DoS & DDoS attacks – and how they can be prevented

Perhaps, one of the oldest attack maneuvers is some variation of "denial of service". The target service may stay standing and even online, but the ultimate goal of non-service will be achieved. The most primitive and naive way to achieve this works amazingly well even today, but defenses are higher than they were some years ago. Think about a simple example of real-life DDoS attack, like having two competing restaurants, A and B operating next to each other. For some reason, one of the owners, let's say A, or perhaps the authorities, C, get annoyed and want to deny B from serving pizza any more. There are good and bad ways to achieve this goal, and then there are the ugly ones. DDoS attacks, perhaps, fall between the categories of bad and ugly.

The key thing in primitive denial of service attack is to send the target, the pizzeria B in our case, many customers. And many means really many. Too much. And a bit more. No matter how good the chef is and how cheap the food is, quite quickly the pizzeria cannot serve any more customers. Job well done or not. A similar attack works in the cyberspace, although variations are perhaps more than in the case of pizzeria. Sending packets from all over the world; millions of packets, angry packets, malformed packets and even confusing packets will make just about any server or router stop working for a while.

What kind of damage can these attacks do to an organization?

Companies can stay resilient and recover quickly even after a massive denial of service attack. Even when their servers were on fire and offices collapsing, a well-prepared company might get up and running soon again. But in real life, variations of denial of service attacks are many and also some companies keep a full backup running all the time, while others may not have resources for such. In the pizzeria case, the very worst case can be ultimate bankruptcy of the restaurant if those customers all file a complaint demanding refund, for example, or if they come in as trainees and steal all the time and attention of the staff, etc. Reasons for fatal failure can be many. Some companies might lose millions, like what recently happened in the Petya attack of the global logistics company, Maersk and TNT who got days, if not weeks, of "denial of service", as a nasty piece of ransomware took their assets hostage.

In the best case, this kind of attacks may cause only temporal or distributed downtime. One of the most common strategies today, for web applications, is to distribute the downtime and treat suspicious requests as potential attacks. This way, the whole application can stay up longer, while small parts of it might experience even fatal and long-lasting downtime.

How can Safewalk help prevent DDoS attacks?

There are basically two layers to a DDoS protection. One, and the most basic, is on the networking level. Traditionally, DDoS attacks were performed on this level by flooding the application with millions and millions of random or malformed, if not even hostile, packets. There is nothing an application can do to prevent this, as they occur at a low level and will attack the network interface and routers, and may not even reach an application that could block them off.

However, some modern DDoS attacks may target specific functions of an application. An attacker may want to launch an attack against the authentication function of an application and prevent access to the application by locking down users' accounts or overloading the authentication procedure. This could be done either on purpose to brute-force login credentials or just by issuing random login requests without an intention to be successful.

A traditional coping strategy against application-level DDoS attacks, as well as brute-force attack, is setting a request failure counter - too many failed attempts and the user gets blocked.

However, having a user blocked and then manually unblocked is not a practical solution and can still lead to a denial of service. As such, a more advanced mechanism has been introduced into the Safewalk platform where a user can get blocked temporarily when the system detects too many failed attempts within a predefined time interval. After the temporary lock-down time has passed, the user can still authenticate and gain access to the system. This will work also as practical defense against simple application-level DDoS attacks.

By locking down the user account, network source, or both, this type of defensive maneuver can ensure that a continuing attack would not overload the database, audit functions, application-level features, etc. This time-dependent block is valid for a short time and will recover automatically.