XKEYSCORE

Introduction to XKS Application IDs and Fingerprints

27 August 2009

DERIVED FROM: NSA/CSSM 1-52

Agenda

- Overview of Application IDs and Fingerprints
- Background of the 4 generations of AppIDs+Fingerprints
- Examples of how they are used for target development SIGDEV

What is an AppID?

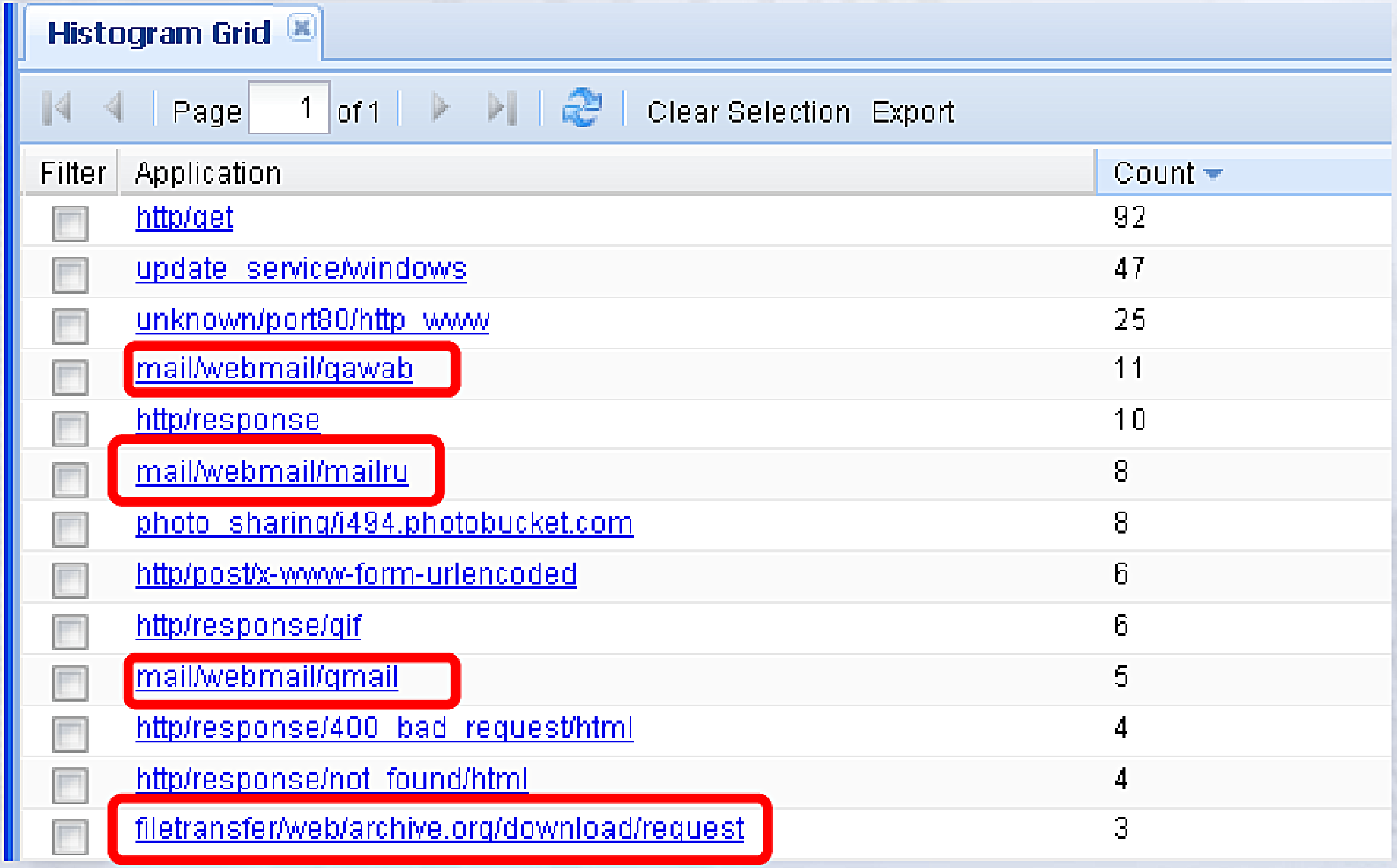
- An Application ID (AppID) is a meta-data tag given to a session to help describe what application is being seen in the traffic
- Examples:
  - mail/webmail/yahoo indicates that the traffic was Yahoo Webmail
  - chat/msn_messenger indicates the traffic was MSN Messenger
  - http/get indicates that the traffic was an HTTP Get

Why even have AppIDs/Fingerprints?

- What’s the point of AppIDs/Fingerprints?
- For one, they give you a powerful tool for the quick analysis of what applications are being seen in your traffic.
- A simple histogram on AppID allows you to quickly identify all of the applications seen for a given result set, without needing to view each piece of content

Why even have AppIDs/Fingerprints?

- Ex: Histogram the applications used during Target activity:



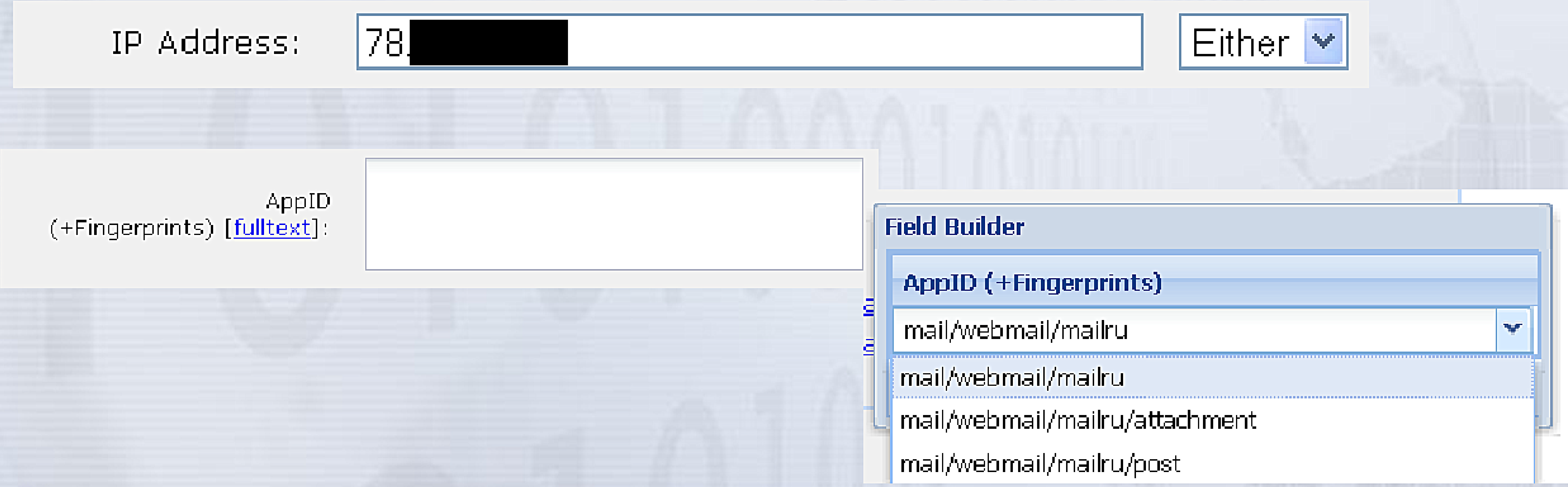
Filter	Application	Count
☐	http/get	92
☐	update_service/windows	47
☐	unknown/port80/http_www	25
☐	mail/webmail/gawab	11
☐	http/response	10
☐	mail/webmail/mailru	8
☐	photo_sharing/i494.photobucket.com	8
☐	http/post/x-www-form-urlencoded	6
☐	http/response/gif	6
☐	mail/webmail/gmail	5
☐	http/response/400_bad_request/html	4
☐	http/response/not_found/html	4
☐	filetransfer/web/archive.org/download/request	3

Why even have AppIDs/Fingerprints?

- Secondly, they provide an additional criteria that you can use in your query.
- **NOTE: It's important to point out that since most AppIDs + Fingerprints are tagging technology and/or applications, they SHOULD NOT be the sole criteria for your queries in X-KEYSCORE!**

Why even have AppIDs/Fingerprints?

- **EX: I'm looking for targets using mail.ru from behind a large Iranian proxy:**



IP Address: 78.[redacted] Either

AppID (+Fingerprints) [fulltext]:

Field Builder

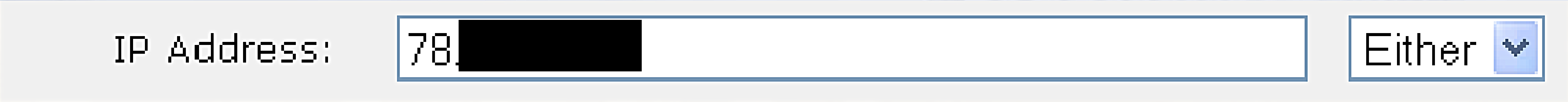
AppID (+Fingerprints)

mail/webmail/mailru

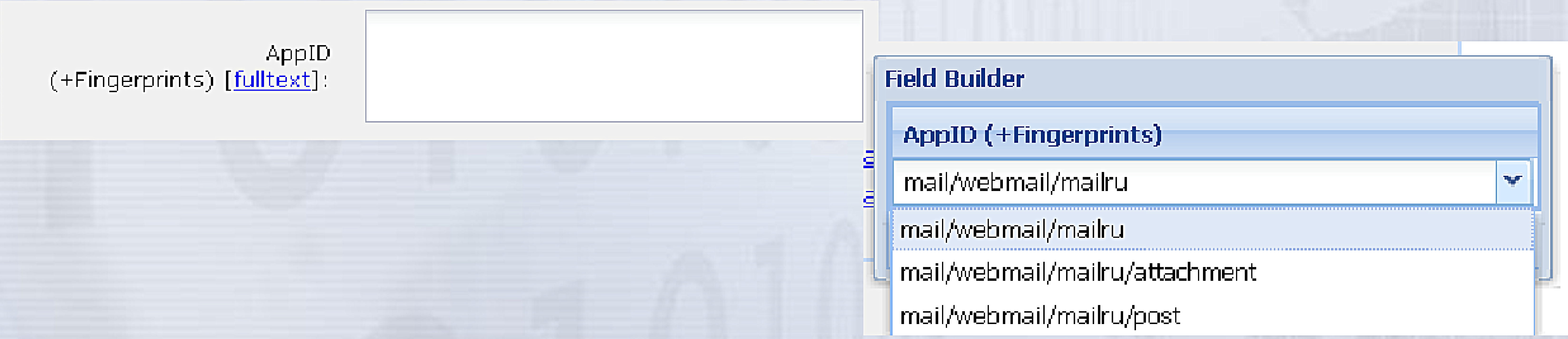
mail/webmail/mailru
mail/webmail/mailru/attachment
mail/webmail/mailru/post

Why even have AppIDs/Fingerprints?

- **EX: I’m looking for targets using mail.ru from behind a large Iranian proxy:**



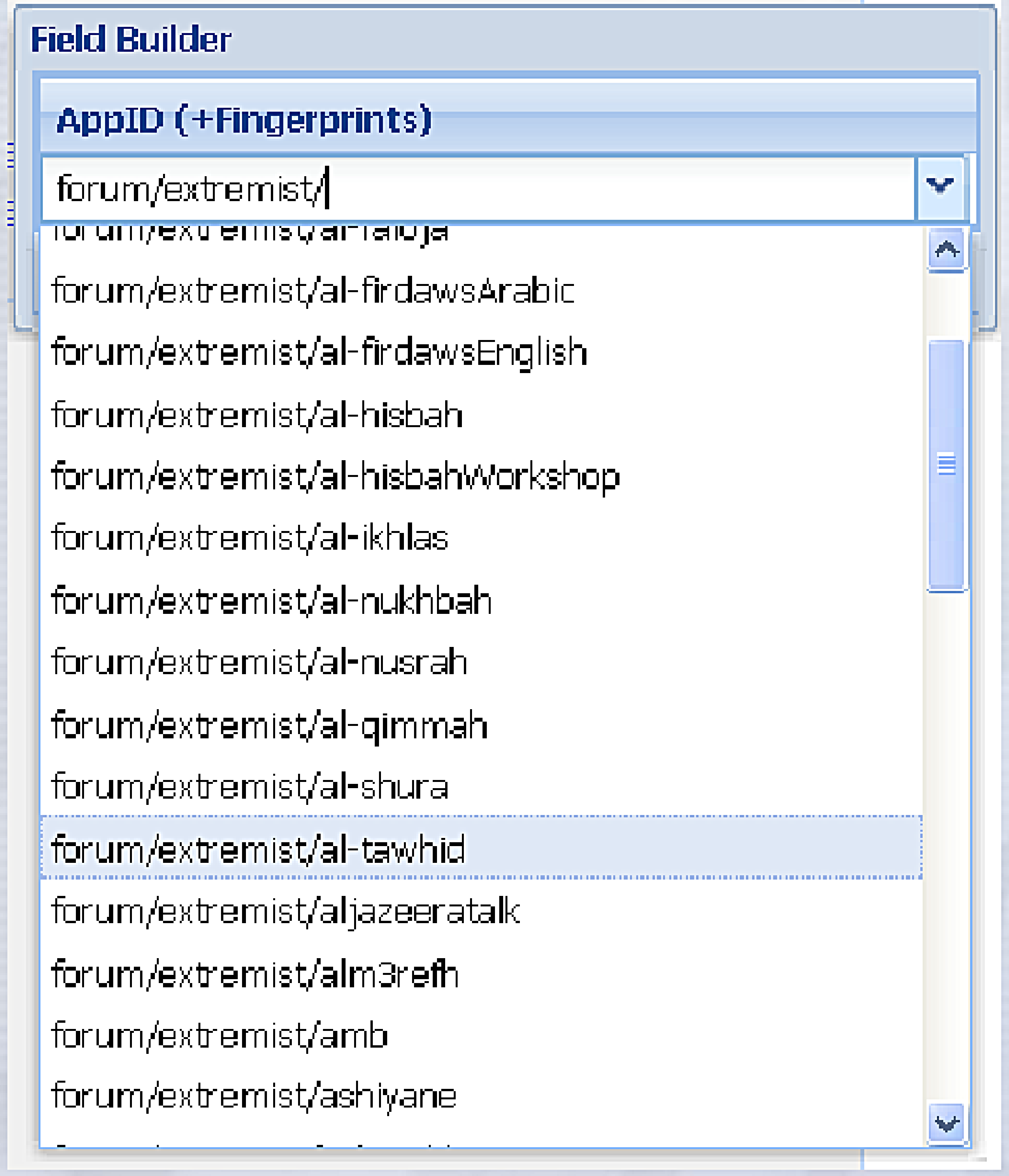
IP Address: 78.[redacted]



AppID (+Fingerprints) [fulltext]:

Why even have AppIDs/Fingerprints?

- **EX: I’m looking for Mojaheden Secrets 2 use in extremist web forums:**

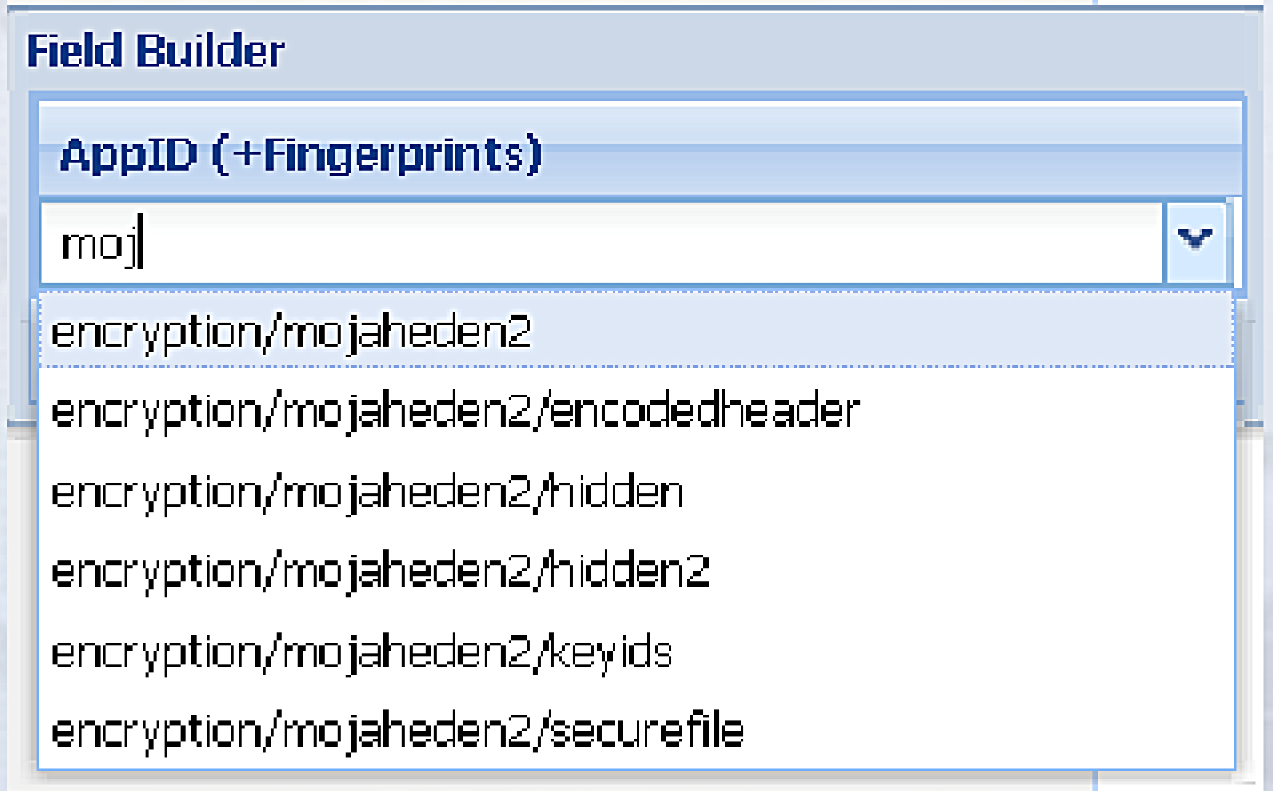


Field Builder

AppID (+Fingerprints)

forum/extremist/

forum/extremist/al-faloja
forum/extremist/al-firdawsArabic
forum/extremist/al-firdawsEnglish
forum/extremist/al-hisbah
forum/extremist/al-hisbahWorkshop
forum/extremist/al-ikhlas
forum/extremist/al-nukhbah
forum/extremist/al-nusrah
forum/extremist/al-qimmah
forum/extremist/al-shura
forum/extremist/al-tawhid
forum/extremist/aljazeeratalk
forum/extremist/alm3refh
forum/extremist/amb
forum/extremist/ashiyane



Field Builder

AppID (+Fingerprints)

moj

encryption/mojaheden2
encryption/mojaheden2/encodedheader
encryption/mojaheden2/hidden
encryption/mojaheden2/hidden2
encryption/mojaheden2/keyids
encryption/mojaheden2/securefile

How do AppIDs work?

- AppID’s are effectively looking for keywords in order to assign the AppID tag.
- Example, let’s say that this is the definition for mail/webmail/yahoo:

```
appid('mail/webmail/yahoo', 9.0) = 'Host: mail.yahoo';
```

Example

- Here is a client side Yahoo session:

```
GET /login.html HTTP/1.1
Referer: http://us.f359.mail.yahoo.com/ym/ShowLetter
Accept-Language: ar
Accept-Encoding: gzip, deflate
User-Agent: Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1; SV1)
Host: mail.yahoo.com
Connection: Keep-Alive
Cookie: B=fn50ehd2612o2&b=3&s=rp; YMBM=d=&v=1;
```

Example

```
appid('mail/webmail/yahoo', 9.0) = 'Host: mail.yahoo';
```

```
GET /login.html HTTP/1.1
Referer: http://us.f359.mail.yahoo.com/ym/ShowLetter
Accept-Language: ar
Accept-Encoding: gzip, deflate
User-Agent: Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1; SV1)
Host: mail.yahoo.com
Connection: Keep-Alive
Cookie: B=fn50ehd2612o2&b=3&s=rp; YMBM=d=&v=1;
```

Application: mail/webmail/yahoo

How AppIDs work

- What does the number in the AppID mean?
  appid('mail/webmail/yahoo', 9.0)=
- Each session can have only one AppID
- The goal is for the AppID to be as descriptive as possible
- Any given session might qualify under multiple AppIDs definitions, but only the most specific AppID that applies to the session is assigned
- Lowest number wins, so the lower the number, the more specific the AppID definition

How do AppIDs work?

- Let’s say there’s another more descriptive appid for mail/webmail/yahoo/login:

```
appid('mail/webmail/yahoo/login, 8.0) = 'Host: mail.yahoo' and
'/login';
```

- It has a lower number than mail/webmail/yahoo, so if it “hits” it will be applied

Example

```
appid('mail/webmail/yahoo', 9.0) = 'Host: mail.yahoo';
appid('mail/webmail/yahoo/login, 8.0) = 'Host: mail.yahoo' and
                                             '/login';
```

```
GET /login.html HTTP/1.1
Referer: http://us.f359.mail.yahoo.com/ym/ShowLetter
Accept-Language: ar
Accept-Encoding: gzip, deflate
User-Agent: Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1; SV1)
Host: mail.yahoo.com
Connection: Keep-Alive
Cookie: B=fn50ehd2612o2&b=3&s=rp; YMBM=d=&v=1;
```

Application: mail/webmail/yahoo/login

AppID Structure

- Note that the AppIDs have a directory-like structure:
- mail/webmail/yahoo and mail/webmail/yahoo/login
- If you wanted to search for all webmail activity you could search for mail/webmail/*
- If you wanted to search for all Yahoo mail activity you could search for mail/webmail/yahoo/*
- etc

How AppIDs work

- Some session can hit on many AppIDs.
- For example a single session might hit on:

```
appid('http/response', 9.2)
appid('mail/webmail', 8.9)
appid('mail/webmail/yahoo', 6.0)
appid('mail/webmail/yahoo/attachment', 5.0)
```

- Which one will be assigned as the winning AppID?

How AppIDs work

- When you see an AppID how do you know what was used to define that AppID?
- Through the XKS AppID signature page available through “go xkeyscore”
- Or by simply clicking on the hyperlink AppID from the new GUI!

What is a fingerprint?

- AppIDs were built to describe applications, of which there *should* only be one application seen per session.
- How do we describe other attributes of a session that aren’t necessarily tied to a particular application?

What is a fingerprint?

- One great example is encryption
- A particular type of encryption could be used in Yahoo Email, Gmail Email, SMTP Email.
- It could be used inside of a Word Document being uploaded to a free file website.
- It could be used inside of a private message sent through Facebook.
- Etc.

What is a fingerprint?

- How can we tag anytime we see that type of encryption regardless of the application we saw it in?
- Answer - Fingerprints
- Think of Fingerprints as “attributes” of a session.
- A session can have as many fingerprints as is needed to best describe it.

Example

```
appid('mail/webmail/yahoo', 9.0) = 'Host: mail.yahoo';
appid('mail/yahoo/login, 8.0) = 'Host: mail.yahoo' and '/login';

fingerprint( ‘mail/arabic') = 'mail' and /language[:=] ?ar/;
```

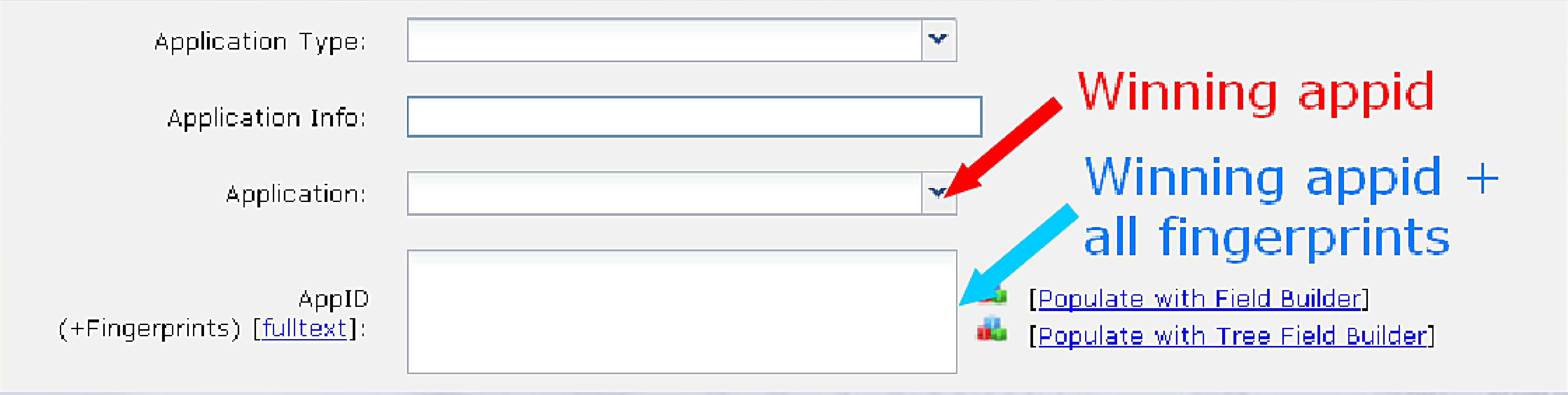
```
GET /login.html HTTP/1.1
Referer: http://us.f359.mail.yahoo.com/ym/ShowLetter
Accept-Language: ar
Accept-Encoding: gzip, deflate
User-Agent: Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1; SV1)
Host: mail.yahoo.com
Connection: Keep-Alive
Cookie: B=fn50ehd2612o2&b=3&s=rp; YMBM=d=&v=1;
```

Application: mail/webmail/yahoo/login
Fingerprint: mail/webmail/yahoo/login mail/arabic

Appid vs Fingerprint

Each session gets *one* appid -- lowest level wins. It gets databased in the 'application' field.

All matching fingerprints are stored in the 'fingerprint' field.



Application Type:

Application Info:

Application:

AppID (+Fingerprints) [fulltext]:

[Populate with Field Builder]

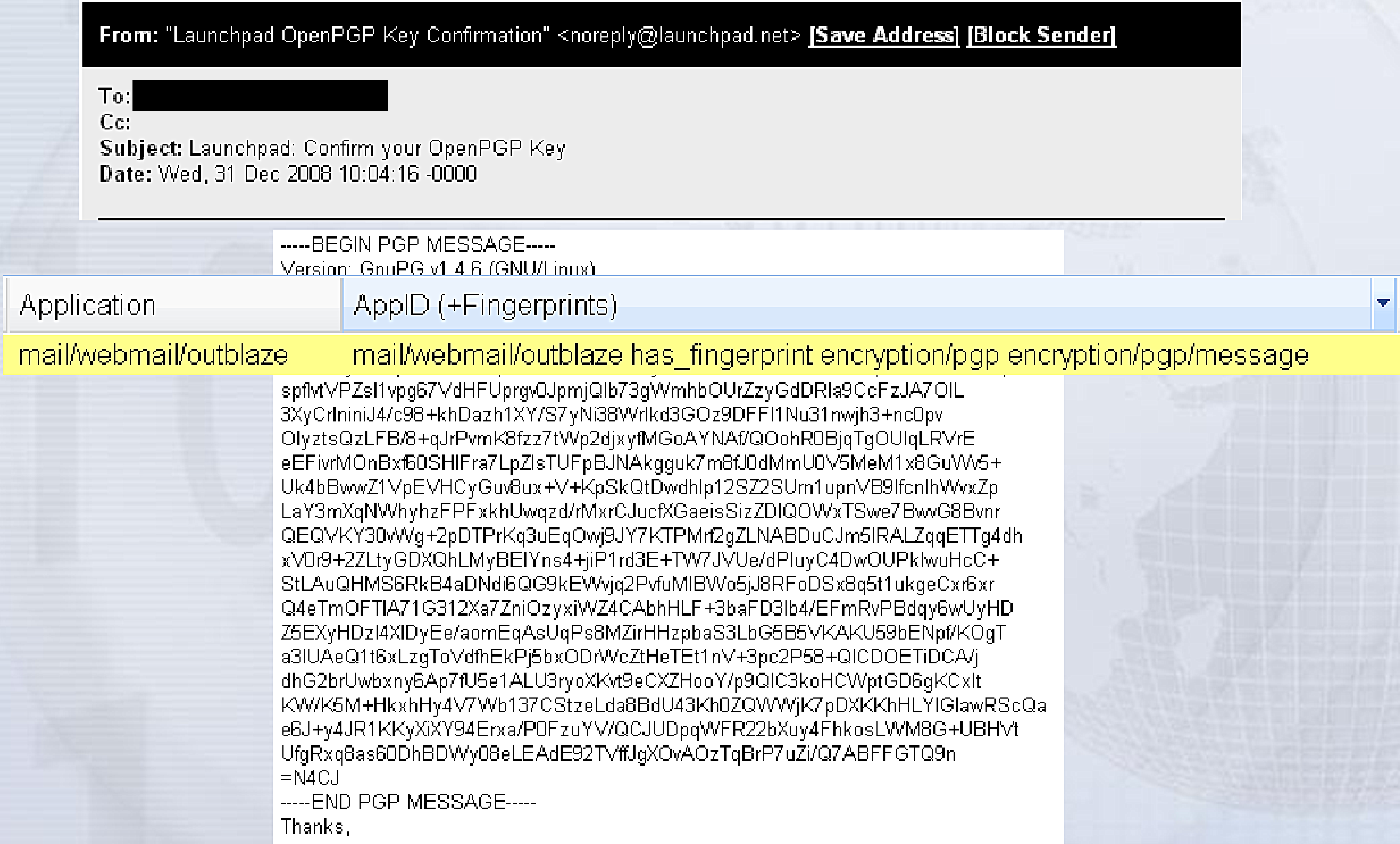
[Populate with Tree Field Builder]

Winning appid

Winning appid + all fingerprints

Fingerprint Examples

Ex: E-Mails with encryption



From: "Launchpad OpenPGP Key Confirmation" <noreply@launchpad.net> [Save Address] [Block Sender]

To: [redacted]
Cc:
Subject: Launchpad: Confirm your OpenPGP Key
Date: Wed, 31 Dec 2008 10:04:16 -0000

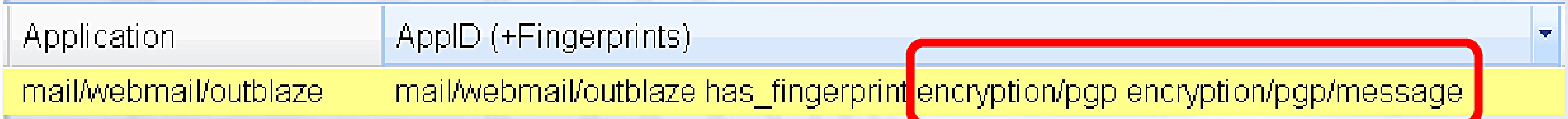
Application	AppID (+Fingerprints)
mail/webmail/outblaze	mail/webmail/outblaze has_fingerprint encryption/pgp encryption/pgp/message

```
-----BEGIN PGP MESSAGE-----
Version: GnuPG v1.4.6 (GNU/Linux)
spfMVPZsl1vpg67VdHFUprgv0JpmjQlb73gWmhbOUrZzyGdDRla9CcFzJA7OIL
3XyCrlniniJ4/c98+khDazh1XY/S7yNi38Wrlkd3GOz9DFFI1Nu31nwjh3+nc0pv
OlyztsQzLFB/8+qJrPvmK8fzz7tWp2djxyfMGoAYNAf/QOohR0BjqTgOUlqLRVrE
eEFivrMOnBxf60SHlFra7LpZlsTUFpBJNAkgguk7m8fJ0dMmU0V5MeM1x8GuWv5+
Uk4bBwwZ1VpEVHCyGuv8ux+V+KpSkQtDwdhlp12SZ2SUm1upnVB9lfcnlhWvxZp
LaY3mXqNWhyhzFPFxkhUwqzd/rMxrCJucfXGaeisSizZDlQOWxTSwe7BwG8Bvnr
QEQVKY30WWg+2pDTPrKq3uEqOwj9JY7KTPMrf2gZLNABDuCJm5lRALZqqETTg4dh
xV0r9+2ZLtyGDXQhLMyBElYns4+jiP1rd3E+TW7JVUe/dPluyC4DwOUPklwuHcC+
StLAuQHMS6RkB4aDNdi6QG9kEWjq2PvfuMlBWo5jJ8RFoDSx8q5t1ukgeCxr6xr
Q4eTmOFTlA71G312Xa7ZniOzyxiWZ4CAbhHLF+3baFD3lb4/EFmRvPBdqy6wUyHD
Z5EXyHDzl4XlDyEe/aomEqAsUqPs8MZirHHzpbaS3LbG5B5VKAKU59bENpf/KOgT
a3lUAeQ1t6xLzgToVdfhEkPj5bxODrWcZtHeTEt1nV+3pc2P58+QICDOETiDCA/j
dhG2brUwbxny6Ap7fU5e1ALU3ryoXKvt9eCXZHooY/p9QIC3koHCWptGD6gKCxlt
KW/K5M+HkxhHy4V7Wb137CStzeLda8BdU43Kh0ZQWWjK7pDXKKhHLYIGlawRScQa
e6J+y4JR1KKyXiXY94Erxa/P0FzuYV/QCJUDpqWFR22bXuy4FhkosLWM8G+UBHVt
UfgRxq8as60DhBDWy08eLEAdE92TVffJgXOvAOzTqBrP7uZi/Q7ABFFGTQ9n
=N4CJ
-----END PGP MESSAGE-----
```

Thanks,

Fingerprint Examples

What caused those fingerprints to hit?

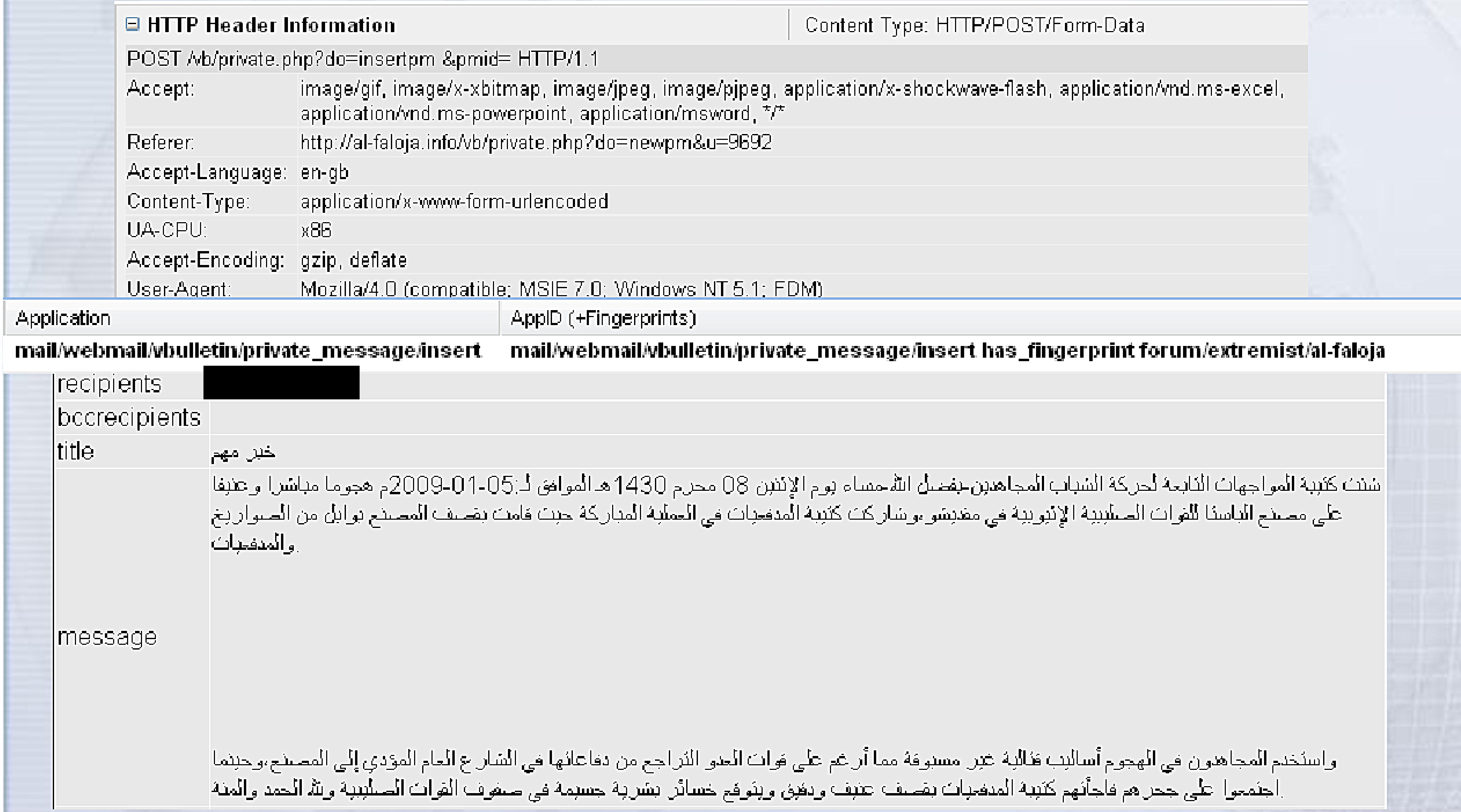


Application	AppID (+Fingerprints)
mail/webmail/outblaze	mail/webmail/outblaze has_fingerprint encryption/pgp encryption/pgp/message

Look at the definitions (notice any overlap?):

```
fingerprint('encryption/pgp') =
    'begin pgp message' or 'begin+pgp+message';

fingerprint('encryption/pgp/message‘)=
    /(?:BEGIN|END) PGP MESSAGE/;
```

Ex: Extremist Forum Private Messages



⊟ HTTP Header Information	Content Type: HTTP/POST/Form-Data
POST /vb/private.php?do=insertpm &pmid= HTTP/1.1	
Accept:	image/gif, image/x-xbitmap, image/jpeg, image/pjpeg, application/x-shockwave-flash, application/vnd.ms-excel, application/vnd.ms-powerpoint, application/msword, */*
Referer:	http://al-faloja.info/vb/private.php?do=newpm&u=9692
Accept-Language:	en-gb
Content-Type:	application/x-www-form-urlencoded
UA-CPU:	x86
Accept-Encoding:	gzip, deflate
User-Agent:	Mozilla/4.0 (compatible; MSIE 7.0; Windows NT 5.1; FDM)

Application	AppID (+Fingerprints)
mail/webmail/vbulletin/private_message/insert	**mail/webmail/vbulletin/private_message/insert has_fingerprint forum/extremist/al-faloja**

recipients	[redacted]
bccrecipients	
title	خبر مهم
message	شنت كتيبة المواجهات التابعة لحركة الشباب المجاهدين بفضل الله مساء يوم الإثنين 08 محرم 1430 هـ الموافق لـ:05-01-2009م هجوما مباشرا وعنيفا على مصنع الباستا للقوات الصليبية الإثيوبية في مقديشو، وشاركت كتيبة المدفعيات في العملية المباركة حيث قامت بقصف المصنع بوابل من الصواريخ والمدفعيات. واستخدم المجاهدون في الهجوم أساليب قتالية غير مسبوقة مما أرغم على قوات العدو التراجع من دفاعاتها في الشارع العام المؤدي إلى المصنع، وحينما اجتمعوا على جحرهم فاجأتهم كتيبة المدفعيات بقصف عنيف ودقيق ويتوقع خسائر بشرية جسيمة في صفوف القوات الصليبية ولله الحمد والمنة.

AppID vs Fingerprint

- AppIDs and Fingerprints use the exact same language inside of XKS.

- You can tell which one it is by the definition:

appid (mail/webmail/yahoo)

fingerprint (encryption/pgp)

AppID/Fingerprint Language Evolution

- There have been 4 generations of XKS AppID/Fingerprint languages
- 1st Generation: Simple Keyword Scanning
- 2nd Generation: Context Aware Keyword Scanning
- 3rd Generation: Code based AppIDs/Fingerprints
- 4th Generation: Code based AppIDs that can extract meta-data (also known as Micro Plugins)

1st Generation AppIDs/Fingerprints

- In the beginning, AppIDs and Fingerprints were just keyword scanning similar to CADENCE tasking Ex:

```
appid('mail/webmail/yahoo', 9.0) =
        'Host: mail.yahoo';
appid('mail/yahoo/login, 8.0) =
        ‘Host: mail.yahoo' and '/login';
```

1st Generation AppIDs/Fingerprints

- 1st Generation would also support Regular Expression (REGEX’s):

```
fingerprint('encryption/pgp/message‘)=
    /(?:BEGIN|END) PGP MESSAGE/;
```

(instead of quotes REGEX’s are enclosed by forward slashes)

1st Generation AppIDs/Fingerprints

- As well as Hex scanning:

```
appid('database/ms_sql_server(tds)/login', 7.5)=
    '\x06\x83\xf2\xf8\xff\x00\x00\x00\x00\xe0\
  x03\x00\x00\x88\xff\xff\xff\x36\x04\x00\x00';
```

(Hex characters are prefaced by \x)

2nd Generation AppIDs/Fingerprints

- 2nd Generation AppIDs/Fingerprints introduced XKS's context sensitive scanning engine.
- For example, rather than scanning an entire session top to bottom to look for 'facebook.com' we can just use the dictionary context http_host to target the scan for the host field only.

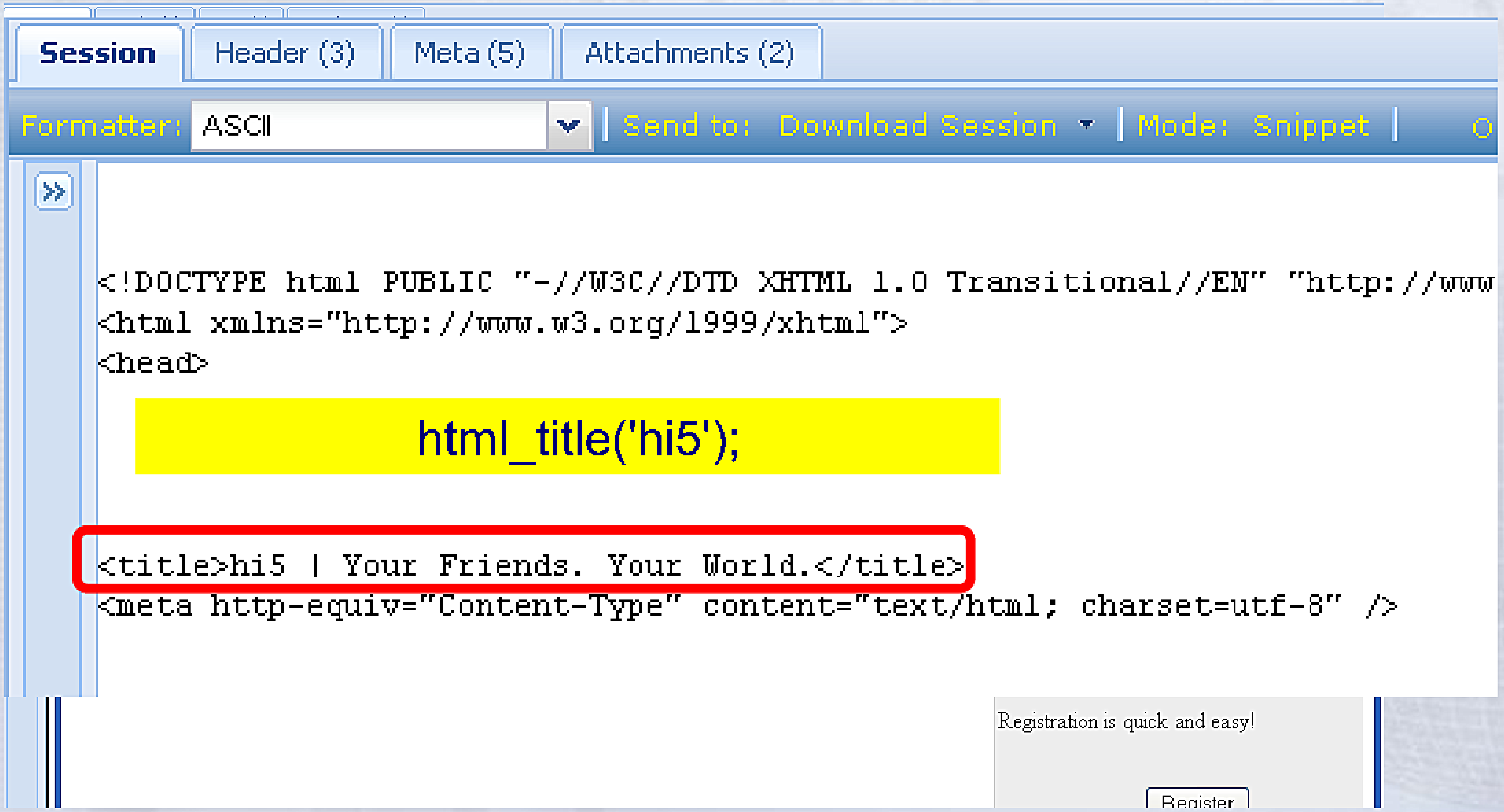
How do AppIDs work?

- AppID’s are effectively looking for keywords in order to assign the AppID tag.
- Example, this is the definition for Hi5

```
appid('mail/webmail/hi5', 6.0')=
                'hi5loggedIn'c or
                http_host('hi5.com') or
                html_title('hi5');
```

What do AppID's look like?

- If you look at the raw text of this traffic, one of the definitions for the mail/webmail/hi5 will hit:



Formatter: ASCII | Send to: Download Session | Mode: Snippet |

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
```

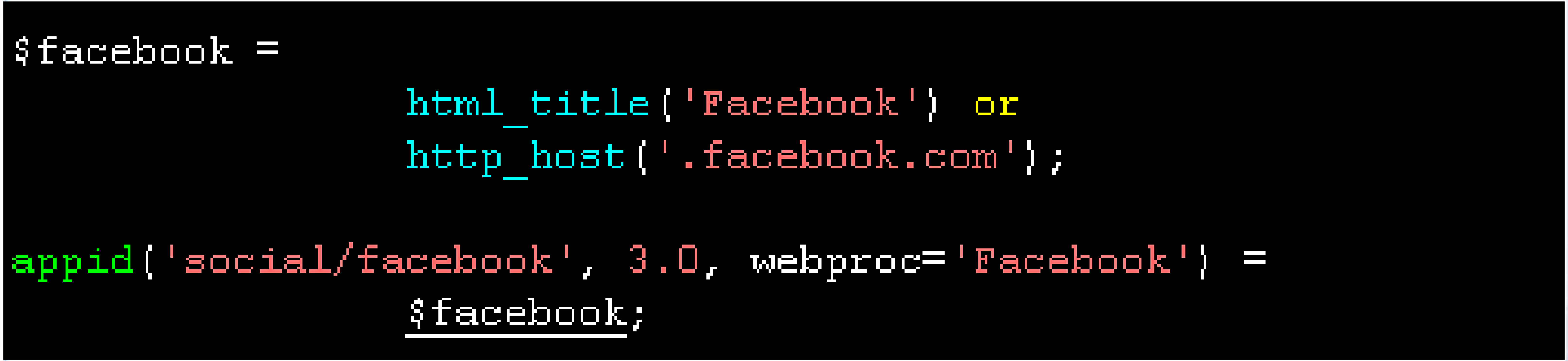
html_title('hi5');

```
<title>hi5 | Your Friends. Your World.</title>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
```

Registration is quick and easy!

2nd Generation AppIDs/Fingerprints

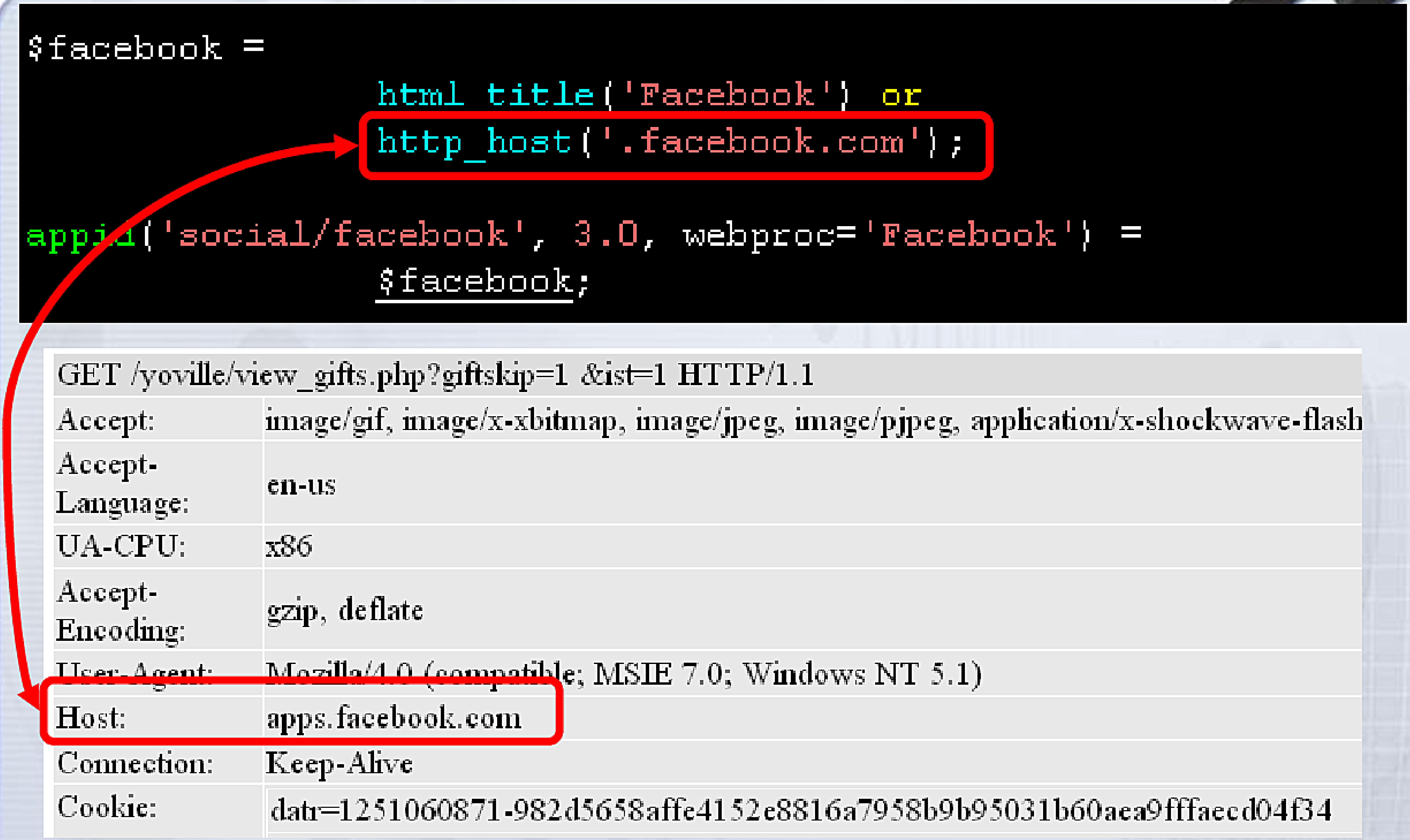
- Example:



```
$facebook =
                 html_title('Facebook') or
                 http_host('.facebook.com');

appid('social/facebook', 3.0, webproc='Facebook') =
                 $facebook;
```

Note the use of the chain word $facebook in the AppID definition

2nd Generation AppIDs/Fingerprints

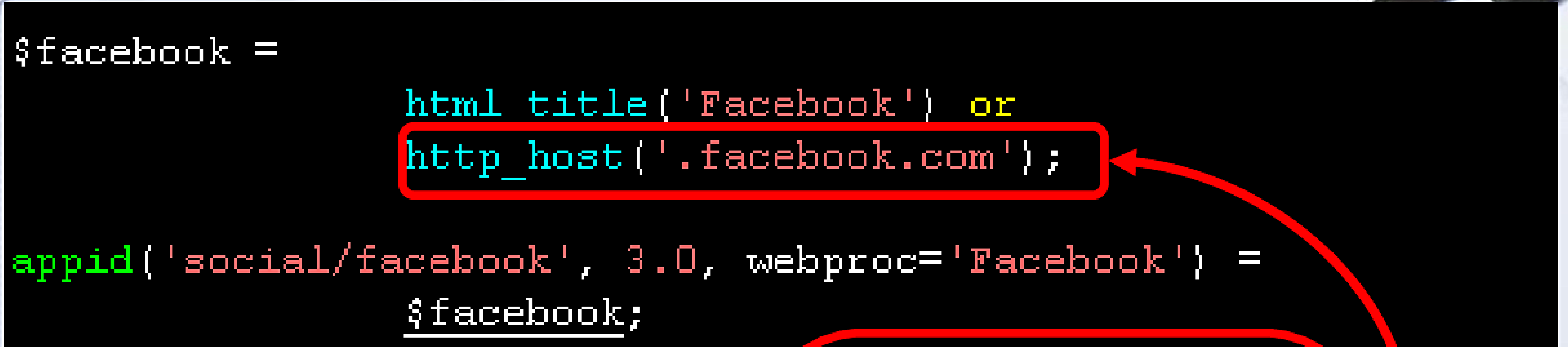


```
$facebook =
                    html_title('Facebook') or
                    http_host('.facebook.com');

appid('social/facebook', 3.0, webproc='Facebook') =
                    $facebook;
```

GET /yoville/view_gifts.php?giftskip=1 &ist=1 HTTP/1.1

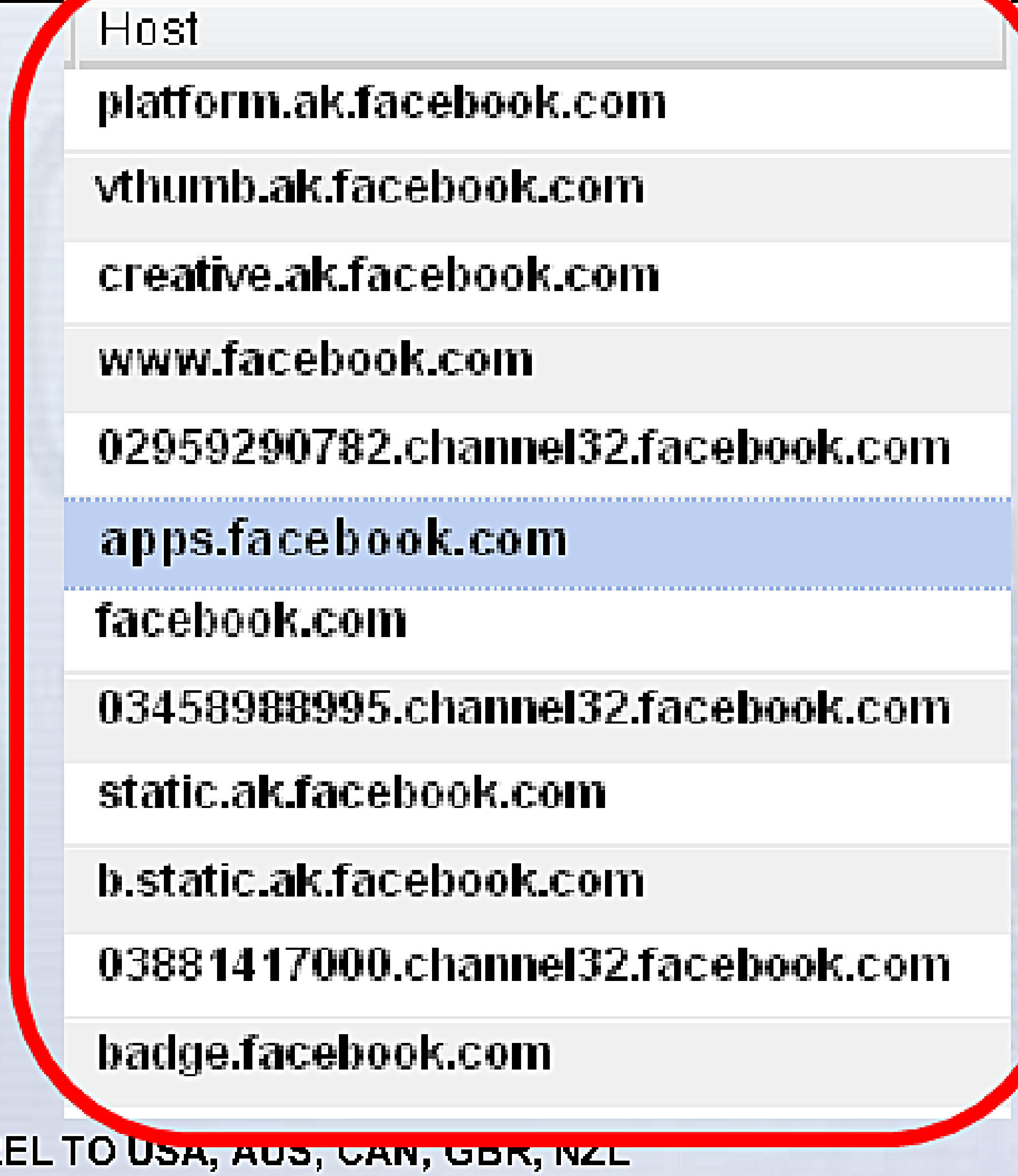
Accept:	image/gif, image/x-xbitmap, image/jpeg, image/pjpeg, application/x-shockwave-flash
Accept-Language:	en-us
UA-CPU:	x86
Accept-Encoding:	gzip, deflate
User-Agent:	Mozilla/4.0 (compatible; MSIE 7.0; Windows NT 5.1)
Host:	apps.facebook.com
Connection:	Keep-Alive
Cookie:	datr=1251060871-982d5658affe4152e8816a7958b9b95031b60aea9fffaecd04f34

2nd Generation AppIDs/Fingerprints



```
$facebook =
                 html_title('Facebook') or
                 http_host('.facebook.com');

appid('social/facebook', 3.0, webproc='Facebook') =
                 $facebook;
```

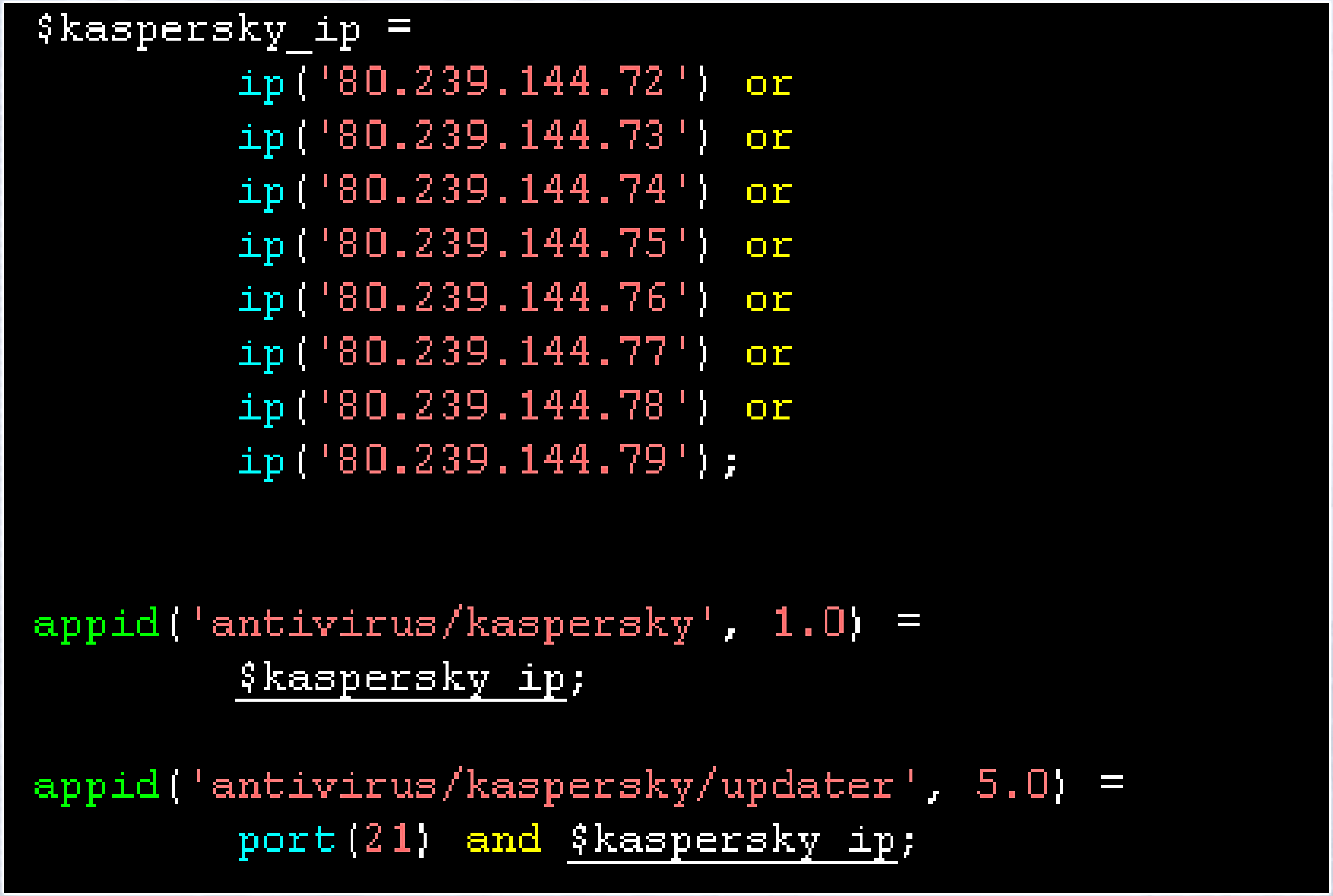
All of these hosts would match this AppID:



Host
platform.ak.facebook.com
vthumb.ak.facebook.com
creative.ak.facebook.com
www.facebook.com
02959290782.channel32.facebook.com
apps.facebook.com
facebook.com
03458988995.channel32.facebook.com
static.ak.facebook.com
b.static.ak.facebook.com
03881417000.channel32.facebook.com
badge.facebook.com

2nd Generation AppIDs/Fingerprints

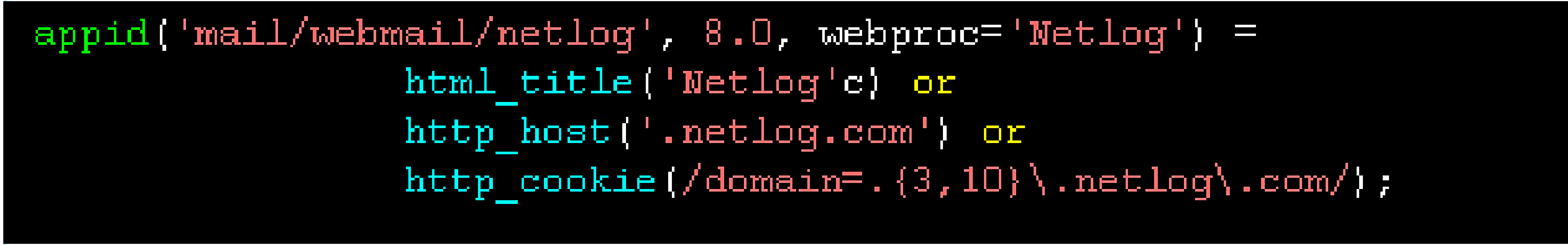
- Example:



```
$kaspersky_ip =
        ip('80.239.144.72') or
        ip('80.239.144.73') or
        ip('80.239.144.74') or
        ip('80.239.144.75') or
        ip('80.239.144.76') or
        ip('80.239.144.77') or
        ip('80.239.144.78') or
        ip('80.239.144.79');


appid('antivirus/kaspersky', 1.0) =
        $kaspersky_ip;

appid('antivirus/kaspersky/updater', 5.0) =
        port(21) and $kaspersky_ip;
```

2nd Generation AppIDs/Fingerprints

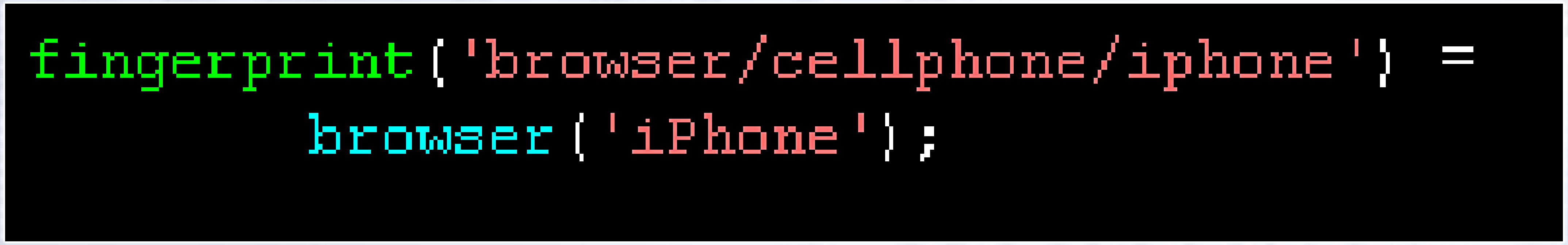
- Can you tell what's going on here?



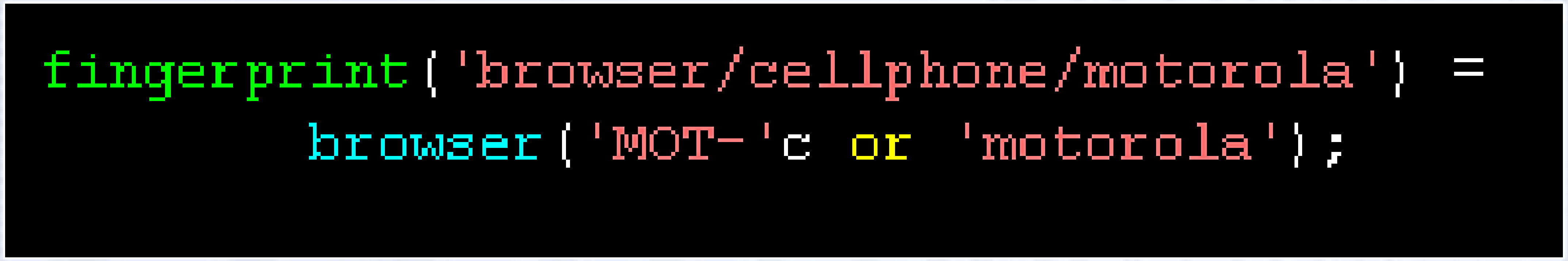
```
appid('mail/webmail/netlog', 8.0, webproc='Netlog') =
                html_title('Netlog'c) or
                http_host('.netlog.com') or
                http_cookie(/domain=.{3,10}\.netlog\.com/);
```

2nd Generation AppIDs/Fingerprints

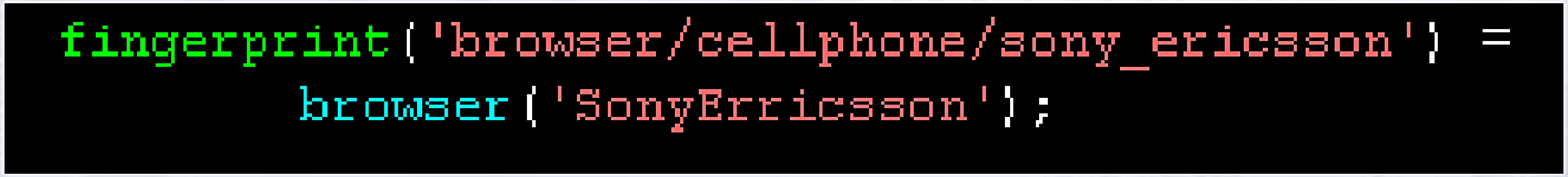
- Mobile User Agent fingerprints:



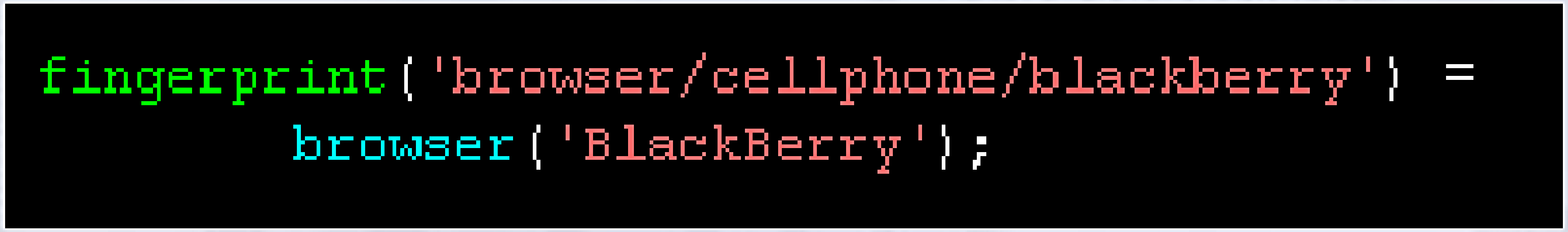
```
fingerprint('browser/cellphone/iphone') =
        browser('iPhone');
```



```
fingerprint('browser/cellphone/motorola') =
       browser('MOT-'c or 'motorola');
```



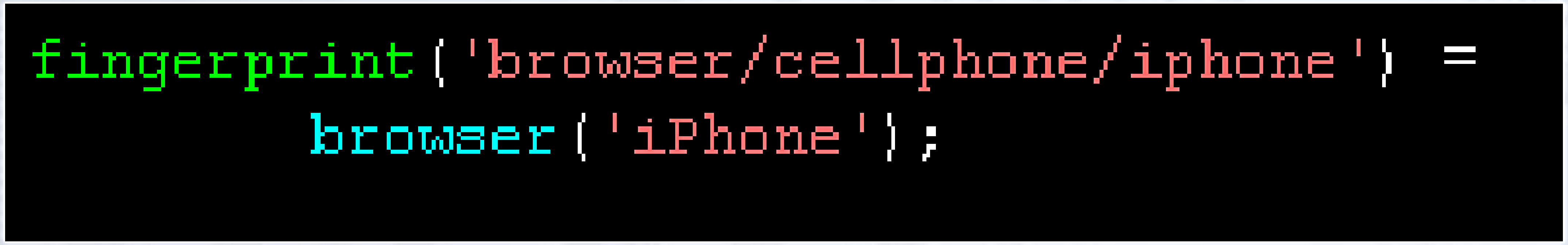
```
fingerprint('browser/cellphone/sony_ericsson') =
       browser('SonyErricsson');
```



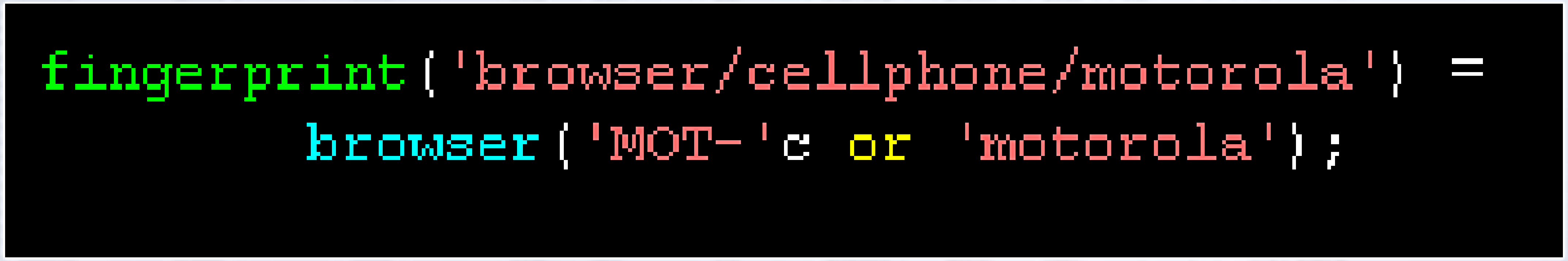
```
fingerprint('browser/cellphone/blackberry') =
       browser('BlackBerry');
```

USSID18 Considerations!

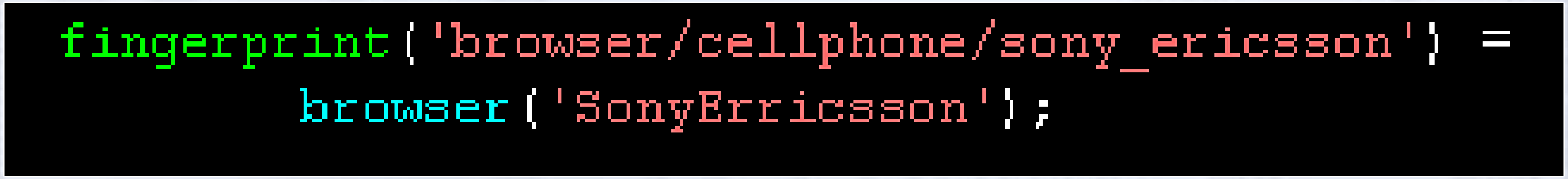
- If you were to query on any of these fingerprints by themselves, would your auditor be happy?



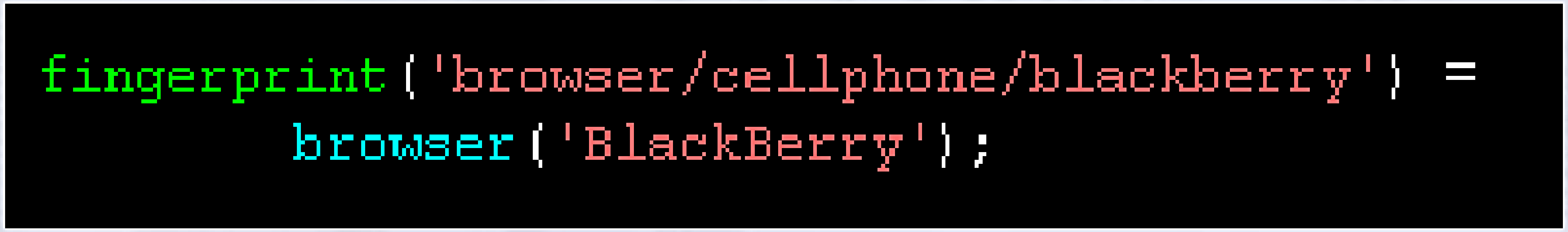
```
fingerprint('browser/cellphone/iphone') =
        browser('iPhone');
```



```
fingerprint('browser/cellphone/motorola') =
        browser('MOT-'c or 'motorola');
```



```
fingerprint('browser/cellphone/sony_ericsson') =
        browser('SonyErricsson');
```



```
fingerprint('browser/cellphone/blackberry') =
        browser('BlackBerry');
```

USSID18 Considerations!

- But if you were to query on an Afghan IP address that was a valid foreign intel target, and then "AND" it with those fingerprints, that would be a USSID18 compliant query (and your auditor would be happy)

3rd Generation AppIDs/Fingerprints

- 3rd Generation AppIDs/Fingerprints introduced the ability to have code-based scanning
- Why is this important? Because scanning sessions for keywords, hex values and regular expression can only take you so far.
- Using Code-based AppIDs, we can run statistical tests of the data that can help determine what type of data it is when keyword scanning can’t give us a result.

4th Generation AppIDs/Fingerprints

- 4th Generation AppIDs/Fingerprints introduce the ability to extract and database meta-data from Appid/Fingerprints
- Why is this important?
- With the dynamic nature of DNI applications, we need the ability to quickly react and deploy solutions to extract new fields of meta-data that are important to analysts

4th Generation AppIDs/Fingerprints

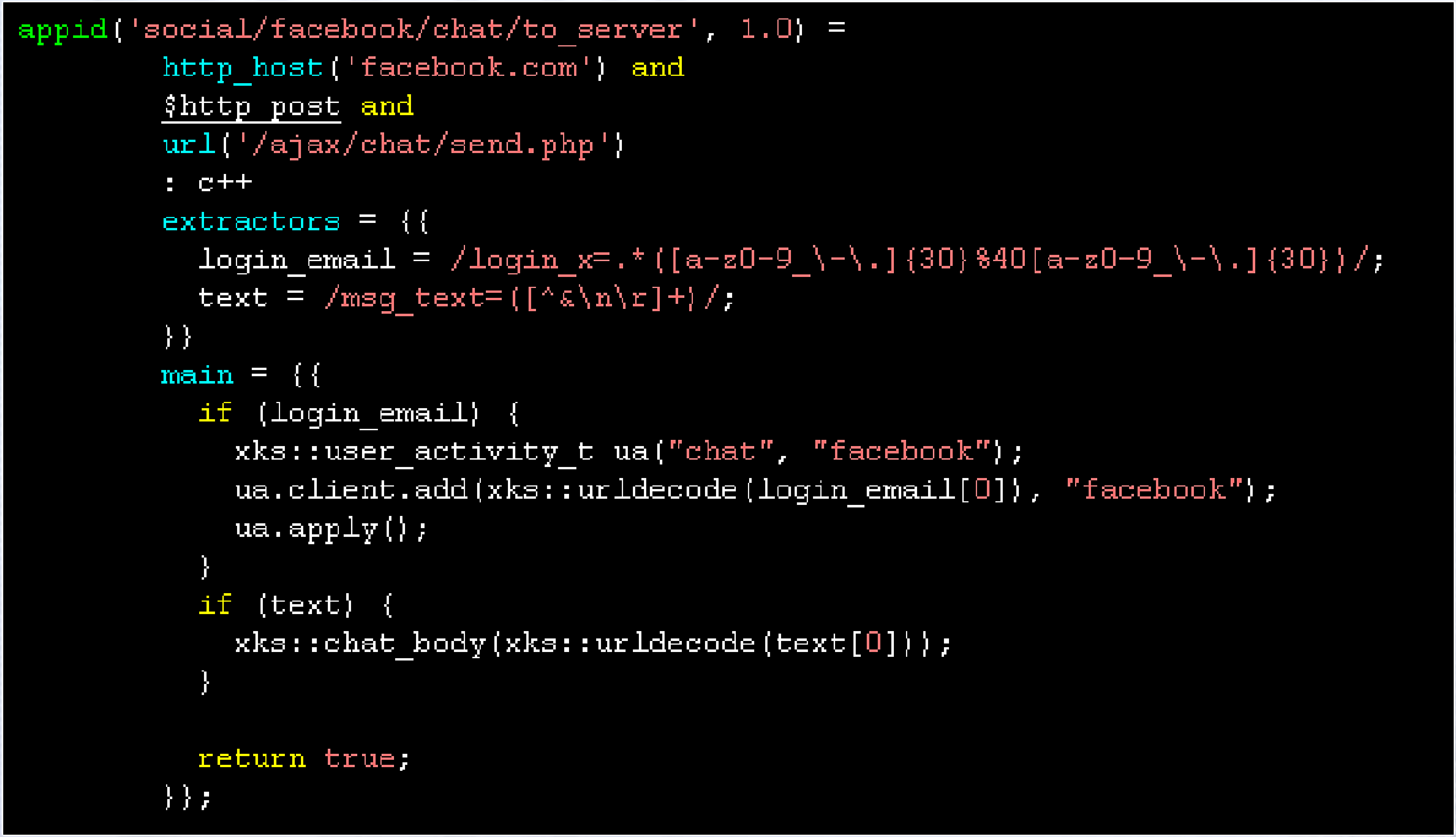
- Previously, if we identified a new protocol or a new field that we wanted to extract meta-data, we would need to upgrade a "core" plug-in and wait until we could upgrade the field sites.
- With 130 field sites, each on their own upgrade schedule, this could take months for a simple change to get out in the field

4th Generation AppIDs/Fingerprints

- With 4th generation AppIDs, a new protocol, meta-data value, can be properly processed within an hour of updating the AppID/Fingerprint.

4th Generation AppIDs/Fingerprints

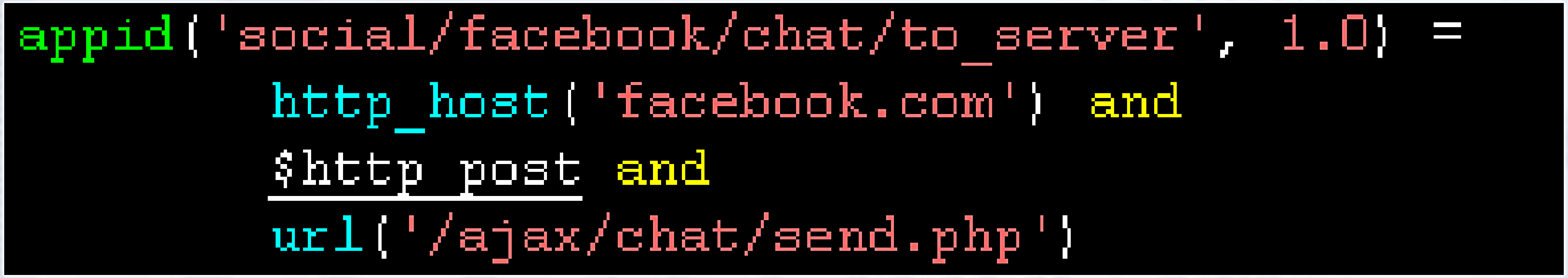
- Examples:



```
appid('social/facebook/chat/to_server', 1.0) =
        http_host('facebook.com') and
        $http_post and
        url('/ajax/chat/send.php')
        : c++
        extractors = {{
          login_email = /login_x=.*([a-z0-9_\-\.]{30}%40[a-z0-9_\-\.]{30})/;
          text = /msg_text=([^&\n\r]+)/;
        }}
        main = {{
          if (login_email) {
            xks::user_activity_t ua("chat", "facebook");
            ua.client.add(xks::urldecode(login_email[0]), "facebook");
            ua.apply();
          }
          if (text) {
            xks::chat_body(xks::urldecode(text[0]));
          }

          return true;
        }};
```

Facebook Chat V4 Appid Example

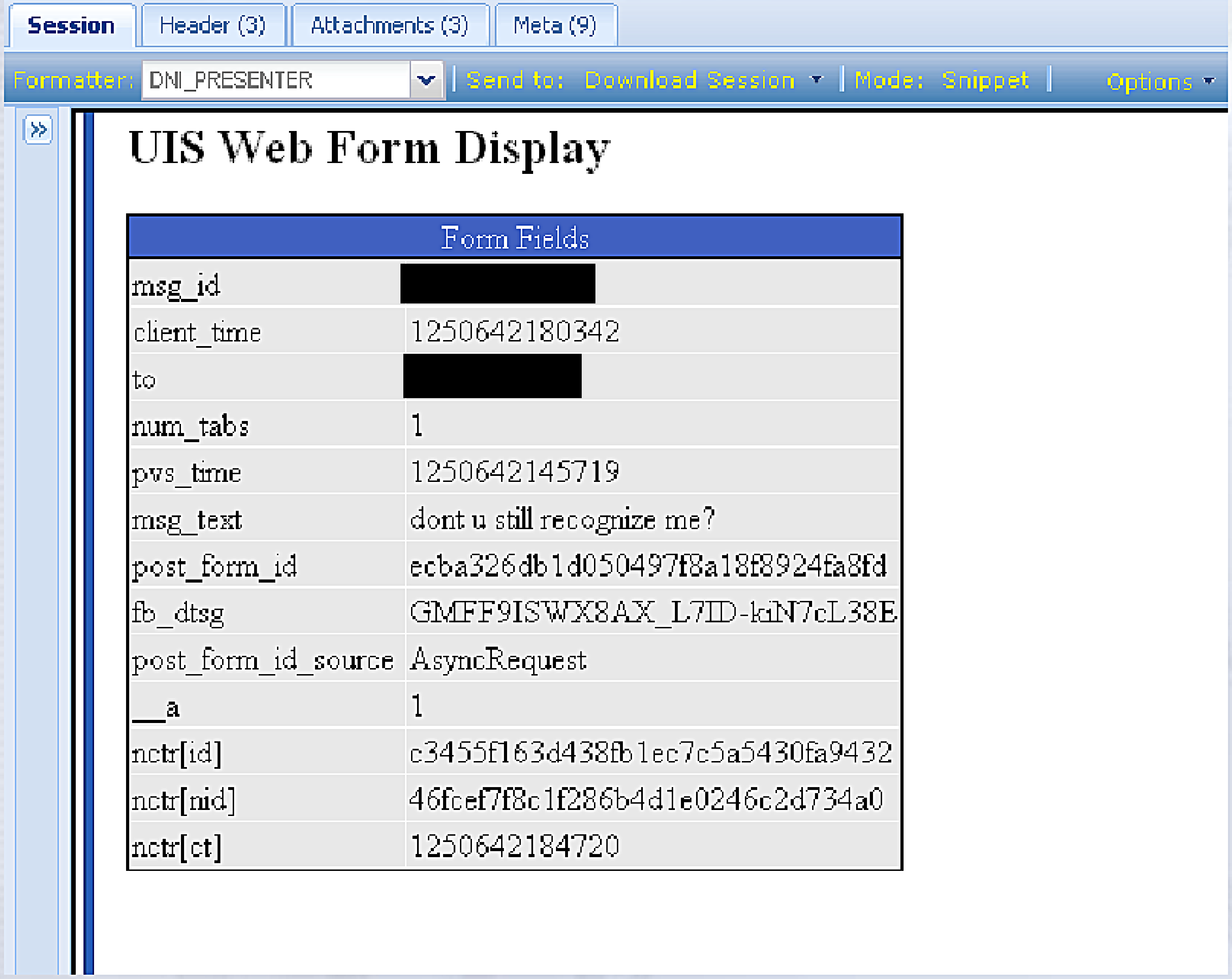
- Let’s take a closer look:
- First a V4 AppID needs to be “anchored”. The anchor is the beginning part of the AppID



```
appid('social/facebook/chat/to_server', 1.0) =
        http_host('facebook.com') and
        $http_post and
        url('/ajax/chat/send.php')
```

Facebook Chat V4 Appid Example

- DNI Presenter Display:

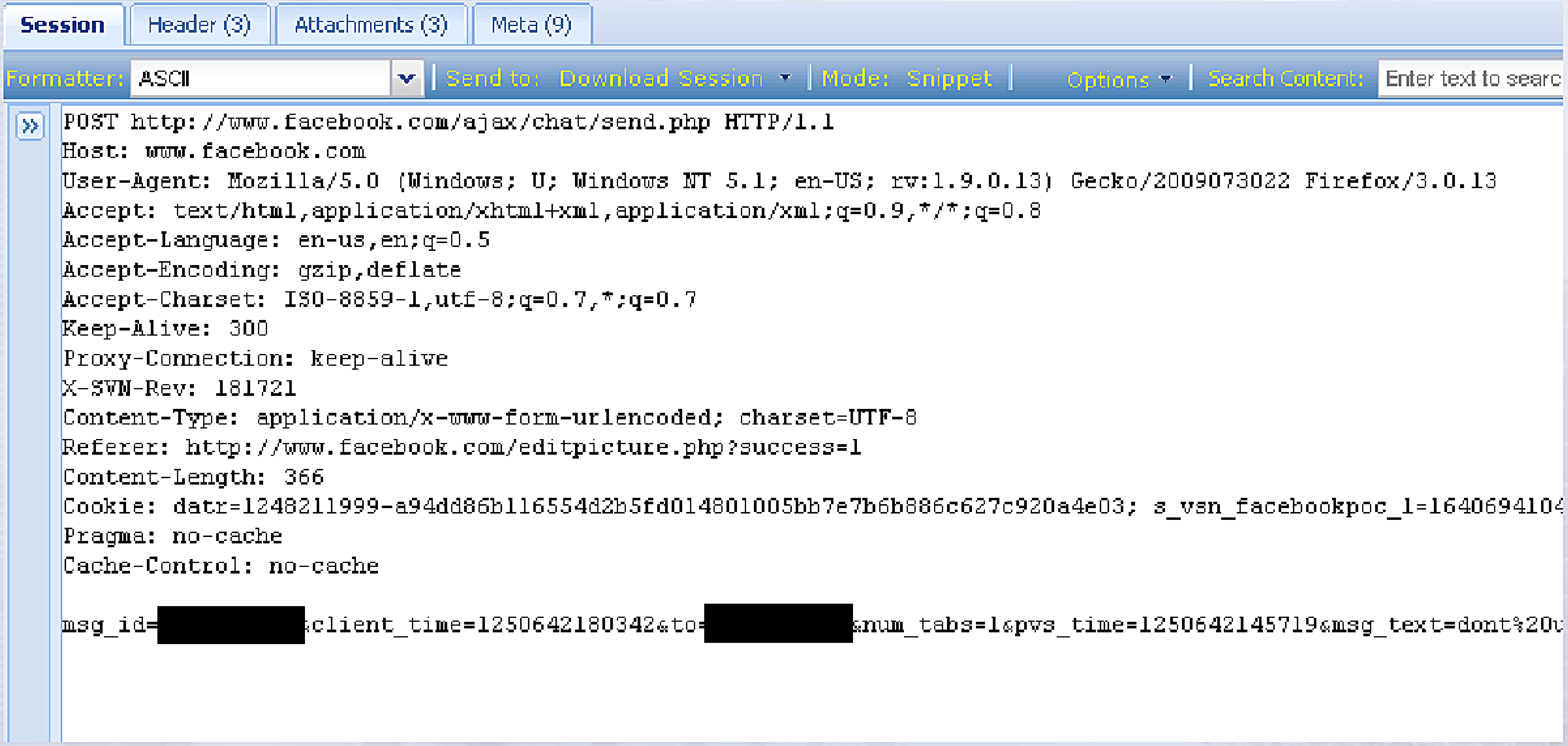


UIS Web Form Display

Form Fields	
msg_id	[redacted]
client_time	1250642180342
to	[redacted]
num_tabs	1
pvs_time	1250642145719
msg_text	dont u still recognize me?
post_form_id	ecba326db1d050497f8a18f8924fa8fd
fb_dtsg	GMFF9ISWX8AX_L7ID-kiN7cL38E
post_form_id_source	AsyncRequest
__a	1
nctr[id]	c3455f163d438fb1ec7c5a5430fa9432
nctr[nid]	46fcef7f8c1f286b4d1e0246c2d734a0
nctr[ct]	1250642184720

Facebook Chat V4 Appid Example

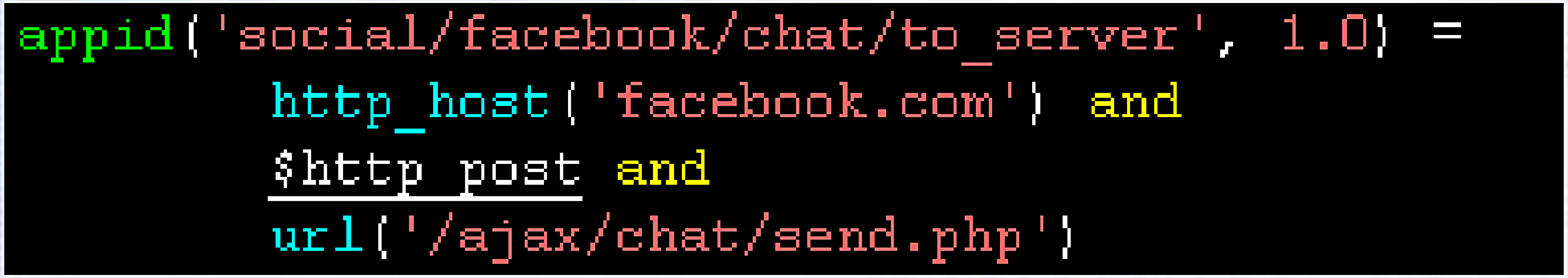
- Lets look at the raw:



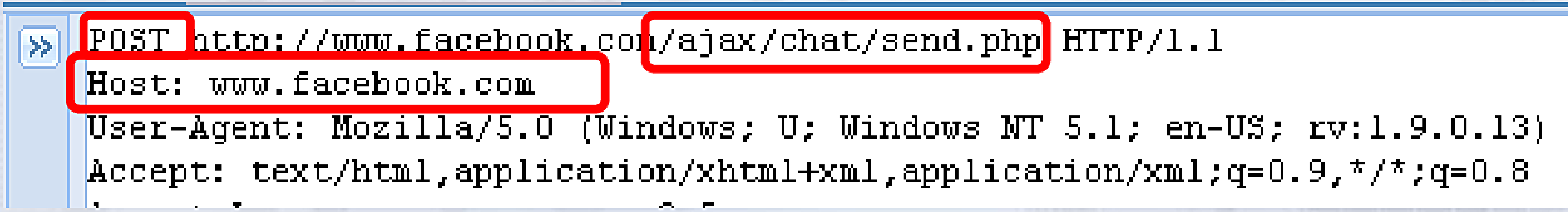
```
POST http://www.facebook.com/ajax/chat/send.php HTTP/1.1
Host: www.facebook.com
User-Agent: Mozilla/5.0 (Windows; U; Windows NT 5.1; en-US; rv:1.9.0.13) Gecko/2009073022 Firefox/3.0.13
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-us,en;q=0.5
Accept-Encoding: gzip,deflate
Accept-Charset: ISO-8859-1,utf-8;q=0.7,*;q=0.7
Keep-Alive: 300
Proxy-Connection: keep-alive
X-SVN-Rev: 181721
Content-Type: application/x-www-form-urlencoded; charset=UTF-8
Referer: http://www.facebook.com/editpicture.php?success=1
Content-Length: 366
Cookie: datr=1248211999-a94dd86b116554d2b5fd014801005bb7e7b6b886c627c920a4e03; s_vsn_facebookpoc_1=1640694104
Pragma: no-cache
Cache-Control: no-cache

msg_id=[redacted]&client_time=1250642180342&to=[redacted]&num_tabs=1&pvs_time=1250642145719&msg_text=dont%20u
```

Facebook Chat V4 Appid Example

- The “anchor” of this V4 AppID was present:



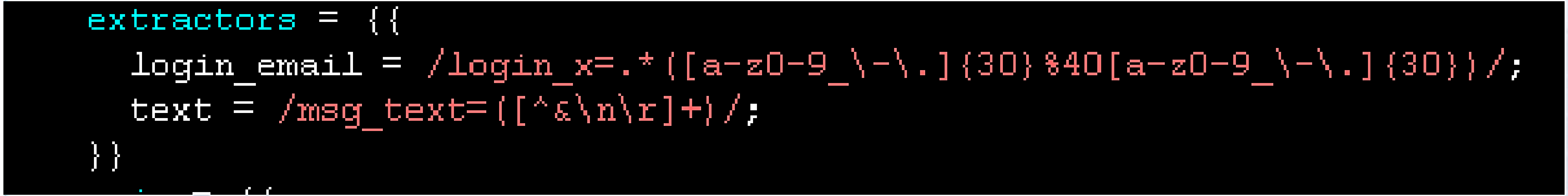
```
appid('social/facebook/chat/to_server', 1.0) =
        http_host('facebook.com') and
        $http_post and
        url('/ajax/chat/send.php')
```



```
POST http://www.facebook.com/ajax/chat/send.php HTTP/1.1
Host: www.facebook.com
User-Agent: Mozilla/5.0 (Windows; U; Windows NT 5.1; en-US; rv:1.9.0.13)
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
```

Facebook Chat V4 Appid Example

- Once the “anchor” hits, the rest of the code executes. In this case, we’re looking for these two REGEX’s from the “Extractors” section:



```
extractors = {{
  login_email = /login_x=.*([a-z0-9_\-\.]{30}%40[a-z0-9_\-\.]{30})/;
  text = /msg_text=([^&\n\r]+)/;
}}
```

Facebook Chat V4 Appid Example

This REGEX hits within the large cookie string

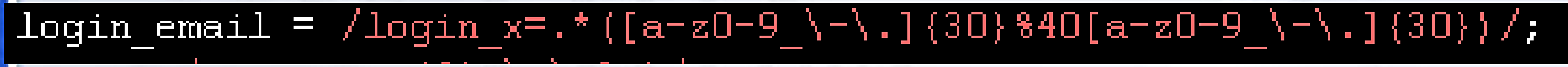
```
login_email = /login_x=.*([a-z0-9_\-\.]{30}%40[a-z0-9_\-\.]{30})/;
```

Cookie:

[redacted]

Facebook Chat V4 Appid Example

A close look

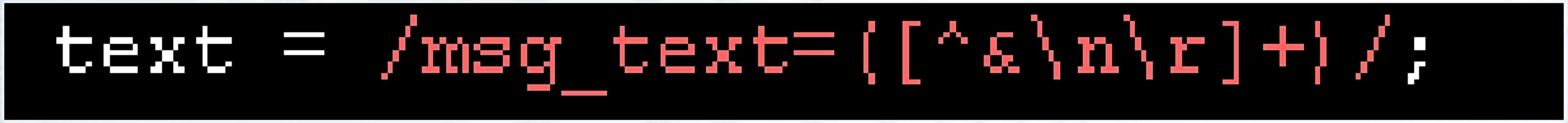


```
login_email = /login_x=.*([a-z0-9_\-\.]{30}%40[a-z0-9_\-\.]{30})/;
```

login_x=a%3A2%3A%7Bs%3A5%3A%22email%22%3Bs%3A26%3A%22[redacted]%40yahoo.com%22%3Bs%3A19%3A%22remember_me_default%22%3Bb%3A1%3B%7D;

Facebook Chat V4 Appid Example

- The other REGEX:

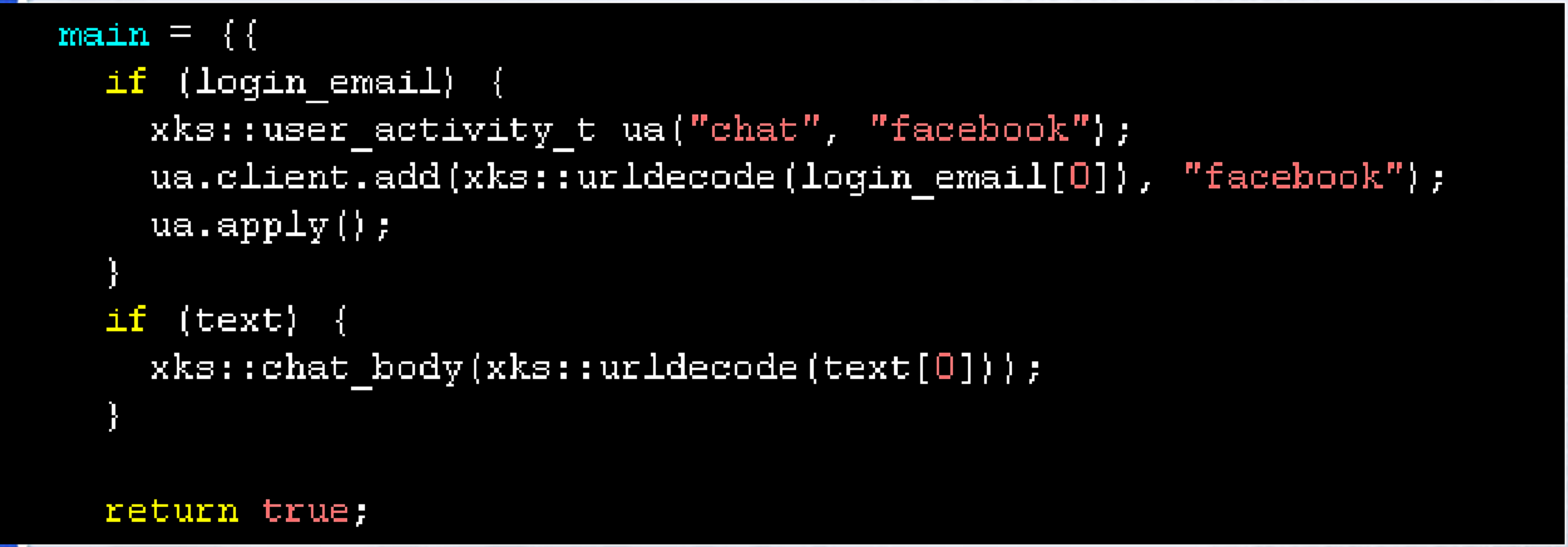


```
text = /msg_text=([^&\n\r]+)/;
```

- msg_text=dont%20u%20still%20recognize%20me%3F&post_form_id

Facebook Chat V4 Appid Example

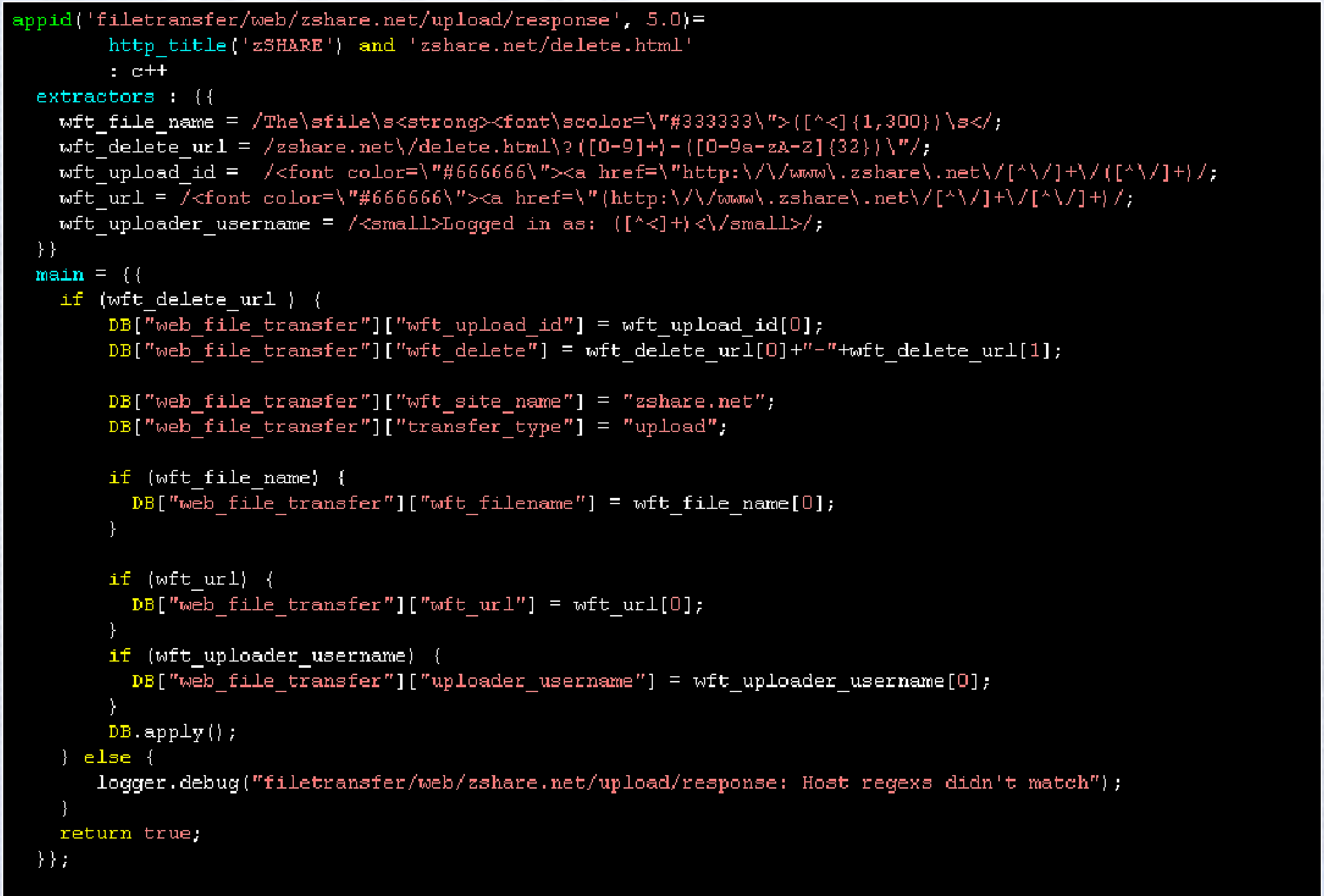
- Finally, in the “Main” section, if those REGEX’s found the data they were looking for, they get databased



```
main = {{
  if (login_email) {
    xks::user_activity_t ua("chat", "facebook");
    ua.client.add(xks::urldecode(login_email[0]), "facebook");
    ua.apply();
  }
  if (text) {
    xks::chat_body(xks::urldecode(text[0]));
  }

  return true;
```

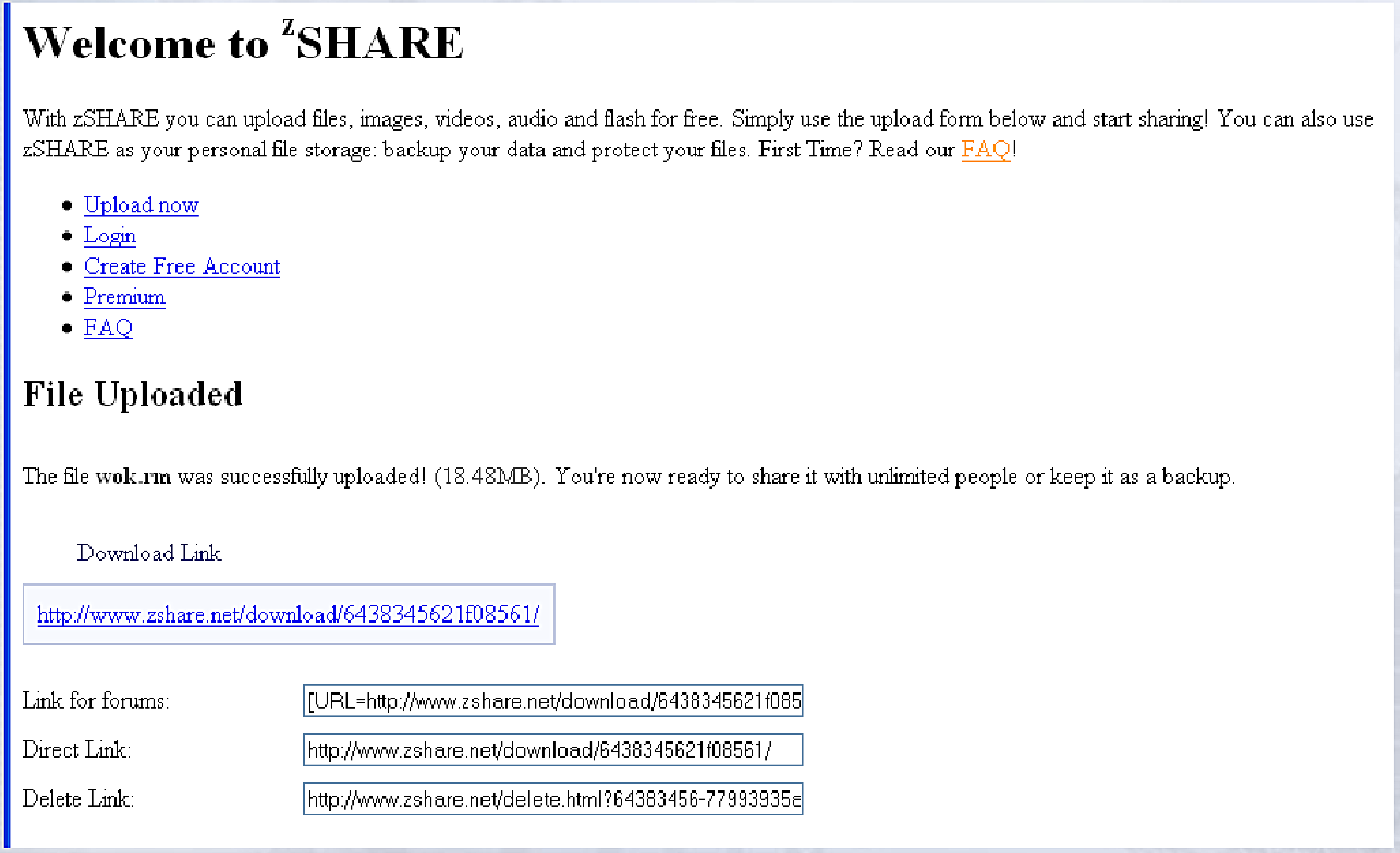
4th Generation AppIDs/Fingerprints

- Another example:



```
appid('filetransfer/web/zshare.net/upload/response', 5.0)=
        http_title('zSHARE') and 'zshare.net/delete.html'
        : c++
  extractors : {{
    wft_file_name = /The\sfile\s<strong><font\scolor=\"#333333\">([^<]{1,300})\s</;
    wft_delete_url = /zshare.net\/delete.html\?([0-9]+)-([0-9a-zA-Z]{32})\"/;
    wft_upload_id =  /<font color=\"#666666\"><a href=\"http:\/\/www\.zshare\.net\/[^\/]+\/([^\/]+)/;
    wft_url = /<font color=\"#666666\"><a href=\"(http:\/\/www\.zshare\.net\/[^\/]+\/[^\/]+)/;
    wft_uploader_username = /<small>Logged in as: ([^<]+)<\/small>/;
  }}
  main = {{
    if (wft_delete_url ) {
        DB["web_file_transfer"]["wft_upload_id"] = wft_upload_id[0];
        DB["web_file_transfer"]["wft_delete"] = wft_delete_url[0]+"-"+wft_delete_url[1];

        DB["web_file_transfer"]["wft_site_name"] = "zshare.net";
        DB["web_file_transfer"]["transfer_type"] = "upload";

        if (wft_file_name) {
          DB["web_file_transfer"]["wft_filename"] = wft_file_name[0];
        }

        if (wft_url) {
          DB["web_file_transfer"]["wft_url"] = wft_url[0];
        }
        if (wft_uploader_username) {
          DB["web_file_transfer"]["uploader_username"] = wft_uploader_username[0];
        }
        DB.apply();
    } else {
       logger.debug("filetransfer/web/zshare.net/upload/response: Host regexs didn't match");
    }
    return true;
  }};
```

FFU Successful Upload Pages



Welcome to ᶻSHARE

With zSHARE you can upload files, images, videos, audio and flash for free. Simply use the upload form below and start sharing! You can also use zSHARE as your personal file storage: backup your data and protect your files. First Time? Read our FAQ!

- Upload now
- Login
- Create Free Account
- Premium
- FAQ

File Uploaded

The file **wok.rm** was successfully uploaded! (18.48MB). You're now ready to share it with unlimited people or keep it as a backup.

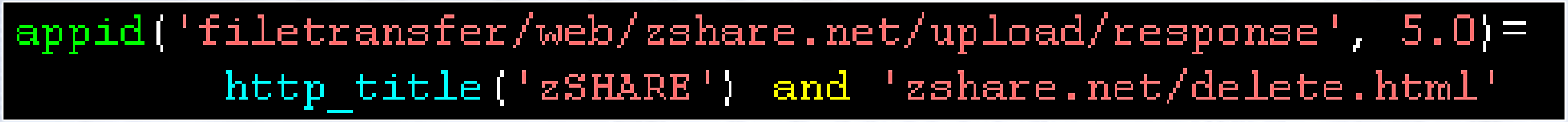
Download Link

http://www.zshare.net/download/6438345621f08561/

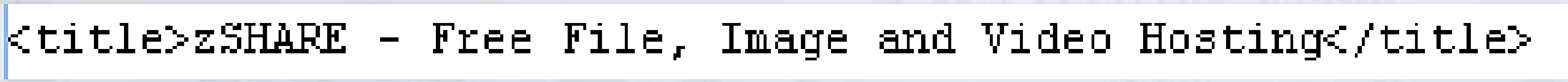
Link for forums:	[URL=http://www.zshare.net/download/6438345621f085
Direct Link:	http://www.zshare.net/download/6438345621f08561/
Delete Link:	http://www.zshare.net/delete.html?64383456-77993935e

FFU Successful Upload Pages

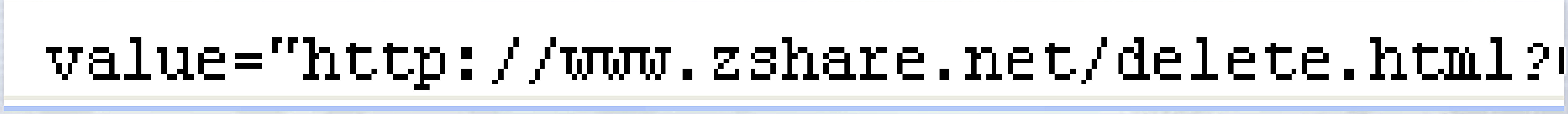
- Again look for the anchor to hit in the raw traffic



```
appid('filetransfer/web/zshare.net/upload/response', 5.0)=
       http_title('zSHARE') and 'zshare.net/delete.html'
```



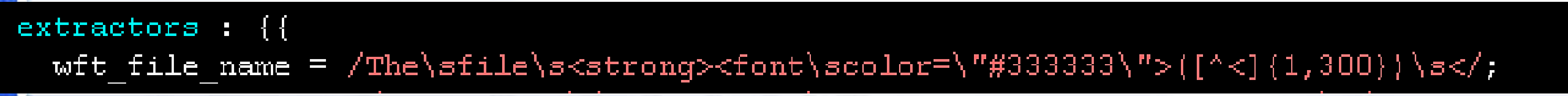
```
<title>zSHARE - Free File, Image and Video Hosting</title>
```



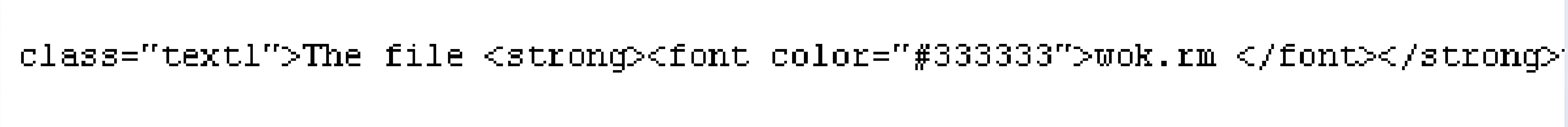
```
value="http://www.zshare.net/delete.html?
```

FFU Successful Upload Pages

- Next look for the extractor REGEX's to match

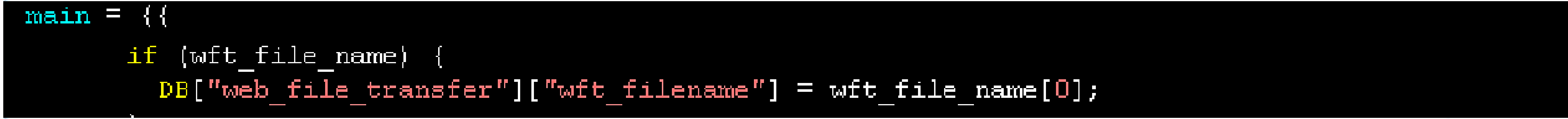


```
extractors : {{
  wft_file_name = /The\sfile\s<strong><font\scolor=\"#333333\">([^<]{1,300})\s</;
```



```
class="text1">The file <strong><font color="#333333">wok.rm </font></strong>
```

- Then database what was extracted



```
main = {{
       if (wft_file_name) {
         DB["web_file_transfer"]["wft_filename"] = wft_file_name[0];
```