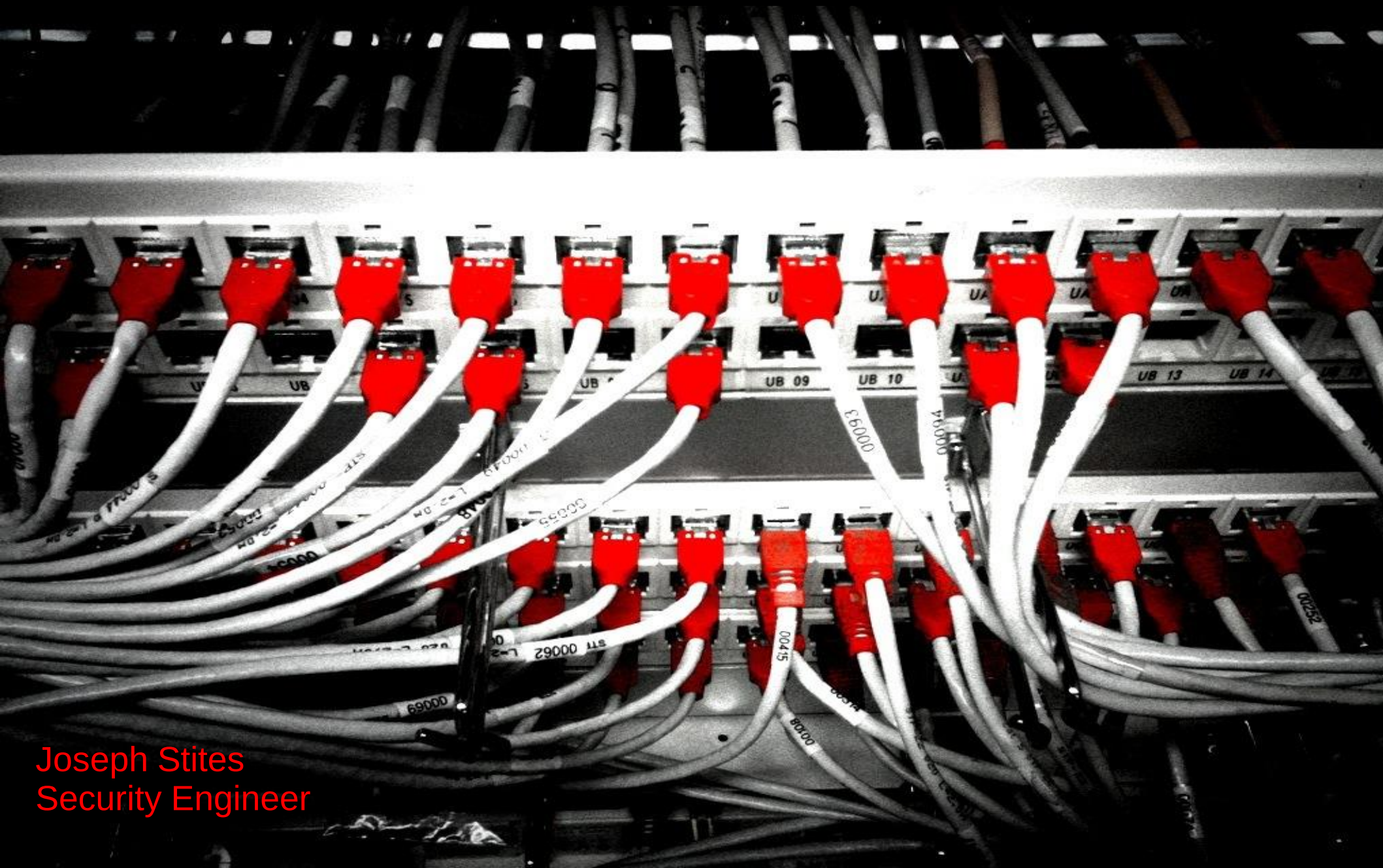
Creating and Leveraging Computer Science Competencies for Information Assurance Career Opportunities [1]



Joseph Stites
Security Engineer

Disclaimer

I do not work for Tennessee Technological University, the Board of Regents, or the Department of Computer Science

Defended thesis in November of 2012

The views and opinions expressed in this presentation are those of the author and do not necessarily reflect the official policy or position of my employer.

Please don't break the law. There is no parole in the Federal system.

- Background
- Security Job Search Lessons Learned
- Things to read
- Things to do

Background

- Advisor Dr. Siraj
- Security Concentration
- Worked as a Teaching Assistant:
  - CSC 2100
  - CSC 2110
- Research Topics:
  - Security Education
  - Smart Grid Security
- Currently employed as a Security Consultant

Security Job Search Lessons Learned

- Interviews
- Technical Interviews
- Technical Skills Evaluation

Security Job Search Lessons Learned

- Interviews
  - Group/Panel Interviews
  - Phone Interviews
  - Stress Interviews

Security Job Search Lessons Learned

- Technical Interviews
  - Describe and Explain Common Security Concepts
    - Example:
      - Explain the difference between public and private key cryptography
  - Answer Basic Security Questions:

    Example:
      - Questions from a Security Final at Tech during job interviews

Security Job Search Lessons Learned

- Technical Skills Evaluation
  - Security Testing Skills:
    - Given a URL you have 48 Hours to identify as many vulnerabilities as possible in a web application
    - On site test on a virtual network
  - Write code:
    - Remote into a shared workspace and write code to solve problems
    - Verify that the application works before the interview
    - Verify that you have a comfortable place to write code and talk on the phone
    - Talk through the code you write

Things to Read

- Books
- Blogs
- Papers
- Standards

Things to Read

- Books:
  - Ross Anderson
    - Book: “Security Engineering”
      - Over a 1,000 Pages
      - Free online

Things to Read

- Blogs and Websites:
  - WIRED
    - Blog: “Threat Level”
  - Cambridge University
    - Blog: “Light Blue Touch Paper
  - OWASP
  - hackerhighschool

Things to Read

- Papers and Essays:
  - Ross Anderson
    - Paper: “How Cryptography Fails”
  - Fred Brooks
    - Essay : “Mythical Man Month”
  - Ambareen Siraj
    - Paper: “Information assurance measures and metrics-state of practice and proposed taxonomy“

Things to Read

- NIST standards
- FIPS standards
- Your industry's standards

Things to Do

- Build a lab
- Write code
- Things Graduate Students can do

Build a lab

- Download VirtualBox or VMPlayer

Build a lab

- Build an Attack Machine:
  - Find an assurance testing distribution or build your own
- Build a Target Machine(s)
  - Find a good target or build your own

Build a lab

- Target Machine(s)
  - Download Operating Systems:
    - DVL
    - ADHD
  - Download Vulnerable Applications:
    - SuperSecureBank
    - DVWA
    - Mutilidae
  - Write your own vulnerable Applications:
    - Implement common security vulnerabilities
  - See also: Cybereagles site

Build a lab

- Set up a virtual network between your attack machine and your target machine
- Save a snapshot before you start
- Verify that the network is isolated

Write your own vulnerable code

- Use off the shelf technology
- Use your MSDN Account:
  - Visual Studio
    - C#
    - Visual Basic
  - SQL Server
- Write it the fastest, dirtiest way possible

Write your own vulnerable code

Randori is a training tool that can generate:

- vulnerable source code
- key describing the flaws in the file or files created

from XML templates.

Randori

```
python randori.py -i authN.xml -n 30
##############################################
randori engine
##############################################
  The files have been created
```

Randori

```
import hashlib
import base64
import getpass

def hashIt(password):
        m = hashlib.md5()
        m.update(password)
        passHash = m.hexdigest()
        passHash64 = base64.b64encode(str(passHash))
        return passHash64
username = str(raw_input("Please enter your username: "))
password = getpass.getpass()
g = open("1.password", 'r+')
authenticated = False
userFound = False

for line in g:
        if line.find(username) != -1:
                userFound = True
                verifyPass = hashIt(password)
                if verifyPass in line:
                        authenticated = True
if userFound == False:
        print "This user was not found"
if authenticated == True:
        print "Access granted"
else:
        print "Access denied"
```

Randori

```
admin:NWY0ZGNjM2I1YWE3NjVkNjFkODMyN2RlYjg4MmNmOTk=
```

Randori

```
This is an unsalted md5 hash.
This is not a secure hashing algorithm.
Hashes should be salted.The password is stored locally.
The password file is accessible by the user.
The password is an md5 hash of 'password'
There is no security warning or banner for users accessing this system
```

Things Graduate Students can do

- Make every research project a security project
- Be bold. Try to publish everything.

Things Graduate Students can do

- Get a job as a TA:
  - 30 students per semester
  - 4 programs a semester
  - 120 applications to grade/test

“You can’t cross the sea merely by standing and staring at the water.”

- Rabindranath Tagore

References

1. network by twicepix: https://www.flickr.com/photos/twicepix/4333178624/sizes/o/in/photostream/

contact

cybereaglefeedback@gmail.com

Links

Book: “Security Engineering”

http://www.cl.cam.ac.uk/~rja14/book.html

Blog: “Threat Level”

http://www.wired.com/category/threatlevel/

Blog: “Light Blue Touch Paper

https://www.lightbluetouchpaper.org/

OWASP

https://www.owasp.org/index.php/Main_Page

hackerhighschool