Sality

P. Kleissner[1]

[1] Kleissner & Associates s.r.o., Prague, Czech Republic
A Lookingglass Cyber Solutions Company

E-mail contact of main author: info@kleissner.org

Abstract. Sality is one of the longest-alive threats and probably the most underrated botnet ever. It made its first appearance in 2003 and is still active in 2015. There are more than 2 million active infections (as per 24 hours) and it has advanced features like a peer-to-peer botnet, a rootkit which is able to kill AVs and a nasty file infector. This paper provides technical insight into the threat.

Key Words: Sality, File Infector, P2P Botnet, Rootkit

1. Executive Summary

According to Symantec, Sality appeared first in April 30, 2003. [1] Today, there appear more than 2 million unique (as per IP address) Sality infections per day on Virus Tracker, a botnet monitoring system based on sinkholes. It is estimated (based on the observation in Virus Tracker and its coverage of sinkholed botnets) that there are right now about 4 million infected devices with Sality worldwide.

Sality is likely the oldest still actively maintained Trojan. The success of Sality lies in the simplicity but effectiveness of the virus and thus staying under the radar. It is known to be a general purpose botnet, sending out spam, stealing login information and installing additional malware; however it is not known for financial fraud, espionage or ddos attacks.

The key features are a rootkit, a peer-to-peer (P2P) algorithm and its file infector. There are two active separate P2P networks. They only differ in the encryption used for signing commands (RSA 1024 vs RSA 2048) and a file transfer functionality via TCP.

Research revealed 3 potential co-authors of Sality, of which one's real name was likely discovered because an email address used by early samples for sending exfiltration emails was used for a Russian social media profile.

2. Authors of Sality

Research indicates that in the beginning there are 3 co-authors of Sality with the nicknames "Sector", "iMAGER" and "Alien-Z". Those nicknames appear in the exfiltration emails (containing stolen information) sent by early versions from the infected machines to the operators (see Figure 1 below).

Symantec provides information on the origin of the Trojans name and other hints [1]:

"*the curious reader asking where the name "Sality" originated from now has the answer: it is derived from "Salavat City", a Russian town from which the author may originate. This threat bears a couple of other names, also related to strings found inside the payload: "Kuku" (which means Hide-and-Seek in Russian), or "Sector" (the nickname of the author).*"

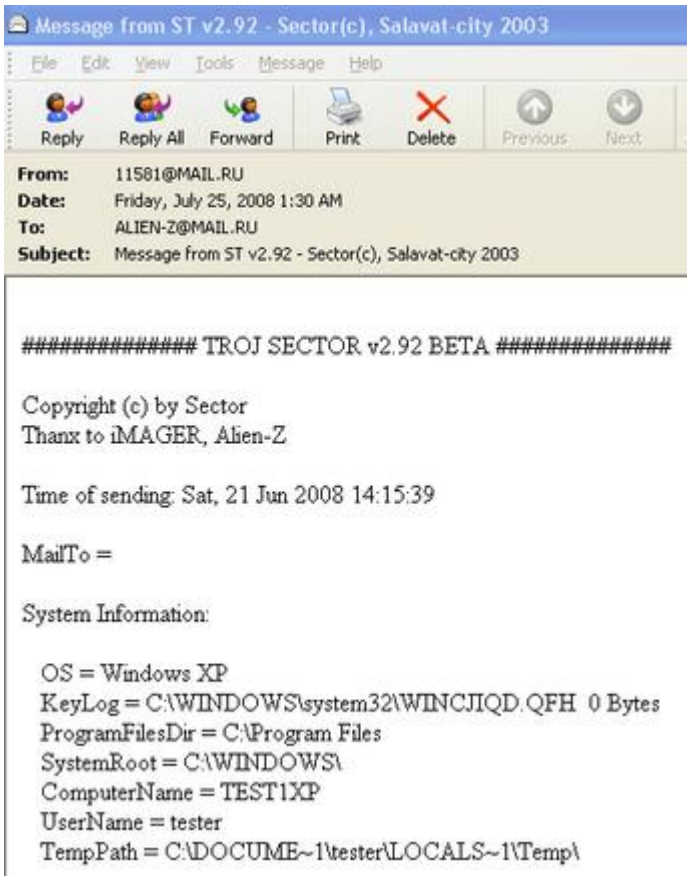


FIG. 1. Screenshot of an exfiltration email, © ESET, [6]

Analysis of multiple samples from its early days revealed that the email 11581@mail.ru was always used as sender, and as receiver the mails lamercool@rambler.ru, alien-z@mail.ru, and imager@mail.ru were observed.

The email address alien-z@mail.ru is associated with a social media profile at http://my.mail.ru/mail/alien-z/ (under the assertion that the email address was not re-used or the profile intentionally faked). Additional information, attribution and conclusions on this handle is not presented in this paper for privacy reasons and law enforcement investigation purposes.

A lookup on the Russian forum "verified.ru" (now under verified.mn) shows 2 users with the names "Sector" and "sect0r", the first one registered on 10/24/2006. There is an email sect0r@mail.ru registered with a social media profile under http://my.mail.ru/mail/sect0r/, however, whether it is attributed to the same person remains uncertain.

FIG. 2. Users named "Sector" and "sect0r" on the verified.ru hacking forum.

On the third email and nickname imager@mai.ru no additional information was discovered.

3. Early Version Numbers

Early versions date back to 2003. Two version strings of the packer used by Sality were found in samples, "Simple Poly Engine v1.1a (c) Sector" and "Simple Poly Engine v1.2a (c) sector". A Sophos report [5] states that "*On the 10-12th of the month, when the minute equals the hour, the following message is displayed with the title 'Win32.HLLP.Kuku v2.91'*":

```
<<<<<Hey, Lamer! Say "Bye-bye" to your data! >>>>>
Copyright (c) by Sector'
```

FIG. 3. Message displayed by early Sality samples. Via [5].

By looking up "Win32.HLLP" multiple version strings are found, which reveal different version numbers and years:

String in Binary	***Title in Email***
Win32.HLLP.Kuku v1.02	
Win32.HLLP.Kuku v1.09	
Win32.HLLP.Kuku v2.05	Message from ST v2.05 - Sector(c), Salavat-city 2003
Win32.HLLP.Kuku v2.91	
Win32.HLLP.Kuku v2.92	Message from ST v2.92 - Sector(c), Salavat-city 2003
	Message from ST v2.93 - Sector(c), Salavat-city 2004
Win32.HLLP.Kuku v3.09	

Table I. Version indicators from strings found in samples.

Figure 4 shows the SMTP traffic observed in Wireshark when running sample MD5 52AE3B7F8F383F169363B5D4F5D5DECA. It was not sending successfully the email because the SMTP relay mail.ru is now only accepting encrypted connections with SSL or TLS.

```
220 smtp47.i.mail.ru ESMTP ready
HELO MAIL.RU
250 smtp47.i.mail.ru
MAIL FROM:<11581@MAIL.RU>
250 2.0.0 OK
RCPT TO:<11581@MAIL.RU>
550 SMTP is available only with SSL or TLS connection enabled.


220 smtp19.mail.ru ESMTP ready
HELO MAIL.RU
250 smtp19.mail.ru
MAIL FROM:<11581@MAIL.RU>
250 2.0.0 OK
RCPT TO:<IMAGER@MAIL.RU>
550 SMTP is available only with SSL or TLS connection enabled.
```

FIG. 4. SMTP traffic of an early Sality sample.

4. Infection Statistics via Virus Tracker

All statistics are based on unique IP addresses per day. If there are multiple infections on the same day behind one single IP (as it's the case with NAT) it will be still counted as one infection. If there is one infection connecting on the same day from multiple IPs (for example

because of reconnecting to the ISP or a laptop travelling to multiple sites) it will be counted as multiple infections. Statistics in this section are accurate as of 9/23/2015.

Figure 5 shows the total unique daily infections. The increase in 2014 is due to broader coverage of Sality botnets in Virus Tracker (i.e. more Sality botnets sinkholed). The outages shown are due to server upgrades, ddos attacks or other server failures. The general decrease of infections over time is a combined result of people installing AVs (and removing Sality) and throwing away their computers and buying new ones.

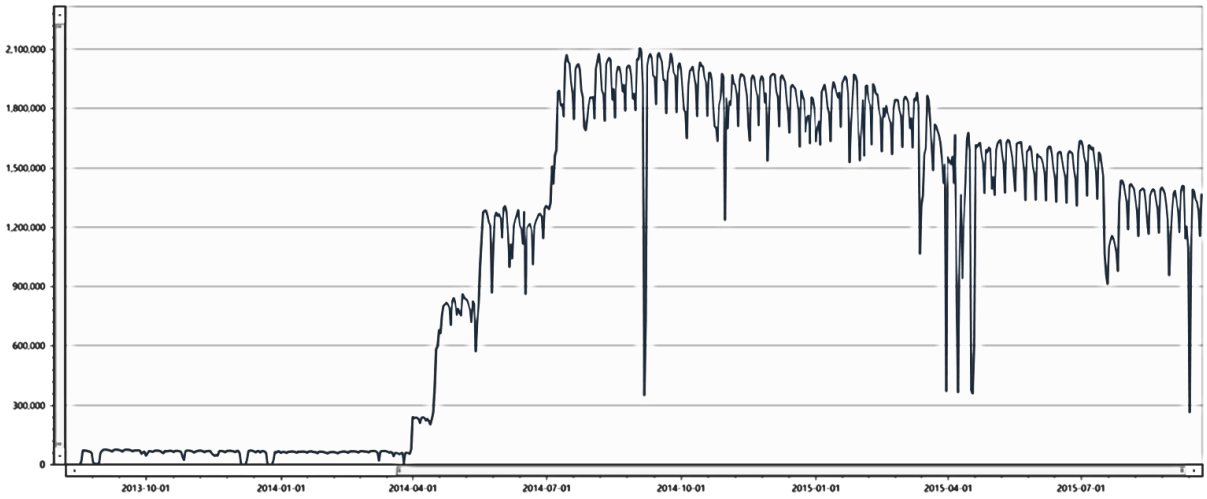
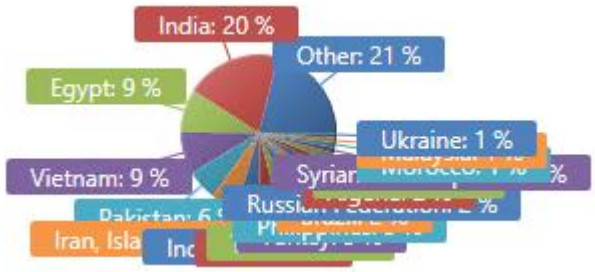


FIG. 5. Total unique daily infections 2013 – 2015.

20% of the total infections observed were seen via P2P – Kleissner & Associates has a custom written P2P crawler that implements the P2P algorithm of Sality and crawls the botnet 24/7. 80% of the total infections were observed on the domain sinkholes.

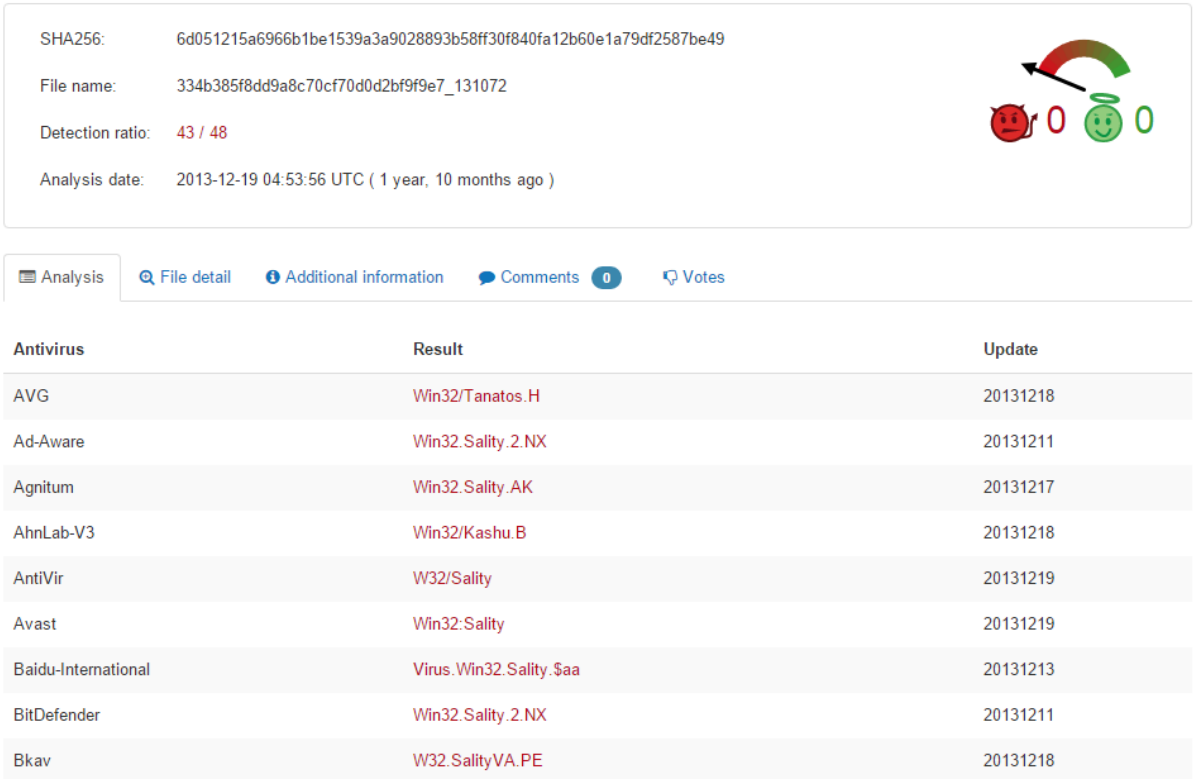
Figure 6 shows the country statistics. Most of the infections were reported from India, followed by Egypt and Vietnam. Countries with <1% are summarized to "Other".



%	Absolute	Country
21.21	284.029	Other
19.67	263.403	India
8.99	120.421	Egypt
8.69	116.346	Vietnam
6.27	83.975	Pakistan
5.51	73.763	Iran, Islamic Republic of
4.73	63.282	Indonesia
3.97	53.154	China
3.33	44.616	Thailand
2.62	35.111	Turkey
2.57	34.376	Philippines
2.48	33.198	Brazil
2.29	30.644	Russian Federation

FIG. 6. Country statistics.

The reason that mostly emerging/3rd world countries are most infected is because computers in those countries typically lack anti-virus software and use pirated Windows versions with OS updates disabled. Antivirus detection of Sality samples is always very high, Figure 7 shows 43/48 for sample MD5 334B385F8DD9A8C70CF70D0D2BF9F9E7:



SHA256: 6d051215a6966b1be1539a3a9028893b58ff30f840fa12b60e1a79df2587be49

File name: 334b385f8dd9a8c70cf70d0d2bf9f9e7_131072

Detection ratio: 43 / 48

Analysis date: 2013-12-19 04:53:56 UTC (1 year, 10 months ago)

0 0

Analysis | File detail | Additional information | Comments 0 | Votes

Antivirus	Result	Update
AVG	Win32/Tanatos.H	20131218
Ad-Aware	Win32.Sality.2.NX	20131211
Agnitum	Win32.Sality.AK	20131217
AhnLab-V3	Win32/Kashu.B	20131218
AntiVir	W32/Sality	20131219
Avast	Win32:Sality	20131219
Baidu-International	Virus.Win32.Sality.$aa	20131213
BitDefender	Win32.Sality.2.NX	20131211
Bkav	W32.SalityVA.PE	20131218

FIG. 7. High detection rates for Sality samples.

5. Ddos attacks against Virus Tracker

There were 3 ddos attacks against Virus Tracker which were allegedly operated by the Sality operators. Table 2 shows them over time. In first two attacks the only IP observed to ddos via TCP was known as a Sality infection on the very same day. The attack was executed against a Virus Tracker sinkhole server which was used for sinkholing among other viruses also Sality.

#	*Date*	*Type*
#1	November 27, 2014	1 Gbps ddos ICMP + UDP + TCP
#2	January 30, 2015	10 Gbps ddos UDP + TCP + NTP amplification
#3	March 13, 2015	120 Gbps ddos NTP + DNS amplification

Table II. Ddos attacks launched against Virus Tracker allegedly from Sality.

Additional information on these ddos attacks are available in the Botconf presentation to this paper.

6. Technical Information

There are 4 different P2P networks, of which only network 3 (appeared in 2009) and 4 (since 2010) are still active. Except different key length (RSA 1024 vs 2048) and a file exchange functionality via TCP the protocol of both networks are identical.

In the P2P network the bots exchange URL packs, IPs of other peers and for network 4 also binaries. Both URL packs and binaries exchanged in the P2P network are signed; therefore only the operator can send updates.

Peers also keep a "goodcount" value of each peer to know which ones to keep and drop from its peer lists – which makes disruption attacks more difficult. Peers also automatically detect whether they are supernodes (i.e. reachable from the outside, no NAT or port-forwarding enabled) or not.

The entire protocol is pretty flat and simple; it uses UDP for sending & receiving commands. Its implementation reminds to TCP; peers send "ok" responses (handshake) and it opens a new UDP port for every connection and has a time-out. The default local port to listen is 9674 but calculated from the local computer name if available.

There are 3 commands: 1 = Announcement & Promotion, 2 = Peer Exchange, 3 = Pack Exchange. Figure 8 shows the packet (and payload) structures.

```
#pragma pack(push, 1)

struct Sality_Packet
{
    WORD Hash;          // 16 bit custom hash
    WORD Size;          // of the data following
    BYTE Version;       // = 4 current
    DWORD UrlPackId;    // Id number of URL pack
    BYTE Command;       // 1 = Announcement & Promotion, 2 = Peer Exchange, 3 = Pack
Exchange

    // followed by additional data, actual payload
};

struct Sality_Payload_UrlPack   // version sanitized
{
    DWORD UrlPackId;
    DWORD FlagsUnknown;
    DWORD SizeFollowing;
    WORD UrlCount;      // of following

    // followed by list of URLs
};

struct Sality_Payload_PackExchange
{
    WORD Status;        // 0 = request, "OK" = response
    DWORD Status2;      // 0 = no newer peer list, 0FEFEFEFEh = newer peer list

    // only Sality v4: DWORD Empty; also 2048 bit certificate instead of 1024 bit

    // if newer peer list:
    BYTE Certificate[128];  // 1024 bit certificate

    Sality_Payload_UrlPack UrlPack;
};

struct Sality_Payload_AnnouncementPromotion
{
    DWORD PeerId;
    WORD ControlPort;
};

struct Sality_Payload_PeerExchange
```

```
{
    DWORD Ipv4;
    WORD Port;
    DWORD PeerId;
};

#pragma pack(pop)
```

FIG. 8. Header file for structure definitions of P2P packets.

Sality in its current version injects itself into a random process (with network 2 it was injecting itself into explorer.exe), but is easy to spot by scanning the RAM or simply observing connections with TCPView. In Figure 9 it was injecting itself into TCPView (sample MD5 B2FB74393D65E8CF91158D6DAAADC70A, SHA1 257E841963D52D2691D34AAE3E1EF7FCB95F4C99). Figure 10 shows actual P2P traffic as observed with Wireshark.

Tcpview.exe:...	UDP	0.0.0.0:8513	*:*	
Tcpview.exe:...	TCP	10.0.2.15:1074	192.185.116.203:80	CLOSE_WAIT
Tcpview.exe:...	TCP	10.0.2.15:1075	208.87.149.250:80	ESTABLISHED
Tcpview.exe:...	TCP	10.0.2.15:1078	184.107.58.100:80	CLOSE_WAIT
Tcpview.exe:...	TCP	10.0.2.15:1079	97.74.47.128:80	CLOSE_WAIT
Tcpview.exe:...	UDP	0.0.0.0:1285	*:*	
Tcpview.exe:...	UDP	0.0.0.0:1286	*:*	
Tcpview.exe:...	UDP	0.0.0.0:1287	*:*	
Tcpview.exe:...	UDP	0.0.0.0:1288	*:*	
Tcpview.exe:...	UDP	0.0.0.0:1289	*:*	
Tcpview.exe:...	UDP	0.0.0.0:1290	*:*	
Tcpview.exe:...	UDP	0.0.0.0:1291	*:*	
Tcpview.exe:...	UDP	0.0.0.0:1292	*:*	
Tcpview.exe:...	UDP	0.0.0.0:1293	*:*	
Tcpview.exe:...	UDP	0.0.0.0:1294	*:*	

FIG. 9. Sality injected code being active from within TCPView.exe process.

```
10.0.2.15   89.40.29.148      UDP    Source port: amx-icsp  Destination port: 6599
10.0.2.15   121.175.78.61     UDP    Source port: amx-axbnet  Destination port: 5817
10.0.2.15   210.182.247.240   UDP    Source port: pip  Destination port: 10647
10.0.2.15   112.144.153.58    UDP    Source port: novation  Destination port: 7374
10.0.2.15   78.97.239.70      UDP    Source port: brcd  Destination port: 6203
10.0.2.15   91.191.15.200     UDP    Source port: delta-mcp  Destination port: 6827
10.0.2.15   77.232.212.93     UDP    Source port: dx-instrument  Destination port: 7204
10.0.2.15   84.123.94.171     UDP    Source port: wimsic  Destination port: 7435
10.0.2.15   93.120.75.42      UDP    Source port: ultrex  Destination port: 9853
10.0.2.15   94.52.174.83      UDP    Source port: ewall  Destination port: 9674
10.0.2.15   82.229.4.35       UDP    Source port: netdb-export  Destination port: 5114
10.0.2.15   89.137.57.111     UDP    Source port: streetperfect  Destination port: 7167
```

FIG. 10. P2P traffic of Sality observed in Wireshark.

Currently in both network 3 and 4 combined there are 320.652 infections (vs 507.692 a year ago) as Table III shows. Interestingly, there are only 217 supernodes (infections that can be reached from the outside).

Network	***Inactive***	***Active***	***Supernode***	***Total***
#3	19	251.279	161	251.459
#4	218	68.919	56	69.193
Total	237	320.198	217	320.652

Table III. P2P statistics from 9/24/2015.

The exchange URL packs are lists of URLs pointing to binaries exchanged between peers. Each URL points to a binary which is a RC4 (in rounds) encrypted Windows executable. Those URLs are always compromised websites and the attacker always only upload the binary file (there is no panel or dynamic code behind it). Table IV shows the latest URL pack (as of writing this document) #241.

URL	*Status*	*IP*
http://maxsmotion.com/image/bottom.gif	Parked/expired	203.78.99.56
http://malkarafestival.com/images/bottom.gif	Active	89.19.30.10
http://kilinos.za.pl/bottom.gif	Parked/expired	193.203.99.115
http://www.kereninnegev.com/index.files/bottom.gif	Active	194.90.8.20
http://aruljayastone.com/images/logo.gif	Active	216.185.114.138
http://healthcenter.ueuo.com/images/bottom.gif	Active	10.6.139.112
http://gupiarainstrumentos.com/bottom.gif	Parked/expired	201.87.225.17
http://greatio.xp3.biz/images/logo.gif	Active	10.6.139.112
http://tunisie-plus.com/index_fichiers/bottom.gif	Active	91.216.107.150
http://www.sumanplastics.com/images/logo.gif	Ghosted	

Table IV. Latest URL pack 241.

The code to decrypt and encrypt these files is shown in Figure 11. Interestingly, the key, among many other things in the code was never changed by the developers.

```
char * Sality_C2_Key = "GdiPlus.dll";

for (unsigned n = 0; n < Sality_C2_Payload_Size; n += 1024)
    rc4((BYTE *)Sality_C2_Payload + n, 1024, (BYTE *)Sality_C2_Key,
strlen(Sality_C2_Key));
```

FIG. 11. Encryption/decryption code for binaries via URL packs.

Some URLs observed in URL packs from network 4 are plain invalid. Examples include “padrup.com.ds” (there is no .ds TLD) or “slwocfd”. The reason why the operators introduced such invalid URLs is unknown. A possible explanation is that the URL pack channel is not needed in network 4 because it has a binary exchange in the P2P botnet implemented.

There were other URLs observed that look interesting but are perfectly valid, for example “0-0-1-1-1-0-1-1-1-0-0-1-1-0-1-0-1-1-0-0-1-0-0-1-1-1-1-1-1-1-1-1-.0-0-0-0-0-0-0-0-0-0-0-0-0-49-0-0-0-0-0-0-0-0-0-0-0-0-0.info”.

Additional technical information about additional forensic footprints like mutexes, startup functionality and the rootkit is not presented here because it is already available through the references (see end of this document).

7. Remediation

As mentioned earlier, Sality is a highly detected virus even with its rootkit. Microsoft added removal of Sality to its Microsoft Malicious Software Removal Tool (MSRT) in 2012 [7]:

"*The second of the families added to the February release of the Microsoft Malicious Software Removal Tool (MSRT) is Win32/Pramro. Win32/Pramro is a family of trojans that can act as a SOCKS proxy on an infected computer. In this case, this proxy may be used to relay spam and HTTP traffic. Detection was first added for Pramro variants in January 2008.*

There is a strong connection with the polymorphic file infector Win32/Sality, which shares portions of code with Pramo."

MSRT gets distributed through Windows Update. Therefore, it is sufficient to enable Windows Update to remove Sality. Having Windows Update enabled and an up-to-date antivirus software installed is enough to protect against the known variants of Sality.

On a global level, the only update channels are be P2P botnet and the URL pack based executable installation channel. Sending commands via P2P (updates of the URL pack or the executable itself) is not feasible because it requires the RSA key to sign those; one would have to break an RSA 1024 key for network 3 and RSA 2048 key for network 4.

However, the executables installed via the URL packs are not signed. If someone obtains access to the URLs (web-servers/websites) that are used in the various URL packs someone could potentially use this channel to install a dedicated Sality remover. It is important to point out however, that those URLs used are always compromised (legitimate) websites.

Kleissner & Associates does not advise to launch disruption attacks against P2P botnets as it would be just a temporary "solution" instead of a long term one.

8. Final Notes

Some of above presented information and threat intelligence was shared appropriately within the security and intelligence community.

REFERENCES

[1] Symantec, Sality, http://www.symantec.com/connect/blogs/all-one-malware-overview-sality, http://www.symantec.com/connect/blogs/sality-botnet, http://www.symantec.com/content/en/us/enterprise/media/security_response/whitepapers/sality_peer_to_peer_viral_network.pdf, http://www.symantec.com/security_response/writeup.jsp?docid=2006-011714-3948-99

[2] SoK: P2PWNED, http://www.christian-rossow.de/publications/p2pwned-ieee2013.pdf

[3] Artem Baranov, Sality Rootkit Analysis, http://artemonsecurity.blogspot.cz/2013/01/sality-rootkit-analysis.html

[4] Total Defense, Sality gets upgrade http://www.totaldefense.com/security-blog/sality-gets-upgrade

[5] Sophos, Sality analysis https://www.sophos.com/en-us/threat-center/threat-analyses/viruses-and-spyware/W32~Sality-H/detailed-analysis.aspx

[6] ESET, Sality.NAB https://www.eset.hu/virus/sality-nab

[7] Microsoft, MSRT 21 Feb 2012, Pramro and Sality - two PEs in a pod http://blogs.technet.com/b/mmpc/archive/2012/02/21/pramro-and-sality-two-pes-in-a-pod.aspx