a new way to detect firefox extensions

Author: www.80vul.com [Email:5up3rh3i#gmail.com]
Release Date: 2011/06/10
References: http://www.80vul.com/firefox/detect%20firefox%20extensions.txt

The beginning of this game is from RSnake's <Detecting FireFox Extentions> in 2006,using chrome:// protocol to load extension's resource and to detect the extension. but now,chrome:// protocol can't load extension's resource What are jar packaging.So we need some new ways to do it.

Interact with HTML page's content can help me to achieve it. like some extension register some window functions,and modify the html's stlye ...

some demos

1.detect firebug

```
<img src="chrome://firebug/skin/infoIcon.png" onload="alert("FIREBUG:yes");if
(typeof(loadFirebugConsole)=='undefined') {alert('unfirebuging');}else{alert('firebuging');}" style="visibility:hidden">
```

firebug don't pack by jar ,so first can use "chrome://firebug/skin/infoIcon.png" to detect install,then use window functon like "loadFirebugConsole" to detect using :)

the firebug's codz that to register loadFirebugConsole():

```
getConsoleInjectionScript: function()
    {
        if (!this.consoleInjectionScript)
        {
            var script = "";
            script += "window.__defineGetter__('console', function console() {\n";
            script += " return (window._firebug ? window._firebug : window.loadFirebugConsole()); })\n\n";

            script += "window.loadFirebugConsole = function loadFirebugConsole() {\n";
            script += "window._firebug =  _createFirebugConsole();";

            if (FBTrace.DBG_CONSOLE)
                script += " window.dump('loadFirebugConsole '+window.location+'\\n');\n";

            script += " return window._firebug };\n";

            var theFirebugConsoleScript = getResource("chrome://firebug/content/consoleInjected.js");
            script += theFirebugConsoleScript;


            this.consoleInjectionScript = script;
        }
        return this.consoleInjectionScript;
    },
```

2.detect NOSCRIPT

```
<script>if (typeof(toStaticHTML)=="function"){alert("NOSCRIPT:yes")}</script>
```

hah,Using well-known function toStaticHTML() :)

3. how about requestpolicy

```
<img class="blogimg" small="0" src="http://hiphotos.baidu.com/hi_heige/pic/item/f76803deadeef740632798ff.jpg"
border="0" width="691">
<script>
setTimeout(function(){for(var i=0;i<document.images.length;i++){alert(document.images[i].style.border)}},1000);
</script>
```

when requestpolicy block the remote <img>, it modify the img tags's stlye,the codz like this :

```
        for (var i = 0; i < images.length; i++) {
            var img = images[i];
            // Note: we're no longer checking img.requestpolicyBlocked here.
            if (!img.requestpolicyIdentified && img.src in rejectedRequests) {
                img.requestpolicyIdentified = true;
                img.style.border = "solid 1px #fcc";
                img.style.backgroundRepeat = "no-repeat";
                img.style.backgroundPosition = "center center";
                img.style.backgroundImage = "url('" + this._missingImageDataUri + "')";
                if (!img.width) {
                    img.width = 50;
                }
                if (!img.height) {
                    img.height = 50;
                }
                img.title = "[" + this._rpService.getUriIdentifier(img.src) + "]"
                        + (img.title ? " " + img.title : "")
                        + (img.alt ? " " + img.alt : "");
                img.src = this._transparentImageDataUri;
            }
        }
    },
```