Communication over unsecured UDP

Many UDP communications are susceptible to amplification attacks. You might want to monitor communication over UDP on your network for the following situations:

- Detecting UDP service amplification abuse

Prerequisites

In order to execute this procedure in your environment, the following data, services, or apps are required:

- Splunk Enterprise or Splunk Cloud Platform
- Network protocol data

Example

You want to compare bytes sent to bytes received by your memcached server to determine whether the payload was amplified, which might indicate an attempt at a Denial of Service attack.

To optimize the search shown below, you should specify an index and a time range. In addition, these sample searches use AWS VPC Flow logs and Splunk Stream. You can replace this source with any other network protocol data used in your organization.

Option 1

Run the following search:

```
sourcetype=aws:cloudwatchlogs:vpcflow (src_port=<memcached server port number> OR dest_port=<memcached server port number>)
|head 4
|table _time duration account_id region interface_id src src_port dest dest_port bytes protocol packets vpcflow_action
```

Search explanation

The table provides an explanation of what each part of this search achieves. You can adjust this query based on the specifics of your environment.

The information provided in Splunk Lantern is intended for informational and educational purposes only. All information is provided in good faith, however, Splunk disclaims any and all representations and warranties, express and implied, regarding the information provided, including without limitation any warranties and representations regarding the completeness, adequacy or accuracy of the information. You agree to take full responsibility for the results arising from the use of the information provided.
**Splunk Search**

```
sourcetype=aws:cloudwatchlogs:vpcflow
```

**Explanation**

Search only AWS CloudWatch VPCFlow data.

```
(src_port=<memcached server port number> OR dest_port=<memcached server port number>)
```

**Return results originating from or going to the memcached server.**

```
|head 4
```

**Return the 4 most recent events.**

```
VCPFlows are collections of unidirectional packets. Therefore, 4 results represent 2 network connections. You can adjust this number as needed.
```

```
|table _time duration account_id region interface_id src src_port dest dest_port bytes protocol packets vpcflow_action
```

**Display the results in a table with columns in the order shown.**

External hosts cannot communicate with an AWS-hosted RFC1918 address, so the actual network communications must have come through a public IP address such as a load balancer.

**Option 2**

If you already know what port number the external host uses, use this search.

```
sourcetype=stream:udp (src_port=<client port number> OR src_port=<another client port number> OR src_port=<memcached server port number>)
|head 2
|eval short_src_content=substr(src_content,1,75)
|eval short_dest_content=substr(dest_content,1,75)
|table _time bytes_in bytes_out src src_port dest dest_port short_src_content short_dest_content
```

**Search explanation**

The table provides an explanation of what each part of this search achieves. You can adjust this query based on the specifics of your environment.
**Splunk Search**

(src_port=<client port number> OR src_port=<another client port number> OR src_port=<memcached server port number>)

**Explanation**

Return results originating from the external host or the memcached server.

You can add as many client port numbers to this search as needed.

|head 2

Return the 2 most recent events.

Stream data shows a complete network connection, bytes sent and received.

|eval short_src_content=substr(src_content,1,75)

Return characters 1 through 75 of the src_content string, and rename the column as short_src_content.

|eval short_dest_content=substr(dest_content,1,75)

Return characters 1 through 75 of the dest_content string, and rename the column as short_dest_content.

|table _time bytes_in bytes_out src src_port dest dest_port short_src_content short_dest_content

Display the results in a table with columns in the order shown.

---

**Result**

Compare the bytes sent (bytes_out) to the bytes received (bytes_in) to see whether the payload has been amplified. For example, if 50 bytes were sent, but 51327 bytes were received, there is an amplification factor of almost 1027. Note that the byte values will vary slightly between the two search options provided in this guide due to IP overhead, such as packet options.

Save this search to continue to monitor for packet size disparities. You might also want to monitor packet count disparities.