Somaroute: emotional contagion in social networks through communications infrastructure controls

Philip Daian and Jack LaSota
University of Illinois at Urbana-Champaign

Abstract—**In this project, we propose extending the techniques used by Facebook to experiment with the effects of emotional contagion in social networks to an attack on the communication infrastructure underlying web communications. We create a router firmware capable of altering packet routing times in real-time based on their emotional content, discouraging users from viewing content with certain emotional characteristics. We explore the practicality of this attack, its deployment vectors, its potential efficacy, and available and effective countermeasures, including a demonstration of a possible metadata-based attack on TLS encrypted web traffic.**

I. Context

The effect now known as emotional contagion was first described by psychologists in 1992 after noticing that their personal mood and emotional state was strongly related to the moods of their clients during their psychotherapy sessions. Emotional contagion refers to a transfer of emotional state through social networks - that is to say, if people around you experience happiness, the happiness has a tendency to positively impact your mood. This social transfer is often subtle, with the affected party being unaware of the effects of the contagion on their emotions. The original authors who noticed this effect explained its action by claiming that ”people do tend to automatically mimic or synchronize with the facial expressions, vocal expressions, patterns, and movements of those around them”, mechanisms which in turn were able to powerfully and subtly impact the mood of their subjects [2]. While this proposed mechanism has not been experimentally validated, this effect was originally studied by several other groups in physical social networks, attempting to find evidence of such mood transfers and to explain and take advantage of their effects.

One such group analyzed a dataset collected by the Framingham heart study, a study that sought to identify populational risks factors for heart disease and which recorded, among other things, data about emotional state and relationships between users. The group detected powerful clusters of participants with similar outlooks and emotions, noting that ”[happiness] is not merely a function of individual experience or individual choice but is also a property of groups of people... Indeed, changes in individual happiness can ripple through social networks and generate large scale structure in the network, giving rise to clusters of happy and unhappy individuals” [1]. This effect can be exploited to both predict and influence the emotions of an individual, and if accurate can powerfully model changes to emotional state on a macro level.

Despite this initial experimental evidence supporting the theory of emotional contagion, this theory remains controversial as a causal explanation for the emotional changes observed, with a limited body of evidence and many confounding factors.

In 2014, a study performed by Facebook Inc. and Cornell University analyzed 689,003 Facebook users in an attempt to answer two questions: whether this effect in real-world social networks extended to online text-based social networks, and whether the effect could be leveraged to alter the emotional state of these users in statistically significant ways. In doing so, the team also attempted to validate the theory of emotional contagion by producing statistically significant observable changes predicted by a contagion-based model. By tweaking the order of the content on the news feed of these participants split into testing and control groups, the study ”tested whether exposure to emotions led people to change their own posting behaviors, in particular whether exposure to emotional content led people to post content that was consistent with the exposurethereby testing whether exposure to verbal affective expressions leads to similar verbal expressions, a form of emotional contagion”, concluding that a statistically significant change in emotional state could be induced by manipulating the display of Facebook news feed posts and measuring this change in emotional state using the same metrics used to evaluate news feed items [5].

This study sparked intense moral and ethical controversy, generating over 40 comments on the PNAS website and thousands of mentions on Twitter and other social networks. The questions of whether it was ethical for Facebook to experiment on human subjects without anything interpretable as informed consent, whether it was ethical for PNAS to publish the study, and whether it should be legal for corporations to modify their products in opaque ways to affect the user in ways that may be outside the scope of the service were all raised and discussed extensively on the Internet as a result.

The Facebook study also highlights the application of this emotional insight to online advertising. Historically, advertising techniques are born from psychological analysis to attempt to influence purchasing decisions of market consumers. Facebook's interest in artificially introducing emotional contagion strongly suggests that advertising applications of such techniques are being actively considered and developed.

While the Facebook study and other previous work suggest that mood can and does spread through social networks, both physical and text-based, the implications and applciations of techniques leveraging this effect and their potential impact on and importance to the average social media user are as of yet unclear.

II. Problem

To understand the problem, we must first bring the work done by Facebook into a security and privacy context. We consider any changes in news feed algorithm designed to interact with the user in any way other than what is clearly presented (such as for reasons of altering an emotional state) an attack on the integrity of the service being used by the user. Although in this case the service is doing the manipulation and such manipulation can be considered routine and legal, it is likely that the end user is unaware of the motivation for such decisions and thus unable to anticipate their consequences. Such a content-based change may also be considered a violation of the confidentiality of a user's data, which may be used in ways not anticipated by or directly agreed to in an informed fashion by the user.

Having defined the violation of security above, the questions general to all attacks on the security of a system now present themselves. We first consider the attack vector, and whether it is possible to extend this attack to areas outside the scope of the service itself. For example, the HTTP protocol mainly used for communication on the Internet is widely known to be vulnerable to manipulation, packet inspection, and man-in-the-middle attacks. Similarly, the HTTPS protocol provides little security against such attacks: it is often possible to strip SSL from a session entirely, or perform an equally effective man in the middle with only a certificate warning displayed to the user.

Naturally, questions of scaling the attack then arise - if it is possible to systematically target the communications layer to leverage emotional contagion based effects to alter mood, due to the mass of data available on the Internet it is likely possible to do so in a targeted way, either for a particular region, topic, or subset of the population. The problems addressed by this work stem from the insufficiency of Facebook's study in viewing changes of users' news feeds for the purpose of inducing emotional changes as a security and privacy violation. In failing to adopt this mentality, Facebook's study did not publicly explore the full ramifications and impact of such a technique, opting instead to focus on only one specialized application directly relevant to their service.

The problem our work will attempt to address is thus that of the vectors for, efficacy of, and countermeasures against the manipulation of social networks to introduce contagion on the network communications level.

There are several questions orthogonal to but important in the understanding of our work. One is the ability of this technique to be used on a small scale - naturally, having proved the efficacy of this technique, Facebook has shown that even small scale and localized manipulation (eg - at a business or coffee shop) may be effective in altering emotional state over time for the subset of users using that network, as no conclusions they draw require manipulation of all users in a full social network. The open questions related to this include how many network users are required to generate a significant effect capable of having real-world consequences, and how much time is required for contagion-based effects to propagate through a social network. We will consider and explore these questions briefly as well in the work.

Another question inherent to this attack is its possible impact potential and severity outside of advertising-based applications. Can this technique be used, for example, to influence elections? A study on voting habits performed in 2010 analyzed how voters make decisions on their satisfaction with the government, and concluded that "important real-world decisions can be influenced by shifts in affect caused by events that are orthogonal to the decision at hand" [3], finding significant evidence that positive or negative emotional state could impact voting decisions in a statistically significant way. Leveraging emotional contagion could be one vector of influencing these emotions, prociding an extremely lucrative attack for which significant financial motivation exists. Additionally, the ability to invoke pro-government sentiment through positive emotions can be useful for furthering specific time-sensitive issues or interests.

This attack-based viewpoint also raises two questions of generalizability. The first is whether we can introduce emotional contagion by influencing users' moods in ways other than altering the content they are exposed to posted by their friends. One easy and potentially impossible to detect alteration is that of page load time - by merely introducing delays in the network, past analyses of eCommerce-based sites have shown potentially massive behavioral changes of web users, finding that slowing page loading by a second decreased queries after the page had loaded by 1% and ad clicks by 2.5%, with an additional second causing declines of 2.5% and 4.4% respectively [4]. To arrive at such numbers, a standard A/B testing framework for web applications was created, a method also generalizable to test the effectiveness of our traffic modifications given a sufficiently large sample dataset.

If the clearly psychologically significant effects of page load time can be used to trigger user unhappiness and frustration, it is possible they can be used in addition to the reordering techniques used in the Facebook study to influence certain users' moods, an effect which would be amplified by emotional contagion to potentially cause major populational changes in mood and behavior. It also follows logically from the contagion effect that only a very small percentage of the population needs to be influenced to cause widespread changes in emotional state, with effects of mood alterations being amplified by natural social networks.

The second question of generalizability concerns the applicability results of Facebook's study outside of Facebook. Are there other social-centric sites we can target, such as reddit, Twitter, or GMail, which may have the same impact on emotional state? Are there any unique social properties of Facebook that amplify this effect outside of its social features?

Both questions of generalizability are critical and as of yet unexplored aspects of the problem that will determine the utility of any solution we develop. Despite this, we will address them in our work only briefly and orthogonally, focusing on the technical evaluation of possible attack vectors, manipulation strategies, and countermeasures while deferring questions of efficacy evaluation to future work.

III. Approach

To address the problem, we propose first exploring the efficacy, practicality, and cost of altering network requests on-the-fly. We will use attempt to deploy a proxy in two places - on the user's computer and on a router we will modify for experimental purposes. We will use mitmproxy (http://mitmproxy.org/) to analyze what data is available by inspecting HTTP and HTTPS requests as users browse the Internet normally, and to test several techniques for altering this data. We will then also apply these techniques to a standard WRT54G running custom firmware, again through a custom HTTP proxy. After building these prototypes, we will analyze the cost of reaching users through various attack vectors, including router-based viruses/worms and solutions targeted at the end-user. We will then consider deployments at the infrastructure level, and again analyze the cost of the deep packet inspection techniques required to perform such an attack. We will also consider attacks with a small attack surface and high impact, such as targeted attacks on datacenter routers or other backbone services. To ensure we have considered all possible attack services, we will briefly analyze the flow of packets to popular web services through the various routers and entities responsible for processing them during routine requests.

We will briefly consider localized attacks, such as attacks deployed by a local business or service provider. While such attacks may not be effective as standalone attacks, it is possible that certain business (such as telecommunications providers) or large groups of businesses have enough network influence and control to alter the traffic of a potentially significant number of local residents.

Having explored the efficacy and cost of altering traffic on both mass and local scales, we will proceed to evaluate countermeasures available to the user for detecting various manipulation techniques leveraging the attack vectors previously explored. Such countermeasures have already been explored in previous work - work on web tripwire systems has for example explored client side integrity checks, being able to detect that page modifications in flight have not occured and concluding that such modifications do occur at a rate of $1 - 2\%$ in the observed sample [6]. These modifications were concluded to often be due to a middleware proxy similar to what we will be exploring for the purposes of traffic manipulation, indicating that significant precedent exists in both the traffic manipulation and manipulation detection approaches we propose.

Lastly, we will briefly discuss an evaluation framework for measuring the effects of such changes on the users they target. Such a framework will rely on the content of the pages users are visiting, as well as on gather other metrics which may be potentially statistically significant in determining user mood.

Together, the traffic analysis, manipulation, and evaluation network form the basis for ”Somaroute”, the prototype of a complete traffic manipulation system capable of mainpulating and evaluating network traffic on the fly.

IV. Contributions

Figure 1 shows a typical system architecture for Internet communications, in which a user is communicating bidirectionally with an application server through routes consisting of a large number of routing nodes, only some of which are in control of the user and the target organization. Any of the routers on this bidirectional path is potentially succeptible to traffic manipulation, and because there are no strict timing or latency guarantees in typical routing protocols, any of these routers can potentially both read unencrypted traffic and cause delays in network communications to the end user. The confidentiality, integrity, and availability of the traffic passing through these complex routing paths is unclear and difficult to distill to precise guarantees over the traffic being processed, providing ample room for attacks which are difficult to detect.

To build a prototype router which dynamically alters packet routing latency in real time based on the emotional content of the packets being routed, we first choose appropriate hardware for a scalable and realistic proof of concept.

A. Preparing the Hardware

Figure 2 shows the two hardware devices we obtained for testing our architecture. We aimed for devices deployed at the consumer level, which are more affordable to analyze than costly infrastructure-level equipment routing packets for service providers. The first router we chose is a Linksys WRT54G, one of the most ubiquitious WiFi routers ever sold [7]. Because the WRT54G only supports Wireless G and is several years old, one can easily be obtained for under 20USD.

The second device was a Ubiquiti Unifi Enterprise AP, with an off-the-shelf price of around 100USD new. This router is commonly used in enterprise deployments and to provide public WiFi in a managed mesh network, with a focus on handling a large range and a substantial amount of traffic. The device we used had the maximum wattage allowable by the FCC for consumer wireless devices, with a range of up to 1km out of the box.

While we chose the WRT54G to demonstrate how cost-effective and low resource our attack could become, the Ubiquiti Unifi shows the potential for the attack to be extended to cover large areas, potentially by institutions to exercise control in areas in which they control infrastructure.

Another reason for choosing consumer hardware was to prove the concept of malicious routing firmware, which could potentially spread like a virus and infect such consumer routers to implement a larger Somaroute network. With an attack that runs comfortably at the endpoints of a user’s connection, it also follows that the attack can easily be generalized to more powerful or specially purposed infrastructure-oriented routing equipment.

B. Bare HTTP MITM with TinyProxy

Tor implement our emotion-based traffic shaping for HTTP sessions and requests, we designed the architecture in Figure 3. We chose Tinyproxy as an HTTP proxy for our purposes, mainly due to its high performance, extensibility and extremely small code size. Strictly speaking, a proxy is not required for this attack, with any program able to analyze and modify the speed of packet flow being sufficient for our conditions.

In our implementation, a firewall redirected HTTP traffic on Port 80 to proxy traffic on Port 3128. The first step was to strip the accept-encoding header from all HTTP requests, ensuring traffic between our client and the webserver is uncompressed. Compressed traffic naturally makes analysis more difficult by requiring decompression for analysis, a costly and slow procedure which requires knowledge of the entire response to perform.

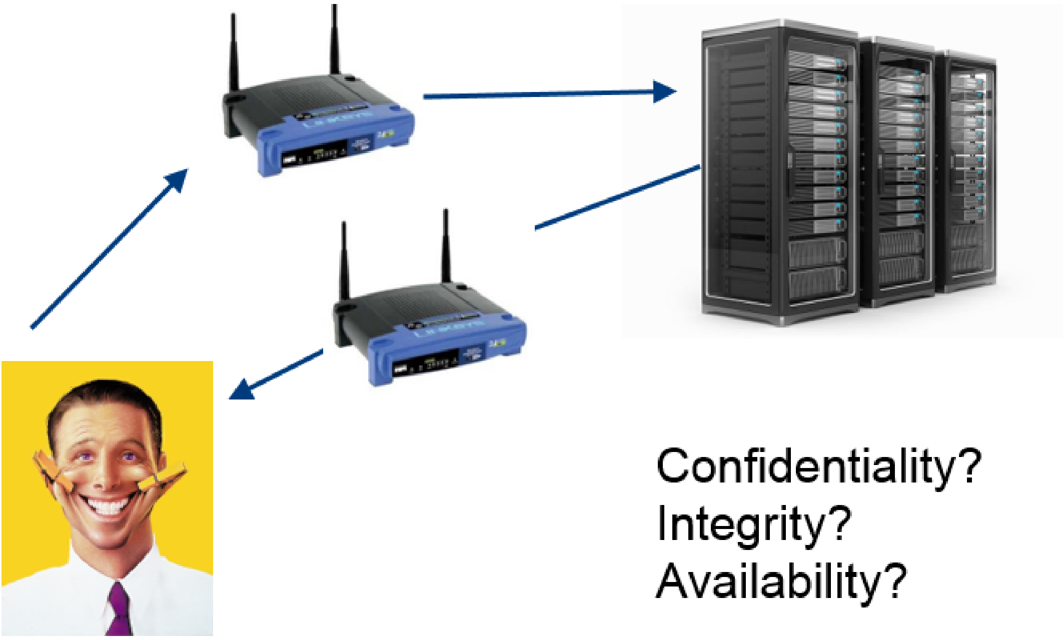


Fig. 1: Overview of traditional Internet architecture

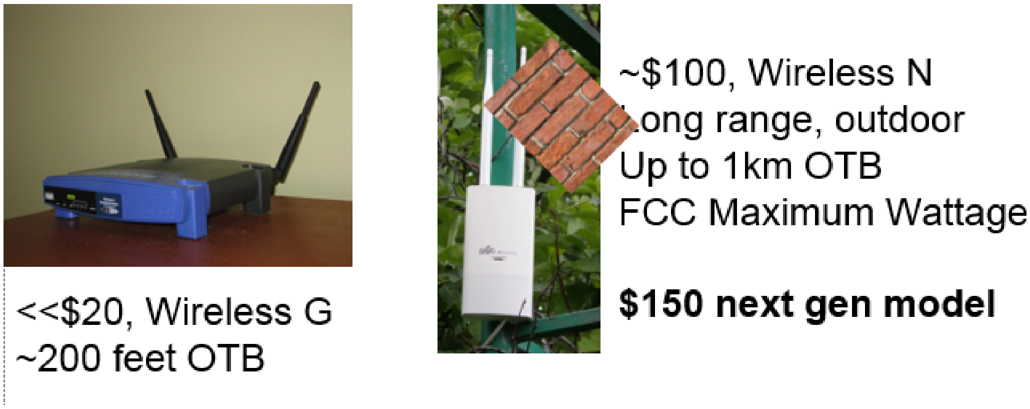


Fig. 2: Two OpenWRT-enabled hardware devices supporting Somaroute

After this, our outgoing traffic is placed in a buffer that is then sent to the destination webserver. The return traffic flows through a similar buffer which is modified to analyze the content for emotional content and decide a content delay. Before being sent back to the user from this buffer, the content is delayed by the decided time. This allows us to precisely control and manage both the delays and content analysis of our user's browsing.

Figure 4 shows the word lists used in our simplistic content analysis heuristic. More comprehensive emotional mappings of language exist, like the LIWC [8], which is the result of analyzing and categorizing over 250 million words for their emotional content, cognitive and biological functions, grammatical purpose, personal concerns, and the core drives or needs they represent. With a sophisticated classification like the LIWC, ascertaining the emotional content of pages is possible to an extremely high degree of accuracy.

Unlike these accurate measures, our heuristic simply counted the number of occurences of words considered both ”negative” and ”positive” on a page, with these words shown in Figure 4. If the number of negative emotional words was at least 50% higher than the number of positive words, we classified the page as emotionally

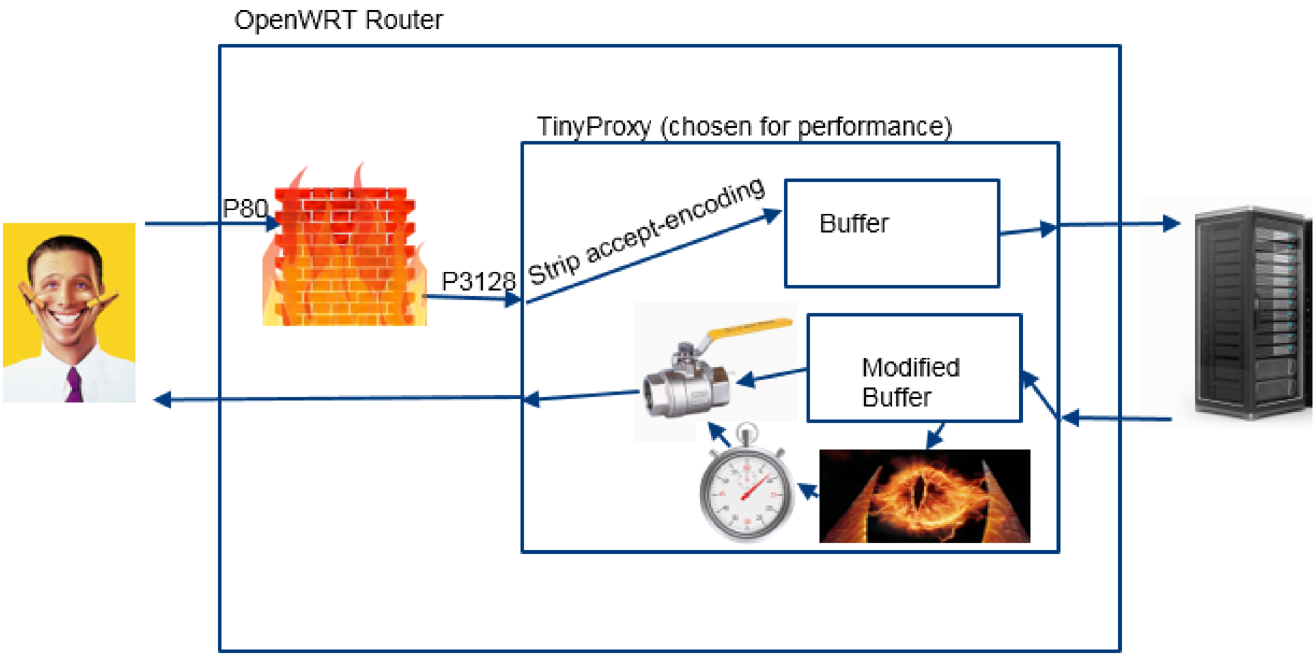
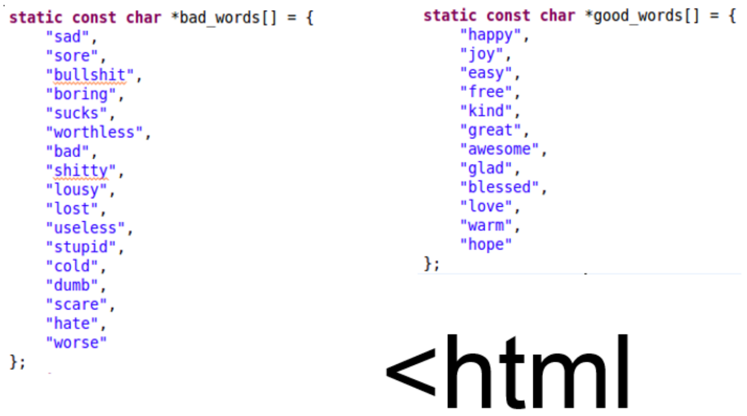


Fig. 3: Implementation of tinyproxy HTTP MITM



```
static const char *bad_words[] = {
    "sad",
    "sore",
    "bullshit",
    "boring",
    "sucks",
    "worthless",
    "bad",
    "shitty",
    "lousy",
    "lost",
    "useless",
    "stupid",
    "cold",
    "dumb",
    "scare",
    "hate",
    "worse"
};
```

```
static const char *good_words[] = {
    "happy",
    "joy",
    "easy",
    "free",
    "kind",
    "great",
    "awesome",
    "glad",
    "blessed",
    "love",
    "warm",
    "hope"
};
```

Fig. 4: Word lists for simple proxy filtering

negative and applied an appropriate delay. Conversely, pages that were classified as emotionally positive were not subjected to any delays.

While this heruistic is merely a proof of concept, it accurately mirrors Facebook's experiment in emotional contagion on their social media pages, with users being more likely to be exposed to content that is positive than content that is negative. Because Facebook saw statistically significant results in its simplistic proof of concept, we can likely expect the same results with our proof of concept, though for true efficacy our wordlist should likely be expanded further.

C. Handling SSL

The most popular method for preserving security properties of Internet communications is by far the SSL protocol [9], formalized as transport-level security. SSL is designed to provide confidentiality and integrity of communications through a hierarchy of trusted certificate servers.

Because SSL encrypts the contents of its communications, the simple man in the middle attack we previously performed using TinyProxy to detect the emotional content of pages being routed through our router is no longer possible. Despite this, fundamental flaws in the design of SSL allow us to extend our attack to SSL-based sessions.

Despite this, there is a fundamental flaw in the SSL protocol that does not allow it to provide for full confidentiality of data. Because each SSL session occurs over some transport layer protocol that is unencrypted, both the IP and the port of these sessions are visible as the user browses. This metadata can be extremely powerful in determining exactly what content a user is viewing, as another stipulation of the SSL protocol is that each certificate is uniquely bound to a host address and port. Additionally, SSL certificates are exchanged in the clear, allowing them to be extracted from a user's sesison as they browse. Because SSL certificates are often bound to a particular service or subdomain (with for example a different certificate protecting mail.google.com as opposed to news.google.com, we can often know what a service a user is browsing with some granularity.

By fetching this service in the background, our attack can thus proceed similary to that of HTTP: we first extract the certificate, then fetch the page of the same host as the user in the background (or query an internal database of which services we consider "positive" and which we consider "negative"), then slow down or speed up the traffic accordingly.

To accomplish the latter, we employ a token-bucket based quality of services approach, shown in Figure 6. Such quality of service approaches classify traffic into various token buckets, allowing traffic into the buckets from a buffer at a set rate, and dropping extra packets from the buffer entirely (which means that because SSL runs over TCP, they must be resent). Through this approach, we can control user traffic with a high level of granularity.

Figure 7 shows our rules for token bucket filtering, with four traffic classes. The first with a token bucket rate of 20 megabit, the second with a token bucket rate of 19 megabit (the two faster token buckets), the third with a token bucket rate of 2 megabits (some noticeable slowdown in traffic), and the fourth with a token bucket rate of 1 kilobit (extremely major slowdown in traffic with a severe impact on usability).

Based on a static cached table which we fill out by fetching the hosts the users are browsing to concurrently with their activity, we then classify traffic into one of the four buckets. As with the previous attack, users are psychologically discourage from browsing any hosts we filter into the two slower buckets.

So, because of the lack of confidentiality in SSL metadata, which leaks user session information through the SSL certificates of the services being browsed, we are able to extend our attack to even traffic which is supposed to be protected for integrity and confidentiality by the most popular cryptographic security protocol on the Internet. This leak suggests the need for a new routing protocol to counter our attack, with strong metadata protection and formal confidentiality and integrity guarantees.

D. Attack Legality and Practicality

There are two key questions concerning the application of such an attack to the general population: is it legal, and does it scale? The right of ISPs to both inspect user packets for traffic analysis and discriminate against the routing speed of certain packets is well established. Comcast is well known to use traffic shaping approaches similar to those discussed to discriminate against traffic it considered to be over P2P networks, and has been known to use spoofing techniques in the past, with the EFF saying that Comcast's "new system appears to be a reasonable attempt at sharing limited bandwidth amongst groups of users. Unlike TCP RST spoofing, it doesn't explicitly discriminate against some applications, and it doesn't threaten protocol developers with interoperability problems and uncertainty about network behavior" [10].

Thus, the attack remains both legal and scalable for service providers owning the backbone infrastructure through which network data is routed. There are a large number of such providers [11] with complex economic and regional interests all over the world, and it is very difficult for a user to tell exactly when and which such providers their traffic is passing over. Because any one of these providers can implement the attacks we describe in the paper, we strongly believe in the practicality of such an attack for a motivated provider.

E. Effective Countermeasures

Having established the practicality of such an attack, we now analyze effective countermeasures that can be taken by users against this attack. Figure 8 shows the impact of our static delay infrastructure on Firefox's network timing analysis module, showing a clear delay

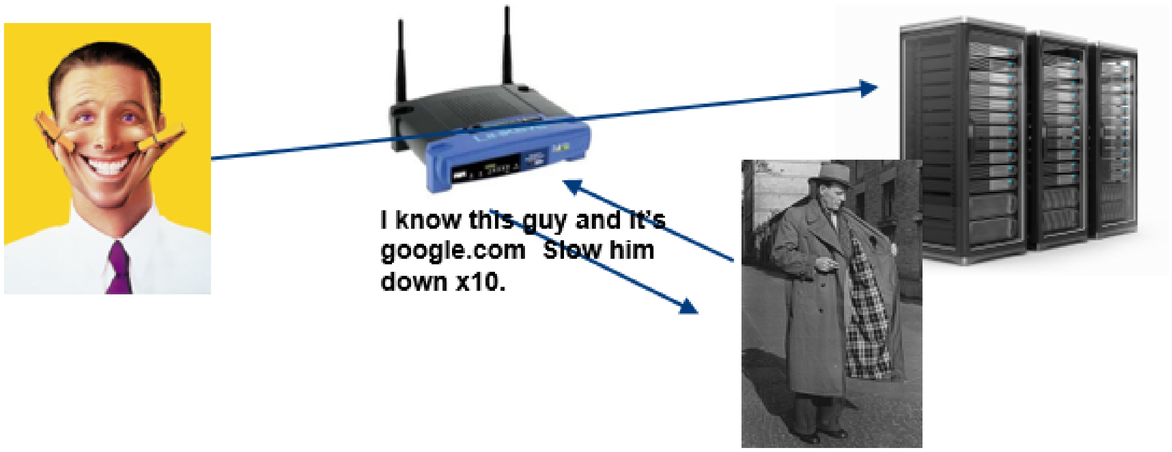


Fig. 5: SSL-based filtering architecture

in traffic on some websites. Thus, timing analysis is one potential avenue for the detection of such shaping early. Unfortunately, with randomizable delays and delays that can be caused by any number of opaque infrastructure factors, such attacks can be extremely difficult to detect through timing analysis. An analysis of a large number of aggregate timing data can however still be used, detecting additional congestion on some routes on average over the sessions of a large number of users. Such strategies have been implemented by the EFF to detect the P2P traffic shaping we mentioned previously, with a conservative test suite called Glasnost available to detect the potential throttling of various P2P protocols [12].

Other potential countermeasures include approaches designed to obscure the source and destination of Internet traffic, including VPNs, IPSEC, and the Tor routing protocol. While these protocols are useful for this purpose, with current Internet infrastructure they are still required to interface with unencrypted TCP routing at some endpoint through which traffic flows: for VPNs and IPSEC, this is the server being routed through, and for the Tor network this is the endpoints. Because our attacks are not user-targetted, targetting these interfaces between the anonymity technologies and the Internet at large can be an effective strategy for applying our attack to otherwise confidential traffic.

The true countermeasure we suggest is that of an encrypted routing protocol providing strong security and anonymity guarantees. The backbone infrastructure of the Internet would need to be updated to support such a protocol natively, requiring no legacy interfaces with the insecure and vulnerable TCP or UDP routing protocols. One such routing protocol is the Onion network, which is designed to obscure both the source and destination of Internet traffic when used internally to connect to ”hidden services” [13]. Widespread adoption of a similar protocol at the infrastrucuture level will prevent entirely the shaping and analysis we discuss, though such an effort is extremely expensive and likely impractical given the current demands of users and businesses for low-latency, high-speed infrastructure, and the reduced emphaiss placed on privacy in this context.

V. Future Work and Challenges

Any future work we expect on this issue will likely largely focus on proving the efficacy of our techniques in an experimental setting. While we will provide a theoretical framework for traffic modification in a variety of settings, the efficacy in the real world of such frameworks is unclear, as is the applicability of Facebook's statistically significant results to such a generalized framework. Furthermore, the questions of generalizability to other techniques (such as page load times) and to diverse applications presented in the Problems section of our work will remain unclear without extensive further A/B studies. Lastly, the important question of how to select which users or data to manipulate for maximum effect is not only critical, but non-obvious and complicated. These questions can only be solved through psychological-style

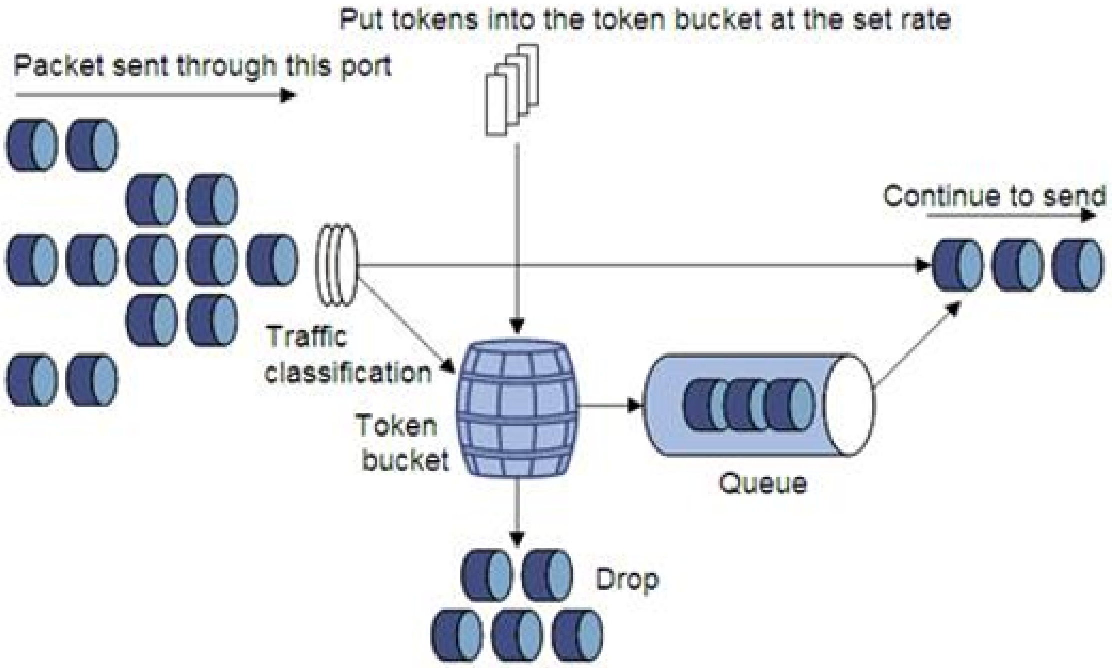

Put tokens into the token bucket at the set rate
Packet sent through this port
Continue to send
Traffic classification
Token bucket
Queue
Drop

Fig. 6: Token bucket-based quality of service

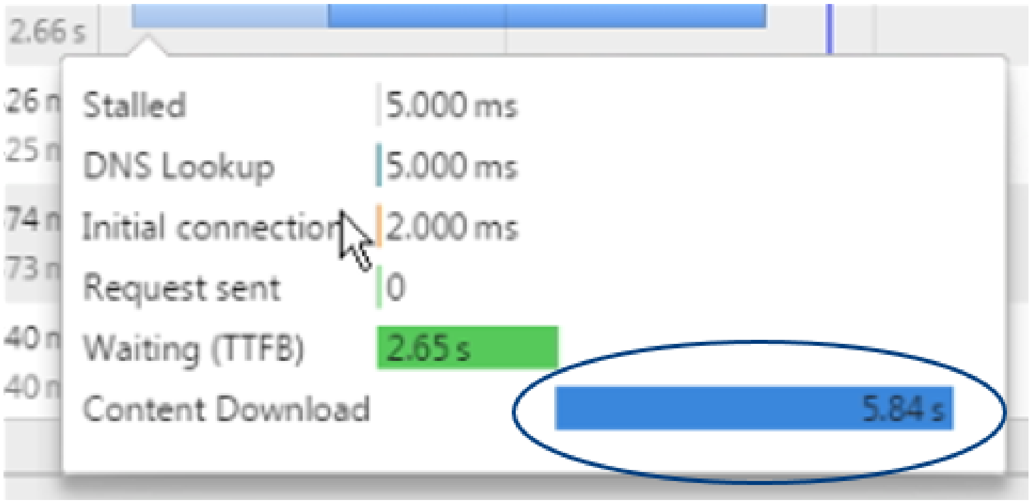
```
tc qdisc add dev br-lan root handle 1: htb default 30
tc class add dev br-lan parent 1: classid 1:1 htb rate 20mbit burst 15k
tc class add dev br-lan parent 1:1 classid 1:10 htb rate 19mbit burst 15k
tc class add dev br-lan parent 1:1 classid 1:20 htb rate 1mbit ceil 2mbit burst 15k
tc class add dev br-lan parent 1:1 classid 1:30 htb rate 1kbit ceil 20kbit burst 15k

tc qdisc add dev br-lan parent 1:10 handle 10: sfq perturb 10
tc qdisc add dev br-lan parent 1:20 handle 20: sfq perturb 10
tc qdisc add dev br-lan parent 1:30 handle 30: sfq perturb 30

iptables -t mangle -A POSTROUTING -j CLASSIFY --set-class 1:10
```

Fig. 7: Token bucket-based quality of service

A/B testing on a large number of users, due to the the subtlety of the effects noted by the Facebook study.

Such testing may be illegal, impossible to obtain consent for, or impractical. Completing such experiments with a large sample size will require the cooperation of the large coroprations and services targeted by these attacks, with such services having no incentive to publicly produce such experimentation. Furthermore, the extensive ethical questions raised by the Facebook study remain important to any future experimentation.

The practical challenges faced in the continuation of such work are likely thus enormous. Nonetheless, we believe it is possible to demonstrate that entities incentivized and financially able to complete such attacks do exist, many of whom financially support related research in the area. It is thus likely that similar non-public research is proceeding somewhere, and therefore important to evaluate such attacks and techniques in a public context to the maximum possible extent, highlighting the possible risks and pitfalls of the widespread overuse and trust of centralized and opaque routing infrastructure that occurs in the clear, leaks metadata, or does not provide strong cryptographic integrity and confidentiality



Stalled 5.000 ms

DNS Lookup 5.000 ms

Initial connection 2.000 ms

Request sent 0

Waiting (TTFB) 2.65 s

Content Download 5.84 s

Fig. 8: Analyzing responses for timing

guarantees.

References

[1] Flowler H and Christakis NA(2008), *Dynamic spread of happiness in a large social network: longitudinal analysis over 20 years in the Framingham Heart Study*. BMJ 337:a2338.

[2] Hatfield E, Cacioppo JT, and Rapson RL(1993), *Emotional contagion*. Curr Dir Psychol Sci 2(3):96100.

[3] Healy AJ, Malhotrab N, and Hyunjung C(2010), *Irrelevant events affect voters' evaluations of government performance*. PNAS 107(29):12804-12809.

[4] Kohavi R, Longbotham R, Sommerfield D, and Henne R(2009), *Controlled experiments on the web: survey and practical guide*. Data Min Knowl Disc 18:140181.

[5] Kramera ADI, Guillory JE, and Hancock JT(2014), *Experimental evidence of massive-scale emotional contagion through social networks*. PNAS 111(24):8788-8790.

[6] Reis C, Gribble SD, Weaver NC, and Kohno T(2008), *Detecting in-flight page changes with web tripwires*. NSDI published online at https://www.usenix.org/legacy/events/nsdi08/tech/full_papers/reis/reis_html/index.html (accessed 10/22/2014).

[7] Higgins, T(2006), *Yes, the Linksys WRT54G V5 Really Is a Lousy Router*. SmallNetBuilder published online at http://www.smallnetbuilder.com/wireless/wireless-reviews/26843-linksyswrt54gv5reallyisalousyrouter (accessed 11/22/2015).

[8] *Linguistics Inquiry and Word Count*. Pennebaker Conglomerates, Inc. published online at http://www.liwc.net/tryonline.php (accessed 11/22/2015).

[9] Vehent, J(2014) *SSL/TLS analysis of the Internet's top 1,000,000 websites*. LinuxWall published online at https://jve.linuxwall.info/blog/index.php?post/TLS_Survey (accessed 11/22/2015).

[10] Eckersley, P(2008) *Comcast Unveils Its New Traffic Management Architecture* . Electronic Frontier Foundation published online at https://www.eff.org/deeplinks/2008/09/comcast-unveils-its-new-traffic-management-archite (accessed 11/22/2015).

[11] Malecki, E(2002) *The Economic Geography of the Internet's Infrastructure* . Economic Geography, 78(4), 399424. doi:10.2307/4140796.

[12] *Glasnost: Test if your ISP is shaping your traffic* . Electronic Frontier Foundation published online at http://broadband.mpi-sws.org/transparency/bttest.php (accessed 11/22/2015).

[13] *Tor: Hidden Service Protocol* . The Tor Project published online at https://www.torproject.org/docs/hidden-services.html.en (accessed 11/22/2015).