DDoS Attack and Detection Using Hadoop

Project Report

Dixit Verma, Himanshu Jethwa

dv6cb@mst.edu, hj5y3@mst.edu

CPE 5420 Project – Prof. Egemen K. Çetinkaya

15 December 2016

Abstract

DDoS attacks have the destructive power to render servers inaccessible within a short period. These attacks have affected a wide range of services like banking, government websites, and even gaming industry. From the first documented attack in 1999 to the enormous 1TBPS DDoS attack against OVH hosting in 2016 [2], both the complexity and scale of these attacks have increased significantly over the years. The consequences of these attacks have escalated accordingly, ranging from inaccessibility of services to the legitimate users to the loss of financial income. The inability of standard DDoS defence mechanisms to identify and prevent such attacks have bolstered the need for new defence mechanisms. The purpose of this project is to simulate such attack and use big data techniques, specifically, Hadoop MapReduce algorithm to analyse the traffic pattern and identifying such attacks. The algorithm implemented is counter based method detection.

1. Introduction

Today most of the services like file sharing, video and audio streaming, VOIP, social media, banking need internet. DoS attack is an attempt to make the services unusable to the legitimate user. Since DoS attacks and Distributed DoS attacks tend to affect the availability of a system, the damage done can be huge in case of services like banking where every second, a transaction is recorded. The system which suffers from attack or whose services are attacked is called as “primary victim” and on other hand “secondary victim” is the system that is used to launch the attack.

In this project, we have used Hadoop map reduce framework to detect DoS traffic. We have implemented our project in 3 parts. First part sends DoS traffic to a web server, the second part analyses the DoS traffic and passes it on to the third part which is the prevention module. The prevention module uses filtering techniques to prevent a suspected attacker from sending further traffic.

2. Literature Survey

2.1.DDoS attack motivation

Attackers who perform DDoS attacks can be motivated by a lot of factors. Most of the time, notoriety is the primary motivation. The attackers aim to gain fame to become known in the hacking community. Politics motivates some hackers. For example, Anonymous believes that the Turkish government supports ISIS [1], so it launched DDoS attacks that crippled websites of Turkish government agencies and financial institutions. [4] New World Hackers launched a DDoS attack on U.S. presidential candidate Donald Trump's website because they opposed his political views.

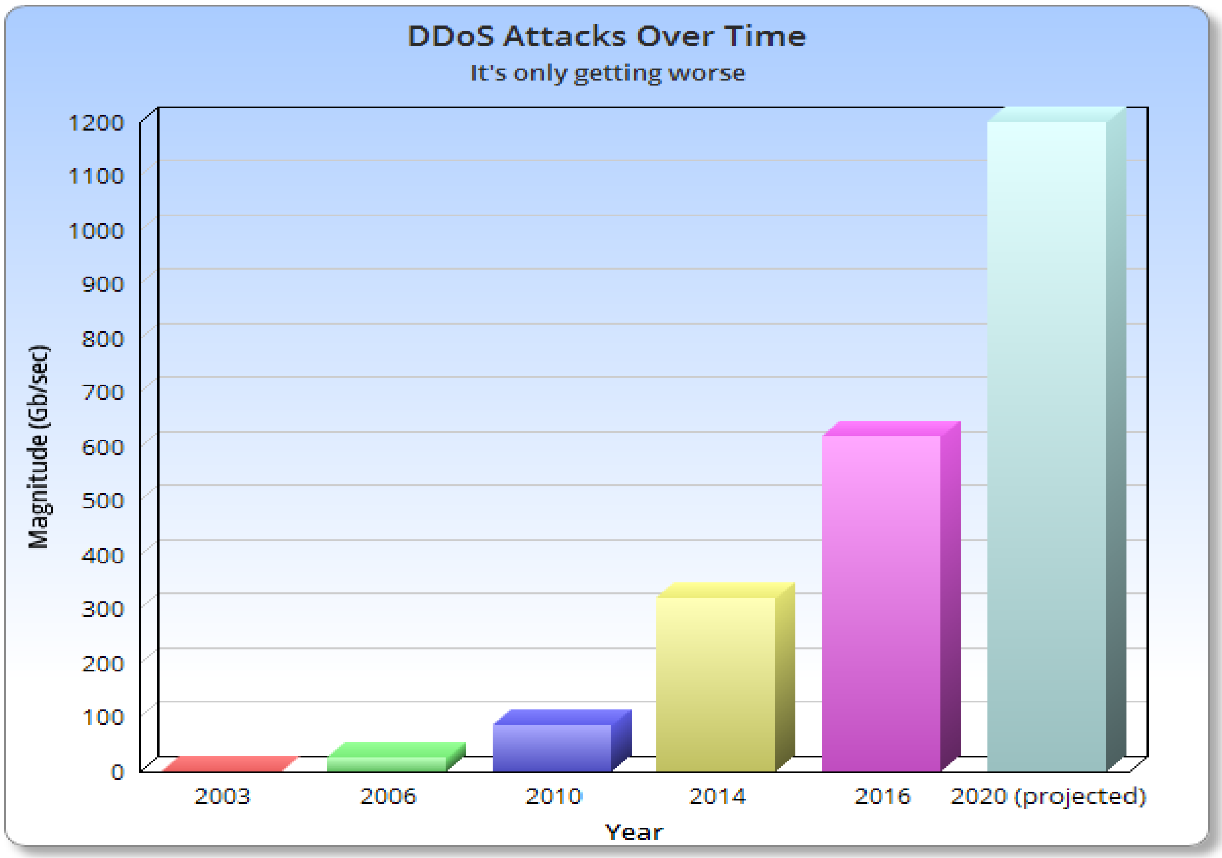
Some theories state that company websites under attack are being attacked by competing companies. This can disrupt or damage the target company's services, which may very well boost the sales or amount of website views for the competing company. There have also been cases where DDoS has been used against companies for ransom purposes. The attackers may attack first and disrupt the company's website until it is almost unusable or until it is down and demand ransom money in exchange for stopping the attacks. Since, a lot of devices are connected to the internet the risk of cyber-extortion is increasing. It could be as small as leaking your personal information to as large as hacking into a company's customer database.

2.2.History and Recent attacks

While it's hard to ascertain exactly when the first DDoS was launched, the first large-scale incident is said to have occurred in 1999 against the IRC server of the University of Minnesota. It left 227 systems affected, and the university's server was rendered unusable for days.

This event kicked off a series of similar attacks that put DDoS on the map as a viable tool for hackers and cybercriminals. In February 2000, some of the world's most popular Websites were toppled by DDoS attacks, and the compounded losses were enormous. Major traffic generators such as Yahoo!, eBay, CNN, Amazon.com and ZDNet suffered from assaults that paralyzed their systems and barred users from accessing their services for hours. DoS attacks have grown considerably since then and the graph depicts the frequency and magnitude of DoS attacks over the years.

In September 2016, one of the largest DDoS attack was reported against OVH hosting [5], a French service provided. The attack was reportedly a little above 1Tbps and the attackers used Mirai botnet to infect 145607 CCTV cameras/dvr and they sent >1.5Tbps DDoS traffic (tcp/ack, tcp/ack+psh, tcp/syn) towards OVH hosting servers. October 2016 saw another major DDoS attack which targeted a DNS service provider company called DYN [6]. The massive attack crashed the company's servers with requests more than 1Tbps. The company tried to mitigate the attack but major websites like Twitter, Netflix and Reddit were temporarily taken down. This led to the massive internet outage on the east coast.



3. DDOS Attack Toolkits

3.1 Trinoo

It uses TCP to communicate between attacker and control master program. The communication between the trinoo master and daemon is held using UDP packets. It implements UDP flood attack against victim. The master and daemons are password protected and prevent system administrators to take control of the trinoo network.

3.2 WIN TRINOO

This is a variant trinoo that works on Windows platform. It sends large amount of UDP packets to the victim as an action of attack.

3.3 MStream

The mstream program which is based on the “stream.c” attack, includes a “master controller” and a “zombie”. As the name indicates master controller controls all of the zombie agents. There is no encryption in the communications between the client, master, and zombie. An attacker connects to the master controller using Telnet to control the zombies. The zombie can slow a computer down by using up CPU cycles via a modified version of stream’s attack. The attack consumes network bandwidth when the target host tries to send TCP RST packets to non-existent IP addresses in addition to the incoming ACK packets which cause Routers to return ICMP host/network unreachable packets to the victim, consequential the starvation of bandwidth. This consumes large amount of network bandwidth and at the same time distributed method of attack multiplies the effect on the CPU.

3.4 Tribe Flood Network (TFN)

In this technique, a command line interface is used to communicate between attacker and control master program. The communication between the two is done through ICMP Echo reply packets. Following attacks are implemented through TFN's attack daemons: Smurf attack, SYN flooding, UDP flood and ICMP flood attack.

3.5 LOIC

Low Orbit Ion Cannon (LOIC) [8] is an open source network stress testing and denial-of-service attack application, written in C#. LOIC was initially developed by Praetox Technologies, but was later released into the public domain, and now is hosted on several open source platforms such as sourceforge.net.

4. Types of Attacks

DDoS attacks can be broadly divided into three types [12]:

4.1 Volume Based Attacks: Includes UDP floods, ICMP floods, and other spoofed-packet floods. The attack's goal is to saturate the bandwidth of the attacked site, and magnitude is measured in bits per second. Some examples are listed below.

- **UDP flood:** This DDoS attack leverages the User Datagram Protocol (UDP) and floods random ports on a remote host with numerous UDP packets, causing the host to repeatedly check for the application listening at that port, and (when no application is found) reply with an ICMP Destination Unreachable packet. Moreover, this process consumes host resources, and can ultimately lead to inaccessibility.
- **PING flood:** An ICMP flood overwhelms the target resource with ICMP Echo Request (ping) packets, generally sending packets as fast as possible without waiting for replies. SThis type of attack can consume both outgoing and incoming bandwidth, since the victim's servers will often attempt to respond with ICMP Echo Reply packets, resulting a significant overall system slowdown.

4.2 Protocol Attacks: Includes SYN floods, fragmented packet attacks, Ping of Death, Smurf DDoS and more. This type of attack consumes actual server resources, or those of intermediate communication equipment, such as firewalls and load balancers, and is measured in Packets per second. Some of the attacks are:

- **SYN flood:** A SYN flood DDoS attack exploits a known weakness in the TCP connection sequence (the "three-way handshake"), wherein a SYN request to initiate a TCP connection with a host must be answered by a SYN-ACK response from that host, and then confirmed by an ACK response from the source. In a SYN flood scenario, the requester sends multiple SYN requests, but either does not respond to the host's SYN-

ACK response, or sends the SYN requests from a spoofed IP address. Either way, the host system continues to wait for acknowledgement for each of the requests, binding resources until no new connections can be made, and ultimately resulting in denial of service.

- **Ping Flood:** A ping of death ("POD") attack involves the attacker sending multiple malicious pings to a computer. The maximum packet length of an IP packet (including header) is 65,535 bytes. However, the Data Link Layer usually poses limits to the maximum frame size - for example 1500 bytes over an Ethernet network. In this case, a large IP packet is split across multiple IP packets (known as fragments), and the recipient host reassembles the IP fragments into the complete packet. In a Ping of Death scenario, the recipient ends up with an IP packet which is larger than 65,535 bytes when reassembled. This can overflow memory buffers allocated for the packet, causing denial of service for legitimate packets.

4.3 Application Layer Attacks: These include GET floods, attacks that target Apache, Windows vulnerabilities and more. The attacks are comprised of seemingly legitimate and innocent requests, the goal of these attacks is to crash the web server, and the magnitude is measured in Requests per second. Application-layer attacks generally require a lot less packets and bandwidth to achieve the goal.

- **Http flood:** This attack forces the server or application to allocate the maximum resources possible in response to each single request. Thus, the attacker will generally aim to overwhelm the server or application with multiple requests that are each as processing-intensive as possible.

5. Detection Techniques

5.1 Detection by Timing

Detection can be done using two ways: passive and proactive. Passive detection is a form of detection which is done by analyzing the logs, after the attacker has finished this mission, the detection can be on time if the attack can be detected during the time of attack. Proactive detection is the detection of attack before it approaches the target machine or before the service is ruined

5.2 Detection by Activity

Based on detection activity the categorization is as follows.

a) Signature based—It involves priori knowledge of attack signatures. SNORT is the widely-used signature-based detection approach.

b) Anomaly based—It treats any incoming traffic that is violating the normal profile as an anomaly. For detecting DDoS attacks, it is first required to know the normal behavior of the host and then finding deviations from that behavior.

Limitation: The common challenge for all anomaly based intrusion detection systems is that it is difficult to consider the data that provides all types of normal traffic behavior. Thus, legitimate traffic can be classified as attack traffic which will result in a false positive. To reduce the false positive rate, many parameters are used to provide more accurate normal profiles, which may increase the computational overhead to detect attack.

c) Hybrid attack detection—

Hybrid attack detection has the optimistic features of both: 1) pattern-and 2) anomaly-based attack detection models to achieve high detection accuracy, low false positives and negatives, and increased level of cyber conviction. Even though hybrid attack detection approach decreases false positive rate, it also increases complexity and cost of implementation.

d) Third party detection—

Mechanisms that deploy third-party detection do not handle the detection process themselves but rely on an external third-party that signals the occurrence of the attack. Examples of mechanisms that use third-party detection are easily found among traceback mechanisms.

6 Detection Techniques using Big Data:

6.1 Hadoop Map Reduce Overview:

Hadoop MapReduce is a software framework for easily writing applications which process vast amounts of data in-parallel on large clusters of commodity hardware in a reliable, fault-tolerant manner.

A MapReduce *job* [9] usually splits the input data-set into independent chunks which are processed by the *map tasks* in a completely parallel manner. The framework sorts the outputs of the maps, which are then input to the *reduce tasks*. Typically, both the input and the output of the job are stored in a file-system. The framework takes care of scheduling tasks, monitoring them and re-executes the failed tasks.

Generally, the compute nodes and the storage nodes are the same, that is, the MapReduce framework and the Hadoop Distributed File System (see HDFS Architecture Guide) are running on the same set of nodes. This configuration allows the framework to effectively schedule tasks on the nodes where data is already present, resulting in very high aggregate bandwidth across the cluster.

The MapReduce framework operates exclusively on <key, value> [10] pairs, that is, the framework views the input to the job as a set of <key, value> pairs and produces a set of <key, value> pairs as the output of the job, conceivably of different types. The key and value classes have to be serializable by the framework and hence need to implement the Writable interface.

Additionally, the key classes have to implement the WritableComparableinterface to facilitate sorting by the framework.

Input and Output types of a MapReduce job [11]:

(input) <k1, v1> -> **map** -> <k2, v2> -> **combine** -> <k2, v2> -> **reduce** -> <k3, v3> (output)

6.1 Counter Based Method

Counter-based method Counter-based detection is a simple method that counts the total traffic volume or the number of web page requests. Since the DDoS attack with the low volume of traffic such as the HTTP GET incomplete attack is prevalent in these days, the frequency of page requests from clients will be a more effective factor.

This algorithm needs three input parameters of time interval, threshold and unbalance ratio, which can be loaded through the configuration property or the distributed cache mechanism of MapReduce. Time interval limits monitoring duration of the page request. Threshold indicates the permitted frequency of the page request to the server against the previous normal status, which determines whether the server should be alarmed or not. The unbalance ratio variable denotes the anomaly ratio of response per page request between a specific client and a server. This value is used for picking out attackers from the clients. In our MapReduce algorithm, the map function filters non-HTTP GET packets and generates key values of server IP address, masked timestamp, and client IP address.

The masked timestamp with time interval is used for counting the number of requests from a specific client to the specific URL within the same time duration. The reduce function summarizes the number of URL requests, page requests, and server responses between a client and a server. Finally, the algorithm aggregates values per server. When total requests for a specific server exceeds the threshold, the MapReduce job emits records whose response ratio against requests is greater than unbalance ratio, marking them as attackers. While this algorithm has the low computational complexity and could be easily converted to the MapReduce implementation, it needs a prerequisite to know the threshold value from historical monitoring data in advance.

6.2 Access pattern-based Method

The access pattern-based detection method assumes that clients infected by the same bot conduct the same behavior and that attackers could be differentiated from normal clients. This method requires more than two MapReduce jobs: the first job obtains access sequence to the web page between a client and a web server and calculates the spending time and the bytes count for each request of the URL; the second job hunts out infected hosts by comparing the access sequence and the spending time among clients trying to access the same server. The drawback of this approach is the highly computational complexity to spot the DDoS pattern.

7 Implementation

The project is implemented in three parts:

a) Attack Module
b) Detection Module
c) Prevention Module

Each of these modules are explained henceforth:

7.1 Attack Module:

Apache Tomcat web server is installed on a windows machine. Configuration for logging visiting clients' IP addresses is done. This means that when any client accesses any web page deployed on web server, the IP address of that client will be logged into a log file.

Attack module consists of two virtual machines installed on 2 hosts. Each module executes a code which continuously sends access requests to web server. Web server logs these requests. As the number of access requests keep on increasing, the web server is slowed down causing delay in providing service.

7.2 Detection Module:

The windows server deployed with web server delivers the log file, the size of which will grow exponentially as DDoS attack is initiated, periodically to a Hadoop cluster which will analyze the Log file using MapReduce framework as follows:

MapReduce Framework input: Log file, T set to 1000, Ti

Mapper:

Input – Line Number, Log Entry key value pairs

Output – IP Address, 1

Reducer:

Input – IP Address, 1

Output – IP Address, N key value pairs $\exists N > T$

Symbol	Meaning
T	Threshold value beyond which if an IP address is not allowed to access the web pages
Ti	Time interval required for analysis, concatenation of start time and end time

N	Number of times IP Address entry is there in the given time interval

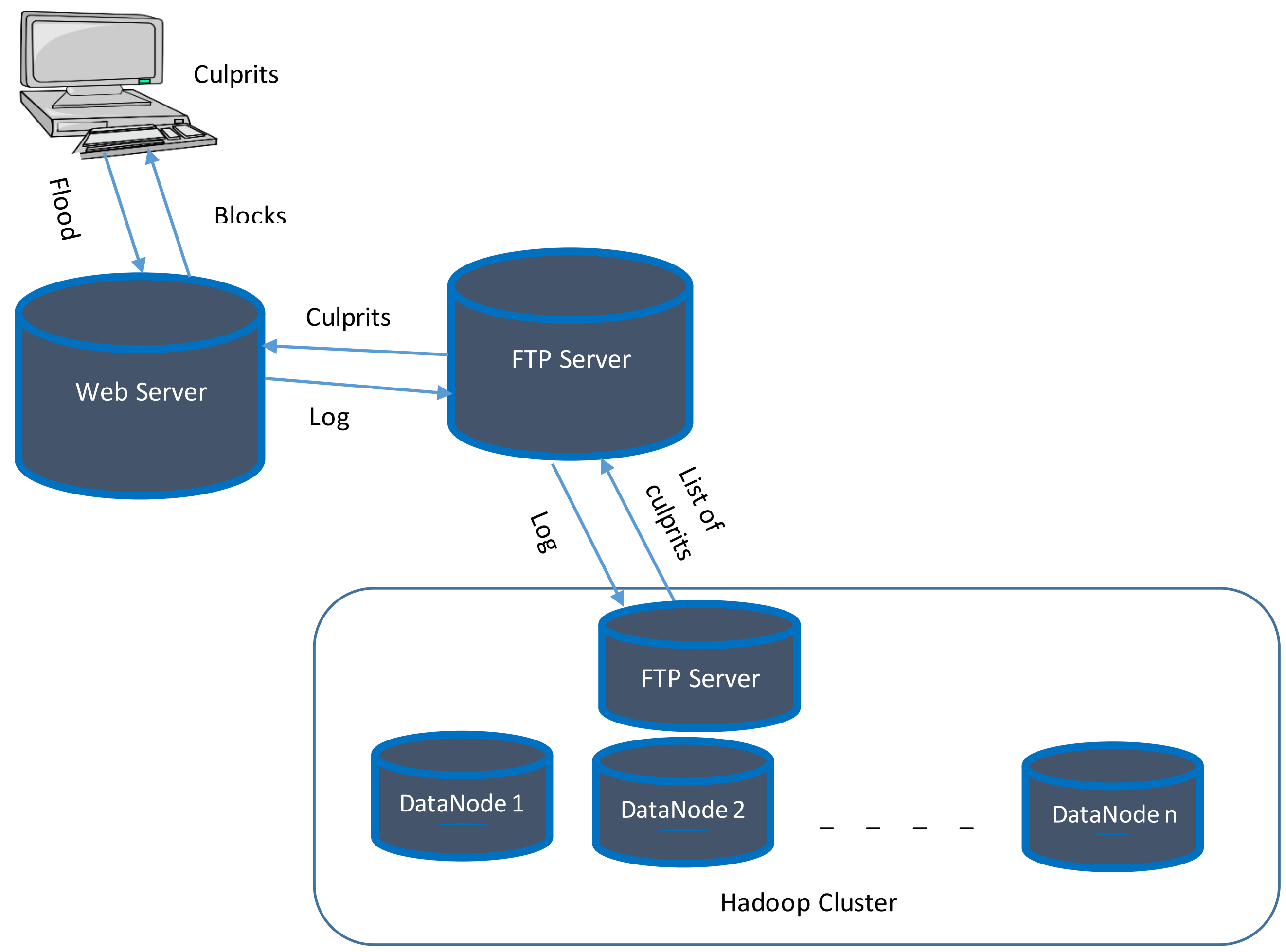
It is important to note that Hadoop cluster is deployed on a pseudo-distributed mode on a linux virtual machine.

7.3 Prevention Module:

After detection module evaluates the culprit IP Addresses, it sends it to the prevention module which is located on the windows server deployed with the web server.

The main function of this module is to change the configuration of web server in such a way that it denies any further requests from client with the culprit IP addresses, or in other words to block the culprits from accessing and hence preventing them from denying service.

System Architecture:



The working of our project can be explained in below steps:

- We have an apache server deployed on a virtual machine. We have used two other virtual machines to flood the apache server with http GET requests. LOIC tool and java code is used for this purpose.
- Then we extract the apache logs and pass them on to the Hadoop file system using a shell script.
- Once the log file is available in the Hadoop cluster we start Map reduce code which reads line by line and looks for the IP address which is sending requests which exceed a threshold value (suspected attacker).
- The code can detect multiple attackers and when detected, these IP addresses can be explicitly banned from sending further requests to the web server by modifying the configuration file.

8. Contribution:

Dixit	Himanshu
Attack using LOIC,java code	Code for mapper
Code for reducer, script for log file transfer	Shell script for Hadoop execution
Blocking the IP from apache server	Setting up Hadoop cluster and apache on VM

9. Conclusion

It is proved that Hadoop MapReduce framework that falls under big data analysis techniques can be used for detection of DDoS attack in its preliminary stages and hence can be used to prevent the same.

As a future work, the results obtained can be compared with result of traditional databases or other data analysis techniques. Some other big data techniques like mongoDB, pig, Mahout, etc. can also be used to analyze the same and eventually, the results of these techniques in regards to the delay in detection and other factors like cost of implementation and efficient use of resources especially for this use case can be done. Furthermore, a specification sheet might be made based on the correlation of requirements with the technique best suited for a website along with the infrastructure available.

10. References

[1] https://www.rt.com/news/326853-anonymous-war-turkey-isis

[2] Twitter: Octave Klaba CTO OVH hosting

[3] https://www.hackread.com/hackers-shut-down-donald-trump-election-campaign-website

[4] Mark Jeftovic, “Managing Mission Critical Domains and DNS” and https://www.arbornetworks.com/blog/asert/the-internet-goes-to-war

[5] http://securityaffairs.co/wordpress/51726/cyber-crime/ovh-hit-botnet-iot

[6] http://dyn.com/blog/dyn-statement-on-10212016-DDoS-attack

[7] http://thehackernews.com/2016/11/DDoS-attack-mirai-liberia

[8] https://en.wikipedia.org/wiki/Low_Orbit_Ion_Cannon

[9] Tripathi, S., Gupta, B., Almomani, A., Mishra, A. and Veluru, S., 2013. Hadoop based defense solution to handle distributed denial of service (DDoS) attacks. Journal of Information Security, 4(3), p.150.

[10] Dayama, R.S., Bhandare, A., Ganji, B. and Narayankar, V., 2015. Secured Network from Distributed DOS through HADOOP. International Journal of Computer Applications, 118(2).

[11] Lee, Y. and Lee, Y., 2011, December. Detecting DDoS attacks with hadoop. In Proceedings of The ACM CoNEXT Student Workshop (p. 7). ACM.

[12] www.incapsula.com