



SkyDog 2016 Catch Me If You Can VM — Vulnhub.com



Andrew Hilton · [Follow](#)

16 min read · Jan 17, 2017



105



So this is the first one of these that I've done in quiet a few months. Please bare with me as this might be a slow process as I get back into it haha.

You can download the VM from [here](#), I had a great time working through it.

Lets get into it.

So after a default **Netdiscover** scan I get the IP as **192.168.1.155**

Lets see what we can find from an nmap scan

```
[root@parrot]—[/home/andrew]  
└─── #nmap -p- -A 192.168.1.155
```

So we can see from the above scan command that ports

22/tcp closed ssh

80/tcp open http

443/tcp open https

22222/tcp open easyengine

are on the system. A standard **80** is open so lets fire up firefox and see what the website has on it.

On first viewing the website there is nothing that jumps out straight away.
I'll come back to the website in a bit.

Let's check out the **ssh port 22222**

```
[root@parrot]—[/home/andrew]
```

```
└── #ssh -p 22222 192.168.1.155
```

The authenticity of host '[192.168.1.155]:22222 ([192.168.1.155]:22222)'
can't be established.

ECDSA key fingerprint is

SHA256:DeCMZ74o5wesBHFLyaVY7UTCA7mW+bx6WroHm6AgMqU.

Are you sure you want to continue connecting (yes/no)? yes

Warning: Permanently added '[192.168.1.155]:22222' (ECDSA) to the list of known hosts.

```
# WARNING #####  
# FBI — Authorized access only! #  
# Disconnect IMMEDIATELY if you are not an authorized user!!! #  
# All actions Will be monitored and recorded #  
# Flag{53c82eba31f6d416f331de9162ebe997} #  
#####
```

Cool, there's the first flag

Flag{53c82eba31f6d416f331de9162ebe997}

Lets run it through crackstation and see what that hash decrypts as.

53c82eba31f6d416f331de9162ebe997 md5 encrypt

So the flag is “**encrypt**” I have no idea what to do with that at the moment. I'll make a note of it in case I need to come back to it later.

Lets go back to the website and do a deeper dive on it. Looking at the source code there doesn't seem to be much going on, one comment does seem out of place though

```
<!-- [If IE4]><script src="/oldIE/html5.js"></script><!--[Make sure to  
remove this before going to PROD] →
```

Viewing the directory I can see it's a huge dump of what look likes obfuscated javascript. The first line though seems like a hash of some kind.

666c61677b3763303133323037306130656637316435343236363365396463
3166356465657d

Running it through crackstation though doesnt return anything useful, at this point I was stuck for while not really knowing what to do next. It was only after rereading the flag clues that I noticed the first one was

Don't go Home Frank! There's a Hex on Your House.

So maybe that string isn't a hash but a hexadecimal string, heading over to asciitohex.com i can try to convert it.

Bingo!! it comes back as

flag{7c0132070a0ef71d542663e9dc1f5dee} cool, so now heading back over to crackstaion and decrypting the hash value gives me

7c0132070a0ef71d542663e9dc1f5dee md5 nmap

Cool, so I've found two flags but I think I've found them in the wrong order as the one I just found (2nd) decrypts as "nmap" which I think is a clue to head back to nmap and do another scan to find the non standard ssh service running on port 22222. Which would lead me to the ssh flag and get the word "encrypt"

Ok so the clue for flag 3 is **"Be Careful Agent, Frank Has Been Known to**

Intercept Traffic Our Traffic” after a little googling I came across a website that said about traffic intercepting via invalid SSL certificates. So I am going to start my search with the SSL cert.

Loading the webpage in firefox and then adding an exception for the invalid cert allows me to view the contents and there is it the third flag.

flag3{f82366a9ddc064585d54e3f78bde3221} taking it over to crack station and it decrypts as

f82366a9ddc064585d54e3f78bde3221 md5 personnel

Ok so the next clue is “**personnel**” maybe it’s another directory on the website. Back in Firefox and I can see

192.168.1.155/personnel is a valid directory but I’m not authorised to view it as I’m not coming from an FBI Workstation.

Ok so how can I make it think I’m coming from an FBI workstation, viewing the source doesn’t give me anything useful at all. After lots of Googling and trying to get a foot hold on anything useful and failing. I decide to go back through my notes and look for anything I can use. The only other real piece of useful info I have so far is that huge Javascript dump from the first flag.

After trawling through the dump there were two lines that stuck out.

```
/* maindev — 6/7/02 Adding temporary support for IE4 FBI Workstations
*/
/* newmaindev — 5/22/16 Last maindev was and idoit and IE4 is still
Gold image -@Support doug.perterson@fbi.gov */
```

This looks like a goldmine. First we have the user agent profile needed to access the site from an FBI workstation but also the second line gives us an email address for support. We can use this address possibly as a username elsewhere in the VM.

Doug Perterson

So now I need to change my UA string to allow us to appear to be connecting through an IE4 browser. So after a quick Google search to find out what the string is meant to contain (as the default downloadable list only goes back to IE5.5) I came across this website

<http://www.useragentstring.com/pages/useragentstring.php?name=Internet+Explorer> which allowed me to add a new string

Now I can test the UA string once it has been changed by selecting it and reloading the page.

BINGO!!! It worked and now the page give us this:

And as another bonus of gaining access to the page the 4th flag is there too.

flag{14e10d570047667f904261e6d08f520f}

So lets see how that decrypts on crackstation, it decrypts as

14e10d570047667f904261e6d08f520f md5 evidence

Just below the flag on the FBI portal is another clue

So reading it literally, it gives me **Clue=newevidence**. I dont really understand what that wants me to do at the moment so lets look at the clue from the fifth flag which is

The Devil is in the Details — Or is it Dialogue? Either Way, if it's Simple, Guessable, or Personal it Goes Against Best Practices

There are a few more interesting parts to the FBI Portal the first is

An Unordered List

Chase Manhattan Bank

Dr. Frank Connors

France

July 16, 2009.

Great American Masterpiece.

Heidelberg or Man

and then the second is

An Ordered List

France

Dr. Frank Connors

Chase Bank

Heidelberg

Miami

Sixty-one on 7/4/6008

I'm sure this is really important but I'm not sure how right now. I have entered all of into my working notes and I'll come back to it when I get stuck again.

So on the off chance that the clue was another directory to be searched I checked out **192.168.1.155/newevidence** and got to a pop up window that needs authentication. I'm guessing this is the next step

I can see that the Personnel page has us logged in as an **Agent Hanratty**, a quick Google search of the name gives me his first name as **Carl**.

Ok so how are the usernames formatted, I spent way too long thinking about this until I remembered the support email from the JS dump
@Support doug.perterson@fbi.gov

It looks like the naming convention is **firstname.lastname** so using that with the info we have so far I get

carl.hanratty

cool so what can the password be, the clue hints at it being **Simple, Guessable, or Personal**.

It also mentions **Dialogue** in the clue. So I'm thinking it's something to do with the characters dialogue in the film. Lets head over to IMDB and see

if we can find out any simple/personal info.

<http://www.imdb.com/character/ch0004040/bio>

After trying a lot of wrong passwords using details from IMDB and dialogue from the film I tried the name **Grace** (Carls daughters name) he says it in the line **“She was four when I left. Now she’s 15. My wife’s been remarried for 11 years. I see Grace every now and again.”**

Bingo, I’m in.

So lets take a look around the site and see if we can find the flag, I’m going to start looking in the **possible locations** link. Which contains this picture

Let's download it and run some **exiftools** on it to see if there's anything in the metadata

Running it through **exiftool**, **identify** and **strings** didn't return anything useful at all.

Ok so back to the drawing board, I could probably use a reverse Google image lookup to find the actual location in the image but I won't get the flag from that, so what am I missing?

Let's go back and check out the rest of the links on the webpage. Clicking the **Evidence Summary File link** and there it is another flag string haha, I should have just clicked on that before falling down the exif rabbit hole. Ok so like the others let's see what it decrypts to.

flag{117c240d49f54096413dd64280399ea9} becomes
117c240d49f54096413dd64280399ea9 md5 panam

Cool so the next clue has something to do with **Panam** (the airline company featured in the film).

Let's move onto the next flag no.6 the clue is "**Where in the world is Frank**" I'm thinking its got something to do with the picture I found a minute ago, so I'm going to have to go back and hit it again.

Whilst I'm on the site let's see what the last link has to offer. It contains 1 invoice from a company called **Hetzl and Associates** They are invoicing an **Agent Earl Amdursky** for an encryption consultation project.

Cool I'll save this to my working notes file and also make a note of the agents name, as this gives me another user should I need it.

So still no real leads on flag 6 so it's back to the image file, running a **ls -la** command on the dir I can see that the image file is 4.2mb, thats not huge but it does seem weird, I'm nearly 100% sure this has something else going on in it, I just dont know how to find it. Lets go back to google and see if I can find another Stego app that I can use to scan the image with.

Googling the term **steg app kali** and clicking on the first link brings me to an app called **steghide**

<http://steghide.sourceforge.net/>

What's really interesting here is at the bottom of the page the authors

name is **Stefan Hetzl**, that second name sounds familiar right?!

It was the name of the person sending the encryption invoice from earlier, so now I know I'm on the right path. Steghide is built into kali linux so a quick command of

```
└─[root@parrot]─[/home/andrew/Desktop]
```

```
└─ #steghide — extract -sf image.jpg
```

Enter passphrase:

and I hit another wall, haha ok.

It only took a few seconds of going back through my notes to remember flag 5 decrypted as panam so I tried that as the password and it worked :-)

```
─[root@parrot]─[/home/andrew/Desktop]
```

```
└─ #steghide — extract -sf image.jpg
```

Enter passphrase:

wrote extracted data to “flag.txt”.

It dumped a **flag.txt** file into the directory and opening the flag file gave me the next flag and a clue

flag{d1e5146b171928731385eb7ea38c37b8}

=ILoveFrance

clue=iheartbrenda

The flag decrypts as

d1e5146b171928731385eb7ea38c37b8 Unknown Not found. (That's really

strange that it isnt a real hash) Hmmm ok I'll make a note of it and then come back it if I need to.

We also have the clues

=ILoveFrance and clue=iheartbrenda

A Google search confirms Brenda is another character from the film. Brenda Strong is Franks girlfriend. Ok noted and lets move on.

Trying **/iheartbrenda** as a dir doesnt return anything neither does the same but with **/ILoveFrance** so what is the clue telling me to do??

The actual flag clue for the 7th flag is **“Frank Was Caught on Camera Cashing Checks and Yelling — I’m The Fastest Man Alive!”** with nothing else to try at this stage I Googled the clue string and after removing parts of it I was left with **“The fastest man alive”** which came back as a possible hint, it refers to the 2nd episode of the The Flash.

This is a hint I think because in the film Franks character also takes the name **Barry Alan** who is the alter ego of The Flash. It all seems to fit with the clue but still doesn't lead me to the flag.

So lets use this new info and try to get somewhere. I still haven't gotten into the SSH login yet so maybe barry is a user for that lets see.

```
└─[X]─[root@parrot]─[/home/andrew/Desktop]  
└─── #ssh 192.168.1.155 -p 22222 -l barry.alan
```

```
# WARNING #####  
# FBI — Authorized access only! #
```



```
# Disconnect IMMEDIATELY if you are not an authorized user!!! #
# All actions Will be monitored and recorded #
# Flag{53c82eba31f6d416f331de9162ebe997} #
#####
barry.alan@192.168.1.155's password:
Permission denied, please try again.
```

using the passwords from the clues gives me nothing.
It's then that I realise my mistake. I have been using the wrong spelling.
In the film it's spelt Barry Allen not Alan. So let's pretend that didn't
happen and try to log into the SSH again this time with the correct
spelling haha

DAMN IT!! Still no luck. Ok after what seemed like every combination of
Barry and Allen I could possibly think of I hit gold with **barryallen** and
the password **iheartbrenda**.

```
└─[X]─[root@parrot]─[/home/andrew/Desktop]
└─ #ssh 192.168.1.155 -p 22222 -l barryallen
#####
# WARNING #
# FBI — Authorized access only! #
# Disconnect IMMEDIATELY if you are not an authorized user!!! #
# All actions Will be monitored and recorded #
# Flag{53c82eba31f6d416f331de9162ebe997} #
#####
barryallen@192.168.1.155's password:
Welcome to Ubuntu 16.04.1 LTS (GNU/Linux 4.4.0-38-generic x86_64)
```

- * Documentation: <https://help.ubuntu.com>
- * Management: <https://landscape.canonical.com>
- * Support: <https://ubuntu.com/advantage>

14 packages can be updated.

7 updates are security updates.

```
barryallen@skydogconctf2016:~$
```

I'm in, lets see whats on the system

```
barryallen@skydogconctf2016:~$ ls
flag.txt security-system.data
```

Theres the flag and a simple CAT command gives me

flag{bd2f6a1d5242c962a05619c56fa47ba6} which decrypts to

bd2f6a1d5242c962a05619c56fa47ba6 md5 theflash (haha I knew it was going to be that) cool, so just one flag left to get and the clue is

Flag #8 Franks Lost His Mind or Maybe it's His Memory. He's Locked Himself Inside the Building. Find the Code to Unlock the Door Before He Gets Himself Killed!

The other file on the system was a **security-system.data** file. Lets see what we can do with that.

```
barryallen@skydogconctf2016:~$ file security-system.data
```

security-system.data: Zip archive data, at least v2.0 to extract

So lets unzip it and see what it contains, I couldnt unzip it all on the system so decided to copy the file back to my local VM to work on it.

I have recently been using Parrot Security OS <https://www.parrotsec.org/> its an OS similar to Kali. I really like it, I cant see any benefit to using it exclusively over Kali but it's making a nice change.

```
└─[X]─[root@parrot]─[/home/andrew/Desktop]
└─ #scp -P 22222
barryallen@192.168.1.155:/home/barryallen/security-system.data
/home/andrew/Desktop
#####
# WARNING #
# FBI — Authorized access only! #
# Disconnect IMMEDIATELY if you are not an authorized user!!! #
# All actions Will be monitored and recorded #
# Flag{53c82eba31f6d416f331de9162ebe997} #
#####
barryallen@192.168.1.155's password:
security-system.data 100% 1024MB 58.5MB/s 00:17
```

So having saved it back to my desktop I can work on it locally now, lets see if I can unzip it.

Success it unzips and appears to be a memory dump. Which actually matches up with the clue

Franks Lost His Mind or Maybe it's His Memory. He's Locked Himself

Inside the Building. Find the Code to Unlock the Door Before He Gets Himself Killed!

So the past few weeks I've come to the conclusion that memory forensics is definitely one of my weak points. So I know what I have to do but have no idea how to do it.

I can fix that with a bit of google fu. I think I'm going to try and use the app Volatility (*I need to use it for a cyber EPQ course I am currently enrolled on, so it'll be two birds with one stone*)

I need to take a break here and grab a coffee whilst researching how to use Volatility.

So after a bit of forum trawling I came across this site

<http://resources.infosecinstitute.com/memory-forensics-and-analysis-using-volatility/> and for the next few steps I will be following along with the tutorial. (update, this wasn't so helpful after the first step, for this CTF anyway)

I need to find out what OS the dump came from so that I can apply the correct profile in Volatility to be able to move forward.

```
└─[X]─[root@parrot]─[/home/andrew/Desktop]  
└─── #volatility imageinfo -f /home/andrew/Desktop/security-  
system.data
```

Volatility Foundation Volatility Framework 2.6

INFO : volatility.debug : Determining profile based on KDBG search...

Suggested Profile(s) : WinXPSP2x86, WinXPSP3x86 (Instantiated with WinXPSP2x86)

AS Layer1 : IA32PagedMemoryPae (Kernel AS)
AS Layer2 : FileAddressSpace (/home/andrew/Desktop/security-system.data)
PAE type : PAE
DTB : 0x33e000L
KDBG : 0x80545b60L
Number of Processors : 1
Image Type (Service Pack) : 3
KPCR for CPU 0 : 0xffdff000L
KUSER_SHARED_DATA : 0xffdf0000L
Image date and time : 2016-10-10 22:00:50 UTC+0000
Image local date and time : 2016-10-10 18:00:50 -0400

Cool, ok so it's a memory dump from an XP machine. The next step in the tutorial is to see if there were any running services when the memory dump was captured.

```
└─[root@parrot]─[/home/andrew/Desktop]
└─── #volatility — profile=WinXPSP2x86 pslist -f security-system.data
Volatility Foundation Volatility Framework 2.6
Offset(V) Name PID PPID Thds Hnds Sess Wow64 Start Exit
-----
0x867c6830 System 4 0 57 171 — — — 0
0x86262900 smss.exe 332 4 3 19 — — — 0 2016-10-10 21:59:14
0x8623b978 csrss.exe 560 332 10 423 0 0 2016-10-10 21:59:14
0x865ed020 winlogon.exe 588 332 24 512 0 0 2016-10-10 21:59:14
0x8662d808 services.exe 664 588 15 263 0 0 2016-10-10 21:59:14
0x866a5670 lsass.exe 676 588 25 356 0 0 2016-10-10 21:59:14 0x86358a70
vmacthlp.exe 848 664 1 25 0 0 2016-10-10 21:59:14
```

(THERE WERE ABOUT ANOTHER 30 PROCESSES THAT I'VE TRIMMED OFF HERE)

First lets see what plugins I have available to me

Volatility Foundation Volatility Framework 2.6

Usage: Volatility — A memory forensics analysis platform.

filescan Pool scanner for file objects

filescan

Volatility Foundation Volatility Framework 2.6

Offset(P) #Ptr #Hnd Access Name

0x000000005e52f90 1 1 RW-rw-

\\Device\\HarddiskVolume1\\WINDOWS\\WindowsUpdate.log

0x000000005e531d8 1 1 R — rw-

\Device\HarddiskVolume1\WINDOWS\WinSxS\x86_Microsoft.Windows.
 Common-Controls_6595b64144ccf1df_6.0.2600.5512_x-ww_35d4ce83
 0x0000000005e53270 1 1 R — rw-
 \Device\HarddiskVolume1\WINDOWS\WinSxS\x86_Microsoft.Windows.
 Common-Controls_6595b64144ccf1df_6.0.2600.5512_x-ww_35d4ce83
 0x0000000005e538a8 3 1 R — rwd \Device\HarddiskVolume1\Documents
 and Settings\test\NetHood
 0x0000000005e53df8 1 1 R — rw-
 \Device\HarddiskVolume1\WINDOWS\WinSxS\x86_Microsoft.VC90.CRT_
 1fc8b3b9a1e18e3b_9.0.30729.4148_x-ww_d495ac4e
 0x0000000005e54de8 1 1 R — rw-
 \Device\HarddiskVolume1\WINDOWS\WinSxS\x86_Microsoft.Windows.
 GdiPlus_6595b64144ccf1df_1.0.2600.5512_x-ww_dfb54e0c
 0x0000000005e54e80 1 1 R — rw-
 \Device\HarddiskVolume1\WINDOWS\WinSxS\x86_Microsoft.Windows.
 Common-Controls_6595b64144ccf1df_6.0.2600.5512_x-ww_35d4ce83
 0x0000000005e58300 1 0 R — rw-
 \Device\HarddiskVolume1\WINDOWS\system32\mydocs.dll
 0x0000000005e5bd18 1 1 — — —
 \Device\NamedPipe\Ctx_WinStation_API_service
 0x0000000005e5c2f0 1 0 R — r-d
 \Device\HarddiskVolume1\WINDOWS\system32\ipconfig.exe
 0x0000000005e5c5b0 1 1 RW-rw-
 \Device\HarddiskVolume1\WINDOWS\WindowsUpdate.log
 0x0000000005e5c6f8 1 1 RW-rw-
 \Device\HarddiskVolume1\WINDOWS\WindowsUpdate.log

This is just a quick copy and paste of the HUGE amount of files it pulled
 off the system. So I need to tweak the output slightly lets see if I can use

GREP with Volatility.

```
[root@parrot]—[/home/andrew/Desktop]  
└── #volatility — profile=WinXPSP2x86 -f security-system.data  
filescan | grep 'flag'
```

Volatility Foundation Volatility Framework 2.6

searching for the string flag didnt return anything useful. Let's see if I can search again using jsut the file extension. Let's try with .txt

```
[X]—[root@parrot]—[/home/andrew/Desktop]  
└── #volatility — profile=WinXPSP2x86 -f security-system.data  
filescan | grep '.txt'
```

Volatility Foundation Volatility Framework 2.6

```
0x0000000005e612f8 1 0 -W-r — \Device\HarddiskVolume1\Documents  
and Settings\test\Desktop\code.txt
```

Bingo (I think)!! Ok so how do I read the code.txt file. Let's see if I can just run CAT against it using the file path.

Nope that didnt work, lets go back to the plugin list and see what else might be available to use.

After trying loads, I purely stumbled across the cmdscan plugin and got this returned

```
[root@parrot]—[/home/andrew/Desktop]  
└── #volatility — profile=WinXPSP2x86 -f security-system.data  
cmdscan
```


Volatility Foundation Volatility Framework 2.6

CommandProcess: csrss.exe Pid: 560

CommandHistory: 0x10186f8 Application: cmd.exe Flags: Allocated, Reset

CommandCount: 2 LastAdded: 1 LastDisplayed: 1

FirstCommand: 0 CommandCountMax: 50

ProcessHandle: 0x2d4

Cmd #0 @ 0x1024400: cd Desktop

Cmd #1 @ 0x4f2660: echo 66 6c 61 67 7b 38 34 31 64 64 33 64 62 32 39 62 30 66 62 62 64 38 39 63 37 62 35 62 65 37 36 38 63 64 63 38 31 7d > code.txt

Awesome, it's returned a cmd line input from where the user echoed a hex string into the code.txt file. I still don't know how to read the actual code.txt file using Volatility but I got lucky this time with the cmd line. (I've made a note to go back and do a deep dive on using Volatility)

So let's take the hex input and convert it back to something I can read. Using asciitohex.com it decrypts as the flag :-)

flag{841dd3db29b0fbbd89c7b5be768cdc81} quick jump over to crackstation.net and it returns

841dd3db29b0fbbd89c7b5be768cdc81 MD5 :

Two[space]little[space]mice

A google search to confirm my thought and Two little mice is from the film also. The quote is:

“Two little mice fell in a bucket of cream. The first mouse quickly gave up and drowned. The second mouse, wouldn’t quit. He struggled so hard that eventually he churned that cream into butter and crawled out. Gentlemen, as of this moment, I am that second mouse.”

I completely feel like that second little mouse right now, I almost gave up on that last flag when I wasn’t getting anywhere with Volatility but I’m so glad I kept on going and eventually “churned that cream into butter” haha.

SO much fun, huge thanks to James Bower for creating this VM and for Vulnhub.com for hosting it and all the other VM’s

Infosec

Cybersecurity

Ethical Hacking

Hacking

Ctf



Written by Andrew Hilton

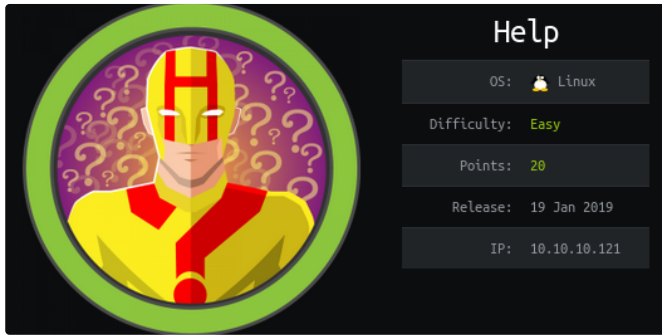
808 Followers

Tactical Threat Manager @ FinTech | Purple Team | Offensive Operations

Follow



More from Andrew Hilton



 Andrew Hilton

HTB: “Help” Walkthrough

So this is one of the first boxes from Hack the Box that I have decided to publish a...

Jul 13, 2019  14



 Andrew Hilton

HTB: “Jerry” Walkthrough

Ok so lets dive in and try to get this box — its rated as easy!!!

Jul 14, 2019  20

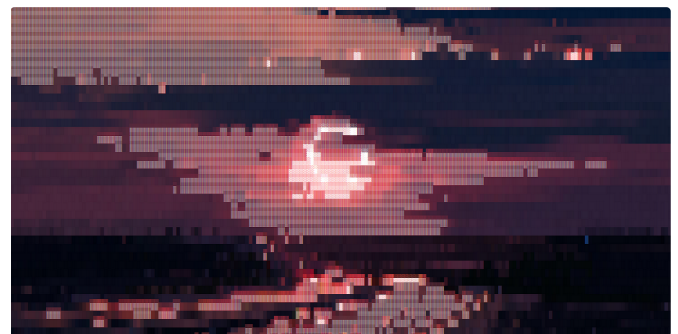


 Andrew Hilton

Web Application Penetration Testing

I was given a PDF a few months back by a friend. It was a result of asking them if the...

Jan 7, 2019  408  4



 Andrew Hilton

Sunset:Nightfall—Vulnhub.com

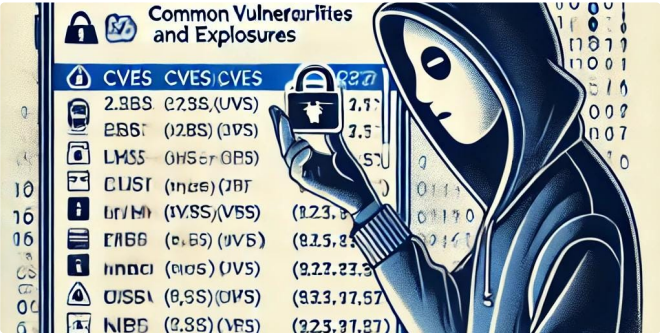
We had a little down time again at work between pen tests so my apprentice and I...


Sep 5, 2019  5



See all from Andrew Hilton

Recommended from Medium



 Jonathan Mondaut

How ChatGPT Turned Me into a Hacker


Discover how ChatGPT helped me become a hacker, from gathering resources to...

★ Jun 18 🖱 1.4K 💬 44 📌⁺

Enter OTP

OTP

Verify

 Yahia Fouda

IEEE Web CTF Challenge (2024)

Hi, i'm Yahia Fouda Cyber Security Researcher and Software Engineer.

6d ago 🖱 17 📌⁺

Lists



Tech & Tools
19 stories · 315 saves



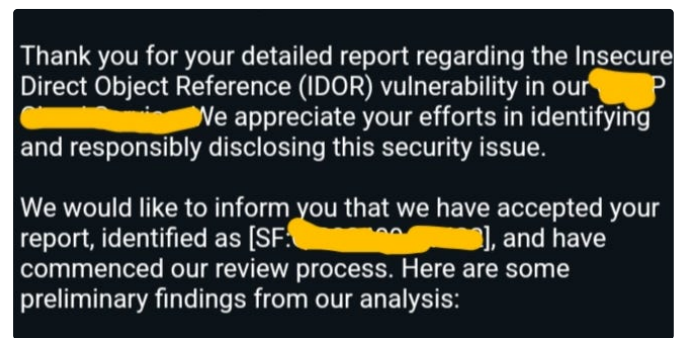
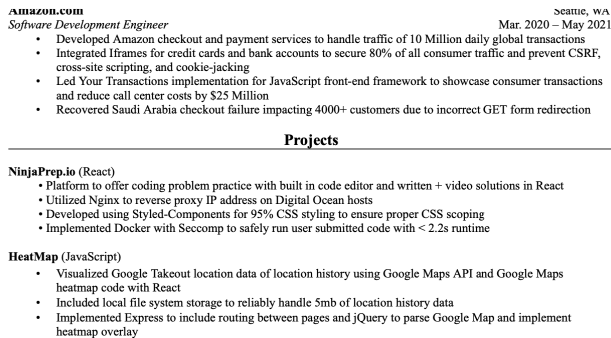
Staff Picks
744 stories · 1341 saves





Medium's Huge List of Publications Accepting...
334 stories · 3631 saves



Natural Language Processing
1738 stories · 1318 saves



 Alexander Nguyen in Level Up Coding

 Being nice pentester

The resume that got a software engineer a \$300,000 job at...

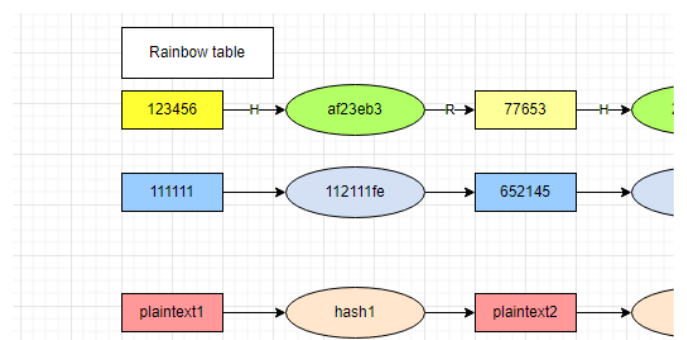
1-page. Well-formatted.

my first easy IDOR in a big company in bug bounty

IDOR is very easiest bug a new hunter can also find it.

★ Jun 1 🖱️ 23K 💬 454 📌⁺

★ Sep 27 🖱️ 251 💬 2 📌⁺



 Devon Price 🧠 in Human Parts

 LORY

Laziness Does Not Exist

Psychological research is clear: when people procrastinate, there's usually a goo...

A basic question in security Interview: How do you store...

Explained in 3 mins.

★ Mar 23, 2018 🖱️ 326K 💬 1683 📌⁺

★ May 12 🖱️ 5K 💬 66 📌⁺

See more recommendations

