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 : 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

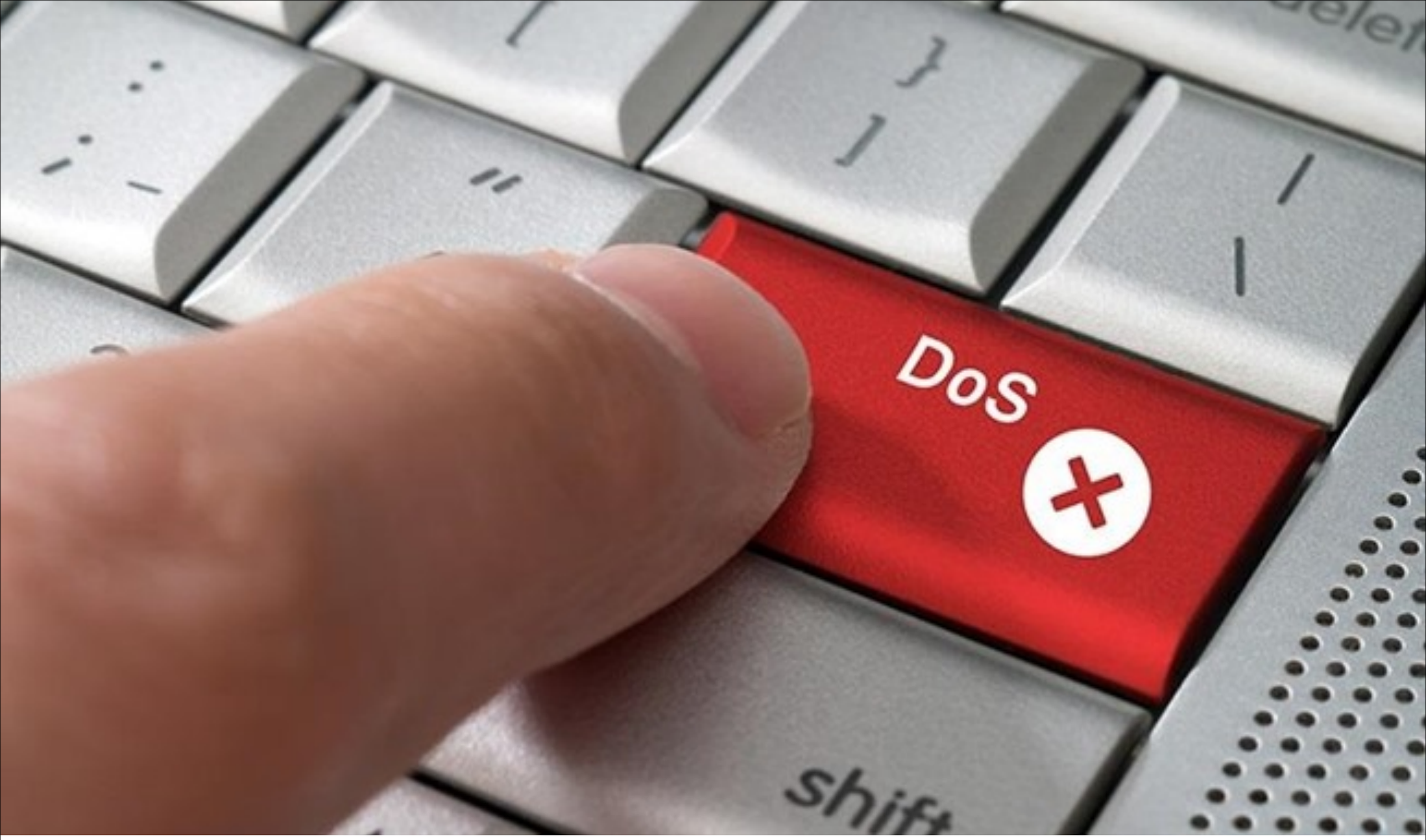
- Layer 3 - muscle-based attacks
- Flood of TCP/UDP/ICMP/IGMP packets, overloading infrastructure due to high rate processing/discarding of packets and filling up the packet queues, or saturating pipes
- Introduce a packet workload most gear isn't designed for
- Example - UDP flood to non-listening port

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



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

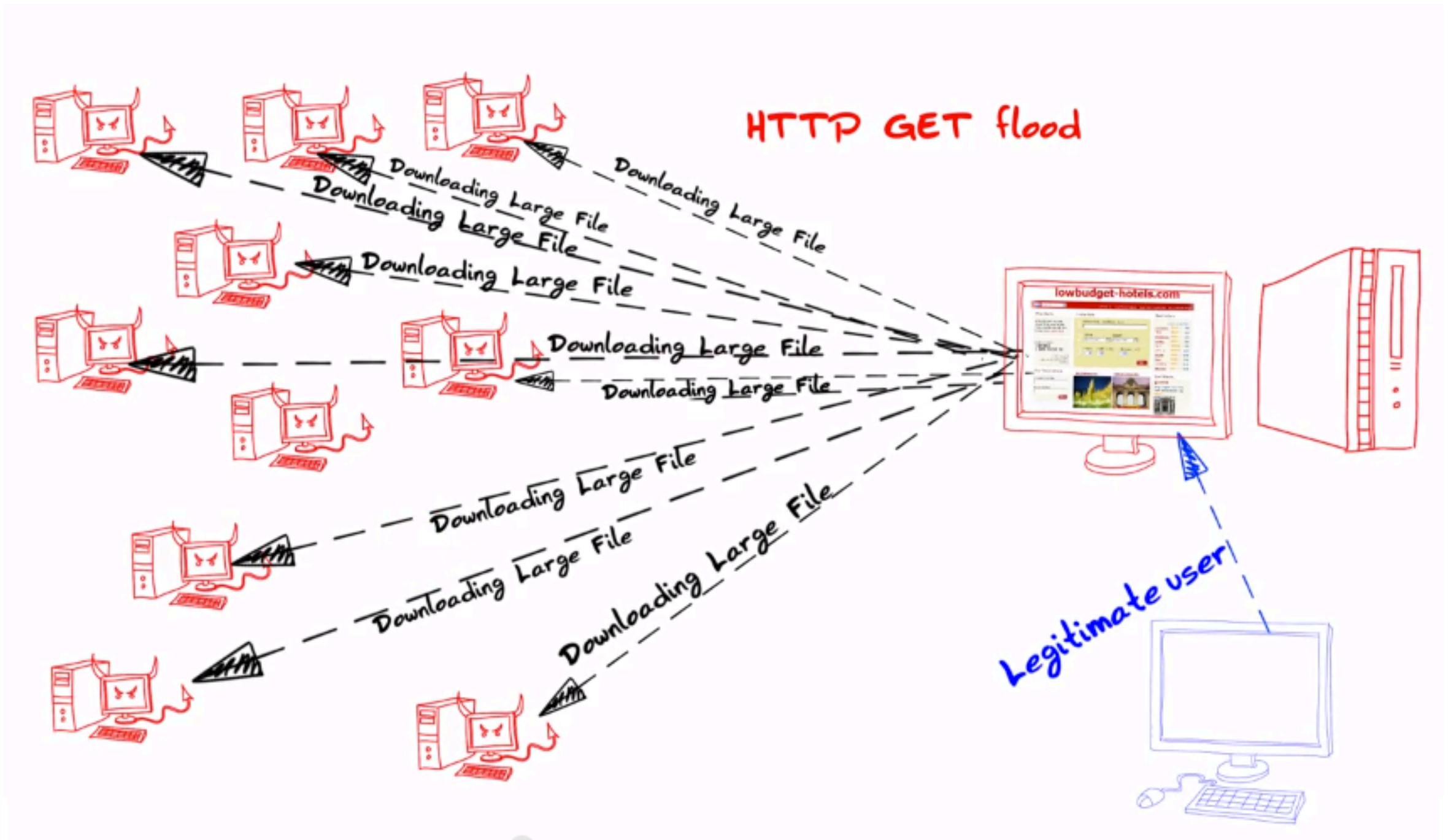
Effectiveness of Layer 7 DDoS

- Higher Obscurity
- Higher Efficiency
- Higher Lethality

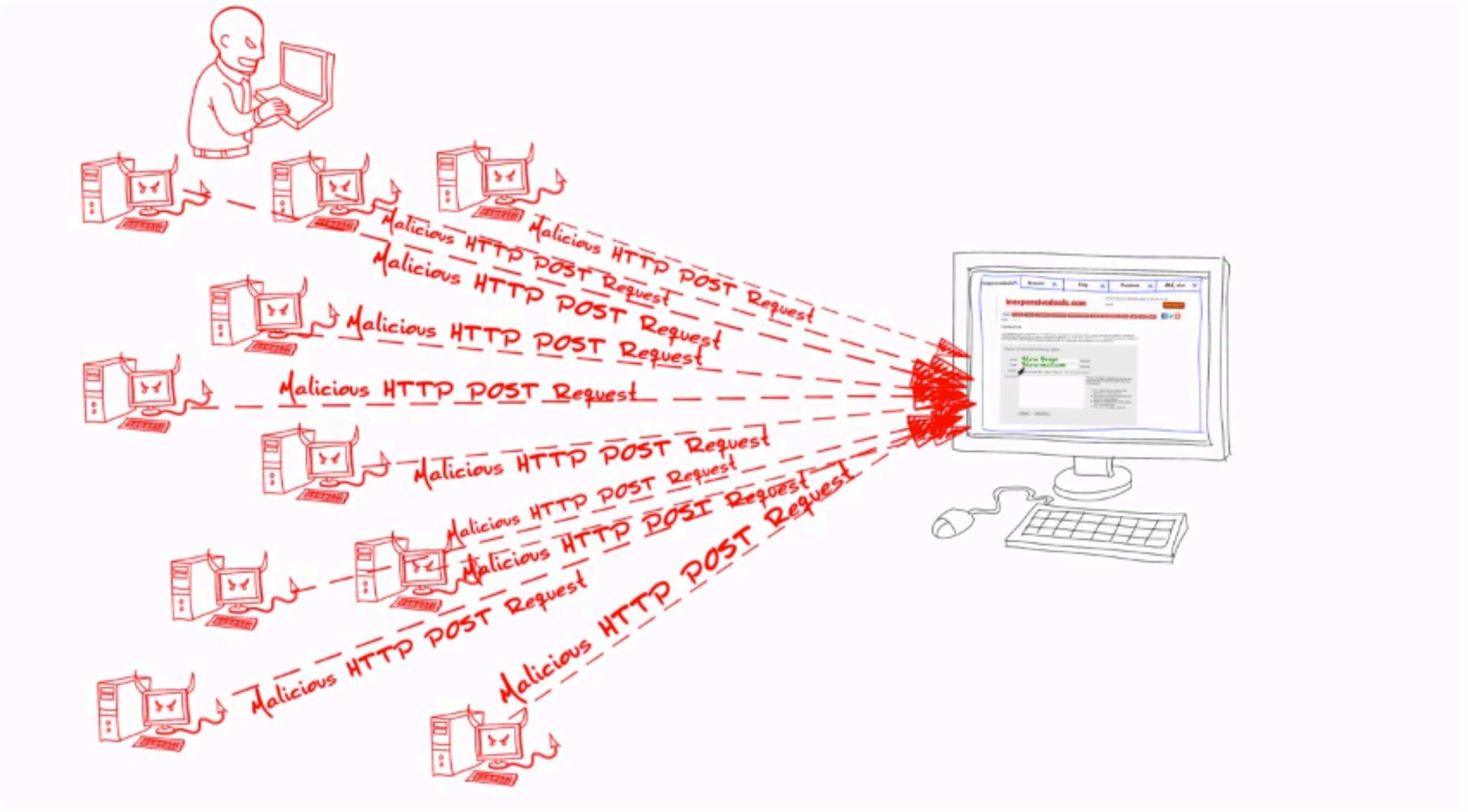
Layer 7 DDoS Web Attack

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

HTTP GET DDoS Attack



HTTP POST DDoS Attack

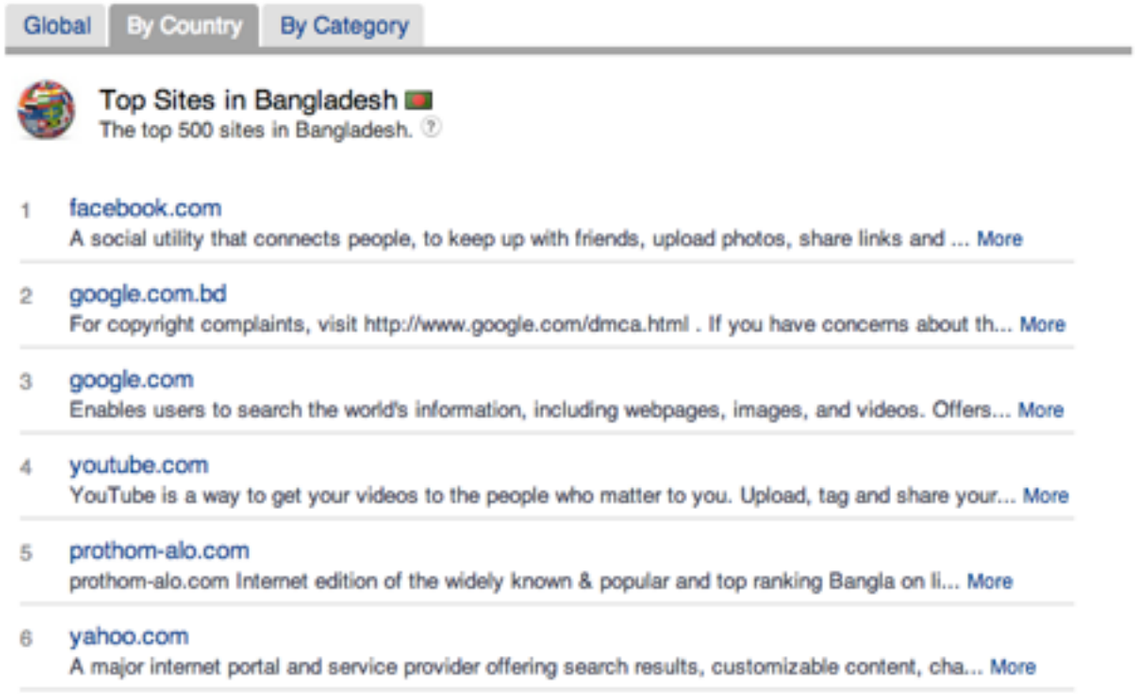


CASE STUDY

PRACTICAL APPROACH

Case Study

- Time: August 2012
- Country: Bangladesh
- Site: www.prothom-alo.com
- Ranked top 5 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

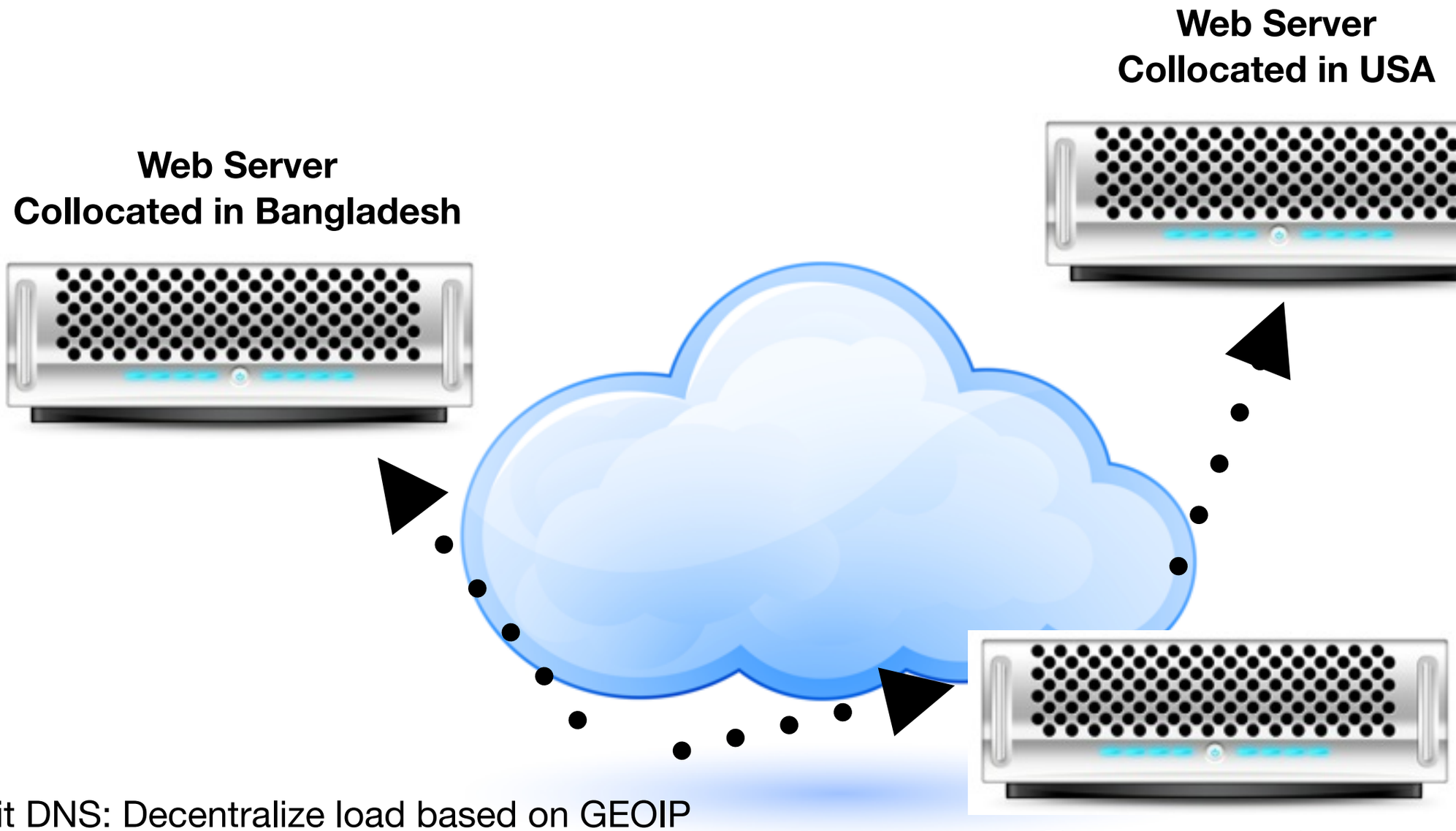
Approach 4

- 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

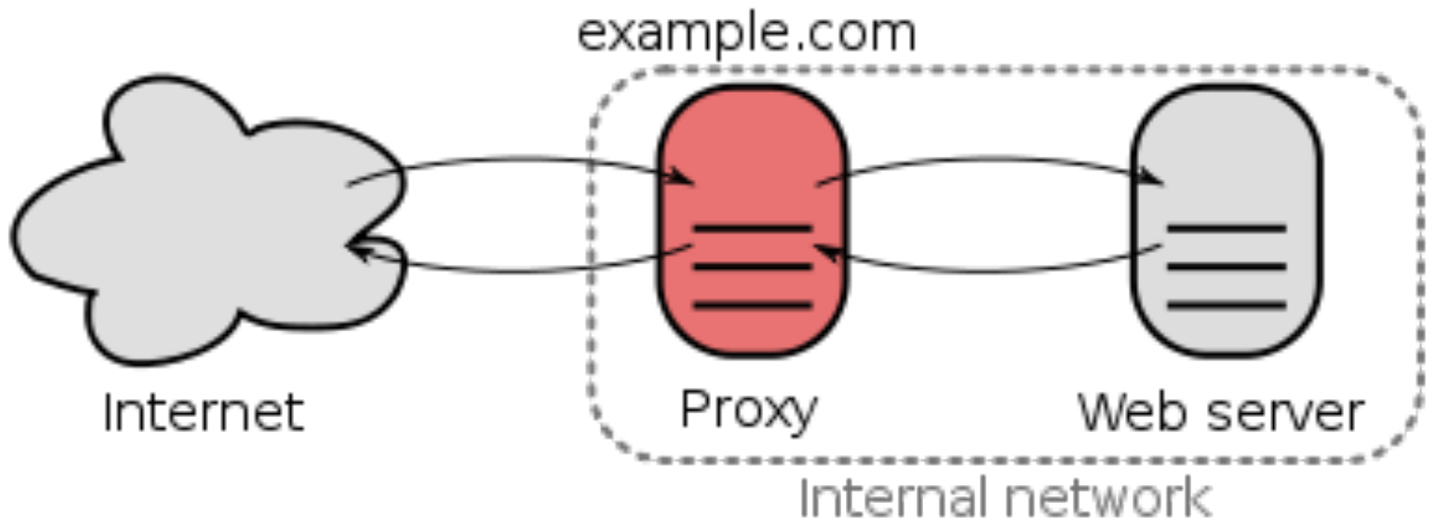
Solution Architecture



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]

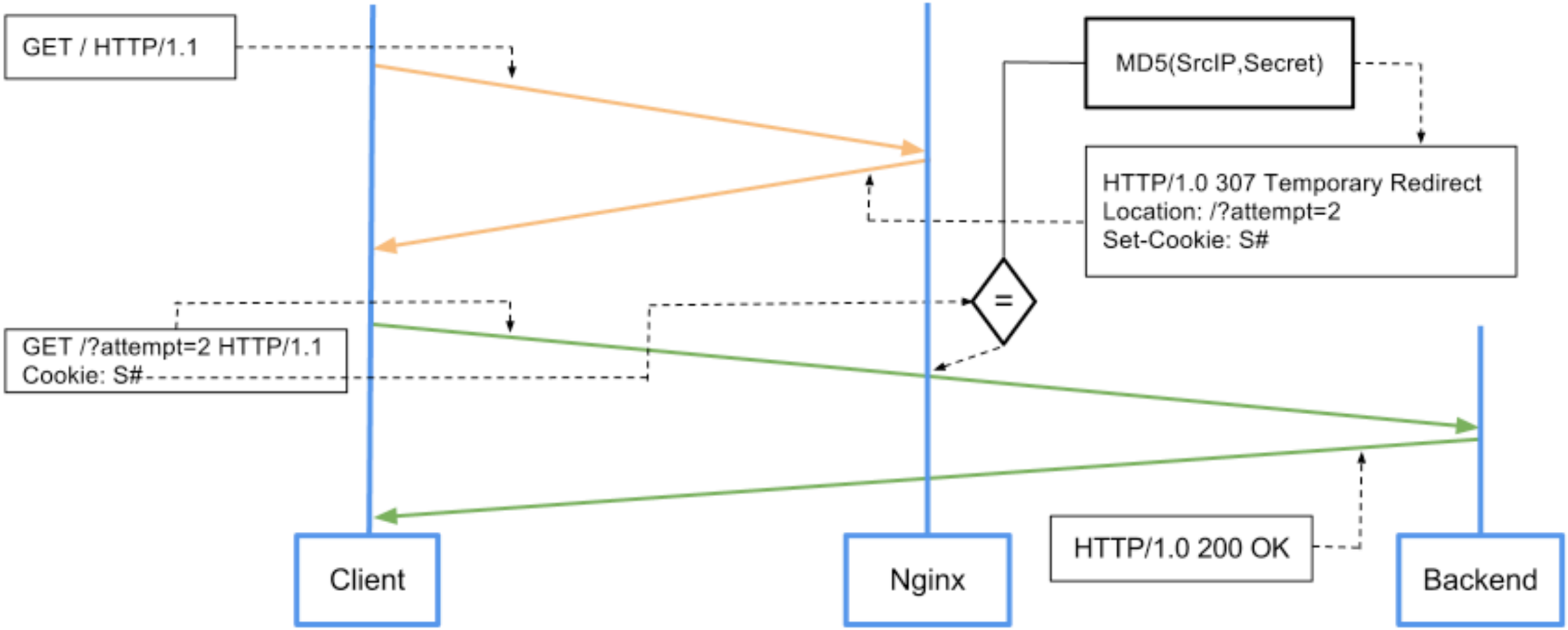
testcookie-nginx-module

- 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

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

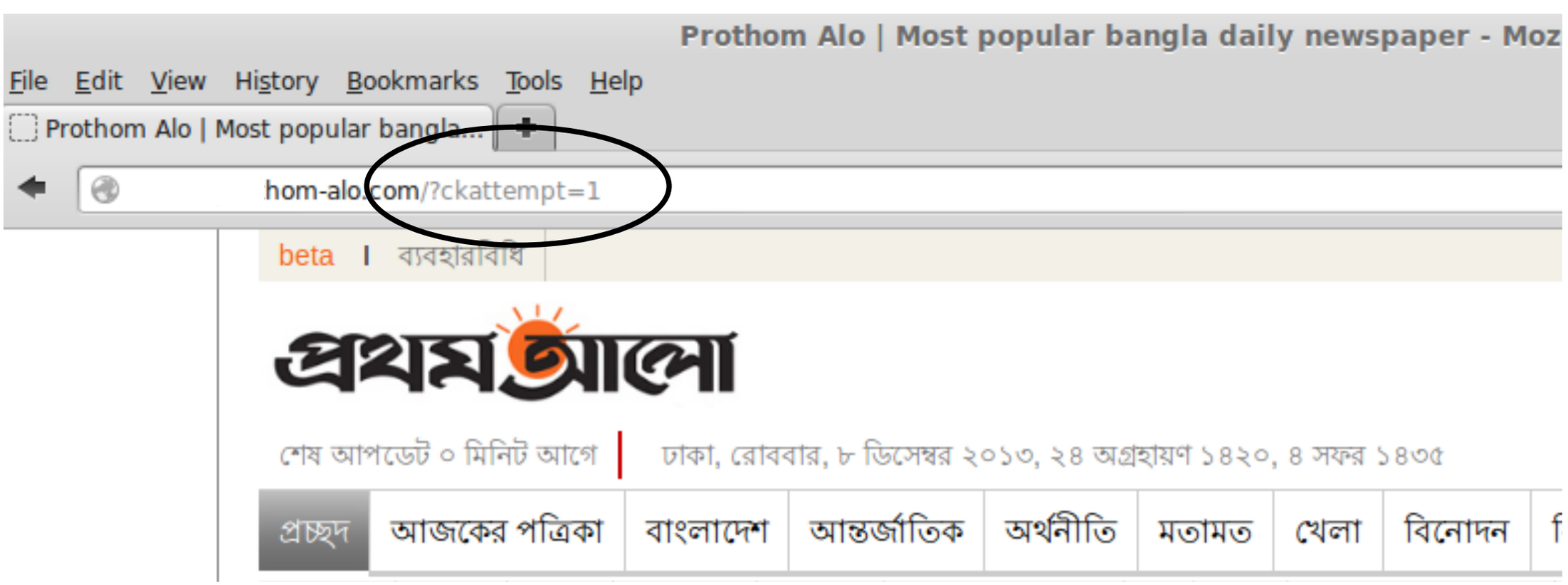
```
202.4.100.35 - - [28/Nov/2013:14:05:10 +0600] "GET / HTTP/1.1" 200
669 "-" "Mozilla/5.0 (Macintosh; Intel Mac OS X 10_9_0) AppleWebKit/
537.36 (KHTML, like Gecko) Chrome/31.0.1650.57 Safari/537.36"

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

1

Web Server : NGINX

Reverse Proxy (nx.bdnog.org)
192.168.1.100

2

Web Server : Apache

Web Server (web.bdnog.org)
192.168.1.150

3

Attack Server

Attack Server (attack.bdnog.org)
192.168.1.200

- 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

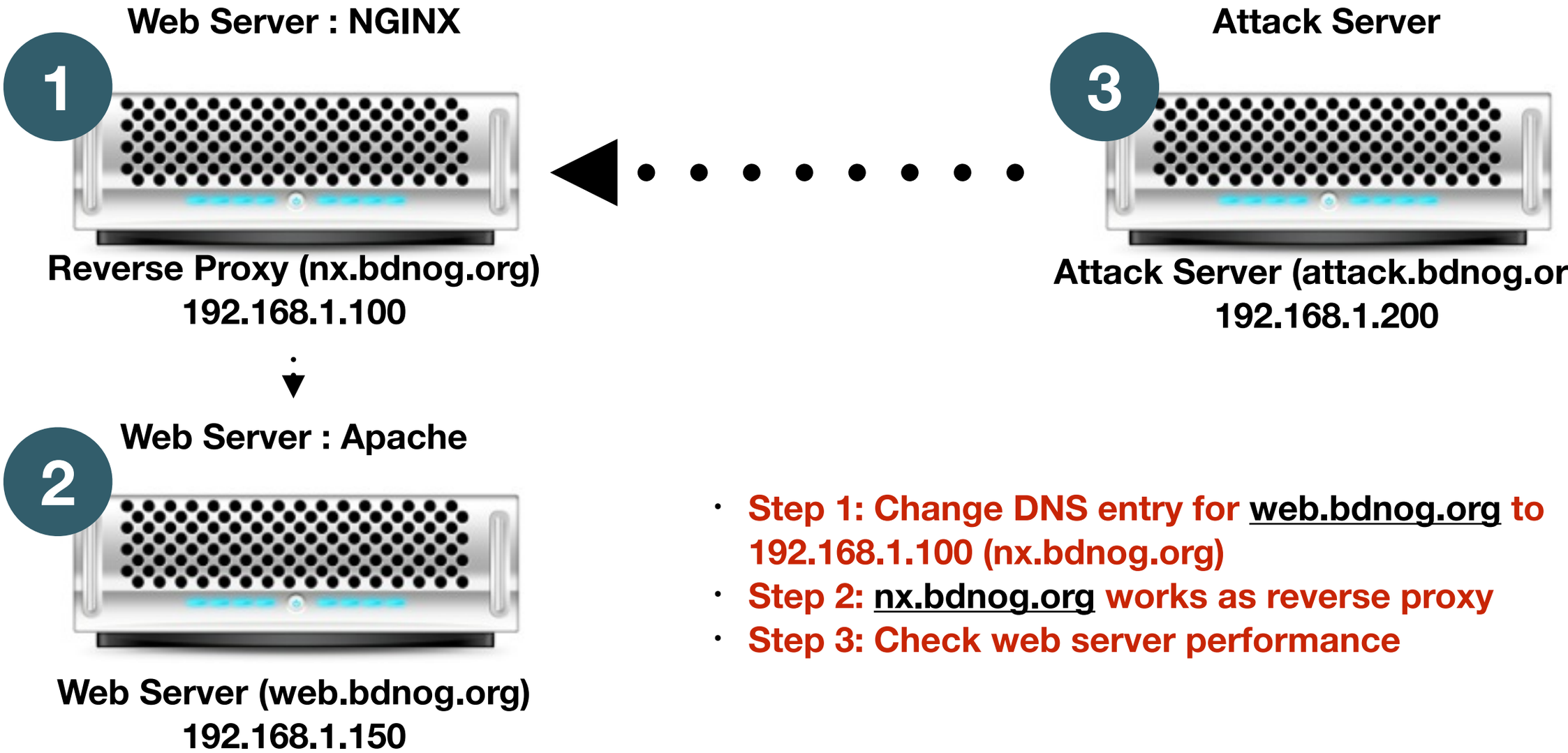
Web Server : Apache

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/

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.

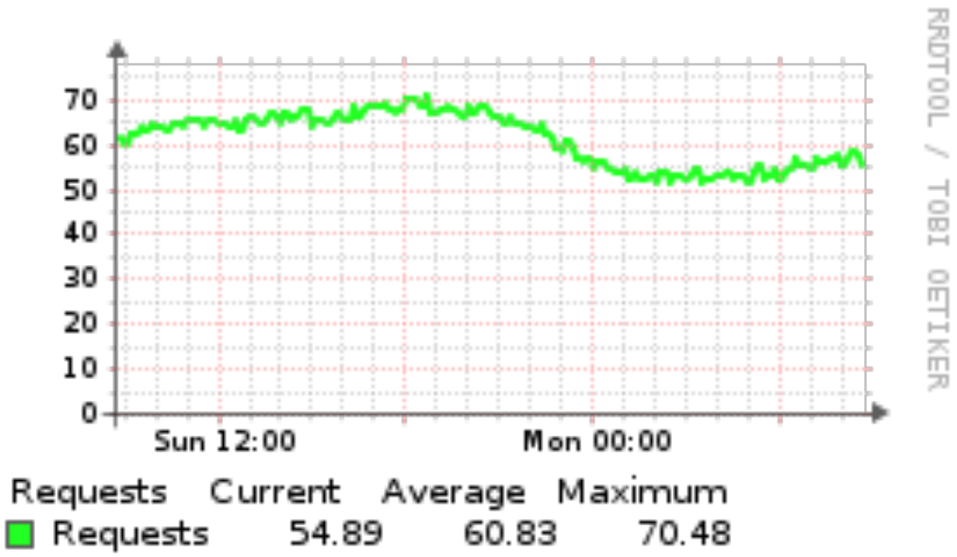
FINDINGS

Findings

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

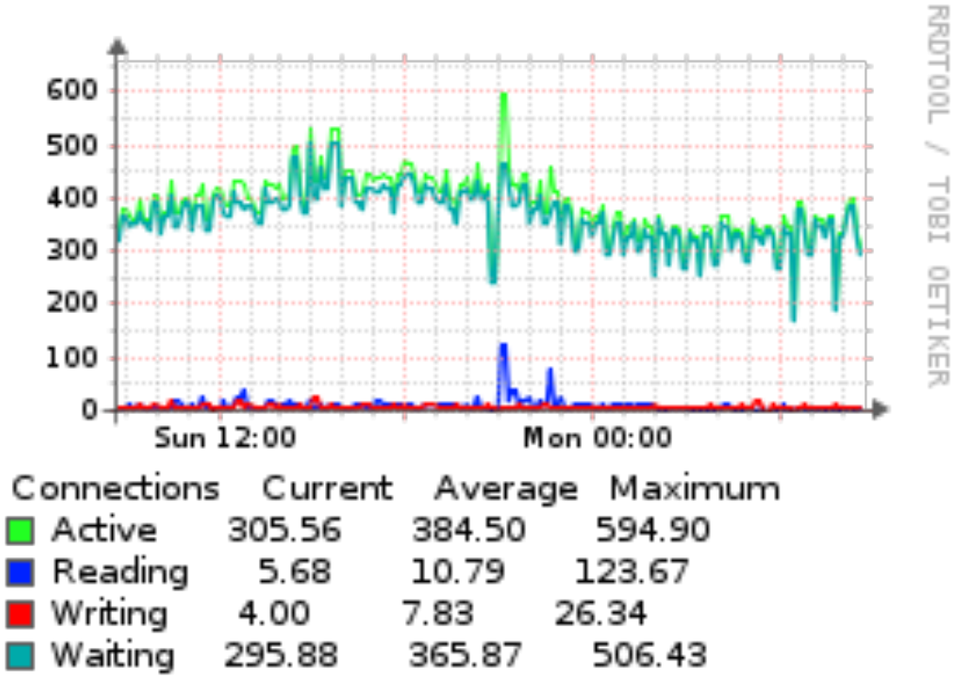
Monitoring

- Monitoring NGINX/Apache with Observium



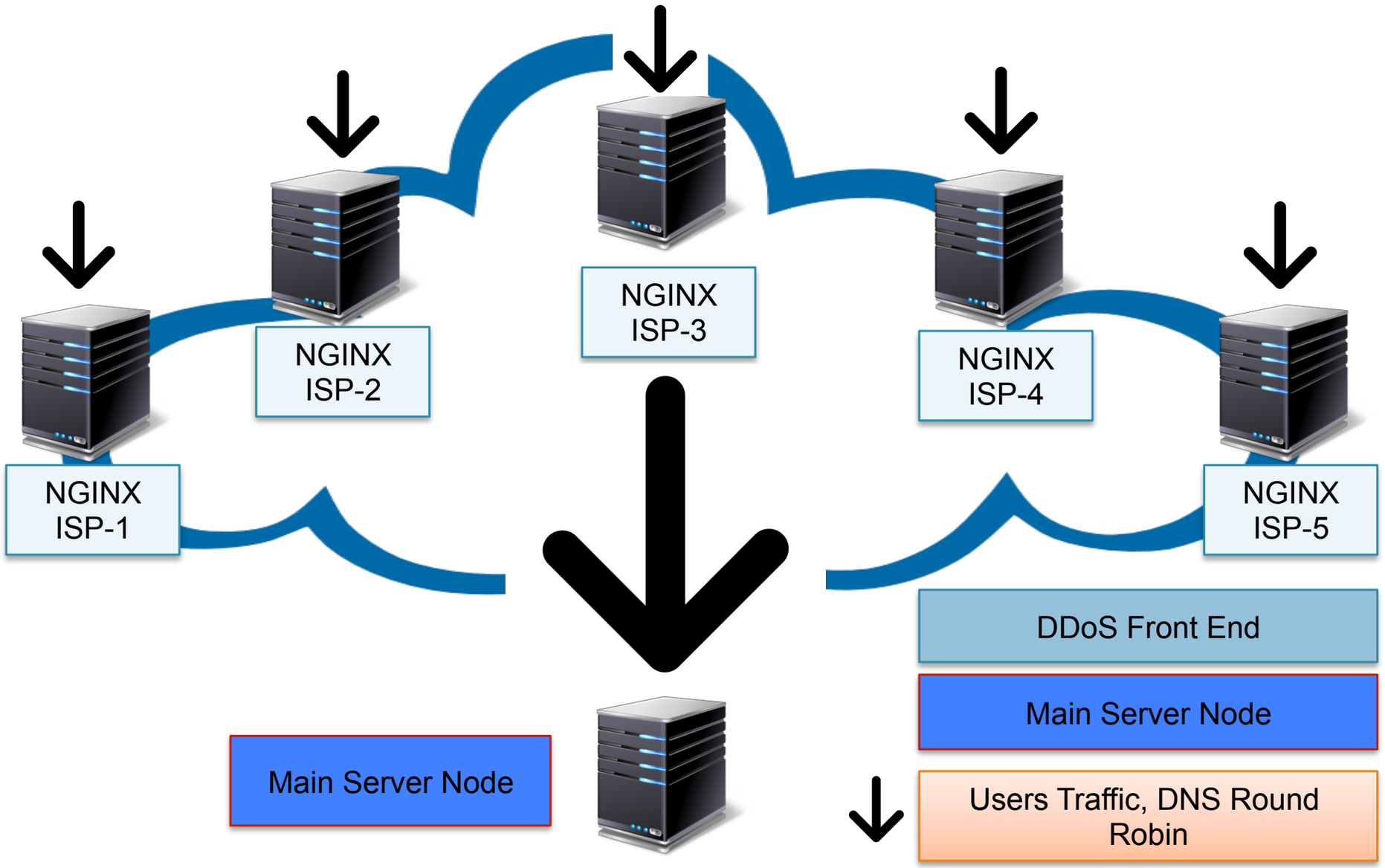
Request Statistics

Server Status



Log Analysis (logstalgia)

Off-loading & Splitting Traffic

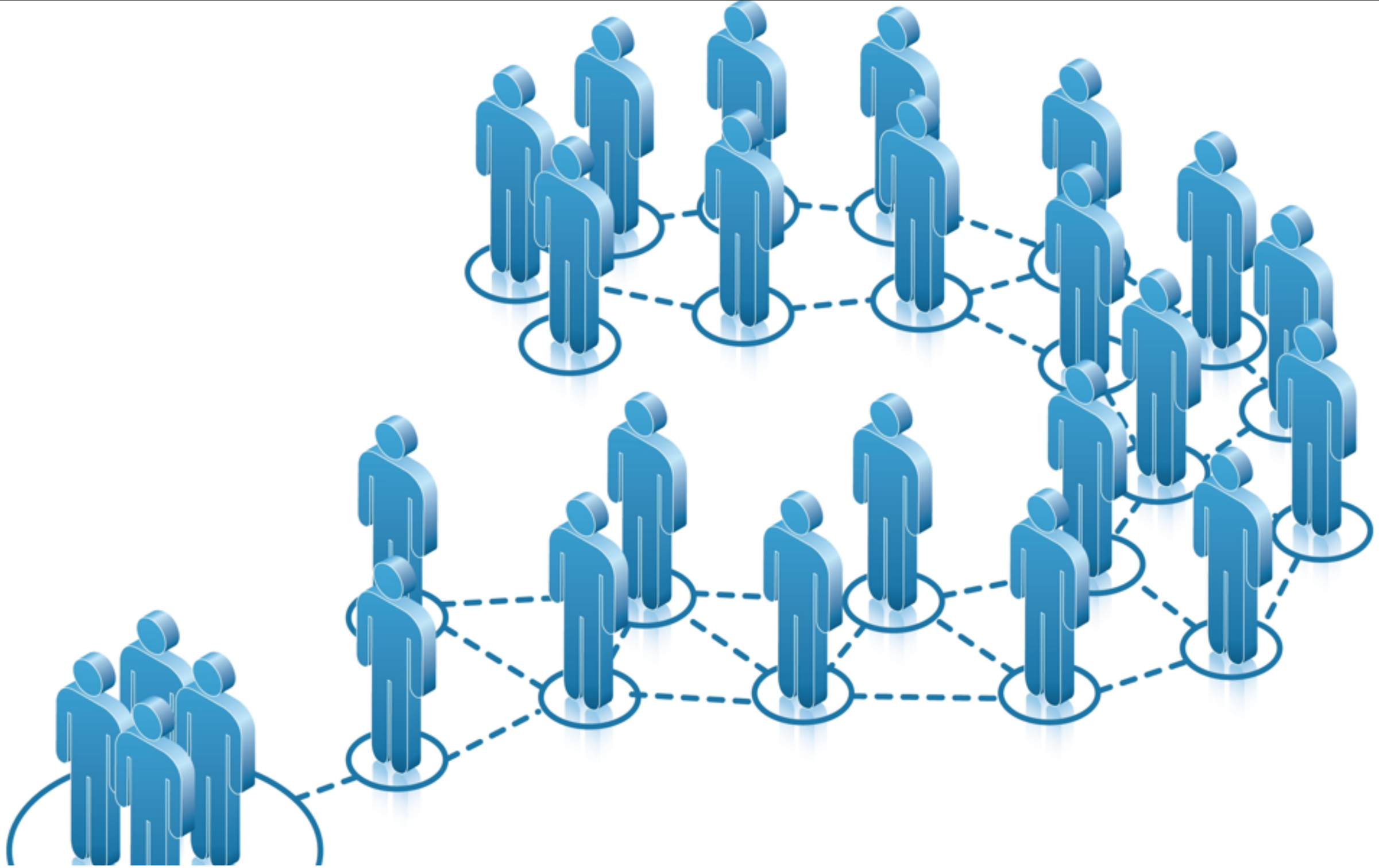


Issues

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

Special Thanks

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



QUESTION