Cross Site Printing

Printer Spamming

By using only JavaScript, an Internet web site can remotely print to an internal network based printer by doing an HTTP Post. The web site initiating the print request can print full text, enter PostScript commands allowing the page to be formatted, and in some cases send faxes. For the attack to succeed the user needs to visit a web site that contains this JavaScript.

Aaron Weaver – aaron.weaver2 [at] gmail [dot] com

11/23/2007

Cross Site Printing

Printer Spamming

Introduction and Background

Many network printers listen on port 9100 for a print job (RAW Printing or Direct IP printing). You can telnet directly to the printer port and enter text. Once you disconnect from the printer it will print out the text that you send it. Network printers also accept PostScript, and Printer Control language. The security around this is usually minimal – connect to the port, send the print job, disconnect and the printer prints the page.
Within the last year there have been new discoveries on attacking the Intranet from the Internet[1]. This involves setting an image tag or script tag to an internally addressable IP address and then the browser will request the "image" resource. Several attacks can be accomplished; port scanning, fingerprinting devices, and changing internal router settings.

Attack

A simple proof of concept is creating an image and setting the source to the printer. Since the printer is waiting for the connection to close the browser will still try to download the "image" in the browser until it eventually times out.[2]

<img src="myprinter:9100/Printed_from_the_web">

Results in the printer outputting:

[1] Hacking the Internet from the Intranet – Jeremiah Grossman
www.whitehatsec.com/home/assets/presentations/blackhatusa07/0807blackhat_**hacking**.pdf

[2] Hacking Network Printers – Adrian "Irongeek" Crenshaw
http://www.irongeek.com/i.php?page=security/networkprinterhacking

```
GET /Printed_from_the_web HTTP/1.1
Accept: */*
Accept-Language: en-us
UA-CPU: x86
Accept-Encoding: Mozilla/4.0
Host: myprinter:9100
Connection: Keep-Alive
```

While this is interesting if the end goal is for someone to pick up the paper and read it then most likely it will be ignored. Since data on a GET request is URL encoded the first line will run off the page because there isn't a line break.

However by using a POST and setting the form type to *multipart/form-data* (so that the data is not encoded) we will be able to send a properly formatted request to the printer. The simple HTML form below submits text to a printer.

```
<FORM ACTION='HTTP://YOURPRINTER:9100' ID='MSGFORM' ENCTYPE='MULTIPART/FORM-DATA' METHOD='POST'>

<TEXTAREA NAME='MSG' ID='MSG' WRAP='NONE' ROWS='50' COLS='100'>

    TESTING THIS PRINTER OUT.

</TEXTAREA><INPUT TYPE=SUBMIT VALUE=SUBMIT></FORM>
```

Some of the other things we tried were sending ASCII art. So how about an advertisement for frogs?

Sending plain text works very well and it would be compatible with most printers, but if we could send PostScript commands then we can format the page and make it look however we would like. The Javascript below sets the printer language to PostScript using PCL (Printer Command Language). PCL has to be used to set PostScript since PostScript requires that the first command be set as %!PS. (The request from the browser starts out with POST.) The JavaScript below could be used to send a PostScript job to a printer.

```
var msg=String.fromCharCode(27) + "%-12345X@PJL ENTER LANGUAGE
= POSTSCRIPT\r\n”
+ "%!PS\r\n"
+ "/Courier findfont\r\n"
+ "20 scalefont\r\n"
+ "setfont\r\n"
+ "72 500 moveto\r\n"
+ "(Your printer is mine!) show\r\n"
+ "showpage\r\n"
+ String.fromCharCode(27) + "%-12345X
```

Which prints the following in Courier 20.

Your printer is mine!

Additionally there are PostScript files on the internet that convert HTML directly to PostScript. So you could easily create a JavaScript function that first sends the HTML to PostScript command followed by the web page you want to print.

Attack Outline

The attack could be initiated by creating a hidden iframe, and then creating a form and submitting the contents to the printer. Since the connection will not close, a setTimeout could be used to cancel the request so that the printer would print the request. A for loop could be setup to iterate through the

192.X.X.X or 10.X.X.X and send multiple requests. Smarter attacks could use an applet to determine the internal IP address of the user and then start with that subnet – since most network printers are on the same subnet as the user.

Faxing

PCL can be also used to send out faxes. Fax PCL tends to be proprietary so it will vary from printer to printer. Though not tested it looks possible to send faxes from a Xerox machine.

```
<ESC>%-12345X
@PJL SET RESOLUTION=400
@PJL COMMENT XRXbegin
@PJL COMMENT OID_ATT_FAX_CONFIRMATION TRUE;
@PJL COMMENT OID_ATT_JOB_TYPE OID_VAL_JOB_TYPE_FAX_SEND;
@PJL COMMENT OID_ATT_FAX_TYPE OID_VAL_FAX_TYPE_G3_AUTO;
@PJL COMMENT OID_ATT_FAX_DESTINATION_PHONE "0123456789";
@PJL COMMENT XRXend
```

Persistent Printer Spam

Use PCL to create a banner page. Then when any print job is sent out it will have the banner page attached. This can be a good way to get your message across.

Remediation

There are several possible ways to protect from this type of attack. First always have an administrator password set on your printer. Secondly look at restricting access to the printer so that it only accepts print jobs from a centralized print server.

Summary

The end result is that by visiting a web site on the Internet you could end up sending printer spam to your printer without even knowing that anything happened. Since most printers don't have any security set it is possible to print anything, control the printer, change the print settings and even send faxes.

Special thanks to Dan Fields, Darian Patrick, and Jeremiah Grossman for adding input and reviewing.