Heimdal®

2020

# POLYMORPHIC MALWARE SPAWNING PSEUDO-OPERATING SYSTEM PROCESSES.

Circumventing behavioral-based and TTPC detection grids.

**Author**

Vladimir-Alexandru Unterfingher

WWW.HEIMDALSECURITY.COM

**Author**
Vladimir-Alexandru Unterfingher

# TABLE OF CONTENTS

# 1.

# INTRODUCTION

# 1. Introduction

Heimdal's cybercrime research unit has recently uncovered a criminal infrastructure that employs multiple domains in order to release malware into the wild. Despite the domains being taken offline, per request, the malicious software distributed through them appears to elude any known simple behavioral-based detection methodology. This type of malicious activity, which has yet to be classified, focuses on machines running on multi-core architecture.

We have ascertained that a new variety of malware may have been released into the wild. More worrisome is the fact that controlled, sandboxing-type tests performed on the sample have yielded some interesting results. The infection appears to be targeting multi-core machines and has so far evaded most behavioral and some simple threat detection tools.

On their own, these continuously-spawned processes have the same properties as regular OS processes, and will therefore not maliciously impact the system. However, the ensemble, which is triggered by an exterior compiler, can be just as viral and damaging as single-process malware.

Keywords: polymorphic malware, multi-process malware, mirrored process, behavior-based detection, system process, heuristic analysis, threat-to-process correlation.

# 2.

# SIGNATURE- VS. BEHAVIOR-BASED

DETECTION IN MALWARE ANALYSIS

# 2. Signature- vs. behavior-based
Detection in malware analysis

## 2.1. Signature-based Detection

Critical to the malware removal process is the detection methodology. The available literature on the topic defines two malware-detection approaches: signature- and behavior-based.

In signature-based detection, an object can be label 'malicious' if its signature matches one found in the antivirus database. This is the most common type of detection approach. However, with the advent of second-generation malware, signature-based antivirus software is becoming less efficient. Malware creators implement simple (or complex) obfuscation techniques to increase the likelihood of success, regardless of intent (i.e. wanton destruction, data exfiltration, hacktivism, etc.).

Obfuscation techniques examples:

- 'Crypters'
- 'Packers'
- Instruction changers
- Dead-code insertion
- Rotate 13 (ROT13)
- Base64 Encoding
- XOR

'Crypters' are used to abscond the original, malicious code. This is achieved through a cryptographical algorithm (symmetric or asymmetric). In the case of packers, the malicious code is compressed in a .zip, .rar, .tar archive. This serves two purposes: reduces the sized of the malicious code and provides an alternate means of execution – auto-unpacking. By default, archives don't have the auto-unpack feature enabled. Instruction changers rely on minute code alterations in order to pass the malicious code as safe. Besides, instruction changers also alter the appearance of the code, although, in essence, it remains unchanged. Dead-code insertion, as the name suggests, involves inserting pieces of garbage code into the original (and malicious code) in an attempt to abscond its purpose. ROT13, short for Rotation 13, is a simple cryptographical algorithm based on letter permutation (first later substituted by the 13th letter that comes after it); ROT13-encrypted codes can be de-obfuscated by applying the algorithm twice). Base64 encoding (i.e. binary data to ASCII string encoding) is another way of obfuscating malicious code. Trivial, at best, since base64 encoding can be easily recognized, and, therefore reverse-engineered, it still confuses most signature-based detection engines.

## Base64 encoding ‘algorithm’

In the Base64 encoding, there are 64 characters (upper- and lowercase letters of the alphabet, numbers, operators such as the “+” and “- “, and the “=” sign, which is used for padding). The algorithm is as follows: three characters are concatenated as to create a 24-bit string. After that, the string is broken down into four blocks, each containing six bits. The resulting string is further encoded using base64 characters.

XOR, standing for Exclusive OR, is yet another common malicious code obfuscation technique. Exclusive OR is mostly used to generate parity bits and to test for errors. This is achieved via input bit comparison. The result is one output bit that follows this logic: if the compared bits match, the result is 0, else the result is 1. Despite being considered a less refined obfuscation method as compared to dead-code insertion or packing, it can be improved upon (i.e. adding more complexity).

Signature-based detection implies or, rather, involves the existence of a curated virus signature database. Such a knowledge base can be difficult to maintain. Another disadvantage is the method’s inability to detect polymorphic or metamorphic malware. For example purposes, we have included three virus signatures – Accom.128, Die.448, Xany.979. In the figure below, you will find a simplistic representation of an antivirus’ agent database. The examples provided are from the paper “Evolution of Computer Virus Concealment and Anti-Virus Techniques: A Short Survey”.

| Virus Name | String Patern |
|---|---|
| Accom.128 | 89C3 B440 8A2E 2004 8A0E 2104 BA00 05CD 21E8 D50 BF50 04CD |
| Die.448 | B440 B9E8 0133 D2CD 2172 1126 8955 15B4 40B9 050 BA5A 01CDv |
| Xany.979 | 8B96 0906 B0 E85C FF8B D5B9 D303 E864 FFC6 8602 0401 F8C3 |

Figure 1. Antivirus database entries

In signature-based detection, the assumption is that malware follows a certain pattern, without minimal deviations. As a result, signature-based detection becomes inefficient in identifying zero-day attacks or malware that employ advanced evasion techniques. As a possible workaround, blacklisting and whitelisting were introduced. In essence, the two techniques were meant to complement or, at the very least, add some degree of padding to counter the deficiencies of this detection method. Whitelisted software can be safely installed on the targeted machine; no scanning or AV inspection is required. Blacklisting is the opposite of whitelisting. The software found on the blacklist cannot be installed on the targeted machine if this rule is admin-enforced. In practice, working with black and whitelist can be difficult because even whitelisted processes or software can exhibit malicious behavior (i.e. browsers can sometimes harbor malicious content).

## 2.2. Behavior-based detection

In behavior-based detection, the main assumption is that malware can follow behavioral patterns. Furthermore, this method proposes a new type of approach in malware analysis and identification – executing the potentially malicious software in a controlled and insulated environment in order to determine how it interacts with systems process. Instead of comparing the potentially malicious code or software to a pre-defined list – as in the case of signature-based detection – the behavioral approach involves the close and almost clinical observation of executables. The results of this impromptu investigations are compared to known malicious behavioral patterns and cross-referenced with normal software behavior. An important milestone in behavior-based detection is anomalous behavior profiling.

In “Analysis Signature-based and behavior-based anti-malware approaches”, the authors compared anomaly detection to the spending profiles banks use in order to identify credit card frauds. In the case of software, the behavior-based engine can flag certain types of system activities as anomalous if it does not conform to normal, software ops (i.e. the software attempts to overwrite Windows registries or gain access to a sensitive directory that would otherwise be unavailable).

## Illustrating malware behavioral profiling

| Malware* | Behavior type | Description |
|---|---|---|
| Farseer, KerrDown, Ocean Lotus | Hijacking the DLL load order | Threat actors can modify the DLL bootstrapping order of an executable. Hardcoding a DLL path can solve the issue. Malicious code can be inserted in the search order. Threat actors can also substitute DLLs or change the search order. |
| Rootkits, COMpfun, COVID-19 | System binaries with trojan-like behavior | Threat actors compromise frequently used system binaries to redirect to malicious code. |
| Svchost.exe *32 Trojan Miner, Smurf Trojan, Carberp, DorkBot | Svchost for persistence | Svchost.exe is a generic host process name for numerous services that rely on DLLs for execution. Since it's a shared service process operator, it can be used by threat actors for malware persistence. |
| APT19, TROJ_POWELIKS.A | Windows registry infiltration | Infiltration and persistence can be achieved by modifying the Windows registry. |
| APT32, Kovter, Dridex | APC injection | Asynchronous Procedure Call injection involves appending a piece of malicious code to the corresponding APC queues, inside a system process. |
| Trickbot, Zberp, Zeus | Hook injection | Inject and run malicious code in another software environment.  Can also be employed to intercept API calls. |
| Trojan: W32/Injector, Smokeloader | DLL injection | Also called direct injection. Hacker rewrites the DLL path inside the app's process. Remote access used to trigger execution. |

| | | |
|---|---|---|
| W32.W.Stuxnet.ad! c, Worm/ Win32.Stuxnet. N495400904, Worm.Win32. Stuxnet | Process injection | Common processes are injected with malicious codes. The compromised processes retain the same appearance as the original ones. |
| SpyEye, Ice IX, Shylock | Credentials exfiltration | Threat actors will exfiltrate credentials (i.e. passwords, usernames) using various transfer methods such as HTTP or email. Keystrokers can also be employed for stealing passwords and /or usernames. |
| Emotet, Zebrocy, FlawedGrave | Backdoors and downloaders | Upon gaining access to the machine, threat actors may install downloaders to maintain control over the infected device. |

*malware strains that exhibit the behavioral patterns described in the middle column.

# 3.

# ‘MULTI-PROCESS MALWARE’

TERMINOLOGY, CLASSIFICATION, DISPERSAL PATTERN, AND ‘TAINTING’ MECHANISM

# 3. ‘Multi-process malware’
Terminology, Classification, Dispersal Pattern, and ‘Tainting’ mechanism

## 3.1. Proposed terminology

In analyzing the dispersal and tainting mechanism of multi-process malware, we propose the following terminology:

1. Polymorphism. Considering that multi-process malware can change its form in order to evade detection, it would be only fair to include it in this larger category.

2. System call dependency. In existing malware detection methodology, a process can be flagged down as either “malicious” or “safe” based on the way it is interacting with various OS functions. Sandboxed malware samples perform very specific system call sequences and/or graphs.

3. Threat-to-Process Correlation (TTPC). A new method in behavioral analysis that aims to establish a connection between the malware and the process it is attempting to access on the infected machine.

4. Malware “Download-Execution” algorithm. In a typical malware propagation schema, the malicious process beings by queuing two types of operations: “recv” (instructs FPT that a file transfer will ensue) and “open”. In turn, this will trigger a “write” operations and finally, the malicious package executes itself on the target machine.

5. Multi-process malware ‘internal’ C&C (coordination and communication). To coordinate malicious code dissemination, all ‘mirrored’ processes communicate between them. Furthermore, an ‘overseer’ type of entity is warranted for malicious package recompilation.

## 3.2. Example of possible multi-process malware execution in a controlled environment

On executing the malware sample in a sandbox environment, we have observed the following behavior.

a. D&E (download & execute) behavior

File-manipulation operations are commencing – a “recv” command is issued at the same time as a “write” operation. Concomitantly, the targeted processes’ kernels are opened before the “write” command is executed.

This primary dyad is processed & executed along with a secondary one: “recv” before “write” and “write” before “execute”. The entire sequence allows the multi-process malware sample to tamper with a key system call sequence that would ultimately allow it to commit modifications to sensitive registry entries.

b. Proxy behavior

On the proxy side, the malware begins by piping data from a socket to a buffer. After that, it will simply store the data from a compromised comm socket into the same buffer.

c. Registry modification

As far as registry modification is concerned, there are two possible outcomes:

- The malware can execute a registry key creation command before deleting the value of a preexisting registry key.
- The malware can execute a registry-creation key command before closing the value-modification process for that key.

This is how the malware sample behaved in a sandbox-type environment. Further research has revealed that once the multi-process strain is released into the wild, it employs rogue domains in order to call up various system processes, which further reinforces the hypothesis of a criminal infrastructure hiding behind malicious domains.

# 4.

# MODUS OPERANDI, HITS, AND PROCESS CALLS

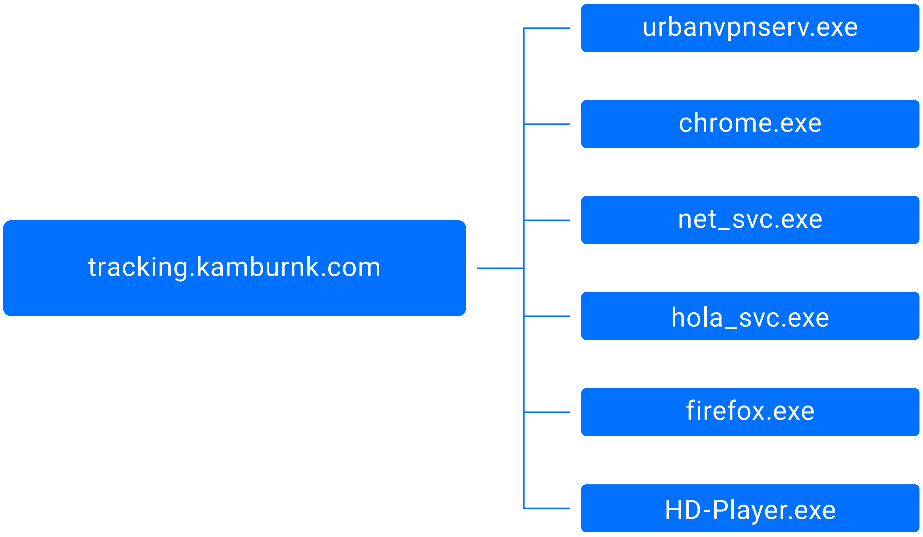
# 4. Modus Operandi, hits, and process calls.

Heimdal's internal data has yielded that a single domain can call up to 35 different system processes. Here are the highlights of Heimdal's investigation.

a. A malicious domain named "tracking.kamburnk.com" (domain blocked and sanitized by Heimdal) called up urbanvpnserv.exe no less than 2,400 times.

> The process in question is a PE32 executable (GUI) Intel 80386 for Microsoft Windows, developed by Urban Security. It has been designed to install a VPN service called UrbanVPN.

The malicious pattern was later established after the executable attempted to drop a viral payload or rewrite itself. The same domain called up various other system processes such as chrome.exe, net_svc.exe, hola_svc.exe, firefox.exe, and HD-Player.exe.

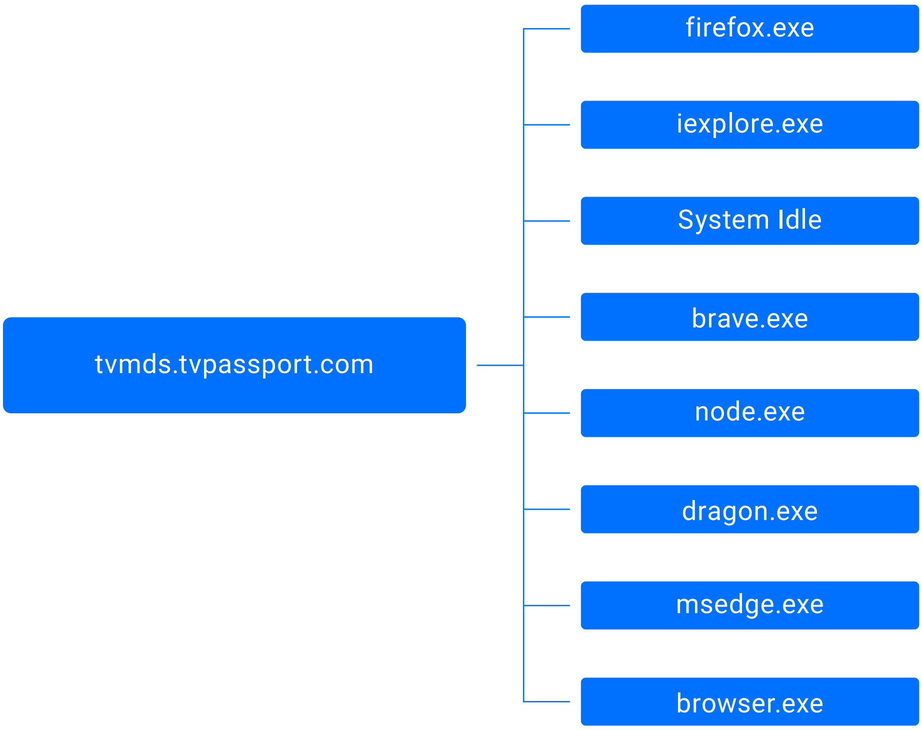


Regular behavioral-based analysis returned inconclusive results. No attack pattern could have been established. TTPC analysis indicated a high probability of malicious activity based on system processes opened concurrently.

Multi-process malware attack pattern established, signaling the agent to sever communications to C&C servers.

b. A malicious domain, called “tvmds.tvpassport.com” (domain blocked and sanitized by Heimdal), attempts to open chrome.exe.

Hit count has been estimated at around 270. The same domain called up additional system processes. Sorted by hit count, the processes are firefox.exe, iexplore.exe, System Idle, brave.exe, node.exe, dragon.exe, msedge.exe, and browser.exe.



Applied behavioral-based analysis has rendered inconclusive results. No attack pattern could have been established.

TTPC detects a high probability of malicious activity, based on the holistic analysis performed on the accessed processes’ log trail. Multi-process malware attack pattern established. C&C connection severed.

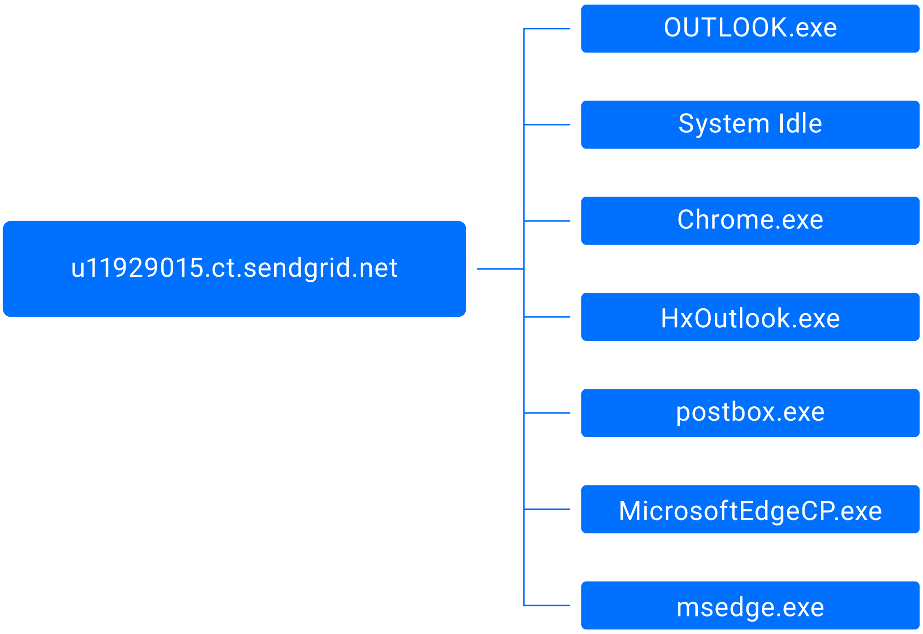
c. A malicious domain, call sign “u11929015.ct.sendgrid.net” (domain blocked and sanitized by Heimdal) attempts to open OUTLOOK.exe.

Hit count estimated at around 35. Concomitantly, the same domain attempts to open other system processes.

Sorted by hit count, the processes are as follows:

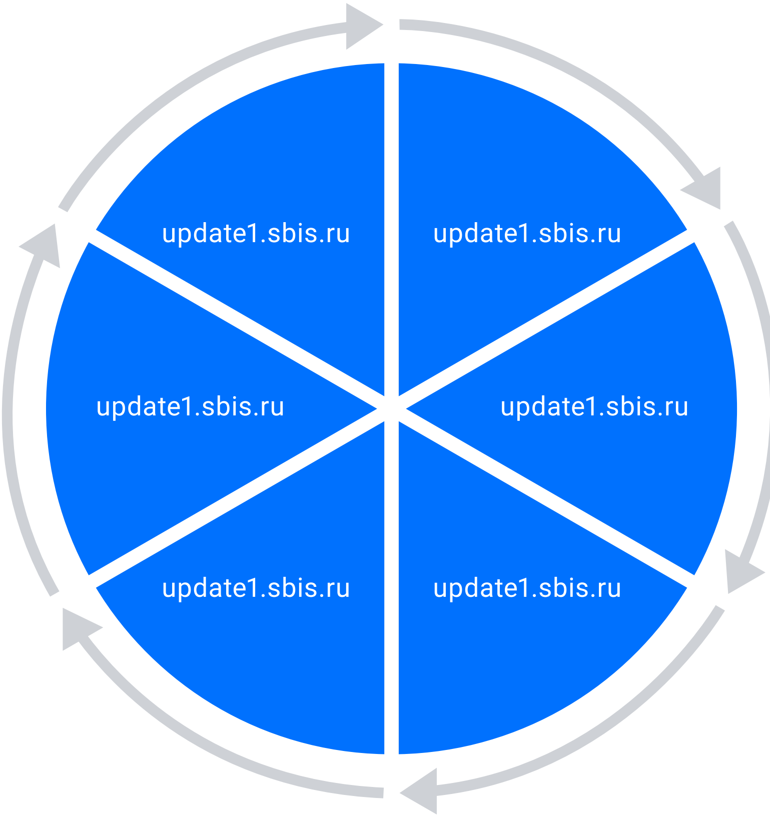
1. System Idle.
2. Chrome.exe.
3. HxOutlook.exe.
4. Postbox.exe.
5. MicrosoftEdgeCP.exe.
6. Msedge.exe.

⚠ HxOutlook.exe is a 32-bit executable that serves various Outlook functions such as task managing, note-making, calendar, and journaling.



Behavioral analysis performed on the sample marked the connection as safe. Subsequent TTPC analysis established multi-process malware-type behavior. The connection to C&C servers was severed.

d. A malicious domain called “update1.sbis.ru” (domain blocked and sanitized by Heimdal) attempts to open several system processes, including svchost.exe.



Low hit count on the first pass – one time for svchost.exe and around 15 times for sbis3plugin.exe. Polymorphic pathology established. Domain mutates into “update5.sbis.ru” (blocked & sanitized by Heimdal).

On the second pass, the hit count and the accessed processes are as follows: 14 registered attempts for sbi3plugin.exe and one attempt for svchost.exe.

Second mutation – attack launched from “update6.saby.ru” (blocked & sanitized by Heimdal). Remark he subdomain permutation (. sbis   .saby); same two processes are targeted (svchost.exe and sbis3plugin.exe).

Hit count grows exponentially – around 28 times for the Service Process Host and once for the sbis3 plugin.

Upon establishing beachhead, the subdomain is again changed (. saby   .sbis). From “updade6.sbis.ru”, svchost and sbi3plugin are again called up. Hit count reaches 1,400 for the sbi plugin and around 800 for the Service Process Host.

Other processes are called up during the second wave: SbisMon.exe and Launcher.exe.

The fourth pass – subdomain and form are changed again.

The new call sign is “update7.saby.ru” (blocked & sanitized by Heimdal). Subdomain targeting attempts to access the sbi3 plugin, with a hit count of 35.

Once the foray has seized, the subdomain is changed once more. saby to. sbis. “update7.sbis.ru” (blocked and sanitized by Heimdal) aims for sbi3plugin.exe, svchost.exe, and other processes.

A pattern variation can be observed: the attack launched upon subdomain shift would have commenced with a different system process. This time, the attack starts by accessing sbis3, as opposed to the last round, before moving on to svc host and other system processes.

Hit count is over 1,000 of sbi3, followed by 856 for System Idle, and 43 for Service Process Host.

Fifth pass (“update8.saby.ru” and “update8.sbis.ru”, both blocked & sanitized by Heimdal) is symmetrical to the last: sbi3 is the round-opener; subdomain switch (saby   sbis).

Hit count for the sbi3 plugin is 37 on “. saby” pass, and over 1,000 from the. sbis subdomain. Other ‘tainted’ system processes: SbisMon.exe, sbis.exe, chrome.exe.

Behavioral analysis has yielded inconclusive results. TTPC indicated a high probability of malware activity.

Given the attack pattern and the number of domains and subdomains used to deploy the malware, it stands to reason that a criminal infrastructure may be at work.

Polymorphic and multi-process malware pathologies confirmed. Connections to C&C servers had been severed.

# 5.

# STATISTICAL ANALYSIS

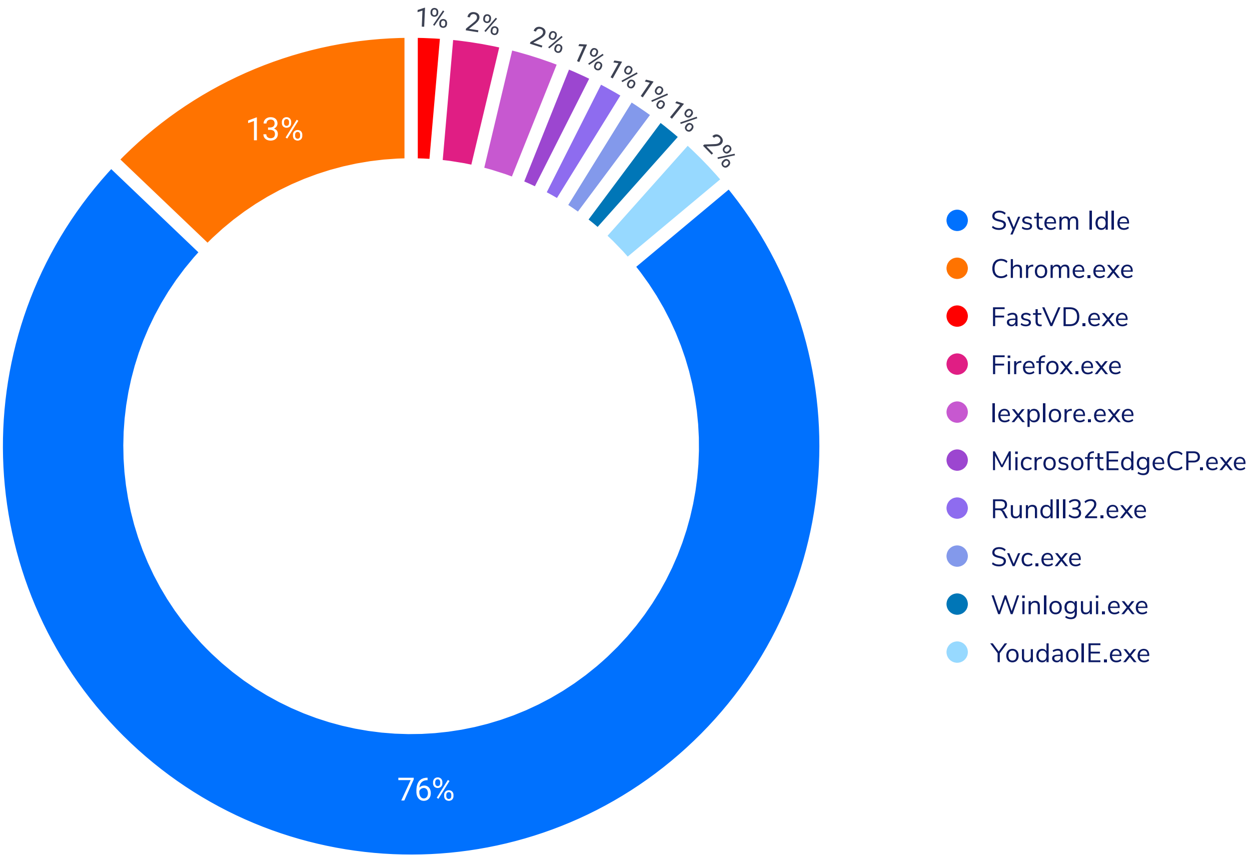
# 5. Statistical analysis

A total of 3,148,815 attacks have been observed in the last three months. They are as follows:

- 13,24% of attacks originated from a domain called “pixel.yabidos.com” (blocked & sanitized by Heimdal).
- In 62.39% of cases, the attackers targeted the System Idle Process. Chrome.exe – 10.16%, firefox.exe – 1.57%, and iexplore.exe -1.49%.
- 1,665 system processes have been targeted within the last three months.
- Total hit count for svc.exe and svchost was 28,185. In only 3,27% of cases, the attackers attempted to tamper with svchost.

## Top 10 ‘brutalized’ processes based on hit count

| System Process | Hit Count |
|---|---|
| System Idle | 1,964,589 |
| Chrome.exe | 320,069 |
| FastVD.exe | 28,069 |
| Firefox.exe | 49,300 |
| Iexplore.exe | 46,916 |
| MicrosoftEdgeCP.exe | 31,519 |
| Rundll32.exe | 35,350 |
| Svc.exe | 27,263 |
| Winlogui.exe | 30,262 |
| YoudaoIE.exe | 44,418 |


1%
2%
2%
1%
1%
1%
1%
2%
13%
76%
System Idle
Chrome.exe
FastVD.exe
Firefox.exe
Iexplore.exe
MicrosoftEdgeCP.exe
Rundll32.exe
Svc.exe
Winlogui.exe
YoudaoIE.exe

# 6.

# CONCLUSIONS

# 6. Conclusions

Considering the factual data, we can safely conclude that a criminal infrastructure is using multiple domains to coordinate malicious attacks. In all instances, behavioral-, code-, and signature-based would have returned conflicting results. None of the access attempts appear to be malicious in nature. However, once intercommunication is established, the malware would have executed a multi-pronged attack that would have more than likely cripple key processes.

The discovery of this APT has been facilitated by correlating the data from Thor Foresight Enterprise's TTPC backlog with the number of connections severed by DarkLayerGuard, Heimdal's DNS traffic-filtering solution.

Is multi-process malware the next stage in APT's evolutionary process? Most definitely, considering that it's far easier to deploy compared to single-process malware and has a higher stealth factor.

# 7.

# RELATED WORK

# 7. Related Work

In “Shadow Attacks: Automatically Evading System-Call-Behavior Based Malware Detection”, the authors describe a new class of cybernetic attacks called “Shadow Attack(s)”. As noted in the paper, shadow attacks have partitioning and system-call exportation capabilities. For obfuscation purposes, this novel class of malware can generate shadow processes, which are described as ‘bits’ of malicious code that lay dormant inside the infected machine in the absence of an external compiler. To maintain access over the infected device or network and to compile the malware, the attack will employ an inter-process, holistic, communication schema called S.P.C (Shadow Process Communication). Local chatter between shadows processes is obfuscated through RMC (Remote Network Coordination), which is achieved via external “stepping nodes”.

Detecting type of covert communication is onerous, partly due to the fact that behavioral-based detection engines cannot establish a relation between the individual components. The same conclusion is corroborated by Ramili, Bishop, and Sun, who noted that multi-process malware’s increased stealth factor can be attributed to the dissolution of spatial and temporal relationships, as observed in the case of system events. In behavioral-based detection, the antiviral agent will have had examined the temporal relationship of system events to determine if there is a pattern.

Malware attempting to call the C&C server for instructions will attempt to force-dial the infected process. Behavioral-based agents may label this activity as malicious if the number of requests exceeds a predetermined baseline value. Spatiality refers to the order of process occurrence; should the order be broken, the behavioral-based engine will become incapable of recognizing the signature associated with that process tree.

# 8.

## GLOSSARY

# 8. Glossary

## 8.1. Terms

TTPC = Threat-to-Process correlation acronym. Behavioral-based detection method proposed by Heimdal. The malicious agent is matched with the system process being called.

Polymorphic malware= a type of malware capable of changing its form and identifiable feature to circumvent detection systems.

APT = Advanced Persistent Threat acronym; refers to a state-funded threat actor that gains access to networks or endpoints for espionage or data exfiltration purposes.

DNS filtering= technique used to either block and restrict the user’s access to specific websites, domains, or IP addresses.

C&C server= Command and Control server acronym. Refers to a server controlled and maintained by a threat actor and used to send commands to endpoints or networks compromised by infections.

C&C server= Command and Control server acronym. Refers to a server controlled and maintained by a threat actor and used to send commands to endpoints or networks compromised by infections.

Multi-pronged (cyber) attack = cybernetic attack employing multiple attack vectors.

Sandboxing = malware analysis technique involving potentially malicious code execution in a controlled (and isolated) environment.

## 8.2. Malware references

| Malware | Short description |
|---|---|
| APT19 | Chinese threat group that has targeted major industries with phishing emails and spearphishing. |
| APT32 (Ocean Lotus) | Vietnamese cyber-espionage group collecting COVID-19 intel from Chinese targets. |
| Carberp | The Banking Trojan family first observed in 2009. Can steal credentials through monitoring user activity or hooking up network APIs. |
| COVID-19 | Ransomware. Has been observed to target universities, hospitals, and governmental institutions. |
| DorkBot | Worms family. Has been observed to spread through social media, thumb drives, IM, and certain websites. Can download and run malicious fires from a given URL, grab credentials, redirect or block secure websites or domains. |
| Dridex | Banking trojan disseminated through Word, Excel, or email attachments. |
| Emotet | Rolled out as a banking trojan in 2014. Later turned into a loader. Botnet creator. Can infect machines through email attachments or macro viruses. |
| Farseer | Command & Control capabilities. Affiliated with HenBox Android. Cyber-espionage. Uses DLL sideloading and VBS scrips to infect machines. |
| FlawedGrace | RAT (remote access tool). Observed in 2017. |

| | |
|---|---|
| Ice IX | Banking trojan from the Zeus family. Used to steal financial and personal info. |
| KerrDown | Trojan. Downloader. Mostly employed by OceanLotus. |
| Kovter | Ransomware turned filess malware. Begun as police ransomware. Some observed KOVTER strains behaved like click-fraud malware. |
| Shylock | Banking trojan. First observed in early 2011. Employs MiB attacks to steal login credentials. |
| Smokeloader | Customizable malware. Sold in MaaS schemas. Can accept multiple viral payloads Malware downloaded and installed on the victim's machine through Smoke Loader. |
| Smurf Trojan (Troj/Smurf-A) | A DDoS attack that exploits IP and ICMP vulnerabilities. |
| SpyEye | Keystroke logger and form grabber. Affect MS Windows and frequently used browsers such as Firefox, IE, Chrome, and Opera. |
| Svchost.exe *32 Trojan Miner | Cryptocurrency miner. Utilizes a vast amount of CPU and GPU power. |
| Trickbot | Modular banking trojan. Uses MiB attacks to steal financial information. Utilized as a dropper for other types of malware. |
| TROJ_POWELIKS.A | Helps threat actors gather sensitive system information. Acts as a dropper for other types of malware. |
| Trojan: W32/Injector | Process injecting trojan. Injects malicious code into the running process. Can act as a dropped. |
| W32.W.Stuxnet.ad! | Computer worm from the Stuxnet family. First observed in 2010. Targets PLCs (Programmable Logic Controllers). Utilized to gain authorized access to industrial and control machinery. |

# 9.

# REFERENCES

# 9. References

1. Ramili, M, Bishop M., Shining S., Mutiprocess Malware;

2. Mujumdar A., Masiwal G., Dr. Meshram B.B., Analysis of Signature-Based and Behavior-Based Anti-Malware Approaches;

3. Bidoki S. M., Jalil S., Tajoddin A., PbMMD: A novel policy based multi-process malware detection;

4. Fan L., Wang Y., Cheng X., Li J., Jin S., Privacy theft malware multi-process collaboration analysis

Heimdal®

Leading the fight against cybercrime.

www.heimdalsecurity.com

Vat No. 35802495, Vester
Farimagsgade 1, 2 Sal, 1606 København V