Black Vectors of Web Exploitation

[Craniological Dissection of Web 2.0 Attacks.]

Analysis Through Live Case Studies.

Aditya K Sood aka 0kn0ck
Sec Niche Security
http://www.secniche.org

OWASP

6 September 2007

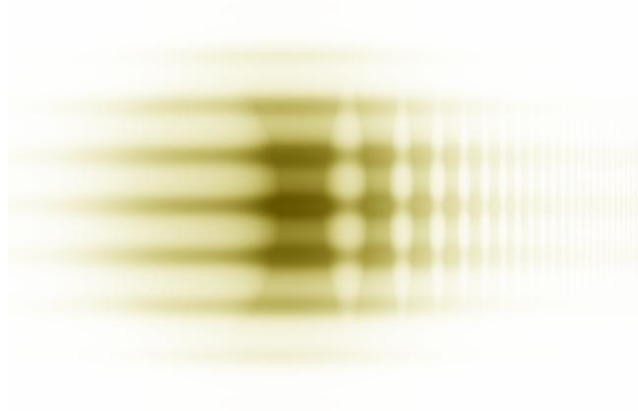
The OWASP Foundation

http://www.owasp.org

[] Who Am I ?

- Independent Security Researcher.
- Founder , Sec Niche Security. [http://www.secniche.org]
- Active Speaker : CERT-IN , XFocus-XCON (China) etc.
- IS Author Hakin9 , Hakin9 Linux+ etc. Authored Number of Security Related Papers. Research Featured as Global Security Perspective at FIRST.
- Released Advisories : Yahoo , AOL , MSN , Google , Verisign , Microsoft etc.
- Projects:
- M-Labs : Digital Intelligence [http://mlabs.secniche.org]
- CERA : Web Application Analysis. [http://cera.secniche.org]
- Trio Sec : An Active Penetration Testing Arena.[http://triosec.org]

[] Traversing Through Talk.



- Technology Variance.
- Application Bug Anatomy.
- Live Case Studies.
  - Double Trapping Injections.
  - Untamed Phishing.
  - Simulated Web Third Party Attacks.

Summary ? Technology Variance.

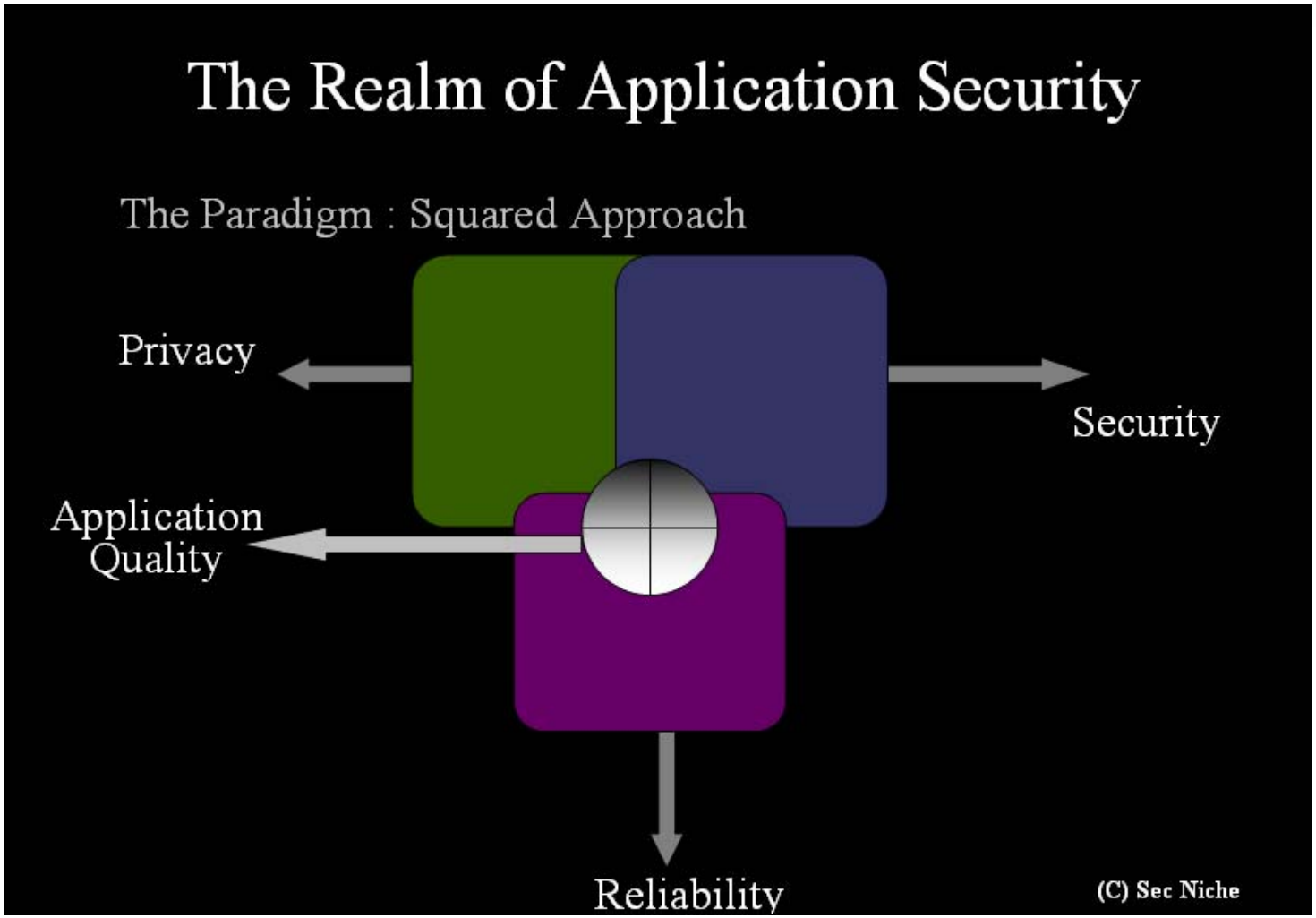
[] Technology Variance !

- Flourishing of Development Matrix.
- Innovation for more Efficient Products.
- Cross Platform Interdependencies.
- Paradigm of Exploitation.
- Workflow and Custom Deadlines.
- Survivability with Ever Changing Requirements.
- Cross Referenced Matrix of Technologies.

[] Shifting Vector Towards Web

- System Bugs are Hard to Exploit.
- Internally Structured Protection Mechanisms.
- Enhanced Security Features To Dethrone System Bugs.
- Organizations are on *RED ALERT.*
- Web : The Hottest Place of Attackers.
- Web Application : An Easy Interface To Exploitation.
- Interconnection Through Web. Diversified Infection Vector.

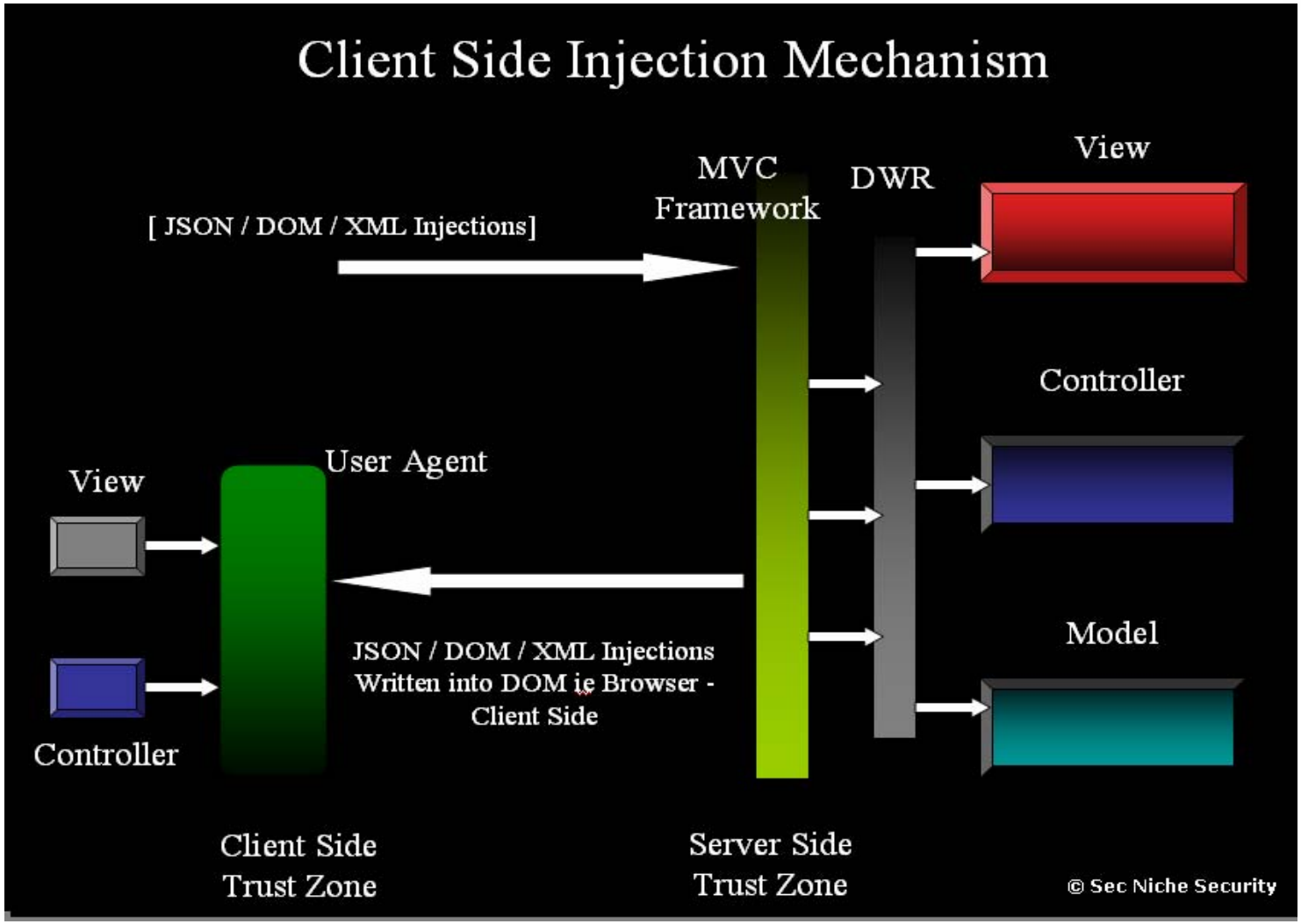
[] Squared Approach : Application Security



[] Application Bug Anatomy

- Scratching the Cause of Bugs.
- Too Many Eye Balls Misses The Point All Together.
- The Application Development Matrix.
- The Exploitation Vector.

[] Injection Mechanism



[] Case Studies : Vulnerabilities

1. *Double Trap Injections [One Step Ahead]*

    1.1 Case Study of a Company.

    1.2 Learning Through Hard Knocking.

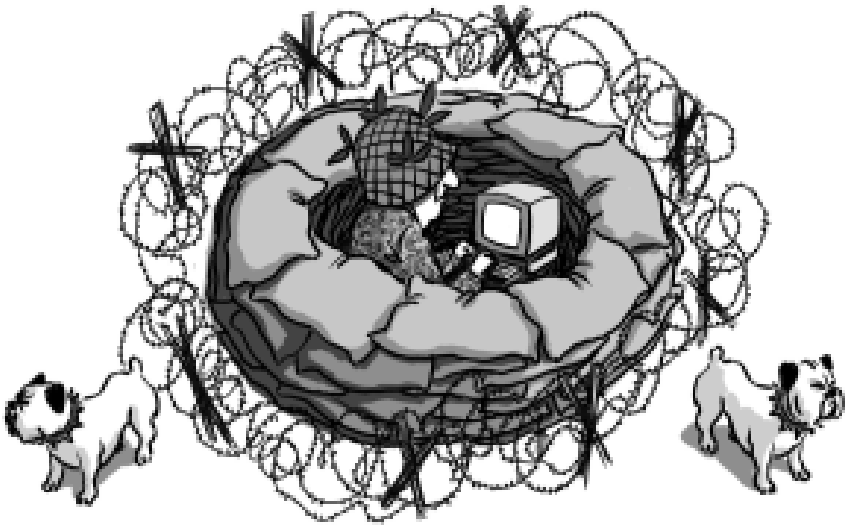
2. *Untamed Phishing [Digging Deeper]*

    2.1 Yahoo Search Engine Vulnerability.

    2.2 Yahoo Network Redirection and Phishing Vulnerability.

    2.3 Verisign Phishing Stringent Cases.

[] Case Studies : Double Trap XSS



[] Double Trap Injections : Core

1. *URL Banging*

   Injecting input parameters in the uniform resource locator's.

2. *Form Splitting*

   Injecting false arguments in the form values.

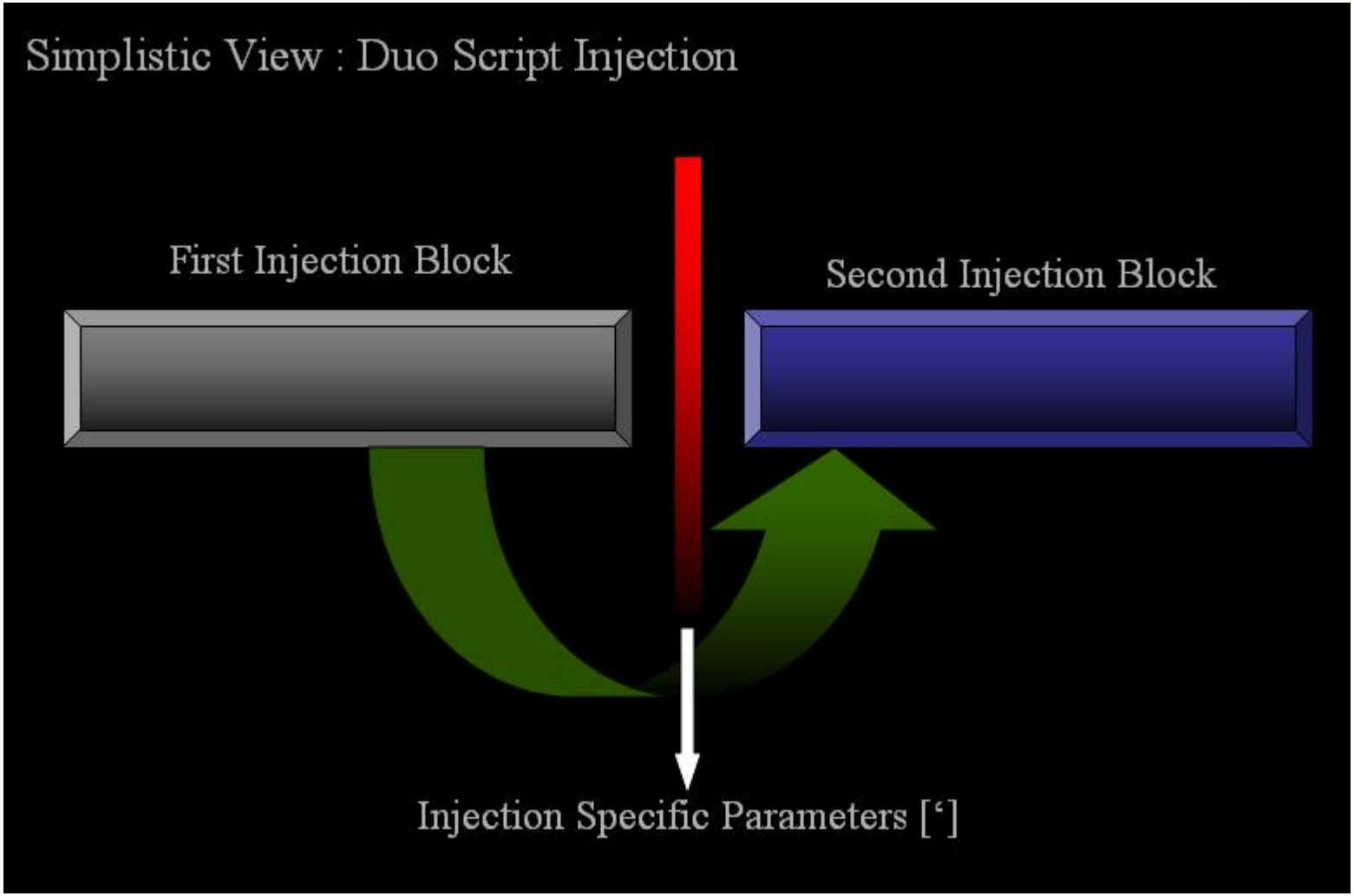
3. Input Validation points.

4. Security Loopholes Intrinsic Structure.

5. Insecure Coding.

6. Use of Obsolete Scripts.

[] Double Trap Injections : Model



[] Double Trap Injections : Live Case Stud

1. Attack Simulated in Security Consultation Website.
2. The Cause : Use of Obsolete Scripts.
3. Global Exploitation of Variables.

The Injection View :

[] Double Trap Injections : Live Case Study

```
<p>Please fill in the following fields:</p><FORM ACTION="contact.cgi"
METHOD="POST"><p><div id="contact" align="right">Your name:<INPUT
TYPE='TEXT' NAME="login" style="width:300px" MAXLENGTH=100
value="'&lt;script&gt;alert(&quot;XSS Says : Let Me IN!");</script>"></div>
```

```
<input type="TEXT" value="">
```

Attack Undertaken :

```
'<script>alert("XSS Says : Let Me In");</script>'"<h1>XSS : I am In!</h1>
```

Single Trap XSS

Double Trap XSS

[] Double Trap Injections : Live Case Study

References:

1. http://ha.ckers.org/blog/20070316/forgetting-global-replace-xss-woes/
2. http://cera.secniche.org/dbltrap.xhtml

Detailed Papers :

[Double Trap]

http://www.xssed.com/article/3/Paper_Double_Trap_XSS_Injection_An_Analysis/

[End Points Malfeasance]

http://sla.ckers.org/forum/read.php?6,8680

[] Yahoo Search Engine Flaw : Live Case Study

[] Yahoo Search Engine Flaw : Explanation

1. Vulnerability persisted in Yahoo Search Engine.
2. Links can be used by Phishers for Malicious Attacks.
3. Possible Cause : Handling of Redirection Variables.
4. An Ingrained Flaw.

The Vulnerable Link :

*http://rds.yahoo.com/_ylt=A0geu4qjI3hGYOEAIjJXNyoA/SIG=14oi6m38j/EXP=1182364963/**http%3A//search.yahoo.com/search%3Fp=Hacking%26y=Search%26rd=r1%26meta=vc%253Din%26fr=yfp-t-501%26fp_ip=IN%26xargs=0%26pstart=1%26b=11*

[] Yahoo Search Engine Flaw : Explanation

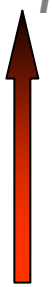
YAHOO! SEARCH

Persistent Link :

*http://rds.yahoo.com/_ylt=A0geu4qjI3hGYOEAIjJXNyoA/SIG=14oi6m38j/EXP=1182364963/**http%3A//%5B [Phishing Website]*

Exploited Link :

*http://rds.yahoo.com/_ylt=A0geu4qjI3hGYOEAIjJXNyoA/SIG=14oi6m38j/EXP=1182363/**http%3A//www.metasploit.com*



The Traffic is Redirected to the desired Link.

[] Yahoo Network Flaw : Explanation

YAHOO! SEARCH

1. The specific URL linked to any further yahoo website can be manipulated by the attacker to redirect the traffic and used for phishing.

2. The critical point is the URL can be called by third party for phishing.

Vulnerable Links :

*https://us.ard.yahoo.com/SIG=12gb00bbf/M=341232.9804850.11489914.6055752/D=regst/S=150001465:R2/Y=YAHOO/EXP=1182284104/A=4651436/R=0/SIG=1255of0p5/*http://help.yahoo.com/l/us/yahoo/mail/yahoomail/tools/tools-08.html*

[] Yahoo Network Flaw : Explanation

YAHOO! SEARCH

The Website Network Links:

https://us.ard.yahoo.com/SIG=12gb00bbf/M=341232.9804850.11489914.6055752/D=regst/S=150001465:R2/Y=YAHOO/EXP=1182284104/A=4651436/R=0/SIG=1255of0p5/ [Website Link]

The Manipulated URL's

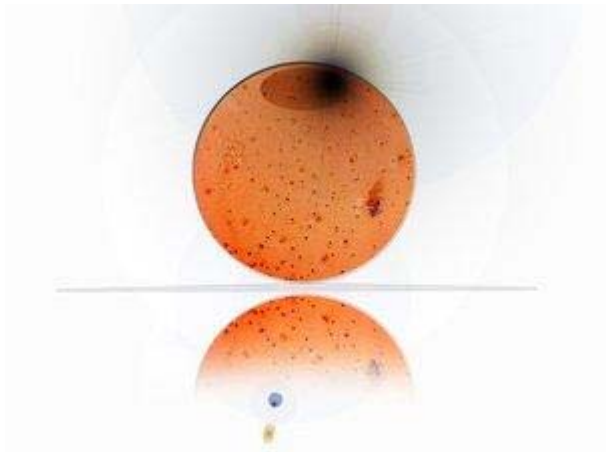
https://us.ard.yahoo.com/SIG=12gb00bbf/M=341232.9804850.11489914.6055752/D=regst/S=150001465:R2/Y=YAHOO/EXP=1182284104/A=4651436/R=0/SIG=1255of0p5/*http://www.google.com

https://us.ard.yahoo.com/SIG=12gb00bbf/M=341232.9804850.11489914.6055752/D=regst/S=150001465:R2/Y=YAHOO/EXP=1182284104/A=4651436/R=0/SIG=1255of0p5/*http://www.hushmail.com

[] Yahoo Flaws : Response

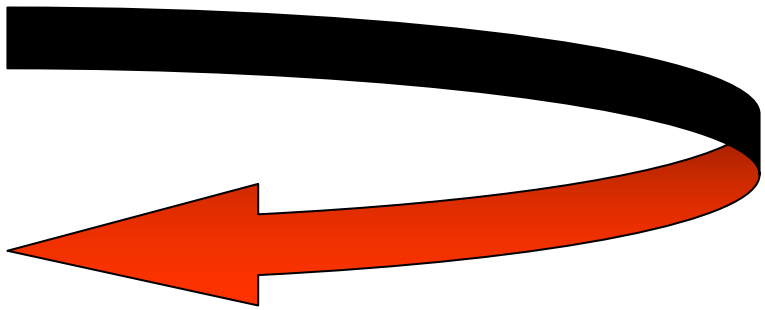
YAHOO! SEARCH

1. Website Vulnerability was Patched in 24 hours.
2. Search Engine Patch is in Development Stage.



[] Verisign Network Flaw

VeriSign™
The Internet Trust Company℠



[] Verisign Network Flaw : Explanation

The Verisign Secured Network and Verisign Weblogs network is vulnerable to phishing attacks. The problem persists in the redirection links present which allows third party redirection.

The cause :

1. Redirection of traffic directly without visiting website.
2. The website wont check the link that is being called by the phisher.
3. Third party linking is possible.
4. Looping attack is also possible.

Vulnerable Links :

http://www.verisignsecured.com/Redirect.aspx?%5B [Website Name]

http://www.weblogs.com/clickthru?url=%5B [Website Name]

[] Verisign Network Flaw : Explanation

Attack Examples :

[Third Party SQL Injection Check]

http://www.weblogs.com/clickthru?url=http://www.unep.org/Documents.Multilingual/Default.asp?DocumentID= [Injection Parameter]

[Multiple Redirections]

http://www.verisignsecured.com/Redirect.aspx?http://www.weblogs.com/clickthru?url=http://www.weblogs.com/clickthru?url=http://www.weblogs.com/clickthru?url=http://www.google.com

[Blind SQL Check]

http://www.verisignsecured.com/Redirect.aspx?http://www.weblogs.com/clickthru?url=http://www.weblogs.com/clickthru?url=http://www.pewinternet.org/report_display.asp?r=

[] Google URL Flaw

Google™ India

[] Google URL Flaw : Overall

1. Time to Time Google is Vulnerable To Phishing and XSS Attacks.
2. The Cause : Chaining of Ever Changing Technology.

[Google Redirection Flaw]

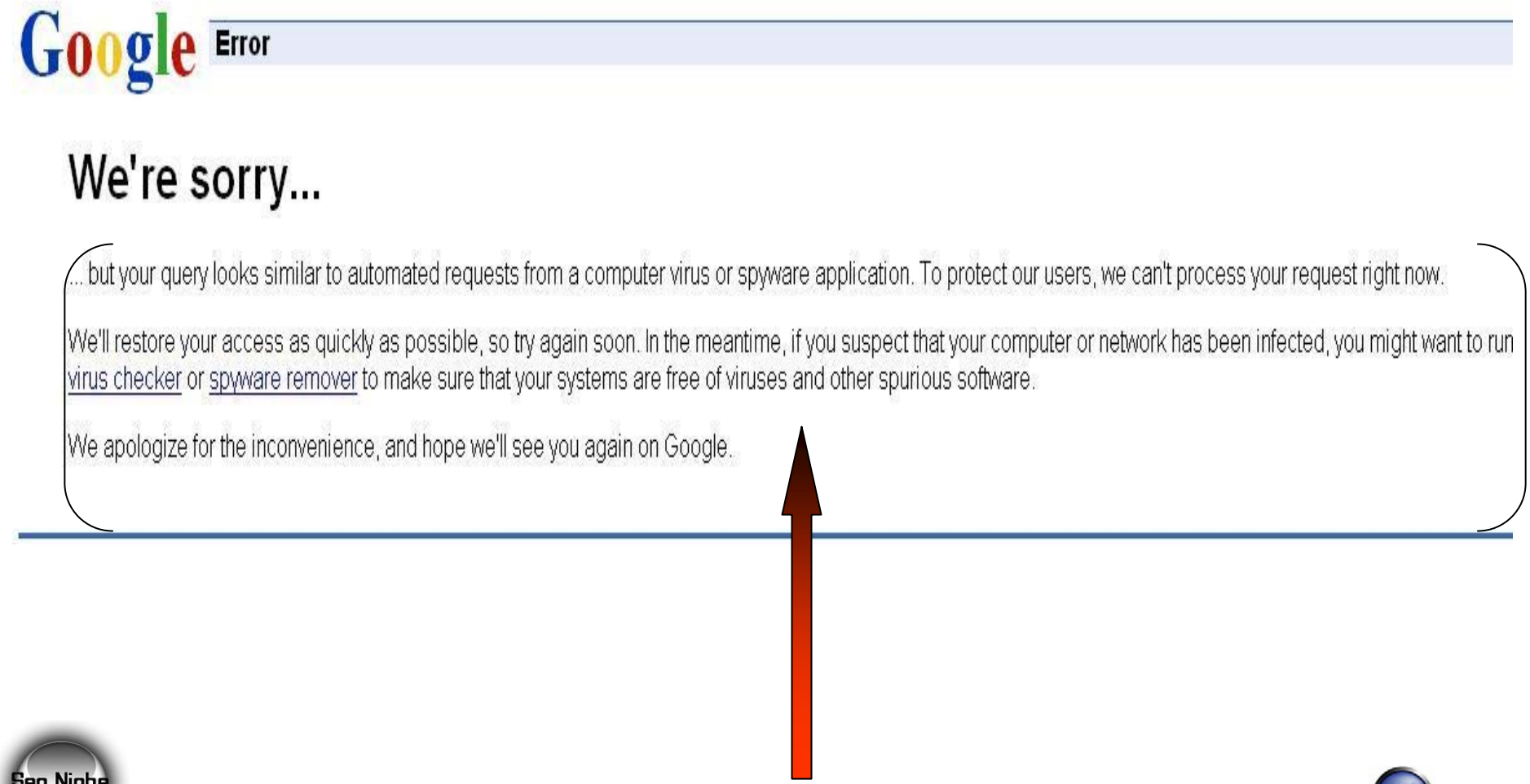
http://www.google.com/url?q= < Website Link> /&sa=D&sntz=1&usg=1

The Link : http://www.google.com/url?q=

The Link is undertaken and applied as such.

From Previous Time This Link has shown discrepancies a lot.

[] Google Search Error ?????????



[] Case Studies : References

http://www.secniche.org/advisory/YahooSearchPhishing_Vul.pdf

http://www.secniche.org/advisory/YahooNetPhishing_Vul.pdf

http://www.secniche.org/advisory/Verisign_Phish_Red_Vul.pdf

http://www.spamfighter.com/News-8704-Two-Critical-Flaws-Found-in-Yahoo.htm

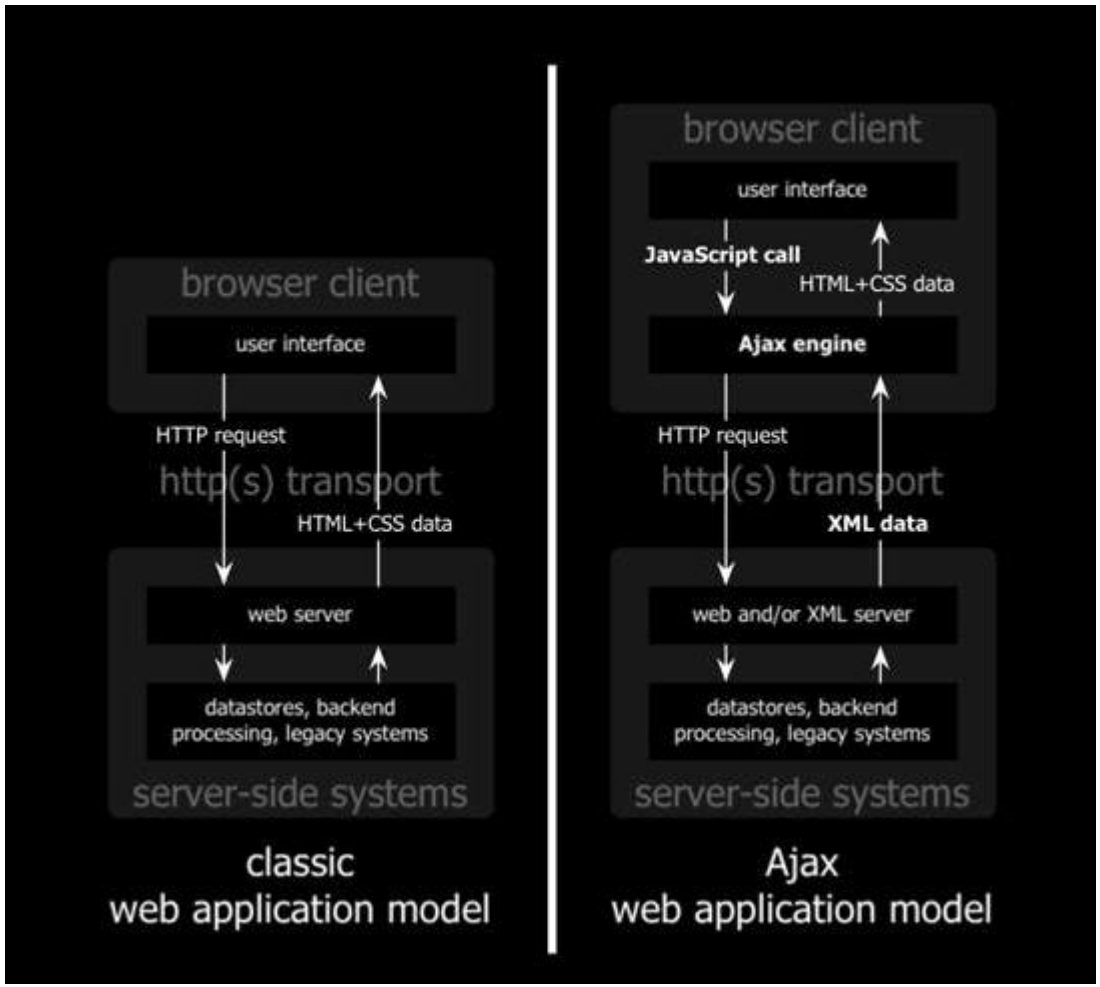
http://spamnews.com/Newsflashes/Newsflash/Two_Critical_Flaws_Found_in_Yahoo_200707116933.html

http://www.internetnews.com/security/article.php/3685131

http://article.wn.com/view/2007/06/23/Yahoo_Moves_Quickly_To_Plug_Phishing_Hole/

[] Digging Deeper : Web 2.0 Attacks

[] The Shift Towards Web 2.0



[] The Overall Structure : Web 2.0

[] The Shifting Points

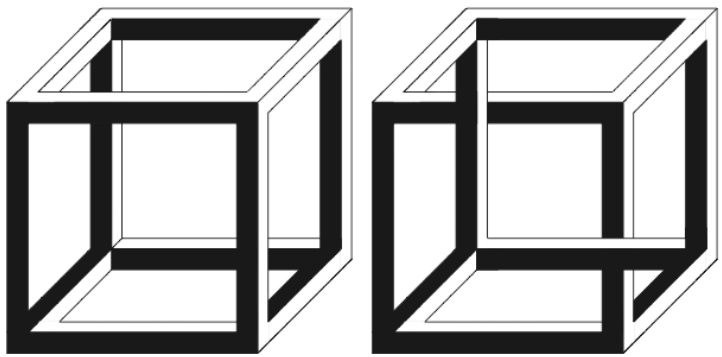
1. Iframes Subtle Functioning and Implementation for Asynchronous Requests.
2. The XHR [XmlHttpRequest] on Run.
3. Extensibility Behavior of XHR over Iframe.
4. Cross Domain Calls are not processed Directly in XHR.
5. Fusion of JSON-BISON , JDOM in AJAX. Simulated Code.
6. Injections [JSON,DOM] Through Data Serialization.
7. Asynchronous Implementation With an Ease through XHR.
8. The Endpoints Consideration is always Javascript Code [XHR+JSON]

[] Web 2.0 Most Favorable Attacks.

1. Incore XSS Attacks.
2. Cross Site Request Forging Attacks

    2.1 Direct Simulation.
    2.2 Indirect Simulation by Fusing Proxy.

3. The Serialization Stringent Attack Anatomy.
4. Denial of Service Attacks Through URL Concatenation.



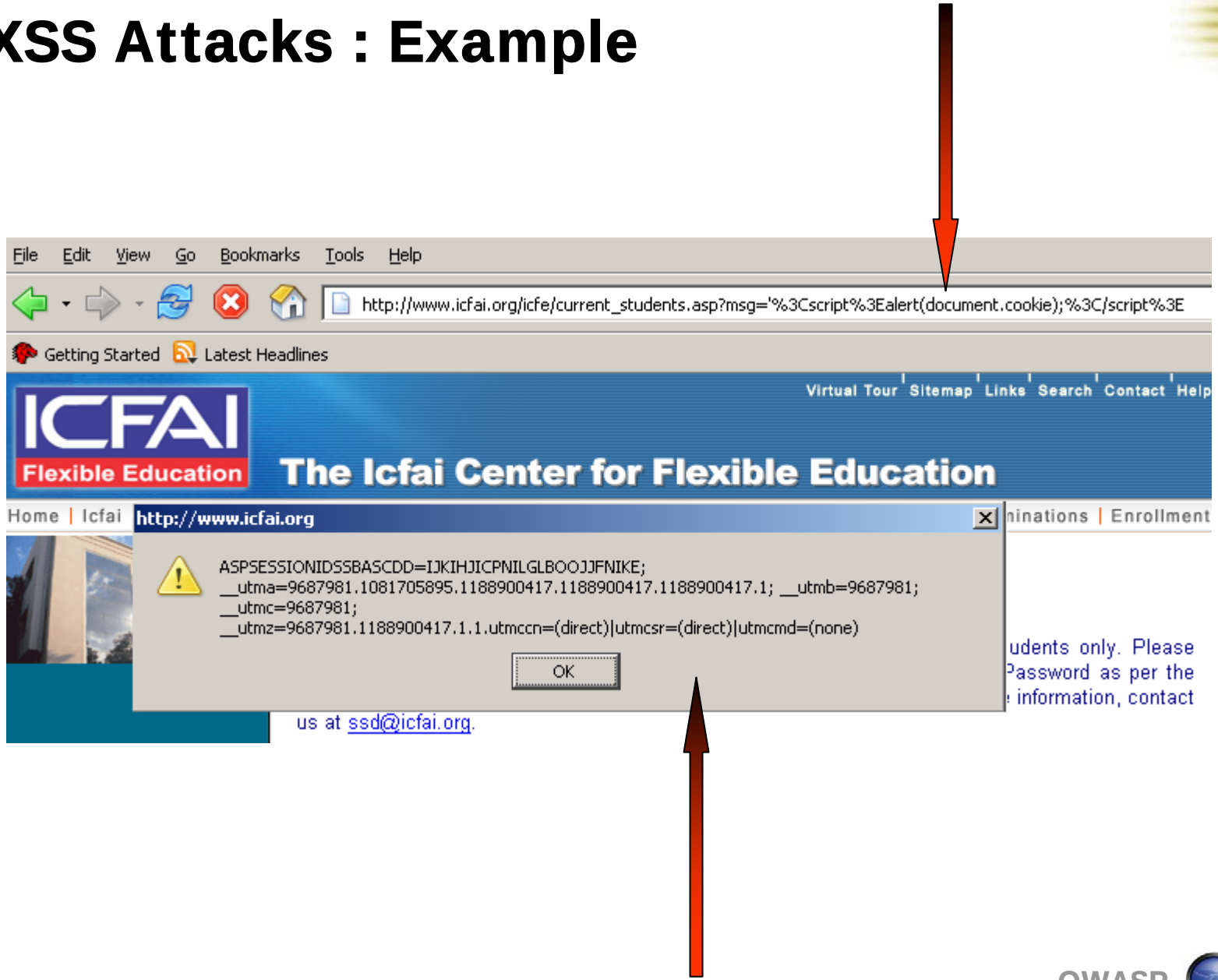
[] XSS Attacks.

1. Cross Site Scripting Attacks are High.
2. Injected Parameter Processed by Server Renders the DOM on the Client Side to Cause an Injection.
3. Javascript Simulation in Dynamic Code.
4. Injections Use :

    4.1 <script>alert(“XSS”);</script>
    4.2 document.cookie , document.domain etc.
    4.3 <img src=“malicious.jpg”></img>
    4.4 Eval etc.

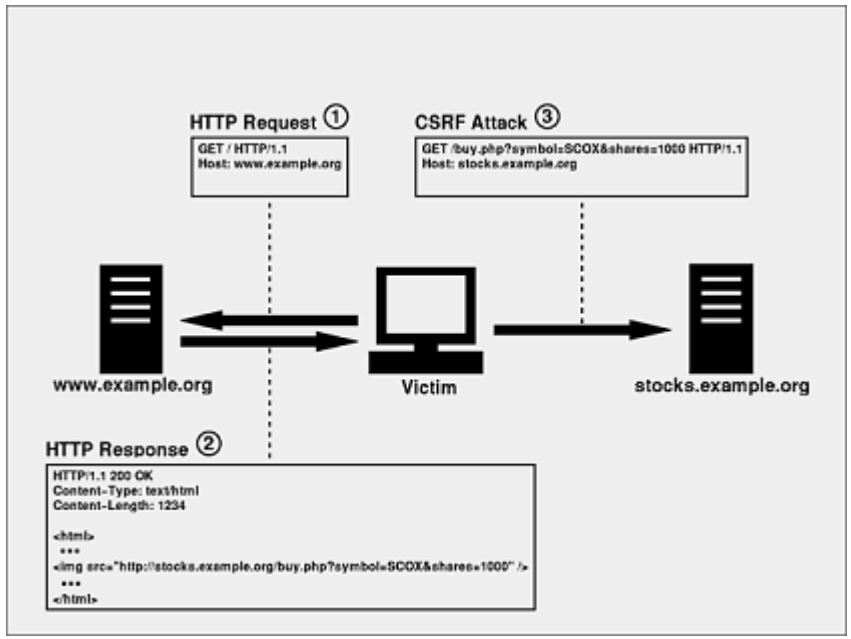
5. Information Disclosure at Full.

[] XSS Attacks : Example



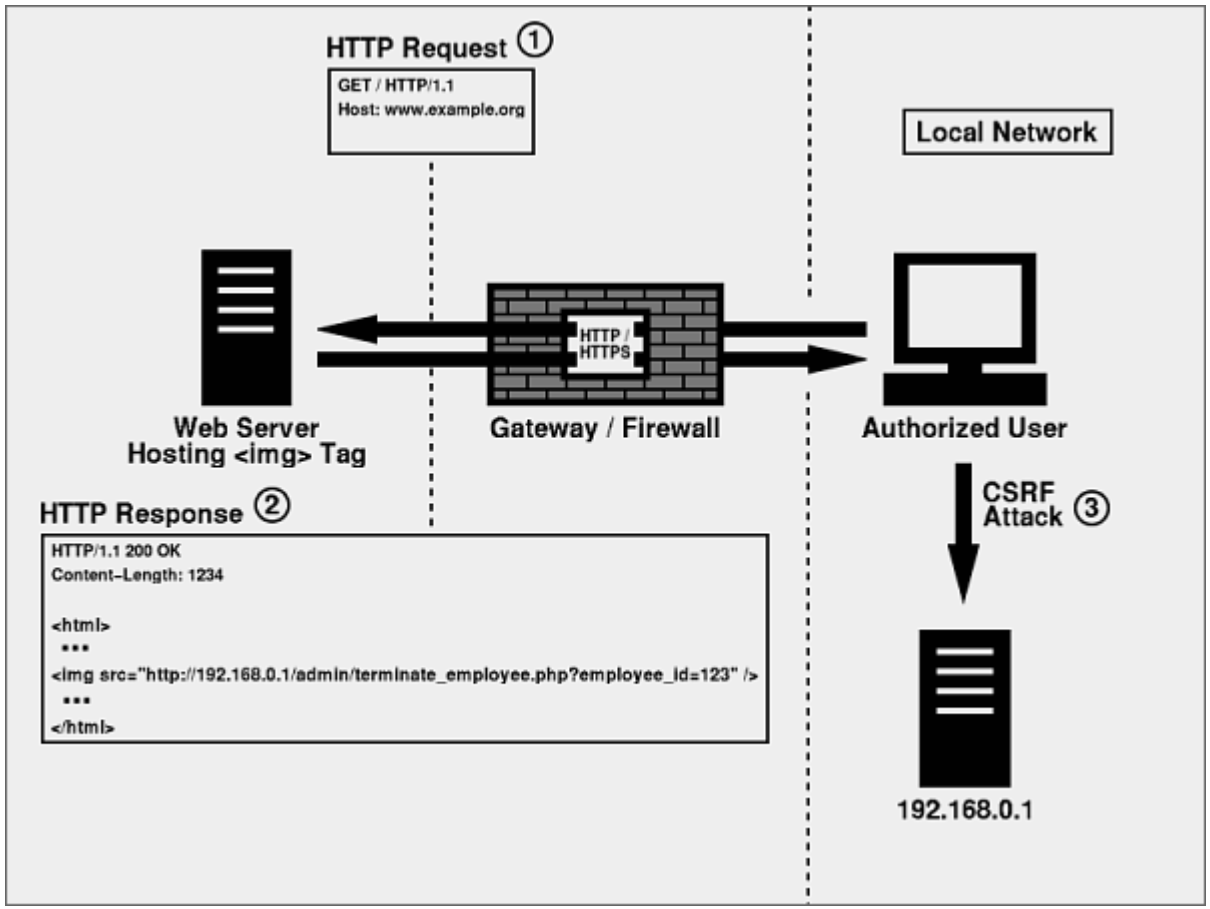
[] Cross Site Request Forging : Direct

1. Cookie Hijacking with Session Undertaking.
2. Authenticating on the Behalf of User by an Attacker.
3. No Stealing of Cookies but Dynamic Manipulation.
4. Lets see :



[] Cross Site Request Forging : Indirect

1. Fusing Proxy between End Points.



[] Cross Site Request Forging : Indirect

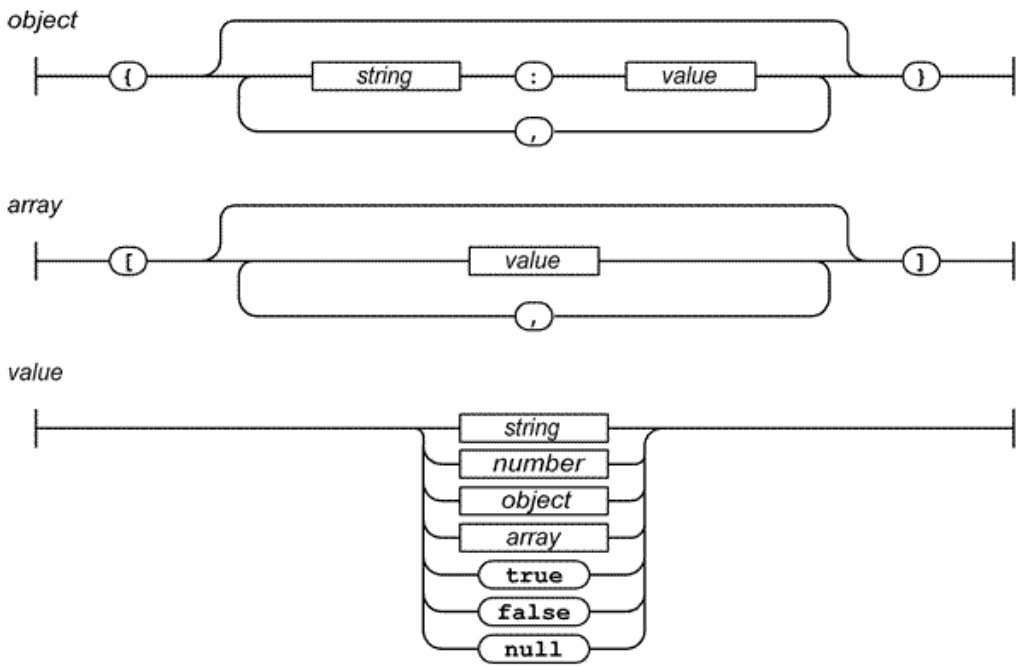
1. Example : JSON Attack Point

```
{ "menu":
 {
  "id": "<img
src="https://books.example.com/clickbuy?book=ISBNhere&quantity=100">",

"value": "<img
src="https://trading.example.com/xfer?from=MSFT&to=RHAT&confirm=Y">",

  "popup":
"<scriptsrc="https://www.google.com/accounts/UpdateEmail?service=adsense&Email
=mymail@newmail.net&Passwd=cool&save="></script>"
                        }
            }

}
```

[] Serialization : JSON/BISON/AJAX

1. Object Interoperability.
2. Concept is Based on Serializing Data i.e Strings.
3. Web 2.0 Finest Edge Driven Attack Vector.

A JSON Layout

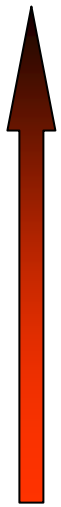


[] Serialization : JSON/BISON/AJAX

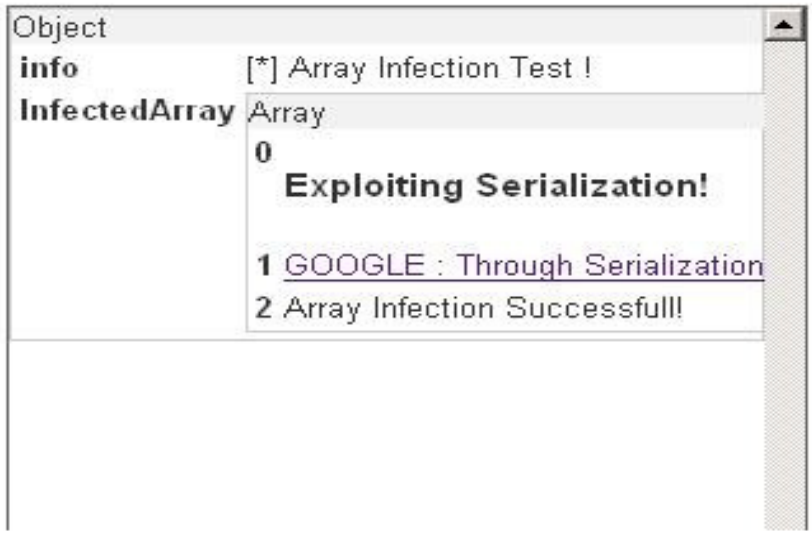
Example :

Send

```
{
info: "[*] Array Infection Test !",

InfectedArray:
["<h3>Exploiting Serialization!</h3>",
 "<a href='http://www.google.com'>
  GOOGLE : Through Serialization</a>",
 "Array Infection Successfull!" ]

}
```



Receive



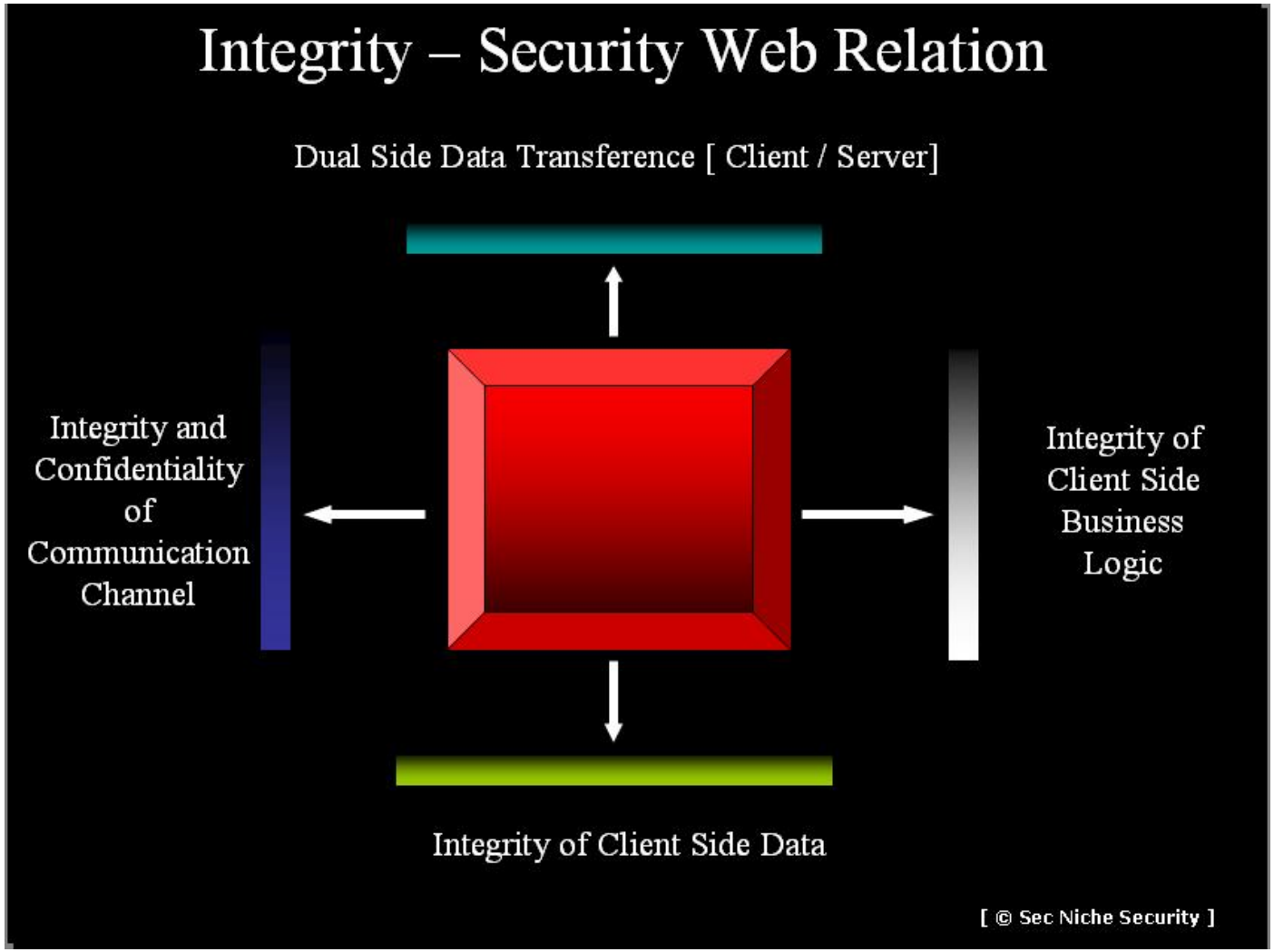
[]Dos : Web Denial of Service Attacks

1. Degradation of Web Service Through Denial of Service.
2. Recursive Calling of URL through Concatenation.
3. Looping Iframe Tags against Entangled Web Entity.

Example :
http://www.verisignsecured.com/Redirect.aspx?http://www.weblogs.com/clickthru?url=http://www.weblogs.com/clickthru?url=http://www.pewinternet.org/report_display.asp?

```
Loop {
        Iframe Tagging.
    }
// Load The Script
```

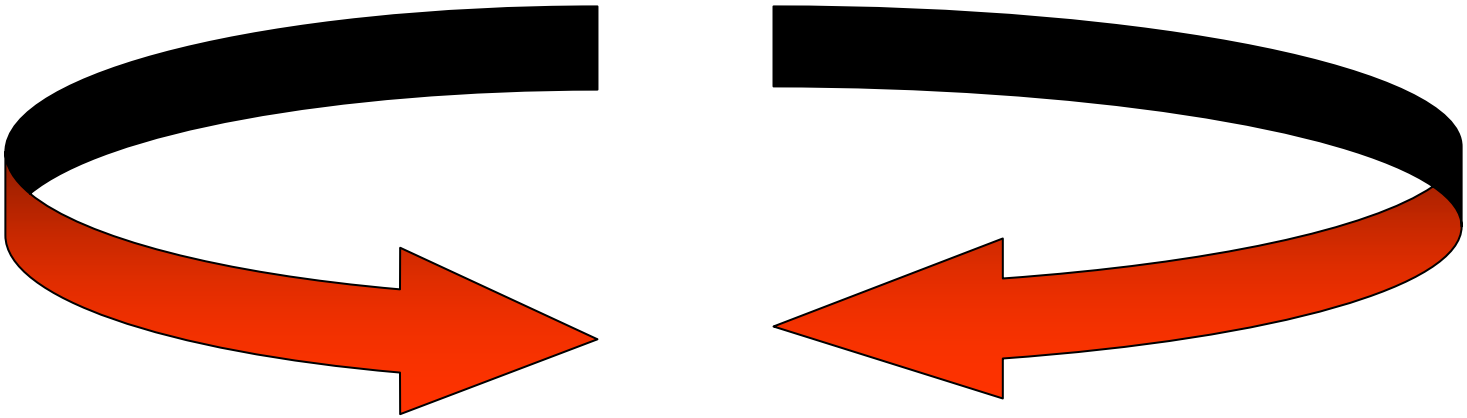
[] Conclusion


Integrity – Security Web Relation
Dual Side Data Transference [Client / Server]
Integrity and Confidentiality of Communication Channel
Integrity of Client Side Business Logic
Integrity of Client Side Data
[© Sec Niche Security]

[] Questions

Contradictory View Always Welcomed !

[] Thanks



Aditya K Sood , Security Researcher.

aditya_ks [at] secniche.org

Zeroknock [at] secniche.org