Application Layer DDoS

A Practical Approach & Mitigation Techniques

Mohammad Fakrul Alam

bdHUB Limited

fakrul@bdhub.com

http://www.as58656.net

http://www.bdnog.org

Disclaimer

Tools used to demonstrate DDoS attack is for educational / knowledge sharing purpose only. No intention to generate DDoS attack on production network.

Agenda

- Background
- Application / Layer 7 DDoS
- Practical Approach (Case Study)
- Mitigation
- Simulation
- Key findings & Issues


pedobear
does he look like a bitch?
newfags can't triforce
4chan
battletoads?
boxxy
sametag
fist
uckroll
we are legion
shit tyrone
troll
tits or g
orly?
MOAR!
get it together
it's a tra
Anonymous
/b/
so i herd u liek mudkipz
.this
wat
none of us are as cruel as all of us
sau
is
sparta
lurk moar
oldfag
delete system32
forever alone
new
anon
moot
YARLY
noko
inb4
the game
desu
anhero
rollin'
newfags can't >greento
summerfag
mind=blown
you're doing wron
boom headshot
when you see it you'll shit brix
imma chargin' mah laz
C-C-Combo breaker!
oh noes
fuck yeah, seaking
OP is a
numa numa
the cake is a lie
ROFLcopter
raptor jesus
ceiling cat is watching you
lolno
hackers on stetoi
never gonna give you up
i see what you did there
how do i shot web?
in soviet russia
for the lulz
i came
buttsecks
i can has?
hurr durr
oh hai gu
nirite?
do want
you raff you ruse
shoop da woop
thats some delicious copypa
cool story bro
how is babby formed?
your argument is invalid
gentle
norlylurk less
a cat is fine too
gold membership require
bitches don't know 'bout my cock
i came buckets
newfags can't se

BACKGROUND

Background : What is DDoS

- Denial of Service (DoS) / Distributed Denial of Service (DDoS) is the act of performing an attack which prevents the system from providing services to legitimate users

- Denial of Service attacks take many forms, and utilize many attack vectors

- When successful, the targeted host may stop providing any service, provide limited services only or provide services to some users only

- DDoS attack sometime refer as Distributed Reflection Denial of Service (DrDoS) Attack

Background : DDoS Attack Phases

- Phase One: Target Acquisition
- Phase Two: Groundwork
- Phase Three: ATTACK

Background : About Botnets

A Botnet can generate **1 Million Times** the available bandwidth of a business

It takes just **64,000 PCs** infected with a virus like Conficker to generate **10 gigabits Per second** of traffic

Mariposa, the largest known Botnet, affected **12 million PCs** It could have generate a DDoS attack as large as **31.2 Terabytes Per second**

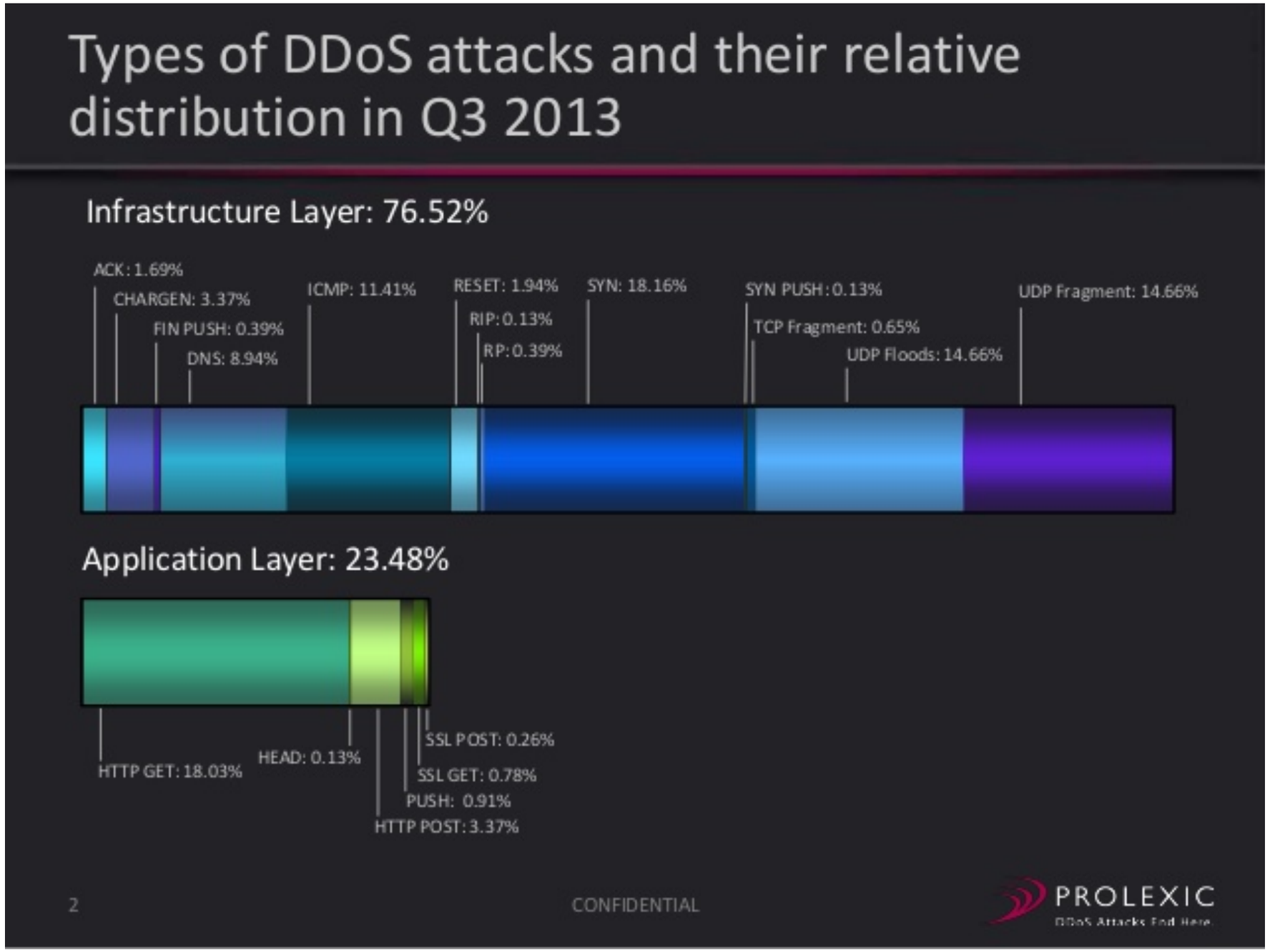
Background : DDoS Insurance

- Insurance is money you pay to be protected from
- Happen / Might Not Happen
- You can be prepared
  - Incident response plan
  - Tools
  - Gear
  - Partnerships
- It may not be sufficient – you should have picked the higher premium policy…

Background : Types of Attacks

- Volume Based Attacks
- Protocol Attacks
- Application Layer Attacks

Background : Statistics



Source: PROLEXIC Q3 2013 Attack Report

Background : DDoS Attack Surface

- Past DDOS attacks were mainly Layer 3 / Layer 4 attacks

Layer 7 Application	Application Telnet / FTP / NFS / WWW
Layer 6 Presentation	Session RPC
Layer 5 Session	Transport Sockets / Streams
Layer 4 Transport	TCP / UDP
Layer 3 Network	Network IP + ARP / RARP / ICMP
Layer 2 Data Link	Physical Protocol Ethernet / FDD / PPP
Layer 1 Physical	Transmission Media Coax / Fiber

Layer 3 DDoS Attack

- Layer 3 - muscle-based attacks
- Flood of TCP/UDP/ICMP/IGMP packets, overloading infrastructure due to high rate processing/discarding of packets and filling up the packet queues, or saturating pipes
- Introduce a packet workload most gear isn't designed for
- Example - UDP flood to non-listening port

Layer 4 DDoS Attack

- Layer 4 – slightly more sophisticated
- DoS attacks consuming extra memory, CPU cycles, and triggering responses
  - TCP SYN flood
  - TCP new connections flood
  - TCP concurrent connections exhaustion
  - TCP/UDP garbage data flood to listening services (ala LOIC)
- Example – SYN flood

Layer 7 DDoS

- Layer 7 - The Evil
- DoS attacks abusing application-server memory and performance limitations – masquerading as legitimate transactions
  - HTTP page flood
  - HTTP bandwidth consumption
  - DNS query flood
  - SIP INVITE flood
  - Low rate, high impact attacks – e.g. Slowloris, HTTP POST DoS

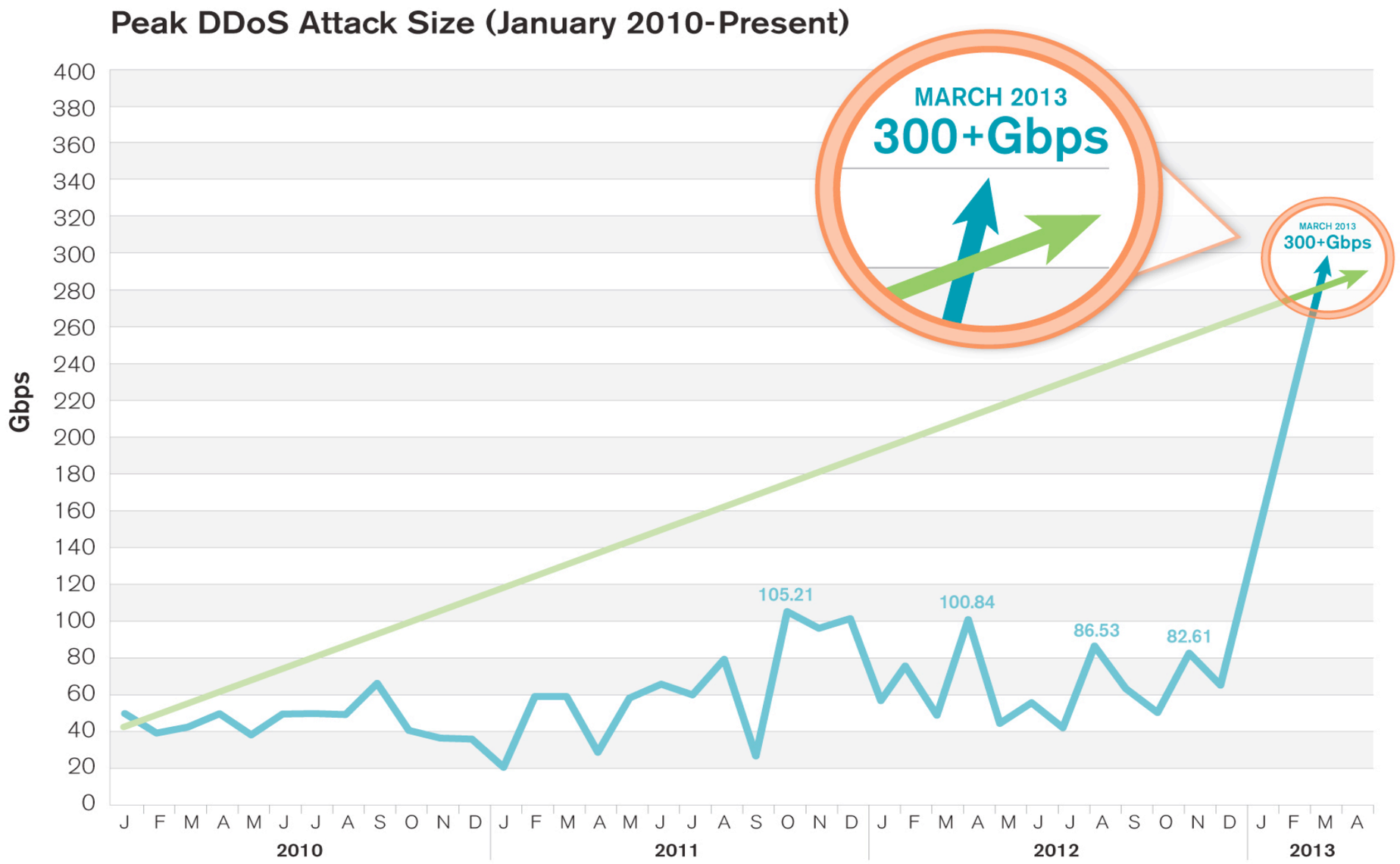
Background : DDoS Attack Costs

Damage to Your Brand	Loss of Revenue	Bad Customer Exeprience
If your site is down, account holders will question if you provide safe service	If your website is down, you loos revenue	Call center agent get overwhelmed
Ruins years of work building your brand	No online banking, bill pay, forms or application, account opening	Account holders frustration
		People seek alternatives

Background : DDoS Additional Threat

- DDoS attacks are more frequently being used to hide security breaches and data theft
    - Attention focuses on attack
    - Log files get massive, to difficult to analyze quickly
    - Servers and routers are rebooted, often destroying forensic evidence
    - Attacks end long before any intrusion is identified

Background : DDoS Attack Volume



Source: Arbor Networks, Inc.

ARBOR NETWORKS

Background : DDoS Attack Volume (DNS)

1 Attacker Laptop controlling

+ 5-7 compromised server on

+ 3 network allowed spoofing of

```
dig ANY isoc.org @OpenResolverIP
   +edns=0 +notcp +bufsize=4096
```

+ 9 Gbps DNS request to

+ 0.1% of open resolvers resulted in

= 300 Gbps+ DDoS attack traffic

[source: cloudflare]

Background : DDoS Attack Volume (DNS)

```
fkfkfkfa.com.           84930 IN    A     204.46.43.27
fkfkfkfa.com.           84930 IN    A     204.46.43.28
fkfkfkfa.com.           84930 IN    A     204.46.43.29
fkfkfkfa.com.           84930 IN    A     204.46.43.30
fkfkfkfa.com.           84930 IN    A     204.46.43.31
fkfkfkfa.com.           84930 IN    A     204.46.43.32
fkfkfkfa.com.           84930 IN    A     204.46.43.33
fkfkfkfa.com.           84930 IN    A     204.46.43.34
fkfkfkfa.com.           84930 IN    A     204.46.43.35
fkfkfkfa.com.           84930 IN    A     204.46.43.36
fkfkfkfa.com.           84930 IN    A     204.46.43.37
fkfkfkfa.com.           84930 IN    A     204.46.43.38
fkfkfkfa.com.           84930 IN    A     204.46.43.39
fkfkfkfa.com.           84930 IN    A     204.46.43.40
fkfkfkfa.com.           84930 IN    A     204.46.43.41
fkfkfkfa.com.           84930 IN    A     204.46.43.42
fkfkfkfa.com.           84930 IN    A     204.46.43.43
fkfkfkfa.com.           84930 IN    A     204.46.43.44
fkfkfkfa.com.           84930 IN    A     204.46.43.45
fkfkfkfa.com.           84930 IN    NS    us3.fkfkfkfa.com.
fkfkfkfa.com.           84930 IN    A     204.46.43.46
fkfkfkfa.com.           84930 IN    NS    us4.fkfkfkfa.com.

;; AUTHORITY SECTION:
fkfkfkfa.com.           84930 IN    NS    us3.fkfkfkfa.com.
fkfkfkfa.com.           84930 IN    NS    us4.fkfkfkfa.com.
```

```
dig ANY fkfkfkfa.com @103.12.178.XXX
+edns=0 +notcp +bufsize=4096
```

4002/64 = 62x amplification

```
;; Query time: 83 msec
;; SERVER: 103.12.178.xxx#53(103.12.178.xxx)
;; WHEN: Sat Dec 28 17:46:24 2013
;; MSG SIZE  rcvd: 4002
```

Background : DDoS Attack Volume (NTP)

```
77.68.201.102           32823 198.123.30.132        1 3 4    190      29       29
117.195.46.81           20609 198.123.30.132       12 3 3    190      29       29
87-114-180-251.ether      123 198.123.30.132        4 3 4    190      29       29
d54C0EAF1.access.telen    123 198.123.30.132        1 3 4    190      29       29
77.68.132.75             1060 198.123.30.132        1 3 4    190      29       29
187-162-58-152.static.    123 198.123.30.132        1 3 4    190      29       29
ynrfw23001.yr.com         480 198.123.30.132       13 3 3    590       4       29
115-39-41-60.mie1.comm   4913 198.123.30.132        1 3 3    190      29       29
primary.idb.com.au        199 198.123.30.132        1 3 4    190      29       29
206-248-138-67.dsl.tek    123 198.123.30.132        1 3 4    190      29       29
pool-173-66-167-144.wa  60137 198.123.30.132        2 3 1    590      14       29
dslsonicwall.btconline  23891 198.123.30.132        1 3 1    190      30       30
108-192-128-52.lightsp    123 198.123.30.132        1 3 4    190      30       30
static-50-44-202-198.o  11547 198.123.30.132        1 3 1    190      30       30
117.136.19.162          21829 198.123.30.132        1 3 3    190      30       30
cm68.eta123.maxonline.    123 198.123.30.132        1 3 4    190      30       30
89.169.64.236             123 198.123.30.132        6 3 4    190       6       30
121.61-66-87.adsl-dyn.    123 198.123.30.132        1 3 4    190      30       30
pool-98-118-84-72.bstn    123 198.123.30.132        1 3 4    190      30       30
218.56.20.142            1100 198.123.30.132        1 3 3    190      30       30
66.162.156.150          41755 198.123.30.132        4 3 4    190      10       30
173-11-46-77-Minnesota   4847 198.123.30.132        2 3 4    190      17       30
61-195-152-9.cust.bit-    123 198.123.30.132        1 3 4    190      31       31
rionicapital1.pndsl.co  50071 198.123.30.132        1 3 3    190      31       31
vespa.ndc.nasa.gov        123 198.123.30.132        1 3 4    190      31       31
host87-93-dynamic.183-  32850 198.123.30.132        1 3 3    190      31       31
odin000956850.ndc.nasa    123 198.123.30.132        1 3 4    190      31       31
241.wolainfo.com.pl       123 198.123.30.132        1 3 3    190      31       31
112.90.239.142          43793 198.123.30.132        1 3 3    190      31       31
cpe-071-070-131-156.nc    123 198.123.30.132        1 3 4    190      31       31
88.230.191.53.dynamic.  23460 198.123.30.132        1 3 3    190      31       31
ARouen-652-1-378-137.w  28247 198.123.30.132        2 3 3    190      16       31
77.68.194.63            32825 198.123.30.132        1 3 4    190      32       32
77.68.154.162            1042 198.123.30.132        1 3 4    190      32       32
```

```
ntpdc -c monlist ntp.nasa.gov
```

Background : DDoS Attack Volume (NTP)

Public Safety Canada

Canada

Find what you're looking for.. Search

National Security | Border Strategies | Countering Crime | Emergency Management | Resources

Home > Resources > Cyber Security Bulletins > AV14-001: Network Time Protocol Vulnerability

Resources

- Acts and Regulations
- Frequently Asked Questions
- Canadian Disaster Database
- Cyber Security Bulletins
- Help
- Library and Information Centre
- News Releases
- Publications and Reports

Network Time Protocol Vulnerability

Number: AV14-001
Date: 08 January 2014

Purpose

The purpose of this advisory is to bring attention to a vulnerability in the NTP service that could be used in a reflection/amplification distributed denial of service (DDoS) attack.

Assessment

CCIRC is aware of a vulnerability in older versions of the NTP service that could allow a remote attacker to use the device unknowingly in a reflection/amplification DDoS attack. Subverted devices used in these attacks are not the ultimate target, but are unknowing accomplices to a DDOS attack on an external system.

CVE Reference: CVE-2013-5211

Suggested action

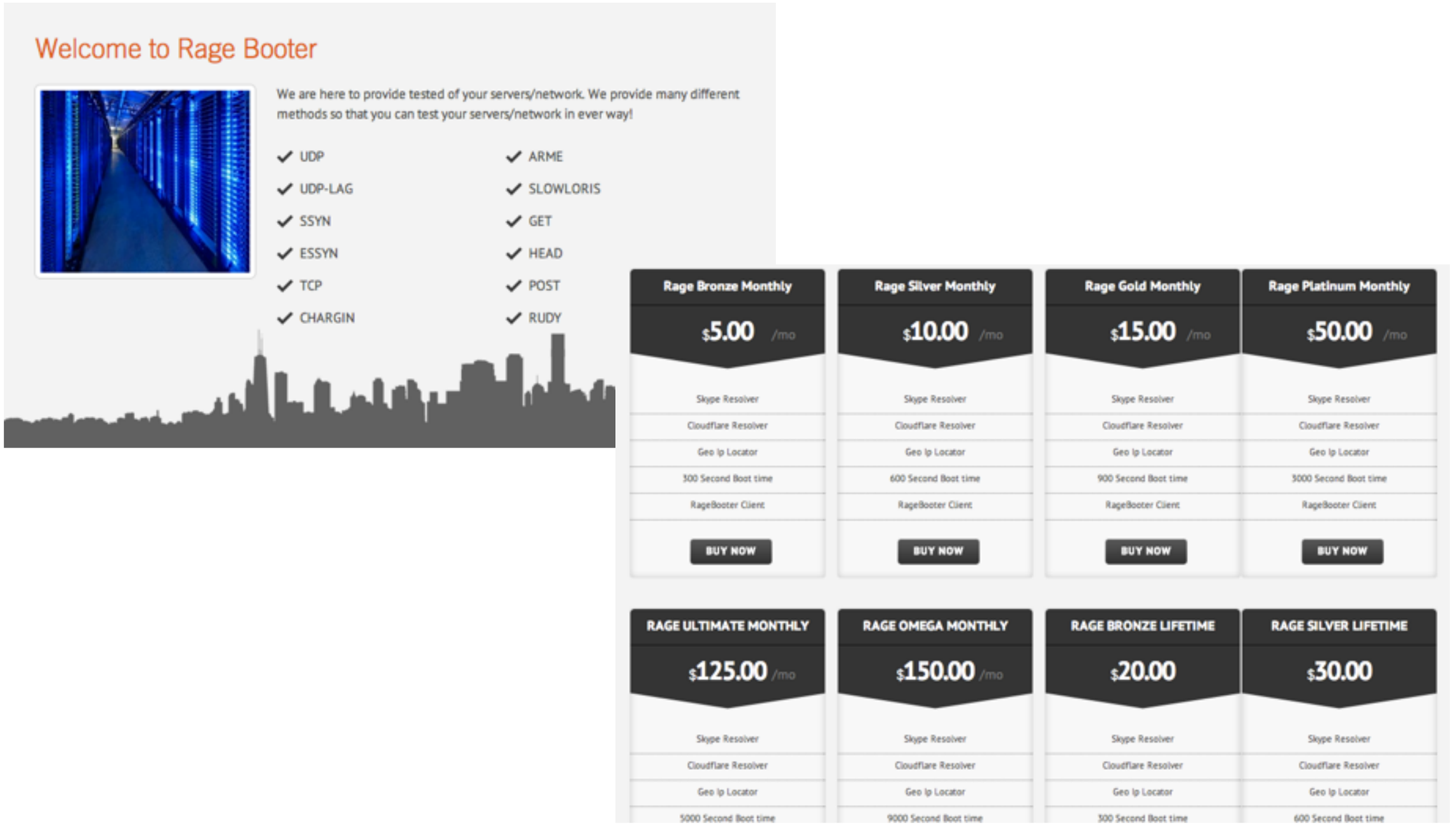
CCIRC recommends organizations review common best practices to harden NTP servers, consider disabling the service if it is not required or test and deploy the latest version of NTP.

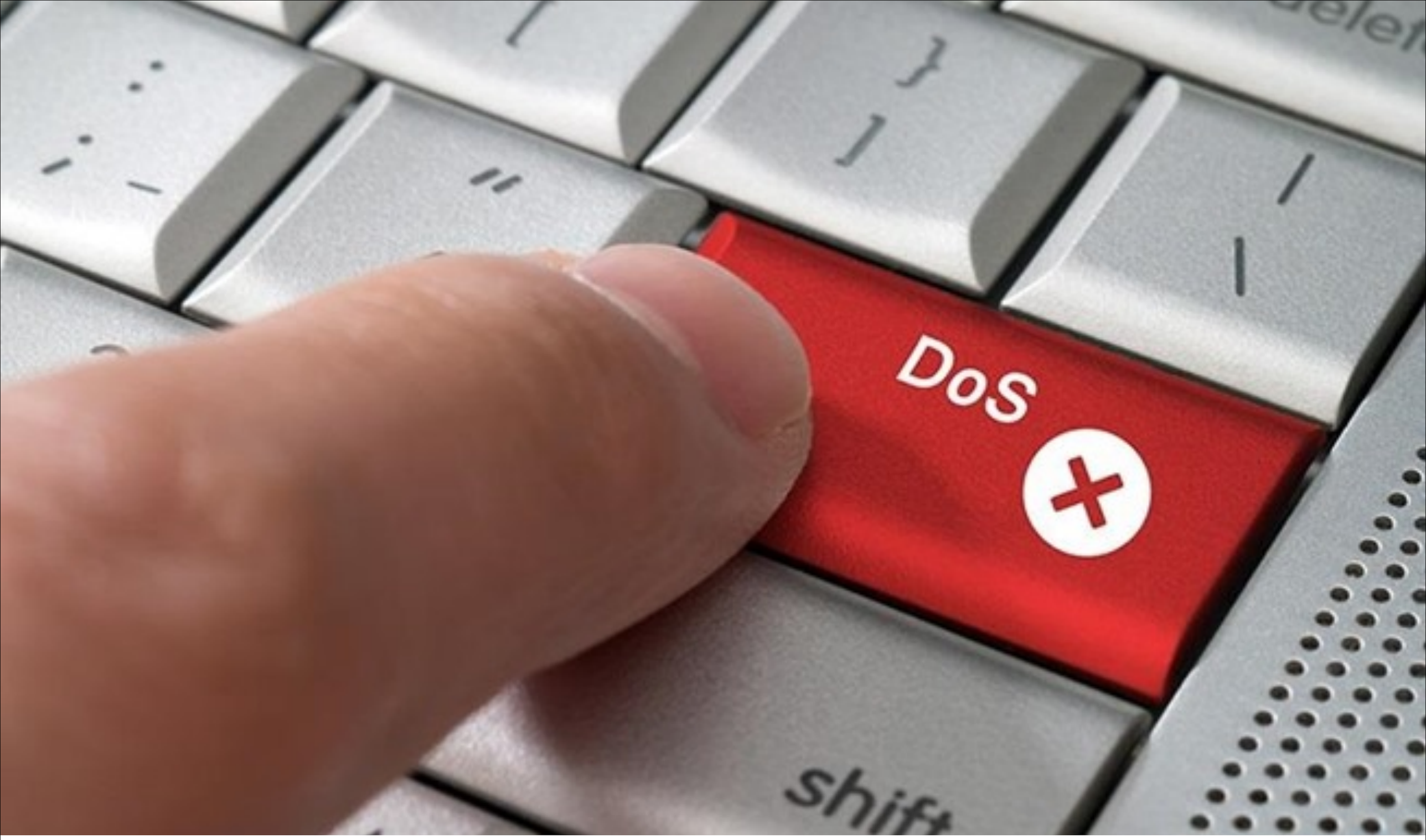
Additional guidance on NTP hardening can be found at the following reference:
http://www.team-cymru.org/ReadingRoom/Templates/secure-ntp-template.html

Background : DDoS Attack Volume

- Source Address Spoofing
    - Ingress Filtering / BCP 38 (http://tools.ietf.org/html/bcp38)
- Secure DNS
    - http://www.cymru.com/Documents/secure-bind-template.html
- Secure NTP
    - http://www.team-cymru.org/ReadingRoom/Templates/secure-ntp-template.html

Background : DDoS or Network Stress Test!





LAYER 7 DDOS

Layer 7 DDoS : Overview

- Application layer DoS attacks are evolving as part of the evolution of application attacks
- The denied service is the application itself (rather than the host) – effectively preventing usage of the system.
- Take advantage of flaws in the code to perform the DoS
- The benefit for the attacker – does not require the same effort to achieve as a DDoS attack

Layer 7 DDoS : Overview

DoS can be achieved in various ways:

- Application Crashing
- Data Destruction
- Resource Depletion

Layer 7 DDoS : Application Crashing

- Common way of performing a Denial of Service attack
- In many cases, certain types of inputs may yield an error in the application which it did not anticipate, and will cause it to crash:
  - Buffer Overflows
  - Malformed data – causing parser exception
  - Terminating with error
  - SQL Injection

Layer 7 DDoS : Data Destruction

- One way to cause a DoS attack is by tampering with the data instead of the service itself

- If a site is vulnerable to SQL Injection, for instance, it may be possible to DELETE all data from all tables

- Although the Web site will keep being ‘online’, it will actually be useless without the information from the Database

Layer 7 DDoS : Resource Depletion

- Resource Depletion is a technique of performing DoS attacks on any site or application
- Classical Resource Depletion simply utilizes very large amounts of attacker resources which includes
  - Memory
  - CPU
  - Disk Space
- Sophisticated attacks pinpoint the weak points of the application to achieve maximum effect using minimal resources

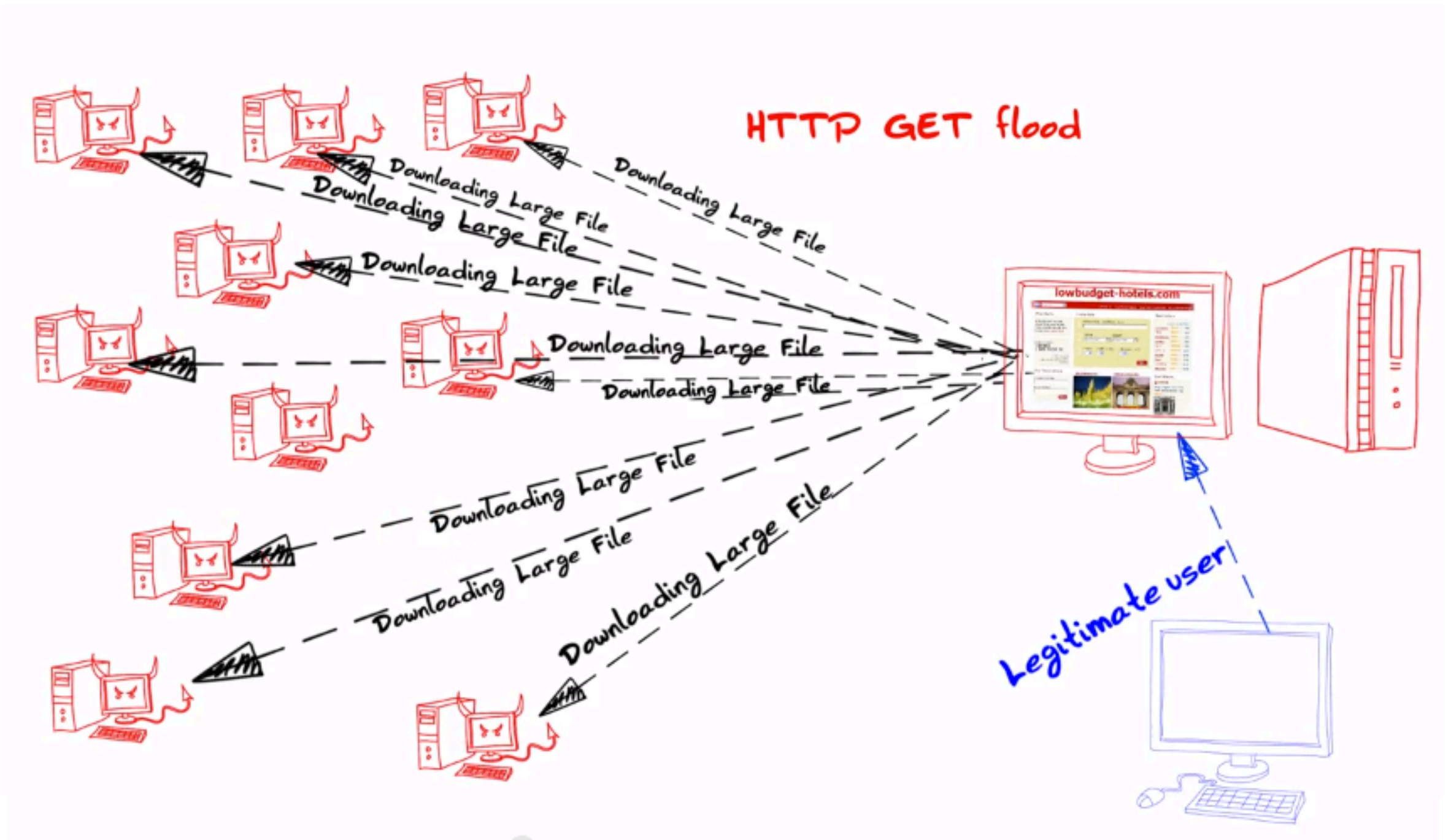
Effectiveness of Layer 7 DDoS

- Higher Obscurity
- Higher Efficiency
- Higher Lethality

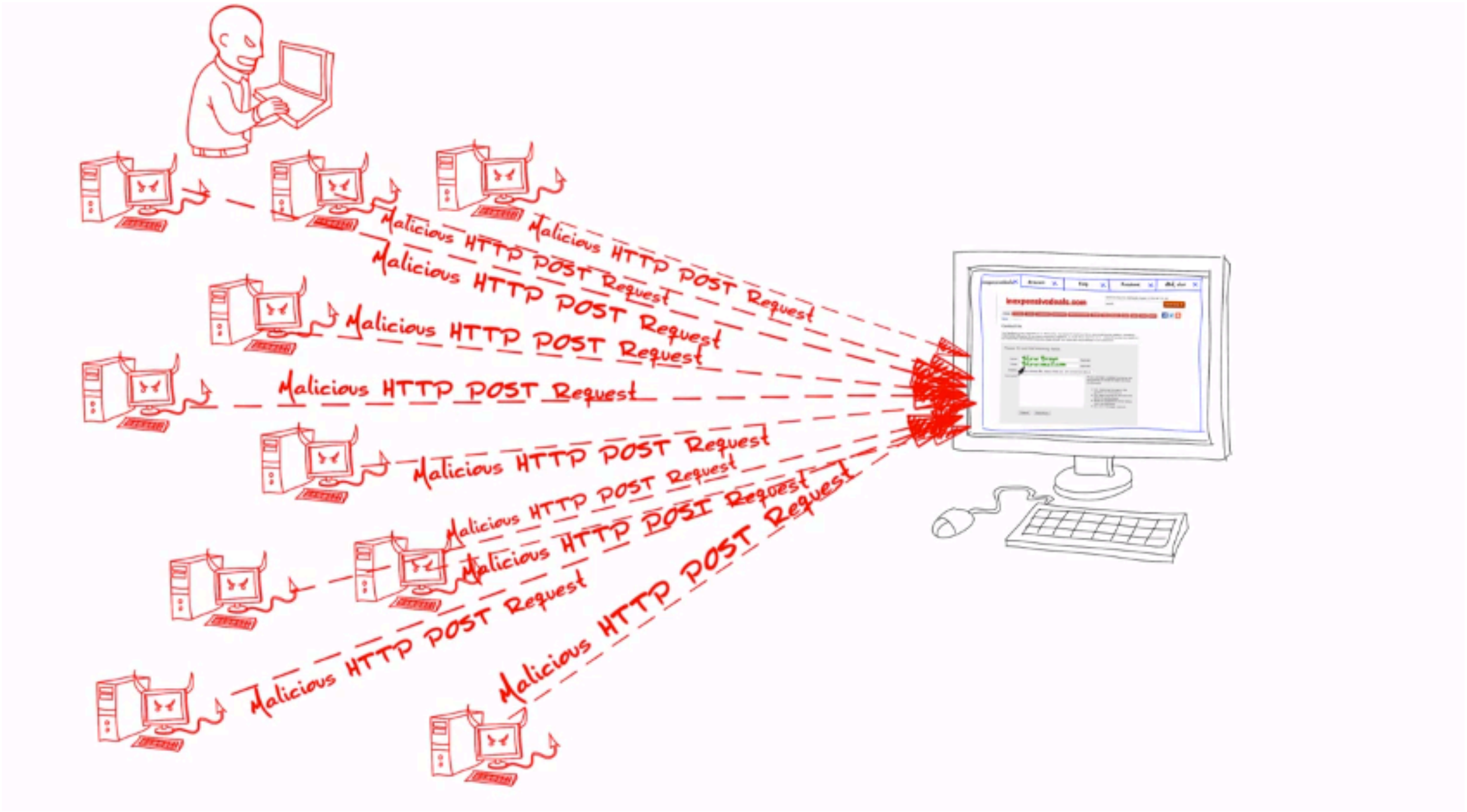
Layer 7 DDoS Web Attack

- Causes related to your inefficient codes
- Protocol Weakness
  - HTTP GET
  - HTTP POST

HTTP GET DDoS Attack



HTTP POST DDoS Attack

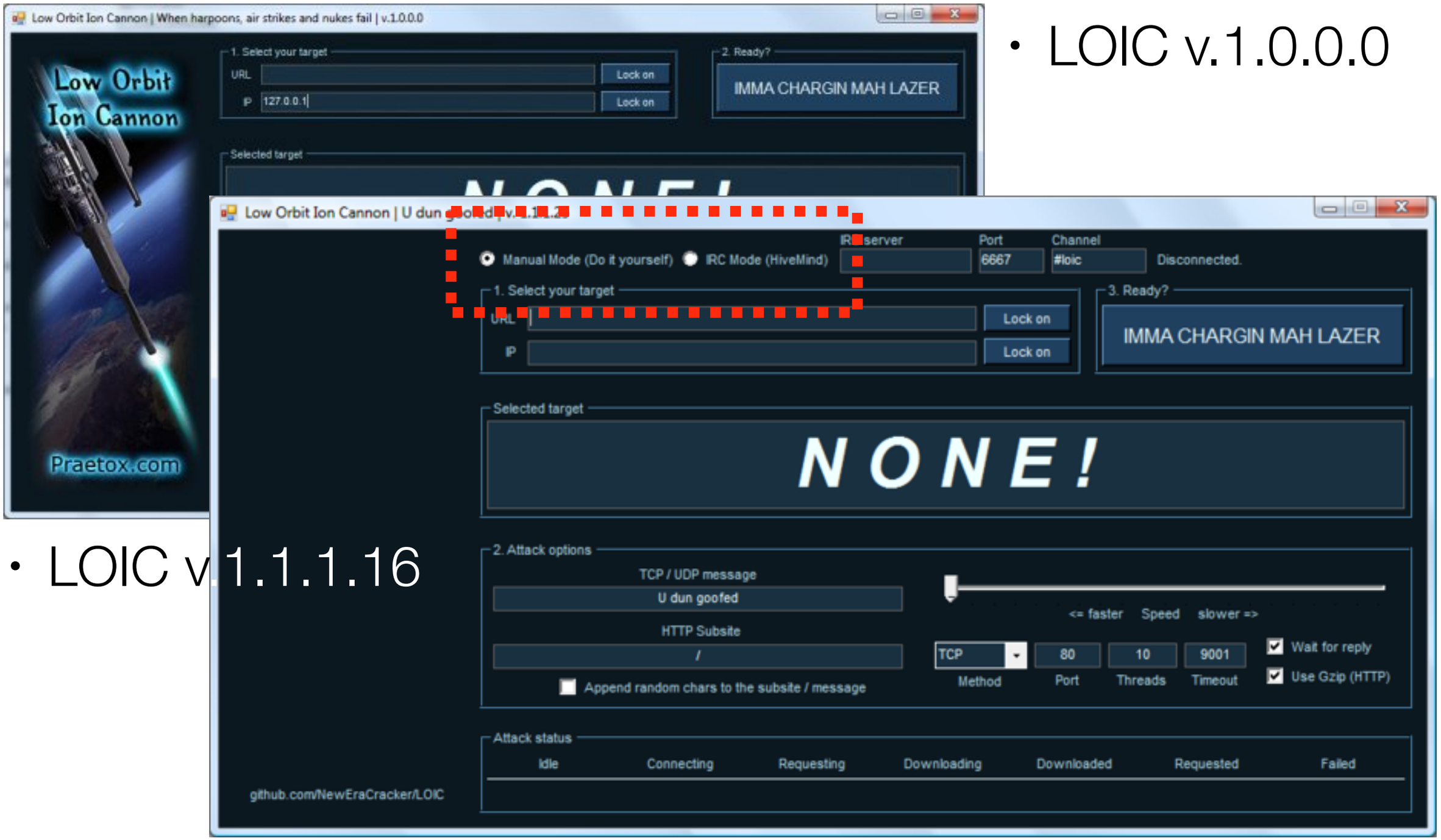


Layer 7 DDoS Tools

- **Slowloris** abuses handling of HTTP request headers sssloooowly…
- Written by RSnake
- Iteratively injects one custom header at a time and goes to sleep
- Web server vainly awaits the line space that will never come

- **R-U-Dead-Yet?** abuses HTTP web form fields
- Iteratively injects one custom byte into a web application post field and goes to sleep
- Application threads become zombies awaiting ends of posts till death lurks upon the website

Low Orbit Ion Cannon (LOIC)



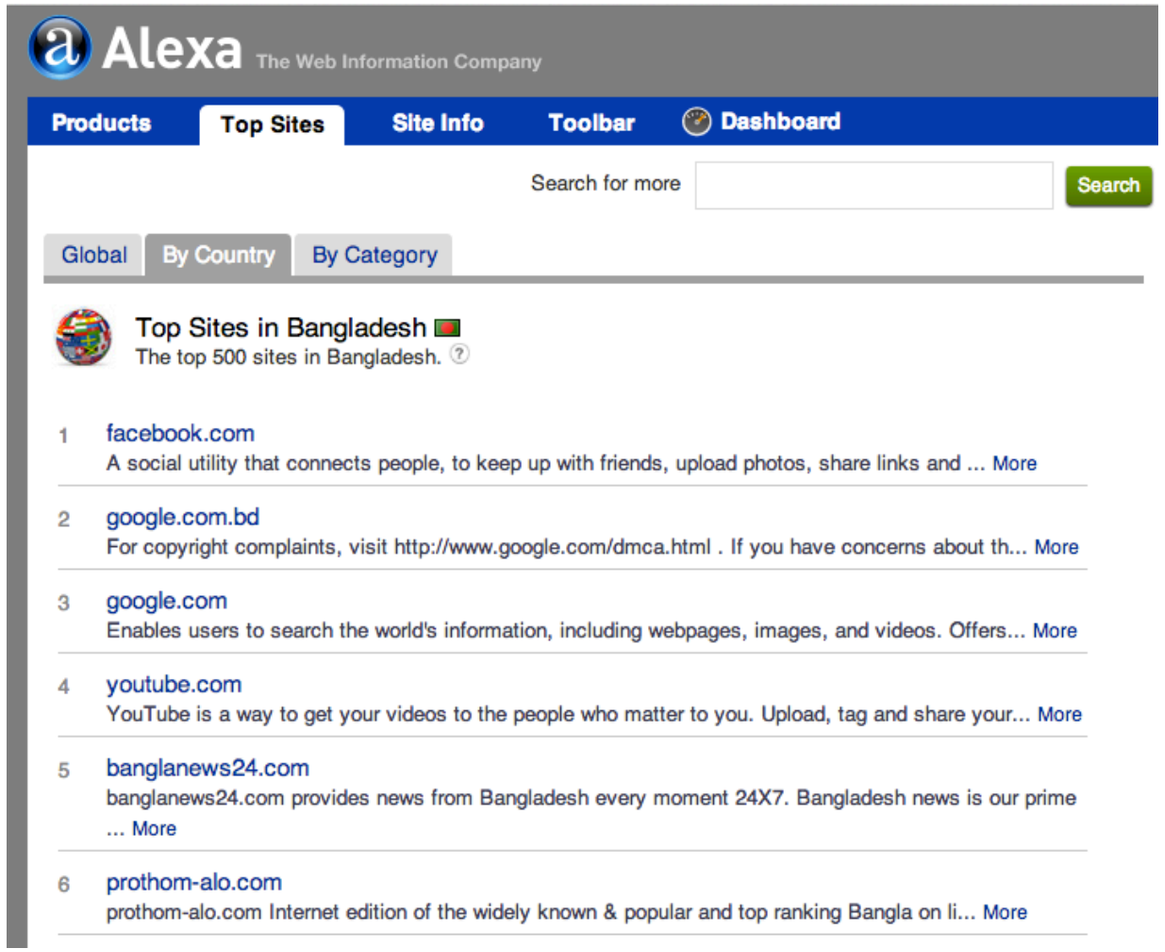
- LOIC v.1.0.0.0
- LOIC v.1.1.1.16

CASE STUDY

PRACTICAL APPROACH

Case Study

- Time: August 2012
- Country: Bangladesh
- Site: www.prothom-alo.com
- Ranked top 10 sites in Bangladesh (Source: Alexa)



Initial Findings

- Massive HTTP GET Flood
- Site is not accessible
- There is no major changes in bandwidth utilization
- Proper monitoring not in place to identify the actual attack
- Attack source is from Russia, China and some countries from Africa

Initial Findings : Logs

```
186.58.179.33 - - [21/Aug/2012:00:10:06 +0600] "GET / HTTP/1.1" 200 12474 "-" "Mozilla/
4.0 (compatible; MSIE 8.0; Windows NT 6.0; WOW64; Trident/4.0; SLCC1; .NET CLR
2.0.50727; .NET CLR 3.5.21022; .NET CLR 3.5.30729; .NET CLR 3.0.30618)"

189.76.197.117 - - [21/Aug/2012:00:10:06 +0600] "GET / HTTP/1.1" 200 12474 "-"
"Mozilla/5.0 (Windows; U; Windows NT 5.1; ru; rv:1.8.1.19) Gecko/20081201 Firefox/
2.0.0.19”

186.58.179.33 - - [21/Aug/2012:00:10:06 +0600] "GET / HTTP/1.1" 200 12474 "-" "Mozilla/
4.0 (compatible; MSIE 8.0; Windows NT 6.0; WOW64; Trident/4.0; SLCC1; .NET CLR
2.0.50727; .NET CLR 3.5.21022; .NET CLR 3.5.30729; .NET CLR 3.0.30618)”

186.6.168.11 - - [21/Aug/2012:00:10:06 +0600] "GET / HTTP/1.1" 200 12474 "-" "Mozilla/
4.0 (compatible; MSIE 5.0; Windows 2000) Opera 6.03 [en]" 197.0.165.121 - - [21/Oct/
2010:00:10:07 -0400] "GET / HTTP/1.1" 200 12474 "-" "Mozilla/5.0 (Windows; U; Windows
NT 5.1; en-US) AppleWebKit/525.19 (KHTML, like Gecko) Chrome/0.4.154.25 Safari/525.19”

189.76.197.117 - - [21/Aug/2012:00:10:06 +0600] "GET / HTTP/1.1" 200 12474 "-"
"Mozilla/5.0 (Windows; U; Windows NT 5.1; ru; rv:1.8.1.19) Gecko/20081201 Firefox/
2.0.0.19”

197.0.165.121 - - [21/Aug/2012:00:10:06 +0600] "GET / HTTP/1.1" 200 12474 "-" "Mozilla/
5.0 (Windows; U; Windows NT 5.1; en-US) AppleWebKit/525.19 (KHTML, like Gecko) Chrome/
0.4.154.25 Safari/525.19”
```

Approach 1

- Solution from hosting company
- Conventional host based firewall using IPTABLES.
- Fine tune TCP parameters
- Enable SYN Cookies
    - `echo 1 > /proc/sys/net/ipv4/tcp_syncookies`
- Enable socket reuse
    - `echo 1 > /proc/sys/net/ipv4/tcp_tw_recycle`
    - `echo 1 > /proc/sys/net/ipv4/tcp_tw_reuse`
- Increase local port range
    - `echo 1024 65535 > /proc/sys/net/ipv4/ip_local_port_range`

Issue with Approach 1

- Solution from hosting company required additional $$$$ which is significantly high
- Hard to justify management
- Host based firewall works only in Layer 3 & Layer 4
- Not capable to filter Layer 7 DDoS Attack

Approach 2

- Split DNS
  - DNS configured to resolve host based on GEOIP.
  - External uses request redirected to external server hosted in USA
- One new server co-located in Bangladesh
- Internal (Bangladesh) traffic has been redirected to new server
- Load has been distributed

Issue with Approach 2

- Issue with Split DNS
  - 4.2.2.2, 8.8.8.8 and other Open DNS
  - Lots of users from Bangladesh is using open DNS like 4.2.2.2 & 8.8.8.8.
  - For those users DNS is still resolving USA data center server IP

Approach 3

- Anycast
  - Failed
  - Most of the upstream provider and datacenter doesn’t allow anycast
  - It’s good in handling volumetric attack

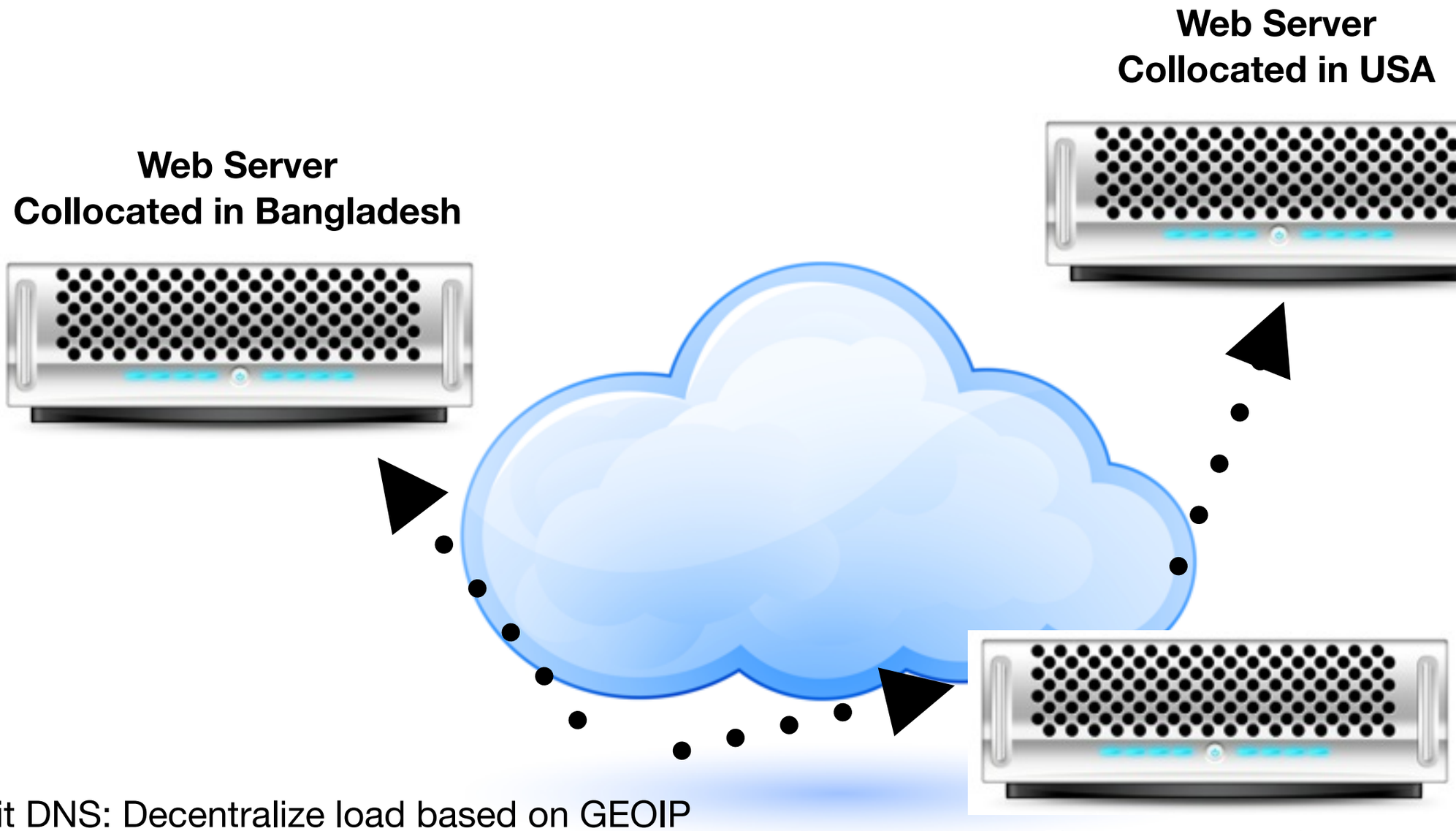
Approach 4

- Reverse Web Proxy
  - Use Reverse Proxy as frontend
  - Anti DDoS plugins along with other parameters
  - Minimize the attack vector
  - Distribute end user load and mitigation solution

MIGRATION

SOLUTION ARCHITECTURE

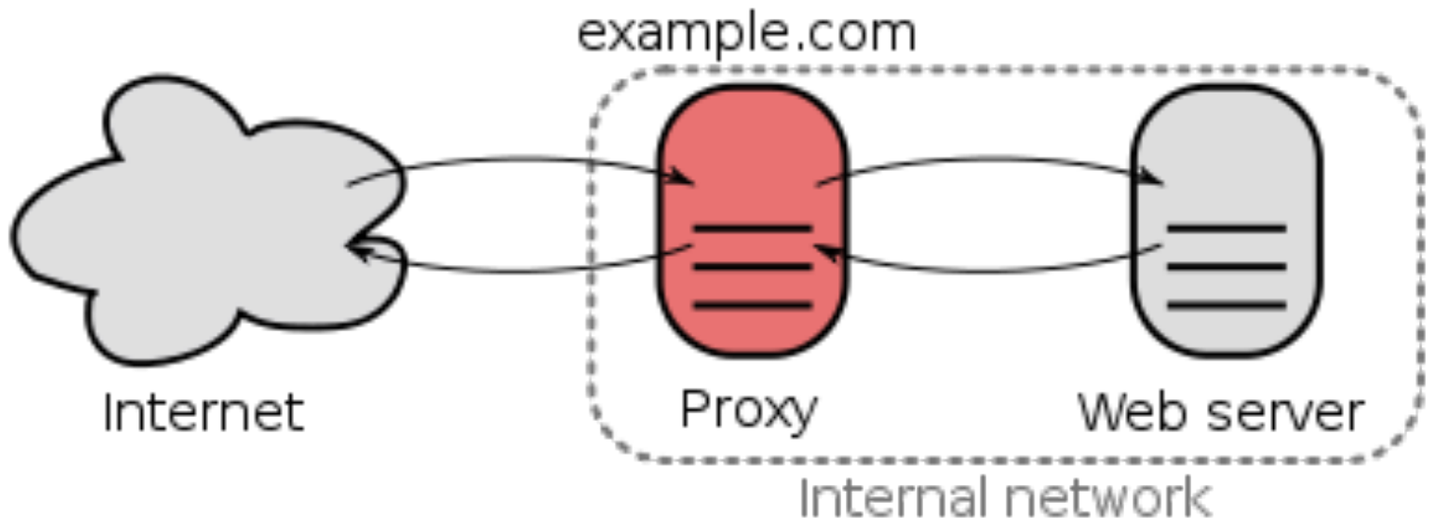
Solution Architecture



Spit DNS: Decentralize load based on GEOIP
Reverse Web Proxy with DDoS plugins as front end

Reverse Web Proxy

- A reverse proxy is a type of proxy server that retrieves resources on behalf of a client from one or more servers. These resources are then returned to the client as though they originated from the server itself (or servers themselves) –Wikipedia



Reverse Web Proxy

Why NGINX

- Event Driven
- Asynchronous
- Single Threaded

Nginx DDoS Plugins

- Available plugins:
  - **testcookie-nginx-module** [http://kyprizel.github.io/testcookie-nginx-module/]
  - **Roboo : HTTP Robot Mitigator** [http://www.ecl-labs.org/2011/03/17/roboo-http-mitigator.html]

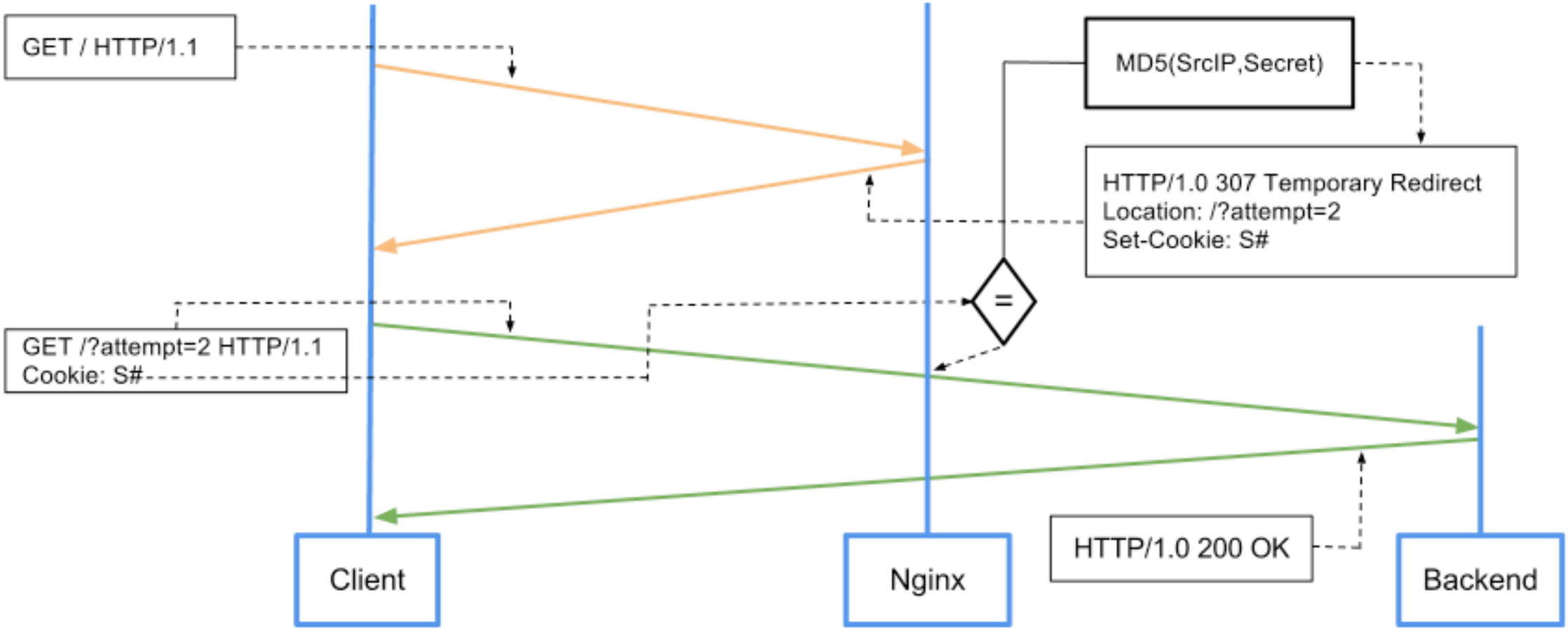
testcookie-nginx-module

- testcookie-nginx-module is a simple robot mitigation module using cookie based challenge/response technique.

- Challenge cookies can be set using different methods:
  - "Set-Cookie" + 307/302 HTTP Location redirect
  - "Set-Cookie" + HTML meta refresh redirect

- If you need Captcha or Flash, check testcookie-flash-processor

testcookie-nginx-module



Roboo : HTTP Robot Mitigator

- Uses advanced non-interactive HTTP challenge/response mechanisms to detect & mitigate HTTP Robots
- Weeds out the larger percentage of HTTP robots which do not use real browsers or implement full browser stacks, resulting in the mitigation of various web threats:
  - HTTP Denial of Service tools - e.g. Low Orbit Ion Cannon
  - Vulnerability Scanning - e.g. Acunetix Web Vulnerability Scanner, Metasploit Pro, Nessus
  - Web exploits
  - Automatic comment posters/comment spam as a replacement of conventional CAPTCHA methods
  - Spiders, Crawlers and other robotic evil
- Available at https://github.com/yuri-gushin/Roboo

Roboo : HTTP Robot Mitigator

- Will respond to each GET or POST request from an unverified source with a challenge:
    - Challenge can be Javascript or Flash based, optionally Gzip compressed
    - A real browser with full HTTP, HTML, Javascript and Flash player stacks will re-issue the original request after setting a special HTTP cookie that marks the host as “verified”
- Marks verified sources using an HTTP Cookie
- Integrates with Nginx web server and reverse proxy as an embedded Perl module

Configuration Snap (nginx.conf)

```
user www-data;
worker_processes 8;
pid /var/run/nginx.pid;

events {
        worker_connectior
        use epoll;
}

http {

        ## Disable Nginx
        server_tokens off

        ## Add here all k
        map $request_meth
                default 1
                ~(?i)(GET
        }

        ## Add here all u
        map $http_user_ag
                default (
                ~(?i)(htt
:simple|Havij) 1;
        }

        ## Add here all r
        map $http_referer
                default (
                ~(?i)(bak
        }

        include /etc/ngin

        default_type  app
        log_format main '
        access_log /var/l
        server_names_hash
        #server_names_has
        #types_hash_bucke
        types_hash_max_size 2048;

        client_header_timeout 10m;
        client_body_timeout 10m;
        send_timeout 10m;
        connection_pool_size 256;
        client_body_buffer_size 16
        large_client_header_buffer
        request_pool_size 4k;
        sendfile on;

        gzip off;
        gzip_min_length 0;
        gzip_buffers 4 8k;
        gzip_types text/plain appl
        gzip_proxied any;
        gzip_http_version 1.0;
        output_buffers 1 32k;
        postpone_output 1460;

        tcp_nopush on;
        tcp_nodelay on;
        keepalive_timeout 75 20;
        ignore_invalid_headers on;

        #set_real_ip_from 202.4.96
        #real_ip_header X-Forwarde
        #real_ip_recursive on;

        proxy_cache_path /usr/shar
        proxy_cache_key "$scheme:/


        # creates zone "req_limit_
        limit_req_zone  $binary_re
        limit_conn_zone $binary_re

        include /etc/nginx/conf.d/
        include /etc/nginx/sites-e

        #default config, module di
        postpone_output 1460;

        tcp_nopush on;
        tcp_nodelay on;
        keepalive_timeout 75 20;
        ignore_invalid_headers on;

        #set_real_ip_from 202.4.96.250;
        #real_ip_header X-Forwarded-For;
        #real_ip_recursive on;

        proxy_cache_path /usr/share/nginx/cache levels=1:2 keys_zone=cache:10m inactive=10m ma
        proxy_cache_key "$scheme://$host$request_uri";


        # creates zone "req_limit_per_ip" allocating 10MB for this session then limits queries
        limit_req_zone  $binary_remote_addr zone=req_limit_per_ip:10m rate=1r/s;
        limit_conn_zone $binary_remote_addr zone=conn_limit_per_ip:10m;

        include /etc/nginx/conf.d/*.conf;
        include /etc/nginx/sites-enabled/*;

        #default config, module disabled
        testcookie off;
        #setting cookie name
        testcookie_name BPC;
        #setting secret
        testcookie_secret keepmesecret;
        #setting session key
        testcookie_session $remote_addr;
        #setting argument name
        testcookie_arg ckattempt;
        #setting maximum number of cookie setting attempts
        testcookie_max_attempts 3;
        testcookie_get_only on;
        testcookie_fallback /cookies.html?backurl=http://$host$request_uri;

        testcookie_whitelist {
                66.249.76.0/24;
        }
```

Configuration Snap (Roboo)

```
server {
        listen   80;
        server_name www1.prothom-alo.com;
        add_header  Cache-Control public;
        #access_log  /var/log/nginx/www1.prothom-alo.com.access
        #error_log   /var/log/nginx/www1.prothom-alo.com.error.

        ## Request-range protection fix.
        if ($http_range ~ "(?:d*s*-s*d*s*,s*){5,}") {
                return 416;
        }

        ## Deny access based on HTTP method
        if ($bad_method = 1) { return 444; }

        ## Deny access based on the User-Agent header
        if ($bad_bot = 1) { return 403; }

        ## Deny access based on the Referer header
        if ($bad_referer = 1) { return 403; }

        location = /robots.txt  { access_log off; log_not_found
        location = /favicon.ico { access_log off; log_not_found

        location / {
            limit_req zone=req_limit_per_ip burst=5;
            limit_conn conn_limit_per_ip 10;

                resolver 210.4.77.180;
                perl Roboo::handler;
                set $Roboo_challenge_modes      "JS,gzip";
                #set $Roboo_challenge_modes      "SWF,gzip";

                # Defaults
                set $Roboo_cookie_name "Anti-Robot";          #
                set $Roboo_validity_window 600;               #
                set $Roboo_whitelist "IP(),UA(''),URI('')";   #
                set $Roboo_charset "UTF-8";                   #
                set $Roboo_challenge_hash_input $remote_addr;#
                error_page 555 = @proxy;
                expires epoch;
```

```
location / {
    limit_req zone=req_limit_per_ip burst=5;
    limit_conn conn_limit_per_ip 10;

        resolver 210.4.77.180;
        perl Roboo::handler;
        set $Roboo_challenge_modes      "JS,gzip";
        #set $Roboo_challenge_modes      "SWF,gzip";

        # Defaults
        set $Roboo_cookie_name "Anti-Robot";          # Cookie name used f
        set $Roboo_validity_window 600;               # Authentication val
        set $Roboo_whitelist "IP(),UA(''),URI('')";   # Whitelist - IP add
        set $Roboo_charset "UTF-8";                   # Charset used durin
        set $Roboo_challenge_hash_input $remote_addr;# Advanced - challen
        error_page 555 = @proxy;
        expires epoch;
                add_header Last-Modified "";
        if ($Roboo_challenge_modes ~ gzip) {
                gzip on;
        }
        access_log /var/log/nginx/www1.prothom-alo.com.challenged.log;
}
location @proxy {
    resolver 210.4.77.180;
    proxy_pass http://103.16.72.4$request_uri;
    proxy_next_upstream error timeout invalid_header http_500 http_502 ht
    proxy_redirect off;
    proxy_buffering off;
    proxy_set_header Host $http_host;
    proxy_set_header X-Real-IP $remote_addr;
    proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
    proxy_connect_timeout 10;
    proxy_read_timeout 10;
    proxy_set_header Range "";
    proxy_cache cache;
    proxy_cache_valid 5m;
    gzip on;
        access_log /var/log/nginx/www1.prothom-alo.com.verified.log;
}
}
```

Configuration Snap (testcookie-nginx-module)

```
server {
        listen 80;
        server_name www.prothom-alo.com;
        #add_header  Cache-Control public;
        access_log  /var/log/nginx/www.prothom-alo.com.access.log;
        error_log   /var/log/nginx/www.prothom-alo.com.error.log error;

        ## Request-range protection fix.
        if ($http_range ~ "(?:d*s*-s*d*s*,s*){5,}") {
                return 416;
        }

        ## Deny access based on HTTP method
        if ($bad_method = 1) { return 444; }

        ## Deny access based on the User-Agent header
        if ($bad_bot = 1) { return 403; }

        ## Deny access based on the Referer header
        if ($bad_referer = 1) { return 403; }

        location = /robots.txt  { access_log off; log_not_found off; }
        location = /favicon.ico { access_log off; log_not_found off; }

        location = /aes.min.js {
                gzip on;
                gzip_min_length 1000;
                gzip_types text/plain;
                root /var/www;
            }

        location = /cookies.html {
                 root /var/www;
         }

            location / {
                testcookie on;
                proxy_set_header Host $host;
                proxy_set_header X-Real-IP $remote_addr;
                proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
                proxy_pass http://103.16.72.4;
```

Key Configuration Parameters

Variables	Description
`worker_processes`	This number should be, at maximum, the number of CPU cores on your system.
`worker_connections`	Determines how many clients will be served by each worker process. (Max clients = worker_connections * worker_processes)
`perl_modules /opt/local/share/nginx;` `perl_require Roboo.pm;`	Enabling Roboo Plugings
`map $http_user_agent`	Define http agent (httrack|WinHTTrack|htmlparser|libwww|Python)

Key Configuration Parameters

Variables	Description
`$http_referer`	(babes|click|forsale|jewelry|nudit)
`limit_req_zone $binary_remote_addr zone=req_limit_per_ip:10m rate=1r/s; limit_conn_zone $binary_remote_addr zone=conn_limit_per_ip: 10m;`	creates zone “req_limit_per_ip” allocating 10MB for this session then limits queries for remote ip address to 1 request per second
`include /etc/nginx/ allow_only.conf`	Can define IP address for where site is only accessible

Logs : Roboo

challenged.log

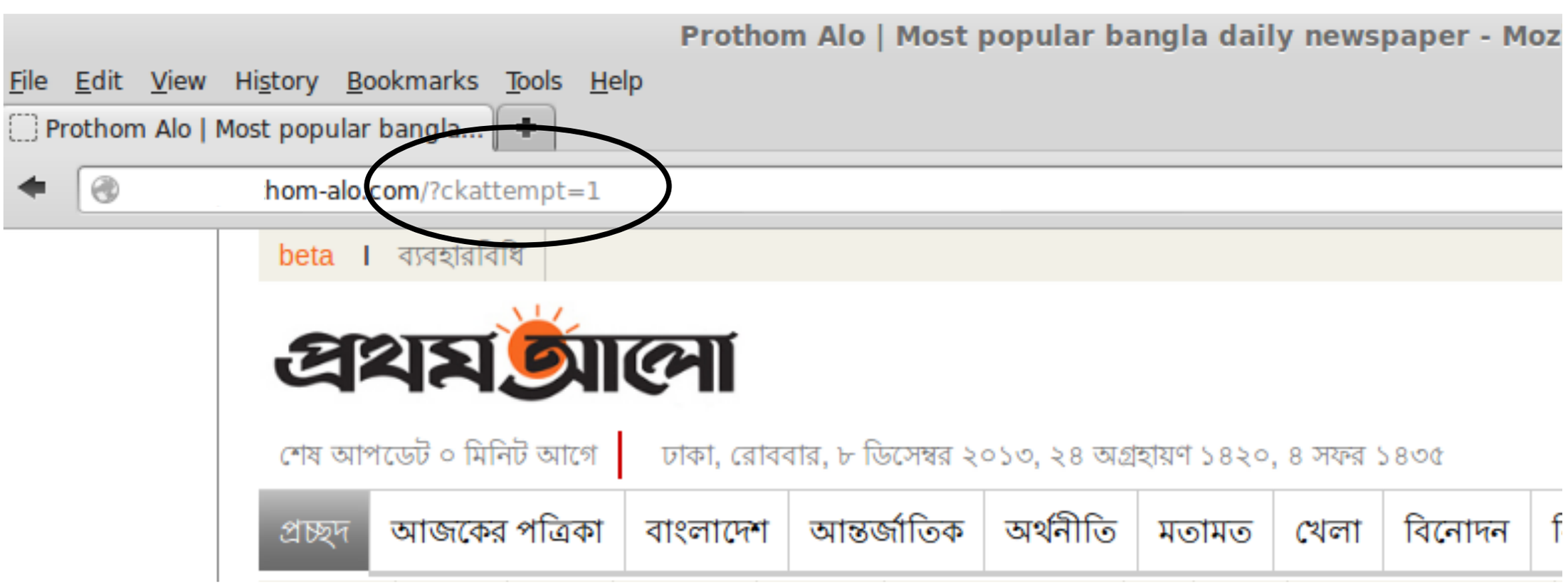
```
202.4.100.35 - - [28/Nov/2013:14:05:10 +0600] "GET / HTTP/1.1" 200
669 "-" "Mozilla/5.0 (Macintosh; Intel Mac OS X 10_9_0) AppleWebKit/
537.36 (KHTML, like Gecko) Chrome/31.0.1650.57 Safari/537.36"

202.4.100.35 - - [28/Nov/2013:14:05:11 +0600] "GET /Anti-Robot-
GET-2babb2739558804 2480c.swf HTTP/1.1" 200 1025 "http://
www1.prothom-alo.com/" "Mozilla/5.0 (Macintosh; Intel Mac OS X
10_9_0) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/31.0.1650.57
Safari/537.36”
```

verified.log

```
202.4.100.35 - - [28/Nov/2013:14:05:12 +0600] "GET / HTTP/1.1" 200
31942 "http://www1.prothom-alo.com/" "Mozilla/5.0 (Macintosh; Intel
Mac OS X 10_9_0) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/
31.0.1650.57 Safari/537.36"
```

Logs : testcookie-nginx-module

```
202.4.100.35 - - [30/Nov/2013:18:06:53 +0600]
"GET /?ckattempt=1 HTTP/1.1" 200 31643 "-"
"Mozilla/5.0 (iPhone; CPU iPhone OS 7_0_4 like Mac
OS X) AppleWebKit/537.51.1 (KHTML, like Gecko)
Version/7.0 Mobile/11B554a Safari/9537.53"
```



SIMULATION

Solution Architecture

Web Server : NGINX

1

Reverse Proxy (nx.bdnog.org)
192.168.1.100

Attack Server

3

Attack Server (attack.bdnog.org)
192.168.1.200

Web Server : Apache

2

Web Server (web.bdnog.org)
192.168.1.150

- All the hardwares are configured in Virtual Box
- DDoS launched in closed network
- Please don’t try in production network

Simulation : Phase 1

Web Server : NGINX

1

Reverse Proxy (nx.bdnog.org)
192.168.1.100

Attack Server

3

Attack Server (attack.bdnog.org)
192.168.1.200

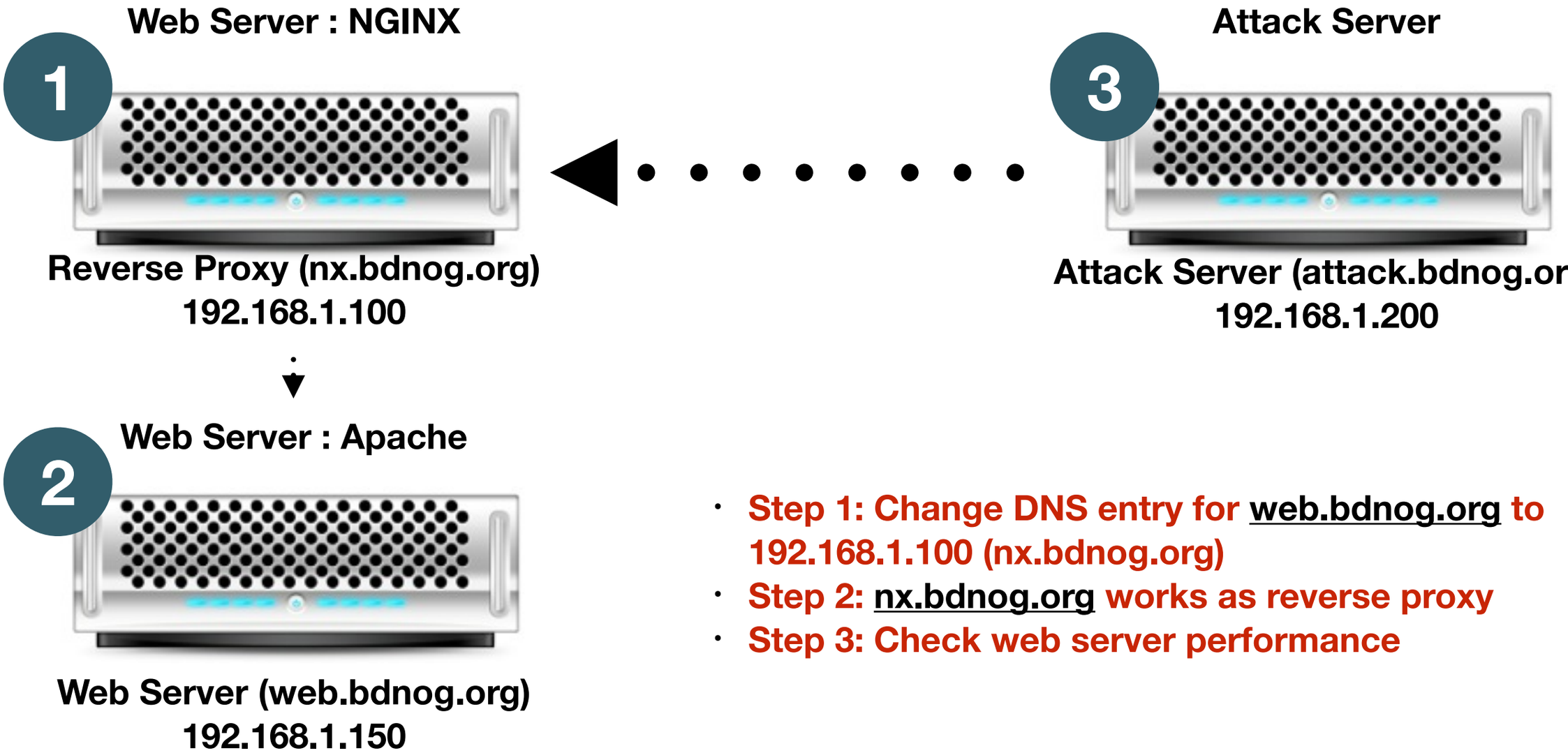
Web Server : Apache

2

Web Server (web.bdnog.org)
192.168.1.150

- Step 1: Launch attack on web.bdnog.org
- Step 2: Check web server performance

Simulation : Phase 2



- Step 1: Change DNS entry for web.bdnog.org to 192.168.1.100 (nx.bdnog.org)
- Step 2: nx.bdnog.org works as reverse proxy
- Step 3: Check web server performance

Simulation : Available Tools

DDOSIM Layer 7 DDoS Simulator	http://sourceforge.net/projects/ddosim/
BONESI The DDoS Botnet Simulator	https://code.google.com/p/bonesi/
Slowhttptest L7 DoS simulator	http://code.google.com/p/slowhttptest/

Tools Used : ddosim

- ddosim is a tool that can be used in a laboratory environment to simulate a distributed denial of service (DDOS) attack against a target server

- ddosim simulates several zombie hosts (having random IP addresses) which create full TCP connections to the target server.

- After completing the connection, ddosim starts the conversation with the listening application (e.g. HTTP server).

Tools Used : ddosim

- ddosim is written in C++ and runs on Linux. Its current functionalities include:
  - HTTP DDoS with valid requests
  - HTTP DDoS with invalid requests (similar to a DC++ attack)
  - SMTP DDoS
  - TCP connection flood on random port

Tools Used : ddosim

- Running DDOSIM out of lab is not really possible because it simulates distributed (multiple source IPs) attacks using a connection-oriented protocol (TCP) which needs at least the 3way handshake before sending any useful data.

- So the communication must be bidirectional. The packets (TCP SYN-ACK) sent by the server must reach the attacker (having random IP address)

Tools Used : ddosim

1. Establish 10 TCP connections from random IP addresses to www server and send invalid HTTP requests

```
./ddosim   -d 192.168.1.2   -p 80   -c 10
-r HTTP_INVALID  -i eth0
```

2. Establish infinite connections at higher speed to www server and make HTTP valid requests:

```
./ddosim   -d 192.168.1.2   -p 80   -c 0
-w 0   -t 10   -r HTTP_VALID  -i eth0
```

Simulation


INTERNAL AUDIT EFFECTIVENESS REVIEW
KEY ACTIONS
1. REVAMP 'TIRED' REPORTING
2. NO SURPRISES
3. USE TECHNOLOGY EFFECTIVELY
I'M JUST TWEETING THE LATEST FINDINGS TO THE AUDIT COMMITTEE
INTERNAL AUDIT DEPT.

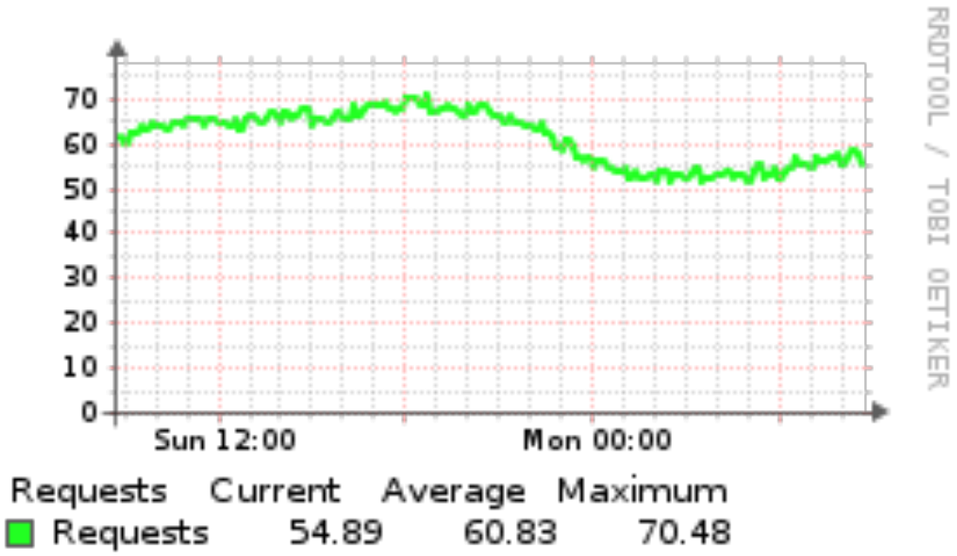
FINDINGS

Findings

- Proper monitoring
- Log analysis (logstalgia)
- Off-loading & Splitting Traffic / DDoS Mitigation in broader scale

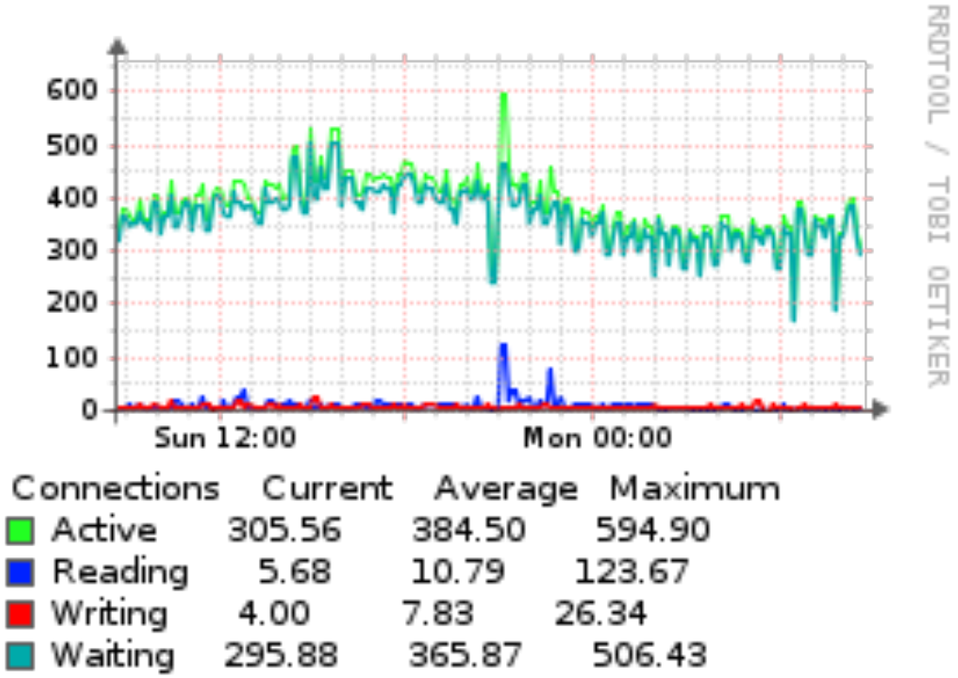
Monitoring

- Monitoring NGINX/Apache with Observium



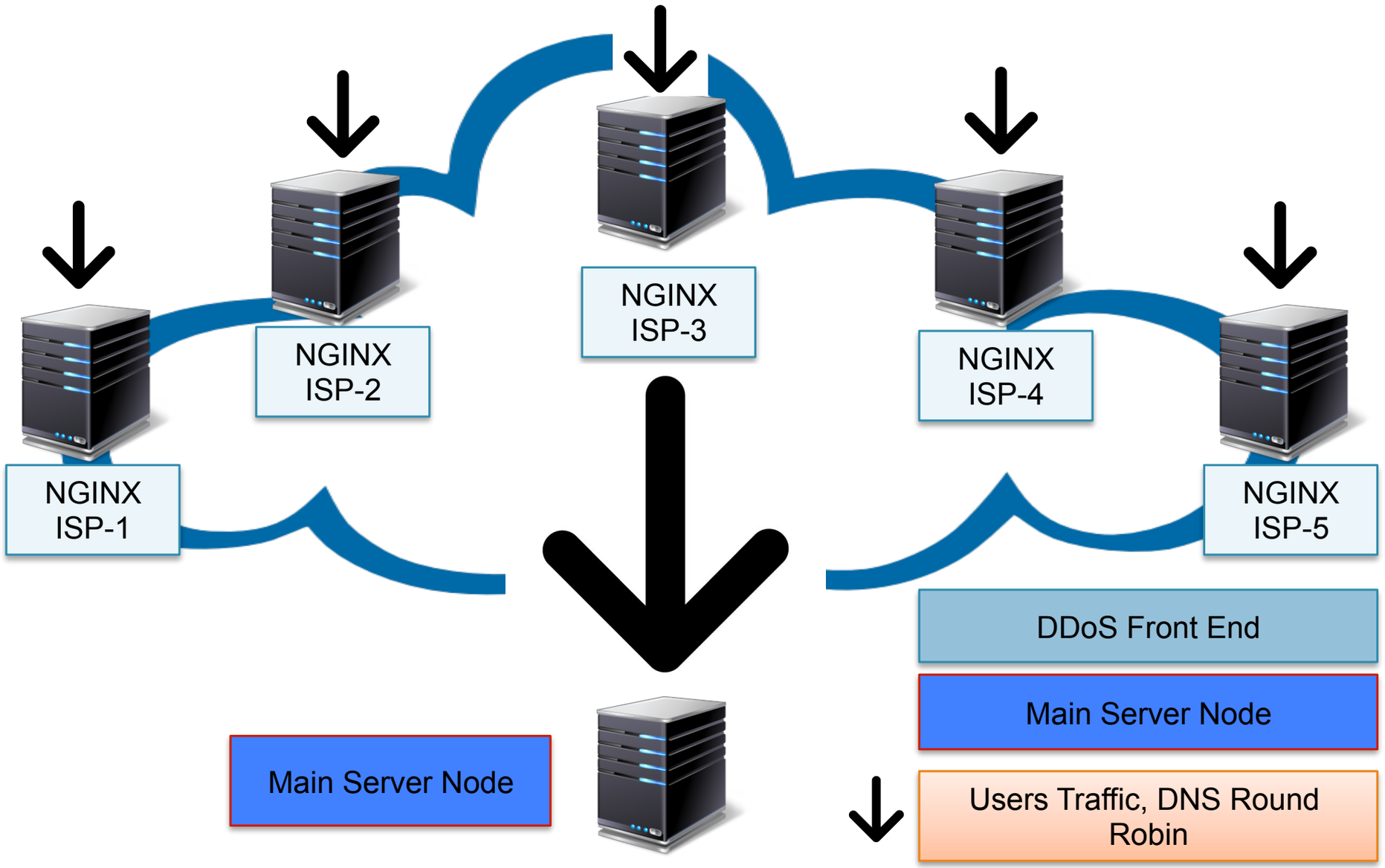
Request Statistics

Server Status



Log Analysis (logstalgia)

Off-loading & Splitting Traffic



Issues

- Scalability
- Performance Optimization
- Integrate DDoS mitigation solution with routing infrastructure
- Integrate ExaBGP / BGP FlowSpec

Scripts (Finding the BOT)

```
# more /var/log/apache2/access.log | grep ”bdnog\.org"
| grep "GET / HTTP”
```

```
203.188.170.218 - - [26/Oct/2013:17:58:25 +0600] "GET / HTTP/1.1" 200
537 "http://www.24livenewspaper.com/site/index.php?url=www.bdnog.org"
"Mozilla/5.0 (compatible; MSIE 9.0; Windows NT 6.1; Trident/5.0)"

94.109.96.192 - - [26/Oct/2013:17:58:26 +0600] "GET / HTTP/1.1" 200
537 "http://www.bdnog.org/" "Mozilla/5.0 (Linux; U; Android 2.3.4;
en-gb; SonyEricssonWT19iv Build/4.0.2.A.0.58) AppleWebKit/533.1
(KHTML, like Gecko) Version/4.0 Mobile Safari/533.1"

203.188.170.218 - - [26/Oct/2013:17:58:26 +0600] "GET / HTTP/1.1" 200
537 "http://www.24livenewspaper.com/site/index.php?url=www.bdnog.org"
"Mozilla/5.0 (compatible; MSIE 9.0; Windows NT 6.1; Trident/5.0)””
```

Scripts (Finding the BOT)

```
# more /var/log/apache2/access.log | grep ”bdnog
\.org" | grep "GET / HTTP" | cut -d " " -f1
```

```
94.109.96.192

203.188.170.218

94.109.96.192

203.188.170.218

94.109.96.192

203.188.170.218
```

Scripts (Finding the BOT)

```
# more /var/log/apache2/access.log | grep
"bdnog\.org" | grep "GET / HTTP" | cut -d " " -
f1 | sort | uniq -c| awk '{if($1>K){print $2}}'
```

Replace **K** with value

```
114.130.136.182

117.18.229.59

117.18.231.60

175.140.219.213

202.134.10.135
```

Scripts (Finding the BOT)

```
# ipset create blacklist hash:net

# more /var/log/apache2/access.log | grep ”bdnog
\.org" | grep "GET / HTTP" | cut -d " " -f1 |
sort | uniq -c| awk '{if($1>100){print $2}}' |
xargs -tl -I _ ipset -A blacklist _
```

```
ipset -A blacklist 114.130.136.182

ipset -A blacklist 117.18.229.59

ipset -A blacklist 117.18.231.60

ipset -A blacklist 175.140.219.213
```

Scripts (Finding the BOT)

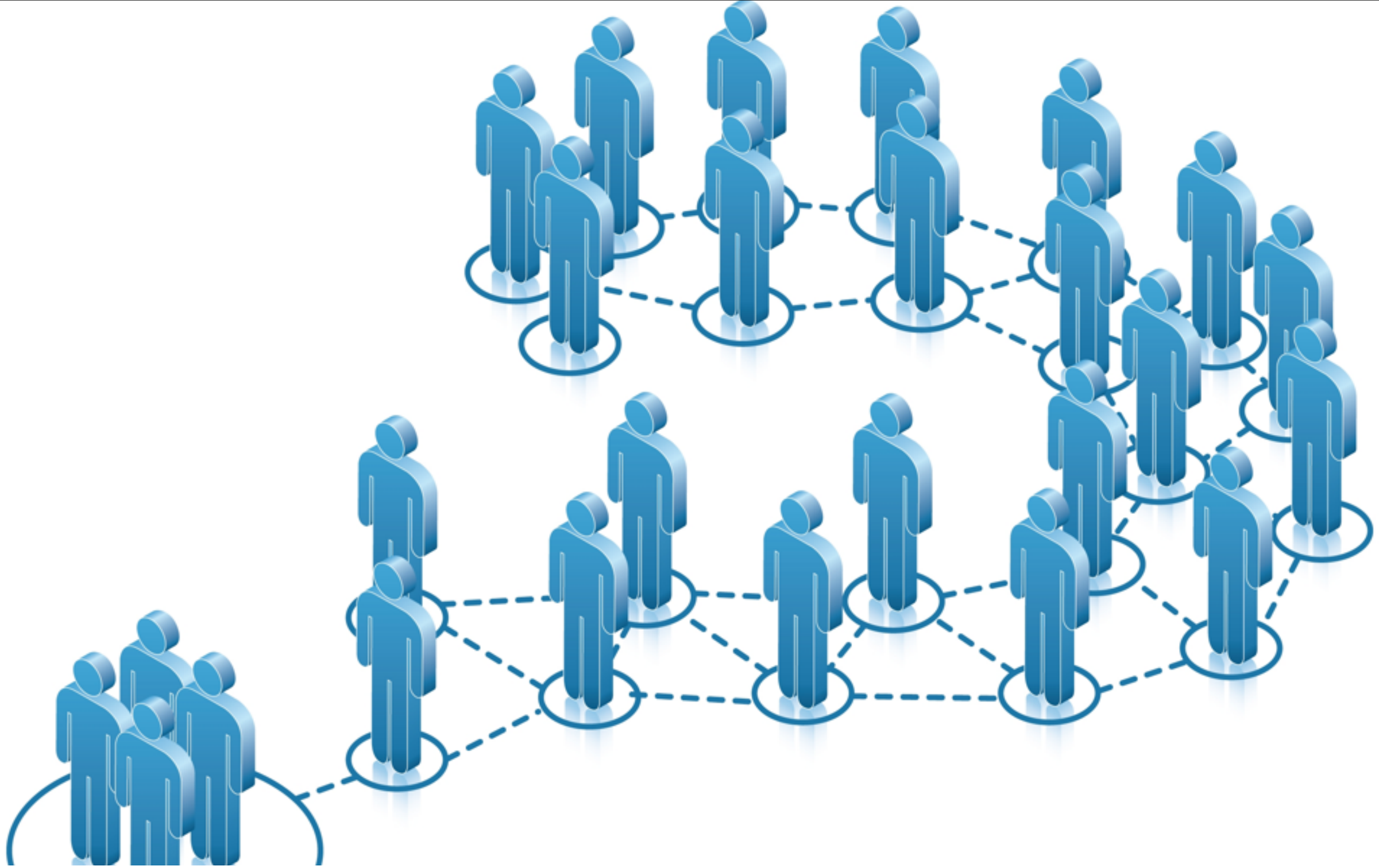
```
# ipset list

Name: blacklist

Type: hash:net

Header: family inet hashsize 1024 maxelem 65536

Size in memory: 16984

References: 0

Members:

114.130.136.182

203.188.170.218

202.134.10.135

37.160.132.237
```

Scripts (Finding the BOT)

```
/sbin/iptables -X DDOS_HTTP_FILTER

/sbin/iptables -N DDOS_HTTP_FILTER

/sbin/iptables -A DDOS_HTTP_FILTER -p tcp
--syn --dport 80 -m set --match-set
blacklist src -j DROP
```

Special Thanks

- GZ Kabir, BDCOM
- Sumon Ahmed Sabir, Fiber@Home
- Technical Team of Prothom Alo.Com
- Attackers



QUESTION