

# Ericom Shield 網頁瀏覽無害化

為您的企業員工提供零威脅無縫的安全網路瀏覽環境



Ericom Shield 為您的深度防禦策略加上了一層強大的防護罩，並阻絕了惡意程式、勒索軟體等威脅，減少檔案及文件下載的風險。不僅防止企業網路及用戶端受到攻擊與迫害，同時降低資訊人員的營運成本。

## 網頁瀏覽器：企業資安威脅的最大來源

### 來自瀏覽器的惡意程式造成莫大的損失

在日常的工作環境中，許多經常性的作業充滿資安風險，像是瀏覽網頁等行為，約莫 80% 的資安漏洞與威脅是來自網站或電子郵件的超連結。二十五年來，電腦犯罪與資訊安全產品之間的競爭從不停歇。電腦犯罪暫時贏了，在美國受到勒索軟體攻擊的公司，每年損失超過十億美元，而且攻擊越來越頻繁。

### 現有的解決辦法為何不足以應付這些威脅

被動的掃毒方式無法輕易辨識惡意程式與一般程式之間的差異。在網站分類不易的情況下，容易將帶有惡意程式碼的網站加入安全網站清單。這兩種都是被動的方式，只有在發現到惡意威脅時才會採取行動，而且必須不斷地更新程式與病毒碼。

### Remote Browser Isolation 預先隔離惡意威脅才是最佳解決方案

Remote Browser Isolation 能夠在隔離瀏覽器中執行每個網站，而這網站存在於暫存的 Linux 軟體容器中，讓資安風險遠離企業內部網路。接著再讓安全的資訊及網站內容傳送到內部使用者的瀏覽器，並提供原始、互動式及無縫的瀏覽方式。此方式不會讓原始網站可能的威脅進入到內部的用戶端，就算發生零時差攻擊也能夠提供絕佳的保護。

## 經過Ericom Shield 隔離後的比較

典型的 不安全的瀏覽內容

152 Java Scripts | 3 CSS Scripts | 22 External URLs

```
String.prototype.trimLeft = function() { return this.slice(1); };
String.prototype.trimRight = function() { return this.slice(0, -1); };
String.prototype.trim = function() { return this.trimLeft().trimRight(); };
function getElementsByClassName(className) {
    var elements = [];
    for (var i = 0; i < document.getElementsByTagName('*').length; i++) {
        var element = document.getElementsByTagName('*')[i];
        if (element.className === className) {
            elements.push(element);
        }
    }
    return elements;
}
function getElementsByTagName(tagName) {
    var elements = [];
    for (var i = 0; i < document.getElementsByTagName(tagName).length; i++) {
        var element = document.getElementsByTagName(tagName)[i];
        elements.push(element);
    }
    return elements;
}
function getElementsByTagNameNS(namespaceURI, localName) {
    var elements = [];
    for (var i = 0; i < document.getElementsByTagNameNS(namespaceURI, localName).length; i++) {
        var element = document.getElementsByTagNameNS(namespaceURI, localName)[i];
        elements.push(element);
    }
    return elements;
}
function getElementsByTagNameWithNamespace(namespaceURI, localName) {
    var elements = [];
    for (var i = 0; i < document.getElementsByTagNameNS(namespaceURI, localName).length; i++) {
        var element = document.getElementsByTagNameNS(namespaceURI, localName)[i];
        elements.push(element);
    }
    return elements;
}
function getElementsByTagNameWithNamespaceAndPrefix(namespaceURI, localName, prefix) {
    var elements = [];
    for (var i = 0; i < document.getElementsByTagNameNS(namespaceURI, localName).length; i++) {
        var element = document.getElementsByTagNameNS(namespaceURI, localName)[i];
        elements.push(element);
    }
    return elements;
}
function getElementsByTagNameWithNamespaceAndPrefixAndAttribute(namespaceURI, localName, prefix, attribute) {
    var elements = [];
    for (var i = 0; i < document.getElementsByTagNameNS(namespaceURI, localName).length; i++) {
        var element = document.getElementsByTagNameNS(namespaceURI, localName)[i];
        elements.push(element);
    }
    return elements;
}
function getElementsByTagNameWithNamespaceAndPrefixAndAttributeAndValue(namespaceURI, localName, prefix, attribute, value) {
    var elements = [];
    for (var i = 0; i < document.getElementsByTagNameNS(namespaceURI, localName).length; i++) {
        var element = document.getElementsByTagNameNS(namespaceURI, localName)[i];
        elements.push(element);
    }
    return elements;
}
function getElementsByTagNameWithNamespaceAndPrefixAndAttributeAndValueAndType(namespaceURI, localName, prefix, attribute, value, type) {
    var elements = [];
    for (var i = 0; i < document.getElementsByTagNameNS(namespaceURI, localName).length; i++) {
        var element = document.getElementsByTagNameNS(namespaceURI, localName)[i];
        elements.push(element);
    }
    return elements;
}
function getElementsByTagNameWithNamespaceAndPrefixAndAttributeAndValueAndTypeAndClass(namespaceURI, localName, prefix, attribute, value, type, class) {
    var elements = [];
    for (var i = 0; i < document.getElementsByTagNameNS(namespaceURI, localName).length; i++) {
        var element = document.getElementsByTagNameNS(namespaceURI, localName)[i];
        elements.push(element);
    }
    return elements;
}
function getElementsByTagNameWithNamespaceAndPrefixAndAttributeAndValueAndTypeAndClassAndId(namespaceURI, localName, prefix, attribute, value, type, class, id) {
    var elements = [];
    for (var i = 0; i < document.getElementsByTagNameNS(namespaceURI, localName).length; i++) {
        var element = document.getElementsByTagNameNS(namespaceURI, localName)[i];
        elements.push(element);
    }
    return elements;
}
function getElementsByTagNameWithNamespaceAndPrefixAndAttributeAndValueAndTypeAndClassAndIdAndStyle(namespaceURI, localName, prefix, attribute, value, type, class, id, style) {
    var elements = [];
    for (var i = 0; i < document.getElementsByTagNameNS(namespaceURI, localName).length; i++) {
        var element = document.getElementsByTagNameNS(namespaceURI, localName)[i];
        elements.push(element);
    }
    return elements;
}
function getElementsByTagNameWithNamespaceAndPrefixAndAttributeAndValueAndTypeAndClassAndIdAndStyleAndClassList(namespaceURI, localName, prefix, attribute, value, type, class, id, style, classList) {
    var elements = [];
    for (var i = 0; i < document.getElementsByTagNameNS(namespaceURI, localName).length; i++) {
        var element = document.getElementsByTagNameNS(namespaceURI, localName)[i];
        elements.push(element);
    }
    return elements;
}
function getElementsByTagNameWithNamespaceAndPrefixAndAttributeAndValueAndTypeAndClassAndIdAndStyleAndClassListAndAttributeNamespaces(namespaceURI, localName, prefix, attribute, value, type, class, id, style, classList, attributeNamespaces) {
    var elements = [];
    for (var i = 0; i < document.getElementsByTagNameNS(namespaceURI, localName).length; i++) {
        var element = document.getElementsByTagNameNS(namespaceURI, localName)[i];
        elements.push(element);
    }
    return elements;
}
function getElementsByTagNameWithNamespaceAndPrefixAndAttributeAndValueAndTypeAndClassAndIdAndStyleAndClassListAndAttributeNamespacesAndAttributeValues(namespaceURI, localName, prefix, attribute, value, type, class, id, style, classList, attributeNamespaces, attributeValues) {
    var elements = [];
    for (var i = 0; i < document.getElementsByTagNameNS(namespaceURI, localName).length; i++) {
        var element = document.getElementsByTagNameNS(namespaceURI, localName)[i];
        elements.push(element);
    }
    return elements;
}
function getElementsByTagNameWithNamespaceAndPrefixAndAttributeAndValueAndTypeAndClassAndIdAndStyleAndClassListAndAttributeNamespacesAndAttributeValuesAndAttributeTypes(namespaceURI, localName, prefix, attribute, value, type, class, id, style, classList, attributeNamespaces, attributeValues, attributeTypes) {
    var elements = [];
    for (var i = 0; i < document.getElementsByTagNameNS(namespaceURI, localName).length; i++) {
        var element = document.getElementsByTagNameNS(namespaceURI, localName)[i];
        elements.push(element);
    }
    return elements;
}
function getElementsByTagNameWithNamespaceAndPrefixAndAttributeAndValueAndTypeAndClassAndIdAndStyleAndClassListAndAttributeNamespacesAndAttributeValuesAndAttributeTypesAndAttributeValues(namespaceURI, localName, prefix, attribute, value, type, class, id, style, classList, attributeNamespaces, attributeValues, attributeTypes, attributeValues) {
    var elements = [];
    for (var i = 0; i < document.getElementsByTagNameNS(namespaceURI, localName).length; i++) {
        var element = document.getElementsByTagNameNS(namespaceURI, localName)[i];
        elements.push(element);
    }
    return elements;
}
function getElementsByTagNameWithNamespaceAndPrefixAndAttributeAndValueAndTypeAndClassAndIdAndStyleAndClassListAndAttributeNamespacesAndAttributeValuesAndAttributeTypesAndAttributeValuesAndAttributeTypes(namespaceURI, localName, prefix, attribute, value, type, class, id, style, classList, attributeNamespaces, attributeValues, attributeTypes, attributeValues) {
    var elements = [];
    for (var i = 0; i < document.getElementsByTagNameNS(namespaceURI, localName).length; i++) {
        var element = document.getElementsByTagNameNS(namespaceURI, localName)[i];
        elements.push(element);
    }
    return elements;
}
function getElementsByTagNameWithNamespaceAndPrefixAndAttributeAndValueAndTypeAndClassAndIdAndStyleAndClassListAndAttributeNamespacesAndAttributeValuesAndAttributeTypesAndAttributeValuesAndAttributeTypes(namespaceURI, localName, prefix, attribute, value, type, class, id, style, classList, attributeNamespaces, attributeValues, attributeTypes, attributeValues) {
    var elements = [];
    for (var i = 0; i < document.getElementsByTagNameNS(namespaceURI, localName).length; i++) {
        var element = document.getElementsByTagNameNS(namespaceURI, localName)[i];
        elements.push(element);
    }
    return elements;
}
```

## Ericom Shield 提供了...

- 一個關鍵的深度防禦層機制，讓網路上的惡意程式在觸及您的網路及用戶端前止步。
- 為您防護已知或未知的惡意程式、勒索軟體以及零時差攻擊的威脅。
- 針對網站上的惡意程式、電子郵件中的超連結或是下載的檔案提供完全的防護。

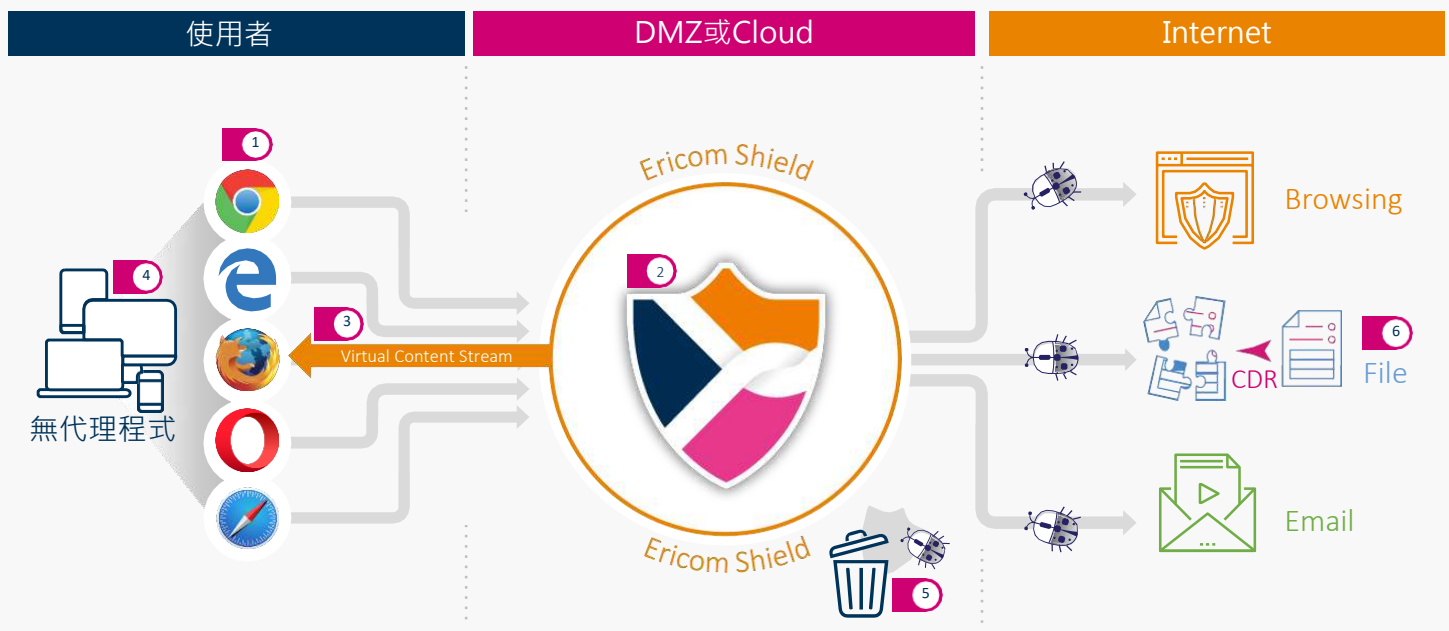
Ericom Shield 所提供的安全瀏覽內容

4 Java Scripts | 0 CSS Scripts | 0 External URLs

```
1 <html>
2 <head>
3 <meta name="viewport" content="width=device-width, user-scalable=no, initial-scale=1, maximum-scale=1">
4 <base href="/" target="">
5 <link rel="stylesheet" href="/css/acecss.min.402d4e02c77a5492e41eb52ab7466c.js">
6 <link rel="stylesheet" href="/icap-server-service-console/css/accessnow.min.65a713c7b536bfc6d645588a737abc3.css">
7 </head>
8 <body>
9 <script src="https://icap-server.service.console/client/id.js"></script>
10 <script src="/icap-server-service-console/console/id.js?474134594d9ef4807fe956e90.js"></script>
11 <script src="/icap-server-service-console/src/vendor.min.402d4e02c77a5492e41eb52ab7466c.js"></script>
12 <script src="/icap-server-service-console/src/accessnow.min.f0028e0c4d07b9385b48bc275750.js"></script>
13 </body>
14 </html>
```

# Ericom Shield 的解決方案

Ericom Shield 是如何不受惡意程式侵害?



- 1 當用戶透過他的瀏覽器開啟一個未經過分類，不確定是否為安全的網站時...
- 2 Ericom Shield 同時在遠端暫存的 Linux 軟體容器中開啟一個虛擬瀏覽器。
- 3 網站的原始碼會在虛擬的瀏覽器中執行，經過過濾，安全的虛擬網站內容會傳送至用戶端。
- 4 網站內容傳送到內部使用者的裝置，並提供原始、互動式及無縫的瀏覽方式
- 5 為了防止傳輸過程的機密資料被外流，例如跨網站指令碼 (簡稱 XSS)，會為每個瀏覽過程的每個連線提供獨立的隔離區，當瀏覽過程結束時，獨立的隔離區將會連同惡意程式一起被銷毀。
- 6 下載的檔案在傳送給用戶端之前，亦會透過內建檔案無害化與重組技術 (Content Disarm and Reconstruction) 程序掃描或清除。

## 為什麼需要 Ericom Shield

- 阻止最新流行的網路攻擊威脅到公司或組織。
- 讓用戶端可以安全瀏覽常用的網站以及下載檔案，以提高公司的生產力。
- 防止網路攻擊所導致代價極高的業務中斷、系統服務停止及商譽損失。
- 無代理程式技術節省部署的成本，同時讓管理變得更容易。
- 降低使用者叫修服務的維運成本，並提高IT部門的生產力。
- 用戶端能夠自由的選擇常用的作業系統、手持設備以及瀏覽器。
- 利用預先整合的檔案清理技術，以排除檔案或文件下載所造成的風險。
- 收集可匯出的事件後鑑識數據。

## 關於 Votiro

Votiro 是由一群資深的資安專家所組成，專家們的技能和企業系統安全的維護能力，無論在公司內外外部，都具有相當豐富的經驗。身為產業界中備受肯定的領導者，Votiro 對於企業內部機密資料和私有財產資料的安全性需求，擁有相當程度的了解，因此 Votiro 研發出能滿足客戶網路防護和基礎設備，不受外部網路威脅的解決方案。Votiro 防護機制也被運用在全球各地的政府、執法機構和不同產業領域的客戶，包含銀行業、國防單位、電信業、製藥業和其他產業等。

**VOTIRO**  
SECURED.

 Cyberview  
力悅資訊

力悅資訊股份有限公司  
台北市中山區松江路54號4F-4  
電話：02-25429758  
Email：Sales@cyberview.com.tw  
官方網站：<http://www.cyberview.com.tw/>