PClub Project Report
Ethical Hacking

team

May 2016

Member 1: Tanmay Seth

Basics Completed

- basic html, css, js, sql
- basics of vim text editor
- cyber security course on coursera first week

Challenges completed

- hts basic level 10
- bandit level 1 (recently started and currently working on)
- natas level 3

Others

- made my website (not uploaded it yet)

Timeline of work done

till 22nd May

- covered basics of html, css, js, sql

till 26th May

- challenges of hack this site till basics level 10
- started creation of my homepage
- basics of vim editor

till 29th May

- created my homepage
- first week of cyber security course on coursera

in future

- attempt more challenges and levels.

Member 2: Samarth Chawla

Summary of challenges completed until May 29

Bandit The bandit wargame only focuses on *teaching* command line utilities that might be useful in exploits, not actual exploits

- **Levels 0 - 6:** ssh and basics of command line; `ls, cat, file, find` [1]
- **Levels 7 - 12:** matching text with `grep`, finding a uniquely string with `sort` and `uniq`, decoding base64 data with `base64`, deciphering a ROT13 cipher with `tr`, uncompressing a hexdump with `xxd` and `tar`
- **Levels 13 - 16:** working with network protocols; ssh using a private key, using `telnet` and `openssl` to exhange data, using `nmap` to scan for open ports
- **Level 17:** used `diff` to compare two files and extract the password
- **Level 18:** used `scp` to transfer files from an ssh server that logs you out immediately on logging in due to a modified .bashrc, thus bypassing the shell on the remote PC
- **Levels 19 - 20:** learned about setuid, learned to start network daemons from the command line using `nc`
- **Levels 21-23:** learned about `cron`; in level 23, abused a cronjob running all scripts from a particular directory to copy the password to a particular directory

Natas

- **Levels 0 - 2:** Password commented in source code of levels 0 and 1, simply opened the developer panel to access the password; level 2 had an image from subdirectory `/files`, and the index of `files` contained an unprotected file with the password
- **Level 3:** Comment in source code hinting at Google not being able to find the password. Had to look up the solution after a lot of trying, a directory listed in the site's `robots.txt` had the password
- **Level 4:** Had to again seek help (from Prannay this time) as I could not figure out what to do after a lot of trying. Tried `window.history.pushState("http://natas5.n` first, but found I had to edit the HTTP request header before it got sent, and did so using the Tamper extension on Chromium.
- **Level 5:** Edited the loggedin cookie using `document.cookie="loggedin=1"`

[1] Reading up on `find` was useful for me, as initially I was using `ls` to `grep` for non-executables, then stripping out filenames and piping to `du` and `file`

- **Level 6:** Found the secret from `/includes/secret.inc`
- **Level 7:** Modified `href` of link to `/etc/natas_webpass/natas8`
- **Level 8:** The secret is now encoded into base64, then reversed and converted to hex. Ideally, I would have written either a PHP script or something like a bash or python script, but since I wasn't familiar enough with either, I wrote a c program to convert the hex into ASCII, then ran the output through the unix programs `rev` and `base64_decode`.
- **Level 9:** Injected `; cat /etc/natas_webpass/natas10` into the PHP script to access the password.
- **Level 10:** Since ";—&" are now filtered, (after floundering about for a long while) I input `-e '.' /etc/natas_webpass/natas11 #` to output every line from `/etc/natas_webpass/natas11`

Hackthissite basic missions , except the last basic mission

Hackthissite javascript missions

Summary of tasks completed until June 7

Natas

- **Level 11:** First base64 decoded the cookie, which gave me xor encrypted json data, and json encoded the expected data. xor encrypting one with the other as the key gave me the key which was used. Used that key to encrypt the json encoding of data with "showpassword" set to "yes" and base64 encoded it back into the cookie.
- **Level 12:** This took me a while, and I had to give up and look up the solution. I was thinking about something along the lines of running a bash script, but the fact that this is a server and would happily run any php script I gave it did not occur to me.
- **Level 13:** I looked up how the `exif_imagetype` function works, and a bit of searching led me to magic numbers and file signatures. I simply prepended the file signature for png before a php script and uploaded that.
- **Level 14:** Reeked of SQL injection. I looked up a common tactic for overriding the `where` clause (1=1) and injected that.

Narnia

- **Level 0:** Simple buffer overflow to set the val as `0xdeadbeef`. Ran into a spot of trouble as ASCII only goes up to `0x7f`, but found that `echo -e '\xef\xbe\xad\xde'` works for inserting arbitrary hex values.

Microcorruption

- **New Orleans:** Looking at the disassembled code and stepping through an execution of `check_password`, I found out that my input is being compared against something stored at the location in `r13`. From the memory dump, I found the password stored at location in `r13`.
- **Sydney:** This time, my input is not compared against something in memory, but the password is hard-coded in the program, viewable in disassembly as hex values.

Backdoor

- **2013-bin-50:** Running `strings` on the binary throws up lines of the form "The password is something". The password is indeed one of those.
- **2013-web-50:** Used JavaScript injection to change cookie username to admin
- **file reader:** Simply typed ".../flag.php"
- **authorized persons only:** Another cookie change

Completed Week 2 of Software Security course on Coursera

Member 3: Prannay Khosla

Work done till now

- Completed till Level 17 NATAS, Level 10 of HTS Basics and all Javsascript challenges on HTS.
- Completed till week 2 of the coursera course of Software Security by the University of Maryland.
- Studied basic of Architecture and 3 address code and a little bit about processors.

Work was done in a discontinous manner without a proper timeline, since more time was spent on learning and developing rather than actual hacking.

- *NATAS Level 0 to Level 2* : Basic HTML source code checking, browser tools etc
- *NATAS Level 3 to Level 5* : Basic scripting and request tampering
- *NATAS Level 6 to Level 8* : Basic PHP injection and reverse engineering from source code.
- *NATAS Level 9 to 11* : Injecting bash commands from PHP and reverse engineering from known encryptions.
- *NATAS Level 12 to 13* : Injecting files that run system commands into the website to hack data.
- *NATAS Level 14 to 17* : SQL injection using arguments, sometimes requiring python based brute forcing.
- *HTS Basic Level 1 to 10* : Basic PHP, SQL injection and command line injection
- *HTS Javascript Level 1 to 7* : All challenges were trivial except the last which required brute forcing from the console.

Future work:

- Work forward with system architecture and reverse engineering binaries.
- Move on to Narnia wargames for hacking
- Start blogging about good challenges.

Member 4: Jaismin Kaur

Work done so far

- Basics of HTML, CSS JavaScript, SQL, and begun PHP.
- written the code for the homepage, request is yet to be approved.
- on week two of software security course on coursera, project one is yet to be done.
- on level 15 of bandit, basci level 8 on HTS, level 4 natas

summary of challanges

* bandit level 0-6 : basic commands like ls, cd, cat, du, to look for the password
* bandit level 6-10 : working on commands like grep, strings, tar, etcetra
* bandit level 10-15 : working with private ssh keys, decoding data with base64, working with network protocols
* basic HTS : working with source codes, editing source code to send password to desired email,working with UNIX command line

  future tasks
  - going with software security course on coursera
  - do basics of PHP
  - learn assembly language
  - progress with challenges
* learnt commands like ls, cd, cat, file, du, grep, sort ,tr, base64, strings, uniq, ssh, etcetra

Member 5: Bhuvi Gupta

Work done till now

As a part of the ethical hacking project, I have completed the following:

- Basics of HTML, CSS(Moderate), JS, SQL, PHP(beginner)
- Completed Week 2 of the coursera course on Software Security, Project 1 is still pending
- Level 18 of bandits, level 10 of natas, basic challeges in HTS level 10

Time line of work done

Till 20th May

- Basics of HTMl, CSS, JS, SQL
- Udacity's course on git and git hub
- Basics of vim
- Week 1 of Software Security(Coursera), excluding project
- Creation of home page directory

TILL 25TH MAY

Bandits level 15 *First Five Levels*

- Basic terminal commands such as:ls, la, escaping special characters, du, file, cat, find
- Connect to a server using ssh

Next Five Levels (6-10)

- Terminal commands like grep, tar, bzip2, gzip, xxd, base64, diff, sort, uniq, tr

Next Five Levels(10-15)

- Terminal commands such as:tr, telnet, nmap, openssl
- Connect to a server using ssh key

Natas level 5

- Basic intro to using the developer panel
- robots.txt
- Exploiting vulnerabilities in functions which mail the flag on a specified mail id

Hack this Site Java script challenges

- Completed all levels except for last

Till 28th May

Bandits level 19 *Level 16-19*

- Terminal commands such as:nmap, diff

Natas level 6-10

- Accessing password by exploiting general terminal commands such as grep and cat.
- Using tamper to change referer
- Basic intro to cookies
- Basic intro to php

Hack this Site Basic Challenges

- Increased familiarity with web hacking

Week 2 of coursera course Basic knowledge about the following:

- Stack canaries
- ROP
- Return to libC

As of 8th June 2016

The following occur in no particular order of completion

- Completed Project 1 of the coursera course

  The project was about exploiting buffer overflow vulnerabilities. A c code needed to be examined and a stack smashing attack needed to be carried out. Project involved the use of gdb primarily(and virtual box too). Learnt the basic usage of gdb like setting break points, printing address of variables/registers.

- Two levels of microcorruption.com

  Level 1 I spent a considerable duration of time on this, given my unfamiliarity with assembly. Although in the end, I realized this level was a cake-walk. Learnt basic assembly instructions like push, move, move.b, cmp, jz, je etc. The password of this level was stored in a register upon calling the getpassword() function. One could easily track the register and get the password.

  Level 2 I was stuck on this level for a long time too. I was not taking into account 'little endianness'. Once I learnt about that, it was easy peasy.

- Backdoor CTF, SDSLabs

  I succesfully completed two levels.
  One of these required the use of hashing. Learnt about md5 and sha-256 hashing The other only required to tamper with the cookie, i order to be able to view the page as 'admin'.

- Micellaneous

  Read a bit and tried to implement some basic stuff in jekyll, in order to be able to maintain a blog in future. Hope to upload it soon on gh-pages.