Icon

Description automatically generated

Practical Malware Analysis & Triage

Malware Analysis Report

DemoWare Cryptor-Dropper Malware

Oct 2021 | HuskyHacks | v1.0

# Table of Contents

[Table of Contents 2](#_Toc85207993)

[Executive Summary 3](#_Toc85207994)

[High-Level Technical Summary 4](#_Toc85207995)

[Malware Composition 5](#_Toc85207996)

[srvupdate.exe 5](#_Toc85207997)

[crt1.crt: 5](#_Toc85207998)

[Basic Static Analysis 6](#_Toc85207999)

[Basic Dynamic Analysis 7](#_Toc85208000)

[Advanced Static Analysis 8](#_Toc85208001)

[Advanced Dynamic Analysis 9](#_Toc85208002)

[Indicators of Compromise 10](#_Toc85208003)

[Network Indicators 10](#_Toc85208004)

[Host-based Indicators 11](#_Toc85208005)

[Rules & Signatures 13](#_Toc85208006)

[Appendices 14](#_Toc85208007)

[A. Yara Rules 14](#_Toc85208008)

[B. Callback URLs 14](#_Toc85208009)

[C. Decompiled Code Snippets 15](#_Toc85208010)

# Executive Summary

|  |  |
| --- | --- |
| SHA256 hash | A6AA84358130078F9455773AF1E9EF2C7710934F72DF8514C9A62ABEB83D2E81 |

DemoWare is a cryptor-dropper malware sample first identified on Oct 15th, 2021. It is a GoLang-compiled dropper that runs on the x64 Windows operating system. It consists of two payloads that are executed in succession following a successful spearphishing attempt. Symptoms of infection include infrequent beaconing to any of the URLs listed in Appendix B, random blue screen popups on the endpoint, and an executable named “srvupdate.exe” appearing in the %APPDATA% directory.

YARA signature rules are attached in Appendix A. Malware sample and hashes have been submitted to VirusTotal for further examination.

# High-Level Technical Summary

DemoWare consists of two parts: an encrypted stage 0 dropper and an unpacked and decoded stage 2 command execution program. It first attempts to contact its callback URL (hxxps://demowarecallback.local) and unpacks its stage 2 payload if successful. Then, loren ipsum….

# Malware Composition

DemoWare consists of the following components:

|  |  |
| --- | --- |
| File Name | SHA256 Hash |
| srvupdate.exe | A6AA84358130078F9455773AF1E9EF2C7710934F72DF8514C9A62ABEB83D2E81 |
| crt1.crt | A6AA84358130078F9455773AF1E9EF2C7710934F72DF8514C9A62ABEB83D2E81 |

## srvupdate.exe

The initial executable that runs after a successful spearphish. Loren ipsum…

crt1.crt:

A Base64 encoded CRT file containing the second stage payload. Loren ipsum…

A screenshot of a computer

Description automatically generated with medium confidence

*Fig 1: Base64 encoded cert of the stage 1 payload.*

# Basic Static Analysis

{Screenshots and description about basic static artifacts and methods}

# Basic Dynamic Analysis

{Screenshots and description about basic dynamic artifacts and methods}

# Advanced Static Analysis

{Screenshots and description about findings during advanced static analysis}

# Advanced Dynamic Analysis

{Screenshots and description about advanced dynamic artifacts and methods}

# Indicators of Compromise

The full list of IOCs can be found in the Appendices.

## Network Indicators

{Description of network indicators}

Text

Description automatically generated

*Fig 3: WireShark Packet Capture of initial beacon check-in*

A picture containing table

Description automatically generated

*Fig 4: WireShark Packet Capture of stage 2 executable download.*

## Host-based Indicators

{Description of host-based indicators}

Graphical user interface, text, application, website

Description automatically generated

# Rules & Signatures

A full set of YARA rules is included in Appendix A.

{Information on specific signatures, i.e. strings, URLs, etc}

# Appendices

## Yara Rules

Full Yara repository located at: http://github.com/HuskyHacks/PMAT-lab

rule Yara\_Example {

    meta:

        last\_updated = "2021-10-15"

        author = "PMAT"

        description = "A sample Yara rule for PMAT"

    strings:

        // Fill out identifying strings and other criteria

*$string1* = "YOURETHEMANNOWDOG" ascii

*$string2* = "nim"

*$PE\_magic\_byte* = "MZ"

*$sus\_hex\_string* = { FF E4 ?? 00 FF }

    condition:

        // Fill out the conditions that must be met to identify the binary

*$PE\_magic\_byte* at 0 and

        (*$string1* and *$string2*) or

*$sus\_hex\_string*

}

## Callback URLs

|  |  |
| --- | --- |
| Domain | Port |
| hxxps://demowaredomain.local | 443 |
| hxxps://ec2-109-80-34-2.local | 443 |
| Hxxp://srv3.freetshirts.local | 80 |

## Decompiled Code Snippets

Text

Description automatically generated

*Fig 5: Process Injection Routine in Cutter*