# Investigating End-To-End Integrity Violations in Internet Traffic

Alan Mislove David Choffnes Taejoong Chung
Northeastern University

**Abstract**

Internet applications are commonly implemented with the implicit assumption that network traffic is transported across the Internet without modification and without having application-level data being monitored; we refer to this end-to-end integrity. Put simply, most applications assume that the data they send will be received intact by the host they are communicating with (barring transient errors and normal packet loss). This expectation is encoded in the Federal Communications Commission (FCC) Open Internet Order, which states that Internet Service Providers (ISPs) should not impose "unreasonable interference" with customers' network traffic. However, it is increasingly common for ISPs to deploy middleboxes that silently manipulate customers' traffic in ways that impact security, privacy, and integrity.

This paper describes a methodology to test for such end-to-end modifications, as well as evidence of multiple ISPs that modify customers' traffic in-flight. We use a HTTP/S proxy service with millions of end hosts in residential networks to study the behavior of over 14K networks worldwide. Using this system, we route benign traffic via over 1.2M hosts in these networks to test for end-to-end integrity. We find end-to-end integrity violations including hijacking of certain DNS responses—often sending users to pages with advertisements—by AT&T, Verizon, and Cox Communications (as well as a number of non-US ISPs). We also find content injection in web pages—often adding trackers or advertisements to web pages or censoring content—by a number of non-US ISPs as well. Worse, we find evidence that a number of hosts' web requests are being monitored, suggesting that subscriber browsing data is being shared with third parties.

Given the increasing amounts of critical and privacy-sensitive information that is exchanged online, we recommend that regulators leverage active auditing technologies to inform and enforce current and future policies. Our methodology can be deployed with low overhead and is scalable to millions of hosts and thousands of networks.

## 1 Introduction

Internet applications today typically operate with the implicit assumption that network traffic is delivered to the destination without any modifications, which is referred as *end-to-end integrity*. For example, when a user's browser is communicating with a website (e.g., browsing a news site or using online banking), the browser typically assumes that its request for a web page is received by the website without modification, and the data the website sends back is delivered to it unmodified as well. Moreover, users typically assume that the content transmitted by their browser is not monitored by the network providers through which the data passes.

In terms of policy, there are two rules that may speak to the legal basis for end-to-end integrity. First, the Federal Communications Commission's (FCC's) Open Internet Order [8], adopted in 2015, states that Internet Service Providers (ISPs) should not *unreasonably interfere with* customers' network traffic, thereby prohibiting ISPs from throttling, prioritizing, or blocking network traffic outside of what is necessary for reasonable network management. However, the FCC is currently in the process of reviewing this order, and the order may be further revised in the near future. Second, the FCC also recently adopted new privacy rules [16], which limit how providers can use and share customer data (e.g., users' browsing history) derived from network traffic. However, President Trump signed into law Senate Joint Resolution 34 [11], which nullified this FCC rule. Despite the removal of the new FCC privacy rules, ISPs are still classified as common carriers, which has some implications for what they can collect and share.

Recent regulatory proposals and laws have suffered from a lack of empirical evidence of ISP practices to (a) inform potential harms to subscribers and content/service providers, and (b) guide policymakers in the design of rules that mitigate them. As a result, it is increasingly important to understand how users' traffic could be subject to end-to-end integrity violations today, e.g., by being modified or monitored. Unfortunately, such end-to-end violations are often quite difficult to detect: ISPs typically do not announce the presence or function of the devices that can perform monitoring or modification, nor do they usually declare how they modify and monitor customer's traffic. Moreover, running active tests to observe modification or monitoring requires the ability to send traffic to and from many different network providers.

In this paper, we describe our recent work that aims to understand and protect consumer's security and privacy, focusing on two questions: *what kinds of customer's traffic are modified?* and *who modifies the customer's traffic?* To answer the above questions, we leverage the commercial Peer-to-Peer (P2P)-based HTTP/S proxy service, *Luminati*, which is based on *Hola Unblocker* browser plugin. By using Luminati, we measure and analyze end-to-end integrity violations from over 1.2M hosts across 14K Autonomous Systems[1] (ASes) in 172 countries. In particular, we find multiple ISPs that modify customer's traffic in-flight, and others that monitor customer's traffic.

In the remainder of this paper, we first provide background on end-to-end integrity and other related approaches. We then describe our methodology of using the Luminati network. Next, we present our findings about two different types of content modification (DNS NXDOMAIN hijacking and HTTP content modification), as well as content monitoring. We concluding with a discussion of the policy implications of our findings.

It is important to note that we make all of our analysis code and data public to the research community at

`https://tft.ccs.neu.edu`

allowing other researchers to use a similar approach to detect end-to-end connectivity violations in DNS and HTTP modification and content monitoring. Additionally, this paper is based on work we published at the IMC conference in 2016 [1]; we refer the reader to this paper for technical details of many of our experiments.

## 2 Background and Related work

In this section, we briefly outline existing attempts to measure end-to-end violation in Internet and examples of end-to-end integrity violations.

### 2.1 End-to-end integrity violations

We define an end-to-end integrity violation as any time the application-level data sent between two hosts on the internet is modified or monitored in transit. To make this more clear, we briefly describe two common end-to-end integrity violations that we explore in the latter part of the paper.

**DNS response modification** The Domain Name System (DNS) is the Internet's equivalent of the "yellow pages." DNS provides a mapping, translating (human readable) domain names such as `example.com` to (machine friendly) Internet Protocol (IP) addresses such as `10.0.0.1`. DNS works based on a query–response protocol, where each computer is configured with a local DNS *resolver* to whom it sends requests.

If the DNS response modified on-the-fly and replaced with different IP address, a user would end up contacting a different machine and, as a result, could see a different web page. This process is called *DNS hijacking*. Recently, Kuhrner et al. [7] measured open (public) DNS resolvers found that many resolvers deliberately manipulated DNS responses and returned incorrect IP address information to restrict the access to certain web pages for censorship, inject advertisements, serve malware, or perform phishing. Similarly, Dagon et al. [2] found in 2008 that 2.4% of DNS queries to open DNS resolvers are returned with incorrect answers. This behavior has been also observed at a variety of ISPs like AT&T [3] and Verizon [14].

[1] Each AS corresponds to one ISP, and an ISP may have multiple ASes.

One particular form of DNS hijacking is *NXDOMAIN hijacking*. NXDOMAIN is a DNS response that indicates that a domain name does not exist (i.e., it is not registered). For example, if a user types a domain name that does not exist into their browser, the browser would receive a NXDOMAIN response and would display a message like "The destination could not be found." However, if the NXDOMAIN response is instead replaced with a hijacked response, the browser—and, therefore, the user—would never know that the domain does not actually exist. Previous studies [2, 15] have found that NXDOMAIN responses are also often hijacked by ISPs to "assist" users by sending them to a "search help" page (or one simply filled with advertisements) instead of allowing the browser to show a connection error to the user.

**HTTP content modification** Another important type of end-to-end integrity violation is HTTP content modification, which happens when a third-party modifies HTTP (web) content between servers and clients. Due to the fact that HTTP by itself does not have integrity checks, violations of the end-to-end integrity of HTTP traffic have been occurring for many years. The most prominent example is an ad injection, where a third-party modifies the content of a web page to insert an advertisement; this is typically done without the web site's knowledge or consent, and the website typically does not receive any compensation for the ad placement. Thomas et al. [12] found that more than 5% of unique daily IP addresses accessing Google are exposed to injected advertisements, often due to malicious Chrome extensions or Window binaries. Also, Zhang et al. [18] identified nine ISPs (including AT&T, Level3, and Suddenlink) that redirect users to web servers hosting modified content.

Content modification has also been observed in mobile networks as well; for example, Xu et al. [17] revealed that Sprint, a major US cellular, performs image compression for mobile users. Lowering the quality of images may benefit users by reducing their bandwidth usage, but also may inadvertently hurt their quality of experience.

## 2.2 Measurement Platforms

Properly detecting violations of end-to-end integrity at scale is often challenging, as it requires an approach that can achieve broad coverage (i.e., many network vantage points). In this subsection, we introduce two previously-proposed approaches to detect end-to-end integrity violations.

**Dedicated hardware and software** The first class of approaches is distributing dedicated hardware or software to users. For example, RIPE Atlas [10] is a project that uses small devices that plug into users' networks (currently 9,300 have been deployed) to conduct network measurement experiments. Each device can run experimenter-configured tests, each of which can use multiple protocols (e.g., ICMP and DNS) to collect information about Internet connectivity and reachability. RIPE collects and publishes the results of these tests from thousands of its measurement devices deployed around the world.

BISmark (Broadband Internet Service Benchmark) [9] is a home router equipped with custom software, along with backend infrastructure to manage experiments and collect measurements. BISmark routers collect information on access link performance, network connectivity, Web page load times, and user behavior and activity. As the BISmark router is directly attached to the network, it enables researchers to measure the network performance continuously, directly, and comprehensively. More than 400 active BISmark routers have been deployed to over 34 countries.

The Netalyzr project [6] is a Web page with an embedded Java applet; its purpose is to diagnose users' network problems. Users can simply access it via their Web browser, and a Java applet performs multiple tests on their machines to check for end-to-end integrity violations such as DNS responses manipulation or HTTP content modification. This approach is relatively simple—users do not need to deploy dedicated hardware or install software—but it is hard to measure the same users over time as they need to visit the Web page again. Over the past six years, over 1.2M users have run the Netalyzr software.

All of these projects required substantial effort to recruit users: In the cases of RIPE Atlas and BISmark, hardware needed to be purchased, configured, distributed, and installed. In the case of Netalyzr, users needed to be recruited. Later in the paper, we present an approach that is as scalable as these systems, but requires no dedicated hardware or recruitment of users.

**Using popular websites** Popular websites also have the option of embedding JavaScript or Flash into their web pages that run measurement code. For example, Google detected content integrity violations by embedding a script in each served Web page that compared the origin content with the actual content that users received. Google found that more than 5% of unique daily IP addresses accessing Google are impacted by ad injection, and also observed that 192 deceptive Chrome extensions that inject advertisements; these together affect up to 14 million users [12]. Similarly, Facebook [4] embedded a script that detects the occurrence of SSL/TLS man-in-the-middle attacks, where the attacker potentially relays and alters the secure communication between an user and Facebook. Facebook analyzed over 3 million real-world SSL connections to Facebook web site, and found that 0.2% of them were tampered with modified SSL connections. This approach can quickly measure the end-to-end integrity violation from a large number of diverse users due to the popularity of these sites. However, it requires privileged access to a popular Web site, something that most researchers and policy makers lack.

# 3 Methodology and Dataset

We first introduce our methodology to detect end-to-end integrity violations in the Internet, and the resulting dataset, before proceeding with our analysis.

**Hola Unblocker** The Hola Unblocker (`http://hola.org/`) is a platform operated by Hola Networks that allows users to route their traffic via a large number of servers (i.e., proxies) over the globe. It is typically used to bypass geo-restrictions to "unblock" any websites regardless of the user's location. Users can use this service by installing software, which is provided in various different forms, including a Windows application, a Firefox add-on, a Chrome extension, and even an Android application. Hola Networks [5] claims that more than 123 million people across the globe have installed the software. When users install the Hola Unblocker, they are required to choose one of the following options to use the service: (1) Users can choose to pay $5 per month (or $45 per year) for a subscription, or (2) users can choose to allow other Hola users to route traffic via their machine, and then can use the Hola Unblocker with free of charge. If users choose the second option, users of *Luminati* (described below) can route their traffic through their machines.

**Luminati** The *Luminati* is the *paid* HTTP/S proxy service that routes traffic via Hola Unblocker users (we refer to these as *nodes*) who choose to use Hola Unblocker for free. Hola Networks claims that more than 23 million IP addresses are accessible through Luminati service. It also provides API (Application Program Interface) that enables users to automate their requests, as well as choose which Hola client (which we refer to as a *node*) will be selected to route their traffic. More specifically, it allows users to (1) select the country/AS that the node is located in, (2) choose the same node over time for subsequent requests, and (3) request DNS resolution be done by the node (using the node's DNS server).

**Dataset** Using Luminati, we obtained our datasets between April 13 and April 18, 2016 for the DNS and content monitoring studies and between May 4 and May 8, 2016 for the HTTP study. In total, we routed traffic via 650K hosts in more than 165 countries.

# 4 Content Modification

We begin our analysis by focusing on two forms of content modification: (1) DNS `NXDOMAIN` response modification, where DNS responses are hijacked by a third party and replaced with another DNS response, and (2) HTTP content modification, where web content is modified before it reaches the user's browser. In both cases, users browsing the web could unknowingly be presented with content that they did not intend to browse. For both kinds of content modification, we first introduce our methodology, then describe our dataset, and close by analyzing the causes and prevalence of these two forms of content modification.

## 4.1 DNS NXDOMAIN hijacking

We begin by considering DNS `NXDOMAIN` hijacking.

| Rank | Country | Tested nodes<br>Hijacked | <br>Total | Ratio |
|---|---|---|---|---|
| 1 | Malaysia | 3,652 | 6,983 | 52.3% |
| 2 | Indonesia | 3,178 | 8,568 | 37.1% |
| 3 | China | 237 | 671 | 35.3% |
| 4 | UK | 9,553 | 37,156 | 25.7% |
| 5 | Germany | 4,703 | 19,076 | 24.7% |
| 6 | US | 6,108 | 33,398 | 18.3% |
| 7 | India | 1,127 | 6,868 | 16.4% |
| 8 | Brazil | 3,190 | 24,298 | 16.4% |
| 9 | Benin | 90 | 716 | 12.6% |
| 10 | Jordan | 76 | 1,117 | 7.7% |

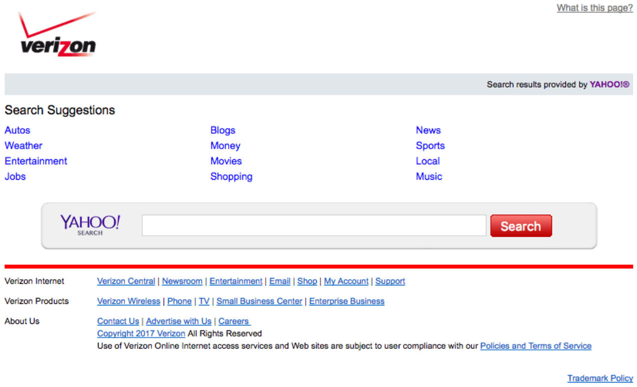
**Table 1:** Table showing the top 10 countries sorted by the fraction of nodes experiencing NXDOMAIN hijacking.

**Methodology and datasets** To test for NXDOMAIN hijacking, we configure a DNS resolver to host a domain name we control, and configure the resolver to return a NXDOMAIN response for all queries. We then request that Luminati nodes look up our domain and check whether they receive a NXDOMAIN DNS response (i.e., determine whether the NXDOMAIN response is hijacked or not).

Using this methodology, we measured a total of 753,111 unique nodes from 167 countries and 10,197 ASes. We find that these nodes are configured to use a total of 33,446 unique DNS resolvers. Among these nodes, we observe that responses served to 35,800 nodes (4.8%) are intercepted and modified.

**Overall results** We first obtain a macroscopic view of NXDOMAIN hijacking phenomena by grouping nodes according to country, and focus on the groups where we have at least 100 nodes. Doing so allows us to understand the prevalence of hijacking, and in some cases determine what is responsible for the hijacking. Table 1 shows the top 10 countries sorted by the fraction of nodes with hijacked DNS responses. For example, we found that in Malaysia, more than 52% of nodes we measured experienced hijacking. Surprisingly, in the US, we found that about 18.3% of the nodes have their NXDOMAIN responses hijacked. We now explore who is doing the hijacking, and where they are sending the users.

**Hijacking culprits** When an NXDOMAIN DNS response is hijacked and replaced with a different response, it could be done at the DNS resolver, a middlebox, or end-host software. In this subsection, we investigate and determine which culprits hijack NXDOMAIN DNS responses. In the case of hijacking taking place at the DNS resolver, we examine nodes that use ISP-provided DNS resolvers separately from nodes that use other (public) DNS resolvers (e.g., Google's open DNS service).


What is this page?
verizon
Search results provided by YAHOO!®
Search Suggestions
Autos
Weather
Entertainment
Jobs
Blogs
Money
Movies
Shopping
News
Sports
Local
Music
YAHOO! SEARCH
Search
Verizon Internet
Verizon Central | Newsroom | Entertainment | Email | Shop | My Account | Support
Verizon Products
Verizon Wireless | Phone | TV | Small Business Center | Enterprise Business
About Us
Contact Us | Advertise with Us | Careers
Trademark Policy

Copyright 2017 Verizon All Rights Reserved

Use of Verizon Online Internet access services and Web sites are subject to user compliance with our Policies and Terms of Service

**Figure 1:** Verizon's Search Assist web page. Verizon subscribers using certain Verizon-supplied DNS resolvers are redirected to this web page instead of receiving an NXDOMAIN DNS response.

*ISP DNS resolvers* Previous work [2, 15] found NXDOMAIN responses are sometimes hijacked by local ISPs to redirect users to "search help" pages. For example, as shown in Figure 1, Verizon users are redirected to a "Search Assist" web page that shows suggestions for other web destinations, as well as a search bar for Yahoo. As many Verizon hardware devices (routers, gateways, or DSL modems) are configured to use

| Country | ISP | DNS Resolvers | Nodes |
|---|---|---|---|
| US | Verizon | 98 | 2,102 |
| | Cox Communications | 63 | 1,789 |
| | AT&T | 37 | 561 |
| | Suddenlink | 9 | 98 |
| | Mediacom Cable | 6 | 219 |
| | Cable One | 4 | 108 |
| | WideOpenWest | 1 | 39 |
| Argentina | Telefonica de Argentina | 14 | 276 |
| Australia | Dodo Australia | 21 | 1,404 |
| Brazil | Oi Fixo | 21 | 2,558 |
| | CTBC | 4 | 290 |
| Germany | Deutsche Telekom AG | 8 | 1,385 |
| India | Airtel Broadband | 9 | 735 |
| | Ntl. Int. Backbone | 8 | 245 |
| | BSNL | 2 | 71 |
| Malaysia | TMnet | 8 | 1,676 |
| Spain | ONO | 2 | 71 |
| UK | Talk Talk | 46 | 3,738 |
| | BT Internet | 6 | 479 |

**Table 2:** Table showing ISP DNS resolvers that hijack responses for more than 90% of nodes. Also shown is the number of DNS resolvers and nodes per ISP.

Verizon DNS resolvers by default, Verizon users will be redirected to this page when they should otherwise receive a `NXDOMAIN` response.

To see how many ISPs have DNS resolvers that hijack `NXDOMAIN` DNS responses, we first group the nodes by the DNS resolver they use, and identify the ISP-provided DNS resolver as ones where all nodes and the DNS resolver belong to the *same* ISP. We then identify ISP DNS resolvers that hijacked the responses as the ones where more than 90% of their nodes experience hijacking; this represents 366 unique ISP-provided DNS resolvers (3.8% of all ISP resolvers) covering a total of 17,358 nodes.

We list the 19 ISP who operate these resolvers in Table 2. Surprisingly, we found seven ISPs in US hijack `NXDOMAIN` DNS responses; this complements previous findings [2,15] and confirms that `NXDOMAIN` hijacking is still happening on a large scale in the US. For example, we identified 98 different Verizon DNS resolvers that we observed to be performing `NXDOMAIN` hijacking.

*Public DNS resolvers* We first identify public/external DNS resolvers by focusing on the ones where we observe nodes coming from more than two countries, which leaves us 1,110 public DNS resolvers. Then we focus on the public/external DNS resolvers that hijack the `NXDOMAIN` DNS response when 90% of nodes using the resolver experienced hijacking; this represents 21 resolvers used by 1,512 nodes. We identify the owner of these resolvers by investigating the DNS resolver's IP address, and issuing DNS queries directly to each DNS resolver to see if it responds. We identified four public DNS services that perform `NXDOMAIN` hijacking: (1) nine resolvers operated by *Comodo DNS*[2], (2) four resolvers of *UltraDNS*, (3) two resolvers operated by *LookSafe*, a piece of malware that changes users' DNS settings,[3], and (4) three resolvers operated by *Level 3*. Thus, even users who opt-out from using their ISP-provided DNS resolver can end up experiencing `NXDOMAIN` hijacking anyway.

*ISP middleboxes and malware* The remaining potential culprits to hijack `NXDOMAIN` DNS are ISP middleboxes (i.e., devices that hijack the response in the middle of network), or software on the node itself. In general, it is challenging to disambiguate these cases as we have little visibility into the network path or the software on the node. To find the nodes that may experience DNS hijacking from ISP middleboxes or malware, we first focus on the nodes that use a DNS resolver where we *know* the resolver does not hijack responses: namely, one of Google's public DNS resolvers.

Specifically, we focus on the nodes that (1) use a Google DNS resolver, but (2) their `NXDOMAIN` DNS response is still hijacked, which leaves us 927 (0.12%) nodes. We then look at the content that nodes received to get clues to infer the source of the hijacking. Across these 927 nodes, we find many nodes that receive

[2] `http://www.comodo.com/secure-dns/`

[3] `http://www.spyware-techie.com/looksafe-removal-guide`

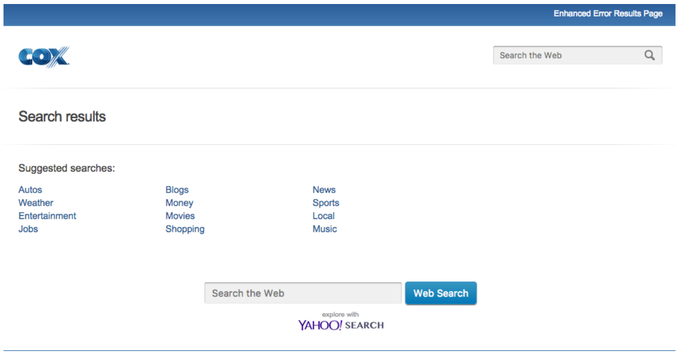


**Figure 2:** Cox's redirected page; it does not specify how to opt-out from `NXDOMAIN` hijacking.

similar content; manually investigating these, we find that the pages include 12 URLs containing ISP names and 2 URLs containing software names. For example, Figure 2 shows the page that is loaded by nodes in the Cox network, where we suspect that Cox hijacks `NXDOMAIN` DNS responses. It is interesting to note the similarity of this page to Verizon's Search Assist in the Figure 1; the similarity suggests the networks may be using hardware/software from the same vendor to implement hijacking.

## 4.2 HTTP Content Modification

Next, we turn to looking for modification of HTTP (web) content.

**Methodology and Datasets** As our focus is to see whether the original content is modified, we deploy a web server to our domain. Then, we simply fetch content from our web server via a node and check whether it is modified. For this experiment, we fetch four different pieces of content through each node: a HTML page, a JPEG image, a JavaScript library, and a CSS file. Using this methodology, we measured 49,545 nodes in 12,658 ASes across 171 countries. We detected HTML content modification for 472 nodes (0.95%), image modification for 694 (1.4%), JavaScript modification for 45 (0.09%), and CSS modification for 11 (0.002%).

**Results** There are different reasons why different types of content may be modified, so we examine each content type separately below.

*HTML* We find that 472 nodes (0.95%) received modified HTML pages. Among them, we are able to find only one AS, AS 42925 (*Internet Rimon ISP*), where *all* nodes received modified HTML content by inserting an `meta` tag which could be used as an identifier. For the remaining cases, we investigated the JavaScript code injected into HTML content by manually extracting URLs or keywords that characterize the code; through this analysis, we found they all appear to be spyware, likely injected by software running on the node itself.

*Images* We find that 694 nodes in 22 ASes receive modified images; over 87% of these nodes come from the top 12 ASes. Interestingly, we found that *all* 12 of these ASes are mobile ISPs, suggesting that the ISPs are doing on-the-fly image compression to reduce network bandwidth. To verify this behavior, we checked the size of the delivered image versus the original image; all of the ISPs except for two appeared to have fixed compression ratios. While this approach does reduce the bandwidth requirements on mobile networks, transparently compressing images without users' consent likely violates network neutrality principles as well as degrades the users' quality of experience.

*Javascript and CSS* We observe 45 nodes and 11 nodes received modified JavaScript and CSS content, respectively, all of which are error pages or empty responses; this suggests the requests were filtered at either the ISP or the host itself. We could not observe any modification to the original Javascript and CSS content in the other nodes.

| Monitoring entity | | Monitored users | | |
|---|---|---|---|---|
| Name | IPs | Nodes | ASes | Countries |
| TalkTalk | 6 | 2,233 | 5 | 1 |
| Tiscali U.K. | 2 | 363 | 6 | 1 |
| Trend Micro | 55 | 6,571 | 734 | 13 |
| Commtouch | 20 | 1,154 | 371 | 79 |
| AnchorFree | 223 | 461 | 225 | 98 |
| Bluecoat | 12 | 453 | 162 | 64 |

**Table 3:** Table showing the top six ASes where unexpected requests originated, indicating content monitoring. The shaded row represent ISP-level content monitoring and the others represent anti-virus or VPN software content monitoring.

# 5 Content Monitoring

DNS `NXDOMAIN` response modification and HTTP content modification are end-to-end integrity violations where content was modified. However, in the course of conducting our study, we found another end-to-end violation: *content monitoring*. This is where middleboxes or ISPs are silently monitoring HTTP requests that users are downloading.

Detecting content *modification* is relatively easy, as we can simply compare the original content and received content to look for differences. In contrast, content *monitoring* is significantly more difficult to detect as there is no change to the content itself. However, we were able to measure some cases of content monitoring by observing unexpected requests coming to our web server from sources other than the node that was supposed to make the request.

## 5.1 Methodology and datasets

Our methodology is relatively straightforward: we simply fetch a content from our web server only *once*, and monitor our web server to see if there are other requests coming from other IP addresses that are not the node. As each node makes a request for a unique page, we expect that each page would only be requested once. We monitor the web server for up to 24 hours after making the initial request. Using this methodology, we measured a total of 747,449 nodes, and 11,234 (1.5%) of them resulted in unexpected requests arriving at our web server.

## 5.2 Analysis

To identify the culprit behind the content monitoring, we first manually examine the IP address, AS, and HTTP headers of the unexpected requests. From this analysis, we found that 424 unique IP addresses generate unexpected requests, which we can group in 54 sets with common characteristics. Table 3 provides more details on the most popular groups out of these 54; all together, these six sources generated 94.0% of the unexpected requests. In the sections below, we examine these groups in more detail.

**ISP level monitoring** We first notice that TalkTalk (a U.K. ISP) and Tiscali U.K. (acquired by TalkTalk in 2009) generate unexpected requests for 2,333 and 363 nodes, respectively. All of these affected nodes are constrained to a small number of ASes (all in TalkTalk and Tiscali network, respectively), which strongly indicates that two ISPs monitor users' HTTP requests and visit the same pages using their own servers later. We note that for both ISPs, some but not all nodes in both ISPs experience content monitoring. Potential explanations are that content monitoring could be done non-deterministically (e.g., only $x$% of requests are monitored), or it may be due to ISP-provided additional services like parental content controls.[4] Regardless, monitoring and its potential for violating users' privacy has significant implications, and should be made transparent.

**Anti-virus and VPN Software** We found that TrendMicro (an anti-virus software vendor), Commtouch (another anti-virus software vendor), Anchorfree (a VPN software vendor), Bluecoat (a computer security company) generate unexpected requests where the affected nodes are not confined to a small number of

[4] For example, TalkTalk's opt-in SuperSafe feature (`https://help2.talktalk.co.uk/supersafe-boost-overview`).

ASes. For TrendMicro and Commtouch, we believe that this behavior is due to anti-virus software installed on a node. Such software providers diagnose websites and blacklists those that correspond to malicious sites to prevent users from harm.[5] Thus, the unexpected requests may be due to the anti-virus vendor checking such malicious sites. We also observed that Anchorfree, a VPN service provider for web browsing, monitors generates the unexpected traffic from 223 IP addresses that are all from Anchorfree's AS. In fact, we observe that the first request comes from one of 10 different locations around the globe, but the second request always comes from Menlo Park, California. This is likely due to their "malware protection" feature provided as part of the "Hotspot Shield" service. Bluecoat, a computer security company that sells a proxy service, also generates unexpected requests *prior to* the arrival of a request from a node. Thus it appears that Bluecoat first downloads the content before allowing the node's request to proceed.

### 5.3 Summary

In this section, we developed and deployed techniques that can detect certain instances of content monitoring. We found that over 1.5% of all nodes suffered from content monitoring where their requests are re-requested by a third party, which are mainly from anti-virus software, VPN services, or user's ISP. All of these behaviors have significant security, privacy, and performance implications for end users, given that in most cases they are likely unaware that their HTTP browsing history is monitored in real-time.

## 6 Conclusion and Policy Recommendations

In this paper, we introduced a new methodology for measuring end-to-end integrity violations in internet traffic. Based on the Luminati proxy network, this methodology allows for large-scale measurements of HTTP and DNS to be conducted without the need to recruit users beforehand; we were able to measure over 1.2M nodes in under 5 days. We applied this methodology to look for DNS `NXDOMAIN` hijacking, HTTP content modification, and HTTP content monitoring, and found a surprisingly large fraction of nodes are subject to one or multiple of these behaviors. Below, we discuss some of the policy implications of our study.

**Privacy and awareness** All three forms for end-to-end violations raise significant concerns over user consent, awareness, and even potential privacy issues. For example, with `NXDOMAIN` hijacking, most of these DNS server-provided "search assist" services appear to be implemented as opt-out services, evidenced by the large number of nodes that we observe using them; for example, in Figure 1, a user can only be informed how to opt-out by clicking "What is this page?" link in the upper-right corner. More worrisome, it is unclear how non-web applications will handle `NXDOMAIN` hijacking, as they may be built to expect `NXDOMAIN` responses that will never arrive. Similarly, it is unclear the extent to which users are aware of HTTP content injection and content monitoring. There is a need for regulation that provides clear guidelines on policies for transparency and consent, both to ensure users are aware of such practices and that they can opt out if desired.

**Enabling measurements for auditing** End-to-end violations are typically difficult to measure, for two reasons: *First*, it is often difficult to get measurement points in a large variety of networks, and *Second*, it is often challenging to recruit users to deploy measurements. Making this situation more dire is that it is necessary to run repeated experiments on a single node, as well as large numbers of experiments for nodes in particular networks, in order to properly diagnose different violations. Comparing the previous studies introduced in Section 2, our methodology can be deployed with low overhead and easily scaled up to millions of hosts. In this regard, we believe that our methodology will be attractive to both regulations and researchers as a means to quickly study end-to-end violations in networks around the world. Such information is useful not only for auditing network providers' compliance with existing regulations, but also for providing empirical evidence to inform future policies.

[5] For example, this may be TrendMicro's Web Reputation Services [13].

# References

[1] T. Chung, D. Choffnes, and A. Mislove. Tunneling for Transparency: A Large-Scale Analysis of End-to-End Violations in the Internet. *IMC*, 2016.

[2] D. Dagon, C. Lee, W. Lee, and N. Provos. Corrupted DNS Resolution Paths: The Rise of a Malicious Resolution Authority. *NDSS*, 2008.

[3] DNS Error Assist. `http://dnserrorassist.att.net`.

[4] L.-S. Huang, A. Rice, E. Ellingsen, and C. Jackson. Analyzing Forged SSL Certificates in the Wild. *IEEE S&P*, 2014.

[5] Hola VPN. `http://hola.org/`.

[6] C. Kreibich, N. Weaver, B. Nechaev, and V. Paxson. Netalyzr: Illuminating the Edge Network. *IMC*, 2010.

[7] M. Kührer, T. Hupperich, J. Bushart, C. Rossow, and T. Holz. Going Wild: Large-Scale Classification of Open DNS Resolvers. *IMC*, 2015.

[8] Open Internet Order - Federal Communications Commission. `https://apps.fcc.gov/edocs_public/attachmatch/FCC-15-24A1.pd`.

[9] Project BISmark. `http://projectbismark.net`.

[10] RIPE NCC Annual Report 2015. `https://www.ripe.net/publications/docs/ripe-665`.

[11] Senate Join Resolution 34. `https://www.congress.gov/bill/115th-congress/senate-joint-resolution/34`.

[12] K. Thomas, E. Bursztein, C. Grier, G. Ho, N. Jagpal, A. Kapravelos, D. McCoy, A. Nappa, V. Paxson, P. Pearce, N. Provos, and M. A. Rajab. Injection at Scale: Assessing Deceptive Advertisement Modifications. *IEEE S&P*, 2015.

[13] TrendMicro Web Reputation Services. `http://esupport.trendmicro.com/solution/en-US/1058991.aspx`.

[14] Verizon Search Assist. `http://searchassist.verizon.com`.

[15] N. Weaver, C. Kreibich, B. Nechaev, and V. Paxson. Implications of Netalyzr's DNS Measurements. *SATIN*, 2011.

[16] Wheeler and R. Protecting the Privacy of Customers of Broadband and other telecommunications services. 2016. `https://apps.fcc.gov/edocs_public/attachmatch/FCC-16-148A1.pdf`.

[17] X. Xu, Y. Jiang, T. Flach, E. Katz-Bassett, D. Choffnes, and R. Govindan. Investigating Transparent Web Proxies in Cellular Networks. *PAM*, 2015.

[18] C. Zhang, C. Huang, K. W. Ross, D. A. Maltz, and J. Li. Inflight Modifications of Content: Who Are the Culprits? *LEET*, 2011.