Claims

1. A method of preventing unwanted code execution in a client/server computing environment executing a client-side scripting language and associated scripting environment, said associated scripting environment comprising functions, objects and properties, and their instances, wherein said computing environment comprises at least one server-side resource and a client side resource, wherein the client-side resource requests and receives from the server-side resource at least one message containing a script and executes the script in the associated scripting environment, the method comprising:

determining “safe” and “at risk” or “restricted” portions of the associated scripting environment, wherein a portion of the associated scripting environment comprises any instance of the functions, objects and properties of the scripting environment;

determining at least one “trusted” server-side resource;

determining “trusted” and “untrusted” script received by the client-side resource, wherein “trusted” refers to responses received from the “trusted” server-side resource, and “untrusted” refers to responses received from other external script sources;

receiving from the “trusted” server-side resource at least one further message containing one or more passwords; and

providing an unwanted code execution protection mechanism by~~:~~

re-writing said “at risk” or “restricted” portions of the associated scripting environment to require presentation of the one or more passwords in order to execute;

wherein re-writing of the “at risk” or “restricted” portions of the scripting environment comprises replacing certain objects, functions and property “accessors” to require the passing of the one or more passwords as a parameter in order for execution to occur.

Formatted: Indent: First line: 0 cm

Commented [JC1]: Basis = prev claim 5

2. The method of claim 1, wherein the at least one further message comprises a whole script file.

3. The method of claim 1 or 2, wherein the one or more passwords are reproducible across a number of client-server requests by virtue of a client-server session-identifier or a persistent client-server connection.

4. The method of claim 3, wherein the one or more passwords are produced by the server-side resource as a secure hash of the client-server session-identifier and a session password stored server-side and unique to the client-server session-identifier.

~~5. The method of any of claims 1 to 4, wherein re-writing of the “at risk” or “restricted” portions of the scripting environment comprises replacing certain objects, functions and property “accessors” to require the passing of the one or more passwords as a parameter in order for execution to occur.~~

~~6.~~5. The method of claim ~~5~~1, wherein re-writing of the “at risk” or “restricted” potions of the scripting environment further comprises renaming certain objects, functions and properties to provide each with a new identity.

~~7.~~6. The method of claim ~~6~~5, wherein each new identity is determined by one of the passwords.

~~8.~~7. The method of claim 1, wherein re-writing the “at risk” or “restricted” portions of the scripting environment further comprises marking certain “restricted” objects within the associated environment with one or more passwords as “scope identifier” and replacing certain objects, functions and properties with equivalent functionality that limits access to the associated environment depending on the “scope identifier” and that requires the passing of the one or more “scope identifiers” and the one or more passwords as a parameter in order for execution and subsequent access to occur.

~~9.~~8. The method of claim ~~5~~1, wherein re-writing the “at risk” or “restricted” portions of the scripting language and associated environment further comprises replacing certain objects, functions and properties with equivalent functionality that cascades the protection to newly created portions of the environment or newly created ancillary environments.

~~10.~~9. The method of any preceding claim, wherein the scripting language and associated environment is self-mutable and the re-writing of said “at risk” or “restricted” portions of the scripting language and associated environment comprises using its native functionality to render the changes immutable and the one or more passwords undiscoverable to scripts from the “untrusted” portion.

~~11.~~10. The method of any preceding claim wherein the “untrusted” portion includes an HTML webpage served by a webserver in response to an HTTP or HTTPS GET or POST request; and the “trusted” portion includes a response to separate HTTP requests made to the server-side resource arising from scripting language and associated environment elements embedded in the said HTML webpage.

~~12.~~11. The method of claim ~~11~~10, wherein the one or more passwords are different for each HTML page load by using an additional page identifying parameter submitted to the server-side resource in combination with the client-server session identifier and the session password.

~~13.~~12. The method of any preceding claim, wherein the scripting language is ECMAScript (RTM) (JavaScript (RTM) /Jscript (RTM)) or VBScript (RTM).

~~14.~~13. The method of any preceding claim, wherein the method comprises a method of preventing cross site scripting (XSS) attack.

~~15.~~14. The method of any preceding claim, further comprising testing how effective the unwanted code execution protection mechanism has been or will be in applying the protection by attempting to circumvent the protection with known circumvention techniques.

~~16.~~15. The method of any preceding claim, further comprising reporting activity blocked by the code execution protection mechanism.

~~17.~~16. A client/server computing environment operable to execute a client-side scripting language and associated scripting environment, said associated scripting environment comprising functions, objects and properties, and their instances, wherein said computing environment comprises at least one server-side resource and a client-side resource, wherein the client-side resource and the at least one server-side resource are arranged to carry out any of method claims 1 to ~~16~~15.

~~18.~~17. A computer readable medium containing instructions, which when executed by at least one processor is operable to carry out any of method claims 1 to ~~16~~15.

~~19.~~18. A method substantially as herein described with reference to the accompanying drawings.

~~20.~~19. A computing environment substantially as herein described with reference to the accompanying drawings.