Report
Alert Analyst
Brute Force and SQL Injection

Author : Mochammad Arif Rizki
***Date analyst alert** : Sat Feb 25 06:50:26 WIB 2023*

Summary The agent's hostname webpanel has WordPress installed

Pentest SSH Brute Force Attack

- **Server Hostname Agent : webpannel**
- **IP Agent : 192.168.31.165**

Tools use **hydra** for SSH Brute Force.

```
[arif@fedora ~]$ hydra -l root -P passwd.txt 192.168.31.165 ssh
```

Threat Hunting
Check Discovery:

- rule.id : **31103**
- id : **1677278005.965906**
- IP Attacker : **192.168.31.10**


```
Feb 25, 2023 @ 06:29:47.897  agent.name: webpanel rule.mitre.technique: Brute Force agent.id: 005 agent.ip: 192.168.31.165 data.dstuser: root data.srcip: 192.168.31.10 decoder.name: sshd decoder.parent: sshd
full_log: Feb 25 06:29:34 webpanel sshd[81852]: PAM 2 more authentication failures; logname= uid=0 euid=0 tty=ssh ruser= rhost=192.168.31.10 user=root id: 1677281387.1004319
input.type: log location: /var/log/secure manager.name: siem predecoder.hostname: webpanel predecoder.program_name: sshd predecoder.timestamp: Feb 25 06:29:34 rule.description: syslog:
User missed the password more than one time rule.firedtimes: 1 rule.gdpr: IV_35.7.d, IV_32.2 rule.gpg13: 7.8 rule.groups: syslog, access_control, authentication_failed

Feb 25, 2023 @ 06:29:46.042  agent.name: webpanel rule.mitre.technique: Valid Accounts, Brute Force agent.id: 005 agent.ip: 192.168.31.165 data.dstuser: root data.srcip: 192.168.31.10 data.srcport: 37796
decoder.name: sshd decoder.parent: sshd full_log: Feb 25 06:29:32 webpanel sshd[81848]: Accepted password for root from 192.168.31.10 port 37796 ssh2 id: 1677281386.1001917
input.type: log location: /var/log/secure manager.name: siem predecoder.hostname: webpanel predecoder.program_name: sshd predecoder.timestamp: Feb 25 06:29:32 rule.description: Multiple
authentication failures followed by a success. rule.firedtimes: 1 rule.frequency: 2 rule.gdpr: IV_35.7.d, IV_32.2 rule.gpg13: 7.1, 7.8 rule.groups: syslog, attacks rule.hipaa: 164.312.b

Feb 25, 2023 @ 06:29:43.937  agent.name: webpanel rule.mitre.technique: Brute Force agent.id: 005 agent.ip: 192.168.31.165 data.dstuser: root data.srcip: 192.168.31.10 data.srcport: 37758 decoder.name: sshd
decoder.parent: sshd full_log: Feb 25 06:29:30 webpanel sshd[81843]: Failed password for root from 192.168.31.10 port 37758 ssh2 id: 1677281383.990117 input.type: log
location: /var/log/secure manager.name: siem predecoder.hostname: webpanel predecoder.program_name: sshd predecoder.timestamp: Feb 25 06:29:30 previous_output: Feb 25 06:29:30 webpanel
sshd[81845]: Failed password for root from 192.168.31.10 port 37768 ssh2 Feb 25 06:29:30 webpanel sshd[81850]: Failed password for root from 192.168.31.10 port 37806 ssh2 Feb 25 06:29:30

Feb 25, 2023 @ 06:29:43.864  agent.name: webpanel rule.mitre.technique: Brute Force agent.id: 005 agent.ip: 192.168.31.165 data.dstuser: root data.euid: 0 data.srcip: 192.168.31.10 data.tty: ssh data.uid: 0
decoder.name: pam full_log: Feb 25 06:29:28 webpanel sshd[81850]: pam_unix(sshd:auth): authentication failure; logname= uid=0 euid=0 tty=ssh ruser= rhost=192.168.31.10 user=root
id: 1677281383.981356 input.type: log location: /var/log/secure manager.name: siem predecoder.hostname: webpanel predecoder.program_name: sshd predecoder.timestamp: Feb 25 06:29:28
previous_output: Feb 25 06:29:28 webpanel sshd[81848]: pam_unix(sshd:auth): authentication failure; logname= uid=0 euid=0 tty=ssh ruser= rhost=192.168.31.10 user=root Feb 25 06:29:28
```

More Brute Force Alert Details

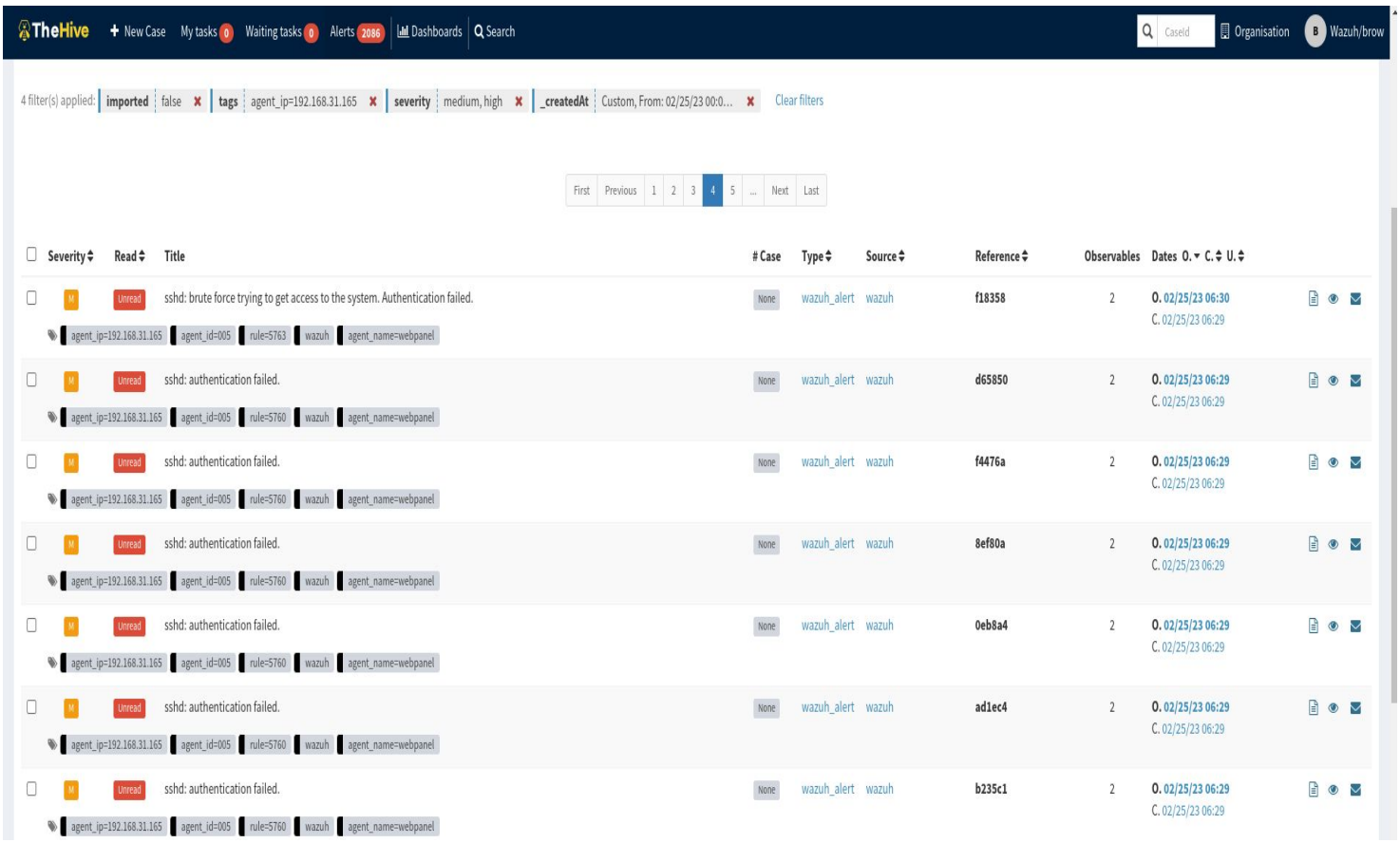
Table JSON

Field	Value
_index	wazuh-alerts-4.x-2023.02.24
@timestamp	Feb 25, 2023 @ 06:29:43.937
agent.id	005
agent.ip	192.168.31.165
agent.name	webpanel
data.dstuser	root
data.srcip	192.168.31.10
data.srcport	37758
decoder.name	sshd
decoder.parent	sshd
full_log	Feb 25 06:29:30 webpanel sshd[81843]: Failed password for root from 192.168.31.10 port 37758 ssh2
id	1677281383.990117
input.type	log
location	/var/log/secure
manager.name	siem
predecoder.hostname	webpanel
predecoder.program_name	sshd
predecoder.timestamp	Feb 25 06:29:30
previous_output	Feb 25 06:29:30 webpanel sshd[81845]: Failed password for root from 192.168.31.10 port 37768 ssh2 Feb 25 06:29:30 webpanel sshd[81850]: Failed password for root from 192.168.31.10 port 37806 ssh2 Feb 25 06:29:30 webpanel sshd[81848]: Failed password for root from 192.168.31.10 port 37796 ssh2 Feb 25 06:29:30 webpanel sshd[81851]: Failed password for root from 192.168.31.10 port 37814 ssh2 Feb 25 06:29:30 webpanel sshd[81852]: Failed password for root from 192.168.31.10 port 37830 ssh2 Feb 25 06:29:30 webpanel sshd[81846]: Failed password for root from 192.168.31.10 port 37772 ssh2

rule.description	sshd: brute force trying to get access to the system. Authentication failed.
rule.firedtimes	1
rule.frequency	8
rule.gdpr	IV_35.7.d, IV_32.2
rule.groups	syslog, sshd, authentication_failures
rule.hipaa	164.312.b
rule.id	5763
rule.level	10
rule.mail	false
rule.mitre.id	T1110
rule.mitre.tactic	Credential Access
rule.mitre.technique	Brute Force
rule.nist_800_53	SI.4, AU.14, AC.7
rule.pci_dss	11.4, 10.2.4, 10.2.5
rule.tsc	CC6.1, CC6.8, CC7.2, CC7.3
timestamp	Feb 25, 2023 @ 06:29:43.937

Dashboards Wazuh
Traffic Brute Force with **agent webpanel**

Analyst Alert TheHive
SSH Brute Force Throw from **Wazuh**



View More Alert Details

Alert Preview New

M sshd: brute force trying to get access to the system. Authentication failed.

ID: ~864432 **Date:** 02/25/23 06:30 **Type:** wazuh_alert **Reference:** f18358 **Source:** wazuh

Basic Information

Tags agent_ip=192.168.31.165 agent_id=005 rule=5763 wazuh agent_name=webpanel

Description

Timestamp

key	val
timestamp	2023-02-25T06:29:43.937+0700

Rule

key	val
rule.level	10
rule.description	sshd: brute force trying to get access to the system. Authentication failed.
rule.id	5763
rule.mitre.id	['T1110']
rule.mitre.tactic	['Credential Access']
rule.mitre.technique	['Brute Force']
rule.frequency	8
rule.firedtimes	1
rule.mail	False
rule.groups	['syslog', 'sshd', 'authentication_failures']

rule.gdpr	['IV_35.7.d', 'IV_32.2']
rule.hipaa	['164.312.b']
rule.nist_800_53	['SI.4', 'AU.14', 'AC.7']
rule.pci_dss	['11.4', '10.2.4', '10.2.5']
rule.tsc	['CC6.1', 'CC6.8', 'CC7.2', 'CC7.3']

Agent

key	val
agent.id	005
agent.name	webpanel
agent.ip	192.168.31.165

Manager

key	val
manager.name	siem

Id

key	val
id	1677281383.990117

Previous_output

key	val
previous_output	Feb 25 06:29:30 webpanel sshd[81845]: Failed password for root from 192.168.31.10 port 37768 ssh2

Feb 25 06:29:30 webpanel sshd[81850]: Failed password for root from 192.168.31.10 port 37806 ssh2 Feb 25 06:29:30 webpanel sshd[81848]: Failed password for root from 192.168.31.10 port 37796 ssh2 Feb 25 06:29:30 webpanel sshd[81851]: Failed password for root from 192.168.31.10 port 37814 ssh2 Feb 25 06:29:30 webpanel sshd[81852]: Failed password for root from 192.168.31.10 port 37830 ssh2 Feb 25 06:29:30 webpanel sshd[81846]: Failed password for root from 192.168.31.10 port 37772 ssh2 Feb 25 06:29:30 webpanel sshd[81849]: Failed password for root from 192.168.31.10 port 37804 ssh2 |

Full_log

key	val
full_log	Feb 25 06:29:30 webpanel sshd[81843]: Failed password for root from 192.168.31.10 port 37758 ssh2

Predecoder

key	val
predecoder.program_name	sshd
predecoder.timestamp	Feb 25 06:29:30
predecoder.hostname	webpanel

Decoder

key	val
decoder.parent	sshd
decoder.name	sshd

Data

key	val
data.srcip	192.168.31.10
data.srcport	37758
data.dstuser	root

Location

key	val
location	/var/log/secure

Additional fields Layout

No additional information has been specified

Observables 2 Similar cases 3

List of observables (2 of 2)

Filters 15 per page

Flags	Type	Data	Date Added
	ip	192[.]168[.]31[.]10	02/25/23 06:29
	ip	192[.]168[.]31[.]165	02/25/23 06:29

Cancel Mark as read Ignore new updates Merge into case Delete

Import alert as -- Empty case -- Yes, Import

Next Pentest SQL Injection

Here I set the domain agent using siptesting.local so it's easy for me to pentest
IP Agent : 192.168.31.165

```
[arif@fedora ~]$ curl -XGET "http://siptesting.local/users/?id=SELECT+*+FROM+users";
```

View the alerts
Check discovery

- **rule.id : 31103**
- **id : 1677278005.965906**
- **IP Attacker : 192.168.31.10**

Feb 25, 2023 @ 05:33:25.921 agent.name: webpanel agent.id: 005 agent.ip: 192.168.31.165 data.id: 404 data.protocol: GET data.srcip: 192.168.31.10 data.url: /users/?id=SELECT+*+FROM+users decoder.name: web-accesslog full_log: 192.168.31.10 - - [25/Feb/2023:05:33:11 +0700] "GET /users/?id=SELECT+*+FROM+users HTTP/1.1" 404 41496 "-" "curl/7.85.0" id: 1677278005.965906 input.type: log location: /www/wwwlogs/siptesting.local.log manager.name: siem rule.description: SQL injection attempt. rule.firedtimes: 6 rule.gdpr: IV_35.7.d rule.groups: web, accesslog, attack, sql_injection rule.id: 31103 rule.level: 7 rule.mail: false rule.mitre.id: T1190 rule.mitre.tactic: Initial Access rule.mitre.technique: Exploit Public-Facing Application

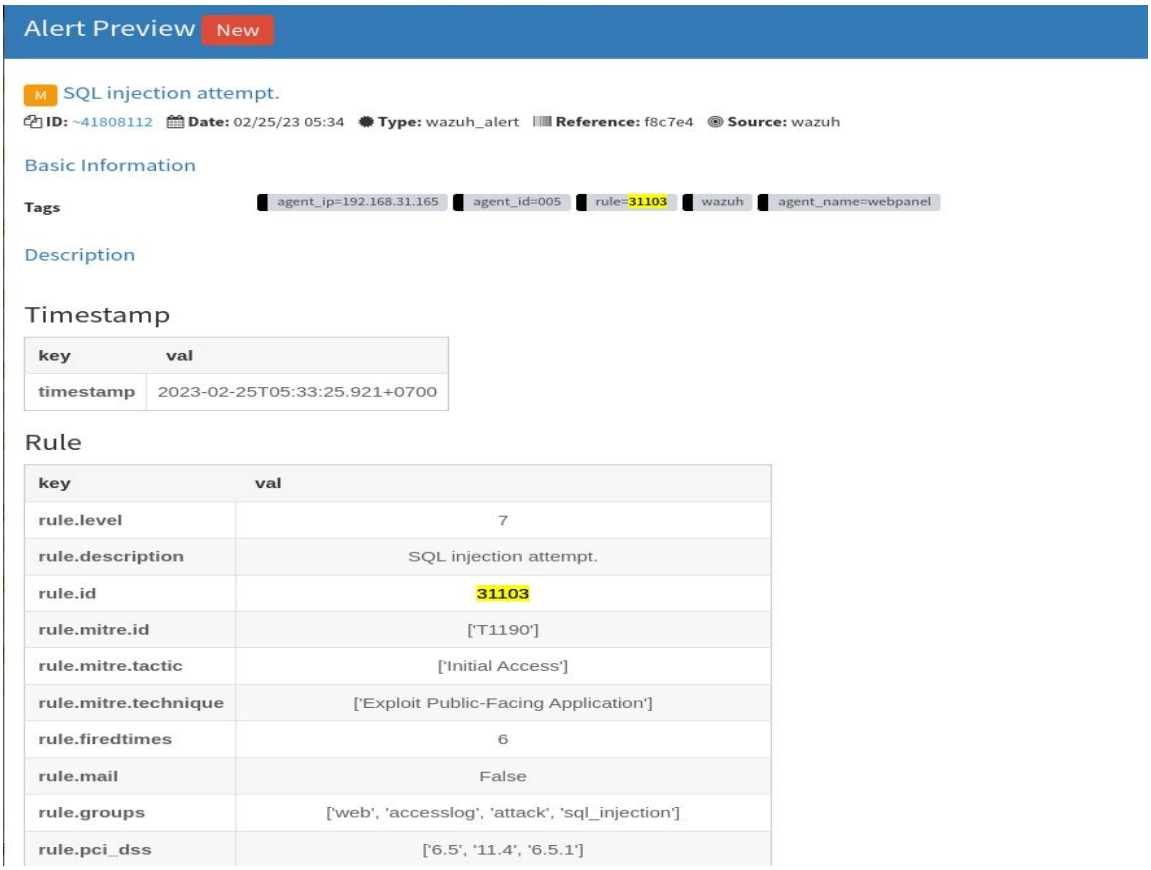
Expanded document

View surrounding documents View single document

Table JSON

Actions	Field	Value
	_index	wazuh-alerts-4.x-2023.02.24
	agent.id	005
	agent.ip	192.168.31.165
	agent.name	webpanel
	data.id	404
	data.protocol	GET
	data.srcip	192.168.31.10
	data.url	/users/?id=SELECT+*+FROM+users
	decoder.name	web-accesslog
	full_log	192.168.31.10 - - [25/Feb/2023:05:33:11 +0700] "GET /users/?id=SELECT+*+FROM+users HTTP/1.1" 404 41496 "-" "curl/7.85.0"
	id	1677278005.965906
	input.type	log
	location	/www/wwwlogs/siptesting.local.log
	manager.name	siem
	rule.description	SQL injection attempt.
	rule.firedtimes	6
	rule.gdpr	IV_35.7.d
	rule.groups	web, accesslog, attack, sql_injection
	rule.id	31103
	rule.level	7
	rule.mail	false
	rule.mitre.id	T1190
	rule.mitre.tactic	Initial Access
	rule.mitre.technique	Exploit Public-Facing Application
	rule.nist_800_53	SA.11, SI.4
	rule.pci_dss	6.5, 11.4, 6.5.1
	rule.tsc	CC6.6, CC7.1, CC8.1, CC6.1, CC6.8, CC7.2, CC7.3
	timestamp	Feb 25, 2023 @ 05:33:25.921

Analyst With TheHive
SQL Injection Alert Throw from **Wazuh**



rule.gdpr	['IV_35.7.d']
rule.nist_800_53	['SA.11', 'SI.4']
rule.tsc	['CC6.6', 'CC7.1', 'CC8.1', 'CC6.1', 'CC6.8', 'CC7.2', 'CC7.3']

Agent

key	val
agent.id	005
agent.name	webpanel
agent.ip	192.168.31.165

Manager

key	val
manager.name	siem

Id

key	val
id	1677278005.965906

Full_log

key	val
full_log	192.168.31.10 - - [25/Feb/2023:05:33:11 +0700] "GET /users/?id=SELECT+*+FROM+users HTTP/1.1" 404 41496 "-" "curl/7.85.0"

Decoder

key	val
decoder.name	web-accesslog

Data

key	val
data.protocol	GET
data.srcip	192.168.31.10
data.id	404
data.url	/users/?id=SELECT+*+FROM+users

Location

key	val
location	/www/wwwlogs/siptesting.local.log

Additional fields Layout

No additional information has been specified

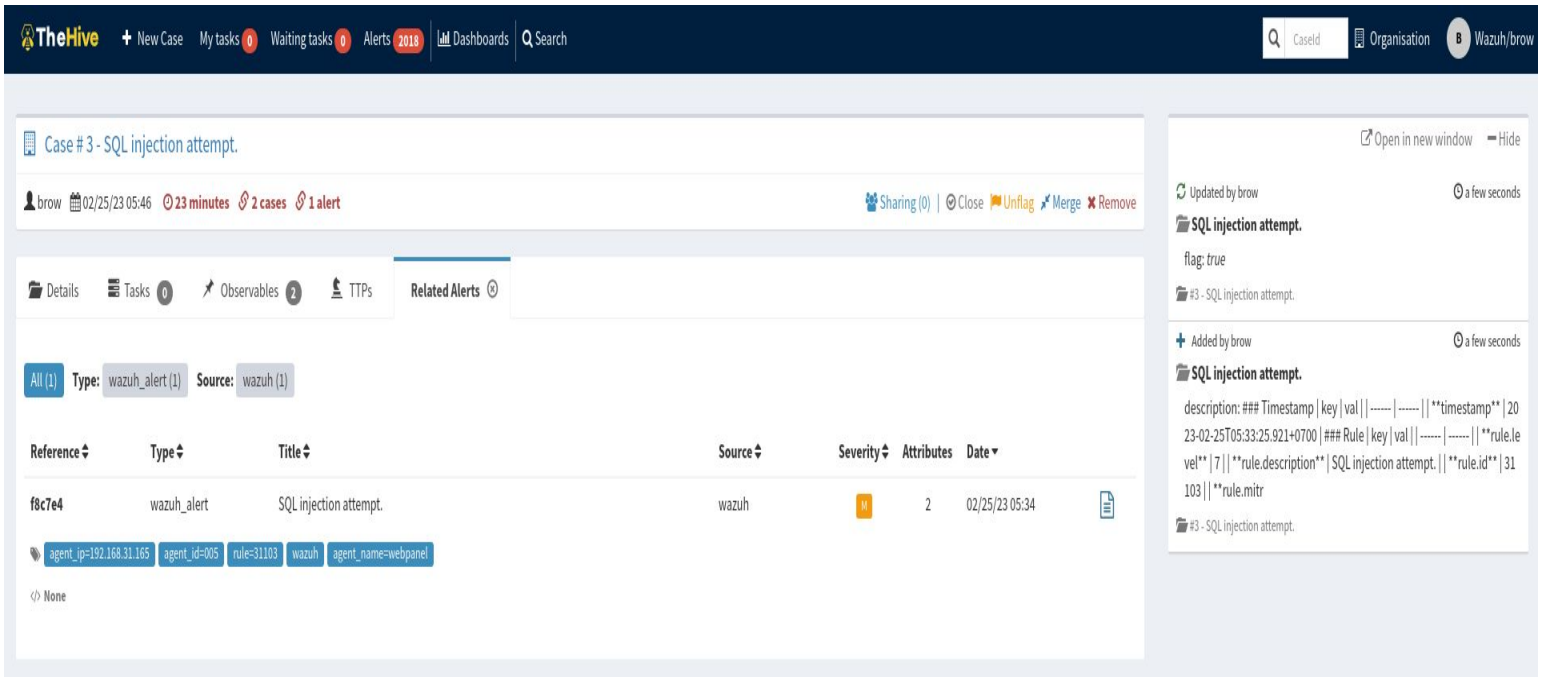
Observables 2 Similar cases 2

List of observables (2 of 2) Filters 15 per page

Flags	Type	Data	Date Added
	ip	192[.]168[.]31[.]10	02/25/23 05:34
	ip	192[.]168[.]31[.]165	02/25/23 05:34

Cancel Mark as read Ignore new updates Merge into case Delete Import alert as -- Empty case -- Yes, Import

Create Case


TheHive
New Case
My tasks 0
Waiting tasks 0
Alerts 2018
Dashboards
Search
Caseid
Organisation
Wazuh/brow
Case # 3 - SQL injection attempt.
brow
02/25/23 05:46
23 minutes
2 cases
1 alert
Sharing (0)
Close
Unflag
Merge
Remove
Details
Tasks 0
Observables 2
TTPs
Related Alerts
All (1)
Type: wazuh_alert (1)
Source: wazuh (1)
Reference
Type
Title
Source
Severity
Attributes
Date
f8c7e4
wazuh_alert
SQL injection attempt.
wazuh
M
2
02/25/23 05:34
agent_ip=192.168.31.165
agent_id=005
rule=31103
wazuh
agent_name=webpanel
None
Open in new window
Hide
Updated by brow
a few seconds
SQL injection attempt.
flag: true
#3 - SQL injection attempt.
Added by brow
a few seconds
SQL injection attempt.
description: ### Timestamp | key | val | | ------ | ------ | | **timestamp** | 2023-02-25T05:33:25.921+0700 | ### Rule | key | val | | ------ | ------ | | **rule.level** | 7 | | **rule.description** | SQL injection attempt. | | **rule.id** | 31103 | | **rule.mitr
#3 - SQL injection attempt.

MITRE ATT&CK
Techniques (T1190) SQL Injection

Procedure Examples

ID	Name	Description
G0007	APT28	APT28 has used a variety of public exploits, including CVE 2020-0688 and CVE 2020-17144, to gain execution on vulnerable Microsoft Exchange; they have also conducted SQL injection attacks against external websites
S0032	gh0st RAT	gh0st RAT can inject malicious code into process created by the "Command_Create&Inject" function
C0014	Operation Wocao	During Operation Wocao, threat actors injected code into a selected process, which in turn launches a command as a child process of the original

Mitigations

ID	Mitigations	Description
M1040	Behavior Prevention on Endpoint	Some endpoint security solutions can be configured to block some types of process injection based on common sequences of behavior that occur during the injection process. For example, on Windows 10, Attack Surface Reduction (ASR) rules may prevent Office applications from code injection
M1026	Privileged Account Management	Utilize Yama (ex: /proc/sys/kernel/yama/ptrace_scope) to mitigate ptrace based process injection by restricting the use of ptrace to privileged users only. Other mitigation controls involve the deployment of security kernel modules that provide advanced access control and process restrictions such as SELinux, grsecurity, and AppArmor

Detection

ID	Data Source	Data Component	Detects
DS0022	File	File Metadata	File Metadata Monitor for contextual data about a file, which may include information such as name, the content (ex: signature, headers, or data/media), user/ower, permissions, etc.
		File Modification	Monitor for changes made to files that may inject code into processes in order to evade process-based defenses as well as possibly elevate privileges.
DS0011	Module	Module Module Load	Monitor DLL/PE file events, specifically creation of these binary files as well as the loading of DLLs into processes. Look for DLLs that are not recognized or not normally loaded into a process.
DS0009	Process	OS API Execution	Monitoring Windows API calls indicative of the various types of code injection may generate a significant amount of data and may not be directly useful for defense unless collected under specific circumstances for known bad sequences of calls, since benign use of API functions may be common and difficult to distinguish from malicious behavior. Windows API calls such as CreateRemoteThread, SuspendThread/SetThreadContext/ResumeThread, QueueUserAPC/NtQueueApcThread, and those that can be used to modify memory within another process, such as VirtualAllocEx/WriteProcessMemory, may be used for this technique. Monitoring for Linux specific calls such as the ptrace system call should not generate large amounts of data due to their specialized nature, and can be a very effective method to detect some of the common process injection methods
		Process Access	Process Access Monitor for processes being viewed that may inject code into processes in order to evade process-based defenses as well as possibly elevate privileges.
		Process Metadata	Process Metadata Monitor for process memory inconsistencies, such as checking memory ranges against a known copy of the legitimate module
		Process Modification	Monitor for changes made to processes that may inject code into processes in order to evade process-based defenses as well as possibly elevate privileges.

MITRE ATT&CK
Techniques (T1110) SSH Brute Force

Procedure Examples

ID	Name	Description
G0007	APT28	APT28 can perform brute force attacks to obtain credentials
S0220	Chaos	Chaos conducts brute force attacks against SSH services to gain initial access.
C0022	Operation Dream Job	During Operation Dream Job, Lazarus Group performed brute force attacks against administrator accounts

Mitigations

ID	Mitigations	Description
M1036	Account Use Policies	Set account lockout policies after a certain number of failed login attempts to prevent passwords from being guessed. Too strict a policy may create a denial of service condition and render environments un-usable, with all accounts used in the brute force being locked-out. Use conditional access policies to block logins from non-compliant devices or from outside defined organization IP ranges.
M1032	Multi-factor Authentication	Use multi-factor authentication. Where possible, also enable multi-factor authentication on externally facing services.
M1027	Password Policies	Refer to NIST guidelines when creating password policies.
M1018	User Account Management	Proactively reset accounts that are known to be part of breached credentials either immediately, or after detecting bruteforce attempts.

Detection

ID	Data Source	Data Source Data Component	Detects
DS0015	Application Log	Application Log Content	Monitor authentication logs for system and application login failures of Valid Accounts. If authentication failures are high, then there may be a brute force attempt to gain access to a system using legitimate credentials.
DS0017	Command	Command Execution	Monitor executed commands and arguments that may use brute force techniques to gain access to accounts when passwords are unknown or when password hashes are obtained.
DS0002	User Account	User Account Authentication	Monitor for many failed authentication attempts across various accounts that may result from password spraying attempts. It is difficult to detect when hashes are cracked, since this is generally done outside the scope of the target network.

To handle **SQL** and **Brute Force**, you can simply use the **Wazuh** or **TheHIve** tools.
Wazuh config for attack responses for rules using **id 1677278005.965906**
Config **TheHive WebHook** for notifications to the **SOC L1** Team **E-mail** to make it easier
IoCs can use **IRIS** tools.

Because the title of the report is only alert analyst for **SOC L1**, here I am not explaining how to respond to the attack. This alert report is sent to **SOC L2** as Incident Responses for responses to **SQL** attacks, **Brute Force**.

Thank You
Regards,

Mochammad Arif Rizki