e-ISSN: 2455-5703

DDoS Attack Detection and Elimination

[1]Ragu Raman R [2]Vinoth Ram S S [3]Ms. Anitha E
[1,2]Student [3]Assistant Professor
[1,2,3]Department of Information Technology Engineering
[1,2,3]Loyola-ICAM Chennai, India

Abstract

Distributed Denial-of-Service (DDoS) attacks are usually launched through the botnet, an "army" of compromised nodes hidden in the network. Inferential tools for DDoS mitigation should accordingly enable an early and reliable discrimination of the normal users from the compromised ones. Unfortunately, the recent emergence of attacks performed at the application layer has multiplied the number of possibilities that a botnet can exploit to conceal its malicious activities. New challenges arise, which cannot be addressed by simply borrowing the tools that have been successfully applied so far to earlier DDoS paradigms. In this work, we offer basically three contributions: i) we introduce an abstract model for the aforementioned class of attacks, where the botnet emulates normal traffic by continually learning admissible patterns from the environment; ii) we devise an inference algorithm that is shown to provide a consistent (i.e., converging to the true solution as time elapses) estimate of the botnet possibly hidden in the network; and iii) we verify the validity of the proposed inferential strategy on a testbed environment. Our tests show that, for several scenarios of implementation, the proposed botnet identification algorithm needs an observation time in the order of (or even less than) one minute to i dentify correctly almost all bots, without affecting the normal users' activity.
Keyword- Distributed Denial-of-Service, DDoS, Inference Algorithm, Botnet, Botmaster

I. INTRODUCTION

It is a strange face up to trackback the basis of Distributed Denial-of-Service (DDoS) attacks in the Internet [1]. In DDoS attacks, attackers engender a massive sum of requests to fatalities from side to side compromised computers (zombies), in the midst of the intend of refuse usual examine or corrupting of the excellence of services[2]. It has been a foremost hazard to the Internet since years, and a current analysis on the prime Internet operators in the world established that DDoS attacks are growing severely, and individual attacks are stronger and refined.

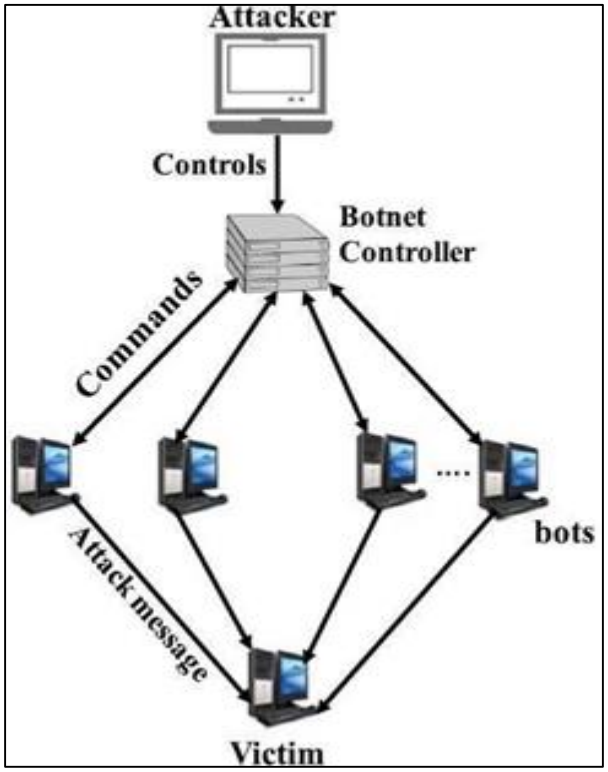
Additionally, the review moreover set up with the intention of the crest of 40 gigabit DDoS attacks almost doubled in 2008 contrast with the earlier year [3]. The major grounds behind this occurrence is to facilitate the network security community does not have useful and capable sketch back methods to situate attackers as it is effortless for attackers to masquerade themselves by taking compensation of the vulnerabilities of the World Wide Web, such as the vibrant, stateless, and unidentified character of the Internet.

AS advances in networking technology help connect every nook and corner of the globe, the openness and scalability are giving birth to a large number of innovative, interesting and useful online applications. With the proliferation of these IT enabled applications, more and more important and valuable information is flowing across public networks. Networks are usually designed to make efficient use of shared assets among network users [1] and network-based computer systems play a vital role in our personal as well as professional activities. Today, the Internet interconnects billions of computers, tablets and smart phones, providing a global communication, storage and computation infrastructure. Furthermore, the integration of mobile and wireless technologies with the Internet is currently ushering in an impressive array of new devices and applications.

These tremendous technological advances in terms of speed, accuracy, reliability and robustness of the modern Internet has made significant impacts on our day-to-day activities and people now rely on the Internet to share valuable and confidential personal as well as business information with other network users. On the other hand, because of the high reliance on the Internet, some people also make us e of the weaknesses of the Internet to paralyze it.

For example, one of the major weaknesses of the Internet is speed mismatch between edge routers and core routers. Inappropriate configuration of routers is also another common weakness of the Internet. Due to such weaknesses, networked systems have often become the targets of various attacks, which are launched in order to unlawfully gain access to important and confidential information or to damage useful resources of a business competitor using different attack tools [2], [3]. Although, in the recent past, many significant developments in constructing firewalls and cryptographic systems have taken place, they are not free from limitations. Defence mechanisms that identify intrusions provide another way to protect networked systems from attacks. In spite of all these safeguards, almost every day, new and complex attacks are being created. However, denial of service attacks are still the most frequent and usually the most devastating ones.

A. Architecture



II. EXISTING SYSTEM

DDoS is a coordinated attack, generated by using many compromised hosts. An attacker initially identifies the vulnerabilities in a network to install malware programs on multiple machines to bring them under his control. Then the attacker uses these compromised hosts to send attack packets to the victim without the knowledge of the compromised hosts. Depending on the attack packet intensity and the number of hosts used for attacking, commensurate damage occurs in the victim network. If the number of compromised hosts is very large, it may disrupt a network or a Web server in a very short period of time. Some examples of DDoS attacks include smurf, fraggle and SYN flooding. The aim of a DDoS attacker is to disrupt a network so that it cannot provide any services to legitimate users (Fig. 1). To launch an attack, an attacker generally follows four basic steps.
Information gathering to scan a network to find vulnerable hosts to use them later to launch an attack,

1) Compromising the hosts to install malware or malicious programs in the compromised hosts so that they can be controlled only by the attacker,
2) Launching the Attack To Command The Zombies To Send Attack Packets With Specified Intensities To The Victim, And
3) Cleaning up to remove all records or history files from memory.

III. PROPOSED SYSTEM

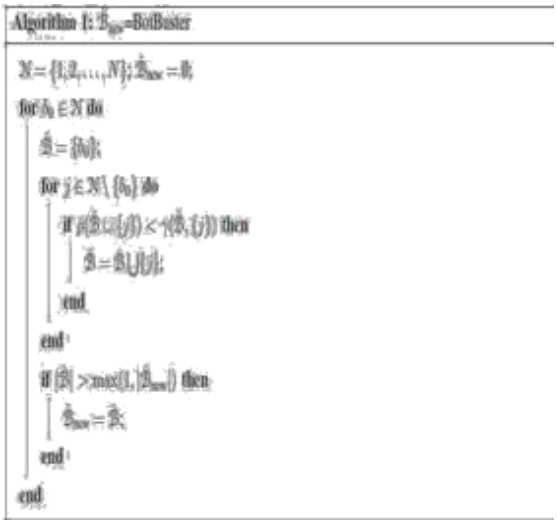
In recent times, most sophisticated DDoS attacks have been launched using botnet technology. The growth of botnet technology is enabling the generation of various types of DDoS attacks due to the flexibility and power of the technology. The four main reasons behind the preference for this technology for the attackers are:

1) Inclusion of a large number of zombie nodes allow generation of a powerful flooding attack quickly,
2) Great difficulty in finding the identity of the actual attacker,
3) Ability to use protocols to bypass security mechanisms, and
4) Difficulty in detecting in real time due to its similarity with the normal traffic.

We now focus on the derivation of the inference algorithm aimed at disclosing a botnet possibly hidden in the network. The Bot Buster algorithm is described by the pseudo-code reported in the right column above, and basically exploits the fact that, given two disjoint subnets, the BIC allows to discriminate the situation where both subnets are part of a botnet, from the situation where at least one of them is made of normal users. We shall show that the proposed algorithm possesses the fundamental requirement of consistency, namely, the guarantee that the botnet is correctly identified as it grows.

We show using a sample data between the two inference as well as the Botbuster algorithm the various advantages of the inference algorithm over the Botbuster algorithm. Moreover the Botbuster algorithm does not guarantee a hundred percent success of detecting and eliminating of a botnet virus unlike the inference algorithm which is proven.

Here we use a simulation tool called Network Animator to illustrate how the attack is being made on a receiver through the nodes by an attacker and how we are using the algorithm to detect the attack and put an end to the attacking nodes. Although we are not preventing all types of attacks we are able to prevent a few types of malwares effectively and efficiently in the shortest possible time.



BotBuster Algorithm

IV. THE BOTBUSTER ALGORITHM

We now focus on the derivation of the inference algorithm aimed at disclosing a botnet possibly hidden in the network. The BotBuster algorithm is described by the pseudo-code reported in the right column above, and basically exploits the fact that, given two disjoint subnets, the BIC allows to discriminate the situation where both subnets are part of a botnet, from the situation where at least one of them is made of normal users. We shall show that the proposed algorithm possesses the fundamental requirement of consistency, namely, the guarantee that the botnet is correctly identified as t grows. Let us examine how the algorithm works. First, note that a botnet made of one user, besides making little sense in practice, is by definition non-identifiable, since we assumed that the characteristics of the messages at a single-user level do not reveal any special information. Now, at the beginning of the algorithm, user 1 is initially declared as a bot, namely, ˆB = {1}. Then, it is checked whether users 1 and 2 form a botnet. If so, Bˆ = {1, 2} is taken as the current botnet estimate. If not, Bˆ = {1} is retained. Then, it is checked whether the currently estimated botnet Bˆ forms a bot with user 3, and so on. At the end of the inner loop, the algorithm ends up with an estimate Bˆ . If the cardinality of the estimated set is greater than one, it is taken as a current estimate. The procedure is then restarted by choosing user 2 as initial pivot, and sequentially checking the remaining users as explained before. At the end of the inner loop, the algorithm ends up with another estimate Bˆ. If the cardinality of the estimated set is greater than one and greater than the cardinality of the previously estimated set8, then it is taken as a current estimate. Otherwise, the previous estimate is retained. The procedure ends when all users have been scanned as pivots.

From all this we can see that the Botbuster algorithm although effective is only useful for a particular subset of data and is not the entire range of data. Therefore we must not blindly use this algorithm for all cases and only for cases which fall under this category of data.

V. MODULES

A. Detection Approaches and Methods

Intrusion is an attempt to bypass or violate the security mechanisms of a system and an intrusion detection system uses enhanced processes to identify intrusions. Intrusion detection systems are designed to detect anomalous traffic in a network. To design an effective DDoS defence mechanism, the designers consider various security issues in a network. The main purpose of a detection system is to provide security to a system by detecting anomalous traffic that can disrupt system services.

1) Detection Approaches

In this section, we discuss four deployment points for DDoS defence mechanisms, viz., source end, and victim-end, intermediate and distributed [56].

a) Source-end

A source-end DDoS defence system is very effective in stopping attacks as close to the source as possible. It reduces network traffic congestion and saves network resources. Placing a defence system in the source network is better than placing it further

downstream. In this approach, network attack traffic can be stopped before reaching the target network, and it also reduces the chance of collision by filtering attack traffic before it aggregates with other attack traffic flow in the network

2) Victim-end

Most DDoS defence mechanisms are deployed at the victim-end for effective detection and defence of a system. Victim-end detection systems detect attacks either in a reactive or proactive manner.

B. Botnet Attack Detection: Approaches and Methods

For about a decade, botnets attacks have been growing rapidly on the Internet and this has created many security problems for defence mechanisms used by network administrators. Many solutions have been proposed to detect botnet attacks based on attack behaviour. People use traffic statistics, nature of communication protocols used, general analysis of network behaviour, graphical representations of behaviours, actions in honeypots, and collaborative feedback in large networks to detect botnet attacks. The majority of current DDoS attacks including mimicking attacks are performed by botnets, and it is possible to distinguish legitimate cyber behaviour from botnets attacks using different detection methods.

C. IRC-based Botnet Detection

Although IRC-based communication is fairly old, due to its simple command and control mechanism it is still used by bots. To detect botnets based on the IRC protocol, many methods have been proposed. Lu and Ghorbani [116] propose an algorithm to detect and characterize botnets in a large enterprise WiFi network. First, they apply the K-means clustering algorithm on payload signatures to classify network traffic into different applications. The IRC applications are then analyzed using temporal-frequent characteristics of flows to discriminate malicious IRC channels created by bots from legitimate IRC traffic.

D. IP Traceback in DDoS Prevention

1) Link Testing Schemes

In such a scheme, the victim tests each of its incoming links as a probable input link for DDoS traffic and contacts the upstream router which is closest to the victim. This router then interactively traces back to its upstream routers until it finds the source of any potential attack. This recursive procedure is performed on every upstream router to reach the original source. The main advantages of link test schemes are that they can reliably detect flooding attacks, the network overhead is low and the distribution is very efficient. However, the major drawback is that the scheme generates additional traffic and consumes large network resources. There are two types of link testing mechanisms: (i) input debugging and (ii) controlled flooding.

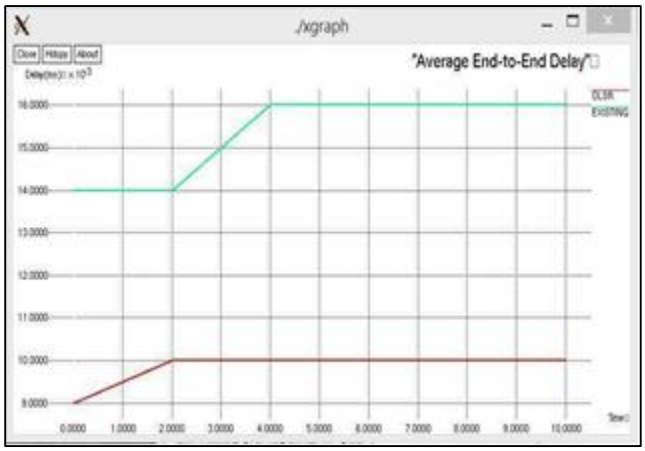
a) Input Debugging

In this mechanism, packets are filtered on every router at some egress ports to determine from which ingress ports they have arrived. During input debugging, the victim must recognize an attack first and then construct an attack signature that describes common features contained in attack packets. The victim then communicates with the upstream router to install input debugging filters on egress ports. Such a filter reveals the associated input ports and hence the upstream routers that originate the traffic. This process is repeated recursively until the originating source is reached. This mechanism is efficient in finding the true attack source because of its distributed nature. But, it has many cons such as high management cost, high network/router overhead, need for significant amount of time during communication to upstream routers in a large network and need for expert and skilled network operators, without whom traceback will be slow and impossible to complete.

b) Controlled Flooding

A mechanism is proposed by Burch and Cheswick [127] to test links by flooding them with large bursts of network traffic and then observing the effect from attackers. This traceback mechanism is an automatic process that does not require any support from network operators. It floods each incoming link on the router closest to the victim using a pre-generated map of Internet topology containing a few selected hosts. Any packets (including the packets sent by attackers) travelling across the loaded links must have high packet dropping probability. As a consequence, the victim can infer attack links from changes in the rate packets arrive from attackers. This basic procedure is used recursively on the upstream routers until the source is reached. This is a very skilful, practical and effective traceback mechanism. The main drawbacks of this mechanism are that it has high management overhead, needs coordination among routers or switches or even IPSs, and requires skilled network administrators. Packet marking schemes: Packet marking is one of the best ways to trace sources of flooding attacks. Routers mark forwarding packets either probabilistically or deterministically, with their own addresses. During an attack, the victim uses the marked information in the packet to traceback the attack source. Packet marking schemes are categorized into two classes, viz., probabilistic and deterministic.

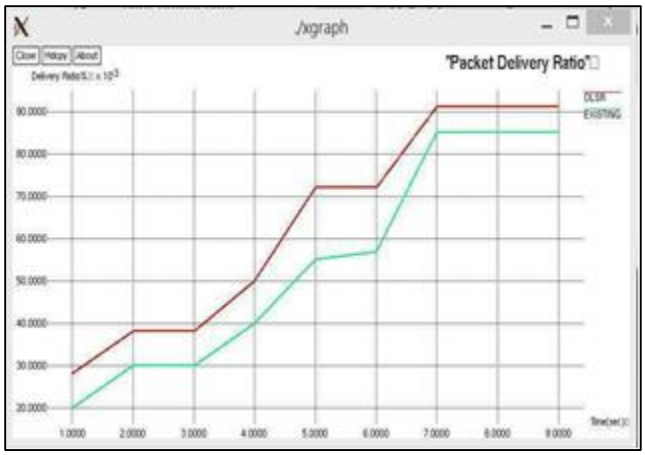
VI. OUTPUT ANALYSIS



Average end to end delay

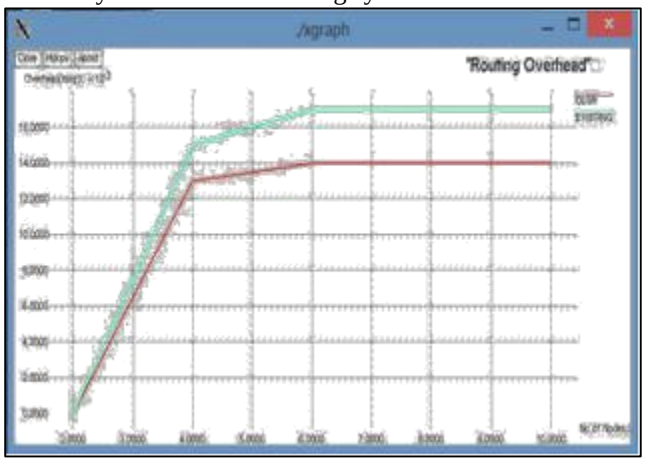
From the given figure we can come to the conclusion that the proposed algorithm functions as a much higher level by reducing the average end to end delay which is time taken for the packets to reach the receiver by the total time taken for all packets. The difference between the existing and proposed algorithm is very high and can be easily seen by the user.

Of all the factors which the current algorithm changes this is the only one which produces a large difference compared to the other factors.

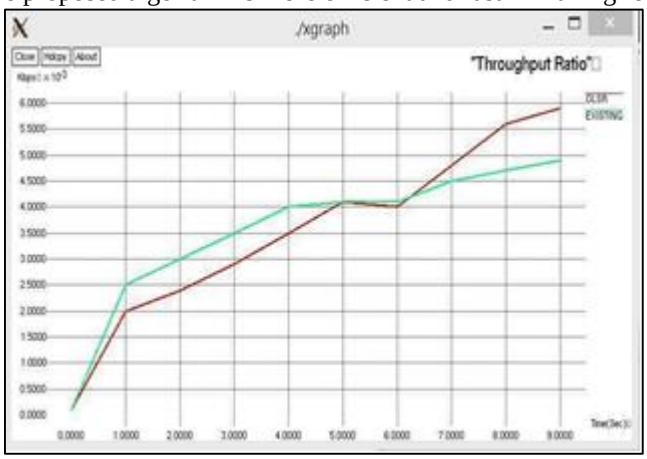


Packet Delivery Ratio

From the given graph we can see that the ratio of packets received by the destination to those generated by the sources for the existing system is slightly lower than the proposed system which although cannot be seen for a small subset we can see the change when we user a larger subset of data. The proposed algorithm although not providing a huge difference in the packet delivery we can see a more stable and steady ratio than the existing system.



The small packets which are used as routing packets usually produce a overhead as time passes and more and more nodes are present there we can see that the proposed algorithm is more efficient and fast in making routes.



Throughput Ratio

From the graph we can observe that even though initially the proposed algorithm functions poorer when compared to the existing system in the long run our algorithm beats the existing algorithm comfortably.

We can infer that for a small subset our algorithm will perform poorly where as in a subset of a reasonably large size our algorithm will function at a higher capacity than the existing one.

VII. CONCLUSION

We start this paper with a discussion of DDoS attacks and present a classification of DDoS attacks and characteristics of each class of DDoS attacks. We then introduce botnet technology, the primary facilitator of modern DDoS attacks, and recent trends in the launching of various types of DDoS attacks using botnets. A discussion of mobile botnets and traditional PC based botnets is given, followed by brief comparison between the two. We provide a detailed discussion of botnet-based DDoS defence approaches and methods. Though, in the past two decades, a good number of defence solutions have been introduced to counter DDoS attacks with increasing sophistication, still there are several important issues and research challenges which are open and yet to be addressed. Some of the prominent issues and research challenges are reported next to push the envelope further.

- Existing methods have been designed to be effective in detecting either low-rate or high-rate DDoS attacks, but usually not for both. So, developing a robust method that can detect both these types of attacks in real time remains a problem that needs attention.
- The performance of most methods is dependent on network conditions and their performance is also highly influenced by multiple user parameters. Hence, developing a defense solution free from these limitations as far as possible should be an important research initiative.
- Due to lack of unbiased evaluation frameworks, including benchmark datasets, it is difficult to properly evaluate the performance of the methods being developed. So, creating an unbiased framework for appropriate evaluation of a defense solution happens to be an important issue for investigation.

REFERENCES

[1] W. Stallings, Cryptography and Network Security: Principles and Prac-tice, 6th ed., Pearson, 2013.
[2] N. Hoque, D. Bhattacharyya, and J. Kalita, “Botnet in DDoS attacks:trends and challenges,” IEEE Commun. Surveys Tuts., vol. 17, no. 4, pp. 2242–2270, fourth quarter 2015.
[3] L. Feinstein, D. Schnackenberg, R. Balupari, and D. Kindred, “Statistical approaches to DDoS attack detection and response,” in Proc. DARPA Information Survivability Conference and Exposition, Washington, DC, USA, Apr. 2003, pp. 303–314.
[4] J. Yuan and K. Mills, “Monitoring the macroscopic effect of DDoS flooding attacks,” IEEE Trans. Depend. Secure Comput., vol. 2, no. 4, pp. 324–335, Oct. 2005.
[5] L. Li, J. Zhou, and N. Xiao, “DDoS attack detection algorithms based on entropy computing,” in Proc. ICICS 2007, Zhengzhou, China, Dec. 2007, pp. 452–466.
[6] Y. Xiang, K. Li, and W. Zhou, “Low-rate DDoS attacks detection and traceback by using new information metrics,” IEEE Trans. Inf. Forensics and Security, vol. 6, no. 2, pp. 426–437, Jun. 2011.
[7] J. Luo, X. Yang, J. Wang, J. Xu, J. Sun, and K. Long, “On a mathematical model for low-rate shrew DDoS,” IEEE Trans. Inf. Forensics and Security, vol. 9, no. 7, pp. 1069–1083, Jul. 2014.
[8] “Layer 7 DDoS.” http://blog.sucuri.net/2014/02/layer-7-ddos-blockinghttp- flood-attacks.html.

[9] "Taxonomy of DDoS attacks." http://www.riorey.com/types-of-ddosattacks/# attack-15.
[10] "Global DDoS threat landscape." https://www.incapsula.com/blog/ddosglobal-threat-landscape-report-q2-2015.html.
[11] S. Ferretti and V. Ghini, "Mitigation of random query string DoS via gossip," Commun. in Comput. and Inf. Sci., vol. 285, pp. 124–134, 2012.
[12] S. Marano, V. Matta, and L. Tong, "Distributed detection in the presence of Byzantine attacks," IEEE Trans. Signal Process., vol. 57, no. 1, pp. 16–29, Jan. 2009.
[13] S. Marano, V. Matta, and P. Willett, "Distributed detection with censoring sensors under physical layer secrecy," IEEE Trans. Signal Process., vol. 57, no. 5, pp. 1976–1986, May 2009.
[14] M. Barni and B. Tondi, "The source identification game: an information theoretic perspective," IEEE Trans. Inf. Forensics and Security, vol. 8, no. 3, pp. 450–463, Mar. 2013.
[15] B. Kailkhura, S. Brahma, B. Dulek, Y. S Han, and P. Varshney, "Distributed detection in tree networks: Byzantines and mitigation techniques," IEEE Trans. Inf. Forensics and Security, vol. 10, no. 7, pp. 1499–1512, Jul. 2015
[16] Yan Ou,and.Chanan Singh, (2002), "Assessment of Available Transfer Capability and Margins", IEEE transaction on Power systems, Vol.,17, No., 2. pp.463-68
[17] Zimmerman R, MATPOWER, A MATLAB Power system simulation package (version 3.0) Cornell University, New York.
[18] University of Washington Electrical Engineering, Power Systems Test Case Archive, 1993, Available from: http://www.ee.washington.edu/research/pstca

