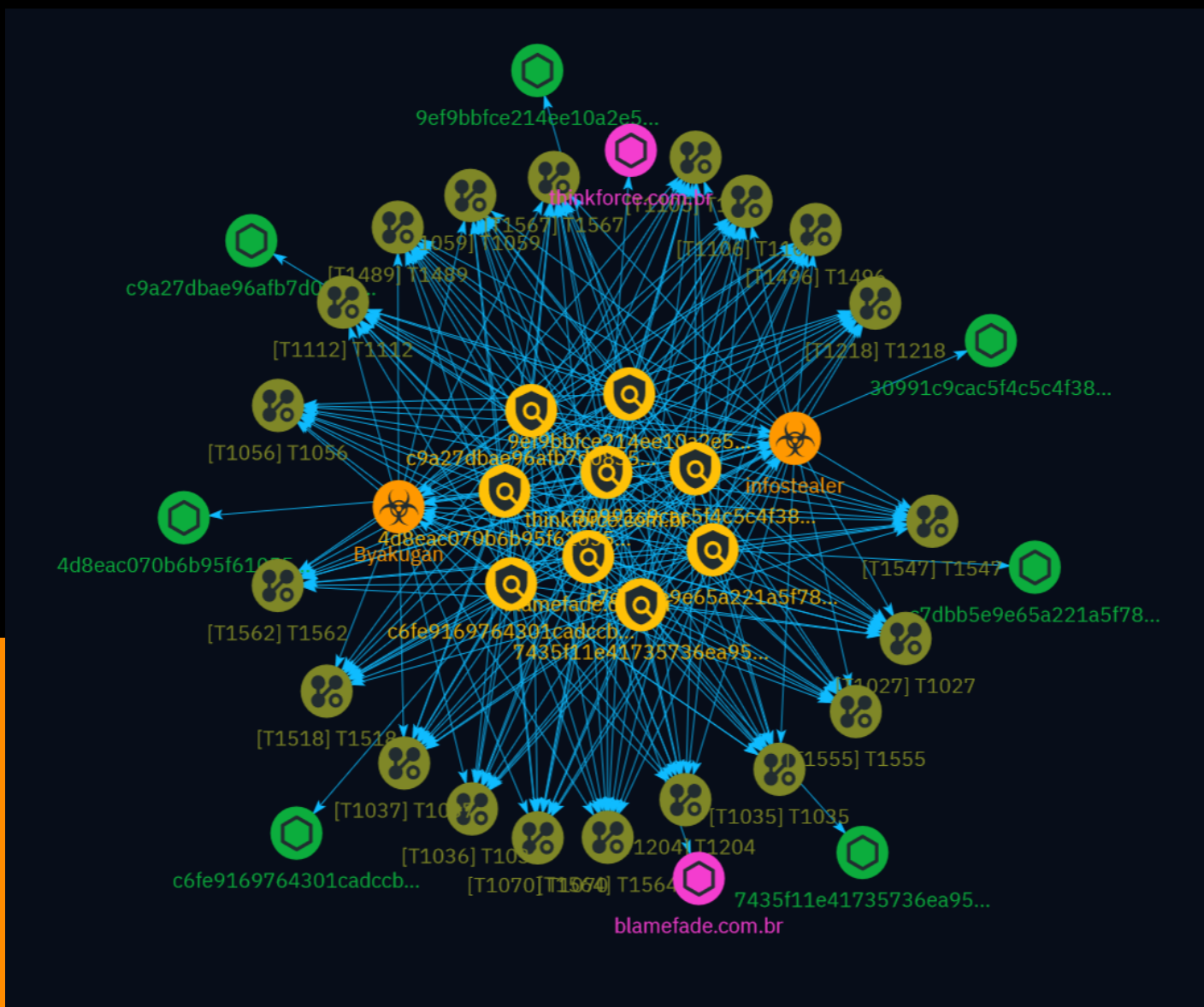


# NETMANAGEIT

## Intelligence Report

# Byakugan - The Malware Behind a Phishing Attack



# Table of contents

---

## Overview

---

● Description	4
● Confidence	4
● Content	5

---

## Entities

---

● Indicator	6
● Malware	10
● Attack-Pattern	11

---

## Observables

---

● Domain-Name	24
● StixFile	25



## External References

- External References

26

# Overview

## Description

The report analyzes Byakugan, a multi-functional malware distributed through a phishing campaign using malicious PDF files. It details the infection vector, command-and-control infrastructure, and various capabilities like screen monitoring, mining, keylogging, and data exfiltration. The malware employs evasion techniques, persistence mechanisms, and utilizes legitimate tools like OBS Studio. Fortinet's protection measures against this threat are also provided.

## Confidence

*This value represents the confidence in the correctness of the data contained within this report.*

100 / 100

# Content

N/A

# Indicator

## Name

thinkforce.com.br

## Description

- **Unsafe:** True - **Server:** N/A - **Domain Rank:** 0 - **DNS Valid:** True - **Parking:** False - **Spamming:** False - **Malware:** True - **Phishing:** True - **Suspicious:** True - **Adult:** False - **Category:** Web Tracker - **Domain Age:** {'human': '2 years ago', 'timestamp': 1651584054, 'iso': '2022-05-03T09:20:54-04:00'} - **IPQS: Domain:** thinkforce.com.br - **IPQS: IP Address:** 209.145.55.141

## Pattern Type

stix

## Pattern

[domain-name:value = 'thinkforce.com.br']

## Name

blamefade.com.br

## Description

- **Unsafe:** True - **Server:** N/A - **Domain Rank:** 0 - **DNS Valid:** True - **Parking:** False - **Spamming:** False - **Malware:** True - **Phishing:** True - **Suspicious:** True - **Adult:** False - **Category:** Web Tracker - **Domain Age:**

{'human': '3 months ago', 'timestamp': 1704903173, 'iso': '2024-01-10T11:12:53-05:00'} - \*\*IPQS: Domain:\*\* blamefade.com.br - \*\*IPQS: IP Address:\*\* 207.244.251.87

**Pattern Type**

stix

**Pattern**

[domain-name:value = 'blamefade.com.br']

**Name**

c9a27dbae96afb7d083577d30b2947c8ba9d1a6cb7e10e5f259f0929ef107882

**Pattern Type**

stix

**Pattern**

[file:hashes:'SHA-256' =  
'c9a27dbae96afb7d083577d30b2947c8ba9d1a6cb7e10e5f259f0929ef107882']

**Name**

c7dbb5e9e65a221a5f78328b5a6141dd46a0459b88248e84de345b2a6e52b1d9

**Pattern Type**

stix

**Pattern**

[file:hashes:'SHA-256' =  
'c7dbb5e9e65a221a5f78328b5a6141dd46a0459b88248e84de345b2a6e52b1d9']

**Name**

c6fe9169764301cadccb252fbed218a1a997922f0df31d3e813b4fe2a3e6326d

**Pattern Type**

stix

**Pattern**

[file:hashes:'SHA-256' =  
'c6fe9169764301cadccb252fbed218a1a997922f0df31d3e813b4fe2a3e6326d']

**Name**

9ef9bbfce214ee10a2e563e56fb6486161c2a623cd91bb5be055f5745edd6479

**Pattern Type**

stix

**Pattern**

[file:hashes:'SHA-256' =  
'9ef9bbfce214ee10a2e563e56fb6486161c2a623cd91bb5be055f5745edd6479']

**Name**

7435f11e41735736ea95e0c8a66e15014ee238c3a746c0f5b3d4faf4d05215af

**Pattern Type**

stix

**Pattern**



[file:hashes!'SHA-256' =  
'7435f11e41735736ea95e0c8a66e15014ee238c3a746c0f5b3d4faf4d05215af']

**Name**

4d8eac070b6b95f61055b96fb6567a477dbc335ef163c10514c864d9913d23cb

**Pattern Type**

stix

**Pattern**

[file:hashes!'SHA-256' =  
'4d8eac070b6b95f61055b96fb6567a477dbc335ef163c10514c864d9913d23cb']

**Name**

30991c9cac5f4c5c4f382f89055c3b5e9bb373c98ce6a5516d06db3f8a478554

**Pattern Type**

stix

**Pattern**

[file:hashes!'SHA-256' =  
'30991c9cac5f4c5c4f382f89055c3b5e9bb373c98ce6a5516d06db3f8a478554']

# Malware

## Name

Byakugan

## Name

infostealer

# Attack-Pattern

**Name**

T1035

**ID**

T1035

**Name**

T1056

**ID**

T1056

**Description**

Adversaries may use methods of capturing user input to obtain credentials or collect information. During normal system usage, users often provide credentials to various different locations, such as login pages/portals or system dialog boxes. Input capture mechanisms may be transparent to the user (e.g. [Credential API Hooking](https://attack.mitre.org/techniques/T1056/004)) or rely on deceiving the user into providing input into what they believe to be a genuine service (e.g. [Web Portal Capture](https://attack.mitre.org/techniques/T1056/003)).

**Name**

T1496

**ID**

T1496

**Description**

Adversaries may leverage the resources of co-opted systems to complete resource-intensive tasks, which may impact system and/or hosted service availability. One common purpose for Resource Hijacking is to validate transactions of cryptocurrency networks and earn virtual currency. Adversaries may consume enough system resources to negatively impact and/or cause affected machines to become unresponsive.(Citation: Kaspersky Lazarus Under The Hood Blog 2017) Servers and cloud-based systems are common targets because of the high potential for available resources, but user endpoint systems may also be compromised and used for Resource Hijacking and cryptocurrency mining.(Citation: CloudSploit - Unused AWS Regions) Containerized environments may also be targeted due to the ease of deployment via exposed APIs and the potential for scaling mining activities by deploying or compromising multiple containers within an environment or cluster. (Citation: Unit 42 Hildegard Malware)(Citation: Trend Micro Exposed Docker APIs) Additionally, some cryptocurrency mining malware identify then kill off processes for competing malware to ensure it's not competing for resources.(Citation: Trend Micro War of Crypto Miners) Adversaries may also use malware that leverages a system's network bandwidth as part of a botnet in order to facilitate [Network Denial of Service](<https://attack.mitre.org/techniques/T1498>) campaigns and/or to seed malicious torrents.(Citation: GoBotKR) Alternatively, they may engage in proxyjacking by selling use of the victims' network bandwidth and IP address to proxyware services.(Citation: Sysdig Proxyjacking)

**Name**

T1070

**ID**

T1070

**Description**

Adversaries may delete or modify artifacts generated within systems to remove evidence of their presence or hinder defenses. Various artifacts may be created by an adversary or something that can be attributed to an adversary's actions. Typically these artifacts are used as defensive indicators related to monitored events, such as strings from downloaded files, logs that are generated from user actions, and other data analyzed by defenders. Location, format, and type of artifact (such as command or login history) are often specific to each platform. Removal of these indicators may interfere with event collection, reporting, or other processes used to detect intrusion activity. This may compromise the integrity of security solutions by causing notable events to go unreported. This activity may also impede forensic analysis and incident response, due to lack of sufficient data to determine what occurred.

**Name**

T1059

**ID**

T1059

**Description**

Adversaries may abuse command and script interpreters to execute commands, scripts, or binaries. These interfaces and languages provide ways of interacting with computer systems and are a common feature across many different platforms. Most systems come with some built-in command-line interface and scripting capabilities, for example, macOS and Linux distributions include some flavor of [Unix Shell](<https://attack.mitre.org/techniques/T1059/004>) while Windows installations include the [Windows Command Shell](<https://attack.mitre.org/techniques/T1059/003>) and [PowerShell](<https://attack.mitre.org/techniques/T1059/001>). There are also cross-platform interpreters such as [Python](<https://attack.mitre.org/techniques/T1059/006>), as well as those commonly associated with client applications such as [JavaScript](<https://attack.mitre.org/techniques/T1059/007>) and [Visual Basic](<https://attack.mitre.org/techniques/T1059/005>). Adversaries may abuse these technologies in various ways as a means of executing arbitrary commands. Commands and scripts can be embedded in [Initial Access](<https://attack.mitre.org/tactics/TA0001>) payloads delivered to victims as lure documents or as secondary payloads downloaded from an existing C2. Adversaries may also execute commands through interactive terminals/shells, as well as utilize various [Remote Services](<https://attack.mitre.org/techniques/T1021>) in order to achieve remote Execution.

(Citation: Powershell Remote Commands)(Citation: Cisco IOS Software Integrity Assurance - Command History)(Citation: Remote Shell Execution in Python)

**Name**

T1027

**ID**

T1027

**Description**

Adversaries may attempt to make an executable or file difficult to discover or analyze by encrypting, encoding, or otherwise obfuscating its contents on the system or in transit. This is common behavior that can be used across different platforms and the network to evade defenses. Payloads may be compressed, archived, or encrypted in order to avoid detection. These payloads may be used during Initial Access or later to mitigate detection. Sometimes a user's action may be required to open and [Deobfuscate/Decode Files or Information](<https://attack.mitre.org/techniques/T1140>) for [User Execution](<https://attack.mitre.org/techniques/T1204>). The user may also be required to input a password to open a password protected compressed/encrypted file that was provided by the adversary. (Citation: Volexity PowerDuke November 2016) Adversaries may also use compressed or archived scripts, such as JavaScript. Portions of files can also be encoded to hide the plain-text strings that would otherwise help defenders with discovery. (Citation: Linux/Cdorked.A We Live Security Analysis) Payloads may also be split into separate, seemingly benign files that only reveal malicious functionality when reassembled. (Citation: Carbon Black Obfuscation Sept 2016) Adversaries may also abuse [Command Obfuscation](<https://attack.mitre.org/techniques/T1027/010>) to obscure commands executed from payloads or directly via [Command and Scripting Interpreter](<https://attack.mitre.org/techniques/T1059>). Environment variables, aliases, characters, and other platform/language specific semantics can be used to evade signature based detections and application control mechanisms. (Citation: FireEye Obfuscation June 2017) (Citation: FireEye Revoke-Obfuscation July 2017)(Citation: PaloAlto EncodedCommand March 2017)

**Name**

T1518

**ID**

T1518

**Description**

Adversaries may attempt to get a listing of software and software versions that are installed on a system or in a cloud environment. Adversaries may use the information from [Software Discovery](<https://attack.mitre.org/techniques/T1518>) during automated discovery to shape follow-on behaviors, including whether or not the adversary fully infects the target and/or attempts specific actions. Adversaries may attempt to enumerate software for a variety of reasons, such as figuring out what security measures are present or if the compromised system has a version of software that is vulnerable to [Exploitation for Privilege Escalation](<https://attack.mitre.org/techniques/T1068>).

**Name**

T1105

**ID**

T1105

**Description**

Adversaries may transfer tools or other files from an external system into a compromised environment. Tools or files may be copied from an external adversary-controlled system to the victim network through the command and control channel or through alternate protocols such as [ftp](<https://attack.mitre.org/software/S0095>). Once present, adversaries may also transfer/spread tools between victim devices within a compromised environment (i.e. [Lateral Tool Transfer](<https://attack.mitre.org/techniques/T1570>)). On Windows, adversaries may use various utilities to download tools, such as `copy`, `finger`, [certutil](<https://attack.mitre.org/software/S0160>), and [PowerShell](<https://attack.mitre.org/techniques/T1059/001>) commands such as `Invoke-WebRequest` and `Invoke-WebRequest`. On Linux and macOS systems, a variety of utilities also exist, such as `curl`, `scp`, `sftp`, `tftp`, `rsync`, `finger`, and `wget`. (Citation: t1105\_lolbas) Adversaries may also abuse installers and package managers, such as `yum` or `winget`, to download tools to victim hosts. Files can also be transferred using various [Web Service](<https://attack.mitre.org/techniques/T1102>)s as well as native or

otherwise present tools on the victim system.(Citation: PTSecurity Cobalt Dec 2016) In some cases, adversaries may be able to leverage services that sync between a web-based and an on-premises client, such as Dropbox or OneDrive, to transfer files onto victim systems. For example, by compromising a cloud account and logging into the service's web portal, an adversary may be able to trigger an automatic syncing process that transfers the file onto the victim's machine.(Citation: Dropbox Malware Sync)

**Name**

T1204

**ID**

T1204

**Description**

An adversary may rely upon specific actions by a user in order to gain execution. Users may be subjected to social engineering to get them to execute malicious code by, for example, opening a malicious document file or link. These user actions will typically be observed as follow-on behavior from forms of [Phishing](<https://attack.mitre.org/techniques/T1566>). While [User Execution](<https://attack.mitre.org/techniques/T1204>) frequently occurs shortly after Initial Access it may occur at other phases of an intrusion, such as when an adversary places a file in a shared directory or on a user's desktop hoping that a user will click on it. This activity may also be seen shortly after [Internal Spearphishing](<https://attack.mitre.org/techniques/T1534>). Adversaries may also deceive users into performing actions such as enabling [Remote Access Software](<https://attack.mitre.org/techniques/T1219>), allowing direct control of the system to the adversary, or downloading and executing malware for [User Execution](<https://attack.mitre.org/techniques/T1204>). For example, tech support scams can be facilitated through [Phishing](<https://attack.mitre.org/techniques/T1566>), vishing, or various forms of user interaction. Adversaries can use a combination of these methods, such as spoofing and promoting toll-free numbers or call centers that are used to direct victims to malicious websites, to deliver and execute payloads containing malware or [Remote Access Software](<https://attack.mitre.org/techniques/T1219>). (Citation: Telephone Attack Delivery)

**Name**

T1112



**ID**

T1112

**Description**

Adversaries may interact with the Windows Registry to hide configuration information within Registry keys, remove information as part of cleaning up, or as part of other techniques to aid in persistence and execution. Access to specific areas of the Registry depends on account permissions, some requiring administrator-level access. The built-in Windows command-line utility [Reg](<https://attack.mitre.org/software/S0075>) may be used for local or remote Registry modification. (Citation: Microsoft Reg) Other tools may also be used, such as a remote access tool, which may contain functionality to interact with the Registry through the Windows API. Registry modifications may also include actions to hide keys, such as prepending key names with a null character, which will cause an error and/or be ignored when read via [Reg](<https://attack.mitre.org/software/S0075>) or other utilities using the Win32 API. (Citation: Microsoft Reghide NOV 2006) Adversaries may abuse these pseudo-hidden keys to conceal payloads/commands used to maintain persistence. (Citation: TrendMicro POWELIKS AUG 2014) (Citation: SpectorOps Hiding Reg Jul 2017) The Registry of a remote system may be modified to aid in execution of files as part of lateral movement. It requires the remote Registry service to be running on the target system. (Citation: Microsoft Remote) Often [Valid Accounts](<https://attack.mitre.org/techniques/T1078>) are required, along with access to the remote system's [SMB/Windows Admin Shares](<https://attack.mitre.org/techniques/T1021/002>) for RPC communication.

**Name**

T1218

**ID**

T1218

**Description**

Adversaries may bypass process and/or signature-based defenses by proxying execution of malicious content with signed, or otherwise trusted, binaries. Binaries used in this technique are often Microsoft-signed files, indicating that they have been either downloaded from Microsoft or are already native in the operating system.(Citation: LOLBAS

Project) Binaries signed with trusted digital certificates can typically execute on Windows systems protected by digital signature validation. Several Microsoft signed binaries that are default on Windows installations can be used to proxy execution of other files or commands. Similarly, on Linux systems adversaries may abuse trusted binaries such as `split` to proxy execution of malicious commands.(Citation: split man page)(Citation: GTFO split)

**Name**

T1036

**ID**

T1036

**Description**

Adversaries may attempt to manipulate features of their artifacts to make them appear legitimate or benign to users and/or security tools. Masquerading occurs when the name or location of an object, legitimate or malicious, is manipulated or abused for the sake of evading defenses and observation. This may include manipulating file metadata, tricking users into misidentifying the file type, and giving legitimate task or service names. Renaming abusible system utilities to evade security monitoring is also a form of [Masquerading](https://attack.mitre.org/techniques/T1036).(Citation: LOLBAS Main Site) Masquerading may also include the use of [Proxy](https://attack.mitre.org/techniques/T1090) or VPNs to disguise IP addresses, which can allow adversaries to blend in with normal network traffic and bypass conditional access policies or anti-abuse protections.

**Name**

T1567

**ID**

T1567

**Description**

Adversaries may use an existing, legitimate external Web service to exfiltrate data rather than their primary command and control channel. Popular Web services acting as an exfiltration mechanism may give a significant amount of cover due to the likelihood that hosts within a network are already communicating with them prior to compromise. Firewall rules may also already exist to permit traffic to these services. Web service providers also commonly use SSL/TLS encryption, giving adversaries an added level of protection.

**Name**

T1555

**ID**

T1555

**Description**

Adversaries may search for common password storage locations to obtain user credentials. Passwords are stored in several places on a system, depending on the operating system or application holding the credentials. There are also specific applications and services that store passwords to make them easier for users to manage and maintain, such as password managers and cloud secrets vaults. Once credentials are obtained, they can be used to perform lateral movement and access restricted information.

**Name**

T1562

**ID**

T1562

**Description**

Adversaries may maliciously modify components of a victim environment in order to hinder or disable defensive mechanisms. This not only involves impairing preventative

defenses, such as firewalls and anti-virus, but also detection capabilities that defenders can use to audit activity and identify malicious behavior. This may also span both native defenses as well as supplemental capabilities installed by users and administrators. Adversaries may also impair routine operations that contribute to defensive hygiene, such as blocking users from logging out of a computer or stopping it from being shut down. These restrictions can further enable malicious operations as well as the continued propagation of incidents.(Citation: Emotet shutdown) Adversaries could also target event aggregation and analysis mechanisms, or otherwise disrupt these procedures by altering other system components.

**Name**

T1106

**ID**

T1106

**Description**

Adversaries may interact with the native OS application programming interface (API) to execute behaviors. Native APIs provide a controlled means of calling low-level OS services within the kernel, such as those involving hardware/devices, memory, and processes. (Citation: NT API Windows)(Citation: Linux Kernel API) These native APIs are leveraged by the OS during system boot (when other system components are not yet initialized) as well as carrying out tasks and requests during routine operations. Adversaries may abuse these OS API functions as a means of executing behaviors. Similar to [Command and Scripting Interpreter](<https://attack.mitre.org/techniques/T1059>), the native API and its hierarchy of interfaces provide mechanisms to interact with and utilize various components of a victimized system. Native API functions (such as `NtCreateProcess``) may be directed invoked via system calls / syscalls, but these features are also often exposed to user-mode applications via interfaces and libraries.(Citation: OutFlank System Calls)(Citation: CyberBit System Calls)(Citation: MDSec System Calls) For example, functions such as the Windows API `CreateProcess()` or GNU `fork()` will allow programs and scripts to start other processes.(Citation: Microsoft CreateProcess)(Citation: GNU Fork) This may allow API callers to execute a binary, run a CLI command, load modules, etc. as thousands of similar API functions exist for various system operations.(Citation: Microsoft Win32)(Citation: LIBC) (Citation: GLIBC) Higher level software frameworks, such as Microsoft .NET and macOS Cocoa, are also available to interact with native APIs. These frameworks typically provide language wrappers/abstractions to API functionalities and are designed for ease-of-use/ portability of code.(Citation: Microsoft NET)(Citation: Apple Core Services)(Citation: MACOS

Cocoa)(Citation: macOS Foundation) Adversaries may use assembly to directly or indirectly invoke syscalls in an attempt to subvert defensive sensors and detection signatures such as user mode API-hooks.(Citation: Redops Syscalls) Adversaries may also attempt to tamper with sensors and defensive tools associated with API monitoring, such as unhooking monitored functions via [Disable or Modify Tools](https://attack.mitre.org/techniques/T1562/001).

**Name**

T1564

**ID**

T1564

**Description**

Adversaries may attempt to hide artifacts associated with their behaviors to evade detection. Operating systems may have features to hide various artifacts, such as important system files and administrative task execution, to avoid disrupting user work environments and prevent users from changing files or features on the system. Adversaries may abuse these features to hide artifacts such as files, directories, user accounts, or other system activity to evade detection.(Citation: Sofacy Komplex Trojan) (Citation: Cybereason OSX Pirrit)(Citation: MalwareBytes ADS July 2015) Adversaries may also attempt to hide artifacts associated with malicious behavior by creating computing regions that are isolated from common security instrumentation, such as through the use of virtualization technology.(Citation: Sophos Ragnar May 2020)

**Name**

T1489

**ID**

T1489

**Description**

Adversaries may stop or disable services on a system to render those services unavailable to legitimate users. Stopping critical services or processes can inhibit or stop response to an incident or aid in the adversary's overall objectives to cause damage to the environment.(Citation: Talos Olympic Destroyer 2018)(Citation: Novetta Blockbuster) Adversaries may accomplish this by disabling individual services of high importance to an organization, such as `MSExchangeIS`, which will make Exchange content inaccessible (Citation: Novetta Blockbuster). In some cases, adversaries may stop or disable many or all services to render systems unusable.(Citation: Talos Olympic Destroyer 2018) Services or processes may not allow for modification of their data stores while running. Adversaries may stop services or processes in order to conduct [Data Destruction](https://attack.mitre.org/techniques/T1485) or [Data Encrypted for Impact](https://attack.mitre.org/techniques/T1486) on the data stores of services like Exchange and SQL Server.(Citation: SecureWorks WannaCry Analysis)

**Name**

T1547

**ID**

T1547

**Description**

Adversaries may configure system settings to automatically execute a program during system boot or logon to maintain persistence or gain higher-level privileges on compromised systems. Operating systems may have mechanisms for automatically running a program on system boot or account logon.(Citation: Microsoft Run Key)(Citation: MSDN Authentication Packages)(Citation: Microsoft TimeProvider)(Citation: Cylance Reg Persistence Sept 2013)(Citation: Linux Kernel Programming) These mechanisms may include automatically executing programs that are placed in specially designated directories or are referenced by repositories that store configuration information, such as the Windows Registry. An adversary may achieve the same goal by modifying or extending features of the kernel. Since some boot or logon autostart programs run with higher privileges, an adversary may leverage these to elevate privileges.

**Name**

T1037

**ID**

T1037

**Description**

Adversaries may use scripts automatically executed at boot or logon initialization to establish persistence. Initialization scripts can be used to perform administrative functions, which may often execute other programs or send information to an internal logging server. These scripts can vary based on operating system and whether applied locally or remotely. Adversaries may use these scripts to maintain persistence on a single system. Depending on the access configuration of the logon scripts, either local credentials or an administrator account may be necessary. An adversary may also be able to escalate their privileges since some boot or logon initialization scripts run with higher privileges.

# Domain-Name

## Value

thinkforce.com.br

blamefade.com.br



# StixFile

**Value**

c9a27dbae96afb7d083577d30b2947c8ba9d1a6cb7e10e5f259f0929ef107882

c7dbb5e9e65a221a5f78328b5a6141dd46a0459b88248e84de345b2a6e52b1d9

c6fe9169764301cadccb252fbed218a1a997922f0df31d3e813b4fe2a3e6326d

9ef9bbfce214ee10a2e563e56fb6486161c2a623cd91bb5be055f5745edd6479

7435f11e41735736ea95e0c8a66e15014ee238c3a746c0f5b3d4faf4d05215af

4d8eac070b6b95f61055b96fb6567a477dbc335ef163c10514c864d9913d23cb

30991c9cac5f4c5c4f382f89055c3b5e9bb373c98ce6a5516d06db3f8a478554

# External References

- 
- <https://www.fortinet.com/blog/threat-research/byakugan-malware-behind-a-phishing-attack>
- 
- <https://otx.alienvault.com/pulse/6613cbd1b29dd49178a9434d>