net square

secure.automate.innovate

The Future Of Automated Web Application Testing

Preview

- Web 1.0
  - Application architecture and it’s traditional analysis methodology
  - Automated web application testing and it's limitation
- Web 2.0
  - How it works
  - Challenges and limitation of web 2.0 application testing
- Next generation auditing tool

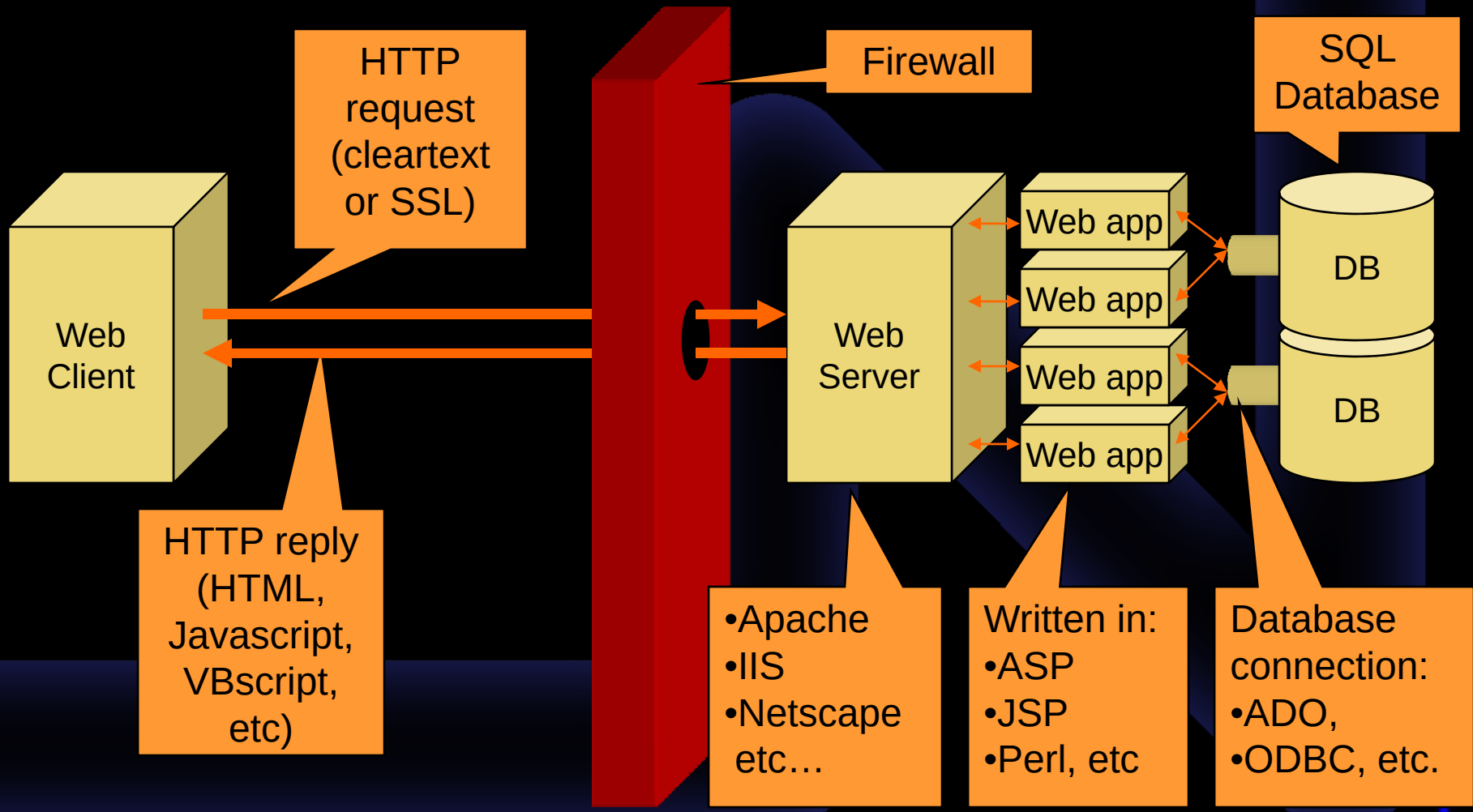
net square

secure.automate.innovate

Web 1.0

Application behavior and it’s traditional analysis methodology

Web 1.0 Application Architecture



Web 1.0 Application Architecture

- Works with page refresh
- Form submitting model
  - Inputs submitted via query string or form parameters
- Browser generates http requests for images, js, etc. while rendering html response through DOM
- Request also can be sent by javascript, ActiveX, Applets, Flash, etc. directly

Web 1.0 Application Architecture

- Server & Web Application
  - Parses http request and map URL with web application physical resource
  - Generates HTML Response based on the supplied resource query and input parameters

net square

secure.automate.innovate

Web 1.0

Traditional Analysis methodology

Traditional Analysis methodology

- Information gathering
  - Http Response Code – 2xx, 4xx, 5xx
  - Http contents
    - Extract forms and query string parameters.
    - Hidden fields, comments, mail ids,
    - Cookie name / value
    - Java scripts,
    - ActiveX and Applets
- Find injection point, suspicious field or query string parameters

Traditional Analysis methodology

- Manipulate field with malicious characters and send request
- Look at the html response, get some clue, modify parameters and send request.
- Do same again and again until…. Bingo !!
- Resources used,
  - Browser
  - Plug-ins (livehttp header or web browser toolbar)
  - Sniffer

net square

secure.automate.innovate

Web 1.0

Automated web application testing and
It's Challenges & limitation

Automated web application testing

- Input – index page or list of stored URLS
- Configurations – depth, within domain, max links, include / exclude, user-agent, etc.
- Testing methodology
  - Crawls web application recursively and collects URLS
  - Find injection point or attack vector for URL
    - Query String parameters
    - It’s Html response form fields
    - Cookie

Automated web application testing

- Popular Web Application Scanners
  - NTObjective's NTOSpider
  - IBM/Watchfire's AppScan
  - HP/SPI Dynamics' WebInspect
- Demo
  - NTOInsight

It's Challenges & Limitations

- Building correct attack request
  - Forms submission by “onclick” event
    - Wrong action or target picked up by automated tool
- Manage context through out the session
  - Logout innocently
- Crawl a site in certain order – logical action
- Infinite crawl – Dynamic URL creation

It's Challenges & Limitations

- Executing java script like a Browser
  - Dynamic menus and css
  - URL decryption on the fly by java scripts
- Identify correct attack vector in URL
  - No question mark in a URL
  - Strange extension
  - Custom techniques to supply inputs.

It's Challenges & Limitations

- False positive/negative and duplication
  - Detects vulnerability through http response code
  - Or regex pattern search in html response
- How to detect persistent XSS??
- Custom response code (obfuscated 200)
- Random 404 pages

It's Challenges & Limitations

- Authenticated scanning
  - Login automatically on authenticated URL,
    - Where to go after authentication ?
  - Form based authentication
    - Success or fail, how to decide ?
- Captcha, how to handle ?
- Broken access controls
- Information leakage
- Design issues

Scanners are also getting smarter

- Page Signature technology being used to identify obfuscated 200, random 404 pages and Form based authentication
- Java scripts based URLs can be fetched by regex based search

- Most of the scanners identify technical vulnerabilities like SQL Injection, XSS, etc.

Web 2.0

How it works !!

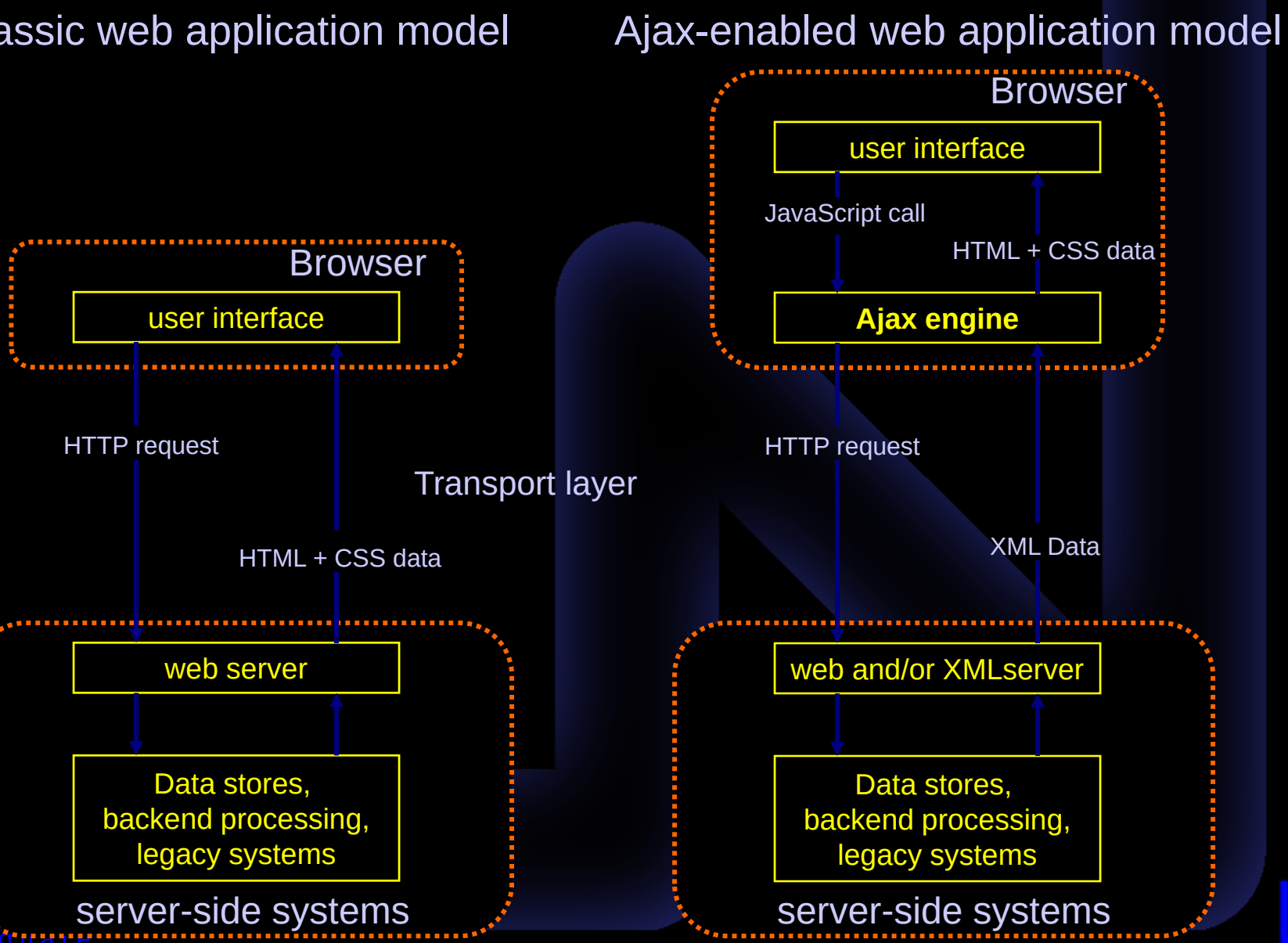
Web 2.0 Technology

- Web 2.0 Applications are on the rise
- Rich Internet Applications (RIA) – reshaping application front
- Web Services on the rise – forming backend of applications
- Gartner is advising companies to take up Web services now, or risk losing out to competitors embracing the technology.

Web 2.0 Technology

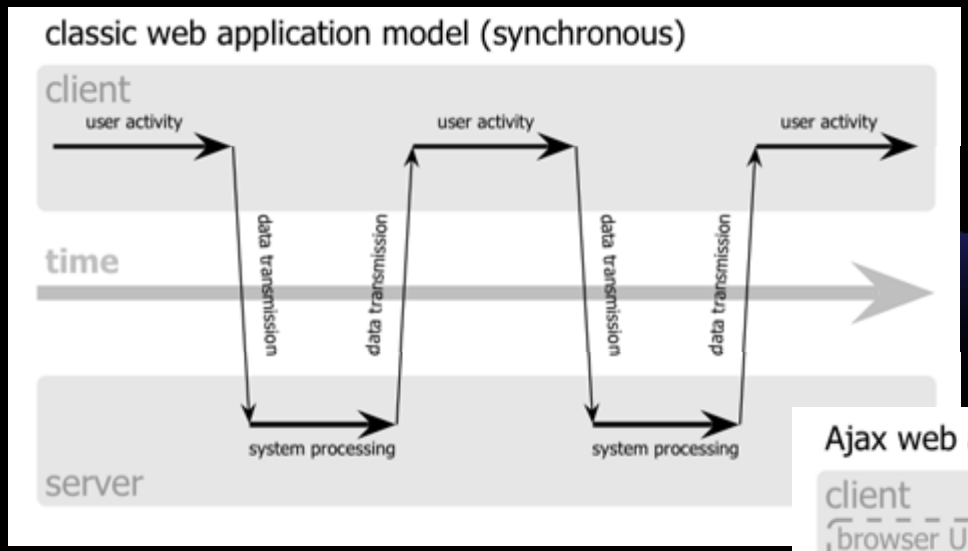
- Web Services is forming back end and accessible on SOAP
- AJAX – empowering browsers
- XML based services
- Rich Internet Applications are consuming back end web services
- Search engines and mechanisms for web services publishing and accessing
- Security evolving around web services

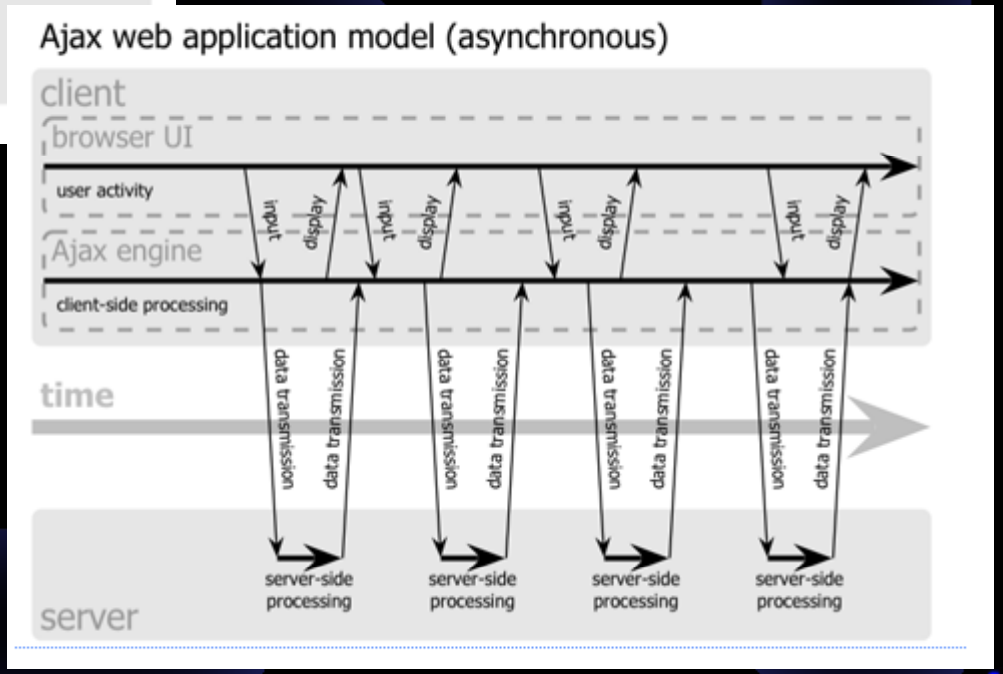
Ajax model



Source : http://www.adaptivepath.com/publications/essays/archives/000385.php

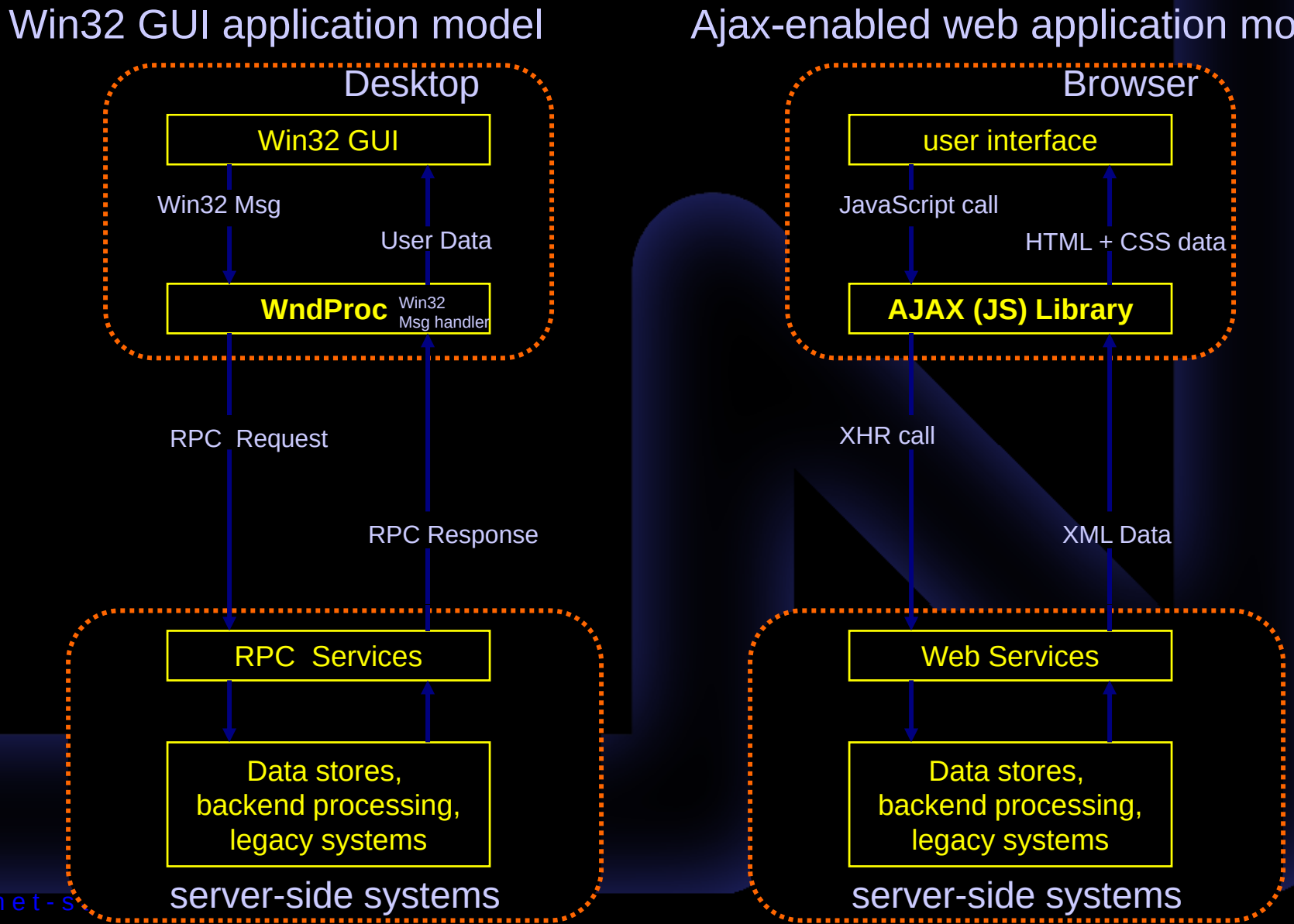
AJAX introduction





Source : http://www.adaptivepath.com/publications/essays/archives/000385.php

Win32 APP vs Web 2.0



net square

secure.automate.innovate

Web 2.0

Challenges and limitation of web 2.0 application testing

Impact of Web 2.0

- Application Infrastructure

Changing dimension	Web 1.0	Web 2.0
(AI1) Protocols	HTTP & HTTPS	SOAP, XML-RPC, REST etc. over HTTP & HTTPS
(AI2) Information structures	HTML transfer	XML, JSON, JS Objects etc.
(AI3) Communication methods	Synchronous Postback Refresh and Redirect	Asynchronous & Cross-domains (proxy)
(AI4) Information sharing	Single place information (No urge for integration)	Multiple sources (Urge for integrated information platform)

Impact of Web 2.0

- Security Threats

Changing dimension	Web 1.0	Web 2.0
(T1) **Entry points**	Structured	Scattered and multiple
(T2) **Dependencies**	Limited	• Multiple technologies • Information sources • Protocols
(T3) **Vulnerabilities**	Server side [Typical injections]	• Web services [Payloads] • Client side [XSS & XSRF]
(T4) **Exploitation**	Server side exploitation	Both server and client side exploitation

Impact of Web 2.0

- Methodology

Changing dimension	Web 1.0	Web 2.0
Footprinting	Typical with "Host" and DNS	Empowered with search
Discovery	Simple	Difficult with hidden calls
Enumeration	Structured	Several streams
Scanning	Structured and simple	Difficult with extensive Ajax
Automated attacks	Easy after discovery	Difficult with Ajax and web services
Reverse engineering	On the server-side [Difficult]	Client-side with Ajax & Flash
Code reviews	Focus on server-side only	Client-side analysis needed

Impact of Web 2.0

- Countermeasure

Changing dimension	Web 1.0	Web 2.0
Owner of information	Single place	Multiple places [Mashups & RSS]
Browser security	Simple DOM usage	Complex DOM usage
Validations	Server side	Client side [incoming content]
Logic shift	Only on server	Client side shift
Secure coding	Structured and single place	Multiple places and scattered

Challenges and Limitation

- No success with http response parsing
- Everything is generated run time
- Path of execution is dynamic
- Cannot predict next URL
- Need to grab data in runtime through DOM
  - cannot use anything other than browser
  - human element is must

Future Approach

Automated Web Application Testing

Future Approach

- "only about half of the required tests for a security assessment can be performed on a purely automated basis. The other half require human involvement, typically for identifying vulnerabilities in business logic.“
  - Jeremiah Grossman (CTO, Whitehat Security)
- So, finally you need a tool which will have both the things at one place..
  - Browser based Web Application Scanner

Future Approach

- Browser based toolbar Advantages
  - Hybrid – Automated + Manual both
  - Uses Browser DOM directly
  - Crawling is possible but it is not required because It’s allow you to test per page basis, so test as you traverse normally,
    - Following challenges get resolved,
      - Infinite crawl
      - Crawl a site in particular order

Future Approach

- Authenticated scanning – login first and then start testing, context will be managed automatically by browser
  - Following challenges get resolved,
    - Manage context through out the session
    - Logout innocently
    - Where to go after authentication ?
    - Form based authentication, Success or fail, how to decide ?
    - Captcha.

Future Approach

- The field value manipulation will be in a DOM itself.
  - Following challenges get resolved,
    - Building correct attack request
    - Forms submission by “onclick” event
    - Wrong action or target picked up by automated tool
    - Dynamic URL creation
- Java scripts execution automatically,
  - Following challenges get resolved,
    - Dynamic menus and css
    - URL decryption on the fly by java scripts

Future Approach

- False positive will be reduced by real html view,
  - Following challenges get resolved,
    - False positives
    - XSS detection with no false positives, popup will be there.
    - Information leakage can be identified by html view.

Future Approach

- So, only approach is browser based tool, i.e toolbar, like human clicks and automation together!!

- Security QA Toolbar
  http://www.isecpartners.com/SecurityQAToolbar.html

net square

secure.automate.innovate

Thanks!!

umesh@net-square.com

amish@net-square.com