Taylor Rosby

CST – 221

Security

December 15, 2019

**Buffer Overflow**

A buffer overflow is an anomaly where a program overruns the buffer’s boundary and overwrites adjacent memory locations white writing data to a buffer. Buffer overflows can be triggered by malformed inputs. A malformed input is when all inputs are assumed to be smaller than and the buffer is created to be that size. An anomalous transaction that produces more data could case the buffer to be overwritten.

If the buffer overwrites adjacent or executable code, it could cause erratic program behavior like memory access errors, incorrect results, and crashes. This can easily occur in the C language because of the lack of built in protection against accessing or overwriting data in any part of memory and it does not check that data is written to an array within its boundaries. The C language is notorious for being subject to buffer overflow because of this reason.

To prevent buffer overflows additional code needs to be added to check for boundaries which will increase the processing time. Modern operating systems randomize the layout of memory or leave space between buffers and look for actions that write into the canaries. Canaries are used as an early warning system and crashes the program rather than continuing and overwriting unwanted data.

A screenshot of a cell phone

Description automatically generated

**Zero Day Exploit**

A zero-day exploit is an exploitable bug that are not known until after they have been used. It is a cyber attack that occurs the same day a weakness is discovered in software. These are considered rare, and reliable exploitable zero days are rarer because of the effectiveness of mitigation measures. Zero-day refers to a newly discovered software vulnerability. This means that an official patch or update hasn’t been released because the developers have just found out about the vulnerability. Until the vulnerability is fixed, hackers can exploit it to adversely affect computers, programs, data, or a network.

Ethical issues occur with zero-day exploits because they are almost always taken advantage of by cyber criminals. Knowing about zero day exploits can benefit people by encouraging programmers to do more testing and to do different and more abstract types of tests to make sure their sensitive information is as secured as possible.

**Kali Linux**

Kali Linux is a Linux distribution aimed at advanced penetration testing and security auditing. It contains several hundred tools geared towards various information security tasks information security tasks including penetration testing, security research, computer forensics, and reverse engineering. It is an open source project that is maintained and funded by Offensive Security.

There is a Kali Linux Certified Professional (KLCP) that testifies a person’s knowledge and fluency in using the Kali Linux penetration testing distribution. This certification also provides foundational knowledge for any information security professional allowing people to use it as a solid base for their career. To prepare for this certification test the person must familiarize themselves with the Kali Documentation site and the Kali Linux Revealed Book.

The tools that come with Kalu Linux could be used for illegal and malicious purposes instead for the purpose of education considering a good amount of the tools are used to find exploits, crack passwords, and sniff data packets. Knowing that Kali can be used for malicious intent could benefit people by encouraging people to use more secure networks, passwords etc. and to remain vigilant to what they do on public networks.

|  |  |  |
| --- | --- | --- |
| Tools | Description | How Its Used |
| Aircrack-ng | A network software suite consisting of a detector, packet sniffer, WEP and WPA/WPA2 cracker and analysis tool | Used for password cracking |
| Armitage | A graphical cyber attack management tool for the Metasploit Project that visualizes and recommends exploits | Used to visualize targets, recommends exploits, and exposes the advanced post-exploitation features in the Metasploit framework |
| Burp suite | A graphical tool for testing web applications security | Used to perform security testing of web applications |
| Cisco Global Exploiter | A hacking tool | Used to find and exploit vulnerabilities in Cisco Network Systems |
| Ettercap | A free and open source network security tool for man-in-the-middle attacks on LAN | Used for computer network protocol analysis and security auditing |
| John the Ripper | A free password cracking software tool | Used to detect weak UNIX passwords |
| Kismet | A network detector, packet sniffer, and intrusion detection system | Used to identify networks and decloak hidden networks |
| Maltego | An interactive data mining tool that renders directed graphs for link analysis | Used to find relationships between pieces of information from various sources located on the Internet |
| Metasploit | A computer security projects that provides info about security vulnerabilities and aids in penetration testing and IDS signature development | Used to create security testing tools and exploit modules and as a penetration testing system |
| Nmap | A free and open source network scanner | Used to discover hosts and services on a computer network by sending packets and analyzing the responses |
| OWASP ZAP | An open source web application security scanner | Used during penetration tests to assess web applications for vulnerabilities |
| Social engineering tools | The psychological manipulation of people into divulging confidential information | Used to trick users into making security mistakes or giving away sensitive information |
| Sqlmap | A penetration testing tool that automates the process of detecting and exploiting SQL injection | Used to detect and exploit database vulnerabilities and provides options for injections malicious code |
| Wireshark | A free and open source packet analyzer | Used for network troubleshooting, analysis, and software and communications protocol development |
| Hydra | A password detection tool | Used in brute force cracking |
| Reverse engineering tools | Process of deconstructing an object to reveal its designs, architecture, or to extract knowledge | Used to detect and neutralize viruses and malware and to protect intellectual property |
| Binwalk | A tool for searching a given binary image for embedded files and executable code | Used to identify files and code embedded inside of firmware images |
| Foremost | A forensic data recovery program | Used to recover lost files based on header, footer and internal data structures |
| Volatility | An open source memory forensics framework for incident response and malware analysis | Used to extract digital artifacts from RAM samples |

**Password Strength Bash Script**

#!/bin/bash

echo "Enter a password : "

read password

i=$(echo ${#password})

if [ $i -lt 8 ]; then

echo "$password is smaller than 8 characters"

exit 1

else

if ! [ -z `echo $password & $password =~ "[[:punct:]]" & $password =~ "[[:digit:]]"` ]; then

echo "$password is a weak password"

else

echo "$password is a strong password"

fi

fi

**Manage Users Bash Script**

#!/bin/bash

echo -n "Enter a filename : "

read filename

filename=$1

echo -n "Enter a group : "

read group

group=$2

echo -n "Enter a flag : "

read flag

flag=$3

if grep $2 /etc/group

then

echo "$2 already exists"

else

groupadd $2

fi

if [[ "$3" == "-a" ]]; then

$1

while read -r username password

do

uname="$username"

pword ="$password"

echo "Username: $uname"

echo "Password: $pword

if [ $(id -u) -eq 0 ]; then

egrep -i "^$uname" /etc/passwd

if [ $? -eq 0 ]; then

echo "$uname already has an account"

exit 1

else

useradd -m -p $ pword $uname

[ $? -eq 0 ] && echo "User has been added to system" || echo "Failed to add user"

fi

else

echo "root access required"

exit 2

fi

usermod -a -G $2 $uname

fi

if [[ "$3" == "-r" ]]; then

$1

while read -r username password

do

uname="$username"

pword ="$password"

userdel -r $uname

echo "$uname has been deleted."

if grep $2 /etc/group

then

groupdel $2

else

echo "Group does not exist."

fi

fi