A WILD KOBALOS APPEARS
Tricksy Linux malware goes after HPCs

Simone Landi

UniPi - ICT Risk Assessment

Table of contents

1. Introduction
2. Operation
3. Technical Analysis
4. OpenSSH Credential Stealer
5. MITRE ATT&CK Techniques
6. Conclusion

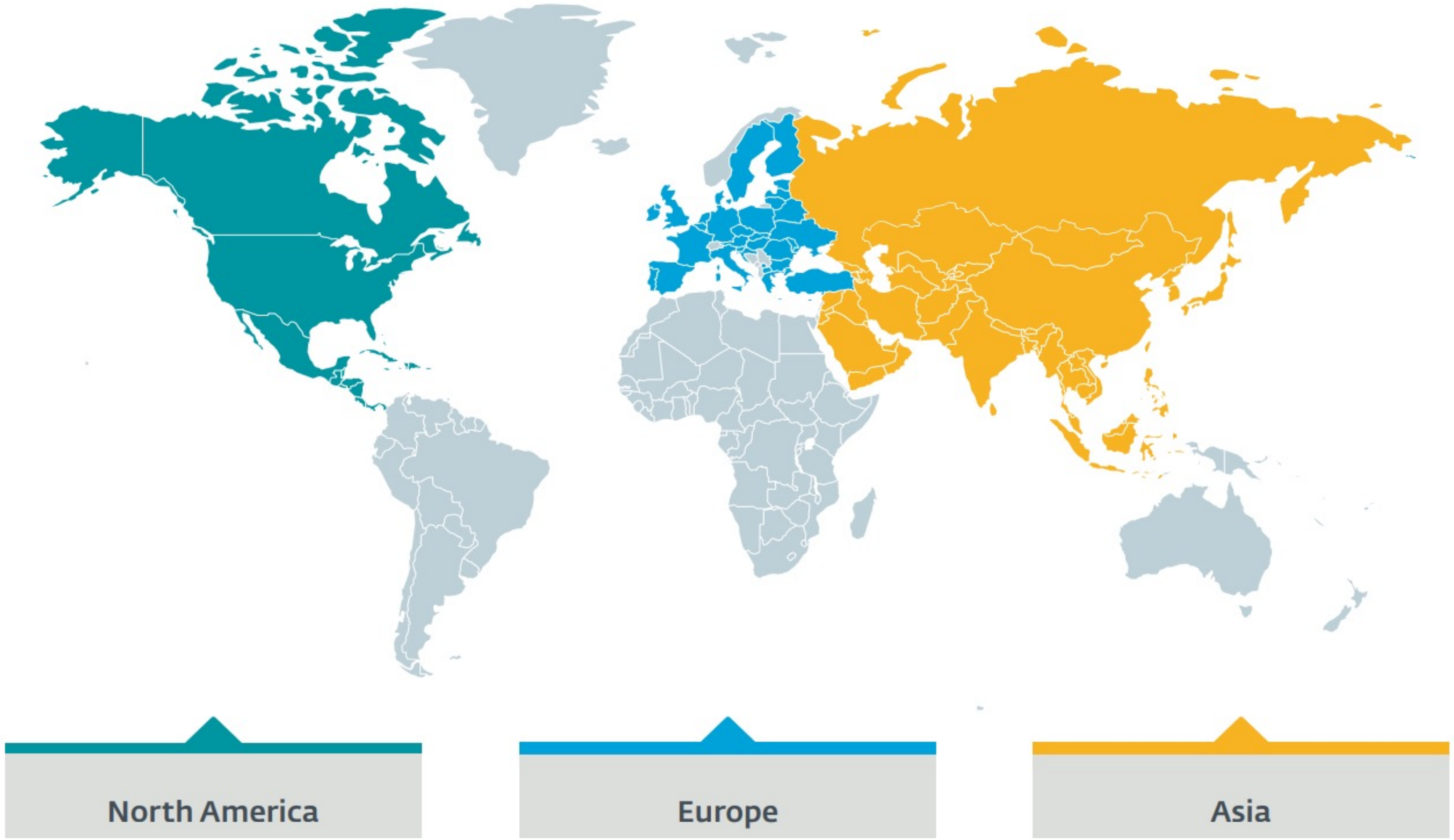
Introduction

- Kobalos is a multiplatform backdoor that works on Linux, FreeBSD and Solaris
- The targets of Kobalos are high profile and include high-performance computers, servers in academia, an endpoint security vendor, and a large internet service provider
- It was deployed in servers located in Europe, North America and Asia
- Kobalos uses a complex obfuscation mechanism that makes its analysis challenging
- Most of the hosts compromised by Kobalos also had an OpenSSH credential stealer installed
- Kobalos's operations seem not to be at a large scale
- The intent of the authors of this malware is still unknown

Operation

Victimology

- very limited number of victims
- Targets tend to be high-performance computers (HPC) and servers that are part of academic and research networks



OVERVIEW OF KOBALOS - ACCESS TO THE COMPROMISED SYSTEM

- The Kobalos malware contains generic commands to read from and write to the file system and spawn a terminal to execute arbitrary commands
- it doesn’t contain any specific payload that could indicate the intentions of its authors

OVERVIEW OF KOBALOS - REACHABILITY

- The Kobalos malware supports multiple ways to make itself reachable from the outside:
  1. By opening a TCP port and waiting for an incoming connection (passive backdoor)
  2. By connecting to another instance of Kobalos configured to run as C&C server
  3. By waiting for connections to an already running, legitimate service but coming from a specific TCP source port
- In all cases the sshd file was completely replaced, so the malware is persistent across service or system restarts

OVERVIEW OF KOBALOS - AUTHENTICATION AND NETWORK ENCRYPTION

- triggering the backdoor requires its clients to authenticate
- Clients must possess an RSA-512 private key and a password
- Once both are validated, Kobalos generates and encrypts two 16-byte keys with the RSA-512 public key and sends them to the attackers
- These two keys are used to RC4 encrypt subsequent inbound and outbound traffic

OVERVIEW OF KOBALOS - ALTERNATE PORT

- during the authentication phase, the operator can choose to continue the communication on another TCP connection
- Kobalos will start listening on the requested TCP port and the rest of the communication will use that connection
- Data going through this channel is encrypted using the RC4 keys

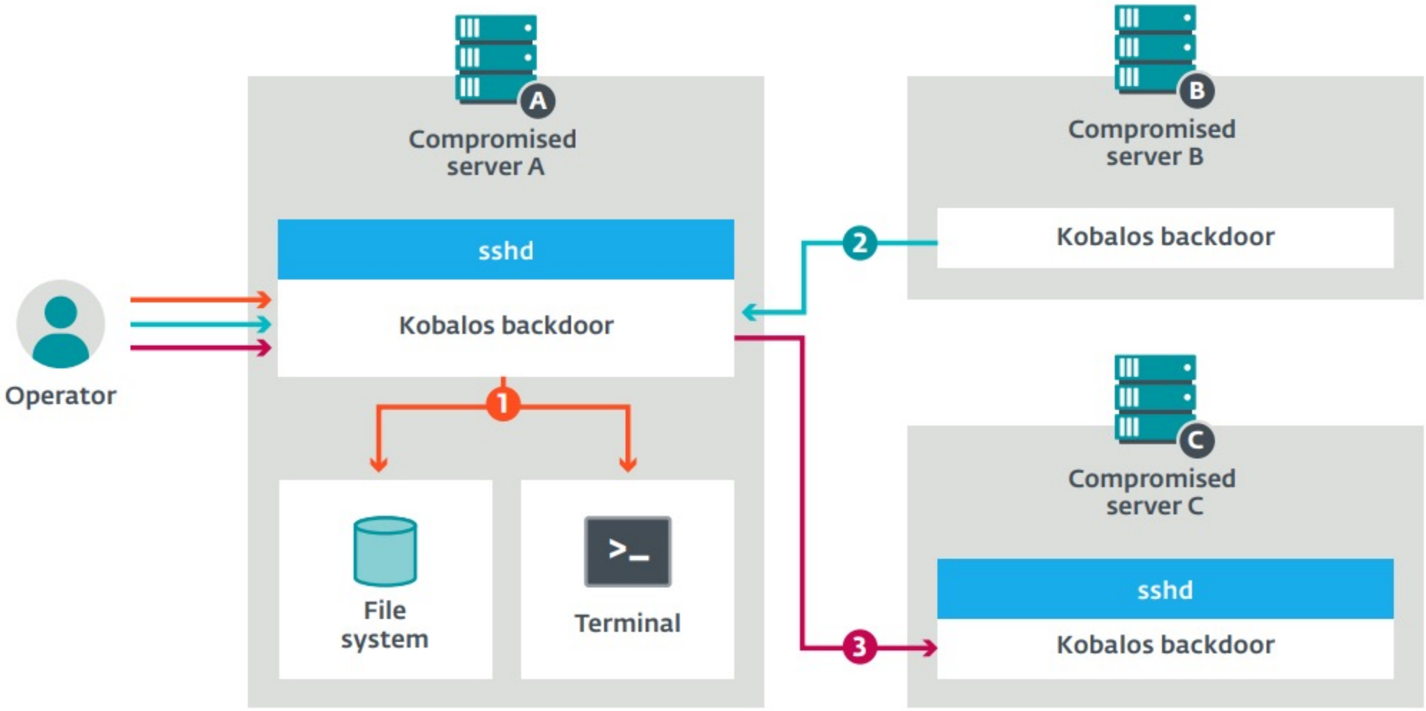
OVERVIEW OF KOBALOS - PROXYING TO OTHER COMPROMISED MACHINES

- Kobalos can also be used as a proxy to connect other servers compromised with Kobalos
- It also supports the alternate port option: a command can be sent to the proxy to “switch” the connection to a new TCP port
- Proxies can be chained, the operators can use multiple Kobalos-compromised machines to reach their targets

OVERVIEW OF KOBALOS - ALL TOGETHER

- Direct connection to the backdoor
  - the backdoor is running inside a compromised OpenSSH server process
  - Communication with the backdoor requires the right TCP source port from the operator
- Server B uses server A as C&C server
  - Once started, it will manage a list of connected bots giving the operator the ability to connect to any of them
  - Authentication is still required on the final node and end-to-end encryption is enforced using the exchanged RC4 keys
  - the Kobalos malware sample running on Server B needs to have in its configuration the IP address and port of the C&C server running on Server A
  - Server B will only see traffic to and from Server A, hiding the IP address of the operator
- Server A is used to proxy connection to Server C
  - authentication and end-to-end encryption is enforced
  - it can be used to connect to Kobalos instances that expect a specific source port when connecting

OVERVIEW OF KOBALOS - ALL TOGETHER



Scenario 1

Direct connection to the backdoor.

Scenario 2

Server B uses Server A as C&C server.

Scenario 3

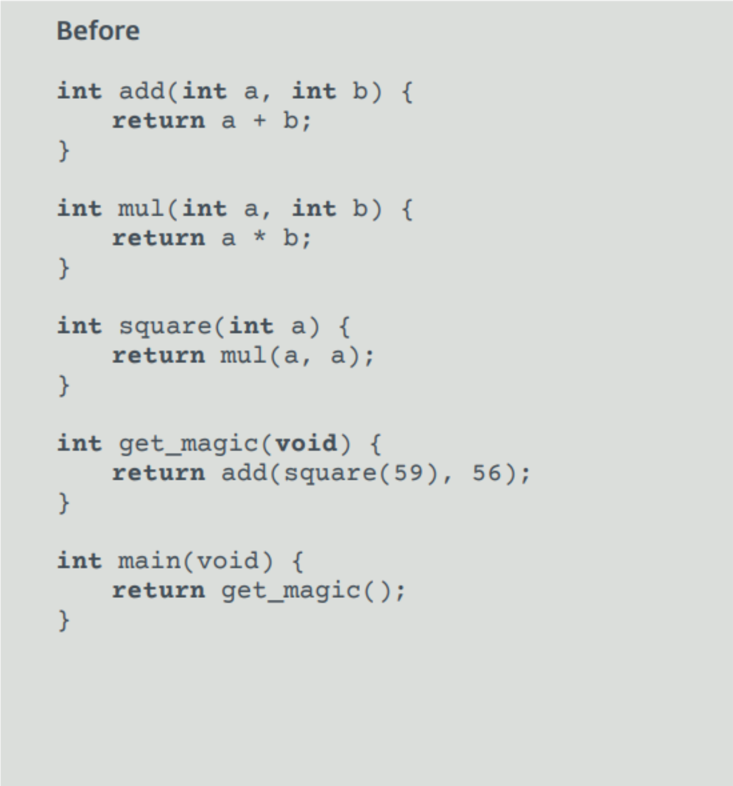
Server A is used to proxy connection to Server C.

Technical Analysis

- Obfuscation
- Configuration
- Deployment and Persistence
- Interacting with the backdoor
- Malware Operation

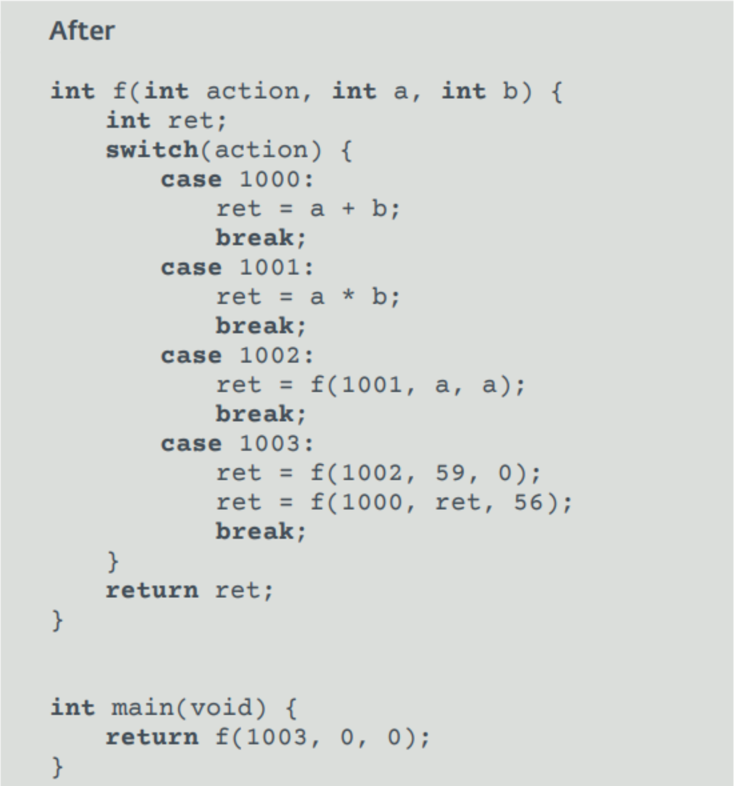
Obfuscation (I)

- Exceptional control flow flattening
  - Kobalos recursively calls the function to perform whatever subtask it needs to do
  - The first parameter to the function is the action to perform (They are 37)
  - the function also serves as a signal handler for SIGCHLD to let child process terminate gracefully and SIGALRM to handle connection timeout



Before

```
int add(int a, int b) {
    return a + b;
}

int mul(int a, int b) {
    return a * b;
}

int square(int a) {
    return mul(a, a);
}

int get_magic(void) {
    return add(square(59), 56);
}

int main(void) {
    return get_magic();
}
```



After

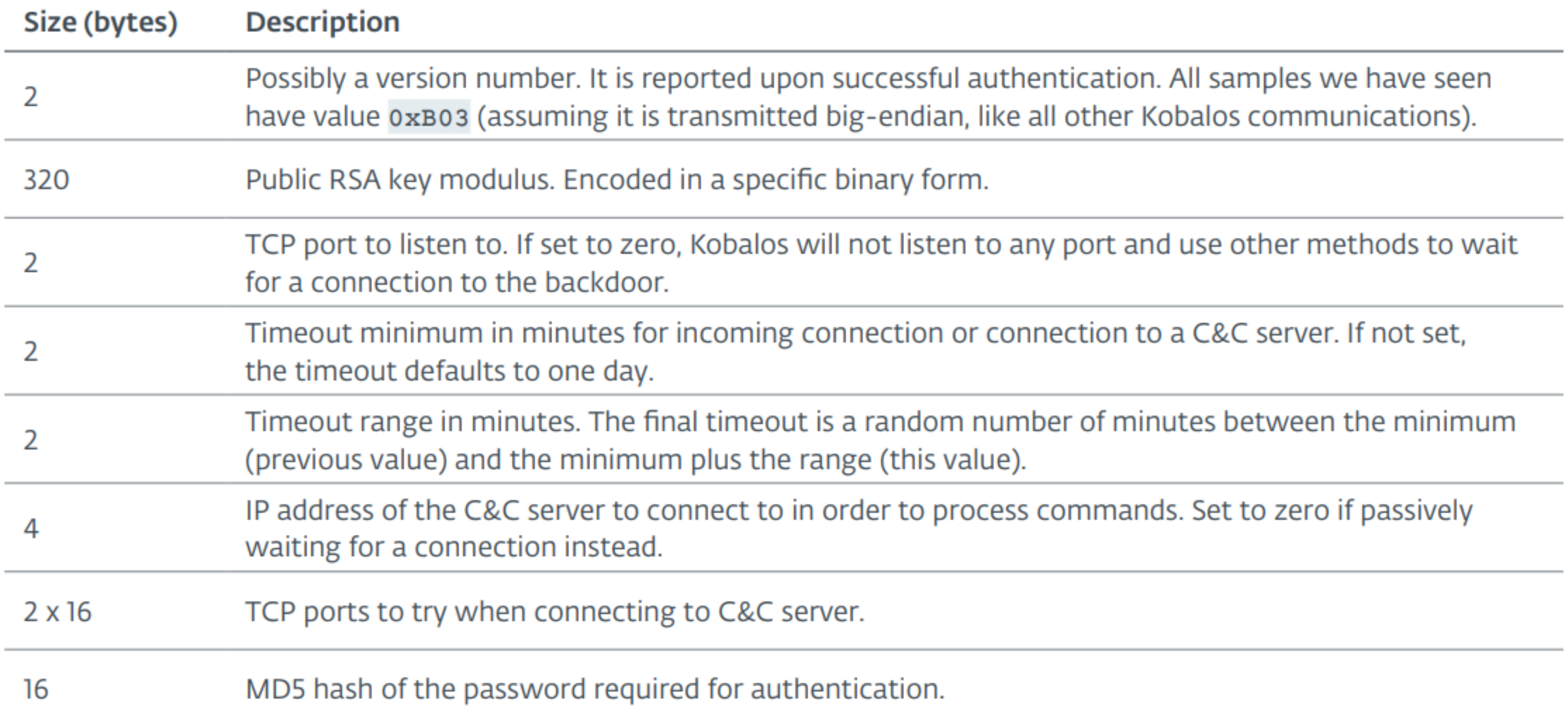
```
int f(int action, int a, int b) {
    int ret;
    switch(action) {
        case 1000:
            ret = a + b;
            break;
        case 1001:
            ret = a * b;
            break;
        case 1002:
            ret = f(1001, a, a);
            break;
        case 1003:
            ret = f(1002, 59, 0);
            ret = f(1000, ret, 56);
            break;
    }
    return ret;
}


int main(void) {
    return f(1003, 0, 0);
}
```

Obfuscation (II)

- Encrypted strings
  - Kobalos doesn’t have any readable plain text string in its code or its data
  - It only uses a few small strings, which are encrypted using RC4 (decrypted right after the initial communication but before the authentication)
- Anti-forensics techniques
  - Set RLIMIT_CORE to zero to prevent core-dump generation if the process crashes
  - Ignore most signals to make it more difficult to interrupt the process
- Timestomping
  - Timestamps of replaced files, such as ssh to add the credential stealer or sshd to deploy Kobalos, are tampered with to reduce suspicion

Configuration

- static configuration that enables or disables functions of the malware
- this configuration differs when Kobalos is running inside sshd or as a stand-alone executable, it requires either a remote C&C server address (remote_c2_addr) or a port to listen on (listen_port)



Size (bytes)	Description
2	Possibly a version number. It is reported upon successful authentication. All samples we have seen have value `0xB03` (assuming it is transmitted big-endian, like all other Kobalos communications).
320	Public RSA key modulus. Encoded in a specific binary form.
2	TCP port to listen to. If set to zero, Kobalos will not listen to any port and use other methods to wait for a connection to the backdoor.
2	Timeout minimum in minutes for incoming connection or connection to a C&C server. If not set, the timeout defaults to one day.
2	Timeout range in minutes. The final timeout is a random number of minutes between the minimum (previous value) and the minimum plus the range (this value).
4	IP address of the C&C server to connect to in order to process commands. Set to zero if passively waiting for a connection instead.
2 x 16	TCP ports to try when connecting to C&C server.
16	MD5 hash of the password required for authentication.

Deployment and Persistence

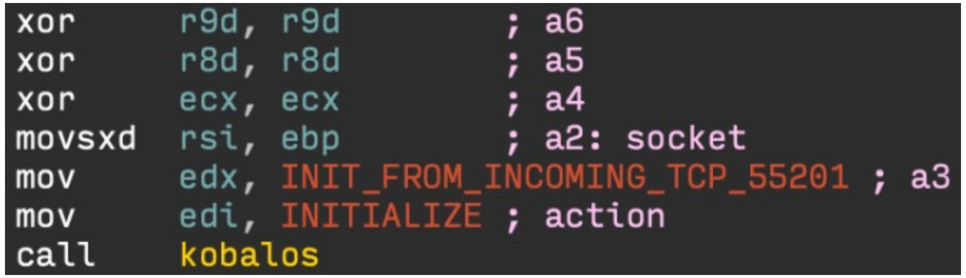
- When Kobalos is deployed to run as part of the OpenSSH server, the sshd file must be recompiled to include the malicious code
- It seems that the operators compile Kobalos using the proper OpenSSH source, the one that was already installed
- This is done in order to enable persistence while preventing mismatching version anomalies, such as library incompatibilities
- It's worth noting that replacing sshd requires root privileges. However, there exist stand-alone variants that either connect to a C&C server or listen on a TCP port

Interacting with the backdoor – Connecting to the backdoor (I)

- One of the notable features of Kobalos is the malleability of the ways in which the link between its operators and the compromised host can be established
  1. Listen on a given TCP port (passive mode)
  2. Connect to a C&C server (active mode) and wait for operator to connect through the C&C server
  3. Replace an existing service that listens on a TCP port and wait for connections from specific TCP source ports

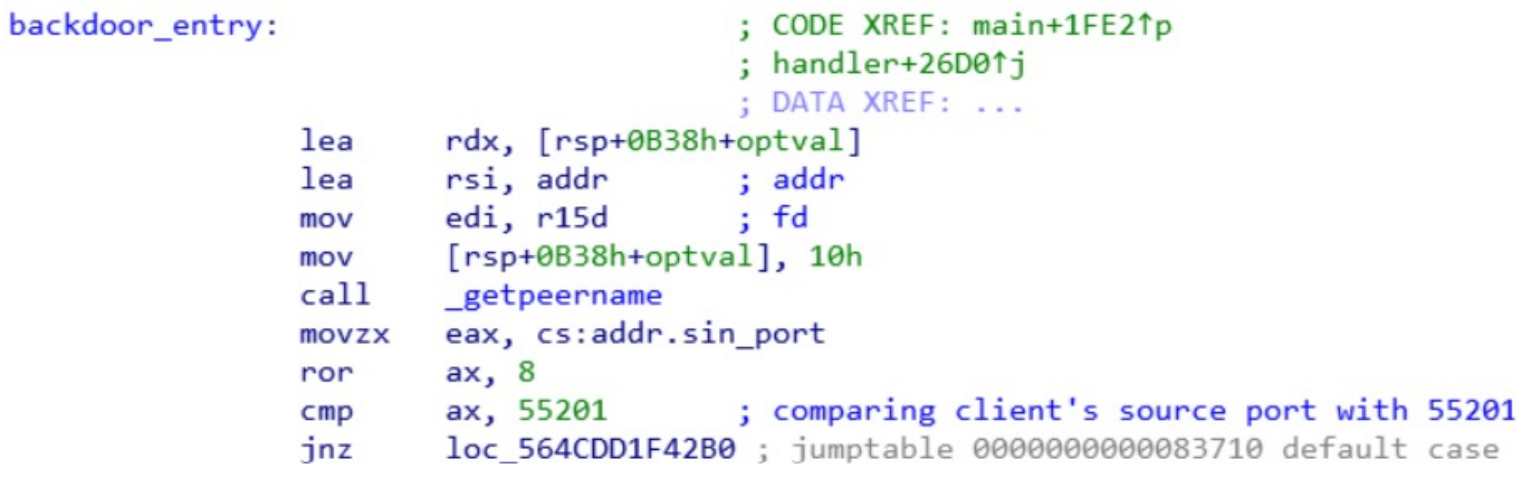
Interacting with the backdoor – Connecting to the backdoor (II)

- The trojanized version calls Kobalos's code each time a new TCP connection is accepted



```
xor     r9d, r9d         ; a6
xor     r8d, r8d         ; a5
xor     ecx, ecx         ; a4
movsxd  rsi, ebp         ; a2: socket
mov     edx, INIT_FROM_INCOMING_TCP_55201 ; a3
mov     edi, INITIALIZE ; action
call    kobalos
```

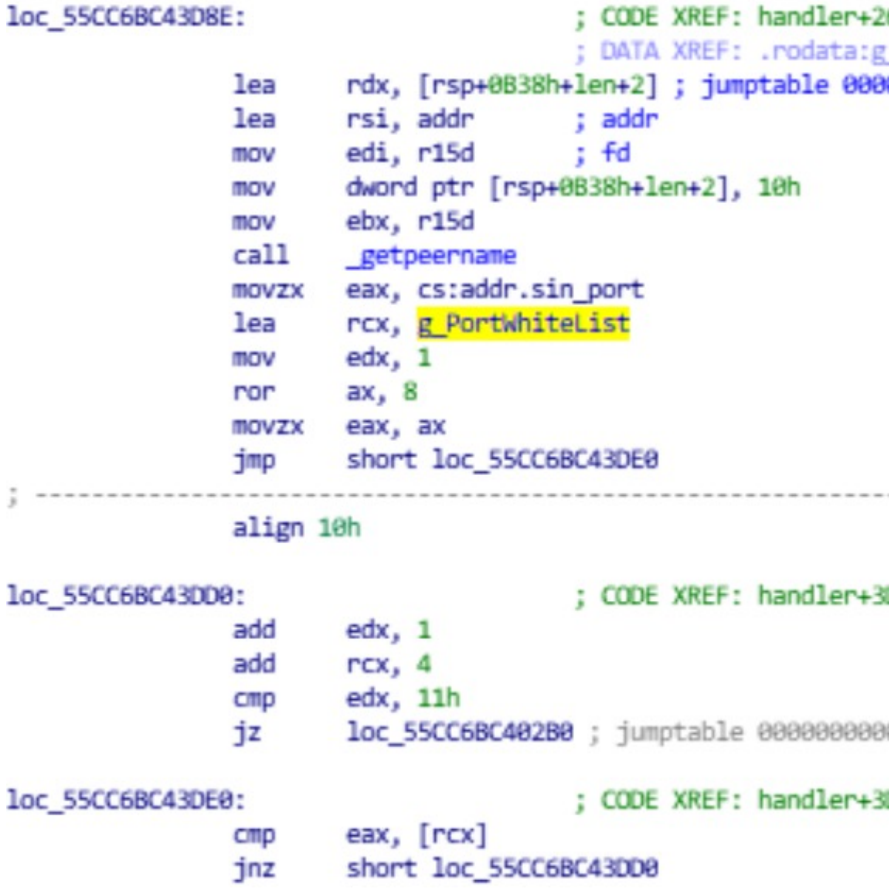
- the OpenSSH server being abused with this method, it's the most popular method



```
backdoor_entry:                          ; CODE XREF: main+1FE2↑p
                                         ; handler+26D0↑j
                                         ; DATA XREF: ...
                lea     rdx, [rsp+0B38h+optval]
                lea     rsi, addr       ; addr
                mov     edi, r15d       ; fd
                mov     [rsp+0B38h+optval], 10h
                call    _getpeername
                movzx   eax, cs:addr.sin_port
                ror     ax, 8
                cmp     ax, 55201       ; comparing client's source port with 55201
                jnz     loc_564CDD1F42B0 ; jumptable 0000000000083710 default case
```

Interacting with the backdoor – Connecting to the backdoor (III)

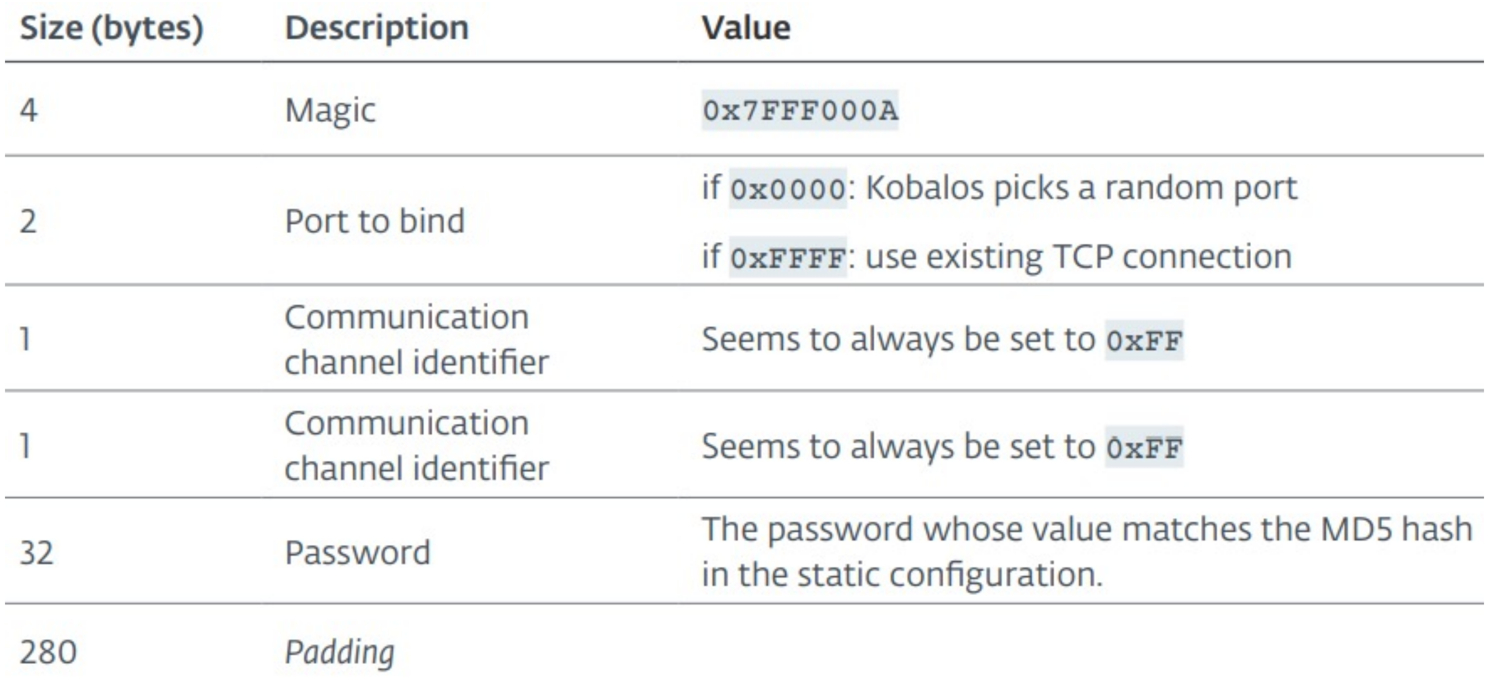
- There is also an additional method implemented for filtering incoming TCP connections in Kobalos, which instead compares the source port against a list of 16 ports



```
loc_55CC6BC43D8E:                       ; CODE XREF: handler+2
                                        ; DATA XREF: .rodata:g
                lea     rdx, [rsp+0B38h+len+2] ; jumptable 000
                lea     rsi, addr       ; addr
                mov     edi, r15d       ; fd
                mov     dword ptr [rsp+0B38h+len+2], 10h
                mov     ebx, r15d
                call    _getpeername
                movzx   eax, cs:addr.sin_port
                lea     rcx, g_PortWhiteList
                mov     edx, 1
                ror     ax, 8
                movzx   eax, ax
                jmp     short loc_55CC6BC43DE0
; ---------------------------------------------------------------
                align 10h

loc_55CC6BC43DD0:                       ; CODE XREF: handler+3
                add     edx, 1
                add     rcx, 4
                cmp     edx, 11h
                jz      loc_55CC6BC402B0 ; jumptable 000000000

loc_55CC6BC43DE0:                       ; CODE XREF: handler+3
                cmp     eax, [rcx]
                jnz     short loc_55CC6BC43DD0
```

Interacting with the backdoor – Authentication (I)

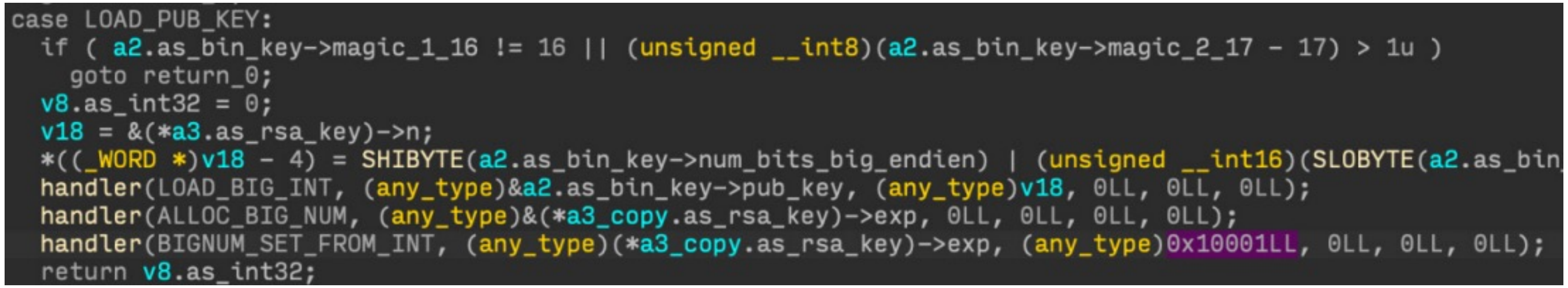
- A private RSA key and a 32-byte password are required
- An initial 320-byte packet is sent by the Kobalos client to the backdoored server



Size (bytes)	Description	Value
4	Magic	`0x7FFF000A`
2	Port to bind	if `0x0000`: Kobalos picks a random port if `0xFFFF`: use existing TCP connection
1	Communication channel identifier	Seems to always be set to `0xFF`
1	Communication channel identifier	Seems to always be set to `0xFF`
32	Password	The password whose value matches the MD5 hash in the static configuration.
280	*Padding*	

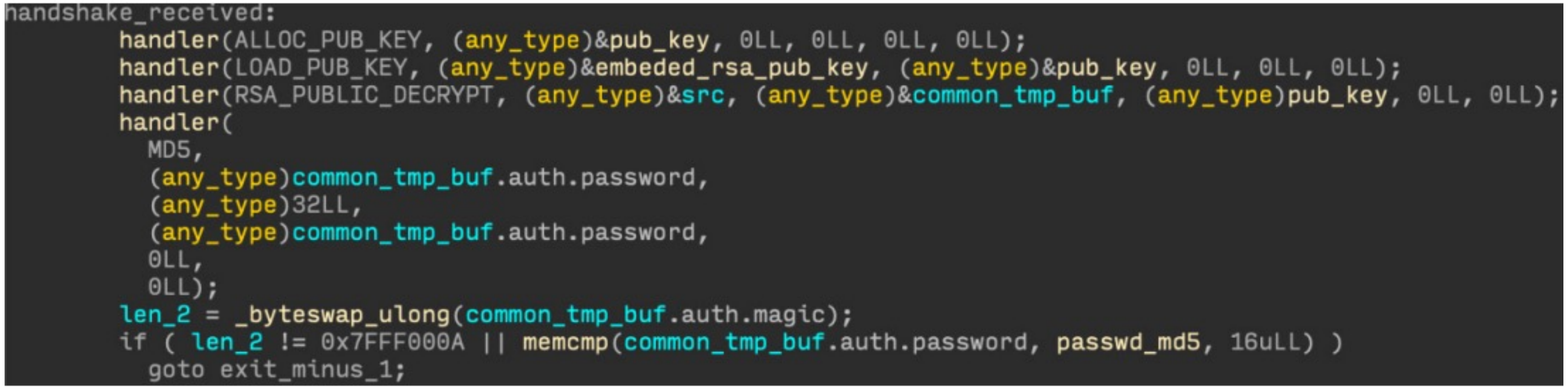
Interacting with the backdoor – Authentication (II)

- The first 64 bytes of the packet are decrypted using the RSA-512 public key modulus provided in the configuration and the 0x10001 exponent



```
case LOAD_PUB_KEY:
  if ( a2.as_bin_key->magic_1_16 != 16 || (unsigned __int8)(a2.as_bin_key->magic_2_17 - 17) > 1u )
    goto return_0;
  v8.as_int32 = 0;
  v18 = &(*a3.as_rsa_key)->n;
  *((_WORD *)v18 - 4) = SHIBYTE(a2.as_bin_key->num_bits_big_endien) | (unsigned __int16)(SLOBYTE(a2.as_bin
  handler(LOAD_BIG_INT, (any_type)&a2.as_bin_key->pub_key, (any_type)v18, 0LL, 0LL, 0LL);
  handler(ALLOC_BIG_NUM, (any_type)&(*a3_copy.as_rsa_key)->exp, 0LL, 0LL, 0LL, 0LL);
  handler(BIGNUM_SET_FROM_INT, (any_type)(*a3_copy.as_rsa_key)->exp, (any_type)0x10001LL, 0LL, 0LL, 0LL);
  return v8.as_int32;
```

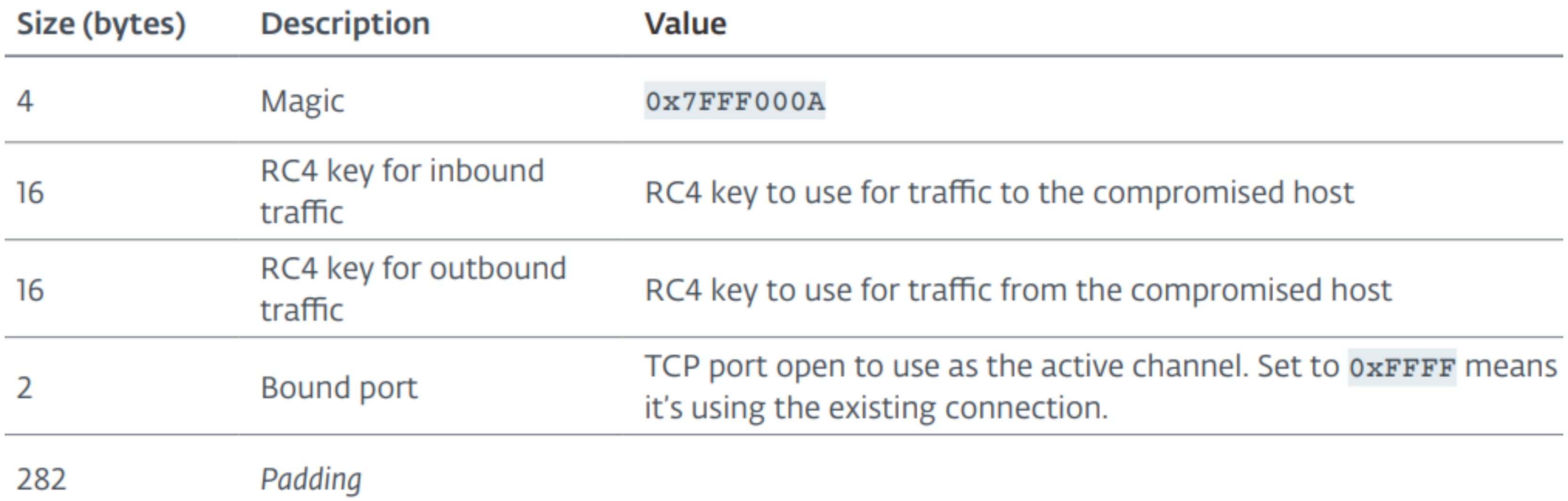
- the 32-byte password is MD5-hashed and compared to the digest found in the static configuration



```
handshake_received:
        handler(ALLOC_PUB_KEY, (any_type)&pub_key, 0LL, 0LL, 0LL, 0LL);
        handler(LOAD_PUB_KEY, (any_type)&embeded_rsa_pub_key, (any_type)&pub_key, 0LL, 0LL, 0LL);
        handler(RSA_PUBLIC_DECRYPT, (any_type)&src, (any_type)&common_tmp_buf, (any_type)pub_key, 0LL, 0LL);
        handler(
          MD5,
          (any_type)common_tmp_buf.auth.password,
          (any_type)32LL,
          (any_type)common_tmp_buf.auth.password,
          0LL,
          0LL);
        len_2 = _byteswap_ulong(common_tmp_buf.auth.magic);
        if ( len_2 != 0x7FFF000A || memcmp(common_tmp_buf.auth.password, passwd_md5, 16uLL) )
          goto exit_minus_1;
```

Interacting with the backdoor – Authentication (III)

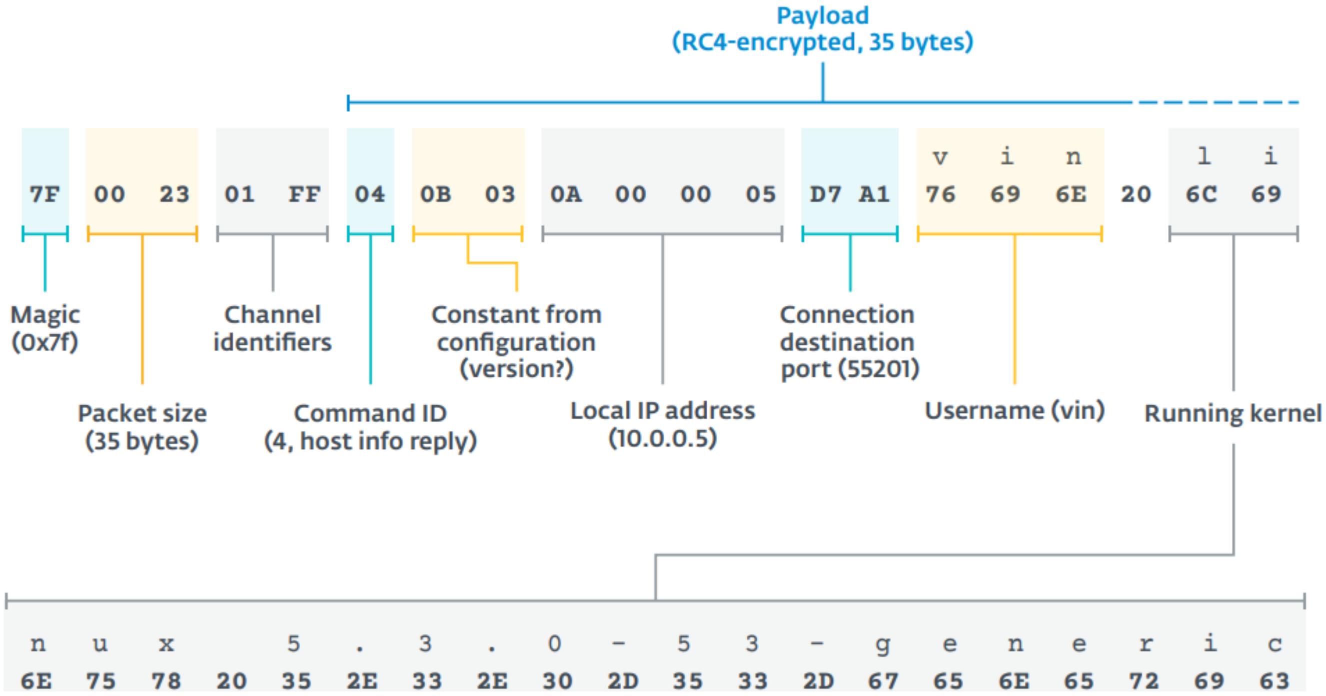
- Kobalos will use the public RSA key to encrypt a set of RC4 keys to use for the rest of the communication: one for inbound traffic and one for outbound traffic
- Those encrypted keys are sent back in the reply



Size (bytes)	Description	Value
4	Magic	`0x7FFF000A`
16	RC4 key for inbound traffic	RC4 key to use for traffic to the compromised host
16	RC4 key for outbound traffic	RC4 key to use for traffic from the compromised host
2	Bound port	TCP port open to use as the active channel. Set to `0xFFFF` means it's using the existing connection.
282	*Padding*	

Interacting with the backdoor – Active channel

- After authentication is accepted, the active channel may use a different port than the one used for authentication
- the client authenticating to Kobalos must provide a “port to bind” in its encrypted message
  - If this value is different from 0xFFFF, Kobalos will start listening to the given TCP port
  - If this value is zero, it will start listening to a random port above 1024
- the newly opened port number is included in the authentication reply alongside the pair of RC4 keys
- The Kobalos malware will be the first to send a packet to the connected operator



Payload
(RC4-encrypted, 35 bytes)

```
                                                          v  i  n     l  i
7F 00 23 01 FF 04 0B 03 0A 00 00 05 D7 A1 76 69 6E 20 6C 69
```

Magic (0x7f)

Packet size (35 bytes)

Channel identifiers

Command ID (4, host info reply)

Constant from configuration (version?)

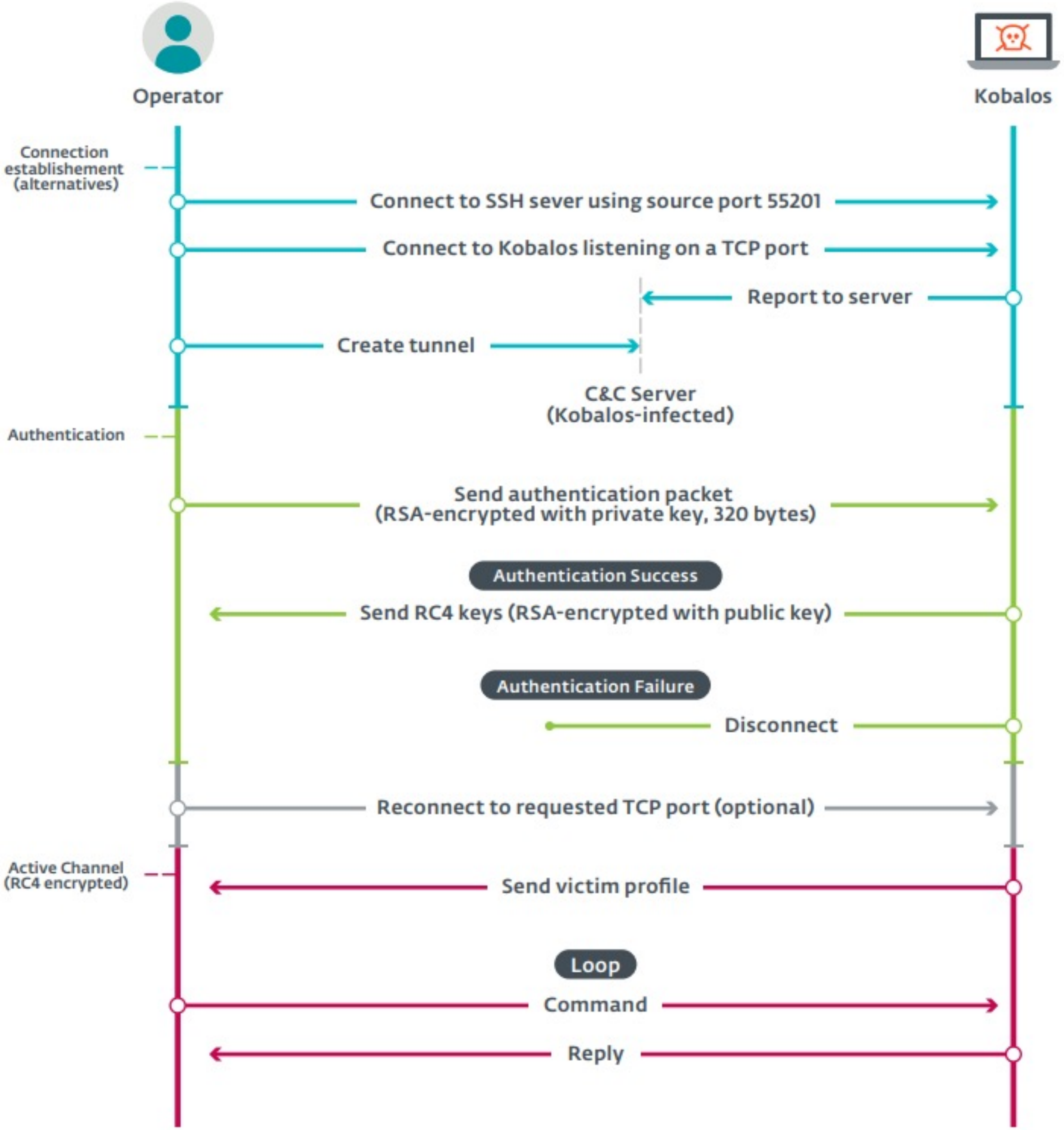
Local IP address (10.0.0.5)

Connection destination port (55201)

Username (vin)

Running kernel

```
n  u  x     5  .  3  .  0  -  5  3  -  g  e  n  e  r  i  c
6E 75 78 20 35 2E 33 2E 30 2D 35 33 2D 67 65 6E 65 72 69 63
```



Operator

Kobalos

Connection establishement (alternatives)

Connect to SSH sever using source port 55201

Connect to Kobalos listening on a TCP port

Report to server

Create tunnel

C&C Server (Kobalos-infected)

Authentication

Send authentication packet (RSA-encrypted with private key, 320 bytes)

Authentication Success

Send RC4 keys (RSA-encrypted with public key)

Authentication Failure

Disconnect

Reconnect to requested TCP port (optional)

Active Channel (RC4 encrypted)

Send victim profile

Loop

Command

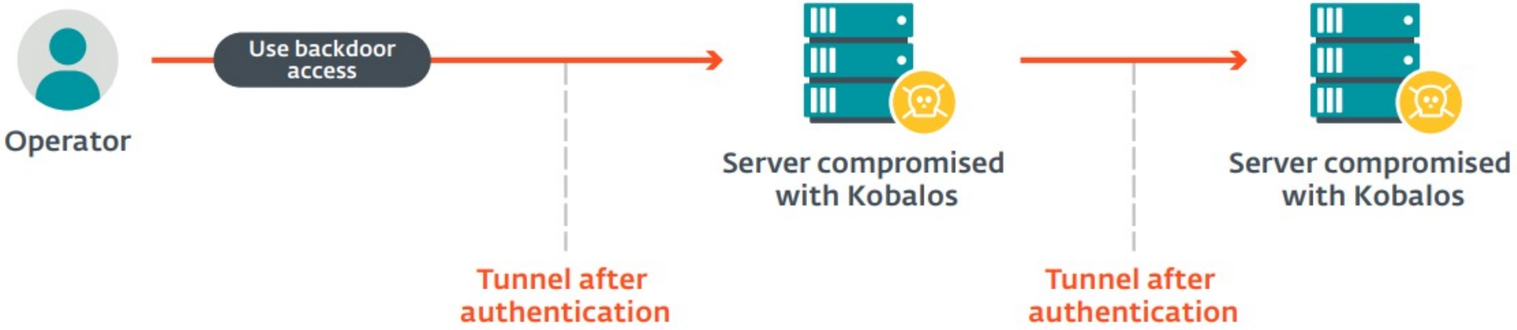
Reply

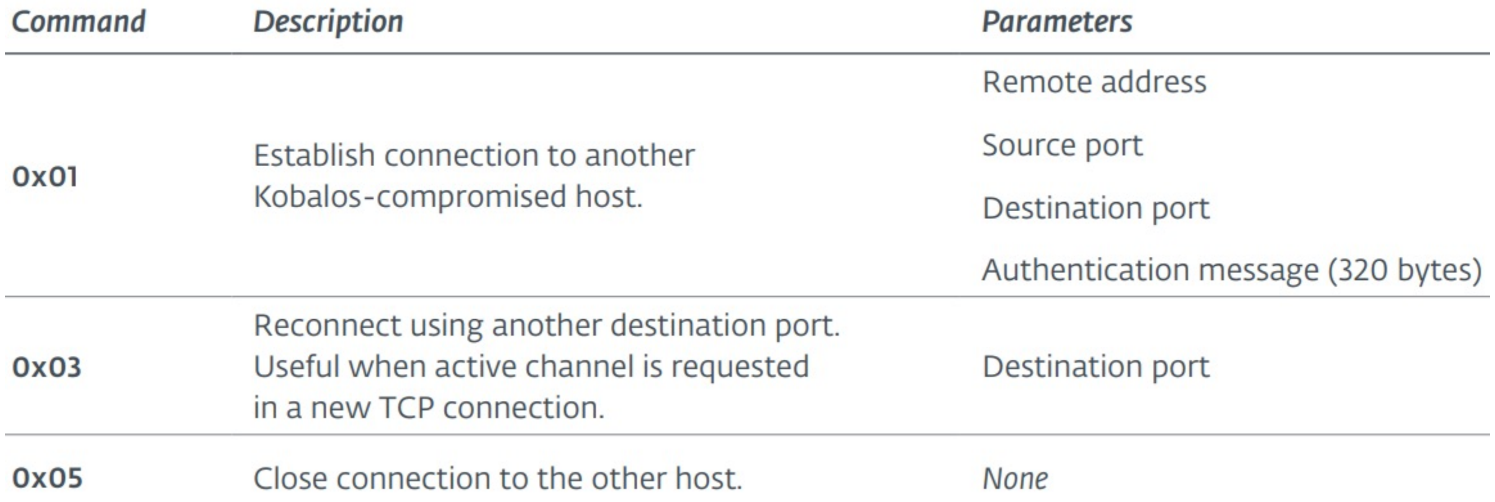
Malware Operation

- Once authenticated, an operator can issue various commands to the backdoor
- We can split them into the following categories:
  1. Connect to other Kobalos-compromised servers and act as a proxy
  2. Read and write any files on the file system
  3. Launch and access a pseudo-terminal on the compromised host
  4. Run and manage Kobalos C&C servers and access connected bots
- Commands are encapsulated in the active channel and start with a single byte to identify the command, followed by parameters parsed by that command
- Replies from Kobalos have a similar format, with their first byte being an identifier, and are handled by that custom client

Malware Operation – Use as a proxy

- It’s not a simple TCP proxy, it expects specific packet sizes for authentication and encapsulation specific to Kobalos
- When connecting to a third-party machine, the operator can choose the TCP source port
- The end node would only see the IP address of another compromised machine, and not the IP address of the operator

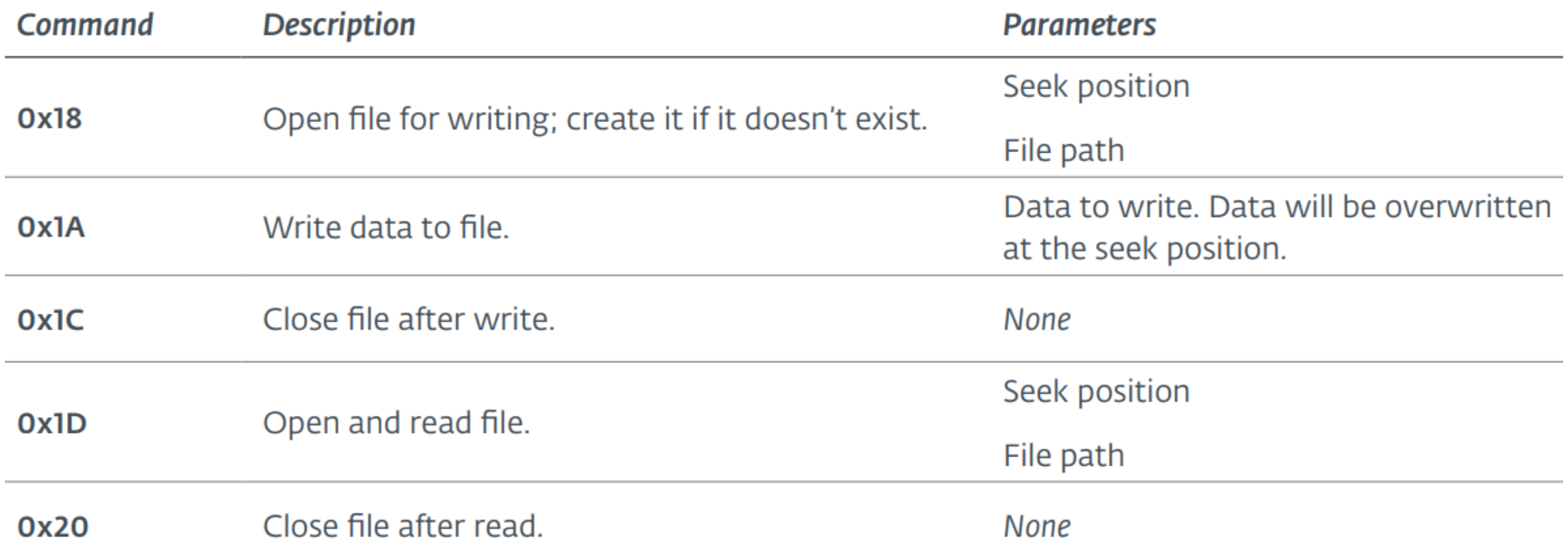

Operator
Use backdoor access
Server compromised with Kobalos
Tunnel after authentication
Server compromised with Kobalos
Tunnel after authentication



Command	*Description*	*Parameters*
0x01	Establish connection to another Kobalos-compromised host.	Remote address Source port Destination port Authentication message (320 bytes)
0x03	Reconnect using another destination port. Useful when active channel is requested in a new TCP connection.	Destination port
0x05	Close connection to the other host.	*None*

Malware Operation – File system access

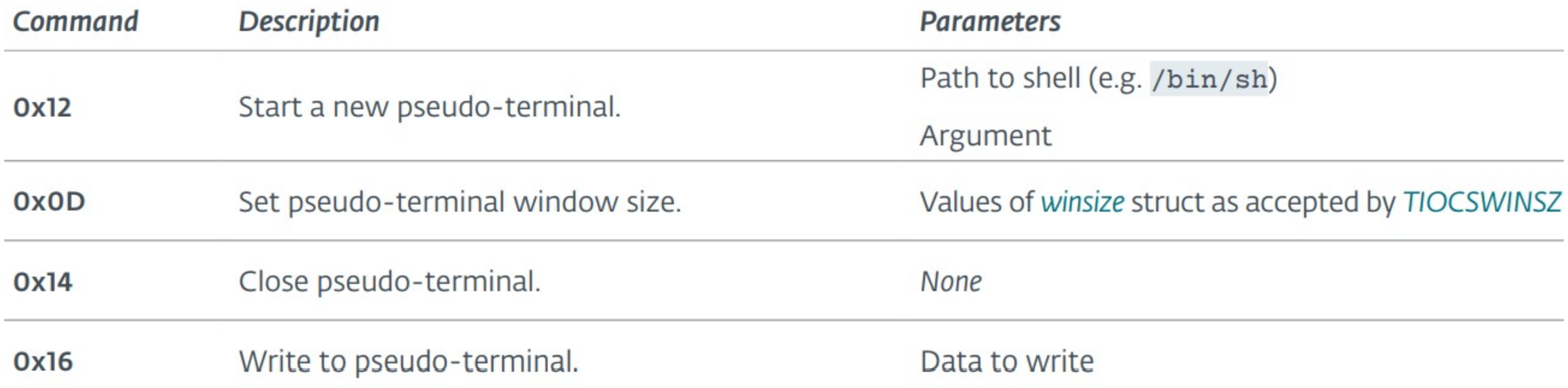
- Once authenticated, an operator can read or write any file on the system
- operator can only send a 64 kB packet
- only 1000 bytes of data are read and sent at a time



Command	*Description*	*Parameters*
0x18	Open file for writing; create it if it doesn't exist.	Seek position File path
0x1A	Write data to file.	Data to write. Data will be overwritten at the seek position.
0x1C	Close file after write.	*None*
0x1D	Open and read file.	Seek position File path
0x20	Close file after read.	*None*

Malware Operation – Pseudo-terminal creation

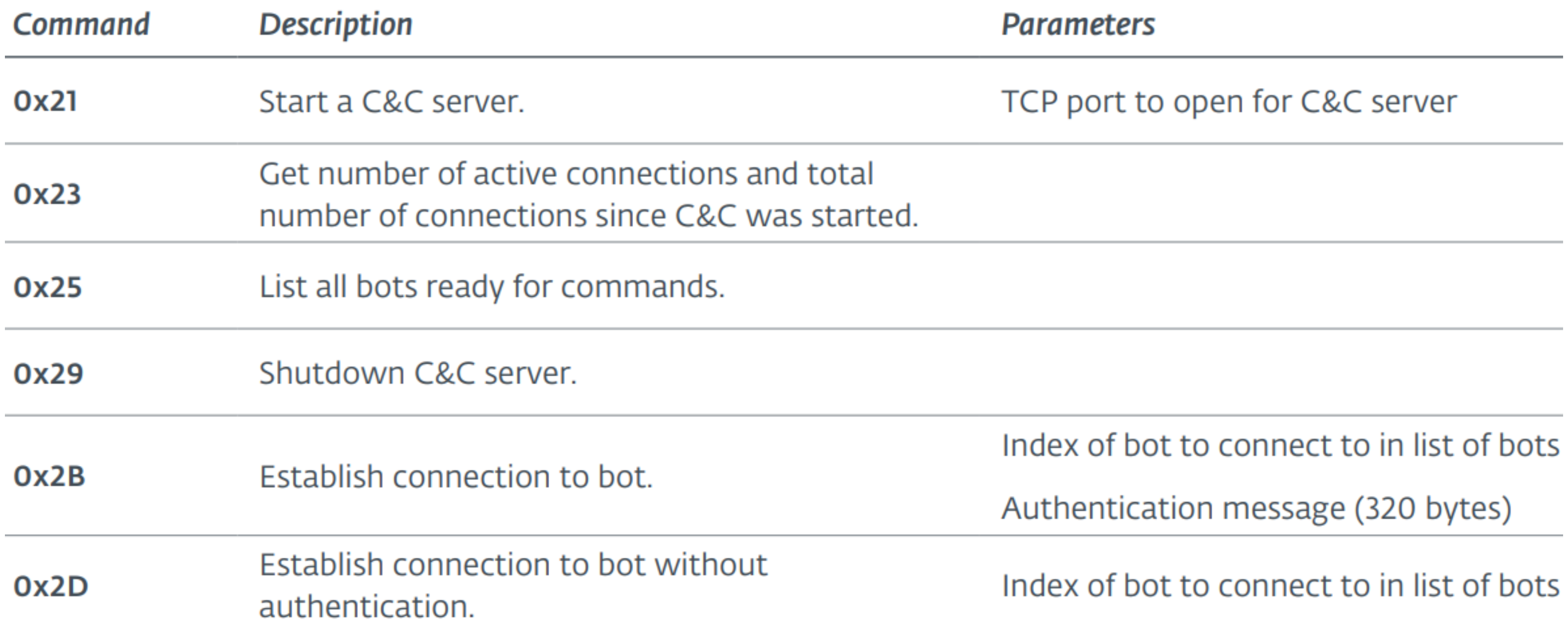
- This functionality allows an authenticated operator to spawn a shell in a new pseudo-terminal and execute arbitrary commands at the command prompt
- The data output from the terminal is sent back to the operators and uses the command ID 0x17 followed by the data



Command	*Description*	*Parameters*
0x12	Start a new pseudo-terminal.	Path to shell (e.g. `/bin/sh`) Argument
0x0D	Set pseudo-terminal window size.	Values of *winsize* struct as accepted by *TIOCSWINSZ*
0x14	Close pseudo-terminal.	*None*
0x16	Write to pseudo-terminal.	Data to write

Malware Operation – Use as a C&C server

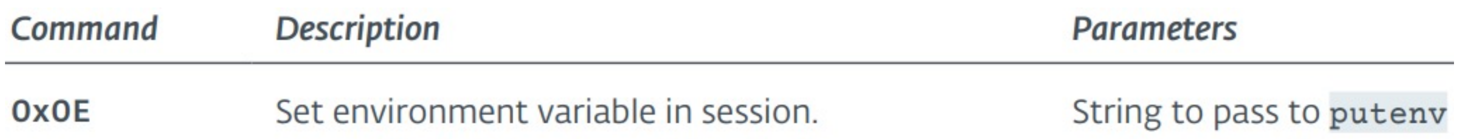
- One of the most unique features of Kobalos is that the code that runs the C&C server is in the malware itself
- This enables the perpetrator to use any Kobalos-compromised machine to turn it into a C&C server simply by sending a single command
- Two main advantages:
  - It allows using compromised resources as C&C servers
  - It allows using the C&C server as a pivot point to machines behind firewalls that are not normally reachable from the internet



Command	*Description*	*Parameters*
0x21	Start a C&C server.	TCP port to open for C&C server
0x23	Get number of active connections and total number of connections since C&C was started.	
0x25	List all bots ready for commands.	
0x29	Shutdown C&C server.	
0x2B	Establish connection to bot.	Index of bot to connect to in list of bots Authentication message (320 bytes)
0x2D	Establish connection to bot without authentication.	Index of bot to connect to in list of bots

Malware Operation – Other commands

- No-op
  - they do not perform any operations
  - They were removed because they were used in previous versions or they are platform-specific and do not apply to the Linux or FreeBSD variants of Kobalos
- Environment variables



Command	*Description*	*Parameters*
0x0E	Set environment variable in session.	String to pass to `putenv`

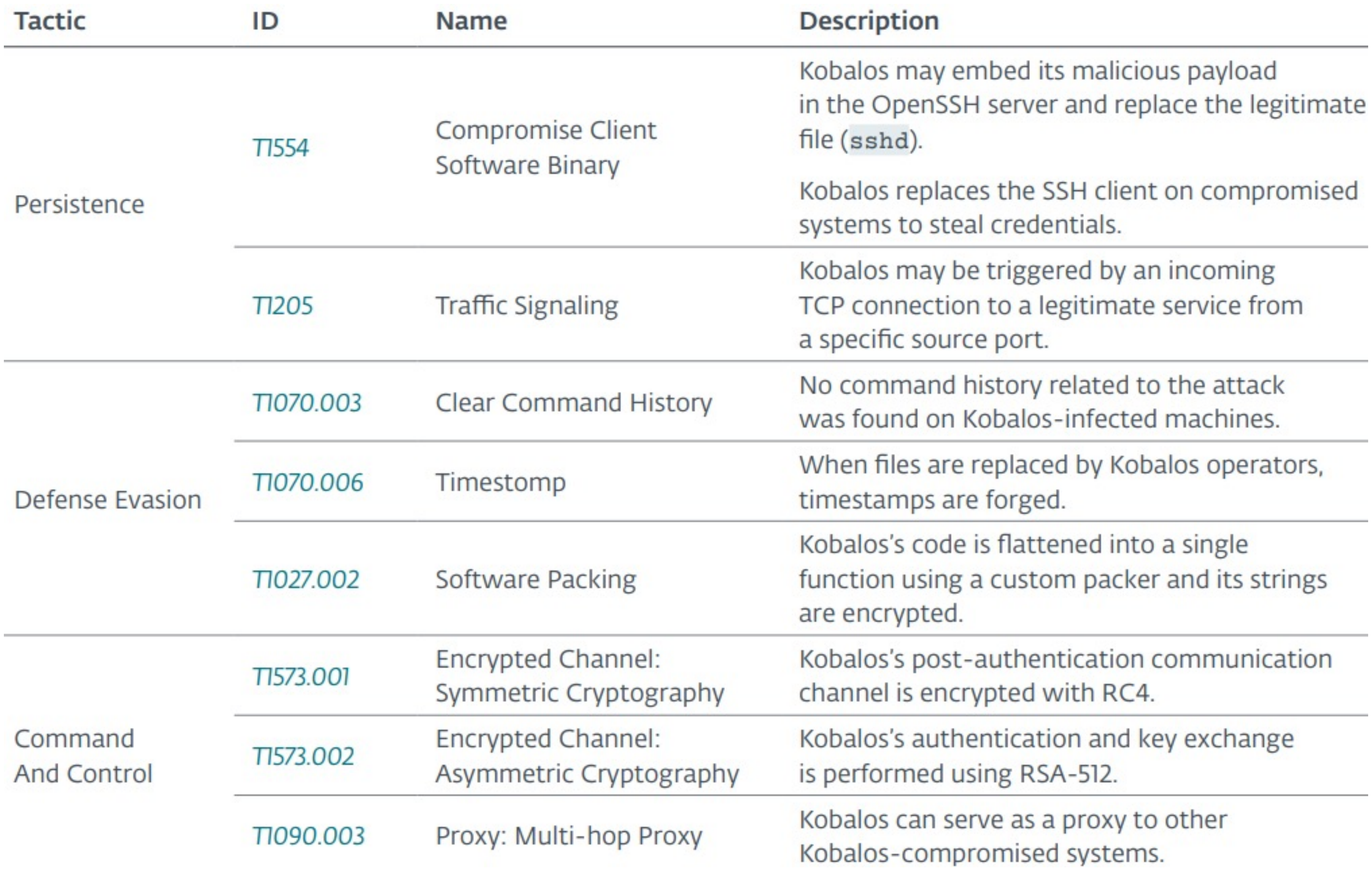
OpenSSH Credential Stealer

- malware was also deployed to steal SSH credentials via a trojanized SSH client
- Different variants including Linux and FreeBSD instances
- Unlike Kobalos, this credential stealer features almost no obfuscation
- Their main capabilities consist of stealing hostname, port, username and password used to establish an SSH connection from the compromised host
- They are saved into an encrypted file
- All samples create a file under /var/run with a ".pid" extension
- All samples found use the same simple cipher: it adds 123 to each byte of data to be saved

OpenSSH Credential Stealer – An evolving malware family

- This new version contains an encrypted configuration and adds the functionality to exfiltrate credentials over UDP to a remote host specified in the configuration
- The choice of UDP could be to bypass a firewall and avoid creating TCP network flow to potentially untrusted hosts
- It can only use one exfiltration method, file or network, because the configuration holds the target hostname and file path in the same variable

MITRE ATT&CK Techniques



Tactic	ID	Name	Description
Persistence	*T1554*	Compromise Client Software Binary	Kobalos may embed its malicious payload in the OpenSSH server and replace the legitimate file (`sshd`). Kobalos replaces the SSH client on compromised systems to steal credentials.
	T1205	Traffic Signaling	Kobalos may be triggered by an incoming TCP connection to a legitimate service from a specific source port.
Defense Evasion	*T1070.003*	Clear Command History	No command history related to the attack was found on Kobalos-infected machines.
	T1070.006	Timestomp	When files are replaced by Kobalos operators, timestamps are forged.
	T1027.002	Software Packing	Kobalos's code is flattened into a single function using a custom packer and its strings are encrypted.
Command And Control	*T1573.001*	Encrypted Channel: Symmetric Cryptography	Kobalos's post-authentication communication channel is encrypted with RC4.
	T1573.002	Encrypted Channel: Asymmetric Cryptography	Kobalos's authentication and key exchange is performed using RSA-512.
	T1090.003	Proxy: Multi-hop Proxy	Kobalos can serve as a proxy to other Kobalos-compromised systems.

Conclusion

- The attackers behind Kobalos are much more knowledgeable than the typical malware author targeting Linux and other non-Windows systems
- Its small footprint and network evasion techniques may explain why it went undetected until we approached victims
- One of the questions we cannot answer is what the intentions of the attackers are
- Another thing we couldn’t determine is how long this malware has been in use. We know there were new compromises in 2019 and 2020, but couldn’t find evidence of its usage before that time
- there are still a few weaknesses to Kobalos
  - issues with the cryptography due to advances in the field as Kobalos actually is very old
  - it’s possible to fingerprint variants passively listening, especially if it requires a specific TCP source port

References

- The MITRE Corporation. MITRE ATT&CK. https://attack.mitre.org/
- Marc-Etienne M. Léveillé Ignacio Sanmillan. A WILD KOBALOS APPEARS Tricksy Linux malware goes after HPCs . Technical report, ESET Research.
- Recent Attacks Against Supercomputers. https://www.cadosecurity.com/recent-attacks-against-supercomputers/
- Attacks on multiple HPC sites. https://csirt.egi.eu/attacks-on-multiple-hpc-sites/
- THE DARK SIDE OF THE FORSSHE. https://www.welivesecurity.com/wp-content/uploads/2018/12/ESET-The_Dark_Side_of_the_ForSSHe.pdf
- National Institute of Standards and Technology (NIST) Cybersecurity Framework: https://www.nist.gov/cyberframework