

# Lab 5.1 Wazuh WAF

💡 In this lab we are going to augment web01 by adding a web application firewall (WAF). The wazuh agent should currently be able to forward apache error logs so a good deal of our work is done for us already. We are then going to run malicious http requests against web01 to see how our WAF performs.

## Adding software to web01

💡 web01's ability to talk to the WAN and the WANs ability to talk to web01 might be currently restricted. Updating and patching the server is one of the things we must do from time to time. VYOS itself cannot filter by domain name such as allowing traffic to updates.centos.org. It has to be by IP address or subnet. For this reason, many organizations go to an internal mirror for this purpose. We will use a work around.

## WAN-to-DMZ

If not already present, we need to add a new permanent rule to vyos such that established connections from the DMZ-to-WAN are allowed back through the WAN-to-DMZ firewall. If that rule (typically rule 1) is not there, add it.

## DMZ-to-WAN

Again, we may need to add a temporary rule for software updates that we either delete, disable or discard when complete. This rule should have the following characteristics.

- Set the rule number to 999 or similar
- Set the action to accept (this is wide open)
- Set the source ip address to be the web server

## Adding mod\_security, the core rule set and php to web01

There are far too many inaccurate guides on mod\_security out there, so please just use the following unless you have done this a bunch of times before. The following command will install mod\_security, the core ruleset associated with this layer 7 firewall and the php necessary to make a webshell work.

```
sudo yum install mod_security mod_security_crs php php-common php-opcache  
php-cli php-gd php-curl php-mysqlnd -y
```

Once the installation has worked, make sure to delete, disable or discard rule 999 if you created one. Make sure to commit so that we are locked down once again.

Updated Sep 24, 2023

Deliverable 1. Restart httpd on web01. Provide two screenshots similar to the ones below that shows that the security2\_module is loaded.

```
[hermione@web01-hermione ~]$ sudo httpd -M | grep security2
AH00558: httpd: Could not reliably determine the server's fully qualified domain
name, using fe80::1222:a0b6:bd93:5076. Set the 'ServerName' directive globally
to suppress this message
security2_module (shared)
[hermione@web01-hermione ~]$
```

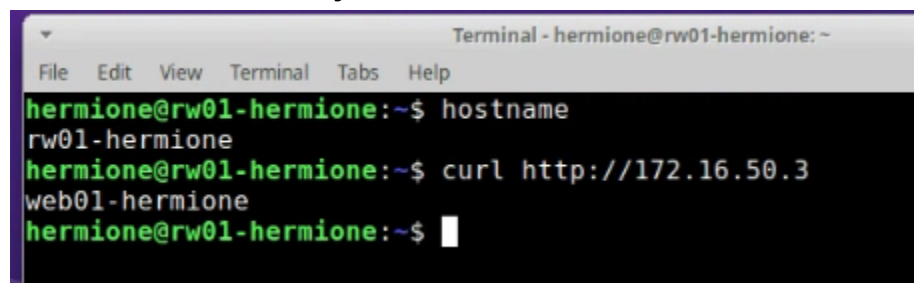
For the next one - you need to find the Apache (aka httpd) "error\_log"

```
[root@web01-hermione httpd]# cat error_log | grep ModSecurity
[Fri Sep 23 17:12:03.865329 2022] [:notice] [pid 3374] ModSecurity for Apache/2.
9.2 (http://www.modsecurity.org/) configured.
[Fri Sep 23 17:12:03.865344 2022] [:notice] [pid 3374] ModSecurity: APR compiled
version="1.4.8"; loaded version="1.4.8"
[Fri Sep 23 17:12:03.865351 2022] [:notice] [pid 3374] ModSecurity: PCRE compile
d version="8.32 "; loaded version="8.32 2012-11-30"
[Fri Sep 23 17:12:03.865359 2022] [:notice] [pid 3374] ModSecurity: LUA compiled
version="Lua 5.1"
[Fri Sep 23 17:12:03.865364 2022] [:notice] [pid 3374] ModSecurity: LIBXML compi
led version="2.9.1"
[Fri Sep 23 17:12:03.865368 2022] [:notice] [pid 3374] ModSecurity: Status engin
e is currently disabled, enable it by set SecStatusEngine to On.
[root@web01-hermione httpd]#
```

## Testing ModSecurity

Deliverable 2. Provide a screenshot showing that you can get to web01 from rw01. Make sure you show you are on rw01 (hostname). Depending on the default rules, modsecurity may not allow you to browse by IP. If this happens, figure out how to allow IP's in urls.

**NOTE: If Web01 returns the sample Apache page - create a custom index.html file in /var/www/html that has your name, SEC-350, and Web-01 identified.**

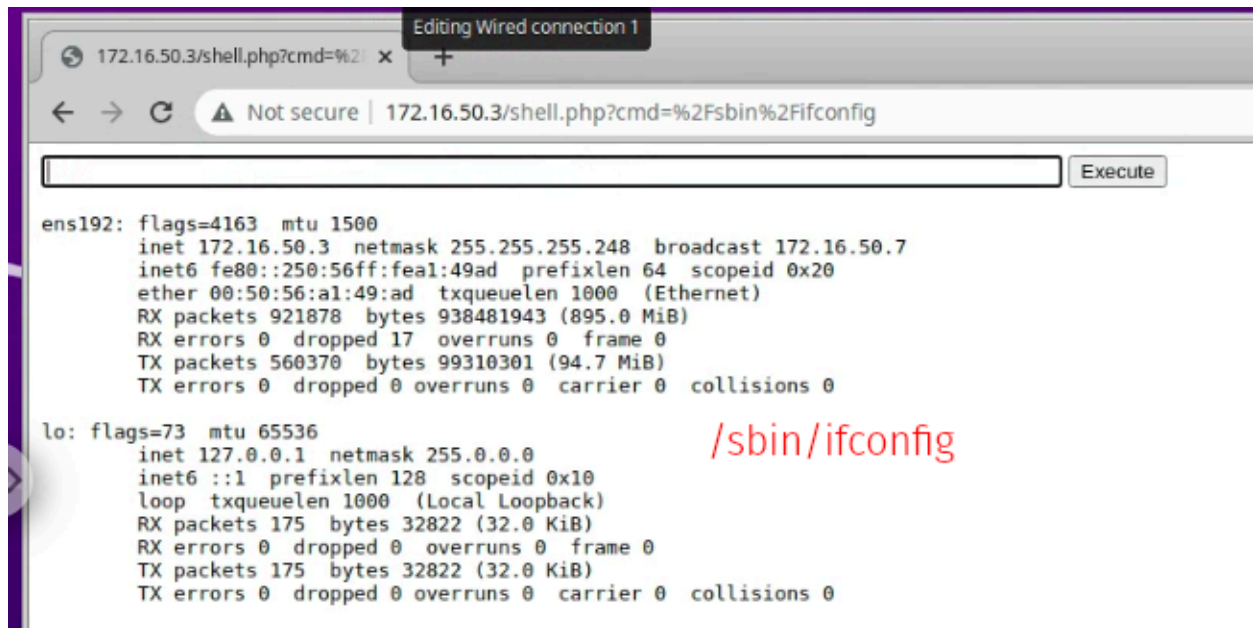


```
Terminal - hermione@rw01-hermione: ~
File Edit View Terminal Tabs Help
hermione@rw01-hermione:~$ hostname
rw01-hermione
hermione@rw01-hermione:~$ curl http://172.16.50.3
web01-hermione
hermione@rw01-hermione:~$
```

Download a php webshell to /var/www/html/shell.php on web01. I used this [one](#) but you are welcome to try your own (see your Tech Journal from SEC-260).

Updated Sep 24, 2023

Deliverable 3. Use the php webshell to execute a command like `ifconfig`, `hostname` or `whoami`. Provide a screenshot showing remote code execution like the one below. You may need to be explicit about the path of the program you wish to run



Deliverable 4. provide a screenshot that shows what happens when you attempt to run the following command within your webshell.

```
cat /etc/passwd
```

Deliverable 5. Find the error or warning associated with Deliverable 4 in the apache error\_log. Provide a screenshot.

Deliverable 6. Find the same alert in wazuh, provide a screenshot similar to the one below. You will see two types of events. 404 events directly from the apache logs but you should see some modsecurity events as well.

able JSON

```
{
  "_index": "wazuh-alerts-4.x-2023.02.11",
  "agent.id": "001",
  "agent.ip": "172.16.50.3",
  "agent.name": "web01-hermione",
  "data.id": "403",
  "data.protocol": "GET",
  "data.srcip": "10.0.17.100",
  "data.url": "/shell.php?cat%20/etc/passwd",
  "decoder.name": "web-accesslog",
  "full_log": "10.0.17.100 - - [11/Feb/2023:17:28:42 -0500] \"GET /shell.php?cat%20/etc/passwd HTTP/1.1\" 403 199 \"-\" \"Mozilla/!me/101.0.4951.64 Safari/537.36\"",
  "id": "1676154524.57576",
  "input.type": "log",
  "location": "/var/log/httpd/access_log",
  "manager.name": "wazuh-hermione"
}
```