

SECURITY RESPONSE

Regin: Top-tier espionage tool enables stealthy surveillance

Symantec Security Response

Version 1.1 – August 27, 2015

“ *Regin is a multi-staged, modular threat, meaning that it has a number of components, each depending on others, to perform attack operations.* ”

CONTENTS

OVERVIEW	3
Introduction	5
Timeline.....	5
Target profile.....	6
Infection vector	6
Regin framework.....	8
Architecture.....	8
Command-and-control	14
64-bit version	17
Conclusion.....	18
Protection.....	18
Appendix	20
Data files	20
Indicators of compromise	20
File MD5s.....	20
File names/paths.....	21
Extended attributes	21
Registry	22

OVERVIEW

In the world of malware threats, only a few rare examples can truly be considered groundbreaking and almost peerless. What we have seen in Regin is just such a class of malware.

Regin is an extremely complex piece of software that can be customized with a wide range of different capabilities which can be deployed depending on the target. It is built on a framework that is designed to sustain long-term intelligence-gathering operations by remaining under the radar. It goes to extraordinary lengths to conceal itself and its activities on compromised computers. Its stealth combines many of the most advanced techniques that we have ever seen in use.

The main purpose of Regin is intelligence gathering and it has been implicated in data collection operations against government organizations, infrastructure operators, businesses, academics, and private individuals. The level of sophistication and complexity of Regin suggests that the development of this threat could have taken well-resourced teams of developers many months or years to develop and maintain.

Regin is a multi-staged, modular threat, meaning that it has a number of components, each depending on others, to perform attack operations. This modular approach gives flexibility to the threat operators as they can load custom features tailored to individual targets when required. Some custom payloads are very advanced and exhibit a high degree of expertise in specialist sectors. The modular design also makes analysis of the threat difficult, as all components must be available in order to fully understand it. This modular approach has been seen in other sophisticated malware families such as [Flamer](#) and [Weevil](#) (The Mask), while the multi-stage loading architecture is similar to that seen in the [Duqu/Stuxnet](#) family of threats.

Regin is different to what are commonly referred to as “traditional” advanced persistent threats (APTs), both in its techniques and ultimate purpose. APTs typically seek specific information, usually intellectual property. Regin’s purpose is different. It is used for the collection of data and continuous monitoring of targeted organizations or individuals. This report provides a technical analysis of Regin based on a number of identified samples and components. This analysis illustrates Regin’s architecture and the many payloads at its disposal.

INTRODUCTION

“ Regin has a wide range of standard capabilities, particularly around monitoring targets and stealing data.”

Introduction

Regin is a multi-purpose data collection tool which dates back several years. Symantec first began looking into this threat in the fall of 2013. Multiple versions of Regin were found in the wild, targeting several corporations, institutions, academics, and individuals.

Regin has a wide range of standard capabilities, particularly around monitoring targets and stealing data. It also has the ability to load custom features tailored to individual targets. Some of Regin's custom payloads point to a high level of specialist knowledge in particular sectors, such as telecoms infrastructure software, on the part of the developers.

Regin is capable of installing a large number of additional payloads, some highly customized for the targeted computer. The threat's standard capabilities include several remote access Trojan (RAT) features, such as capturing screenshots and taking control of the mouse's point-and-click functions. Regin is also configured to steal passwords, monitor network traffic, and gather information on processes and memory utilization. It can also scan for deleted files on an infected computer and retrieve them. More advanced payload modules designed with specific goals in mind were also found in our investigations. For example, one module was designed to monitor network traffic to Microsoft Internet Information Services (IIS) web servers, another was observed collecting administration traffic for mobile telephony base station controllers, while another was created specifically for parsing mail from Exchange databases.

Regin goes to some lengths to hide the data it is stealing. Valuable target data is often not written to disk. In some cases, Symantec was only able to retrieve the threat samples but not the files containing stolen data.

Timeline

Symantec is aware of two distinct versions of Regin. Version 1.0 appears to have been used from at least 2008 to 2011. Version 2.0 has been used from 2013 onwards, though it may have possibly been used earlier.

Version 1.0 appears to have been abruptly withdrawn from circulation in 2011. Version 1.0 samples found after this date seem to have been improperly removed or were no longer accessible to the attackers for removal.

This report is based primarily on our analysis of Regin version 1.0. We also touch on version 2.0, for which we only recovered 64-bit files.

Symantec has assigned these version identifiers as they are the only two versions that have been acquired. Regin likely has more than two versions. There may be versions prior to 1.0 and versions between 1.0 and 2.0.

Target profile

The Regin operators do not appear to focus on any specific industry sector. Regin infections have been observed in a variety of organizations, including private companies, government entities, and research institutes.

Infections are also geographically diverse, having been identified mainly in ten different countries.

Infection vector

The infection vectors of Regin are largely unknown. On one computer, log files revealed that Regin originated from Yahoo! Instant Messenger through an unconfirmed exploit. The sophisticated nature of the platform would indicate this vector is also highly complex in nature. The vector may vary per target and there is a high likelihood the attacker has access to zero-day exploits.

Targets may be tricked into visiting spoofed versions of well-known websites and the threat may be installed through a web browser or by exploiting an application.

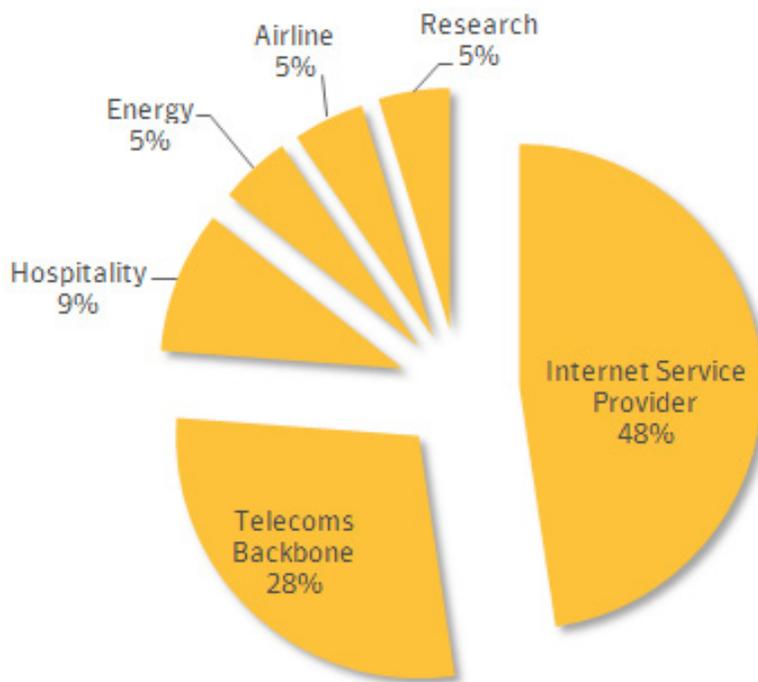


Figure 1. Confirmed Regin infections by sector

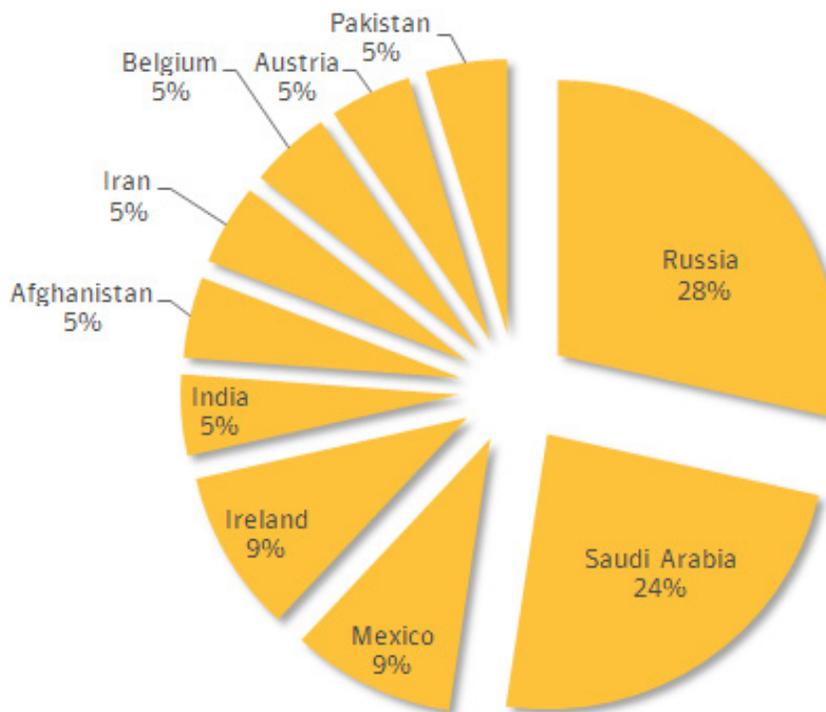
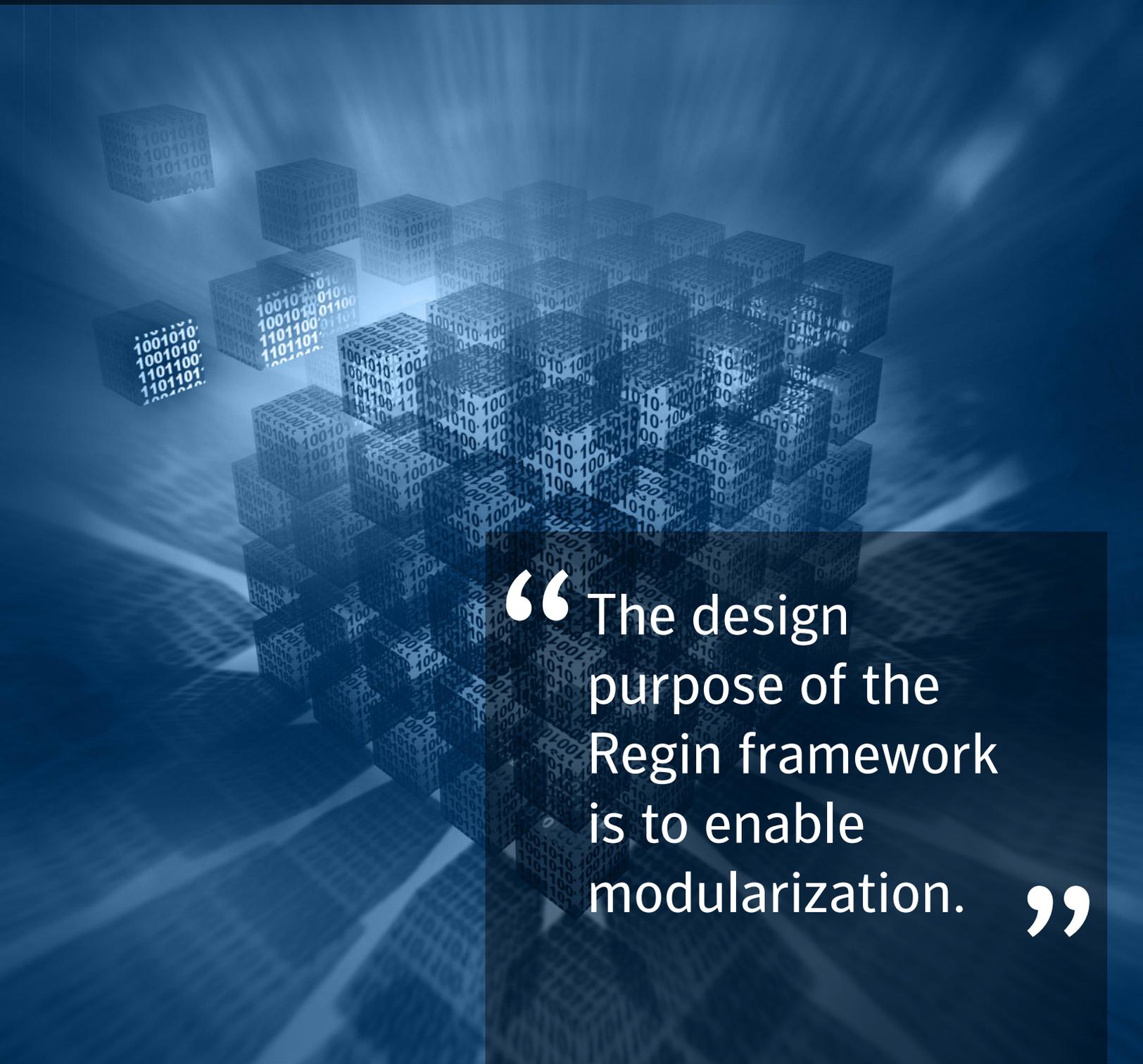


Figure 2. Confirmed Regin infections by region

REGIN FRAMEWORK



“ The design purpose of the Regin framework is to enable modularization. ”

Regin framework

Architecture

Regin has a six-stage architecture. The initial stages involve the installation and configuration of the threat's internal services. The later stages bring Regin's main payloads into play. This section of the paper presents an overview of the format and purpose of each stage. The most interesting aspects are the executables and data files stored in Stages 4 and 5.

The initial Stage 1 driver is the only plainly visible code on the computer. All other stages are stored as encrypted data blobs, as a file, or within a non-traditional file storage area such as the registry, extended attributes, or raw sectors at the end of disk.

The design purpose of the Regin framework is to enable modularization. The modularization is very fine-grained, starting with basic functionality and extending up to specific attacks, all implemented with common framework. For example, compression, encryption, logging, and password stealing are implemented as four separate units. Using fine-grained units allows for in-the-field updates of specific functionality or easy deployment of extensions when necessary (almost everything is extensible).

Stage 0 (dropper)

To date, Symantec Security Response has yet to obtain a Regin dropper. Symantec expects the dropper to install and execute the Stage 1 driver. It's likely that Stage 0 is also responsible for creating the extended attributes and/or registry keys and values that contain the additional stages providing Regin's remaining functionality.

Stages	Components
Stage 0	Dropper. Installs Regin onto the target computer
Stage 1	Loads driver
Stage 2	Loads driver
Stage 3	Loads compression, encryption, networking, and handling for an encrypted virtual file system (EVFS)
Stage 4	Utilizes the EVFS and loads additional kernel mode drivers, including payloads
Stage 5	Main payloads and data files

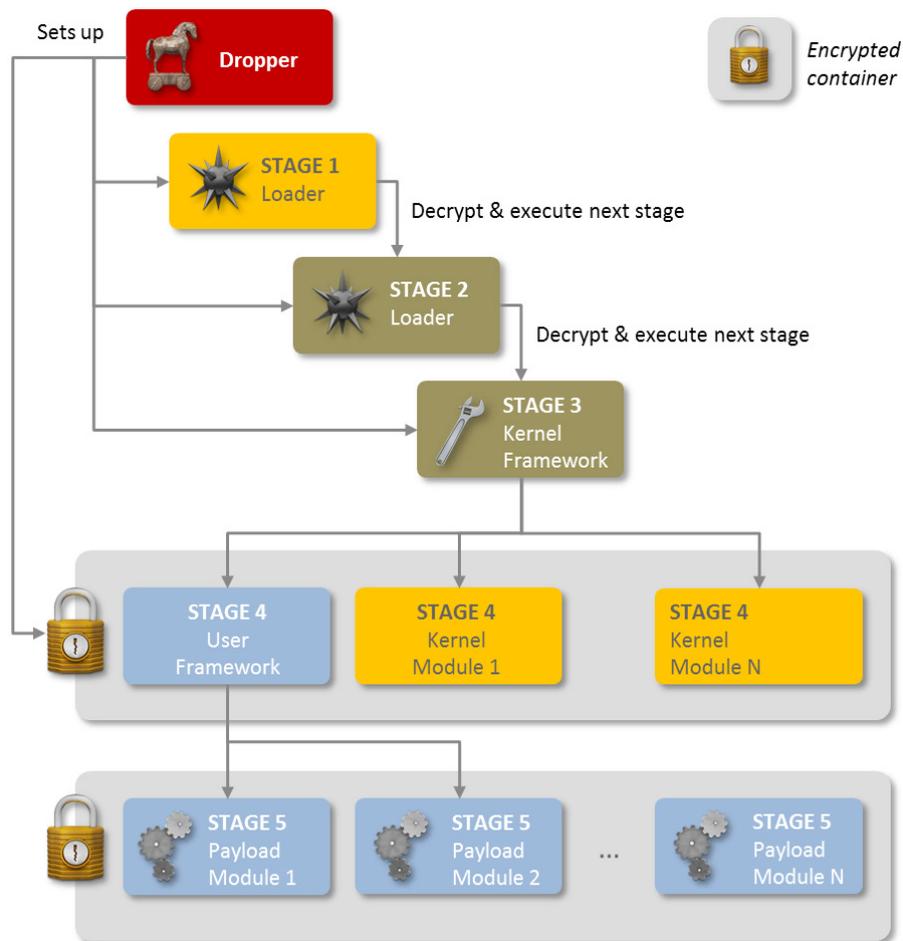


Figure 3. Regin's architecture

Since the dropper doesn't appear as an actual file on the system, it could be transient code existing only in memory rather than acting as an executable file. The dropper is quite likely built into the infection vector exploit code and is never written to disk.

Stage 1

Stage 1 can be considered a support module, whose purpose is to facilitate the loading of Stage 2. These kernel drivers may be registered as a system service or may have an associated registry key to load the driver while the computer is starting up. These loaders are the only part of the framework that may exist as an executable on the victim's computer.

Stage 1 will simply read and execute Stage 2 from NTFS extended attributes or registry key blobs, depending on which operating system is infected. The module exists in two forms, as a standalone malicious driver or as malicious code embedded within code of a legitimate device driver.

The following are example file names of known Stage 1 modules:

- usbclass.sys (version 1.0)
- adpu160.sys (version 2.0)

Note: For a full list of known file names, see the appendix.

Stage 2

Stage 2 is a kernel driver that simply extracts, installs, and runs Stage 3. Stage 2 is not stored in the traditional file system, but is encrypted within an extended attribute or a registry key blob.

This stage can also hide running instances of Stage 1. Once this happens, there are no remaining plainly visible code artifacts. Similar to previous stages, Stage 2 finds and loads an encrypted version of Stage 3 from either NTFS extended attributes or a registry key blob.

Stage 2 can also monitor the state of the threat. This stage drops the file msrdc64.dat, which appears to always be 512 bytes in size. The first two bytes are used and the remaining bytes are set to zero. The second byte indicates the exclusive maximum number of instances allowed to run, which is set to two. This means no more than one instance should run at any time. The first byte indicates how many instances were run or attempted to

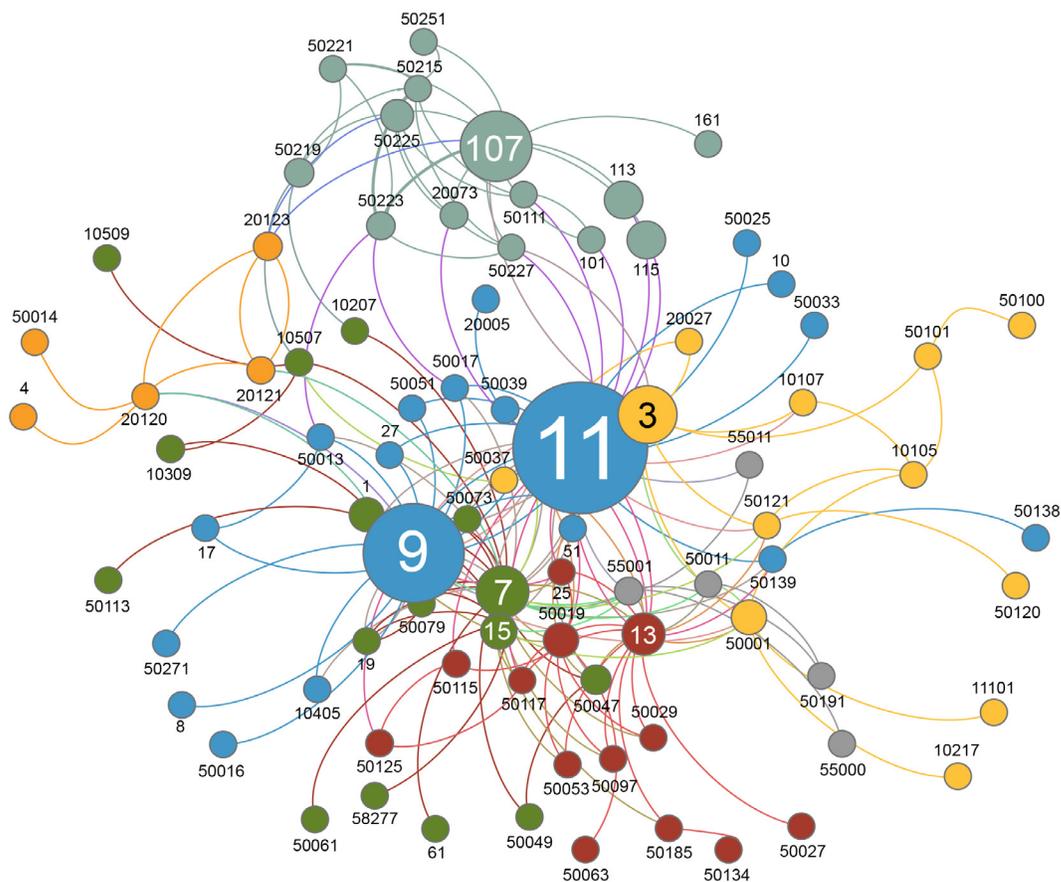


Figure 4. Regin's interdependent units

run. Therefore, the potential combinations for the first two bytes are:

- 00 02 (the threat is not running)
- 01 02 (the threat is running)
- 02 02 (the threat was running and a second instance has started)

The following configurations are examples of where Stage 2 can be found:

Example extended attribute:

- %Windir%
- %Windir%\fonts
- %Windir%\cursors (possibly only in version 2.0)

Example registry subkey

- HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\Class\{4F20E605-9452-4787-B793-D0204917CA58}
- HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\RestoreList\VideoBase (possibly only in version 2.0)

Note: For a full list of known folder names and registry keys, see the appendix.

Stage 3

Stage 3 is a kernel mode DLL and is not stored in the traditional file system. Instead, this file is encrypted within an extended attribute or registry key blob. The file is six to seven times the size of the driver in Stage 2. In addition to loading and executing Stage 4, Stage 3 offers a framework for the higher level stages.

Stages 3 and above are based on a modular framework of code modules. These modules offer functions through a private, custom interface (RPC Mechanism). The units in stages 3 and above can export functionality to other parts of the framework.

Stage 3 is internally known as *VMEM.sys* and exposes the functionality in Table 2.

The purpose of the Stage 3 modules are as follows:

- The orchestrator, which parses custom records found in the appended data of the executable files for stages 3 and above. These records contain a list of Regin functionalities to be executed. A record starts with the number 0xD912FEAB (little-endian ordering)
- Compression and decompression routines
- Encryption and decryption routines
- Routines to retrieve storage locations of higher-level (Stage 4) components
- Routines to handle an encrypted virtual file system used by Stage 4
- Network primitives

Example configurations of where Stage 3 can be found are as follows:

Extended attribute

- %Windir%\system32
- %Windir%\system32\drivers

Table 2. An example of Regin's methods organized into 12 groups

Unit (dec)	Unit (hex)	Functionality
101	0065h	Orchestrator
107	006bh	Virtual file system (VFS) access manager
113	0071h	Compression/decompression
115	0073h	Encryption/decryption (RC5)
161	00a1h	VFS access
50111	c3bfh	Inter process communication
50215	c427h	System information (OS/process)
50225	c431h	API hooking engine

Registry subkey

- HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\Class\{4F20E605-9452-4787-B793-D0204917CA5A}

Stage 4

The files for Stage 4, which are loaded by Stage 3, consist of a user-mode orchestrator and multiple kernel payload modules.

They are stored in two EVFS containers as files:

- %System%\config\SystemAudit.Evt
- %System%\config\SecurityAudit.Evt (DISP.DLL)

Note: Stage 4 uses the same export methodology described in Stage 3.

The user mode component of Regin's Stage 4 payload provides the functionality in Table 3.

The kernel mode component of Regin's Stage 4 payload provides the functionality in Table 4.

When the attackers behind Regin cleaned up compromised computers they often failed to remove Stage 4 and 5 artifacts from systems.

Stage 5

Stage 5 consists of the main Regin payload functionality. The files for Stage 5 are injected into services.exe by Stage 4.

Stage 5 files are EVFS containers containing other files:

- %System%\config\SystemLog.evt: Contains Stage 5 user mode DLLs that constitute Regin's payload
- %System%\config\SecurityLog.evt: Contains Stage 5 data files, used by the Stage 4 and 5 components to store various data items
- %System%\config\ApplicationLog.evt: Another Stage 5 log container, which is referenced by Stage 5 data files
- %Windir%\ime\imesc5\dicts\pintlgbp.imd (version 2.0)
- %Windir%\ime\imesc5\dicts\pintlgbbs.imd (version 2.0)

Regin's payload involves the DLLs contained in the SystemLog.evt EVFS container. The payload functionality differs depending on the targeted computer. Custom payload files will likely be delivered for each specific environment. Example payload functionality seen to date includes:

- Sniffing low-level network traffic
- Exfiltrating data through various channels (TCP, UDP, ICMP, HTTP)
- Gathering computer information
- Stealing passwords
- Gathering process and memory information
- Crawling through the file system
- Low level forensics capabilities (for example, retrieving files that were deleted)
- UI manipulation (remote mouse point & click activities, capturing screenshots, etc.)
- Enumerating IIS web servers and stealing logs

Table 3. Stage 4 usermode payloads (DISP.DLL)

Unit (dec)	Unit (hex)	Functionality
1	0001h	Core (user level)
7	0007h	VFS access management
9	0009h	Networking
11	000Bh	Event logging
13	000Dh	Compression/decompression
15	000Fh	Encryption/decryption (RC5)
17	0011h	Remote procedure call (RPC) helper
19	0013h	Peer nodes
25	0019h	UDP transport
51	0033h	Winlogon Autostart
61	003Dh	EVFS handling
50035	C373h	TCP Transport

Table 4. Stage 4 kernel payloads

Unit (dec)	Unit (hex)	Functionality
3	0003h	Universal unit interface
20073	4e69h	Port blocking
50115	c3c3h	Network packet filter driver
50211	c42dh	DLL loader (Unit from VFS)
50219	c42bh	PE loader (Unit from VFS)
50227	c433h	Rootkit (File System/Network)

- Sniffing GSM BSC administration network traffic

Framework units

Regin can be upgraded with various payload modules or receive payload modules after infection. The extensible nature of Regin and its custom payloads indicate that many additional payloads are likely to exist in order to enhance Regin's capabilities. Furthermore, we have found data files accompanying payload modules that have not been recovered.

Note: Some units listed below exist as payloads and are also embedded in the VMEM.SYS/DISP.DLL.

Table 5 describes the additional payloads which we have seen used by several variants of Regin.

Unit (dec)	Unit (hex)	Functionality
25	0019h	UDP transport
27	001bh	ICMP transport
10105	2779h	Call scheduling
10107	277bh	Impersonation
10207	27dfh	Logging (keyboard/clipboard/mouse)
10217	27e9h	Logging (IIS web server)
10309	2845h	Networking (mailslot/named pipe)
10405	28a5h	Timestamp conversion
10507	290bh	Credential retrieval (Windows/Outlook)
11101	2b5dh	Process/file forensics
20027	4e3bh	Browser stealing (proxy/sessions/user accounts)
20120	4e98h	User level keylogging
20121	4e99h	Utility keylogging
20123	4e9bh	Keyboard driver hooking
50001	c351h	File system forensics
50011	c35bh	File system monitoring
50013	c35dh	Security product identification
50015	c35fh	Retrieve system time (UNIX format)
50017	c361h	Crypto support
50019	c363h	Network packet capture
50025	c369h	Discovery (system/network)
50027	c36bh	User interface manipulation (screenshots, mouse, keyboard)
50029	c36dh	Packet sniffer and parser utility
50033	c371h	Hooking (Windows event log)
50035	c373h	TCP transport
50037	c375h	HTTP cookies transport
50047	c37fh	Packet sniffer and parser utility
50049	c381h	Credential harvesting (network captures)
50051	c383h	SSL transport
50053	c385h	Packet sniffer and parser utility
50061	c38dh	Asymmetric cryptography
50063	c38fh	Packet string filter compiler
50073	c399h	Event logging
50079	c39fh	Temporary file access
50081	c3a1h	RPC network interface

50097	c3b1h	Logging (DNS)
50101	c3b5h	System information collection
50113	c3c1h	Utility (linked lists)
50117	c3c5h	Network information collector
50121	c3c9h	List files
50125	c3cdh	Network communications intermediary
50139	c3dbh	Windows event log retrieval
50185	c409h	Credential harvesting (SAM/LSA)
50223	c42fh	Notifier (LoadImage/CreateProcess)
50231	c437h	Copy payload
50251	c44bh	Keyboard hooking
50271	c45fh	SMB transport
55001	d6d9h	Email read/write
55011	d6e3h	Log parser (MS Exchange)

Standalone units

Additional units, which also appear to be used in Regin but do not share the same export methodology as the framework units, have also been identified.

Table 6. Regin's standalone units

Internal name	Description
U_Alice	Download and execute remote payload
Hopscotch	Create and execute payload on remote computer
Legspin	Interactive console administration tool

Remote procedure call (RPC) mechanism

The Regin framework facilitates communication between units by providing a lightweight RPC mechanism. In a typical scenario, units that provide a higher-level functionality (for example, networking) rely on services provided by other units (such as TCP transport, UDP transport, etc.)

The RPC mechanism appears to be a custom implementation for the following reasons:

- No optimization is implemented for local invocations.
- Simple data representation; for example, serialized strings are terminated by NULL. Standard protocols are expected to handle embedded NULLs.
- Server implements inline de-serialization. Standard frameworks typically split de-serialization to reduce maintenance overhead.

Regin units expose their functionality as procedures accessible through this RPC mechanism. The procedure to be invoked by RPC is identified by the tuple described in Table 7.

Table 7. Regin RPC tuple format

Size	Element	Description
DWORD	Node address	Identifies specific Regin node (i.e infected computer)
WORD	Major id	Identifies specific Regin unit within the node
BYTE	Procedure id	Identified specific procedure within the Regin node

Regin used a virtual private network where the node address is similar to IPv4, but addresses are distinct from the actual computer IPv4 address. When Regin wants to communicate with the local computer it uses the constant 0x7F000001 which can be interpreted as 127.0.0.1 (localhost).

This RPC mechanism allows for procedure calls locally and across the network of Regin-infected computers. Operators can directly call any procedure on the Regin network to remotely control, install or update units or change unit configuration by replacing EVFS files.

Encrypted virtual file system containers

Regin stores data files and payloads on disk in encrypted virtual file system files. Such files are accessed by the major routines 3Dh. Files stored inside EVFS containers are encrypted with a variant of RC5, using 64-bit blocks and 20 rounds. The encryption mode is reverse cipher feedback (CFB).

Known extensions for EVFS containers are .evt and .imd. The structure of a container is similar to the FAT file system. One major difference is that files do not have a name; instead, they are identified using a binary tag. The tag itself is the concatenation of a major number and a minor number. The major number typically indicates the major function group that will handle the file.

A container starts with the header shown in Table 8 (little-endian ordering).

Table 8. The container's header

Offset	Type	Description
00h	WORD	Sector size in bytes
02h	WORD	Maximum sector count
04h	WORD	Maximum file count
06h	BYTE	File tag length (taglen)
07h	DWORD	Header CRC
0Bh	DWORD	File table CRC
0Fh	WORD	Number of files
11h	WORD	Number of sectors in use
13h	-	Sector-use bitmap

The header is followed by the file entry table (Table 9). Each file entry is 13h+taglen bytes long.

Table 9. The container's file entry table

Offset	Type	Description
00h	DWORD	CRC
04h	DWORD	File offset
08h	DWORD	Offset to first sector holding the file data
0Ch	BYTE[taglen]	File tag

The sectors follow (Table 10). A sector of sectsize bytes starts with a DWORD pointing to the next sector (if the file does not fit within a single sector), followed by sectsize-4 bytes of payload data.

Table 10. The container's sectors

Offset	Type	Description
00h	DWORD	Next sector offset, or 0
04h	BYTE[sectsize-4]	Data

As previously mentioned, the files are encrypted. Other layers of encryption and compression may also be in place, although those would be handled by higher-level components.

Logging

Regin logs data to the ApplicationLog.dat file. This file is not an encrypted container, but it is encrypted and compressed.

Command-and-control

Regin's command-and-control (C&C) operations are extensive. The C&C communications can be relayed through the network of Regin-infected computers. The networking protocols are extensible and they are configurable between



Figure 5. Physical layout of an EVFS container

each pair of Regin-infected computers. Furthermore, compromised computers can serve as a proxy for other infections and command and control can also happen in a peer-to-peer fashion.

Protocols

All communications are strongly encrypted and can happen in a two-stage fashion where the attacker may contact a compromised computer using one channel (knock) to instruct it to begin communications on another channel (conversation).

The protocol for establishing communications between a pair of computers is as follows:

1. A Regin node sends a knock to establish bidirectional communication with a target node. The knock includes details of requested transport to be used for bidirectional communication.
2. The nodes work together to establish which transport should be used for bidirectional communications.
3. RPC messages are then exchanged over the established transport in a secure conversation.

The operators can configure a variety of communication mechanisms, including a combination of different networking protocols for knock and conversation. For example, a pair of Regin nodes could use the following combination of protocols:

- Knock—Custom transport over UDP
- Conversation—Transport over named pipes (SMB)

The core of these knock and conversation protocols is implemented in unit 0009h. The actual representation of knock and conversation messages to be used with a specific network protocol is delegated to one of the transport providers, implemented as separate units within framework.

Unit 0x0009 is configured through its EVFS files. The configuration includes information about transport provider units to be used for exchanging knock and conversation messages over the network. This provides flexibility to the operators to configure and deploy additional transport providers by simply installing them and reconfiguring unit 0009h.

The units responsible for C&C, including these transports, are illustrated in Figure 6.

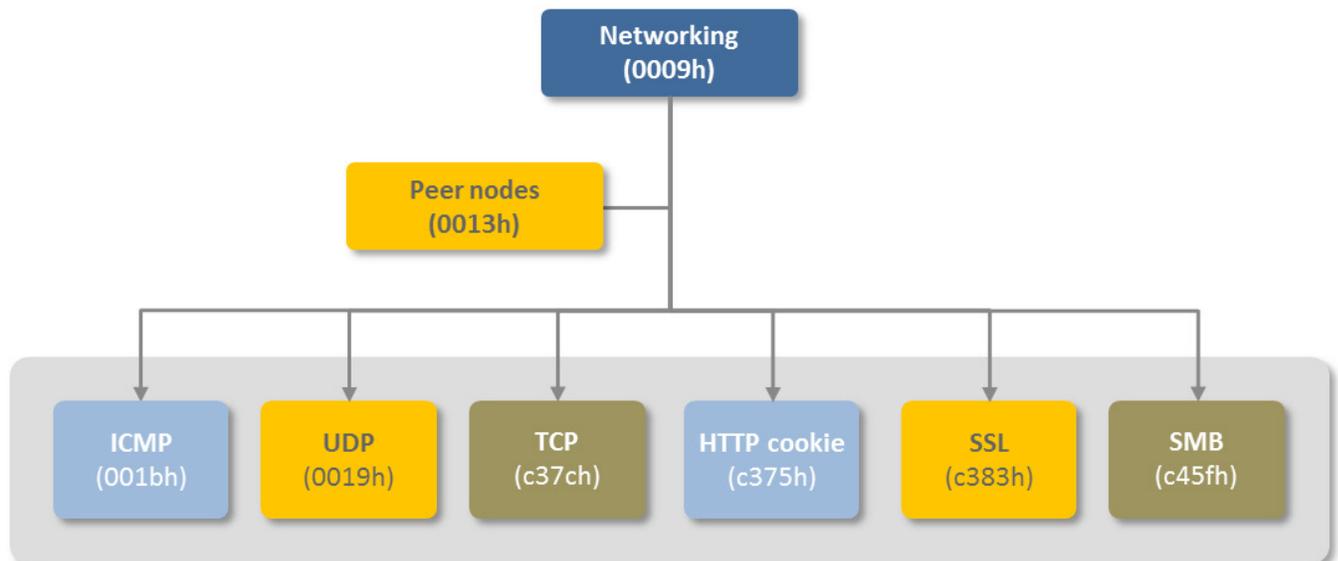


Figure 6. Regin's extendible C&C units

A total of six transport protocols have been identified for command and control between nodes, which include ICMP, UDP, TCP, HTTP Cookies, SSL, and SMB.

ICMP

ICMP transport is provided by unit 001bh, providing transport for the knock only. Regin communicates over ICMP using a custom protocol with 'shit' markers embedded in the communications for data validation. In addition, CRC checks using the seed '31337' are performed.

UDP

UDP transport is provided by unit 0019h, providing transport for both the knock and conversation. Regin communicates over UDP using a custom protocol with 'shit' markers embedded in the communications for data validation.

TCP

TCP transport is provided by unit c373h, providing transport for both the knock and conversation. Regin communicates over TCP using a custom protocol with 'shit' markers embedded in the communications for data validation.

HTTP cookie

HTTP cookie transport is provided by unit c375h, providing transport for the knock only. Regin communicates over HTTP cookies using a custom protocol with 'shit' markers embedded in the communications for data validation.

The request must use one of the following cookie names, the value is not important:

- USERID, TK=UID, GRID, UID, PREF=ID, TM, __utma, LM, TMARK, VERSION, CURRENT

The second cookie must use one of the following names, and contains the encoded message:

- SESSID, SMSWAP, TW=WINKER, TIMESET, LASTVISIT, ASP.NET_SessionId, PHPSESSID, phpAds_id

SSL

SSL transport is provided by unit c383h, providing transport for both the knock and conversation. Regin communicates over SSL using a custom protocol. This unit appears to be built from a version of Open SSL (0.9.7b).

SMB

SMB transport is provided by unit c475fh, providing transport for both the knock and conversation. Regin communicates over SMB using a custom protocol with 'shit' markers embedded in the communications for data validation. The named pipes created are generated by a seed.

Topology

Regin is designed as a peer-to-peer network. The network adopts a virtual private network (VPN) on top of the physical network of the infected host. Each Regin node is assigned a virtual IP address, which forms the virtual network on top of the physical network. Unit 0009h and 0013h handle communications on the virtual network between peers, whilst the various transport providers exchange the data over the underlying physical network.

The Regin operators can configure communications between nodes to enable:

1. Deep access to critical assets within a compromised organization
2. Establish "trusted" communication links between trusted organizations/sub-organizations
3. Mask core attack infrastructure

The operators also appear to configure the knock and conversation traffic to match expected protocols depending on where the infected Regin node exists on the network.

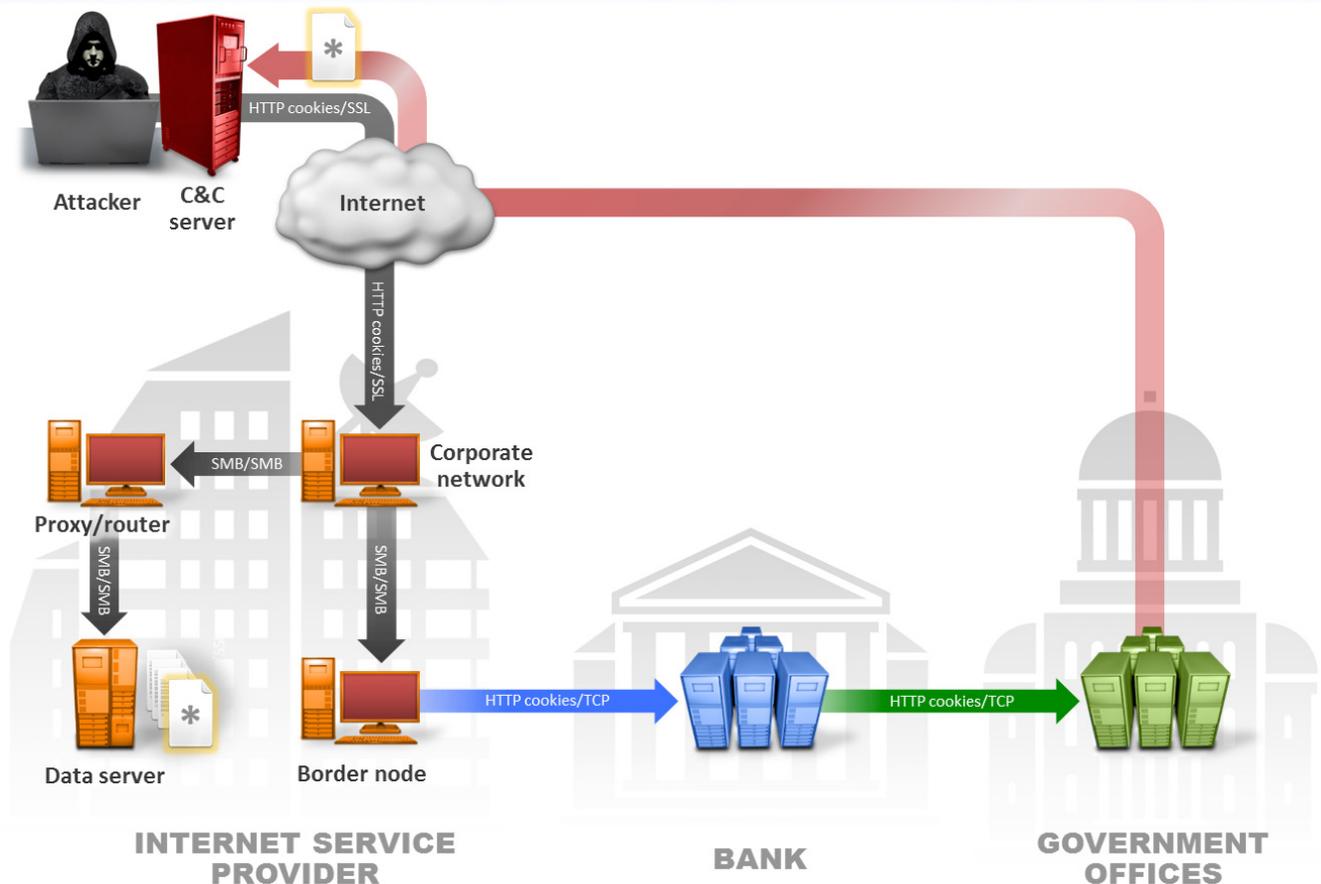


Figure 7. Example network topology of infected organisation

64-bit version

Only a small amount of the 64-bit Regin files have been recovered. These samples may represent version 2.0 or their differences may possibly be solely specific to 64-bit versions of Regin. We also recovered files from infected computers that may or may not be associated with 64-bit Regin, including several variants of `svcsstat.exe`, a file that aims to retrieve binary data over pipes or sockets and execute the data.

File names

The recovered files do not appear to fundamentally vary from their 32-bit counterparts, apart from a few noteworthy differences.

The 32-bit and 64-bit versions of Regin use different file names. These differences are shown in the first section of this paper as well as in the appendix. Most importantly, in the 64-bit version of Regin, the names of containers are changed:

- PINTLGBP.IMD replaces SystemLog.Evt
- PINTLGBPS.IMD replaces SecurityLog.Evt

Stage differences

The 64-bit version of Regin's Stage 1 (`wshnetc.dll`) is no longer a kernel mode driver, as drivers under 64-bit Windows must be signed. Instead, Stage 1 is a user mode DLL loaded as a Winsock helper when the computer is starting up. Rather than loading Stage 2 from an NTFS extended attribute, Stage 1 looks for the last partition (in terms of physical location) on disk and searches for the payload in the raw sectors in this area of the disk.

The 64-bit Regin's Stage 3 has not been recovered. We believe that it may not exist, as the 32-bit version is a driver. Stage 4 is an orchestrator, just like its 32-bit counterpart, and it uses the same major and minor values to export functionality.

Stage 5 uses the following file names:

- %Windir%\IME\IMESC5\DIRTS\PINTLGBP.IMD contains Stage 5 user payloads, replacing SystemLog.Evt in the 32-bit version
- %Windir%\IME\IMESC5\DIRTS\PINTLGBS.IMD contains Stage 5 data files, replacing SecurityLog.Evt in the 32-bit version
- The equivalent files for SystemAudit.Evt and SecurityAudit.Evt were not recovered

No Stage 5 payload modules have been recovered.

Conclusion

Regin is a highly-complex threat which has been used for large-scale data collection or intelligence gathering campaigns. The development and operation of this threat would have required a significant investment of time and resources. Threats of this nature are rare and are only comparable to the Stuxnet/Duqu family of malware. The discovery of Regin serves to highlight how significant investments continue to be made into the development of tools for use in intelligence gathering. Many components of Regin have still gone undiscovered and additional functionality and versions may exist.

Protection

Symantec and Norton products detect this threat as [Backdoor.Regin](#).

APPENDIX



Appendix

Data files

Regin's data files are classified as Stage 5 components and are contained in an EVFS container.

As the data files are stored in a container, they do not have names. Just like Stage 5 modules, they are referenced by their filetag, which is the aggregation of the major and minor identifiers. The major identifier indicates which major routine group likely handles or creates the file.

Not all data files have been recovered, so the information in Table 11 remains incomplete.

Table 12 lists recovered data files used by Stage 5 modules:

The associated modules that supposedly manipulate those data files were not recovered.

Indicators of compromise

The following details can be used to help determine whether you have been impacted by this threat.

File MD5s

2c8b9d2885543d7ade3cae98225e263b

4b6b86c7fec1c574706cecedf44abded

187044596bc1328efa0ed636d8aa4a5c

06665b96e293b23acc80451abb413e50

d240f06e98c8d3e647cbf4d442d79475

6662c390b2bbbd291ec7987388fc75d7

ffb0b9b5b610191051a7bdf0806e1e47

b29ca4f22ae7b7b25f79c1d4a421139d

1c024e599ac055312a4ab75b3950040a

ba7bb65634ce1e30c1e5415be3d1db1d

b505d65721bb2453d5039a389113b566

b269894f434657db2b15949641a67532

bfbe8c3ee78750c3a520480700e440f8

Table 11. Data files used by Stage 4's framework DLL

Major	Minor	Description
0001	-	-
000D	-	-
000F	01	High-entropy blobs, cryptographic data
	02	High-entropy blobs, cryptographic data
003D	-	-
0007	-	-
000B	01	Contains a path to the log file. Typically, %System\config\ApplicationLog.Evt
	02	Small 8 byte files
0033	01	A single DWORD, such as 111Ch
	03	A single DWORD, such as 1114h
0011	-	-
0013	01	Unknown list of records
	02	A single byte, such as 3
C373	01	BPF bytecode for the netpcap driver—allows UDP passthrough
	02	A WORD value, such as 1
0019	01	BPF bytecode for the netpcap driver—allows TCP passthrough
	02	A WORD value, such as 1
0009	00	A single DWORD, such as 11030B15h
	01	Contains C&C location information
	02	C&C routines to be executed: <ul style="list-style-type: none"> • (C375, 1) param= 08 02 • (19, 1) param= 44 57 58 00 • (C373, 1) param= 08 02 • (1B, 1) param= 20 00
	03	Routines to be executed <ul style="list-style-type: none"> • (4E69, 2) • (19, 2) • (1B, 2) • (C373, 2) • (C375, 2) • (C383, 2) • (C363, 2)
	07	RC5 key used to decrypt command-and-control packets
	09	Unknown data
	0B	Unknown data
	12	A single byte, such as 1
	17	Unknown data

File names/paths

abiosdsk.sys
 adpu160.sys
 floppy.sys
 parclass.sys
 rio8drv.sys
 ser8uart.sys
 usbclass.sys
 vidscfg.sys
 msrdc64.dat
 msdcsvc.dat
 %System%\config\SystemAudit.Evt
 %System%\config\SecurityAudit.Evt
 %System%\config\SystemLog.evt
 %System%\config\ApplicationLog.evt
 %Windir%\ime\imesc5\dicts\pintlgs.imd
 %Windir%\ime\imesc5\dicts\pintlgbp.imd
 %Windir%\system32\winhttp.dll
 %Windir%\system32\wshnetc.dll
 %Windir%\SysWow64\wshnetc.dll
 %Windir%\system32\svcstat.exe
 %Windir%\system32\svcsstat.exe

Extended attributes

%System%\CertSrv
 %System%\mui
 %System%\npp
 %System%\Spool\Printers
 %Windir%\cursors
 %Windir%\fonts
 %Windir%\help
 %Windir%\inf

Table 12. Data files used by Stage 5's modules (payloads)

Major	Minor	Description
C363	02	6 bytes (01 00 00 00 00 00)
4E3B	-	
	02	High-entropy blobs, cryptographic data
290B	-	
C375	01	Dword (1)
	02	Dword (0)
C383	01	Dword (1)
	02	Dword (0)
	10	64 bytes (512 bits) Diffie Hellman, p (prime)
	11	Byte (2) iffie Hellman, g (generator)
C361	10	File containing timestamps and high entropy data
	11	Dword (E10h)
	12	Dword (2)
001B	-	
C399	-	
C39F	00	Small file, 18h bytes, low entropy
	01	Unencrypted unicode path, %Temp%\~B3Y7F.tmp
C3A1	01	Small file, 6 bytes (08 01 00 00 00 01)
28A5	02	Small file, 18h bytes, unknown
C3C1	-	-
C3B5	-	-
C36B	-	-
C351	-	-
2B5D	-	-
C3CD	-	-
C38F	-	-
C3C5	-	-
27E9	-	-

Table 13. Orphaned data files

Major	Minor	Description
4E25	00	Byte (1)
	01	Byte (2)
28A4	00	Unknown
	02	Small file, 8 bytes (01 00 00 00 00 00 00 00)
DEAB	01	Small file, 8 bytes (00 00 01 01 04 00 00 00)

%Windir%\msagent

%Windir%\msapps

%Windir%\repair

%Windir%\security

%Windir%\temp

%Windir%\System32

%Windir%\System32\drivers

Registry

HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\Class\{4F20E605-9452-4787-B793-D0204917CA58}

HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\Class\{4F20E605-9452-4787-B793-D0204917CA5A}

HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\Class\{9B9A8ADB-8864-4BC4-8AD5-B17DFDBB9F58}

HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\RestoreList\VideoBase

HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\Session\{5D42A36B-12C4-DE7C-4BD1-0612BD1CF324}



About Symantec

Symantec Corporation (NASDAQ: SYMC) is an information protection expert that helps people, businesses and governments seeking the freedom to unlock the opportunities technology brings -- anytime, anywhere. Founded in April 1982, Symantec, a Fortune 500 company, operating one of the largest global data-intelligence networks, has provided leading security, backup and availability solutions for where vital information is stored, accessed and shared. The company's more than 19,000 employees reside in more than 50 countries. Ninety-nine percent of Fortune 500 companies are Symantec customers. In fiscal 2015, it recorded revenues of \$6.5 billion.

To learn more go to www.symantec.com or connect with Symantec at: go.symantec.com/social/.

 Follow us on Twitter
@threatintel

 Visit our Blog
<http://www.symantec.com/connect/symantec-blogs/sr>

For specific country offices and contact numbers, please visit our website.

Symantec World Headquarters
350 Ellis St.
Mountain View, CA 94043 USA
+1 (650) 527-8000
1 (800) 721-3934
www.symantec.com

Copyright © 2015 Symantec Corporation. All rights reserved. Symantec, the Symantec Logo, and the Checkmark Logo are trademarks or registered trademarks of Symantec Corporation or its affiliates in the U.S. and other countries. Other names may be trademarks of their respective owners.

Any technical information that is made available by Symantec Corporation is the copyrighted work of Symantec Corporation and is owned by Symantec Corporation.

NO WARRANTY . The technical information is being delivered to you as is and Symantec Corporation makes no warranty as to its accuracy or use. Any use of the technical documentation or the information contained herein is at the risk of the user. Documentation may include technical or other inaccuracies or typographical errors. Symantec reserves the right to make changes without prior notice.