**PWN3: MAGIKARP**

**Dillon Wu**

Executive Summary:

The goal of the penetration testing was to identify all security flaws in the designated scope, to exploit these flaws in such a way that they would grant access to the targeted machines, and to find the hash values of the proof.txt files. The engagement was carried out with approval from the PWN Challenge #3 partner.

The main attack vector was a phishing email sent to an employee. After the executable file on the phishing email was opened, I utilized unpatched holes in the operating system to attain administrative privileges and find the passwords to other machines in the domain. All exploits were made using publicly available software. I recommend that the organization institute company-wide training sessions on the dangers of opening links from miscellaneous third parties.

The impact of these exploits are significant. I was able to gain access to the company’s proprietary software and trade secrets. Disclosure of this intellectual property to a third party would result in substantial financial loss for the company.

Detailed Findings:

\*\*Severity levels are determined according to two primary factors: (1) Impact of security flaw (2) Cost of upgrading

Vulnerability Name: Phishing Link

**Description:** I crafted an email with a malicious payload and sent it out to various individuals in the company. An individual using the email [olivia@pwn3.local](mailto:olivia@pwn3.local) happened to open the executable and I was able to get access to machine 10.20.160.86 as a result.

Severity: 10/10; Employees should not open miscellaneous links.

Affected Hostname: 10.20.160.86

Recommended Mitigations: I recommend that the company implement security training procedures for individuals at the company to make them aware of the dangers of opening a link coming from an unidentified source. Another policy would be to create a deterrence system whereby individuals who open random links and have introduced malware into the host computers are liable for a portion of the damages.

A close up of a logo

Description automatically generated

Vulnerability Name: Firewall Evasion

**Description:** I used Veil-Evasion to create payloads written in different software languages to bypass firewall detection. Veil-Evasion was unable to detect the executable written in Powershell.

Severity: 2/10; The firewall did its job in defending against some of the payloads, but was unable to detect all variations.

Recommended Mitigations: The firewall the company has installed may not be stringent enough to meet industry standards. Disregarding the inability to detect all of Veil’s packages, modern firewalls generally detect when port 4444 is trying to be reached, but this firewall was unable to do so. I recommend that the company update its firewall system, or find a new provider.

A screenshot of a cell phone screen with text

Description automatically generated

Vulnerability Name: Unpatched Software

**Description:** Admin privileges were obtained on machine 10.20.160.112 by exploiting the MS16-014 Kerberos Security Feature Bypass. The more technical details of the exploit can be found on Exploit-DB and Microsoft’s website, but this exploit basically granted me access to the admin account in the system without me having knowledge of the password. Using this module, I was able to find the proof file in the admin account

Severity: 10/10; unpatched software should be patched.

Affected Hostname: 10.20.160.112

Recommended Mitigations: I recommend that the company always patch their systems with the latest versions of Windows. In this instance, the problem could have been avoided with the latest patch from September 2, 2016 via Windows Update.

A screenshot of a newspaper

Description automatically generated

Attack Path

First, I used an nmap scan to identify the open ports on the machines.

Machine 10.20.160.10 contained the following open ports:

A close up of a sign

Description automatically generated

Machine 10.20.160.86 contained the following open ports:



I created the executable using Veil, and sent it to [olivia@pwn3.local](mailto:olivia@pwn3.local) via swaks.

A screenshot of a cell phone screen with text

Description automatically generated

A close up of a logo

Description automatically generated

While the email was being opened, I set up a listener in the background to catch the meterpreter.

A screenshot of a cell phone

Description automatically generated

A close up of a sign

Description automatically generated

I then located the local file on Olivia’s desktop.



I saw that I was using an x86 meterpreter on an x64 Windows machine so I converted the meterpreter to x64 as well.

A screenshot of a cell phone

Description automatically generated

A screenshot of a cell phone

Description automatically generated

In order to find the missing patches on the machine, I used metasploit’s local exploit suggester.

A screenshot of a newspaper

Description automatically generated

I used ms16\_014\_wmi\_recv\_notif to obtain admin privileges, and subsequently, the proof file.

A screenshot of a newspaper

Description automatically generated

A screenshot of a cell phone

Description automatically generated

I then used metasploit’s post exploit tool to identify the password on machine 10.20.160.10.

A screenshot of a cell phone

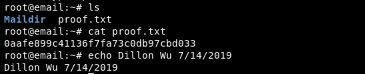
Description automatically generated

I used the decrypted value of the default password to log into the machine via ssh.

A screenshot of a cell phone

Description automatically generated

Finally, I navigated to the proof file.



Technical Details

Hostname: 10.20.160.86

Open Ports: 3389

Vulnerability Description: Unpatched software, Phishing Link, Firewall Evasion

Local file: f4d586bcb8603be92b3371010fea00c

Proof file: 4ccd6782ba1a890c74ed9f7f409e752b

Hostname: 10.20.160.10

Open Ports: 22, 25, 110, 143

Vulnerability Description: Password Management Protocols

Proof file: 0aafe899c41136f7fa73c0db97cbd033

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*ONE MORE PAGE\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

I guess you could say the phishing email was… a hook, line, and sinker.



