PROJEKT

Multitenantné operačné centrum kybernetickej bezpečnosti riešené ako otvorená cloudová služba s prvkami strojového učenia

V16 Funkčný PoC

KPB5 – Realizácia riešenia

typ – dokumentácia

mesiac odovzdania – M22

Financované Európskou úniou
NextGenerationEU

PLÁN OBNOVY

ÚRAD VLÁDY
SLOVENSKEJ REPUBLIKY

Obsah

1 Úvod ... 3
1.1 Účel dokumentu ... 3
1.2 Východiská a vstupy do PoC ... 3
1.3 Rozsah PoC ... 3
1.4 Prevádzkové prostredie PoC ... 3
1.5 Obmedzenia PoC ... 4
2 Implementované služby v PoC INNOSOC ... 5
2.1 SIEM - ELK Stack ... 5
2.2 Shuffle ... 17
2.3 Wazuh ... 19
2.4 Greenbone Community edition ... 28
2.5 OpenCTI ... 29
2.6 MISP ... 31
2.7 GLPI ... 31
2.8 NetBox ... 33
2.8.1 Úloha NetBox v PoC ... 33
2.8.2 Nasadenie a základná konfigurácia NetBoxu ... 34
2.8.3 Využitie NetBoxu v rámci PoC ... 35
2.8.4 Aktuálny stav a obmedzenia NetBoxu v PoC ... 39
2.9 DFIR IRIS ... 39
2.10 Kibana (ELK) ... 41
2.11 Wiki.js ... 44
3 Nasadenie a základná konfigurácia PoC ... 47
3.1 Model nasadenia ... 47
3.2 Použité virtuálne stroje a služby ... 47
3.3 Základné integračné väzby ... 47
4 Záver ... 48
4.1 Aa ... **Chyba! Záložka nie je definovaná.**
4.1.1 nadpis ... **Chyba! Záložka nie je definovaná.**

1 ÚVOD

1.1 Účel dokumentu

Účelom tohto dokumentu je zdokumentovať funkčný PoC v rámci pracovného balíka KPB5 – Realizácia riešenia projektu INNOSOC. Dokument nadväzuje na predchádzajúce architektonické a realizačné výstupy projektu a sumarizuje stav, v ktorom je navrhované riešenie implementované ako funkčný experimentálny koncept.

Dokument zachytáva funkčný stav realizovaného riešenia ako integrovaného celku, teda ako multitenantnej cloudovej služby typu SOC, ktorá je nasadená v experimentálnom režime a pripravená na formálne overenie. Jeho hlavným cieľom je preukázať, že navrhované riešenie bolo implementované do funkčnej a prevádzkyschopnej podoby v rozsahu PoC a predstavuje technickú a prevádzkovú dokumentáciu experimentálne vytvoreného riešenia..

1.2 Východiská a vstupy do PoC

PoC vychádza z výsledkov predchádzajúcich pracovných balíkov a z nadväzujúcich výstupov projektu. Základný rámec realizácie bol vytvorený v KPB4 – Architektúra riešenia, ktorého cieľom bolo na základe predchádzajúcich analytických a návrhových výstupov vytvoriť ucelený technologický koncept. V rámci KPB4 boli definované funkčné a nefunkčné požiadavky, biznis architektúra, technologická architektúra a finálny DNR – Detailný návrh riešenia, ktoré predstavujú priamy vstup do implementačnej etapy KPB5.

Za vecné vstupy do realizácie PoC je potrebné považovať aj výsledky predchádzajúcich návrhových a analytických činností projektu, najmä návrh virtualizačného riešenia, návrh služby SOCaaS a návrh integračnej architektúry jednotlivých komponentov. Projektový zámer medzi hlavné úlohy realizácie radí konfiguráciu infraštruktúry, inštaláciu softvérových aplikácií, ich integráciu a testovanie navrhovaného riešenia. PoC vzniká ako výsledok postupnej nadväznosti na predchádzajúce architektonické, technologické a infraštruktúrne výstupy projektu.

1.3 Rozsah PoC

Tu opísaný PoC zachytáva implementované časti navrhovaného riešenia v rozsahu potrebnom na overenie základnej funkčnosti služby. Predmetom tohto výstupu je najmä opis nasadených komponentov, ich základných väzieb a prevádzkového usporiadania v prostredí projektu. Dokument sa sústreďuje na to, čo bolo v rámci KPB5 reálne nasadené, nakonfigurované a overené. PoC nepredstavuje finálne produkčné riešenie ani kompletnú komerčnú službu. Ide o experimentálny koncept, ktorého účelom je preukázať technickú realizovateľnosť navrhovanej architektúry a základnú prevádzkyschopnosť riešenia ako celku. Rozsah PoC je preto vedome obmedzený na funkcie, ktoré sa podarilo implementovať a overiť v laboratórnych podmienkach.

1.4 Prevádzkové prostredie PoC

PoC je prevádzkovaný nad privátnym cloudovým prostredím OpenStack, ktoré tvorí základnú infraštruktúrnu vrstvu riešenia. Jednotlivé komponenty sú nasadené vo virtuálnych strojoch, pričom podľa charakteru konkrétneho nástroja je využitá aj kontajnerizácia prostredníctvom Docker a Docker Compose. Tento model umožňuje oddeliť jednotlivé funkčné bloky riešenia, zjednodušiť ich správu a zároveň zachovať flexibilitu pri experimentálnom

nasadzovaní. Prevádzkové prostredie PoC má charakter testovacieho a overovacieho prostredia. Jeho účelom nie je poskytovanie plnohodnotnej produkčnej služby, ale vytvorenie funkčnej platformy na integráciu, konfiguráciu a overenie navrhovaných komponentov a ich vzájomnej spolupráce. Prostredie je preto navrhnuté tak, aby umožňovalo opakované nasadenie, úpravy konfigurácie a realizáciu testovacích scenárov.

1.5 Obmedzenia PoC

PoC je poskytuje experimentálny dôkaz koncepcie. Nie všetky plánované komponenty a integračné väzby boli v čase spracovania tohto výstupu dotiahnuté do finálnej úrovne funkčnosti. V niektorých prípadoch boli komponenty úspešne nasadené samostatne, avšak ich integrácia zostala len čiastočne realizovaná. Dokument tak zachytáva reálny stav implementácie, nie finálny stav architektúry. Obmedzenia sa týkajú najmä miery dokončenia jednotlivých integrácií, rozsahu overených scenárov a celkovej úrovne prevádzkovej zrelosti riešenia. PoC však poskytuje dostatočný základ na technické overenie konceptu a na následné vyhodnotenie dosiahnutých výsledkov v ďalšom výstupe projektu.

.

BRAIN:IT

FRI

2 IMPLEMENTOVANÉ KOMPONENTY V POC INNOSOC

2.1 SIEM - ELK Stack

2.1.1 Model nasadenia ELK Stack

ELK je v rámci PoC riešenia INNOSOC využitý ako SIEM, ktorý sa skladá z Elasticsearch indexovej databázy, Logstash parsovacieho nástroja, ktorý predpripravuje formát dát, pred uložením do databázy, Kibany, ktorá slúži ako WEB GUI pre celý stack. Kibana zobrazuje nazbierané dáta, poskytuje správu používateľov a rolí, tvorbu bezpečnostných pravidiel, riešenie alertov, správu indexov, správu uzlov (nodes) a iné.

Celý ELK stack je nasadený v LXC linuxových kontajneroch. Pre každú službu je vytvorený samostatný kontajner. Operačný systém nasadený v kontajneroch je Ubuntu 24.04.2.

Všetky LXC kontajnery sú nasadené na jednom fyzickom serveri. Server má procesor Intel(R) Xeon(R) CPU E5-2690, 2 fyzické procesory, 8 jadier na procesor s frekvenciou 2.90GHz. Server disponuje s 251GB RAM pamäte. Úložný priestor je tvorený raidom HDD diskov o veľkosti 2.7T a SSD diskom 1.8T WD Red.

2.1.2 Nasadenie Elasticsearch

Elasticsearch kontajner má IP adresu 192.168.1.178. Beží na ňom Elasticsearch cluster, ktorý je tvorený 7 elasticsearch uzlami. Uzly majú role master, hot, warm a cold. Master node je hlavný node, ostatné data nodes sa naň pripájajú a začleňujú sa do clustera. Hot nodes sú uzly, kde sa uchovávajú najaktuálnejšie dáta. Dáta sa ukladajú na SSD disk, aby sa stíhali zapisovať v čo najkratšom časovom intervale. Tieto dáta sa tam uchovávajú po dobu 5 dní. Po tejto dobe sa presúvajú na warm nodes. Warm nodes sú 2 a používajú HDD disky. Dáta sú uložené vo warm nodes po dobu od 5 do 14 dní. Následne sa dáta presúvajú do cold uzla a po 21 dňoch sa mažú.

Node	Disky	Retenčná doba	Pridelená RAM
master1	HDD	---	16GB
hot1	SSD	<5 dní	31GB
hot2	SSD	<5 dní	31GB
hot3	SSD	<5 dní	31GB
warm1	HDD	<14 dní	31GB
warm2	HDD	<14 dní	31GB
cold1	HDD	<21 dní	16GB

V projekte je kladený dôraz na zabezpečenie. Elasticsearch predvolene používa, od verzie 7, x-pack modul, ktorý zabezpečuje viacero bezpečnostných funkcií, vrátane autentifikácie, autorizácie, šifrovania komunikácie a správy používateľov.

Komunikácie je medzi uzlami šifrovaná pomocou TLS certifikátov. Každý node má svoj TLS certifikát. Komunikácie medzi Logstash a Elastich search aj Kibana a Elasticsearch sú taktiež šifrované. Každý Elasticsearch uzol má vlastný konfiguračný súbor.

Použitá certifikačná autorita (CA), ktorá vystavuje jednotlivé certifikáty, je vlastná. Elastic poskytuje možnosť vygenerovať CA pomocou nástroja *elasticsearch-certutil*, ktorý je súčasťou distribúcie *Elasticsearch*. Tento nástroj umožňuje vytvárať certifikát certifikačnej autority, ale aj vystavovať ďalšie certifikáty podpísané touto certifikačnou autoritou.

Elasticsearch okrem toho využíva tzv. keystore. Keystore slúži ako úložisko pre citlivé údaje, ako sú prihlasovacie údaje, tokeny alebo heslá systémových používateľov. V našom prípade využívame keystore na uloženie prihlasovacích údajov preddefinovaných systémových účtov. Tento mechanizmus zvyšuje bezpečnosť systému tým, že citlivé údaje nie sú priamo uložené v konfiguračných súboroch.

Okrem zabezpečenia citlivých údajov pomocou keystore je potrebné zabezpečiť aj sieťovú komunikáciu samotného *elk-elasticsearch* kontajneru. Na tento účel sme využili nástroj iptables. Pomocou neho sme definovali firewall pravidlá. Prístup pre službu Secure Shell (SSH) je povolený z univerzitného rozsahu a prístup pre *Elasticsearch* aplikáciu je povolený len zo strojov *Kibana*, *Logstash* a *Arkime*, definované na základe IP adries.

V rámci monitorovania dostupnosti bolo okrem, už spomínaného, monitorovania ELK stack pomocou metricbeat zavedené monitorovanie pomocou nástroja libreNMS a pomocou nástroja zabbix.

2.1.2.1 Ukážka konfiguračného súboru hot uzla

```
cluster.name: elastic-cluster
node.name: hot1
node.roles: [data_hot]
path.data: /mnt/hot-nodes/elasticsearch-hot1
path.logs: /var/log/elasticsearch-hot1
network.host: 192.168.1.178
http.port: 9201
transport.port: 9301
discovery.seed_hosts:  ["192.168.1.178:9300",  "192.168.1.178:9302",
"192.168.1.178:9303",  "192.168.1.178:9304",  "192.168.1.178:9305",
"192.168.1.178:9306"]
cluster.initial_master_nodes: ["master1"]
xpack.security.enabled: true
xpack.security.enrollment.enabled: true
xpack.security.http.ssl:
  enabled: true
  keystore.path: certs/elasticsearch/hot1/http.p12
xpack.security.transport.ssl:
  enabled: true
  verification_mode: certificate
  client_authentication: required
  keystore.path: certs/elastic-certificates.p12
  truststore.path: certs/elastic-certificates.p12
http.host: 0.0.0.0
transport.host: 0.0.0.0
xpack.security.audit.enabled: true
xpack.security.audit.logfile.events.include:          access_denied,
anonymous_access_denied,  authentication_failed,  connection_denied,
tampered_request, run_as_denied, security_config_change
xpack.security.audit.logfile.emit_node_host_address: true
```

```
xpack.security.audit.logfile.emit_node_name: true
xpack.security.audit.logfile.emit_node_host_name: true
```

2.1.3 Nasadenie Logstash

Logstash spracováva dáta prijaté od auditbeat a filebeat agentov, ktorí sú nasadení na ostatných serveroch na FRI UNIZA. Komunikácia medzi agentami a Logstash je taktiež šifrovaná. Logstash pipeline je rozdelená do troch konfiguračných súboroch. Pre vstup, filtrovanie a výstup zozbieraných dát.

Pre vstup, filtrovanie a výstup zozbieraných dát:

- input.conf – definuje zdroje prijatých dát,
- filters.conf – spracováva prijaté dáta, aplikuje na dáta filtre, rieši normalizáciu dát
- output.conf – spravuje odoslanie dát.

Obdobne ako pri nástroji Elasticsearch aj pre nástroj Logstash sme vykonali čistú inštaláciu. Na stroji ucs-lxc-1 s IP adresou 192.168.2.11 bol pre potreby nasadenia Logstash nástroja vytvorený LXC kontajner s operačným systémom Ubuntu 24.04 LTS. Kontajner je pomenovaný elk-logstash s pridelenou IP adresu 192.168.1.177.

Logstash je nasadený vo verzii 8.17.2. Pri jeho konfigurácii je definovaná veľkosť operačnej pamäte, obdobne ako pri Elasticsearch uzloch, pričom je alokovaných 16 GB. Táto hodnota poskytuje dostatočnú rezervu aj pre budúce rozšírenie infraštruktúry a pripojenie ďalších monitorovaných zariadení.

Komunikácia Logstash a Elasticsearch je zabezpečená certifikátom na strane Elasticsearch. Overenie Logstash voči Elasticsearch prebieha pomocou mena a hesla. Keďže sme sa snažili dodržať princíp najnižších potrebných oprávnení tzv. „least privilege“, bolo potrebné vytvoriť používateľa ktorému bola pridelená rola s oprávneniami. Vytvorená rola nesie názov logstash_writer a vytvorený používateľ logstash_internal. Danej roli sme pridelili nasledujúce oprávnenia pre cluster:

- monitor - sledovanie stavu klastra a uzlov
- manage_index_template - správa šablón indexov
- manage_ilm - správa životného cyklu indexov
- manage_pipeline - správa ingest pipeline

A oprávnenia pre jednotlivé indexy:

- write - zapisovanie dát do indexu
- create - vytváranie nových dokumentov
- create_index - vytváranie nových indexov
- manage - plná správa indexu
- read - čítanie dát z indexu
- index - pridávanie alebo aktualizácia dokumentov

Indexy pre ktoré sa majú privilégia prideliť sme definovali pomocou tzv. index pattern. Index pattern je vzor alebo pravidlo, ktoré definuje konkrétny index. Využívajú sa v nich tzv. wildcards, v podobe znaku „ * “, spolu v kombinácií textom, ktorý tvorí časť názvu indexu. Pre našu rolu sme definovali nasledujúce index pattern:

1. gvm-greenbone*
2. elasticsearch-elk-elasticsearch*
3. syslog-fortigate*
4. waf-filebeat*

5. auditbeat*
6. suricata*
7. threatintel-filebeat*

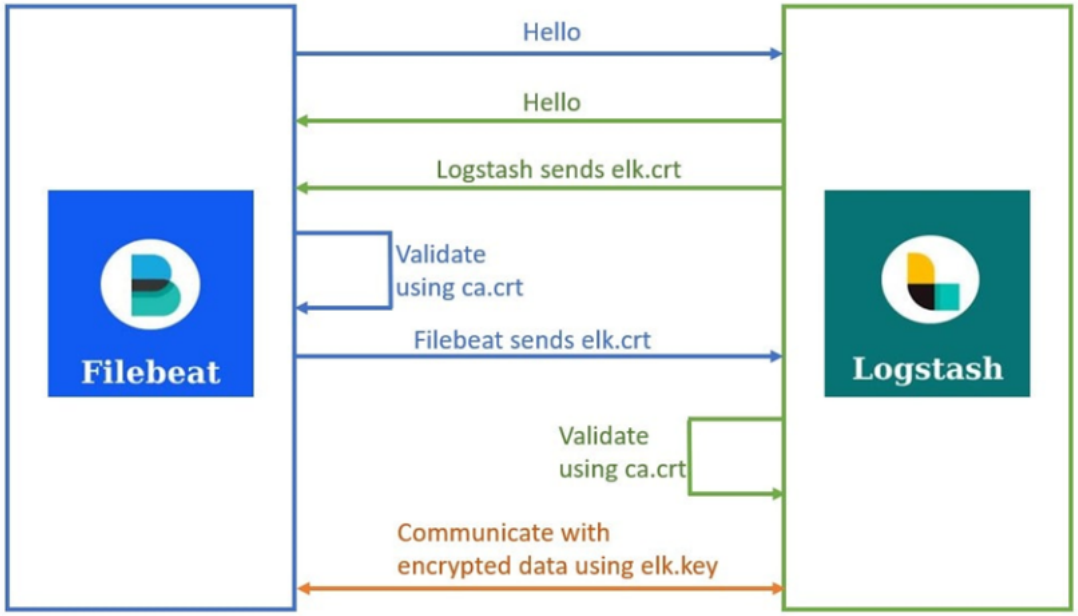
Logstash využíva keystore na uloženie hesla používateľa pre Logstash API a na uloženie hesla pre používateľa logstash_internal. Sieťová komunikácia je zabezpečená pomocou nástroja iptables, ktorým sme definovali firewall pravidlá. Prístup pre službu SSH a pre príjem dát pomocou Logstash je povolený zo špecifického IPv4 rozsahu.

Logstash je v rámci monitorovania dostupnosti monitorovaný pomocou metricbeat a taktiež pomocou nástrojov zabbix a libreNMS.

2.1.3.1 Ukážka input.conf konfiguračného súboru pre nasadený Logstash

V konfiguračnom súbore je nastavený príjem dát pomocou modulu beat, ktorý dáta prijíma na porte 5044. Ďalej je nastavené Secure Sockets Layer (SSL)/ Transport Layer Security (TLS) šifrovanie a definované cesty k certifikátom. Certifikáty boli podpísané vlastnou certifikačnou autoritou. Logstash vyžaduje aby sa klienti preukázali platným certifikátom, inak spojenie nebude nadviazané, a taktiež poskytuje vlastný TLS certifikát ktorým si klienti overia identitu Logstash. Ide o obojstranné TLS overovanie.

```
input {
       beats {
              port => 5044
              ssl_enabled => true
              ssl_certificate                                    =>
            "/etc/logstash/certs/logstash/logstash.crt"
              ssl_key => "/etc/logstash/certs/logstash/logstash.key"
              ssl_certificate_authorities                        =>
            ["/etc/logstash/certs/ca.crt"]
              ssl_client_authentication => "required"
       }
}
```



Znázornenie obojstranného TLS

2.1.3.2 Ukážka filters.conf konfiguračného súboru pre nasadený Logstash

```
filter { if ("suricata-moloch" in [tags]) or ("suricata-suricata2"
in [tags]) { json { source => "[event][original]" target =>
"suricata_parsed" } date { match => ["[suricata_parsed][timestamp]",
"yyyy-MM-dd'T'HH:mm:ss.SSSSSSZ"] target => "@timestamp" } mutate {
rename => { "[suricata_parsed][src_ip]" => "source.ip"
"[suricata_parsed][src_port]" => "source.port"
"[suricata_parsed][dest_ip]" => "destination.ip"
"[suricata_parsed][dest_port]" => "destination.port"
"[suricata_parsed][proto]" => "network.protocol"
"[suricata_parsed][event_type]" => "event.type"
"[suricata_parsed][flow_id]" => "network.flow_id"
"[suricata_parsed][app_proto]" => "network.application_protocol"
"[suricata_parsed][in_iface]" => "network.interface"
"[suricata_parsed][alert][signature]" => "rule.name"
"[suricata_parsed][alert][signature_id]" => "rule.id"
"[suricata_parsed][alert][category]" => "rule.category"
"[suricata_parsed][alert][severity]" => "rule.severity"
"[suricata_parsed][alert][action]" => "rule.action"
"[suricata_parsed][alert][rev]" => "rule.revision"
"[suricata_parsed][alert][gid]" => "rule.gid"
"[suricata_parsed][flow][pkts_toserver]" =>
"network.packets.to_server" "[suricata_parsed][flow][pkts_toclient]"
=> "network.packets.to_client"
"[suricata_parsed][flow][bytes_toserver]" =>
"network.bytes.to_server" "[suricata_parsed][flow][bytes_toclient]"
=> "network.bytes.to_client" "[suricata_parsed][flow][start]" =>
"network.flow_start"
"[suricata_parsed][alert][metadata][signature_severity]" =>
```

```
"rule.signature_severity"
"[suricata_parsed][alert][metadata][created_at]" =>
"rule.created_at" "[suricata_parsed][alert][metadata][updated_at]"
=> "rule.updated_at" "[suricata_parsed][alert][metadata][cve]" =>
"rule.cve_list" "[suricata_parsed][alert][metadata][deployment]" =>
"rule.deployment"
"[suricata_parsed][alert][metadata][attack_target]" =>
"rule.attack_target"
"[suricata_parsed][alert][metadata][affected_product]" =>
"rule.affected_product" "[suricata_parsed][alert][metadata][tag]" =>
"rule.tag" } convert => { "network.flow_id" => "integer"
"source.port" => "integer" "destination.port" => "integer"
"network.packets.to_server" => "integer" "network.packets.to_client"
=> "integer" "network.bytes.to_server" => "integer"
"network.bytes.to_client" => "integer" "rule.id" => "integer"
"rule.severity" => "integer" "rule.revision" => "integer" "rule.gid"
=> "integer" } } geoip { source => "source.ip" target => "source"
database => "/etc/logstash/geoip/GeoLite2-City.mmdb" } geoip {
source => "destination.ip" target => "destination" database =>
"/etc/logstash/geoip/GeoLite2-City.mmdb" } mutate { remove_field =>
[ "source.ip", "destination.ip"] rename => {
"[suricata_parsed][src_ip]" => "source.ip"
"[suricata_parsed][dest_ip]" => "destination.ip" } } mutate {
remove_field => [ "message", "suricata_parsed"] } }

else if ("auditbeat-studenti" in [tags]) or ("auditbeat-admin" in
[tags]) or ("auditbeat-testing" in [tags]) or ("auditbeat-elk" in
[tags]){ mutate { add_field => { "[@metadata][index]" => "auditbeat-
logs" } add_field => { "event.ingested" => "%{+YYYY-MM-
dd'T'HH:mm:ss.SSSZ}" } rename => {
"[process][parent][process][name]" => "process.parent.name"
"[process][parent][process][executable]" =>
"process.parent.executable" "[process][parent][process][args]" =>
"process.parent.args" "[process][parent][process][pid]" =>
"process.parent.pid" "[process][parent][process][entity_id]" =>
"process.parent.entity_id" "[process][parent][process][title]" =>
"process.parent.title" "[process][parent][process][start_time]" =>
"process.parent.start_time" "[process][parent][process][owner]" =>
"process.parent.owner" "[process][parent][process][group]" =>
"process.parent.group" "[process][parent][process][thread]" =>
"process.parent.thread" } } mutate { replace => {
"process.parent.pid" => "%{[process][parent][pid][0]}" } }

mutate {
  remove_field => ["process.parent.pid"]
}
mutate {
  convert => { "process.parent.pid" => "integer" }
}
if [process][args] and ( [process][args] != "" ) {

  mutate {
```

```
add_field => { "[process][command_line]" => "%{[process][args]}" }
  }
  mutate {
    gsub => [
      "[process][command_line]", '\"', '',
      "[process][command_line]", '\[', '',
      "[process][command_line]", '\]', '',
      "[process][command_line]", ',', ' '
    ]
  }
}


} else if "syslog-fortigate" in [log_type] { grok { match => {
"message" => "%{SYSLOGTIMESTAMP:syslog_timestamp}
%{IPORHOST:syslog_host} %{GREEDYDATA:event.original}" } overwrite =>
["event.original"] } kv { source => "event.original" field_split =>
" " value_split => "=" trim_key => """ trim_value => """ } mutate {
rename => { "srcip" => "source.ip" "dstip" => "destination.ip" }
convert => { "srcport" => "integer" "dstport" => "integer"
"sentbyte" => "integer" "rcvdbyte" => "integer" "sentpkt" =>
"integer" "rcvdpkt" => "integer" "duration" => "integer" } add_field
=> { "datetime" => "%{date} %{time}" } } date { match =>
["datetime", "yyyy-MM-dd HH:mm:ss"] target => "@timestamp" timezone
=> "+0100" } mutate { remove_field => ["datetime", "message",
"date"] } if [agent] { mutate { add_field => { "agent_name" =>
"%{[agent][name]}" "agent_id" => "%{[agent][id]}" "agent_type" =>
"%{[agent][type]}" "agent_version" => "%{[agent][version]}"
"agent_ephemeral_id" => "%{[agent][ephemeral_id]}" } } mutate {
remove_field => ["agent"] }

 } } else if "CTF-win" in [tags]{ mutate { add_field => {
"[@metadata][index]" => "ctf-win-logs" } add_field => {
"event.ingested" => "%{+YYYY-MM-dd'T'HH:mm:ss.SSSZ}" } } if
[winlog][provider_name] == "Microsoft-Windows-Sysmon" { mutate {
rename => { "[winlog][event_id]" => "event.sysmon_id"
"[winlog][event_data][Image]" => "process.name"
"[winlog][event_data][CommandLine]" => "process.command_line"
"[winlog][event_data][ParentImage]" => "process.parent.name"
"[winlog][event_data][ParentCommandLine]" =>
"process.parent.command_line" "[winlog][event_data][User]" =>
"user.name" "[winlog][event_data][Hashes]" => "file.hashes"
"[winlog][event_data][TargetFilename]" => "file.target_name"
"[winlog][event_data][CreationUtcTime]" => "file.creation_time"
"[winlog][event_data][ProcessId]" => "process.id"
"[winlog][event_data][ParentProcessId]" => "process.parent.id" } }
if [file][hashes] { ruby { code => ' hashes =
event.get("[file][hashes]") if hashes hashes.split(",").each do
|hash| key_value = hash.split("=") if key_value.length == 2 end end
end ' } } } if [winlog][provider_name] == "PowerShell" { mutate {
rename => { "[winlog][event_id]" => "event.powershell_id"
"[winlog][event_data][param1]" => "powershell.command"
```

```
"[winlog][event_data][param2]" => "powershell.details"
"[winlog][event_data][param3]" => "powershell.output" } } } if
[host][name] { mutate { rename => { "[host][name]" =>
"system.hostname" "[host][os][name]" => "system.os_name"
"[host][os][version]" => "system.os_version" } } } if
[agent][version] { mutate { rename => { "[agent][version]" =>
"logstash.agent_version" } } } mutate { remove_field => [
"[winlog][provider_guid]", "[winlog][record_id]",
"[winlog][opcode]", "[winlog][api]" ]

 } } else if "filebeat-gvm" in [tags] {

if [message] =~ "^\{.*\}$" {
  json {
    source => "message"
  }
} else {
  grok {
    match => { "message" => "%{IPORHOST:client_ip} - -
\[%{HTTPDATE:log_timestamp}\] \"%{WORD:http_method}
%{URIPATHPARAM:request} HTTP/%{NUMBER:http_version}\"
%{NUMBER:response_code} (?:%{NUMBER:bytes_sent}|-)
\"%{DATA:referrer}\" \"%{DATA:user_agent}\"" }
  }
}
date {
  match => [ "log_timestamp", "dd/MMM/yyyy:HH:mm:ss Z" ]
  target => "@timestamp"
}
mutate {
  remove_field => ["log_timestamp"]
}


} else if "filebeat-elasticsearch" in [tags] { json { source =>
"message" target => "audit" skip_on_invalid_json => true } mutate {
add_field => { "user" => "%{[audit][user.name]}" "action" =>
"%{[audit][event.action]}" "timestamp" => "%{[audit][timestamp]}" }
} date { match => ["[audit][@timestamp]", "yyyy-MM-
dd'T'HH:mm:ss,SSSZ"] target => "@timestamp" } } else if "filebeat-
WAF" in [tags] { if [log][file][path] == "/var/log/nginx/access.log"
{ mutate { add_tag => [ "nginx-access" ] } grok { match => {
"message" => '%{IPORHOST:remote_ip} - %{DATA:remote_user}
[%{HTTPDATE:timestamp}] "(?:%{WORD:method}
%{URIPATHPARAM:request}(?: HTTP/%{NUMBER:http_version})?)"
%{NUMBER:status} (?:%{NUMBER:bytes}|-) "(?:%{URI:referrer}|-)"
"(?:%{DATA:user_agent}|-)"' } remove_field => [ "event.original" ] }
date { match => [ "timestamp", "dd/MMM/yyyy:HH:mm:ss Z" ] target =>
"@timestamp" } } else if [log][file][path] ==
"/var/log/nginx/error.log" { mutate { add_tag => [ "nginx-error" ] }
grok { match => { "message" => '[%{DAY} %{MONTH} %{MONTHDAY} %{TIME}
%{YEAR}] [%{LOGLEVEL:log_level}]( [%{DATA:module}])?
```

```
%{GREEDYDATA:error_message}' } remove_field => [ "event.original" ]
} date { match => [ "timestamp", "dd/MMM/yyyy:HH:mm:ss Z" ] target
=> "@timestamp" } } else if [log][file][path] ==
"/var/log/modsec/kibana.fri.uniza.sk_audit.log" { mutate { add_tag
=> [ "nginx-audit" ] } grok { match => { "message" => "(?m)---
%{DATA:modsec_id}---A--\[%{HTTPDATE:timestamp}\] %{NUMBER:unique_id}
%{IP:src_ip} %{NUMBER:src_port} %{IP:dst_ip} %{NUMBER:dst_port}.*?--
-%{DATA:modsec_id}---B--%{GREEDYDATA:http_request}---
%{DATA:modsec_id}---F--%{GREEDYDATA:http_response}---
%{DATA:modsec_id}---H--%{GREEDYDATA:modsec_alerts}" } tag_on_failure
=> [ "_modsec_grok_fail" ] } date { match => [ "timestamp",
"dd/MMM/yyyy:HH:mm:ss Z" ] target => "@timestamp" } mutate {
remove_field => [ "event.original" ] } } } else if "filebeat-
threatintel" in [tags] { date { match => ["[audit][@timestamp]",
"yyyy-MM-dd'T'HH:mm:ss,SSSZ"] target => "@timestamp" } if
[fileset][name] == "abuseurl" { json { source => "message" target =>
"abuseurl" } mutate { add_field => { "threat_type" =>
"%{[abuseurl][threat]}" "url" => "%{[abuseurl][url]}" "ip" =>
"%{[abuseurl][host]}" "reporter" => "%{[abuseurl][reporter]}"
"malware_tag" => "%{[abuseurl][tags][0]}" "urlhaus_ref" =>
"%{[abuseurl][urlhaus_reference]}" } } } else if [fileset][name] ==
"abusemalware" { json { source => "message" target => "abusemalware"
} mutate { add_field => { "sha256" =>
"%{[abusemalware][sha256_hash]}" "md5" =>
"%{[abusemalware][md5_hash]}" "signature" =>
"%{[abusemalware][signature]}" "vt_result" =>
"%{[abusemalware][virustotal][result]}" "vt_percent" =>
"%{[abusemalware][virustotal][percent]}" "download_url" =>
"%{[abusemalware][urlhaus_download]}" } } } else if [fileset][name]
== "otx" { json { source => "message" target => "otx" } mutate {
add_field => { "indicator" => "%{[otx][indicator]}" "indicator_type"
=> "%{[otx][type]}" "description" => "%{[otx][description]}" } } }

 } else if "arkimecapture-moloch" in [tags] { grok { match => {
"message" => "%{MONTH:log_month} %{MONTHDAY:log_day}
%{TIME:log_time} %{GREEDYDATA:rest_of_message}" } } mutate {
add_field => { "log_timestamp" => "%{log_month} %{log_day}
%{log_time}" } } date { match => [ "log_timestamp", "MMM dd
HH:mm:ss" ] target => "@timestamp" timezone => "Europe/Bratislava" }
grok { match => { "message" => "drop: %{NUMBER:drop_count:int}
(%{NUMBER:drop_percent:float})" } } grok { match => { "message" =>
"recv: %{NUMBER:recv_count:int}" } } } }
```

2.1.3.3 Ukážka konfiguračného súboru output.conf pre nasadený Logstash

```
output {
  if "auditbeat-studenti" in [tags] {
    elasticsearch {
      hosts => ["https://192.168.1.178:9200"]
      user => "logstash_internal"
      password => "${ES_psswd}"
      ssl_enabled => true
      ssl_certificate_authorities => "/etc/logstash/certs/ca.crt"
```

```
    ilm_enabled => true
    ilm_rollover_alias => "auditbeat-logs-studenti"
    ilm_pattern => "{now/d}-000001"
    ilm_policy => "auditbeat-logs-ILM"
    manage_template => false
  }
}
else if "suricata-moloch" in [tags] {
  elasticsearch {
    hosts => ["https://192.168.1.178:9200"]
    user => "logstash_internal"
    password => "${ES_psswd}"
    ssl_enabled => true
    ssl_certificate_authorities => "/etc/logstash/certs/ca.crt"
    ilm_enabled => true
    ilm_rollover_alias => "suricata-moloch"
    ilm_pattern => "{now/d}-000001"
    ilm_policy => "suricata-moloch-ILM"
    manage_template => false
  }
}
else if "syslog-fortigate" in [log_type] {
  elasticsearch {
    hosts => ["https://192.168.1.178:9200"]
    user => "logstash_internal"
    password => "${ES_psswd}"
    ssl_enabled => true
    ssl_certificate_authorities => "/etc/logstash/certs/ca.crt"
    ilm_enabled => true
    ilm_rollover_alias => "syslog-fortigate"
    ilm_pattern => "{now/d}-000001"
    ilm_policy => "syslog-fortigate-ILM"
    manage_template => false
  }
}
else if "auditbeat-admin" in [tags] {
  elasticsearch {
    hosts => ["https://192.168.1.178:9200"]
    user => "logstash_internal"
    password => "${ES_psswd}"
    ssl_enabled => true
    ssl_certificate_authorities => "/etc/logstash/certs/ca.crt"
    ilm_enabled => true
    ilm_rollover_alias => "auditbeat-logs-admin"
    ilm_pattern => "{now/d}-000001"
    ilm_policy => "auditbeat-logs-ILM"
    manage_template => false
  }
}
else if "suricata-suricata2" in [tags] {
  elasticsearch {
    hosts => ["https://192.168.1.178:9200"]
```

```
    user => "logstash_internal"
    password => "${ES_psswd}"
    ssl_enabled => true
    ssl_certificate_authorities => "/etc/logstash/certs/ca.crt"
    ilm_enabled => true
    ilm_rollover_alias => "suricata-suricata2"
    ilm_pattern => "{now/d}-000001"
    ilm_policy => "suricata-suricata2-ILM"
    manage_template => false
  }
}
else if "auditbeat-testing" in [tags] {
  elasticsearch {
    hosts => ["https://192.168.1.178:9200"]
    user => "logstash_internal"
    password => "${ES_psswd}"
    ssl_enabled => true
    ssl_certificate_authorities => "/etc/logstash/certs/ca.crt"
    index => "auditbeat-testing"
    manage_template => false
  }
}
else if "filebeat-gvm" in [tags] {
  elasticsearch {
    hosts => ["https://192.168.1.178:9200"]
    user => "logstash_internal"
    password => "${ES_psswd}"
    ssl_enabled => true
    ssl_certificate_authorities => "/etc/logstash/certs/ca.crt"
    ilm_enabled => true
    ilm_rollover_alias => "gvm-greenbone"
    ilm_pattern => "{now/d}-000001"
    ilm_policy => "gvm-greenbone-ILM"
    manage_template => false
  }
}
else if "auditbeat-elk" in [tags] {
  elasticsearch {
    hosts => ["https://192.168.1.178:9200"]
    user => "logstash_internal"
    password => "${ES_psswd}"
    ssl_enabled => true
    ssl_certificate_authorities => "/etc/logstash/certs/ca.crt"
    ilm_enabled => true
    ilm_rollover_alias => "auditbeat-logs-elk"
    ilm_pattern => "{now/d}-000001"
    ilm_policy => "auditbeat-logs-ILM"
    manage_template => false
  }
}
else if "filebeat-elasticsearch" in [tags] {
  elasticsearch {
```

```
    hosts => ["https://192.168.1.178:9200"]
    user => "logstash_internal"
    password => "${ES_psswd}"
    ssl_enabled => true
    ssl_certificate_authorities => "/etc/logstash/certs/ca.crt"
    ilm_enabled => true
    ilm_rollover_alias => "elasticsearch-elk-elasticsearch"
    ilm_pattern => "{now/d}-000001"
    ilm_policy => "elasticsearch-elk-elasticsearch-ILM"
    manage_template => false
  }
}
else if "filebeat-WAF" in [tags] {
  elasticsearch {
    hosts => ["https://192.168.1.178:9200"]
    user => "logstash_internal"
    password => "${ES_psswd}"
    ssl_enabled => true
    ssl_certificate_authorities => "/etc/logstash/certs/ca.crt"
    ilm_enabled => true
    ilm_rollover_alias => "waf-filebeat"
    ilm_pattern => "{now/d}-000001"
    ilm_policy => "waf-filebeat-ILM"
    manage_template => false
  }
}
else if "filebeat-threatintel" in [tags] {
  elasticsearch {
    hosts => ["https://192.168.1.178:9200"]
    user => "logstash_internal"
    password => "${ES_psswd}"
    ssl_enabled => true
    ssl_certificate_authorities => "/etc/logstash/certs/ca.crt"
    ilm_enabled => true
    ilm_rollover_alias => "threatintel-filebeat"
    ilm_pattern => "{now/d}-000001"
    ilm_policy => "threatintel-filebeat-ILM"
    manage_template => false
  }
}
else if "arkimecapture-moloch" in [tags] {
  elasticsearch {
    hosts => ["https://192.168.1.178:9200"]
    user => "logstash_internal"
    password => "${ES_psswd}"
    ssl_enabled => true
    ssl_certificate_authorities => "/etc/logstash/certs/ca.crt"
    ilm_enabled => true
    ilm_rollover_alias => "arkimecapture-moloch"
    ilm_pattern => "{now/d}-000001"
    ilm_policy => "arkimecapture-moloch-ILM"
    manage_template => false
```

```
        }
    }
}
```

2.2 Shuffle

Shuffle Automation je v rámci PoC riešenia INNOSOC využitý pre proces automatizovaného vykonávania procesov a transformácie dát medzi rôznymi bezpečnostnými nástrojmi a systémami. V našom prípade ide o orchestráciu dátových tokov, ktorá zabezpečuje, že údaje z jedného systému sú automaticky spracované, transformované a presmerované do ďalších systémov bez ľudskej intervencie.

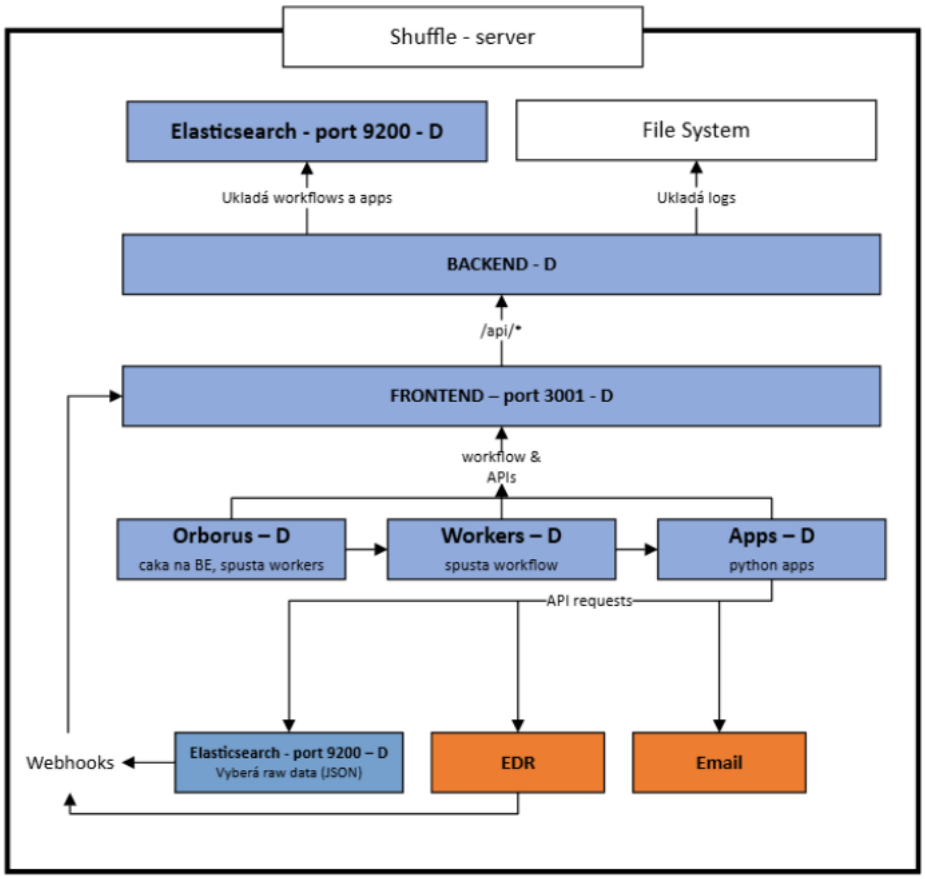
2.2.1 Model nasadenia Shuffle

Riešenie beží v Docker kontajneroch, pričom backend je postavený na OpenSearch, ktorý slúži ako úložisko pre workflows a aplikácie v pythone. Webové rozhranie je prístupné na porte 3001.

Techncká špecifikácia nasadenia Shuffle:

- Prostredie – OpenStack
- Meno inštancie – shuffe
- IP adresa – 192.168.4.188
- vCPU – 4
- RAM – 16 GB
- Disk – 80 GB
- OS – Ubuntu 24.04

Softvér Shuffle je založený na dockeri. To znamená, že k tomu aby vedel plne fungovať musí mať systém, na ktorom bude nasadený na najaktuálnejšiu verziu dockera. Spolu s inštaláciou Shuffle sú 2 súbory, docker-compose a súbor .env. Oba súbory slúžia na konfiguráciu portov, port forwardingu, backendu a frontendu. K tomu aby Shuffle vedel plne fungovať, stačí ho zapnúť pomocou príkazu sudo docker-compose up –d.



Obrázok 1 Architektúra komponentov Shuffle

Obrázok znázorňuje architektúru SOAR platformy Shuffle na serveri a tok spracovania dát. Frontend beží na porte 3001 a slúži ako webové rozhranie, cez ktoré používateľ pracuje s workflowmi a API. Požiadavky z frontendu sú smerované na backend, ktorý zabezpečuje logiku aplikácie a komunikáciu s databázou Elasticsearch, kde sa ukladajú workflows a aplikácie, a so súborovým systémom, v ktorom sa ukladajú logy. V runtime prostredí sa nachádzajú komponenty Orborus, Workers a Apps, pričom Orborus dohliada na backend a spúšťa workers, ktoré vykonávajú jednotlivé kroky workflowov. Apps predstavujú skripty, najčastejšie v jazyku Python, ktoré komunikujú s externými službami. Prostredníctvom API požiadaviek a webhookov je Shuffle prepojený s externými systémami, ako sú Elasticsearch, EDR alebo e-mailové služby.

2.2.2 Základné integračné väzby komponentu Shuffle v PoC

Aktuálne prepojené a otestované služby pomocou Shuffle:

- ELK – vieme si vyberať dáta pomocou volania API, vyhľadávať indexy a vytvárať nové indexy. Je ale nutné mať v iptables na zariadení kde beží ELK, povolené IP zariadenia na ktorom beží Shuffle.
- Discord – vieme posielať notifikácie vo formáte JSON pomocou webhooku ktorý sa vytvára v skupine na Discorde.
- OpenCTI – vieme vytvárať pomocou API nové indikátory, tieto vytvárame ako JSON a posielame pomocou API volania.
- DFIR IRIS – vieme pomocou API, vytvárať nové eventy, ktoré taktiež vieme poslať ako formát JSON.

2.3 Wazuh

Wazuh je v rámci PoC riešenia INNOSOC využitý ako centrálna platforma pre zber, koreláciu a analýzu bezpečnostných udalostí s funkcionalitou EDR.

2.3.1 Model nasadenia komponentu Wazuh

Architektúra nášho riešenia je založená na nasadení univerzálneho Wazuh-agenta na cieľové systémy a troch kľúčových centrálnych komponentov:

- Wazuh-manager (server): Zabezpečuje analýzu dát, správu agentov a detekciu hrozieb.
- Wazuh-indexer (databáza): Vysoko výkonný full-textový vyhľadávací a analytický nástroj slúžiaci na indexovanie a ukladanie bezpečnostných záznamov.
- Wazuh-dashboard: Webové používateľské rozhranie určené na vizualizáciu dát, analýzu incidentov a konfiguráciu systému

Centrálne komponenty sú nasadené na virtuálnom stroji (VM) v prostredí XenOrchestra. VM má k dispozícii: 8 virtuálnych CPU, 8GB RAM pamäte a 250GB úložiska.

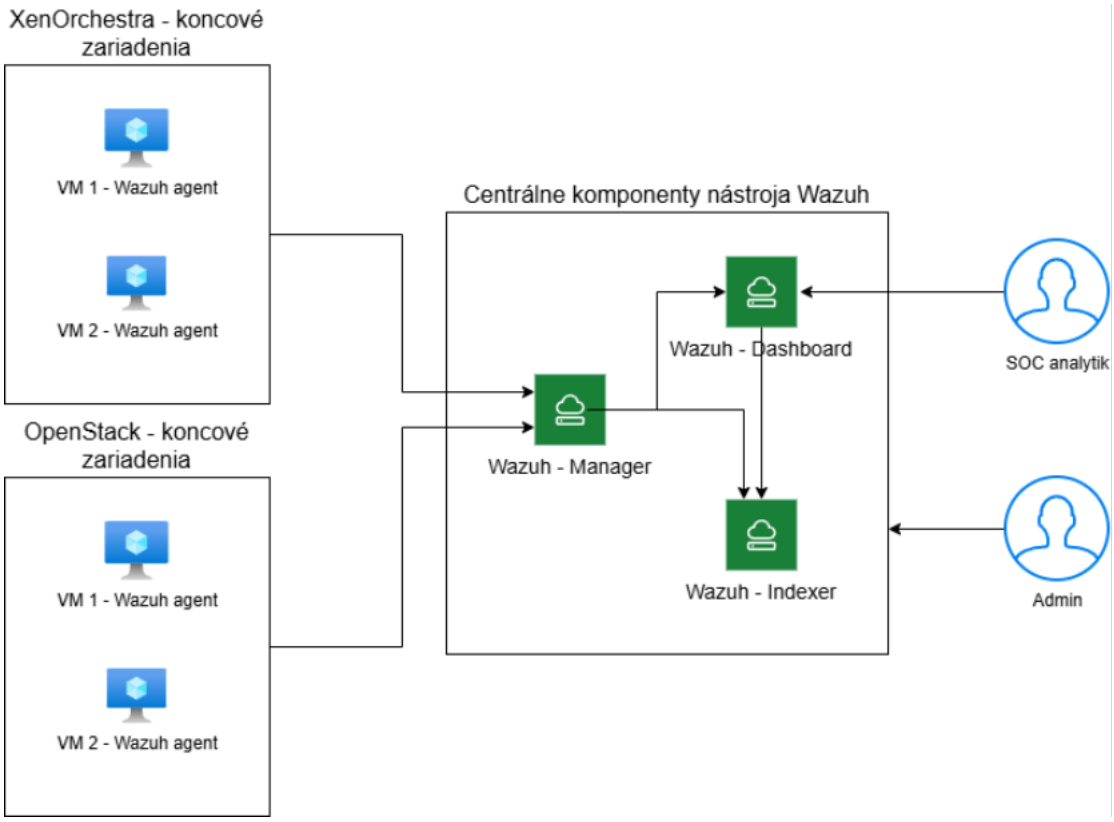
2.3.2 Monitorované prostredia nástrojom Wazuh

Monitorujeme prostredia dvoch tenatov. Prvým sú cloudové prostredia na Katedry informačných sietí na Fakulte riadenia a informatiky XenOrchestra a OpenStack. Druhé je prostredie vybranej ZŠ v Žiline.

2.3.2.1 Prostredie Katedry informačných sietí FRI UNIZA

V podmienkach KIS je systém konfigurovaný ako All-in-one inštalácia, pričom všetky kľúčové komponenty platformy Wazuh sú integrované do jedného virtuálneho stroja (VM). Toto riešenie zabezpečuje centralizovaný zber a analýzu dát z monitorovaných uzlov nachádzajúcich sa v prostrediach XenOrchestra a OpenStack. Grafické znázornenie výslednej sieťovej architektúry a hierarchie komponentov je ilustrované na obrázku nižšie.

Keďže sa jedná o spôsob nasadenia All-in-one tak všetky centrálne komponenty majú rovnakú IP adresu: 192.168.1.184. V prípade inštalácie Single alebo multi-node je potrebné aby mal každý komponent vlastnú IP adresu (keďže každý komponent bude na samostatnom servery alebo VM).



2.3.2.2 *Monitorovanie vybranej ZŠ v Žiline*

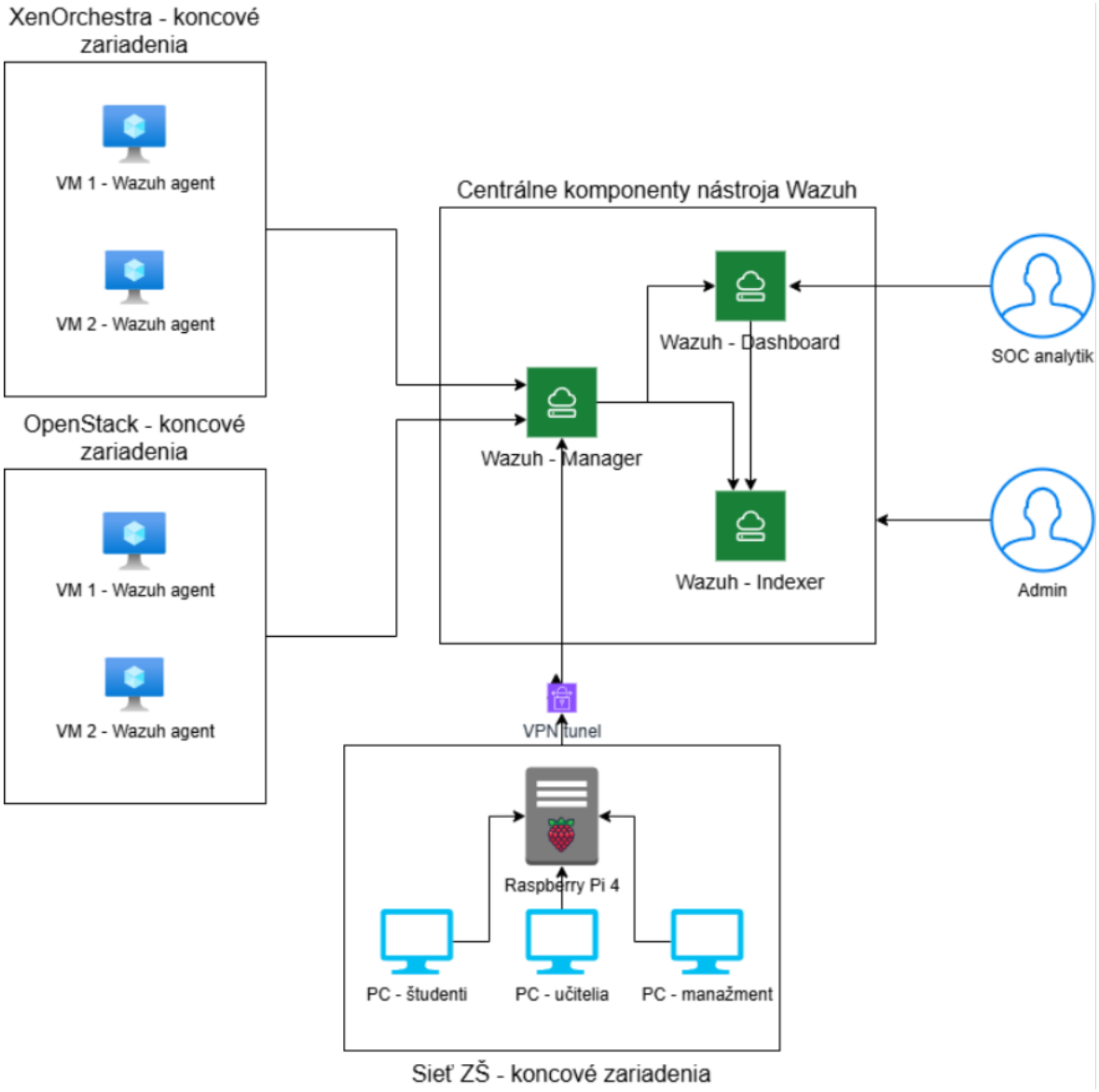
Druhá inštancia monitorovacieho systému je implementovaná v infraštruktúre vybranej základnej školy (ZŠ) v Žiline. Pôvodná architektúra v tomto prostredí pozostávala z lokálneho servera, na ktorom boli nasadené základné komponenty platformy Wazuh. Tento server zabezpečoval zber a ukladanie dát z koncových zariadení v rámci lokálnej siete ZŠ, pričom vzdialený prístup k rozhraniu Wazuh Dashboard bol realizovaný prostredníctvom VPN technológie ZeroTier.

Z dôvodu neefektívnej správy a nutnosti redundantnej konfigurácie dvoch samostatných manažérov došlo k revízii architektúry. Rozhodlo sa o centralizácii hlavných komponentov do prostredia XenOrchestra na KIS FRI. Dáta z koncových zariadení umiestnených v ZŠ sú po novom presmerované a spracovávané centrálne na KIS.

Prepojenie oboch lokalít je technicky riešené nasledovne:

- **Strana KIS:** Na centrálnom serveri sú okrem manažéra Wazuh prevádzkované aj serverové komponenty VPN Wireguard.
- **Strana ZŠ:** V lokálnej sieti je nasadené zariadenie Raspberry Pi 4, ktoré plní úlohu VPN klienta.

- **Sieťová konektivita:** Medzi lokalitami je vytvorený funkčný VPN tunel s adresným rozsahom 10.0.0.0/24.
- **Smerovanie (Routing):** Na lokálnom routeri v ZŠ je definovaná statická cesta (route), ktorá smeruje všetku prevádzku určenú pre sieť `10.0.0.0/24` cez bránu Raspberry Pi.



- **Wazuh-agenti** nainštalovaný na koncových zariadeniach v ZŠ odosielajú zozbierané dáta priamo na Wazuh-manažéra v sieti KIS prostredníctvom zabezpečeného tunela.

2.3.3 Nasadené a aktivované moduly systému Wazuh

V úvodnej fáze boli aktivované a konfigurované natívne moduly systému Wazuh, ktoré tvoria základný rámec pre monitorovanie bezpečnosti a dodržiavanie súladu (compliance). Nižšie sú detailne špecifikované vybrané funkcie:

2.3.3.1 *File Integrity Monitoring (FIM)*

Tento modul monitoruje zmeny v kritických systémových súboroch a adresároch. Funguje na princípe porovnávania kontrolných súčtov (checksums) a atribútov súborov v čase.

- **Mechanizmus:** Pri inicializácii vytvorí agent databázu hashov vybraných súborov. Pri každej zmene (vytvorenie, modifikácia, vymazanie) odošle upozornenie manažérovi.
- **Využitie:** Detekcia neoprávnených zmien v konfiguráciách alebo injekcia malvéru.

2.3.3.2 *Rootcheck (Rootkit Detection)*

Modul zameraný na detekciu skrytých hrozieb, ako sú rootkity, trójske kone alebo neoprávnené modifikácie jadra operačného systému.

- **Mechanizmus:** Vykonáva hĺbkovú kontrolu anomálií v systéme, porovnáva výstupy systémových volaní s priamym čítaním pôvodných (tzv. raw) dát a hľadá skryté procesy alebo porty.

2.3.3.3 *Vulnerability Detection (Detekcia zraniteľností)*

Tento modul automaticky identifikuje slabé miesta v operačnom systéme a nainštalovaných aplikáciách.

- **Mechanizmus:** Wazuh agent zhromažďuje zoznam nainštalovaných balíkov, ktorý následne manažér porovnáva s globálnymi databázami zraniteľností (**CVE** – Common Vulnerabilities and Exposures) v NVD - National Vulnerability Database.

2.3.3.4 *Policy monitoring & SCA (Security Configuration Assesment)*

Slúži na overenie, či je systém nakonfigurovaný v súlade s bezpečnostnými štandardmi (napr. CIS Benchmarks).

- **Mechanizmus:** Modul spúšťa skripty, ktoré kontrolujú nastavenia systému (napr. silu hesiel, zakázané protokoly). Výsledkom je skóre súladu a zoznam odporúčaní na nápravu.

2.3.3.5 *System Inventory (Systémová inventarizácia)*

Poskytuje komplexný prehľad o hardvérových a softvérových vlastnostiach monitorovaného endpointu.

Mechanizmus: Zhromažďuje údaje o sieťových rozhraniach, bežiacich procesoch, nainštalovaných balíkoch a verziách OS. Tieto dáta sú dostupné v reálnom čase cez wazuh-dashboard.

2.3.4 Základné integračné väzby - posielanie notifikácií

Ekosystém nástroja Wazuh ponúka široké spektrum integračných možností pre automatizované zasielanie výstrah prostredníctvom externých komunikačných kanálov a nástrojov tretích strán.

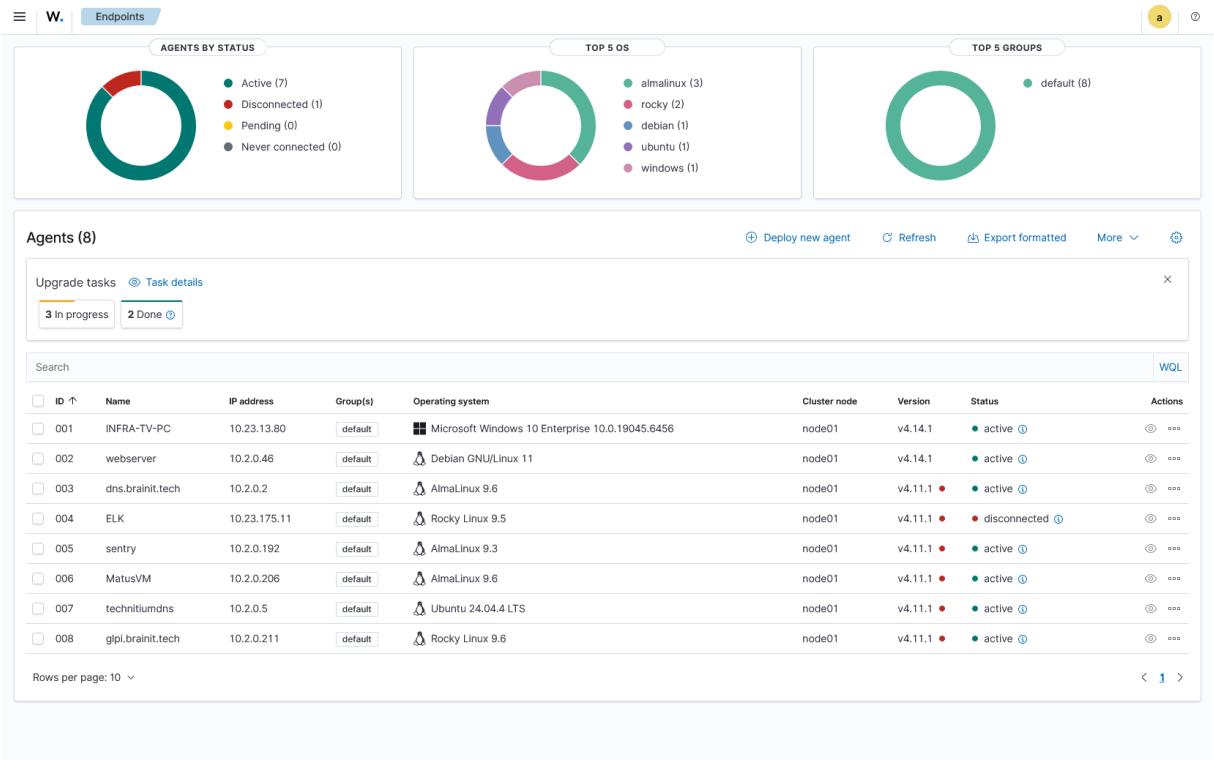
Posielanie notifikácií do externých platforiem (Slack, MS Teams, PagerDuty a pod.) je vo Wazuhu riešené cez modul Integrations. Integrácia je realizovaná prostredníctvom aplikačného rozhrania Webhook, pričom systém umožňuje granulárne filtrovanie odosielaných udalostí na základe ich závažnosti. Tento proces umožňuje transformovať interné výstrahy (Alerts) na okamžité správy pre bezpečnostný tím, čím sa výrazne skracuje reakčný čas (MTTR – Mean Time to Respond).

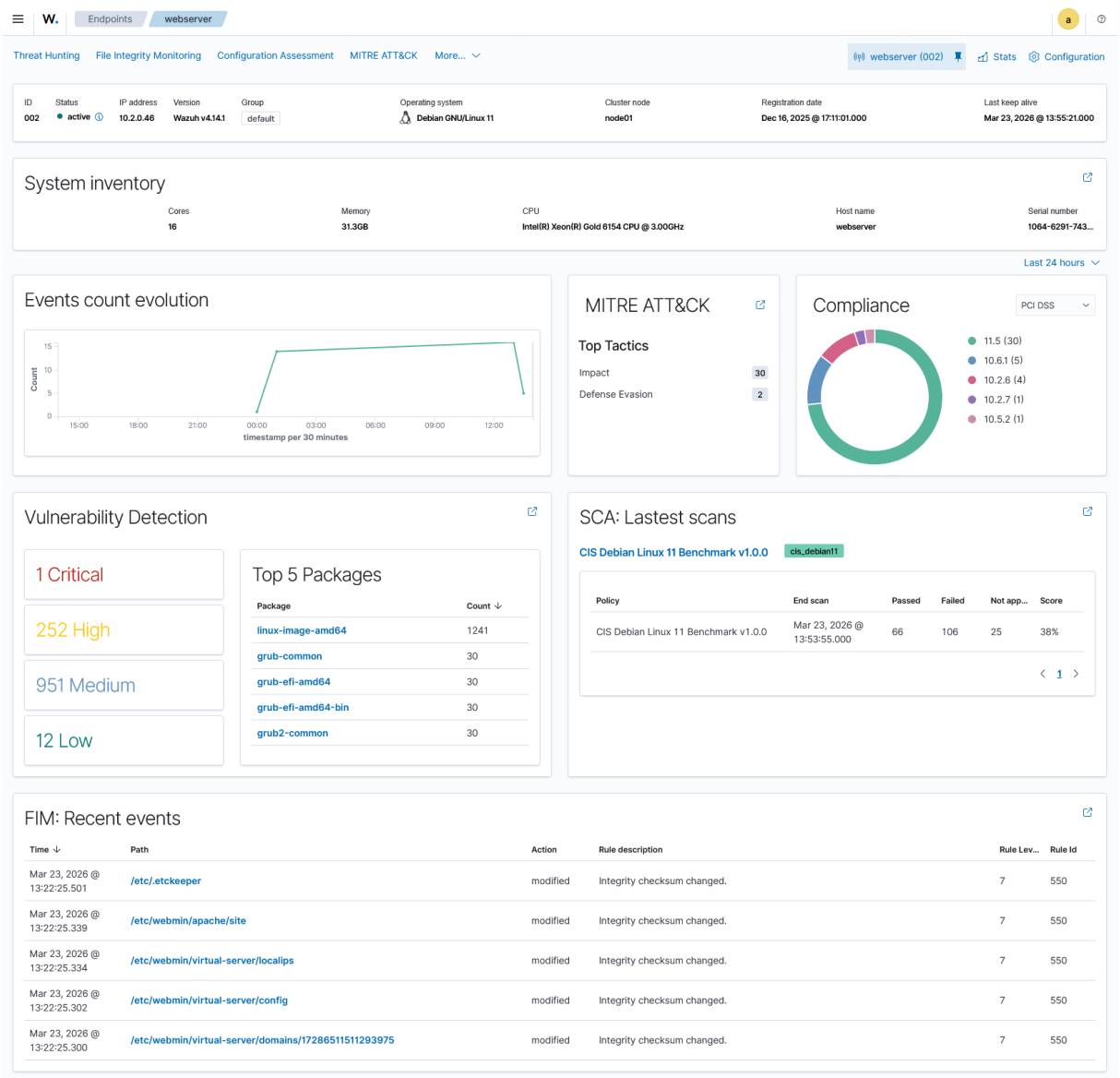
Analyzovali sme rôzne nástroje pre posielanie notifikácií ako Slack, PagerDuty a MS Teams. Pre účely monitorovania sme sa rozhodli použiť MS Teams. Integrácia je postavená na využití Incoming Webhook (HTTP endpoint). Platforma Wazuh odosiela dáta vo formáte JSON, ktoré sa v Teams transformujú na vizuálne štruktúrované karty (Adaptive Cards). Tento formát umožňuje prehľadné zobrazenie kľúčových metrík, ako sú úroveň závažnosti (severity), identifikácia agenta, ID pravidla a podrobný opis udalosti.

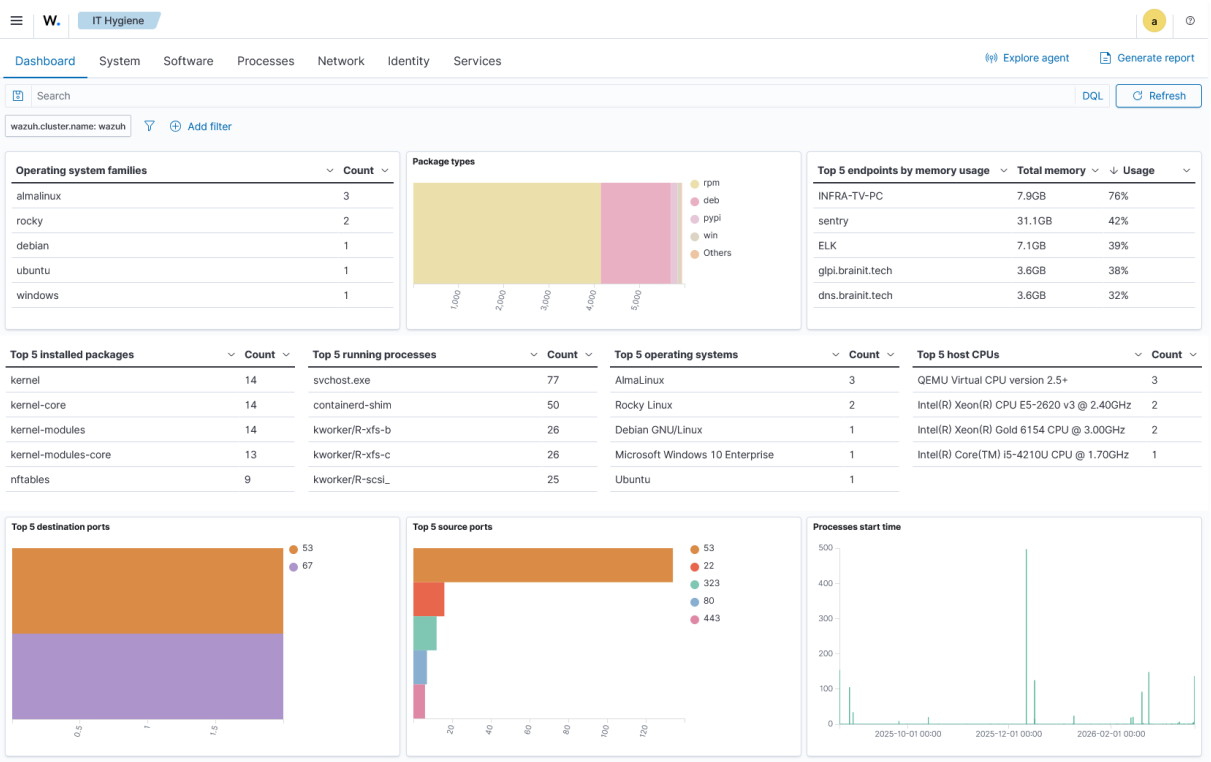
Hlavným dôvodom výberu bola skutočnosť, že táto platforma je plne integrovaná v existujúcej digitálnej infraštruktúre univerzity. Využitie ekosystému Microsoft 365 nám prináša viaceré výhody:

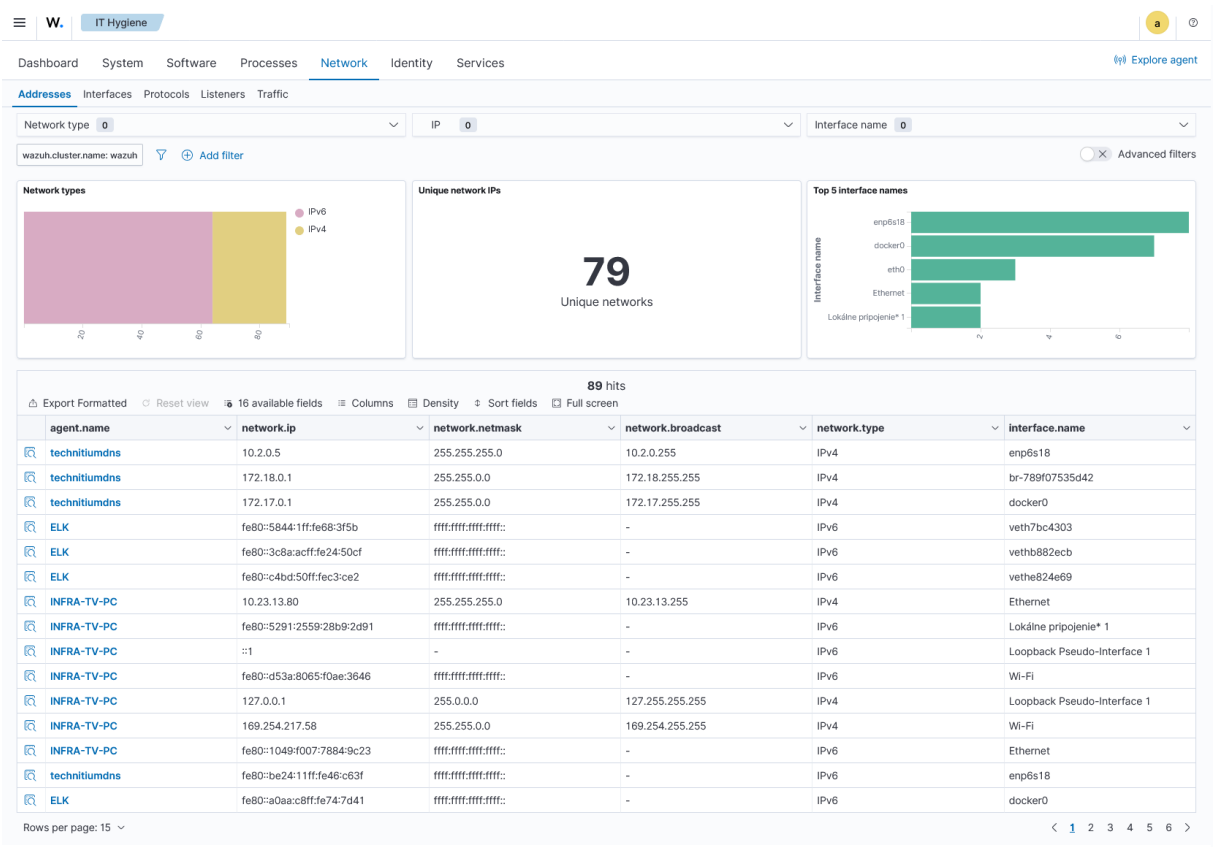
- **Efektivita nasadenia:** Odpadá nutnosť budovania novej komunikačnej infraštruktúry alebo správy ďalších používateľských účtov.
- **Konektivita:** Integrácia prebieha v rámci overeného a zabezpečeného prostredia univerzity.
- **Dostupnosť:** Zabezpečuje okamžitý prístup k notifikáciám pre všetkých členov tímu bez potreby inštalácie dodatočného softvéru tretích strán.

Ukážka z dashboardu nástroja Wazuh:

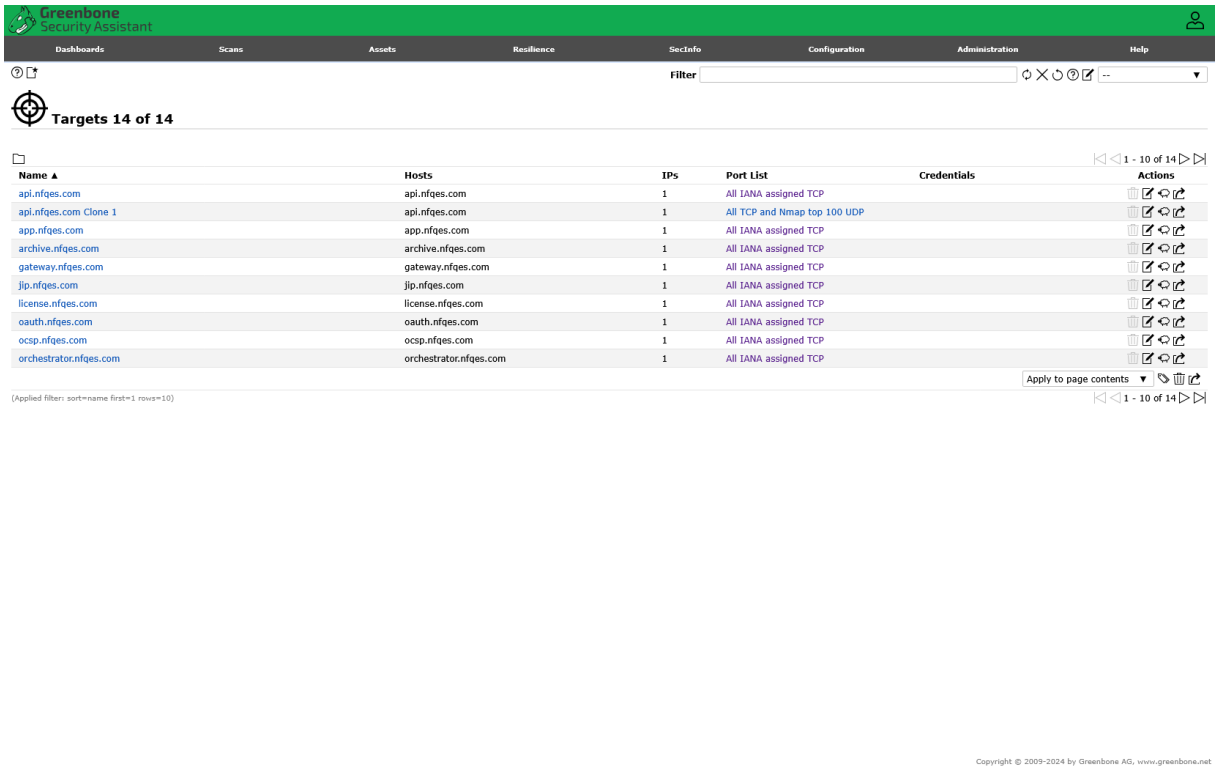

W.
Endpoints
a
AGENTS BY STATUS
Active (7)
Disconnected (1)
Pending (0)
Never connected (0)
TOP 5 OS
almalinux (3)
rocky (2)
debian (1)
ubuntu (1)
windows (1)
TOP 5 GROUPS
default (8)
Agents (8)
Deploy new agent
Refresh
Export formatted
More
Upgrade tasks
Task details
3 In progress
2 Done
Search
WQL
ID
Name
IP address
Group(s)
Operating system
Cluster node
Version
Status
Actions
001 INFRA-TV-PC 10.23.13.80 default Microsoft Windows 10 Enterprise 10.0.19045.6456 node01 v4.14.1 active
002 webserver 10.2.0.46 default Debian GNU/Linux 11 node01 v4.14.1 active
003 dns.brainit.tech 10.2.0.2 default AlmaLinux 9.6 node01 v4.11.1 active
004 ELK 10.23.175.11 default Rocky Linux 9.5 node01 v4.11.1 disconnected
005 sentry 10.2.0.192 default AlmaLinux 9.3 node01 v4.11.1 active
006 MatusVM 10.2.0.206 default AlmaLinux 9.6 node01 v4.11.1 active
007 technitiumdns 10.2.0.5 default Ubuntu 24.04.4 LTS node01 v4.11.1 active
008 glpi.brainit.tech 10.2.0.211 default Rocky Linux 9.6 node01 v4.11.1 active
Rows per page: 10
1


Endpoints
webserver
Threat Hunting
File Integrity Monitoring
Configuration Assessment
MITRE ATT&CK
More...
webserver (002)
Stats
Configuration
ID 002
Status active
IP address 10.2.0.46
Version Wazuh v4.14.1
Group default
Operating system Debian GNU/Linux 11
Cluster node node01
Registration date Dec 16, 2025 @ 17:11:01.000
Last keep alive Mar 23, 2026 @ 13:55:21.000
System inventory
Cores 16
Memory 31.3GB
CPU Intel(R) Xeon(R) Gold 6154 CPU @ 3.00GHz
Host name webserver
Serial number 1064-6291-743...
Last 24 hours
Events count evolution
Count
timestamp per 30 minutes
MITRE ATT&CK
Top Tactics
Impact 30
Defense Evasion 2
Compliance
PCI DSS
11.5 (30)
10.6.1 (5)
10.2.6 (4)
10.2.7 (1)
10.5.2 (1)
Vulnerability Detection
1 Critical
252 High
951 Medium
12 Low
Top 5 Packages
Package
Count
linux-image-amd64 1241
grub-common 30
grub-efi-amd64 30
grub-efi-amd64-bin 30
grub2-common 30
SCA: Lastest scans
CIS Debian Linux 11 Benchmark v1.0.0
cis_debian11
Policy
End scan
Passed
Failed
Not app...
Score
CIS Debian Linux 11 Benchmark v1.0.0
Mar 23, 2026 @ 13:53:55.000
66
106
25
38%
FIM: Recent events
Time
Path
Action
Rule description
Rule Lev...
Rule Id
Mar 23, 2026 @ 13:22:25.501 /etc/.etckeeper modified Integrity checksum changed. 7 550
Mar 23, 2026 @ 13:22:25.339 /etc/webmin/apache/site modified Integrity checksum changed. 7 550
Mar 23, 2026 @ 13:22:25.334 /etc/webmin/virtual-server/localips modified Integrity checksum changed. 7 550
Mar 23, 2026 @ 13:22:25.302 /etc/webmin/virtual-server/config modified Integrity checksum changed. 7 550
Mar 23, 2026 @ 13:22:25.300 /etc/webmin/virtual-server/domains/17286511511293975 modified Integrity checksum changed. 7 550

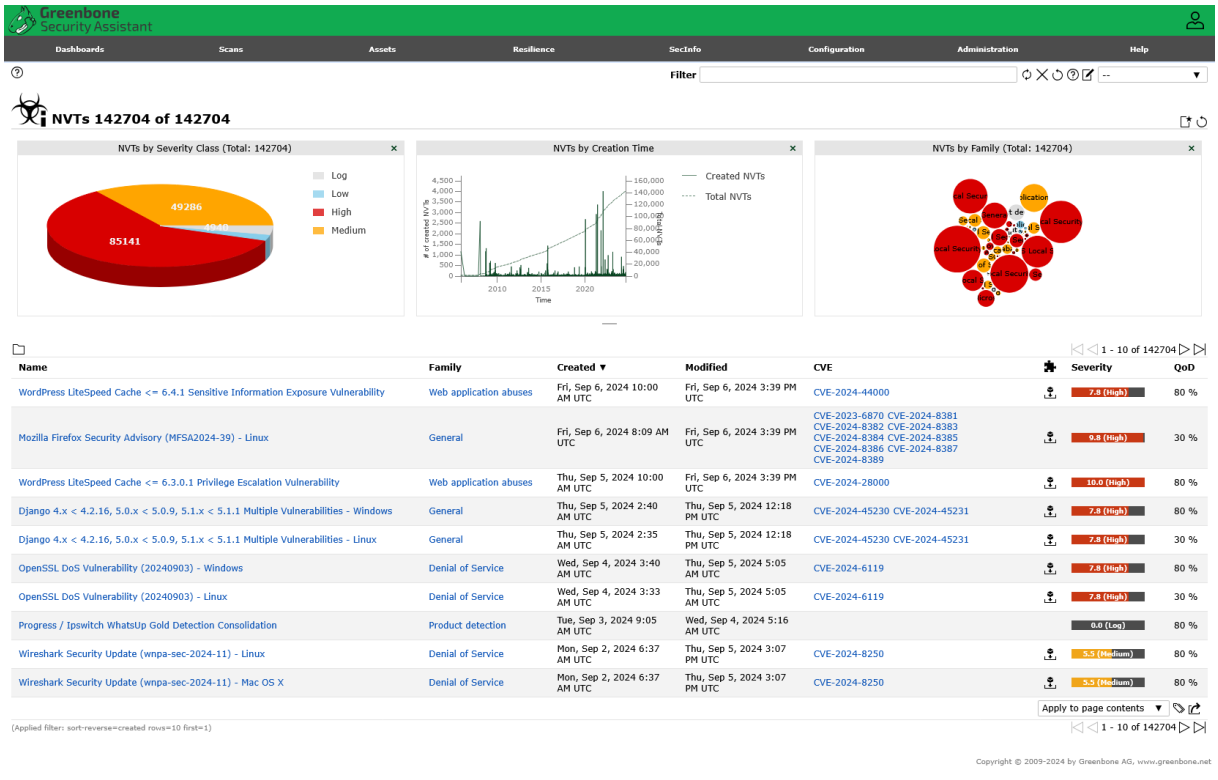

IT Hygiene
Dashboard
System
Software
Processes
Network
Identity
Services
Explore agent
Generate report
Search
DQL
Refresh
wazuh.cluster.name: wazuh
Add filter
Operating system families
Count
almalinux 3
rocky 2
debian 1
ubuntu 1
windows 1
Package types
rpm
deb
pypi
win
Others
1,000
2,000
3,000
4,000
5,000
Top 5 endpoints by memory usage
Total memory
Usage
INFRA-TV-PC 7.9GB 76%
sentry 31.1GB 42%
ELK 7.1GB 39%
glpi.brainit.tech 3.6GB 38%
dns.brainit.tech 3.6GB 32%
Top 5 installed packages
Count
kernel 14
kernel-core 14
kernel-modules 14
kernel-modules-core 13
nftables 9
Top 5 running processes
Count
svchost.exe 77
containerd-shim 50
kworker/R-xfs-b 26
kworker/R-xfs-c 26
kworker/R-scsi_ 25
Top 5 operating systems
Count
AlmaLinux 3
Rocky Linux 2
Debian GNU/Linux 1
Microsoft Windows 10 Enterprise 1
Ubuntu 1
Top 5 host CPUs
Count
QEMU Virtual CPU version 2.5+ 3
Intel(R) Xeon(R) CPU E5-2620 v3 @ 2.40GHz 2
Intel(R) Xeon(R) Gold 6154 CPU @ 3.00GHz 2
Intel(R) Core(TM) i5-4210U CPU @ 1.70GHz 1
Top 5 destination ports
53
67
0.5
1
1.5
Top 5 source ports
53
22
323
80
443
20
40
60
80
100
120
Processes start time
500
400
300
200
100
0
2025-10-01 00:00
2025-12-01 00:00
2026-02-01 00:00


IT Hygiene
Dashboard
System
Software
Processes
Network
Identity
Services
Explore agent
Addresses
Interfaces
Protocols
Listeners
Traffic
Network type
IP
Interface name
wazuh.cluster.name: wazuh
Add filter
Advanced filters
Network types
IPv6
IPv4
Unique network IPs
79
Unique networks
Top 5 interface names
enp6s18
docker0
eth0
Ethernet
Lokálne pripojenie* 1
Interface name
89 hits
Export Formatted
Reset view
16 available fields
Columns
Density
Sort fields
Full screen
agent.name
network.ip
network.netmask
network.broadcast
network.type
interface.name
technitiumdns 10.2.0.5 255.255.255.0 10.2.0.255 IPv4 enp6s18
technitiumdns 172.18.0.1 255.255.0.0 172.18.255.255 IPv4 br-789f07535d42
technitiumdns 172.17.0.1 255.255.0.0 172.17.255.255 IPv4 docker0
ELK fe80::5844:1ff:fe68:3f5b ffff:ffff:ffff:ffff:: - IPv6 veth7bc4303
ELK fe80::3c8a:acff:fe24:50cf ffff:ffff:ffff:ffff:: - IPv6 vethb882ecb
ELK fe80::c4bd:50ff:fec3:ce2 ffff:ffff:ffff:ffff:: - IPv6 vethe824e69
INFRA-TV-PC 10.23.13.80 255.255.255.0 10.23.13.255 IPv4 Ethernet
INFRA-TV-PC fe80::5291:2559:28b9:2d91 ffff:ffff:ffff:ffff:: - IPv6 Lokálne pripojenie* 1
INFRA-TV-PC ::1 - - IPv6 Loopback Pseudo-Interface 1
INFRA-TV-PC fe80::d53a:8065:f0ae:3646 ffff:ffff:ffff:ffff:: - IPv6 Wi-Fi
INFRA-TV-PC 127.0.0.1 255.0.0.0 127.255.255.255 IPv4 Loopback Pseudo-Interface 1
INFRA-TV-PC 169.254.217.58 255.255.0.0 169.254.255.255 IPv4 Wi-Fi
INFRA-TV-PC fe80::1049:f007:7884:9c23 ffff:ffff:ffff:ffff:: - IPv6 Ethernet
technitiumdns fe80::be24:11ff:fe46:c63f ffff:ffff:ffff:ffff:: - IPv6 enp6s18
ELK fe80::a0aa:c8ff:fe74:7d41 ffff:ffff:ffff:ffff:: - IPv6 docker0
Rows per page: 15
Configuration A...
webserver
Dashboard
Inventory
Events
webserver (002)
CIS DEBIAN LINUX 11 BENCHMARK V1.0.0
Passed (66)
Failed (106)
Not applicable (25)
CIS Debian Linux 11 Benchmark v1.0.0
Passed 66
Failed 106
Not applicable 25
Score 38%
End scan Mar 23, 2026 @ 13:53:55.000
Checks (197)
Refresh
Export formatted
Search
WQL
ID
Title
Target
Result
29500 Ensure /tmp is a separate partition. Command: findmnt --kernel /tmp Failed
29501 Ensure nodev option set on /tmp partition. Command: findmnt --kernel /tmp Failed
29502 Ensure noexec option set on /tmp partition. Command: findmnt --kernel /tmp Failed
29503 Ensure nosuid option set on /tmp partition. Command: findmnt --kernel /tmp Failed
29504 Ensure separate partition exists for /var. Command: findmnt --kernel /var Failed
29505 Ensure nodev option set on /var partition. Command: findmnt --kernel /var Failed
29506 Ensure nosuid option set on /var partition. Command: findmnt --kernel /var Failed
29507 Ensure separate partition exists for /var/tmp. Command: findmnt --kernel /var/tmp Failed
29508 Ensure noexec option set on /var/tmp partition. Command: findmnt --kernel /var/tmp Failed
29509 Ensure nosuid option set on /var/tmp partition. Command: findmnt --kernel /var/tmp Failed
Policy
CIS Debian Linux 11 Benchmark v1.0.0
Rows per page: 10

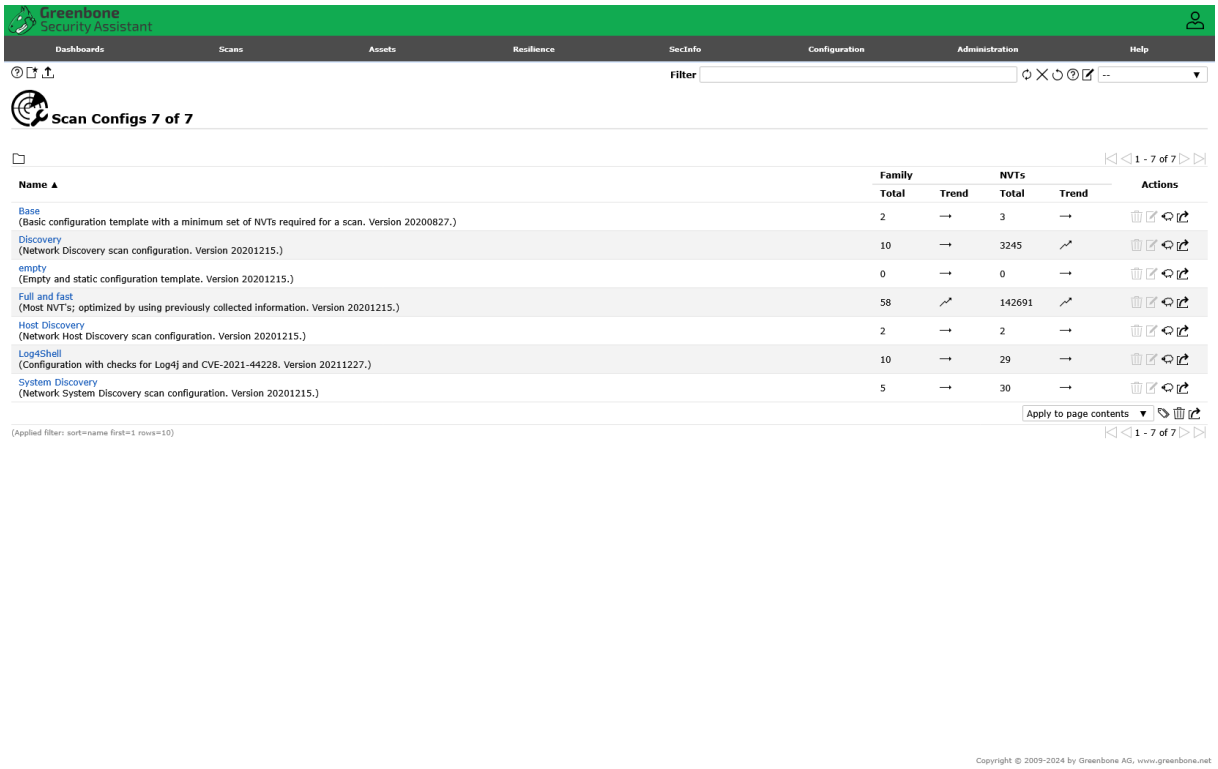
2.4 Greenbone Community edition



Copyright © 2009-2024 by Greenbone AG, www.greenbone.net



Copyright © 2009-2024 by Greenbone AG, www.greenbone.net



2.5 OpenCTI

OpenCTI je v rámci PoC riešenia INNOSOC využitý ako ako centrálne úložisko hrozieb, kde sa zhromažďujú indikátory kompromitácie (IoC), TTP techniky, kampane a aktéri. Platforma umožňuje tieto dáta analyzovať a následne distribuovať do ďalších bezpečnostných nástrojov bez nutnosti manuálneho zásahu.

OpenCTI (Open Cyber Threat Intelligence Platform) je open-source platforma určená na správu, analýzu a zdieľanie informácií o kybernetických hrozbách (CTI). Umožňuje agregovať dáta z rôznych zdrojov, korelovať ich a transformovať do jednotného modelu založeného na štandarde STIX.

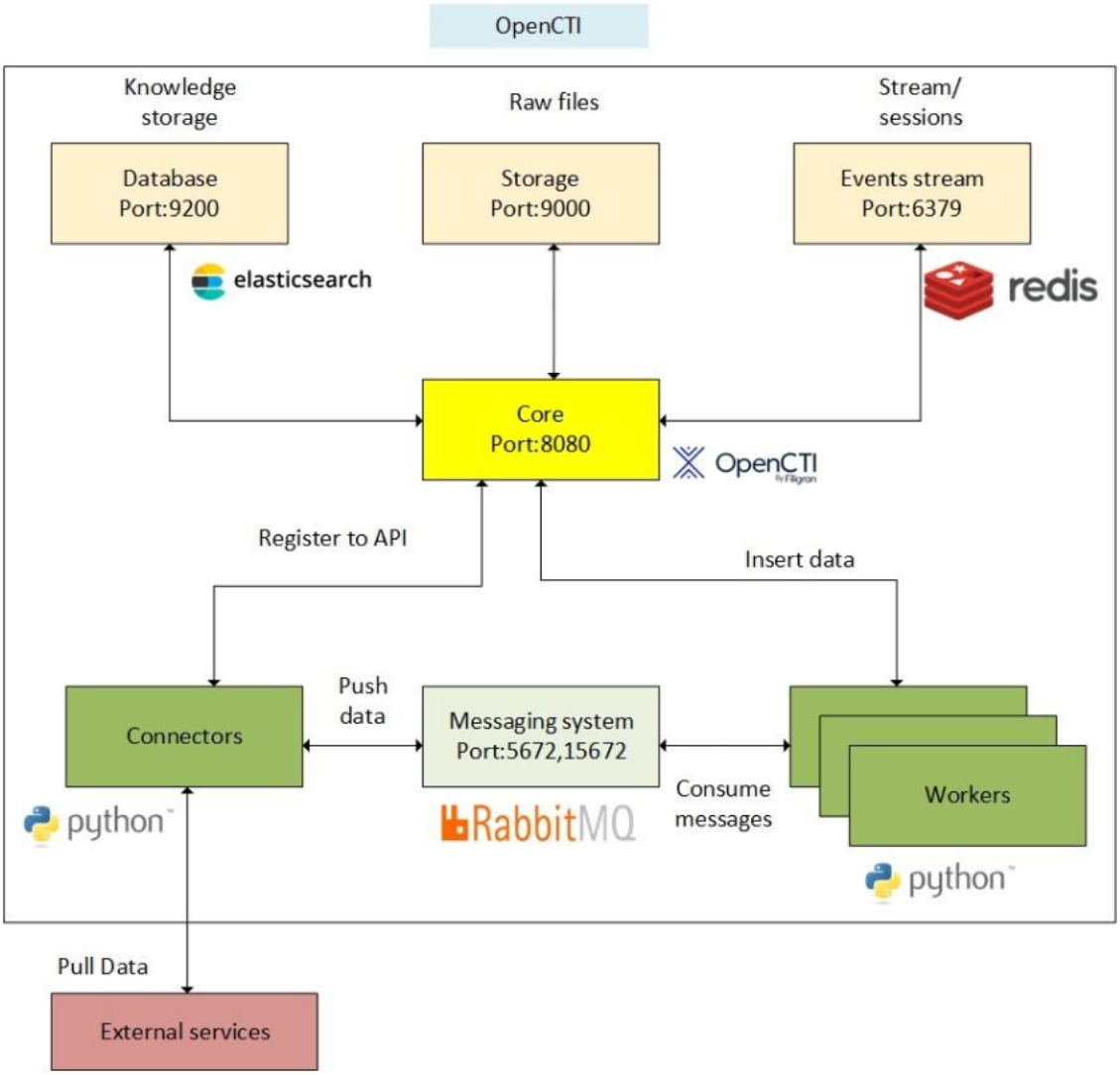
2.5.1 Nasadenie Shuffle v prostredí OpenStack

Technická špecifikácia nasadenia Shuffle:

- Prostredie – OpenStack
- Meno inštancie – OpenCTI_s
- IP adresa – 192.168.4.121
- vCPU – 4
- RAM – 16 GB
- Disk – 80 GB
- OS – Ubuntu 22.04

Schéma na obrázku znázorňuje architektúru platformy OpenCTI na serveri a tok spracovania dát medzi jednotlivými komponentmi. Centrálnym prvkom je OpenCTI Core bežiaci na porte 8080, ktorý zabezpečuje logiku aplikácie a komunikáciu medzi službami. Core komunikuje s databázou Elasticsearch na porte 9200, kde sa ukladajú všetky CTI dáta, ako aj s

úložiskom súborov na porte 9000, ktoré slúži na ukladanie raw dát a príloh. Na správu eventov a cache sa využíva Redis na porte 6379.



V runtime prostredí sa nachádza messaging systém RabbitMQ (porty 5672 a 15672), ktorý zabezpečuje asynchrónnu komunikáciu medzi komponentmi. Konektory získavajú dáta z externých zdrojov, registrujú sa cez API do Core a následne posielajú dáta do RabbitMQ. Workers tieto správy spracovávajú, vykonávajú transformáciu a enrichment dát a následne ich ukladajú späť do OpenCTI cez Core.

2.5.2 Základné integračné väzby pre OpeCTI komponent

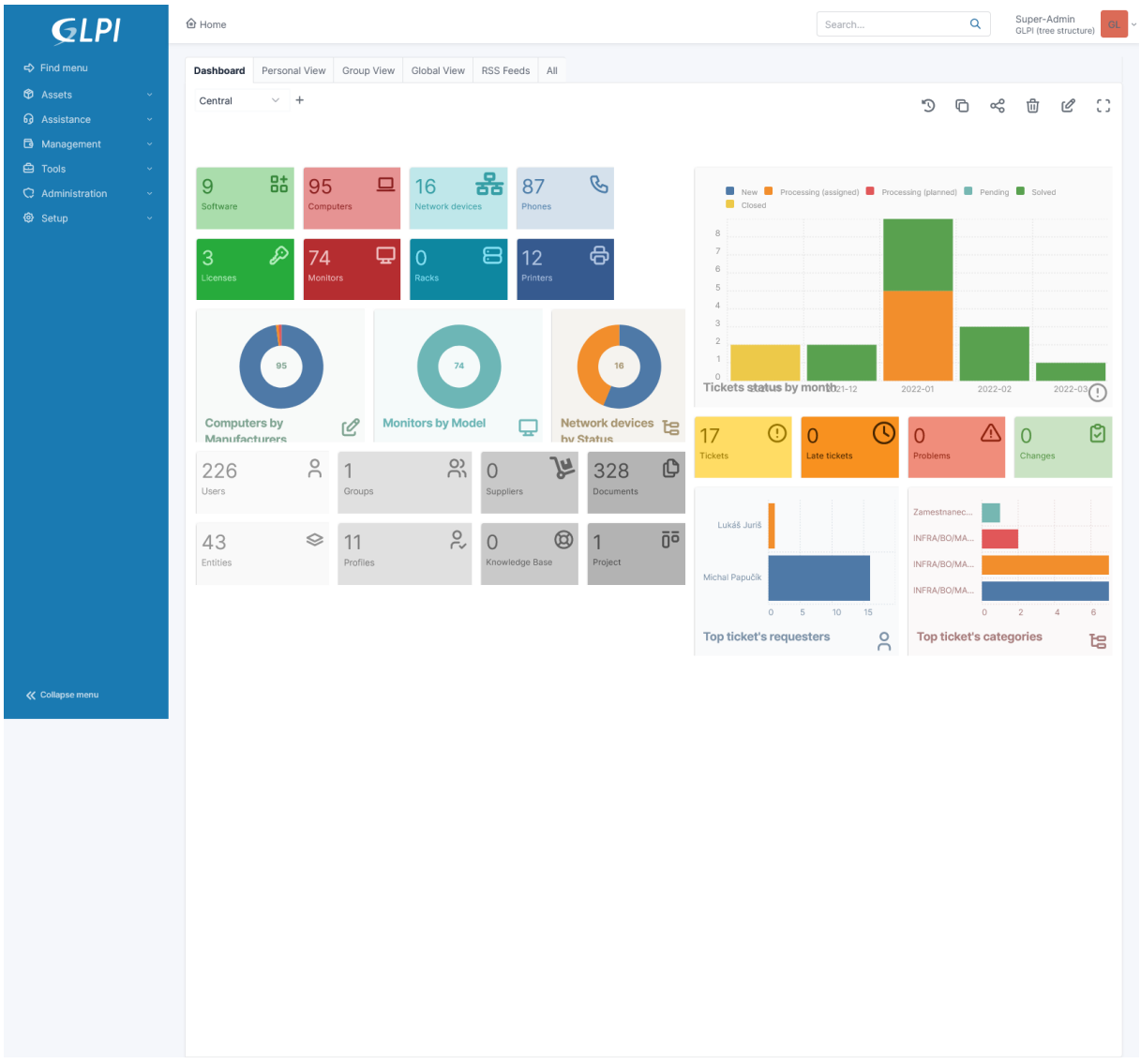
Aktuálne prepojené a otestované služby pomocou OpenCTI:

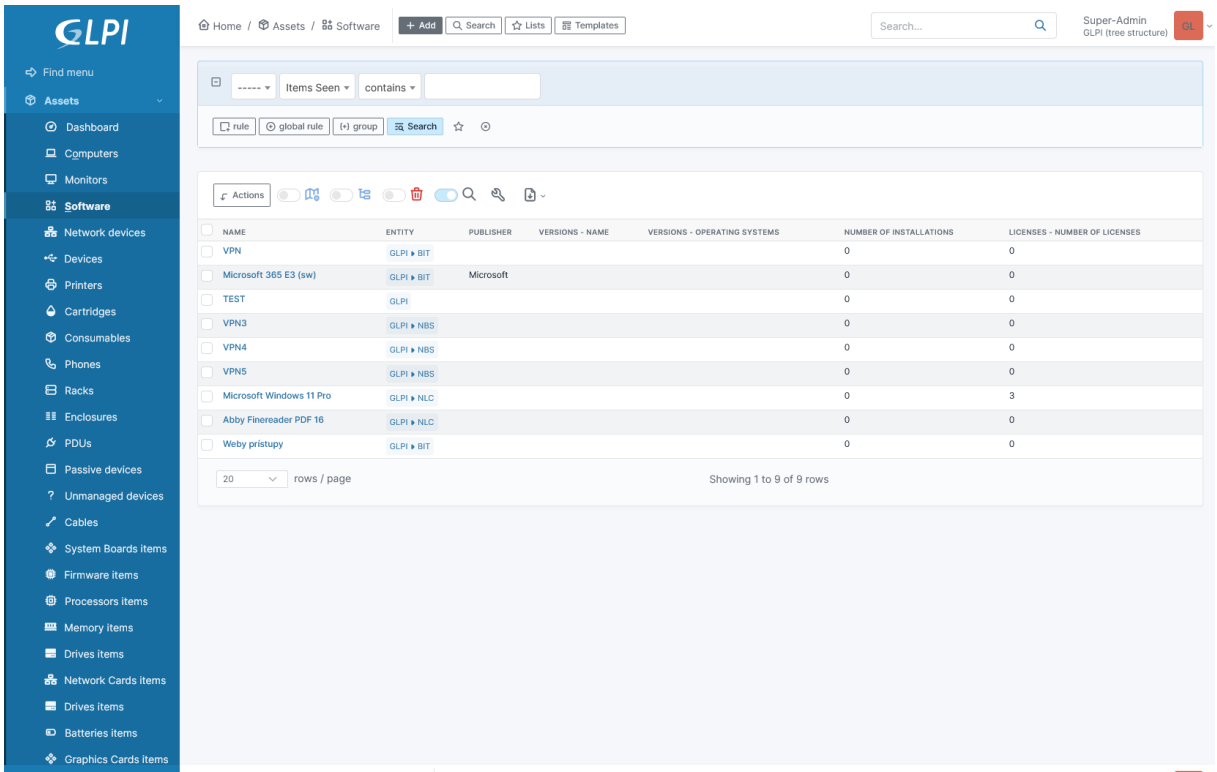
- **Abuse.ch (ThreatFox, URLhaus, Malwarebazzar)** – automatický import IoC
- **MITRE ATT&CK** – mapovanie techník a taktík
- **SOAR (Shuffle)** – automatizácia reakcií na incidenty

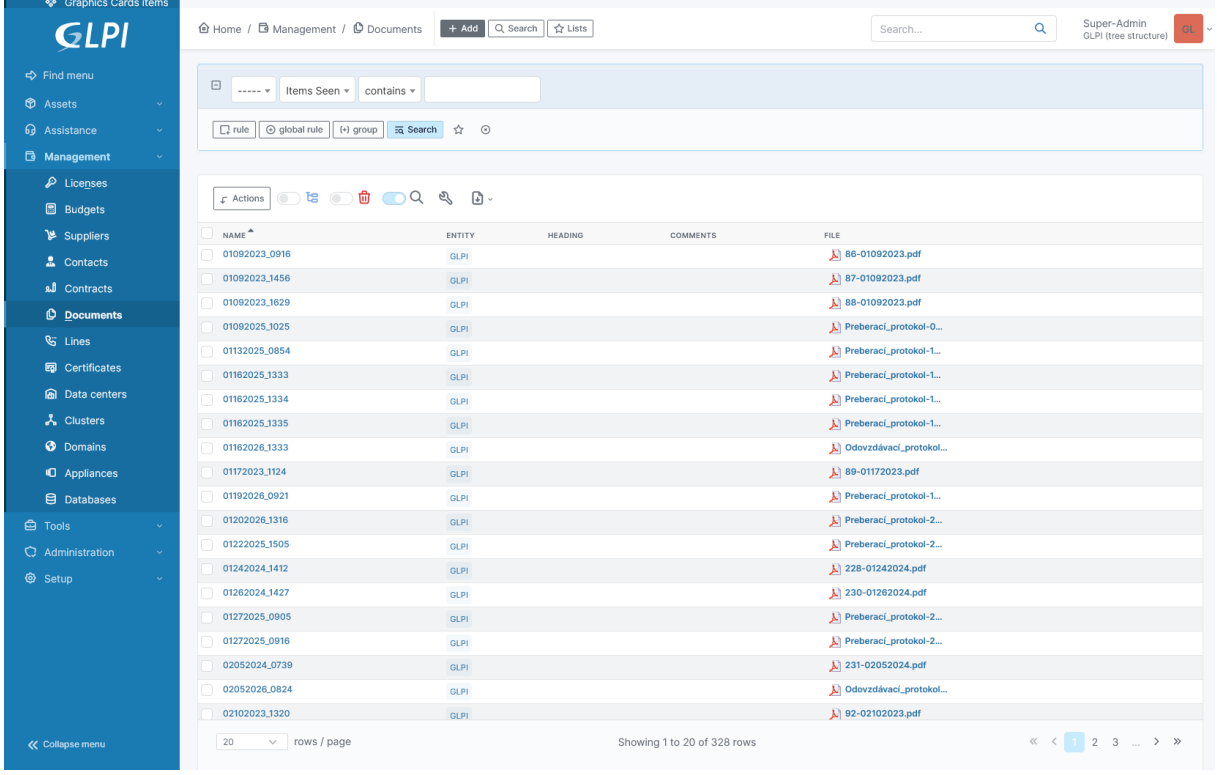
2.6 MISP

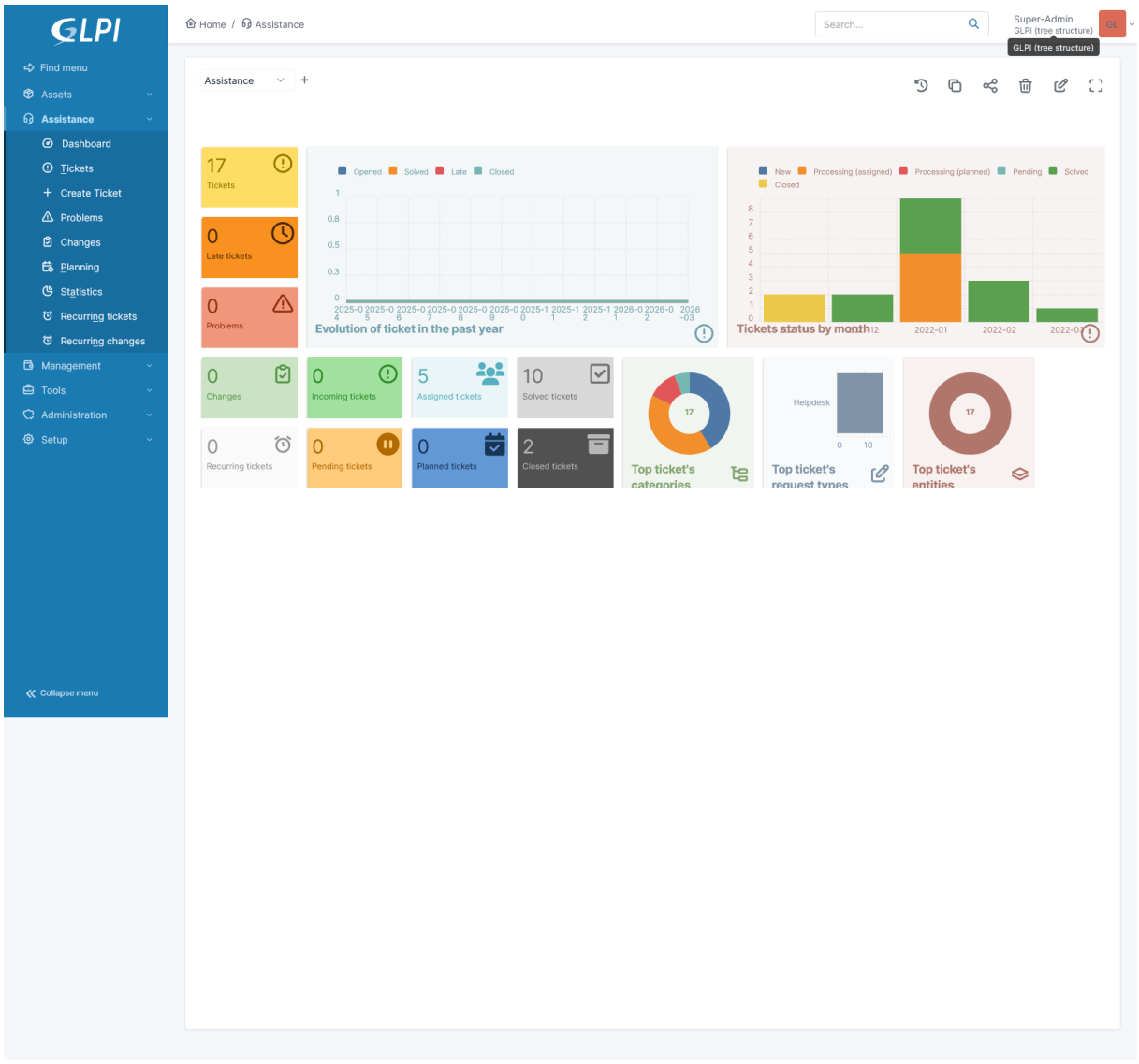
Nebude, nemame nic, CTI komponent bude zastresene iba platformou OpenCTI.

2.7 GLPI




GLPI
Find menu
Assets
Dashboard
Computers
Monitors
Software
Network devices
Devices
Printers
Cartridges
Consumables
Phones
Racks
Enclosures
PDUs
Passive devices
Unmanaged devices
Cables
System Boards items
Firmware items
Processors items
Memory items
Drives items
Network Cards items
Drives items
Batteries items
Graphics Cards items
Home / Assets / Software
+ Add
Search
Lists
Templates
Search...
Super-Admin
GLPI (tree structure)
Items Seen
contains
rule
global rule
group
Search
Actions
NAME
ENTITY
PUBLISHER
VERSIONS - NAME
VERSIONS - OPERATING SYSTEMS
NUMBER OF INSTALLATIONS
LICENSES - NUMBER OF LICENSES
VPN
GLPI ▸ BIT
0
0
Microsoft 365 E3 (sw)
GLPI ▸ BIT
Microsoft
0
0
TEST
GLPI
0
0
VPN3
GLPI ▸ NBS
0
0
VPN4
GLPI ▸ NBS
0
0
VPN5
GLPI ▸ NBS
0
0
Microsoft Windows 11 Pro
GLPI ▸ NLC
0
3
Abby Finereader PDF 16
GLPI ▸ NLC
0
0
Weby prístupy
GLPI ▸ BIT
0
0
20
rows / page
Showing 1 to 9 of 9 rows


GLPI
Find menu
Assets
Assistance
Management
Licenses
Budgets
Suppliers
Contacts
Contracts
Documents
Lines
Certificates
Data centers
Clusters
Domains
Appliances
Databases
Tools
Administration
Setup
Collapse menu
Home / Management / Documents
+ Add
Search
Lists
Search...
Super-Admin
GLPI (tree structure)
Items Seen
contains
rule
global rule
group
Search
Actions
NAME
ENTITY
HEADING
COMMENTS
FILE
01092023_0916
GLPI
86-01092023.pdf
01092023_1456
GLPI
87-01092023.pdf
01092023_1629
GLPI
88-01092023.pdf
01092025_1025
GLPI
Preberací_protokol-0...
01132025_0854
GLPI
Preberací_protokol-1...
01162025_1333
GLPI
Preberací_protokol-1...
01162025_1334
GLPI
Preberací_protokol-1...
01162025_1335
GLPI
Preberací_protokol-1...
01162026_1333
GLPI
Odovzdávací_protokol...
01172023_1124
GLPI
89-01172023.pdf
01192026_0921
GLPI
Preberací_protokol-1...
01202026_1316
GLPI
Preberací_protokol-2...
01222025_1505
GLPI
Preberací_protokol-2...
01242024_1412
GLPI
228-01242024.pdf
01262024_1427
GLPI
230-01262024.pdf
01272025_0905
GLPI
Preberací_protokol-2...
01272025_0916
GLPI
Preberací_protokol-2...
02052024_0739
GLPI
231-02052024.pdf
02052026_0824
GLPI
Odovzdávací_protokol...
02102023_1320
GLPI
92-02102023.pdf
20
rows / page
Showing 1 to 20 of 328 rows
1
2
3



2.8 NetBox

2.8.1 Úloha NetBox v PoC

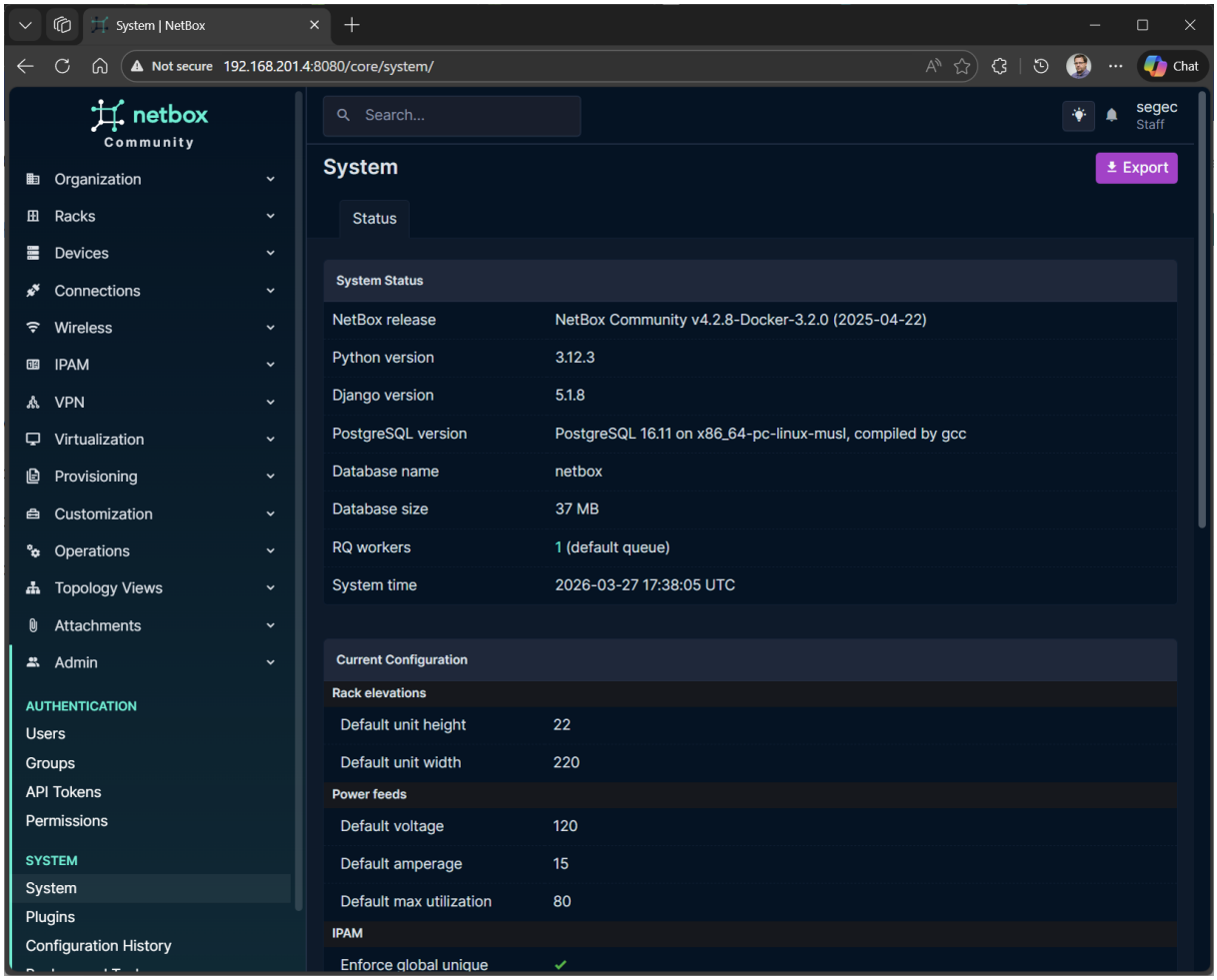
NetBox je v rámci PoC riešenia INNOSOC využitý ako podporný nástroj pre evidenciu a správu infraštruktúrneho kontextu. V navrhovanom SOC riešení neplní úlohu detekčného ani reakčného komponentu, ale slúži ako zdroj informácií o aktívach, adresácii, sieťových segmentoch a vybraných väzbách medzi prvkami infraštruktúry. Jeho význam spočíva najmä v tom, že umožňuje lepšie pochopiť, ku ktorému zariadeniu, systému alebo časti siete sa konkrétna udalosť vzťahuje. V rámci PoC je NetBox chápaný ako kontextová vrstva, ktorá dopĺňa technické údaje získané z ostatných komponentov riešenia. Pri analýze bezpečnostných udalostí tak môže slúžiť na dohľadanie základných informácií o zariadeniach, IP rozsahoch, lokalitách alebo logickom zaradení prvkov infraštruktúry. Takýto prístup zvyšuje prehľadnosť riešenia a vytvára predpoklad na presnejšie vyhodnocovanie bezpečnostných situácií.

NetBox v SOC ponúka kontext prostredia, v ktorom bezpečnostné logy a alerty vznikajú. V tomto zmysle predstavuje ***doplnkový***, ale dôležitý komponent riešenia, ktorý podporuje orientáciu v infraštruktúre a pripravuje základ pre ďalšie integračné využitie v rozšírených scenároch.

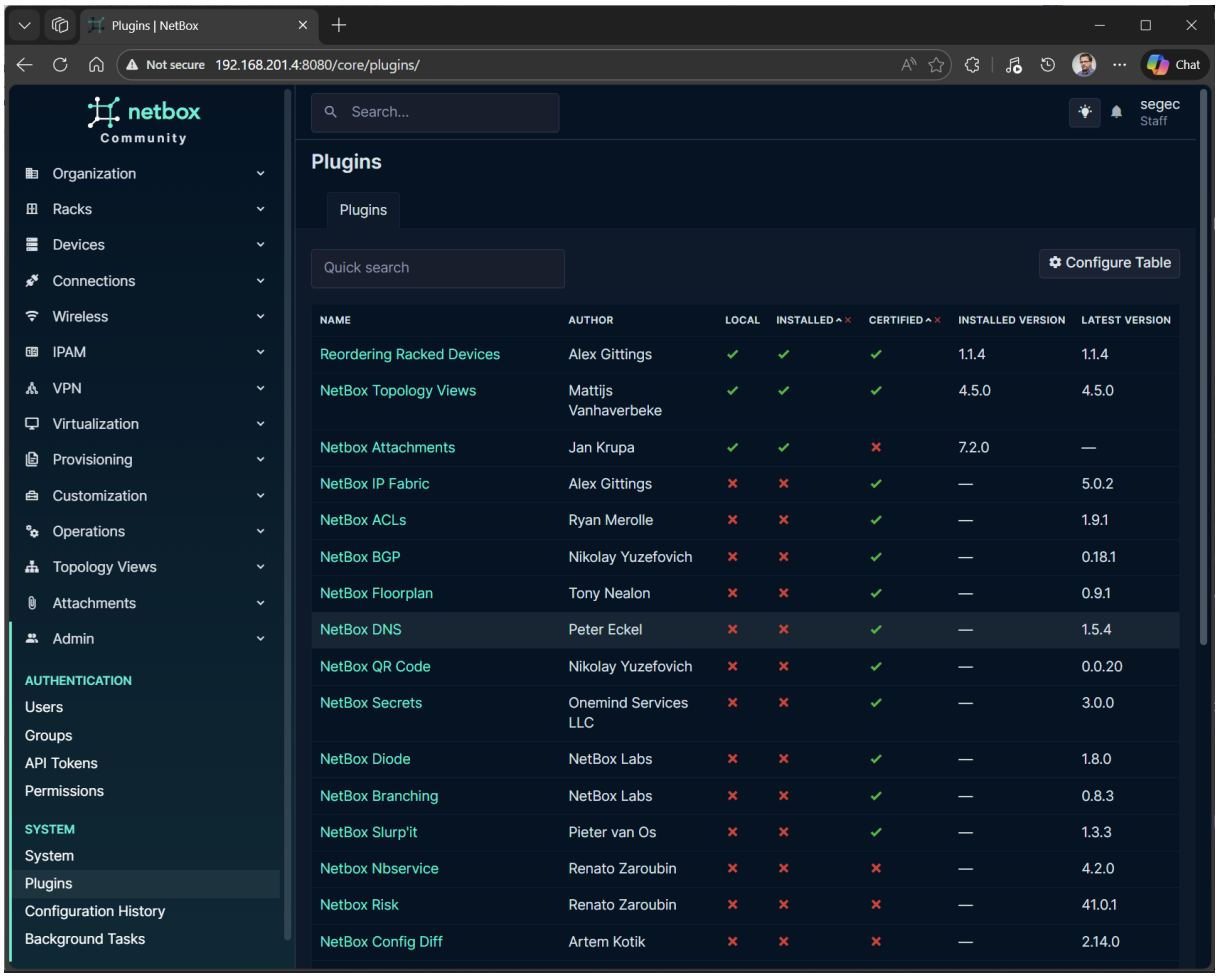
2.8.2 Nasadenie a základná konfigurácia NetBoxu

V rámci PoC bol NetBox nasadený v prostredí OpenStack ako samostatná služba určená pre evidenciu a správu infraštruktúrnych údajov. Z pohľadu implementácie bol riešený kontajnerizačne, pričom boli využité aplikačné a podporné komponenty potrebné na jeho beh, najmä databázová vrstva a cache/queue služby. Takýto spôsob nasadenia zodpovedá charakteru experimentálneho nasadenia a zároveň umožňuje jednoduchšiu správu, aktualizáciu a opakovateľné nasadenie.

Základná konfigurácia NetBoxu bola v PoC zameraná na sprístupnenie webového rozhrania a na prípravu dátového modelu pre evidenciu vybraných objektov infraštruktúry. Konfiguračne bol systém pripravený na prácu s databázou PostgreSQL a s oddelenou Redis vrstvou pre úlohy a cache. V rámci PoC boli do systému zavedené základné objekty, najmä lokality, zariadenia, adresné priestory a súvisiaci infraštruktúrny kontext.



Obrázok 1 Informácie o systéme



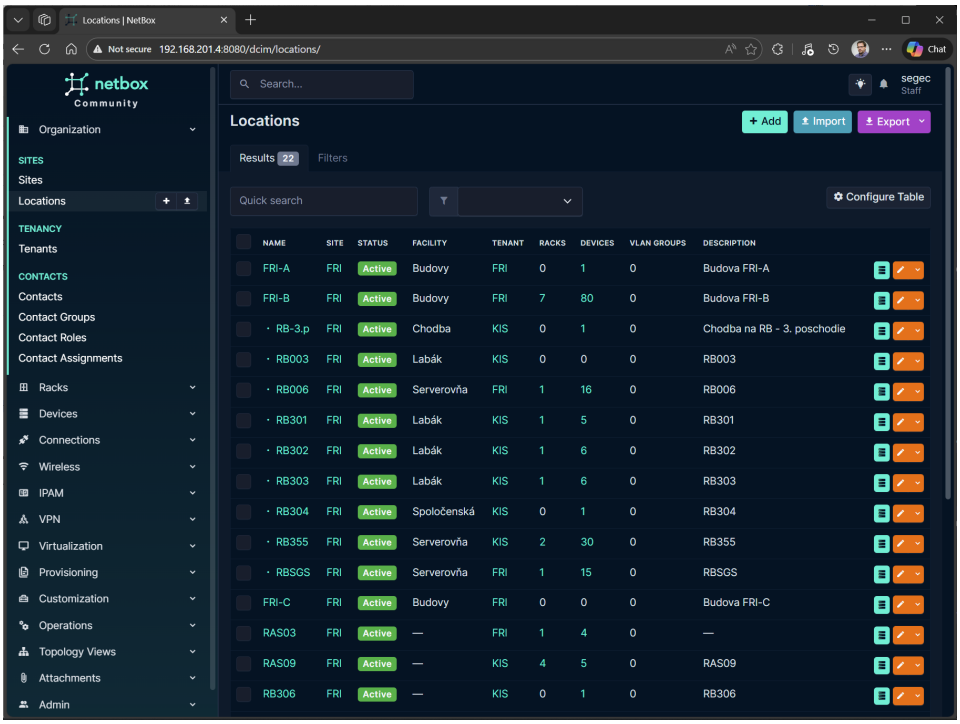
Obrázok 2 Pluginy

Cieľom nebolo vytvoriť úplnú produkčnú CMDB alebo detailný model celej siete, ale pripraviť reprezentatívnu množinu údajov, na ktorej je možné demonštrovať využitie NetBoxu v navrhovanom riešení.

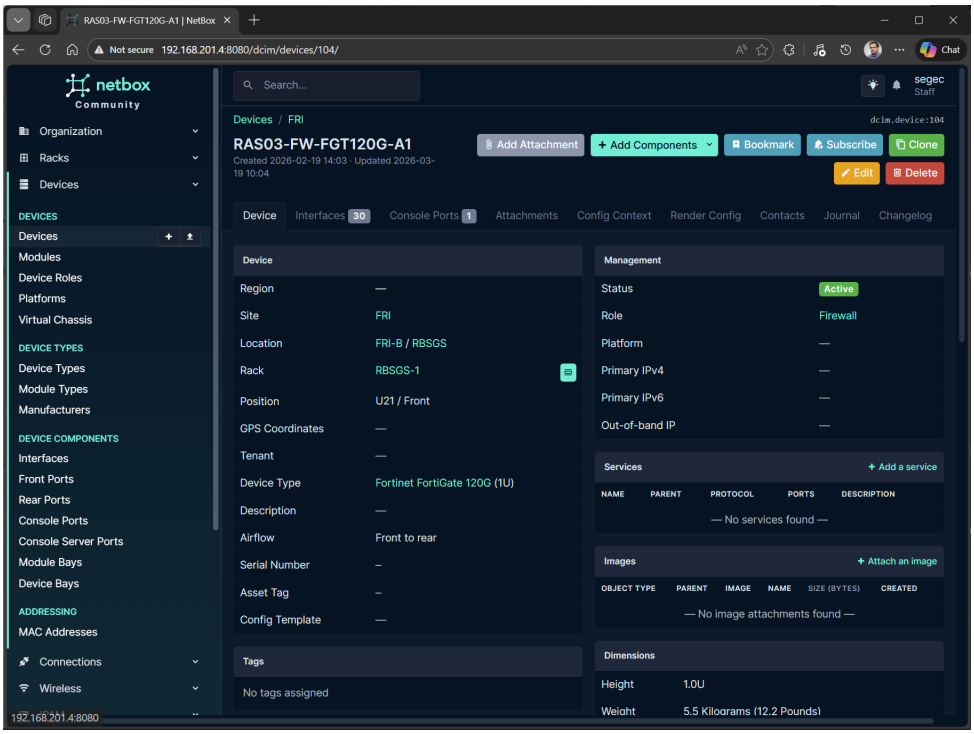
2.8.3 Využitie NetBoxu v rámci PoC

V rámci PoC bol NetBox využívaný ako podporný zdroj infraštruktúrneho kontextu pre ostatné časti riešenia. Jeho hlavný prínos spočíval v tom, že umožňoval dohľadanie základných informácií o zariadeniach, IP adresácii, lokalitách a sieťových segmentoch, ku ktorým bolo možné priradiť udalosti alebo zistenia z ďalších nástrojov.

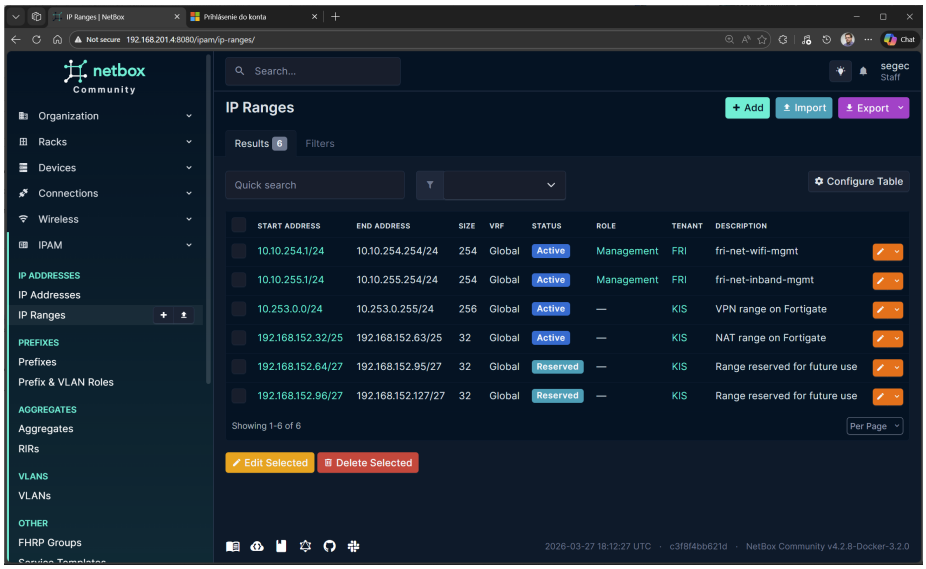
Praktické využitie NetBoxu v tejto etape bolo orientované najmä na evidenciu a manuálne alebo čiastočne podporené dohľadanie kontextu. NetBox tak neplnil úlohu detekčného ani reakčného nástroja, ale dopĺňal riešenie o údaje potrebné na lepšie pochopenie technického prostredia. V tomto rozsahu jeho nasadenie potvrdilo, že ide o vhodný doplnok PoC riešenia a o použiteľný základ pre ďalšie rozšírenie integračných scenárov.



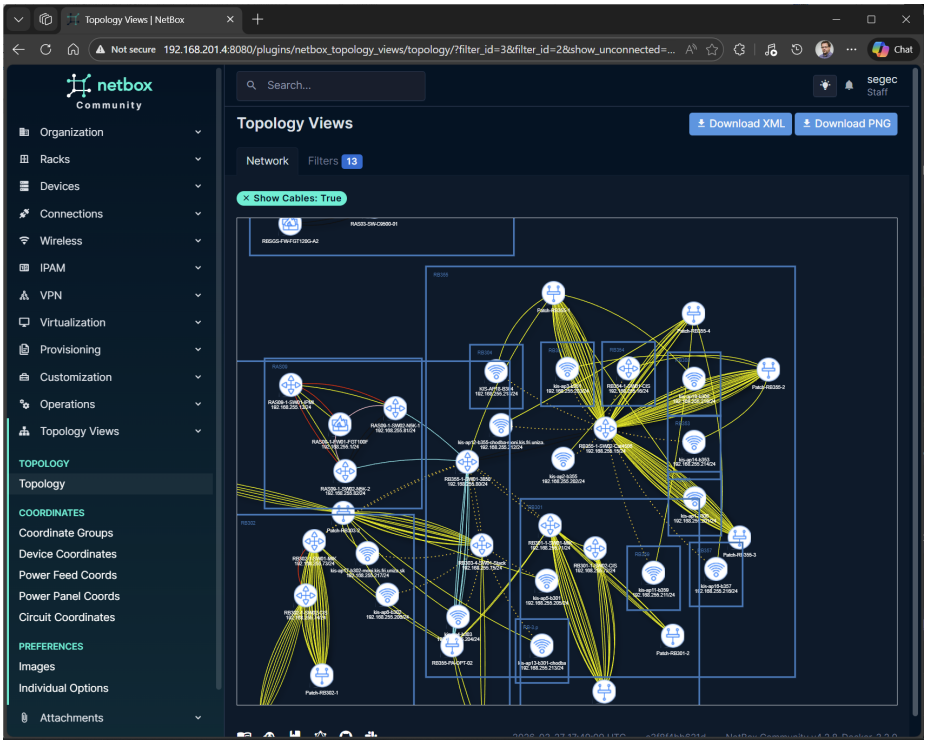
Obrázok 3 Modelovanie organizácie

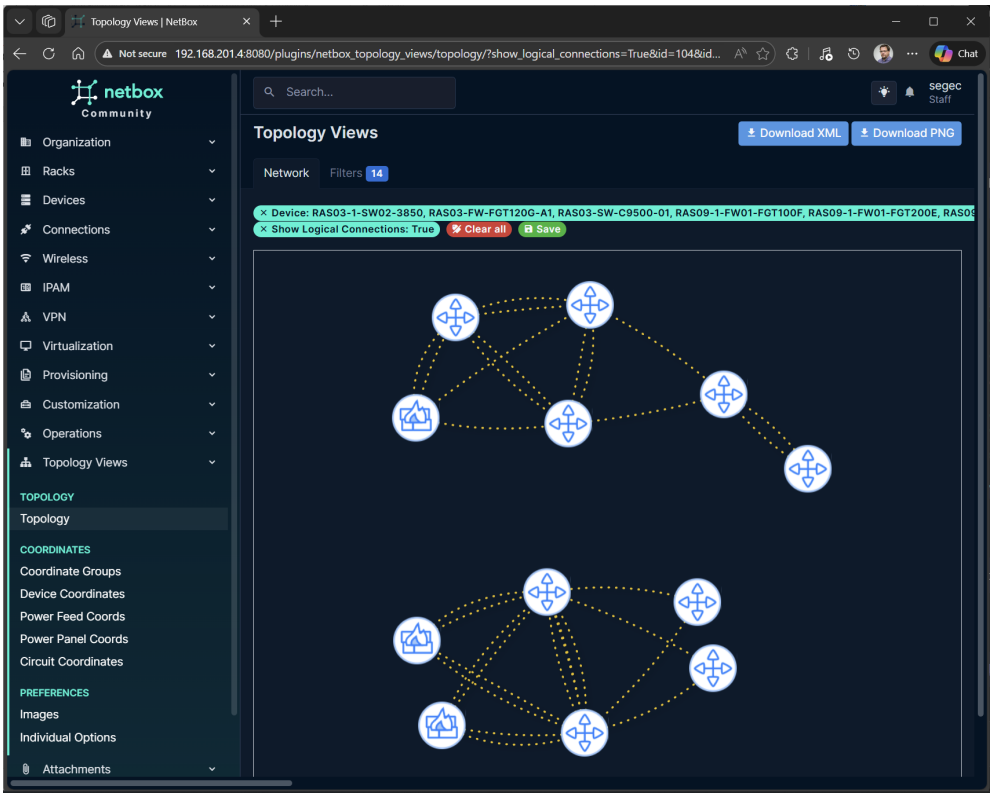


Obrázok 4 Modelovanie zariadenia – Firewallu

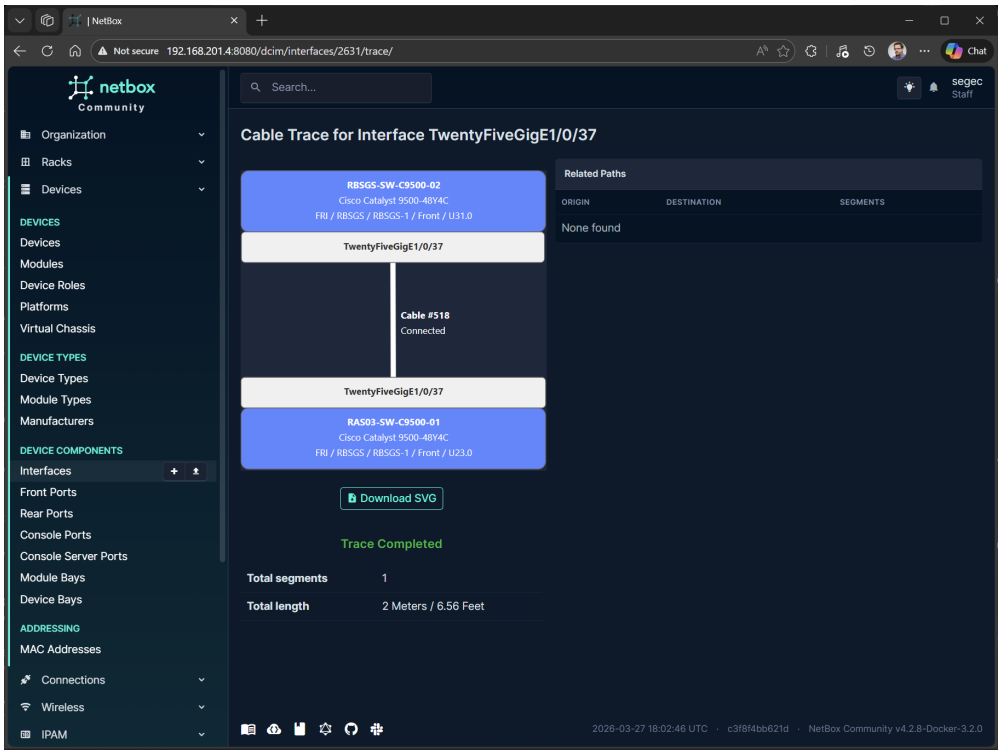


Obrázok 5 Manažment adries





Obrázok 6 Generovanie topológií



Obrázok 7 Trasovanie kabeláže

2.8.4 Aktuálny stav a obmedzenia NetBoxu v PoC

NetBox bol v rámci PoC úspešne nasadený a základne nakonfigurovaný ako funkčný nástroj pre evidenciu infraštruktúrnych údajov. V tomto stave poskytuje použiteľný základ pre správu vybraných objektov a pre orientáciu v prostredí navrhovaného riešenia. Z pohľadu PoC sa potvrdilo, že NetBox je vhodný na doplnenie SOC riešenia o infraštruktúrny kontext a na podporu práce s aktívami a adresáciou.

Na druhej strane je potrebné uviesť, že jeho využitie v PoC ostalo na úrovni základnej evidencie a čiastočného kontextového doplnenia. Neboli dotiahnuté všetky plánované integračné väzby voči ostatným komponentom riešenia a NetBox tak v tejto etape nefunguje ako plne automatizovaný zdroj enrichment údajov. Jeho úloha je zatiaľ skôr podporná a pripravuje základ pre ďalšie rozšírenie v budúcich etapách riešenia.

Z tohto dôvodu je NetBox v PoC potrebné chápať ako funkčný, ale zatiaľ nie plne rozvinutý komponent. Napriek tomu jeho nasadenie preukázalo praktickú využiteľnosť v navrhovanom riešení a potvrdilo vhodnosť jeho zaradenia do širšej architektúry SOC služby.

2.8.5 DFIR IRIS

DFIR-IRIS je v rámci PoC riešenia INNOSOC využitý ako open-source nástroj určený na riadenie a evidenciu bezpečnostných incidentov (case management). V rámci nášho PoC slúži ako centrálne miesto, kde sú prijímané alerty zo systému Wazuh a následne spracovávané ako incidenty (cases). Umožňuje evidenciu artefaktov, poznámok, úloh, zmenu stavov incidentov a koordináciu práce analytikov.

2.8.6 Model nasadenia DFIR IRIS v prostredí OpenStack

Systém je nasadený v prostredí OpenStack na virtuálnom stroji s operačným systémom Ubuntu 24.04. DFIR-IRIS beží v kontajneroch pomocou nástroja Docker Compose, čo umožňuje jednoduché nasadenie, správu aj škálovanie riešenia.

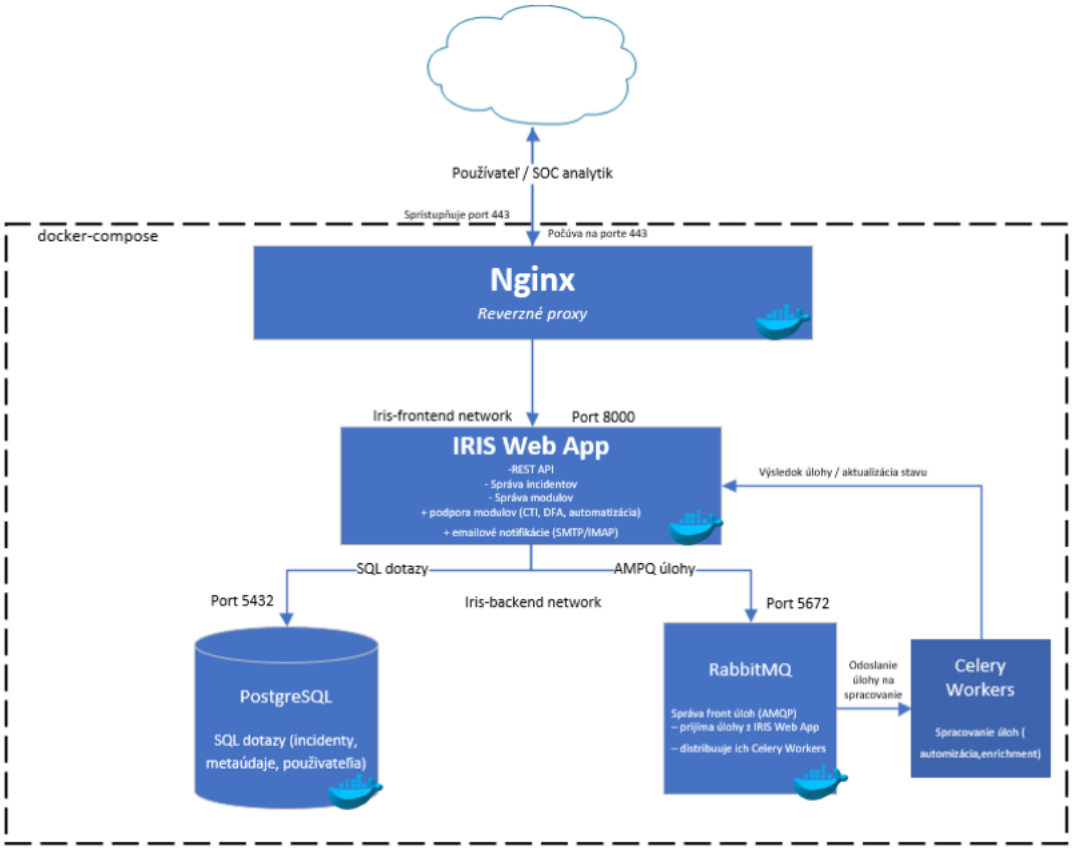
Technická špecifikácia nasadenia DFIR-IRIS:

- Prostredie – OpenStack
- Meno inštancie – ubuntu-24.04-kis
- IP adresa – 192.168.20.45 (Interná IP v Openstacku), 192.168.4.97 (Externá IP)
- vCPU – 4
- RAM – 8 GB
- Disk – 40 GB
- OS – Ubuntu 24.04

Riešenie je postavené na kontajnerovej architektúre, kde jednotlivé komponenty (webové rozhranie, backend a databázová vrstva) bežia oddelene v Docker kontajneroch. Spustenie celej aplikácie je realizované pomocou príkazu:

docker compose up -d

DFIR-IRIS poskytuje webové rozhranie, cez ktoré analytik pracuje s incidentmi, pričom backend zabezpečuje spracovanie logiky aplikácie a komunikáciu s databázou, kde sú ukladané všetky incidenty, artefakty a história zmien



Obrázok 2 Architektúra DFIR-IRIS

Obrázok znázorňuje architektúru nástroja DFIR-IRIS a spracovanie incidentov v rámci INNOSOC prostredia. Riešenie je založené na kontajnerizovanej architektúre, kde jednotlivé komponenty komunikujú prostredníctvom interných služieb.

Frontend predstavuje webové rozhranie pre analytikov, ktoré slúži na prácu s incidentmi (cases), artefaktami a úlohami. Požiadavky z rozhrania sú spracované backendom, ktorý zabezpečuje aplikačnú logiku a riadenie workflow incidentov. Backend komunikuje s databázovou vrstvou, kde sú uložené všetky informácie o incidentoch vrátane stavu, používateľov, artefaktov a časovej osi. Táto architektúra umožňuje konzistentnú evidenciu a spätné dohľadanie vykonaných operácií.

2.8.7 Základné integračné väzby komponentu DFIR IRIS v INNOSOC

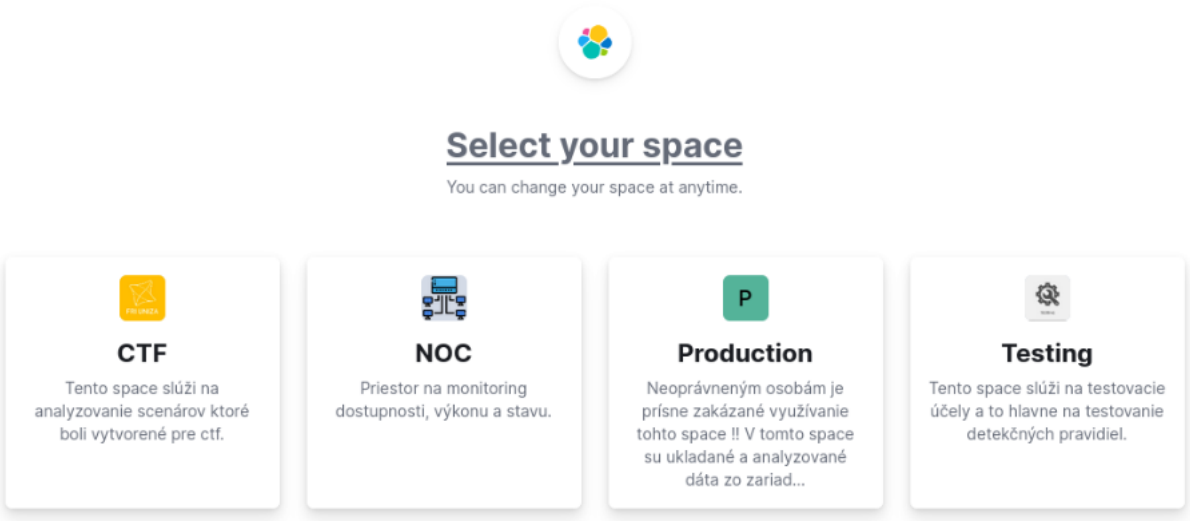
DFIR-IRIS podporuje integráciu pomocou REST API a webhookov. V rámci PoC je prepojený so systémom Wazuh, z ktorého prijíma alerty vo formáte JSON a spracováva ich ako incidenty alebo aktualizácie existujúcich prípadov. Systém Shuffle je s DFIR-IRIS prepojený a bude slúžiť ako automatizačná vrstva pre operácie nad incidentmi, ako napríklad enrichment, aktualizácia prípadov alebo spúšťanie reakcií. Toto prepojenie je implementované a jeho funkčnosť bola overovaná testovaním.

Aktuálne prepojené a otestované služby pomocou DFIR-IRIS:

- Wazuh – príjem alertov a vytváranie incidentov (JSON / webhook)
- Shuffle – automatizácia a aktualizácia incidentov pomocou API

2.9 Kibana (ELK)

Kibana je v rámci PoC riešenia INNOSOC využitá ako grafické rozhranie v ELK stack. Na FRI UNIZA je nasadené na url adrese v doméne FRI. Logické delenie pravidiel, prístupov a iné, je realizované pomocou Spaces. Nasadené máme nasledovné Spaces. Na obrázku je vidieť aj ich úlohu / využitie.



Hlavné funkcie Kibany využité v PoC:

- Discover – prezeranie záznamov,
- Dashboards (Analytics) – zobrazovanie záznamov do prehľadných grafov,
- Dashboards (Security) - fungovanie pravidiel,
- Rules - správa detekčných pravidiel,
- Alerts,
- Dev Tools - konzola na správu,
- Stack Monitoring,
- Stack Management.

V Stack Managment sa spravuje celý stack.

2.9.1 Vytvorené dashboardy

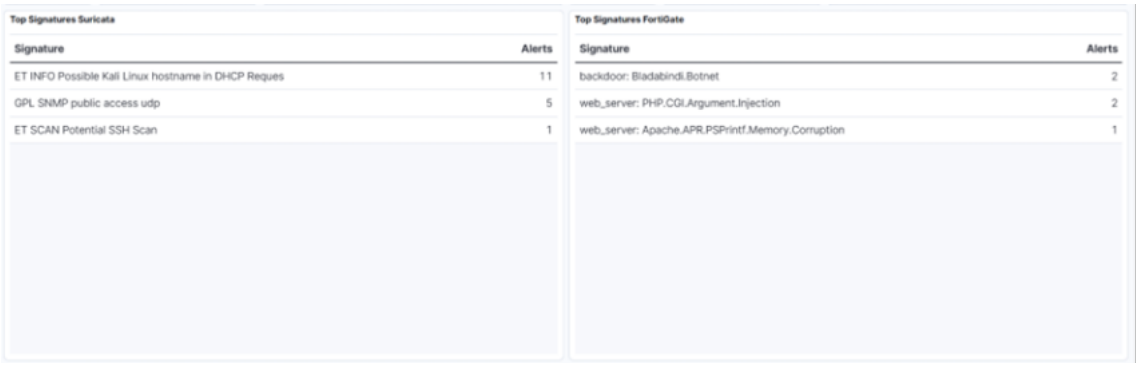
Dashboardy v Kibane slúžia na vizualizáciu dát z Elasticsearch pomocou grafov, máp a filtrov, čím poskytujú prehľad o sledovaných údajoch.

Vytvorené dashboardy:

- Suricata UNIZA
- Arkime UNIZA
- F_kibana_dashboards
- Noc výskumníkov
- IDS DMZ: Vizualizuje dáta z IDS senzorov v demilitarizovaných zónach a umožňuje porovnávať detekcie medzi nástrojmi Suricata a Fortigate.
- Login Activity Monitoring: Zameriava sa na úspešné a neúspešné prihlásenia
- Zaťaženie Stacku: Slúži na monitorovanie využitia operačnej pamäte jednotlivými uzlami Elasticsearch, Logstash a Kibany



Dashboard IDS DMZ, alerty



Dashboard IDS DMZ, top signatúry



Dashboard IDS DMZ, top source IPs



Dashboard IDS DMZ, mapa

2.9.2 Technické parametre a overenie funkčnosti

Pre nástroj Kibana sme pri nasadení definovali maximálnu veľkosť operačnej pamäte, s ktorou môže pracovať. Definovali sme 8 GB. Kibana nemá predvolene definované obmedzenie operačnej pamäte, pridelením obmedzenia sme zabezpečili, že Kibana nealokuje všetku pamäť pre svoje využitie. 8 GB sme zvolili pre poskytnutie dostatočných pamäťových zrojov pri zobrazovaní zložitých vizualizácií.

Kibana je schopná overiť TLS certifikát nástroja Elasticsearch, čo zabezpečuje šifrovanú komunikáciu. Elasticsearch si nástroj Kibana pri pripojení overuje pomocou mena a hesla, sú využité predvolené používateľské účty. Autentifikácia sa konfiguruje v hlavnom konfiguračnom súbore nástroja Kibana. Pre zvýšenie bezpečnosti nie je heslo predvoleného používateľa uložené v tzv. „plaintaxt“ podobe, ale pre uloženia hesla využívame keystore, obdobne ako aj pri ostatných nástrojoch. Sieťová komunikácie je zabezpečená pomocou nástroja iptables, ktorým sme definovali firewall pravidlá. Pre prihlásenie do UI nástroja kibana sa používajú používateľské účty s rolami na to určenými, tzv. riadenie prístupu na základe rolí (RBAC).

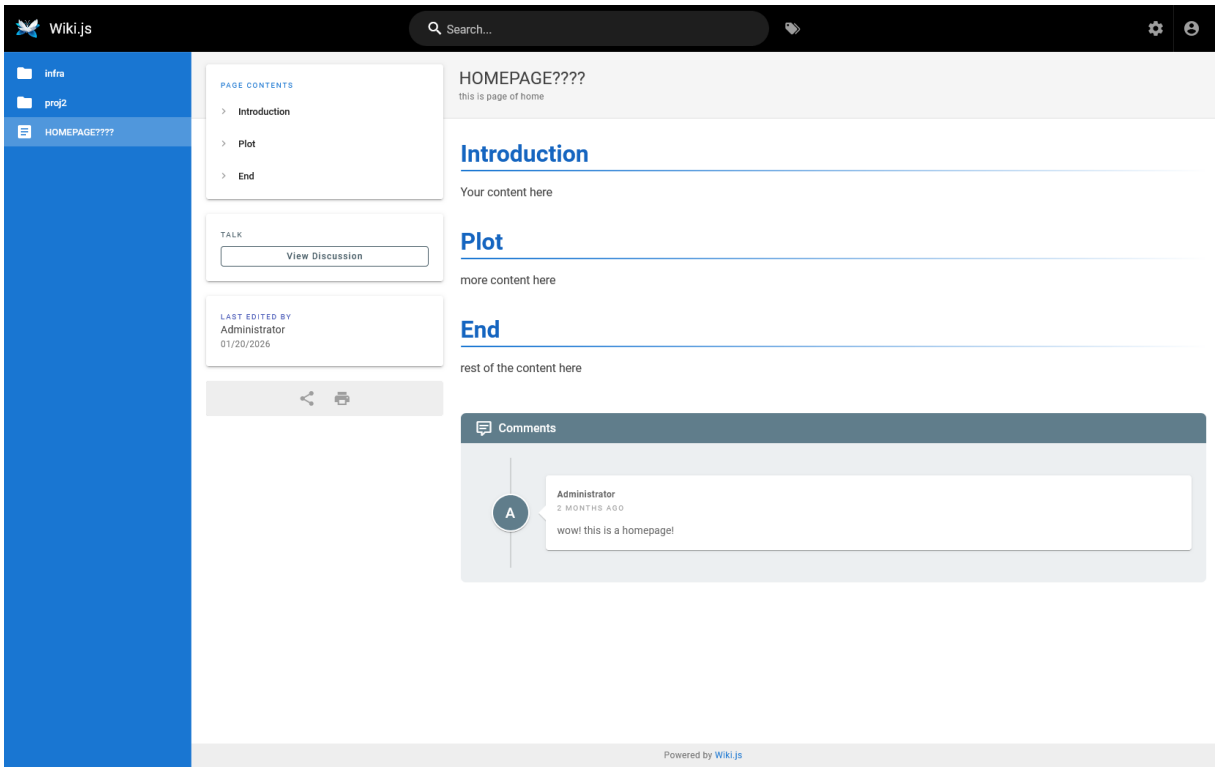
Kibana je monitorovaná nástrojmi Zabbix, LibreNMS a taktiež pomocou metricbeat. Takýmto monitorovaním sledujeme dostupnosť a stabilitu danej služby.

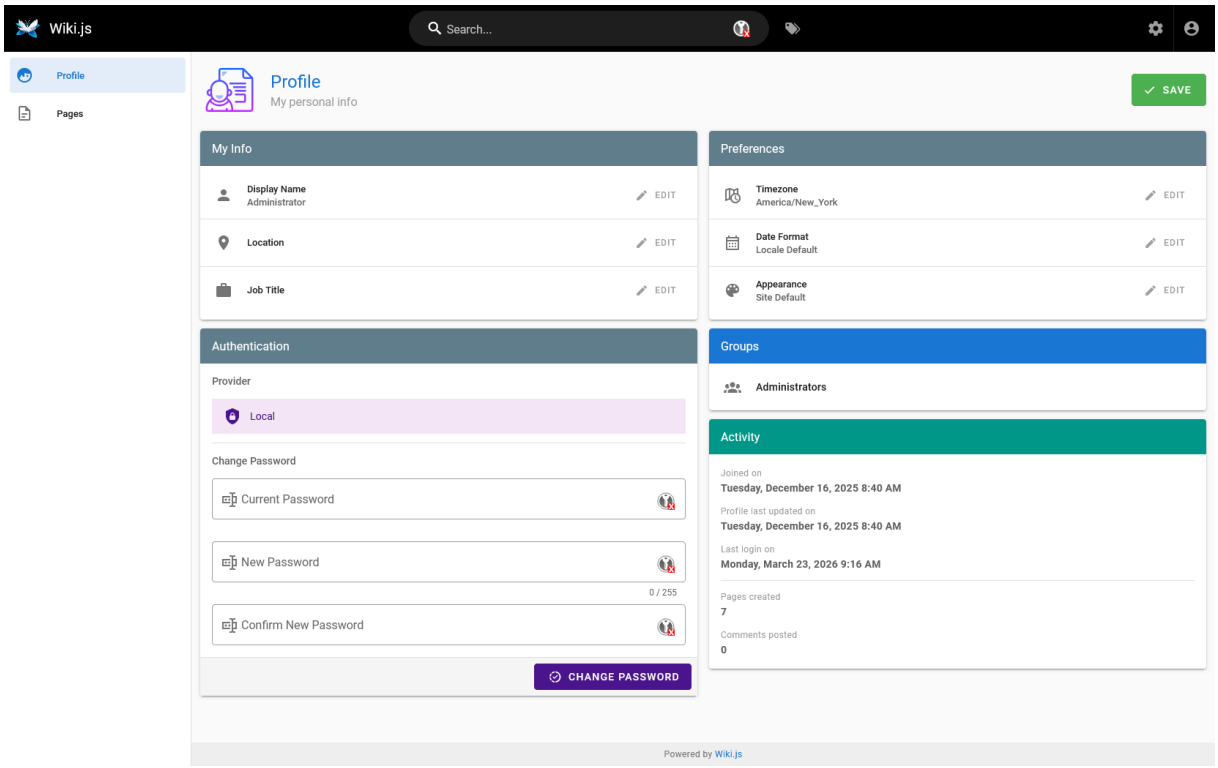
Služba Kibana taktiež beží vo svojom LXC kontajnery na rovnakom fyzickom servery ako zvyšok ELK staku. Kontajner má IP adresu 192.168.2.188, ktorá je dostupná iba zo špecifického IP rozsahu. V adresáry Kibana sa nachádza konfiguračný súbor, rovnako ako aj certifikáty s certifikačnou autoritou elasticsearch na zabezpečenie šifrovanej komunikácie. Ukážka konfiguračného súboru kibana.yml:

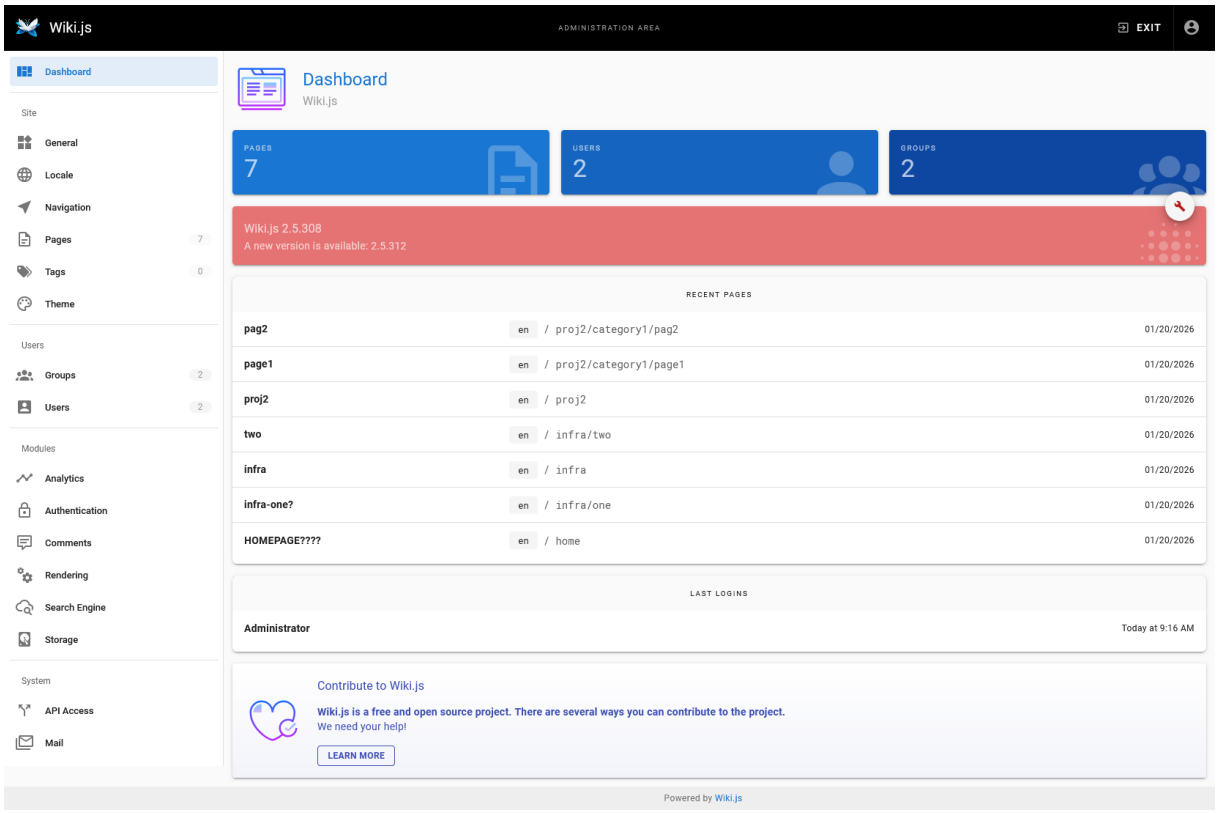
```
server.port: 5601
server.host: "0.0.0.0"
server.publicBaseUrl: "https://[interná doména]"
server.ssl.enabled: false
elasticsearch.hosts: ["https://192.168.2.178:9200"]
elasticsearch.username: "kibana_system"
```

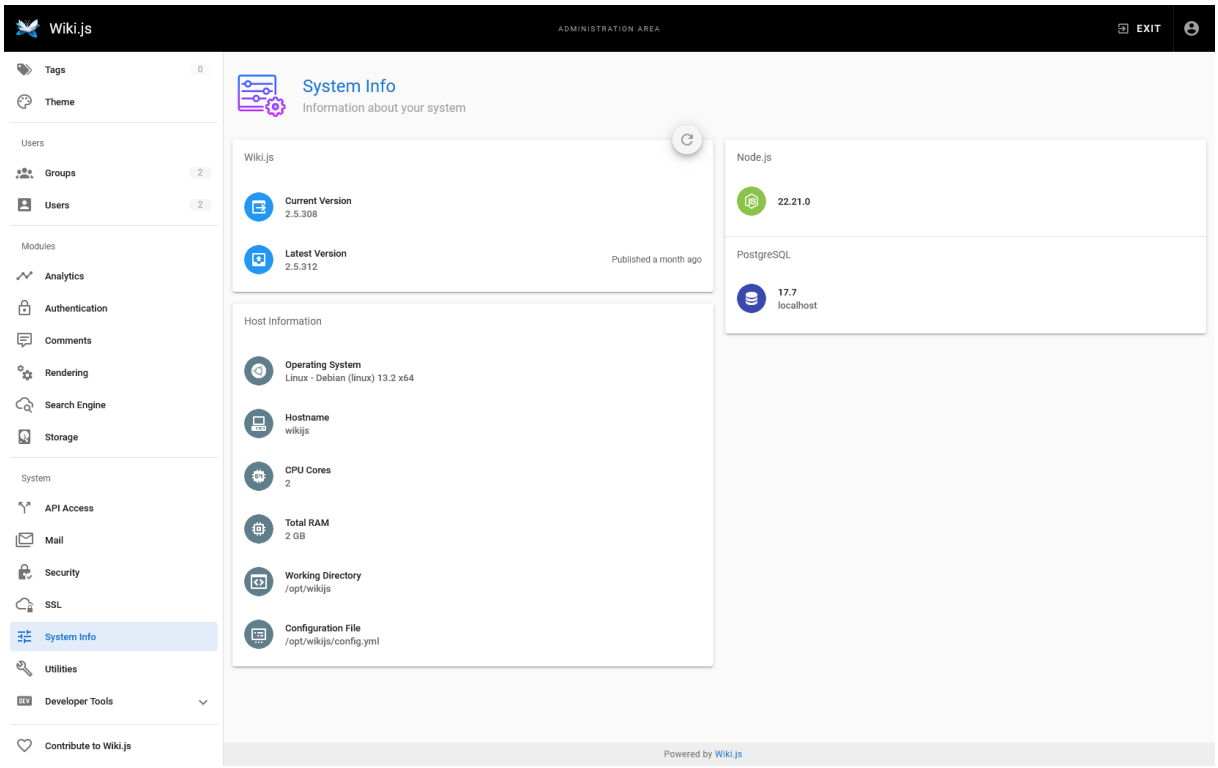
```
elasticsearch.requestTimeout: 180000
elasticsearch.ssl.certificateAuthorities:                                    [
"/etc/kibana/certs/kibana/elasticsearch-ca.pem" ]
logging:
  appenders:
    file:
      type: file
      fileName: /var/log/kibana/kibana.log
      layout:
        type: json
  root:
    appenders:
      - default
      - file
pid.file: /run/kibana/kibana.pid
xpack.encryptedSavedObjects.encryptionKey: [key1]
xpack.reporting.encryptionKey: [key2]
xpack.security.encryptionKey: [key3]
monitoring.kibana.collection.enabled: false
monitoring.ui.enabled: true
monitoring.ui.elasticsearch.hosts: ["https://192.168.2.178:9200"]
     monitoring.ui.ccs.enabled: false
```

2.10 Wiki.js




Wiki.js
Search...
Profile
Pages
Profile
My personal info
SAVE
My Info
Display Name
Administrator
EDIT
Location
EDIT
Job Title
EDIT
Preferences
Timezone
America/New_York
EDIT
Date Format
Locale Default
EDIT
Appearance
Site Default
EDIT
Authentication
Provider
Local
Change Password
Current Password
New Password
0 / 255
Confirm New Password
CHANGE PASSWORD
Groups
Administrators
Activity
Joined on
Tuesday, December 16, 2025 8:40 AM
Profile last updated on
Tuesday, December 16, 2025 8:40 AM
Last login on
Monday, March 23, 2026 9:16 AM
Pages created
7
Comments posted
0
Powered by Wiki.js


Wiki.js
ADMINISTRATION AREA
EXIT
Dashboard
Site
General
Locale
Navigation
Pages
Tags
Theme
Users
Groups
Users
Modules
Analytics
Authentication
Comments
Rendering
Search Engine
Storage
System
API Access
Mail
Dashboard
Wiki.js
PAGES
7
USERS
2
GROUPS
2
Wiki.js 2.5.308
A new version is available: 2.5.312
RECENT PAGES
pag2 en / proj2/category1/pag2 01/20/2026
page1 en / proj2/category1/page1 01/20/2026
proj2 en / proj2 01/20/2026
two en / infra/two 01/20/2026
infra en / infra 01/20/2026
infra-one? en / infra/one 01/20/2026
HOMEPAGE???? en / home 01/20/2026
LAST LOGINS
Administrator Today at 9:16 AM
Contribute to Wiki.js
Wiki.js is a free and open source project. There are several ways you can contribute to the project.
We need your help!
LEARN MORE
Powered by Wiki.js



Plus v zvyšnej dokumentácii je aj nejaký screen z nastavenia emailov alebo čosi také myslím...

3 NASADENIE A ZÁKLADNÁ KONFIGURÁCIA POC

Tu stručne opísať, ako bol PoC nasadený v OpenStack prostredí, ako boli použité VM a kde bol použitý Docker. Neísť do detailov ako vo V14, len zhrnúť reálny stav.

Podla mna sme sa vytrieskali z obsahu vyssie – pri kazdom komponente mame aj model nasadenia, aj pouzite irtualne stroje a sluzby, aj zakladne integracne vazby ... tu neviem ci nieco stvorime este... nebude sil.... Podla mna to mame hore.. Pri komponentoch

3.1 Model nasadenia

3.2 Použité virtuálne stroje a služby

3.3 Základné integračné väzby

4 ZÁVER

4.1 XX