### **Network Topology**

*TODO: Fill out the information below.*

The following machines were identified on the network:

* ELK
  + **Operating System**: Linux
  + **Purpose**: Kibana Server
  + **IP Address**: 192.168.1.100
  + **Operating System**: Ubuntu - Linux
  + **Purpose**:
  + **IP Address**: 192.168.1.105
* Capstone
  + **Operating System**: Linux
  + **Purpose**: Apache server
  + **IP Address**: 192.168.1.115
* Target 1
  + **Operating System**: Debian - Linux
  + **Purpose**: Wordpress server
  + **IP Address**: 192.168.1.110

### **Description of Targets**

**The target of this attack was:** Target 1. The Debian machine at 192.168.1.110

**Target 1 is an Apache web server and has SSH enabled, so ports 80 and 22 are possible ports of entry for attackers. As such, the following alerts have been implemented:**

The SSH port should have logging for the local machine, and be pushing all these logs to who attempts to connect through this port to the ELK stack.

The HTTP port should also have logging enabled for what is being requested on the server itself. All GET/PUT/PUSH and other HTTP codes should have alerts for too many, or ones that breach the baseline.

There should be a alert set up that indicates if there was any traffic that is sent or received by the machine on port 22 and 80 that is not HTTP or SSH traffic.

### **Monitoring the Targets**

**Traffic to these services should be carefully monitored. To this end, we have implemented the alerts below:**

#### **Increased HTTP requests**

Alert 1 is implemented as follows:

* **Metric**: destination.port:80
* **Threshold**: 2,000
* **Vulnerability Mitigated**: This will increase the response time to DDoS attacks from increased HTTP queries. This will also show if there is any malicious activity where the bad actor is connecting via port 80.
* **Reliability**: There is a possibility for increased false positives that could be caused by increased traffic from actual customers. Though with basic investigation it should be easily identified.

#### **HTTP error codes**

Alert 2 is implemented as follows:

* **Metric**: http.response.status\_code:404
* **Threshold**: 1,500
* **Vulnerability Mitigated**: Attacker seemed to have “walked the dog” through the apache site while looking for hidden directories or pages that should not be visible.
* **Reliability**: With a higher threshold it will have low false positives. And the incorrect alerts that are received can show an issue with the site itself that will need to be resolved as well.

#### **SSH connection packets**

Alert 3 is implemented as follows:

* **Metric**: destination.port : 22
* **Threshold**: 400
* **Vulnerability Mitigated**: Allowing bad actors to connect to the server either by SSH or another malicious mean. This does not allow them to stay connected for long before being noticed.
* **Reliability**: Can be reliable, though may trigger many false positives if there needs to be work done on the machine. A notification in advance to the infosec team would allow them to disable the alert for a short period while maintenance is being performed.