DEVO

Search the community ...

Introducing the Devo Cyber Information Model: Going Live in March

‹ Use Cases

How to Create a Denial of Service (DoS) Detection Alert

7 months ago • 3 replies

yul pertierra

Requirement	Security Operations Application

Description

A [denial-of-service (DoS) attack](#) **occurs when legitimate users are unable to access information systems, devices, or other network resources due to the actions of a malicious cyber threat actor**. Services affected may include email, websites, online accounts (e.g., banking), or other services that rely on the affected computer or network. A denial-of-service condition is accomplished by flooding the targeted host or network with traffic until the target cannot respond or simply crashes, preventing access for legitimate users.

There are many different methods for carrying out a DoS attack. **The most common method of attack occurs when an attacker floods a network server with traffic**. In this type of DoS attack, the attacker sends several requests to the target server, overloading it with traffic.

In this article, you will find a way to prevent possible DOS by grouping and analyzing activity by source IP. For this purpose, you will create an alert so that you can know when there are more than 5000 requests per hour.

Prerequisites

web.all.access Table

- You will be querying the `web.all.access` table.
- Here you can find an example of a log in the `web.all.access` table.

> Notice that the source IP corresponds to the ***yourIp*** field.

Field Name	**Example Value**
eventdate	2020-01-15 13: 43: 01.415
source	apache
environment	web
site	pro
clone	www-ssl
serverdate	2018-05-14 08: 31: 30,000
yourIp	8.8.8.8
method	GET
url	/dists/Packages.html
protocol	HTTP / 1.1
statusCode	404
referer	-
userAgent	Debian APT-HTTP / 1.3 (1.0.1ubuntu2)
user	-
serverName	repository.example.net
serverPort	80
cookies	-: -

requestLength	217
responseLength	410
responseTime	554

Creating Lookup Tables

In order to create an alert on your query, you must create two lookup tables as part of that query. Here are two examples:

Lookup SecOpsAlertDescription Example Registration

Lookup SecOpsLocation Example Registration

Field Name	Example Value
Hostname	8.8.8.8
City	Santa Clara
State	CA
Country	US
Lat	37.389437
Lon	-121.962005

SecOps Alert Query Example

Here is the query:

```
from web.all.access
where ispublic(IPField)
where isnotnull(IPField)
group every 30m by IPField
every 1h
select count() as count
where count > 5000
select str(IPField) as entity_sourceIP
select `lu/SecOpsLocation/country`(entity_sourceIP) as
```

```
enrichStream_entity_sourceIP_locationCountry
select `lu/SecOpsLocation/city`(entity_sourceIP) as
enrichStream_entity_sourceIP_locationCity
select `lu/SecOpsLocation/state`(entity_sourceIP) as
enrichStream_entity_sourceIP_locationState
select `lu/SecOpsLocation/lat`(entity_sourceIP) as
enrichStream_entity_sourceIP_locationLat
select `lu/SecOpsLocation/lon`(entity_sourceIP) as
enrichStream_entity_sourceIP_locationLon
select `lu/SecOpsAlertDescription/alertType`("SecOpsDenialOfService") as
alertType
select `lu/SecOpsAlertDescription/alertMitreTactics`("SecOpsDenialOfService") as
alertMitreTactics
select `lu/SecOpsAlertDescription/alertMitreTechniques`("SecOpsDenialOfService")
as alertMitreTechniques
select `lu/SecOpsAlertDescription/alertPriority`("SecOpsDenialOfService") as
alertPriority
```

And here you can find a brief explanation of each part:

Part of the Query	Explanation
`from web.all.access`	Select the Devo table ***web.all.access***
`where ispublic (IPField)`	Verify that the source IP is public using queries language function.
`where isnotnull (IPField)`	Check that the source IP is not null usir queries language function.
`every 30m group by IPField`	Group all the above fields to be display the alert every hour.
`select count () as count`	Count how many occurrences there ar

`where count> 5000`	If they are greater than *5000* generate
`select str(IPField) as entity_sourceIP`	Create an alias for field **IPField** calling **entity_sourceIP** since we want the val to be enriched with the services Sightir GreyNoise, Enigma, Misp, and DomainT

``select 'lu/SecOpsLocation/country` (entity_sourceIP) as enrichStream_entity_sourceIP_locationCountry``
``select `lu/SecOpsLocation/city` (entity_sourceIP) as enrichStream_entity_sourceIP_locationCity``
``select `lu/SecOpsLocation/state` (entity_sourceIP) as enrichStream_entity_sourceIP_locationState``
``select `lu/SecOpsLocation/lat` (entity_sourceIP) as enrichStream_entity_sourceIP_locationLat``
``select `lu/SecOpsLocation/lon` (entity_sourceIP) as enrichStream_entity_sourceIP_locationLon``
	Obtain the geolocation of the value of **entity_sourceIP** of the lookup SecOpsl
	These fields should be called obligato
	• *enrichStream_entity_sourceIP_l*
	• *enrichStream_entity_sourceIP_l*
	• *enrichStream_entity_sourceIP_l*
	• *enrichStream_entity_sourceIP_l*

	Get the alert type, MITRE Tactic, MITRE T Priority with the name we will give to th case **SecOpsDenialOfService)**. **These fields should be called obligato** • *alertType* • *alertMitreTactics* • *alertMitreTechniques* • *alertPriority*

Alerts Security Operations Denial of Service AdvKnowledge