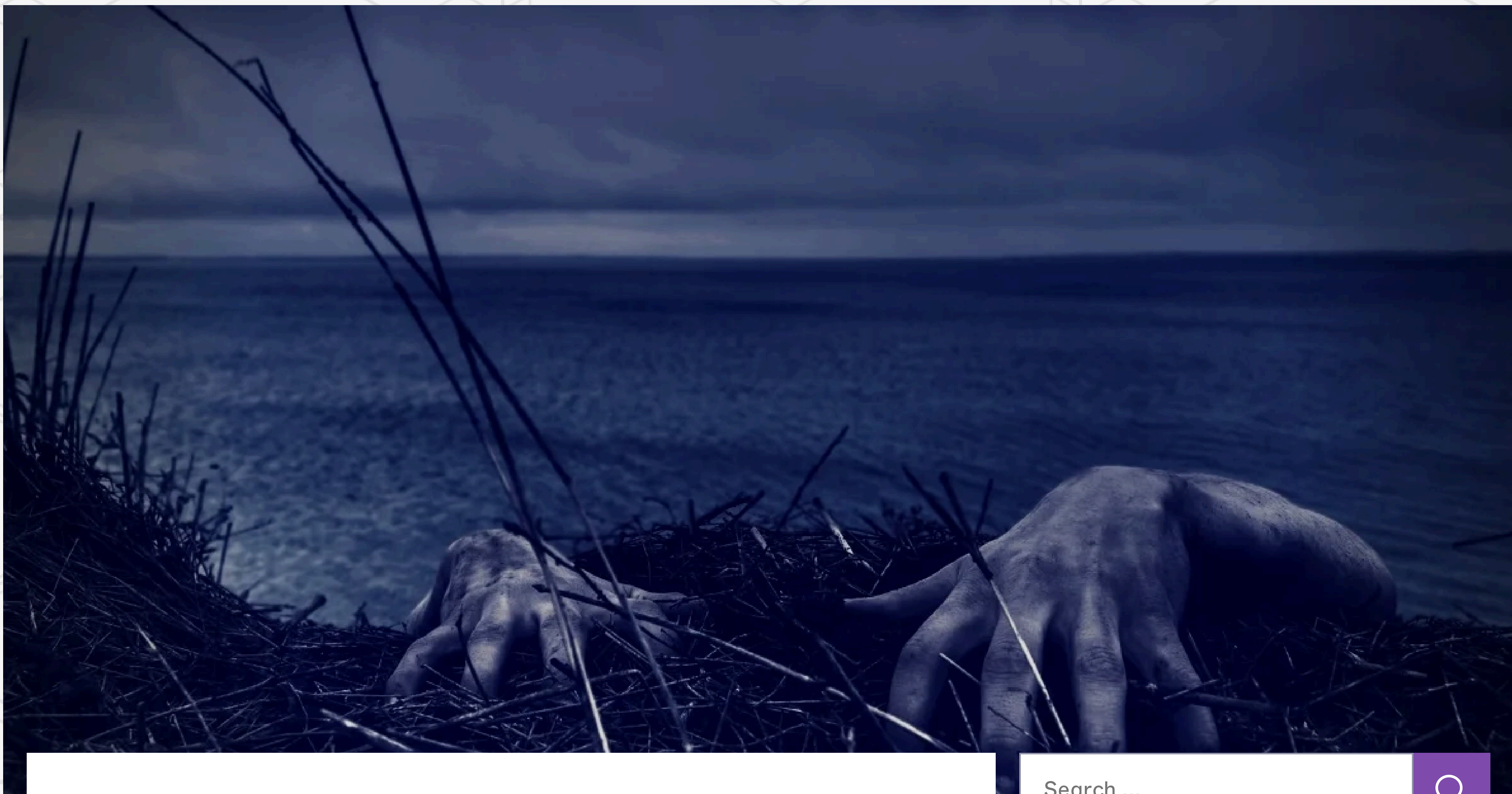
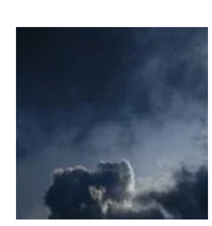


Search ...

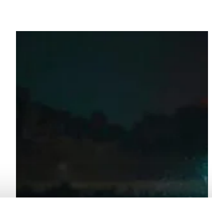
Background

Ranzy ransomware emerged in September/October this year, and appears to be an evolution of ThunderX and, to a lesser extent, Ako ransomware. Ranzy shares many features and under-the-hood elements with its predecessors. However there have been a few key updates, including tweaks to encryption, methods of exfiltration, and the (now commonplace) use of a public “leak blog” to post victim data for those who do not comply with the ransom demand.

RECENT POSTS



Cloud Malware | A Threat Hunter’s Guide to Analysis, Techniques and Delivery
OCTOBER 24, 2024



China’s Influence Ops | Twisting Tales of Volt Typhoon at Home and Abroad

by downloads). The "rebrand" from ThunderX to Ranzy occurred after free-decryption programs for ThunderX started to appear. A free decryption tool for ThunderX was posted to the NoMoreRansom project in September of this year.

This 'rebrand' distances the actors from ThunderX as well as improves upon the encryption mechanism so as to reduce the feasibility of future, free, decryption tools. With ThunderX emerging around August 2020, it would seem as though the lifecycle of this particular family has been rather short throughout its evolution. Note that some early samples of Ako were observed around January 2020.

As we observed with Ako and ThunderX, the primary delivery method observed is email (phish) with the malicious payload attached. Current samples (Ranzy Locker 1.1) append a `.ranzy` extension to encrypted files (with early versions using just `.RNZ`). Also of note, current Ranzy Locker payloads tend to include the same PDB patch as their ThunderX ancestors:

`C:UsersGh0StDesktopThunderXReleaseLockerStub.pdb`

```
uStack4 = 0x54;
local_64 = (void *)0x0;
local_60 = 0;
local_5c = 0;
local_8 = 0;
local_58 = param_1;
DVar1 = GetLogicalDrives();
uVar2 = 0;
do {
  local_58 = (void *)0x104;
  local_54 = uVar2 + 0x41 & 0xffff;
  local_20 = (WCHAR)local_54;
  local_1c = 0;
  local_1e = 0x3a;
  FUN_00403437(local_38,(void *)0x104);
  lpRemoteName = (undefined8 *)local_38;
  if (7 < local_24) {
    lpRemoteName = local_38[0];
  }
  WNetGetConnectionW(&local_20,(LPWSTR)lpRemoteName,(LPDWORD)&local_58);
  FUN_00406a49((short **)local_38);
  FUN_00403401(local_50,(undefined8 *)local_38);
  FUN_00403715(local_38);
  local_8._0_1_ = 1;
  if (local_40 == 0) {
    if ((DVar1 & 1 << ((byte)uVar2 & 0x1f)) != 0) {
      FUN_00403740(local_50,(undefined8 *)&local_54,(void *)0x1);
      FUN_0040438b(local_50,(undefined8 *)&DAT_0041d598,2);
      goto LAB_00406b85;
```

Improved Encryption Routines

Ranzy uses a combination of encryption algorithms to affect

2024

LABS CATEGORIES

- Crimeware
- Security Research
- Advanced Persistent Threat
- Adversary
- LABScon
- Security & Intelligence

file types by extension while excluding specific extensions and/or paths based on strings. Files that do not contain the `.dll`, `.exe`, `.ini`, `.lnk`, `.key`, `.rdp` are subject for inclusion. The ransomware will also exclude specific critical paths with strings including **AppData**, **boot**, **PerfLogs**, **PerfBoot**, **Intel**, **Microsoft**, **Windows** and **Tor Browser**.

Once launched, Ranzy payloads take a number of steps in order to both ensure maximum impact (encryption) as well as inhibiting standard recovery options where possible. Specific commands, and syntax, can vary across Windows versions and flavors. This includes the use of standard system tools to manipulate VSS and boot time recovery options.

After execution, the ransomware will swiftly call WMIC.EXE with the following syntax:

`wmic.exe SHADOWCOPY /nointeractive`

The following WBADMIN, BCDEDIT, and VSSADMIN commands are then issued to shift the victim host to the desired, compromised, state:

`wbadmin DELETE SYSTEMSTATEBACKUP`
`wbadmin DELETE SYSTEMSTATEBACKUP -deleteOldest`

`bcdedit.exe /set {default} recoveryenabled No`
`bcdedit.exe /set {default} bootstatuspolicy ignoreallfailures`

`vssadmin.exe Delete Shadows /All /Quiet`

```
    if (7 < (uint)param_1[5]) {
      param_1 = (undefined4 *)*param_1;
    }
    local_60 = param_1;
    iVar1 = RmRegisterResources(local_50,1,&local_60,0,0,0,0);
    if (iVar1 == 0) {
      local_54 = 0;
      local_58 = 0;
      pvVar2 = (void *)0x0;
      local_5c = 0;
      uVar3 = 0;
      do {
        iVar1 = RmGetList(local_50,&local_58,&local_54,pvVar2,&local_5c);
        if (iVar1 == 0) {
          if (local_5c == 0) {
            RmShutdown(local_50,0,0);
          }
          break;
        }
        if (iVar1 != 0xea) goto LAB_004077d9;
        local_54 = local_58;
        if (pvVar2 != (void *)0x0) {
          thunk_FUN_0040c5cd(pvVar2);
        }
        pvVar2 = (void *)FUN_004088c6(-(uint)((int)((ulonglong)local_54 * 0x29c >> 0x20) != 0) |
                                      (uint)((ulonglong)local_54 * 0x29c));
        bVar4 = uVar3 < 3;
        uVar3 = uVar3 + 1;
      } while (bVar4);
      if (pvVar2 != (void *)0x0) {
        thunk_FUN_0040c5cd(pvVar2);
      }
      if (local_50 != -1) {
        RmEndSession(local_50);
      }
```

Both Ranzy versions analyzed appear to retain the same multithreading capabilities that first appeared in ThunderX. The payload will first identify the number of processors available via `GetSystemInfo()`. Following this, the ransomware will leverage `IoCompletionPort` to generate a queue of files which are to be encrypted. Then, the ransomware is able to allocate a number of threads (equal to 2x the count of processors identified). This allows for fairly competitive (and therefore dangerous) encryption speeds when compared to the likes of Maze or NetWalker.

Post Encryption Behavior

Ranzy's ransom notes are deposited into each folder containing affected files/data. Across the analyzed versions, these are always identified with the name **readme.txt**. There are minor variations in the ransom notes across versions of the ransomware. That being said, the basic structure and content across ThunderX, Ranzy and Ranzy 1.1 are all quite similar.

Examples of the Ranzy and Ranzy 1.1 ransom notes can be seen below.

Perhaps the most significant difference between the ransom notes is with Ranzy 1.1, victims are instructed to access a TORengineering, security & investigations, and Government administration industries.

Conclusion

The Ranzy, ThunderX and Ako family is yet another example of how nimble and aggressive these threats and the actors behind them are becoming. With little to no barrier for entry (beyond a small investment of cash), any enterprising cybercriminal can gain access to, and manage, ransomware like Ranzy, potentially causing a great deal of financial damage. As we know, this damage is not limited to the direct payment of the ransom (which you should avoid), but now also includes any penalties associated with data breaches, public posting of private data, GDPR / compliance fallout, and beyond.

These threats are very agile, and it is clear that the actors behind them are paying attention to the efforts on the defense side. For example, when decryptor utilities are released, they quickly update their code and start distributing better and stronger payloads to nullify any workarounds.

Indicators of Compromise

SHA256

c4f72b292750e9332b1f1b9761d5aefc07301bc15edf31adeaf2e608000ec1c9

393fd0768b24cd76ca653af3eba9bff93c6740a2669b30cf59f8a064c46437a2

90691a36d1556ba7a77d0216f730d6cd9a9063e71626489094313c0afe85a939

bbf122cce1176b041648c4e772b230ec49ed11396270f54ad2c5956113caf7b7

ade5d0fe2679fb8af652e14c40e099e0c1aaea950c25165cebb1550e33579a79

SHA1

Indicator Removal on Host: File Deletion T1070.004

Modify Registry T1112

Query Registry T1012

System Information Discovery T1082

Peripheral Device Discovery T1120

Inhibit System Recovery T1490

Create or Modify System Process: Windows Service T1031

Exfiltration TA0010

RAAS RANSOMWARE

SHARE

PDF

JIM WALTER

Jim Walter is a Senior Threat Researcher at SentinelOne focusing on evolving trends, actors, and tactics within the thriving ecosystem of cybercrime and crimeware. He specializes in the discovery and analysis of emerging cybercrime "services" and evolving communication channels leveraged by mid-level criminal organizations. Jim joined SentinelOne following ~4 years at a security start-up, also focused on malware research and organized crime. Previously, he spent over 17 years at McAfee/Intel running their Threat Intelligence and Advanced Threat Research teams.

PREV NEXT

SentinelLABS

Kryptina RaaS | From Unsellable Cast-Off to Enterprise Ransomware
SEPTEMBER 23 2024

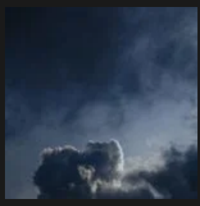
Xeon Sender | SMS Spam Shipping Multi-Tool Targeting SaaS Credentials
AUGUST 19 2024

NullBulge | Threat Actor Masquerades as Hacktivist Group Rebelling Against AI
JULY 16 2024

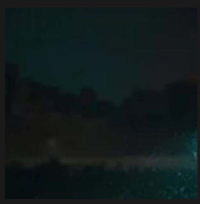
SENTINELLABS

In the era of interconnectivity, when markets, geographies, and jurisdictions merge in the melting pot of the digital domain, the perils of the threat ecosystem become unparalleled. Crimeware families achieve an unparalleled level of technical sophistication, APT groups are competing in fully-fledged cyber warfare, while once decentralized and scattered threat actors are forming adamant alliances of operating as elite corporate espionage teams.

RECENT POSTS



Cloud Malware | A Threat Hunter’s Guide to Analysis, Techniques and Delivery
OCTOBER 24, 2024



China’s Influence Ops | Twisting Tales of Volt Typhoon at Home and Abroad
OCTOBER 16, 2024

Kryptina RaaS | From Unsellable Cast-Off to Enterprise Ransomware
SEPTEMBER 23, 2024

SIGN UP

Twitter LinkedIn