

**RFI Response**  
**National Institute of Standards and Technology**  
**Request for Information (RFI) Docket# 130208119-3119-01**  
**Developing a Framework to Improve Critical Infrastructure  
Cybersecurity**

**April 8, 2013**



**Proposal Submitted By:**

Daston Corporation  
Attn: Melanie Newton  
Melanie.newton@daston.com  
2010 Corporate Ridge, Suite 165  
McLean, VA 22102  
Phone: 256-777-0425  
Fax: 703-288-3315  
**Cage Code:** 03HK9  
**DUNS:** 931904486  
**Size Status:** Woman-Owned Small Business  
(WOSB)  
**TIN:** 541638058  
**GSA Schedule 70 #:** GS35F0555K

**Submitted To :**

*National Institute of Standards and Technology (NIST)*  
*100 Bureau Drive*  
*Gaithersburg, Maryland 20899*  
Attn: [cyberframework@nist.gov](mailto:cyberframework@nist.gov)

---

## Table of Contents

RFI Response Section One: Introduction .....**Error! Bookmark not defined.**  
RFI Response Section Two: Team Daston Response Summary ..... 2

---

## Section One: Introduction

Daston Corporation, and our browser security partner Authentic8 (Team Daston) are pleased to submit our response to the National Institute of Science and Technology (NIST) Request for Information (RFI) Docket # 130208119-3119-01 for comments on Developing a Framework To Improve Critical Infrastructure Cybersecurity. Team Daston's comments on how to improve critical infrastructure are centered on the importance of improving browser security.



is an *ISO 9001:2008 registered (Certificate # FS676004)* privately held *Woman-Owned Small Business (WOSB)* that maintains a Top Secret facility clearance at its headquarters in McLean, VA. Daston provides a wide range of IT and Cloud Based Services to a number of leading Federal Government Agencies, including the Peace Corps, Defense Information Systems Agency (DISA), US Army, Forest Service, Treasury, and Department of State as well as many State & Local Agencies such as City of Pittsburgh, City of St. Louis, and South Carolina Department of Health and Environmental Controls (SCDHEC). Daston has a 20 year history of successfully providing Information Technology services such as software application development, certification & accreditation (C&A), identity & access management, enterprise infrastructure engineering, service oriented architecture (SOA), database administration, business intelligence, to many leading federal, commercial and education institutions.

Daston is a fully authorized reseller of Authentic8's browser security solutions and have teamed together to bring Authentic8's innovative browser security suite to the Federal Government. Daston is an experienced prime contractor and has an array of contract vehicles available to its clients such as General Services Administration (GSA) IT70, GSA Financial & Business Solutions (FABS), and Mission Oriented Business Integrated Services (MOBIS). In addition, Daston Corporation has an SBA certified 8a Joint Venture (JV) called EnterpriseTech that holds a GSA 8a STARS II Contract Vehicle.



Authentic8, Inc. is privately held and headquartered in Mountain View, California since 2009. Authentic8 puts you back in control by delivering the browser as a service where it is secured and managed on the Authentic8 servers. Users get a secure web experience from any computer, and your policies remain intact regardless of how and from where they're connecting. Authentic8 insulates users from malware, encrypts all your data, and gives your business a central point of command and control for all your web applications.

Founded by members of the original Postini team, the technical resources now include former engineers from Apple, Google, Juniper, Microsoft, Nokia and VMware.

Technical Point of Contact at Authentic8 [questions@authentic8.com](mailto:questions@authentic8.com)

---

## **Section Two: Team Daston Response Summary**

Web browsers have come to represent the client interface to Internet applications. As critical infrastructure components become increasingly managed by web-based administration consoles, the web browser becomes a point of vulnerability where several attack surfaces converge. But web browsers are inherently insecure and unmanageable, exacerbating the dependence on proper user behavior to maintain proper internet hygiene. At the same time, users are finding themselves in the position of integrating disparate services with no common administration framework so they have to self-enforce policy guidelines. We propose a series of modifications to existing practices which can improve security without the need for rebuilding infrastructure or increasing the dependence on proper user behavior.

### **2.1 Users connect from a variety of work and personal machines**

Managing secure access to internal services (i.e. “behind the firewall”) is achievable, but 3rd party web apps open up new vulnerabilities. Users download data, browsers store cookies, and increasingly accessed from machines outside of IT’s control. Maintaining consistency across each end point is impossible. The absence of a ubiquitous policy enforcement tool means that admins cannot adjust browser behavior based on changing circumstances. The result is a wide range of discrepancies in browser configurations, thus amplifying risk.

### **2.2 Users come in from networks that are insecure**

Open wifi networks are abundant. Users are connecting from networks that may be snooped or compromised in another fashion. Even if a user is not explicitly connecting, client apps can connect in the background and exchange sensitive account data.

### **2.3 Web code is suspect**

- Browsers are designed to connect to web services over industry-standard protocols and to execute any code that the service delivers (HTML, Javascript, Flash, HTML5, etc). The ungoverned nature of web code results in the delivery of malicious code, resulting in theft or misrepresentation of data, and more.

### **2.4 User-controlled passwords can be weak**

While users understand the risks, we are all still creatures of habit in creating simple passwords or re-using passwords across multiple sites. As long as the password remains information stored in a user’s head, there will be the tendency to reduce entropy in effort to make passwords easy to remember. At the same time, the proliferation of password databases containing similar records increases the likelihood that a compromise of one data base will yield information useful for compromising other databases.

The culmination of these diverse risks put IT on its heels, and makes protecting the browser an exercise in reacting to threats and attacks and an increasing dependence on proper user behavior. We propose modifications to the existing practices such that the security framework is **not** dependent on user behavior.

---

The solution must address **three** distinct, inter-dependent areas:

### **Sandboxing**

1. move the browser to a constrained environment where other resources are not accessible.
2. The display and user interface events should be delivered to the user with no machine executable content.
3. The browser events (downloading, login, link following, etc) can then be intercepted and subject to policy.

### **Identity and Access Management**

1. Multi-factor authentication to include factors which are not managed by the user.
2. Device identification, so that connections from unknown machines can be processed differently.
3. Obfuscation of credentials. Managing credentials for the users without sharing details, so that users need not duplicate passwords, nor enter them using devices which might be compromised.

### **Policy**

1. Regulate how the browser is used. Users may require specific approval for activities like download, print, access to non-essential sites, etc.
2. Adapt to environmental circumstances such as location, group membership, device trust.
3. Provide administrative oversight both proactively (eg. determining which sites may be accessed) and reactively (eg. triggering additional verification if a machine is changed).

### **Web browsers have these systemic vulnerabilities:**

1. Add-ons which allow communication between the browser and the device (Java Plug-in, ActiveX controls) introduce the possibility of day-zero exploits for which a defense may not be in place.
2. The client device may execute code capable of capturing data from the browser (eg. key loggers and root kits)
3. Browsers transfer data from a remote server and evaluate or execute that data after it has already been received on the device.
4. Some of the core tools, such as JavaScript, DNS and HTTP, have limited tools to ensure authenticity and secrecy.

### **Therefore, attacks tend to result in:**

1. Unauthorized **installation** of malicious code on the user's machine.
2. Unauthorized **access** of data located elsewhere on the user's machine.
3. Unauthorized **collection** of data from the user's browser sessions (eg. using downloaded files or logging keyboard activity)
4. Unauthorized **activity** on the client machine (eg. as a spam bot or malware server).
5. **Mis-direction** of the user to a fraudulent site, where they might be tricked into entering sensitive information.
6. The **replay** of stolen credentials for unauthorized access to internet applications from another location.

**We propose that a cyber-security framework include:**

- 1) Dislocation of the browser from the user's client device. Execute a controlled web client in a sandbox on an alternate machine (a secure container)
- 2) Controlling the flow of data between the user's device and the secure container by establishing a secure connection and using a non-HTTP protocol
- 3) Insulation of the end device by keeping all web code off of it.
- 4) Insulation of the web app from any malicious code that may exist on the untrusted device
- 5) Build a fresh, profile-based web client at session request, and execute all web code in the container. Destroy the container when the session ends
- 6) Pre-provision the container with the specific web apps that the user is authorized to access.
- 7) Execute network-based and content-based malware detection and resource validation routines within the container
- 8) Strong, multi-factor authentication to verify the user's identity as they access the container. Vary authentication requirements depending on machine, time, location, etc changes.
- 9) Authentication methods that mitigate end-device key logger or other password-snooping exploits.

---

10) Identity management methods enabling strong, secure access to downstream web services without exposing credential data to users or snooping exploits.

11) Simple, one-click access to users' apps after they've been authenticated to the container. Unnecessary complexity increases the password fatigue and poor credential management.

12) Establish and enforce a set of policy guidelines to control use of the web client and access to data within web apps, including access controls and data use permissions.

13) Adjust policy guidelines based on changing environmental variables, such as machine, location, time of day, etc.

---

## References

HIPAA Breach report:

<http://www.hhs.gov/ocr/privacy/hipaa/administrative/breachnotificationrule/breachtool.html>

OWASP Top Ten Vulnerabilities: [https://www.owasp.org/index.php/Top\\_10\\_2013](https://www.owasp.org/index.php/Top_10_2013)

Transport secrecy and authentication using TLS/SSL ([pdf](#)):

<http://csrc.nist.gov/publications/nistpubs/800-52/SP800-52.pdf>