**A Divide and Defend Firewall**

*Walter B. Gress V*

*Drexel University*

*Friday, June 7 2013*

This concept of operations details a “Divide and Defend” Firewall. Recently there has been a surge of malware attacks, those attacks typically being orchestrated through botnets. Modern enterprise and personal firewalls are not designed by default to handle massive number of attacks at once like botnets can incur. D&D is designed with that very thought in mind.

# introduction

As said in the abstract, botnets have become more and more of a threat. Instead of taking the offense in this situation, I suggest taking a lightweight approach to the botnet problem until the proper authority can be contacted or

the attacks (bots) are cut off from their ISP, or some other form naturally occurring termination. The lightweight procedures include a divide and conquer, modular, procedure, making the host’s firewall more difficult to crash the firewall. Also, it allows for a crashed component to be recovered and made up and functional once again.

# feature list and ITERATIVE PROPERTIES OF THE D&D FIREWALL

This section consists of a series of requirements for the D&D Firewall .

* The central concept of this piece of software is to have the firewall and its components in linked modules, making it harder to attack than one monolithic piece of software.
* The Firewall Core (hereto after referred to as the Core), the Main Firewall controlling software is isolated from the rest of the network, inaccessible through any outside ports.
* The D&D firewall uses micro or mini firewalls to block incoming attacks. These firewalls are designed to run lightweight threads and to be quick to spawn, despite performing all of the duties of a firewall. Being small and lightweight allows them to not only react quickly, there are multiple firewalls to handle multiple attackers, and should they crash they are lightweight enough that a new mini firewall can be thrown up again for the Core as quicker than would a typical firewall architecture.
* A data abstraction layer (DAL) exists between the Core and the “mini” firewalls that are created to defend against attacks. The Core spawns the micro-firewalls through the DAL and out into the web where they can block botnet attacks and act as “bait” in the sense that they will be the only exposed layer of the network to the rest of the web that can receive input as well as perform output. DAL is a one-way pipeline. The Core can allow spawning of the micro-firewalls through the DAL, but the micro-firewalls cannot communicate back to the Core.
* Each thread spawned through the DAL has firewall software running or “attached to” the thread.
* When one mini firewall goes down, the Core (main firewall) immediately respawns a mini-firewall through the DAL.
* Since the mini-firewalls take up little resources, they can be spawned, respawned, and so on again and again taking much effort to crash the host computer.
* When a mini-firewall is generated, it generates according to the hostname and port currently being attacked. If other hostnames and ports are attacked, additional mini firewalls are spawned.
* Since the Main Firewall (Core) software managing the mini firewalls is isolated, the enemy is unable to easily penetrate and attack the Core and in essence, for an analogy, it “swats away flies”, the flies being the bots from he botnet attempting atbreakinginto the system, the flyswatter(s) being the mini-firewalls.

# general structurE of firewall

The firewall consists of three primary things. 1) The Monolithic Core which is designed to be implemented as small and fast as possible.

2) The Data Abstraction Layer, meant to “confuse” the enemy malware by implementing the “proxy” like behaviour of a firewall. That is, it means it the front door so to speak, or gateway. The DAL is loosely coupled so that it may change its ip address of on the fly. Should a botnet be attacking 101.1268.1.1, the DAL will immediately shifts its ip address to another available (possibly stored similar to how proxy server software does present day).So now its ip address, whose only purpose is to proxy, sitting on the gateway, has changed, and the botnet no longer knows where its target has gone.

Also, as mentioned earlier, the Core of the firewall, that is, the most of the firewall functionality we administer we put in the Core, allowing the the mini/micro firewalls to be very lightweight.

As said, the Firewall spawns micro-firewall from monolithic core to block enemy attack on the dummy host ip address and then shifts the dummy ip address to another (these dummy ip addresses are the address info for the Data Abstraction Layer). If a new dummy host attack occurs, the main firewall generates another micro-firewall and shifts the DAL’s dummy ip, and so on….This part I’m not sure of how to implement: how to shift the DAL’s ip address. I considered adding another proxy, one proxy only outgoing and the other tying the two proxies together. Instead of changing the ip address of the DA, change the connection between the DAL and this new, internal proxy. That way should the attacker try to attack , he wouldn’t get beyond the dummy proxy. (DAL)

3. When not under attack, the DAL operates as a normal gateway between the internal and external networks.

# diagrams of system processes

## Illustration of firewall design



## Alternative Firewall Using Double Proxy







