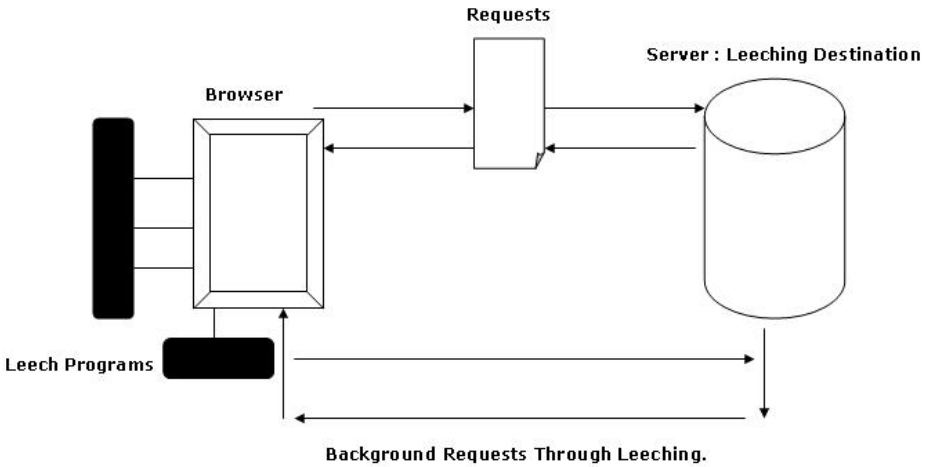
Paradox of Web Leeching

Aditya K Sood aka 0kn0ck
SecNiche Security
www.secniche.org

[1] Abstract: This paper delineates the anti circumventory paradox of Leeching. This matrix co-persists with the attackers and protectors. Leeching is derived from an animal Leech, that is a parasite which nourishes itself from other resources. The same case occurs when an attacker exploits technology constraints of other party thereby providing nothing but loss in one or the other sense. This concept encompasses web, system and network in whole. The process is really critical in its context. The concern is to protect the resources. Is it possible to eliminate them fully? It is not possible to completely break off its sustenance from the matrix though damage can be controlled. The prime point is to protect functional objects. The matrix works like that. Pre and post exploitation revolves around this matrix context. The leeching is always considered to be as a floating object in the computer security panorama. This object is created by evil minds. The other inherited object is Anti Leeching which works entirely opposite to the previous object. The cross matrix wars go on and on between these two objects .If one looks at the security personally then protection has to be implemented to de- stabilize the leeching attacks.

[2] Anatomy: Leech computing is a concept that is based on distributed functionality. Actually in technology terms, the leech computing relates to a hidden program on a client computer and user is not aware of it. The program remotely processes data and report the results without the intervention of user. It works automatically and does not affect the client system. The prime functional part of a designed leech program is that it does not access the hard drive of the client system but is memory driven. It means the leech program entirely runs in the memory of system. The concept ingrained here is that leech program becomes active during run time because of memory operations and not physically bounded to the system. So this makes the program constraint free and activation occurs only on the basis of design. It is considered to be as a distributed environment of working. How this is accomplished? The simple concept is data is downloaded from main server to client hard disk and result of the process is thrown back to main server. The work load can be disseminated into number of nodes i.e. clients that are participating in the transference of objects to process data. If you think then leeching software is running in backward context of system without any cross reference with the working objects with front end. It defines the versatility of leech program because all the relative work is done without tempering the state of system. When a number of client systems work together and processing power is shared it results in distributed environment. It can be considered as virtual concept of distributed computing but it practically applicable.



Generic Flow Model of Leeching Process

Let's analyze the parasitic computing. It comprises of an unauthorized use of any computer that is connected to the internet to process certain part of defined data. Remember network operations are timely reliable but require more processing power than system if overall structure of processing is analyzed. The processing power is much more than to send and receive data from the internet. One can understand clearly how critical the process is. This is because in network, the system has to first fetch data from frames and afterwards the data has to be unwrapped. Once the data is set and divided into stated benchmarks, than processing starts. But in case of raw data it has to be directly processed. So distinction can be made easily. The leech computing and parasitic computing share same base on the underlined reasons.

Data Processing without User Knowledge: The data is processed directly but without the knowledge of user, The user does not know that the system is being used by leech program and processing power is shared with other operation too. It is one of the common base of leech computing and parasitic computing.

No Software Installed: No specific software is installed on the physical hard drive of the system. It means the leech program is software independent as far as processing operations are concerned. The manipulation is done in the system itself. It is considered to be as second base for computing.

Stabilizing System: The system remains in its intact state. So no possible changes are made. It is considered to be as third prime base.

These are similarities between leech computing and parasitic computing. So that's why these are considered to be as two sides of same coin.

[4] Web2.0 Leeching Semantics Now its a time to understand how actually leeching works. It does not using any installed software, not affects system, so what it is. The leech concept revolves around browser panorama. The web browser is one of the prime working object in internet matrix. The web browser is main base for processing internet objects on the network. The browser composition does affect its working state and functional aspect to great extent. Previously only specific tags are used to design web pages. But with the advent of internet technology the scripting and applet objects have taken stride. The use of Active XObjects has increased a lot. It is a requirement to satisfy the need of ever increasing pace of internet technology. The web browsers support JavaScript and Java applets on a large scale to design Active X Objects for automated functioning. The browsers are dependent on it for better functioning .JavaScript is a prime component of website designing and object implementation. If JavaScript is not properly enabled the web browser does not display the contents of web page efficiently. This can even render the web page in a rogue way. The three main objects are underlined to overhaul the leeching process.

1. JavaScript Programs.
2. Active X Objects.
3. Java applets.

The JavaScript have enhanced the context of web page functioning. but at the same time have increased the security risk. You will get an alert box when you have to work with objects like Active X or java objects. This is done to provide stringent security and is applicable only after user consent. But by default JavaScript is enabled. You rarely find any alert box for JavaScript checking. Usually a option is provided in web sever properties but in most of cases it is enabled internally. With the rise of 2.0 the AJAX is on high. Let's get through some of the functional part of AJAX if we are considering an attack scenario. The underlined factors result in feature as well as exploitation through browsers.

a) *Execution Speed:* The real advantage of JavaScript data in the Ajax world is speed. It can take up to half a second to parse through XML on a reasonably fast machine, while the equivalent amount of data encoded as JavaScript is evaluated into arrays and associative arrays in just a few milliseconds.

b) Generating Syndicate Feeds in One Shot: The great part about using RSS for your Ajax data is that you can do things in one action. Not only do you get the data to your JavaScript code, but you also create a syndication feed that people can subscribe to using their RSS readers.

c) Restricting Domains: The domain functionality can be restricted. If your page comes from www.meta.com, your script cannot request data from www.google.com. You can only make requests to www.meta.com, which means that creating an RSS reader that resides just on the client is impossible till some explicit mechanism is used to fetch data in outbound manner from the source.

d) In core Parsing: It means to parse the JavaScript code returned from source file further to XML. You simply run the EVAL function on the code and take the returned value, which is an array of hash tables, then walk through them using a standard JavaScript for loop.

e) Javascript Module Tracing:
It's possible to trace JavaScript functions.

- javascript:eval
- javascript:alert
- javascript iframe | frame
- javascript:escape
- javascript:unescape
- <script></script>

f) Attack Simulation: The advantage of a frame is that you can use it in the ever-diminishing *number* of browsers that don't support the XMLHttp object. You can also use a frame to add items to the user's page history so that the back button actually works on your Ajax page. Unfortunately, the disadvantages are numerous. Using frames for transport works easily only for transmitting HTML, although with some hacking you can get JavaScript code and XML across the wire, as well. With the IFRAME approach, you can get data to the server in either of two ways. The first way is through the URL arguments associated with the SRC attribute on the < IFRAME > tag. The second way is to create a <form> tag with associated <input> elements inside the IFRAME document, then use submit () method on the <form> tag to POST or GET data to the server.

g) *Accessing Web Server Realm:* The XMLHttpRequest object provides two properties that provide access to the server response. The first property, responseText, simply provides the response as a string. The second property, responseXML, provides the response as an XML object. Retrieving the response as simple text is fine for simple use cases, such as when the response is displayed in an alert box or the response is a simple one-word phrase indicating success or failure.

h) *Browser Security:* Browser-based technologies wouldn't be complete without mentioning security. The XMLHttpRequest object is subjected to the browsers security sandbox. Any resources requested by the XMLHttpRequest object must reside within the same domain from which the calling script originated. This security restriction prevents the XMLHttpRequest object from requesting resources outside the domain from which the script was originally served.

The above stated factors can be considered as positive or negative aspects of JavaScript in this world of Web 2.0. The exploitation can be done based on these factors. When the leeching process is undertaken or leech program is designed then we are laying stress mainly on JavaScript infection and manipulation through it.

One more thing to be discussed and that is most of the folks have a knowledge stature in mind about Java and JavaScript is same. This is not so because one is active scripting language and has potential functionality entirely different from Java. The deigning of web applets are done through Java and other related programming.

[5] Drained Facts about Leech: Let's enumerate some of the practical facts of leech to grasp the functional and operational functionality of it.

- It is not considered to be as a virus or Trojan because it does not affect the system state or properties. It does not spread from computer to computer but only transmitted from server to client.

- It's not even considered to be as a spy ware because it does not possess any characteristics to be like that. It only processes data and reply back to server. Basically, it resides in a unique web page. If it is a spy ware then it must affect the other web pages in the browser.

- It's not specific to windows or Linux because JavaScript is platform independent if functionality is to be cross referenced. It means it can run on any platform provided it supports JavaScript. The operational part is defined over browser and platform.

- It becomes very hard for firewalls to trigger any alarm if leech is undertaken. It means its hard to check when data is being sent over the internet. This is because the leech is running inside your browser and it is well connected with the outside world. So access is there.

- HTTP proxy may be useful to some extent in combating leeching attacks. It becomes hard when the JavaScript is obfuscated because the designed filter is unable to match the object signature to set a constraint on its functioning.

- The malware scanner can catch leech objects if signatures are updated properly. Like the Active X Objects, Java applets etc can be checked against execution on system by the scanner. It becomes hard if signatures are not upgraded in scanner.

- Javascript progarms i.e. Leech can be throttled by applying SetTimeout function in code in the browser. It applies the time limit for which the leech performs its work.

[6] Conclusion

The leeching mechanism holds complexity as well as flexibility in the context in which it is implemented. The process can be used in a versatile manner. It depends on the usage. Since interoperability is becoming a grave factor in designing composite networks. But at the same time things should be taken into account regarding induction of new technology and the adverse affects. Anti leeching programs are running high now a days in order to dethrone attacks. The paradox is hard to deny. Protection and implementation best suited together.

[7] Anti Leeching Products and References:

[7.1] IIS Anti Leeching Sniffer Dog by VersalSoft [http://anti-leech.versalsoft.com/].
[7.2] Anti Leech Measure by AntiLeech.com [http://www.anti-leech.com/index.php?go=products]
[7.3] Anti Leech Archives [http://anti-leech-files.qarchive.org/]
[7.4] Anti Leech Project [http://www.antileech.net/]