

School of Engineering

Department of Electronic & Computer Engineering

# Test the testers (come up with a fancy name)

Author: Hlib Fedchuk

Supervisor: Stephen Farrell

A thesis submitted in partial fulfilment of the degree of BAI in

Computer & Electronic Engineering

May 2018

***Table of Contents(needs fixing):***

Abstract……………………………………………………………………………………………………………………………………3

Inroduction………………....................................................................................................................3  
  Background………………………………………………………………………………….…………………………………….xx  
 -crap web sites  
-getting less crap

-measuring via n/w visible indicators  
-assumption is that if you tell people "here's HOWTO get an A"  
-research question: what's the value of the different measures/scores?  
ANS:if they had commonality then site owners might  
                  have less work to do to test their shit and hence  
                  more time to improve their shit

Methods………………………………………………………………………………………………………………………………….xx  
- internet measurement, now practical wasn't before  
        - measuring the measurers  
        - commensurate/incommensurate scoring, 105/100 good example   
          is A == A ? etc. github.com mention that sites start with initial 100 score and get +/- points for features  
        - not all measures make sense (e.g. CSP depends on  
          content being served)  
        - loadsa text about specific tests done (heartbleed, hsts,...)  
        - initial list of possible testers  
                = reduce list based on scriptability  
                = try include most commonly used (sslabs, moz,..)  
        - reproducible: what hlib does needs to be re-doable by all

Design/implementation………………………………………………………………………………………………xx

- set of test sites  
        - set of testers  
        - scripting/screen-scraping  
                = try find CLI/json-output  
                = worst case: hack screen-scraping  
                = even worst case: headless browser & har files  
                        - ok to say thought about but didn't need  
                          (if true)

Results…………………………………………………………………………………………………………………………..xx

Conclusion and future work……………………………………………………………………………………………xx

- it's great but could be better, here's how  
        - ask testers to do X,Y,Z to improve commonality

Acknowledgements……………………………………………………………………………………………………….xx

References…………………………………………………………………………………………………………………….xx

Appendices…………………………………………………………………………………………………………………….xx

***Abstract:***

This project is about comparing and contrasting different web site testers. During the period of this project’s progress, a list of tester and test sites will be picked in order to come up with results. The aim of this project is to differentiate between the tester’s results output in order to see and pick out the key differences between them. A set of web testers will be broken into three different categories: DNS testers, TLS (mail) testers and overall website testers. This separation was introduced in order to have a set of results that would relate to each other and get rid of confusion, where it would be very impractical to compare a TLS tester with a DNS tester, where there may be some similarities between them, but we want to obtain results of the same lookup area. During this project, the results would depend on the:

1. Overall score:   
   Typically, a tester would provide the overall score to the user that would indicate how well or bad their website/mail server is implemented in terms of security aspects. This result value cannot be discarded and should be taken very seriously, as it is the final score that the tester provides on how well the security features are implemented in this website, whether they are coming from a DNS/TLS or overall web tester.
2. Security and privacy issues:  
   A majority of the web testers would provide a detailed description on how they came up with the end score (privacyscore doesn’t). This description will provide the information on the points (that this specific tester thinks are important), whether they are implemented or not. For example, 2 different test may consider the same feature differently, some might penalize for it, giving it a negative mark, and some may just leave it and give a 0 mark for it.
3. Scope of implemented features:   
   This aspect will consider the current amount of implemented features that the web site has.   
   We can consider a small web site and a much bigger one, but the smaller one can get more points depending on the implementation, if more measures are implemented – it wouldn’t be a surprise that the score will be higher.

***Background:***

***Methods:***

***Internet measurement:***

Internet has become a huge aspect of human lives for the past ten years. It includes different branches like: international commerce, communication and technological development. It is important that various groups that affect and use internet, like researches, entrepreneurs, service providers and other members of internet community understand the growth characteristics and its limitations.

At early stages internet measurement was neglected giving it a low priority to increasing network’s speed, capacity and coverage. The increase in interest towards the internet only come pretty recently and it is a logical outcome as internet has expanded, which led to an increase in the number of users and malicious behavior. In order to counter these malicious elements, there needs to be a huge investment into Internet measurement and data analytics. Merit Network operated the NSFnet backbone in its various forms, measured the backbone’s traffic volumes and produced summary statistics through April 1995. But these were primarily oriented toward short-term operational requirements or periodic simplistic traffic reports for funding agencies. As such, they weren’t conducive to workload or performance characterization, much less network-dynamics modeling. As the NSFnet and attached regional infrastructures exploded in popularity among academic and commercial sectors, operators acutely focused on increasing link speeds and router/switch-traffic capacities. Developers worked on improving protocols and inventing new ones to support emerging services. The evolutionary context of the infrastructure left little room for more than mild interest in network measurement. For ordinary people, the Internet has become an integral part of everyday life; it is now used continually to find information, buy products, meet people, do our jobs, and play. As if these circumstances weren’t sufficiently revolutionary, the pervasive adoption of mobile computing expectations and requirements is now prompting service providers to take a strong interest in more strategic measurement and charging schemes.

With its ever-growing user community, the Internet has gradually been forced over the past decade to deal with the “real world.” Like chemical pollutants from industrial production processes, infrastructural pollution — such as viruses, worms, and spam traffic — has become significant in volume and impact on user productivity. Protective technologies such as firewalls and NAT gateways have changed the Internet’s simple end to-end connectivity model. Although these devices can effectively block some malignant packets, they do so by filtering packets according to access control lists (ACLs), which can prevent many applications from working properly.

The Challenges Collection, interpretation, and modeling of empirical Internet data remains challenging. The technologies and protocols involved in generating and delivering Internet traffic were designed for technical expediency, architectural clarity, and functionality, rather than for measurement and analysis. New developments often introduce specifications that are independent of their predecessors; technology developers often deploy them as rapidly as possible, without concerted systematic testing on the vast set of heterogeneous components encountered on the Internet. Indeed, it would be impossible to test certain behaviors against all possible combinations of equipment, software, and configuration. Furthermore, many who develop technologies and protocols contend that the Internet has evolved splendidly thus far without extensive measurement and modeling. Others believe that we should not begin measurement and modeling efforts until doing so proves cheaper than simply expanding the currently available bandwidth. To make matters harder, a variety of legal and privacy issues serve as active disincentives to measurement research and development activity. Nonetheless, every constituency of the Internet (providers, vendors, policymakers, and users) realizes that we need a better understanding of Internet structure and behavior, including the influence of various components and functionalities on macroscopic dynamics. Floyd and Paxson’s landmark paper provided several insights into why the Internet is hard to measure, and thus hard to simulate, making it resistent to modeling and predictive insight.2 The first big challenge is that everything keeps changing. For example, HTTP traffic grew from zero in 1995 to more than 80 percent of the network traffic at many sites by the early 2000s. Yet, HTTP’s proportion of total traffic is now dropping on most links, and peer-to-peer traffic is steadily rising as developers find more ways to use P2P technology. The Internet’s global scale also complicates measurement efforts, as does the fact that many aspects of traffic and behavior change from location to location. Thus, statistics gathered at one location often prove unrepresentative of the global Internet. Instead, we need to make measurements at many sites and correlate the results to derive a comprehensive view. Finally, few Internet protocols and applications were designed to inherently support fine-grained measurement. Instead, researchers have had to find indirect ways to measure network phenomena. For example, traffic-flow measurements rely on data collected from packet headers as they pass across links; counting packets and bytes and classifying them into flows on the basis of values taken from the headers is easy but yields limited insight into higher-layer behavior. Measuring application performance generally remains a challenge, since applications differ as to how they transport application-specific data. For example, while effective tools exist for measuring Web server performance, such tools are often not effective at measuring performance of other applications.

***Results:***

***Conclusion***

***References:***

[http://www.axiompanbiog.com/Pages/HyperGraph.aspx,](http://www.axiompanbiog.com/Pages/HyperGraph.aspx)

[https://networkx.readthedocs.io/en/stable/.](https://networkx.readthedocs.io/en/stable/),

[http://courses.cs.ut.ee/2011/graphmining/Main/Demos,](http://courses.cs.ut.ee/2011/graphmining/Main/Demos)

[https://phys.org/news/2009-02-greedy-routing-enables-network.html,](https://phys.org/news/2009-02-greedy-routing-enables-network.html)

[http://math.stackexchange.com/questions/1039308/graph-theory-clique-graph,](http://math.stackexchange.com/questions/1039308/graph-theory-clique-graph)

<http://www.scielo.br/scielo.php?script=sci_arttext&pid=S1415>[-47572008000400001,](http://www.scielo.br/scielo.php?script=sci_arttext&pid=S1415-47572008000400001)

a lot of links for lit reviews and examples online + tester documentation