

*The following security alert was issued by the Information Security Division of the Mississippi Department of ITS and is intended for State government entities. The information may or may not be applicable to the general public and accordingly, the State does not warrant its use for any specific purposes.*

**DATE(S) ISSUED:**

12/29/2011

**SUBJECT:**

Multiple Vulnerabilities Reported in the .NET Framework (MS11-100)

**OVERVIEW:**

Multiple vulnerabilities have been reported in the Microsoft .NET Framework, specifically in ASP.NET, that could allow remote code execution. ASP.NET allows developers to build dynamic Web applications and Web services. Successful exploitation of some of the vulnerabilities could result in an attacker gaining the same privileges as the logged on user. Depending on the privileges associated with the user, an attacker could then install programs; view, change, or delete data; or create new accounts with full user rights. It should be noted that one vulnerability will cause a Denial of Service condition.

**SYSTEMS AFFECTED:**

- Microsoft .NET Framework 1.1
- Microsoft .NET Framework 2.0
- Microsoft .NET Framework 3.5
- Microsoft .NET Framework 4

**RISK:**

**Government:**

- Large and medium government entities: **High**
- Small government entities: **High**

**Businesses:**

- Large and medium business entities: **High**
- Small business entities: **High**

**Home users: High**

**DESCRIPTION:**

Multiple vulnerabilities have been discovered in the Microsoft ASP.net that could allow for remote code execution. These vulnerabilities only impact systems that have IIS installed. By default, IIS is not installed on any Windows operating system.

**Collisions in HashTable May Cause DoS Vulnerability**

The vulnerability exists because of the way that ASP.NET hashes specially crafted requests and inserts that data into a hash table causing a hash collision. When many of these collisions are chained together, the performance of the hash table is greatly degraded leading to the denial of service condition. It is important to note that the hash collision attacks used to exploit this vulnerability does not only impact ASP.NET. This is an industry-wide issue affecting other Web Platforms, such as PHP and Ruby.

**Insecure Redirect in .NET Form Authentication Vulnerability**

A spoofing vulnerability exists in the way that .NET Framework verifies return URLs during the forms authentication process. An attacker could use this vulnerability to redirect users to a malicious website. In a web-based attack, a user visiting the malicious site can be exploited by this vulnerability. The attacker could also use this vulnerability in the form of a phishing attack by providing a link in an email to the malicious website.

### **ASP.Net Forms Authentication Bypass Vulnerability**

An elevation of privilege vulnerability exists in the way that .NET Framework authenticates users. In order to exploit this vulnerability, an unauthenticated attacker would need to be able to register an account on the ASP.NET application, and must know an existing account name for a targeted user. The attacker could then craft a special web request using a previously registered account name to gain access to that account. The attacker could then take any action in the context of the targeted user, including executing arbitrary commands on the site.

### **ASP.NET Forms Authentication Ticket Caching Vulnerability**

An elevation of privilege vulnerability exists in the way that ASP.NET Framework handles cached content when Forms Authentication is used with sliding expiry. Forms Authentication is a method of tracking users behavior on a site. An attacker would need a victim to click on a link in an email or visit a malicious site for successfully exploit this vulnerability. Successful exploitation of this vulnerability may allow the attacker to run commands that the signed on user currently had.

### **RECOMMENDATIONS:**

The following actions should be taken:

- Apply the appropriate patches provided by Microsoft to vulnerable systems immediately after appropriate testing.
- Run all software as a non-privileged user (one without administrative privileges) to diminish the effects of a successful attack.
- Remind users not to visit un-trusted websites or follow links provided by unknown or un-trusted sources.
- Remind users not to open e-mail attachments from unknown users or suspicious e-mails from trusted sources.

### **References:**

#### **Microsoft**

<http://technet.microsoft.com/en-us/security/bulletin/MS11-100>

<http://technet.microsoft.com/en-us/security/advisory/2659883>

#### **CVE**

<http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2011-3414>

<http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2011-3415>

<http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2011-3416>

<http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2011-3417>

#### **Security Focus**

<http://www.securityfocus.com/bid/51186>