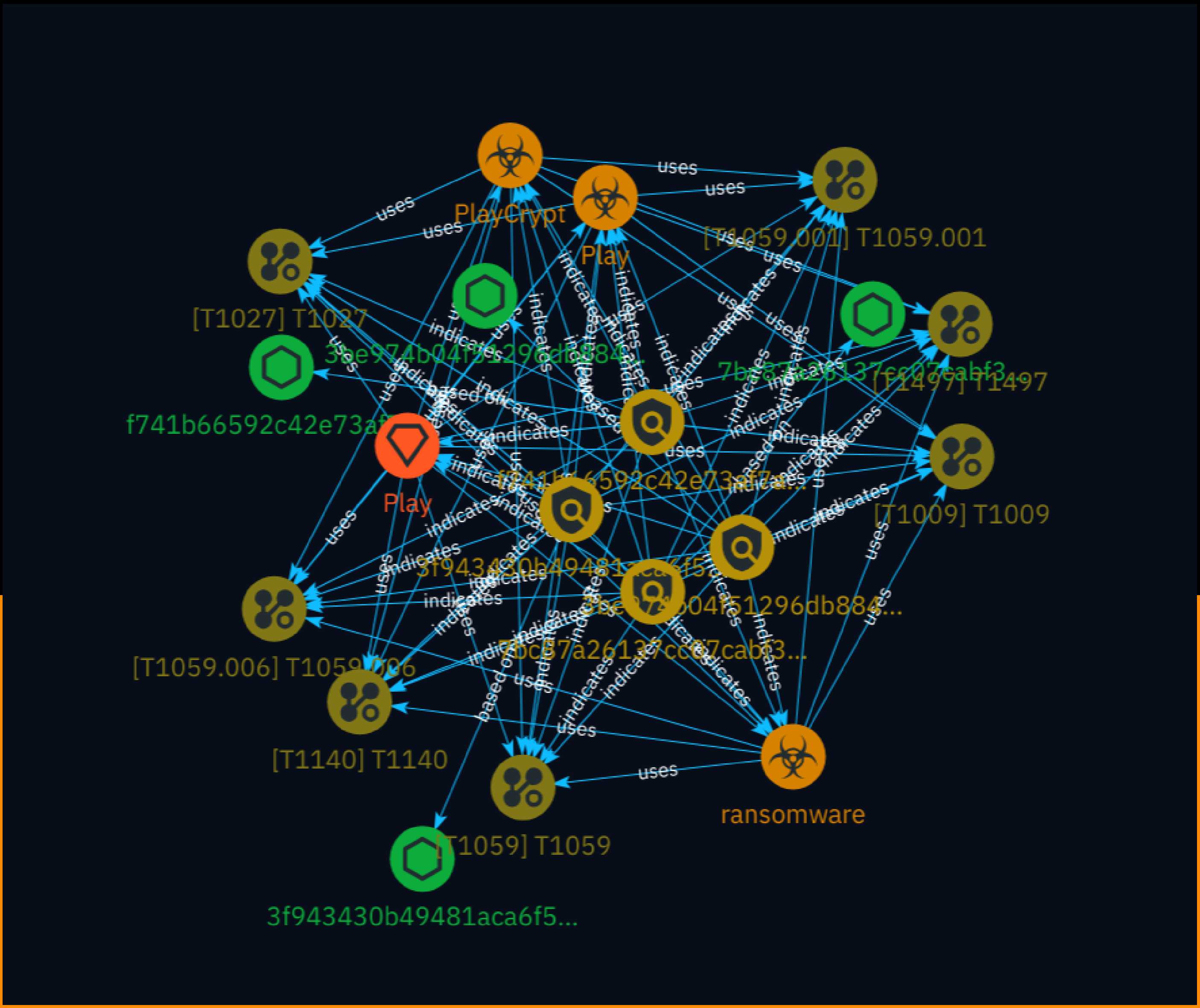
Aug 09 2024

NETMANAGEIT

# Intelligence Report REPLAY: Revisiting Play Ransomware Anti-Analysis Techniques



# Table of contents

# Overview

## Description

This analysis revisits the anti-analysis techniques employed by recent variants of the Play ransomware, which is known for targeting industries like healthcare and telecommunications across various regions. The report explains how the ransomware utilizes techniques like return-oriented programming (ROP), anti-disassembling tricks, junk code insertion, exploiting the Structured Exception Handling (SEH) mechanism, string obfuscation, and API hashing to hinder analysis and detection. Scripts developed by Netskope Threat Labs to aid in countering these techniques are also discussed.

## Confidence

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

100 / 100

# Content

N/A

# Attack-Pattern

| Name |
|---|
| T1009 |
| **ID** |
| T1009 |
| **Description** |
| Adversaries can use binary padding to add junk data and change the on-disk representation of malware without affecting the functionality or behavior of the binary. This will often increase the size of the binary beyond what some security tools are capable of handling due to file size limitations. Binary padding effectively changes the checksum of the file and can also be used to avoid hash-based blacklists and static anti-virus signatures.(Citation: ESET OceanLotus) The padding used is commonly generated by a function to create junk data and then appended to the end or applied to sections of malware.(Citation: Securelist Malware Tricks April 2017) Increasing the file size may decrease the effectiveness of certain tools and detection capabilities that are not designed or configured to scan large files. This may also reduce the likelihood of being collected for analysis. Public file scanning services, such as VirusTotal, limits the maximum size of an uploaded file to be analyzed.(Citation: VirusTotal FAQ) |
| **Name** |
| T1497 |
| **ID** |

T1497

## Description

Adversaries may employ various means to detect and avoid virtualization and analysis environments. This may include changing behaviors based on the results of checks for the presence of artifacts indicative of a virtual machine environment (VME) or sandbox. If the adversary detects a VME, they may alter their malware to disengage from the victim or conceal the core functions of the implant. They may also search for VME artifacts before dropping secondary or additional payloads. Adversaries may use the information learned from [Virtualization/Sandbox Evasion](https://attack.mitre.org/techniques/T1497) during automated discovery to shape follow-on behaviors.(Citation: Deloitte Environment Awareness) Adversaries may use several methods to accomplish [Virtualization/Sandbox Evasion](https://attack.mitre.org/techniques/T1497) such as checking for security monitoring tools (e.g., Sysinternals, Wireshark, etc.) or other system artifacts associated with analysis or virtualization. Adversaries may also check for legitimate user activity to help determine if it is in an analysis environment. Additional methods include use of sleep timers or loops within malware code to avoid operating within a temporary sandbox. (Citation: Unit 42 Pirpi July 2015)

## Name

T1059.006

## ID

T1059.006

## Description

Adversaries may abuse Python commands and scripts for execution. Python is a very popular scripting/programming language, with capabilities to perform many functions. Python can be executed interactively from the command-line (via the `python.exe` interpreter) or via scripts (.py) that can be written and distributed to different systems. Python code can also be compiled into binary executables.(Citation: Zscaler APT31 Covid-19 October 2020) Python comes with many built-in packages to interact with the underlying system, such as file operations and device I/O. Adversaries can use these libraries to download and execute commands or other scripts as well as perform various malicious behaviors.

**Name**

T1059.001

**ID**

T1059.001

**Description**

Adversaries may abuse PowerShell commands and scripts for execution. PowerShell is a powerful interactive command-line interface and scripting environment included in the Windows operating system.(Citation: TechNet PowerShell) Adversaries can use PowerShell to perform a number of actions, including discovery of information and execution of code. Examples include the `Start-Process` cmdlet which can be used to run an executable and the `Invoke-Command` cmdlet which runs a command locally or on a remote computer (though administrator permissions are required to use PowerShell to connect to remote systems). PowerShell may also be used to download and run executables from the Internet, which can be executed from disk or in memory without touching disk. A number of PowerShell-based offensive testing tools are available, including [Empire](https://attack.mitre.org/software/S0363), [PowerSploit](https://attack.mitre.org/software/S0194), [PoshC2](https://attack.mitre.org/software/S0378), and PSAttack.(Citation: Github PSAttack) PowerShell commands/scripts can also be executed without directly invoking the `powershell.exe` binary through interfaces to PowerShell's underlying `System.Management.Automation` assembly DLL exposed through the .NET framework and Windows Common Language Interface (CLI).(Citation: Sixdub PowerPick Jan 2016)(Citation: SilentBreak Offensive PS Dec 2015)(Citation: Microsoft PSfromCsharp APR 2014)

**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**

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**

T1140

**ID**

T1140

**Description**

Adversaries may use [Obfuscated Files or Information](https://attack.mitre.org/techniques/T1027) to hide artifacts of an intrusion from analysis. They may require separate mechanisms to decode or deobfuscate that information depending on how they intend to use it. Methods for doing that include built-in functionality of malware or by using utilities present on the system. One such example is the use of [certutil](https://attack.mitre.org/software/S0160) to decode a remote access tool portable executable file that has been hidden inside a certificate file.(Citation: Malwarebytes Targeted Attack against Saudi Arabia) Another example is using the Windows `copy /b` command to reassemble binary fragments into a malicious payload.(Citation: Carbon Black Obfuscation Sept 2016) Sometimes a user's action may be required to open it for deobfuscation or decryption as part of [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)

# Indicator

| **Name** |
|---|
| 7bc87a26137cc07cabf31e6e4bcd0e514846b5dd727a29132919f2e6b317cde8 |
| **Pattern Type** |
| stix |
| **Pattern** |
| [file:hashes.'SHA-256' = '7bc87a26137cc07cabf31e6e4bcd0e514846b5dd727a29132919f2e6b317cde8'] |
| **Name** |
| f741b66592c42e73af7adc46815cf6183765a2fb6a5f9f96cc75eaaf7dc15402 |
| **Pattern Type** |
| stix |
| **Pattern** |
| [file:hashes.'SHA-256' = 'f741b66592c42e73af7adc46815cf6183765a2fb6a5f9f96cc75eaaf7dc15402'] |
| **Name** |

| |
|---|
| 3be974b04f51296db884e46d0baf9e750a79731376d06887377bde3d6c3be6f6 |
| **Pattern Type** |
| stix |
| **Pattern** |
| [file:hashes.'SHA-256' = '3be974b04f51296db884e46d0baf9e750a79731376d06887377bde3d6c3be6f6'] |
| **Name** |
| 3f943430b49481aca6f57051ed0ced1a08038373f063afdd2423d8d72b19b545 |
| **Pattern Type** |
| stix |
| **Pattern** |
| [file:hashes.'SHA-256' = '3f943430b49481aca6f57051ed0ced1a08038373f063afdd2423d8d72b19b545'] |

# Intrusion-Set

| Name |
|---|
| Play |

# Malware

| Name |
|---|
| ransomware |

| Name |
|---|
| Play |

| Name |
|---|
| PlayCrypt |

# uses

| Name |
|---|
| |
| **Name** |
| |
| **Name** |
| |
| **Name** |
| |
| **Name** |
| |
| **Name** |
| |
| **Name** |
| |
| **Name** |
| |
| **Name** |

| |
|---|
| |
| **Name** |
| |
| **Name** |
| |
| **Name** |
| |
| **Name** |
| |
| **Name** |
| |
| **Name** |
| |
| **Name** |
| |
| **Name** |
| |
| **Name** |
| |
| **Name** |
| |
| **Name** |

| |
|---|
| |
| **Name** |
| |
| **Name** |
| |
| **Name** |
| |
| **Name** |
| |
| **Name** |
| |
| **Name** |
| |
| **Name** |
| |
| **Name** |
| |
| **Name** |
| |
| **Name** |
| |
| **Name** |

# indicates

| **Name** |
|---|
|  |
| **Name** |
|  |
| **Name** |
|  |
| **Name** |
|  |
| **Name** |
|  |
| **Name** |
|  |
| **Name** |
|  |
| **Name** |
|  |
| **Name** |

| |
|---|
| |
| **Name** |
| |
| **Name** |
| |
| **Name** |
| |
| **Name** |
| |
| **Name** |
| |
| **Name** |
| |
| **Name** |
| |
| **Name** |
| |
| **Name** |
| |
| **Name** |
| |
| **Name** |

| |
|---|
| |
| **Name** |
| |
| **Name** |
| |
| **Name** |
| |
| **Name** |
| |
| **Name** |
| |
| **Name** |
| |
| **Name** |
| |
| **Name** |
| |
| **Name** |
| |
| **Name** |
| |
| **Name** |

| |
|---|
| **Name** |
| |
| **Name** |
| |
| **Name** |
| |
| **Name** |
| |
| **Name** |
| |
| **Name** |
| |
| **Name** |
| |
| **Name** |
| |
| **Name** |
| |
| **Name** |
| |
| **Name** |

| Name |
|---|
| |

| Name |
|---|
| |

# based-on

| Name |
|---|
| |

| Name |
|---|
| |

| Name |
|---|
| |

| Name |
|---|
| |

# StixFile

| Value |
|---|
| 3f943430b49481aca6f57051ed0ced1a08038373f063afdd2423d8d72b19b545 |
| f741b66592c42e73af7adc46815cf6183765a2fb6a5f9f96cc75eaaf7dc15402 |
| 7bc87a26137cc07cabf31e6e4bcd0e514846b5dd727a29132919f2e6b317cde8 |
| 3be974b04f51296db884e46d0baf9e750a79731376d06887377bde3d6c3be6f6 |

# External References

- https://www.netskope.com/blog/replay-revisiting-play-ransomware-anti-analysis-techniques
- https://otx.alienvault.com/pulse/66b5fb5b1e1327099012c06c