Inspection and Sanitization Guidance for Cascading Style Sheets

Version 1.0

1 March 2012

National Security Agency
9800 Savage Rd, Suite 6721
Ft. George G. Meade. MD 20755

Authored/Released by:
Unified Cross Domain Capabilities Office
cds_tech@nsa.gov

DOCUMENT CHANGE HISTORY

Date	Version	Description
3/01/2012	1.0	Initial Release
12/13/2017	1.0	Updated Contact information, IAC Logo, Cited Trademarks and Copyrights, Expanded Acronyms, and added Legal Disclaimer

DISCLAIMER OF WARRANTIES AND ENDORSEMENT

The information and opinions contained in this document are provided “as is” and without any warranties or guarantees. References herein to any specific commercial product, process, or service by trade name, trademark, manufacturer or otherwise, does not constitute or imply its endorsement, recommendation or favoring by the United States Government and this guidance shall not be used for advertising or product endorsement purposes.

EXECUTIVE SUMMARY

Cascading Style Sheets (CSS2) is a style sheet language that is typically used to specify the presentation of data in a HyperText Markup Language (HTML) document. As this language allows a web designer to precisely arrange content, CSS2 presents many hidden data risks. Data can be moved off the page. Data can be moved over other data. Data can be made invisible to the eye. All of these risks can be mitigated by removing risky rules or replacing risky rule values with benign values, though these might alter the intended display of the page.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

1. SCOPE

1.1 Purpose of this Document

The purpose of this document is to provide guidance for the development of a sanitization and analysis software tool for Cascading Style Sheets (CSS), which may exist within or external to HyperText Markup Language (HTML) documents.

1.2 Document Organization

Section 1 scopes this document. Section 2 details the structure for each CSS construct. Section 3 introduces CSS2, briefly describing what it is, what it's used for, its benefits, the basic syntax of CSS rules, and how CSS rules can be associated with HTML elements. Section 4 highlights several general issues involved with evaluating CSS rules. Section 5 gives the constructs, each of which explains a specific issue, gives one or more examples, and lists the risks involved and recommendations for mitigation.

1.3 Recommendations

The following subsections summarize the categories of recommendation actions that appear in this document and associated options.

1.3.1 Actions

Each construct description lists recommended actions for handling the construct when processing a document. Generally, inspection and sanitization programs will perform an action on a construct: *Validate*, *Remove*, *Replace*, *External Filtering Required*, *Review, or Reject*.

The Recommendation section in each construct lists each of these actions and corresponding applicable explanations of the action to take. It notes if a particular action does not apply, indicates actions that are not part of the standard set of actions (listed in the previous paragraph). For example, a program may choose to reject a file if it is encrypted. Additionally, for some constructs, an action may further break down to specific elements of a construct (e.g., for hidden data in a dictionary's syntax) to give administrators the flexibility to handle specific elements differently.

Recommendations such as Remove and Replace alter the contents of the document. It is important to fix issues where references may be broken or may inadvertently corrupt the document such that it cannot be rendered.

NOTE

The recommendations in this document are brief explanations rather than a How-To Guide. Readers should refer to the construct description or official documentation for additional details.

Table 1-1 summarizes the recommendation actions.

Table 1-1. Recommendation Actions

Recommendation Action	Comments
Validate	Verify the data structure's integrity, which may include integrity checks on other components in the file.(This should almost always be a recommended action)
Replace	Replace the data structure, or one or more of its elements, with values that alleviate the risk (e.g., replacing a user name with a non-identifying, harmless value, or substituting a common name for all authors).
Remove	Remove the data structure or one or more of its elements and any other affected areas.
External Filtering Required	Note the data type and pass the data to an external action for handling that data type (e.g., extract text and pass it to a dirty word search).
Review	Present the data structure or its constructs for a human to review. (This should almost always be recommended if the object being inspected can be revised by a human)
Reject	Reject the file.

NOTE

No recommendations for logging all actions and found data are included here because all activity logging in a file inspection application should occur "at an appropriate level" and presented in a form that a human can analyze further (e.g., the audit information may be stored in any format but must be parsable and provide enough information to address the issue when presented to a human.)

1.3.2 Action Options

The companion to this document, *Data Transfer Guidance for CSS2 Documents*, specifies four options for each recommended action: *Mandatory*, *Recommended*, *Optional*, or *Ignore.* Depending on the circumstances (e.g., a low to high data transfer versus a classified to unclassified transfer), programs can be configured to handle constructs differently.

Table 1-2 summarizes the recommendation action options.

Table 1-2. Recommendation Action Options

Action Options	Comments
Mandatory	For the given direction (e.g., secure private network to unsecure Internet), the file inspection and sanitization program must perform this recommended action.
Recommended	Programs should implement this action if technically feasible.
Optional	Programs may choose to perform or ignore this recommended action.
Ignore	Programs can ignore this construct or data structure entirely

1.3.3 Naming Convention for Recommendations

Recommendations in this document are numbered sequentially, where applicable, and adhere to a standard naming convention identified by a single number *x*, where *x* is a sequential number following by the recommendation keyword defined in Table 1-2. There may be multiple recommendations of the same type, which remain uniquely identified by its number. There is only one type of content under review in this document (i.e., CSS2).

1.4 Data Transfer Guidance

Each format that is documented for inspection and sanitization analysis has a companion document (i.e., the aforementioned DTG document). The DTG serves as a checklist for administrators and others to describe expected behaviors for inspection and sanitization programs. For example, administrators may only remove certain values in a metadata dictionary. Or, the administrator may decide to remove all hidden data if the document is being transferred to a lower security domain.

The DTG gives the administrator the flexibility to specify behaviors for inspection and sanitization programs. The workbook contains a worksheet for each security domain (i.e., the originating domain). Each worksheet lists the numbered constructs from this document and enumerated recommendations in a row. After the recommendations, the worksheet displays a cell for each possible destination domain. This enables an administrator to select the action option for data transfer from the originating domain to the particular destination domain. Each construct row also contains two comment cells: one for low to high transfers and another for high to low transfers.

The recommended actions address two broad risk types: data hiding and data execution. Most data structures are vulnerable to one risk type, while others are susceptible to both risk types. Each construct row in the DTG worksheet contains a cell for designating the risk type (i.e., data execution, data hiding, or both) and another cell for assessing the risk level for that construct (i.e., high, medium and low). This enables administrators to assign the risk type and risk level to each specific construct.

1.5 Document Limitations

This document covers information from the World Wide Web Consortium (W3C) specification on CSS2. The current version of CSS2 is CSS Level 2 Revision 1 (aka CSS2.1), which is a W3C recommendation (as of 7 June 2011).[1]

This document does not cover various CSS profiles, subsets of CSS that are used in specific environments, such as CSS Mobile,[2] which is intended for use on mobile devices. Nor does it cover the next major version of CSS, CSS3, which is a work in progress at this time.

While CSS can be applied to XML documents (using a processing instruction), this is rarely done in practice today, thus this document does not reference this capability. CSS is most commonly used with HTML documents, and that is the focus of this document.

1.5.1 Covert Channel Analysis

It is nearly theoretically impossible to detect or prevent covert channels during communication. It is impossible to identify all available covert channels in any file format. Because these documents contain free-form text, searching for hidden data becomes increasingly difficult. No tool can possibly analyze every channel, so this document highlights the highest risk areas to reduce or eliminate data spills and malicious content.

Additionally, this document does not discuss steganography within block text or media files, such as a hidden message that is embedded within an innocuous image or paragraph. Separate file format filters that specialize in steganography should be used to handle embedded content, such as text, images, videos, and audio when attempting to reduce the risk of covert channels.

[1] http://www.w3.org/TR/CSS2/.

[2] http://www.w3.org/TR/css-mobile/.

1.5.2 Character Encoding

CSS2 provides syntax that is within the American Standard Code for Information Interchange (ASCII) range and is case-insensitive. CSS identifiers contain only alphanumeric characters, ISO 10646 characters U+00A0 and higher, and hyphen (-) and underscore (_). Identifiers cannot start with a digit, two hyphens, or a hypen followed by a digit.

The backslash (\) character can be used as an escape character in three different ways:

1. Backslash inside a string followed by newline is ignored. Outside a string, a backslash followed by newline is a delimeter followed by a newline.
2. Backslash cancels the meaning of special CSS characters. For example, "\"" is a string with one double quote.
3. Character substitution. In this case the backslash character is followed by at a maximum of six hexadecimal digits. The hexadecimal digits imply the Internationl Organization for Standards (ISO) 10646 character with the matching number. For example "\26B" is equivalent to "&B". If the hexadecimal number is invalid, the User Agent should show some kind of visible symbol indicating a missing character.

2. CONSTRUCTS AND TAXONOMY

2.1 Constructs

This document describes many of the constructs used in CSS; it does not describe each construct and this document is not to be treated as a complete reference. Developers of filter software should consult the official documentation from the W3C on CSS alongside this documentation for the full context. Each construct contains the following information:

- **Overview:** a high level explanation of the data structure or element.
- **Concerns:** an explanation of potential problems posed by the element. For example, some metadata elements can cause inadvertent data leakage and others can be used for data exfiltration.
- **Product**: the applicable standard.
- **Location:** provides a textual description of where to find the element in the document. This can vary by client (or product) or it may apply accorss the entire product line.
- **Examples:** if applicable, the definition will contain an example of the construct.
- **Recommendations:** as described in 1.3.

2.2 Taxonomy

The following table describes the terms that appear in this document.

Table 2-1. Document Taxonomy

Term	Definition
Consistency	A construct state in which object information is set to correct values, and that required objects are implemented as defined in the standard.
Construct	An object that represents some form of information or data in the hierarchy of the CSS2/HTML document structure.
DTG	A list of all ISG constructs and their associated recommendations. DTGs are used to define policies for handling every ISG construct when performing inspection and sanitization.
Grammar	A precise, formal, and rigorous syntax for defining constructs.
Inspection and Sanitization	Activities for processing files to prevent inadvertent data leakage, data exfiltration, and malicious data or code transmission
ISG	A document (such as this) that details a file format or protocol and inspection and sanitization activities for constructs within that file format.
Recommendations	A series of actions for handling a construct when performing inspection and sanitization activities.
Referential Integrity	The construct state in which all associated objects are properly referenced in the construct and that construct entries reference existing objects.

3. OVERVIEW OF CSS

CSS 2 is the acronym for Cascading Style Sheets Level 2.[3] It is "a style sheet language that allows authors and users to attach style (e.g., fonts and spacing) to structured documents."[4] It is a style sheet language, which means it describes the presentation for a structured document. It is cascading, which means that any given element in the structured document can be described by multiple style rules; a predefined order of precedence is used to resolve any conflicts.

[3] From this point forward, the abbreviation CSS will be used to refer to CSS 2.

[4] http://www.w3.org/TR/CSS2/.

3.1 Purpose

CSS does not change the content of a document, but it does change how the content is presented to the user, which is commonly referred to as a document's look-and-feel.

3.2 Benefits

CSS allows HTML content to be arranged with precise control, much more control than HTML itself allows.[5]

CSS allows the web designer to separate content from the presentation of content, which simplifies design and debugging and allows multiple types of developers (e.g., graphic designers and programmers) to work simultaneously and independently.

CSS eliminates duplication. For example, instead of repeatedly describing the presentation of each paragraph element within each paragraph, the web designer can describe the presentation of the paragraph element one time. This reduces errors, especially when changes are required. This also speeds up the load time of the page.

CSS makes it easier to maintain a website. The presentation of an entire website can be described in one CSS file; when a change is made to that file, the look and feel of the entire site is automatically changed.

CSS makes it possible to render content differently for different purposes. A CSS style sheet can have one set of rules for displaying the content in a web browser and another set of rules for printing the content.

CSS can decrease the load time of a web site. As web browsers cache external CSS files, the style rules are only loaded once for an entire web site.

3.3 Referenced Documents

The World Wide Web Consortium (W3C) maintains CSS. The most current version of the Level 2 specification can be found here:

http://www.w3.org/TR/CSS2/

[5] For example, absolute positioning allows elements to be placed at any X/Y location.

3.4 Syntax

Each CSS rule is comprised of two parts, a selector and a declaration. The selector is the HTML element that is being styled; the declaration details how the HTML element is to be styled. Each declaration is comprised of two parts, a property name (often just called a property) and a property value (often just called a value). The property name identifies the particular style that is being applied to the selector; the property value specifies the value of the property name. Each property name and value is surrounded by curly brackets and ends with a semicolon. In the following example, “p” is the selector, color is the property, and green is the value.

```
p { color:green; }
```

A CSS rule can group together multiple selectors, which are separated by commas.

```
h1, h2 { color:green; }
```

A CSS rule can have multiple values, which are separated by commas.

```
p { font-family:Arial, sans-serif; }
```

A CSS rule can have multiple property name and value pairs, which are separated by the ending semicolon.

```
p { color:black; background-color:white; }
```

When a group of HTML elements are associated with a class, the CSS selector can specify a style that is unique to that group. The HTML element uses the class attribute.

```
<p class=“inverted”>The World Wide Web Consortium (W3C) is an international
community that develops standards.</p>
```

The CSS rule uses a period and the class name; this is called a class selector.

```
.inverted { color:black; background-color:white; }
```

When an HTML element is associated with an identifier (ID), the CSS selector can specify a style that is unique to that element. The HTML element uses the id attribute.

```
<p id=“footer”>Copyright 2011 John Doe</p>
```

The CSS rule uses a pound symbol; this is called an ID selector.

```
#footer { color:orange; }
```

When HTML elements are descendents of another HTML element, the CSS selector can specify a style that is unique to these descendents. The CSS rule lists the parent element followed by the descendent; this is called a descendent selector or a contextual selector. The following rule is interpreted as, “Set the color to blue for every paragraph that is a descendent of a heading 1 element.”

```
h1 p { color:blue; }
```

CSS comments are contained within a slash-asterisk pair. They can be inserted inside and outside of CSS rules, but they cannot be nested.

```
/* This is a comment. */
```

3.5 Association with HTML

There are four ways to associate CSS rules with HTML elements. The first way is with inline styles. This way inserts the CSS rules directly into HTML elements. This way forfeits most of the benefits of using CSS, such as separation of content and presentation, elimination of duplication, and ease of maintenance.

```
<p style=“color:red;”>The monster is on the loose!</p>
```

The second way is with an internal style sheet. This way inserts CSS rules into the head element of the HTML document. This way also forfeits most of the benefits of using CSS.

```
<html>
  <head>
    <style type=”text/css”>
      h1 p { color:blue; }
      #warning { color:red; }
    </style>
  </head>
…
</html>
```

The third way is with an external style sheet. This ways places all CSS rules into an external file, which is linked to the HTML file. This way takes advantage of all the benefits of using CSS.

```
<html>
  <head>
    <link href=“someStyle.css” type=“text/css” rel=“stylesheet” />
  </head>
…
</html>
```

The fourth way is to use Javascript®[6], which is covered in section 4.1 below.

When validating CSS, all 4 of these places— inline, internal, external, and Javascript code—of these must be checked for risks. Furthermore, CSS code can import additional CSS files, and Javascript code and HTML can both import additional Javascript files. All of these imported files must be checked for risks.

4. CSS GENERAL CONCERNS

The purpose of this section is to bring awareness to some of the issues associated with CSS. This section does not attempt to resolve these issues or give any recommendations for them; it merely highlights them.

[6] Javascript is a registered trademark of Oracle Corp.

4.1 CSS and Javascript

CSS rules can be dynamically added, edited, and deleted using Javascript. This flexibility makes it more difficult to enforce the guidance provided in section 5. Inspection and sanitization software (ISS) must examine not only the CSS rules associated with an HTML page, but also all the Javascript code that can be executed for that page. This section describes four common ways to modify CSS "on-the-fly" using Javascript.

4.1.1 The Style Property

Every HTML element has a style property that can be used to modify CSS rules. First, a specific element in the HTML document must be selected, which is commonly done using one of two functions, getElementById() or getElementsByTagName(). Second, the style property is used to access a CSS rule. For example, this Javascript changes the font on the element with the id of "important:"

```
document.getElementById("important").style.fontFamily = "Arial";
```

This Javascript changes the list style type for all list items:

```
var listItems = document.getElementsByTagName("li");
for (var i=0; i<listItems.length; i++)
{
    listItems[i].style.listStyleType = "square";
}
```

The style property can be used other ways as well. It can set all the properties for an element, superseding any previous properties:

```
element.style.cssText = "letter-spacing:5px; color:red";
```

It can also be used to remove CSS properties. It can remove a single style property from an element:

```
element.style.color = "";
```

It can also remove all style properties for an element:

```
element.style.cssText = "";
```

4.1.2 The StyleSheets Object

The styleSheets object contains an array of all the style sheets that are linked to or embedded within an HTML document. Each style sheet, in turn, contains an array of all the rules within it. The styleSheets object can be used to add, edit, and delete CSS rules. This snippet accesses the second rule in the first style sheet.[7]

```
document.styleSheets[0].cssRules[1]...
```

There are two challenges when using the styleSheets object. One, browsers implement it very inconsistently. Two, it relies upon index values, which may be unknown and may change over time.

The styleSheets object can add new rules:

```
document.styleSheets[0].addRule("p", "font-family:Arial");
document.styleSheets[0].insertRule("p{font-family:Arial;}", 0);
```

It can change existing rules:

```
document.styleSheets[0].cssRules[0].style.setProperty("background-color",
    "yellow", null);
document.styleSheets[0].cssRules[0].style.color = "red";
```

It can also delete existing rules:

[7] Arrays begin at 0.

```
document.styleSheets[0].removeRule(0);
document.styleSheets[0].deleteRule(0);
```

It can import additional CSS rules:

```
document.styleSheets[0].addImport("imported.css");
document.styleSheets[0].insertRule("@import 'external.css';", 0);
document.styleSheets[0].href = "imported.css";
```

It can even disable a style sheet:

```
document.styleSheets[0].disabled = true;
```

4.1.3 The Write Function

The write() function adds elements to an HTML document using text. It can add a link element:

```
document.write("<link rel='stylesheet' type='text/css'
                href='external.css'>");
```

It can add a style element:

```
document.write("<style type='text/css'>body{color:blue;}</style>");
```

It can also add an element with inline styles:

```
document.write("<p style='font-weight:bold'>Foolish people publicize
                folly.</p>");
```

4.1.4 The AppendChild Function

The appendChild() function adds elements to an HTML document similar to the write() function, but it uses objects instead of text. It can add a link element:

```
var link = document.createElement('link');
link.type='text/css';
link.href='external.css';
link.rel='stylesheet';
document.getElementsByTagName('head')[0].appendChild(link);
```

It can add a style element:[8]

```
var style = document.createElement("style");
style.type="text/css";
var styleText = document.createTextNode("body{color:blue;}");
style.appendChild(styleText);
document.getElementsByTagName("head")[0].appendChild(style);
```

It can also add an element with inline styles:[9]

```
var para = document.createElement("p");
para.setAttribute("style", "font-weight:bold");
var paraText = document.createTextNode("Foolish people publicize folly.");
para.appendChild(paraText);
document.getElementsByTagName("body")[0].appendChild(para);
```

4.1.5 Additional Ways

There are other ways that Javascript can be used to change CSS rules, ways that are not covered in this document, such as:

- Using Javascript to create more Javascript that changes the CSS.
- Using Javascript to call other Javascript code that changes the CSS.

[8] Except Microsoft Internet Explorer (IE), which doesn't allow text nodes to be added to style elements.
[9] Except again for IE, which doesn't completely implement the setAttribute function.

- Using the XMLHttpRequest object to retrieve Javascript from a server, and then using eval() to execute it.

- Using obfuscated (e.g., encoded) Javascript that changes the CSS.

- Using Javascript to change the class and/or id of HTML elements, so that a different set of CSS rules apply.

- Using dynamic properties (aka CSS expressions), an IE-specific feature that makes it "possible to declare property values not only as constants, but also as formulas. The formulas used in a dynamic property can reference property values from other elements, thereby allowing authors unique flexibility when designing their Web pages."[10] Dynamic properties were deprecated as of Microsoft Internet6 Explorer®[11] 8 (IE 8).

- Using the HTML Component (HTC) behavior technique, an IE-specific feature that allows CSS to reference an HTC file, a variation of an HTML file that can contain and execute Jscript code, which uses Microsoft's®[12] Dynamic Hyper Text Mark-up Language Application Program Interface (DHTML API) to dynamically manipulate the original HTML document, including its CSS.[13]

- Using the XML Binding Language (XBL) technique, a Firefox®[14]-specific feature that allows CSS to reference an XML file whose contents are written in the XML User interface Language (XUL). XUL can include and execute Javascript, which can manipulate the original HTML document, including its CSS. XBL was deprecated as of Fire Fox 4 (FF 4).[15]

4.2 Obfuscating CSS

Any malicious use of CSS is more difficult to detect when CSS rules are obfuscated. If ISS cannot detect the existence of CSS rules, then it obviously cannot enforce guidance. There are at least two ways to obfuscate CSS.

[10] http://msdn.microsoft.com/en-us/library/ms537634.aspx.

[11] Microsoft Internet Explorer is a registered trademark of Microsoft Corp.

[12] Microsoft is a registered trademark of Microsoft Corp.

[13] http://en.wikipedia.org/wiki/HTML_Components.

[14] Firefox is a registered trademark of Mozilla Foundation.

[15] https://bugzilla.mozilla.org/show_bug.cgi?id=546857.

4.2.1 Obfuscating with External Software

External software can be used to obfuscate CSS rules, but different types of software are used depending upon the location of the rules. If the rules are in CSS code, then one type of software is used. If the rules are in Javascript, then another type is used. And if the rules are in server side code, such as PHP, then yet another type is used.

4.2.1.1 Obfuscating CSS Rules in CSS Code

CSS rules can, of course, be implemented with CSS code, and at least two commercial software tools have a capability to obfuscate CSS code. HTML Guardian by Protware promises that "no portion of your code can be reused by anyone else."[16] This software works by converting CSS rules to Javascript and encrypting the Javascript file. Jasob Obfuscator by jasob.com promises that code "will become impossible to understand thus preventing anyone to steal and modify it."[17] It works by replacing meaningful names with obscure names and removing extraneous characters.

4.2.1.2 Obfuscating CSS Rules in Javascript

CSS rules can be implemented with Javascript, and if they are several software tools are available to obfuscate Javascript code. All of these can be reverse-engineered, but some are difficult and time-consuming. Some software can convert characters to character codes or hex codes; other software can use a shift cipher or other algorithm.[18]

Still other software can minimize, compress, encrypt, or encode Javascript, such as Yahoo User Interface Library (YUI) Compressor from Yahoo®[19], JSMin©[20] from Douglas Crockford, and Thicket™ ®[21]Javascript Obfuscator from Semantic Designs.

4.2.1.3 Obfuscating CSS Rules in Server Side Scripting

In a similar manner, CSS rules can be implemented with server side scripting using PHP or another scripting language, and if they are several software tools are available to obfuscate the code. Examples include Code Eclipse®[22],[23] ionCube's®[24] Pre Hypertext Processor (PHP) Encoder,[25] and Thicket's Obfuscator for PHP.[26]

[16] http://www.protware.com/.

[17] http://www.jasob.com/JavaScript-Obfuscator html.

[18] These techniques can also be done by hand.

[19] Yahoo is a registered trademark of Yahoo Inc.

[20] JSmin is copyright by Douglas Crockford.

[21] Thicket is a unregistered trademark of Semantic Designs.

[22] Eclipse is a registered trademark of Eclipse Foundation.

4.2.2 Obfuscating with the Data URI Scheme

CSS rules can be base64 encoded and then contained in a link element within the HTML page by using the data Uniform Resource Identifier (URI) scheme defined by Internet Engineering Task Force (IETF) standard Request for Comment (RFC) 2397. The rules cannot be examined unless they are decoded.

4.3 Validation

Web browsers ignore invalid CSS rules, which might allow sensitive data to be embedded within CSS. This issue can be resolved using a validator, such as the one offered by the W3C:

http://jigsaw.w3.org/css-validator/

It can support multiple various "flavors" of CSS (i.e., CSS2, CSS3, the mobile CSS profile, etc.). It can also be downloaded and run locally. Unfortunately, it is also possible to fool this validator. At the time of writing, the following line passed validation without any errors:

```
p {color:white;}This is sensitive data that I am hiding in the CSS file.body
{background-color:black;}
```

Other bugs in the W3C's CSS Validation Service can be found on the W3C's bug/issue tracking page.[27]

4.4 Encoding Data in CSS

Data can be encoded and then embedded in a CSS file by using the data URI scheme. This can be done, for example, with a background image or a cursor icon.

```
div.apple
{
    height:600px;
```

[23] http://www.codeeclipse.com/.
[24] IonCube is a registered trademark of IonCube LTD.
[25] http://www.ioncube.com/.
[26] http://www.semanticdesigns.com/Products/Obfuscators/PHPObfuscator.jsp.
[27] http://www.w3.org/Bugs/Public/buglist.cgi?product=CSSValidator.

```
    width:600px;
    background-image:url(data:image/png;base64,/9j/4AAZJgAB...ICAgICgIP//Z);
    background-repeat:no-repeat;
}
```

Encoding makes it difficult to enforce guidance on the CSS rules. A human reviewer cannot visually examine the encoded data—or even know what it is—until it has been unencoded. Software must be capable of detecting and unencoding the data URI scheme.

4.5 CSS Escaping Characters

Characters can be escaped in CSS using the backslash character or the character code from ISO 10646. The syntax allows for two options:

1. The backslash followed by exactly 6 hexadecimal digits. For example, to set the color of a paragraph to blue, these are equivalent:

```
p { color:blue; }
p { color:\000062\00006C\000075\000065; }
```

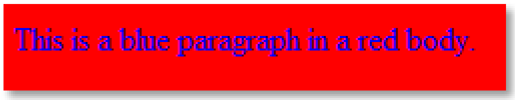
2. The backslash followed by all non-zero hexadecimal digits followed by a space. For example, to set the background color of the body, these are equivalent:

```
body            { background-color:red; }
\62 \6F \64 \79 { background-color:red; }
```

Here’s a complete example of an HTML document with embedded CSS rules that contain escaped characters:

```
<html>
    <head>
        <style type="text/css">
            \62 \6F \64 \79 { background-color:red; }
            p { color:\000062\00006C\000075\000065; }
        </style>
    </head>
    <body>
        <p>This is a blue paragraph in a red body.</p>
    </body>
</html>
```

The result is rendered like this:



This is a blue paragraph in a red body.

Figure 4-1. Escaped Characters

Backslashes in property names or property values that are not followed by character codes are ignored, so long as they are not the first letter of the name or value. These rules are valid and equivalent:

```
p { color:blue; }
p { c\ol\or:bl\ue; }
```

Escaping makes it difficult to enforce guidance, because it's hard to identify and evaluate a CSS rule with escaped characters. A human reviewer cannot visually examine the escaped data until it has been unescaped. Software must be capable of detecting and unescaping escaped data.

4.6 CSS with External Servers

Some CSS elements, notably the background image property and the import rule, can pull data from external servers. Both external images and additional CSS rules should be inspected and sanitized in accordance with appropriate guidance. Data on external servers cannot be examined until it is first retrieved.

```
p { background-image:url("http://evilwebsite.com/picture.jpg"); }
@import url("http://www.external.com/moreStyles.css");
```

5. CSS CONSTRUCTS

This section discusses specific features and risks of CSS2. Each section provides examples and a description, as well as the recommendations for handling each feature or issue.

CSS 5.1 Parsing Errors

OVERVIEW:

CSS parsers are required to ignore some parsing errors, acting as if they do not exist. For example, if a declaration has an unknown property or an invalid value, the browser should ignore it. If the declaration is malformed or has malformed statements, the browser should ignore it. If a declaration has an invalid at-keyword, the browser should ignore everything in its block.[28]

CONCERNS:

Hidden data risk – When invalid CSS is ignored by the parser, it is a hidden data risk.

PRODUCT:

- CSS2

LOCATION:

Parsing errors can exist throughout the CSS information in the document.

EXAMPLE:

The `sensitive` property, an unknown property name, and its value are ignored:

```
<style type="text/css">
   p { sensitive:thisIsVerySensitiveData; }
</style>
```

The property value `thisIsVerySensitiveData,` which is invalid, is ignored:

```
<style type="text/css">
   p { color:thisIsVerySensitiveData; }
</style>
```

The declaration `thisIsVerySensitiveData` is missing a colon and a value, is malformed, and thus is ignored:

```
<style type="text/css">
   p { color:green; thisIsVerySensitiveData }
</style>
```

[28] http://www.w3.org/TR/CSS2/syndata.html#parsing-errors.

The statement `p` has an unexpected at-keyword, `@thisIsVerySensitiveData,` is malformed, and thus is ignored:

```
<style type="text/css">
   p @thisIsVerySensitiveData;
</style>
```

Because the at-keyword `thisIsVerySensitiveData` is unknown, everything in the block is ignored:

```
<style type="text/css">
   @thisIsVerySensitiveData { sensitive:ThisIsSensitiveData;
                              moreSensitive:ThisIsMoreSensitiveData; }
</style>
```

RECOMMENDATIONS:

1. **Validate:** Validate all CSS rules, including properties and values, before processing.

2. **Remove:** If a CSS rule is invalid, strip out the invalid portions of the rule to make it valid. If this is not possible, strip out the entire rule; if valid portions of the rule are stripped out with the invalid portions, the content may not be styled as expected.

3. **Replace:** N/A

4. **External Filtering Required:** N/A

5. **Review:** N/A

6. **Reject:** Reject the entire document if it contains any invalid CSS.

CSS 5.1: END

CSS 5.2 Comments

OVERVIEW:

Programmers use comments to explain their code, but any information can be put into a comment.[29] Comments are not displayed, thus they could be used to hide or disclose text.

[29] http://www.w3.org/TR/CSS2/syndata html#comments.

CONCERNS:

Hidden data risk – All comments are free text and are not displayed, thus comments are a hidden data risk.

Data disclosure risk – If developers inserted comments describing why they made certain choices, they could inadvertently reveal information about the system using the CSS. This could be a data disclosure risk.

PRODUCT:

- CSS2

LOCATION:

Comments are located throughout the document and are contained within the markers `/*` and `*/`.

EXAMPLE:

This CSS comment is not displayed:

```
<style type="text/css">
   /* This is sensitive data that is not displayed. */
</style>
```

RECOMMENDATIONS:

1. **Validate:** N/A

2. **Remove:** Remove all comments, including the comment markers (`/*` and `*/`)

3. **Replace:** Replace comment text with benign text or an empty string.

4. **External Filtering Required:** Extract text from comments and send to an external filter.

5. **Review:** N/A

CSS 5.2: END

CSS 5.3 The Import Rule

OVERVIEW:

The `@import` rule allows style rules to be imported from other style sheets.[30] The URI can be absolute, which would allow the import of style sheets from malicious sites with style rules that contain the various types of risks described in this paper.

CONCERNS:

Hidden data risk – Many CSS capabilities can be used to hide data, thus the imported CSS rules might contain additional hidden data risks.

Data disclosure risk – Many CSS capabilities can be used to inadvertently hold sensitive data, thus the imported CSS rules might contain additional data disclosure risks. Additionally, the URI to the imported style sheet might be sensitive, which is a data disclosure risk.

PRODUCT:

- CSS2

LOCATION:

The `import` rule can be located inline, internally, or externally.

EXAMPLE:

This HTML file imports a style sheet from a malicious site:

```
<style type="text/css">
   @import url('http://www.evil.org/stylesheet.css');
</style>
```

RECOMMENDATIONS:

1. **Validate:** Validate the URI against a known list of approved URIs.

2. **Remove:** Remove import rules with absolute URIs. If the CSS is removed, the content may not be styled as expected.

3. **Remove:** Remove any import rule that imports a CSS file that has recursive imports (a CSS file that attempts to import itself).[31]

4. **Replace:** Replace external URIs with an equivalent relative address. If the style sheet does not exist locally, then the content may not be styled as expected.

5. **External Filtering Required:** Pass URI data to an external filter.

[30] http://www.w3.org/TR/CSS2/cascade html#at-import.

[31] A number of vulnerabilities have been reported when style sheets import recursively; for example, it has caused denial of service and memory corruption.

6. **Review:** N/A

CSS 5.3: END

CSS 5.4 The Margin Property

OVERVIEW:

The `margin` property specifies the width of the margin area of a box.[32] If the margins exceed the limits of the page, an element can be moved off the page and thus not displayed. When a margin moves an element into the space occupied by another element, one of them could be partially or completely obstructed.

CONCERNS:

Hidden data risk – When the margin moves an element off the page, it is a hidden data risk. When the margin moves an element into the space of another element, it is a hidden data risk.

PRODUCT:

- CSS2

LOCATION:

Margins can be located inline, internally, or externally.

EXAMPLE:

If the left margin is set to a large negative value, the paragraph is not displayed:

```
<style type="text/css">
   p.movedData { margin-left:-500px; }
</style>
```

RECOMMENDATIONS:

1. **Validate:** Verify that the margins do not position any part of an element off the page and or over another element.

2. **Remove:** Remove the `margin` property, and the element will have margins of 0 width and be displayed at its default location.

[32] http://www.w3.org/TR/CSS2/box html#margin-properties. Though it may not appear so, every element in HTML is bounded by an invisible box; many CSS properties, including margin, border, and property, alter the display of the element at this boundary.

3. **Replace:** Set the margin values to 0, or set them to some value that ensures the entire element is on the page and is not positioned over another element.

4. **External Filtering Required:** N/A

5. **Review:** N/A

CSS 5.4: END

CSS 5.5 The Display Property

OVERVIEW:

The `display` property specifies the type of box an element generates, thus determining if and where an element is displayed.[33] If a developer wants to hide an element, he can set the display property to "none," and the element will not appear, and layout will not be affected; descendent elements do not generate boxes either.[34]

CONCERNS:

Hidden data risk – As `display` can remove data, it is by definition a hidden data risk.

PRODUCT:

- CSS2

LOCATION:

The `display` property can be located inline, internally, or externally.

EXAMPLE:

When the display of a paragraph is set to `none,` it is not displayed:

```
<style type="text/css">
   p.notDisplayed { display:none; }
</style>
```

RECOMMENDATIONS:

1. **Validate:** N/A

2. **Remove:** If display is set to "none," remove it, and the element will be displayed inline.

3. **Replace:** If display is set to "none," replace it with "inline," and the element will be

[33] For example, boxes can be block, inline, a list item, or a table element.

[34] http://www.w3.org/TR/CSS2/visuren.html#display-prop.

displayed inline or replace it with “block,” and the element will be displayed in a block.

4. **External Filtering Required:** N/A

5. **Review:** N/A

CSS 5.5: END

CSS 5.6 The Position Property

OVERVIEW:

The position property is used to place an element at a specific location on the page. If position is set to “relative,” “absolute,” or “fixed,” one or more box offset properties are used to calculate the exact position. When the offsets exceed the limits of the page, an element can be moved off page, thus it is not displayed. When the offset moves an element into the space occupied by another element, one of them could be partially or completely obstructed.

CONCERNS:

Hidden data risk – When the offsets move an element off the page, position is a hidden data risk. When the offsets move an element into the space of another element, position is a hidden data risk.

PRODUCT:

- CSS2

LOCATION:

The position property can be located inline, internally, or externally.

EXAMPLE:

When the position is set to “absolute” and the top is set to a large negative value, the paragraph is not displayed.

```
<style type="text/css">
   p.movedData { position:absolute; top:-500px;}
</style>
```

(There is a related set of properties that are Microsoft proprietary: pixelTop, pixelLeft, pixelRight, and pixelBottom. These are not supported in every web browser, but they are supported in IE.)

If the `position` properties for two elements are set to `absolute` and the offset properties are identical, then one element will be over the other. If an image is over text, then it will hide the text.

```
<html>
    <head>
        <style type="text/css">
            p.underImage { position:absolute; top:50px; left: 50px;
                           width:250; }
            img { position:absolute; top:50px; left: 50px; }
        </style>
    </head>
    <body>
        <p>This is non-sensitive data that is displayed normally</p>
        <p class="underImage">This is sensitive data that is not
                              displayed.</p>
        <img src="w3c.jpeg"/>
    </body>
</html>
```

A variation of this threat is to give an empty element, such as a span, a background property, set its URL to an image, and then position the element over the text.[35]

RECOMMENDATIONS:

1. Validate: Verify that the offsets do not position the element off the page or over another element.

2. Remove: Remove the `position` property, and the element will be statically positioned. Although the offset properties should be ignored by the browser when there is no position property, remove them to avoid confusion.

3. Replace: Set the values of the offsets to 0, and the element will be on the page. Set the offsets to values that ensure the entire element is on the page and not over another element. Set the value of the position element to “static;” although the offset properties should be ignored by browser when the position property is set to “static,” remove them to avoid confusion.

4. External Filtering Required: N/A

5. Review: N/A

CSS 5.6: END

[35] http://wellstyled.com/css-replace-text-by-image.html.

CSS 5.7 The Overflow Property

OVERVIEW:

The `overflow` property tells the browser what to do with content that doesn't fit in the box.[36] If the box is intentionally made too small for the content (e.g., set width to `0px`) and the overflow property set to "hidden," then the overflowing text will be hidden from the user.

CONCERNS:

Hidden data risk – When overflowing text is hidden from the user, the `overflow` property is a hidden data risk.

PRODUCT:

- CSS2

LOCATION:

The `overflow` property can be located inline, internally, or externally.

EXAMPLE:

If the width of a paragraph is 0 and the overflow is hidden, then the text is not displayed.

```
<style type="text/css">
   p.overflowing { width:0px; overflow:hidden; }
</style>
```

RECOMMENDATIONS:

1. **Validate:** N/A

2. **Remove:** Remove overflow properties with the "hidden" value.

3. **Replace:** Replace all overflow properties with the value of "hidden" to "visible."

4. **External Filtering Required:** N/A

5. **Review:** N/A

CSS 5.7: END

[36] http://www.w3.org/TR/CSS2/visufx html#overflow.

CSS 5.8 The Clip Property

OVERVIEW:

The `clip` property is used to limit some portion of an element's border box, thus making part of it invisible.[37] It only applies to elements that are absolutely positioned. The purpose of clipping is to hide some or all of an element, whether text or an image.

CONCERNS:

Hidden data risk – If the clip hides some or all of the element, clip is a hidden data risk.

PRODUCT:

- CSS2

LOCATION:

The clip property can be located inline, internally, or externally.

EXAMPLE:

If the area of the clip is set to 0, then the element—a paragraph in this example—is not displayed:

```
<style type="text/css">
   p.clipped { position:absolute; clip:rect(0px,0px,0px,0px); }
</style>
```

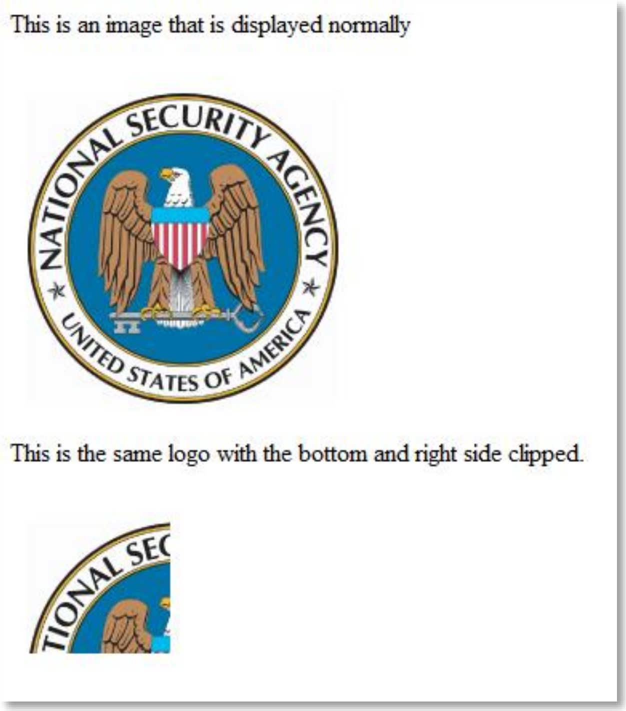
If the area of the clip is larger than 0, then the element—a picture is in this example—is only partially displayed:

```
<html>
    <head>
        <style type="text/css">
            img.clipped { position:absolute;
                          clip:rect(0px 100px 100px 0px); }
        </style>
    </head>
    <body>
        <p>This is an image that is displayed normally</p>
        <img src="nsa_logo.jpg">
        <p>This is the same logo with the bottom and
           right side clipped.</p>
```

[37] http://www.w3.org/TR/CSS2/visufx html#propdef-clip.

```
        <img src="nsa_logo.jpg" class="clipped">
    </body>
</html>
```

This example is rendered like this:



This is an image that is displayed normally

This is the same logo with the bottom and right side clipped.

Figure 5-1. Clipped Picture

RECOMMENDATIONS:

1. **Validate:** N/A

2. **Remove:** Remove the `clip` property.

3. **Replace:** Replace the value of the clip to "auto," and the element will be fully displayed.

4. **External Filtering Required:** N/A

5. **Review:** N/A

CSS 5.8: END

CSS 5.9 The Visibility Property

OVERVIEW:

The `visibility` property specifies whether the browser should render the box generated by an element. If the value is “hidden,” the box is invisible, though it still affects layout. If the value is “collapse” and used on a table row, row group, column, or column group, the entire section is removed and layout is not affected. If the value is “collapse” and used on any other element, it has the same meaning as hidden.[38]

This property does not appear to be implemented consistently in browsers when the value is “collapse” and used on a table row, row group, column, or column group. On some browsers the layout was affected (i.e., the table element was removed), but in others it was not (i.e., the table element was merely hidden); in all cases, the text was not rendered.

CONCERNS:

Hidden data risk – As visibility hides data, it is by definition a hidden data risk.

PRODUCT:

- CSS2

LOCATION:

The `visibility` property can be located inline, internally, or externally.

EXAMPLE:

When the visibility of a paragraph is set to “hidden,” it is invisible.

```
<style type="text/css">
   p.hiddenData { visibility:hidden; }
</style>
```

When the visibility of a table row is set to “collapse,” the row is either invisible or removed (depending upon the browser).

```
<style type="text/css">
   tr.collapsedData { visibility:collapse; }
</style>
```

When the visibility of a row group is set to “collapse,” the rows are either invisible or removed

[38] http://www.w3.org/TR/CSS2/visufx html#visibility; http://www.w3.org/TR/CSS2/tables html#dynamic-effects.

(depending upon the browser).

```
<style type="text/css">
   tbody.collapsedData { visibility:collapse; }
</style>
```

When the visibility of a column is set to "collapse," the column is either invisible or removed (depending upon the browser), or it may not be affected at all, as some browsers do not support the `col` element and thus do not hide the collapsed column.

```
<style type="text/css">
   col.collapsedData { visibility:collapse; }
</style>
```

When the visibility of a column group is set to "collapse," the columns are either invisible or removed (depending upon the browser), or they may not be affected at all, as some browsers do not support the `colgroup` element and thus do not hide the collapsed columns.

```
<style type="text/css">
   colgroup.collapsedData { visibility:collapse; }
</style>
```

When the visibility of a non-table element, such as a paragraph, is set to "collapse," it is invisible.

```
<style type="text/css">
   p.collapsedData { visibility:collapse; }
</style>
```

RECOMMENDATIONS:

1. **Validate:** N/A

2. **Remove:** Remove the `visibility` property.

3. **Replace:** F or the `visibility` property, replace values of "hidden" or "collapse" with "visible."

4. **External Filtering Required:** N/A

5. **Review:** N/A

CSS 5.9: END

CSS 5.10 The Page Rule

OVERVIEW:

The content of an HTML document is typically viewed within a web browser, but this content can be repurposed and viewed in other formats, including the printed page. The `@page` rule specifies instructions for displaying HTML content on other formats that (unlike web browsers) have distinct pages (aka paged media), such as paper or transparencies. This rule does not impact how content is viewed in a web browser, but it does impact how it is viewed in media such as a Portable Document Format (PDF) file or a printed page.[39] The `@page` rule can specify margin boundaries; much like margins for viewing in the browser, these margins can be set to negative values, which can cause data to be off the page and thus not displayed. Not all browsers support this rule.

CONCERNS:

Hidden data risk – If content is reviewed by first printing it out, and if the margins are set to negative values, then the `@page` rule is a hidden data risk.

PRODUCT:

- CSS2

LOCATION:

The `@page` rule can be located inline, internally, or externally.

EXAMPLE:

With a negative margin, the data in this HTML file is not displayed on the page.

```
<style type="text/css">
   @page { size:auto; margin:-3in; }
</style>
```

RECOMMENDATIONS:

1. **Validate:** Verify that the margins do not move the elements off the page.

[39] http://www.w3.org/TR/CSS2/page html#page-intro.

2. **Remove:** Remove the margin property in `@page` rules, and the element will have margins within the page.

3. **Replace:** Replace the margin property in `@page` rules with a value of 0, and the element will have margins within the page.

4. **External Filtering Required:** N/A

5. **Review:** N/A

CSS 5.10: END

CSS 5.11 The Color Property

OVERVIEW:

The `color` property specifies the foreground color of an element's text content.[40] If the color of the text is set to match the background color of any containing element, the text will not be visible (unless selected by the user).

CONCERNS:

Hidden data risk – When text cannot be read because its color blends in with the background, the color property is a hidden data risk.

PRODUCT:

- CSS2

LOCATION:

The `color` property can be located inline, internally, or externally. It applies to all elements (though it only impacts text content).

EXAMPLE:

If the background of the body element and the color of a paragraph are both set to "white," then the text is not visible.

```
<style type="text/css">
   body { background-color:white }
```

[40] http://www.w3.org/TR/CSS2/colors.html#colors.

```
   p.colorless { color:white }
</style>
```

A variation of this is to set the text and background to nearly matching colors.

If a paragraph has an image for a background, and the color of the paragraph's text matches the color of the image, then the text is not visible.

```
<style type="text/css">
   p.imageColor { color:#CDB793;
                  background-image:url("CDB793.png")
                  no-repeat; }
</style>
```

A variation of this is to set the text and background image color to nearly matching colors.

RECOMMENDATIONS:

1. **Validate:** N/A

2. **Remove:** Remove images that are used as backgrounds for text.

3. **Remove:** Remove all color and background color properties. If the content requires color for meaning (e.g., errors are colored with red text), then some meaning will be lost.

4. **Replace:** Replace all foreground and background colors such that the text is visible (e.g., set all color properties for text to "black" and all background color properties for text to "white"). If the content requires color for meaning (e.g., errors are colored with red text), then some meaning will be lost.

5. **External Filtering Required:** If the element has a background image, send it to an external filter for the corresponding image type.

6. **Review:** A human reviewer should be able to see text hidden in this manner (i.e., text color matches text background color) by choosing "Select All" from the Edit menu in the web browser.

CSS 5.11: END

CSS 5.12 The Background Image Property

OVERVIEW:

Image elements can have a background, and this background can be an image. The `background-image` property selects the image.[41]

CONCERNS:

Hidden data risk – When one image is beneath another, it is a hidden data risk.

PRODUCT:

- CSS2

LOCATION:

The `background-image` property can be located inline, internally, or externally. It is one of the valid background properties that apply to all elements.

EXAMPLE:

If a background image is positioned directly beneath another image of the same size, the background image will not be displayed:

```
<html>
    <head>
        <style type="text/css">
            img { background-image:url("sensitivePicture.png"); }
        </style>
    </head>
    <body>
        <img src="normalPicture.png"/>
    </body>
</html>
```

RECOMMENDATIONS:

1. Validate: N/A

2. Remove: If an image element has a background image, remove the background image. If this feature was used to overlay one image upon another (e.g., put a star on a map to indicate position), then removing an image may cause content loss.

3. Replace: N/A

4. External Filtering Required: Send all images, whether in elements or image backgrounds, to an external filter for the corresponding image type.

5. Review: If external software could manipulate the content such that nothing obscures the background image, then it could be reviewed by a human reviewer.

CSS 5.12: END

[41] http://www.w3.org/TR/CSS21/colors.html#background.

CSS 5.13 The Background Position Property

OVERVIEW:

Elements can have a background, and this background can be an image. The `background-position` property sets its initial position.[42] One way to specify position is with x-y coordinates, and if these are large numbers (negative or positive), then the background image can be off screen, and thus not displayed.

CONCERNS:

Hidden data risk – When the `background-position` property is used to position images off screen, it is a hidden data risk.

PRODUCT:

- CSS2

LOCATION:

The `background-position` property can be located inline, internally, or externally. It applies to all elements.

EXAMPLE:

If a background image is positioned off screen, it will not be displayed:

```
<style type="text/css">
   body { background-image:url("sensitivePicture.png");
          background-position:-500px -500px;
          background-repeat:no-repeat;}
</style>
```

RECOMMENDATIONS:

1. **Validate:** Verify that the `background-position` does not move the image off the page.

2. **Remove:** If an element has a background image, remove the `background-position` property, and the image will be displayed on the page.

3. **Replace:** If the background position has a negative `x` or `y` coordinate, set those coordinates to 0, and the image will remain on screen.

4. **External Filtering Required:** Send background images to an external filter for the corresponding image type.

[42] http://www.w3.org/TR/CSS21/colors.html#background.

5. **Review:** N/A

CSS 5.13: END

CSS 5.14 The Background Attachment Property

OVERVIEW:

Elements can have a background, and this background can be an image. The `background-attachment` property determines whether the image is fixed or scrolls with the page.[43] If an element has a fixed background image, that image can be used to hide other elements.

CONCERNS:

Hidden data risk – When the `background-attachment` property is set to "fixed," it can be a hidden data risk.

PRODUCT:

- CSS2

LOCATION:

The `background-attachment` property can be located inline, internally, or externally.

EXAMPLE:

If a fixed image is a solid color, then text can be positioned over it and hidden.

```
<style type="text/css">
   body { background-image:url("CDB793.png");
          background-attachment:fixed;
          background-repeat:no-repeat; }
   p.fixedBackground { position:absolute;
                       top:15px; left:15px;
                       color:#CDB793; }
</style>
```

RECOMMENDATIONS:

1. **Validate:** N/A

2. **Remove:** If the `background-attachment` property set to "fixed," remove the property and its value, and the image will scroll with its element.

3. **Replace:** If the `background-attachment` property has the value "fixed," replace it with

[43] http://www.w3.org/TR/CSS21/colors.html#background.

the value "scroll," and the image will scroll with its element.

4. External Filtering Required: Send background images to an external filter for the corresponding image type.

5. Review: If external software could manipulate the content such that nothing obscures the background image, then it could be reviewed by a human reviewer.

CSS 5.14: END

CSS 5.15 The Font-Size Property

OVERVIEW:

The `font-size` property specifies the size of the font used by a text element.[44] If the size is set to a small number, then the text will be very difficult to see. If the size is set to zero or to a negative number, then it will be hidden from the user.

CONCERNS:

Hidden data risk – When the font size is small, zero, or negative, it is a hidden data risk.

PRODUCT:

- CSS2

LOCATION:

The font size property can be located inline, internally, or externally.

EXAMPLE:

If the font size of a paragraph is set to 0, then the paragraph is not displayed.

```
<style type="text/css">
   p.small { font-size:0px; }
</style>
```

RECOMMENDATIONS:

1. Validate: Verify that the `font-size` property is at or above a threshold to ensure its visibility.

2. Remove: Remove the `font-size` property from text elements, and the text will be displayed at the default size.

3. Replace: If the `font-size` property is below a threshold (e.g., 16px or 12pt), set it to the

[44] http://www.w3.org/TR/CSS2/fonts html#font-size-props.

threshold.

4. **External Filtering Required:** N/A

5. **Review:** N/A

CSS 5.15: END

CSS 5.16 The Text Indent Property

OVERVIEW:

The `text-indent` property specifies the indentation of the first line of text in a block container.[45] If the indentation exceeds the limits of the page negatively (to the left), an element can be moved off the page, thus it is not displayed. This is not an issue with positive numbers, because the scroll bar can be used to scroll over and see text to the right, unless text indent is combined with a narrow height and hidden overflow.

CONCERNS:

Hidden data risk – When indentation moves an element off the page, it is a hidden data risk.

PRODUCT:

- CSS2

LOCATION:

The `text-indent` property can be located inline, internally, or externally.

EXAMPLE:

If the text indent is set to a large negative value, the paragraph is not displayed.

```
<style type="text/css">
   p.movedData { text-indent:-5000px; }
</style>
```

If the text indent is set to a large positive value, the height is set small, and the overflow is set to “hidden,” then the paragraph is not displayed:

```
<style type="text/css">
   p.movedData { text-indent:2000px; height:20px;
                 overflow:hidden }
</style>
```

[45] http://www.w3.org/TR/CSS2/text html#indentation-prop.

RECOMMENDATIONS:

1. **Validate:** Verify that the indentation does move the element off the left or right side of the page.

2. **Remove:** Remove the `text-indent` property, and the text will begin at the start of the block.

3. **Replace:** Set the text indent value either to 0 or to some appropriately small number, such as 20 pixels, and the the text will begin either at or near the start of the block.

4. **External Filtering Required:** N/A

5. **Review:** N/A

CSS 5.16: END

CSS 5.17 The Letter-Spacing, Word-Spacing, and Line-Height Properties

OVERVIEW:

The `letter-spacing` property specifies the space between letters; the `word-spacing` property specifies the space between words; and the `line-height` property specifies the space between lines. If the `letter-spacing` and `word-spacing` properties are set to a very large distance, then only the first letter or first word of an element will be displayed, thus mostly hiding the text. If these properties are set to a negative distance, then letters and words can overlap, thus hiding the text. If the `line-height` property is set to 0 while the font-size is set very large, lines can overlap each other, thus obscuring the text.

CONCERNS:

Hidden data risk – When a large value for letter or word spacing moves most of a text element off the page, it is a hidden data risk. When a negative value for letter or word spacing moves text over itself or off the page, it is a hidden data risk. When a value of 0 for line height causes lines to overlap, it is a hidden data risk.

PRODUCT:

- CSS2

LOCATION:

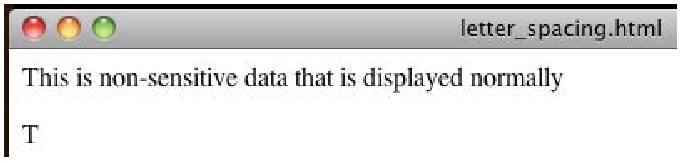
Letter spacing, word spacing, and line height can be located inline, internally, or externally.

EXAMPLE:

If the `letter-spacing` property is set to 5000 pixels, the height is limited to 20 pixels, and the overflow is "hidden," then only the first letter of the paragraph will be displayed. The `height` property prevents wrapping, and the `overflow` property disables the scrollbar.

```
<html>
    <head>
        <style type="text/css">
            p.spaced { letter-spacing: 5000px; overflow:hidden;
                       height:20px;}
        </style>
    </head>
    <body>
        <p>This is non-sensitive data that is displayed normally</p>
        <p class="spaced">This is sensitive data that is not
                          displayed.</p>
    </body>
</html>
```

The result looks like this:



This is non-sensitive data that is displayed normally

T

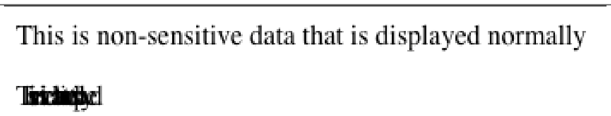
Figure 5-2. Large Letter Spacing Example

The `word-spacing` property would achieve similar results, except that the entire first word (i.e., “This”) would be displayed.

If the `letter-spacing` property is set to -5px, then the letters appear on top of each other and obfuscate the text. A similar effect occurs when using word spacing with a negative value.

```
<html>
    <head>
        <style type="text/css">
            p.shrunk { letter-spacing: -5px; }
        </style>
    </head>
    <body>
        <p>This is non-sensitive data that is displayed normally</p>
        <p class="shrunk">This is sensitive data that is not
                          displayed.</p>
    </body>
</html>
```

The result looks like this:



This is non-sensitive data that is displayed normally

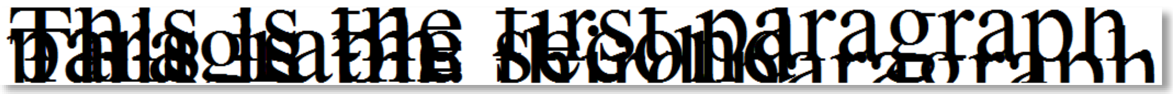
Figure 5-3. Negative Letter Spacing Example

If the `letter-spacing` property is set to an even larger negative number, such as `-50px`, then it disappears from display altogether. A similar effect occurs when using word spacing with a negative value.

If the `line-height` property is set to 0 and the font size to a large number, then web browsers will display lines as overlapping.

```
<html>
    <head>
        <style type="text/css">
            p { line-height: 0; font-size: 100; }
        </style>
    </head>
    <body>
        <p>This is the first paragraph.</p>
        <p>This is the second paragraph.</p>
        <p>This is the third paragraph.</p>
    </body>
</html>
```

All four web browsers that were tested displayed the result slightly differently. The most obscured result looked like this:



This is the first paragraph

Figure 5-4. Data Obscured by Line Height

RECOMMENDATIONS:

1. **Validate:** Validate that the letter spacing values are below some upper threshold (e.g., 10) and above some lower threshold (e.g., 1).

2. **Validate:** Validate that the word spacing values are below some upper threshold (e.g., 100) and above some lower threshold (e.g., 1).

3. **Validate**: Validate that the line height is above 0.

4. **Remove:** Remove the letter and word spacing and the line height attributes and the text will be visible and readable at the default spacing.

5. **Replace:** If the letter spacing is above some upper threshold (e.g., 10), setting it to the upper threshold will make the text visible. If the letter spacing is below some lower threshold (e.g., 1), setting it to the lower threshold will make the text visible.

6. **Replace**: If the word spacing is above some upper threshold (e.g., 100), setting it to the upper threshold will make the text visible. If the word spacing is below some lower threshold (e.g., 1), setting it to the lower threshold will make the text visible.

7. **Replace**: If the line height is below 1, replace the value with it with the value of 1.

8. **External Filtering Required:** N/A

9. **Review:** If external software could manipulate the content such that the spacing no longer obscures the text, then it could be reviewed by a human reviewer.

CSS 5.17: END

CSS 5.18 The Outline Property

OVERVIEW:

The `outline` property is used to create outlines around elements. One way that they are different from borders is that they do not take up space, which means that an outline can overlap nearby elements, thus hiding data.[46]

CONCERNS:

Hidden data risk – If an outline covers another element, outline will be a hidden data risk.

PRODUCT:

- CSS2

LOCATION:

[46] http://www.w3.org/TR/CSS21/ui html#dynamic-outlines.

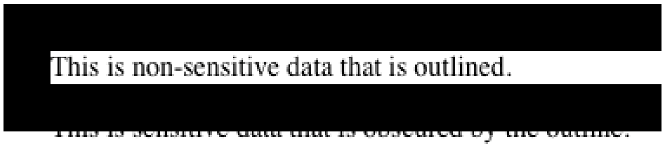
The `outline` property can be located inline, internally, or externally.

EXAMPLE:

If the outline of the first paragraph is sufficiently large, the second paragraph can be obscured or even completely hidden.

```
<html>
    <head>
        <style type="text/css">
            body { margin: 40px; }
            p.outlined { outline-width:27px;
                         outline-color:black;
                         outline-style:solid; }
        </style>
    </head>
    <body>
        <p class="outlined">This is non-sensitive data that is
                            outlined.</p>
        <p>This is sensitive data that is obscured by the outline.</p>
    </body>
</html>
```

The result looks like this:



This is non-sensitive data that is outlined.

This is sensitive data that is obscured by the outline.

Figure 5-5. Outlined Data

If the `outline-width` is increased to 30 pixels, the second paragraph is completely hidden.

RECOMMENDATIONS:

1. **Validate:** Verify that the `outline-width` does not exceed the margin width.

2. **Remove:** Remove the outline property, and it will not obstruct other elements.

3. **Replace:** Set the `outline-width` to a sufficiently small number, such as 1 pixel.

4. **External Filtering Required:** N/A

5. **Review:** N/A

CSS 5.18: END

6. ACRONYMS

Table 6-1 Acronyms

Acronym	Denotation
CSS	Cascading Style Sheets
HTML	HyperText Markup Language
IE	Internet Explorer
PDF	Portable Document Format
URI	Uniform Resource Identifier
URL	Uniform Resource Locator
XHTML	eXtensible HyperText Markup Language
XML	eXtensible Markup Langague

7. TABLE OF DOCUMENT CONSTRUCTS

This section contains an index of all the constructs that appear in this document.

APPENDIX A: REFERENCED DOCUMENTS

Referenced Documents

The following publications were referenced in this document or used to prepare the document.

[1] Cascading Style Sheets Level 2 Revision 1 (CSS 2.1) Specification. Available at: http://www.w3.org/TR/CSS21/ W3C Recommendation 07 June 2011.

APPENDIX B: SUMMARY OF RISKS

ISG Section	Spec	Hiding	Attack	Disclosure
5.1 Parsing Errors	4.2	X		
5.2 Comments	4.1.9	X		X
5.3 Import Rule	6.3	X		X
5.4 Margin Property	8.3	X		
5.5 Display Property	9.2.4	X		
5.6 Position Property	9.3.1	X		
5.7 Overflow Property	11.1.1	X		
5.8 Clip Property	11.1.2	X		
5.9 Visibility Property	11.2 17.5.5	X		
5.10 Page Rule	13.1 13.2	X		
5.11 Color Property	14.1 14.2.1	X		
5.12 Background Image Property	14.2	X		
5.13 Background Position Property	14.2	X		
5.14 Background Attachment Property	14.2	X		
5.15 Font-size Property	15.7	X		
5.16 Text-indent Property	16.1	X		
5.17 Letter-spacing, Word-spacing, Line-height Properties	16.4 10.8	X		
5.18 Outline Property	18.4	X		