The Catch-22 Attack

Lumin Shi
luminshi@cs.uoregon.edu
University of Oregon

Devkishen Sisodia
dsisodia@cs.uoregon.edu
University of Oregon

Mingwei Zhang
mingwei@caida.org
CAIDA, UC San Diego

Jun Li
lijun@cs.uoregon.edu
University of Oregon

Alberto Dainotti
alberto@caida.org
CAIDA, UC San Diego

Peter Reiher
reiher@cs.ucla.edu
UCLA

ABSTRACT

In this work, we introduce the Catch-22 attack, a distributed denial-of-service (DDoS) link-flooding attack that exploits real-world limitations of DDoS defense. An attacker in the Catch-22 attack leverages virtual private server (VPS) providers and residential proxy services as vehicles for assembling a botnet, and employs moving target attack techniques to not only maximize the amount of strain on DDoS defense, but also maximize the amount of collateral damage incurred by attacked networks, thereby wreaking havoc on wide swaths of the Internet. In fact, according to our preliminary evaluation, the Catch-22 attack can cause significant collateral damage to over thousands of websites from a major VPS provider. To the best of our knowledge, no existing work has yet to present a solution for such an attack, let alone study it.

1 INTRODUCTION

The scale, frequency, and complexity of DDoS attacks are forcing edge networks to seek much-needed protection. This is no surprise as edge networks have evolved to connect more end devices with ever-increasing bandwidths, meanwhile, many of these devices are exploitable for launching DDoS attacks. Today, websites and small networks often subscribe to DDoS protection services (DPSes) such as Akamai, Cloudflare, and NETSCOUT Arbor for DDoS defense. Unfortunately, DPSes have finite bandwidth and computational capacity, and thereby cannot provide protection to all networks.

For the remaining networks that are not protected by DPSes, the network community utilizes features baked into modern IP routers – n-tuple traffic matching – for fine-grained DDoS filtering. Protocols such as BGP FlowSpec [4], allow networks to disseminate n-tuple filtering rules to their neighboring autonomous systems (ASes), and the neighboring ASes can then verify and deploy the rules to mitigate the attacks *inline*. Recent research such as SENSS [6], proposed a DDoS defense framework to automate and secure the inline mitigation process.

Inline mitigation assumes attacked networks can derive attack signatures for generating n-tuple rules to filter the DDoS traffic. Attack signatures are derived from packet-level information, such as IP header and payload fields, of the attack traffic. The challenge for the attacked networks is to derive attack signatures that will lead to 1) efficient DDoS filtering and 2) low collateral damage. In this paper, we focus on source IP-based rules, because they allow for more fine-grained filtering, thereby reducing the potential of collateral damage better than any other individual field.

For source IP-based filtering, the main limitation is the limited amount of memory on network routers in which filtering rules can be deployed. Network routers rely on ternary content-addressable memory (TCAM) to forward or discard traffic with low latency in the data plane, but it is extremely difficult to scale TCAM while maintaining a low latency [1]. In fact, most routers today do not support more than 10s of thousands of access control list (ACL) rules in their data plane [2].

In this work, we shed light on a technological disparity of different components in today's Internet: while edge networks have evolved to connect more end devices at faster network speeds, network routers, by and large, have not advanced for sufficient DDoS mitigation. This disparity leads to an imminent threat: the Catch-22 attack, and we use this attack to prove that existing DDoS defense options do not account for this disparity.

2 THE CATCH-22 ATTACK

In this paper, we introduce the Catch-22 attack that fundamentally holds benign hosts hostages that are located within the same subnet as the DDoS bots, and rotates attacked networks to increase its coverage across the Internet. An attacker can obtain machines from virtual private server (VPS) providers or residential proxy services to build a botnet that is co-located in networks with legitimate sources.

Due to limited TCAM space on routers, the Catch-22 attack forces the attacked networks to choose from: (a) deploying coarsely granular rules (e.g., cover /16 source IPv4 prefixes) that incur large amounts of collateral damage, but filtering most attack traffic, or (b) deploying finely granular rules (e.g., cover /32 source IPv4 prefixes) to reduce collateral damage, but risk not mitigating the attack. The Catch-22 attack introduces a *mitigation conundrum* as follows: an attacked network only wishes to filter the bots rather than the hostages, but the Catch-22 attack forces the network to filter hostage traffic due to the rule space limitation.

The Catch-22 attack consists of two critical steps: 1) *finding hostages*: acquire bots that are co-located with the targeted hostages in a small subnet, and 2) *moving target attack*: attack multiple networks to increase the scale of the attack. Figure 1 illustrates the Catch-22 attack when launched from a set of VPS providers, A and B, each of which own two separate /24 IP prefixes. The attacker first acquires a set of bots from the two VPS providers, which are co-located with several legitimate websites, labeled as hostage websites. Then the attacker launches a DDoS attack so that it floods a link upstream to the attacked network, preventing the attacked network from deploying effective filtering rules in its own network (e.g., at router Y). However, the attacked network may not be able to deploy rules on certain routers in upstream networks, either due to them being inaccessible or having no available rule space. The accessible routers may have limited rule space (e.g., router X),

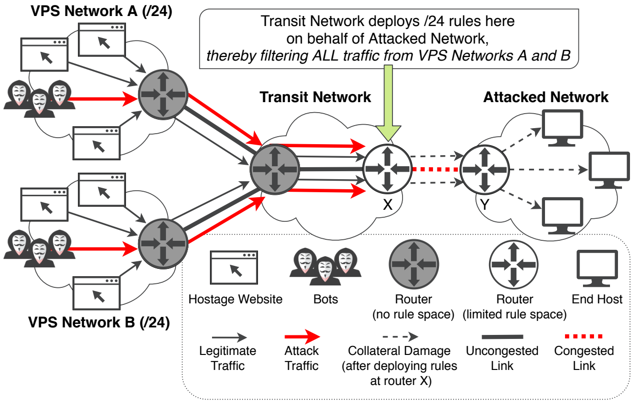

Figure 1: An overview of the Catch-22 attack

thereby forcing the attacked network to deploy coarsely granular source prefix rules (e.g., two /24 rules that filter all traffic from A and B) instead of finely granular individual IP rules (i.e., /32 rules that only filter the individual bots). Unfortunately, along with bot traffic, the rules will filter traffic from the hostage websites, potentially causing significant collateral damage to the attacked network.

2.1 Finding Hostages

Today, VPS providers are the go-to choices for anyone wanting to serve content to the public; anyone can request (virtual) machines to host content. According to *serverhunter.com*, there are over 1,000+ VPS providers on the market. One can rent a machine for as little as than $5 a month, or even pay by minutes they use the machines. At this rate, it would only cost an attacker $70 to launch an hour-long Catch-22 attack with 10,000 attack machines. Note, when attacking networks that mainly consume content (e.g., residential networks), attackers can inflict more damage by taking hostages in VPS networks, or server hostages, that serve content to the attacked networks.

Proxy service providers, such as *Luminati.io*, give users access to millions of residential IPs from distributed geolocations. The Catch-22 attack can thus take full advantage of these proxy service providers, and launch the attack from millions of residential IPs. Note, when attacking networks that mainly serve content (e.g., VPS or cloud networks), attackers can inflict more damage by taking hostages in residential networks, or client hostages, that consume content from the attacked networks.

2.2 Moving Target Attack

Pulsing-based attacks [3] allow the attacker to attack more networks simultaneously with limited resources. These attacks open the door to a technique we call the *moving target attack*. With the moving target attack, the attacker can scale up the Catch-22 attack from three aspects: 1) more attacked networks will block hostages in the attack, thereby translating to more collective damage across the Internet, 2) attacked networks will have less available rule space if they are connected to the same upstream network, and 3) DPS providers can become overbooked if too many attacked networks seek protections from DPS providers.

3 PRELIMINARY RESULTS

As an initial validation, we conduct two experiments on two VPS providers, AWS and DigitalOcean, to answer two obvious questions on server hostages: 1) do VPS providers block potential outbound DDoS flows? and 2) what server hostages can we possibly obtain? To answer the first question, we ran *HTTP* requests for an hour between two virtual machines (VM), each located in a different VPS network. Each of the *HTTP* requests contained a payload size of several megabytes. We then rate limited the VM bandwidth to *20 Mbits/s*, so as to avoid occupying too much shared bandwidth of the neighboring machines. As a result, we saw no throughput degradation within a one-hour experiment period. Secondly, we investigated what hostages an attacker could find from a VPS. We requested 6 VMs from AWS for this experiment, then use each VM's /16 network prefix to mimic the effect of a large-scale Catch-22 attack, where an attacker obtains a large number of bots from VPS providers, and force an attacked network to deploy rules matching large network prefixes to block the attack traffic. We used DNS dataset from OpenINTEL [5] that maps the Alexa top 1 million website domains to their IPs. As a result, we found over 1,000 websites that are potential hostages using the 6 /16 network prefixes from AWS alone.

4 CONCLUSION

In this paper, we presented the Catch-22 attack, an imminent threat to today's Internet that can force attacked networks to incur collateral damage during DDoS defense. The Catch-22 attack is made possible due to the following limitations in real world defense: 1) DPSes have finite filtering capacity, and 2) inline mitigation depends on the scarce TCAM space available in today's networking devices. Furthermore, we leveraged resources from VPS and residential proxy providers to quantify the damage caused by the Catch-22 attack. The core contribution of this work and its future extension is to examine fundamental vulnerabilities in today's DDoS defense infrastructure; the Catch-22 attack is a lens that focuses in on the vulnerabilities in today's Internet. Essentially, by presenting the Catch-22 attack, we are proving the Internet's vulnerabilities in a structured way. We hope this work will motivate the network community to implement defense solutions that do not ignore these vulnerabilities.

REFERENCES

[1] Banit Agrawal and Timothy Sherwood. 2008. Ternary CAM power and delay model: Extensions and uses. *IEEE transactions on very large scale integration (VLSI) systems* 16 (2008).

[2] Cisco. 2019. Effective DDoS Mitigation in Distributed Peering Environments. https://www.cisco.com/c/m/en$_u$s/network-intelligence/service-provider/digital-transformation/distributed-peering-architecture.html.

[3] Yu-Ming Ke, Chih-Wei Chen, Hsu-Chun Hsiao, Adrian Perrig, and Vyas Sekar. 2016. CICADAS: Congesting the Internet with Coordinated and Decentralized Pulsating Attacks. In *Proceedings of the 11th ACM on Asia Conference on Computer and Communications Security*.

[4] Pedro Marques, Robert Raszuk, Danny McPherson, Jared Mauch, Barry Greene, and Nischal Sheth. 2009. Dissemination of Flow Specification Rules. https://tools.ietf.org/html/rfc5575.

[5] OpenINTEL. 2019. OpenINTEL: Active DNS Measurement Project. https://openintel.nl.

[6] Sivaramakrishnan Ramanathan, Jelena Mirkovic, Minlan Yu, and Ying Zhang. 2018. SENSS Against Volumetric DDoS Attacks. In *Proceedings of the 34th Annual Computer Security Applications Conference*.