Michał Dziarkowski,
Aleksandra Rząca,
Szymon Szkarłat

Testy Penetracyjne

Laboratorium 7

Zadanie 1/2

Raport z Wyników Testu Penetracyjnego

Data: 15 stycznia 2025

Autorzy: Michał Dziarkowski, Aleksandra Rząca, Szymon Szkarłat

Analiza wyników

1. Identyfikacja podatności

Wyniki skanowania Nmap:

Host: 192.168.1.10

- Otwarty port 22/tcp: Usługa SSH (OpenSSH 7.4)
- Otwarty port 80/tcp: Usługa HTTP (Apache httpd 2.4.29)
- Otwarty port 443/tcp: Usługa HTTPS (Apache httpd 2.4.29)
- Otwarty port 3306/tcp: Usługa MySQL (MySQL 5.7.21)
- Otwarty port 445/tcp: Usługa SMB (Windows Server 2016 Standard)

Host: 192.168.1.20

- Otwarty port 139/tcp: Usługa NetBIOS (Samba smbd 4.7.6)
- Otwarty port 445/tcp: Usługa SMB (Samba smbd 4.7.6)

Wyniki skanera podatności Nessus:

Host: 192.168.1.10

- CVE-2018-11776 (Krytyczna): Apache Struts Remote Code Execution.
  Opis zagrożenia: Umożliwienie zdalnego wykonania kodu przez atakującego, prowadzenie do przejęcia kontroli nad serwerem.

- CVE-2021-44228 (Krytyczna): Log4j Remote Code Execution.
  Opis zagrożenia: Wykorzystanie podatności może skutkować zdalnym wykonaniem kodu, eskalacją uprawnień i wyciekiem danych.

Host: 192.168.1.20

- CVE-2017-7494 (Wysoka): Samba Arbitrary Module Loading.
  Opis zagrożenia: Umożliwia załadowanie dowolnego modułu przez atakującego, co może skutkować eskalacją uprawnień i przejęciem kontroli nad systemem.

2. Analiza logów

Host 192.168.1.10

[Jan 06 10:12:34]: Nieudana próba logowania na konto "admin" z adresu IP 192.168.1.30.

- **Potencjalne zagrożenie:** Brute-force na usługę SSH.

[Jan 06 10:13:45]: Próba dostępu do nieistniejącego pliku wp-login.php z adresu IP klienta Apache 192.168.1.31.

- **Potencjalne zagrożenie:** Skany pod kątem podatności WordPress.

[Jan 06 10:15:23]: Odrzucone połączenie do MySQL z adresu IP 192.168.1.32 z powodu błędnych danych uwierzytelniających.

- **Potencjalne zagrożenie:** Próba nieautoryzowanego dostępu do bazy danych.

Host 192.168.1.20

[Jan 06 10:20:15]: Nieudana próba uwierzytelnienia użytkownika "guest" z adresu IP 192.168.1.40.

- **Potencjalne zagrożenie:** Brute-force na usługę SMB.

[Jan 06 10:21:30]: Wykrycie potencjalnej próby exploitacji na udział publiczny.

- **Potencjalne zagrożenie:** Próba wykorzystania podatności CVE-2017-7494 (Samba).

3. Rekomendacje

Priorytety napraw podatności:

- CVE-2021-44228 (Log4j Remote Code Execution) - Krytyczna
  Działania: Aktualizacja biblioteki Log4j do najnowszej wersji.

- CVE-2018-11776 (Apache Struts Remote Code Execution) - Krytyczna
  Działania: Aktualizacja Apache Struts do wersji niepodatnej.

- CVE-2017-7494 (Samba Arbitrary Module Loading) - Wysoka
  Działania: Aktualizacja oprogramowania Samba do najnowszej wersji.

Propozycje mechanizmów monitorowania i zabezpieczeń:

- **Monitorowanie logów:** Wdrożenie centralnego systemu SIEM do analizy logów w czasie rzeczywistym i wykrywania podejrzanych zdarzeń.

- **Detekcja i zapobieganie:** Wdrożyć system klasy IDS/IPS (np. Suricata, Snort lub inne popularne rozwiązanie) do detekcji i zapobiegania atakom sieciowym.

- **Blokada adresów IP:** Skonfigurowanie czarnych list dla adresów IP wielokrotnie podejmujących próby nieautoryzowanego dostępu.

- U**wierzytelnianie dwuskładnikowe (2FA):** Wdrożenie 2FA dla usług SSH i SMB oraz analiza innych działających usług pod kątem potrzeby wdrożenia takiego uwierzytelniania.

- **Wdrożenie certyfikacji:** Zastosowanie certyfikatów TLS na stronach hostowanych przez Apache.

- **Aktualizacja oprogramowania:** Regularne aktualizowanie systemów i aplikacji w celu minimalizacji ryzyka wynikającego z nowych podatności.

- **Audyty bezpieczeństwa:** Regularne audyty aplikacji webowych, uprawnień użytkowników, metod uwierzytelniania oraz bezpieczeństwa wersji stosowanego oprogramowania.

4. Podsumowanie

Test penetracyjny wykazał istnienie kilku krytycznych podatności na hostach 192.168.1.10 i 192.168.1.20, w tym podatności Log4j i Apache Struts, które wymagają natychmiastowej reakcji. Dodatkowo, analiza logów pozwoliła zidentyfikować również podejrzane zdarzenia, takie jak próby brute-force oraz skanowanie podatności na serwerach.

Nasz zespół rekomenduje pilne podjęcie działań naprawczych oraz wdrożenie odpowiednich mechanizmów monitorowania. Przeprowadzanie regularnych audytów bezpieczeństwa jest równie kluczowe dla zapewnienia ochrony infrastruktury.

Zadanie 2/2

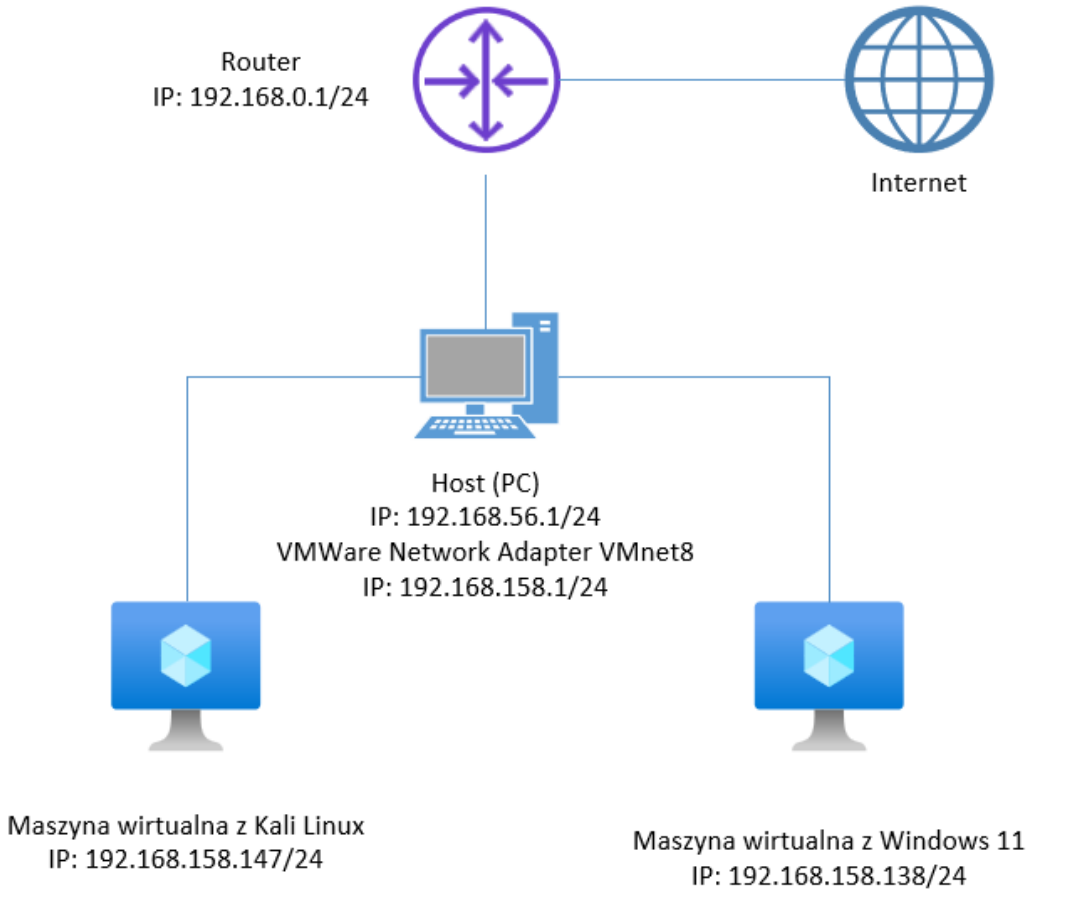
Czym jest Wazuh?

Jak możemy przeczytać chociażby na stronie, w artykule pt. *„Czym jest Wazuh?”*: https://sklep.netcomplex.pl/blog/czym-jest-wazuh, Wazuh to zaawansowana platforma open-source typu IDS (*Intrusion Detection System*), która zapewnia kompleksowe narzędzia do monitorowania bezpieczeństwa, analizy zdarzeń oraz zarządzania zgodnością. System został zaprojektowany z myślą o elastyczności i łatwej skalowalności. Wazuh oferuje wiele funkcji, takich jak: monitorowanie integralności plików (FIM, ang, *File Integrity Monitoring*), analiza zdarzeń systemowych, zarządzanie logami oraz monitorowanie bezpieczeństwa w czasie rzeczywistym.
W raporcie przedstawimy działanie Wazuh na przykładzie monitorowania integralności plików.

Schemat środowiska wirtualizacyjnego

Poniższy rysunek przedstawia środowisko, na którym zainstalowano odpowiednio: serwer Wazuh, jest to maszyna wirtualna z Kali Linux (IPv4: 192.168.158.147), oraz agenta Wazuh, jest to maszyna wirtualna z systemem Windows 11 (IPv4: 192.168.158.138), celem monitorowania działań wykonywanych na tym urządzeniu (urządzeniu z systemem Windows 11).
Działania będą wykonywane na maszynie z systemem Windows 11, z kolei analiza alertów będzie wykonywana na maszynie z systemem Linux.

Rysunek techniczny środowiska wirtualizacyjnego



Proces instalacji

Serwer Wazuh

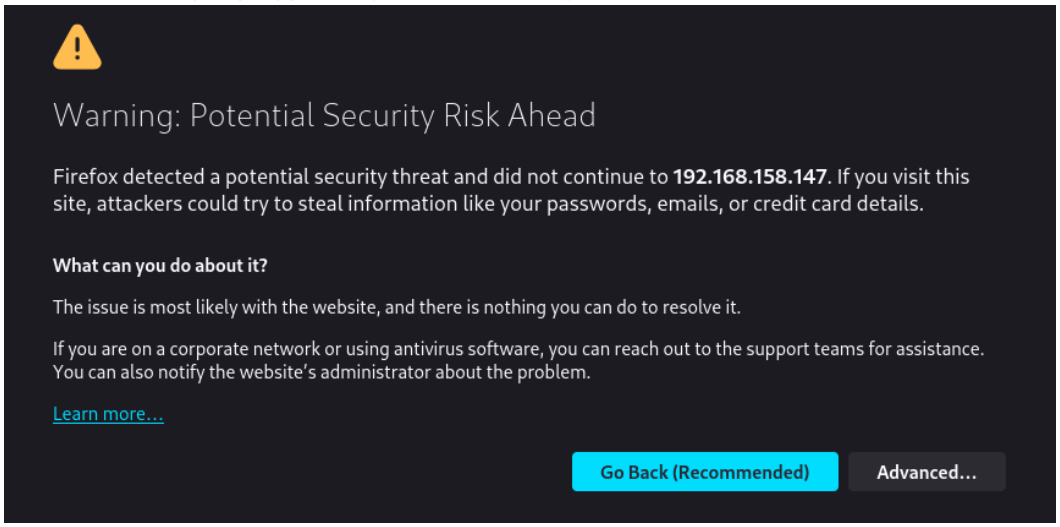
Na początku zainstalowaliśmy serwer Wazuh na maszynie z systemem Kali Linux (192.168.158.147). W tym celu skorzystaliśmy z oficjalnej dokumentacji, gdzie krok po kroku cały proces jest wytłumaczony: https://documentation.wazuh.com/current/quickstart.html.

1. Pierwszym krokiem jest pobranie i uruchomienie asystenta instalacji Wazuh. Realizujemy to za pomocą polecenia: *curl -sO https://packages.wazuh.com/4.10/wazuh-install.sh && sudo bash ./wazuh-install.sh -a*

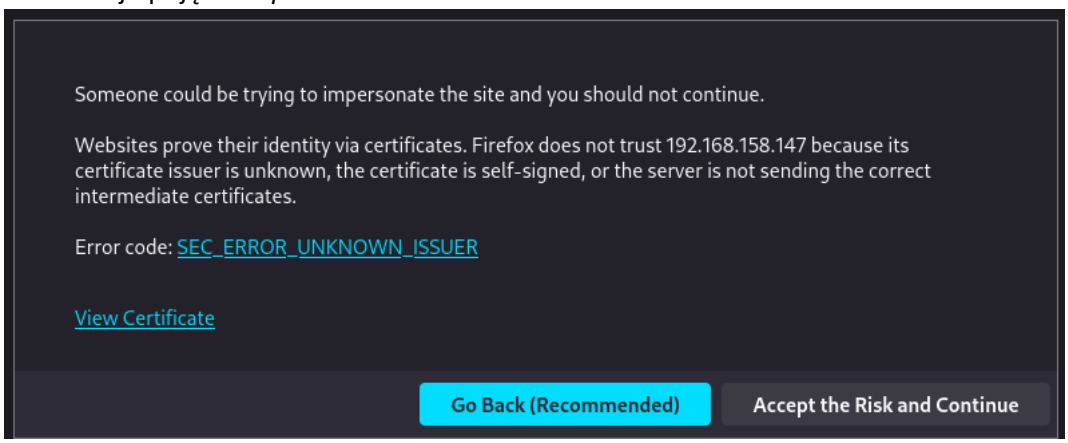
```
┌──(kali㉿kali)-[~]
└─$ curl -sO https://packages.wazuh.com/4.10/wazuh-install.sh && sudo bash ./wazuh-install.s
h -a
```

2. Na końcu, udanego procesu instalacji, na ekranie powinniśmy otrzymać dane dostępowe oraz potwierdzenie pozytywnego zakończenia procesu instalacji serwera Wazuh.
   W naszym przypadku jest to login: *admin* oraz hasło: *9sLtLlFbqPOIVg*06NDUOl2mK63yGAug*.

3. Wchodzimy do dashboardu Wazuh, wprowadzając w przeglądarce adres IP, na którym działa serwer. W naszym przypadku jest to fraza: *https://192.168.158.147*

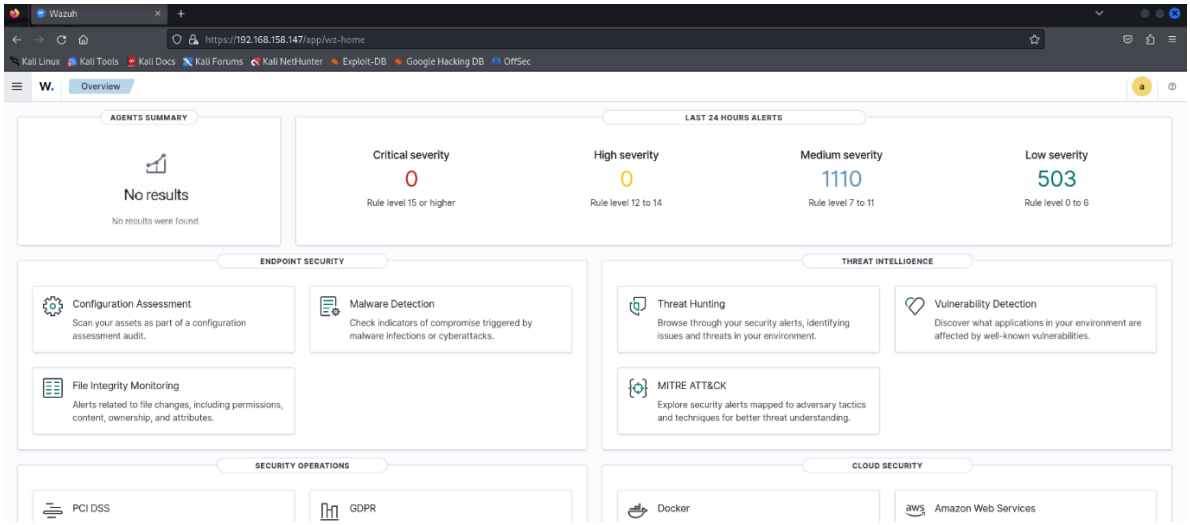


Za pierwszym razem musimy zaakceptować ryzyko (brak certyfikatu), klikamy *Advanced* oraz dalej opcję *Accept the Risk and Continue*.



4. Wprowadzamy dane dostępowe, wcześniej wspomniane login oraz hasło).

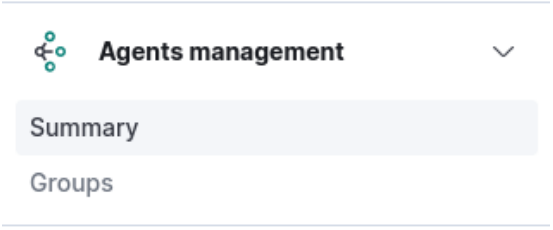
Mamy dostęp do dashboardu. Nie mniej jednak, nie mamy jeszcze żadnego podpiętego agenta.



Agent Wazuh

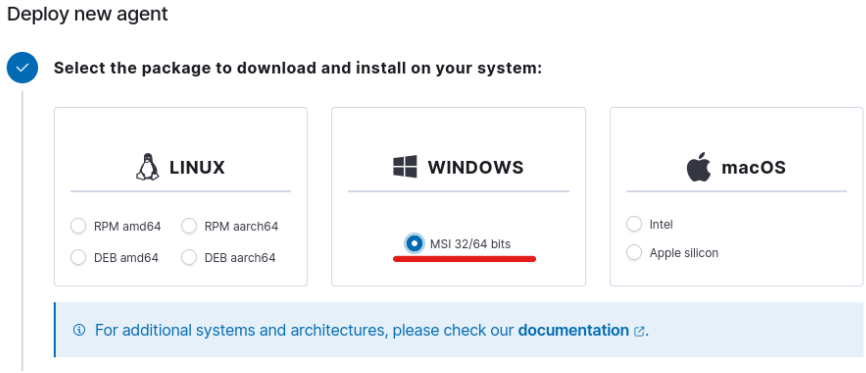
Kolejnym niezbędnym elementem do prawidłowego działania jest instalacja agenta Wazuh. W tym celu korzystamy z dokumentacji, dostępnej pod linkiem, gdzie proces instalacji agenta Wazuh jest kompleksowo przedstawiony: https://documentation.wazuh.com/current/installation-guide/wazuh-agent/index.html.

1. Pierwszym krokiem jest wejście w dashboard Wazuh (robimy to na Kali Linux). Już tam jesteśmy, co pokazuje ostatni zrzut ekranu. Tam wybieramy opcję *Agents management* oraz *Summary* z rozwijanego bocznego menu.

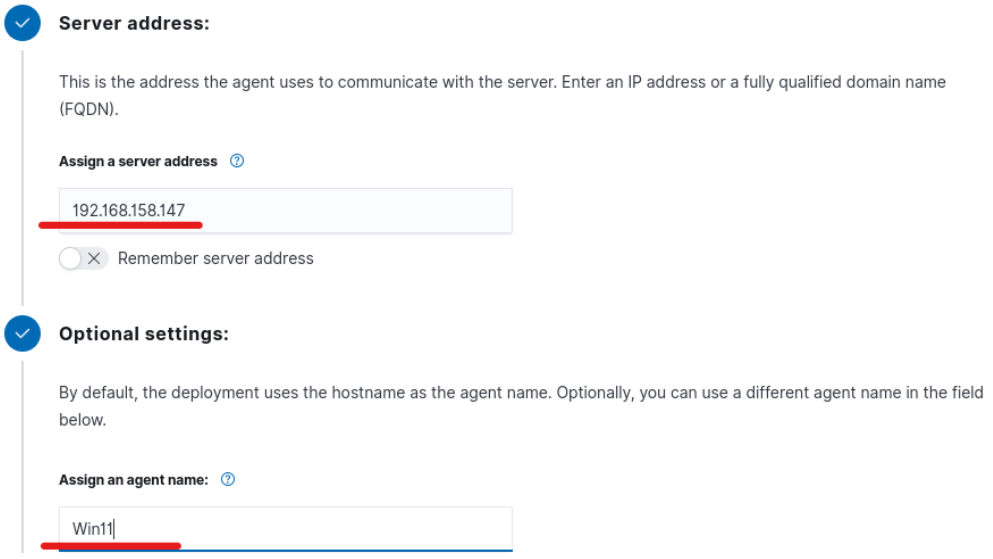


2. Następnie wybieramy opcję *Deploy new agent*.

3. Kolejno wybieramy Windows, ponieważ nasz agent będzie działał na systemie Windows 11.



4. Następnie wpisujmy adres IP, na którym zainstalowano serwer Wazuh, tj. *192.168.158.147*, a także podajmy nazwę dla agenta Wazuh: *Win11*.



5. Kolejny krokiem jest skopiowanie komend, które należy wykonać na systemie, na którym będzie dokonywana instalacja agenta (w naszym przypadku Windows 11). Komendy wprowadzamy w Power Shellu. Należy uruchomić to narzędzie jako *Administrator*. Poniższe polecenie odpowiada za pobranie oraz instalację agenta Wazuh.

4 Run the following commands to download and install the agent:

```
Invoke-WebRequest -Uri https://packages.wazuh.com/4.x/windows/wazuh-agent-4.10.0-1.msi -OutFile
$env:tmp\wazuh-agent; msiexec.exe /i $env:tmp\wazuh-agent /q WAZUH_MANAGER='192.168.158.147'
WAZUH_AGENT_NAME='Win11'
```

Wykonanie komendy na maszynie z Windows 11.

```
Writing web request
   Writing request stream... (Number of bytes written: 48588)

PS C:\WINDOWS\system32>
PS C:\WINDOWS\system32>
PS C:\WINDOWS\system32>
PS C:\WINDOWS\system32> Invoke-WebRequest -Uri https://packages.wazuh.com/4.x/windows/wazuh-agent-4.10.0-1.msi -Out
File $env:tmp\wazuh-agent; msiexec.exe /i $env:tmp\wazuh-agent /q WAZUH_MANAGER='192.168.158.147' WAZUH_AGENT_NAME=
'Win11'
```

6. Ostatnim etapem jest uruchomienie agenta. Za pomocą poniższej komendy wprowadzonej w Power Shell, na tych samych zasadach co wcześniejsze polecenie. Dzięki tym krokom uruchomimy agenta Wazuh, na maszynie wirtualnej Windows 11 (192.168.158.138).

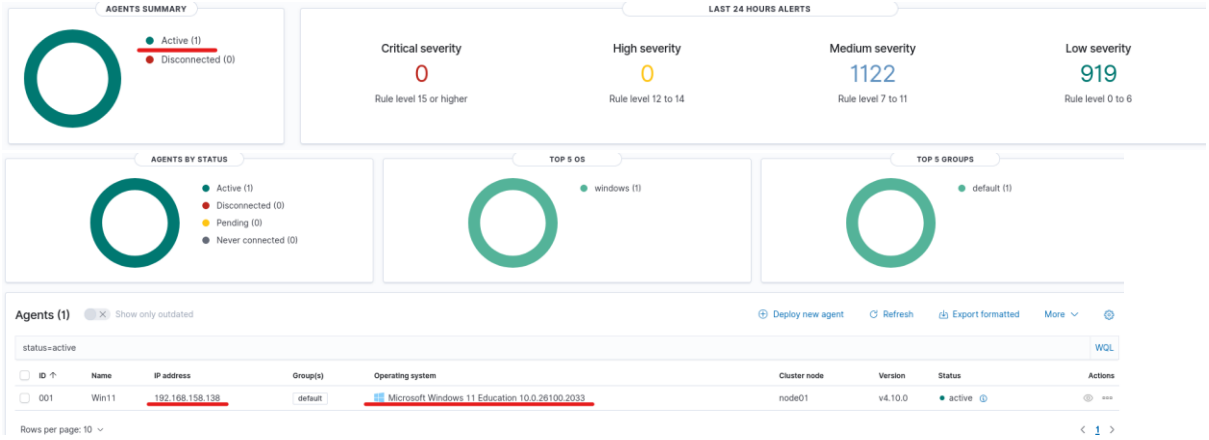
5 Start the agent:

```
NET START WazuhSvc
```

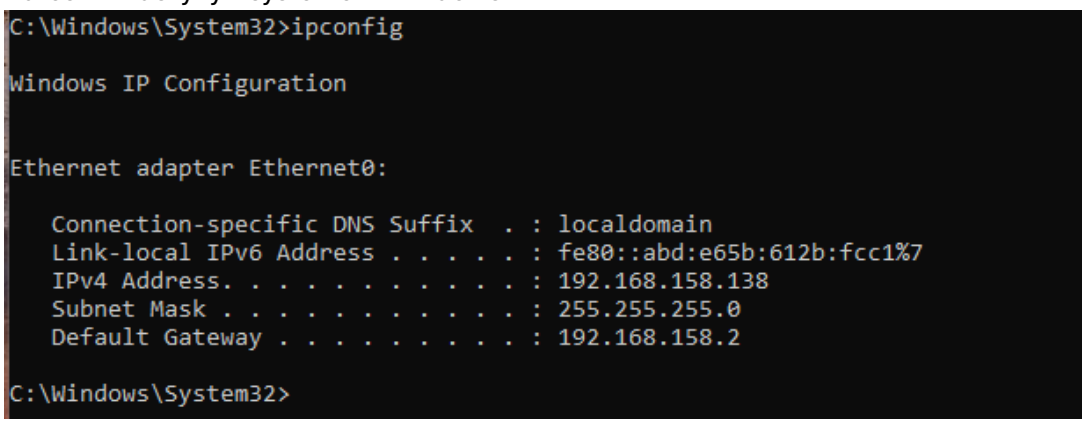
Wykonanie komendy na systemie Windows 11.

```
PS C:\WINDOWS\system32> NET START WazuhSvc
Żądana usługa została już uruchomiona.

Dostępne są dalsze informacje Pomocy; aby je uzyskać, wpisz NET HELPMSG 2182.

PS C:\WINDOWS\system32>
```

7. W dashboardzie mamy potwierdzenie, że agent został pomyślnie zainstalowany, jest on w stanie active – akywny/dostępny.



Adres IP maszyny z systemem Windows 11.

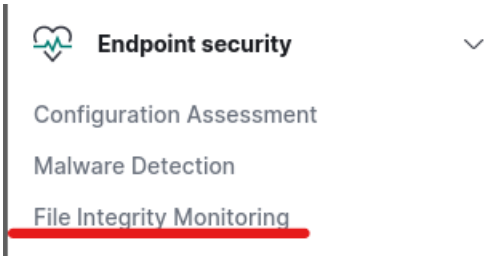


Za pomocą zwykłego polecenia *ipconfig*, potwierdzamy, że maszyna wirtualna Windows 11 posiada adres IPv4 równy 192.168.158.138.

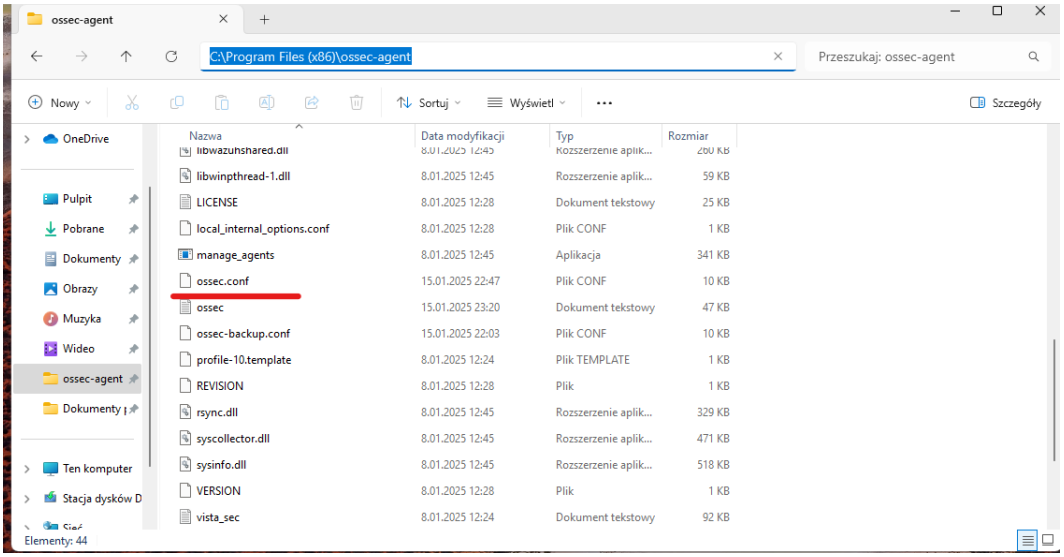
Działanie Wazuh

W naszym raporcie przedstawimy działanie Wazuh na podstawie wykrycia nieautoryzowanej zmiany w systemie plików (dodania nowego pliku oraz modyfikacji istniejącego pliku). Pokażemy jak Wazuh monitoruje zmiany w plikach na maszynie Windows 11 (urządzenie na którym zainstalowano agenta Wazuh).

1. Wszystkie zmiany integralności plików są śledzone w zakładce *Endpoint security* i dalej *File Integration Monitoring* (menu boczne). Tam się przenosimy.



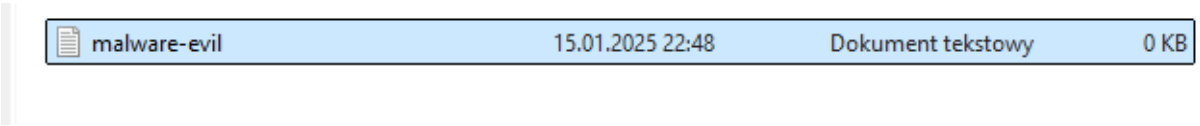
2. Aby móc zobaczyć zmiany integralności plików musimy zmodyfikować odpowiednie opcje w konfiguracji agenta Wazuh. Zatem przechodzimy na maszynę z Windows 11. Poszukujemy pliku ossec.conf. Warto wykonać jego kopię zapasową, na wszelki wypadek.



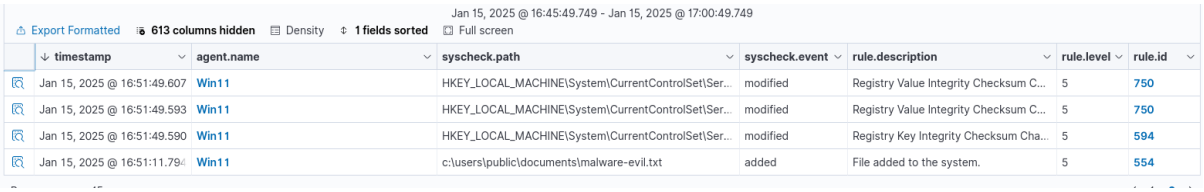
3. Wprowadzamy zmiany w zaznaczonych na czerwono miejscach, jest to sekcja *File integrity monitoring*. Wprowadzone zmiany to, zwiększenie częstotliwości sprawdzania integralności plików, poprzez zmianę czasu z 12 godzin, na 10 sekund. Co 10 sekund zatem będzie weryfikowana integralność plików z podanych lokalizacji. Dodatkowo dodajemy nową lokalizację (C:\Users\Public), w której będziemy dodawać nowe pliki i je następnie modyfikować.

```
    <skip_nfs>yes</skip_nfs>
  </sca>

  <!-- File integrity monitoring -->
  <syscheck>

    <disabled>no</disabled>

    <!-- Frequency that syscheck is executed default every 12 hours -->
    <frequency>10</frequency>

    <!-- Default files to be monitored. -->
    <directories recursion_level="0" restrict="regedit.exe$|system.ini$|win.ini$">%WINDIR%</directories>
    <directories>C:\Users\Public</directories>
    <directories recursion_level="0" restrict="at.exe$|attrib.exe$|cacls.exe$|cmd.exe$|eventcreate.exe$|ftp.exe$|lsass.exe$|net.exe$|net1.exe
$|netsh.exe$|reg.exe$|regedt32.exe|regsvr32.exe|runas.exe|sc.exe|schtasks.exe|sethc.exe|subst.exe$">%WINDIR%\SysNative</directories>
    <directories recursion_level="0">%WINDIR%\SysNative\drivers\etc</directories>
    <directories recursion_level="0" restrict="WMIC.exe$">%WINDIR%\SysNative\wbem</directories>
```

4. Po wprowadzonych zmianach możemy monitorować zmiany wprowadzone w konkretnym katalogu, tj. w katalogu *Public*. Zatem przechodzimy do maszyny Windows 11, gdzie tworzymy nowy plik o nazwie *malware-evil.txt*.

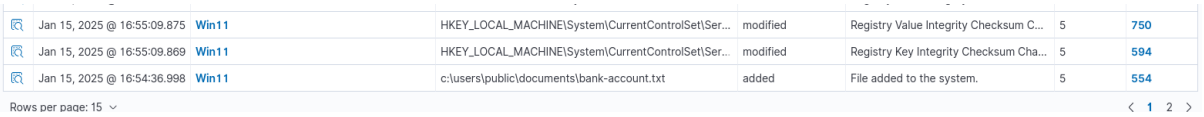


5. Plik został dodany, sprawdźmy czy informacja o tym, że dodano nowym plik została odnotowana w dashboardzie Wazuh.

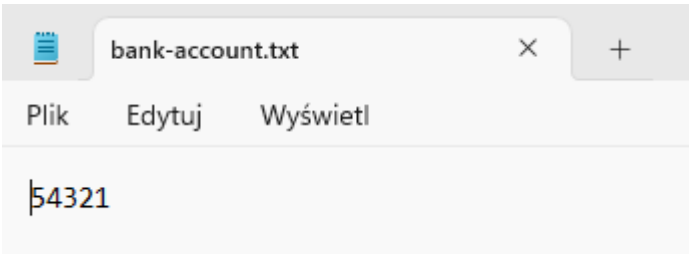


Oczywiście, że tak, potwierdza to powyższy screen. Mamy tu podaną dokładną datę wykonania akcji (ang. *timestamp*) oraz informację o tym na jakim agencie ta akcja została wykonana (*agent.name*), a także informację o tym jakiego rodzaju akcja została wykonana – *added* (*syncheck.event*) oraz jaki jest opis tejże akcji – *File added to the system* (*rule.describtion*).

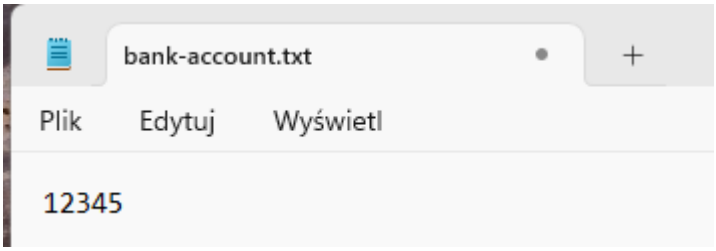
6. Kolejnym przykładem, jest modyfikacja istniejącego w monitorowanej lokalizacji pliku (nazwa istniejącego pliku: *bank-account.txt*). Plik ten został uprzednio dodany, o czym też mamy również informację.



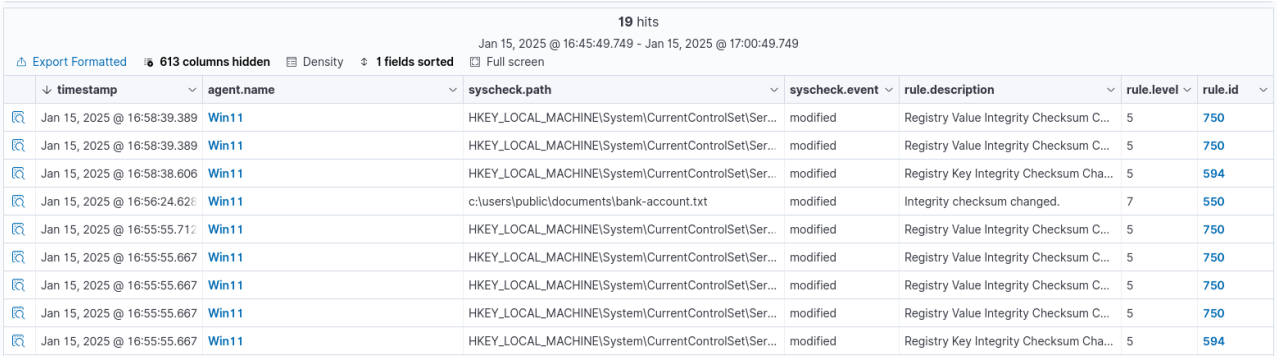
7. Zawartość pliku *bank-account.txt* przed modyfikacją.



8. Zawartość pliku *bank-account.txt* po modyfikacji.



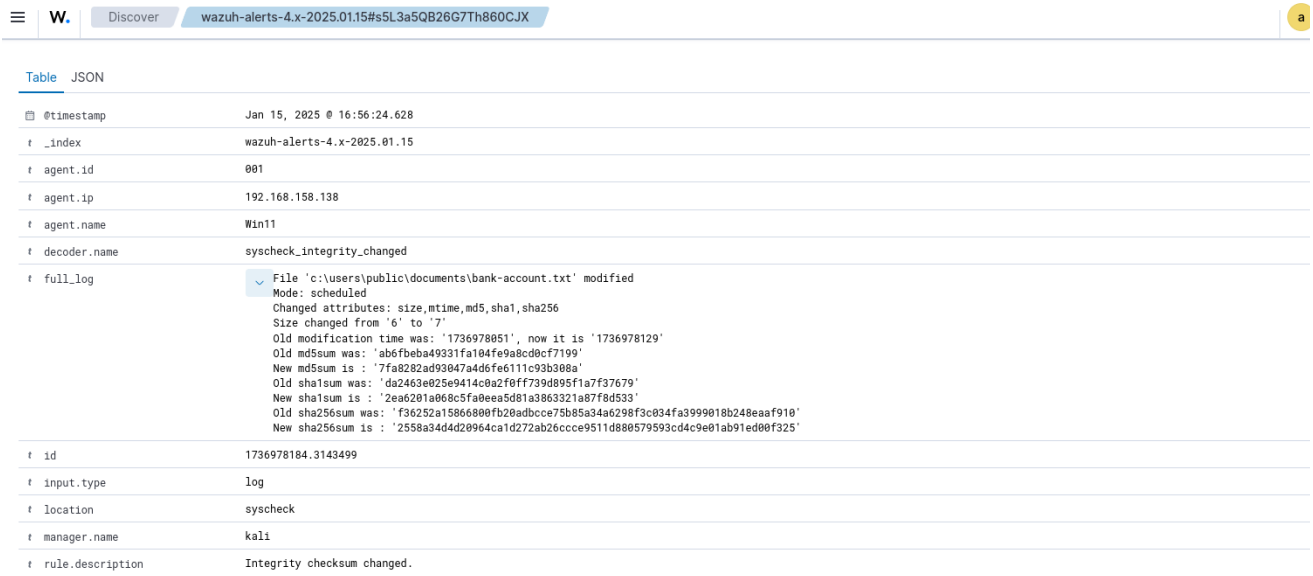
9. Zmiana ta została odnotowana przez Wazuh.



	timestamp	agent.name	syscheck.path	syscheck.event	rule.description	rule.level	rule.id
	Jan 15, 2025 @ 16:58:39.389	Win11	HKEY_LOCAL_MACHINE\System\CurrentControlSet\Ser...	modified	Registry Value Integrity Checksum C...	5	750
	Jan 15, 2025 @ 16:58:39.389	Win11	HKEY_LOCAL_MACHINE\System\CurrentControlSet\Ser...	modified	Registry Value Integrity Checksum C...	5	750
	Jan 15, 2025 @ 16:58:38.606	Win11	HKEY_LOCAL_MACHINE\System\CurrentControlSet\Ser...	modified	Registry Key Integrity Checksum Cha...	5	594
	Jan 15, 2025 @ 16:56:24.628	Win11	c:\users\public\documents\bank-account.txt	modified	Integrity checksum changed.	7	550
	Jan 15, 2025 @ 16:55:55.712	Win11	HKEY_LOCAL_MACHINE\System\CurrentControlSet\Ser...	modified	Registry Value Integrity Checksum C...	5	750
	Jan 15, 2025 @ 16:55:55.667	Win11	HKEY_LOCAL_MACHINE\System\CurrentControlSet\Ser...	modified	Registry Value Integrity Checksum C...	5	750
	Jan 15, 2025 @ 16:55:55.667	Win11	HKEY_LOCAL_MACHINE\System\CurrentControlSet\Ser...	modified	Registry Value Integrity Checksum C...	5	750
	Jan 15, 2025 @ 16:55:55.667	Win11	HKEY_LOCAL_MACHINE\System\CurrentControlSet\Ser...	modified	Registry Value Integrity Checksum C...	5	750
	Jan 15, 2025 @ 16:55:55.667	Win11	HKEY_LOCAL_MACHINE\System\CurrentControlSet\Ser...	modified	Registry Key Integrity Checksum Cha...	5	594

Jak widzimy na screenie opis tej akcji to: *Integrity checksum changed*, czyli *suma kontrolna integralności została zmieniona*. Akcja ta została oznaczona jako *modified* oraz został jej przypisany wyższy poziom niż akcji *addded, czyli* (*rule.level*) równe 7, a nie 5.

10. W przypadku tej akcji analizujmy również szczegóły przechwyconego alertu.



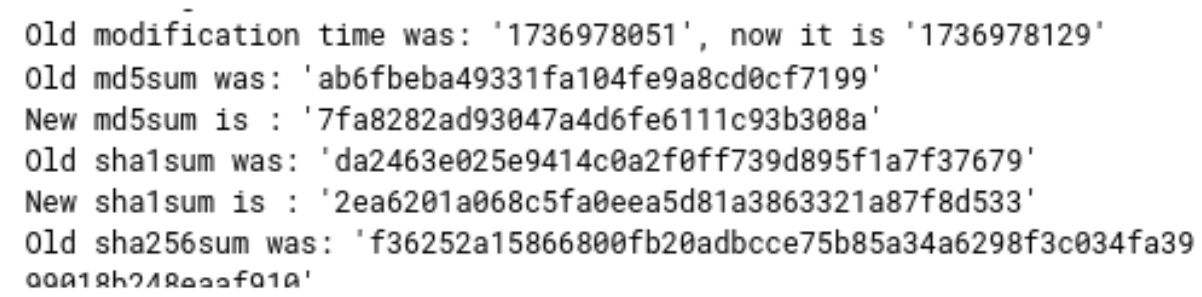
@timestamp	Jan 15, 2025 @ 16:56:24.628
_index	wazuh-alerts-4.x-2025.01.15
agent.id	001
agent.ip	192.168.158.138
agent.name	Win11
decoder.name	syscheck_integrity_changed

full_log

```
File 'c:\users\public\documents\bank-account.txt' modified
Mode: scheduled
Changed attributes: size,mtime,md5,sha1,sha256
Size changed from '6' to '7'
Old modification time was: '1736978051', now it is '1736978129'
Old md5sum was: 'ab6fbeba49331fa104fe9a8cd0cf7199'
New md5sum is : '7fa8282ad93047a4d6fe6111c93b308a'
Old sha1sum was: 'da2463e025e9414c0a2f0ff739d895f1a7f37679'
New sha1sum is : '2ea6201a068c5fa0eea5d81a3863321a87f8d533'
Old sha256sum was: 'f36252a15866800fb20adbcce75b85a34a6298f3c034fa3999018b248eaaf910'
New sha256sum is : '2558a34d4d20964ca1d272ab26ccce9511d880579593cd4c9e01ab91ed00f325'
```

id	1736978184.3143499
input.type	log
location	syscheck
manager.name	kali
rule.description	Integrity checksum changed.

I tak widzimy, że zmiany w pliku dokonano na maszynie z Windows 11 (*agent.id* = 1).



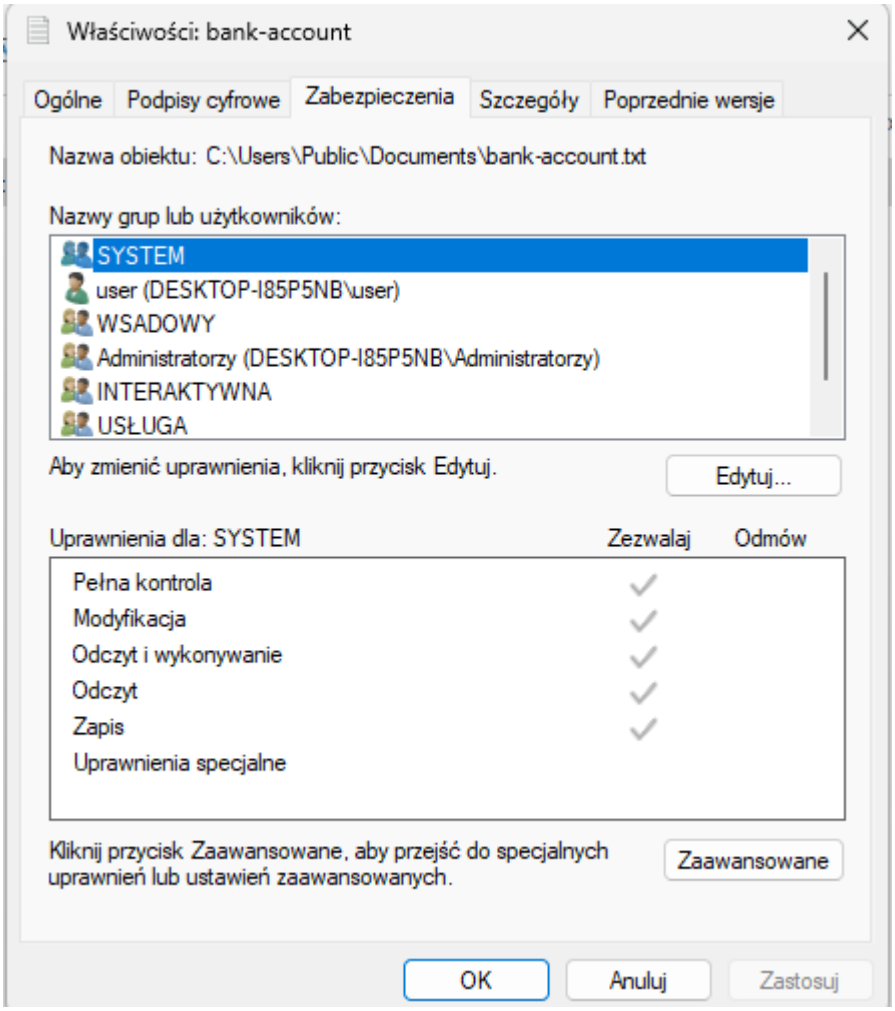
```
Old modification time was: '1736978051', now it is '1736978129'
Old md5sum was: 'ab6fbeba49331fa104fe9a8cd0cf7199'
New md5sum is : '7fa8282ad93047a4d6fe6111c93b308a'
Old sha1sum was: 'da2463e025e9414c0a2f0ff739d895f1a7f37679'
New sha1sum is : '2ea6201a068c5fa0eea5d81a3863321a87f8d533'
Old sha256sum was: 'f36252a15866800fb20adbcce75b85a34a6298f3c034fa39
99018b248eaaf910'
```

Potwierdzenie w postaci hashy, że zmiana została dokonana. Sumy kontrolne się od siebie różnią, mowa tu o starych (*old*) oraz nowych (*new*) hashach.

11. Alert wyświetla również informację o tym, kto ma jakie prawa dostępu do pliku.

syscheck.mtime_after	Jan 15, 2025 @ 11:55:29.000
syscheck.mtime_before	Jan 15, 2025 @ 11:54:11.000
syscheck.path	c:\users\public\documents\bank-account.txt
syscheck.sha1_after	2ea6201a068c5fa0eea5d81a3863321a87f8d533
syscheck.sha1_before	da2463e025e9414c0a2f0ff739d895f1a7f37679
syscheck.sha256_after	2558a34d4d20964ca1d272ab26ccce9511d880579593cd4c9e01ab91ed00f325
syscheck.sha256_before	f36252a15866800fb20adbcce75b85a34a6298f3c034fa3999018b248eaaf910
syscheck.size_after	7
syscheck.size_before	6
syscheck.uid_after	S-1-5-21-2512249615-3878565102-2367661405-1001
syscheck.uname_after	user
syscheck.win_perm_after.allowed	DELETE, READ_CONTROL, WRITE_DAC, WRITE_OWNER, SYNCHRONIZE, READ_DATA, WRITE_DATA, APPEND_DATA, READ_EA, WRITE_EA, EXECUTE, READ_ATTRIBUTES, WRITE_ATTRIBUTES, DELETE, READ_CONTROL, WRITE_DAC, WRITE_OWNER, SYNCHRONIZE, READ_DATA, WRITE_DATA, APPEND_DATA, READ_EA, WRITE_EA, EXECUTE, READ_ATTRIBUTES, WRITE_ATTRIBUTES, DELETE, READ_CONTROL, WRITE_DAC, WRITE_OWNER, SYNCHRONIZE, READ_DATA, WRITE_DATA, APPEND_DATA, READ_EA, WRITE_EA, EXECUTE, READ_ATTRIBUTES, WRITE_ATTRIBUTES, DELETE, READ_CONTROL, SYNCHRONIZE, READ_DATA, WRITE_DATA, APPEND_DATA, READ_EA, WRITE_EA, EXECUTE, READ_ATTRIBUTES, WRITE_ATTRIBUTES, DELETE, READ_CONTROL, SYNCHRONIZE, READ_DATA, WRITE_DATA, APPEND_DATA, READ_EA, WRITE_EA, EXECUTE, READ_ATTRIBUTES, WRITE_ATTRIBUTES
syscheck.win_perm_after.name	⚠ Administratorzy, user, SYSTEM, INTERAKTYWNA, US$UGA, WSADOWY
timestamp	Jan 15, 2025 @ 16:56:24.628

Zabezpieczenia pliku *bank-account.txt*, wyświetlone na maszynie Windows 11.



Użytkownicy, którzy mają dostęp do pliku, są tacy sami jak w przechwyconym alercie.

Podsumowanie

Przeprowadzone działania miały na celu zademonstrowanie, jak Wazuh monitoruje integralność systemu plików. W tym celu zainstalowano serwer Wazuh, działający na maszynie Kali Linux (IPv4: 192.168.158.147) oraz skonfigurowano agenta Wazuh na maszynie wirtualnej z systemem Windows 11 (IPv4: 192.168.158.138), do obserwowania plików katalogu tj. *C:\Users\Public*, co umożliwia rejestrowanie wszelkich zmian w tym obszarze. Następnie w katalogu dokonano celowych zmian, takich jak utworzenie (dodanie nowego pliku *malware-evil.txt*) czy edycja istniejącego pliku (*bank-account.txt*), co symulowało potencjalnie nieautoryzowane operacje. W dashboardzie Wazuh zaprezentowano alerty wygenerowane przez system, zawierające szczegółowe informacje o zmodyfikowanych plikach, czasie wykonanych akcji oraz typie operacji. Działania te pokazały, jak Wazuh pomaga wykrywać naruszenia bezpieczeństwa w czasie rzeczywistym, co jest kluczowe dla ochrony systemów przed nieautoryzowanym dostępem lub modyfikacją danych.