Střední průmyslová škola na Proseku
Novoborská 610/2, 190 00 Praha 9

OBOR

18-20-M/01 INFORMAČNÍ TECHNOLOGIE

ŠVP: Správa sítí a IT bezpečnost

MATURITNÍ PROJEKT

TÉMA PRÁCE

Centralizovaný logserver a detekce bezpečnostních událostí

2025/2026

Thomas Shon

PROHLÁŠENÍ

Prohlašuji, že jsem svou maturitní práci vypracoval samostatně a použil jsem pouze podklady (literaturu, projekty, SW, atd.) uvedené v seznamu.

Nemám závažný důvod proti užití tohoto školního díla ve smyslu § 60 zákona č. 121/2000 Sb., o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon).

V Praze dne XX. X. 2026

……

podpis

PODĚKOVÁNÍ

Rád bych touto cestou poděkoval vedoucímu mé maturitní práce Ing. Jiřímu Šilhánovi za odborné vedení a věcné připomínky, které mi pomohly ke zdárnému dokončení tohoto projektu. Dále děkuji panu učiteli Ondřeji Němcovi za jeho cenné rady, připomínky a motivaci v průběhu celého zpracování.

Můj velký dík patří panu Tomáši Lonskému, mému kolegovi z CENDIS s.p., který zde zastupuje roli bezpečnostního analytika. Velmi si vážím toho, že mě díky svým bohatým zkušenostem s těmito systémy dokázal vždy správně nasměrovat ve chvílích, kdy to bylo potřeba.

V neposlední řadě děkuji své rodině za jejich neutuchající morální podporu během celého studia. Zvláštní poděkování si zaslouží má starší sestra za její důslednou diakritickou kontrolu textu a neocenitelnou pomoc s formálními náležitostmi této práce.

ANOTACE

Jméno autora	Thomas Shon
Název maturitní práce	Centralizovaný logserver a detekce bezpečnostních událostí
Rozsah práce	50s.
Školní rok vyhotovení	2025/2026
Škola	Střední průmyslová škola na Proseku
Vedoucí práce	Ing. Jiří Šilhán
Využití práce	Projekt slouží jako praktická metodika pro administrátory, kteří potřebují implementovat efektivní monitorování infrastruktury bez vysokých nákladů na komerční SIEM řešení. Díky integraci honeypotu systém navíc poskytuje cenná data pro analýzu aktuálních hrozeb. Práce může dále sloužit jako studijní materiál pro studenty oboru IT se zaměřením na síťovou bezpečnost a správu serverů.
Abstrakt	Tato maturitní práce se věnuje problematice centralizovaného sběru a analýzy systémových a bezpečnostních logů v rámci počítačové sítě. Hlavním cílem bylo navrhnout a realizovat monitorovací systém založený na open-source platformě Graylog, který umožňuje efektivní detekci anomálií a bezpečnostních incidentů. Teoretická část definuje koncept SIEM a popisuje architekturu použitých technologií, včetně databází MongoDB a OpenSearch. Praktická část detailně popisuje instalaci systému na platformu Ubuntu, konfiguraci šifrovaného přenosu dat a nasazení agentů Beats na koncová zařízení. Významným prvkem je integrace SSH honeypotu Cowrie pro sběr dat o útocích a následná vizualizace těchto dat v přehledných

	dashboardech. Práce také řeší automatizaci procesů pomocí Pipelines pro obohacování dat o geografické údaje útočníků.
Klíčová slova	Graylog, centralizovaná správa logů, kybernetická bezpečnost, Pipeline, honeypot, monitorování sítě, detekce incidentů.

ANNOTATION

Autor	Thomas Shon
Title of graduation work	Centralized log server and security event detection
Extend	50p.
Academic year	2025/2026
School	Střední průmyslová škola na Proseku
Supervisor	Ing. Jiří Šilhán
Application	The project serves as a practical methodology for administrators who need to implement effective infrastructure monitoring without the high costs of commercial SIEM solutions. Thanks to the integration of a honeypot, the system also provides valuable data for analyzing current threats on the Internet. The work can also serve as study material for IT students focusing on network security and server administration.
Abstrakt	This graduation thesis focuses on the issue of centralized collection and analysis of system and security logs within a computer network. The main objective was to design and implement a robust monitoring system based on the open-source Graylog platform, enabling efficient detection of anomalies and security incidents. The theoretical part defines the SIEM concept and describes the architecture of the technologies used, including MongoDB and OpenSearch databases. The practical part details the system installation on the Ubuntu platform, configuration of encrypted data transmission, and deployment of Beats agents on end devices. A significant element is the integration of the Cowrie SSH honeypot for collecting attack data and the

	subsequent visualization of this data in clear dashboards. The thesis also addresses process automation using Pipelines for enriching data with attackers' geographic information.
Key words	Graylog, centralized log management, cybersecurity, Pipeline, honeypot, Cowrie, network monitoring, incident detection

Obsah

1 Úvod

V dnešní rychlé době, kdy už téměř každý zaměstnanec vlastní mobilní telefon nebo jinou chytrou elektroniku a stává se tak součástí tohoto obrovského ekosystému, jsou bez pochyb informační technologie považovány za jeden ze základních stavebních kamenů většiny firem a jejich firemních procesů. Jelikož dochází k rapidnímu nárůstu využívání těchto moderních technologií během každodenních procesů uvnitř firem, je potřeba tento faktor zohlednit a zvýšit tak nároky na dohled, správu a zabezpečení všech prvků v celé firemní infrastruktuře. Absolutně klíčovou roli zde zaujímají logy, které poskytují velice cenné, a také obzvláště důležité informace o stavu systémů. Důsledná práce s nimi je tak nesmírně důležitá pro bezpečný a plynulý chod firmy a všech procesů uvnitř, ale i mimo ni.

Na základě těchto okolností si autor zvolil jako téma své maturitní práce návrh, a také následnou realizaci centralizovaného logserveru, který však bude disponovat i prvky systému pro správu bezpečnostních událostí. Hlavní cílem projektu je vytvořit řešení, které bude schopné ze serverů s operačními systémy Windows i Linux logy shromažďovat, filtrovat, a také vyhodnocovat. Tyto informace budou poté nejen ukládány, ale také analyzovány a vizualizovány pomocí webového rozhraní s jasně přehlednými dashboardy.

Hlavní motivací k uskutečnění tohoto projektu jsou pro autora jeho zkušenosti z pracovního poměru. Již více než rok působí autor ve firmě jako součást týmu kybernetické bezpečnosti, kde má možnost pozorovat každodenní fungování IT infrastruktury a jiných úkonů spojených s kyberbezpečností. Události z interních firemních serverů jsou monitorovány individuálně dle bezpečnostních politik firmy a následné vyhodnocování je tedy složitější a časově náročné. Při odhalování bezpečnostních incidentů tudíž dochází ke zpoždění, protože aktuálně neexistuje ve firmě řešení, jak mít vše přehledně na jednom místě.

Autor tedy vidí v uskutečnění projektu nejen možnost rozšíření svých znalostí ze studia na střední škole, ale také příležitost propojit tyto zkušenosti s praxí, která by měla značný přínos pro autorovo pracovní prostředí. Centralizovaný logserver by ve firmě velmi usnadnil dohled nad interními servery a díky tomu by urychlil případnou reakci na potenciální bezpečnostní problémy a zároveň zvýšil celkovou úroveň bezpečnosti a integrity firmy. Projekt tedy není pouze školní maturitní prací, ale zároveň představuje reálné řešení nedostatku, se kterým se autor během svého dosavadního působení ve firmě setkal.

Metodika realizace vychází z architektury klient–server. Jako hlavní nástroj bude využit Graylog, který zajistí sběr a zpracování logů i jejich následnou vizualizaci. O ukládání dat se postará Graylog Data Node s integrovaným vyhledávacím enginem OpenSearch, zatímco metadata a konfigurace budou spravovány prostřednictvím MongoDB.

Pilotní verze projektu bude implementována na virtuálním serveru s Debianem, což poskytne stabilní a uživatelsky přívětivé prostředí pro testování. Autor se dále zaměří na pravidelné zálohování a konfiguraci základních bezpečnostních pravidel pro detekci podezřelých událostí, jako jsou například opakovaná neúspěšná přihlášení nebo přihlášení mimo pracovní dobu. Výsledkem projektu tedy bude funkční systém, který prokáže, že pomocí open source nástrojů lze vytvořit efektivní a spolehlivé řešení, které je připravené pro nasazení v různých prostředích, od menších po rozsáhlé infrastrukturní systémy.

2 Použité technologie

V této kapitole se autor zaměřuje na podrobnější popsání a vysvětlení konkrétních technologií, které byly použity při realizaci projektu.

2.1 Operační systémy

Při návrhu a následné realizaci centralizovaného logserveru bylo zapotřebí vybrat spolehlivé a také především stabilní prostředí, které by dlouhodobě umožnilo bezproblémový provoz všech potřebných součástí. Pro logserver autorova projektu se jako nejlepší řešení ukázal operační systém Linux, konkrétně se zvolením distribuce Debian. Tato linuxová distribuce je díky své vysoké stabilitě, otevřenému vývoji a dlouhodobé podpoře široce využívána v serverovém prostředí, a také nabízí uživatelům vysokou míru bezpečnosti, jednoduchou správu prostřednictvím příkazového řádku a široce rozsáhlou komunitní podporu. Díky těmto všem faktorům je Debian ideální linuxovou distribucí pro provoz klíčových prvků, jako je Graylog, MongoDB nebo OpenSearch, které společně tvoří jádro celého projektu.

Kromě sběru logů z Linuxových serverů bylo nutné brát v potaz i operační systémy Windows, které tvoří značnou část firemní infrastruktury, a i právě z těchto systémů je potřeba shromažďovat systémové a bezpečnostní logy, které autorovi poskytují potřebný přehled o činnosti uživatelů ale i aplikací. Systémy Windows využívají svůj velice specifický mechanismus pro uchování událostí, pojmenovaný jako Windows Event Log, a proto pro jejich

přenos na centrální logserver autor využívá nástroj Winlogbeat. O tom ale více v samostatné kapitole, nástroje pro sběr a přenos logů. Tento nástroj umožní okamžité odesílání záznamů z prostředí Windows přímo na centrální server do Graylogu, ve kterém dochází k jejich dalšímu zpracování.

Výsledná práce tak kombinuje dva různé operační systémy a stává se tedy multiplatformním řešením, které je schopné a připravené efektivně zpracovávat logy napříč prostředími, díky čemuž umožňuje komplexní dohled nad celou firemní infrastrukturou.

2.2 Hlavní nástroj – Graylog

Graylog je bezpochyby nejdůležitější technologií tohoto maturitního projektu a tvoří samotné jádro celého řešení. Jedná se o open-source systém pro centralizovanou správu, analýzu a také vizualizaci logů. Graylog byl vytvořen s cílem efektivně shromažďovat a zpracovávat logy z různorodých zdrojů, od serverů, přes síťové prvky až po samotné aplikace. Hlavním přínosem je snadná orientace v systému, možnost rozšíření a okamžitý přehled o infrastrukturních procesech. Díky tomu je ideálním nástrojem nejen pro správce systémů, ale i pro bezpečnostní analytiky, kteří potřebují rychle reagovat na bezpečnostní incidenty.

2.2.1 Inputy – vstupní brána do systému

Inputy představují pomyslnou vstupní bránu, kterou logy poprvé pronikají do systému Graylog. Každý input má svůj konkrétní účel a definovaný formát, který odpovídá typu zdroje dat, ze kterého zprávy přicházejí. Právě správná volba a konfigurace inputu hraje zásadní roli nejen v přehlednosti, ale především ve stabilitě a bezpečnosti celého procesu přenosu dat.

V rámci tohoto projektu se autor rozhodl plně spoléhat na moderní rozhraní Beats Input. Tato volba byla učiněna na základě potřeby maximální spolehlivosti a bezpečnosti při sběru dat z různorodých operačních systémů. Beats Input v Graylogu standardně naslouchá na portu 5044 a slouží jako centrální sběrný bod pro specializované agenty rodiny Beats.

Hlavním pilířem pro sběr dat jsou v této architektuře nástroje Filebeat a Winlogbeat. Zatímco Filebeat zajišťuje plynulý přísun dat z Linuxových distribucí a vzdáleného VPS, Winlogbeat se zaměřuje na specifické události systému Windows (Windows Event Log). Využití této Beats architektury přináší oproti tradičním metodám (jako je například Syslog) značné výhody v podobě nativní strukturovanosti dat a možnosti šifrování celého přenosu.

Logy již na vstupu do systému obsahují užitečná metadata, jako je název hostitele nebo typ operačního systému, což výrazně usnadňuje jejich další třídění. V kombinaci s takzvanými Extractory, které lze na inputy aplikovat, je možné data okamžitě po přijetí rozparsovat na jednotlivá pole. Tímto krokem se hned na začátku procesu zvyšuje kvalita a čitelnost informací, což je klíčové pro veškerou následnou analýzu a detekci bezpečnostních incidentů.

2.2.2 Streams – třídění a směrování logů

Streamy jsou po inputech dalším klíčovým prvkem. Zajišťují především organizaci dat uvnitř Graylogu. Je možné přirovnat je k filtrům, které rozhodují o tom, kam logy budou směrovány dle svých vlastností. Každý stream má určená pravidla, podle nichž Graylog automaticky rozřazuje přicházející zprávy. Tento přístup umožňuje přehledné uspořádání logu například podle typu zařízení nebo závažnosti události.

Ve vytvořeném prostředí bude existovat stream pro Windows servery, stream pro Linuxové servery, dále stream pro vzdálené VPS, a také stream pro bezpečnostní incidenty. V praxi to tedy bude fungovat tak, že logy porušující nastavená pravidla například o připojení mimo pracovní dobu, či neúspěšná přihlášení skončí ve streamu pro bezpečnostní incidenty, kde nad nimi bude prováděna patřičná analýza a vyhodnocení.

Kromě klíčového třídění logů nabízejí streamy i správu oprávnění uživatelů. Díky této možnosti lze zajistit aby určitá osoba/skupina měla přístup pouze ke svým relevantním logům.

2.2.3 Pipelines – zpracovávání a obohacování dat

Pipelines jsou pokročilým ale velmi užitečným nástrojem, který umožňuje pracovat s logy ještě před uložením do databáze. Pomocí nich je možné logy modifikovat, třídit nebo doplňovat. Každá pipeline je sestavena z několika menších částí takzvaných pipeline rules, což jsou malé skripty psané v jednoduchém jazyce Graylogu.

Právě díky pipelines je možné z logů snadno a automatizovaně odstraňovat nepotřebné údaje, normalizovat formát časových údajů, nebo přidávat nové informace. Typicky lze pomocí pipelines log rozšířit o geolokační informace anebo ho třeba označit tagem. V rámci projektu autor využije pipelines především pro rozšíření logu o geolokační data a tagování, čímž se výrazně ulehčí jejich vyhledávání a následná analýza.

2.2.4 Dashboards, vyhledávání a upozornění

Uživatelsky přívětivé webové rozhraní, které Graylog poskytuje, zaujme především svou přehledností a snadnou možností vizualizace a analýzy logu. Pomocí dashboardů je možné vytvářet a znázorňovat přehledy o aktivitě v síti. Právě díky těmto vizualizacím získá uživatel okamžitý přehled o stavu infrastruktury.

Zásadní součástí celého systému jsou i alerty, ty zajišťují automatické upozornění na konkrétní, předem stanovené události. Systém umožňuje zasílání notifikací pomocí e-mailu, ale lze využít i integraci s dalšími nástroji. Díky této možnosti se z Graylogu stává aktivní nástroj pro bezpečnostní monitoring.

Vyhledávání dat je zajištěno pomocí dotazovacího jazyka podobného Lucene Query Language, díky čemuž lze vyhledávat konkrétní události a vytvářet přesné dotazy i nad velkým množstvím záznamů. Díky tomu, lze snadno během několika vteřin dohledat přesný čas, kdy k danému incidentu došlo a zjistit okolní události.

2.2.5 Indexy a Index Sets – správa a uchovávání logů

Graylog využívá indexy jako hlavní mechanismus pro ukládání a organizaci všech logů. Každý přijatý záznam je po zpracování uložen do databázového systému, kde je zařazen do konkrétního indexu. Ten lze chápat jako takový logický kontejner určený k uchovávání dat z daného časového rámce nebo zdroje. Správné nastavení indexů má zásadní vliv na výkon, ale i na rychlost vyhledávání a celkovou stabilitu systému.

Graylog umožňuje práci s takzvanými Index Sets tedy skupinami indexů, které sdílejí společná pravidla a parametry, například strategii rotace, dobu uchování dat nebo způsob archivace. Rotace indexů zajišťuje, že po dosažení určité velikosti nebo po uplynutí stanoveného intervalu se začne zapisovat do nového indexu. Díky tomu si systém zachovává vysoký výkon a rychlost při vyhledávání dat. Retenční politika pak určuje, jak dlouho zůstanou logy v aktivních indexech, než budou odstraněny. Tento mechanismus pomáhá efektivně nakládat s úložným prostorem a brání jeho přeplnění.

V případech, kdy je nutné logy uchovávat delší dobu, lze využít funkci archivace. Ta umožňuje přesun starších indexů na sekundární úložiště, například do cloudu nebo na síťový disk, aniž by to ovlivnilo výkon systému.

2.3 MongoDB

MongoDB tvoří nedílnou součást systému Graylog a hraje klíčovou roli v jeho vnitřní architektuře. Tato databáze slouží především k uchovávání konfigurací, metadat a interních informací nezbytných pro správný provoz aplikace.

Ukládají se zde například informace o uživatelských účtech, oprávněních, definicích vstupů, streamů, pipeline pravidel či o nastavení jednotlivých dashboardů a alertů.

Jedná se o NoSQL databázi, která místo tabulek používá dokumenty ve formátu JSON. MongoDB tak umožňuje systému Graylog snadno přidávat nová nastavení nebo rozšíření. Díky této vlastnosti je správa systému rychlá, stabilní a odolná i vůči výpadkům. Databáze rovněž zajišťuje konzistenci po restartu systému, protože veškerá konfigurace zůstává zachována.

V rámci projektu bude MongoDB nasazena společně s Graylogem na jednom serveru, což je vzhledem k rozsahu prostředí zcela dostačující. Slouží jako centrální úložiště konfigurací, které zajišťuje, že systém lze po restartu či obnově spustit ve stejném stavu, v jakém byl před výpadkem. MongoDB je tedy tichým, ale nezastupitelným prvkem celého řešení bez kterého by Graylog nedokázal správně fungovat.

2.4 Cowrie – SSH/Telnet Honeypot

Honeypot Cowrie představuje v projektu aktivní prvek obrany a sběru dat o útočnících. Jedná se o tzv. „medium-interaction“ honeypot, který simuluje zranitelný SSH a Telnet server. Jeho hlavním úkolem je klamat útočníky a boty, kteří se snaží o prolomení hesel pomocí metod brute-force. Cowrie zaznamenává veškerou interakci útočníka – od pokusů o přihlášení až po příkazy, které se útočník pokouší v systému vykonat po domnělém úspěšném vniknutí. Data z honeypotu jsou v reálném čase odesílána do Graylogu, kde jsou podrobena analýze, což umožňuje identifikovat aktuální trendy v útocích a odhalit IP adresy botnetu v internetu.

2.5 Docker – Kontejnerizace služeb

Pro efektivní a bezpečný provoz honeypotu Cowrie byla využita technologie Docker. Tato platforma umožňuje izolovat aplikaci a její závislosti do lehkého kontejneru, který sdílí jádro hostitelského operačního systému, ale běží ve vlastním odděleném prostředí. V kontextu bezpečnosti je toto řešení klíčové, neboť případné kompromitování honeypotu útočníkem

zůstává izolováno uvnitř kontejneru a neohrožuje stabilitu ani bezpečnost samotného hostitelského serveru. Docker navíc usnadňuje správu, aktualizaci a rychlé nasazení celého bezpečnostního uzlu.

2.6 GeoIP – Geografická lokalizace IP adres

V rámci analýzy síťových útoků a monitoringu přístupů je často nezbytné znát nejen IP adresu útočníka, ale i jeho přibližnou geografickou polohu. K tomuto účelu autor v projektu využívá technologii GeoIP, konkrétně databáze od společnosti MaxMind. Tato technologie umožňuje systému Graylog automaticky mapovat veřejné IP adresy na konkrétní státy, města nebo souřadnice.

Proces probíhá v rámci Pipelines, kde je na každou příchozí zprávu obsahující veřejnou IP adresu aplikována vyhledávací funkce. Výsledkem je obohacení logu o nová metadata, která následně umožňují vytvářet vizuálně atraktivní a informačně bohaté mapové dashboardy. Pro bezpečnostního analytika je tato funkce klíčová, neboť mu dovoluje okamžitě identifikovat anomálie, jako jsou například pokusy o přihlášení z exotických zemí, ve kterých se nenacházejí žádní oprávnění uživatelé spravované infrastruktury. Integrace GeoIP tak transformuje abstraktní číselné údaje na srozumitelný kontext hrozeb.

3 Realizace a konfigurace systému

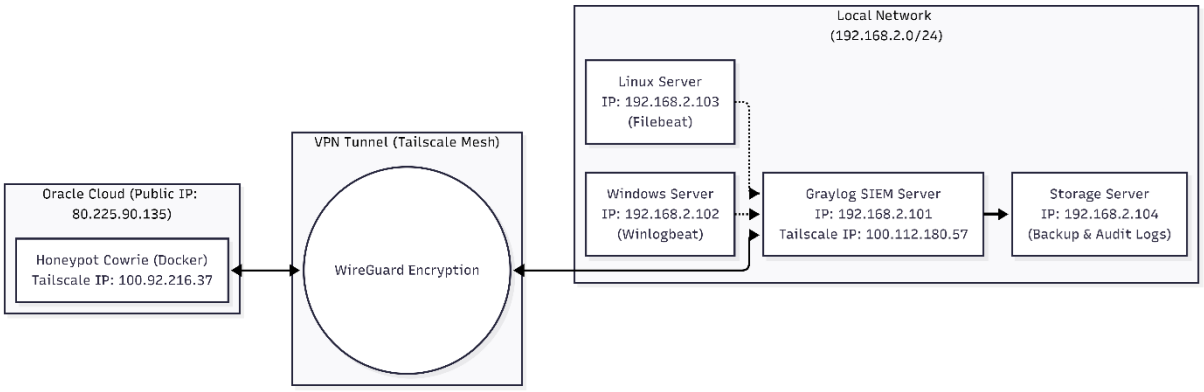
V této části práce se autor věnuje praktickému postupu při budování celého bezpečnostního uzlu. Proces realizace nebyl pouhou instalací softwarových balíčků, ale vyžadoval precizní plánování parametrů systému tak, aby byla zajištěna jeho stabilita, propustnost a schopnost dlouhodobého uchovávání dat. Jako základní stavební kámen byl zvolen operační systém Debian 12 (Bookworm), který je v komunitě bezpečnostních expertů ceněn pro svou minimalistickou architekturu a vysokou úroveň zabezpečení.

Právě s ohledem na stabilitu a vzájemnou kompatibilitu všech prvků byly pro projekt vybrány konkrétní verze softwarových komponent. Jádro logserveru tvoří Graylog Server ve verzi 7.0.2, přičemž jako indexační backend slouží Graylog Data Node ve stejné verzi, tedy 7.0.2, v němž je přímo integrován vyhledávací engine OpenSearch. Metadata a konfigurace celého systému jsou spravovány v databázi MongoDB verze 8.0.16. Pro samotný sběr dat z koncových stanic byly využity osvědčené verze agentů rodiny Beats, konkrétně Filebeat 9.2.3

pro Linuxové systémy a Winlogbeat 9.2.3 pro události z OS Windows. Bezpečnostní složku projektu doplňuje honeypot Cowrie, nasazený pomocí oficiálního Docker image v prostředí Dockeru verze 28.2.2. Celá infrastruktura je pak síťově propojena a zabezpečena prostřednictvím nástroje Tailscale ve verzi 1.92.1, který vytváří bezpečnou mesh VPN síť mezi všemi uzly bez ohledu na jejich fyzické umístění.

3.1 Příprava serverového prostředí a instalace jádra

Prvním krokem k vybudování funkčního řešení bylo zprovoznění centrálního serveru s pevnou IP adresou 192.168.2.101. Autor zvolil instalaci v režimu „netinst“, což umožnilo nasadit pouze nezbytné systémové součásti bez grafického rozhraní, čímž se minimalizovalo vytížení procesoru a operační paměti. Po dokončení základní instalace byla provedena kompletní aktualizace všech systémových balíčků a instalace nezbytných utilit, jako jsou curl, gnupg a wget, které jsou klíčové pro bezpečnou komunikaci s repozitáři třetích stran.



Obr. 1 Topologie sítě (Zdroj: Autor)

3.1.1 Implementace a zabezpečení MongoDB 8.0

Jako první z klíčových komponent byla nasazena databáze MongoDB ve verzi 8.0.16, která v rámci Graylogu plní roli správce veškerých metadat. Instalace začala importem oficiálního GPG klíče, který zajišťuje, že veškeré stahované balíčky jsou autentické a nebyly během přenosu modifikovány (1).

```
root@maturita-graylog:~# cat /etc/mongod.conf
# mongod.conf

# for documentation of all options, see:
#   http://docs.mongodb.org/manual/reference/configuration-options/

# Where and how to store data.
storage:
  dbPath: /var/lib/mongodb
#  engine:
#  wiredTiger:

# where to write logging data.
systemLog:
  destination: file
  logAppend: true
  path: /var/log/mongodb/mongod.log

# network interfaces
net:
  port: 27017
  bindIp: 192.168.2.101


# how the process runs
processManagement:
  timeZoneInfo: /usr/share/zoneinfo

#security:

#operationProfiling:

#replication:

#sharding:

## Enterprise-Only Options:

#auditLog:
root@maturita-graylog:~#
```

Obr. 2 Konfigurace databáze MongoDB pro ukládání metadat systému Graylog (Zdroj: Autor)

Po úspěšném přidání repozitáře a instalaci samotného balíku mongodb-org přistoupil autor k důležitému kroku, zafixování verze balíčku. Tento úkon zabrání automatickým aktualizacím, které by mohly v budoucnu způsobit nekompatibilitu se serverem Graylog. V rámci síťové konfigurace autor upravil soubor mongod.conf, kde definoval, že databáze bude naslouchat na statické IP adrese serveru na portu 27017. Toto nastavení je bezpečnější než otevření databáze do všech síťových rozhraní a zároveň zaručuje, že Graylog i Data Node budou mít ke svým konfiguracím vždy přístup (1).

3.1.2 Konfigurace Graylog Data Node a optimalizace OS

Následovala implementace komponenty Graylog Data Node, která tvoří výkonné vyhledávací jádro postavené na OpenSearch. Tato část vyžadovala hlubší zásah do nastavení

```
root@maturita-graylog:~# cat /etc/sysctl.d/99-graylog-datanode.conf
vm.max_map_count=262144
```

Obr. 3 Optimalizace jádra systému navýšením limitu mapování paměti pro korektní běh datového uzlu (Zdroj: Autor)

samotného linuxového kernelu. Autor musel v souboru /etc/sysctl.d/99-graylog-datanode.conf navýšit limit pro mapování paměti vm.max_map_count na hodnotu 262144. Bez této úpravy by indexovací proces nemohl správně fungovat a docházelo by k nestabilitě celého systému (1).

Bezpečnost přenosu dat mezi jednotlivými komponentami byla zajištěna vygenerováním silného šifrovacího klíče password_secret. Tento klíč, vytvořený jako náhodný hexadecimální řetězec, slouží k šifrování citlivých údajů v MongoDB a autor jej musel identicky nastavit jak v konfiguraci Data Node, tak později i v Graylog Serveru(1).

3.1.3 Instalace Graylog Serveru a správa paměti

Po přípravě datové vrstvy byla zahájena instalace hlavní aplikace Graylog Server. Prvním bezpečnostním úkonem bylo vygenerování hashe pro administrátorské heslo pomocí algoritmu SHA-256. Tento hash byl následně vložen do konfiguračního souboru server.conf pod parametr root_password_sha2, což zajišťuje, že ani administrátorské heslo není v systému uloženo v čitelné podobě (1).

V rámci ladění výkonu autor upravil soubor /etc/default/graylog-server, kde definoval parametry JVM. Nastavením fixní velikosti paměti -Xms2g a -Xmx2g se předešlo zbytečné režii systému při dynamickém přidělování RAM, což v kombinaci s moderním algoritmem Garbage Collectoru (UseG1GC) zajišťuje plynulý chod i při nárazovém příjmu velkého množství logů. Pro přístup k managementu systému byla webová adresa nasměrována na port 9000 statické IP adresy serveru (1).

```
root@maturita-graylog:~# cat /etc/default/graylog-server
# Path to a custom java executable. By default the java executable of the
# bundled JVM is used.
#JAVA=/usr/bin/java

# Default Java options for heap and garbage collection.
GRAYLOG_SERVER_JAVA_OPTS="-Xms2g -Xmx2g -server -XX:+UseG1GC -XX:-OmitStackTraceInFastThrow"
```

Obr. 4 Konfigurace parametrů Java Virtual Machine (JVM) a alokace operační paměti pro stabilní běh Graylog serveru (Zdroj: Autor)

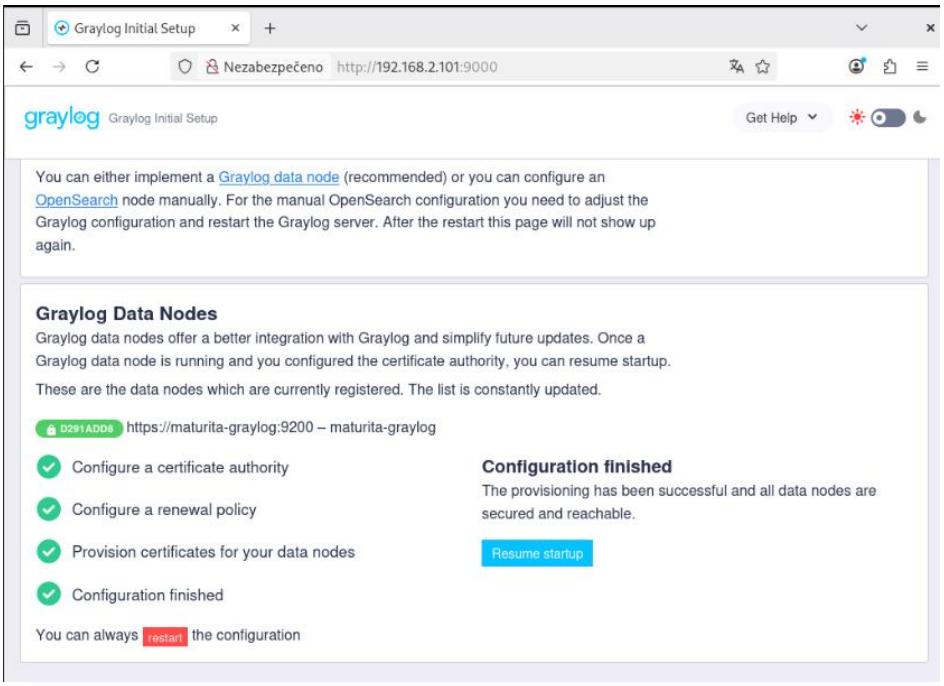
3.1.4 Analýza logů a dokončení Preflight konfigurace

Závěrečná fáze zprovoznění začala spuštěním všech služeb a následnou kontrolou systémového logu /var/log/graylog-server/server.log. Graylog při svém prvním spuštění přechází do speciálního režimu „Preflight“, ve kterém vygeneruje unikátní jednorázové heslo pro úvodní nastavení (1).

Obr. 5 Analýza systémového logu Graylogu pro potvrzení úspěšné inicializace a získání prvotních přihlašovacích údajů k webovému rozhraní (Zdroj: Autor)

3.1.5 Zabezpečení vnitřní komunikace a certifikační autorita

Během závěrečné fáze úvodní konfigurace v rozhraní „Pre-flight“ kladl autor zásadní důraz na zabezpečení vnitřní architektury celého systému. Moderní verze Graylogu vyžadují, aby veškerá komunikace mezi hlavní aplikací a Data Node probíhala v šifrovaném režimu, což eliminuje riziko odposlechu citlivých logů v rámci vnitřní sítě serveru. Z tohoto důvodu byla přímo v systému definována a aktivována vlastní interní certifikační autorita.



Obr. 6 Dokončení úvodní konfigurace a úspěšné zprovoznění certifikační autority pro zabezpečenou komunikaci mezi uzly (Zdroj: Autor)

Tato autorita slouží jako důvěryhodný bod, který v rámci SIEMu vydává a podepisuje certifikáty pro jednotlivé komponenty. V praxi to znamená, že autor nechal touto CA vygenerovat a nainstalovat unikátní certifikát pro instanci Data Node, čímž bylo vytvořeno zabezpečené a autentizované spojení. Data Node se díky tomu prokazuje serveru svou identitou a veškerý přenos dat je šifrován. Pro zajištění dlouhodobé bezúdržbovosti systému byla rovněž

nastavena politika automatické obnovy těchto certifikátů, čímž autor předešel potenciálním výpadkům způsobeným vypršením platnosti bezpečnostních prvků. Po dokončení těchto nezbytných úkonů a finálním potvrzení certifikačních cest byl systém plně připraven k trvalému provozu a autor se mohl poprvé přihlásit do administrátorského dashboardu pomocí svého trvalého hesla

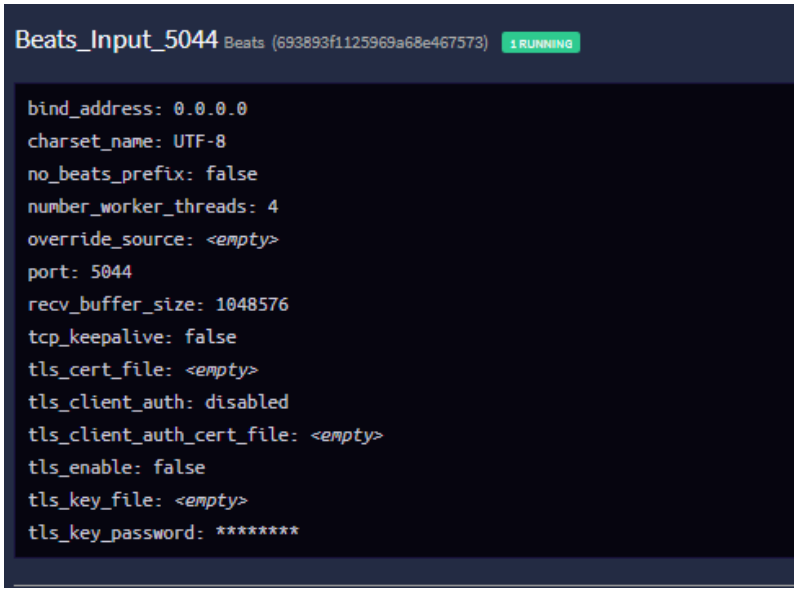
3.2 Sběr dat z interní sítě a konfigurace agentů

Jakmile bylo jádro systému stabilně v provozu, bylo nutné připravit mechanismy pro samotný sběr logů. Autor se v této fázi rozhodl nevyužívat zastaralé a méně bezpečné metody, ale plně se soustředil na moderní architekturu založenou na agentech rodiny Beats, kteří zajišťují spolehlivý přenos dat i při případných výpadcích konektivity.

3.2.1 Konfigurace centrálního Beats Inputu

Prvním krokem v rámci webového rozhraní Graylogu bylo vytvoření nového vstupu typu Beats Input. Tento vstup funguje jako centrální naslouchající uzel, na který se připojují všichni agenti z celé sítě. Autor zvolil pro komunikaci standardní port 5044 a nastavil vstup jako globální, aby byl dostupný pro všechna zařízení.

Výhodou tohoto přístupu je, že Graylog automaticky rozpoznává strukturu dat přicházejících od Beats agentů, což usnadňuje jejich následné parsování.



Obr. 7 Konfigurace a monitoring síťového vstupu typu Beats na portu 5044 pro příjem logů z Beats agentů (Zdroj: Autor)

3.2.2 Monitoring Linuxového prostředí pomocí Filebeatu

Pro sběr systémových zpráv z Linuxových serverů a z budoucího Honeypotu byl vybrán agent Filebeat. Jeho instalace na cílové stanice probíhala skrze oficiální repozitáře Elastic.

Hlavní úsilí bylo věnováno konfiguračnímu souboru filebeat.yml. Zde autor deaktivoval výchozí výstup pro Elasticsearch a nahradil jej výstupem pro už nastavený Filebeat input v Graylogu. Jako cílovou adresu (hosts) nastavil statickou IP adresu logserveru 192.168.2.101:5044. Filebeat byl následně nakonfigurován tak, aby sledoval systémový žurnál (journald), což umožňuje sbírat veškeré události od startu služeb až po pokusy o SSH přihlášení v reálném čase (2).

3.2.3 Monitoring Windows stanic pomocí Winlogbeatu

Pro monitoring interního serveru s operačním systémem Windows byl nasazen agent Winlogbeat, jehož konfigurace byla optimalizována pro sběr kritických bezpečnostních událostí. Autor se v konfiguračním souboru zaměřil nejen na standardní kanály Application, System a Security, ale pro hloubkovou analýzu incidentů aktivoval i sledování pokročilých zdrojů událostí (3).

winlogbeat.yml – Poznámkový blok
Soubor Úpravy Formát Zobrazení Nápověda

```
winlogbeat.event_logs:
  - name: Application
    ignore_older: 72h

  - name: System

  - name: Security

  - name: Microsoft-Windows-Sysmon/Operational

  - name: Windows PowerShell
    event_id: 400, 403, 600, 800

  - name: Microsoft-Windows-PowerShell/Operational
    event_id: 4103, 4104, 4105, 4106

  - name: ForwardedEvents
    tags: [forwarded]
```

Obr. 8 Konfigurace agenta Winlogbeat pro selektivní sběr systémových protokolů, událostí Sysmon a aktivit PowerShellu z platformy Windows (Zdroj: Autor)

Klíčovým prvkem je integrace logů z nástroje Sysmon (Microsoft-Windows-Sysmon/Operational), který poskytuje detailní vhled do procesů v systému, a monitoring PowerShellu. Jako výstupní cíl (output) byl definován logserver na adrese 192.168.2.101:5044 (3).

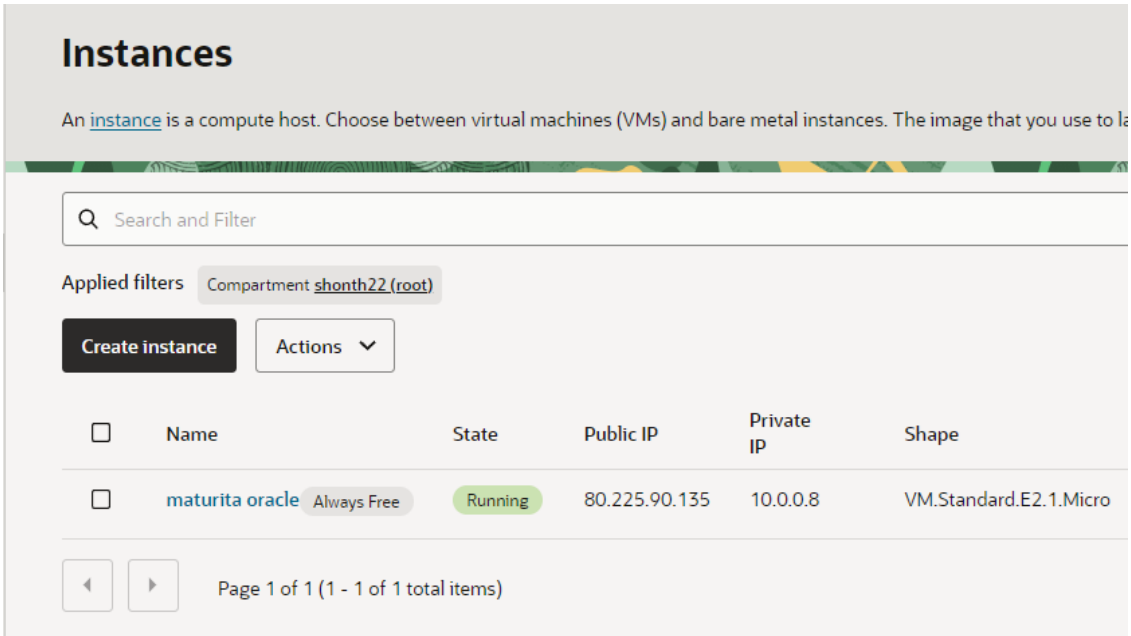
Díky strukturovanému formátu odesílaných dat jsou v Graylogu okamžitě dostupná pole jako zdrojová IP adresa, uživatelské jméno nebo příkaz spuštěný v PowerShellu, a to bez nutnosti dodatečného parsování textu. To výrazně zrychluje reakci na bezpečnostní incidenty a zjednodušuje tvorbu analytických dashboardů (3).

3.3 Příprava cloudové instance pro externí sběr dat

Aby byl výsledný SIEM systém schopen monitorovat hrozby nejen uvnitř lokální sítě, ale i v globálním měřítku internetu, bylo nutné expandovat infrastrukturu do cloudu. Autor se rozhodl pro využití platformy Oracle Cloud Infrastructure, která nabízí vysokou úroveň flexibility a bezpečnosti pro provozování veřejně dostupných uzlů.

3.3.1 Provisioning VPS v Oracle Cloud Infrastructure

Prvním krokem bylo vytvoření virtuální instance v rámci bezplatného programu Oracle Cloud. Tato instance byla nakonfigurována s operačním systémem Canonical Ubuntu verze 24.04. Autor zvolil instanci s architekturou ARM (Always Free), která poskytuje dostatečný výkon pro běh kontejnerizovaných služeb i odesílání logů v reálném čase (4).



Obr. 9 Správa a monitoring běžící instance virtuálního serveru v prostředí Oracle Cloud Infrastructure (Zdroj: Autor)

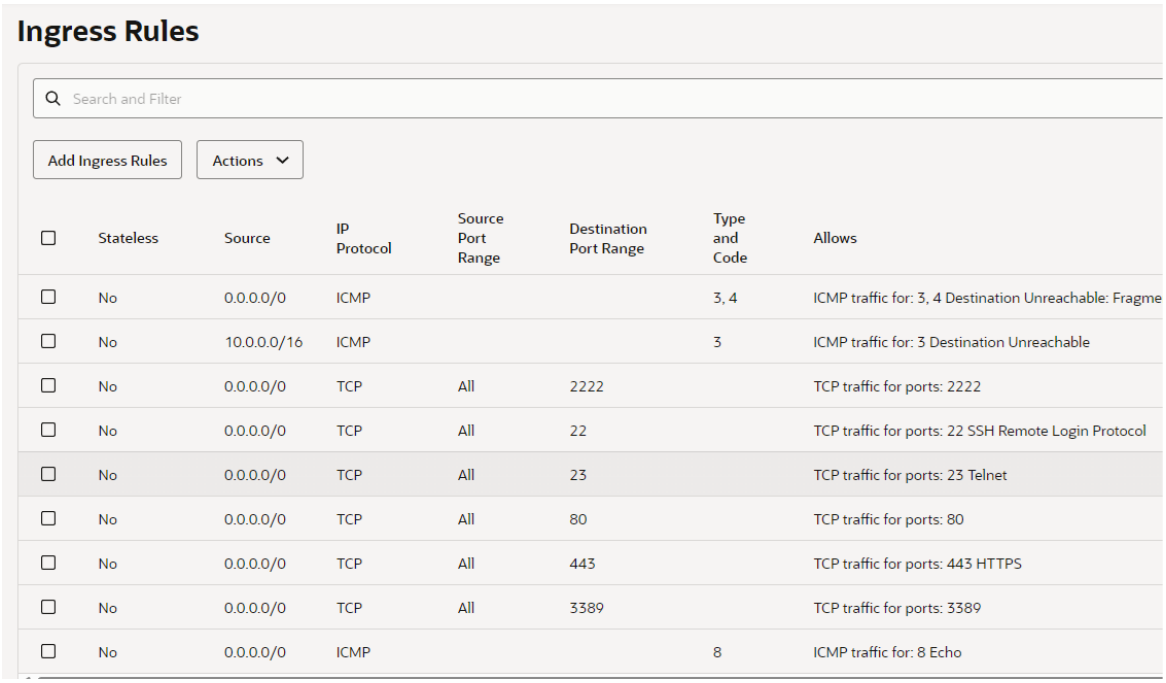
Během procesu vytváření instance byla vygenerována dvojice SSH klíčů. Veřejný klíč byl vložen do konzole Oracle pro autorizaci přístupu, zatímco soukromý klíč byl bezpečně uložen na lokálním stroji autora. Tímto bylo zajištěno, že administrace cloudu probíhá výhradně šifrovaně a bez možnosti zneužití standardních hesel útočníky (4).

3.3.2 Konfigurace Virtual Cloud Network a Ingress pravidel

Jedním z nejdůležitějších úkolů v cloudovém prostředí byla správná konfigurace virtuální sítě a firewallu, jelikož Oracle Cloud standardně blokuje veškerý příchozí provoz. Tato restriktivní politika vyžadovala precizní úpravu tzv. Security Lists, v rámci kterých autor definoval několik úrovní síťové propustnosti pro zajištění správy systému i funkčnosti detekčních senzorů.

Pro potřeby správy a bezpečného přenosu dat autor povolil protokol TCP na standardním portu 22 (SSH Remote Login Protocol), čímž byla zajištěna administrátorská kontrola nad celou instancí. Klíčovou částí konfigurace však bylo vystavení specifických služeb do veřejného internetu za účelem simulace zranitelného prostředí. V rámci honeypot úrovně autor otevřel porty pro protokoly TCP 23 (Telnet), TCP 80 (HTTP) a TCP 443 (HTTPS), které slouží jako primární návnady pro automatizované botnety a útočníky. Dále byl implementován port 2222 určený pro specifickou SSH návnadu Cowrie a port 3389 simulující protokol vzdálené plochy (RDP) (5).

Pro diagnostiku síťové konektivity byl rovněž povolen protokol ICMP, konkrétně typy pro Echo request a hlášení o fragmentaci dat. Veškerý tento provoz je v reálném čase monitorován a odesílán do centrálního SIEMu, kde jsou data o pokusech o vniknutí agregována a následně vizualizována na geografických mapách útoků v hlavním dashboardu (5).



Ingress Rules

Search and Filter

Add Ingress Rules | Actions

	Stateless	Source	IP Protocol	Source Port Range	Destination Port Range	Type and Code	Allows
☐	No	0.0.0.0/0	ICMP			3, 4	ICMP traffic for: 3, 4 Destination Unreachable: Fragme
☐	No	10.0.0.0/16	ICMP			3	ICMP traffic for: 3 Destination Unreachable
☐	No	0.0.0.0/0	TCP	All	2222		TCP traffic for ports: 2222
☐	No	0.0.0.0/0	TCP	All	22		TCP traffic for ports: 22 SSH Remote Login Protocol
☐	No	0.0.0.0/0	TCP	All	23		TCP traffic for ports: 23 Telnet
☐	No	0.0.0.0/0	TCP	All	80		TCP traffic for ports: 80
☐	No	0.0.0.0/0	TCP	All	443		TCP traffic for ports: 443 HTTPS
☐	No	0.0.0.0/0	TCP	All	3389		TCP traffic for ports: 3389
☐	No	0.0.0.0/0	ICMP			8	ICMP traffic for: 8 Echo

Obr. 10 Konfigurace Ingress pravidel ve virtuálním firewallu pro povolení nezbytného síťového provozu (Zdroj: Autor)

3.3.3 Systémový hardening a optimalizace VPS

Po prvním přihlášení k VPS provedl autor základní hardening systému. Byla provedena deaktivace nepotřebných služeb a aktualizace jádra na nejnovější verzi. Protože se jedná o uzel vystavený do veřejného internetu, byla věnována zvýšená pozornost logování samotného přístupu k VPS.

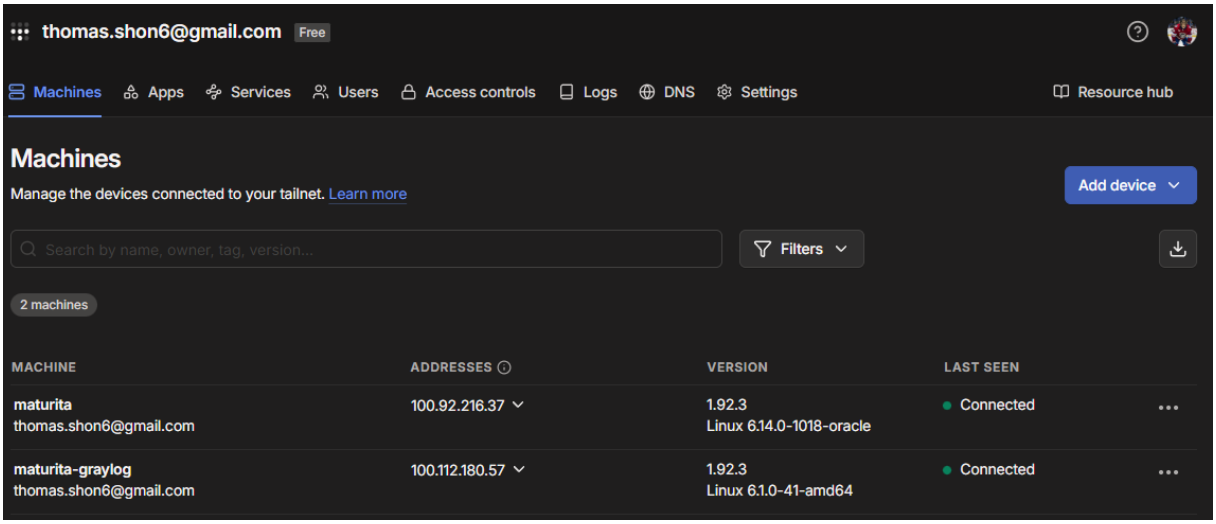
Na instanci byl nainstalován balík iptables-persistent, který zajišťuje, že veškerá pravidla síťového provozu zůstanou zachována i po restartu cloudového serveru. Dále byl připraven adresářový prostor pro budoucí instalaci Dockeru a agenta Filebeat. Tímto krokem byla cloudová instance plně připravena k tomu, aby se stala aktivním detekčním senzorem (Honeypotem) a začala plnit databázi Graylogu informacemi o reálných útocích z celého světa.

3.4 Zabezpečení komunikace a nasazení Honeypot senzoru

Po úspěšné přípravě cloudové instance v Oracle Cloudu a lokálního Graylog serveru vyvstala zásadní otázka: jak zajistit bezpečný a spolehlivý přenos logů mezi těmito dvěma světy? Protože logy z honeypotu obsahují citlivé údaje o útočnících a odesílání dat přes veřejný internet by vyžadovalo složité nastavování firewallů a šifrování na úrovni aplikací, autor zvolil moderní řešení v podobě privátní mesh sítě.

3.4.1 Implementace Mesh VPN Tailscale

Pro propojení lokální infrastruktury a cloudového VPS byla implementována technologie Tailscale, což je zero-config VPN postavená na moderním a extrémně rychlém protokolu WireGuard. Samotná instalace proběhla na obou uzlech současně a po následné autorizaci zařízení v administrátorském panelu Tailscale vznikl šifrovaný tunel. V rámci této privátní sítě oba stroje získaly unikátní IP adresy v rozsahu 100.x.x.x, což přineslo několik zásadních výhod pro bezpečnost celého projektu. (6)



Obr. 11 Nasazení SSH/Telnet honeypotu Cowrie v kontejneru Docker pro simulaci zranitelných služeb a sběr útočných dat (Zdroj: Autor)

Hlavním přínosem je úplná absence nutnosti nastavovat port-forwarding, díky čemuž nebylo na lokálním firewallu nutné otevírat žádné porty směrem dovnitř sítě, což dramaticky snižuje útočnou plochu. Veškerá komunikace mezi agentem Filebeat na cloudovém uzlu a Graylog serverem v lokální síti je nyní díky této technologii automaticky šifrována na úrovni síťové vrstvy (end-to-end encryption). Toto řešení navíc vyniká vysokou stabilitou, neboť Tailscale dokáže efektivně pracovat se změnami IP adres i přechody mezi různými sítěmi, čímž je zajištěna kontinuita sběru dat bez rizika výpadků konektivity. (6)

3.4.2 Nasazení Honeypotu Cowrie pomocí Dockeru

Jako hlavní „lákadlo“ pro útočníky byl na cloudovém VPS zvolen honeypot Cowrie. Tento systém simuluje prostředí SSH serveru a Telnetu a je schopen zaznamenávat nejen pokusy o přihlášení, ale i veškerou interakci útočníka po úspěšném vniknutí do systému.

```
root@maturita:/# docker ps
CONTAINER ID   IMAGE           COMMAND                  CREATED      STATUS       PORTS
                                                NAMES
65bf2b2b9723   cowrie/cowrie   "/cowrie/cowrie-env/…"   7 days ago   Up 3 hours   0.0.0.0:22->2222/tcp, [::]:22->2222/tc
p, 0.0.0.0:23->2223/tcp, [::]:23->2223/tcp   cowrie
```

Obr. 12 Zabezpečení komunikace mezi servery pomocí privátní sítě Tailscale pro bezpečnou výměnu dat (Zdroj: Autor)

Aby byla zajištěna maximální bezpečnost hostitelského systému VPS, autor nasadil Cowrie v izolovaném prostředí pomocí platformy Docker. Tento přístup zaručuje, že útočník, který se domnívá, že ovládl server, je ve skutečnosti uzavřen v kontejneru s omezenými prostředky, odkud nemůže nijak napadnout skutečný operační systém Debian nebo proniknout dále do sítě. Kontejner byl nakonfigurován tak, aby své podrobné logy v reálném čase ukládal do souboru cowrie.json, což je klíčové pro následnou analýzu (7).

3.4.3 Konfigurace Filebeatu pro sběr dat z Honeypotu

Posledním krokem této fáze byla instalace a konfigurace agenta Filebeat přímo na cloudovém VPS. Úkolem agenta je sledovat zmíněný soubor cowrie.json a každou novou událost okamžitě odeslat ke zpracování.

V konfiguračním souboru filebeat.yml autor definoval cestu k logům v rámci Docker svazků a jako cílový výstup nastavil Tailscale IP adresu lokálního Graylog serveru na portu 5044. Díky tomuto nastavení logy putují šifrovaným tunelem Tailscale přímo do Beats Inputu v Graylogu. Tímto byl vytvořen ucelený řetězec: útok na cloudu – záznam v kontejneru – transport přes VPN – uložení v Graylogu (2).

```
# ============================== Filebeat inputs ===============================

filebeat.inputs:

# Each - is an input. Most options can be set at the input level, so
# you can use different inputs for various configurations.
# Below are the input-specific configurations.

# filestream is an input for collecting log messages from files.
- type: filestream

  # Unique ID among all inputs, an ID is required.
  id: my-filestream-id

  # Change to true to enable this input configuration.
  enabled: true

  # Paths that should be crawled and fetched. Glob based paths.
  paths:
    - /var/log/*.log
    #- c:\programdata\elasticsearch\logs\*
- type: log

  enabled: true

  id: cowrie-honeypot-id

  paths:
    - /var/log/cowrie/cowrie.json
```

Obr. 13 Konfigurace agenta Filebeat pro automatizovaný sběr a odesílání logů z honeypotu Cowrie ve formátu JSON (Zdroj: Autor)

3.4.4 Právní a etické aspekty provozu honeypotu

Součástí navrženého řešení je nasazení honeypotu Cowrie, jehož účelem je pasivní zachytávání pokusů o neoprávněný přístup a analýza chování útočníků. Provoz takového systému může vést ke zpracování údajů, které lze v určitých případech považovat za osobní údaje, zejména IP adresy a textové příkazy zadávané útočníkem.

S ohledem na platnou legislativu, zejména Nařízení Evropského Parlamentu a Rady (EU) 2016/679 o ochraně fyzických osob v souvislosti se zpracováním osobních údajů a o volném pohybu těchto údajů a o zrušení směrnice 95/46/ES (GDPR), je honeypot v rámci tohoto projektu provozován výhradně v pasivním režimu a neslouží k aktivnímu provokování útoků. Zachycená data jsou využívána pouze pro studijní a analytické účely v rámci tohoto projektu a nejsou poskytována třetím stranám.

Pro minimalizaci rizik je honeypot provozován v izolovaném prostředí, odděleném od interní infrastruktury. Zaznamenaná data slouží výhradně k analýze bezpečnostních událostí.

Provoz honeypotu je navržen tak, aby nedocházelo k porušení etických zásad ani právních předpisů. Systém neslouží k identifikaci konkrétních osob, ale k demonstraci principů detekce útoků a zpracování bezpečnostních událostí v rámci centralizovaného logovacího systému.

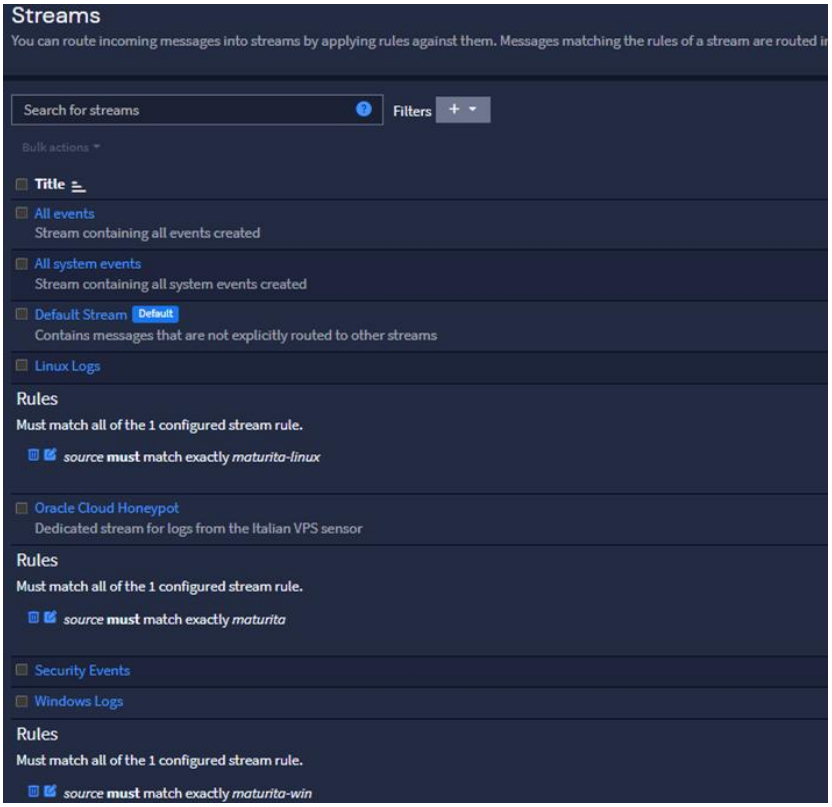
3.5 Organizace a inteligentní zpracování dat

Po úspěšném navázání konektivity se všemi zdroji bylo nutné zajistit, aby příchozí data nebyla pouze ukládána, ale aktivně tříděna a analyzována. K tomuto účelu autor využil kombinaci Streamů pro základní kategorizaci a Pipelines pro pokročilou logiku a obohacování dat.

3.5.1 Segmentace provozu pomocí Streamů

Prvním stupněm organizace dat v systému Graylog bylo vytvoření streamů, které fungují jako inteligentní směrovače logů na základě jejich původu. Pro zajištění správné kategorizace autor definoval specifická pravidla, která v reálném čase analyzují příchozí zprávy a okamžitě identifikují jejich zdroj. V rámci této konfigurace byl vytvořen stream pro Linux Logs, který filtruje zprávy, kde pole *source* přesně odpovídá názvu virtuálního stroje „maturita-linux“, a obdobně nastavený stream Windows Logs směrující data ze zdroje „maturita-win“ (8).

Samostatnou a klíčovou kategorii tvoří stream Oracle Cloud Honeypot, který je dedikován výhradně pro příjem dat z cloudového VPS senzoru identifikovaného zdrojem „maturita“. Tato logická segmentace je pro systém SIEM zásadní, neboť umožňuje aplikovat různé sady analytických pravidel na různé typy provozu. Zároveň tato struktura výrazně zjednodušuje a zrychluje následné vyhledávání incidentů a vizualizaci dat v dashboardech, protože analytik může pracovat s konkrétní izolovanou skupinou logů namísto prohledávání celého objemu přijatých dat (8).

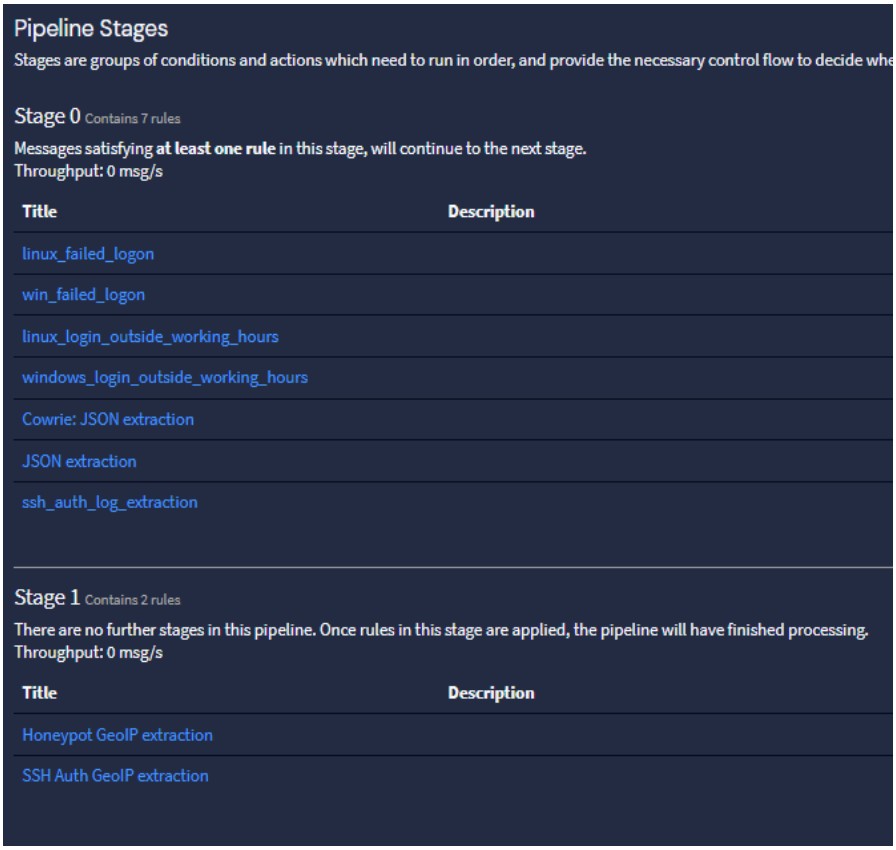


Obr. 14 Logické třídění příchozích dat do samostatných proudů (Streams) pro efektivní správu událostí z Linuxu, Windows a honeypotu (Zdroj: Autor)

3.5.2 Implementace Pipeline pravidel a normalizace

Hlavním nástrojem pro analýzu dat v reálném čase se stala Pipeline 1, která je přímo napojena na definované streamy. Autor v rámci této pipeline navrhl dvě logické fáze (Stages), které zajišťují postupné a strukturované zpracování příchozích událostí. První fáze, označovaná jako Stage 0, je zaměřena na primární detekci incidentů a extrakci surových dat. V této části běží celkem sedm pravidel, která zajišťují normalizaci záznamů z různých zdrojů (9).

Pravidla win_failed_logon a linux_failed_logon automaticky identifikují neúspěšné pokusy o autentizaci, zatímco pravidla JSON extraction a Cowrie: JSON extraction využívají funkci parse_json k rozkladu surového textu na strukturovaná pole. Pro analýzu přímých útoků na VPS bylo implementováno pravidlo ssh_auth_log_extraction, které vytahuje uživatelská jména a IP adresy útočníků ze systémových logů.



Obr. 15 Konfigurace procesních řetězců (Pipelines) pro automatizované zpracování, filtraci a logickou úpravu příchozích událostí (Zdroj: Autor)

Navazující Stage 1 se zaměřuje na geografické obohacení již rozparsovaných dat. Jakmile jsou v nulté fázi úspěšně identifikovány zdrojové IP adresy útočníků, nastupují pravidla Honeypot GeoIP extraction a SSH Auth GeoIP extraction. Tato pravidla využívají extrahované adresy k provedení dotazu v integrované databázi GeoIP. Výsledkem tohoto procesu je automatické doplnění každé bezpečnostní zprávy o název země, města a přesné zeměpisné souřadnice útočníka, což je klíčové pro finální vizualizaci hrozeb v dashboardech, jak je detailněji popsáno v kapitole 3.6 (9) (10).

3.5.3 Detekce anomálií: Kontrola pracovní doby

Unikátním prvkem celého systému je schopnost detekovat lidský faktor a neobvyklé časy aktivit. Autor vytvořil pravidla linux_login_outside_working_hours a windows_login_outside_working_hours, která využívají pokročilé časové funkce Graylogu (10).

Obr. 16 Implementace detekční logiky v Pipeline Rule pro identifikaci podezřelých přihlášení mimo definovanou pracovní dobu (Zdroj: Autor)

Logika těchto pravidel prověřuje každé úspěšné přihlášení. Systém kontroluje, zda den v týdnu neodpovídá víkendu (dayOfWeek >= 6) nebo zda se čas události nachází mimo definované rozmezí (před 8:00 nebo po 17:00). Pokud je podmínka splněna, zpráva je označena příznakem user_login_after_hours a automaticky přesměrována do streamu Security Events. Tento mechanismus umožňuje okamžitě identifikovat podezřelé aktivity, které by v běžném provozu mohly zaniknout (10).

3.5.4 Analýza autentizačních pokusů systému (VPS)

Kromě izolovaného honeypotu se autor zaměřil také na monitoring reálných pokusů o přístup k operačnímu systému VPS instance v Oracle Cloud. Pomocí agenta Filebeat jsou sbírány logy ze souboru /var/log/auth.log (10).

Rule source

```
rule "ssh_auth_log_extraction"
when
    has_field("filebeat_log_file_path") &&
    to_string($message.filebeat_log_file_path) == "/var/log/auth.log" &&
    contains(to_string($message.message), "sshd") &&
    (contains(to_string($message.message), "invalid user") || contains(to_string($message.message), "authenticating user"))
then
    set_field("security_event_vps", true);
    set_field("event_type", "ssh_bruteforce");

    let extract = regex("user (\\S+) (\\d{1,3}\\.\\d{1,3}\\.\\d{1,3}\\.\\d{1,3}) port (\\d+)", to_string($message.message));

    set_field("attacker_user", extract["0"]);
    set_field("attacker_ip", extract["1"]);
    set_field("attacker_port", extract["2"]);

end
```

Obr. 17 Implementace detekčního pravidla pro automatickou extrakci dat z neúspěšných SSH pokusů (Zdroj: Autor)

Protože tyto záznamy přicházejí jako prostý text, byla vytvořena speciální Pipeline rule využívající regulární výrazy (Regex). Ta z textu automaticky extrahuje klíčová pole: attacker_user, attacker_ip a attacker_port. Zprávy jsou následně označeny příznakem security_event_vps: true, což umožňuje jejich separátní vizualizaci v dashboardu nezávisle na honeypotu. Tento přístup umožňuje v reálném čase sledovat intenzitu Brute-force útoků přímo na systémové služby serveru (10).

3.6 Obohacování dat a GeoIP lokalizace

Jednou z nejvíce přínosných funkcí celého systému je schopnost přiřadit k anonymní IP adrese útočníka konkrétní geografické údaje. To umožňuje transformovat strohá data do podoby přehledných map a statistik.

3.6.1 Konfigurace MaxMind databáze

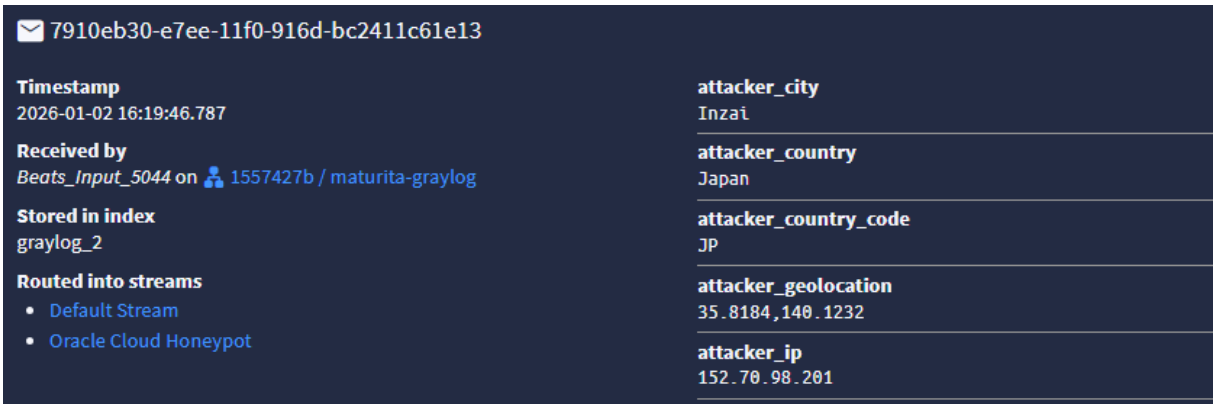
Pro realizaci této funkce autor integroval do Graylogu databázi MaxMind GeoIP. Proces spočíval v nahrání databázových souborů (GeoLite2-City) na server a následné konfiguraci tzv. Lookup Tables a jejich adaptéru.

V rámci Graylogu byl vytvořen datový adaptér, který funguje jako most: pokaždé, když systémem projde log obsahující IP adresu, Graylog se „podívá“ do této databáze a zjistí, ke kterému státu a městu daná adresa patří. Tento proces probíhá v milisekundách a nijak nezdržuje zpracování logů (11).

3.6.2 Vizualizace geografického původu útoků

Výsledkem tohoto obohacení je vytvoření nových polí u každého logu, jako jsou attacker_country_name nebo attacker_geolocation. Tato pole autor následně využil při tvorbě World Map widgetu (10).

Díky GeoIP lokalizaci je na první pohled patrné, ze kterých koutů světa přichází nejvíce pokusů o prolomení honeypotu. Tato data mají nejen vizuální hodnotu pro prezentaci projektu, ale v reálném nasazení umožňují bezpečnostním technikům identifikovat botnety operující z konkrétních zemí a případně na ně reagovat úpravou pravidel na firewallu.



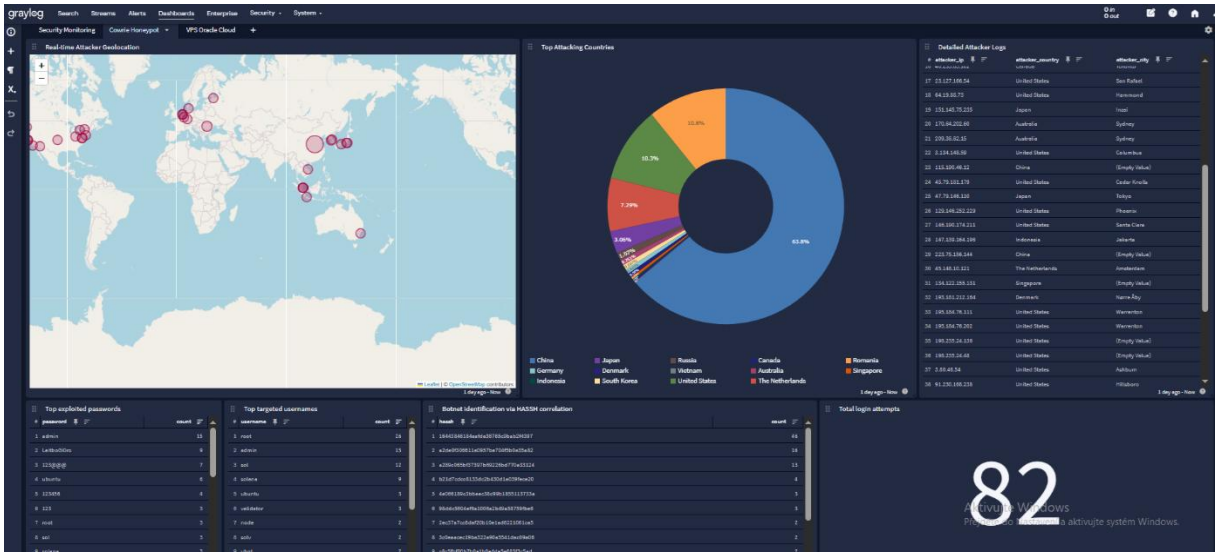
Obr. 18 Detail zpracované události s automatickým obohacením o geografické údaje útočníka na základě databáze GeoIP (Zdroj: Autor)

3.7 Vizualizace dat a bezpečnostní Dashboardy

V této fázi projektu autor přeměnil surová a normalizovaná data do přehledných vizuálních výstupů. Byly vytvořeny tři hlavní dashboardy, které slouží k okamžitému monitoringu bezpečnosti infrastruktury v reálném čase.

3.7.1 Cowrie Honeypot dashboard

Tento dashboard je primárně zaměřen na vizualizaci a analýzu aktivity vnějšího světa zachycené cloudovým honeypotem Cowrie. Dominantním prvkem celého rozhraní je mapa světa v reálném čase (Real-time Attacker Geolocation), která vizualizuje přesný geografický původ útoků na základě dat získaných z GeoIP lookupu.

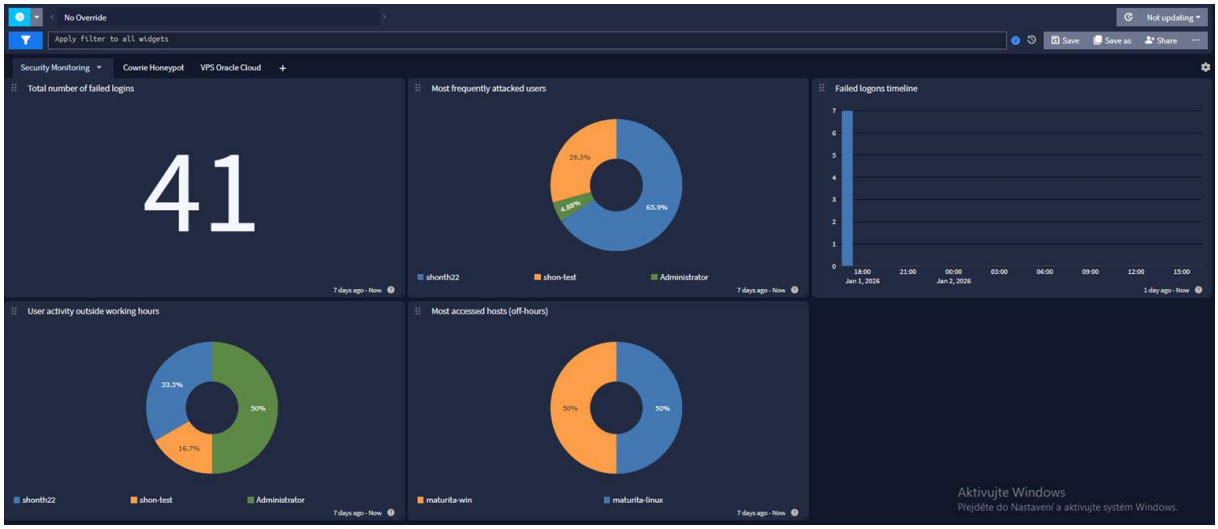


Obr. 19 Dashboard pro vizualizaci útoků na honeypot a analýzu aktivit útočníků v reálném čase (Zdroj: Autor)

Podrobnější geografické rozložení hrozeb pak doplňuje koláčový graf Top Attacking Countries, který přehledně zobrazuje procentuální podíl útočníků z jednotlivých zemí. Kromě polohy útočníků se dashboard soustředí i na analýzu útoků prostřednictvím widgetů Top exploited passwords a Top targeted usernames. Tyto prvky odhalují nejčastější přihlašovací údaje používané botnety. Pro hlubší analýzu autor implementoval také sekci Botnet identification via HASSH correlation. Tato pokročilá metoda sledování pomocí otisků SSH klientů umožňuje identifikovat a propojovat aktivity stejných útočných skupin i v případech, kdy mění své zdrojové IP adresy.

3.7.2 Bezpečnostní události & interní monitoring

Tento dashboard slouží k systematickému dohledu nad interními stroji v rámci infrastruktury (Windows a Linux), přičemž se zde v praxi projevuje logika analytických pravidel definovaných v předchozích fázích projektu. Ústředním ukazatelem je widget Total number of failed logins, který agreguje celkový počet neúspěšných pokusů o přístup do celé monitorované sítě. Pro detailní časovou analýzu incidentů slouží graf Failed logons timeline, jenž analytikovi umožňuje přesně identifikovat nárazové útoky typu Brute-force v konkrétních časových úsecích.



Obr. 20 Dashboard pro monitoring interních systémů, sledování neúspěšných přihlášení a detekci uživatelské aktivity mimo pracovní dobu (Zdroj: Autor)

Zásadní roli v monitoringu anomálií hraje sekce User activity outside working hours, která vizualizuje události zachycené pipeline pravidlem pro kontrolu času. Přidružené grafy, zejména Most accessed hosts (off-hours), jasně indikují, na které konkrétní servery docházelo k pokusům o přihlášení v nestandardní době, jako je noc nebo víkend. Tato kombinace přehledových statistik a časové analýzy poskytuje komplexní vhled do bezpečnosti vnitřní sítě a umožňuje včasnou reakci na podezřelé aktivity uživatelů.

3.7.3 Analýza a vizualizace systémových událostí VPS

Tato sekce dashboardu představuje kritickou vrstvu dohledu zaměřenou na reálné síťové rozhraní VPS instance, která je přímo vystavena do veřejné sítě internet. Na rozdíl od honeypotu Cowrie, který slouží jako klamný cíl, tento dashboard interpretuje pokusy o narušení skutečného operačního systému. Ústředním indikátorem bezpečnosti serveru je widget Total Authentication Failures, který v reálném čase sčítá neúspěšné pokusy o přihlášení a slouží jako primární varovný signál zvýšené útočné aktivity.

Analýza útoku je realizována prostřednictvím widgetů Top Targeted Usernames SSH a Top Targeted Destination Ports. Tyto prvky umožňují analytikovi identifikovat, zda útočníci cílí na standardní systémové účty, jako je „root“ nebo „admin“, a na které konkrétní porty

směřují své požadavky. Tato data jsou klíčová pro validaci nastavení Ingress pravidel v cloudové infrastruktuře Oracle a umožňují včasnou detekci skenování portů.

Obr. 21 Dashboard v systému Graylog pro hloubkovou analýzu SSH útok (Zdroj: Autor)

Geografický kontext a původ hrozeb jsou vizualizovány na mapě Real-time Attacker Geolocation, která ve spojení s tabulkou Detailed Attacker Logs poskytuje přesné informace o zemi a městě, ze kterého útoky přicházejí. Propojení těchto údajů s přehledem Top Attacking Countries dává správci možnost identifikovat specifické regiony, které vykazují vysokou míru agresivity vůči VPS, a následně implementovat preventivní blokaci na úrovni IP rozsahů. Tato ucelená vizualizace tak tvoří nezbytný podklad pro aktivní správu bezpečnosti cloudového uzlu.

3.8 Alerting a automatizovaná oznámení (E-mail)

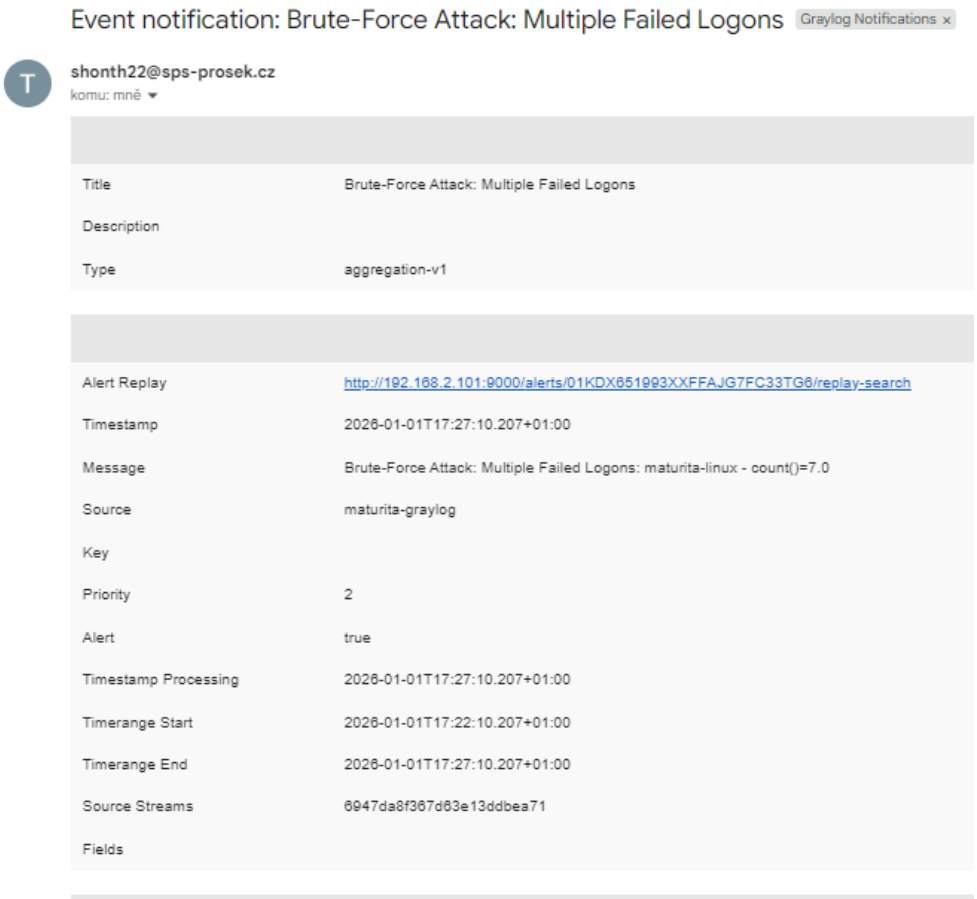
Aby systém nevyžadoval nepřetržitou pozornost administrátora, autor implementoval systém varování. Ten zajišťuje, že kritické incidenty jsou okamžitě eskalovány přímo k odpovědné osobě.

3.8.1 Definice událostí (Event Definitions)

V rámci systému Graylog autor vytvořil specifickou definici událostí, která v reálném čase prohledává a příchozí logy. Autor tuto událost nazval „Brute-Force Attack: Multiple Failed Logons“, jehož logika je postavena na sledování četnosti neúspěšných pokusů o přístup. Tato událost se automaticky aktivuje v momentě, kdy systém detekuje více než pět neúspěšných přihlášení (event_type: failed_logon) z jednoho unikátního zdroje během minutového okna. (12).

3.8.2 Konfigurace e-mailových notifikací

Pro okamžité doručování varování administrátorovi byl vytvořen dedikovaný notifikační kanál „Email Admin - Brute Force“, využívající SMTP protokol pro odesílání na adresu shonth22@sps-prosek.cz. Výjimečným prvkem této implementace je plně přizpůsobená HTML šablona e-mailu, zajišťující maximální čitelnost a informační hodnoty varování (12).



Obr. 22 Ukázka automatického e-mailového upozornění generovaného systémem Graylog při detekci brute-force útoku na server (Zdroj: Autor)

Tato šablona v doručené poště přehledně zobrazuje přesný čas incidentu, popis hrozby a identifikaci zdrojového zařízení včetně IP adresy útočníka. Klíčovou funkcí je zahrnutí tzv. backlogu zpráv, což je výpis konkrétních logů, které alarm bezprostředně vyvolaly. Díky tomuto technickému detailu může administrátor provést okamžitou analýzu závažnosti útoku přímo z e-mailové zprávy, aniž by se musel v první fázi přihlašovat do rozhraní Graylogu, což výrazně zrychluje proces incident response (12).

3.9 Strategie zálohování a Disaster Recovery

V poslední fázi realizace se autor zaměřil na kritický aspekt provozu každého bezpečnostního systému, zajištění integrity a dostupnosti dat v případě technické havárie nebo kompromitace hlavního uzlu. Byla implementována víceúrovňová strategie zálohování, která kombinuje statické denní snímky systému s real-time replikací kritických logů na oddělené úložiště.

3.9.1 Implementace dedikovaného úložného uzlu

Pro účely archivace byl v lokální síti zprovozněn sekundární server s IP adresou 192.168.2.104 (maturita-storage), běžící na minimalistické instalaci systému Debian. Aby byla zajištěna maximální bezpečnost a automatizace, autor implementoval autentizaci založenou na asymetrické kryptografii. Mezi hlavním Graylog serverem a úložným uzlem byl vytvořen důvěryhodný vztah pomocí SSH klíčů s délkou 4096 bitů, což umožňuje bezpečný přenos dat bez nutnosti uchovávat hesla v čitelném formátu v zálohovacích skriptech.

3.9.2 Automatizované zálohování indexů (Cold Backup)

Jádrem disaster recovery plánu je pravidelná archivace surových dat z datového uzlu. Autor vytvořil komplexní bash skript zalohuj_indexy.sh, který využívá nástroj tar s kompresním algoritmem gzip pro sbalení celého adresářového stromu /var/lib/graylog-datanode, kde jsou uloženy veškeré indexy. Vytvořený archiv je následně zabezpečeně odeslán pomocí protokolu SCP na sekundární uzel. Vzhledem k nepravidelnému provozu testovacích uzlů byl proces automatizovaného zálohování svěřen plánovači Anacron. Na rozdíl od standardního nástroje cron, který vyžaduje nepřetržitý běh systému v přesně definovaný čas, Anacron zajišťuje spuštění zálohovacího skriptu i v případě, že byl uzel v plánované době vypnut. Autor definoval úlohu s denní periodou. Tímto je garantována pravidelná aktualizace záloh bez nutnosti 24/7 provozu infrastruktury, přičemž je zachována efektivita přenosu dat mimo špičku (13).

```
root@maturita-storage:~# ls -lh /zalohy_graylog/indexy/
celkem 123M
-rw-r--r-- 1 root root 60M  1. led 18.45 zaloha_2026-01-01.tar.gz
-rw-r--r-- 1 root root 63M  2. led 13.26 zaloha_2026-01-02.tar.gz
```

Obr. 23 Výpis na serveru zobrazující pravidelné archivy databázových indexů pro potřeby dlouhodobého uchování dat (Zdroj: Autor)

3.9.3 Real-time Forwarding (Hot Backup)

Pro eliminaci rizika ztráty dat mezi jednotlivými denními zálohami autor implementoval tzv. „živý auditní log“. Tento mechanismus funguje na principu zrcadlení všech příchozích zpráv v reálném čase. V rozhraní Graylogu byl vytvořen nový výstup typu GELF Output na UDP portu 12201, který je směrován na IP adresu úložného serveru. Tento výstup byl následně napojen na centrální „Default Stream“, čímž bylo zajištěno, že každá zpráva o útoku je v momentě svého vzniku okamžitě replikována mimo hlavní server. Na straně úložného serveru byl zprovozněn proces využívající utilitu nc, který naslouchá na definovaném portu a příchozí data ukládá do souboru live_audit.json. Tento soubor slouží jako nezpochybnitelný důkaz v případě, že by se útočníkovi podařilo proniknout do hlavního systému a smazat místní logy (14).

```
root@maturita-storage:~# ls -lh /zalohy_graylog/zive_streamy/
celkem 12M
-rw-rw-rw- 1 root root 12M  2. led 16.33 live_audit.json
```

Obr. 24 Výpis na serveru zobrazující soubor pro živý audit událostí pro potřeby průběžné kontroly a analýzy (Zdroj: Autor)

3.9.4 Automatizovaná údržba a rotace logů

Vzhledem k vysoké intenzitě útoků na honeypot a s tím spojeným rychlým nárůstem objemu dat v živém archivu musel autor vyřešit otázku kapacity úložiště. Byla nasazena utilita logrotate, která spravuje narůstající soubor s auditním logem. Konfigurace využívá metodu copytruncate, která umožňuje odrotovat a zkomprimovat soubor po dosažení velikosti 100 MB bez nutnosti přerušit běžící proces zápisu. Tímto krokem autor zajistil, že systém zálohování je dlouhodobě udržitelný, autonomní a nevyžaduje manuální zásahy administrátora při zachování kontinuity sběru dat.

4 Uživatelská příručka

Tato kapitola slouží jako metodický návod pro obsluhu a základní údržbu vytvořeného systému. Administrátorské rozhraní je pro oprávněné osoby dostupné na vnitřní adrese portu 9000, kde je vyžadována autentizace pomocí administrátorských údajů definovaných během instalace. Po přihlášení má uživatel k dispozici hlavní ovládací panel, odkud může přistupovat k jednotlivým streamům a sledovat příchozí události v reálném čase.

Pro efektivní dohled nad bezpečností celé infrastruktury jsou v systému připraveny přehledové dashboardy, které v reálném čase vizualizují příchozí události. Správce má k dispozici grafické výstupy, jako jsou interaktivní mapy a statistické grafy, které umožňují okamžitou identifikaci aktuálních hrozeb a geografického původu pokusů o narušení. V případě potřeby detailního prošetření konkrétní aktivity lze využít vyhledávací pole a filtry, které umožňují rychle vyhledat záznamy podle IP adresy útočníka, cílového portu nebo uživatelského jména.

Pro ověření funkčnosti zálohovacího mechanismu může správce využít kontrolní příkaz ls -lh /zalohy_graylog/zive_streamy/ na úložném serveru, kde by měl pozorovat kontinuální nárůst velikosti auditního souboru. V případě obnovy dat po havárii jsou archivy připraveny v adresáři /zalohy_graylog/denni_indexy/ ve formátu .tar.gz, připravené k okamžité dekompresi a importu zpět do datového uzlu.

V rámci běžné údržby je doporučeno pravidelně kontrolovat stav systémových logů a vytížení zdrojů, aby byla zachována vysoká propustnost systému. V případě technických potíží nebo výpadku toku dat z agentů je nezbytné prověřit stav systémových služeb Graylogu a databáze přímo v terminálu serveru. Tato příručka tak poskytuje základní rámec pro efektivní využívání i osobami, které nebyly přímo zapojeny do jeho technické realizace.

5 Závěr

Cílem této maturitní práce bylo navrhnout a realizovat centralizovaný logovací systém pro sběr, zpracování a analýzu událostí z operačních systémů Windows a Linux. Tento cíl byl splněn vytvořením funkčního řešení založeného na platformě Graylog, doplněné o OpenSearch, MongoDB a další podpůrné nástroje.

V rámci práce byla navržena architektura logserveru a provedena instalace a konfigurace všech klíčových komponent systému. Pro sběr logů z koncových stanic a serverů byly využity nástroje Filebeat a Winlogbeat, které umožňují bezpečný a spolehlivý přenos dat do centrálního úložiště. Zpracování a třídění událostí je realizováno pomocí streamů a pipeline pravidel, což umožňuje efektivní filtrování a další analýzu dat.

Součástí řešení je také vizualizace dat pomocí přehledných dashboardů a základní nastavení upozornění na vybrané bezpečnostní události. Pro rozšíření detekčních schopností systému byl nasazen honeypot Cowrie, jehož výstupy jsou integrovány do logovací infrastruktury. Zabezpečený vzdálený přístup k systému je zajištěn prostřednictvím privátní sítě Tailscale.

Výsledkem práce je funkční centralizovaný logovací systém, který lze využít pro monitoring a základní bezpečnostní dohled v menším až středně velkém prostředí. Navržené řešení je možné dále rozšiřovat o pokročilé detekční mechanismy, automatizaci reakcí na incidenty nebo integraci s dalšími bezpečnostními nástroji. Práce tak splňuje stanovené cíle a potvrzuje praktické využití získaných znalostí v oblasti správy sítí a IT bezpečnosti.

Seznam zkratek

Zkratka	Význam
API	Application Programming Interface; rozhraní pro programování aplikací umožňující komunikaci mezi systémy.
ARM	Advanced RISC Machine; efektivní procesorová architektura využívaná v cloudové infrastruktuře Oracle.
CA	Certificate Authority; certifikační autorita vydávající digitální certifikáty pro šifrování vnitřní sítě.
EU	Evropská unie; politická a ekonomická unie členských států, která definuje legislativní rámec pro kybernetickou bezpečnost a ochranu dat.
GB	Gigabyte; jednotka kapacity digitálních dat (1024 MB), určuje velikost RAM nebo diskového prostoru.
GDPR	General Data Protection Regulation (Obecné nařízení o ochraně osobních údajů); nařízení Evropské unie, které ukládá povinnosti při zpracování a uchovávání osobních údajů, což se přímo dotýká správy a archivace systémových logů
GELF	Graylog Extended Log Format; strukturovaný formát logů navržený pro efektivní přenos zpráv v Graylogu.
GPG	GNU Privacy Guard; nástroj pro digitální podepisování a šifrování, použitý pro ověření balíčků MongoDB.
GZIP	GNU zip; softwarová utilita a formát souborů používaný pro kompresi záloh a archivů logů.
HASSH	Algoritmus pro tvorbu unikátních otisků (fingerprinting) SSH komunikace k identifikaci útočníků.
HTML	HyperText Markup Language; značkovací jazyk použitý pro tvorbu vizuálních šablon e-mailových notifikací.

HTTP	Hypertext Transfer Protocol; protokol pro přenos nešifrovaného webového obsahu.
HTTPS	Hypertext Transfer Protocol Secure; zabezpečená verze protokolu HTTP využívající šifrování TLS.
ICMP	Internet Control Message Protocol; protokol pro diagnostiku sítě a zasílání chybových hlášení (např. ping).
ID	Identifier; unikátní identifikační číslo entity v systému (např. ID streamu nebo zprávy).
IP	Internet Protocol; základní protokol síťové vrstvy zajišťující adresaci a směrování v síti.
IT	Information Technology; informační technologie; obor zabývající se správou a zpracováním dat.
JSON	JavaScript Object Notation; textový formát pro výměnu dat, ve kterém ukládá záznamy honeypot Cowrie.
JVM	Java Virtual Machine; virtuální stroj umožňující běh aplikací v jazyce Java (jádro Graylogu).
MB	Megabyte; jednotka kapacity dat; v práci použitá pro nastavení limitů rotace a velikosti logů.
NC	Netcat; síťový nástroj pro čtení a zápis dat napříč síťovými spojeními TCP/UDP.
NoSQL	Not Only SQL; kategorie databází (např. MongoDB), které nevyužívají klasický relační model.
OS	Operating System; operační systém; základní programové vybavení serveru (např. Debian 12).
RAM	Random Access Memory; operační paměť serveru nezbytná pro rychlé zpracování dat v reálném čase.

RDP	Remote Desktop Protocol; protokol vzdálené plochy Windows simulovaný na honeypotu jako cíl útoku.
SCP	Secure Copy Protocol; protokol pro bezpečný přenos souborů mezi servery využívající šifrování SSH.
SHA-256	Secure Hash Algorithm 256; kryptografická hashovací funkce použitá pro bezpečné uložení hesel.
SIEM	Security Information and Event Management; systém pro centralizovaný sběr a analýzu bezpečnosti.
SMTP	Simple Mail Transfer Protocol; protokol pro odesílání e-mailových varovných upozornění správci.
SSH	Secure Shell; protokol pro zabezpečené vzdálené ovládání serverů a šifrovaný přenos dat.
TCP	Transmission Control Protocol; transportní protokol zajišťující spolehlivý a potvrzovaný přenos dat.
TLS	Transport Layer Security; kryptografický protokol pro zabezpečení komunikace mezi agenty a serverem.
UDP	User Datagram Protocol; rychlý nespojovaný protokol použitý pro real-time odesílání auditních logů.
VPN	Virtual Private Network; virtuální privátní síť pro šifrované propojení cloudu a LAN přes Tailscale.
VPS	Virtual Private Server; virtuální server běžící v cloudovém prostředí (Oracle Cloud).

Zdroje

Při přípravě této práce autor použil nástroj Gemini (Google) za účelem optimalizace textové formulace a zvýšení srozumitelnosti odborného výkladu. Po použití této služby autor obsah podle potřeby zkontroloval a upravil a přebírá plnou odpovědnost za obsah publikace.

1. **GRAYLOG, Inc.** Installing Graylog. *Graylog Documentation.* [Online] [Cited: 18 Listopad 2025.] https://go2docs.graylog.org/current/downloading_and_installing_graylog/installing_graylog.html.

2. **ELASTIC.** Filebeat Guide. *Elastic Docs.* [Online] [Cited: 20 Listopad 2025.] https://www.elastic.co/docs/reference/beats/filebeat.

3. —. Winlogbeat installation and configuration. *Elastic Docs.* [Online] [Cited: 20 Listopad 2025.] https://www.elastic.co/docs/reference/beats/winlogbeat/winlogbeat-installation-configuration.

4. **ORACLE.** Tutorial - Launching Your First Linux Instance. *Oracle Help Center.* [Online] [Cited: 19 Prosinec 2025.] https://docs.oracle.com/en-us/iaas/Content/Compute/tutorials/first-linux-instance/overview.htm.

5. —. Adding Ingress Rules to a Compute Instance. *Oracle Help Center.* [Online] [Cited: 19 Prosinec 2025.] https://docs.oracle.com/en-us/iaas/mysql-database/doc/adding-ingress-rules-compute-instance-bastion-session-or-vpn-connection.html.

6. **TAILSCALE.** What is Tailscale? *Tailscale Docs.* [Online] [Cited: 19 Prosinec 2025.] https://tailscale.com/kb/1151/what-is-tailscale.

7. **COWRIE.** Docker Documentation (Cowrie Honeypot). *GitHub Documentation.* [Online] [Cited: 20 Prosinec 2025.] https://github.com/cowrie/cowrie/tree/main/docs/docker.

8. **GRAYLOG, Inc.** Streams and Stream Rules. *Graylog Documentation.* [Online] [Cited: 21 Prosinec 2025.] https://go2docs.graylog.org/current/making_sense_of_your_log_data/streams.html.

9. —. Pipelines and Pipeline Usage. *Graylog Documentation.* [Online] [Cited: 22 Prosinec 2025.] https://go2docs.graylog.org/current/making_sense_of_your_log_data/pipeline_usage.html.

10. —. Rules and Functions. *Graylog Documentation.* [Online] [Cited: 22 Prosinec 2025.] https://go2docs.graylog.org/current/making_sense_of_your_log_data/rules.html.

11. —. Lookup Tables. *Graylog Documentation.* [Online] [Cited: 28 Prosinec 2025.] https://go2docs.graylog.org/current/making_sense_of_your_log_data/lookup_tables.html.

12. —. Alerts and Notifications. *Graylog Documentation.* [Online] [Cited: 20 Prosinec 2025.] https://go2docs.graylog.org/current/interacting_with_your_log_data/alerts.html.

13. —. Index Rotation Configuration. *Graylog Documentation.* [Online] [Cited: 30 Prosinec 2025.] https://go2docs.graylog.org/current/setting_up_graylog/index_model.html#IndexRotationConfiguration.

14. —. Outputs. *Graylog Documentation.* [Online] [Cited: 1 Leden 2026.] https://go2docs.graylog.org/current/interacting_with_your_log_data/outputs.html.

Seznam obrázků