Antlion: Chinese APT Uses Custom Backdoor to Target Financial Institutions in Taiwan

symantec-enterprise-blogs.security.com/blogs/threat-intelligence/china-apt-antlion-taiwan-financial-attacks

The attackers spent a significant amount of time on victim networks.

Chinese state-backed advanced persistent threat (APT) group Antlion has been targeting financial institutions in Taiwan in a persistent campaign over the course of at least 18 months.

The attackers deployed a custom backdoor we have called xPack on compromised systems, which gave them extensive access to victim machines.

The backdoor allowed the attackers to run WMI commands remotely, while there is also evidence that they leveraged EternalBlue exploits in the backdoor. The attackers appeared to have the ability to interact with SMB shares, and it's possible that they used mounted shares over SMB to transfer files from attacker-controlled infrastructure. There is also evidence that the attackers were able to browse the web through the backdoor, likely using it as a proxy to mask their IP address.

The goal of this campaign appears to have been espionage, as we saw the attackers exfiltrating data and staging data for exfiltration from infected networks.

Technical details

As well as the attack on the financial institution outlined in the case study below, Antlion compromised the networks of at least two other organizations in Taiwan, including another financial organization and a manufacturing company. The activity the group carried out on those networks was largely similar to the activity that is detailed in the case study, with the xPack backdoor frequently deployed and a lot of evidence of credential dumping. In the manufacturing target, also, we see the attackers attempting to download malicious files via SMB shares.

The attackers also spent a significant amount of time on both these targeted networks, spending close to 250 days on the financial organization and around 175 days on the manufacturing organization.

Symantec, a division of Broadcom, cannot state with certainty what the initial infection vector used by the attackers in this campaign was, though in one instance they were seen utilizing the MSSQL service to execute system commands, which indicates that the most likely infection vector was exploitation of a web application or service. However, Antlion are also known to have previously used malicious emails to gain initial access to victim networks.

The main custom backdoor used by Antlion in this campaign was the xPack backdoor, which is a custom .NET loader that decrypts (AES), loads, and executes accompanying .bin files. Its decryption password is provided as a command-line argument (Base64 encoded string), and xPack is intended to be run as a standalone application or as a service (xPackSvc variant). The xPack malware and its associated payload seems to be used for initial access; it appears that xPack was predominantly used to execute system commands, drop subsequent malware and tools, and stage data for exfiltration. The attackers also used a custom keylogger and three custom loaders.

- EHAGBPSL loader - custom loader written in C++ - loaded by JpgRun loader
- JpgRun loader - customer loader written in C++ - similar to xPack, reads the decryption key and filename from the command line - decodes the file and executes it
- CheckID - custom loader written in C++ - based on loader used by BlackHole RAT

The attackers also used a custom SMB session enumeration tool (NetSessionEnum), a custom bind/reverse file transfer tool named ENCODE MMC, and a Kerberos golden ticket tool based on Mimikatz.

The attackers also used a variety of off-the-shelf tools, as well as leveraging living-off-the-land tools such as PowerShell, WMIC, ProcDump, LSASS, and PsExec. The legitimate AnyDesk tool was also abused by the attackers for remote access in one of the victim organizations. The attackers were also observed leveraging exploits such as CVE-2019-1458 for privilege escalation and remote scheduled tasks to execute their backdoor. CVE-2019-1458 is an elevation-of-privilege vulnerability that occurs in Windows when the Win32k component fails to properly handle objects in memory.

Legitimate versions of WinRAR appear to have been exploited by the attackers for data exfiltration, while there is also evidence of data exfiltration via PowerShell, specifically using the BitsTransfer module to initiate an upload to attacker-controlled infrastructure. There is also evidence that the attackers likely automated the data collection process via batch scripts, while there is also evidence of instances where data was likely staged for further exfiltration, though it was not actually observed being exfiltrated from the network. In these instances, it appears the attackers were interested in collecting information from software pertaining to business contacts, investments, and smart card readers.

Case study: Attack on a financial organization

The attackers spent a significant amount of time on victims' networks, and deployed both custom and off-the-shelf malware. In one financial sector victim in Taiwan the attackers spent almost nine months on the victim network.

The first suspicious activity on this victim network occurred in December 2020 when WMIC was used to execute two commands:

- wmic process get CSName,Description,ExecutablePath,ProcessId /format:";CSIDL_SYSTEM\wbem\zh-tw\htable.xsl";
- wmic os get name,version,InstallDate,LastBootUpTime,LocalDateTime,Manufacturer,RegisteredUser,ServicePackMajorVersion,SystemDirectory /format:";CSIDL_SYSTEM\wbem\zh-tw\htable.xsl";

The first command was used to list the computer name, description of processes, executable path, and process ID. The output was written to a suspicious file named htable.xsl under the wbem directory. The second command was used to collect information about the system, which was written out to the same file (htable.xsl). Information collected included:

- Version of the operating system (OS)
- The installation date
- The last time the system was booted
- The local date and time of the system
- The manufacturer
- The registered user
- Service pack information - this can be used to determine what patches are installed
- System directory path

Five minutes after those commands were issued, WMIC was used to dump credentials:

- reg save HKLM\SAM CSIDL_COMMON_DOCUMENTS\sam.hiv
- reg save HKLM\SYSTEM CSIDL_COMMON_DOCUMENTS\sys.hiv
- reg save hklm\security CSIDL_COMMON_DOCUMENTS\security.hiv

The commands listed above were all executed via Antlion's custom xPack backdoor.

Several days later, during the Christmas holiday period, the attackers returned over a period of a few days and executed the xPack backdoor again. They also executed an unknown VBS script via PsExec multiple times:

> ";cscript.exe"; CSIDL_SYSTEM_DRIVE\update.vbs

On December 28, the attackers used xPack to launch a command prompt to dump credentials from several machines within the compromised organization with the following commands:

- upload.exe -accepteula -ma lsass.exe 16.dmp (a renamed version of Sysinternals procdump64.exe)
- reg save hklm\sam CSIDL_PROFILE\publicsam.hive
- reg save hklm\system CSIDL_PROFILE\public\system.hive
- reg save hklm\security CSIDL_PROFILE\public\security.hive

Over the following couple of weeks, the attackers continued to return intermittently to launch the xPack backdoor or to dump credentials via the registry. Then, following a few weeks of inactivity, they become active on the infected network once again.

The attackers used the xPack backdoor to launch a command prompt to execute the following commands:

- ";cmd"; /K CHCP 950
- CHCP 950
- query user
- ";CSIDL_SYSTEM\quser.exe";
- tasklist /v
- findstr explorer
- cmd /c dir ";CSIDL_PROFILE\desktop";
- CSIDL_SYSTEM\cmd.exe /c cmd /c dir \users /b
- cmd /c dir ";CSIDL_PROFILE\desktop";
- cmd /c dir \users /b

- reg save hklm\security CSIDL_COMMON_DOCUMENTS\security.hiv
- rar a -r -hp1qaz@WSX3edc!@# W22-009-099.tmp “;CSIDL_COMMON_DOCUMENTS\w22-009-099_file”;
- reg save hklm\system CSIDL_COMMON_DOCUMENTS\system.hiv
- reg save hklm\sam CSIDL_COMMON_DOCUMENTS\sam.hiv

The above commands were used to firstly change the code page to 950, which is the Windows code page for Traditional Chinese. The attackers then executed 'query user' to list any logged-in users on the system, as well as running ‘tasklist’ to get a list of all the running processes on the system. They also tried to discover what processes were running, before listing all contents of the Desktop directory and the Users directory. After this, the attackers dumped credentials again via the registry.

The attackers returned to the network a couple of weeks later and carried out largely the same activity. The attackers remained active on the network for March, April, and May 2021, intermittently returning to launch their xPack backdoor or dump credentials from the registry. Dumping credentials appears to be a main focus of the attackers, with them likely using these credentials to move laterally across the network to identify machines of interest from which they can exfiltrate data.

The last activity on this network, after a gap of three months, occurred in August 2021, when the attackers returned and listed all available shares. They then dumped credentials from the registry and proceeded to collect account, group, and workstation configuration information.

They then dumped credentials from the registry once again. This was the last activity seen on this network.

Experienced actor stays active

Antlion is believed to have been involved in espionage activities since at least 2011, and this recent activity shows that it is still an actor to be aware of more than 10 years after it first appeared.

The length of time that Antlion was able to spend on victim networks is notable, with the group able to spend several months on victim networks, affording plenty of time to seek out and exfiltrate potentially sensitive information from infected organizations. The targeting of Taiwan is perhaps unsurprising given we know Chinese state-backed groups tend to be interested in organizations in that region.

Protection

For the latest protection updates, please visit the Symantec Protection Bulletin.

Indicators of Compromise (IOCs)

If an IOC is malicious and the file is available to us, Symantec Endpoint products will detect and block that file.

Type	IOC	Description
SHA2	85867a8b4de856a943dd5efaaf3b48aecd2082aa0ceba799df53ba479e4e81c5	checkID
SHA2	12425edb2c50eac79f06bf228cb2dd77bb1e847c4c4a2049c91e0c5b345df5f2	xPack
SHA2	e4a15537f767332a7ed08009f4e0c5a7b65e8cbd468eb81e3e20dc8dfc36aeed	xPack
SHA2	e488f0015f14a0eff4b756d10f252aa419bc960050a53cc04699d5cc8df86c8a	xPack
SHA2	9456d9a03f5084e44f8b3ad936b706a819ad1dd89e06ace612351b19685fef92	xPack
SHA2	730552898b4e99c7f8732a50ae7897fb5f83932d532a0b8151f3b9b13db7d73c	xPack
SHA2	de9bd941e92284770b46f1d764905106f2c678013d3793014bdad7776540a451	xPack
SHA2	390460900c318a9a5c9026208f9486af58b149d2ba98069007218973a6b0df66	xPack
SHA2	4331d1610cdedba314fc71b6bed35fea03bc49241eb908a70265c004f5701a29	xPack
SHA2	9b5168a8f2950e43148fe47576ab3ac5b2cfa8817b124691c50d2c77207f6586	xPack
SHA2	a74cb0127a793a7f4a616613c5aae72142c1166f4bb113247e734f0efd48bdba	xPack
SHA2	e5259b6527e8612f9fd9bba0b69920de3fd323a3711af39f2648686fa139bc38	xPack
SHA2	eb7a23136dc98715c0a3b88715aa7e936b88adab8ebae70253a5122b8a402df3	xPack
SHA2	789f0ec8e60fbc8645641a47bc821b11a4486f28892b6ce14f867a40247954ed	Keylogger

Type	IOC	Description
SHA2	3db621cac1d026714356501f558b1847212c91169314c1d43bfc3a4798467d0d	Keylogger
SHA2	443f4572ed2aec06d9fb3a190de21bfced37c0cd2ee03dd48a0a7be762858925	JpgRun
SHA2	f4534e04caced1243bd7a9ce7b3cd343bf8f558982cbabff93fa2796233fe929	JpgRun
SHA2	e968e0d7e62fbc36ad95bc7b140cf7c32cd0f02fd6f4f914eeb7c7b87528cfe2	EHAGBPSL
SHA2	0bbb477c1840e4a00d0b6cd3bd8121b23e1ce03a5ad738e9aa0e5e0b2e1e1fea	EHAGBPSL
SHA2	55636c8a0baa9b57e52728c12dd969817815ba88ec8c8985bd20f23acd7f0537	EHAGBPSL
SHA2	2a541a06929dd7d18ddbae2cb23d5455d0666af7bdcdf45b498d1130a8434632	EHAGBPSL
SHA2	85867a8b4de856a943dd5efaaf3b48aecd2082aa0ceba799df53ba479e4e81c5	checkID
SHA2	29d7b82f9ae7fa0dbaf2d18c4d38d18028d652ed1ccc0846e8c781b4015b5f78	checkID
SHA2	f7cab241dac6e7db9369a4b85bd52904022055111be2fc413661239c3c64af3d	checkID
SHA2	2aa52776965b37668887a53dcd2374fc2460293b73c897de5d389b672e1313ff	checkID
SHA2	79a37464d889b41b7ea0a968d3e15e8923a4c0889f61410b94f5d02458cb9eed	checkID
SHA2	48d41507f5fc40a310fcd9148b790c29aeb9458ff45f789d091a9af114f26f43	NetSessionEnum
SHA2	f01a4841f022e96a5af613eb76c6b72293400e52787ab228e0abb862e5a86874	MMC
SHA2	e1a0c593c83e0b8873278fabceff6d772eeaaac96d10aba31fcf3992bc1410e5	MMC
SHA2	dfee6b3262e43d85f20f4ce2dfb69a8d0603bb261fb3dfa0b934543754d5128b	Mimikatz

Yara Rules

```
rule xpack_loader
{
  meta:
author = "Symantec, a division of Broadcom"
    hash = "12425edb2c50eac79f06bf228cb2dd77bb1e847c4c4a2049c91e0c5b345df5f2"
  strings:
    $s1 = "Length or Hash destoryed" wide fullword
    $s2 = "tag unmatched" wide fullword
    $s3 = "File size mismatch" wide fullword
    $s4 = "DESFile" wide fullword
    $p1 = "fomsal.Properties.Resources.resources" wide fullword
    $p2 = "xPack.Properties.Resources.resources" wide fullword
    $p3 = "foslta.Properties.Resources.resources" wide fullword
  condition:
    uint16(0) == 0x5A4D and uint32(uint32(0x3C)) == 0x00004550 and (2 of ($s*) or any of ($p*))
}
rule xpack_service
{
  meta:
author = "Symantec, a division of Broadcom"
    hash = "390460900c318a9a5c9026208f9486af58b149d2ba98069007218973a6b0df66"
  strings:
    $s1 = "C:\\Windows\\inf\\wdnvsc.inf" wide fullword
    $s2 = "PackService" wide fullword
    $s3 = "xPackSvc" wide fullword
    $s4 = "eG#!&5h8V$" wide fullword
  condition:
    uint16(0) == 0x5A4D and uint32(uint32(0x3C)) == 0x00004550 and 3 of them
}
rule EHAGBPSL_loader
{
  meta:
author = "Symantec, a division of Broadcom"
    hash = "e968e0d7e62fbc36ad95bc7b140cf7c32cd0f02fd6f4f914eeb7c7b87528cfe2"
    hash = "2a541a06929dd7d18ddbae2cb23d5455d0666af7bdcdf45b498d1130a8434632"
  strings:
    $s1 = {45 00 00 00 48 00 00 00 41 00 00 00 47 00 00 00 42 00 00 00 50 00 00 00 53 00 00 00 4C} // EHAGBPSL
    $s2 = {74 00 00 00 61 00 00 00 72 00 00 00 57 00 00 00 6F 00 00 00 6B} // tarWok
    $b1 = "bnRtZ3M=" fullword // ntmgs
    $b2 = "TmV0d29yayBNYW5hZ2VtZW50IFNlcnZpY2U=" fullword // Network Management Service
    $b3 = "UHJvdmlkZXMgYWJpbGl0eSB0byBtYW5hZ2UgbmV0d29yayBvdmVyIHRoZSBuZXQgcHJvdG9jb2wu" fullword //
Provides ability to manage network over the net protocol.
    $b4 = "bnRtZ3MuZG" // ntmgs.dll / ntmgs.dat
    $b5 = "aW1nMS5qcGc=" fullword // img1.jpg
    $c1 = "Wscms.nls" fullword
    $c2 = "Wscms.dat" fullword
    $c3 = "Wscms.dll" fullword
    $c4 = "Wscms.ini" fullword
    $c5 = "Images01.jpg" fullword
    $e1 = "StartWork" fullword
    $e2 = "ServiceMain" fullword
    $h1 = {DD 9C BD 72} // CreateRemoteThread
    $h2 = {C0 97 E2 EF} // OpenProcess
    $h3 = {32 6D C7 D5} // RegisterServiceCtrlHandlerA
```

```
        $h4 = {A1 6A 3D D8} // WriteProcessMemory
    condition:
        uint16(0) == 0x5A4D and uint32(uint32(0x3C)) == 0x00004550 and all of ($e*) and (all of ($s*) or any of ($b*) or 3 of ($c*) or all of ($h*))
}
rule keylogger
{
    meta:
author = "Symantec, a division of Broadcom"
        hash = "3db621cac1d026714356501f558b1847212c91169314c1d43bfc3a4798467d0d"
        hash = "789f0ec8e60fbc8645641a47bc821b11a4486f28892b6ce14f867a40247954ed"
    strings:
        $m1 = "BKB_Test" fullword
        $m2 = "KLG_sd76bxds1N" fullword
        $k1 = "[%d/%02d/%02d %02d:%02d:%02d K-E-Y-L-O-G]" fullword
        $k2 = "[%d/%02d/%02d %02d:%02d:%02d C-L-I-P-B-D]" fullword
        $k3 = "< Title--%s-- >" fullword
        $k4 = "ImpersonateLoggedOnUser Error(%d)" fullword
        $f1 = {55 73 65 72 ?? ?? ?? 00 00 00 ?? ?? ?? 6B 65 79 2E} // Userkey.
        $f2 = {55 73 65 72 ?? ?? ?? 00 00 00 ?? ?? ?? 64 61 74 2E} // Userdat.
    condition:
        uint16(0) == 0x5A4D and uint32(uint32(0x3C)) == 0x00004550 and (2 of ($k*) or (any of ($m*) and any of ($f*)))
}
rule checkid_loader
{
    meta:
author = "Symantec, a division of Broadcom"
        description = "BlackHole/BlackSwan / QuasarRAT/xClient loader"
        hash = "29d7b82f9ae7fa0dbaf2d18c4d38d18028d652ed1ccc0846e8c781b4015b5f78"
    strings:
        $s1 = "Call %s.%s(\"%s\") => %d" fullword wide
        $s2 = "Assembly::CreateInstance failed w/hr 0x%08lx" fullword wide
        $s3 = "checkID"
        $s4 = "NULL == checkID hMutex" fullword
        $s5 = "checkID Mutex ERROR_ALREADY_EXISTS" fullword
        $s6 = "dllmain mutex ERROR_ALREADY_EXISTS" fullword
        $x1 = "xClient.Program" fullword wide

        $x2 = "LoadPayload" fullword

        $m1 = "SFZJ_Wh16gJGFKL" ascii wide

        $m2 = "d5129799-e543-4b8b-bb1b-e0cba81bccf8" ascii wide

        $m3 = "USA_HardBlack" ascii wide

        $b1 = "BlackHole.Slave.Program" fullword wide

        $b2 = "NuGet\\Config" wide

        $b3 = "VisualStudio.cfi" wide

        $p = {E1 F6 3C AC AF AC AC AC A8 AC AC AC 53 53 AC AC 14}

        $t = "0s+Nksjd1czZ1drJktPO24aEjISMtsvLy5LJzNjdyNnL1dLY08uS39PRh0SMhIy2jYyPkomNk02IjJKEiIaEjISM"

    condition:

        uint16(0) == 0x5A4D and uint32(uint32(0x3C)) == 0x00004550 and 2 of ($s*) and (all of ($x*) or any of ($m*) or all of ($b*) or
$p or $t)

}
```

The Threat Hunter Team is a group of security experts within Symantec whose mission is to investigate targeted attacks, drive enhanced protection in Symantec products, and offer analysis that helps customers respond to attacks.