



Hacker | Halted™

U S A
2 0 1 0

October 9-15th 2010 Miami, Florida

Web Maniac Hacking Trust

Aditya K Sood [[adi_ks \[at\] secniche.org](mailto:adi_ks@secniche.org)]
SecNiche Security

→Disclaimer

Web Maniac - Hacking Trust

Pentesting web applications in a hacker's way. Attack surface varies from application to application. How to think below the surface? That's the aim!

All contents of this presentation represent my own beliefs and views and do not, unless explicitly stated otherwise, represent the beliefs of my current, or any of my previous in that effect, employers.

Screenshots have been shared from various resources. This is done to show the comparative model of various methodologies.



→ About Me



- Founder , SECNICHE Security Labs.
<http://www.secniche.org>
- PhD Candidate at Michigan State University
- Worked previously for Armorize as Senior Security Practitioner , COSEINC as Senior Security Researcher and Security Consultant for KPMG
- Written content Author for HITB E-Zine, Hakin9 ,ELSEVIER, USENIX Journals.
- Like to do Bug Hunting and Malware dissection.
- Released Advisories to Forefront Companies.
- Active Speaker at Security Conferences including RSA etc.
- Blog: <http://secniche.blogspot.com> | <http://zeroknock.blogspot.com>

→ Agenda

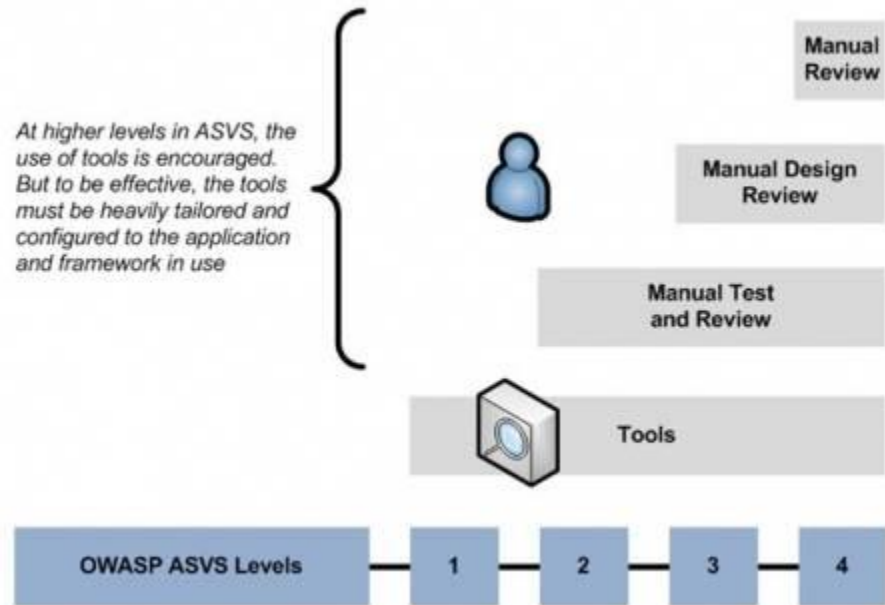
- ✓ Web Application Security Standards
- ✓ Web Application Security- A view of Reality
- ✓ Web Application – Testing and Development Methodologies
- ✓ Facets of Web Application Pen Testing (WAPT)
- ✓ Demonstrations – Live Targets



→ Web Application Security Standards – Really

WASC Threat Classification v2	OWASP Top Ten 2010 RC1
WASC-19 SQL Injection	A1 - Injection
WASC-23 XML Injection	
WASC-28 Null Byte Injection	
WASC-29 LDAP Injection	
WASC-30 Mail Command Injection	
WASC-31 OS Commanding	
WASC-39 XPath Injection	
WASC-46 XQuery Injection	
WASC-08 Cross-Site Scripting	A2 -Cross Site Scripting (XSS)
WASC-01 Insufficient Authentication	A3 - Broken Authentication and Session
WASC-18 Credential/Session Prediction	
WASC-37 Session Fixation	
WASC-47 Insufficient Session Expiration	
WASC-01 Insufficient Authentication	A4 - Insecure Direct Object References
WASC-02 Insufficient Authorization	
WASC-33 Path Traversal	
WASC-09 Cross-site Request Forgery	A5 - Cross-Site Request Forgery
WASC-14 Server Misconfiguration	A6 - Security Misconfiguration
WASC-15 Application Misconfiguration	
WASC-02 Insufficient Authorization	A7 - Failure to Restrict URL Access
WASC-10 Denial of Service	
WASC-11 Brute Force	
WASC-21 Insufficient Anti-automation	
WASC-34 Predictable Resource Location	
WASC-38 URL Redirector Abuse	A8 - Unvalidated Redirects and Forwards
WASC-50 Insufficient Data Protection	A9 - Insecure Cryptographic Storage
WASC-04 Insufficient Transport Layer Protection	A10 -Insufficient Transport Layer Protection

OWASP Top 10 – 2007 (Previous)	OWASP Top 10 – 2010 (New)
A2 – Injection flaws	A1 – Injection
A1 – Cross Site Scripting (XSS)	A2 – Cross Site Scripting (XSS)
A7 – Broken Authentication and Session Management	A3 – Broken Authentication and Session Management
A4 – Insecure Direct Object Reference	A4 – Insecure Direct Object References
A5 – Cross Site Request Forgery (CSRF)	A5 – Cross Site Request Forgery (CSRF)
<was T10 2004 A10 – Insecure Configuration Management>	A6 – Security <u>Misconfiguration</u> (NEW)
A10 – Failure to Restrict URL Access	A7 – Failure to Restrict URL Access
<not in T10 2007>	A8 – <u>Unvalidated</u> Redirects and Forwards (NEW)
A8 – Insecure Cryptographic Storage	A9 – Insecure Cryptographic Storage
A9 – Insecure Communications	A10 - Insufficient Transport Layer Protection
A3 – Malicious File Execution	<dropped from T10 2010>
A6 – Information Leakage and Improper Error Handling	<dropped from T10 2010>

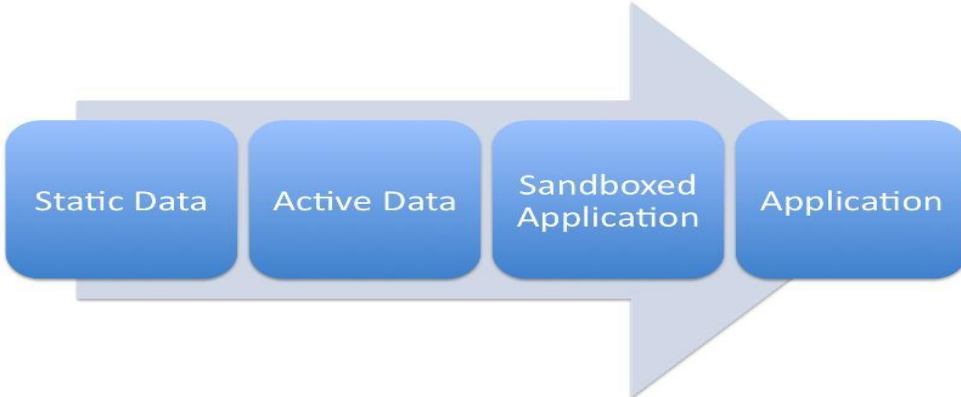
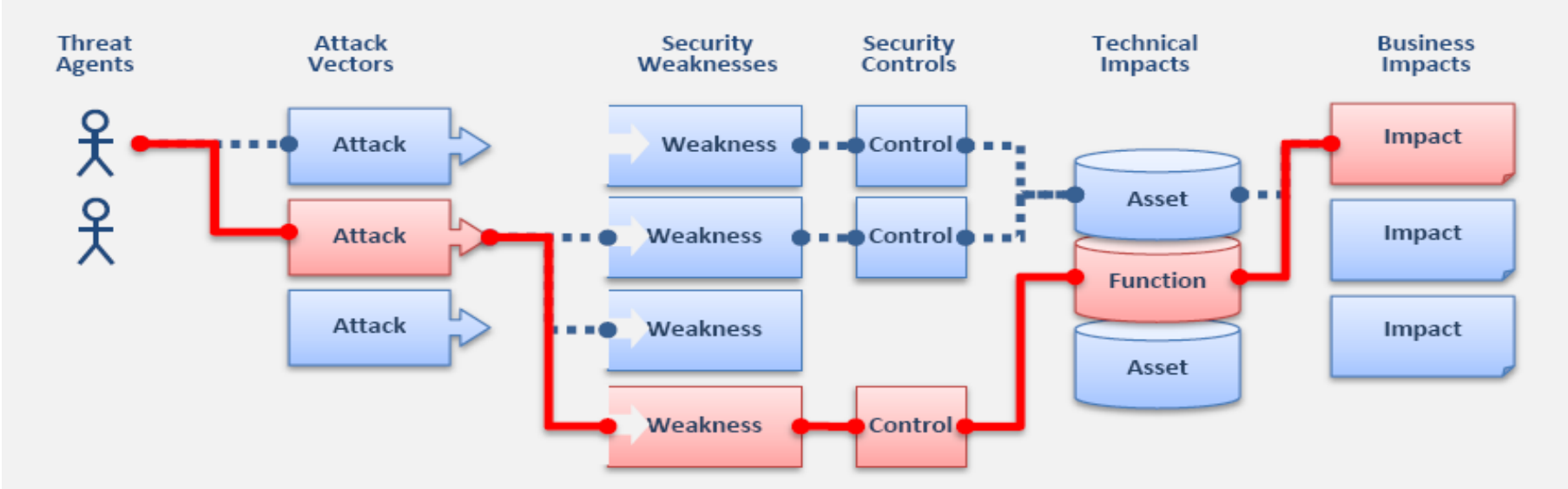


→ Web Application Security Standards - ?? ! Answers !

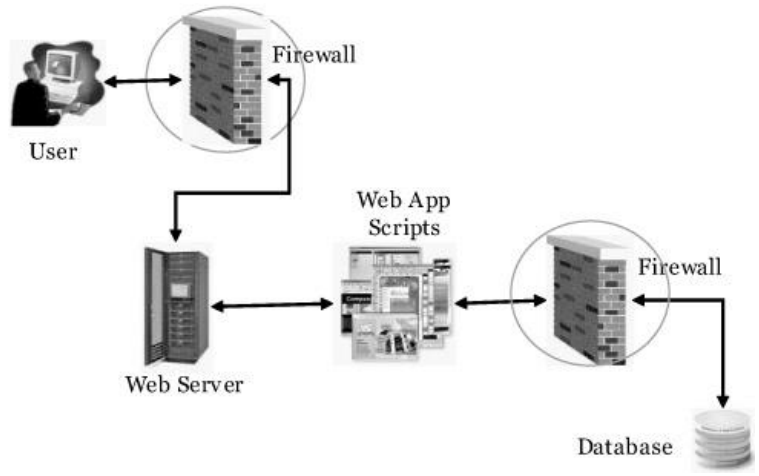
- Standards provide specific classification of vulnerabilities
- Do they comprise of all types of vulnerabilities ?
- Are all types of web attacks predefined in them?
- Do you think the **design** of web application matters? [to what extent]
- A view of web application and a website under testing.
- Do platforms and web servers matter while web application assessment?
- Do you think penetration testing of web applications is beyond these standards ?



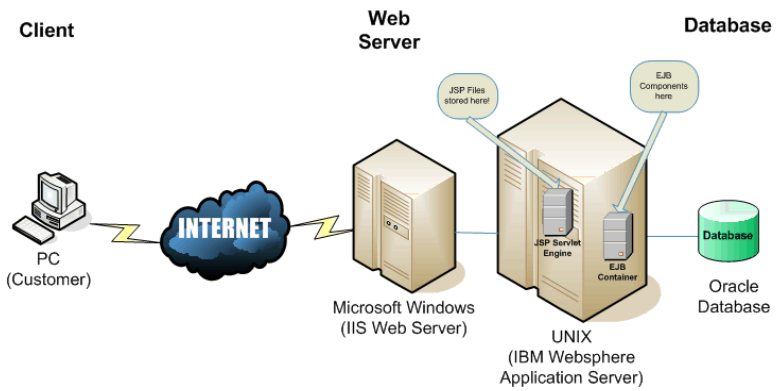
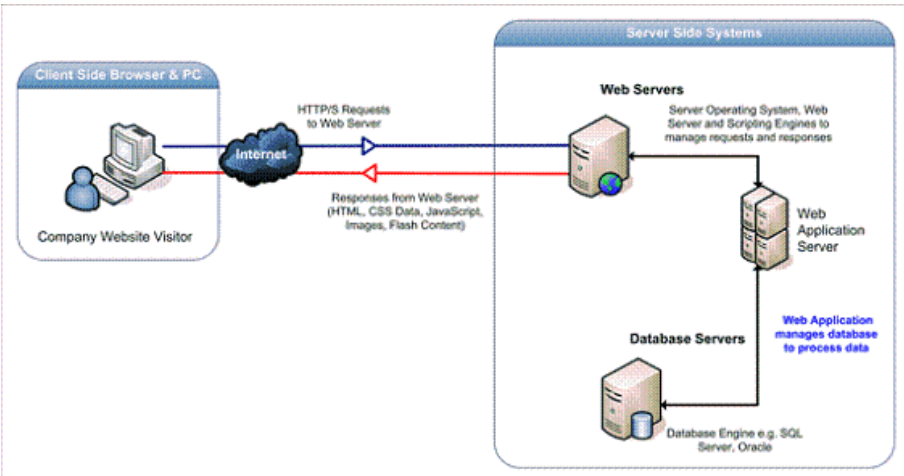
→ Web Application State and Risks



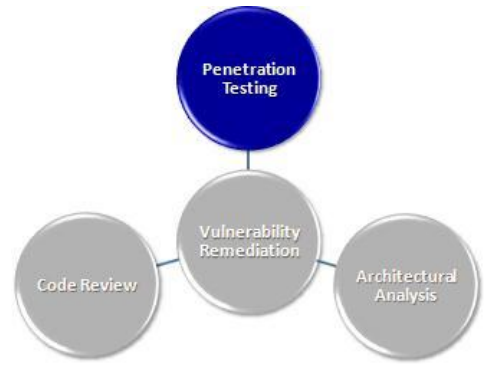
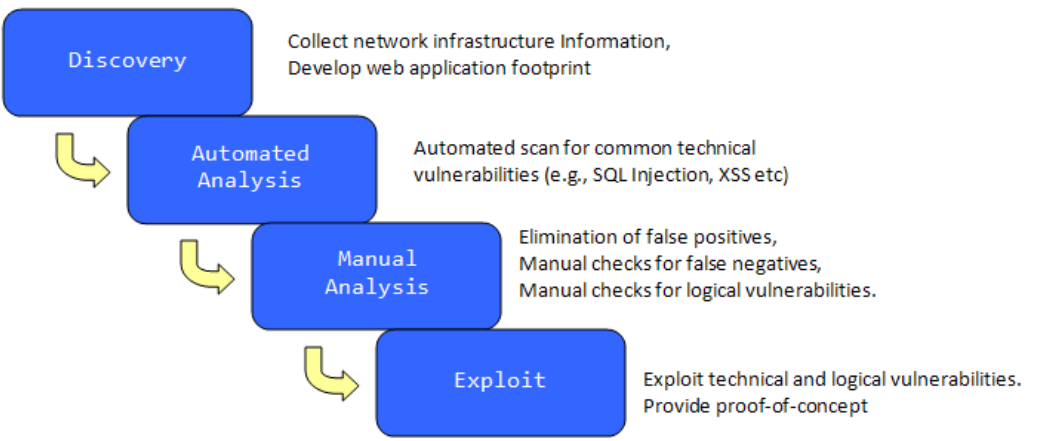
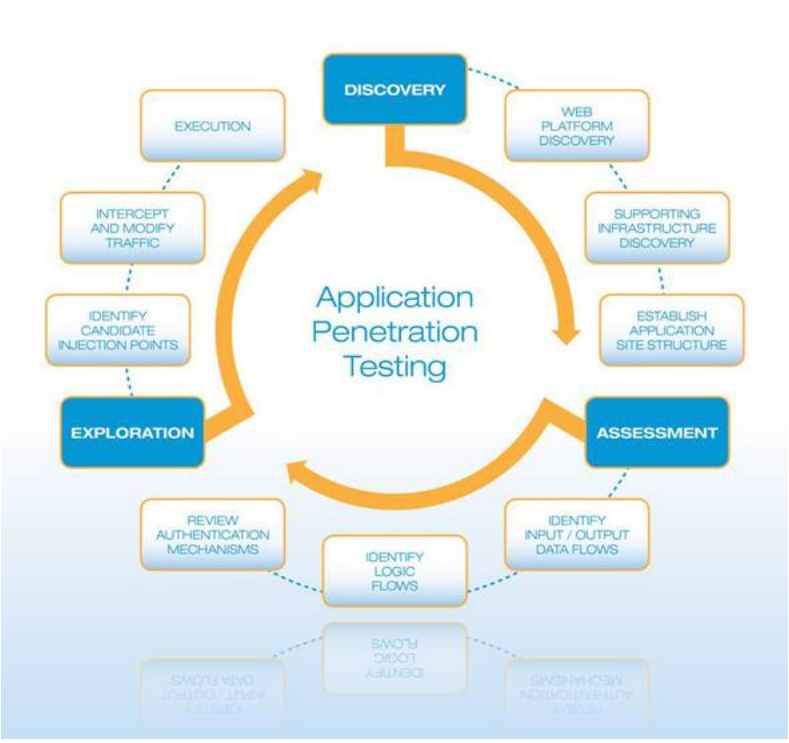
→ Web Application Architecture - Development



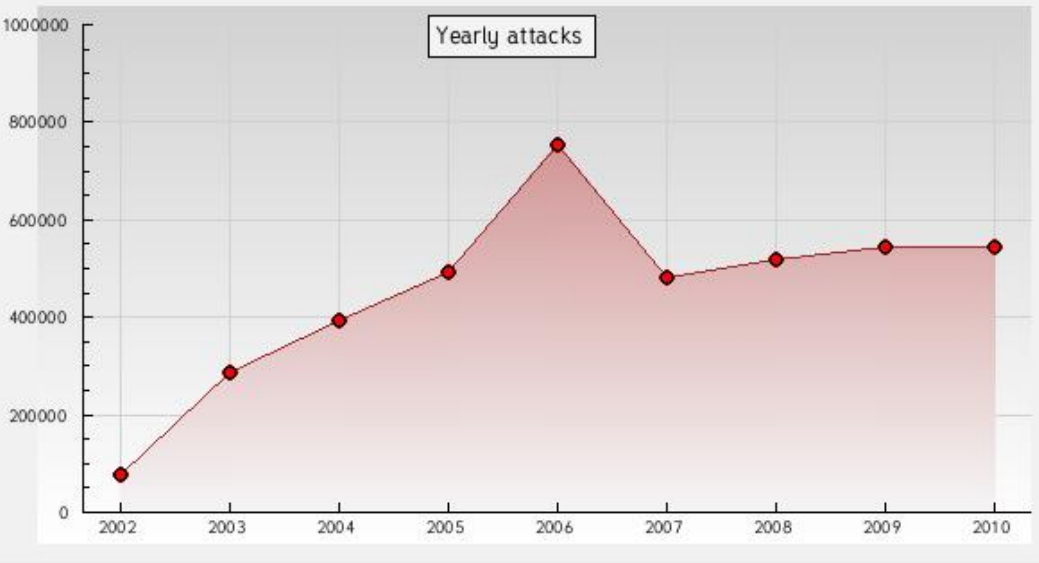
Web Development Process



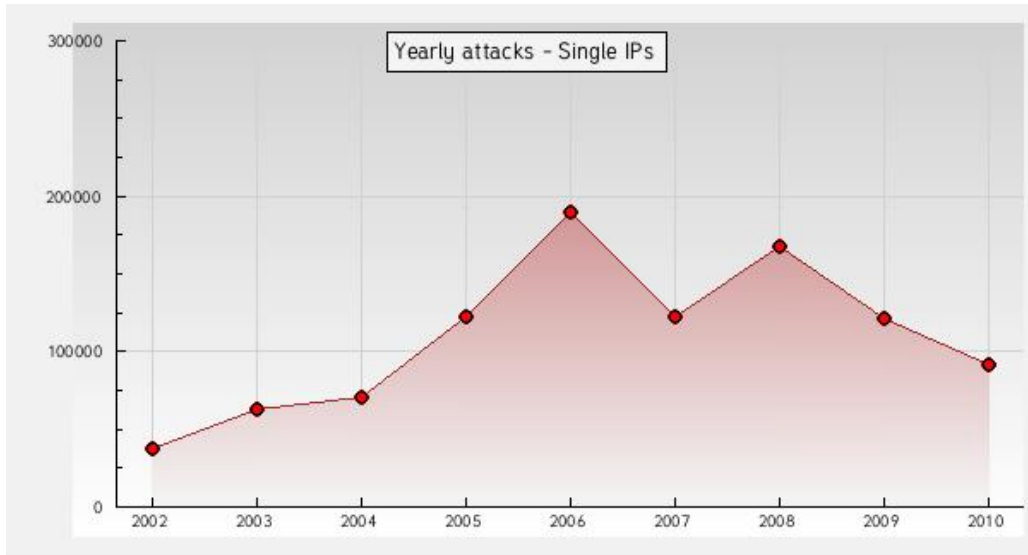
→ Web Application Testing - Methodologies



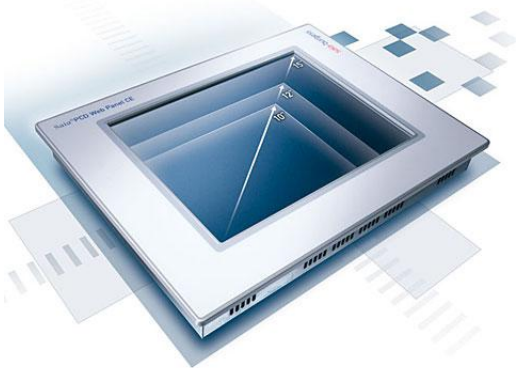
→ Why Security Testing ?



Defacement Statistics



→ Web Application Security ! Reality - Broken



→ Is that Ethical?

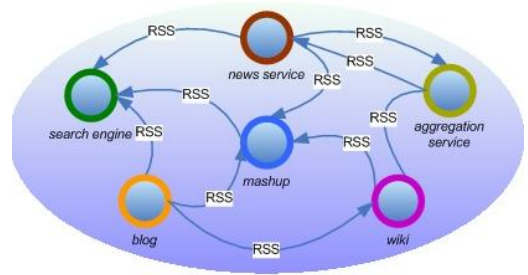
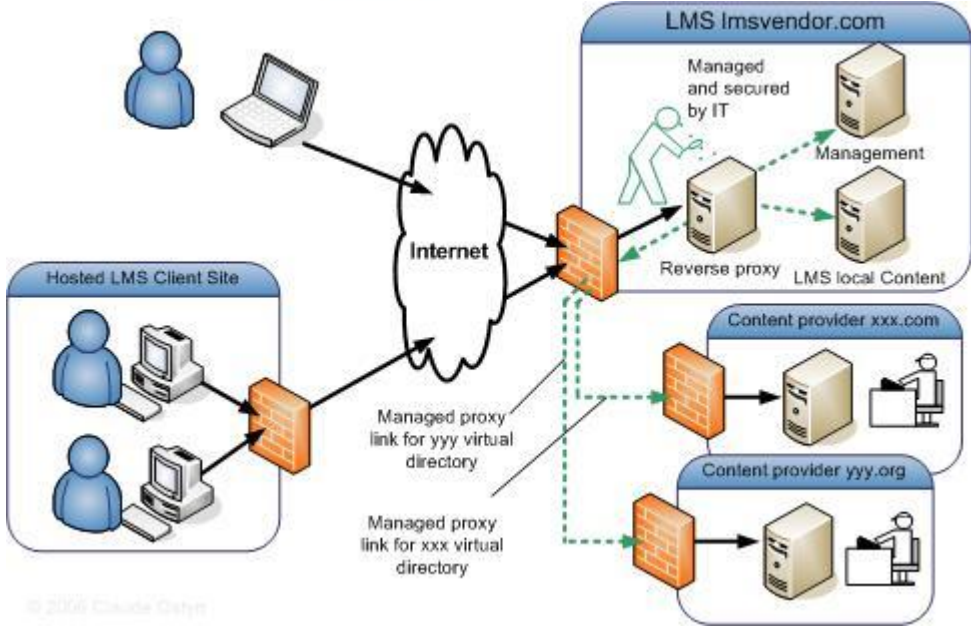
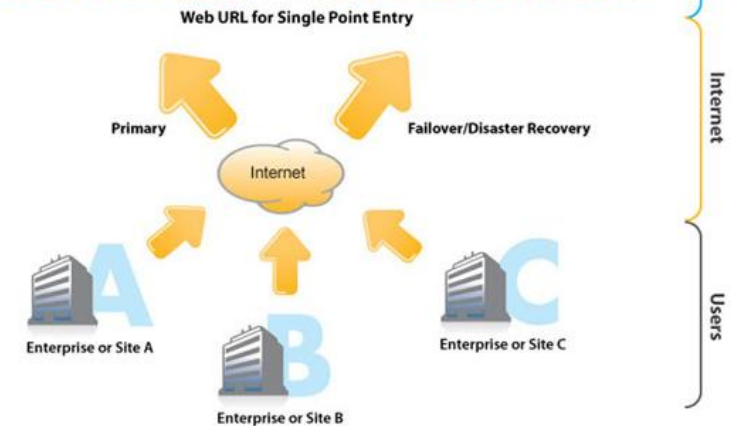
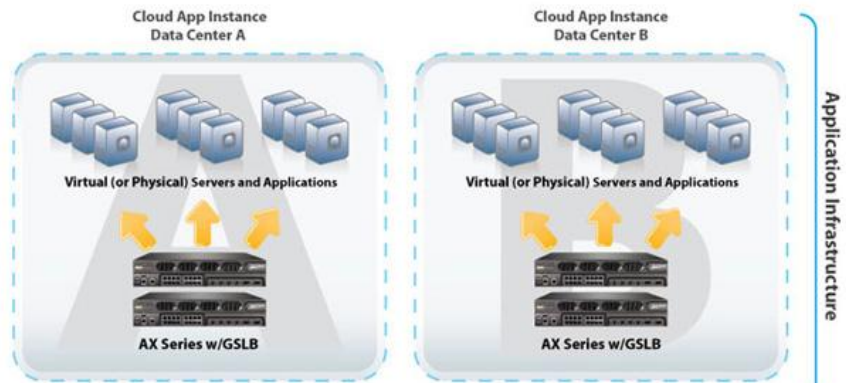
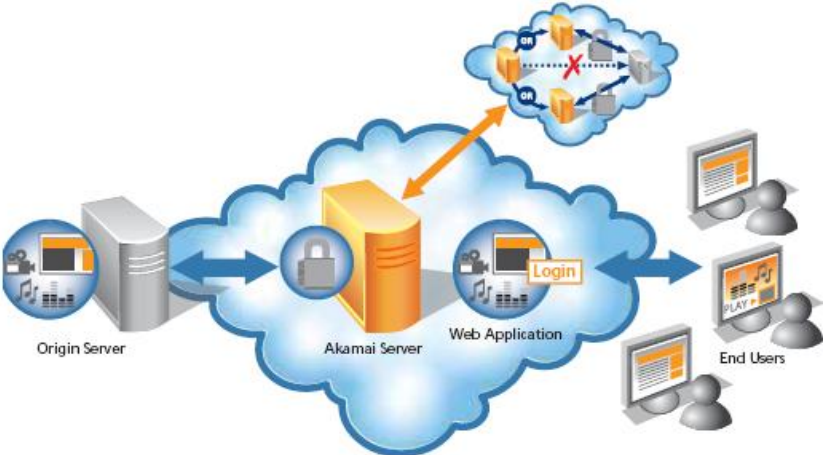


→ Existence and Reality – Web Penetration Test

- Is this all about compliance (PCI) ?
- Is this all about reporting generic issues and using reports for cert's?
- Do you think organizational teams patch all the reported issues?
- White box or Black box – Changed definitions.
- Security Assessment ! = Penetration Testing [**Mismatch**]
- Time dependency – A big factor in determining the effectiveness
- Penetration Tests – Does not provide security / That's the **Truth**
 - Applied security comes out of the actions taken to remove those vulnerabilities which are exploited during the course of penetration testing.
 - Vulnerability assessment provides a glimpse of security to some degree
 - Penetration tests emulate real world attacks to exploit the network and web infrastructure
 - Effective penetration tests provide a degree to which systems can be exploited. It can be more.

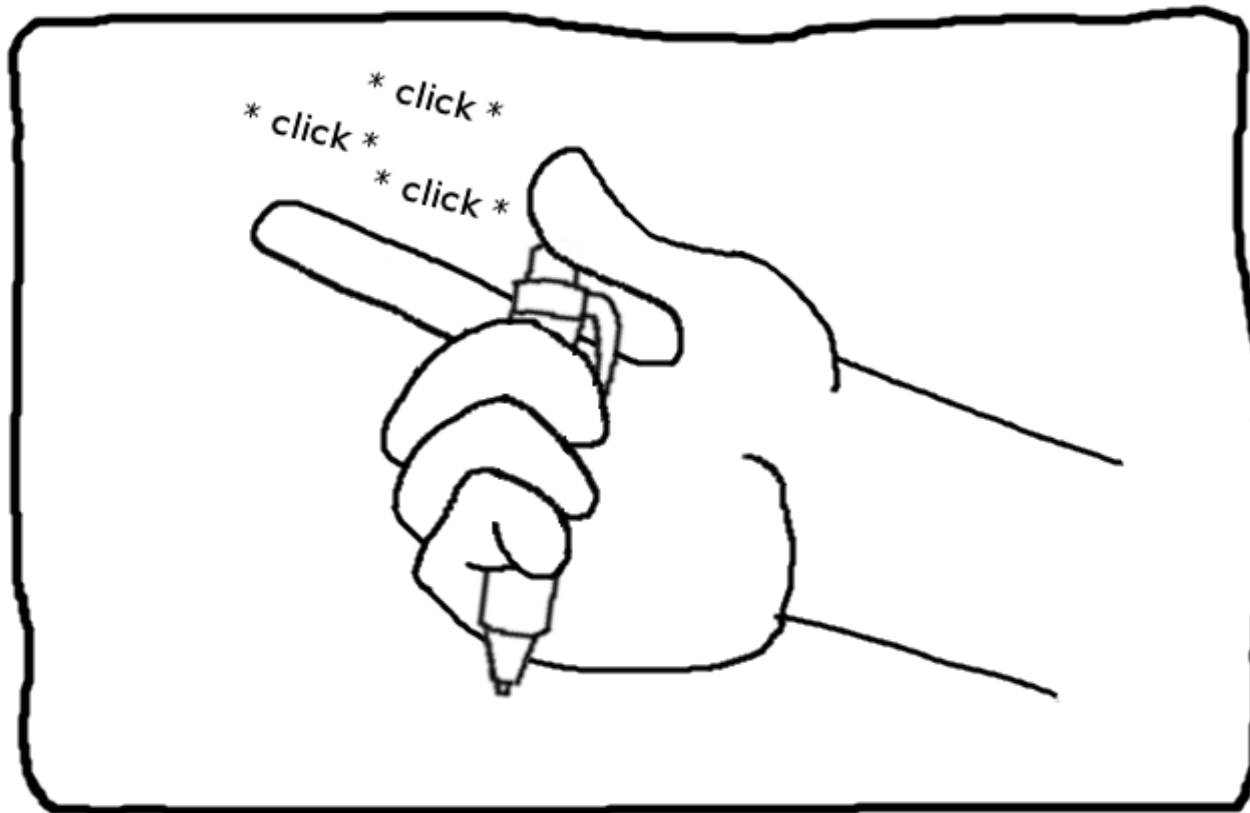


→ Pentesting Stringency in Real World



→ Is that True ?

Pen-testing is overrated

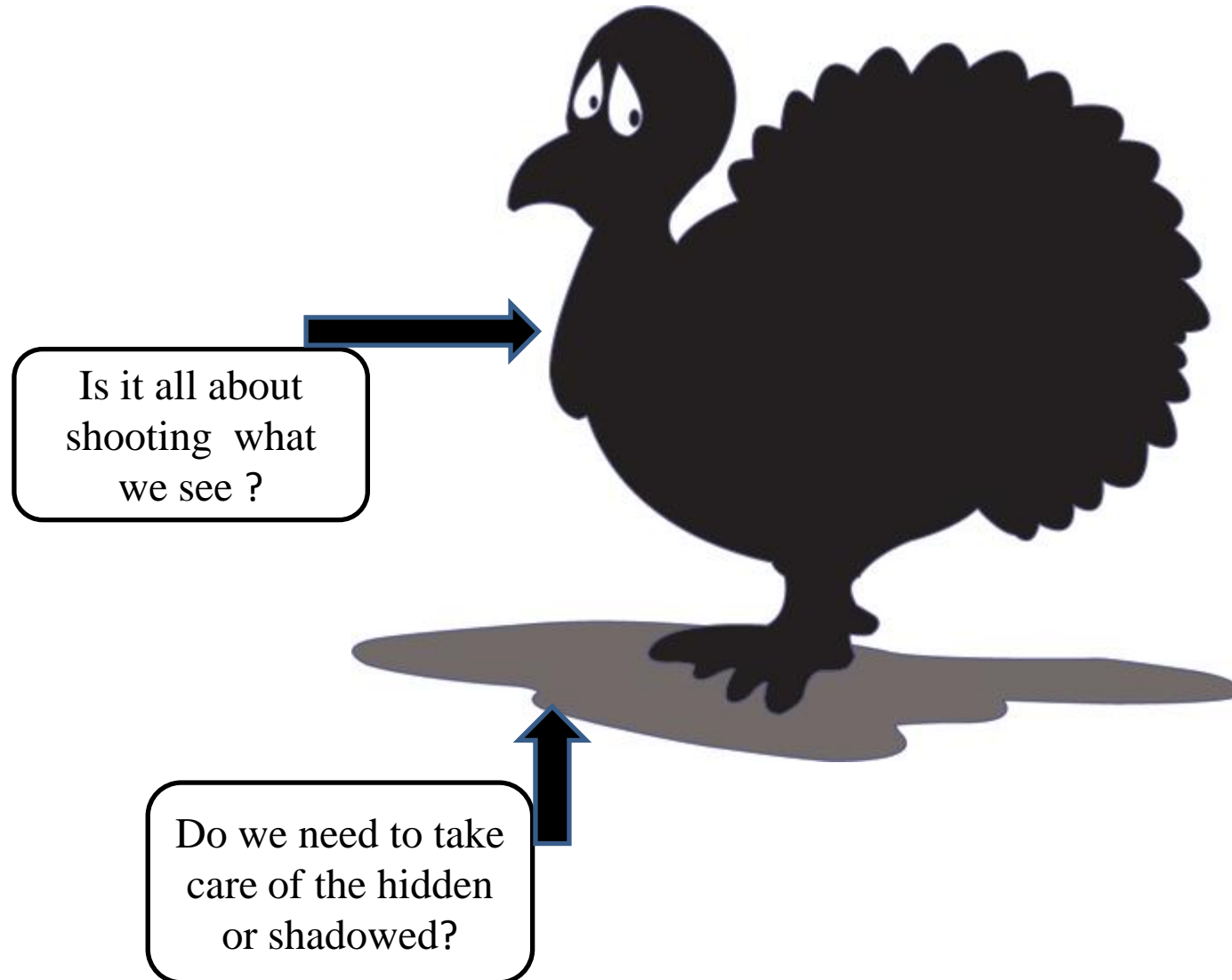


Then what about Human Ignorance ?

A critical component in every sphere. Hard to beat it.

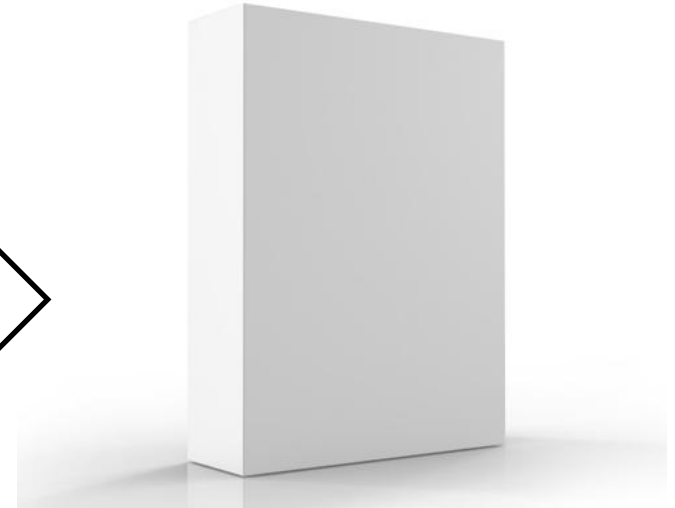


→ Thinking in the Wild – Web Penetration Testing.



→ Web Penetration Test – The Refined Art

- Turning the Black Box Testing into White box Testing
- Expertise – Hacking in a controlled manner
- Meeting the expectations



→ The One – Murphy's Law (Variation)

- Pen Tester – The Word of Advice

“Everything that goes wrong on the target host, network, or on the Internet from two weeks before you plug in to two weeks after you submit the report will be your fault.”

Murphy's Law

*“If anything
can go wrong,
it will”*

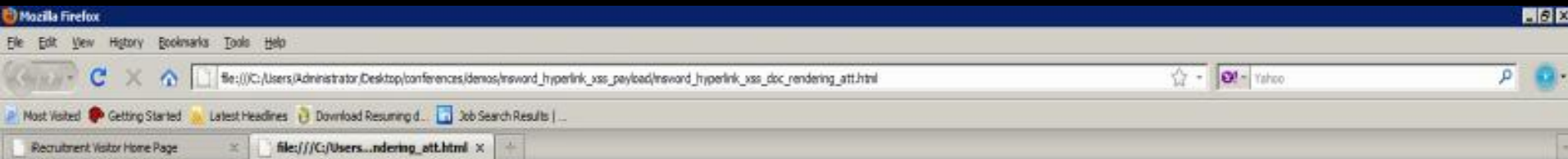
-

→ Demonstration

- Attacking Web Apps through Content Rendering – 4: 15 M
- SQLXSSI – XSS through SQL Injections : Yahoo – 5:30 M
- Persistent Redirection Attacks and Malware - 4:00 M
- Content Delivery Networks – Infection Behavior - 4:09 M
- Widget Redirection Attacks – Outbrain – 3:20 M



→ Demo 1: Document Content Rendering Attacks



Demonstrations

Document Hyperlinking XSS Payload Rendering Attacks Vulnerability in Web applications and Enterprise Solutions

This issue is a potential part of web application functionality
Disclaimer - This is only for demonstration purposes. All on your mobility.

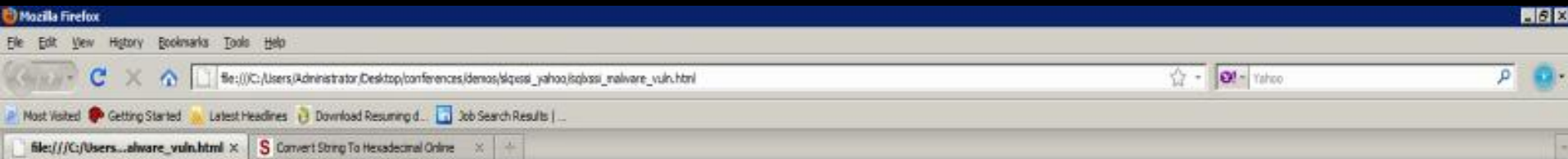
Proof of Concept

- Step 1 - Finding a upload module in the web application and understanding the design
- Step 2 - Setting XSS payload in MSWord Hyperlinks
- Step 3 - Uploading the file on the web application and allow it to render
- Step 4 - Rendering the MS Word document while translation results in XSS
- Step 5 - All at your own risk.

(C) SecNiche Security (<http://www.secniche.org>)



→ Demo 2 : SQLXSSI – Using SQLI to conduct XSS



Demonstrations

SQLXSSI – XSS Payloads in SQL Variables

Database – MYSQL / MSSQL / ORACLE

Vulnerability in Web applications and Enterprise Solutions

It has been transferred into a malware demo for understanding and leveraging knowledge out of the issue.
Disclaimer - This is only for demonstration purposes. All on your mobility.

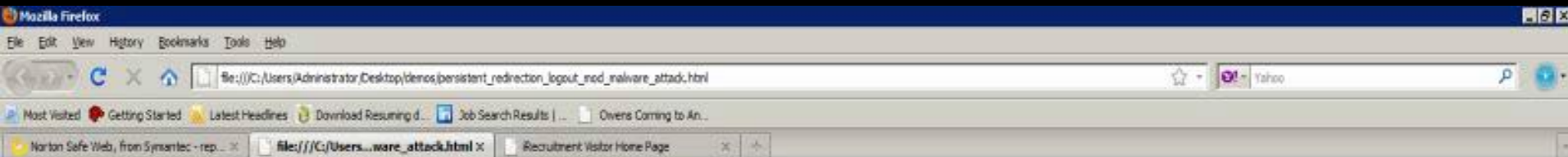
Proof of Concept

- Step 1 – Finding a vulnerable website throwing reflective errors
- Step 2 – Exploiting through UNION SQL poisoning with XSS payloads
- Step 3 - Reflective error renders XSS payload in context of browser
- Step 4 – Possibility of all sorts of attacks.

(C) SecNiche Security (<http://www.secniche.org>)



→ Demo 3 : Persistent Logout Redirection Attacks



Demonstrations

Persistent Redirection - Logout Module - Malware Infection - Attacks

Vulnerability in Web applications and Enterprise Solutions

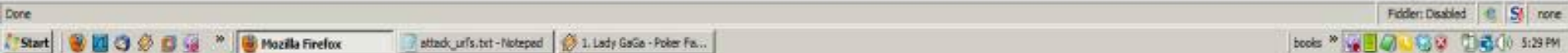
This issue is a potential redirection attack which is persistent in logout module of an enterprise application. It has been transferred into a malware demo for understanding and leveraging knowledge out of the issue.

Disclaimer - This is only for demonstration purposes. All on your mobility.

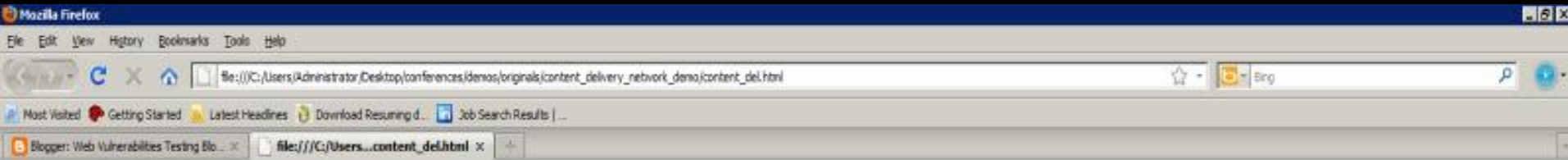
Proof of Concept

- Step 1 - Analysing the URL and inherent application parameters. Debug it to get the parameter used for redirection on logout module.
- Step 2 - URL is designed with malicious link as a logout redirection parameter value
- Step 3 - User is allowed to log into the application
- Step 4 - User logged in and started doing some work and logged out.
- Step 5 - During logout redirection occurs to malicious website and starts downloading files into system.

(C) SecNiche Security (<http://www.secniche.org>)



→ Demo 4 : Third Party Content Delivery Infections



Demonstrations

Content Delivery Networks – Website Hacking and Malware Infection

Third Party Content Rendering

Vulnerability in Web applications and Enterprise Solutions

It has been transferred into a malware demo for understanding and leveraging knowledge out of the issue.
Disclaimer - This is only for demonstration purposes. All on your mobility.

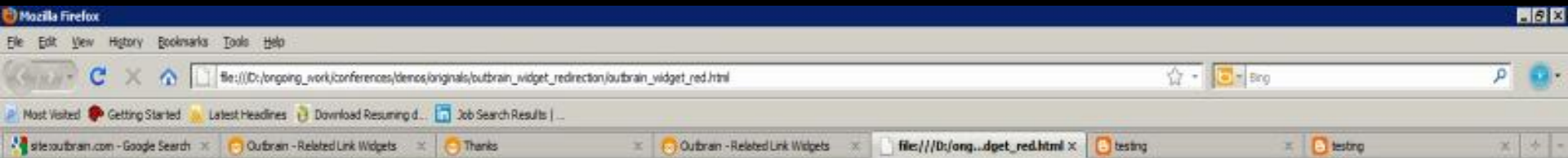
Proof of Concept



(C) SecNiche Security (<http://www.secniche.org>)



→ Demo 5 : Widget Redirection Attacks



Demonstrations

Widget Redirection Attacks – Malware Infection

Third Party Content Rendering

Vulnerability in Web applications and Enterprise Solutions

It has been transferred into a malware demo for understanding and leveraging knowledge out of the issue.
Disclaimer - This is only for demonstration purposes. All on your mobility.

Proof of Concept

(C) SecNiche Security (<http://www.secniche.org>)



→ Questions and Queries



→ Thanks

SecNiche Security : <http://www.secniche.org>

Hacker Halted – <http://www.hackerhalted.com>

