A NGSSoftware Insight Security Research Publication

E-mail Spoofing and CDONTS.NEWMAIL

(Protecting Microsoft Active Server Pages Applications)

David Litchfield
(david@ngssoftware.com)
9th January 2002
www.ngssoftware.com

[Abstract]

Many IIS web servers running ASP applications will use the CDONTS.NEWMAIL object to provide the functionality for feedback or contact forms. This paper will examine how the CDONTS.NEWMAIL object can be used by attackers to send arbitrary e-mails via the vulnerable web server and what must be done to prevent an online ASP application being abused in this way. This paper is written to show ASP developers the importance of client input validation and that without it even the most seemingly innocuous code can become dangerous.

[Details]

To add e-mailing functionality for Microsoft Active Server Pages applications running on Internet Information Server a COM object is provided: CDONTS.NEWMAIL. Often this object is used for feedback forms, newsletters and alerts by the application. A typical ASP page that uses the CDONTS.NEWMAIL object may look similar to the following

```
<%
        set objNewMail = CreateObject("CDONTS.Newmail")
        objNewMail.From = "newsletter@company.com"
        objNewMail.To = Request.QueryString("email")
        objNewMail.Subject = "NEWSLETTER"
        objNewMail.Body = "Please find attached the newsletter."
        objNewMail.AttachFile "c:\newsletter.txt", "mailatt.txt"
        objNewMail.Send
%>
```

The first line of this ASP code creates the CDONTS.NEWMAIL object and the remainder of the code sets some of its properties with the last line calling the .send method which causes the ASP page to dispatch the e-mail. The CDONTS.NEWMAIL object acts as a mail client and connects to the IIS SMTP service running on the same machine and sends the mail. To receive a copy of the news letter a user would go to a URL on the web server similar to

http://www.company.com/newsletter.asp?email=david@ngssoftware.com

As can be seen from the code the email parameter in the query string is passed straight to the CDONTS.NEWMAIL object's .To property. When the objects sends the mail an SMTP conversation takes place :

```
..
..
mail from: newsletter@company.com
rcpt to: david@ngssoftware.com
data
Subject: NEWSLETTER
..
..
```

and the e-mail is sent. However, had the email address been entered with the relevant SMTP commands and newline characters such as

http://www.company.com/newsletter.asp?email=victim@spoofed.com%0D%0Adata%0D%0ASubject:%20Spoofed!%0D%0A%0D%0AHi,%0D%0AThis%20is%20a%20spoofed%20email%0D%0A.%0D%0Aquit%0D%0A

then the SMTP conversation would go similar to

```
..
..
mail from: newsletter@company.com
rcpt to: victim@spoofed.com
data
Subject: Spoofed!

Hi,
This is a spoofed e-mail
.
quit
```

In this way a spoofed e-mail has been sent using the NEWMAIL object. Rather than quitting the SMTP conversation, however, an attacker could send an entirely new mail and modify who the mail is from too.

[Impact]

As can be seen it is a trivial task for an attacker to send an arbitrary e-mail from the web server. This could be used by the attacker in any number of nefarious ways limited only by their imagination. For example, they could spoof a press release (seemingly) from company.com. By looking at the e-mail's properties the source would indeed be from company.com. This kind of attack can have the most damaging effect on businesses. In 2000, Emulex lost $2.2 billion of its total market capitalization due to a spoofed press release and in March 2001 a Hong Kong law firm was the victim of a spoofed e-mail that stated one of their cleaners had been murdered. On the less damaging side, this could be used by spammers to fill up mail boxes with even more unwanted mail.

[Resolution]

With all aspects of an online web application it is imperative to ensure that all client side input is validated. Validated means cleaned up and checked for anything that may subvert the application’s security. To make safe client input for CDONTS.MAIL all new line type characters such as 0x0A and 0x0D and should be stripped from the input. Whilst these characters have no effect on the safety of the CDONTS.NEWMAIL object's properties itself when it comes to sending the mail they do have a (dangerous) effect. To replace a character in an ASP application the Replace( ) function can be used.

The sample application given here is vulnerable to poisoning of the .To property. Other applications may be vulnerable to the poisoning of other properties such as .From or .Subject. It is important to ensure that before client side input is embedded in these properties that it is made safe.