Systém detekce průniku v linuxovém serveru

Bc. Marek Cigánek

Diplomová práce
2024

Univerzita Tomáše Bati ve Zlíně
Fakulta aplikované informatiky

Univerzita Tomáše Bati ve Zlíně
Fakulta aplikované informatiky
Ústav elektroniky a měření

Akademický rok: 2023/2024

ZADÁNÍ DIPLOMOVÉ PRÁCE

(projektu, uměleckého díla, uměleckého výkonu)

Jméno a příjmení:	**Bc. Marek Cigánek**
Osobní číslo:	**A22354**
Studijní program:	**N1032A020003 Bezpečnostní technologie, systémy a management**
Specializace:	**Bezpečnostní technologie**
Forma studia:	**Kombinovaná**
Téma práce:	**Systém detekce průniku v Linuxovém serveru**
Téma práce anglicky:	**An Intrusion Detection System in a Linux Server**

Zásady pro vypracování

1. Vypracujte literární rešerši na téma systémy detekce průniku.
2. Navrhněte pravidla detekce a implementaci řešení.
3. Nainstalujte a nakonfigurujte Linuxový server s využitím Ansible automatizace.
4. Věnujte pozornost zabezpečení celého systému.
5. Nastavte systém detekce průniku na detekci lokálních a vzdálených nestandardních aktivit.
6. Generujte přehled o bezpečnostních incidentech.
7. Ověřte navržené řešení pomocí penetračních testů s následným vyhodnocením a návrhem možných protiopatření.

Forma zpracování diplomové práce: **tištěná/elektronická**

Seznam doporučené literatury:

1. HOWARD, Michael a LEBLANC, David. *Bezpečný kód: [techniky a strategie tvorby bezpečných webových aplikací]*. Brno: Computer Press, 2008. ISBN 978-80-251-2050-7.
2. TEVAULT, Donald A. *Mastering Linux Security and Hardening*. 3rd ed. Birmingham: Packt Publishing, Limited, 2023. ISBN 978-1-83763-051-6.
3. SWARNKAR, Mayank a RAJPUT, Shyam Singh. *Artificial intelligence for intrusion detection systems*. 2024. Boca Raton, FL: CRC Press. ISBN 978-1-032-38665-2
4. RED HAT, INC. *Ansible*. Online. C2023. Dostupné z: http://www.ansible.com. [cit. 2023-11-14].
5. HICKEY, Matthew a ARCURI, Jennifer. *Hands on hacking*. Indianapolis, Indiana: Wiley, 2020. ISBN 978-1-119-56145-3.
6. BISHOP, Matt; SULLIVAN, Elisabeth a RUPPEL, Michelle. *Computer security: Art and science*. Second. New York: Addison--Wesley, 2019. ISBN 978-0-321-71233-2.
7. PATHAN, Al-Sakib Khan. *The State of the Art in Intrusion Prevention and Detection*. Auerbach Publications, 2014. ISBN 978-1-4822-0351-6.
8. VAN OORSCHOT, Paul C. *Computer security and the internet: Tools and jewels from malware to bitcoin*. Second. Springer, 2021. ISBN 978-3-030-83410-4.

Vedoucí diplomové práce: **doc. Ing. Martin Sysel, Ph.D.**
Ústav počítačových a komunikačních systémů

Datum zadání diplomové práce: **20. listopadu 2023**
Termín odevzdání diplomové práce: **28. května 2024**

doc. Ing. Jiří Vojtěšek, Ph.D. v.r.
děkan

UNIVERZITA TOMÁŠE BATI VE ZLÍNĚ
Fakulta aplikované informatiky
L.S.

Ing. Milan Navrátil, Ph.D. v.r.
ředitel ústavu

Ve Zlíně dne 1. prosince 2023

Prohlašuji, že

- beru na vědomí, že odevzdáním diplomové práce souhlasím se zveřejněním své práce podle zákona č. 111/1998 Sb. o vysokých školách a o změně a doplnění dalších zákonů (zákon o vysokých školách), ve znění pozdějších právních předpisů, bez ohledu na výsledek obhajoby;
- beru na vědomí, že diplomová práce bude uložena v elektronické podobě v univerzitním informačním systému dostupná k prezenčnímu nahlédnutí, že jeden výtisk diplomové práce bude uložen v příruční knihovně Fakulty aplikované informatiky Univerzity Tomáše Bati ve Zlíně;
- byl/a jsem seznámen/a s tím, že na moji diplomovou práci se plně vztahuje zákon č. 121/2000 Sb. o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon) ve znění pozdějších právních předpisů, zejm. § 35 odst. 3;
- beru na vědomí, že podle § 60 odst. 1 autorského zákona má UTB ve Zlíně právo na uzavření licenční smlouvy o užití školního díla v rozsahu § 12 odst. 4 autorského zákona;
- beru na vědomí, že podle § 60 odst. 2 a 3 autorského zákona mohu užít své dílo – diplomovou práci nebo poskytnout licenci k jejímu využití jen připouští-li tak licenční smlouva uzavřená mezi mnou a Univerzitou Tomáše Bati ve Zlíně s tím, že vyrovnání případného přiměřeného příspěvku na úhradu nákladů, které byly Univerzitou Tomáše Bati ve Zlíně na vytvoření díla vynaloženy (až do jejich skutečné výše) bude rovněž předmětem této licenční smlouvy;
- beru na vědomí, že pokud bylo k vypracování diplomové práce využito softwaru poskytnutého Univerzitou Tomáše Bati ve Zlíně nebo jinými subjekty pouze ke studijním a výzkumným účelům (tedy pouze k nekomerčnímu využití), nelze výsledky diplomové práce využít ke komerčním účelům;
- beru na vědomí, že pokud je výstupem diplomové práce jakýkoliv softwarový produkt, považují se za součást práce rovněž i zdrojové kódy, popř. soubory, ze kterých se projekt skládá. Neodevzdání této součásti může být důvodem k neobhájení práce.

Prohlašuji,

- že jsem na diplomové práci pracoval samostatně a použitou literaturu jsem citoval. V případě publikace výsledků budu uveden jako spoluautor.
- že odevzdaná verze diplomové práce a verze elektronická nahraná do IS/STAG jsou totožné.

Ve Zlíně, dne

Marek Cigánek v.r.
podpis studenta

ABSTRAKT

Diplomová práce se věnuje problematice systémů detekce průniku na koncových klientech s linuxovým operačním systémem. Zabývá se implementací instalace a zabezpečení klientského operačního systému prostřednictvím softwarového nástroje konfiguračního managementu. Dále se věnuje výběru vhodného řešení pro detekci průniku na linuxových operačních systémech a následně realizuje jeho implementaci a konfiguraci. V následující části se zabývá otestováním navrženého řešení prostřednictvím penetračního testování. V závěru jsou vyhodnoceny výsledky detekce se zaměřením na problémy vzniklé během testování. Jsou navržena vhodná protiopatření, která jsou taktéž otestována s vyhodnocením výsledků.

Klíčová slova: HIDS, bezpečnost, Linux, Rocky Linux, Kali Linux, penetrační testy, Ansible, Wazuh, Nmap, OpenVAS

ABSTRACT

The thesis deals with the problem of intrusion detection systems on end clients with Linux operating system. It deals with the implementation of the installation and security of the client operating system through a configuration management software tool. It also discusses the selection of a suitable solution for intrusion detection on Linux operating systems and then its implementation and configuration. The next section deals with the testing of the proposed solution through penetration testing.

Finally, the detection results are evaluated with a focus on the problems encountered during testing. Appropriate countermeasures are proposed, which are also tested with evaluation of the results.

Keywords: HIDS, security, Linux, Rocky Linux, Kali Linux, penetration testing, Ansible, Wazuh, Nmap, OpenVAS

Rád bych touto cestou poděkoval vedoucímu práce panu doc. Ing. Martinu Syslovi, Ph.D. za odborné vedení a konzultace během vypracování této diplomové práce. Tímto bych chtěl také poděkovat své rodině a přátelům za podporu během studia.

OBSAH

ÚVOD

Systémy detekce průniku jsou stále velmi důležitým komponentem v rámci zabezpečení informační tak i fyzické bezpečnosti. Tato práce se věnuje systémům detekce v počítačové infrastruktuře na koncových stanicích, na nichž je provozován operační systém Linux. Koncovými stanicemi mohou být jak pracovní stanice, tak i serverová infrastruktura. Úkolem systémů detekce je včas detekovat veškeré pokusy o proniknutí.

Práce se zabývá zabezpečením operačního systému Linux, návrhem pravidel detekce, výběrem vhodného detekčního systému a jeho instalací. Dále penetračním testováním, vyhodnocením výsledků, a nakonec návrhem řešení.

I. TEORETICKÁ ČÁST

1 ÚVOD DO PROBLEMATIKY

Yost [1] definuje systémy detekce narušení jako "*Systémy detekce narušení jsou softwarové systémy, které monitorují auditní data o chování uživatelů počítače (a další události počítačového systému), aby odhalily a označily potenciální neoprávněný přístup k počítačovému systému a jeho nevhodné používání.*[1]".

Hlavní činností detekčních systémů je monitorování typických činností uživatelů na operačních systémech nebo monitorování provozu počítačové sítě. Systém kontroluje pomocí definovaných pravidel, zda nedochází k jejich porušování a jestli se v operačním systému se nevyskytuje škodlivý software jakým mohou být počítačové viry, spyware, keyloggery atd.

RFC 4949 definuje systém IDS (Intrusion Detection Systems) jako [2]: *„Proces nebo subsystém implementovaný v softwaru nebo hardwaru, který automatizuje úkoly a) monitorování událostí v počítačové síti a b) jejich analýzy z hlediska příznaků bezpečnostních problémů.*[2]".

Detekce průniku je [2]: „*(I) Bezpečnostní služba, která sleduje a analyzuje systémové události za účelem vyhledávání pokusů o neoprávněný přístup k systémovým prostředkům v reálném čase nebo téměř v reálném čase a poskytování varování před těmito pokusy*[3].".

IDS je pasivním typem obrany a jeho hlavní úkoly jsou [1;2]:

- Sledování aktivit v síti.
- Detekce bezpečnostních problémů (pokusy o narušení).
- Detekce virů a škodlivý software (spyware, krádeže hesel).

---

[1] Intrusion-detection systems are software-based systems that monitor computer user behavior audit data (and other computer system events) to detect and flag potential unauthorized access to and inappropriate use of a computer system. [1]

[2] A process or subsystem, implemented in software or hardware, that automates the tasks of (a) monitoring events that occur in a computer network and (b) analyzing them for signs of security problems. [2]

[3] (I) A security service that monitors and analyzes system events for the purpose of finding, and providing real-time or near real-time warning of, attempts to access system resources in an unauthorized manner. [2]

1.1 Historie

S nástupem sálových počítačů si lidé začínali uvědomovat potenciální problémy se zabezpečením těchto systémů [1].

1.1.1 Počátky vzniku detekčních systémů

Ve Spojených státech amerických přichází v říjnu roku 1972 James P. Anderson s kolektivem s dokumentem "*COMPUTER SECURITY TECHNOLOGY PLANNING STUDY*"., který se zabývá možnými problémy počítačové bezpečnosti. Dokument se zabývá analýzou bezpečnostních hrozeb a technik průniku, technikami víceuživatelské počítačové bezpečnosti. Výsledkem práce je vypracovaný plán řešení jednotlivých technických oblastí, jejich integrace do konečného návrhu pro splnění požadavků bezpečnosti víceuživatelských a víceúrovňových zabezpečení počítačových systémů, terminálů pro americké letectvo [3].

V roce 1980 James P. Anderson zveřejňuje studii na zlepšení řešení auditu a dohledu nad počítačovou bezpečností systémů pro bezpečnostní pracovníky. Studie se zabývá například monitorováním aktivit uživatelů, dále dohledem nad souborovým systémem a zařízeními a následným auditem těchto záznamů. V tomto dokumentu položil Anderson základy definování detekčních systémů [4].

IDES (Intrusion Detection Expert System) je prvním modelem detekčního systému IDS, který byl navržený a vyvinutý Dorothou E. Denningovou a Peterem Neumannem v letech 1984 až 1986. Systém detekoval známé zranitelnosti (pokusy o vniknutí do systému, maskování nebo úspěšné průniky do systémů, trojský kůň, virus, DoS (Denial of Service definovaný jako „*kybernetický útok, který má za cíl omezit nebo vyřadit služby počítačových systémů.*" [5]), průnik legitimního uživatele, únik informací způsobený legitimním uživatelem) statistickým vyhodnocování abnormálního chování uživatelů. IDES používal hybridní architekturu – detekci anomálii a expertní systém. Expertní systém používal systém pravidel k detekci známých bezpečnostních narušení [6].

V roce 1987 Denningová [7] navrhla model, který obsahuje šest hlavních částí:

- Subjekty:
    - Iniciátoři činnosti v cílovém systému – obvykle uživatelé.
- Objekty:
    - Uživatelé, kteří využívají služby: Prostředky spravované systémem (soubory, příkazy, zařízení atd).

- Záznamy o auditu:
    - Záznamy: Generované cílovým systémem v reakci na akce, které subjekty provedly nebo se pokusily provést na objektech (přihlášení uživatele, provedení příkazu, přístup k souboru atd.)
- Profily:
    - Profily: Struktury, které charakterizují chování subjektů vůči objektům z hlediska statistických metrik a modelů pozorované činnosti. Profily jsou automaticky generovány a inicializovány ze šablon.
- Záznamy o anomáliích: Generují se při zjištění abnormálního chování.
- Pravidla aktivity: Akce prováděné při splnění určité podmínky, které aktualizují profily, detekují abnormální chování, vztahují anomálie k podezření na narušení a vytvářejí zprávy.

Dále se uvádí v [7], že základní myšlenkou je sledování standardních operací v operačním systému a hledání odchylek vůči normálnímu používání. Mezi tyto operace patřilo např. přihlašování, spouštění příkazů, programů, přístupy k souborům, zařízením. Jednalo se o jednoduchý model, který nepřepokládal žádné bezpečnostní chyby v těchto systémech, neobsahoval konkrétní informace o jejich bezpečnostních mechanismech [7].

1.1.2 MIDAS

Detekční systém MIDAS (Multics Intrusion Detection and Alerting System) patří mezi detekční systémy a byl vyvinutý v roce 1988 v National Computer Security Centre, ve spolupráci s laboratoří výpočetní techniky SRI International. Cílem projektu bylo vytvořit systém detekce pro sálový počítač Honeywell DPS-8/70 Dockmaster s operačním systémem Multics. Systém Dockmaster monitoroval příkazy, shromažďoval auditní data a odesílal je do expertního systému k analýze [8].

MIDAS byl vyvinut s pomocí systému P-BEST (Production Based Expert System Toolset), jedná se o expertní systém, („*Podle E. Feigenbauma je expertní systém inteligentní počítačový program, který užívá znalosti a inferenční procedury k řešeni problémů, které jsou natolik obtížné, že pro své řešení vyžadují významnou lidskou expertízu“* [9]), který poskytoval možnosti pro vytváření detekčních pravidel [10].

1.1.3 W&S (Wisdom and Sense)

Detekční systém W&S vznikal mezi roky 1984-1987 v LANL (Los Alamos National Laboratory). Jeho původní určení bylo odhalování chyb v inventarizaci materiálů, později byl systém rozšířen o identifikaci neobvyklých transakcí v materiálovém účetnictví. Nakonec byl systém přepsán, tak aby se věnoval počítačové bezpečnosti, kde se zaměřoval na auditní záznamy [11].

Systém je založen na statistických datech, na základě historických data generuje stromovou strukturu pravidel, podle kterých probíhá detekce anomálií. Subsystém systému Wisdom obsahuje soubor pravidel, která popisují běžné chování systému na základě historických dat z auditu. Jedná se přibližně o soubor s 10 000 záznamy o uživateli, ze kterých se tvoří pravidla. Subsystém Sense je expertní systém, založený na pravidlech ze systému Widsdom a ověřuje auditní data, jestli nedochází k porušování pravidel [12].

1.1.4 Haystack

Systém Haystack, prototyp detekčního systému, byl vyvinut pro potřeby víceuživatelských počítačových systémů amerického letectva Air Force v roce 1988 na mainframe systémech Unisys 1100/60 s operačním systémem OS/1100. Jednalo se behaviorálně založený systém, který generoval každý den reporty o detekovaných anomáliích. Prováděl analýzu činností uživatelů podle předem definovaných údajů podle bezpečnostních politik určené pro vojenské systémy.

Definoval šest typů průniků do systému [13]:

- Pokus o průnik neautorizovanými osobami.
- Maskovaný útok (útočník předstírá, že je tím, kým není).
- Průnik do bezpečnostního systému, únik informací.
- DoS.
- Škodlivé používání systému.

Jednalo se o tzv. offline systém, logy z mainframe systému se pomocí magnetické pásky přenášely a analyzovaly v tzv. host systému [13].

1.1.5 NIDES

Detekční systém NIDES (Next-Generation Intrusion Detection Expert System) byl vyvinutý v roce 1995 v laboratoři výpočetní techniky CSI (Computer Science Laboratory) ve SRI

International. Cílem toho projektu bylo vyvinout systém, který je schopen v reálném čase monitorovat chovaní uživatelů a detekovat podezřelé chování [14].

NIDES vychází ze systému IDES a je stejně jako systém IDES založený na technikách detekcí anomálií a expertního systému. Expertní systém obsahoval pravidla obsahující známé případy narušení, zranitelnosti systému a další porušení systémové a bezpečnostní politiky.

NIDES fungoval v architektuře klient – server, analýza probíhala na tzv. NIDES host systému, který nebyl monitorován, ale sám monitoroval koncové systémy přes síť Ethernet. Koncové systémy zasílaly systému NIDES host záznamy z auditů.

Systém obsahoval několik typů scénářů např. podvržení hostitelských jmen, podvržení IP adresy, a techniku hádání hesel Doorknob Twisting [4] [15].

1.1.6 Tripwire

„*Tripwire je program pro kontrolu integrity napsaný pro prostředí UNIX, který dává správcům systému možnost sledovat souborové systémy z hlediska přidaných, odstraněných a změněných souborů.*[5]“ [16]

Program Tripwire byl navržen v roce 1992 a je dalším detekčním systémem, který vznikl v 90. letech [17].

Cílem programu Tripwire je sledování souborů a adresářů, u nichž nedochází k výrazným změnám. Tripwire používá k detekci změn souborů kontrolní součty (hash), podpisy vytváří si databázi atributů souborů, adresářů a jejich signatur. Tuto databázi používá ke kontrole změn a pokud k nim dojde tak informuje správce. Program Tripwire je dostupný jako komerční nebo jako volně šiřitelný software [17].

[4] Doorknob twisting refers to service access attempts that may be harmless individually but taken together probably represent a break-in attempt.[15]

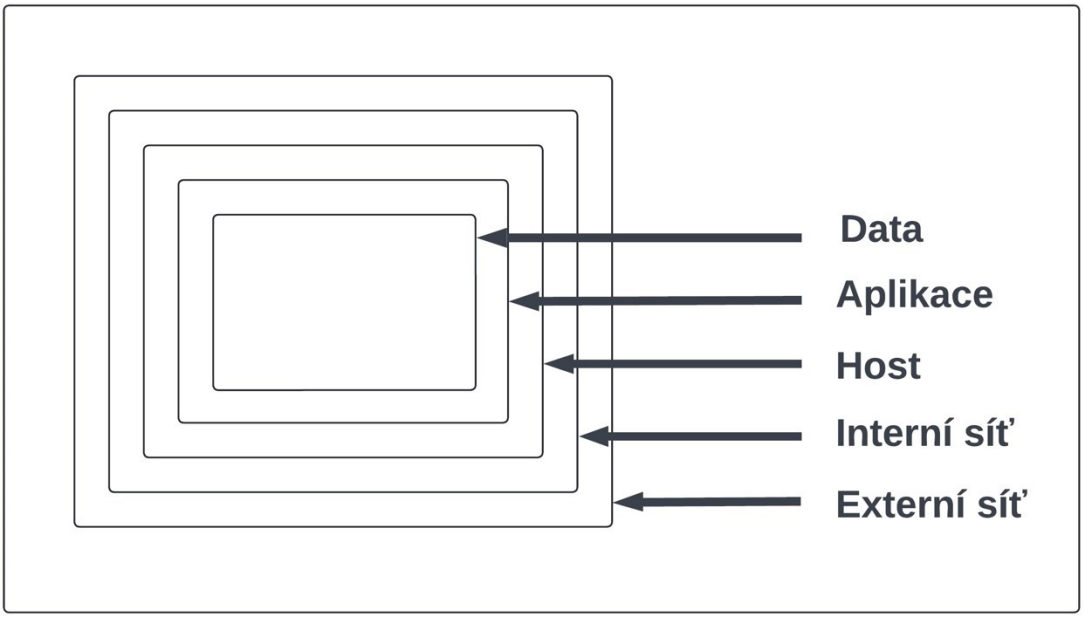
[5] Tripwire is an integrity checking program written for the UNIX environment that gives system administrators the ability to monitor file systems for added, deleted, and modified files.[16]

1.2 Hloubková ochrana (Defense in depth)

Hloubková ochrana zajišťuje celkovou ochranu informací organizace prostřednictvím několika vrstev obrany, kdy každá vrstva obsahuje ochranné mechanismy. Cílem této ochrany je zajištění důvěrnosti, integrity a dostupnosti dat (CIA– Confidentiality Integrity Accessibility) [18]. Celkově jsou všechny vrstvy schopné zabránit potenciálním kybernetickým hrozbám mnohem lépe než jednotlivá opatření (Obrázek 1).

Ochrana je rozdělena na následující části [18]:

- perimetr (např. fyzické zabezpečení – senzory),
- externí síť (např firewall),
- interní síť (např. IDS, Honeypot, segmentace sítě),
- klienti (např. HIDS, EDR, firewall, antivirus),
- aplikace,
- data.



Obrázek 1. Hloubková ochrana [18]

1.3 Komponenty IDS systémů

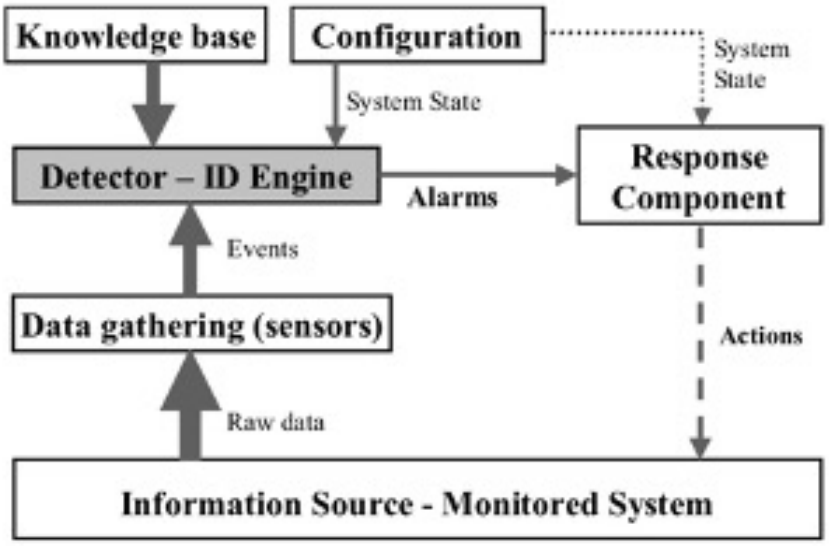
Glass-Vanderlan [19] uvádí, že obecný IDS systém se skládá ze tří hlavních funkčních částí:

- Sběr dat – jeden nebo více datových typů (např. systémová volání).

- Konverze – převod na vybranou funkcí (např. systémové volání se reprezentuje jako seznam atributů).
- Rozhodování – algoritmus, heuristika rozhoduje o tom, zda data obsahují příznaky vektoru útoku.

Lazarevic [20] rozděluje architekturu IDS systému na pět částí:

- Senzor – zařízení/agent, který provádí sběr dat.
- Detektor – zpracovává data ze senzoru a identifikuje hrozby.
- Databáze – obsahuje například signatury kybernetických útoků (data jsou předzpracována ze senzorů).
- Konfigurace – obsahuje informace o stavu IDS systému.
- Reakce – reaguje prostřednictvím akcí v případě detekce průniku.



Obrázek 2.Architektura IDS systému [20]

1.4 Techniky detekce

Detekční systémy pracují na dvou základních principech a těmi jsou detekce signatur tzv. SIDS (Signature Intrusion Detection Systems) systémy a systémy na detekci anomálií tzv. AIDS (Anomaly Intrusion Detection Systems).

Dále existují hybridní systémy, které kombinují obě techniky – detekci signatur a detekci anomálií [21;22;23].

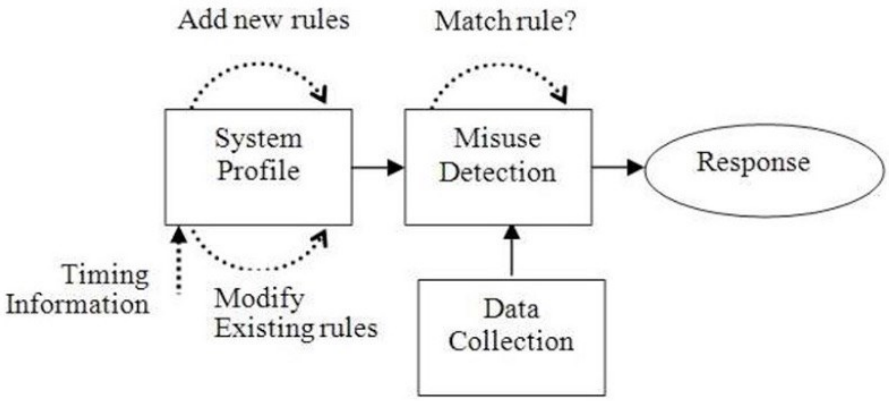
1.4.1 Detekce signatur (SIDS)

První detekční metodou je detekce signatur, která je také uváděna jako detekce zneužití nebo také jako heuristická identifikace na základě pravidel. Metoda pracuje na základě porovnávání signatur s databází známých útoků a je závislá na pravidelné aktualizaci této databáze [24].

Signatury se také nazývají jako IOC (Indicators of Compromise) – indikátory ohrožení [25].

Pathan [26] dále rozděluje detekci signatur na:

- Modelování stavů.
- Expertní systém (množina pravidel, která popisují chování při útoku).



Obrázek 3. Architektura pravidel [23]

Detekci pomocí signatur rozděluje Swarnkar [27] následovně:

- Informační – druhy signatur poskytující informace (např. TCP/UDP spojení).
- Průzkum – druhy signatur spouštěných útoky (např. skenování portů).
- Přístup – druhy signatur souvisejících s pokusy o neoprávněný přístup, zvýšení uživatelských oprávnění.
- Odmítnutí služby – druhy signatur vyvolaných útoky na odepření služeb (např. DoS, TCP SYN, Ping of Death).

Výhody [21;27]:

- Rychlá a efektivní detekce známých útoků,
- Detekce známých útoků s nízkým počtem falešně pozitivních poplachů.

Nevýhody [21;27]:

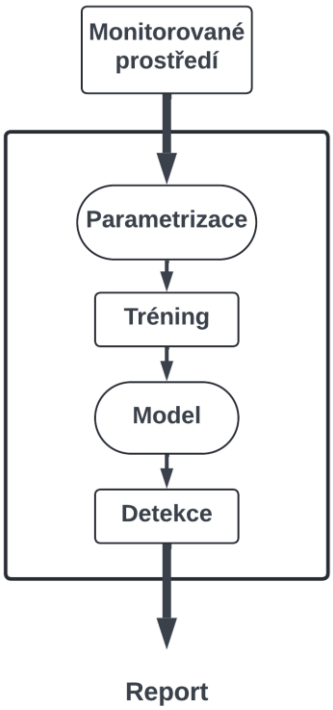
- Neumí detekovat neznámé a nové typy útoků (např. zero-day), než ty, které má uloženy ve své databázi signatur.
- Lze obejít detekci známých hrozeb pozměněním její signatury.
- Nutná častá aktualizace signatur.
- Špatná detekce útoků, které mají více částí.

1.4.2 Detekce anomálií (AIDS)

Systém reaguje na odchylky – anomálie [28] od normálního stavu. Detekce anomálií se nejprve trénuje sledováním běžných aktivit, které vytvoří profily standardního chování z historických dat po určitou dobu. Systém pak detekuje odchylky od běžného stavu.

Agrawal & Agrawal [29] uvádí metodiku detekce anomálií:

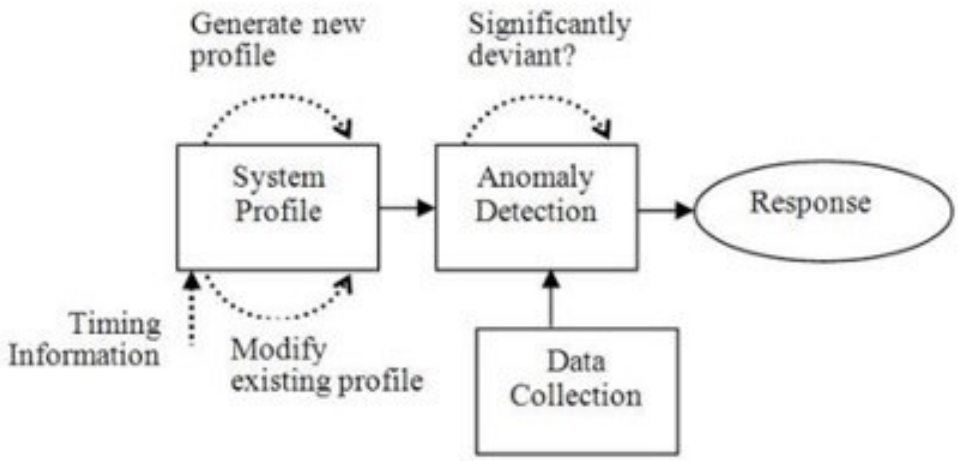
- Parametrizace – je proces zpracování dat, tak aby byla přijatelná nebo v souladu s cílovým systémem.
- Učení – je vytvoření modelu na bázi normálního nebo abnormálního chování systému.
- Detekce – je porovnávání vytvořeného modelu s provozem systému.



Obrázek 4. Metodologie detekce anomálií [29]

Stallings v [24], kde se odvolává na [30] rozděluje klasifikace přístupů na:

- Statistický – pozorování chovaní na základě metrik.
- Znalostní – využití expertního systému, který provádí klasifikaci.
- Strojové učení – určuje klasifikační model z dat, které byly použity při tréningu modelu.



Obrázek 5. Architektura systému detekce anomálií [23]

Strojové učení

V AIDS se dále používají metody strojového učení, které Stallings [24] rozděluje na:

- bayesovské sítě,
- markovovy modely,
- neuronové sítě,
- fuzzy logika,
- genetické algoritmy,
- shlukování a detekce odlehlých hodnot.

Metody využívané k detekci anomálii pomocí strojové učení Swarnkar rozděluje [27] na:

- statistické techniky,
- bayesovské sítě,
- neuronové sítě,
- data mining.

Agrawal & Agrawal [29] rozšiřuje techniky strojového učení o:

- klasifikační strom,
- fuzzy logika,
- genetické algoritmy,
- stroj s podpůrnými vektory.

Výhody [21;22]:

- AIDS lze použít k vytvoření signatury narušení.
- Mohly by být použity k odhalení nových útoků.

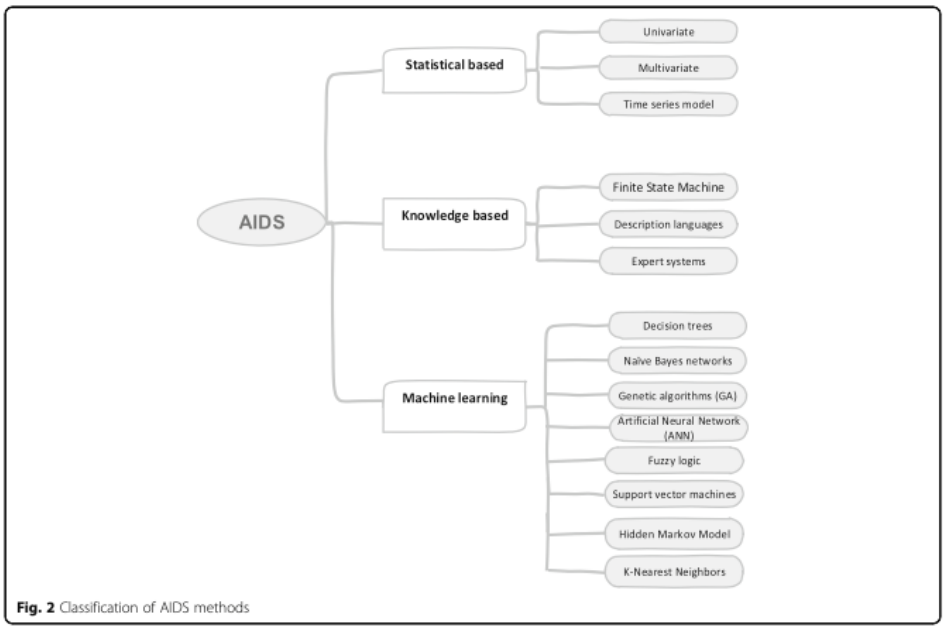
Nevýhody [21;22]:

- Neumí zpracovat šifrovanou komunikaci nebo obfuskací.
- Vysoký počet falešně pozitivních poplachů.
- Nutnost počátečního tréningu.
- Neklasifikované výstrahy.
- Nedokáže rozlišit mezi různými typy útoků.

Stallings [24] uvádí, že další nevýhodou je tréning těchto systémů pouze na legálních datech. Tudíž dochází k závěru, že [24]: *„Nedostatek anomálních trénovacích dat, ke kterým dochází vzhledem k touze odhalit aktuálně neznámé budoucí útoky, omezuje účinnost některých výše uvedených technik.*[6]“.

Celkové rozdělení AIDS systémů je zobrazeno na Obrázku 6.

[6] The lack of anomalous training data, which occurs given the desire to detect currently unknown future attacks, limits the effectiveness of some of the techniques listed above. [24]

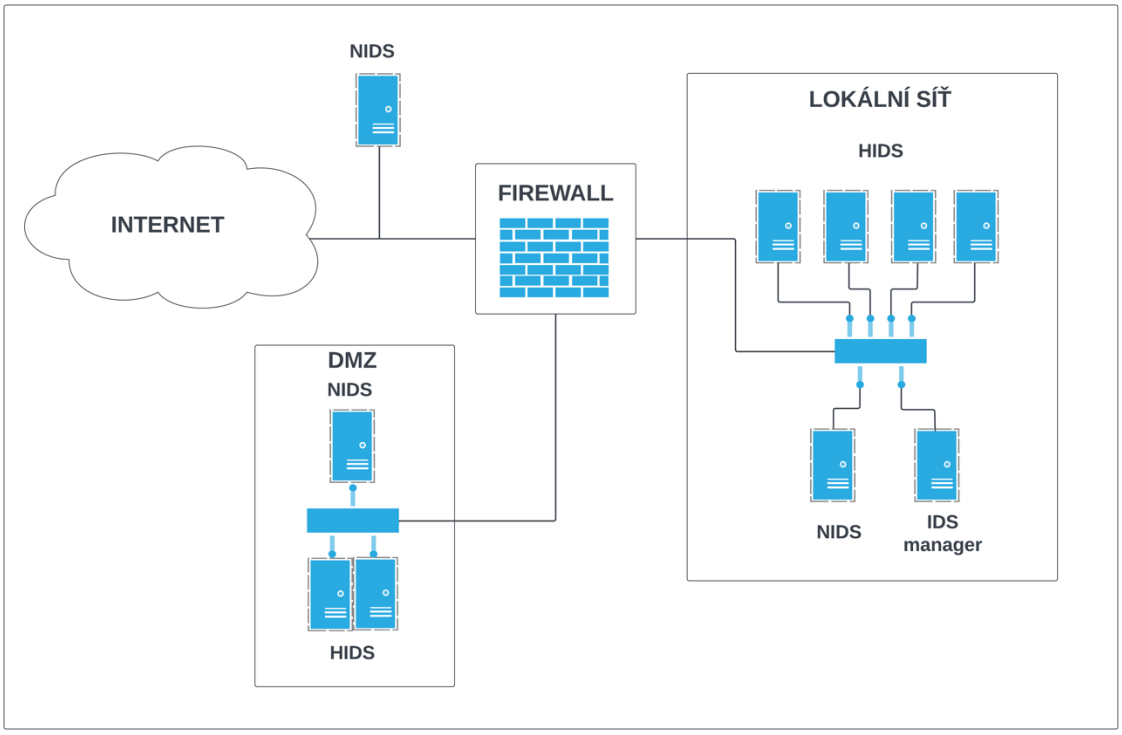


Obrázek 6. Rozdělení AIDS metod [21]

1.5 IDS rozdělení

IDS systémy se podle umístění rozdělují na dvě skupiny (Obrázek 7) [24]:

- NIDS (Network-based Intrusion Detection System).
- HIDS (Host-based Intrusion Detection System).



Obrázek 7. Architektura IDS systému [26]

1.5.1 NIDS

NIDS (Network-based Intrusion Detection System), je síťový systém detekce průniku, který sleduje provoz na lokální síti.

NIDS systém se obvykle umísťuje na následujících segmentech sítě [31]:

- DMZ (Demilitarized Zone) /perimetr,
- wifi síť,
- virtualizovaná síť,
- ostatní segmenty sítě.

Výhody [21;26]:

- Není nutná instalace na každého klienta.
- Možná kontrola různých zdrojů dat z klientů současně.
- Nemá vliv na síťový provoz.
- Detekce velkého rozsahu síťových protokolů.
- Nezávislost na operačním systému.

Nevýhody [21;26]:

- Problém s monitorování šifrované komunikace.
- Podpora pouze síťových útoků.
- Problém s analýzou rychlé komunikace.
- Nároky na kapacitu diskového úložiště.
- Omezené možnosti se sledováním provozu u sítí s vysokým počtem přepínačů a síťových segmentů.

1.5.2 HIDS

HIDS (Host-based Intrusion Detection System), jsou klientské systémy detekce průniku, které se principiálně zabývají daty na klientských systémech. Datovými zdroji se rozumí logy z operačního systému, linuxový auditní démon – *auditd*, systémová volání, příchozí nebo odchozí pakety [21;22].

Výhody [21]:

- Kontroluje šifrovanou komunikaci koncových stanic (end-to-end).

- Detekuje narušení kontrolou integrity souborů, systémových volání, síťové komunikace, podezřelé procesy.

Nevýhody [21]:

- Využívá hardwarové zdroje klientského systému.
- Nutná instalace na každého klienta.
- Monitoruje podezřelé aktivity pouze na klientovi.
- Časová prodleva při nahlášení napadení.

1.6 Ostatní systémy

Mezi ostatní systémy lze do jisté míry zařadit i systém Honeypot (návnada), který není primárně IDS systémem. Lze ho využít jako prostředek, který upoutá pozornost útočníka.

1.6.1 Honeypot

Spitzner definuje Honeypot jako [32]: "*Bezpečnostní prostředek, jehož hodnota spočívá v tom, že je zkoumán, napaden nebo ohrožen.*[7]". NÚKIB uvádí, že Honeypot je [33]: "*Obecný název pro systém, který se používá k nalákání útočníka a k jeho přesvědčení, aby strávil čas zpracováním informací, které se zdají být velmi hodnotné, ale ve skutečnosti jsou uměle vyrobené a pro oprávněného uživatele bezcenné.*".

Honeypot nemá jiný v síti úkol než, přilákat zájem útočníka na tento systém, kdy se tváří jako zcela legitimní systém nebo služba. A následně sledovat a monitorovat jeho aktivitu a maximalizovat dobu po kterou je útočník aktivní na tomto systému, tak aby bylo možné ho sledovat [34].

Tyto systémy mají následující charakteristiky [34]:

- Oklamání – skrytí skutečné funkce (systém se jeví jako již existující systém nebo služba).
- Odhalení – umístění uvnitř sítě v blízkosti systémů, které napodobují.
- Interaktivita – nízká a střední aktivita, aby nedošlo k odhalení skutečné funkce systému.

[7] A security resource whose value lies in being probed, attacked, or compromised. [32]

- Monitorování – zaznamenávání veškeré činnosti, propojení se systémem SIEM (Security Event and Information Management – „*Systém, jehož úkolem je sběr, analýza a korelace dat – událostí v síti.*“ [33]).

1.6.2 EDR systémy

EDR (Endpoint Detection Response) systémy jsou agenti instalovaní na koncových stanicích, jejichž úkolem je sběr dat – telemetrie. “*Telemetrie jsou nezpracovaná data generovaná senzorovou komponentou nebo samotným hostitelem a obránci je mohou analyzovat a zjistit, zda došlo ke škodlivé aktivitě.*[8]“ [35]

Úkolem těchto systémů je detekce podezřelých aktivit, na základě vyhodnocení telemetrických dat. Příkladem může být nalezení podezřelého souboru, kterého kontrolní součet (hash) odpovídá kontrolnímu součtu známého škodlivého software např. malware.

Agent je schopný tuto aktivitu zaznamenat a zaslat informaci dále do centrálního systému EDR pro další zpracování. Další možností, kterou může agent vykonat je zablokování prováděného programu, tím, že programu vrátí chybné návratové hodnoty např. adresy v operační paměti, a dojde k oklamání škodlivého software. Na základě toho dojde škodlivý software k závěru, že došlo k úspěšnému provedení jeho úlohy [35].

[8] Telemetry is the raw data generated by a sensor component or the host itself, and defenders can analyze it to determine whether malicious activity has occurred. [35]

2 NÁVRH DETEKCE PRAVIDEL

Detekční systém by měl být schopen zachytit nestandardní události, které probíhají na Linuxovém serveru. Framework Mitre ATT&CK®[9] definuje taktiky útočníků, které byly registrovány, jako metody útoků. HIDS systém by měl být schopen detekovat tyto jednotlivé techniky jako součást detekčního mechanismu na operačním systému. Dále v této kapitole budou rozebrány pravidla podle vybraných metod protivníků, které lze aplikovat na Linuxové systémy.

2.1 Souborový systém

Na Linuxových a Unixových systémech jsou standardně uloženy konfigurační soubory (definováno v FHS (Filesystem Hierarchy Structure)[10]). Z tohoto důvodu by HIDS systém měl být schopen sledovat následující změny:

- Konfigurační soubory v adresáři /etc.
- Atributů souborů, včetně rozšířených ACL (Access Control List)[11] práv.
- Vlastníků souborů.

2.2 PAM

PAM (Pluggable Authentication Modules Library)[12] je vrstva určená k procesu ověřování mezi aplikací a uživatelem. Obsahuje knihovní moduly, které tuto činnost zajišťují, jedná se například o moduly pro ověření přístupu do systému lokálního – login, démonu SSHD určeného pro zabezpečené přihlášení po sítí atd [36;37].

Systém by měl být schopen kontroly následujících souborů:

- /etc/pam.d,
- /etc/pam.conf.

[9] https://attack.mitre.org
[10] https://www.pathname.com/fhs/
[11] https://linux.die.net/man/5/acl
[12] https://linux.die.net/man/3/pam

2.3 Lokální účty

Pokusy o přidání, odebrání, modifikace uživatelů nebo skupin v systému. Tyto změny lze provádět pomocí příkazů pro přidání uživatelů adduser, useradd, odstranění uživatelů userdel nebo změny příkazem usermod. Dále přidání skupin groupadd, odstranění skupin groupdel nebo modifikace promocí příkazu groupmod a změna nastavení hesla promocí příkazu passwd.

Může také dojít k ruční modifikaci účtů bez použití výše uvedených příkazů. Při této činnosti dochází ke změnám následujících souborů:

- /etc/passwd,
- /etc/shadow,
- /etc/group.

2.3.1 Pokusy o přihlášení – uživatelské účty

Systém by měl být schopen registrovat neúspěšné pokusy o přihlášení běžných uživatelských účtů a slovníkové útoky („*Útok na systém, v rámci kterého jsou využívány seznamy často používaných hesel.*“ [33]).

Může se jednat o následující pokusy:

- lokální přihlášení (login, su),
- sudo[13],
- vzdálené přihlášení po sítí.

2.3.2 Pokusy o přihlášení – privilegovaný účet

Systém by měl být schopen registrovat neúspěšné pokusy o přihlášení na privilegovaného uživatele, kterým je na linuxových a unixových systémech uživatelský účet *root*.

Může se jednat o následující pokusy:

- lokální přihlášení (login, su),
- sudo,
- vzdálené přihlášení po sítí.

[13] https://wiki.debian.org/sudo/

Dále monitorovat i úspěšné přihlášení jako privilegovaný uživatel, které mohou probíhat v nestandardních časech nebo během svátků, víkendů atd.

2.3.3 Neoprávněný přístup

Monitorování pokusů o zvýšení oprávnění jak vertikální (změna neprivilegovaného na neprivilegovaného uživatele), tak horizontální (změna z neprivilegovaného na privilegovaného uživatele).

2.4 Lokální služby

Systém by měl být schopen detekovat spuštění nestandardní služby. Nestandartní službou je myšleno vytvoření služby v rámci správce služeb systemd[14] nebo původního systému init.d.

2.5 Podezřelé aktivity

Manipulace s logy např. démona auditd.

2.6 Procesy

Monitorování nestandardních procesů a služeb (démonů).

2.7 Síťové servery

2.7.1 Skenování portů

Detekce skenování portů.

2.7.2 SSH

Neúspěšné pokusy o vzdálené přihlášení prostřednictvím protokolu SSH (secure shell).

[14] https://systemd.io

3 VÝBĚR DETEKČNÍHO SYSTÉMU

Kapitola se zabývá výběrem vhodného open source systému, který by byl vhodný pro zabezpečení klientského systému.

Tento program by měl splňovat následující základní požadavky:

- Integrita souborů.
- Detekce malware a rootkitů („*Programy umožňující maskovat přítomnost zákeřného software v počítači.*“ [33]).
- Monitorování systémových volání.
- Analýza logů.

3.1 Systémy zajišťující integritu souborů

Mezi další HIDS systémy se řadí, ty, které zajišťují integritu souborů. Vzhledem k tomu, že filozofie unixových systému je „*On a UNIX system, everything is a file; if something is not a file, it is a process.*“ [38] neboli „*V systému UNIX je vše soubor; pokud něco není soubor, je to proces.*“ [38]. Tyto systémy provádějí kontrolu, zda nedošlo ke změně souboru s porovnáním proti jejich databázi (Kapitola 1.1.6).

3.1.1 Samhaim

Program Samhaim zajišťuje integritu souborů a provádí detekci na základě systémových logů umístěných na klientském systému. Je schopen pracovat nezávisle jako samostatný program nebo centrálně, kde sbírá a vyhodnocuje data od ostatních instancí programu nakonfigurovaných jako klienti. Program dále monitoruje aktivitu na lokálních síťových portech a porovnává je se seznamem povolených portů. Další vlastností je detekce integrity linuxového kernelu na rootkity, analyzovat soubory logů, je schopen monitorovat skryté procesy, které nejsou viditelné pomocí linuxového příkazu ps[15].

Pokud by došlo ke kompromitaci systému tak Samhaim podporuje skrytý mód (stealth mode), aby ho nebyl schopen útočník sabotovat.

[15] https://man7.org/linux/man-pages/man1/ps.1.html

Prostřednictvím externího programu Beltane[16] lze doinstalovat webové rozhraní pro tento systém.

Samhaim podporuje operační systémy Linux, BSD systémy, Solaris, AIX, HP-UX, MacOS a Windows [39;40].

3.1.2 AIDE

AIDE (Advanced Intrusion Detection Environment) je dalším systémem, který sleduje integritu souborů a adresářové struktury pomocí předefinovaných pravidel. U souborů monitoruje jejich vlastnosti, jakými jsou typ souboru, oprávnění, inode, uid, gid, název odkazu, velikost, počet bloků, počet odkazů, mtime, ctime a atime., vlastníka a systému SELinux[17].AIDE je podporovaný na operačních systémech Linux, BSD systémech, MacOS a dalších [41].

3.2 Systémy monitorující logy

HIDS systémy monitorující logy kontrolují logy a na základě definovaných pravidel zasílají upozornění správci systému nebo bezpečnostnímu administrátorovi.

3.2.1 Sagan

Sagan není klasickým HIDS systémem, ale provádí analýzu a korelaci dat z logů v reálném čase. Používá podobná pravidla a strukturu pro zachování kompatibility a možnosti korelaci dat s jinými IDS systémy (Snort[18], Surikata[19]). Sagan je schopen exportovat data do jiných formátů, detekovat IP adresy pomocí databáze GeoIP[20] a je schopný spouštět skripty, které reagují na vzniklé bezpečnostní události [42;43].

3.3 Bezpečnostní platformy

Bezpečnostní platformy kombinují několik funkcí dohromady – kontrola integrity souborů, analýzu logů a korelace, zasílání upozornění a další. Mezi nejznámější dva projekty, které

[16] https://www.la-samhna.de/beltane/index.html
[17] https://github.blog/2023-07-05-introduction-to-selinux/
[18] https://www.snort.org
[19] https://suricata.io
[20] https://dev.maxmind.com/geoip/geolite2-free-geolocation-data

nabízejí open source verzi tak i placenou verzi se řadí v první řade program OSSEC a dále Wazuh.

3.3.1 OSSEC

„*OSSEC má výkonný korelační a analytický engine, který integruje analýzu protokolů, monitorování integrity souborů, monitorování registru Windows, centralizované vynucování zásad, detekci rootkitů, upozorňování v reálném čase a aktivní reakci.*[21]“ [44]

Kromě výše zmíněných funkcí platforma monitoruje logy v reálném čase, sleduje spuštěné procesy a detekuje škodlivé aplikace – malware, provádí inventarizaci software a hardware a aktivně reaguje na útoky a systémové změny. Dále také poskytuje možnost auditu systému a zajištění shody například se standardy CIS (Center for Internet Security)[22].

Obrázek 8. OSSEC architektura [45]

Platforma OSSEC pracuje v režimu klient server (Obrázek 8), kde v databázi na centrálním serveru jsou uloženy informace o integritě souborů, protokolů, událostí a záznamy o auditu. Centrální server dále obsahuje hlavní konfiguraci, pravidla detekce a dekodéry.

[21] OSSEC has a powerful correlation and analysis engine, integrating log analysis, file integrity monitoring, Windows registry monitoring, centralized policy enforcement, rootkit detection, real-time alerting and active response. [44]

[22] https://www.cisecurity.org/cis-benchmarks

OSSEC agenti jsou programy nainstalované na klientských systémech a podporují následující operační systémy – Linux, OpenBSD, MacOS, Windows, Solaris, HP-UX a VMware.

OSSEC je schopný monitorovat i systémy na které nelze nainstalovat agenty. Takovými zařízeními jsou firewally, směrovače a síťové přepínače bez agenta za pomoci protokolu SSH [45].

3.3.2 Wazuh

Wazuh vznikl jako odnož (fork) projektu OSSEC v roce 2015, z důvodů jeho nevýrazného pokroku ve vývoji. Na webu projektu uvádějí [46]: „*Proto se tým Wazuh v roce 2015 rozhodl projekt rozdělit. Výsledkem je mnohem komplexnější, snadno použitelné, spolehlivé a škálovatelné řešení. Fork se setkal s vřelým přijetím v open source komunitě a rychle se stal široce používaným řešením v podnikovém prostředí*[23]. “.

Tato platforma spojuje několik funkcí dohromady (sledování integrity souborů, detekce malware, rootkity, posuzování konfiguraci zabezpečení, detekovat zranitelnosti, analyzovat soubory s logy, monitorovat příkazy, systémová volání).

Wazuh funguje stejně jako OSSEC v konfiguraci klient – server. Agent (klient) se instaluje na koncovou stanici. Je také podporována konfigurace bez agenta pomocí SSH protokolu. Na těchto systémech je schopen monitorovat soubory, adresáře, konfigurace nebo vzdáleně spustit příkazy a na základě výsledku spustit výstrahu. Jedná se o síťová zařízení typu firewall, směrovač a router [46;47].

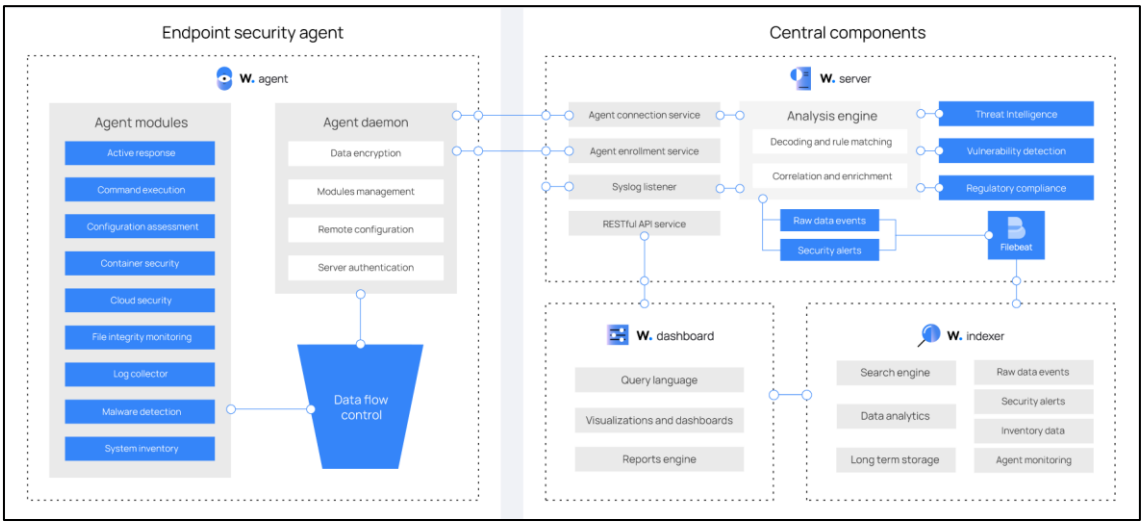
Agenta je možné instalovat na laptop, desktop, server, cloudový server nebo virtuální server. Klient podporuje systémy Linux, Windows, macOS, Solaris, AIX nebo HP-UX. Agent komunikuje se serverovou částí pomocí protokolu AES (Advanced Encryption Standard)[24].

---

[23] This is why, back in 2015, the Wazuh team decided to fork the project. The result is a much more comprehensive, easy-to-use, reliable, and scalable solution. The fork has had great adoption among the open-source community, quickly becoming a broadly used solution in enterprise environments. [46]

[24] https://nvlpubs.nist.gov/nistpubs/FIPS/NIST.FIPS.197-upd1.pdf

Systém podporuje nasazování prostřednictvím nástrojů konfiguračního managementu, kterými jsou Ansible, Chef, Puppet, kontejnerová řešení Docker, cloudová řešení –AWS, Azure a GCP [47;48].



Obrázek 9. Wazuh architektura [47]

Serverová část má tři komponenty (Obrázek 9) [47]:

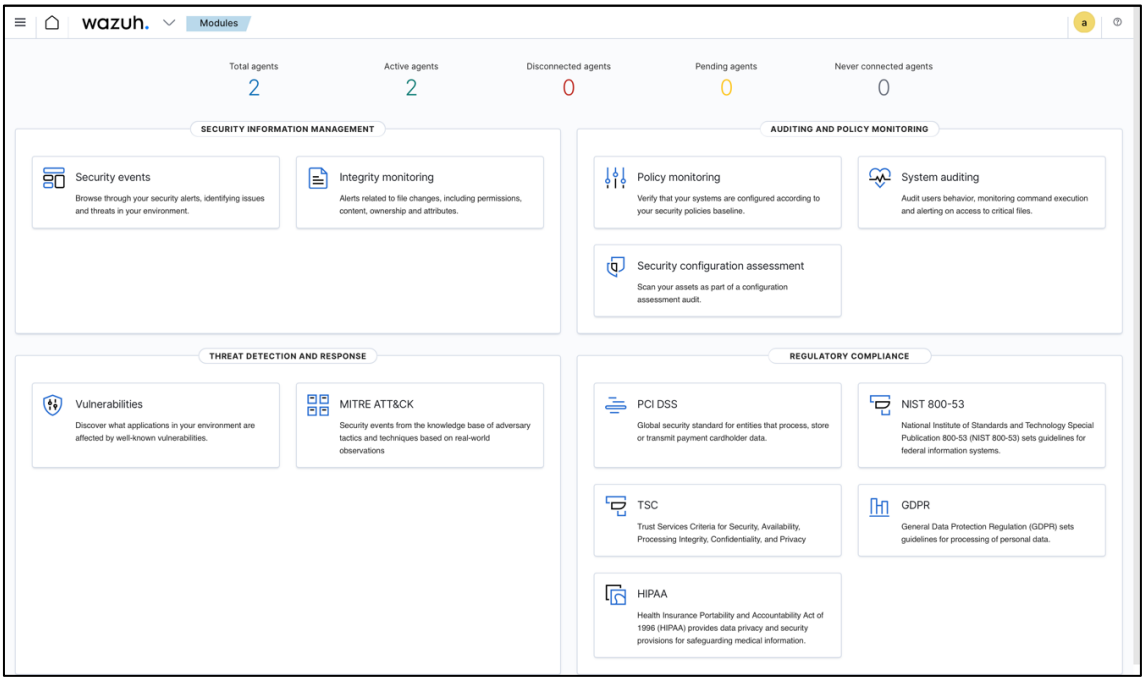
- server,
- indexer,
- dashboard.

Wazuh server provádí analýzu dat přijatých od agentů nainstalovaných na klientských systémech, server zpracovává přijatá data a vyhledává v nich indikátory kompromitace (IOC) [47].

„*Wazuh indexer je vysoce škálovatelný fulltextový vyhledávací a analytický engine*[25].“ [47], který obsahuje indexy a výstrahy z komponenty Wazuh server.

Wazuh dashboard je uživatelské rozhraní poskytující vizualizaci a analýzu získaných dat od agentů [47].

[25] The Wazuh indexer is a highly scalable, full-text search and analytics engine. [47]



Obrázek 10. Wazuh dashboard

Modul FIM (File Integrity Monitoring) umí kontrolovat integritu souborů. FIM monitoruje soubory a adresáře, kontrolní součty, atributy a v případě změny zasílá upozornění o provedených změnách. Dále je schopen detekovat škodlivý software – malware pomocí kontroly signatur. Modul Rootcheck umí detekovat rootkity. Modul SCA (Security Configuration Assessment) posuzuje, zda konfigurace daného systému odpovídá nastaveným pravidlům bezpečnostní politiky. Vulnerability detector je modul, který umožňuje detekovat možné zranitelnosti systémů a aplikací na koncových stanicích. Systém dokáže monitorovat různé příkazy (např. místo na disku, zatížení systému, procesy atd.). Monitorování systémových volání poskytuje možnost detekce podezřelého chování na koncových systémech [48;49].

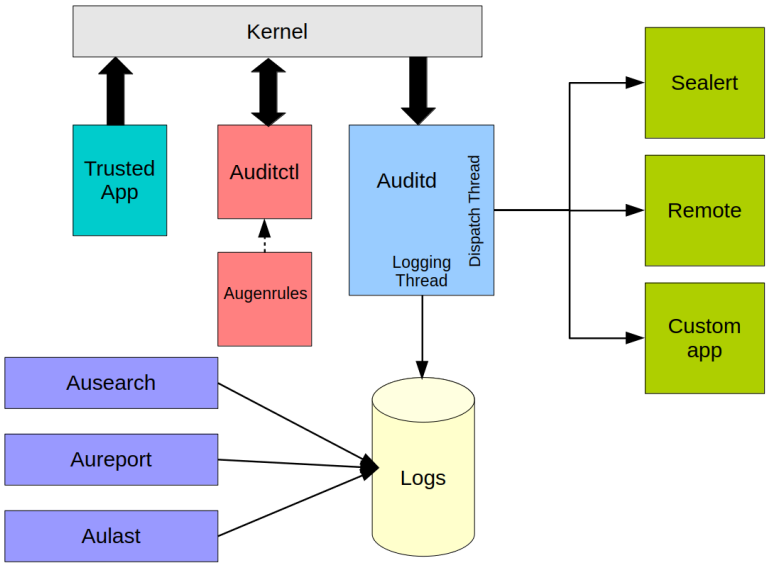
3.4 Ostatní

Následujícím systémem, který je součástí každé linuxové distribuce linuxový auditní systém, který sám o sobě neprovádí aktivní detekci narušení systému, ale shromažďuje a zapisuje záznamy o událostech do logovacího systému.

3.4.1 Linux auditní systém

Úkolem Linuxového auditního systému je sledování informací důležitých pro zabezpečení systému. Auditní systém úzce spolupracuje s linuxovým jádrem od verze 2.6 a sleduje volání

systému. Systém sám generuje logy a nepotřebuje proto externí nástroj jakým je například syslog[26] nebo rsyslog [27]server [50;51].

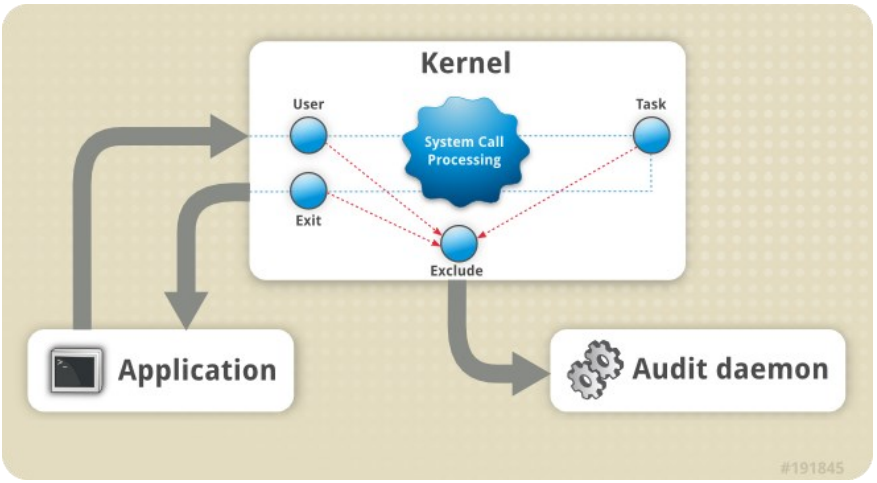


Obrázek 11. Architektura auditního systému [52]

Architektura auditního systému je zobrazena na Obrázku 11, kde je zobrazen vztah mezi jednotlivými částmi, které pracují jak v uživatelském režimu, tak spolupracují s jádrem systému. V uživatelském režimu se jedná o programy o programy ausearch, aureport, aulast a auditctl. Program ausearch, který slouží k vyhledávání událostí v logovacích souborech. Program aureport vytváří reporty, program aulast zobrazuje seznam přihlášených a odhlášených uživatelů, auserach vyhledává uživatelská jména a terminály v uložených záznamech [53].

„*Démon Audit v uživatelském prostoru shromažďuje informace z jádra a vytváří záznamy v souboru logu.*[28]“ [51]

[26] https://linux.die.net/man/8/syslogd
[27] https://www.rsyslog.com
[28] The user-space Audit daemon collects the information from the kernel and creates entries in a log file. [51]



Obrázek 12. Architektura auditního systému [50]

„V režimu jádra běží komponenta, která přijímá systémová volání od aplikací v uživatelském prostoru a filtruje je přes jeden z následujících filtrů: user, task, fstype nebo exit. Poté, co systémové volání projde filtrem exclude, je odesláno přes jeden z výše uvedených filtrů, který jej na základě konfigurace pravidla Audit odešle démonu Audit k dalšímu zpracování.[29]“ [50]

Dále je auditní systém schopen sledovat přístup k souborům, sledovat systémová volání, monitorovat uživatelské příkazy, záznam bezpečnostních událostí, vyhledávání událostí, přehledy událostí, monitorovat síťový přístup ve spolupráci s iptables[30], nftables[31] nebo ebtables[32] [51].

3.5 Systémy HIDS s funkcí IPS

Pro úplnost je nutné zmínit dva HIDS systémy, které v sobě integrují i IPS (Intrusion Prevention Systems) systémy prevence narušení. IPS systémy v porovnání s IDS systémy aktivně reagují na bezpečnostní události.

[29] The kernel component receives system calls from user-space applications and filters them through one of the following filters: user, task, fstype, or exit. After a system call passes the exclude filter, it is sent through one of the aforementioned filters, which, based on the Audit rule configuration, sends it to the Audit daemon for further processing. [50]

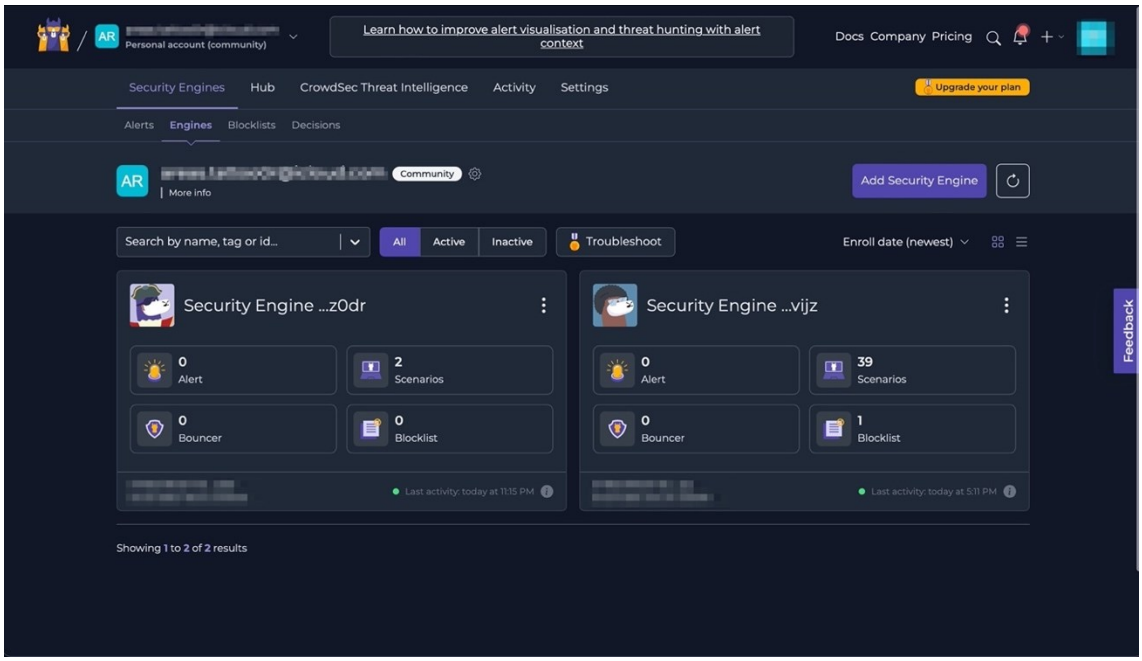
[30] https://linux.die.net/man/8/iptables
[31] https://www.nftables.org
[32] https://linux.die.net/man/8/ebtables

3.5.1 Fail2ban

Fail2ban zajišťuje zabezpečení síťového přístupu ke koncové stanici. Program monitoruje soubory s logy umístěnými v adresáři /var/log např. auth.log, které obsahují záznamy o přihlášení do systému. Fail2ban zablokuje IP adresu vzdáleného systému na základě vysokého počtu zaznamenaných neúspěšných pokusů o vzdálené přihlášení do systému [54].

3.5.2 Crowdsec

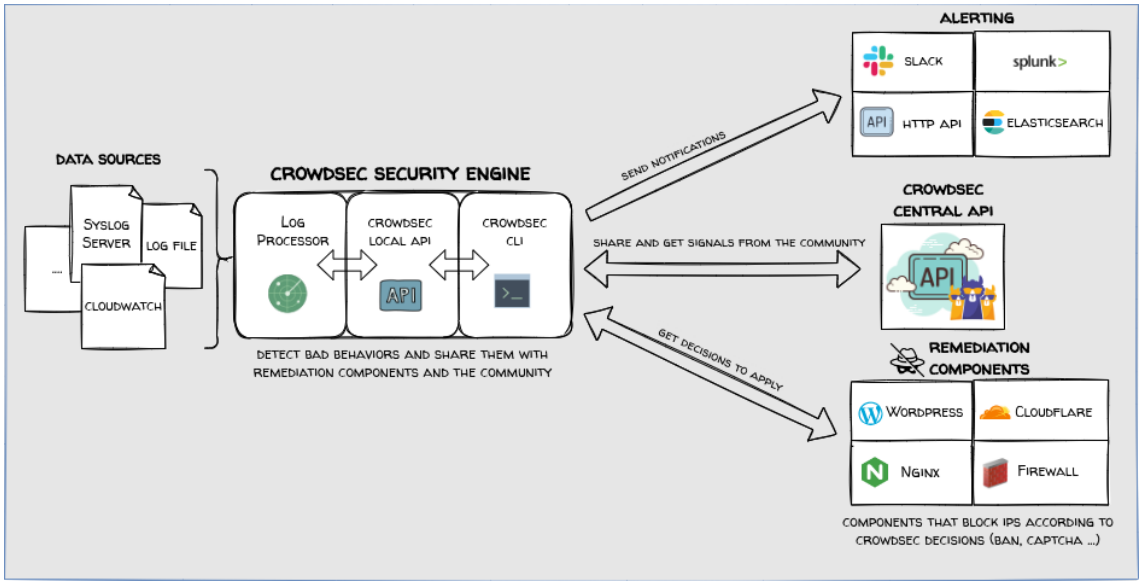
Crowdsec analyzuje síťový provoz a detekuje na podezřelé aktivity v reálném čase. Jedná se modernější verzi programu Fail2ban.



Obrázek 13. Crowsec dashboard

Součástí je programu je subsystém nazvaný „bouncer“, ve spolupráci s firewallem (iptables, nftables) blokuje škodlivé IP adresy. Tento subsystém podporuje instalaci dalších zásuvných modulů (plugins) nebo vlastních skriptů. Dále obsahuje kolekce s detekčními scénáři, které jsou vyvinuté firmou Crowdsec nebo komunitou okolo tohoto produktu. Program Crowdsec pracuje na bázi modelu klient – server.

Server je dostupný jako SAAS (Software As A Service) řešení a jehož komunitní verze je zdarma. [55;56]



Obrázek 14. Architektura programu Crowdsec [57]

3.6 Porovnání HIDS systémů

Na Obrázku 15 je zobrazeno porovnání vlastností jednotlivých systémů. Jak je zřejmé, nejvíce vlastností poskytuje program Wazuh, který je vhodným kandidátem pro nasazení HIDS systémů a splňuje také původní stanovené požadavky na HIDS systém (Kapitola 2). Druhým možným kandidátem, který neposkytuje tolik možností jako je Wazuh je projekt Samhaim.

	OSSEC	Wazuh	Samhaim	AIDE	Sagan	Fail2ban	Crowdsec	Auditd
Open Source	X	X	X	X	X	X	X	X
Podporovaný OS Debian/Ubuntu	X	X	X	X	X	X	X	X
Podporovaný OS RedHat/Klony	X	X	X	X	X	X	X	X
Agent	X	X						
Detekce rootkitu	X	X	X					
Detekce malware	X	X						
Inventarizace	X	X						
Blokování portů		X				X	X	
Analýza logů	X	X	X		X	X	X	
Integrita souboru	X	X	X	X				
Detekce zranitelnosti		X						
Monitorování portů		X	X					
Skrytý mód			X					
Detekce skrytých procesů		X	X					
Kontrola CIS		X						
Integrace s Mitre ATT&CK®		X						
Monitorování zranitelností		X						

Obrázek 15. Srovnání systémů [40;41;42;43;45;46;47;48;51;54;55;60]

Projekt OSSEC je podobný programu Wazuh, ale jak již bylo zmíněno, není tak často aktualizován. Ostatní projekty poskytují pouze omezené vlastnosti detekce. AIDE provádí pouze kontrolu integrity souborů. Sagan provádí pouze kontrolu logovacích souborů. Oba programy Fail2ban a Crowdsec sledují logy a na základě neúspěšných pokusů o přihlášení do systému blokují přístup.

Alternativou by mohl být použit démon Auditd pro monitoring podezřelých aktivit na linuxových systémech, ale vyhodnocovaní událostí by probíhalo na externím SIEM systému. V tomto případě je nutno zajistit bezpečnou komunikaci protokolem TLS (Transport Layer Security) mezi klientem odesílajícím auditní logy a serverem. Dále je třeba zajistit spolehlivé odeslání logů v případě, že je centrální logovací/SIEM systém nedostupný. Toho lze docílit prostřednictvím programu rsyslog[33] a protokolu RELP (Reliable Event Logging Protocol)[34] nebo s ukládáním souborů s logy do fronty na lokálním systému. Auditd má v porovnání se systémem Wazuh pouze omezené vlastnosti, a je nutné ho použít s jiným externím systémem.

[33] https://www.rsyslog.com
[34] https://www.rsyslog.com/doc/configuration/modules/omrelp.html

II. PRAKTICKÁ ČÁST

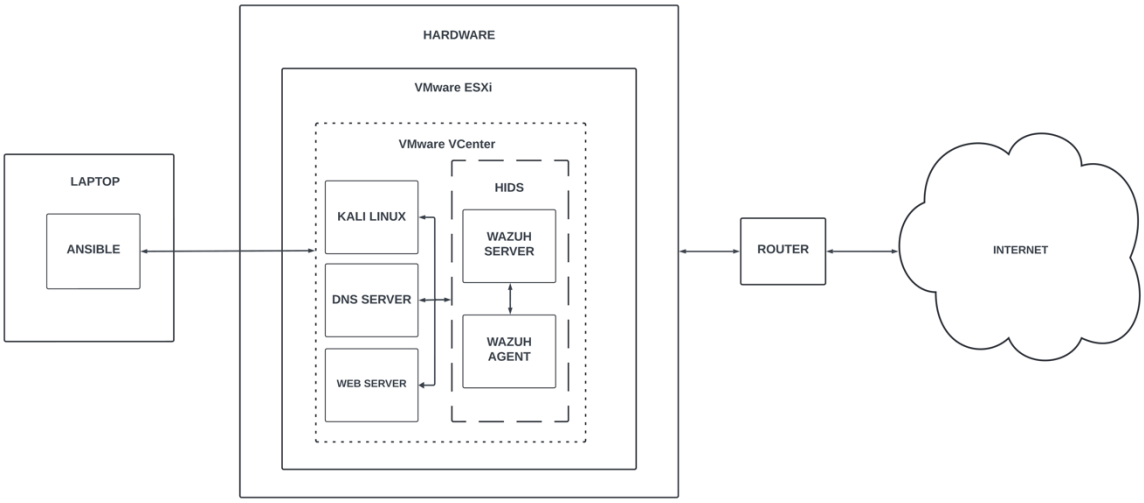
4 ARCHITEKTURA

Kapitola se zabývá architekturu testovacího prostředí a jednotlivými komponenty, které budou podrobněji popsány v dalších kapitolách.

4.1 Architektura testovacího prostředí

Architektura testovacího prostředí (Obrázek 16) obsahuje následující komponenty:

- Fyzický server: DELL PowerEdge T620, 2 CPU (2 x 6 jader), 96 GB RAM, 4TB HDD (RAID 5)
- Hypervisor: VMware ESXi 8.X
- Managment: vCenter 8.X
- Virtuální server Wazuh-agent, 2 vCPU, 8 GB RAM, 20 GB HDD
- Virtuální server Wazuh-server, 4 vCPU, 32 GB RAM, 50 GB HDD
- Virtuální server Kali Linux, 4 vCPU, 16 GB RAM, 20 GB HDD



Obrázek 16. Architektura testovacího prostředí

4.1.1 Hardware

Pro účely této práce bylo zvolen fyzický server DELL PowerEdge T620 (Obrázek 17), který se již nevyrábí. Pro účely této práce je tento server je tento server po hardwarové stránce stále dostačující.

Obrázek 17. Server DELL PowerEdge T620 [58]

4.1.2 Virtualizační prostředí

Jako virtualizační software byl zvolen produkt od firmy VMware, VMware ESXi[35] hypervisor nainstalovaný na fyzický hardware. Pro správu virtuálních serverů byl použit software VMware vCenter Server[36]. Systémy byl licencovány prostřednictvím programu VMUG (VMware User Group), který za poplatek poskytuje roční, dvouleté nebo tříleté předplatné k produktům firmy VMware pro nekomerční účely [59].

Výhodou virtualizační platformy je možnost vytváření obrazů operačního systému (snapshots) z existujícího virtuálního serveru. Tato funkce byla dále využita při testování.

4.1.3 Řídící stanice (laptop)

Pro účely řídí stanice byl použit laptop s nainstalovaným konfiguračním nástrojem Ansible. Z laptopu se spouštěly Ansible scénáře a sloužil také ke komunikaci s platformou VMware ESXi a vCenter (Obrázek 16).

Ansible

Ansible je volně šiřitelný nástroj od firmy Red Hat, který zjednodušuje konfigurační management a nasazování aplikací v rámci IT infrastruktury. Bylo vytvořeno několik Ansible scénářů pro zjednodušení instalačních a konfiguračních kroků [60].

[35] https://www.vmware.com/products/esxi-and-esx.html
[36] https://www.vmware.com/products/vcenter.html

4.1.4 Kali Linux

Kali Linux je volně šiřitelná distribuce zaměřující se na bezpečnostní audity, penetrační testování a forenzní analýzu operačních systémů [61].

V této práci byla tato distribuce použita pro účely penetračního testování (Kapitola 7).

4.1.5 Web server

Jako úložiště instalačních předpisů (kickstart[37]) a softwarových balíčků pro operační systém Linux byl použit dedikovaný server s nainstalovaným HTTP (Hypertext Transfer Protocol) web serverem Apache[38]. Web server poskytuje také RPM (RPM Packing Manager) balíčky a ostatní soubory potřebné pro automatickou instalaci operačního systému Linux a zároveň se používá jako lokální YUM/DNF server[39]. Server nebyl předmětem bezpečnostního testování.

4.1.6 DNS server

DNS (Domain Name System) server byl použit pro účely jmenných služeb v testovacím prostředí. Server nebyl předmětem bezpečnostního testování.

4.1.7 HIDS

HIDS (Host-based Intrusion Detection System) systém (dále popsán v Kapitole 6) se skládá ze dvou komponent, serverové části (Wazuh server) a části klientské (Wazuh agent). Oba systémy komunikují s okolními počítači. Detailní informace o zabezpečení klientského systému je popsána v Kapitole 5 a jeho konfigurace následně v Kapitole 6, kde jsou také informace o nastavení serverové části.

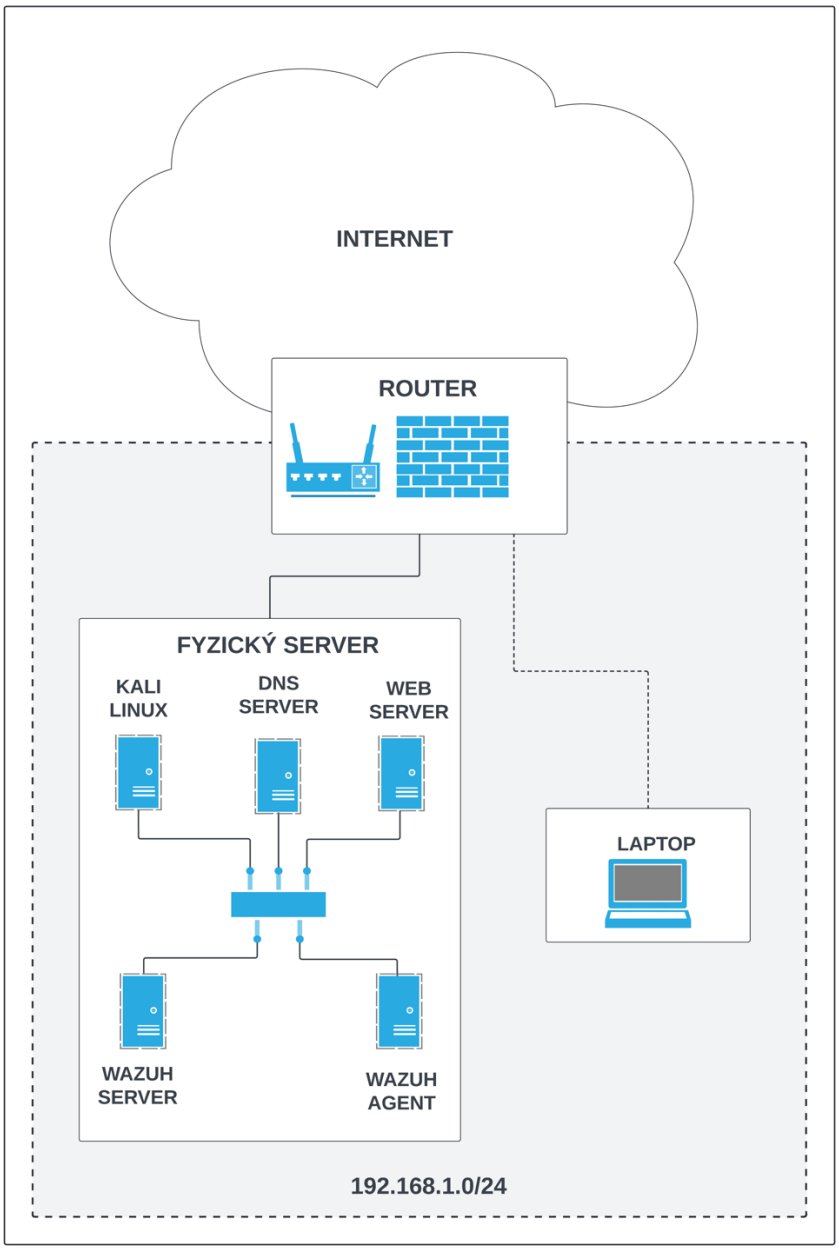
Klientská část HIDS systému s Wazuh agentem byla předmětem bezpečnostního testování.

4.2 Architektura síťového prostředí

Síťová topologie je zobrazena na Obrázku 18, kde jednotlivé virtuální servery používají statické IP adresy z privátního rozsahu podsítě 192.168.1.0/24. Laptopu byla přidělena IP

[37]https://access.redhat.com/documentation/en-us/red_hat_enterprise_linux/9/html-single/performing_an_advanced_rhel_9_installation/index
[38] https://httpd.apache.org
[39] http://yum.baseurl.org

adresa z DHCP serveru provozovaném na routeru, který obsluhuje část adresního rozsahu stejné podsítě jako v případě serverů.



Obrázek 18. Architektura síťového prostředí

5 OPERAČNÍ SYSTÉM LINUX

Tato část práce se zaměřuje na popis instalace operačního systému, konfigurace a následně jeho zabezpečeni.

Jako linuxová distribuce vhodná pro tuto práci byla zvolena distribuce Rocky Linux[40] 9, která vychází z komerční linuxové distribuce Red Hat Enterprise Linux[41] 9. Jedná se o volně šiřitelnou (open source) distribuci, šiřitelnou pod BSD licencí[42]. Aktualizace operačního systému a software jsou volně k dispozici z veřejně dostupných repozitářů [62].

5.1 Instalace a konfigurace

Instalace a konfigurace operačního systému byla provedena prostřednictvím instalačního přepisu (kickstart) (viz Příloha P I CD-ROM), který se používá pro automatickou instalaci systémů odvozených od distribuce Red Hat Enterprise Linux, jakými jsou např. Rocky Linux, Alma Linux[43].

5.1.1 Ansible

Virtuální server byl vytvořen pomocí Ansible scénáře (playbook) *vmware_deploy_vm.yml* na virtualizační platformě VMware. Tento scénář vytvořil virtuální server a spustil instalaci operačního. Na Obrázku 19 je zobrazen průběh této instalace.

[40] https://www.rockylinux.org
[41] https://www.redhat.com/en/technologies/linux-platforms/enterprise-linux
[42] https://opensource.org/license/BSD-3-clause
[43] https://www.almalinux.org

```
→ hids ansible-playbook -i inventories/dev/hosts playbooks/vmware_deploy_vm.yml

PLAY [localhost] ***************************************************************************************************************

TASK [Gathering Facts] *********************************************************************************************************
ok: [localhost]

TASK [Create VM] ***************************************************************************************************************
changed: [localhost]

TASK [Change machine boot order] ***********************************************************************************************
skipping: [localhost]

TASK [Send escape key stroke at boot prompt] ***********************************************************************************
changed: [localhost]

TASK [Send escape key stroke at boot prompt] ***********************************************************************************
changed: [localhost]

TASK [Send kickstart file string BIOS] *****************************************************************************************
changed: [localhost]

TASK [Send enter key stroke to start the automated OS install] *****************************************************************
changed: [localhost]

TASK [Send kickstart file string BIOS] *****************************************************************************************
changed: [localhost]

TASK [Send enter key stroke to start the automated OS install] *****************************************************************
changed: [localhost]

TASK [Send enter key stroke to start the automated OS install] *****************************************************************
changed: [localhost]

TASK [Send enter key stroke to start the automated OS install] *****************************************************************
changed: [localhost]

PLAY RECAP *********************************************************************************************************************
localhost                  : ok=10   changed=9    unreachable=0    failed=0    skipped=1    rescued=0    ignored=0
```

Obrázek 19. Ansible vytvoření serveru a spuštění instalace

Ansible scénář nastavil zavaděč operačního systému GRUB[44] (GRand Unified Bootloader), tak aby si stáhnul instalační předpis kickstart (viz Příloha P I CD-ROM) z lokálního webového serveru (Obrázek 20). A následně spustil instalaci operačního systému.

Obrázek 20. Parametry zavaděče GRUB

[44] https://www.gnu.org/software/grub/grub-documentation.html

5.1.2 Zabezpečení

Základní zabezpečení serveru bylo provedeno již během vytvoření virtuálního serveru a instalace operačního systému pomocí již zmíněného instalačního předpisu (kickstart).

5.1.3 Hardware

Virtuální server byl vytvořen s firmware UEFI (Unified Extensible Firmware Interface) a podporou Secure boot, který je podporován také distribucí Rocky Linux (Obrázek 21). Secure boot zajišťuje vyšší zabezpečení proti rootkitům a jinému škodlivému kódu díky využívání technologie digitálního podpisu k ověření zdroje a jeho integrity. Škodlivým kódem může být například zavaděč systému (bootkit), secure boot poskytne ochranu proti jeho spuštění před zavedením operačního systému [63; 64].

```
[root@hids-sandbox01 ~]# keyctl show %:.platform
Keyring
 292672431 ---lswrv      0     0  keyring: .platform
 799398963 ---lswrv      0     0   \_ asymmetric: VMware, Inc.: 4ad8ba0472073d28127706ddc6ccb9050441bbc7
  27364859 ---lswrv      0     0   \_ asymmetric: Rocky Enterprise Software Foundation: Rocky Linux Secure Boot
 CA: 4c2c6bd7d64ee81581cab8e986661f65e2166fc4
 837655419 ---lswrv      0     0   \_ asymmetric: Microsoft Windows Production PCA 2011: a92902398e16c49778cd90
f9ae17c55af53
 743504225 ---lswrv      0     0   \_ asymmetric: Microsoft Corporation UEFI CA 2011: 13adbf4309bd82709c8cd54f3
22988a1bd4
 117131283 ---lswrv      0     0   \_ asymmetric: VMware, Inc.: VMware Secure Boot Signing: 04597f3e1ffb240bba0
5d5eb05f3e15f6d7
```

Obrázek 21. Zabezpečení Secure boot

5.1.4 GRUB

Během instalace operačního systému byl také zabezpečen přístup pomocí hesla ke změnám parametrů zavaděče GRUB (Obrázek 22).

Změny parametrů v zavaděči GRUB lze po tomto nastavení měnit pouze po zadání uživatelského jména, které je vždy *root* a jeho hesla.

```
bootloader --append="crashkernel=1G-4G:192M,4G-64G:256M,64G-:512M" --location=mbr --boot-drive=sda
--iscrypted --password=grub.pbkdf2.sha512.10000.
```

Obrázek 22. Kickstart zabezpečení boot manageru

5.1.5 Operační systém

Distribuce na bázi operačního systému Red Hat Enterprise Server 9 poskytují možnost nastavit zabezpečení serveru již během instalace prostřednictví bezpečnostních SCAP (Security Content Automation Protocol) profilů (Obrázek 23).

NIST (National Institute of Standards and Technology) institut definuje SCAP jako [65]: „*Souhrn interoperabilních specifikací odvozených z nápadů komunity. Účast komunity je pro SCAP velkou předností, protože komunita zajišťuje, aby se do funkcí SCAP promítla co nejširší škála případů použití.*[45]".

SCAP profily jsou následující [66]:

- CIS Red Hat Enterprise Linux 9 Benchmark for Level 2 – Server.
- CIS Red Hat Enterprise Linux 9 Benchmark for Level 1 – Server.
- CIS Red Hat Enterprise Linux 9 Benchmark for Level 1 – Workstation.
- CIS Red Hat Enterprise Linux 9 Benchmark for Level 2 – Workstation
- French National Agency for the Security of Information Systems (ANSSI) BP-028 Enhanced Level.
- French National Agency for the Security of Information Systems (ANSSI) BP-028 High Level.
- French National Agency for the Security of Information Systems (ANSSI) BP-028 Intermediary Level.
- French National Agency for the Security of Information Systems (ANSSI) BP-028 Minimal Level.

Pro tento server byl zvolen profil *CIS Red Hat Enterprise Linux 9 Benchmark for Level 1 – Server*. Detailní dokument o těchto testech je k dispozici po registraci na oficiálních webových stránkách organizace CIS [67]. CIS je nezisková organizace odpovědná za CIS Controls® a CIS Benchmarks™, metodiku a postupy pro zabezpečení informačních systémů [68].

```
%addon com_redhat_oscap
    content-type = scap-security-guide
    datastream-id = scap_org.open-scap_datastream_from_xccdf_ssg-rhel9-xccdf.xml
    xccdf-id = scap_org.open-scap_cref_ssg-rhel9-xccdf.xml
    profile = xccdf_org.ssgproject.content_profile_ccn_intermediate
%end
```

Obrázek 23. Kickstart nastavení bezpečnostního profilu

[45] A synthesis of interoperable specifications derived from community ideas. Community participation is a great strength for SCAP, because the security automation community ensures the broadest possible range of use cases is reflected in SCAP functionality. [65]

Další informace jsou k dispozici bez registrace na stránkách projektu CIS OpenSCAP[46],kde jsou popsány detaily nastavení zabezpečení podle operačních systémů a jejich profilů [69].

5.1.6 Softwarové balíčky

Systém byl nainstalován se základní skupinou *@Base* instalačních balíčků (*%packages*) a balíček *chrony* pro synchronizaci času (Obrázek 24).

```
%packages
@Base
chrony
```

Obrázek 24. Kickstart konfigurace softwarových balíčků

5.1.7 Účty

Z bezpečnostních důvodu bylo zakázáno přihlašování na účet systémového administrátora – účtu *root* bylo zakázáno. Pro účely vzdáleného přihlašování a testování vytvořen lokální účet *testuser* (Obrázek 25). Při prvním přihlášení do systému je vyžadována změna původního hesla, které bylo nastaveno během instalace.

S lokálním účtem uživatelem *testuser* se lze pomocí příkazu *sudo* přihlásit na uživatele účtu *root* a přihlášená vyžaduje zadání hesla uživatele *testuser*.

```
rootpw --iscrypted --lock [redacted]
user --groups=wheel --name=testuser --password=[redacted] --iscrypted --gecos="testuser"
```

Obrázek 25. Kickstart konfigurace lokálních účtů

V běžné praxi, se pro účely autorizace a autentizace používají adresářové servery typu LDAP, kterými je například OpenLDAP[47] nebo Microsoft Active Directory[48].

[46] https://openscap.org
[47] https://www.openldap.org
[48]https://learn.microsoft.com/cs-cz/windows-server/identity/ad-ds/get-started/virtual-dc/active-directory-domain-services-overview

5.1.8 Diskové oddíly

Pevný disk byl rozdělen na jednotlivé diskové oddíly, jak vyžaduje metodologie CIS, dále vzhledem k volbě UEFI jako firmware byly vytvořeny potřebné diskové oddíly (Obrázek 26).

```
part /boot/efi --fstype="efi" --ondisk=sda --size=600 --fsoptions="umask=0077,shortname=winnt"
part pv.1784 --fstype="lvmpv" --ondisk=sda --size=26612
part /boot --fstype="ext4" --ondisk=sda --size=1024
part /dev/shm --fstype="efi" --ondisk=sda --size=1907 --fsoptions="umask=0077,shortname=winnt"
volgroup rl_hids-sandbox --pesize=4096 pv.1784
logvol /var/log --fstype="ext4" --size=2861 --name=var_log --vgname=rl_hids-sandbox
logvol swap --fstype="swap" --size=3071 --name=swap --vgname=rl_hids-sandbox
logvol /home --fstype="ext4" --size=4768 --name=home --vgname=rl_hids-sandbox
logvol /var --fstype="ext4" --size=3072 --name=var --vgname=rl_hids-sandbox
logvol /var/log/audit --fstype="ext4" --size=1907 --name=var_log_audit --vgname=rl_hids-sandbox
logvol / --fstype="ext4" --size=6144 --name=root --vgname=rl_hids-sandbox
logvol /tmp --fstype="ext4" --size=2861 --name=tmp --vgname=rl_hids-sandbox
logvol /var/tmp --fstype="ext4" --size=1907 --name=var_tmp --vgname=rl_hids-sandbox
```

Obrázek 26. Kickstart nastavení diskových oddílů

5.1.9 Firewall

Součástí instalace je i základní nastavení paketového filtru služby firewallu, který povoluje pouze přístup na protokol SSH (Secure Shell) (Obrázek 27).

```
firewall --enabled --ssh
```

Obrázek 27. Kickstart nastavení firewallu

5.1.10 SELinux

Jako další krok k zabezpečení systému byl nastaven SELinux (Security Enhanced Linux) (Obrázek 28), jedná se o MAC (Mandatory Access Control), tedy systém povinné kontroly přístupu. Přístup k aplikacím, procesům a souborům v operačním systému se nastavuje pomocí definovaných pravidel.

Pokud aplikace nebo proces požádá o přístup k objektu – souboru, aplikaci nebo procesu, dojde ke kontrole jejího oprávnění přístupu pomocí přístupových vektorů. Na tomto základě, pokud je zde nalezeno patřičné oprávnění tak dojde k povolení k přístupu, v opačném případě odešle dotaz na bezpečnostní server. Server zkontroluje bezpečnostní kontext aplikace, souboru nebo procesu a udělí nebo zamítne přístup [70].

```
selinux --enforcing
```

Obrázek 28. Kickstart nastavení SELinux

5.1.11 Nastavení času

Dále byl nastavena synchronizace času prostřednictvím služby chronyd (Obrázek 29), která byla nakonfigurována jako klient využívající veřejné NTP (Network Time Protocol) servery z projektu NTP Pool Project [71;72].

```
%post

if [[ -e /etc/chrony.conf ]];then

cat >/etc/chrony.conf<< EOF_chrony
pool 0.cz.pool.ntp.org iburst
pool 1.cz.pool.ntp.org iburst
pool 2.cz.pool.ntp.org iburst
pool 3.cz.pool.ntp.org iburst

sourcedir /run/chrony-dhcp
driftfile /var/lib/chrony/drift
makestep 1.0 3
rtcsync
keyfile /etc/chrony.keys
ntsdumpdir /var/lib/chrony
leapsectz right/UTC
logdir /var/log/chrony
EOF_chrony

fi

if [[ $? -eq 0 ]];then
  systemctl enable chronyd
  systemctl start chronyd
fi

%end
```

Obrázek 29. Kickstart nastavení služby chronyd

5.2 Další nastavení serveru

Nový server je připravený k použití, ale při prvním přihlášení je vyžadována změna hesla (Obrázek 30) uživatele *testuser*.

```
→  ~ ssh testuser@192.168.1.50
The authenticity of host '192.168.1.50 (192.168.1.50)' can't be established.
ED25519 key fingerprint is SHA256:qXxvbZFDYYmNUgZh9SMj0Vk0VT/w2G8m77sIOj8YlCw.
This key is not known by any other names
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '192.168.1.50' (ED25519) to the list of known hosts.
Authorized uses only. All activity may be monitored and reported.
testuser@192.168.1.50's password:
You are required to change your password immediately (password expired).
You are required to change your password immediately (password expired).
Activate the web console with: systemctl enable --now cockpit.socket

WARNING: Your password has expired.
You must change your password now and login again!
Changing password for user testuser.
Current password:
New password:
Retype new password:
passwd: all authentication tokens updated successfully.
Connection to 192.168.1.50 closed.
```

Obrázek 30. SSH login změna hesla

5.3 Další zabezpečení serveru

Prvotní zabezpečení systému bylo již realizováno během instalace operačního systému. Dále bylo provedeno dodatečné nastavení zabezpečení serveru prostřednictvím Ansible role *RedHatOfficial.rhel9_cis* dostupnou z portálu Github [73].

6 HIDS

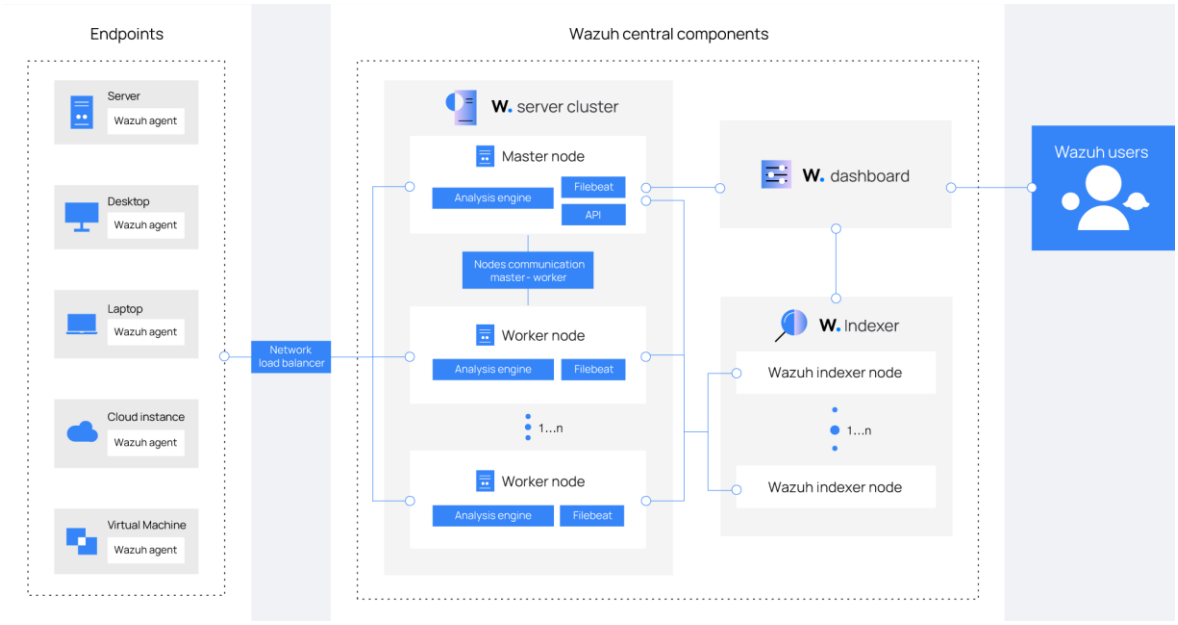
Kapitola se zabývá instalací a nastavením vybraného detekčního systému, jeho serverové části a instalací samotného agenta na koncové stanice. Dále konfigurací detekčních pravidel, která nejsou součástí originálního nastavení software, nastavení dalších služeb, které rozšiřují základní detekční mechanismy tohoto HIDS systému.

Jako HIDS systém byl vybrán software Wazuh (Kapitola 3.3.2), z důvodu jeho možností, kterými není pouze detekce pokusů o průnik, ale i možnosti skenování zranitelností, přehledné zpracování veškerých událostí ve formě dashboardu, propojení s metodologií Mitre ATT&CK®.

6.1 Wazuh serverová část

Projekt Wazuh poskytuje již vytvořený OVA (Open Virtual Appliance) instalační obraz, který obsahuje jeho jednotlivé subsystému (Indexer, Server, Dashboard). Tento instalační obraz se instaluje jako samostatný virtuální server.

Dále jde je zde možnost instalace systému jako clusterové řešení, kterým je myšlena instalace a konfigurace několika serverů jako tzv. pracovní uzly (worker nodes). Toto řešení je vhodné pro využití v rozsáhlé sítí s vysokým počtem klientských systémů. Distribuovaná konfigurace poskytuje lepší škálovatelnost a rozložení zátěže mezi jednotlivými subsystémy [74].



Obrázek 31. Komponenty Wazuh serveru [74]

V případě této práce byla vybrána jednodušší varianta, byl použit již předinstalovaný instalační OVA obraz systému (appliance), který je dostupný ze webových stránek projektu Wazuh.

6.1.1 Instalace

Serverová část byla nainstalovaná pomocí vytvořeného Ansible scénáře (playbook) - *wazuh_deploy_server.yml*. Scénář stáhnul instalační soubor OVA ze stránek projektu Wazuh, a byl uložen na lokální souborový systém počítače a následně byla provedena instalace OVA obrazu do virtualizačního prostředí a nakonfigurována IP adresa a nastaveno jméno serveru.

6.1.2 Konfigurace

Následně byla provedena konfigurace serverové části, konfigurační soubor *ossec.conf* je uložený v adresáři */var/ossec/etc/*.

Obsah celého konfiguračního souboru lze nalézt v Příloze P I CD-ROM. Jednotlivé části konfiguračního souboru jsou popsány dále v této kapitole.

Globální nastavení

Globální část konfigurace obsahuje základní nastavení emailu, který zasílá upozornění o bezpečnostních incidentech (Obrázek 32).

```
<global>
  <jsonout_output>yes</jsonout_output>
  <alerts_log>yes</alerts_log>
  <logall>no</logall>
  <logall_json>no</logall_json>
  <email_notification>yes</email_notification>
  <smtp_server>smtp.lab.local</smtp_server>
  <email_from>wazuh-server@smtp.lab.local</email_from>
  <email_to>wazuh@smtp.lab.local</email_to>
  <email_maxperhour>12</email_maxperhour>
  <email_log_source>alerts.log</email_log_source>
  <agents_disconnection_time>10m</agents_disconnection_time>
  <agents_disconnection_alert_time>0</agents_disconnection_alert_time>
</global>
```

Obrázek 32. Wazuh server globální konfigurace email

Další části je konfigurace portu pro komunikaci s agentem programu Wazuh (Obrázek 33).

```
<remote>
  <connection>secure</connection>
  <port>1514</port>
  <protocol>tcp</protocol>
  <queue_size>131072</queue_size>
</remote>
```

Obrázek 33. Wazuh server globální konfigurace porty

Detekční pravidla

Wazuh server je nainstalován s předdefinovanými detekčními pravidly (Obrázek 34), které lze povolovat nebo zakázat. Systém také podporuje vlastní pravidla detekce. V této práci se použila pouze výchozí pravidla serveru Wazuh.

```
<ruleset>
  <!-- Default ruleset -->
  <decoder_dir>ruleset/decoders</decoder_dir>
  <rule_dir>ruleset/rules</rule_dir>
  <rule_exclude>0215-policy_rules.xml</rule_exclude>
  <list>etc/lists/audit-keys</list>
  <list>etc/lists/amazon/aws-eventnames</list>
  <list>etc/lists/security-eventchannel</list>

  <!-- User-defined ruleset -->
  <decoder_dir>etc/decoders</decoder_dir>
  <rule_dir>etc/rules</rule_dir>
</ruleset>
```

Obrázek 34. Wazuh server nastavení pravidel

Modul SCA

Modul SCA (Security Configuration Assessment) provádí ověření (Obrázek 35), zda koncový klienti mají správně nastavená konfigurační pravidla. Jedním z těchto pravidel je nastavení kontroly metodiky CIS. Kontrola nastavení se provádí každých 12 hodin a její výsledek je zobrazen v UI program Wazuh [75].

```
<sca>
  <enabled>yes</enabled>
  <scan_on_start>yes</scan_on_start>
  <interval>12h</interval>
  <skip_nfs>yes</skip_nfs>
</sca>
```

Obrázek 35. Wazuh server konfigurace modulu SCA

Detekce zranitelností

Dále byl nastaven modul detekce zranitelností (*vulnerability-detection*) na klientských systémech pro linuxové distribuce založené na distribuci Red Hat Enterprise Linux (Obrázek 36). Systém podporuje další klienty, jakými jsou např. Ubuntu, Debian, SuSE a operační systém Windows.

Úkolem funkce detekce zranitelností je odhalování bezpečnostních chyb v operačním systému, aplikaci klientského systému [76].

```
<vulnerability-detector>
  <enabled>yes</enabled>
  <interval>5m</interval>
  <min_full_scan_interval>6h</min_full_scan_interval>
  <run_on_start>yes</run_on_start>

  <!-- RedHat OS vulnerabilities -->
  <provider name="redhat">
    <enabled>yes</enabled>
    <os>5</os>
    <os>6</os>
    <os>7</os>
    <os>8</os>
    <os>9</os>
    <os allow="Rocky Linux-9">9</os>
    <update_interval>1h</update_interval>
  </provider>

  <!-- RockyLinux OS vulnerabilities -->
  <provider name="rockylinux">
    <enabled>yes</enabled>
    <os>8</os>
    <os>9</os>
    <update_interval>1h</update_interval>
  </provider>

  <!-- Aggregate vulnerabilities -->
  <provider name="nvd">
    <enabled>yes</enabled>
    <update_interval>1h</update_interval>
  </provider>

</vulnerability-detector>
```

Obrázek 36. Wazuh server konfigurace modulu detekce zranitelností

Detekce rootkitů

K detekci rootkitů byl použit modul *rootcheck* (Obrázek 37), který monitoruje podezřelé chování a pokud dojde k nějaké anomálii tak zašle upozornění o této události. Kontrola probíhá jednou za 12 hodin, a používá se k tomu databáze rootkitů, nakonfigurovaná s parametry rootkit_files a rootkit_trojans. Tyto soubory jsou součástí instalace [77].

```
<rootcheck>
  <disabled>no</disabled>
  <check_files>yes</check_files>
  <check_trojans>yes</check_trojans>
  <check_dev>yes</check_dev>
  <check_sys>yes</check_sys>
  <check_pids>yes</check_pids>
  <check_ports>yes</check_ports>
  <check_if>yes</check_if>

  <!-- Frequency that rootcheck is executed - every 12 hours -->
  <frequency>43200</frequency>

  <rootkit_files>etc/rootcheck/rootkit_files.txt</rootkit_files>
  <rootkit_trojans>etc/rootcheck/rootkit_trojans.txt</rootkit_trojans>

  <skip_nfs>yes</skip_nfs>
</rootcheck>
```

Obrázek 37. Wazuh server konfigurace modulu rootcheck

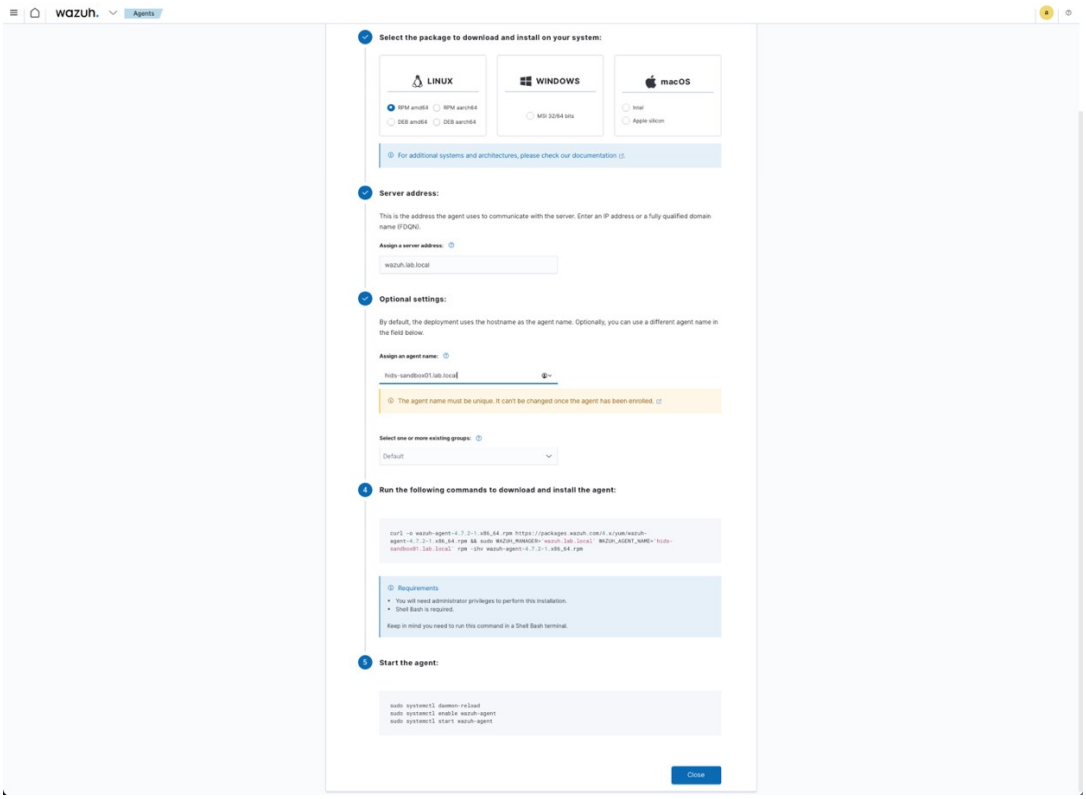
Integrita souborového systému

Součástí detekce proti malware je modul pro sledování integrity souborů (FIM – File Integrity Monitoring). Tato funkce byla nastavena pomocí konfiguračního parametru *syscheck*. Úkolem této kontroly je monitoring změn na souborovém systému jak souborů, tak adresářů. Systém si vytvoří vlastní databázi s kontrolním součty a ostatními sledovanými atributy v zašifrované podobě. Pokud dojde ke změně dojde i ke kontrole kontrolního součtu oproti uložené informaci tak systém upozorní na tuto změnu (viz Příloha P I CD-ROM) [78].

6.2 Wazuh agent

Wazuh agent určený pro linuxové operační systémy je volně dostupný na webových stránkách projektu Wazuh v několika instalačních formátech ve formátu RPM pro systémy odvozené od distribuce Red Hat Enterprise Linux nebo DEB pro systém Debian a jeho deriváty. Agent dále podporuje operační systém Windows and macOS.

Na Obrázku 38 je zobrazena konfigurační stránka z webového uživatelského rozhraní serveru Wazuh – generování instalace klienta. Zde lze zvolit typ operačního systému, zadat jméno Wazuh serveru a jméno agenta. Na této konfigurační stránce se pouze generují výsledné příkazy, které je potřeba následně spustit na klientském systému.



Obrázek 38. Wazuh agent

Vygenerované instalační příkazy (Obrázek 39) stáhnou instalační RPM balíček na lokální disk, deklarují proměnné WAZUH_MANAGER se jménem serverového systému a WAZUH_AGENT_NAME se jménem klienta.

```
[testuser@hids-sandbox01 ~]$ curl -o wazuh-agent-4.7.3-1.x86_64.rpm https://packages.wazuh.com/4.x/yum/
wazuh-agent-4.7.3-1.x86_64.rpm && sudo WAZUH_MANAGER='wazuh-server.lab.localhost' WAZUH_AGENT_NAME='hid
s-sandbox01.lab.local' rpm -ihv wazuh-agent-4.7.3-1.x86_64.rpm
```

Obrázek 39. Wazuh agent manuální instalace

Následně se znovu načte démon *systemd* pomocí příkazu:

```
sudo systemctl daemon-reload
```

Dále se povolí automatický start služby `wazuh-agent`:

```
sudo systemctl enable wazuh-agent
```

A nakonec se služba nastartuje:

```
sudo systemctl start wazuh-agent
```

Pokud nedošlo k žádné komplikaci při startu služby wazuh-agent, tak dojde k zobrazení nového zařízení (agenta) v přehledu agentů na straně serveru.

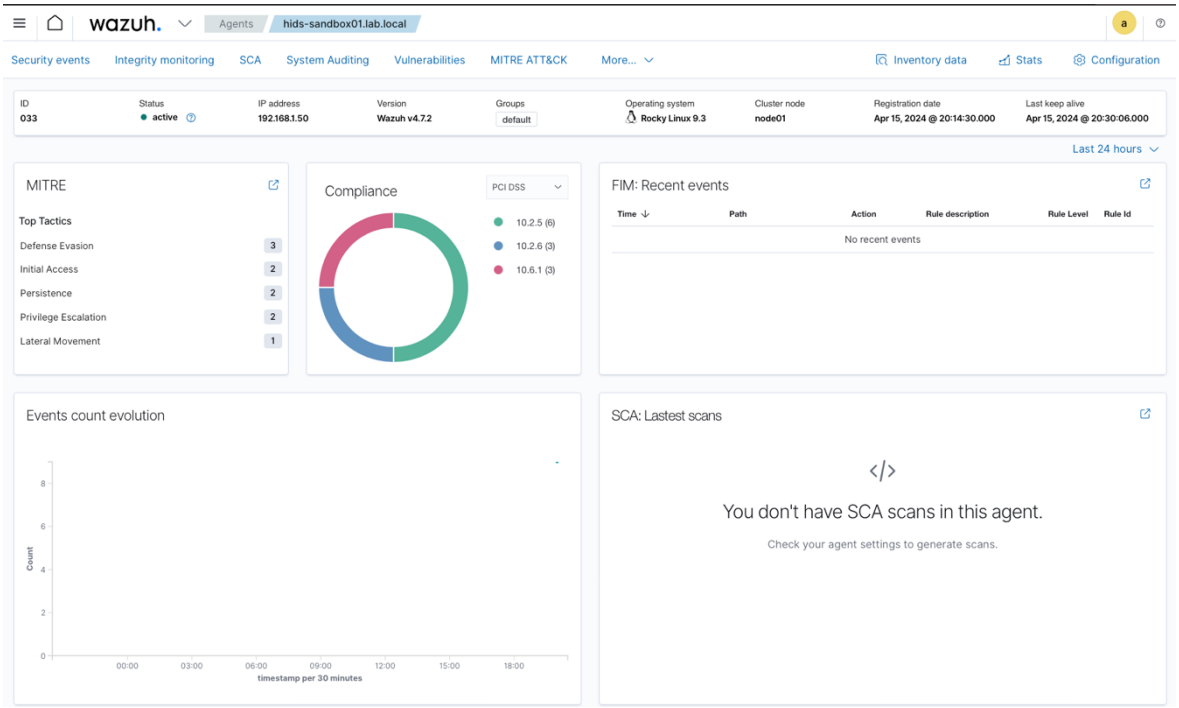
6.2.1 Instalace

Pro zjednodušení instalace Wazuh agenta byl vytvořen další Ansible scénář *wazuh_install_agent.yml*, který přidá repozitář instalačního zdroje na klientský systém a pomocí příkazu yum/dnf (určený pro distribuce odvození od operačního systému Red Hat Linux) nainstaluje požadovaný instalační RPM a dále ho nakonfiguruje, nastaví pravidla detekce a spustí Wazuh agenta. [79].

```
→  hids ansible-playbook -i inventories/dev/hosts playbooks/wazuh_install_agent.yml -K -u testuser
BECOME password:

PLAY [sandbox01] ***************************************************************************************

TASK [Gathering Facts] *********************************************************************************
ok: [192.168.1.50]

TASK [Check if Wazuh is Installed] *********************************************************************
ok: [192.168.1.50]

TASK [Check if a package exists] ***********************************************************************
ok: [192.168.1.50]

TASK [Fetch  Wazuh-agent installation package] *********************************************************
changed: [192.168.1.50]

TASK [Install Wazuh-agent] *****************************************************************************
changed: [192.168.1.50]

TASK [Check if exist directory] ************************************************************************
ok: [192.168.1.50]

TASK [Add custom Wazuh-agent configuration] ************************************************************
changed: [192.168.1.50]

TASK [Wazuh-agent systemd] *****************************************************************************
changed: [192.168.1.50]

PLAY RECAP *********************************************************************************************
192.168.1.50               : ok=8    changed=4    unreachable=0    failed=0    skipped=0    rescued=0
  ignored=0
```

Obrázek 40. Ansible instalace Wazuh agenta

Agent se po spuštění spojí se serverem Wazuh a dojde k jeho registraci. Následně se po úspěšné registraci zobrazí v přehledu agentů s detailními informacemi (Obrázek 40) jakými jsou jméno (name), IP adresa (IP address), grupa (group), operační systém (operating system), verze agenta (agent version), status agenta (agent status) zda je aktivní (active) nebo odpojený (disconnected).



Obrázek 41. Wazuh server souhrnný přehled informací o agentovi

Je nutné poznamenat, že v produkčním prostředí by se nikdy neměl software instalovat přímo ze zdroje projektu z Internetu. Softwarové balíčky by měly být staženy na lokální softwarové úložiště například lokální webserver poskytující RPM balíky. Vždy musí dojít ke kontrole kontrolního součtu balíčku, zda odpovídá kontrolnímu součtu publikovaným na webových stránkách projektu. Nový balíček musí být řádně otestován v testovacím prostředí, než bude nasazen do produkčního prostředí. Tímto způsobem se zabrání možným nekonzistencím ve verzích software a zamezení instalace škodlivého software v rámci organizace.

6.2.2 Konfigurace

Klientská část byla nakonfigurována tak, aby byla schopná detekovat podezřelé aktivity způsobené programy typu malware. Dále kontroluje, zda klientské nastavení odpovídá požadavkům CIS metodologie. Wazuh agent provádí skenování na známé zranitelnosti ve spolupráci se serverovou částí. Systém sleduje aktivity ohledně integrity souborového systému. Všechny zmíněné kontrolní mechanismy zasílají report zpět serverové části.

Konfigurace Wazuh agenta soubor *ossec.conf* je uložený v adresáři */var/ossec/etc/*.

Celý konfigurační soubor je v Příloze P I CD-ROM.

Konfigurace agenta

Konfigurace klientské strany je zobrazena na Obrázku 42, kde je nastavena adresa serveru (address), port (port), protokol (protocol), konfigurační profil (config-profile), jméno agenta (agent_name) atd. [80].

```
<client>
  <server>
    <address>wazuh.lab.local</address>
    <port>1514</port>
    <protocol>tcp</protocol>
  </server>
  <config-profile>rhel, rhel9, rhel9.3</config-profile>
  <notify_time>10</notify_time>
  <time-reconnect>60</time-reconnect>
  <auto_restart>yes</auto_restart>
  <crypto_method>aes</crypto_method>
  <enrollment>
    <enabled>yes</enabled>
    <agent_name>hids-sandbox01.lab.local</agent_name>
    <authorization_pass_path>etc/authd.pass</authorization_pass_path>
  </enrollment>
</client>
```

Obrázek 42. Wazuh konfigurace agenta

Modul Root check

Modul *rootcheck* byl nastaven pro sledování podezřelých aktivit způsobených škodlivým softwarem typu malware (Obrázek 43).

```
<rootcheck>
  <disabled>no</disabled>
  <check_files>yes</check_files>
  <check_trojans>yes</check_trojans>
  <check_dev>yes</check_dev>
  <check_sys>yes</check_sys>
  <check_pids>yes</check_pids>
  <check_ports>yes</check_ports>
  <check_if>yes</check_if>

  <!-- Frequency that rootcheck is executed - every 12 hours -->
  <frequency>43200</frequency>

  <rootkit_files>etc/shared/rootkit_files.txt</rootkit_files>
  <rootkit_trojans>etc/shared/rootkit_trojans.txt</rootkit_trojans>

  <skip_nfs>yes</skip_nfs>
</rootcheck>
```

Obrázek 43. Wazuh agent konfigurace modulu rootcheck

Modul OpenSCAP

OpenSCAP (Security Content Automation Protocol) je určený pro kontrolu bezpečnostních politik a byl nakonfigurován prostřednictvím konfiguračního parametru *wodle* (Obrázek 44) [81].

```
<wodle name="open-scap">
 <content type="xccdf" path="ssg-rhel9-ds.xml">
   <profile>xccdf_org.ssgproject.content_profile_pci-dss</profile>
 </content>
</wodle>
```

Obrázek 44. Wazuh agent konfigurace woodle open-scap

Modul SCA

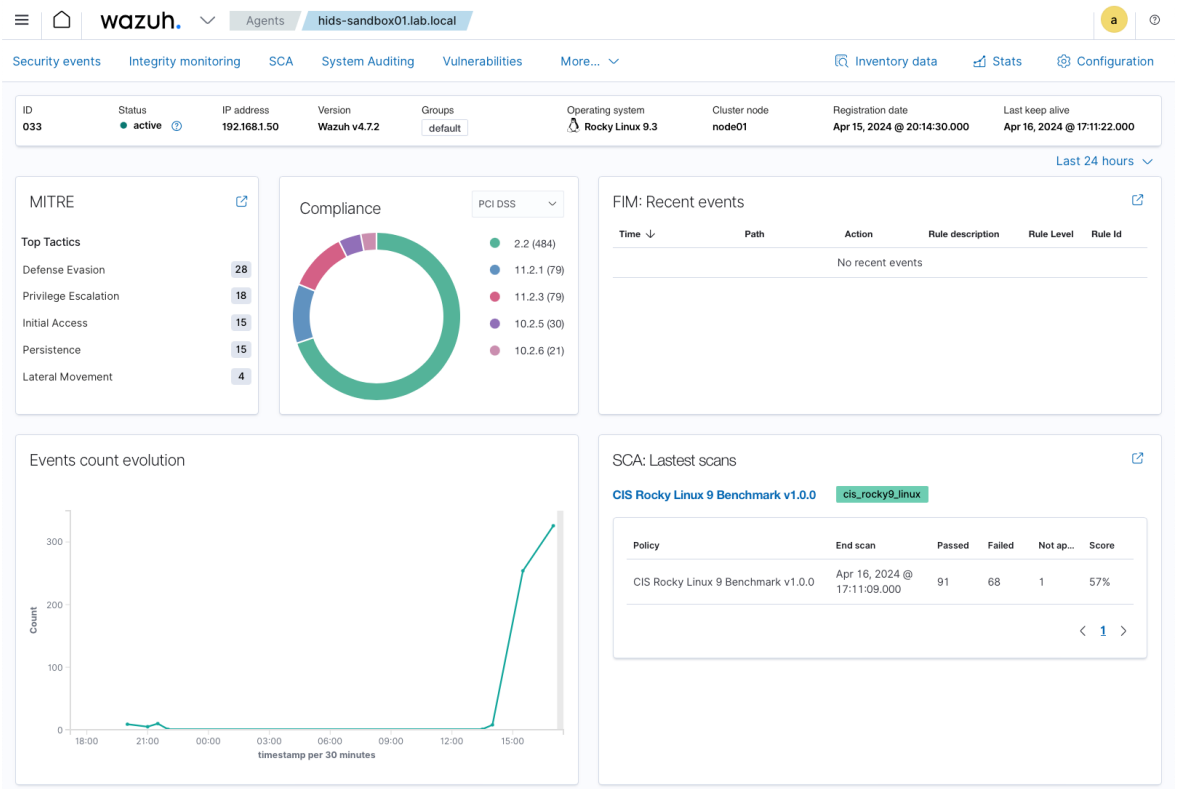
Na klientské straně byl rovněž nastaven modul SCA (Security Control Assessment) pro kontrolu metodiky CIS. Kontrola se provádí každých 12 hodin stejně jako v serverové části a její výsledek je odeslán serverové části programu Wazuh (Obrázek 45).

```
<sca>
  <enabled>yes</enabled>
  <scan_on_start>yes</scan_on_start>
  <interval>12h</interval>
  <skip_nfs>yes</skip_nfs>
</sca>
```

Obrázek 45. Wazuh agent konfigurace modulu SCA

Jak je patrné z Obrázku 41 v okně s přehledem o SCA skenování, tak tato informace není dostupná, a není dostupná ani po stanoveném časovém úseku, tj.12 h. Je to dáno tím, že CIS ukazatel (benchmark) nebyl pro distribuci Rocky Linux 9 definován, a tudíž nebyl na klientské stanici dostupný.

Vzhledem k tomu, že distribuce Rocky Linux vychází z distribuce Red Hat Enterprise Linux, tak byl převzat již existující CIS profil *cis_rhel9_linux.yml* uložený v adresáři */var/ossec/ruleset/sca/* a upravený (byl provedeny změny týkající se jména distribuce) a přejmenovaný na *cis_rocky9_linux.yml*. Po provedení této aktualizace byl agent restartován a v souhrnném přehledu klienta (Obrázek 46) je již zobrazen výsledek CIS testů.



Obrázek 46. Wazuh server souhrnný přehled o agentovi

Kontrola integrity souborů

Na klientské straně byla nakonfigurována kontrola integrity souborů (FIM) stejně jako na serverové části (viz Příloha P I CD-ROM).

Logovací soubory

Pro kontrolu logovacích souborů je na agentu nakonfigurován modul *logcollector* pro sběr logovacích souborů ze systému a jejich zasílání do serverové části pro další vyhodnocení (Obrázek 47) [82].

```
<localfile>
  <log_format>audit</log_format>
  <location>/var/log/audit/audit.log</location>
</localfile>

<localfile>
  <log_format>syslog</log_format>
  <location>/var/ossec/logs/active-responses.log</location>
</localfile>

<localfile>
  <log_format>syslog</log_format>
  <location>/var/log/messages</location>
</localfile>

<localfile>
  <log_format>syslog</log_format>
  <location>/var/log/secure</location>
</localfile>
```

Obrázek 47. Wazuh agent ukázka konfigurace modulu logcollector

Konfigurace zranitelností

Součástí standardní konfigurace klienta je i nastavení modulu *syscollector* (Obrázek 48), který shromažďuje informace o systému. Na straně serveru tyto informace zpracovává již zmíněný modul detekce zranitelností (*vulnerability management*) (Kapitola 6.1.2) [83;84].

```
<!-- System inventory -->
<wodle name="syscollector">
  <disabled>no</disabled>
  <interval>1h</interval>
  <scan_on_start>yes</scan_on_start>
  <hardware>yes</hardware>
  <os>yes</os>
  <network>yes</network>
  <packages>yes</packages>
  <ports all="no">yes</ports>
  <processes>yes</processes>

  <!-- Database synchronization settings -->
  <synchronization>
    <max_eps>10</max_eps>
  </synchronization>
</wodle>
```

Obrázek 48. Wazuh agent konfigurace modulu syscollector

7 TESTOVÁNÍ

Hlavním cílem této kapitoly je otestování HIDS řešení pomocí penetračních testů, které se běžně aplikují v organizacích z důvodů testování bezpečnosti a vyhodnocení možný slabin systémů a následně jejich řešení.

Penetrační testování je definováno jako [33]: „*Zkoumání funkcí počítačového systému a sítí s cílem najít slabá místa počítačové bezpečnosti tak, aby bylo možno tato slabá místa odstranit*.“ a je obvykle vykonáváno etickým hackerem, nebo je také tato osoba označována v terminologii jako white hacker. NÚKIB definoval etického hackera (ethical hacker) následovně [33]: „*Etický hacker, který je často zaměstnáván jako expert počítačové bezpečnosti, programátor nebo správce sítí. Specializuje se na penetrační testy a jiné testovací metodiky k zajištění IT bezpečnosti v organizaci.*“.

Penetračního testování se řídí podle několika mezinárodně definovaných metodologií, jakými jsou např.

- OWASP Web Security Testing Guide[49],
- NIST SP 800-115[50],
- OSSTMM[51],
- PTES[52].

Cílem penetračního testování je simulovat techniky a postupy, které jsou prováděny protivníky, a odhalit bezpečností chyby dříve, než dojde k jejich zneužití.

7.1 Postupy útočníků

Útočníci zpravidla postupují následujícím způsobem podle Mitre ATT&CK® frameworku [85]:

- průzkum (Reconnaissance),
- zjišťování (Discovery),
- zvýšení práv (Escalation Privileges),

[49] https://owasp.org/www-project-web-security-testing-guide/
[50] https://csrc.nist.gov/pubs/sp/800/115/final
[51] https://www.isecom.org/research.html#content5-9d
[52] http://www.pentest-standard.org/index.php/Main_Page

- perzistence (Persistence),
- zahlazení stop (Indicators Removal on Host).

7.1.1 Průzkum (Reconnaissance)

Technika průzkumu TA0043 se zabývá aktivním nebo pasivním hledáním informací, které by mohli vést k provedení přístupu k cílovému systému [86].

7.1.2 Zjišťování (Discovery)

Technika zjišťování TA0007 hledá informace o systémech umístěných ve vnitřní sítí. Tyto informace mu pak pomohou k rozhodování o dalšímu postupu, jak se dostat k cílovému systému [87].

7.1.3 Zvýšení oprávnění (Privilege Escalation)

Technika zvýšení oprávnění TA0004 vede ke zvýšení oprávnění na úrovni operačního systému nebo v rámci lokální počítačové sítě. Útočník se snaží využít zranitelností v systému, které jsou způsobeny například softwarovými chybami, které obsahují bezpečnostní chyby a nebyly odstraněny. Dále mohou být způsobeny chybami v systémové konfiguraci nebo v nastavení programů, které také umožní eskalaci práv [88].

7.1.4 Persistence (Persistence)

Další technikou je perzistence (TA0003) neboli udržení oprávnění v napadeném systému. Technika obsahuje akce, které se snaží uchovat si přístup i po restartu systému. Dále může docházet ke změnám konfiguračních souborů, k nahrazení programů modifikovanými verzemi tzv. zadní vrátka (backdoor), které umožnují přístup do systému a obejití standardních bezpečnostních mechanismů [89].

7.1.5 Zahlazení stop (Indicators Removal on Host)

Techniku zahlazení stop (T0872) využívají útočníci k odstranění změn, které udělali. Mohou měnit logovací soubory nebo měnit historii provedených příkazů. Opět je jejich cílem skrýt jakékoliv záznamy o jejich přítomnosti, které by mohli vést k jejich odhalení [90].

7.2 Testovací prostředí

Jednotlivé testy byly provedeny ve vytvořeném testovacím prostředí na již dříve popsané virtualizační platformě VMware (Kapitola 4).

7.3 Vzdálené testování

Vzdálené testování je dále popsáno v této kapitole a bylo realizováno z virtuálního serveru s distribucí Kali Linux (Kapitola 4.1.4) a byl testován virtuální server s operačním systémem Rocky Linux s agentem programu Wazuh.

7.3.1 OpenVAS

Open source skener zranitelností OpenVAS (Open Vulnerability Assessment System) byl nainstalován na virtuální server s distribucí Kali Linux.

Projekt OpenVAS vznikl v roce 2006 jako odnož (fork) skeneru Nessus[53]. V roce 2005 projekt Nessus změnil své licenční podmínky a přešel z otevřeného zdrojového kódu na uzavřenou proprietární licenci. Cílem projektu OpenVAS a dalších projektů vycházejících z programu Nessus bylo pokračovat ve vývoji tohoto skeneru ve formě otevřené softwarové licence [91].

Instalace skeneru

Program je dostupný jako Greenbone Community Edition z repozitáře distribuce Kali Linux a byl nainstalován na dedikovaný systém s touto distribucí [92].

```
sudo apt install gvm
```

Konfigurace

Následující skript provede konfiguraci skeneru [92].

```
sudo gvm-setup
```

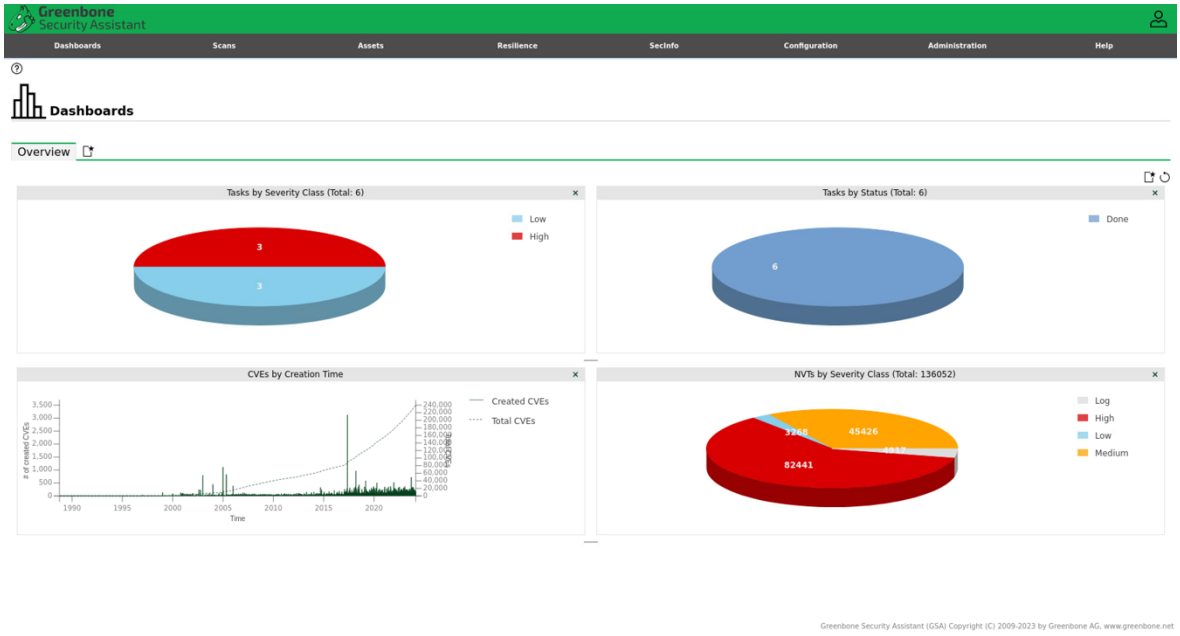
Ověření činnosti

Dále je nutné spustit další skript, který ověří, zda byl program *gvm* správně nastaven [92].

```
sudo gvm-check-setup
```

[53] https://www.tenable.com/products/nessus

Pokud všechny služby fungují správně tak se lze přihlásit do programu pomocí webového prohlížeče z grafického prostředí distribuce Kali Linux a na adrese *https://localhost:9392* , prostředí skeneru OpenVAS je zobrazeno na Obrázku 49 [92].



Obrázek 49. OpenVAS prostředí

7.3.2 Nmap

Program Nmap byl vytvořený Gordonem Lyonem (Fyodor) již v roce 1997 a původně byl napsaný pro jeho soukromé účely na platformě Linux. Během let se popularita tohoto skeneru prudce rozšířila, byly přidány další možnosti a podpora operačních systému. Skener Nmap byl použit pro testování portů [93] a testování zranitelností prostřednictvím NSE (Nmap Scripting Engine) [94].

Skenování portů

Jako první krok byla provedena rekognoskace všech portů (Obrázek 50) s parametrem [95]:

-p- výběr portů od 0 do 65535

```
┌──(kali㉿kali)-[~]
└─$ sudo /usr/bin/nmap -p- 192.168.1.50
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-05-04 09:40 CEST
Nmap scan report for hids-sandbox01.lab.local (192.168.1.50)
Host is up (0.00049s latency).
Not shown: 65383 filtered tcp ports (no-response), 150 filtered tcp ports (admin-prohibited)
PORT     STATE  SERVICE
22/tcp   open   ssh
9090/tcp closed zeus-admin
MAC Address: 00:50:56:87:8B:6F (VMware)

Nmap done: 1 IP address (1 host up) scanned in 144.78 seconds
```

Obrázek 50. Příkaz Nmap sken portů

Program Nmap provedl sken portů v rozmezí 0–65535 a program Wazuh, resp. fronta událostí agenta programu Wazuh byla zahlcena. Z toho důvodu by mohlo dojít ke ztrátě dalších událostí (Obrázek 51). Test byl několikrát zopakován se stejným parametrem a výsledek byl vždy totožný.

Security Alerts

Time	Technique(s)	Tactic(s)	Description	Level	Rule ID
Apr 22, 2024 @ 22:56:50.950			Agent event queue is back to normal load.	3	205
Apr 22, 2024 @ 22:56:47.684			Agent event queue is full. Events may be lost.	9	203
Apr 22, 2024 @ 22:56:45.653			Agent event queue is full. Events may be lost.	9	203
Apr 22, 2024 @ 22:56:43.634			Agent event queue is full. Events may be lost.	9	203
Apr 22, 2024 @ 22:56:41.616			Agent event queue is full. Events may be lost.	9	203
Apr 22, 2024 @ 22:56:39.621			Agent event queue is 90% full.	7	202
Apr 22, 2024 @ 22:56:25.472			Auditd: Process ended abnormally.	10	80711
Apr 22, 2024 @ 22:51:44.394			Agent event queue is back to normal load.	3	205
Apr 22, 2024 @ 22:51:41.113			Agent event queue is full. Events may be lost.	9	203
Apr 22, 2024 @ 22:51:39.137			Agent event queue is full. Events may be lost.	9	203

Rows per page: 10 ‹ 1 2 3 4 5 … 21 ›

Obrázek 51. Wazuh přehled detekovaných událostí

Skenování více typů

Dále bylo provedeno pokročilé skenování klientského systému, které kromě již dříve zmíněného otevřeného portu 22 a služby OpenSSH, dále detekovalo otisky veřejného klíče cílového SSH serveru (ECDSA, ED25519). Nmap odhadnul operační systém jako Linux s verzí jádra 5.0–5.4 na 98 %, což odpovídá běžícímu jádru v 5.14.0 (Obrázek 52). Následně byl spuštěn příkaz traceroute s výsledkem 1 skok (hop) vzhledem k tomu, že oba virtuální servery jsou umístěny na stejné podsíti192.168.1.0/24.

```
[testuser@hids-sandbox01 ~]$ uname -r
5.14.0-362.24.1.el9_3.0.1.x86_64
```

Obrázek 52. Verze jádra

Program Nmap byl spuštěn s parametrem (Obrázek 53) [95]:

-A Pokročilé nebo agresivní skenování jako je detekce OS, služeb, NSE skenování, traceroute

```
┌──(kali㉿kali)-[ ]
└─$ sudo /usr/bin/nmap -A 192.168.1.50
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-04-30 19:20 CEST
Nmap scan report for hids-sandbox01.lab.local (192.168.1.50)
Host is up (0.00053s latency).
Not shown: 988 filtered tcp ports (no-response), 10 filtered tcp ports (admin-prohibited)
PORT     STATE  SERVICE    VERSION
22/tcp   open   ssh        OpenSSH 8.7 (protocol 2.0)
| ssh-hostkey:
|   256 14:5f:1a:5b:c1:00:1e:9b:79:35:be:3d:45:5c:0b:1c (ECDSA)
|_  256 2f:24:4a:b9:fb:92:07:2e:7c:ef:63:d9:b2:46:a2:68 (ED25519)
9090/tcp closed zeus-admin
MAC Address: 00:50:56:87:8B:6F (VMware)
Device type: general purpose|storage-misc|WAP|media device
Running (JUST GUESSING): Linux 5.X|4.X|2.6.X|3.X (98%), HP embedded (89%), Ubiquiti embedded (89%), Infom
ir embedded (88%)
OS CPE: cpe:/o:linux:linux_kernel:5 cpe:/o:linux:linux_kernel:4 cpe:/o:linux:linux_kernel:2.6 cpe:/o:linu
x:linux_kernel:3 cpe:/h:hp:p2000_g3 cpe:/o:linux:linux_kernel:2.6.32 cpe:/h:ubnt:airmax_nanostation cpe:/
h:infomir:mag-250
Aggressive OS guesses: Linux 5.0 - 5.4 (98%), Linux 4.15 - 5.8 (94%), Linux 5.0 - 5.5 (93%), Linux 2.6.32
 - 3.13 (93%), Linux 2.6.39 (93%), Linux 5.1 (92%), Linux 2.6.22 - 2.6.36 (91%), Linux 3.10 - 4.11 (91%),
 Linux 5.0 (91%), Linux 5.4 (90%)
No exact OS matches for host (test conditions non-ideal).
Network Distance: 1 hop

TRACEROUTE
HOP RTT     ADDRESS
1   0.54 ms hids-sandbox01.lab.local (192.168.1.50)

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 14.83 seconds
```

Obrázek 53. Příkaz Nmap pokročilé skenování

Skenování zranitelností s NSE skripty

Skenování zranitelností bylo provedeno s pomocí NSE modulu *vulners*, který porovnává výsledky s databází *http://vulners.com* [96].

Program Nmap byl spuštěn s následujícími parametry [95]:

-p-	výběr portů od 0 do 65535
-sV	test portů za účelem zjištění služeb
--script	NSE skript
-p22	porty
--open	zobrazení pouze otevřených portů

-T4 nastavení časování mezi jednotlivými kroky

Byl proveden sken zranitelností před aktualizací software prostřednictvím příkazu *dnf update* a výsledek tohoto testu je zobrazen na Obrázku 54.

```
┌──(kali㉿kali)-[ ]
└─$ sudo /usr/bin/nmap -sV --script vulners -p22 --open -T4 192.168.1.50
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-04-30 19:22 CEST
Nmap scan report for hids-sandbox01.lab.local (192.168.1.50)
Host is up (0.00068s latency).

PORT   STATE SERVICE VERSION
22/tcp open  ssh     OpenSSH 8.7 (protocol 2.0)
| vulners:
|   cpe:/a:openbsd:openssh:8.7:
|       CVE-2012-1577   7.5     https://vulners.com/cve/CVE-2012-1577
|       CVE-2010-4816   5.0     https://vulners.com/cve/CVE-2010-4816
|       PRION:CVE-2021-41617    4.4     https://vulners.com/prion/PRION:CVE-2021-41617
|       CVE-2021-41617  4.4     https://vulners.com/cve/CVE-2021-41617
|       PRION:CVE-2016-20012    4.3     https://vulners.com/prion/PRION:CVE-2016-20012
|       CVE-2016-20012  4.3     https://vulners.com/cve/CVE-2016-20012
|       CVE-2023-51767  3.5     https://vulners.com/cve/CVE-2023-51767
|       PRION:CVE-2021-36368    2.6     https://vulners.com/prion/PRION:CVE-2021-36368
|_      CVE-2021-36368  2.6     https://vulners.com/cve/CVE-2021-36368
MAC Address: 00:50:56:87:8B:6F (VMware)

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 1.17 seconds
```

Obrázek 54. Příkaz Nmap skenování zranitelností před aktualizací software

Následně byla provedena aktualizace systému příkazem *dnf update* a restart systému a proveden druhý test (Obrázek 55).

```
┌──(kali㉿kali)-[~]
└─$ sudo /usr/bin/nmap -sV --script vulners -p22 --open -T4 192.168.1.50
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-04-30 22:03 CEST
Nmap scan report for hids-sandbox01.lab.local (192.168.1.50)
Host is up (0.00080s latency).

PORT   STATE SERVICE VERSION
22/tcp open  ssh     OpenSSH 8.7 (protocol 2.0)
| vulners:
|   cpe:/a:openbsd:openssh:8.7:
|       CVE-2012-1577   7.5     https://vulners.com/cve/CVE-2012-1577
|       CVE-2010-4816   5.0     https://vulners.com/cve/CVE-2010-4816
|       PRION:CVE-2021-41617    4.4     https://vulners.com/prion/PRION:CVE-2021-41617
|       CVE-2021-41617  4.4     https://vulners.com/cve/CVE-2021-41617
|       PRION:CVE-2016-20012    4.3     https://vulners.com/prion/PRION:CVE-2016-20012
|       CVE-2016-20012  4.3     https://vulners.com/cve/CVE-2016-20012
|       CVE-2023-51767  3.5     https://vulners.com/cve/CVE-2023-51767
|       PRION:CVE-2021-36368    2.6     https://vulners.com/prion/PRION:CVE-2021-36368
|_      CVE-2021-36368  2.6     https://vulners.com/cve/CVE-2021-36368
MAC Address: 00:50:56:87:8B:6F (VMware)

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 1.16 seconds
```

Obrázek 55. Příkaz Nmap skenování zranitelností po aktualizací software

Techniky oklamání detekce

Úkolem testu na oklamání detekce bylo použít fragmentované pakety a sledovat, zda byl tento pokus úspěšný nebo nikoliv (Obrázek 56).

Program Nmap byl spuštěn s následujícími parametry [95]:

-f sken používá malé fragmentované pakety

-T1 nastavení časování mezi jednotlivými kroky

-n bez překladu IP adres na jména

-Pn bez zjišťování hostitele (neověřuje hosta, pokud je dostupný příkazem ping)

--data-length 200 přidání náhodných dat do odesílaných dat

-D [IP1, IP2...] falešné IP adresy

```
┌──(kali㉿kali)-[~]
└─$ sudo nmap -f -T1  -n -Pn --data-length 200 -D 192.168.1.101,192.168.1.102,192.168.1.103,192.168.1.104
,192.168.1.105  192.168.1.50
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-04-29 20:55 CEST
Nmap scan report for 192.168.1.50
Host is up (0.0046s latency).
Not shown: 998 filtered tcp ports (admin-prohibited)
PORT     STATE  SERVICE
22/tcp   open   ssh
9090/tcp closed zeus-admin
MAC Address: 00:50:56:87:A0:BB (VMware)

Nmap done: 1 IP address (1 host up) scanned in 16513.46 seconds
```

Obrázek 56. Příkaz Nmap test oklamání detekce

V tomto případě detekční sytém Wazuh nezachytil žádnou podezřelou aktivitu.

7.3.3 Hydra

Program Hydra je nástroj na prolamování uživatelských jmen a hesel pomocí slovníkového útoku. Jedná se o útok hrubou silou tzv. bruteforce útok. Program podporuje širokou škálu protokolů.

Byl proveden pokus o útok hrubou silou na port 22, služby SSH. Jedná se o velmi dlouhý proces a velmi snadno detekovaný.

```
┌──(root㉿kali)-[~]
└─# hydra -f  -V  -t 4  -L  /usr/share/seclists/Usernames/Names/names.txt -P /usr/share/seclists/Password
s/2020-200_most_used_passwords.txt -s 22 192.168.1.50  ssh
Hydra v9.5 (c) 2023 by van Hauser/THC & David Maciejak - Please do not use in military or secret service
organizations, or for illegal purposes (this is non-binding, these *** ignore laws and ethics anyway).

Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2024-04-30 19:44:19
[DATA] max 4 tasks per 1 server, overall 4 tasks, 2004869 login tries (l:10177/p:197), ~501218 tries per
task
[DATA] attacking ssh://192.168.1.50:22/
[ATTEMPT] target 192.168.1.50 - login "aaliyah" - pass "123456" - 1 of 2004869 [child 0] (0/0)
[ATTEMPT] target 192.168.1.50 - login "aaliyah" - pass "123456789" - 2 of 2004869 [child 1] (0/0)
[ATTEMPT] target 192.168.1.50 - login "aaliyah" - pass "picture1" - 3 of 2004869 [child 2] (0/0)
[ATTEMPT] target 192.168.1.50 - login "aaliyah" - pass "password" - 4 of 2004869 [child 3] (0/0)
[ATTEMPT] target 192.168.1.50 - login "aaliyah" - pass "12345678" - 5 of 2004869 [child 1] (0/0)
[ATTEMPT] target 192.168.1.50 - login "aaliyah" - pass "111111" - 6 of 2004869 [child 3] (0/0)
[ATTEMPT] target 192.168.1.50 - login "aaliyah" - pass "123123" - 7 of 2004869 [child 0] (0/0)
[ATTEMPT] target 192.168.1.50 - login "aaliyah" - pass "12345" - 8 of 2004869 [child 2] (0/0)
[ATTEMPT] target 192.168.1.50 - login "aaliyah" - pass "1234567890" - 9 of 2004869 [child 1] (0/0)
[ATTEMPT] target 192.168.1.50 - login "aaliyah" - pass "senha" - 10 of 2004869 [child 3] (0/0)
[ATTEMPT] target 192.168.1.50 - login "aaliyah" - pass "1234567" - 11 of 2004869 [child 0] (0/0)
[ATTEMPT] target 192.168.1.50 - login "aaliyah" - pass "qwerty" - 12 of 2004869 [child 2] (0/0)
```

Obrázek 57. Hydra slovníkový útok

Program Hydra byl spuštěn s parametry (Obrázek 57) [97]:

-f program se přeruší při prvním nalezení uživatelského jména (login) a hesla (password)

-R dojde k obnovení přerušeného běhu programu, hydra si vytvoří soubor hydra.restore kam si ukládá, již otestované uživatelského jména a hesla

-V zobrazuje každý testovaný pokus

-t počet paralelních testů

-L soubor s uživatelskými jmény

-P soubor s hesly

Pro slovníkový útok byl nainstalován softwarový balíček *seclists*, který obsahuje soubory s uživatelskými účty a hesly [97;98;99].

Security Alerts

Time	Technique(s)	Tactic(s)	Description	Level	Rule ID
Apr 2, 2024 @ 17:14:15.893	T1110.001 T1021.004	Credential Access, Lateral Movement	sshd: Attempt to login using a non-existent user	5	5710
Apr 2, 2024 @ 17:14:13.890	T1110.001 T1021.004	Credential Access, Lateral Movement	sshd: Attempt to login using a non-existent user	5	5710
Apr 2, 2024 @ 17:14:09.886	T1110.001 T1021.004	Credential Access, Lateral Movement	sshd: Attempt to login using a non-existent user	5	5710
Apr 2, 2024 @ 17:14:07.893	T1110.001	Credential Access	PAM: User login failed.	5	5503
Apr 2, 2024 @ 17:14:07.884	T1110.001 T1021.004	Credential Access, Lateral Movement	sshd: Attempt to login using a non-existent user	5	5710
Apr 2, 2024 @ 17:13:37.856	T1110	Credential Access	syslog: User missed the password more than one time	10	2502
Apr 2, 2024 @ 17:13:37.854	T1110.001 T1021.004	Credential Access, Lateral Movement	sshd: Attempt to login using a non-existent user	5	5710
Apr 2, 2024 @ 17:13:37.852	T1110	Credential Access	Maximum authentication attempts exceeded.	8	5758
Apr 2, 2024 @ 17:13:37.851	T1110.001 T1021.004	Credential Access, Lateral Movement	sshd: Attempt to login using a non-existent user	5	5710
Apr 2, 2024 @ 17:13:35.849	T1110	Credential Access	sshd: brute force trying to get access to the system. Non existent user.	10	5712

Obrázek 58. Wazuh detekce útoku hrubou silou

Program Wazuh detekoval test útoku hrubou silou jako techniku (Obrázek 58).

Na základě testu slovníkovým útokem program Wazuh) detekoval techniku *T1110* s dílčí technikou *T1110.001 Password Guessing a* taktikou *Credential Access* z frameworku Mitre ATT&CK®. Tato taktika obsahuje pokusy o uhádnutí přihlašovacích údajů a hádání hesla [100].

Dále Wazuh zaznamenal techniku *T1021* s dílčí technikou *T1021.004* Remote *Services: SSH* [101] a taktikou *Lateral Movement*, kde [102] uvádí následující: *„Laterální pohyb se skládá z technik, které protivníci používají ke vstupu do vzdálených systémů v síti a k jejich ovládání. Sledování jejich primárního cíle často vyžaduje prozkoumání sítě, aby našli svůj cíl a následně k němu získali přístup.*[54]“.

7.4 Lokální testování

Lokální testování bylo provedeno dvěma způsoby. Jako první způsob byly provedeny pokusy se simulováním fyzických pokusů o průnik do daného systému. Fyzickým průnikem jsou myšleny pokusy se zadáváním uživatelských jmen a hádání hesel, dále pokus o restart

[54] Lateral Movement consists of techniques that adversaries use to enter and control remote systems on a network. Following through on their primary objective often requires exploring the network to find their target and subsequently gaining access to it [101].

operačního systému pomocí kombinace kláves CTRL + ALT + DEL a změna parametrů zavaděče systému. U těchto testů se předpokládá, že útočník má fyzický přístup k hardware.

Tyto testy by bylo možné také realizovat pomocí hardwarového USB zařízením jakým je například USB Rubber Ducky (Obrázek 59) od firmy Hak5 specializující se na zařízení pro penetrační testování.

Obrázek 59. Rubber Ducky [103]

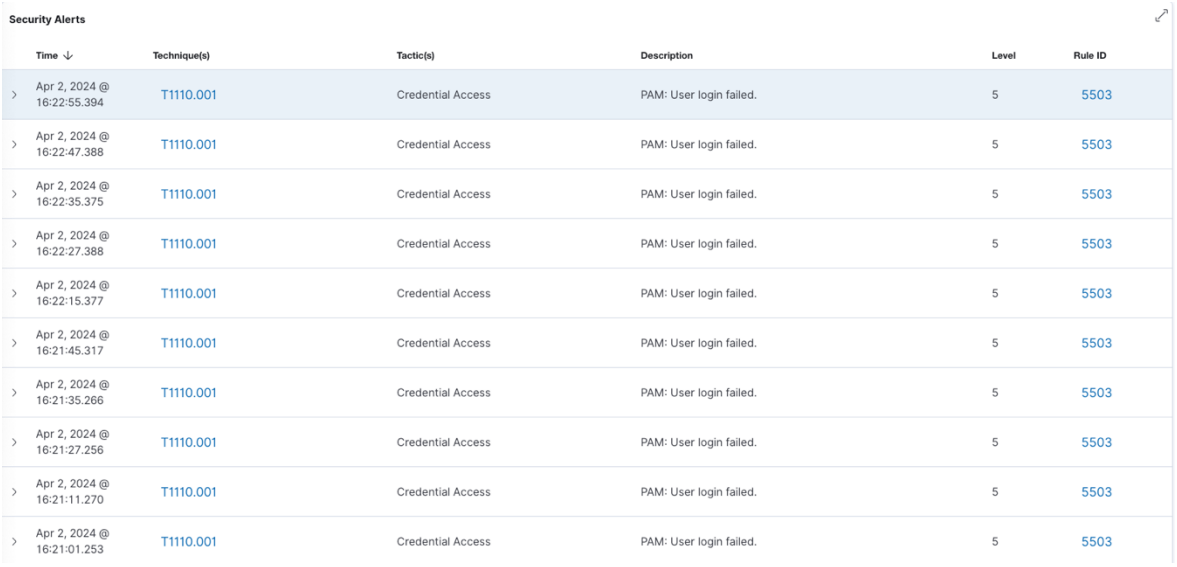
7.4.1 HID

Prostřednictvím konfiguračního nástroje Ansible a modulu *vmware_guest_sendkey* [104] pro komunikaci s virtualizačním nástrojem VMware byly nasimulovány pokusy o průnik pomocí HID (Human Interface Device)[55] zařízení tzn. klávesnice. Testy jsou k dispozici v Příloze P I CD-ROM.

Test zadání uživatele a hesla

Test *vmware_vm_HID_login.yml* simuluje pokusy o zadávání uživatelských jmen a hesel na virtuálním terminálu klientského operačního systému.

[55] https://www.usb.org/hid



Security Alerts

Time	Technique(s)	Tactic(s)	Description	Level	Rule ID
Apr 2, 2024 @ 16:22:55.394	T1110.001	Credential Access	PAM: User login failed.	5	5503
Apr 2, 2024 @ 16:22:47.388	T1110.001	Credential Access	PAM: User login failed.	5	5503
Apr 2, 2024 @ 16:22:35.375	T1110.001	Credential Access	PAM: User login failed.	5	5503
Apr 2, 2024 @ 16:22:27.388	T1110.001	Credential Access	PAM: User login failed.	5	5503
Apr 2, 2024 @ 16:22:15.377	T1110.001	Credential Access	PAM: User login failed.	5	5503
Apr 2, 2024 @ 16:21:45.317	T1110.001	Credential Access	PAM: User login failed.	5	5503
Apr 2, 2024 @ 16:21:35.266	T1110.001	Credential Access	PAM: User login failed.	5	5503
Apr 2, 2024 @ 16:21:27.256	T1110.001	Credential Access	PAM: User login failed.	5	5503
Apr 2, 2024 @ 16:21:11.270	T1110.001	Credential Access	PAM: User login failed.	5	5503
Apr 2, 2024 @ 16:21:01.253	T1110.001	Credential Access	PAM: User login failed.	5	5503

Obrázek 60. Wazuh výsledky detekce

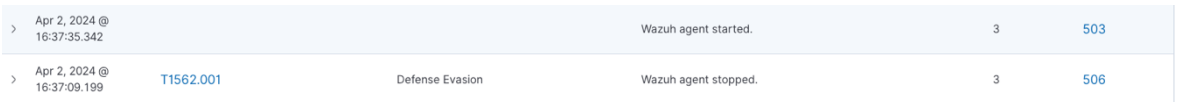
Wazuh agent detekoval neplatné pokusy o přihlášení (Obrázek 60), podle frameworku Mitre ATT&CK® detekoval techniku *T1110* s dílčí technikou *T1110.001 Password Guessing* a taktikou *Credential Access* [100].

Test restartu serveru pomocí kombinace kláves Ctrl + Alt + Del

Test *vmware_vm_HID_CTRL_ALT_DEL.yml* simuluje stisknutí kláves Ctrl + Alt + Del, zda lze restartovat server.

Výsledek testu

Vzhledem k tomu, že došlo k restartu serveru pomocí simulovaného testu, Wazuh detekoval tento test podle frameworku Mitre ATT&CK® detekoval techniku *T1562* s dílčí technikou *T1562.001 Impair Defenses: Disable or Modify Tools* a taktikou *Defense Evasion* [105] (Obrázek 61).



Apr 2, 2024 @ 16:37:35.342			Wazuh agent started.	3	503
Apr 2, 2024 @ 16:37:09.199	T1562.001	Defense Evasion	Wazuh agent stopped.	3	506

Obrázek 61. Wazuh výsledky detekce

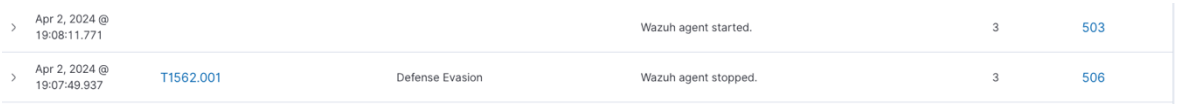
Technika *Defense Evasion* je definována jako [105]: "*Vyhýbání se obraně se skládá z technik, které protivníci používají k tomu, aby se vyhnuli odhalení v průběhu svého*

kompromitování. Mezi techniky používané k vyhýbání se obraně patří odinstalování/vypnutí bezpečnostního softwaru nebo zakrytí/zašifrování dat a skriptů.[56] *“*.

Test přerušení startovací sekvence a start systému do nouzového módu

Test *vmware_vm_HID_CTRL_ALT_DEL_grub.yml* simuluje stisknutí kláves Ctrl + Alt + Del, dále přeruší start systému (boot) a pokusí se změnit konfiguraci zavaděče systému GRUB. Cílem je změna parametrů a zavedení operačního systému do nouzového modu (rescue mode).

Výsledek testu



Obrázek 62. Wazuh výsledky detekce

Výsledkem testu, je úspěšný restart operačního systému, ale neúspěšný pokus o změnu konfigurace zavaděče GRUB. Ke změně těchto parametrů je nutné zadat heslo, které bylo nastaveno během instalace operačního systému (Kapitola 5.1.4). Stejně jako v předchozím testu systém Wazuh zaznamenal zastavení a opětovný start agenta na klientském systému techniku *T1562* s dílčí technikou *T1562.001 Impair Defenses: Disable or Modify Tools* a taktikou *Defense Evasion* [105] (Obrázek 62).

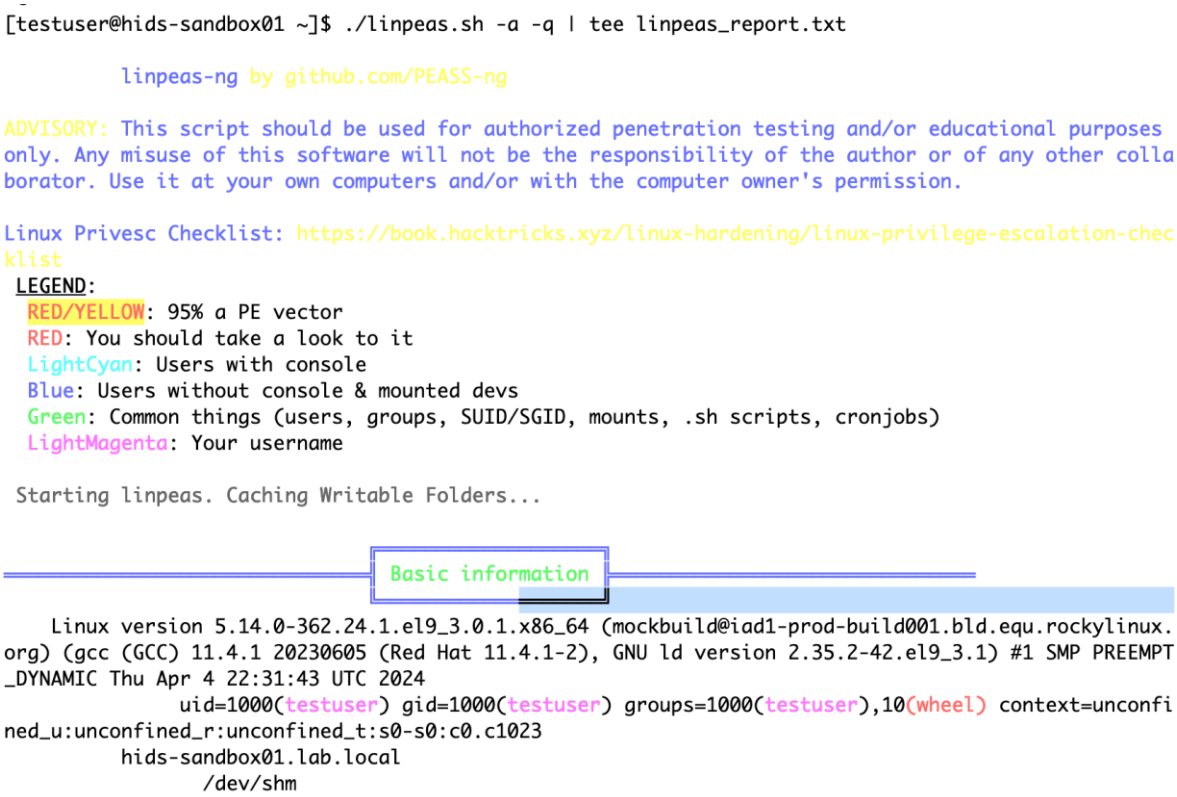
7.4.2 Test eskalace oprávnění

Dále byl proveden test z příkazové řádky pod uživatelským účtem *testuser* a pomocí skriptu *linpeas.sh*. Cílem tohoto skriptu je otestování možnosti zranitelností na lokálním systému a použití techniky eskalace práv (privilege escalation). Ve frameworku Mitre ATT&CK® se jedná o techniku *TA0004 Privilege Escalation*, která je popisována jako [106]: “*Zvýšení oprávnění se skládá z technik, které protivníci používají k získání oprávnění vyšší úrovně v systému nebo síti. Protivníci mohou často vstoupit do sítě a prozkoumat ji s*

[56] Defense Evasion consists of techniques that adversaries use to avoid detection throughout their compromise. Techniques used for defense evasion include uninstalling/disabling security software or obfuscating/encrypting data and scripts.[105]

neprivilegovaným přístupem, ale k dosažení svých cílů potřebují vyšší oprávnění. Obvyklé přístupy spočívají ve využití slabých míst systému, chybné konfigurace a zranitelností.[57]“.

Skript byl stažen z Github repozitáře [107] a následně spuštěn (Obrázek 63). Skript *linpeas.sh* nenalezl zranitelnosti (výsledek skriptu je k nalezení v Příloze P I CD-ROM), které by vedly k eskalaci práv uživatele *testuser*.



Obrázek 63. Skript linpeas.sh

[57] Privilege Escalation consists of techniques that adversaries use to gain higher-level permissions on a system or network. Adversaries can often enter and explore a network with unprivileged access but require elevated permissions to follow through on their objectives. Common approaches are to take advantage of system weaknesses, misconfigurations, and vulnerabilities.[106]

8 VYHODNOCENÍ

Kapitola se zabývá vyhodnocením výsledků penetračních testů realizovaných v Kapitole 7 a návrhu možných protiopatření. Tato protiopatření by měla vést k dalšímu zlepšení detekce a zabezpečení klientského operačního systému.

8.1 Výsledky skenování Nmap

8.1.1 Popis problému

Skenováním programem Nmap byly zjištěny následující informace:

- Otevřený port 22 (služba SSH).
- Operační systém Linux.
- MAC adresa, která odpovídá virtualizačnímu software VMware.
- Možné CVE zranitelnosti.

Během skenování všech portů došlo k zaznamenání aktivity detekčním systémem Wazuh (Kapitola 7.3.2). Jednalo se o zprávy ohledně zahlcení fronty na straně Wazuh agenta.

8.1.2 Návrh řešení

Problém zahlcení fronty příchozích zpráv na straně Wazuh agenta se dá řešit změnou parametrů *queue_size* (velikost fronty) a *events_per_second* (počet událostí za sekundu) v konfiguračním bloku *client_buffer* (Obrázek 64). Jedná se o konfigurační soubor na straně Wazuh agenta [108;109].

```
<client_buffer>
  <!-- Agent buffer options -->
  <disabled>no</disabled>
  <queue_size>5000</queue_size>
  <events_per_second>500</events_per_second>
</client_buffer>
```

Obrázek 64. Wazuh agent konfigurace zásobníku

8.1.3 Výsledek

Byla provedena změna parametru *events_per_second* (počet událostí za sekundu) z hodnoty 500 na 600 a následně na 700 a ponechána *queue_size* (velikost fronty) na hodnotě 5000. Test byl zopakován v každém případě 10x s 30sekundovou přestávkou mezi jednotlivými

testy. Systém Wazuh zaznamenal v každém běhu testu nárůst zásobníku na 90 %. Následně byla změněna hodnota parametru *events_per_second* (počtu událostí za sekundu) na 800 (maximální hodnota parametru je 1000), a celý test byl opět zopakován. V tomto případě už nedošlo k zahlcení fronty, ale zvýšení této hodnoty se projevilo na zátěži serveru (aktivita procesu rsyslog).

Byly vykonány testy se změnou *queue_size* (velikost fronty) na hodnotu 10000 (maximální hodnota parametru je 100000) a byl ponechán parametr *events_per_second* (počet událostí za sekundu) na hodnotě 500. V této konfiguraci již také nedošlo k zahlcení fronty a nebyly pozorována zvýšená zátěž serveru jako v předchozím případě [109].

8.2 Výsledky skenování Nmap zranitelností

Test skenování zranitelností odhalil celkem 6 CVE (Common Vulnerabilities and Exposures) zranitelností, souhrnný přehled je uveden v Tabulce 1.

Následně byla provedena kontrola detekovaných zranitelností, z toho 2 zranitelnosti (CVE-2010-4816, CVE-2012-1577) se týkají operačních systémů FreeBSD a OpenBSD. Chyba CVE-2010-4816 byla chybně detekována, týká se démonu ftpd, který nebyl na skenovaném systému nainstalován [110;111;112].

Zbylé 4 CVE zranitelnosti náleží programu OpenSSH 8.7. Podle informací uvedených v uživatelském portálu firmy Red Hat, zranitelnosti (CVE-2016-20012, CVE-2023-51767) nebudou opraveny a zranitelnosti (CVE-2021-41617, CVE-2021-36368) neovlivňují distribuci Red Hat Enterprise Linux a ani distribuce z něj odvozené jako Rocky Linux [113;114;115;116].

Tabulka 1. Přehled CVE chyb detekovaných programem Nmap [110;111;112;113;114;115;116]

Software	CVE skóre	CVE	Poznámka
ftpd	7.5	CVE-2010-4816	FreeBSD
libc		CVE-2012-1577	Chyba systému OpenBSD lib/libc/stdlib/random.c Opraveno

OpenSSH 8.7	7.0	CVE-2021-41617	RHEL 9.X není ovlivněn
	5.9	CVE-2016-20012	Nebude oprava
	7.0	CVE-2023-51767	Nebude oprava
	0.0	CVE-2021-36368	RHEL 9.X není ovlivněn

8.3 Výsledky testování programem Hydra

8.3.1 Popis problému

V Kapitole 7.3.3 byl proveden test slovníkového útoku programem Hydra. Tento útok byl úspěšně identifikován detekčním systémem Wazuh, ale nedošlo k zablokování zdrojové adresy útočníka [117].

8.3.2 Návrh řešení

Program Wazuh poskytuje řešení na podobné typy útoků, a to prostřednictvím modulu *active-response*. Modul *active-response*, poskytuje možnost aktivní reakce na tento typ útoku pomocí skriptů, které jsou buď uživatelsky definované a nebo jsou již předpřipraveny v adresáři */var/ossec/active-response/bin/*. Dále tento modul nabízí i možnost zablokování uživatelského účtu [117].

8.3.3 Výsledek

Na serveru Wazuh byl nastaven modul *active-response* tak, aby zablokoval pokusy o útok hrubou silou (bruteforce) (Obrázek 65).

```
<ossec_config>
  <active-response>
    <command>firewall-drop</command>
    <location>local</location>
    <rules_id>5763</rules_id>
    <timeout>180</timeout>
  </active-response>
</ossec_config>
```

Obrázek 65. Wazuh server active-response

Následně byl zopakován útok na uživatele *testuser* programem Hydra z virtuálního serveru s distribucí Kali Linux.

```
┌──(kali㉿kali)-[~]
└─$ sudo /usr/bin/hydra  -t 4 -l testuser -P /usr/share/seclists/Passwords/2020-200_most_used_pas
swords.txt 192.168.1.50 ssh -V
Hydra v9.5 (c) 2023 by van Hauser/THC & David Maciejak - Please do not use in military or secret
service organizations, or for illegal purposes (this is non-binding, these *** ignore laws and et
hics anyway).

Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2024-05-04 14:32:02
[DATA] max 4 tasks per 1 server, overall 4 tasks, 197 login tries (l:1/p:197), ~50 tries per task
[DATA] attacking ssh://192.168.1.50:22/
[ATTEMPT] target 192.168.1.50 - login "testuser" - pass "123456" - 1 of 197 [child 0] (0/0)
[ATTEMPT] target 192.168.1.50 - login "testuser" - pass "123456789" - 2 of 197 [child 1] (0/0)
[ATTEMPT] target 192.168.1.50 - login "testuser" - pass "picture1" - 3 of 197 [child 2] (0/0)
[ATTEMPT] target 192.168.1.50 - login "testuser" - pass "password" - 4 of 197 [child 3] (0/0)
[ATTEMPT] target 192.168.1.50 - login "testuser" - pass "12345678" - 5 of 197 [child 1] (0/0)
[ATTEMPT] target 192.168.1.50 - login "testuser" - pass "111111" - 6 of 197 [child 0] (0/0)
[ATTEMPT] target 192.168.1.50 - login "testuser" - pass "123123" - 7 of 197 [child 2] (0/0)
[ATTEMPT] target 192.168.1.50 - login "testuser" - pass "12345" - 8 of 197 [child 3] (0/0)
[ATTEMPT] target 192.168.1.50 - login "testuser" - pass "1234567890" - 9 of 197 [child 1] (0/0)
[ATTEMPT] target 192.168.1.50 - login "testuser" - pass "senha" - 10 of 197 [child 3] (0/0)
[ATTEMPT] target 192.168.1.50 - login "testuser" - pass "1234567" - 11 of 197 [child 0] (0/0)
[ATTEMPT] target 192.168.1.50 - login "testuser" - pass "qwerty" - 12 of 197 [child 2] (0/0)
```

Obrázek 66. Hydra test

Výsledkem tohoto útoku (Obrázek 66) bylo úspěšné zablokování IP adresy útočníka (192.168.1.91) (Obrázek 67).

```
[root@hids-sandbox01 etc]# iptables -L -n -v
Chain INPUT (policy ACCEPT 0 packets, 0 bytes)
 pkts bytes target     prot opt in     out     source               destination
  104  9392 DROP       all  --  *      *       192.168.1.91         0.0.0.0/0

Chain FORWARD (policy ACCEPT 0 packets, 0 bytes)
 pkts bytes target     prot opt in     out     source               destination
    0     0 DROP       all  --  *      *       192.168.1.91         0.0.0.0/0

Chain OUTPUT (policy ACCEPT 0 packets, 0 bytes)
 pkts bytes target     prot opt in     out     source               destination
```

Obrázek 67. Firewall zablokovaná zdrojová adresa

Dále bylo nastaveno automatické uzamknutí účtu uživatele *testuser* po několika neúspěšných pokusech (Obrázek 68).

```
  <command>
    <name>disable-account</name>
    <executable>disable-account</executable>
    <timeout_allowed>yes</timeout_allowed>
  </command>

<ossec_config>
  <active-response>
    <command>disable-account</command>
    <location>local</location>
    <rules_id>100100</rules_id>
    <timeout>300</timeout>
  </active-response>
</ossec_config>
```

Obrázek 68. Wazuh active-response zablokování účtu

Uživatelský účet byl uzamčen, ale po 5 minutách byl odemčen (Obrázek 69).

```
[root@hids-sandbox01 etc]# sudo passwd --status testuser
testuser LK 2024-04-15 1 45 7 30 (Password locked.)
```

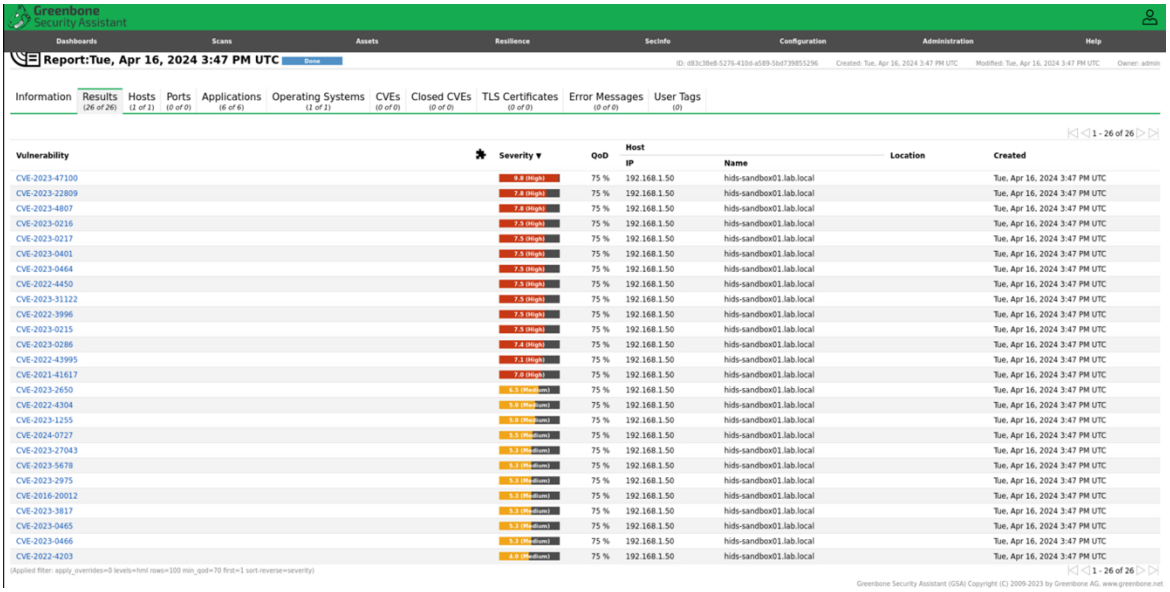
Obrázek 69. Zablokovaný uživatelský účet testuser

8.4 Výsledky skenování programem OpenVAS

8.4.1 Popis problému

Program OpenVAS byl použit na skenování zranitelností operačního systému. Tento sken odhalil několik závažných problémů s instalovanými softwarovými balíčky. Jedná se o bezpečnostní chyby definované podle CVE.

CVE definoval David E. Manne a Steven M. Christey dokumentu „Towards a Common Enumeration of Vulnerabilities" v roce 1999. Autoři v dokumentu navrhují vytvořit systém jehož cílem je poskytnout standardizovaný způsob identifikace a katalogizace zranitelností v oblasti kybernetické bezpečnosti [118].



Obrázek 70. OpenVAS CVE zranitelnosti

8.4.2 Detekované bezpečnostní chyby

Skener OpenVAS nalezl celkem 26 bezpečnostních chyb (Obrázek 70), z toho se jedná o 14 chyb s vysokou až kritickou závažností hodnocených podle CVSS (Common Vulnerability Scoring System) skóre.

CVSS je systém hodnocení zranitelností a závažností problémů. Poskytuje následující aspekty zranitelnosti, jakými jsou vektor útoku[58], složitost útoku, interakce s uživatelem, potřebná oprávnění, rozsah, důvěrnost, integrita a dostupnost. Hodnocení jednotlivých úrovní pro verzi 3 je zobrazeno v Tabulce 2. Další detaily ohledně nové verze 4 jsou k dispozici na webových stránkách organizace First [119;120].

Tabulka 2. Hodnocení úrovní CVSS v3 [120]

CVSS Základní skóre	CVSS Úroveň závažnosti
0	Žádná
0.1–3.9	Nízká
4.0–6.9	Střední
7.0–8.9	Vysoká
9.0–10.0	Kritická

Z těchto 14 chyb je jedna hodnocena kritickou závažností, tj. hodnotou 9.8, jedná se o balíček programovacího jazyku Perl[59] CVE-2023-47100 [121] (Obrázek 71). Dále bylo nalezeno 13 chyb s hodnocením 7.0 až 7.8.

Obrázek 71. OpenVAS CVE-223-47100 detail

[58] https://www.crowdstrike.com/cybersecurity-101/threat-intelligence/attack-vector/
[59] https://www.perl.org

8.4.3 Návrh řešení

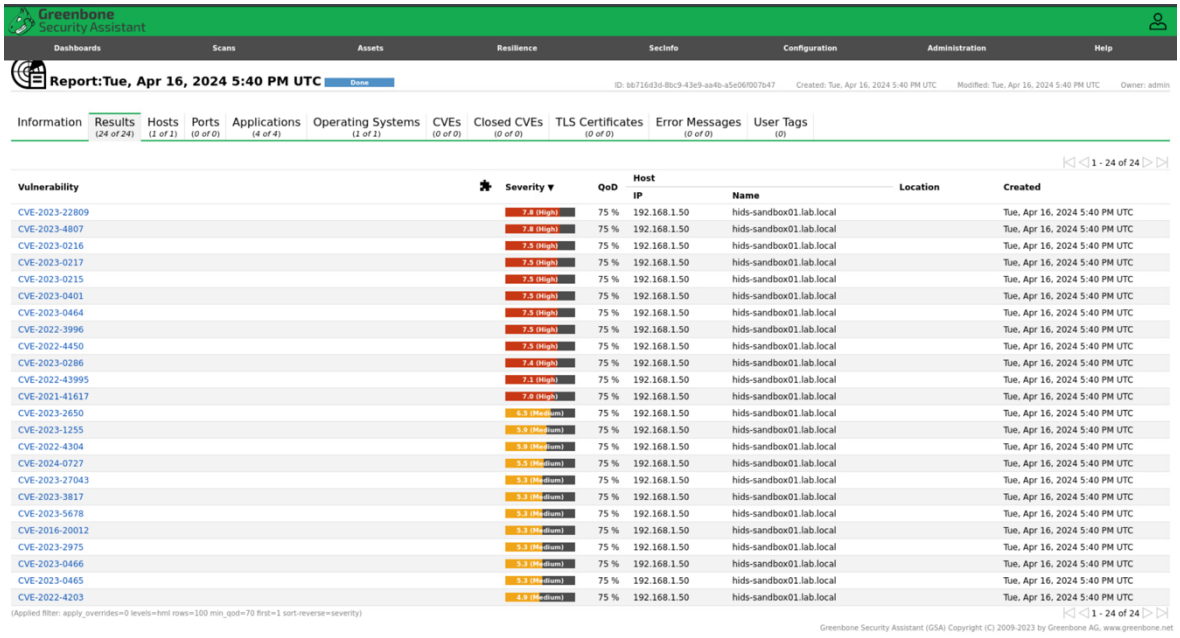
V tomto případě byla provedena aktualizace software z repozitářů distribuce na nejnovější verzi (Obrázek 72). Následně byl proveden nový sken prostřednictvím skeneru OpenVAS (Obrázek 73) a došlo k odstranění kritické chyby, a i ostatních chyb.

```
[root@hids-sandbox01 ~]# yum update
Last metadata expiration check: 3:32:13 ago on Tue 30 Apr 2024 12:18:13 PM CEST.
Dependencies resolved.
================================================================================
 Package                 Arch        Version                      Repository  Size
================================================================================
Installing:
 kernel                  x86_64      5.14.0-362.24.1.el9_3.0.1    baseos     4.6 M
Upgrading:
 NetworkManager          x86_64      1:1.44.0-5.el9_3             baseos     2.2 M
 NetworkManager-libnm    x86_64      1:1.44.0-5.el9_3             baseos     1.8 M
 NetworkManager-team     x86_64      1:1.44.0-5.el9_3             baseos      38 k
 NetworkManager-tui      x86_64      1:1.44.0-5.el9_3             baseos     243 k
```

Obrázek 72. Rocky Linux aktualizace softwarových balíčků

8.4.4 Výsledek

Kritické chyby byly odstraněny aktualizací prostřednictvím příkazu *dnf update*, který aktualizuje softwarové balíčky z veřejného nebo privátního repozitáře distribuce Rocky Linux.



Obrázek 73. OpenVAS CVE zranitelnosti po aktualizaci

Aktualizace operačního systému odstranila předchozí chyby, ale následný sken programem OpenVAS odhalil celkem 24 nových zranitelností (Obrázek 73):

- openssl 3.0.7 (19 CVE zranitelností),

- sudo (2 CVE zranitelností),
- openSSH (2 CVE zranitelností),
- python 3.9.18 (1 zranitelností).

Následně byla provedena kontrola jednotlivých zranitelností pomocí funkcionality changelog pro jednotlivé programy příkazem rpm (viz níže).

```
rpm -q PROGRAM --changelog
```

kde

`PROGRAM` jméno softwarového balíčku

`--changelog` log změn

OpenSSL

V Tabulce 3 je zobrazen přehled jednotlivých opravených a neopravených chyb. Z celkových 19 chyb, bylo opraveno 10.

Tabulka 3. OpenSSL přehled CVE chyb

Software	CVE skóre	CVE opravené	CVE
OpenSSL 3.0.7	7.8	CVE-2023-4807	
	7.5	CVE-2022-3996	
	7.5	CVE-2023-0464	
	5.9	CVE-2023-1255	
	5.5	CVE-2024-0727	
	5.3	CVE-2023-0466	
	5.3	CVE-2023-0465	
	5.3	CVE-2023-2975	
	5.3	CVE-2023-3817	
	5.3	CVE-2023-5678	
	7.5		CVE-2023-0215

	7.5		CVE-2023-0216
	7.5		CVE-2022-4450
	7.5		CVE-2023-0217
	7.5		CVE-2023-0401
	7.4		CVE-2023-0286
	6.5		CVE-2023-2650
	5.9		CVE-2022-4304
	4.9		CVE-2022-4203

Sudo

V případě programu sudo byla ze dvou chyb opravena jedna chyba (Tabulka 4).

Tabulka 4. Sudo přehled CVE chyb

Software	CVE skóre	CVE opravené	CVE neopravené
Sudo 1.9.5	7.8	CVE-2023-22809	
	7.1		CVE-2022-43995

OpenSSH

U programu OpenSSH z nalezených dvou chyb nedošlo k opravě ani jedné chyby (Tabulka 5).

Tabulka 5. OpenSSH přehled CVE chyb

Software	CVE skóre	CVE opravené	CVE neopravené
OpenSSH 8.7	7.0		CVE-2021-41617
	5.3		CVE-2016-20012

Python

U programovacího jazyku Python verze 3 byla identifikována 1 chyba, která podle logu změn byla také odstraněna (Tabulka 6).

Tabulka 6. Python přehled CVE chyb

Software	CVE skóre	CVE opravené	CVE neopravené
python 3	7.8	CVE-2023-27043	

8.4.5 Softwarové aktualizace operačního systému Rocky Linux

Operační systém Rocky Linux vychází z komerční distribuce Red Hat Enterprise Linux a kopíruje i model aktualizací této distribuce.

```
[root@hids-sandbox01 ~]# dnf updateinfo list security --installed
Last metadata expiration check: 1:48:42 ago on Wed 01 May 2024 05:44:16 PM CEST.
RLSA-2023:0340 Moderate/Sec.  bash-5.1.8-6.el9_1.x86_64
RLSA-2023:4099 Important/Sec. bind-libs-2:9.16.23-11.el9_2.1.x86_64
RLSA-2022:8068 Moderate/Sec.  bind-libs-2:9.16.23-5.el9_1.x86_64
RLSA-2023:4099 Important/Sec. bind-license-2:9.16.23-11.el9_2.1.noarch
RLSA-2022:8068 Moderate/Sec.  bind-license-2:9.16.23-5.el9_1.noarch
RLSA-2023:4099 Important/Sec. bind-utils-2:9.16.23-11.el9_2.1.x86_64
RLSA-2022:8068 Moderate/Sec.  bind-utils-2:9.16.23-5.el9_1.x86_64
RLSA-2023:0334 Important/Sec. bpftool-5.14.0-162.12.1.el9_1.0.2.x86_64
RLSA-2023:3559 Important/Sec. c-ares-1.17.1-5.el9_2.1.x86_64
RLSA-2023:0333 Moderate/Sec.  curl-7.76.1-19.el9_1.1.x86_64
RLSA-2022:8299 Low/Sec.       curl-7.76.1-19.el9_1.1.x86_64
RLSA-2023:0335 Moderate/Sec.  dbus-1:1.12.20-7.el9_1.x86_64
RLSA-2023:4569 Moderate/Sec.  dbus-1:1.12.20-7.el9_2.1.x86_64
RLSA-2023:0335 Moderate/Sec.  dbus-common-1:1.12.20-7.el9_1.noarch
RLSA-2023:4569 Moderate/Sec.  dbus-common-1:1.12.20-7.el9_2.1.noarch
RLSA-2023:0335 Moderate/Sec.  dbus-libs-1:1.12.20-7.el9_1.x86_64
RLSA-2023:4569 Moderate/Sec.  dbus-libs-1:1.12.20-7.el9_2.1.x86_64
```

Obrázek 74. Rocky Linux přehled bezpečnostních aktualizaci

Distribuce RHEL poskytuje svým uživatelům na základě předplatného (subscription) přístup k aktualizacím software tzv. Red Hat Errata. Aktualizace adresují bezpečnostní problémy, opravy softwarových chyb nebo další rozšíření funkcí programů.

Red Hat je rozděluje do následujících kategorií [122]:

- RHSA (RedHat Security Advisory) - řeší bezpečnostní problémy.
- RHBA (Red Hat Bug Advisory) - obsahuje záplaty týkající se chyb ve zdrojovém kódu programů.
- RHEA (Red Hat Enhancement Advisory) - obsahují vylepšení funkcí software.

Podobně jako distribuce RHEL i distribuce Rocky Linux, rozděluje aktualizace do následujících kategorií [123;124]:

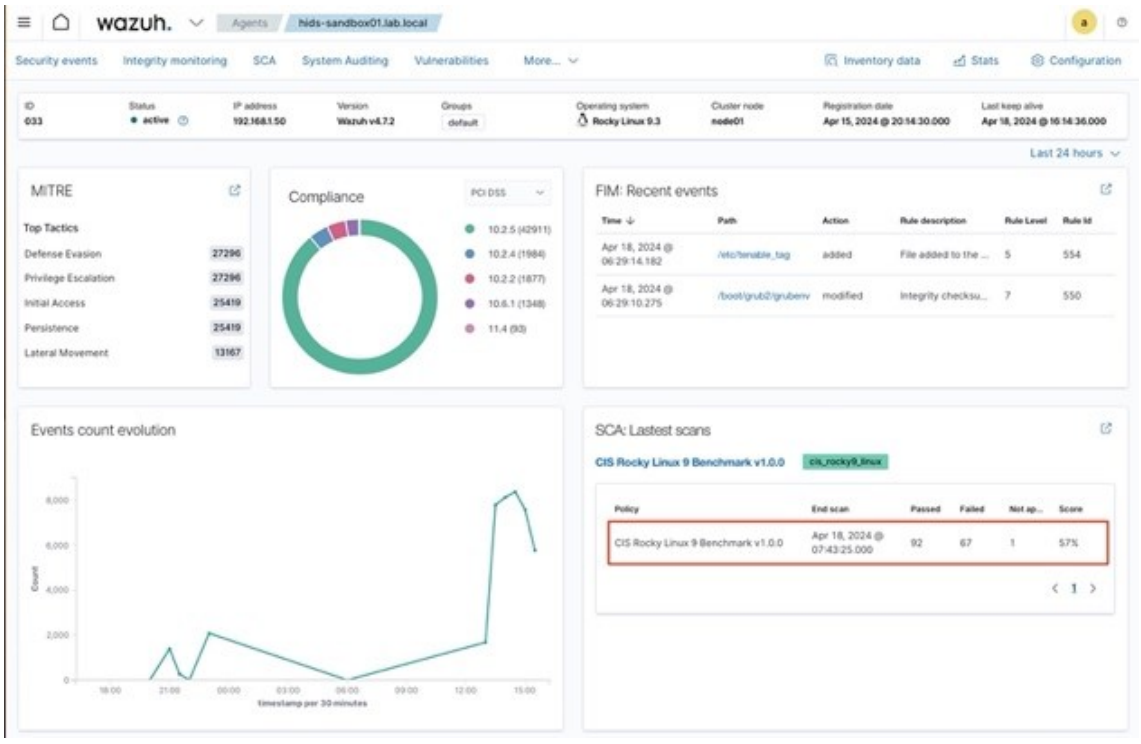
- RLSA (Rocky Linux Security Advisory),
- RLBA (Rocky Linux Bug Advisory),
- RLEA (Rocky Linux Enhancement Advisory).

8.5 Výsledky skenování SCA CIS

8.5.1 Popis problému

Během instalace operačního systému (Kapitola 5) byl operační systém nastaven podle metodologie CIS. Dále byla nastavena kontrola testování kompatibility SCA v programu Wazuh jak na straně serveru, tak klienta (Kapitola 6).

Po prvotní instalaci Wazuh identifikoval 57 % skóre vůči CIS úrovní nastavení (baseline), kde 92 testů bylo úspěšných dalších 67 testů proběhlo neúspěšně (Obrázek 75).



Obrázek 75. Wazuh SCA prvotní výsledky

8.5.2 Návrh řešení

Byla provedena dodatečné konfigurace klientského operačního systému prostřednictvím Ansible scénáře. K vytvoření scénáře byla použita Ansible role *ansible-role-rhel9-cis*,

vytvořená firmou Red Hat a dostupná z jejich oficiálního repozitáře *RedHatOfficials*, hostovaného na verzovacím systému Github [125].

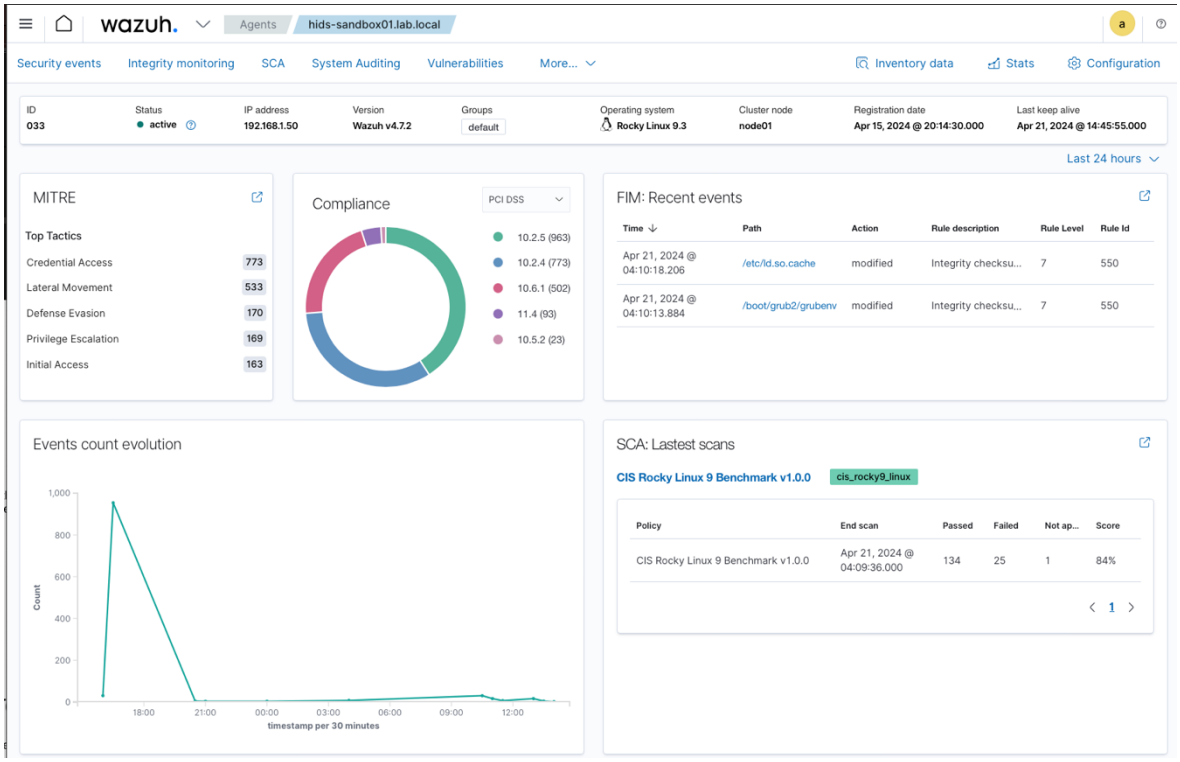
8.5.3 Výsledek

Jak je vidět z Obrázku 76, bylo úspěšně provedeno 2263 úkolů (tasks) a došlo ke změnám nastavení (changed=279) podle základní úrovni CIS (Center for Internet Security).

```
→  hids ansible-playbook -i inventories/dev/hosts playbook.yml -u testuser -K | tee cis_baseline.log
BECOME password:

PLAY [sandbox01] ***************************************************************

TASK [Gathering Facts] *********************************************************
ok: [192.168.1.50]

TASK [RedHatOfficial.rhel9_cis : Ensure aide is installed] *********************
changed: [192.168.1.50]

TASK [RedHatOfficial.rhel9_cis : Ensure AIDE is installed] *********************
ok: [192.168.1.50] => (item=aide)

TASK [RedHatOfficial.rhel9_cis : Build and Test AIDE Database] *****************
changed: [192.168.1.50]

TASK [RedHatOfficial.rhel9_cis : Check whether the stock AIDE Database exists] ***
ok: [192.168.1.50]


TASK [RedHatOfficial.rhel9_cis : Deduplicate values from /etc/ssh/sshd_config] ***
skipping: [192.168.1.50]

TASK [RedHatOfficial.rhel9_cis : Insert correct line to /etc/ssh/sshd_config] ***
changed: [192.168.1.50]

TASK [RedHatOfficial.rhel9_cis : Ensure xorg-x11-server-common is removed] *****
ok: [192.168.1.50]

PLAY RECAP *********************************************************************
192.168.1.50               : ok=2263 changed=279  unreachable=0    failed=0    skipped=820  rescued=0
ignored=2
```

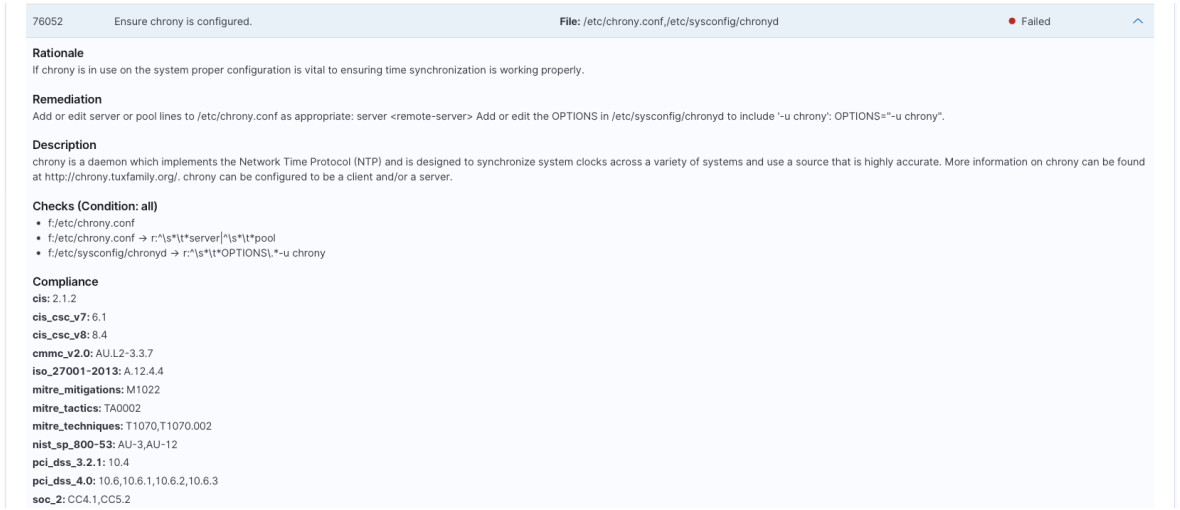
Obrázek 76. Ansible výsledek nastavení systému (zkrácená verze)

Program Wazuh provedl opětovnou kontrolu nastavení oproti CIS prostřednictvím modulu SCA a došlo ke zlepšení výsledného skóre z 51 % na 84 %. Z toho134 testů bylo úspěšných a 25 bylo neúspěšných (Obrázek 77). Je nutné dále věnovat pozornost zbývajícím 25 testům, CIS metodologie poskytuje informace o možné nápravě (Obrázek 78). Zde je nutné projít jednotlivé testy a vyhodnotit, zda akceptujeme tuto chybu nebo zjednáme nápravu.



Obrázek 77. Wazuh SCA výsledky po aktualizaci

Program Wazuh přehledně zobrazí stav jednotlivých testů a v případě chybného výsledku i možnost jeho nápravy (Obrázek 78).



Obrázek 78. Wazuh SCA detail chybného testu

8.6 Výsledky HID testů

8.6.1 Popis problému

Při vykonání testů simulujících pokusy o fyzické proniknutí do operačního systému došlo k odhalení bezpečnostního problému, kterým byla možnost restartu systému stisknutím kláves CTRL + ALT + DEL (Kapitola 7.5.1).

8.6.2 Návrh řešení

Způsobený problém lze odstranit vypnutím služby *ctrl-alt-del.target*, která je symbolickým odkazem na službu *reboot.target*. Příkazy na Obrázku 79 tuto službu vypnou a vytvoří symbolický odkaz na soubor /dev/null[60] [126].

```
[root@hids-sandbox01 ~]# systemctl disable ctrl-alt-del.target
Removed "/etc/systemd/system/ctrl-alt-del.target".
[root@hids-sandbox01 ~]# systemctl mask  ctrl-alt-del.target
Created symlink /etc/systemd/system/ctrl-alt-del.target → /dev/null.
```

Obrázek 79. Systemd zablokování služby ctrl-alt-del.target

8.6.3 Výsledek

Po implementaci této změny byl zopakován test prostřednictvím Ansible scénáře *vmware_vm_HID_CTRL_ALT_DEL.yml* (Obrázek 80) a už nebylo možné provést restart virtuálního serveru použitím kláves CTRL + ALT + DEL. Výsledek testu proběhl s očekávaným výsledkem.

[60] https://www.man7.org/linux/man-pages/man4/null.4.html

```
→  hids ansible-playbook -i inventories/dev/hosts playbooks/hid/vmware_vm_HID_CTRL_ALT_DEL.yml

PLAY [localhost] ***************************************************************************************

TASK [Send CTRL + ALT + DEL] ***************************************************************************
fatal: [localhost]: FAILED! => changed=false
  msg: |-
    Failed to send key (vim.vm.UsbScanCodeSpec.KeyEvent) {
       dynamicType = <unset>,
       dynamicProperty = (vmodl.DynamicProperty) [],
       usbHidCode = 4980743,
       modifiers = (vim.vm.UsbScanCodeSpec.ModifierType) {
          dynamicType = <unset>,
          dynamicProperty = (vmodl.DynamicProperty) [],
          leftControl = true,
          leftShift = false,
          leftAlt = true,
          leftGui = false,
          rightControl = false,
          rightShift = false,
          rightAlt = false,
          rightGui = false
       }
    } to virtual machine due to vim.fault.InvalidPowerState

PLAY RECAP *********************************************************************************************
localhost                  : ok=0    changed=0    unreachable=0    failed=1    skipped=0    rescued=0
ignored=0
```

Obrázek 80. Test restart serveru

Pro případ zabezpečení USB portu proti použití fyzického hardware (USB klíč) je vhodné nasadit program usbguard. Program umožnuje selektivně blokovat připojené zařízení do USB portů, kterými mohou být škodlivé zařízení (USB rouge device) jako např. Rubber Ducky (Obrázek 60) [127].

ZÁVĚR

Systémy detekce průniku (IDS) jsou důležitou součástí detekčního mechanismu zabezpečení počítačové infrastruktury. V dnešní době s dlouhodobým nárůstem škodlivého software se jedná již o nezbytnou součást ochrany v rámci IT infrastruktury, která by měla odpovídat bezpečnostním standardům např. ISO 27001. Klientské detekční systémy (HIDS) monitorují neobvyklé aktivity pouze na koncových stanicích, kterými jsou servery nebo pracovní stanice. Ale již nesledují podezřelé aktivity v rámci síťové infrastruktury. Pro tyto účely je vždy nutné se zabývat komplexně problematikou detekce průniků prostřednictvím dalších komponentů (NIDS), které jsou umístěny v dané infrastruktuře.

Pro účel této práce byl vybrán vhodný hardware a virtualizační prostření, na kterém byla vytvořená virtuální testovací serverová infrastruktura. Jako operační systém byla zvolena volně šiřitelná distribuce operačního systému Linux – Rocky Linux 9, která vychází z komerční distribuce Red Hat Enterprise Linux 9. Konfigurační nástroj Ansible byl použit pro vytvoření virtuálního serveru a automatizace jeho instalace prostřednictvím konfiguračního předpisu kickstart. Byla věnována pozornost zabezpečení serveru již při vytváření a během instalace a konfigurace operačního systému podle metodiky CIS, který je podporován prostřednictvím OpenSCAP profilů.

Následně byla navržena pravidla detekce průniku a byl vybrán vhodný detekční systém. Během tohoto procesu byly porovnány známé i méně známé klientské detekční systémy pro linuxové operační systémy a srovnány jejich základní parametry. Na základě vyhodnocení všech parametrů byl zvolen volně šiřitelný detekční systém Wazuh, který poskytuje nejvíc funkcí oproti ostatním systémům. Serverová část sytému Wazuh byla nainstalována na samostatný server a klientská část na server s linuxovou distribucí Rocky Linux. Oba systémy byly nastaveny tak, aby detekovaly možné průniky podle definovaných pravidel detekce.

Klientský systém byl otestován penetračními testy, které byly provedeny vzdáleně z linuxové distribuce Kali Linux. Testy obsahovaly skenování portů, zranitelností systému, pokus o útok hrubou silou.

Testy dále obsahovaly lokální test na testování možnosti zvýšení uživatelských práv a testy simulující pokusy o průnik z HID (Human Interface Device) zařízení.

Na závěr byly vyhodnoceny nalezené problémy během testování, byla navržena a otestovaná řešení, která tyto problémy odstranila.

SEZNAM POUŽITÉ LITERATURY

[1] YOST, Jeffrey R. The March of IDES: Early History of Intrusion-Detection Expert Systems. Online. *IEEE Annals of the History of Computing*. 2016, roč. 38, č. 4, s. 42-54. ISSN 1058-6180. Dostupné z: https://doi.org/10.1109/MAHC.2015.41. [cit. 2024-01-21].

[2] SHIREY, Robert W. *Internet Security Glossary, Version 2*. RFC Editor, 2007. Dostupné z: https://doi.org/10.17487/RFC4949.

[3] ANDERSON, James P. COMPUTER SECURITY TECHNOLOGY PLANNING STUDY. *DEPUTY FOR COMMAND AND MANAGEMENT SYSTEMS HQ ELECTRONIC SYSTEMS DIVISION (AFSC)* [online]. Massachusetts, 1972, (ESD-TR-73-51), 33 [cit. 2023-03-30]. Dostupné z: https://csrc.nist.gov/csrc/media/publications/conference-paper/1998/10/08/proceedings-of-the-21st-nissc-1998/documents/early-cs-papers/ande72a.pdf

[4] ANDERSON, James P. *Computer Security Threat Monitoring and Surveillance* [online]. Washington, 1980, 53 [cit. 2023-03-30]. Dostupné z: https://seclab.cs.ucdavis.edu/projects/history/papers/ande80.pdf

[5] *Doporučení pro případ napadení DDoS útokem – jak se zachovat a jak postupovat*. Online. NÚKIB. 2013. Dostupné z: https://nukib.gov.cz/cs/infoservis/doporuceni/1452-doporuceni-pro-pripad-napadeni-ddos-utokem-jak-se-zachovat-a-jak-postupovat/. [cit. 2024-05-02].

[6] REQUIREMENTS AND MODEL FOR IDES -A REAL-TIME INTRUSION-DETECTIONEXPERT SYSTEM August 1985 Dorothy Denning Peter G. Neumann , Computer Science Laboratory, Contract No. 83F83-01-00,SRI Project 6169-10] [Intrusion Detection

[7] DENNING, D.E. An Intrusion-Detection Model. Online. *IEEE Transactions on Software Engineering*. 1987, roč. SE-13, č. 2, s. 222-232. ISSN 0098-5589. Dostupné z: https://doi.org/10.1109/TSE.1987.232894. [cit. 2024-01-21].

[8] SEBRING, Michael M; SHELLHOUSE, Eric; HANNA, Mary E a WHITEHURST, R Alan. Expert systems in intrusion detection: A case study: A case study. In: *Proceedings of the 11th National Computer Security Conference*. Baltimore,

Maryland: 1988. National Institute of Standards and Technology / National Computer Security Center, 1988, s. 74-81.

[9] [GÓRECKI, Jan. *Expertní systémy*. Online. Karviná: Slezská univerzita v Opavě, 2017. Dostupné z: https://is.slu.cz/el/opf/leto2021/INMNKESY/um/gorecki2017_expertni_systemy.pdf. [cit. 2024-05-02].

[10] LINDQVIST, U. a PORRAS, P.A. Detecting computer and network misuse through the production-based expert system toolset (P-BEST). Online. In: *Proceedings of the 1999 IEEE Symposium on Security and Privacy (Cat. No.99CB36344)*. IEEE Comput. Soc, 1999, s. 146-161. ISBN 0-7695-0176-1. Dostupné z: https://doi.org/10.1109/SECPRI.1999.766911. [cit. 2024-01-21].

[11] REDLE, K. *Wisdom and Sense*. United States, 1992. Dostupné také z: https://www.osti.gov/biblio/1230146.

[12] GHORBANI, Ali A.; LU, Wei; TAVALLAEE, Mahbod; GHORBANI, Ali A.; LU, Wei et al. Detection Approaches. Online. In: *Network Intrusion Detection and Prevention*. Advances in Information Security. Boston, MA: Springer US, 2010, s. 27-53. ISBN 978-0-387-88770-8. Dostupné z: https://doi.org/10.1007/978-0-387-88771-5_2. [cit. 2024-01-21].

[13] SMAHA, S.E. Haystack: an intrusion detection system. Online. In: *[Proceedings 1988] Fourth Aerospace Computer Security Applications*. IEEE Comput. Soc. Press, 1988, s. 37-44. ISBN 0-8186-0895-1. Dostupné z: https://doi.org/10.1109/ACSAC.1988.113412. [cit. 2024-01-21].

[14] ANDERSON, Debra; FRIVOLD, Thane a VALDES, Alfonso. Next-generation Intrusion Detection Expert System (NIDES). Online. In: *Laboratory SRI-CSL-95-07*. SRI International, May 1995. Dostupné z: https://www.csl.sri.com/papers/4sri/4sri.pdf. [cit. 2024-01-21].

[15] *What is NIDES?* Online. SRI International. C20054. Dostupné z: https://www.csl.sri.com/projects/nides/whatisnides.html. [cit. 2024-01-21].

[16] KIM, Gene a SPAFFORD, Eugene. Experiences with Tripwire: Using Integrity Checkers for Intrusion Detection: Using Integrity Checkers for Intrusion Detection. 1995/03/22.

[17] KIM, Gene H a SPAFFORD, Eugene H. *The design and implementation of tripwire: a file system integrity checker: a file system integrity checker*. 1994. Dostupné také z: https://api.semanticscholar.org/CorpusID:5027061.

[18] *Election Security Spotlight – Defense in Depth (DiD)*. Online. CIS: Center for Internet Security. C2024. Dostupné z: https://www.cisecurity.org/insights/spotlight/cybersecurity-spotlight-defense-in-depth-did. [cit. 2024-05-02].

[19] GLASS-VANDERLAN, Tarrah R; IANNACONE, Michael D; VINCENT, Maria S; QIAN,; CHEN, et al. *A Survey of Intrusion Detection Systems Leveraging Host Data*. 2018. Dostupné také z: http://arxiv.org/abs/1805.06070.

[20] LAZAREVIC, Aleksandar; KUMAR, Vipin a SRIVASTAVA, Jaideep. Intrusion Detection: A Survey: A Survey. In: *Managing Cyber Threats: Issues, Approaches, and Challenges*. Boston, MA: Springer US, 2005, s. 19-78. ISBN 978-0-387-24230-9. Dostupné z: https://doi.org/10.1007/0-387-24230-9_2.

[21] KHRAISAT, Ansam; GONDAL, Iqbal; VAMPLEW, Peter a KAMRUZZAMAN, Joarder. Survey of intrusion detection systems: techniques, datasets and challenges. Online. *Cybersecurity*. 2019, roč. 2, č. 1, s. 4. ISSN 2523-3246. Dostupné z: https://doi.org/10.1186/s42400-019-0038-7. [cit. 2024-01-24].

[22] VAN OORSCHOT, Paul C. *Computer security and the internet: Tools and jewels from malware to bitcoin*. Second. Springer, 2021. ISBN 9783030834104. Dostupné z: https://doi.org/10.1007/978-3-030-83411-1.

[23] GHORBANI, Ali A.; LU, Wei a TAVALLAEE, Mahbod. *Network Intrusion Detection and Prevention*. Online. Advances in Information Security. Boston, MA: Springer US, 2010. ISBN 978-0-387-88770-8. Dostupné z: https://doi.org/10.1007/978-0-387-88771-5. [cit. 2024-02-08].

[24] STALLINGS, William a BROWN, Lawrie. *Computer Security: Principles and Practice: Principles and Practice*. 3rd. USA: Prentice Hall Press, 2014. ISBN 9780133773927.

[25] *Co jsou indikátory ohrožení (IOC)?* Online. Microsoft. C2024, https://www.microsoft.com/. Dostupné z: https://www.microsoft.com/cs-cz/security/business/security-101/what-are-indicators-of-compromise-ioc. [cit. 2024-05-04].

[26] PATHAN, Al-Sakib Khan. *The State of the Art in Intrusion Prevention and Detection*. Auerbach Publications, 2014. ISBN 978-1482203516.]

[27] SWARNKAR, Mayank a RAJPUT, Shyam Singh. *Artificial intelligence for intrusion detection systems*. Boca Raton, FL: CRC Press, 2024. ISBN 978-1032386652.

[28] *Akademický slovník současné češtiny*. Online. 2017 - 2024. Dostupné z: https://slovnikcestiny.cz. [cit. 2024-05-02].

[29] AGRAWAL, Shikha a AGRAWAL, Jitendra. Survey on Anomaly Detection using Data Mining Techniques. Online. *Procedia Computer Science*. 2015, roč. 60, s. 708-713. ISSN 18770509. Dostupné z: https://doi.org/10.1016/j.procs.2015.08.220. [cit. 2024-02-01].

[30] GARCÍA-TEODORO, P.; DÍAZ-VERDEJO, J.; MACIÁ-FERNÁNDEZ, G. a VÁZQUEZ, E. Anomaly-based network intrusion detection: Techniques, systems and challenges. Online. *Computers & Security*. 2009, roč. 28, č. 1-2, s. 18-28. ISSN 01674048. Dostupné z: https://doi.org/10.1016/j.cose.2008.08.003. [cit. 2024-02-03].

[31] DIOGENES, Yuri a OZKAYA, Erdal. Cybersecurity – Attack and Defense Strategies: Improve your security posture to mitigate risks and prevent attackers from infiltrating your system. 3rd ed. Packt Publishing, c2022. ISBN 978-1803248776.

[32] SPITZNER, Lance. *Honeypots tracking hackers*. Boston: Addison-Wesley, 2003. ISBN 0-321-10895-7.

[33] JIRÁSEK, Petr; NOVÁK, Luděk a POŽÁR, Josef. *Výkladový slovník kybernetické bezpečnosti: Cyber security glossary*. Páté doplněné a upravené vydání. Přeložil Karel VAVRUŠKA. Praha: Česká pobočka AFCEA, 2022. ISBN 978-80-908388-4-0.

[34] SANDERS, Chris. *Intrusion detection honeypots: detection through deception*. Oakwood: Chris Snaders, [2020]. ISBN 978-1-7351883-0-0.

[35] HAND, Matt. *Evading EDR: the definitive guide to defeating endpoint detection systems: the definitive guide to defeating endpoint detection systems*. San Francisco: No Starch Press, 2024. ISBN 978-1718503342.

[36] LAUBER, Susan. *An introduction to Pluggable Authentication Modules (PAM) in Linux*. Online. RedHat. 2020. Dostupné

z: https://www.redhat.com/sysadmin/pluggable-authentication-modules-pam. [cit. 2024-05-02].

[37] *Authentication with PAM*. Online. SUSE product documentation. C2024. Dostupné z: https://documentation.suse.com/en-us/sles/15-SP2/html/SLES-all/cha-pam.html#sec-security-pam-whatis. [cit. 2024-05-02].

[38] *3.1. General overview of the Linux file system*. Online. GARRELS, Machtelt. Introduction to Linux: A Hands on Guide. C2008. Dostupné z: https://tldp.org/LDP/intro-linux/html/sect_03_01.html. [cit. 2024-05-04].

[39] *SAMHAIN: THE SAMHAIN FILE INTEGRITY / HOST-BASED INTRUSION DETECTION SYSTEM*. Online. SAMHAIN LABS. Dostupné z: https://www.la-samhna.de/samhain/. [cit. 2023-03-30].

[40] WICHMANN, Rainer. *The Samhain HIDS: Overview of available features*. Online. SAMHAIN LABS. 2011. Dostupné z: https://www.la-samhna.de/samhain/samhain_leaf.pdf. [cit. 2023-03-30].

[41] *AIDE*. Online. Dostupné z: https://aide.github.io. [cit. 2024-05-02].

[42] *What is Sagan?* Online. Sagan User Guide. C2018. Dostupné z: https://sagan.readthedocs.io/en/latest/what-is-sagan.html. [cit. 2024-05-02].

[43] *Sagan*. Online. GitHub. C2024. Dostupné z: https://github.com/quadrantsec/sagan/. [cit. 2024-05-02].

[44] *About OSSEC HIDS: Host Intrusion Detection for Everyone*. Online. OSSEC. C2024. Dostupné z: https://www.ossec.net/about/. [cit. 2024-05-02].

[45] *OSSEC Architecture: Host Intrusion Detection for Everyone*. Online. OSSEC. C2010 - 2021. Dostupné z: https://www.ossec.net/docs/docs/manual/ossec-architecture.html. [cit. 2024-05-02].

[46] *Migrating from OSSEC*. Online. Wazuh. C2024. Dostupné z: https://documentation.wazuh.com/current/migration-guide/migrating-from-ossec/index.html. [cit. 2024-05-02].

[47] *Components*. Online. Wazuh. C2024. Dostupné z: https://documentation.wazuh.com/current/getting-started/components/index.html. [cit. 2024-05-02].

[48] *Use cases*. Online. Wazuh. C2024. Dostupné z: https://wazuh.com/platform/overview/. [cit. 2024-05-02].

[49] *Use cases*. Online. Wazuh. C2024. Dostupné z: https://documentation.wazuh.com/current/getting-started/use-cases/index.html. [cit. 2024-05-02].

[50] *Chapter 7. System Auditing*. Online. Red Hat Customer Portal. 2024. Dostupné z: https://access.redhat.com/documentation/en-us/red_hat_enterprise_linux/6/html/security_guide/chap-system_auditing. [cit. 2024-05-02].

[51] *Chapter 12. Auditing the system*. Online. Red Hat Customer Portal. 2024. Dostupné z: https://access.redhat.com/documentation/en-us/red_hat_enterprise_linux/9/html/security_hardening/auditing-the-system_security-hardening. [cit. 2024-05-02].

[52] *Audit-userspace*. Online. GitHub. C2024. Dostupné z: https://github.com/linux-audit/audit-userspace. [cit. 2024-05-02].

[53] *AUREPORT*. Online. Debian Manpages. 2024. Dostupné z: https://manpages.debian.org/testing/auditd/aureport.8.en.html. [cit. 2024-05-02].

[54] *Fail2ban*. Online. GitHub. C2024. Dostupné z: https://github.com/fail2ban/fail2ban. [cit. 2024-05-02].

[55] *CrowdSec Plans & Pricing*. Online. CrowdSec. C2023. Dostupné z: https://www.crowdsec.net/pricing. [cit. 2024-05-02].

[56] *CrowdSec Security Engine*. Online. CrowdSec. C2023. Dostupné z: https://www.crowdsec.net/product/crowdsec-security-engine. [cit. 2024-05-02]

[57] *Introduction*. Online. CrowdSec. C2024. Dostupné z: https://docs.crowdsec.net/docs/intro/. [cit. 2024-05-02].

[58] *PowerEdge T620: Technical Guide*. Online. Dell Technologies. C2013. Dostupné z: https://i.dell.com/sites/doccontent/shared-content/data-sheets/en/Documents/dell-poweredge-t620-technical-guide.pdf. [cit. 2024-05-02].

[59] *VMUG: VMware User Group*. Online. VMUG. C2024. Dostupné z: https://www.vmug.com. [cit. 2024-05-02].

[60] *Ansible*. Online. C2024. Dostupné z: https://www.ansible.com. [cit. 2024-05-05].

[61] *Kali Linux Features: What is Kali Linux, and what is a Penetration Testing Distribution?* Online. KALI. C2024. Dostupné z: https://www.kali.org/features/. [cit. 2024-05-02].

[62] ROCKY LINUX. *Licensing*. Online. Dostupné z: https://rockylinux.org/legal/licensing. [cit. 2024-05-05].

[63] *SecureBoot*. Online. Debian. 2024. Dostupné z: https://wiki.debian.org/SecureBoot. [cit. 2024-05-05].

[64] *What is UEFI Secure Boot and how it works?* Online. Red Hat Customer Portal. C2024. Dostupné z: https://access.redhat.com/articles/5254641. [cit. 2024-05-05].

[65] *Security Content Automation Protocol SCAP*. Online. NIST. Information Technology Laboratory: COMPUTER SECURITY RESOURCE CENTER. 2023. Dostupné z: https://csrc.nist.gov/projects/security-content-automation-protocol/]. [cit. 2024-05-02].

[66] *What are CIS benchmarks?* Online. IBM. Dostupné z: https://www.ibm.com/topics/cis-benchmarks. [cit. 2024-05-02].

[67] *Red Hat Enterprise Linux*. Online. CIS: Center for Internet Security. C2024. Dostupné z: https://www.cisecurity.org/benchmark/red_hat_linux. [cit. 2024-05-02].

[68] *About us*. Online. CIS: Center for Internet Security. C2024. Dostupné z: https://www.cisecurity.org/about-us. [cit. 2024-05-02].

[69] *Profiles in ComplianceAsCode*. Online. COMPLIANCEASCODE BLOG. C2024. Dostupné z: https://complianceascode.github.io/content-pages/guides/index.html. [cit. 2024-05-02].

[70] *What is SELinux?* Online. Red Hat. 2019. Dostupné z: https://www.redhat.com/en/topics/linux/what-is-selinux. [cit. 2024-05-02].

[71] *Czech Republic — cz.pool.ntp.org*. Online. NTP Pool Project. Dostupné z: https://www.ntppool.org/zone/cz. [cit. 2024-05-05].

[72] *Chapter 18. Configuring NTP Using the chrony Suite*. Online. Red Hat Customer Portal. C2024. Dostupné z: https://access.redhat.com/documentation/en-us/red_hat_enterprise_linux/7/html/system_administrators_guide/ch-configuring_ntp_using_the_chrony_suite. [cit. 2024-05-05].

[73] *Ansible-role-rhel9-cis*. Online. GitHub. C2024. Dostupné z: https://github.com/RedHatOfficial/ansible-role-rhel9-cis. [cit. 2024-05-02].

[74] *Architecture*. Online. Wazuh. C2024. Dostupné z: https://documentation.wazuh.com/current/getting-started/architecture.html. [cit. 2024-05-02].]

[75] *Configuration assessment*. Online. Wazuh. C2024. Dostupné z: https://documentation.wazuh.com/current/getting-started/use-cases/configuration-assessment.html. [cit. 2024-05-02].

[76] *Vulnerability detection*. Online. Wazuh. C2024. Dostupné z: https://documentation.wazuh.com/current/user-manual/capabilities/vulnerability-detection/index.html. [cit. 2024-05-02].

[77] *Malware detection*. Online. Wazuh. C2024. Dostupné z: https://documentation.wazuh.com/current/user-manual/capabilities/malware-detection/index.html. [cit. 2024-05-02].

[78] *File integrity monitoring*. Online. Wazuh. C2024. Dostupné z: https://documentation.wazuh.com/current/user-manual/capabilities/file-integrity/index.html. [cit. 2024-05-02].

[79] *Deploying Wazuh agents on Linux endpoints*. Online. Wazuh. C2024. Dostupné z: https://documentation.wazuh.com/current/installation-guide/wazuh-agent/wazuh-agent-package-linux.html. [cit. 2024-05-02].

[80] *Client*. Online. Wazuh. C2024. Dostupné z: https://documentation.wazuh.com/current/user-manual/reference/ossec-conf/client.html. [cit. 2024-05-02].

[81] *How it works*. Online. Wazuh. C2024. Dostupné z: https://documentation.wazuh.com/current/user-manual/capabilities/policy-monitoring/openscap/how-it-works.html. [cit. 2024-05-02].

[82] *Log data collection*. Online. Wazuh. C2024. Dostupné z: https://documentation.wazuh.com/current/user-manual/capabilities/log-data-collection/index.html. [cit. 2024-05-02].

[83] *Configuring and running scans*. Online. Wazuh. C2024. Dostupné z: https://documentation.wazuh.com/current/user-manual/capabilities/vulnerability-detection/configuring-scans.html. [cit. 2024-05-02].

[84] *Using Syscollector information to trigger alerts*. Online. Wazuh. C2024. Dostupné z: https://documentation.wazuh.com/current/user-manual/capabilities/system-inventory/using-syscollector-information-to-trigger-alerts.html. [cit. 2024-05-02].

[85] *Enterprise tactics*. Online. Mitre ATT&CK®. C2015 - 2024. Dostupné z: https://attack.mitre.org/tactics/enterprise/. [cit. 2024-05-02].

[86] *Reconnaissance*. Online. Mitre ATT&CK®. C2015 - 2024. Dostupné z: https://attack.mitre.org/tactics/TA0043/. [cit. 2024-05-02].

[87] *Discovery*. Online. Mitre ATT&CK®. C2015 - 2024. Dostupné z: https://attack.mitre.org/tactics/TA0007/. [cit. 2024-05-02].

[88] *Privilege Escalation*. Online. Mitre ATT&CK®. C2015 - 2024. Dostupné z: https://attack.mitre.org/tactics/TA0004/. [cit. 2024-05-02].

[89] *Persistence*. Online. Mitre ATT&CK®. C2015 - 2024. Dostupné z: https://attack.mitre.org/tactics/TA0003/. [cit. 2024-05-02].

[90] *Indicator Removal on Host*. Online. Mitre ATT&CK®. C2015 - 2024. Dostupné z: https://attack.mitre.org/techniques/T0872/. [cit. 2024-05-02].

[91] *Background*. Online. Greenbone Community Edition – Documentation. C2021 - 2024. Dostupné z: https://greenbone.github.io/docs/latest/background.html. [cit. 2024-05-02].

[92] *Kali Linux Install Guide*. Online. Greenbone Community Edition – Documentation. C2021 - 2024. Dostupné z: https://greenbone.github.io/docs/latest/22.4/kali/index.html. [cit. 2024-05-02].

[93] *The History and Future of Nmap*. Online. nmap.org. Dostupné z: https://nmap.org/book/history-future.html. [cit. 2024-05-02].

[94] *Nmap Scripting Engine (NSE)*. Online. nmap.org. Dostupné z: https://nmap.org/book/man-nse.html. [cit. 2024-05-02].

[95] LYON, Gordon "Fyodor." *Nmap Network Scanning: The Official Nmap Project Guide to Network Discovery and Security Scanning*. Online. Nmap.org. C2008 - 2022. Dostupné z: https://nmap.org/book/toc.html. [cit. 2024-05-05].

[96] *Script vulners*. Online. NMAP.ORG. Dostupné z: https://nmap.org/nsedoc/scripts/vulners.html. [cit. 2024-05-02].].

[97] *Tool Documentation: hydra Usage Example*. Online. KALI. C2024. Dostupné z: https://www.kali.org/tools/hydra/. [cit. 2024-05-02].

[98] *Tool Documentation: SecLists Usage Examples*. Online. KALI. C2024. Dostupné z: https://www.kali.org/tools/seclists/. [cit. 2024-05-02].

[99] *Thc-hydra*. Online. GitHub. C2024. Dostupné z: https://github.com/vanhauser-thc/thc-hydra. [cit. 2024-05-02].

[100] *Brute Force: Password Guessing*. Online. Mitre ATT&CK®. C2015 - 2024. Dostupné z: https://attack.mitre.org/techniques/T1110/001/. [cit. 2024-05-02].

[101] *Remote Services: SSH*. Online. Mitre ATT&CK®. C2015 - 2024. Dostupné z: https://attack.mitre.org/techniques/T1021/004/. [cit. 2024-05-02

[102] *Lateral Movement*. Online. Mitre ATT&CK®. C2015 - 2024. Dostupné z: https://attack.mitre.org/tactics/TA0008. [cit. 2024-05-02].

[103] *USB RUBBER DUCKY*. Online. HAK5. Dostupné z: https://shop.hak5.org/products/usb-rubber-ducky. [cit. 2024-05-03].

[104] *Community.vmware.vmware_guest_sendkey module: Send USB HID codes to the Virtual Machine's keyboard*. Online. Ansible Community Documentation. C2024. Dostupné z: https://docs.ansible.com/ansible/latest/collections/community/vmware/vmware_guest_sendkey_module.html. [cit. 2024-05-05].

[105] *Impair Defenses: Disable or Modify Tools*. Online. Mitre ATT&CK®. C2015 - 2024. Dostupné z: https://attack.mitre.org/techniques/T1562/001/. [cit. 2024-05-02].

[106] *Privilege Escalation*. Online. Mitre ATT&CK®. C2015 - 2024. Dostupné z: https://attack.mitre.org/tactics/TA0004/. [cit. 2024-05-02].

[107] *PEASS-ng*. Online. GitHub. C2024. Dostupné z: https://github.com/peass-ng/PEASS-ng. [cit. 2024-05-02].

[108] *Anti-flooding mechanism*. Online. Wazuh. C2024. Dostupné z: https://documentation.wazuh.com/current/user-manual/agent/agent-management/antiflooding.html. [cit. 2024-05-02].

[109] *Client_buffer*. Online. Wazuh. C2024. Dostupné z: https://documentation.wazuh.com/current/user-manual/reference/ossec-conf/client-buffer.html. [cit. 2024-05-02].

[110] CVE-2010-4816. Online. CVE. 2023. Dostupné z: *https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2010-4816*. [cit. 2024-05-03].

[111] *CVE-2012-1577*. Online. CVE. 2023. Dostupné z: https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2012-1577. [cit. 2024-05-03].

[112] *CVE-2012-1577*. Online. Https://security-tracker.debian.org. Dostupné z: https://security-tracker.debian.org/tracker/CVE-2012-1577. [cit. 2024-05-03].

[113] *CVE-2021-41617*. Online. Red Hat Customer Portal. C2021, 2024. Dostupné z: https://access.redhat.com/security/cve/cve-2021-41617. [cit. 2024-05-03].

[114] *CVE-2016-20012*. Online. Red Hat Customer Portal. C2021, 2024. Dostupné z: https://access.redhat.com/security/cve/cve-2016-20012. [cit. 2024-05-03].

[115] *CVE-2023-51767*. Online. Red Hat Customer Portal. C2021, 2024. Dostupné z: https://access.redhat.com/security/cve/cve-2023-51767. [cit. 2024-05-03].

[116] *CVE-2021-36368*. Online. Red Hat Customer Portal. C2021, 2024. Dostupné z: https://access.redhat.com/security/cve/cve-2021-36368. [cit. 2024-05-03].

[117] *Active response*. Online. Mitre ATT&CK®. C2015 - 2024. Dostupné z: https://documentation.wazuh.com/current/user-manual/capabilities/active-response/index.html. [cit. 2024-05-02].

[118] *History*. Online. CVE. C1999-2023. Dostupné z: https://www.cve.org/About/History. [cit. 2023-11-12]. , *CVE® Program Mission*. Online. CVE. C1999-2023. Dostupné z: https://www.cve.org.

[119] *Severity ratings*. Online. RED HAT, INC. Red Hat Customer Portal. C2023. Dostupné z: https://access.redhat.com/security/updates/classification#cvss.

[120] *What is Common Vulnerability Scoring System (CVSS)*. Online. RISTO, Jonatha. SANS™ INSTITUTE. Https://www.sans.org. C2023. Dostupné z: https://www.sans.org/blog/what-is-cvss/.]

[121] *CVE-2023-47100*. Online. CVE. 2023. Dostupné z: https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2023-47100. [cit. 2024-05-03].

[122] *Explaining Red Hat Errata (RHSA, RHBA, and RHEA)*. Online. RED HAT, INC. Red Hat Customer Portal. C2020. Dostupné z: https://access.redhat.com/articles/2130961. [cit. 2024-05-03].

[123] *Product Errata*. Online. Rocky Enterprise Software Foundation. Dostupné z: https://errata.rockylinux.org. [cit. 2024-05-03].

[124] *Rocky Linux Errata*. Online. Rocky Linux Errata. C2023. Dostupné z: https://wiki.rockylinux.org/rocky/errata/. [cit. 2024-05-03].

[125] *Ansible-role-rhel9-cis*. Online. GitHub. C2024. Dostupné z: https://github.com/RedHatOfficial/ansible-role-rhel9-cis. [cit. 2024-05-02].

[126] *Systemd.special(7) — Linux manual page*. Online. Michael Kerrisk: man7.org. 2023. Dostupné z: https://www.man7.org/linux/man-pages/man7/systemd.special.7.html. [cit. 2024-05-03].

[127] *USBGuard*. Online. USBGuard. Dostupné z: https://usbguard.github.io. [cit. 2024-05-03].

SEZNAM POUŽITÝCH SYMBOLŮ A ZKRATEK

ACL	Access Control List
AES	Advanced Encryption Standard
AIDE	Advanced Intrusion Detection Environment
AIDS	Anomally Intrusion Detection Systems
ANSSI	Agence Nationale de la Sécurité des Systèmes d'Information
AWS	Amazon Web Services
BSD	Berkley Software Distribution
CD-ROM	Compact Disc Read-Only Memory
CIS	Center for Internet Security
CPU	Central Processing Unit
CIA	Confidentiality Integrity Accessibility
CSI	Computer Science Laboratory
CVE	Common Vulnerabilities a Exposures
CVSS	Common Vulnerability Scoring System
DMZ	Demilitarized Zone
DNF	Dandified YUM
DNS	Domain Name System
DoS	Denial Of Service
DVD	Digital Video Disc
ECDSA	Elliptic Curve Digital Signature Algorithm
EDR	Endpoint Detection Response
FHS	Filesystem Hierarchy Structure
FIM	File Integrity Monitoring
GB	Gigabyte

GCP	Google Cloud Platform
GRUB	Grand Unified Bootloader
HDD	Hard Disk Drive
HID	Human Interface Device
HIDS	Host-based Intrusion Detection System
HTTP	Hypertext Transfer Protocol
IDES	Intrusion Detection Expert System
IDS	Intrusion Detection Systém
IOC	Indicators of Compromise
IP	Internet Protocol
IPS	Intrusion Prevention Systems
ISO	International Organization for Standardization
LANL	Los Alamos National Laboratory
LDAP	Lightweight Directory Access Protocol
LXC	Linux Containers
LXD	Linux Daemon
MAC	Mandatory Access Control
MIDAS	Multics Intrusion Detection and Alerting System
NIDES	Next-Generation Intrusion Detection Expert System
NIDS	Network-based Intrusion Detection System
NIST	National Institute of Standards and Technology
NTP	Network Time Protocol
NÚKIB	Národní úřad pro kybernetickou a informační bezpečnost
OpenVAS	Open Vulnerability Assesment System
OVA	Open Virtual Appliance
P-BEST	Production Based Expert System Toolset

PAM	Pluggable Authentication Modules
QEMU	Quick Emulator
RAM	Random-access Memory
RELP	Reliable Event Logging Protocol
RFC	Request For Comment
SAAS	Software As A Service
SCA	Security Configuration Assesment
SCAP	Security Content Automation Protocol
SELinux	Security Enhanced Linux
SIDS	Signature Intrusion Detections Systems
SIEM	Security Event and Information Management
SSH	Secure Shell
SSHD	Secure Shell Daemon
SSL	Secure Socket Layer
TB	Terabyte
TCP	Transmission Control Protocol
TLS	Transport Layer Security
UDP	User Datagram Protocol
UEFI	Unified Extensible Firmware Interface
VMUG	VMware User Group
W&S	Wisdom and Sense
YUM	Yellowdog Updater Modifier

SEZNAM OBRÁZKŮ

SEZNAM TABULEK

SEZNAM PŘÍLOH

Příloha P I CD-ROM

PŘÍLOHA P I: CD-ROM

Přiložené CD-ROM obsahuje:

README.md	- Dokumentace
ansible	- Adresář s Ansible scénáři
konfigurace	- Adresář s konfiguracemi HIDS, kickstart
reporty	- Adresář s reporty testů