How to [Holiday] Hack It

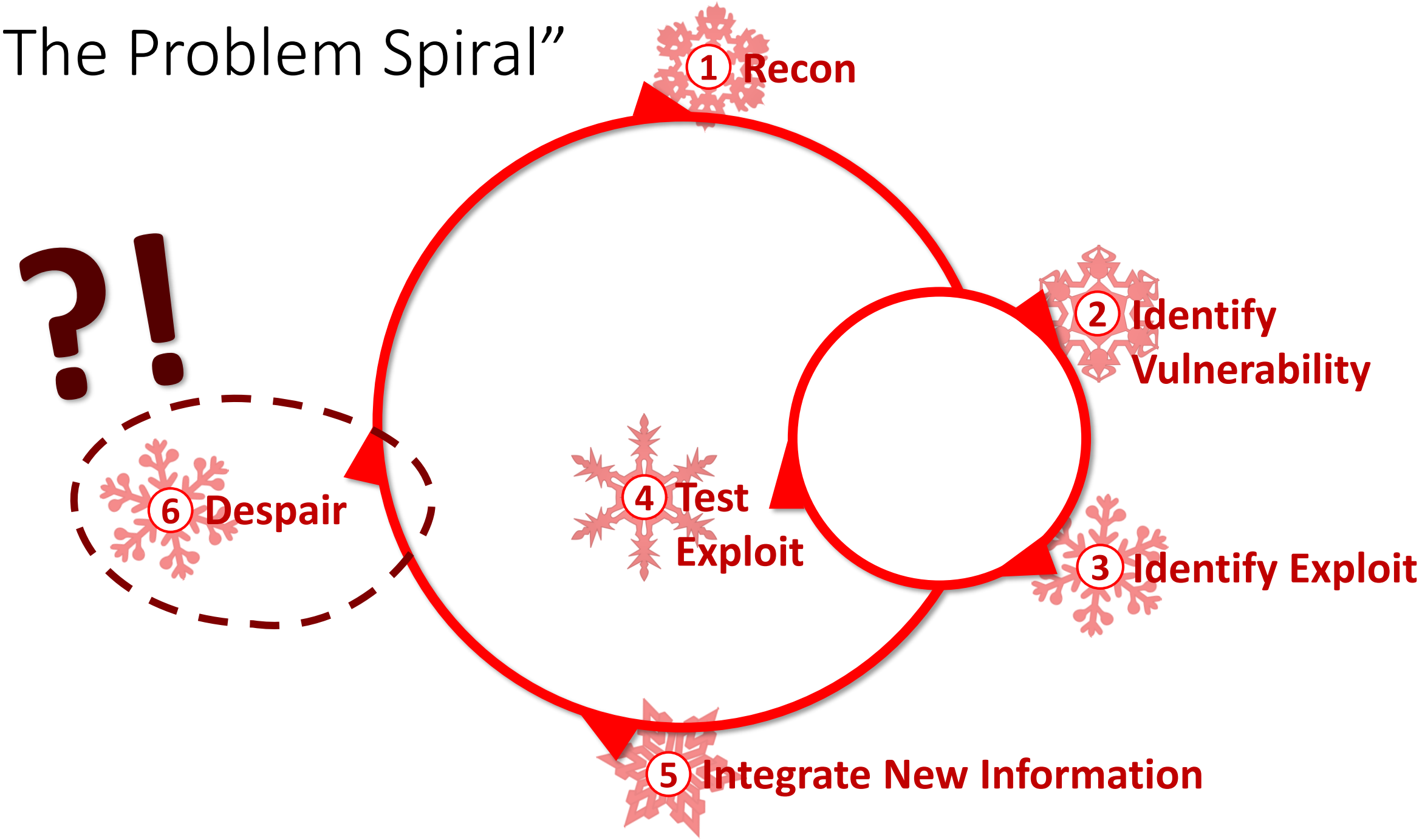
Tips for Crushing CTFs & Pwning Pentests

KringleCon 2019

SANS Holiday Hack

What if <u>EVERYONE</u> solved this year's Holiday Hack?

“The Problem Spiral”



We've even got a CTF right here!!

Problem Solving

- Capture the Flag (CTF) is a fantastic way to build skills:
  - Understanding Attackers
  - Bug Bounty Programs
  - Penetration Testing
- We can't learn the answer to literally everything...
- How can we get unstuck on tricky challenges?
- **Bonus Tier:**

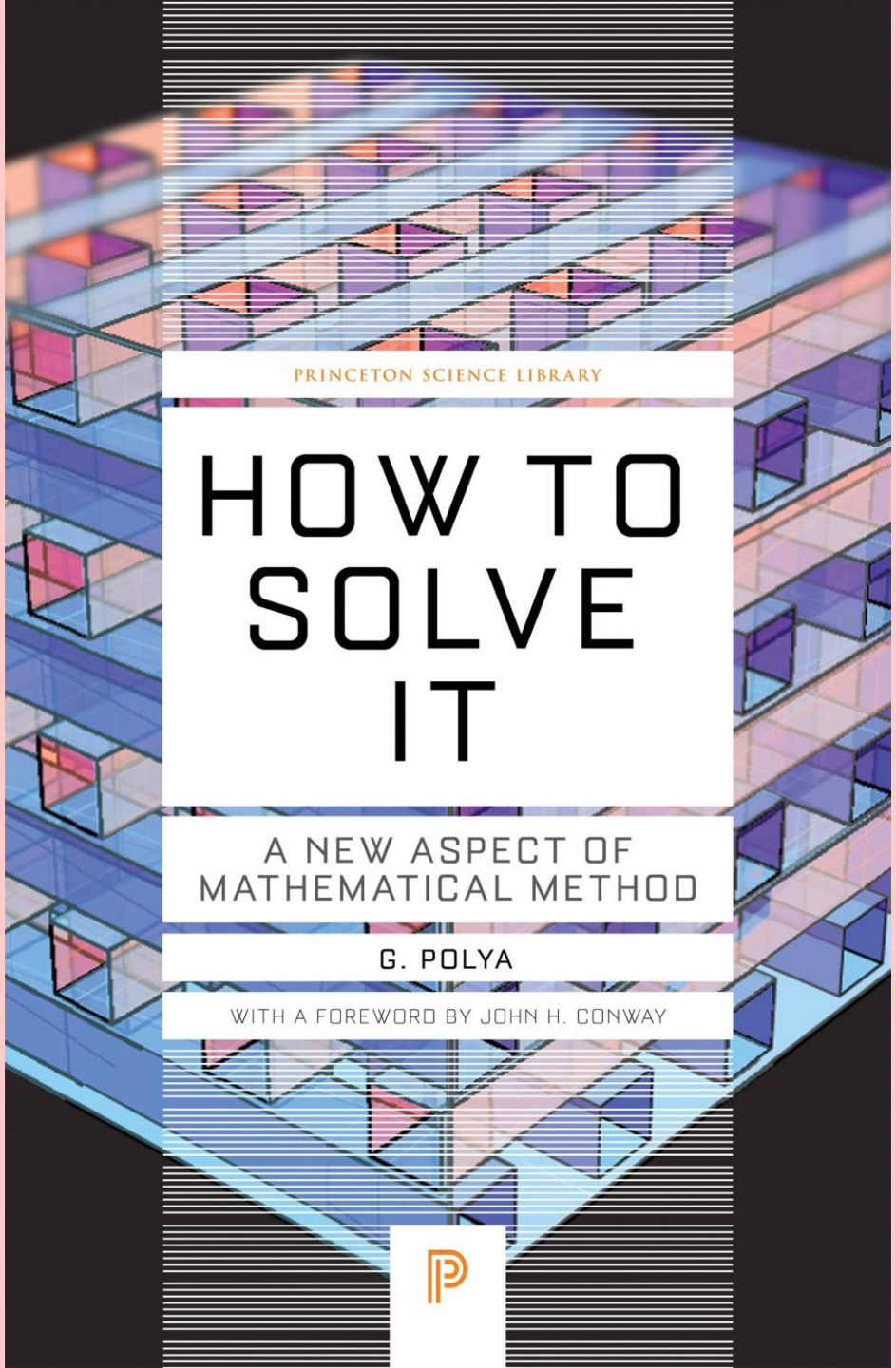
  ...with less Google?

Our Steps:

- *What* exactly are we hacking?
- How can we hack it…
  - …using previous information?
  - …using similar information?
  - …using our awesome brains?
- Go hack it!!
- Looking Back

Katie Knowles *(@_sigil)*

- Security Consultant, F-Secure Consulting
- OSCP, GPEN, CREST CRT



UNDERSTANDING THE PROBLEM

First.
You have to *understand* the problem.

What is the unknown? What are the data? What is the condition? Is it possible to satisfy the condition? Is the condition sufficient to determine the unknown? Or is it insufficient? Or redundant? Or contradictory?

Draw a figure. Introduce suitable notation.

Separate the various parts of the condition. Can you write them down?

DEVISING A PLAN

Second.
Find the connection between the data and the unknown. You may be obliged to consider auxiliary problems if an immediate connection cannot be found. You should obtain eventually a *plan* of the solution.

Have you seen it before? Or have you seen the same problem in a slightly different form?

Do you know a related problem? Do you know a theorem that could be useful?

Look at the unknown! And try to think of a familiar problem having the same or a similar unknown.

Here is a problem related to yours and solved before. Could you use it? Could you use its result? Could you use its method? Should you introduce some auxiliary element in order to make its use possible?

Could you restate the problem? Could you restate it still differently? Go back to definitions.

If you cannot solve the proposed problem try to solve first some related problem. Could you imagine a more accessible related problem? A more general problem? A more special problem? An analogous problem? Could you solve a part of the problem? Keep only a part of the condition, drop the other part; how far is the unknown then determined, how can it vary? Could you derive something useful from the data? Could you think of other data appropriate to determine the unknown? Could you change the unknown or the data, or both if necessary, so that the new unknown and the new data are nearer to each other?

Did you use all the data? Did you use the whole condition? Have you taken into account all essential notions involved in the problem?

CARRYING OUT THE PLAN

Third.
Carry out your plan.

Carrying out your plan of the solution, *check each step*. Can you see clearly that the step is correct? Can you prove that it is correct?

LOOKING BACK

Fourth.
Examine the solution obtained.

Can you *check the result?* Can you check the argument?

Can you derive the result differently? Can you see it at a glance?

Can you use the result, or the method, for some other problem?

How To Solve It

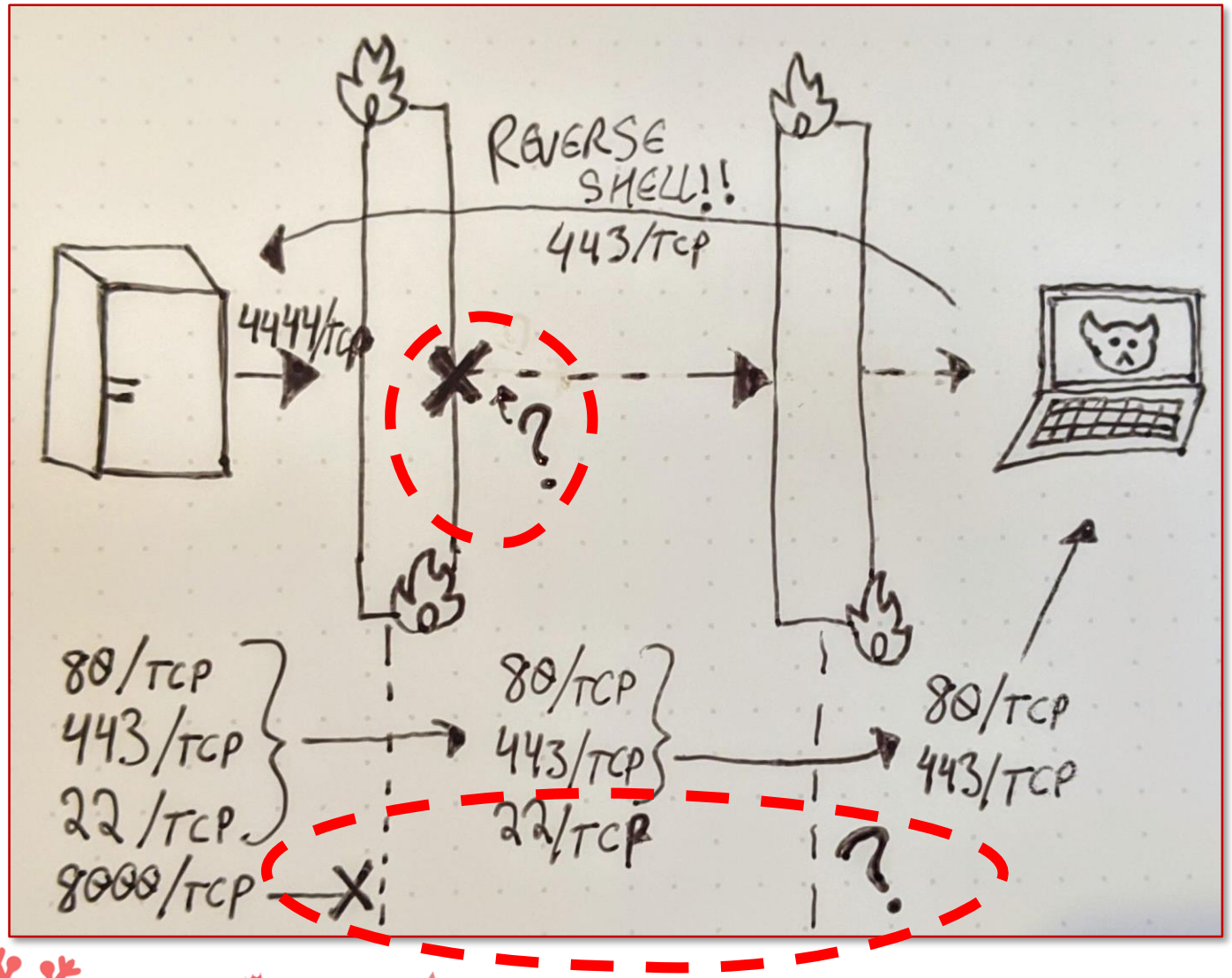
Part 1.

Understanding the Problem

What Are We Hacking?!

- What do we know about the system?
  - What ports, services, versions, software?
  - What does it do?
  - How does it respond?
- Draw it out:
  - System
  - Services
  - Network

  …in relation to other systems & services



Part 2.

Devising a (Devious!) Plan

This Looks Familiar…

- Is there a blog post or challenge write-up about a vulnerable system like this?
- What about a system with a similar service, or OS?
  - How was it solved?
  - What issues did it relate to?
  - How did it work?

CTFtime.org / Writeups

ctftime.org/writeups?tags=rce&hidden-tags=rce

Event	Task	Tags	team
RITSEC CTF 2019	potat0	web rce	r00tn3p@
InCTF 2019	PHP+1	php rce	CBF
FAUST CTF 2019	punchymclochface	python rce cobol punchcard	Bushwha
*CTF 2019	mywebsql	cve perl rce mywebsql	rawsec
ångstromCTF 2019	Server	asm http rce web	Nekocha
0CTF/TCTF 2019 Quals	Wallbreaker Easy	imagick disable_functions web rce	GoN

SANS Penetration Testing

07 Dec 2016

Getting MOAR Value out of PHP Local File Include Vulnerabilities

2 comments Posted by eskoudis
Filed under (no category specified)
By Jeff McJunkin

Wouldn't web application penetration testing be easier if you could look at the source code? Well, when looking to expand my web app pen testing skills, my good friend and co-worker, Josh Wright, mentioned a specific new twist for Local File Include vulnerabilities on PHP-based web servers: PHP wrappers.

fuzzySecurity 2.0 Home Tutorials

Home » Tutorials » Windows Privilege Escalation Fundamentals

Windows Privilege Escalation Fundamentals

Not many people talk about serious Windows privilege escalation which is a shame. I think the reasons for this are probably (1) during penetrating engagements a low-priv shell is often all the proof you need for the customer, (2) in staged environments you often pop the Administrator account, (3) meterpreter makes you lazy (getsystem = lazy-fu), (4) build reviews to often end up being --> authenticated nessus scan, microsoft security baseline analyser…

Contrary to common perception Windows boxes can be really well locked down if they are configured with care. On top of that the patch time window of opportunity is small. So lets dig into the dark corners of the Windows OS and see if we can get SYSTEM.

OK, We're Stuck

- What hints have we been given?
  - How could they be interpreted?
  - Could we interpret them differently?
- Can we describe where you're stuck, in as much detail as possible?
- Have we used all the information we found enumerating?
- Are we missing any information? Could we enumerate more?


```
root@computer:~# nmap -p- 192.168.0.10
Starting Nmap 7.70 ( https://nmap.org )
Nmap scan report for 192.168.0.100
Host is up (0.080s latency).
Not shown: 65521 closed ports
PORT      STATE SERVICE
21/tcp    open  ftp
22/tcp    open  ssh
79/tcp    open  finger
111/tcp   open  rpcbind
512/tcp   open  exec
513/tcp   open  login
514/tcp   open  shell
2049/tcp  open  nfs
5432/tcp  open  postgresql
34716/tcp open  unknown
35328/tcp open  unknown
50948/tcp open  unknown
57565/tcp open  unknown
59197/tcp open  unknown
```

?

1. *Scanned 10.0.0.2*
2. *Accessed https://10.0.0.2*
3. *Uploaded reverse shell to file upload function*
4. *STUCK:*

- *Can't get a shell back!*
- *Is it getting blocked?*
- *Does the shell work?*

Let's try it locally…

Part 3.

Carrying Out the Plan

Hack It!!!

- Did it work?
  - …If it didn't, where and when did it fail?
  - What step or detail could we debug?
- Is there another way to solve this?
- Can new access get us more information?
- Revisit previous steps as needed.

```
msf5 exploit(windows/smb/ms17_010_eternalblue) > run

[*] Started reverse TCP handler on 127.0.0.1:4444
[-] [redacted]:445 - Exploit aborted due to failure: not-vulnerable
[*] Exploit completed, but no session was created.
msf5 exploit(windows/smb/ms17_010_eternalblue) >
```

```
msf exploit(linux/postgres/postgres_payload) > run

[*] Started reverse TCP handler on [redacted]:444
[-] Connection failed
[*] Exploit completed, but no session was created.
```

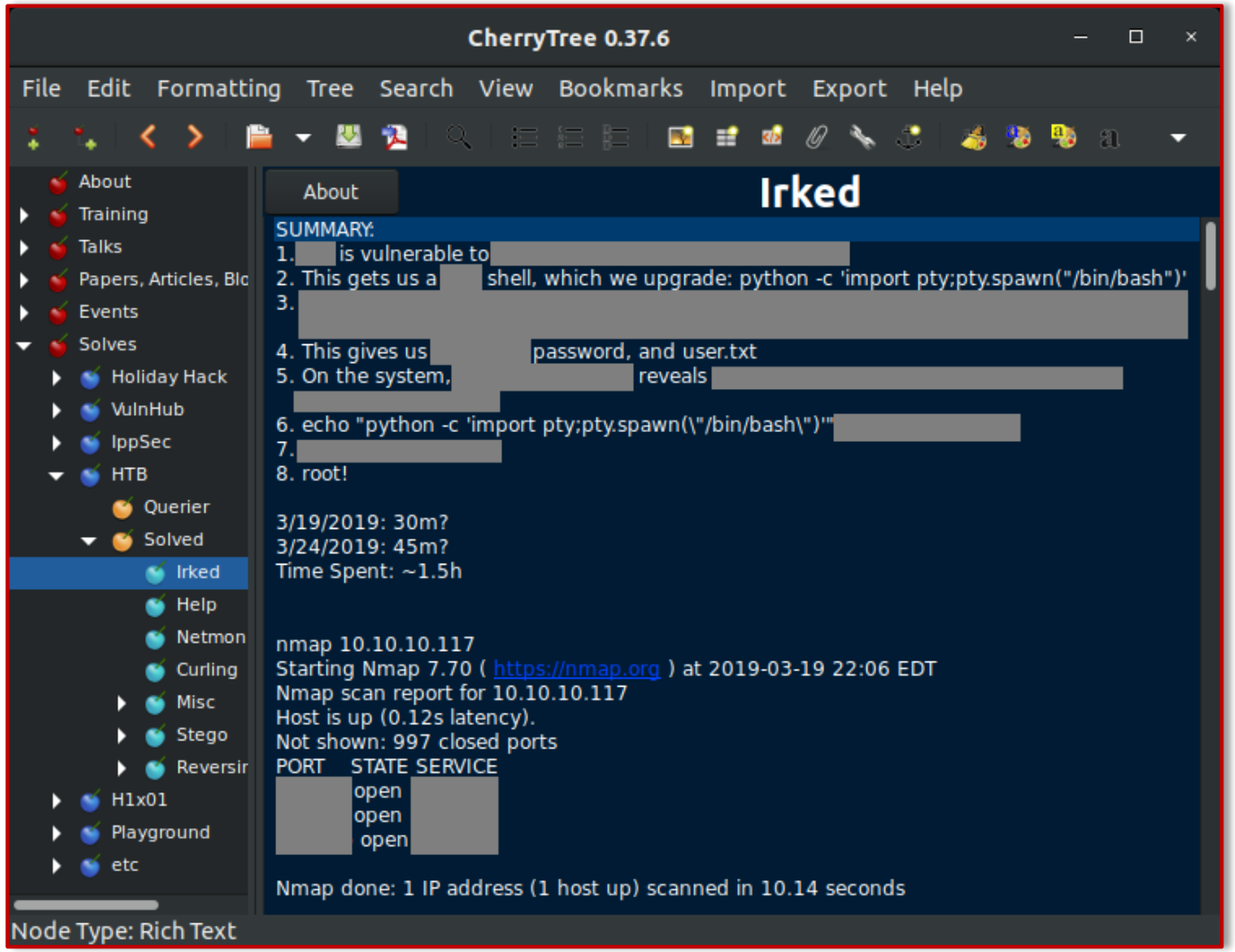
```
msf exploit(windows/smb/ms08_067_netapi) > exploit

[*] Started reverse TCP handler on [redacted]:4444
[*] [redacted]:445 - Automatically detecting the target...
[*] [redacted]:445 - Fingerprint: Windows 2003 R2 - Service Pack 2 - lang:Unknown
[*] [redacted]:445 - We could not detect the language pack, defaulting to English
[*] [redacted]:445 - Selected Target: Windows 2003 SP2 English (NX)
[*] [redacted]:445 - Attempting to trigger the vulnerability...
[*] Exploit completed, but no session was created.
```

Part 4.

Looking Back

Reflect

- Pick a solution:
  - OneNote?
  - Cherrytree?
  - Markdown?
- Questions:
  - How'd we solve it?
  - What scans, tools, and changes?
  - Useful links?
  - Record your thoughts and notes for next time as personal reference.



How to Hack It!

1. Understand the Problem
   - What do we know about the system? What ports, services, versions, software? What does it do? How does it respond?
   - Draw the system, services, or network in relation to other systems & services.
2. Design a (Devious) Plan
   - Is there a blog post or challenge write-up about a vulnerable system like this? What about a system with a similar service, or OS? How was it solved? What issues did it relate to, and how did it work?
   - What hints have we been given? How could they be interpreted? What if we interpreted them differently?
   - Summarize the steps we've taken. Can we describe where we're stuck, in as much detail as possible? Are we missing any information?
   - Have we used all the information we found while enumerating?
3. Execute the Plan
   - Did it work? If it didn't, when did it fail? What step or detail should we debug?
   - Could there be another way to solve this? Can our increased access get us more information to help understand other vulnerabilities?
   - Revisit steps 1 & 2 as needed.
4. Looking Back
   - How'd we solve it? What steps, tools, changes? Useful links?
   - Record your thoughts and notes for next time as personal reference.

HOLIDAYHACKCHALLENGE.COM

SANS HOLIDAY HACK CHALLENGE 2019

#HOLIDAYHACK

@KRINGLECON

Happy Holidays!

- When you're stuck, reframe your thinking for a fresh perspective
- Tackle problems one step at a time
- Take regular cocoa breaks!! ☕

& Enjoy KringleCon!

katie@kknowl.es
@_sigil
/kaknowles