Losing Control of the Internet: Using the Data Plane to Attack the Control Plane

Max Schuchard, Abedelaziz Mohaisen, Denis Foo Kune, Nicholas Hopper, Yongdae Kim
University of Minnesota
Minneapolis, MN
{schuch, mohaisen, foo, hopper, kyd} @ cs.umn.edu

Eugene Y. Vasserman
Kansas State University
Manhattan, KS
eyv@ksu.edu

Categories and Subject Descriptors: C.2.0 [COMPUTER COMMUNICATION NETWORKS]: Security and protection

General Terms: Security

Keywords: DDoS, BGP, botnet, Internet

ABSTRACT

In this work, we introduce the Coordinated Cross Plane Session Termination, or CXPST, attack, a distributed denial of service attack that attacks the control plane of the Internet. CXPST extends previous work that demonstrates a vulnerability in routers that allows an adversary to disconnect a pair of routers using only data plane traffic. By carefully choosing BGP sessions to terminate, CXPST generates a surge of BGP updates that are seen by nearly all core routers on the Internet. This surge of updates surpasses the computational capacity of affected routers, crippling their ability to make routing decisions.

1. INTRODUCTION

The Internet can be divided into two distinct parts; the *data plane*, which forwards packets to their destination, and the *control plane*, which determines the path to any given destination. The control plane is designed to route around connectivity outages, resulting in the Internet's robustness to localized failure. This durability comes with a cost however: "local" events can have nearly global impact on the control plane. An excess of such control plane events can disrupt even core Internet routers. This disruption can lead to network instability, resulting in a loss of connectivity and data. There are several historical examples of such incidents stemming from rare events, such as router mis-configuration, hardware failure, and as side-effects of a fast-propagating worm.

In this work, we introduce the Coordinated Cross Plane Session Termination, or CXPST, attack, a new form of distributed denial of service (DDoS) attack that attempts to exploit the global scope of BGP updates to induce control plane instability on the Internet as a whole. In order to artificially create control plane instability, CXPST applies Zhang et al.'s [6] work on disrupting BGP sessions between routers. Zhang et al. described how an unprivileged adversary in control of a botnet can exploit the fact that the control plane and data plane use the same physical medium; from here on we will refer to this as the ZMW attack. This fate-sharing allows an adversary to convince a BGP speaker that one of its BGP sessions has failed. CXPST computes centrality measures of the network topology and uses this information to intelligently select a collection of BGP sessions to disrupt using the ZMW attack. This results in waves of control plane instability which, because of the choice of links, are broadcast globally. By exerting influence over the location and times of failures, CXPST generates enough updates to overwhelm the computational capacity of routers, crippling the Internet's control plane.

CCS'10, October 4–8, 2010, Chicago, Illinois, USA.
ACM 978-1-4503-0244-9/10/10.

2. BACKGROUND

2.1 Inter-domain Routing and BGP

The Internet is composed of multiple networks called autonomous systems (ASes), which relay traffic to each other on behalf of their customers. ASes are diverse, with a wide range of sizes and numbers of connections to other ASes. Some ASes have very high degrees of connectivity; these ASes are considered *core* ASes. Other ASes have very low degrees of connectivity, sitting at the outskirts of the Internet; these are *fringe* ASes. Fringe ASes require the assistance of core ASes in order to route traffic. The core ASes which agree to forward traffic on behalf of customers are termed *transit* ASes. Routers must determine what series of ASes packets have to traverse to reach their destinations. To this end, routers exchange routing protocol messages advertising other ASes which are reachable through them. The Border Gateway Protocol (BGP) [1] is the *de facto* standard routing protocol spoken by inter-AS routers.

2.2 BGP Stability and Network Performance

BGP is essentially a path vector routing algorithm with support for custom policies. If the network changes, routes that no longer exist will need to be withdrawn, new routes found, and routing changes advertised to other parties. These other parties must do the same, withdrawing routes, determining new routes, and advertising changes. This behavior demonstrates a key fact: small local changes are often seen globally by BGP speakers.

Instability in the control plane arising from network changes has been shown to directly result in vast reductions in the performance of the data plane [3, 5]. For example, when a router fails, paths that pass through it will no longer function, and new routes need to be found. Functioning routers will continue forwarding traffic towards the now non-existent router until they complete the process of finding a new route. All traffic directed toward the failed router will be dropped. Data plane functionality is only restored after the affected routers complete the processing of BGP messages. In the case of large amounts of instability, the load on a router's CPU is increased dramatically, possibly exceeding the capacity of already taxed route processors. This increased load translates into a longer

turnaround time for processing decisions, which in turn extends the duration of the data plane disruption.

2.3 Attacks on BGP Routers

Given the importance of routers and routing protocols, it is unsurprising that there exists a large body of literature exploring their weaknesses. Of particular interest to this work is a paper by Zhang, Mao, and Wang [6] that looks at using brief targeted data plane congestion to trick a pair of routers into disconnecting from each other. In their attack, an unprivileged adversary indirectly interacts with the control plane via the data plane. This is possible because the data plane and the control plane are co-located. Because of this co-location, congestion from data plane traffic can cause the loss of control plane traffic. When resources are scarce, control traffic and data traffic must share these limited resources. If enough consecutive control plane packets are lost, the halt timer of a BGP session will expire and the session will fail. When the BGP session fails, all routes discovered via that session will have to be withdrawn and new routes recalculated on both sides of the "failed" link. Zhang et al. demonstrated in both hardware and software routers the ability to successfully implement this attack.

3. CXPST

In order to create control plane instability, our attacker will apply the ZMW attack [6]. As discussed in Section 2.3, ZMW uses data traffic to trick a pair of routers into disconnecting from each other. This results in a set of route withdrawals, recalculations, and advertisements. Interestingly, the control plane disruption generated is not limited to the one set of withdrawals and advertisements. Since the targeted link is no longer used by routes after the BGP session fails, no traffic will utilize the link. This allows the two attacked routers to communicate with each other once more, as the link will no longer be congested with attack traffic. The targeted routers will, after a small amount of time, re-establish their BGP session. This will result in further BGP updates as the routes that were just withdrawn are re-advertised. Bot traffic will once again shift to the targeted link as the previous routes become utilized once more, and the attack resumes without any intervention from the attacker. The targeted BGP session will again be destroyed and the cycle repeats itself. In essence, CXPST induces targeted route flapping.

While the two routers attacked will be most impacted, routers not directly attacked will be affected as well. As mentioned in Section 2.2, BGP updates that result from local changes tend to be broadcast on a global scale. By creating a series of localized failures that have near global impact, CXPST overwhelms the computational capacity of a large set of routers on the Internet.

3.1 Selecting Targets

Maximizing control plane disruption is equivalent to maximizing the number of BGP update messages that are generated as a result of link failures. Centrality measures from graph theory provide a good starting point for building a heuristic to govern target selection. Our method of selection uses a slightly modified version of edge betweenness as a metric. Normally edge betweenness is defined as: $C_B(e) = \sum_{s \neq t \in V} \frac{\sigma_{st}(e)}{\sigma_{st}}$ where σ_{st} is the number of shortest paths between nodes s and t, and $\sigma_{st}(e)$ is the number of those paths that contain the edge e. BGP does not always use the shortest path between two ASes however. Because of this we use a modified definition of edge betweenness: $C_B(e) = \sum_{s \neq t \in V} path_{st}(e)$ where $path_{st}(e)$ is the number of BGP paths between IP blocks in s and t that use link e. Since each of these routes must be individually withdrawn, recomputed, and re-advertised this will provide an approximation of the number of BGP messages generated if the link were to fail. Consequently, target links are ranked in order of their "BGP Betweenness".

Another reason to use BGP betweenness is that our attacker possesses the resources to measure it. Our attacker controls a botnet distributed across the Internet, this provides him with a large number of distinct vantage points. Bots can perform traceroutes from themselves to a large set of nodes in separate networks and report the results. By aggregating the results an attacker can generate a rough measure of the BGP betweenness of links. Each time we see an edge in our aggregated traceroute data set, it represents an individual route that crosses a given link.

3.2 Dealing With Changing Topology

CXPST actively changes network topology. The attacker must select which bots will attempt to attack a given link with this in mind. Instead of simply checking that a given path contains the target link, the attacker must ensure that the path does not contain other links that are being targeted as well. By doing this, when links targeted by CXPST fail, attack traffic will not be re-routed.

Attack traffic can still be re-routed because of the unintended disruption of a non-targeted link. In order to counter this, an attacker should send more attack traffic toward a targeted link then is needed to congest it. This "safety net" will allow some amount of attack traffic to be diverted because of network dynamics without relaxing pressure on targeted links.

3.3 Fixing the Flow Issue

Our attacker will typically have more bots able to attack a given link than needed. Care must be taken when selecting a subset of these bots to attack the link. In order to minimize the amount of congestion prior to reaching the targeted link, the attacker should keep the attack traffic dispersed until it reaches the target. When the attack traffic reaches the targeted link the attack flows will be aggregated together, causing congestion on that link. After the intersection point traffic takes different paths toward its final destinations, dispersing in an effort to not congest downstream links.

CXPST uses a straight-forward algorithm to automate attacker assignment. Prior to allocating resources, our attacker builds two flow networks based on the traceroutes used to select targets. In one network, bots are treated as sources and target links are treated as sinks. In the other, target links are treated as sources and destination networks are treated as sinks. The attacker can either guess the bandwidth of links involved or actively measure their capacity. When selecting destinations for attack traffic, the attacker runs a max flow algorithm on the first flow network, establishing which bots will be used to attack each targeted link. Then the second flow network is then analyzed to determine which destination networks attackers should address their traffic to. Where possible bots will attempt to send attack traffic to IP address of other bots in the botnet as described by Sunder and Perrig in Coremelt [4]. In this way, traffic sent by the attacker is "wanted" and not reported.

3.4 Thwarting Defenses

There are some mechanisms that exist to reduce the effects of route flapping. Since CXPST is artificially induced route flapping, these defenses might impede it. These defenses though, were designed to deal with random network events, not an adaptive adversary. Two of the defenses, BGP Graceful Restart and Minimum Route Advertisement Intervals, require no changes. Route Damping on the other hand requires some minimal changes to CXPST's behavior. During the course of the attack the bots will need to remove links that get damped from their target set. Bots notice that links are being damped when the paths used to reach their targets

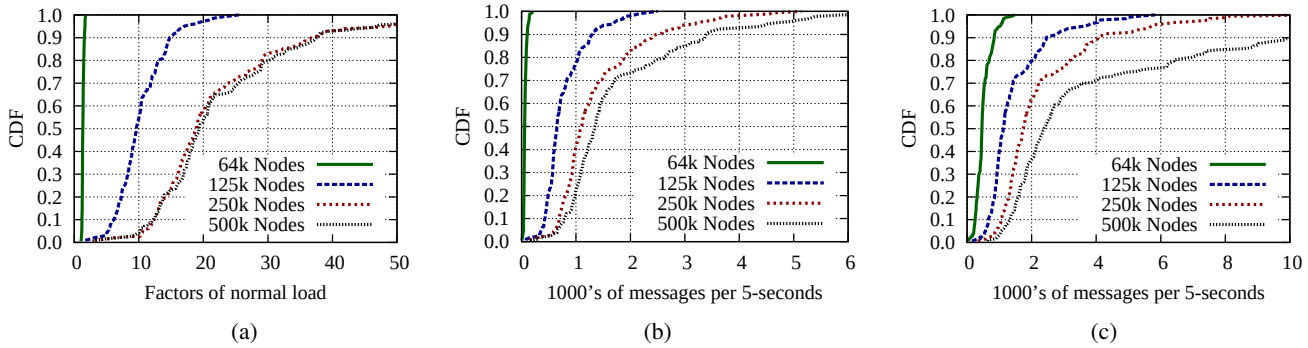

Figure 1: Median router load of targeted routers under attack as a factor of normal load (a). Message loads experienced by routers under attack, measured in BGP updates seen in 5-second windows: 75th percentile (b), and 90th percentile (c).

do not re-appear within a time window. New target links are then chosen from the list of available targets.

4. SIMULATION

In order to answer these questions we built a discrete event driven simulator modeling the dynamics of routers on the Internet. Given the level of complexity found in the system that we were attempting to model, this presented a challenge. Many diverse agents needed to be represented including: ASes, routing polices, the routers themselves, the physical links that connect these routers, and the botnet used by our attacker.

4.1 Simulation Methodology

Using inferred AS relationships from CAIDA, we chose ASes servicing other providers, i.e. all ASes who had at least one customer that itself had customers, and generated a graph modeling the interconnection of these ASes. The result was a connected graph with 1,829 ASes and nearly 13,000 edges. Since we are more concerned with the dynamics of traffic passing between ASes than traffic moving inside an AS, we modeled an AS as a single BGP speaker with a link connecting it to each AS its host AS has a relationships with. This simplification is acceptable for experimenting with CXPST as we focus on the behavior of traffic at the network edges, and are largely unconcerned with internal dynamics.

The bandwidth model for links in our simulator is meant to be as disadvantageous to the attacker as possible. Link capacities are based on the degrees of the connected ASes. Since we are concerned about the ability to fill core AS links we use OC-768 size links, the largest link size currently in the SONET standard, for those links. In the same spirit we connect all fringe ASes, where the majority of the attacker's resources reside, with OC-3 links. It is important to mention that while the aggregate bandwidth between two ASes may be much higher than a single OC-768 link, we are only concerned with attacking *single inter-AS links*, meaning that having to attack an OC-768 link is truly a worst case scenario.

We used the data set for the Waledac botnet [2] to build our model of bot distributions. IP addresses of infected machines were mapped to their parent ASes using the GeoIP database, providing a rough count of infections per AS. We then uniformly scaled these numbers up or down to achieve the botnet size desired. To ensure a proper lower bound for attacker bandwidth, bots were given a basic ADSL connections with an upload capacity capped at 1.0 Mbit/sec. Bots were only given the ability to send network traffic and perform traceroutes. They were *not* given any additional information about the network, such as link capacities or AS relationships.

4.2 Simulation Results

We ran our attack with botnets of 64, 125, 250, and 500 thousand nodes. Targets were selected from the *core routers* in our topology, the top 10% of ASes by degree. There are two reasons behind this selection strategy: the sizes of these ASes would increase the magnitude of control plane instability, and their expansive customer base would increase the impact of the resulting data plane failures.

As mentioned in Section 2.2, large bursts of updates have a significant impact on the performance of the Internet. Simulations show that CXPST successfully creates BGP update message bursts throughout the duration of the attack. For example, during normal operation the 90th percentile load is 182 messages per 5 seconds. During CXPST, the 90th percentile load is dramatically increased for the targeted routers, as seen in a CDF of their 90th percentile loads in Figure 1(c). In the case of the $250,000$-node attacker, more than half of core routers are at or above an order of magnitude increase in load. These bursts of updates are not a few isolated incidents. At the 75th percentile of update load, shown Figure 1(b), we continue to see the same dramatic increases in processing load.

Moreover, these spikes are not the only effect of CXPST, an increase in BGP update rate is felt throughout the attack. Figure 1(a) shows the increase in the median load of routers during the attack. In the case of the $250,000$-node botnet, the median load on nearly half of the core routers increased by a factor of 20 or more. Even using the $125,000$-node botnet results in 50% of routers' median loads increased by an order of magnitude or more. This increased median load shows that routers will not have a chance to recover from the previous bursts of updates.

Acknowledgments This work was supported by NSF grant 0917154. We thank Adrian Perrig and Shubho Sen for helpful discussions about this work.

5. REFERENCES

[1] N. W. Group. RFC4271 - A Border Gateway Protocol 4 (BGP-4). `http://tools.ietf.org/html/rfc4271`, January 2006.

[2] G. Sinclair, C. Nunnery, and B. B. Kang. The Waledac protocol: The how and why. In *In proceeding the IEEE International Conference on Malicious and Unwanted Software (MALWARE)*, pages 69–77, October 2009.

[3] K. Sriram, D. Montgomery, O. Borchert, O. Kim, and D. R. Kuhn. Study of BGP peering session attacks and their impacts on routing performance. *IEEE Journal on Selected Areas in Communications*, 24(10):1901–1915, 2006.

[4] A. Studer and A. Perrig. The Coremelt attack. In *Proceedings of the European Symposium on Research in Computer Security (ESORICS)*, Sept. 2009.

[5] F. Wang, Z. M. Mao, J. Wang, L. Gao, and R. Bush. A measurement study on the impact of routing events on end-to-end Internet path performance. *SIGCOMM Comput. Commun. Rev.*, 36(4):375–386, 2006.

[6] Y. Zhang, Z. M. Mao, and J. Wang. Low-rate TCP-targeted DoS attack disrupts Internet routing. In *Proceedings of the Network and Distributed System Security Symposium (NDSS)*, 2007.