

Capture The Flag

Team Members: Winnona DeSombre, Jackie Faselt, Ramiro Sarabia, Charlie Yang

Executive Summary:

During the Capture the Flag Game our team successfully found and exploited vulnerabilities in the given target server. The vulnerabilities identified included: weak password requirements, missing encryption of sensitive data, and cleartext transmission of sensitive information. Structural problems we found were use of an outdated, insecure operating system with too many open ports. These weaknesses have contributed to public cyber attacks such as the those associated with Hillary Clinton's 2016 campaign, as well as the Dyn denial of service attack. To create a more secure server we recommend that the owner should require stronger password requirements, encrypt sensitive data, and use a more secure OS.

Introduction:

The goal of the Capture the Flag game was to find and exploit vulnerabilities in a target server: 192.168.1.135 and also identify structural flaws in the system. During class time all of the teams were given the given the target server and we attempted to compromise the system by gaining access and leaving a note on the system desktop. No team was able to do this, however many other vulnerabilities were identified and information gathered. We used tools ranging from rdesktop to Burp Suite. Using these tools and others on our Virtual Machines we were able to find which commercial off-the-shelf (COTS) software packages were used as well as the four system users. The sections below provide information on the tools and methods used to make these finding as well as recommendations and policy implications.

Tools and Methods:

This simulation did not require any sophisticated or premium software; all the tools that we ended up using were free or built into our systems. Using only six tools, we were able to find multiple vulnerabilities and gain access to the server in which the casino site was located.

- **Whois** : showed that the system was running on a Windows 2008 server
- **nmap**: we used the command `nmap -p 1-65535 -sV -sS -T4 192.168.1.135` to identify which ports were open. Among the open ports was port 3000, which we

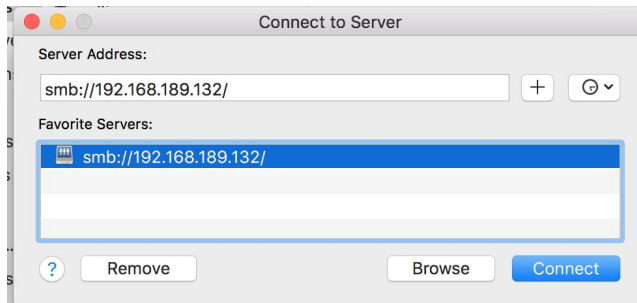
used to then access the website at 192.168.1.135:3000. Additional ports that were open are listed below:

[illegible]

- **sqlmap:** We tried using sqlmap to identify sql injection vulnerabilities. Due to time constraints, however, we pursued other options for breaking into the system. Sqlmap also requires an http request url which we did not have.
- **Burp Suite:** We used the free version of Burp Suite, a security testing software, to look at the packets being sent between us while communicating with the

server. This showed us that session IDs, credit card information, and other vulnerable information was being sent in plain text.

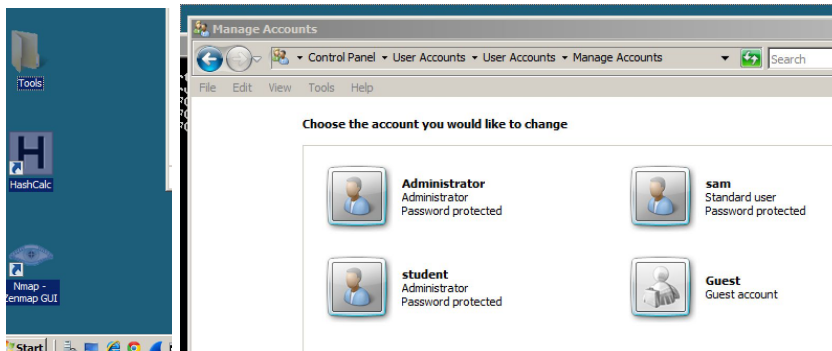
- **Mac Built-in Server Connect:** We attempted to see if we could access shared files on the server by using the built in Mac “Connect to Server” function. If we were able to establish a connection, we could determine that there were shared files on the server. Furthermore, if we were able to log in after establishing the connection, we could see what shared files were on the server. We were able to establish a connection, but were unable to login.



- **Chrome Developer Tools:** By inspecting elements on the webpage in the casino site, we were able to find the credit card numbers of players stored in plaintext within the html.

```
<select name="account">
  <option value="1234-99920-3013">HacmeBank Acct No:
  xxxx-99920-3013</option>
  <option value="123-345-22-043">Offshore Bermuda Acct
  No: xxx-xxx-xx-xxx</option>
</select>
```

- **rdesktop:** We used rdesktop to access the Windows 2008 server where the casino was hosted. Once gaining access to the desktop administrator login using “p@ssw0rd” (thanks for the hint), we were able to find account users.



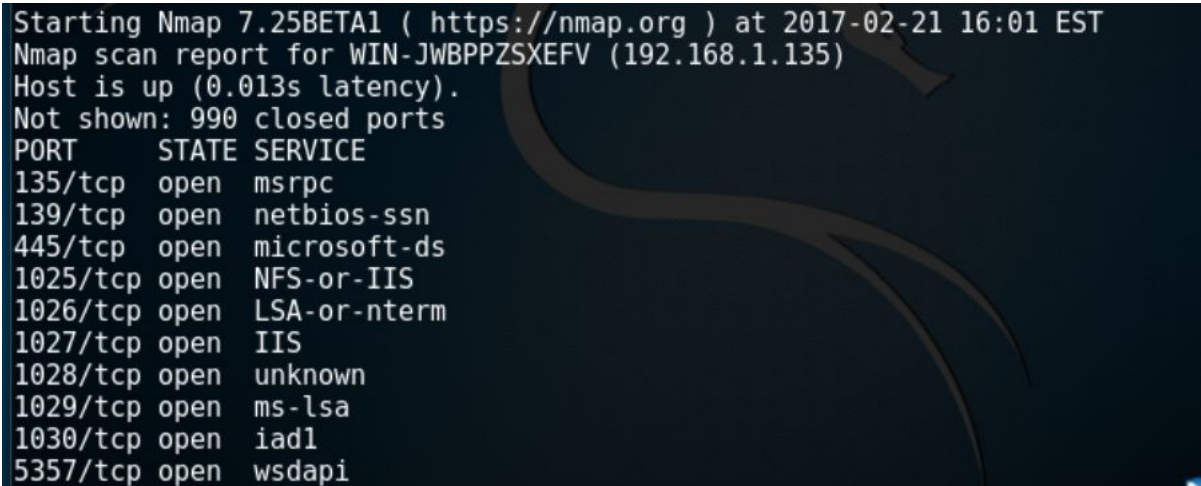
Findings:

Operating System:

- Windows Server 2008

Services / Open Ports:

- 135 (msrpc)
- 139 (netbios-ssn)
- 445 (microsoft-ds)
- 3000 (http)
- 21 (periodically) (ftp)
- Among others (see below)



```
Starting Nmap 7.25BETA1 ( https://nmap.org ) at 2017-02-21 16:01 EST
Nmap scan report for WIN-JWBPPZSXEfv (192.168.1.135)
Host is up (0.013s latency).
Not shown: 990 closed ports
PORT      STATE SERVICE
135/tcp    open  msrcpc
139/tcp    open  netbios-ssn
445/tcp    open  microsoft-ds
1025/tcp   open  NFS-or-IIS
1026/tcp   open  LSA-or-nterm
1027/tcp   open  IIS
1028/tcp   open  unknown
1029/tcp   open  ms-lsa
1030/tcp   open  iad1
5357/tcp   open  wsdaapi
```

Commercial Off-The-Shelf (COTS) Software Packages:

- Wireshark
- setup files
- Various executables for DDoS
- Nmap zenmap GUI
- Hxd - hex editor
- Hashcalc
- Hacme Server Casino START

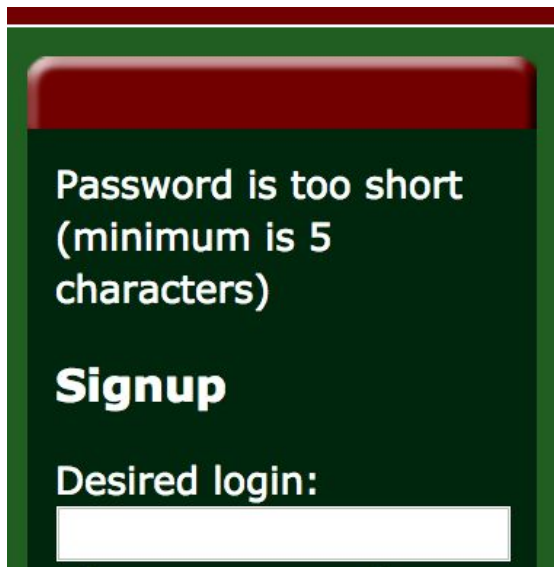
Users:

- Administrator (whose password was p@ssw0rd)
- Sam
- student
- guest

Vulnerabilities:

The target server had the following vulnerabilities (please note that the screenshots of vulnerabilities that we found with the tools we used are above):

- CWE-521: Weak Password Requirements



Not only did the server have a password that was easy to guess, but the website itself imposed only a minimum password requirement of 5+ characters. This is seen when one tries to register an account for the casino site at <http://192.168.1.135:3000/?signup=true>. To mitigate this vulnerability the owner of the server should impose Minimum and maximum password length requirements and mixed character set requirements. We also recommend expirations for passwords.

- CWE-311: Missing Encryption of Sensitive Data/ CWE-319: Cleartext Transmission of Sensitive Information

```
▼ <select name="account">
  <option value="1234-99920-3013">HacmeBank Acct No:
  xxxx-99920-3013</option>
  <option value="123-345-22-043">Offshore Bermuda Acct
  No: xxx-xxx-xx-xxx</option>
</select>
</td>
```

By using Burpsuite, we were able to determine that on the login page of the casino website (<http://192.168.1.135:3000>) was sending username and password information in cleartext. Moreover, the credit card numbers were showing up in the html of the webpage of <http://192.168.1.135/account/options>. It is thus possible to steal money from one account to another. To mitigate this vulnerability the owner of the server should use encryption throughout the system. We recommend the use of strong, current cryptographic algorithms, and the use of encryption before the transferring of critical or sensitive data. Additionally, web application should use SSL from login time to logout time.

- CWE CATEGORY 16 - Configuration

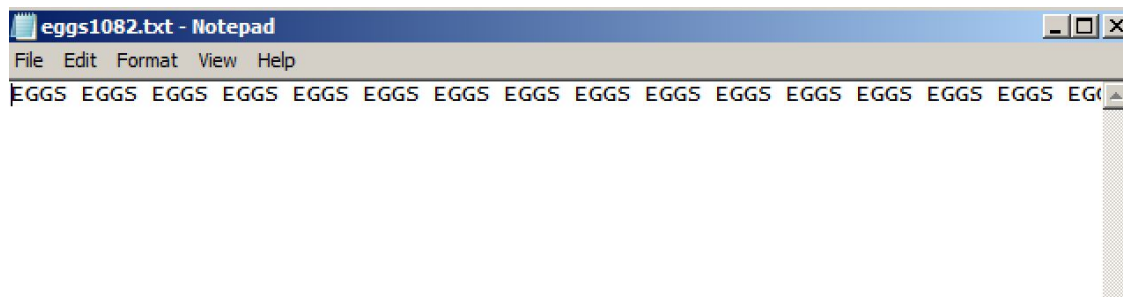
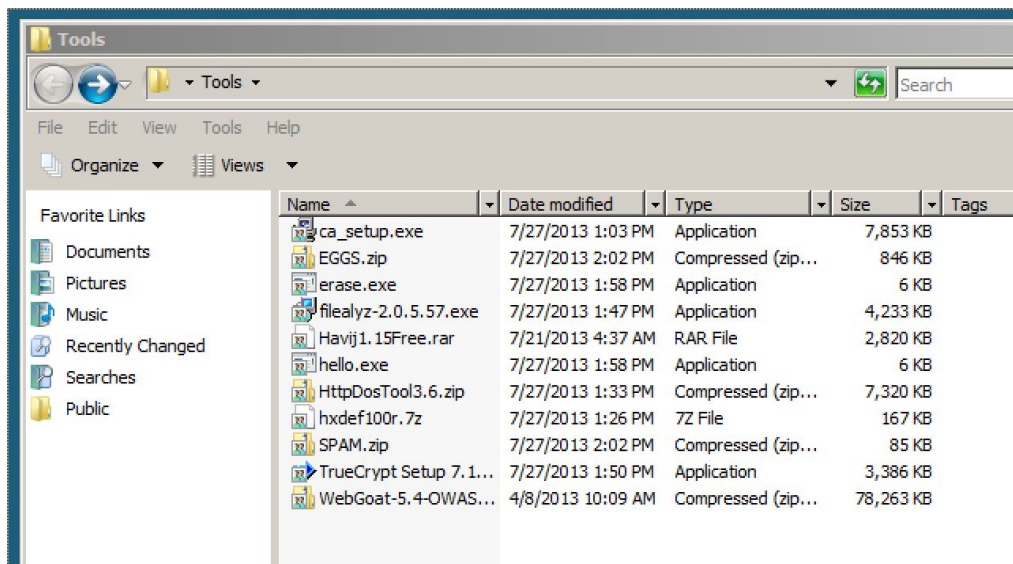


The target server also extensively used outdated software, many of which are riddled with their own CVE's. A glaring example of this is the use of Windows Server 2008 OS, which ceased receiving service pack support many years ago. Moreover, many ports were open on the server that were completely unnecessary, which also comes with many CVE's. The most recent example of open ports wreaking havoc is the Dyn DDoS attack in October 2016, which took advantage of open ports on legacy routers and IoT devices to grow a botnet large enough to take down one of the world's largest DNS providers.

Questions:

- *What is the meaning behind all of the “egg” files?*

Upon getting access to the VM, we decided to poke around and found over 1100 compressed .txt files that only contained the word “EGGS” over and over again. We would love to know why they were there.



Policy Implications:

Cyber hygiene is defined as the protection and maintenance of systems and devices that are connected to the web and the implementation of cyber security best practices.¹ While maintaining good cyber hygiene is easy, many people are unaware of the costs associated with cybercrime or don't care enough to take the time to learn. A 2016 report from Symantec showed that poor cybersecurity "resulted in 689 million people in 17 separate countries being victimized at a cost to them of \$128 billion."² On a geopolitical scale, politicians who don't take the steps to secure their devices or who lack a basic understanding of security, set themselves and therefore their constituents up for harm. Relating to this CTF, John Podesta revived a phishing scam from Russian attackers in March of 2016, asking him to change his password. Despite being suspicious, he changed his password anyway, giving the attackers access to 60,000 emails. This mistake arguably affected the outcome of a presidential election, which further shows how poor cybersecurity can have extreme ramifications. It has also been reported that Russia gained access because Podesta's password was "p@ssw0rd." While this claim has not been fully substantiated, it is important to note because weak passwords are a simple vulnerability that is easily exploitable. Therefore, the first steps for good security for lower risk targets should include stronger password requirements.

While trying to exploit the server we found that simple exploits such as social engineering password guesses, checking developer tools for sloppy code and scanning with small programs like nmap yielded much more than large scale exploit suites such as metasploit, especially when offset by the amount of effort put in. This is consistent with Praetorian's top five attack vectors³ in that the most exploited and most dangerous vulnerabilities are simple flaws such as weak passwords, and cleartext passwords. Therefore, implementing even a baseline level of security could pay dividends in the protection it provides.

One flaw we indicated with the server was that there were too many ports open. Only port 3000 should have been open because it was that was the port that was hosting the website. Additional ports leave entry points for attackers. In addition to weak passwords, this was a vulnerability that allowed for the 2016 Dyn DDoS attack by a Mirai botnet. The Mirai malware scans the internet for IPs of internet of things devices, which typically have a specific open port and default login credentials. We recommend

¹ <http://resources.infosecinstitute.com/the-importance-of-cyber-hygiene-in-cyberspace/#gref>

² <https://www.scmagazine.com/consumers-poor-cyber-hygiene-costs-them-billions/article/573834/>

³

<https://www.praetorian.com/downloads/report/How%20to%20Dramatically%20Improve%20Corporate%20IT%20Security%20Without%20Spending%20Millions%20-%20Praetorian.pdf>

servers minimize the number of open ports in order to protect themselves from such scanning and malware.

Conclusion

This simulation provided invaluable insight into what it looks like to hack into computer systems. By placing ourselves in the shoes of a hacker and setting out with the sole intention of breaking into a targeted computer network, our group witnessed the various of ways that vulnerabilities in computer networks can be exploited, as well as some of the technical limitations. This served to be useful for both our technical and non-technical team members, as we were able to witness the team start with virtually nothing and conclude with direct access to the network's main server within an hour.

Another important lesson from this simulation was the overall vulnerability of computer networks, especially regarding legacy systems, and the many ways in which they can be exploited. While cybersecurity, cybercrime and hacking have all been connotated as long, complex processes that require thousands of lines of code and hours of key-crunching, our group was able to exploit the system's vulnerabilities fairly quickly and with the use of minimal software and code (see: Tools and Methods). During the simulation, we also saw other groups attempt to break into the system using their own unique methods that we had not tried, illustrating the myriad of ways that vulnerabilities in a network's security can be exploited. The multitudes of vulnerabilities were exacerbated by the target's poor cyber hygiene: the two biggest flaws in this simulation that ultimately allowed us to compromise their system were one, having an extremely simple password, and two, using an outdated version of Windows to host their server (see our discussion on Policy Implications). We found that it was not the complex strategies that got us access to the system but instead exploiting the small, common vulnerabilities.

The write-up portion of the exercise has also been extremely valuable for understanding how to translate the digital experience into an understandable story. As Ming discussed, the most important part of being a 'hacker' that aims to expose computer vulnerabilities such as these is being able to succinctly explain to the average person how you did it. This was a great way to understand firsthand the disconnect that Professors Taliaferro and Chow frequently mention in class, in which cyber experts and policy makers fail to fundamentally understand how they relate to and can complement one another.

Q: If you had to do this challenge again, what would you do differently?

A: We would have focused more on grabbing the low-hanging fruit first. (Trying the most simple vulnerabilities and exploits first)