

Example of how an IDS protects you Network:

Snort (the NIDS) has a feature called flexible response. Essentially, it can terminate connections that fit a certain rule. More information on flexible response is available [here](#).

These are the rules I've added (they all pertain to codered/nimda):

- disabling incoming web access of cmd.exe (general command line access)

The above rule is a method commonly used by exploits to access Microsoft Windows Command line. The chances of these being false alerts are extremely slim (most likely 0%). This should essentially prevent any more machines running vulnerable version of IIS from becoming compromised by machines **outside** of the Network.

For anyone unfamiliar with wget, information about it is available [here](#). Wget allows you to make http requests... I'm going to use it to set off IDS by making a request for cmd.exe. Here's an example of flexible response in action:

Before enabling flexible response:

```
[root@scanner root]# wget http://XXX.XXX.XXX.XXX/cmd.exe
--17:23:20--  http://XXX.XXX.XXX.XXX/cmd.exe
           => `cmd.exe'
Connecting to XXX.XXX.XXX.XXX:80... connected!
HTTP request sent, awaiting response... 404 Not Found
17:23:20 ERROR 404: Not Found.
```

note: just ignore the "404: Not Found". I just wanted to prove that the connection was successfully established and the request for "cmd.exe" was successfully made.

After enabling flexible response:

```
wget http://XXX.XXX.XXX.XXX/cmd.exe
--17:26:02--  http://XXX.XXX.XXX.XXX/cmd.exe
           (try: 2) => `cmd.exe'
Connecting to XXX.XXX.XXX.XXX:80... connected!
HTTP request sent, awaiting response...
Read error (Connection reset by peer) in headers.
```

note: the connection is never established, so the request for cmd.exe is never completed.

IDS terminates the incoming request for cmd.exe, thus protecting the IIS server from becoming compromised.