

# **DDOS ATTACKS: PREPARATION-DETECTION-MITIGATION**

SANOG 21

January 27 - Feb 4, 2013

Cox's Bazar, Bangladesh

**Mohammad Fakrul Alam**

**bdHUB**

fakrul [at] bdhub [dot] com

# AGENDA

1. Overview of (D)DoS
2. How to (D)DoS
3. Motivation
4. Attack Type
5. Detection
6. Preparation
7. Mitigation
  - Layer 4 DDoS
  - Layer 7 DDoS
  - Link-Local DoS: IPv6 RA Attack

# (D)DOS: A REAL WORLD EXAMPLE



## Imagine a Restaurant

1. People come & order

2. The waiter takes their order

3. Served as the waiter becomes free

# (D)DOS: A REAL WORLD EXAMPLE



Suddenly, hundreds or thousands of customers come in and order a glass of water.

The waiter becomes overwhelmed with the quantity of requests. As more customers enter the restaurant the waiter is unable to attend to them because they are so overwhelmed.

That is a (D)DoS:  
making a resource unavailable by overloading.

# (D)DOS: IN THE COMPUTING WORLD

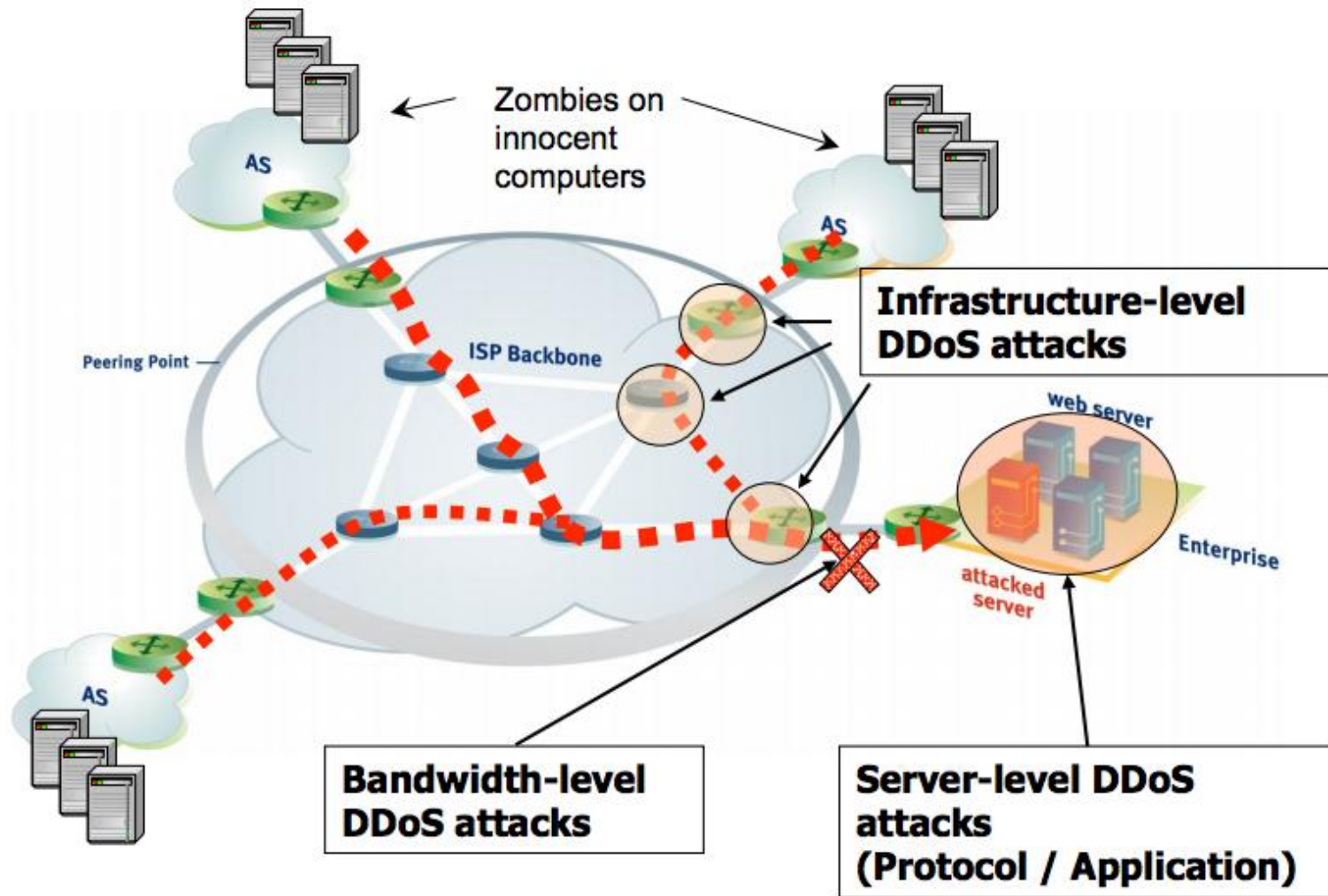
- In computing, a denial-of-service attack (DoS attack) or distributed denial-of-service attack (DDoS attack) is an attempt to make a machine or network resource unavailable to its intended users.
- It is a "Denial of Service". The server is never compromised, the databases never viewed, and the data never deleted. Throughout and after the attack, the server remains intact.
- Compromise "A" of CIA.

**Confidentiality**

**Integrity**

**Availability**

# (D)DOS: IN THE COMPUTING WORLD



# DO I HAVE TO CARE

“To **expect** the world to treat you fairly, because you’re a **good person**, is somewhat like asking a bull not to **attack** you, because you’re a **vegetarian!**”

- Quote from the Reader’s Digest

# HOW TO (D)DOS

Click really, really fast the “retry/reload” button

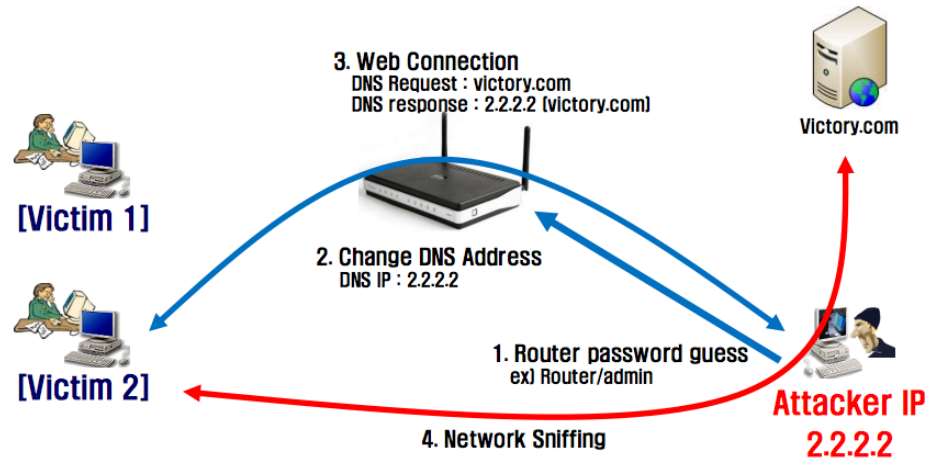
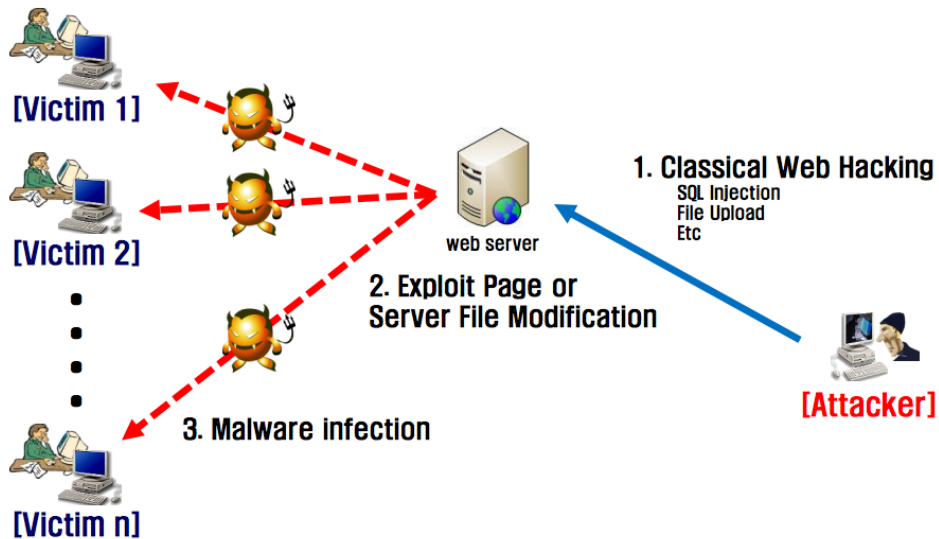
Scale vertically-> Recruit your friends/kids to do so

Scale horizontally -> Get a BOT



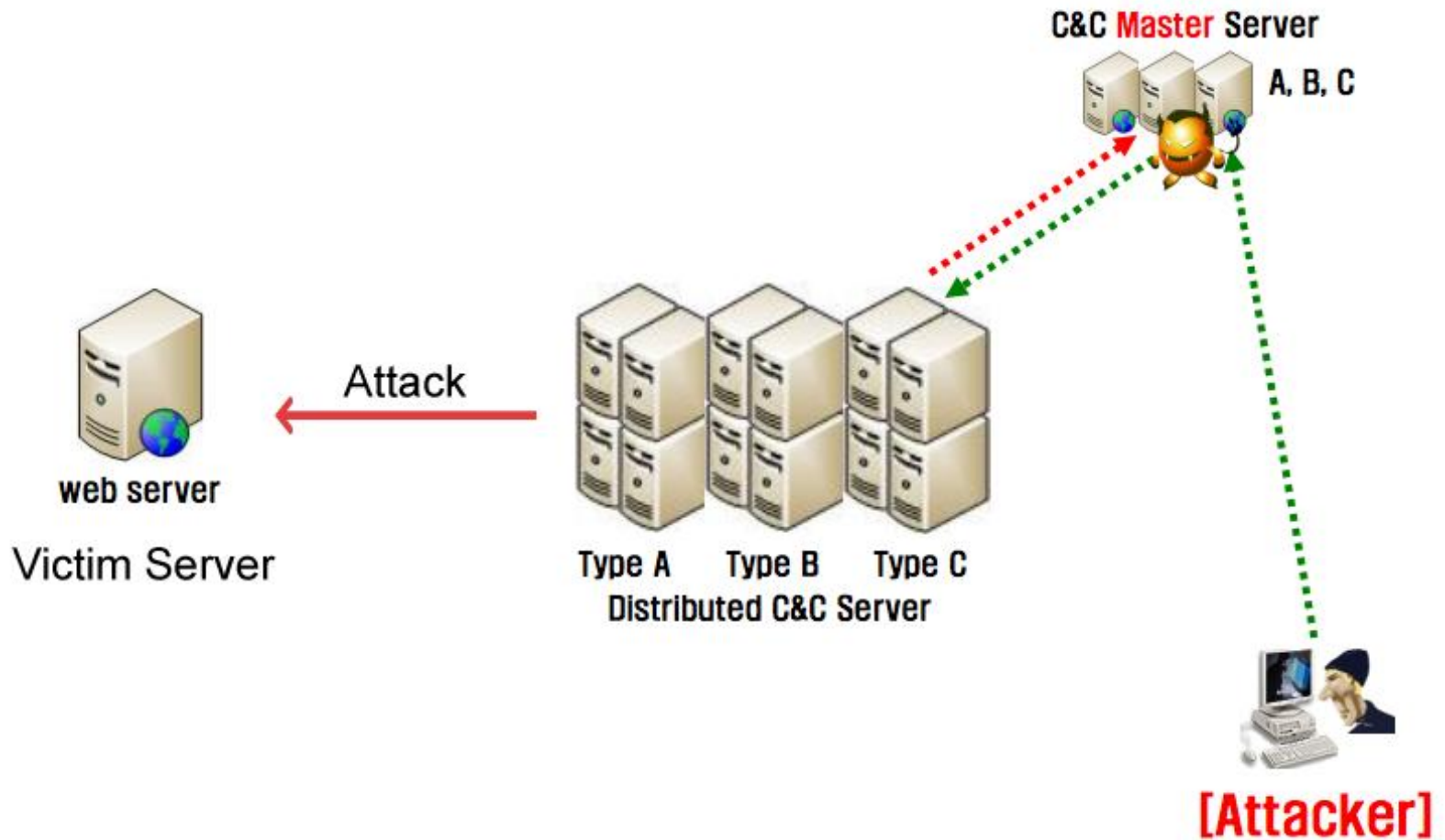
# HOW TO (D)DOS: BOT

Conventional way of creating BOT.



New technique, Rouge AP

# HOW TO (D)DOS: C&C



# MOTIVATION

## Financial

- Competition
- Extortion
- Divert attention
- Proof of Power

## Political

- Hacktivism
- “I’m a cooler kid than you”

# **ATTACK TYPE**

- **Asymmetric**
  - **DNS queries**
  - **SYN flood**
- **Symmetric**
  - **GET flood**
- **Reflected**
  - **Smurf/DNS (spoofed source)**
- **Brute force or logic state attacks**
- **Distributed**
  - **Any of the above (and many more)**

# ATTACK SURFACE (CLASSIFIED BY LAYER)

## TCP/IP Model

Application Layer

Transport Layer

Internet Layer

Network Access Layer

## OSI Model

Application Layer

Presentation Layer

Session Layer

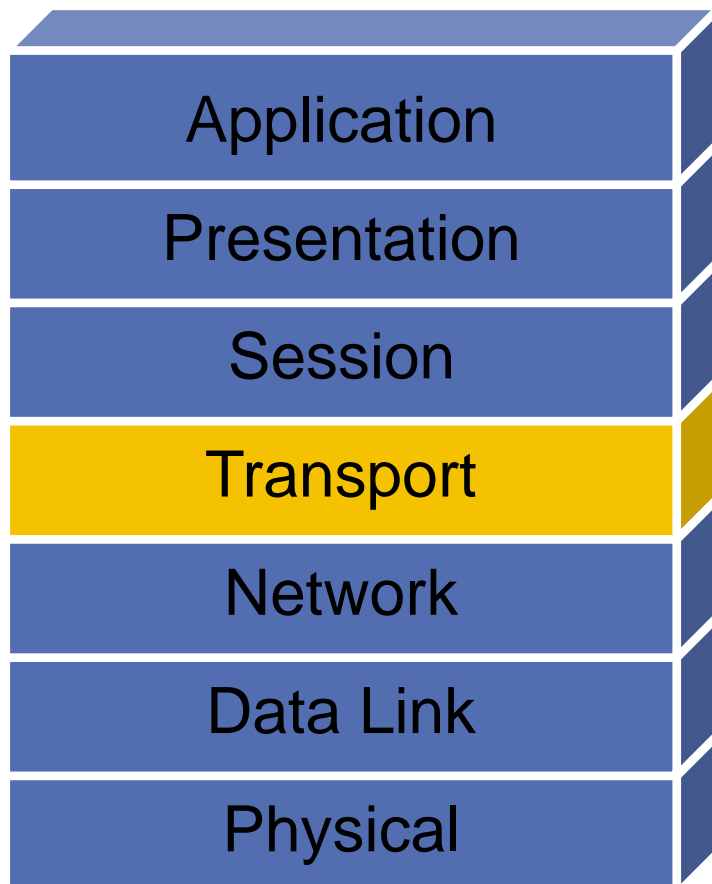
Transport Layer

Network Layer

Data Link Layer

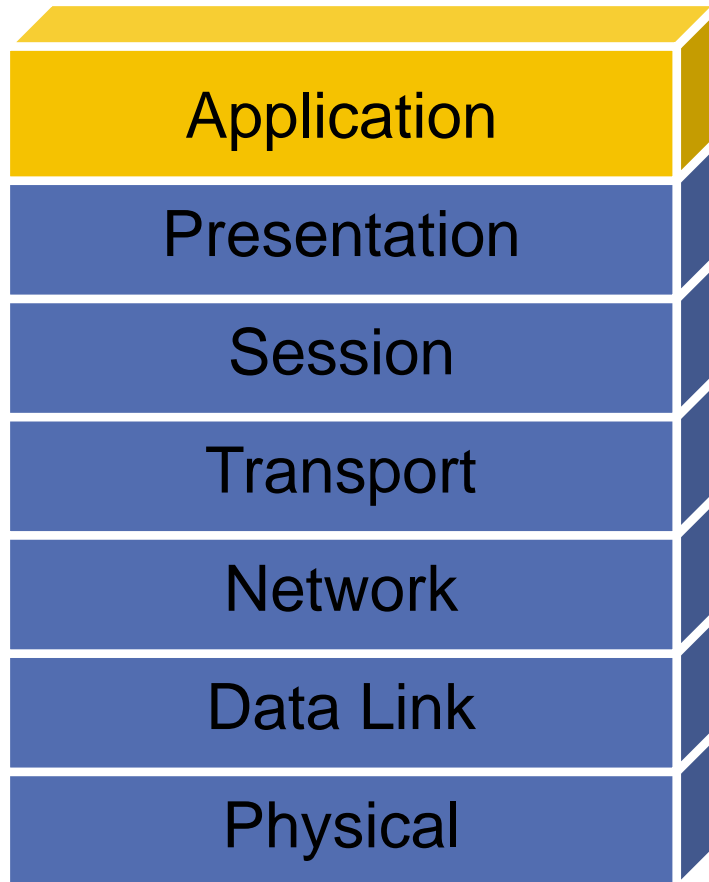
Physical Layer

# LAYER 4 ATTACK



- SYN Flood
- RST Flood
- FIN Flood

# LAYER 7 ATTACK



- SPAM
- DNS Queries
- HTTP GET Flood

# **DETECTION: GAME OF RESOURCE EXHAUSTION**

- Bandwidth
- PPS
- Storage
- CPU
- Application specific



# SIMPLIFIED TCP STATE MACHINE

- LISTEN – waiting for a connection request
- SYN\_RECV – received request still negotiating
- ESTABLISHED – connection working OK
- FIN-WAIT1/2 – one side closed the connection
- TIME-WAIT – waiting for a while

# LIFE OF A SOCKET

- Socket = TCP/UDP port + IP address
- Normal connection

```
root@access:/home/fakrul# netstat -nap | grep 8080
```

```
tcp          0          0 :::8080      :::*         LISTEN      1426/apache2
```

```
root@access:/home/fakrul# netstat -nap | grep 8080
```

```
tcp 0 0 192.168.1.250:8080 192.168.1.35:49560 ESTABLISHED 3918/apache2
```

```
root@access:/home/fakrul# netstat -nap | grep 8080
```

```
tcp 0 0 192.168.1.250:8080 192.168.1.35:49557 TIME_WAIT -
```

# DETECTION ON THE HOST

- Your best friend: netstat

```
netstat -nap
```

- Your next best friend: tcpdump

```
tcpdump -n -i <interface> -s 0 -w <target_file.pcap> -c <packet_count>
```

- Dedicated IDS (snort/suricata)

# PREPARATION & MITIGATION

## Key Points to Note:

1. You can only stop DDoS attacks after your own perimeter.
2. You can't stop DDoS attacks before your perimeter – unless others are ready to help you.
3. Your mitigation is as good as knowledge, tools and techniques you have at your disposal.

# IN PEACE TIME

- You should have your monitoring ahead of time.
- Have a Incident response plan.
- When do you need to escalate?
- Your security gear (if at all present).

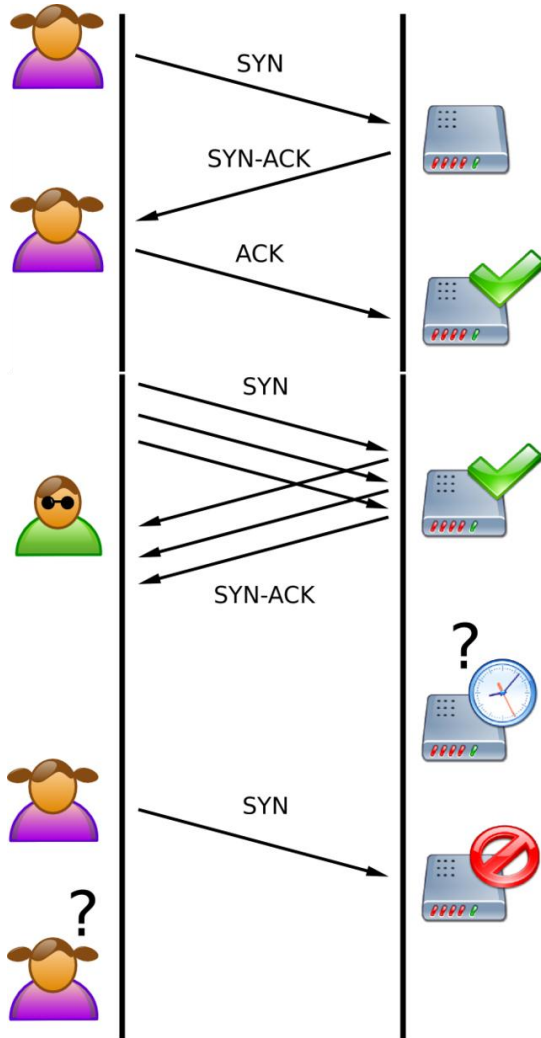
# MONITORING IMPACT

- The most neglected resource.
- No matter how much traffic they throw at you, there is no problem until your users start seeing it.
- Use internal monitoring.
- Use external monitoring services.

# IN THE HEAT OF THE MOMENT

- What is actually happening? Focus on the facts.
- Collect data (from Systems, Network Graphs, Capture Traffic).
- Create a response plan.
- Execute.
- Observe.

# SYN FLOOD



## What does it take:

- Think 3-way handshake.
- Server has a number of slots for incoming connections.
- When slots are full no more connections are accepted.



# HOW TO RECOGNIZE SYN FLOOD

Active Internet connections (servers and established)

Proto Recv-Q Send-Q Local Address Foreign Address State PID/Program

name

tcp 0 0 0.0.0.0:111 0.0.0.0:\* LISTEN 1339/rpcbind

tcp 0 0 0.0.0.0:33586 0.0.0.0:\* LISTEN 1395/rpc.statd

tcp 0 0 192.168.122.1:53 0.0.0.0:\* LISTEN 1962/dnsmasq

tcp 0 0 192.168.1.250:631 0.0.0.0:\* LISTEN 1586/cupsd

tcp 0 0 192.168.1.250:25 0.0.0.0:\* LISTEN 2703/sendmail: acce

**tcp 0 0 192.168.1.250:25 192.168.1.35:49718 SYN\_RECV -**

**tcp 0 0 192.168.1.250:25 192.168.1.35:49717 SYN\_RECV -**

**tcp 0 0 192.168.1.250:25 192.168.1.35:49722 SYN\_RECV -**

**tcp 0 0 192.168.1.250:25 192.168.1.35:49720 SYN\_RECV -**

**tcp 0 0 192.168.1.250:25 192.168.1.35:49719 SYN\_RECV -**

**tcp 0 0 192.168.1.250:25 192.168.1.35:49721 SYN\_RECV -**

**tcp 0 0 192.168.1.250:25 192.168.1.35:49716 SYN\_RECV -**

# SYN MITIGATION

## SYN Cookies

- Special hash
- Enable by:
  - `echo 1 > /proc/sys/net/ipv4/tcp_syncookies`
- Other timeouts to tweak (in `/proc/sys/net/ipv4/`):
  - `tcp_max_syn_backlog` [how many SYN requests to keep in memory that we have yet to get the third packet in a 3-way handshake from]
  - `tcp_synack_retries` [how many times to retransmit the SYN,ACK reply to an SYN request]
  - `tcp_syn_retries` [how many times to try to retransmit the initial SYN packet for an active TCP connection attempt]

# HOW TO RECOGNIZE SOCKET EXHAUSTION

Active Internet connections (servers and established)

Proto Recv-Q Send-Q Local Address Foreign Address State PID/Program

Name

tcp 0 0 192.168.1.250:631 0.0.0.0:\* LISTEN 1586/cupsd

tcp 0 0 192.168.1.250:25 0.0.0.0:\* LISTEN 2703/sendmail: acce

tcp 0 0 192.168.1.250:25 192.168.1.35:49718 TIME\_WAIT -

tcp 0 0 192.168.1.250:25 192.168.1.35:49717 TIME\_WAIT -

tcp 0 0 192.168.1.250:25 192.168.1.35:49722 TIME\_WAIT -

tcp 0 0 192.168.1.250:25 192.168.1.35:49720 TIME\_WAIT -

tcp 0 0 192.168.1.250:25 192.168.1.35:49719 TIME\_WAIT -

tcp 0 0 192.168.1.250:25 192.168.1.35:49721 TIME\_WAIT -

tcp 0 0 192.168.1.250:25 192.168.1.35:49716 TIME\_WAIT -

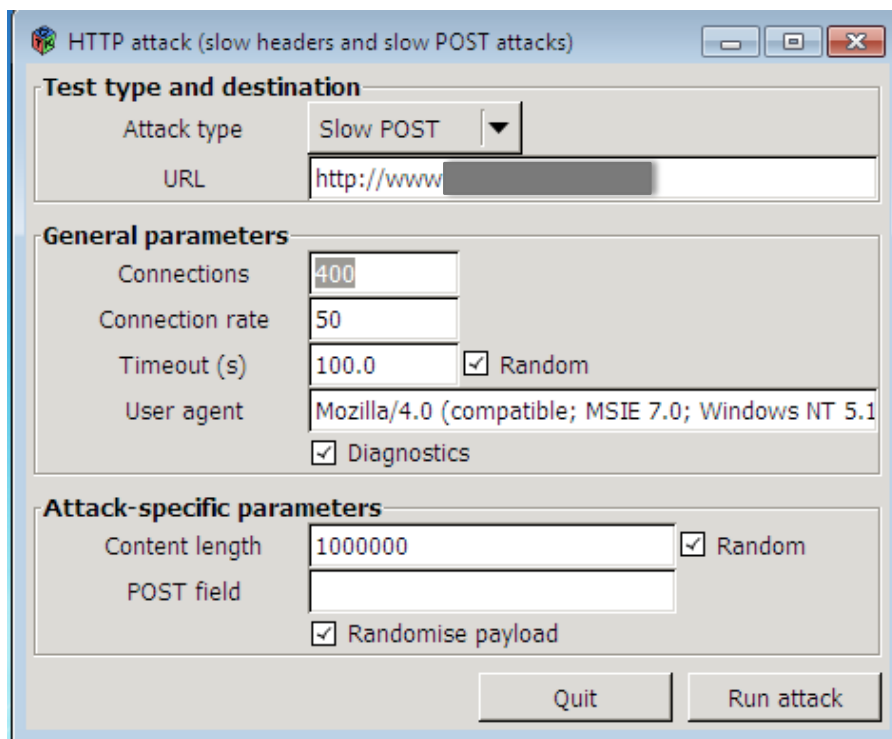
# SOCKET EXHAUSTION/CONNECT MITIGATION

- 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`
- Check learn about the value in
  - `/proc/sys/net/ipv4/tcp_*`



# UPPER LAYER (D)DOS ATTACK

- **OWASP HTTP POST Tool & R-U-Dead-Yet**
  - **Incomplete HTTP POSTs.**
  - **Stops IIS, but requires thousand of packet per seconds.**



```
$ ./r-u-dead-yet.py
```

```
http://localhost/upload.html
```

# MITIGATION UPPER LAYER

- **Architecture of applications**
  - Apache – process based – In Linux kernel level threads
  - Nginx – event based
- **Mitigation through challenges**
  - Nginx plugin – Roboo (ECL-LABS.ORG)
  - Apache - ModSecurity
- **Load Balancer**
- **Split DNS!!**

# MITIGATION UPPER LAYER

- Nginx plugin – Roboo Configuration

```
perl Roboo::handler;  
  
set $Roboo_challenge_modes    "SWF,gzip";  
  
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;
```

## Sample challenged.log

```
192.168.1.250 - - [08/Nov/2012:13:03:47 +0600] "GET /Anti-Robot-GET-f9e5de6f1f226fbb7472.swf HTTP/1.1" 200 1023 "http://192.168.1.1/"  
"Mozilla/5.0 (Macintosh; Intel Mac OS X 10_7_5) AppleWebKit/537.4 (KHTML, like Gecko) Chrome/22.0.1229.94 Safari/537.4"
```



# MITIGATION UPPER LAYER

- **Split DNS!!**

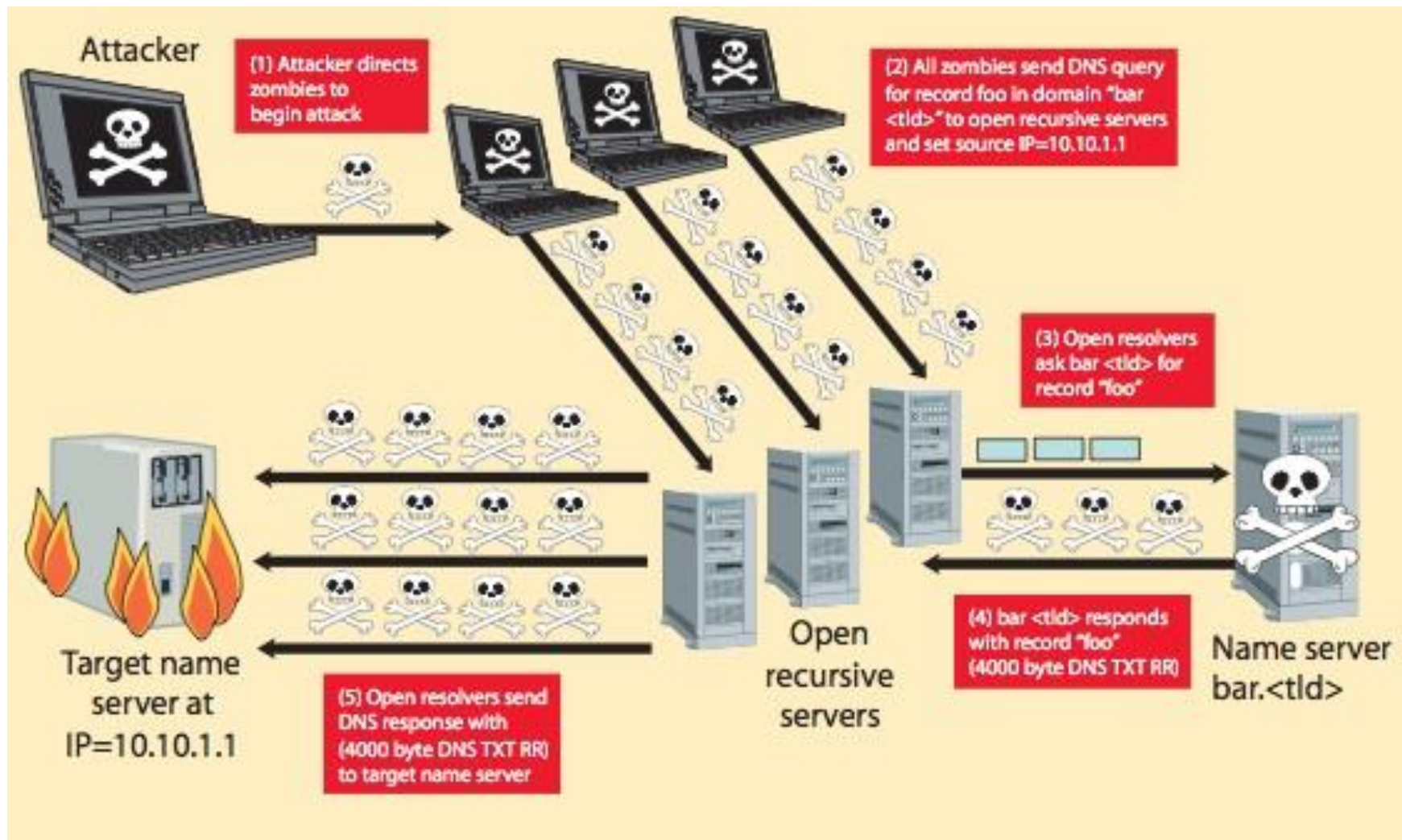


- DNS resolve based on the users source.
- Possible to distribute the (D)DoS load.
- BIND ACL with IP.
- GeoDNS BIND patch.

# DNS AMPLIFICATION DDOS ATTACK

- Attacks using IP spoofed DNS query
  - Generating a traffic overload
  - Bandwidth attack
  - Similar to 'smurf attacks'
- Components are:
  - IP spoofing
  - DNS amplification

# DNS AMPLIFICATION DDOS ATTACK



# DNS AMPLIFICATION DDOS ATTACK

- dig ANY isc.org @x.x.x.x +edns=0

-->output truncated<--

:: AUTHORITY SECTION:

isc.org.	3569	IN	NS	ams.sns-pb.isc.org.
isc.org.	3569	IN	NS	sfba.sns-pb.isc.org.
isc.org.	3569	IN	NS	ord.sns-pb.isc.org.
isc.org.	3569	IN	NS	ns.isc.afiliast-nst.info.

:: ADDITIONAL SECTION:

ns.isc.afiliast-nst.info. 82769	IN	A	199.254.6
ns.isc.afiliast-nst.info. 82769	IN	AAAA	2001:500:

:: Query time: 79 msec

:: SERVER: 103.12.179.12#53(103.12.179.12)

:: WHEN: Mon Nov 26 16:50:09 2012

**:: MSG SIZE rcvd: 3191**

That's a 64 byte query that resulted in a 3,191 byte response. In other words, an attacker is able to achieve a 50x amplification over whatever traffic they can initiate to an open DNS resolver.

# MITIGATION DNS AMPLIFICATION

- Disable Open Recursive DNS
  - For BIND 9.x authoritative servers, apply the following global options:

```
options {  
    recursion no;  
    additional-from-cache no;  
};
```

- For BIND 9.x caching servers

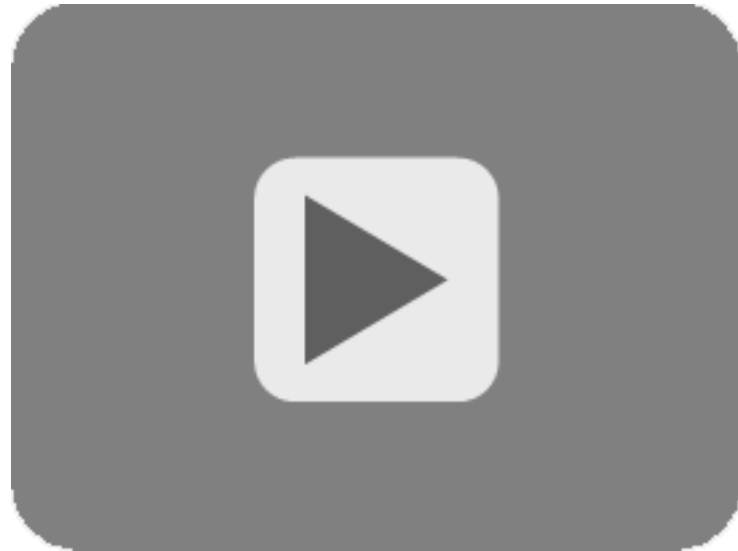
```
acl "trusted" {  
    192.0.2.0/24;  
};  
options {  
    recursion no;  
    additional-from-cache no;  
    allow-query { none; };  
};
```

```
view "trusted" in {  
    match-clients { trusted; };  
    allow-query { trusted; };  
    recursion yes;  
    additional-from-cache yes;  
};
```

# LINK LOCAL DOS: IPV6 RA ATTACK

1. A single device can instantly stop all the Windows machines on a Local Area Network.
2. Effected OS:
  - Windows XP, Vista, Windows 7, Server 2008
  - It is also reported that X-Box & PS3 is also effected.
  - FreeBSD
3. CVE-2010-4669
  - CVSS Severity: High

# LINK LOCAL DOS: IPV6 RA ATTACK



Live Demonstration

# MITIGATION LINK LOCAL DOS

- Disable IPv6.
- Turn of Router Discovery.
- Use a firewall to block rogue Router Advertisements.



# FEW RECOMMENDATION

1. Stop spoofed TCP attacks at your perimeter.
2. Don't let dark address packets pass your perimeter.
3. Block unused protocols and ports.
4. Limit number of access per second per source IP.
5. Limit number of concurrent connections per source IP.
6. Don't forward packets with header anomalies.
7. Monitor self similarity in traffic.

# MAKE YOUR FRIENDS

- Look around, who else might be suffering this?
- Build partnerships.
- Build social contacts.
- Prepare before it hits.

**QUESTIONS?**