob	VENDOR	/gr
npscgsgb	Промка	/gr
xpgs8wb	Статистика сканирования	/gr

grafana_inventory.py

Your access level?

- User – Boring but useful
  - Browse data
  - View dashboards
  - ”Viewers can edit” (Explore)
  - Indirectly modify queries (Vars)
  - /api/ds/query “rawSql” (yep, ”SQLi”)
- Editor – getting hotter
  - (No)SQL Queries (yep, “SQLi” In da box)
  - Test/partially modify DATASOURCES
- Admin – Crown Jewel
  - Create users
  - CREATE and MODIFY DATASOURCES



SQLi by design

Avoid SQL injection when tampering with the queries

Grafana Dashboards

For example, in a Grafana instance with one data source, one dashboard, and one panel that has one query defined, you might assume that a Viewer can only see the result of the query defined in that panel. Actually, the Viewer has access to send any query to the data source. With a command-line tool like curl (there are many tools for this), the Viewer can make their own query to the data source and potentially access sensitive data.

Grafana backend sql injection affected all version

Vuln Description

The open–source platform for monitoring and observability

to exploit this sql injection vulnerability, someone must use a valid account login to the grafana web backend, then send malicious POST request to /api/ds/query “rawSql” entry.

to the grafana web backend, they can use a post request to /api/ds/query
n modify the “rawSql” filed to execute Malicious sql strings leading to time–
ıjection vulnerability, then leak data from databases.

- https://fdlucifer.github.io/2024/04/22/grafana-sql-injection/
- https://community.grafana.com/t/avoid-sql-injection-when-tampering-with-the-queries/102303
- https://community.grafana.com/t/custom-plugin-injects-sql/6049

DATASOURCES?!!1111

- Connects to data sources, Cap!
- In different ways
  - Time-Series Databases: Prometheus, InfluxDB
  - Log Aggregation: Loki, Elasticsearch, OpenSearch
  - Relational Databases: PostgreSQL, MySQL
  - Cloud Services: AWS CloudWatch, Azure Monitor, Google Cloud Monitoring
  - Web-services and API
- [I] [Can] force Grafana to connect to Web-server?..

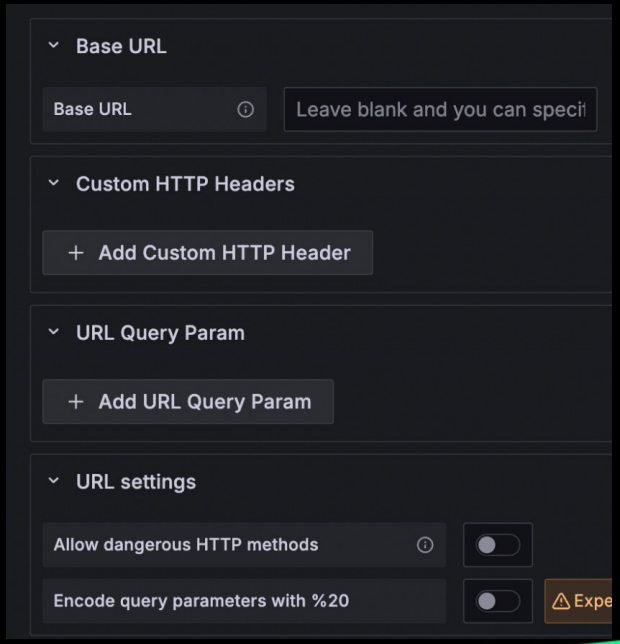
a vulnerability where an attacker tricks a server into making unintended requests to internal or external resources

SSRF definition

SSRF in da box

- Infinity data source – SSRF Swiss army knife
- HTTP/HTTPS, different auth types, proxies
- ”Dangerous” METHODS (ouch)
- Custom headers, query parameters
- JSON, CSV, GraphQL, XML, and HTML …

- So you can connect anywhere
  - Inside the network
  - Outside the network
  - Even use a proxy if needed



OK, ◻ к, CyberOK


```
"data": "\u003c!DOCTYPE html\u003e\n\u003chtml lang=\"ru\"\u003e\n\u003chead\u003e\n    \u003cmeta charset=\"UTF-8\"\u003e\n    \u003cmeta
http-equiv=\"X-UA-Compatible\" content=\"IE=edge\"\u003e\n    \u003cmeta name=\"viewport\" content=\"width=device-width,
initial-scale=1.0\"\u003e\n    \u003c!-- accordion css--\u003e\n    \u003clink rel=\"stylesheet\" href=\"css/accordion.min.css\"\u003e\n
\u003c!-- css --\u003e\n    \u003clink rel=\"stylesheet\" href=\"css/normalize.css\"\u003e\n    \u003clink rel=\"stylesheet\" href=\"css/
style.css\"\u003e\n    \u003ctitle\u003eCYBEROK - пентест, appsec, кибербезопасность\u003c/title\u003e\n\u003c/
head\u003e\n\u003cbody\u003e\n  \u003csvg style=\"display:none\"\u003e\n    \u003csymbol id=\"twitter\" viewbox=\"0 0 50 50\"
xmlns=\"http://www.w3.org/2000/svg\"\u003e\n      \u003cpath d=\"M42.5866 0H7.41336C5.44922 0.0100725 3.56571 0.785812 2.1757
2.17582C0.785691 3.56582 0 5.44908 0 7.41322V42.5863C0.0100725 44.5504 0.785691 46.4342 2.1757 47.8242C3.56571 49.2142 5.44922 50 7.41336
50H42.5866C44.5508 49.9899 46.4343 49.2142 47.8243 47.8242C49.2143 46.4342 50 44.5504 50 42.5863V7.41322C49.9899 5.44908 49.2143 3.56582
47.8243 2.17582C46.4343 0.785812 44.5508 0 42.5866 0ZM45.83 42.5966C45.83 43.4629 45.4875 44.2886 44.8731 44.8929C44.2587 45.4973 43.4327
45.85 42.5766 45.85H7.40328C6.53704 45.85 5.71119 45.5074 5.10684 44.8929C4.49241 44.2785 4.14987 43.4528 4.14987 42.5966V7.42355C4.14987
6.55731 4.49241 5.73109 5.10684 5.12674C5.72126 4.51232 6.54711 4.17015 7.40328 4.17015H42.5766C43.4428 4.17015 44.2688 4.51232 44.8731
5.12674C45.4775 5.74116 45.83 6.56738 45.83 7.42355V42.5966Z\" fill=\"white\"/\u003e\n      \u003cpath d=\"M10.8179 34.0044C13.89 34.1857
16.66 33.4508 19.1983 31.5974C16.2168 30.9528 14.313 29.4317 13.9504 27.4475C14.7159 27.4475 15.4714 27.4475 16.2167 27.4475C14.9677 26.964
```

Turning a Datasource into a Port Mapper

- **Goal: map internal host:port reachability from Grafana's network position.**
- **Method: update a throwaway Infinity or MSSQL datasource → call /health → classify: open / closed / filtered / unknown.**
- **Why it works: Grafana's plugin/driver tries to speak to the target; responses & timing reveal state.**

```
python3 gmap3.py \
  -t "https://grafana.internal/connections/datasources/edit/LLALA" \
  -s "<grafana_session_cookie_value>" \
  -H "10.0.5.0/24 10.0.6.10-20 10.0.7.15" \
  -p "1433,1521,3306,5432,6379,8080-8090" \
  -o results.txt
```

gmap – what's next?

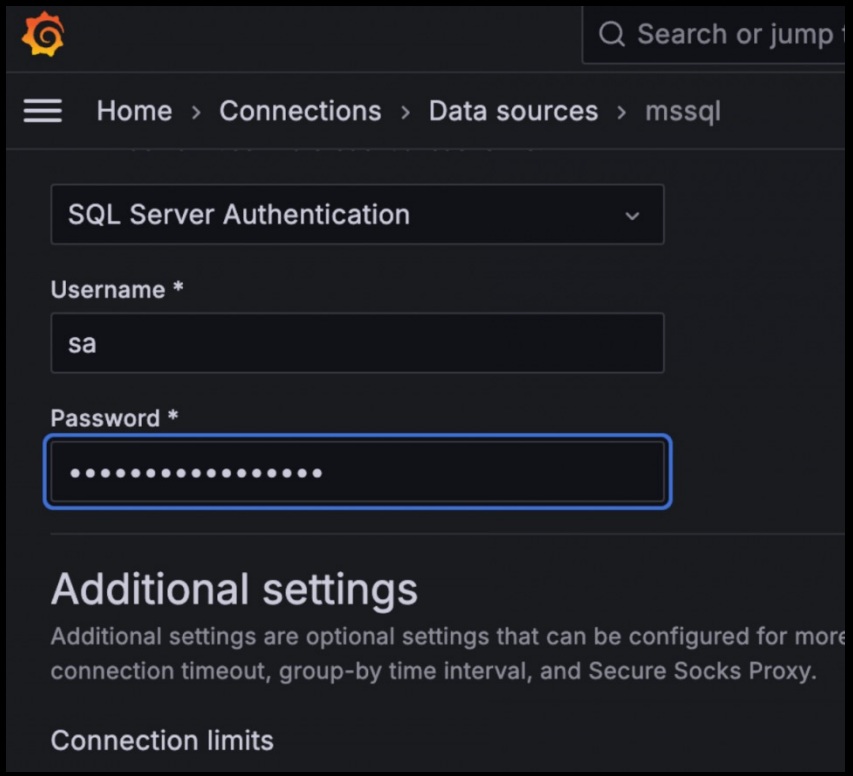
```
10.100.1.182:3000,Grafana
10.100.1.182:9094,connection refused
10.100.1.193:6565,"\\x00\\x00\\x00\\x04\\b\\x00\\x00\\x00\\x00\\x00\\x00\\x0
\x00\\x00\\x7f\\xff\\xff\\xff\\x00\\x00\\x00\\x01Unexpected\"
10.100.1.193:8001,connection refused
10.100.1.193:8080,connection refused
10.100.1.20:6565,"\\x00\\x00\\x00\\x04\\b\\x00\\x00\\x00\\x00\\x00\\x00\\x0f
\x00\\x00\\x7f\\xff\\xff\\xff\\x00\\x00\\x00\\x01Unexpected\""
10.100.1.20:8080,404
10.100.1.53:8080,kube-state-metrics
10.100.1.54:8080,404
10.100.1.54:9093,Alertmanager
10.100.1.54:9094,connection refused
10.100.1.77:6565,\\x00\\x00\\x00\\x04\\b\\x00\\x00\\x00\\x00\\x00\\x00\\x0f\
\x00\\x00\\x7f\\xff\\xff\\xff\\x00\\x00\\x00\\x01Unexpected\"
10.100.1.77:8001,connection refused
10.100.1.84:10250,tls,4040
10.100.1.85:6565,?
10.100.1.85:8080,404
10.100.1.86:8001,connection refused
10.100.1.86:8080,401
10.100.1.86:8090,connection refused
10.100.1.93:8080,404
10.100.1.93:9090,Prometheus Time Series Collection and Processing Server
10.100.3.230:6565,\\x00\\x00\\x00\\x04\\b\\x00\\x00\\x00\\x00\\x00\\x00\\x0f
\x00\\x00\\x7f\\xff\\xff\\xff\\x00\\x00\\x00\\x01Unexpected\"
10.100.3.230:8001,connection refused
10.100.3.247:4443,403,tls
```

HTTP/2 JSON?

Java/Spring gRPC?

- **HTTP →**
  - **Infinity/JSON API**
- **HTTP/2 JSON →**
  - **Infinity/JSON API**
- **gRPC →**
  - **Simple gRPC Datasource**
- **SQL →**
  - **Grafana has it all**
- **File system?**

brute things?



File system?

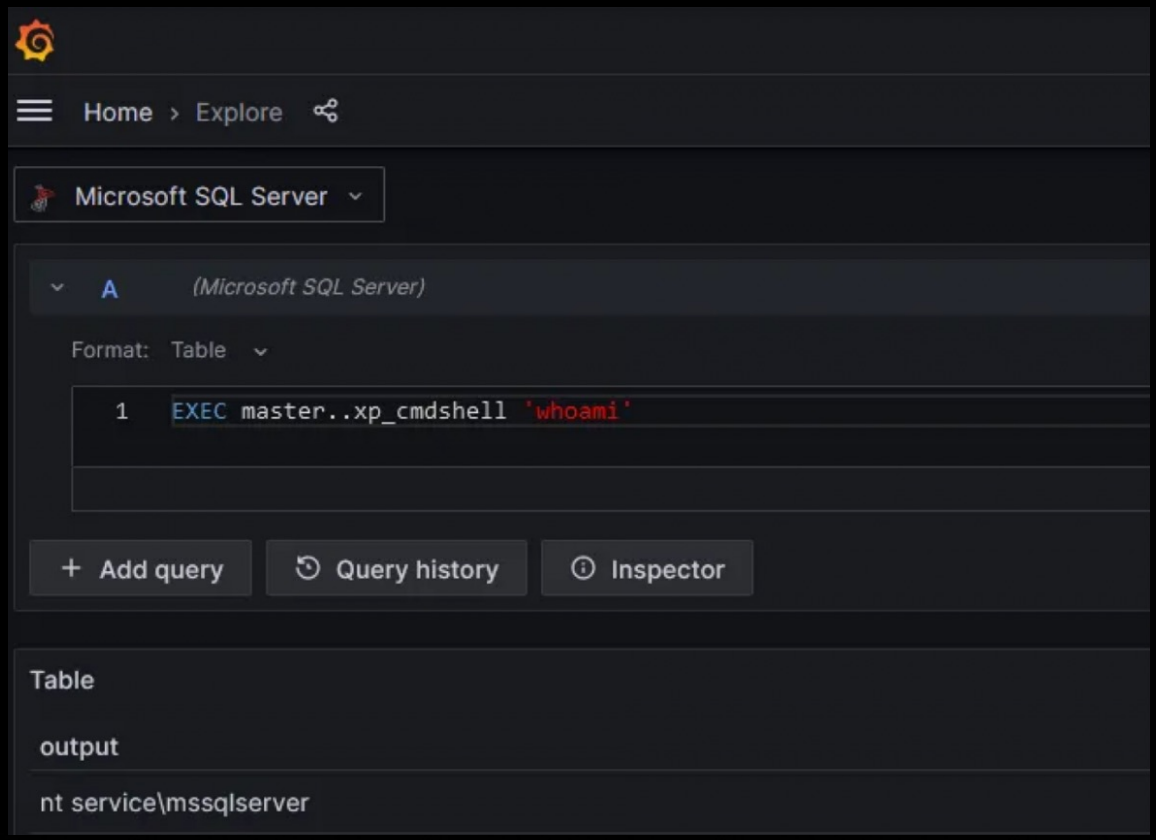
- **Actually no…**
- **But…**
- **Perhaps…**

marcusolsson-csv-datasource	proxy	False

	False	/etc/passwd

- **grafana.ini**
  - allow_local_mode = true
- **worth to try**

Srsly?


Home › Explore
Microsoft SQL Server
A (Microsoft SQL Server)
Format: Table
1 EXEC master..xp_cmdshell 'whoami'
+ Add query
Query history
Inspector
Table
output
nt service\mssqlserver

Let's Put In All Together

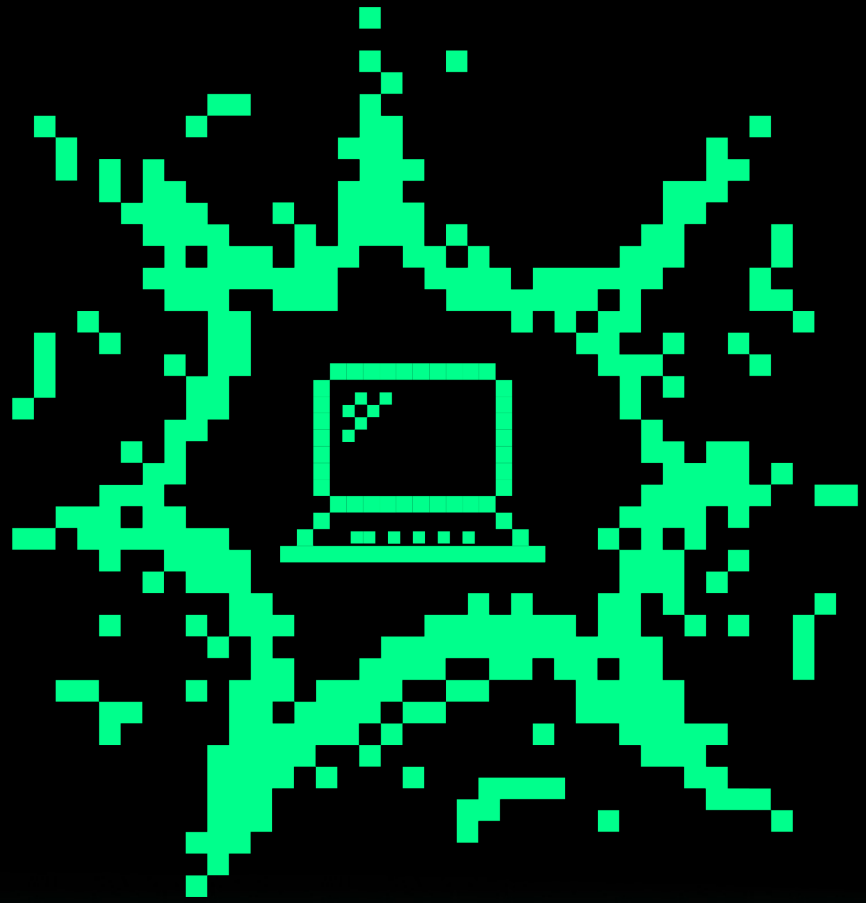
```
[Browser/Burp] ──HTTP──> [Grafana Proxy]
  ▲                  ▲ 1. Parse request (method/URL/qs/headers/body)
  |                  | 2. Build /api/ds/query payload for Infinity
  |                  ▼
  |            [Grafana /api/ds/query]
  |                  ▲ 3. Grafana runs Infinity (GET/POST, headers, params)
  |                  ▼
  |            [Target HTTP endpoint]
  |                  ▲ 4. Grafana returns dataframe/JSON
  |                  | 5. Proxy unwraps/normalizes to HTTP-like response
  ▼                  | 6. Response back to client
Results  <──────── HTTP ────────┘
```

ffuf is a Hacker's Best Friend

Don’t trust defaults

...

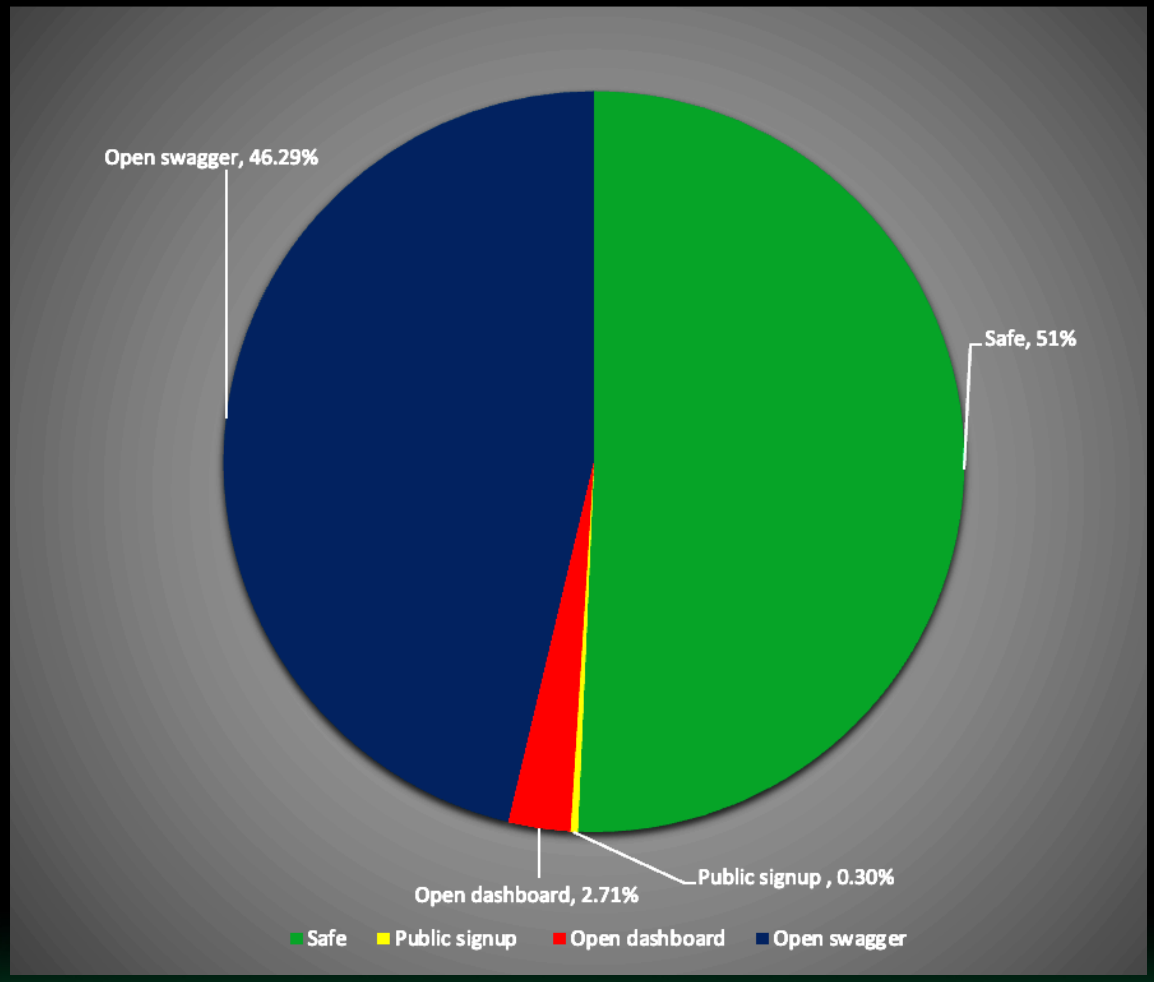
- /metrics/probes
- /total
- /metrics/cadvisor
- /metricsz (l33t)
- /probe
- /metrics
- …



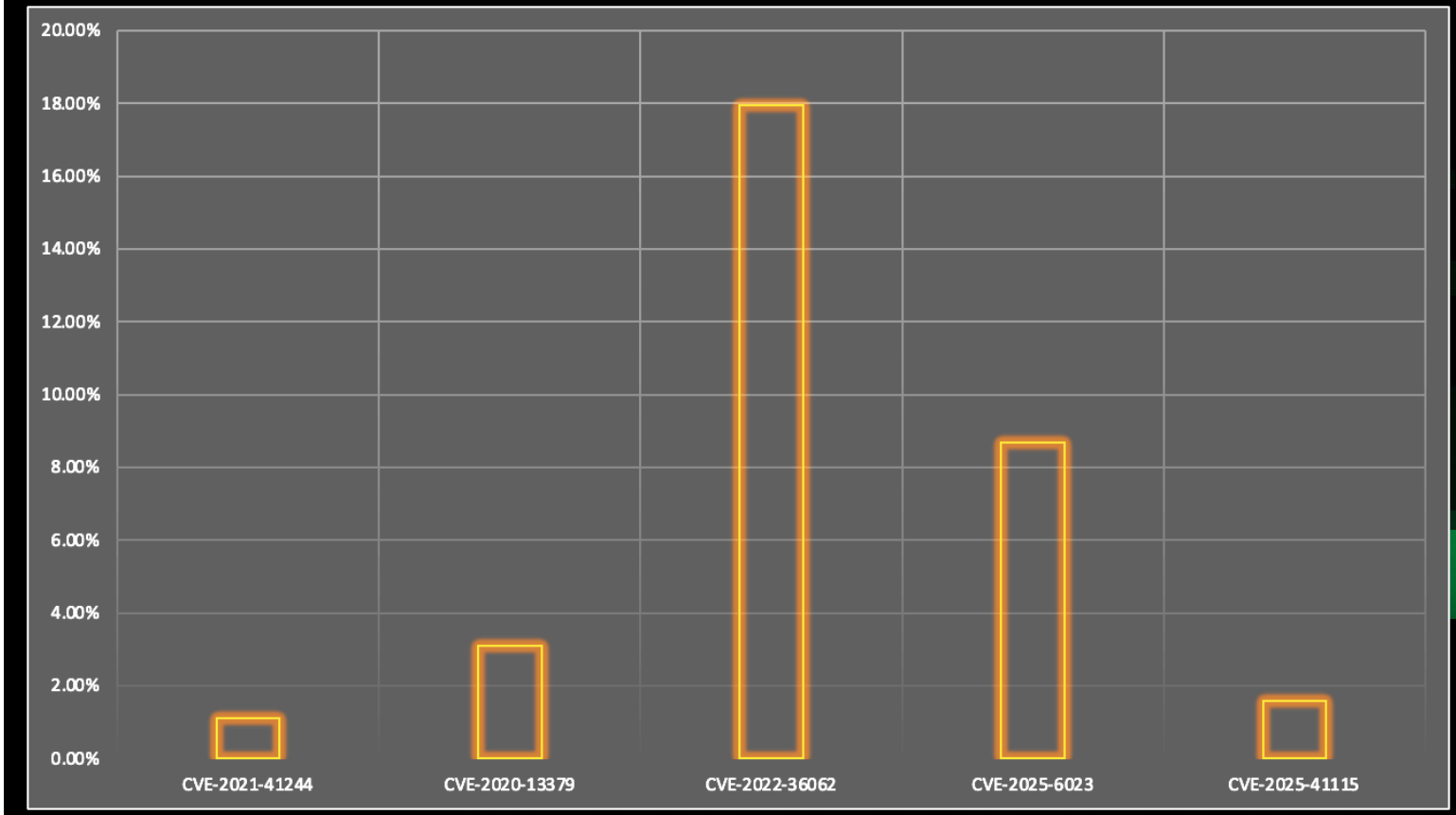
- 0,05% of “403” - hidden Grafana
- 50% - unauth API access

how deep the hole goes

% of misconfigurations



% of risky vulns



Releases

Slides

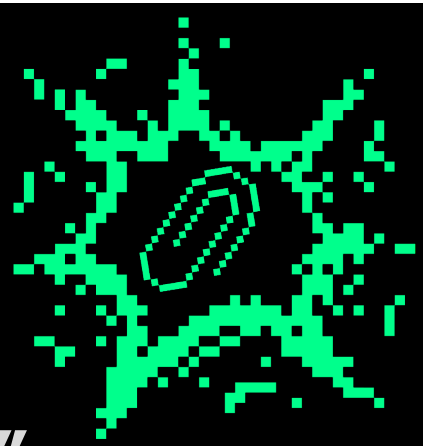
Brute lists

Tools

gmap
grafana_inventory
gproxy

- https://github.com/cyberok-org
- https://github.com/scadastrangelove/

PS



- Monitoring is everywhere
- Insecure by default
- Grafana is both monitoring and “BA for poor”, sometime mixed in one instance
- Point of Trust (in addition to Directory, KSC ☺)
- Stores or can query sensitive data
- Powerful connection and data parsing capabilities
- Cool hacking tool ☺

Red-Team Checklist

1) Reconnaissance and Information Collection:

- **Study users and their roles (Admin, Viewer, Editor).**
- **Analyze datasources: types, URLs, authentication parameters.**
- **Review dashboard configurations for sensitive data (API keys, tokens, secret links).**

2) Scanning and Fingerprinting:

- **Use plugins (e.g., Infinity Data Source Plugin) to create datasources and scan ports.**
- **Analyze service errors and responses to identify application types.**

Red-Team Checklist

3) SSRF Attacks:

- Send arbitrary HTTP(not) requests via Grafana plugins.
- Check the availability of internal APIs (e.g., Kubernetes API, cloud provider metadata).

4) Brute Force and Password Attacks:

- Create new datasources to test authentication on various services.
- Use scripts to automate credential bruteforcing.

Red-Team Checklist

5) (no)SQL injections:

- Modify SQL queries in datasources to execute arbitrary queries.
- Look for datasources with elevated access rights.

7) Using Grafana as a proxy:

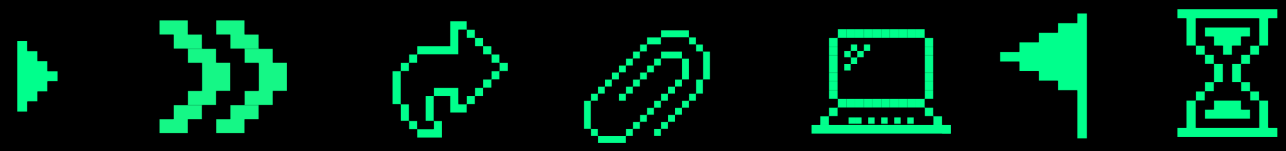
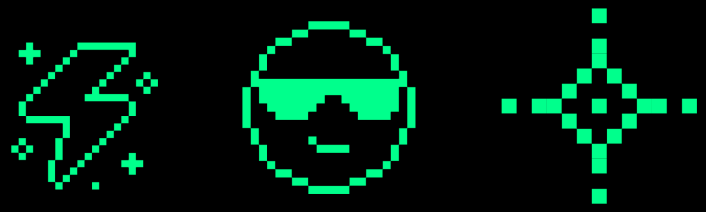
- Use Grafana to redirect requests through Burp Suite or other tools.

8) PWN

Blue-Team Mitigations

- **Private by default:** bind to internal addresses; restrict ports
- **Auth/TLS everywhere:** SSO where possible; mTLS or reverse-proxy with auth; no anonymous access.
- **Secrets hygiene:** no creds in URLs/labels/dashboards; use a vault; rotate tokens; least-priv Datasource creds.
- **Cardinality & flood control:** set label limits, sample limits, rate-limit proxies.
- **Pprof & debug:** never expose debugs to untrusted networks.
- **Monitor** Grafana (with PAG, aha)
- **Pray**

PIXEL MATERIALS



10X YOU!