Advanced HTTP Activity Analysis

2009

Goal

The goal of this training is to get you familiar with basic HTTP traffic and understand how to target and expliot it using X-KEYSCORE

Agenda

- ?

What is HTTP?

HTTP stands for Hypertext Transfer Protocol and it's the primary protocol for transferring data on the World Wide Web

Why are we interested in HTTP?

facebook

YAHOO!

twitter

myspace.com
a place for friends

Because nearly everything a typical user does on the Internet uses HTTP

CNN.com

Google Earth

@mail.ru

WIKIPEDIA
The Free Encyclopedia

Gmail by Google BETA

Why are we interested in HTTP?

- Almost all web-browsing uses HTTP:
  - Internet surfing
  - Webmail (Yahoo/Hotmail/Gmail/etc.)
  - OSN (Facebook/MySpace/etc.)
  - Internet Searching (Google/Bing/etc.)
  - Online Mapping (Google Maps/Mapquest/etc.)

How does HTTP work?

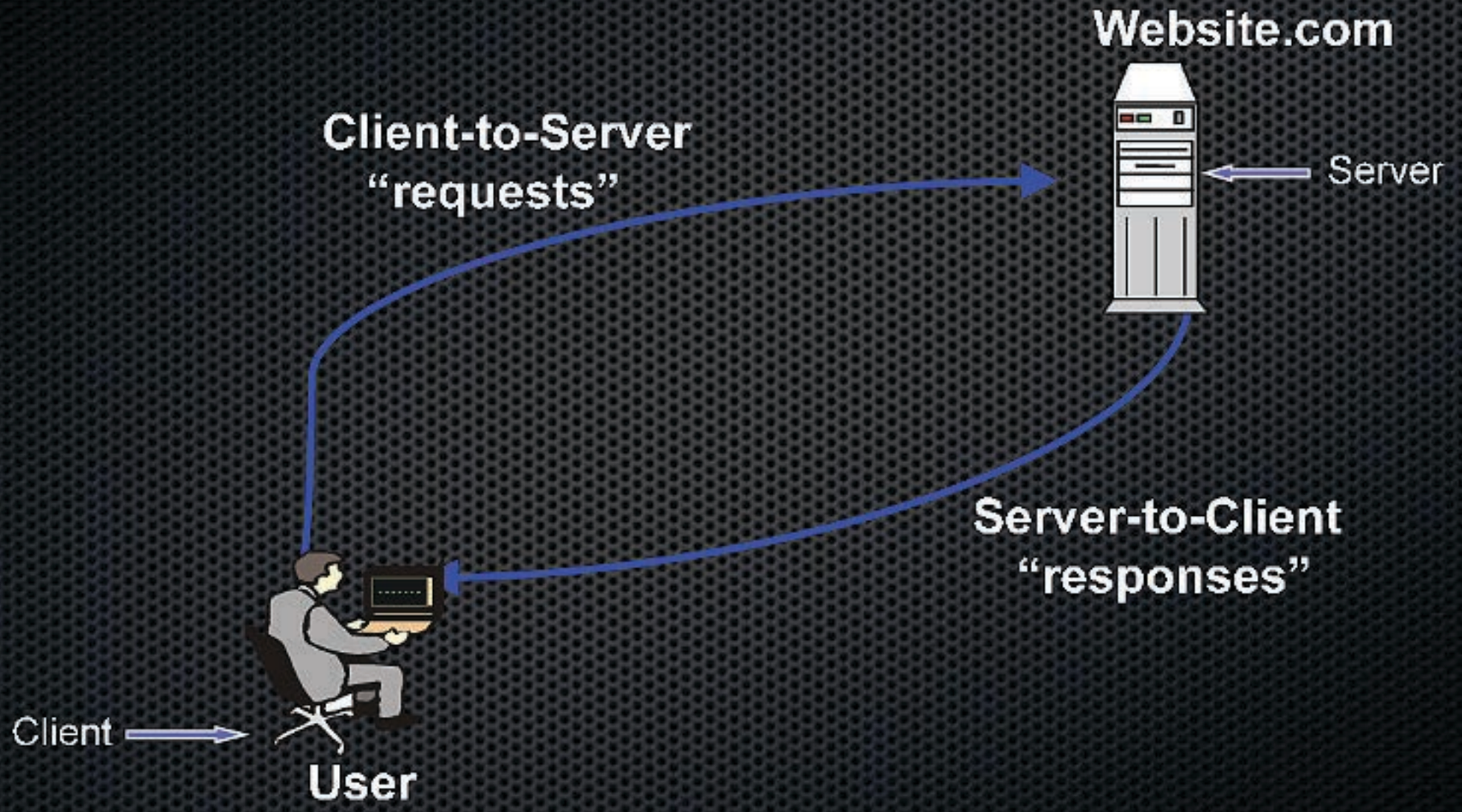
- HTTP is comprised of requests from clients to servers and their corresponding responses
- Many analysts are already familiar with the terms “client-to-server” or “server-to-client” collection (also referred to as “client side” or “server side” collection).

How does HTTP work?

- A “Client” is usually referring to a Browser (like Firefox or IE) which is also referred to as the “User Agent”
- The “Server” can also be referred to as the “web-server” or “origin-server” which is the machine that is storing the data that is being accessed (like a web-page, a map, an inbox, etc)

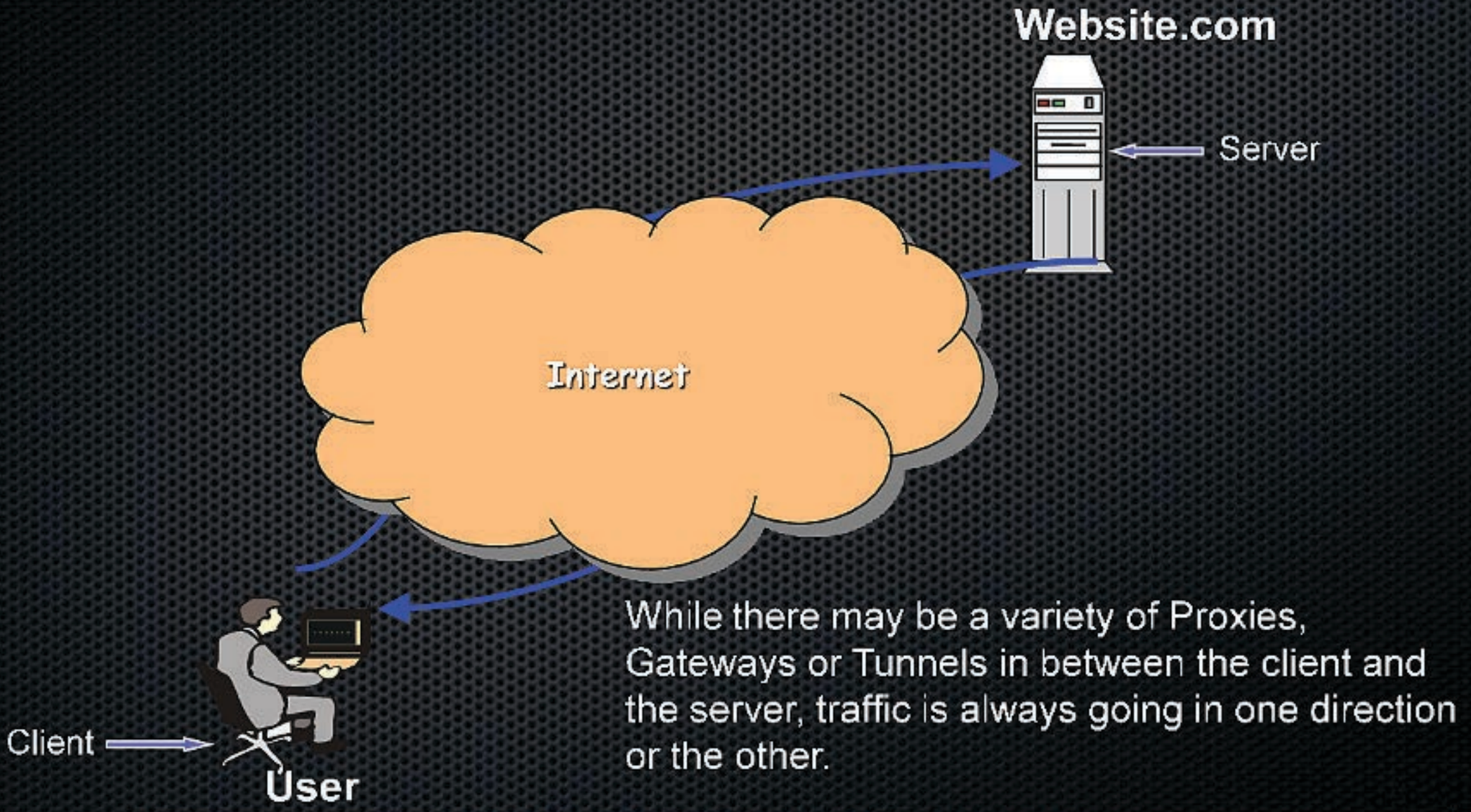
HTTP Activity

- HTTP activity comes in two types:



HTTP Activity

- HTTP activity comes in two types:



While there may be a variety of Proxies, Gateways or Tunnels in between the client and the server, traffic is always going in one direction or the other.

Client vs. Server Side Traffic

- How do you know which side you're looking at?
- Client-to-Server requests are generally small in size and are computers talking to other computers
- They contain standard HTTP header fields like "Host:" "Accept:" "Connection" etc.

HTTP Activity Examples

Client-to-Server request:

TOP SECRET//COMINT//20320108

ID: sess_orig_proc

Type: HTTP-GET | Printer Friendly Version

DNI Display | Raw Data | DNI Format

Services

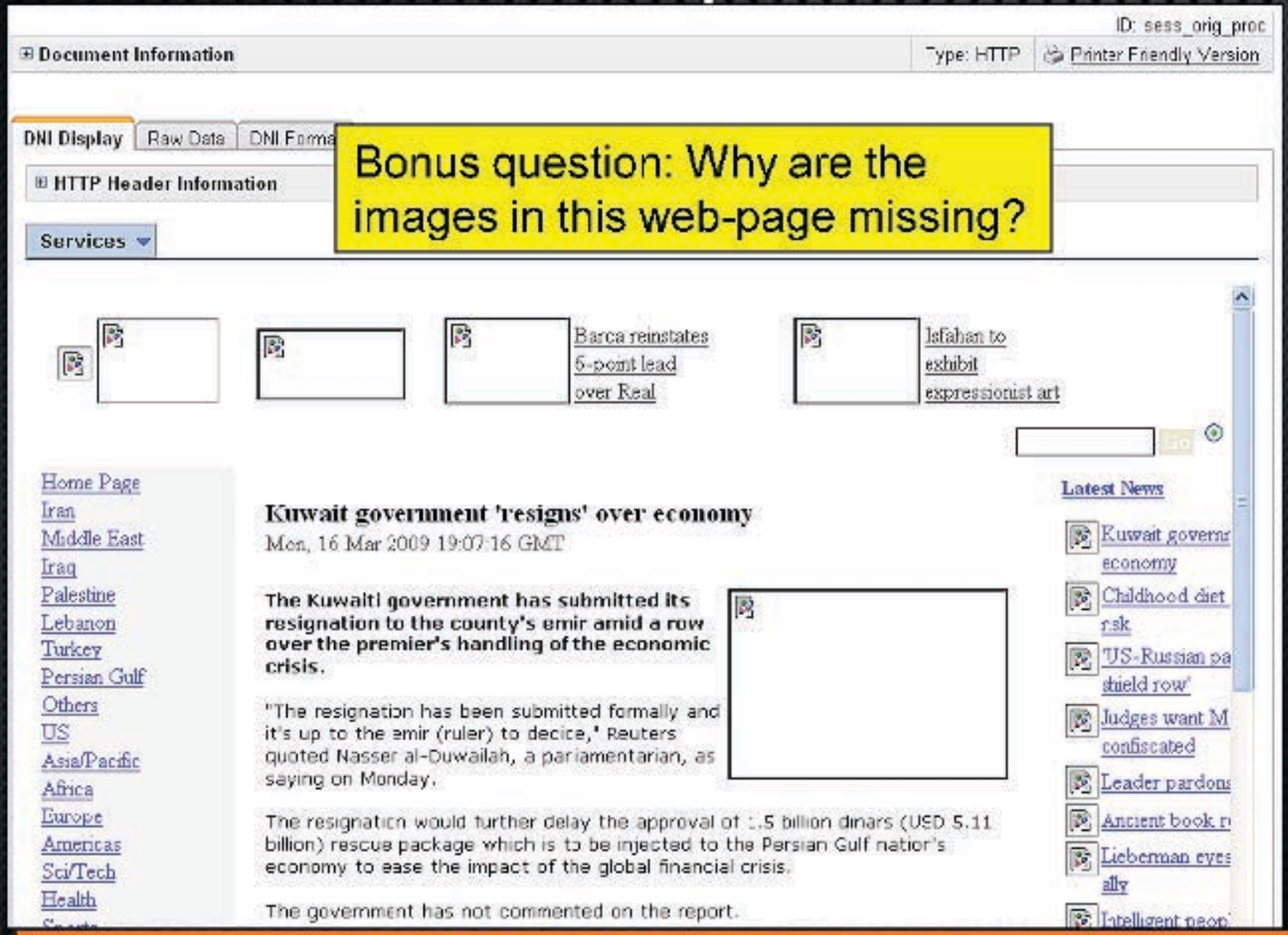
GET /Hezbollah-Terrorism-Judith-Palmer-Harik/dp/1860648932 HTTP/1.1	
User-Agent:	Mozilla/5.0 (Windows; U; Windows NT 5.1; en-US) AppleWebKit/525.19 (KHTML, like Gecko) Chrome/1.0.154.48 Safari/525.19
Referer:	http://www.google.com.pk/search?hl=en&q=wretten books on hizbollah&btnG=Google Search&meta=
Accept:	text/xml,application/xml,application/xhtml+xml,text/html;q=0.9,text/plain;q=0.8,image/png,*/*;q=0.5
Accept-Encoding:	gzip,deflate,bzip2,sdch
Cookie:	ubid-main=185-5525816-8765531
	apn-user-id=P1YXY7QF1PUYQ5
Accept-Language:	en-US,en
Accept-Charset:	ISO-8859-1,*,utf-8
Host:	www.amazon.com
Connection:	Keep-Alive

Client vs. Server Side Traffic

- Server-to-Client responses are generally larger in size and are what web-pages look like at the internet.
- When you're at a computer accessing the Internet, you're only seeing Server-to-Client traffic.

HTTP Activity Examples

Server-to-Client Response:



HTTP Activity

- XKS HTTP Activity Meta-data differs greatly depending on which side of traffic we're collecting
- In nearly all cases it's better to have client-to-server traffic

HTTP Activity Client-to-Server

```
GET /search?tab=urdu&order=sortboth&q=musharraf&start=3&scope=urdu&link=next HTTP/1.1
Accept: */*
Referer: http://search.bbc.co.uk/search?tab=urdu&order=sortboth&q=musharraf&start=2&scope=urdu
Accept-Language: en-us
Accept-Encoding: gzip, deflate
User-Agent: Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1; SV1)
Host: search.bbc.co.uk
Cookie: BBC-UID=b479a5f4ad230a53063d513630203acb22684634a0e0b164c45f96efc054cf950Mozilla%2f4%2e0%20%28cc
Cache-Control: max-stale=0
Connection: Keep-Alive
X-BlueCoat-Via: 66808702E9A98546
```

Host	URL Path	URL Args
search.bbc.co.uk	/search	tab=urdu&order=sortboth&q=musharraf&start=3&scope=urdu&link=next

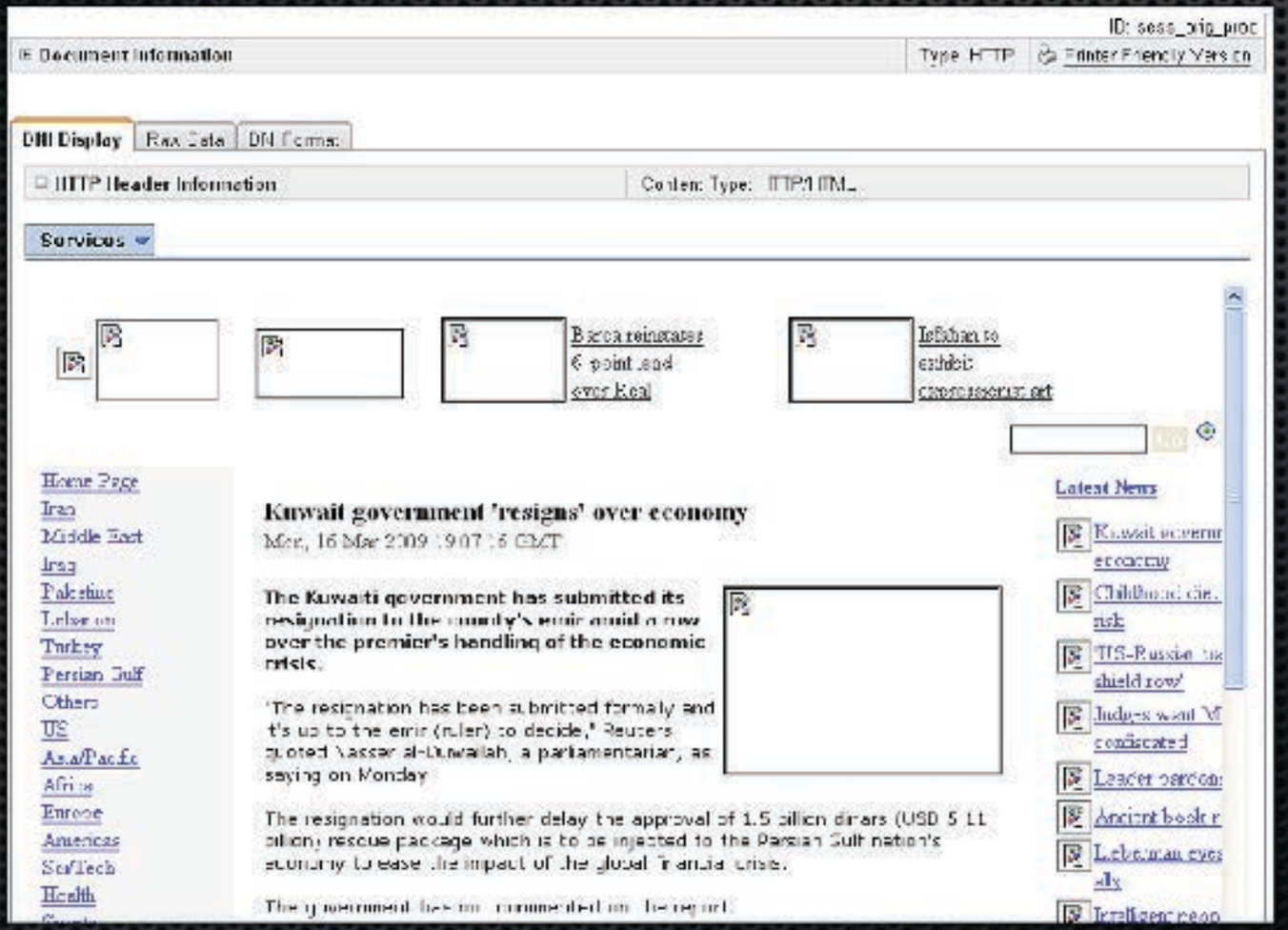
Search Terms	Language	Browser	Via
musharraf	en	Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1; SV1)	66808702E9A98546

Referer
http://search.bbc.co.uk/search?tab=urdu&order=sortboth&q=musharraf&start=2&scope=urdu

Cookie
BBC-UID=b479a5f4ad230a53063d513630203acb22684634a0e0b164c45f96efc054cf950Mozilla%2f4%2e0%20%28com

HTTP Activity Server-to-Client

Application Info	HTTP Type
Press TV - Kuwait government 'resigns' over economy	**response**



HTTP Activity – HTTP Types

- Meta-data will also tell you which side of traffic you're looking at
- Client-to-server has two main types:

HTTP Type
get

HTTP Type
post

- Server-to-client has only one:

HTTP Type
response

HTTP Activity – Get vs Post

- A ‘GET’ is you requesting data from the server (most web surfing)
- A ‘POST’ is you sending data to the server (i.e. signing in, filling out a form, composing an E-mail, uploading a file etc.)

Let's break down the important parts of a client-to-server request

HTTP Client-to-Server

```
GET /home.html
Host: sample.website.com
User-Agent: Mozilla/5.0 (Windows; U; Windows NT 5.1; en-US; rv:1.9.0.10) Gecko/2009042316 (USG-
    25) Firefox/3.0.10
Accept: image/png,image/*;q=0.8,*/*;q=0.5
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
Connection: keep-alive
```

First thing to note is the Host: line which tells you the name of the server that the client is requesting data from

Host Field

It's important to note, that in many cases users think they're at websites like www.yahoo.com, but behind the scenes data is coming from a number of different servers without the user knowing it:

GET /mc/modules/ui/abContacts?mcrumb=RIIDb6i9jm &.jsrand=98037807 &.rand=2127033459 HTTP/1.0	
Accept:	*/*
Accept-Language:	fa
Referer:	http://us.mc575.mail.yahoo.com/mc/showFolder;_ylc=X3oDMTBucmhobGR0BF9TAzM5ODMwMTAyNwRhYwNkZWxNc2dz?mid=1_21857_AERkxEIAANvjSi6wUQ7filZa4fY&fid=Inbox&sort=date&order=up&startMid=36&filterBy=
x-requested-with:	XMLHttpRequest
Accept-Encoding:	gzip, deflate
User-Agent:	Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1; SV1; .NET CLR 2.0.50727)
Host:	us.mc575.mail.yahoo.com
Cookie:	
	MG
	Y

Bonus question: What would the impact of this be in how you formulate your X-KEYSCORE queries using the Host field?

HTTP Client-to-Server

```
GET /home.html
Host: sample.website.com
User-Agent: Mozilla/5.0 (Windows; U; Windows NT 5.1; en-US; rv:1.9.0.10) Gecko/2009042316 (USG-
    25) Firefox/3.0.10
Accept: image/png,image/*;q=0.8,*/*;q=0.5
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
Connection: keep-alive
```

Second the GET line tells you which files the user is requesting from the server.

If you simply take that line and append it to the Host line you have the live public URL that the user is requesting:

http://sample.website.com/home.html

HTTP Client-to-Server

```
GET /example.php?region=iraq
Host: sample.website.com
User-Agent: Mozilla/5.0 (Windows; U; Windows NT 5.1; en-US; rv:1.9.0.10) Gecko/2009042316 (USG-
    25) Firefox/3.0.10
Accept: image/png,image/*;q=0.8,*/*;q=0.5
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
Connection: keep-alive
```

When the GET line has a ? mark in it, then the GET request is also passing information to the server.

So in this case the client is requesting the file example.php but it's also passing along a value that could have been entered by the user.

URL Lines

When there is a ? mark in the URL line, then X-KEYSCORE is breaking it up into two parts. The first part is called the URL Path and the second part is called the URL Argument.

URL Path	URL Args
/search	tab=urdu&order=sortboth&q=musharraf&start=3&scope=urdu&link=next

Notice all of the "arguments" (each separated by &'s) in this URL:

```
GET /search?tab=urdu&order=sortboth&q=musharraf&start=3&scope=urdu&link=next HTTP/1.1
Accept: */*
Referer: http://search.bbc.co.uk/search?tab=urdu&order=sortboth&q=musharraf&start=2&scope=urdu
Accept-Language: en-us
Accept-Encoding: gzip, deflate
User-Agent: Mozilla/4.0 (compatible; M
Host: search.bbc.co.uk
Cookie: BBC-UID=b479a5f4ad230a53063d51
Cache-Control: max-stale=0
Connection: Keep-Alive
X-BlueCoat-Via: 66808702E9A98546
```

Bonus question: Any idea what the information that is being passed in the URL Argument in this example are for?

HTTP Client-to-Server

```
GET /home.html
Host: sample.website.com
User-Agent: Mozilla/5.0 (Windows; U; Windows NT 5.1; en-US; rv:1.9.0.10) Gecko/2009042316 (USG-
    25) Firefox/3.0.10
Accept: image/png,image/*;q=0.8,*/*;q=0.5
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
Connection: keep-alive
```

The User-Agent line gives you information on what type of client is requesting the data. In this case, we can see that it was a Firefox 3.0 browser from a Windows NT 5.1 (XP) machine.

User Agents

User Agents

The User Agent (also known as the "browser") can be very valuable.

While it can not be trusted to be absolutely unique, in many cases you can use it to unwind a proxy or multi-user environment.

It can also help provide hints if the origins of the request came from a mobile device:

User-Agent:	Mozilla/5.0 (SymbianOS/9.2; U; Series60/3.1 NokiaE63-1/100.21.110; Profile/MIDP-2.0 Configurat like Gecko) Safari/413

User-Agent:	NokiaN72/5.0706.4.0.1 Series60/2.8 Profile/MIDP-2.0 Configuration/CLDC-1.1

User-Agent:	iPhone Mail (5H11)

HTTP Client-to-Server

```
GET /home.html
Host: sample.website.com
User-Agent: Mozilla/5.0 (Windows; U; Windows NT 5.1; en-US; rv:1.9.0.10) Gecko/2009042316 (USG-
    25) Firefox/3.0.10
Accept: image/png,image/*;q=0.8,*/*;q=0.5
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
Connection: keep-alive
```

The various “Accept” lines instruct the server on the types of responses the client can accept back.

Let's look at a simplified version of a HTTP request and response

What is Web (HTTP) Activity

This shows how a person logs on to a webpage

From Port 3434*
(client)

Click on http://www.hotmail.com
GET Request

To Port 80
(Server)

The client's port can be any high-numbered port, 3434 is just an example

What is Web (HTTP) Activity

This shows how a person logs on to a webpage

From Port 3434*
(client)

Click on http://www.hotmail.com
GET Request

To Port 80
(Server)

To Port 3434*
(client)

"Welcome to Hotmail"
HTTP Response

From Port 80
(server)

The client's port can be any high-numbered port, 3434 is just an example

What is Web (HTTP) Activity

This shows how a person logs on to a webpage

From Port 3434* (client)

Click on http://www.hotmail.com
GET Request

To Port 80 (Server)

To Port 3434* (client)

"Welcome to Hotmail"
HTTP Response

From Port 80 (server)

From Port 3434* (client)

Email Address: me@hotmail.com
Password: Admin123
POST to the Web server

To Port 80 (Server)

The client's port can be any high-numbered port, 3434 is just an example

What is Web (HTTP) Activity

This shows how a person logs on to a webpage

From Port 3434* (client)

Click on http://www.hotmail.com
GET Request

To Port 80 (Server)

To Port 3434* (client)

"Welcome to Hotmail"
HTTP Response

From Port 80 (server)

From Port 3434* (client)

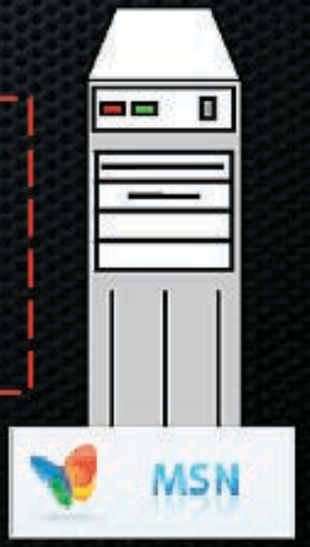
Email Address: me@hotmail.com
Password: Admin123
POST to the Web server

To Port 80 (Server)

To Port 3434* (client)

"Welcome to your Inbox/homepage"
HTTP Response

From Port 80 (server)



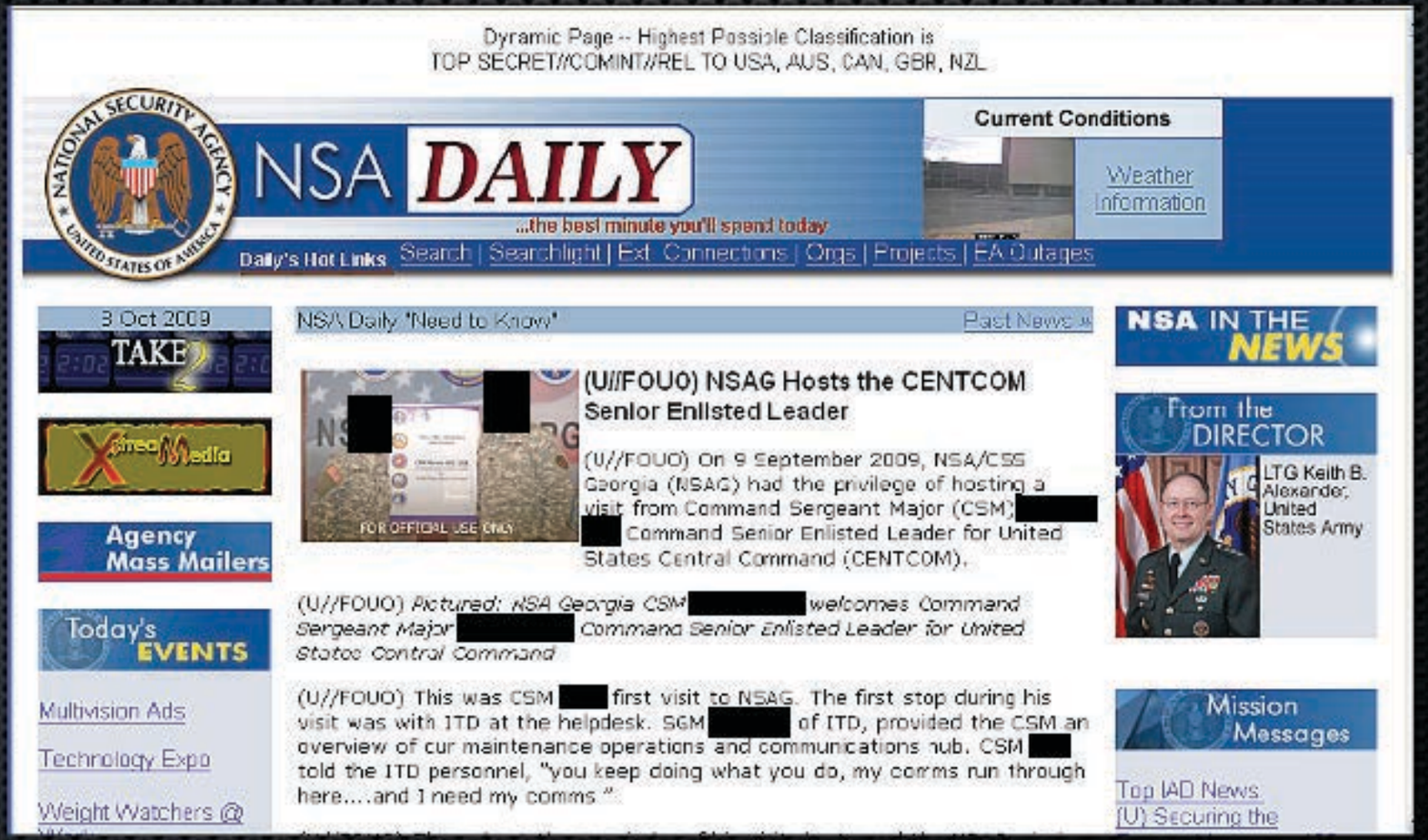
The client's port can be any high-numbered port, 3434 is just an example

HTTP Activity

- Real traffic, however, can be a little more complicated.
- Almost all web pages are built from multiple files.
- For example, every single image or banner ad on a web page is a separate file that needs to be individually requested before the server that has the file can respond

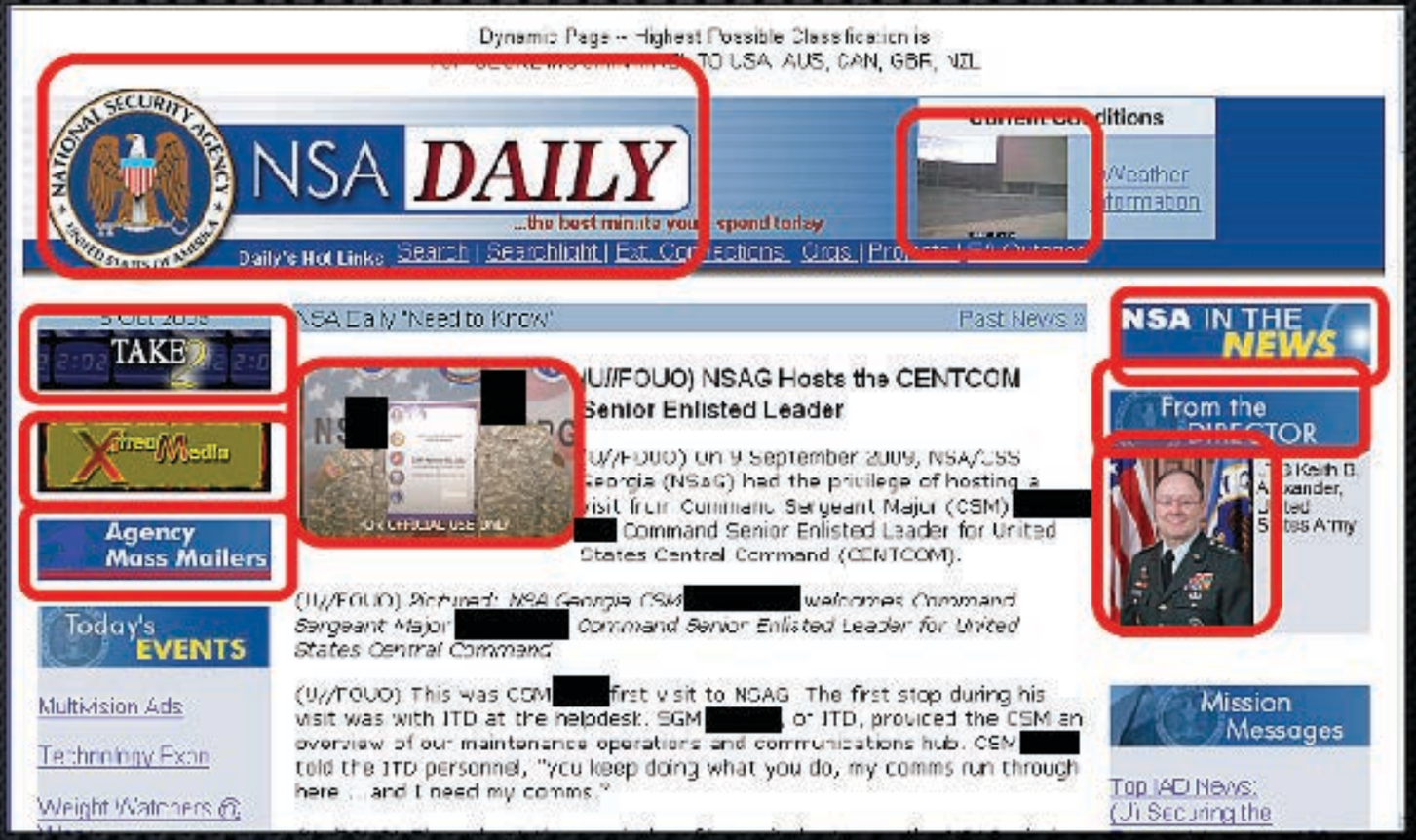
HTTP Activity – Real World

- Let's look at the "NSA Today" home page.



HTTP Activity – Real World

- It looks like one page, but each of the images and banners are separate data files that your browser pieces back together

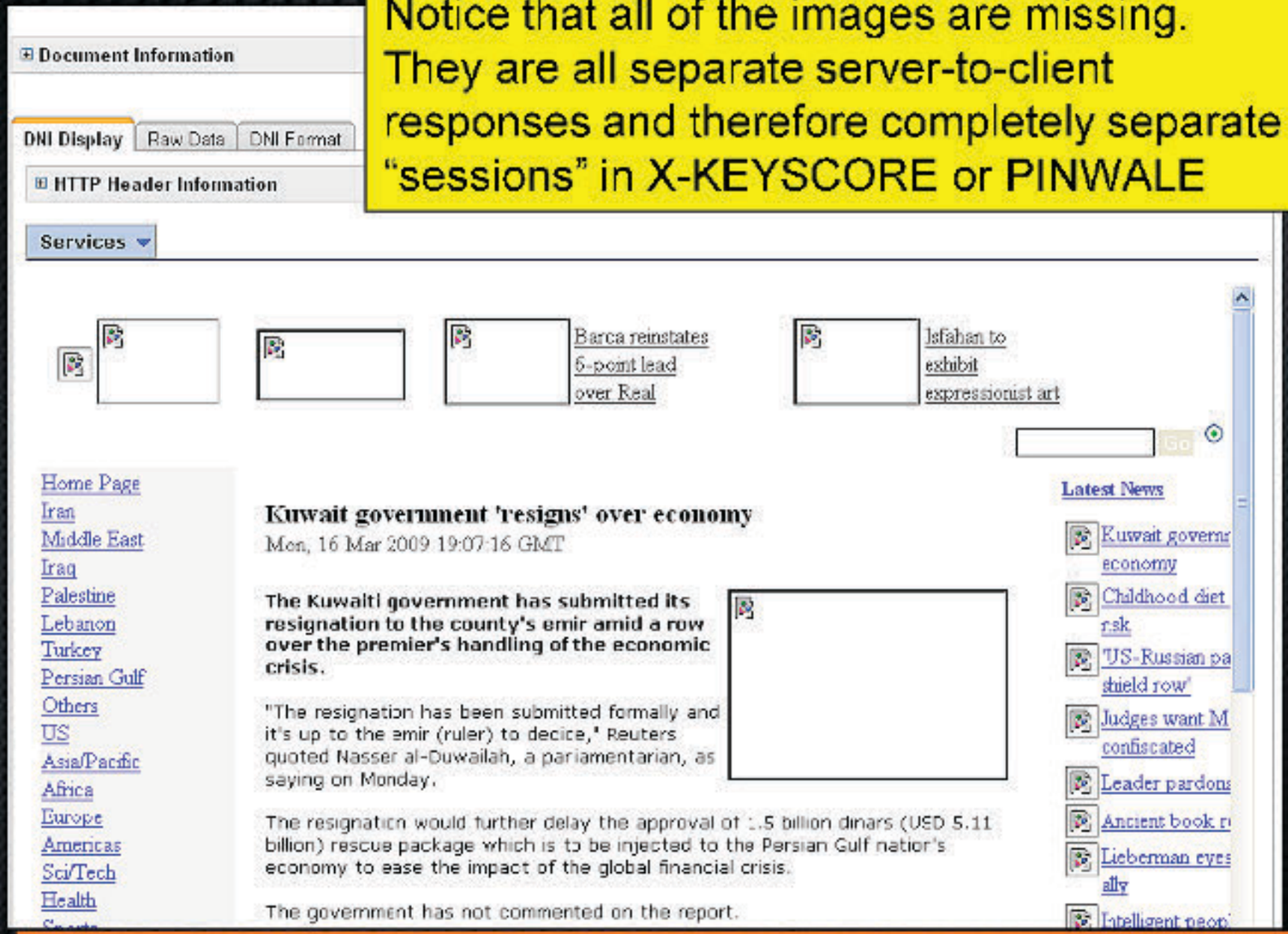


HTTP Activity – Real World

- In fact, to build the NSA Today home page it takes 34 separate files from 4 different servers
- However, most people probably don't notice, because the entire page loads in <300 milliseconds.
- If we had a slow internet connection, we'd notice the images would initially be missing.

HTTP Activity Real-Word

Notice that all of the images are missing. They are all separate server-to-client responses and therefore completely separate "sessions" in X-KEYSCORE or PINWALE



HTTP Activity – Real World

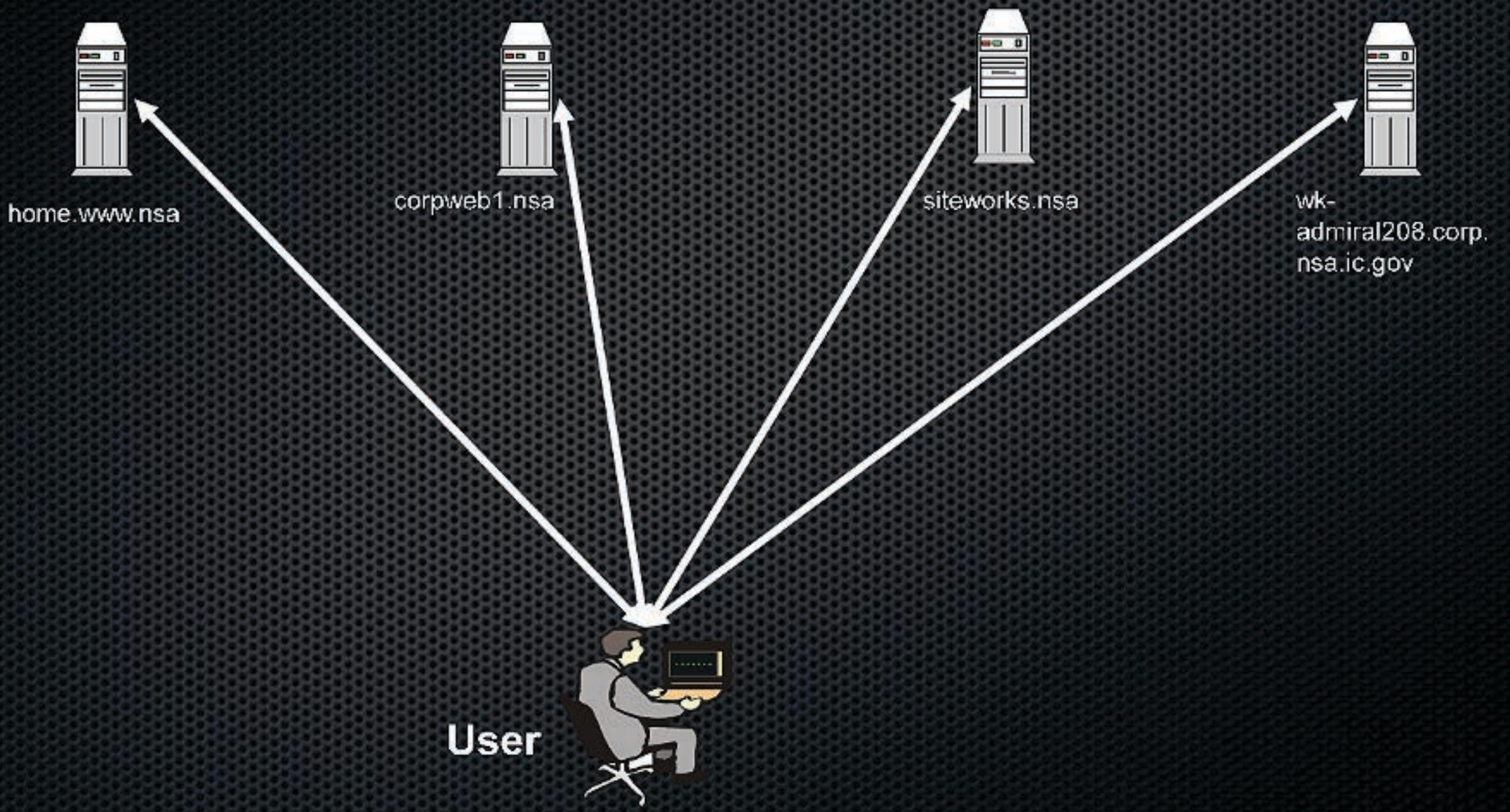
- It's important to note that not all of the data on one web-page came from the same server.
- For example, most of the NSA Today home page come from home.www.nsa, but the image of the current weather conditions came from wk-admiral208.corp.nsa.ic.gov

HTTP Activity – Real World

- This happens all the time on the Internet.
- The cnn.com home page, may have an ad on it that was from the Google ad server and etc.
- And this does have an impact on our collection!

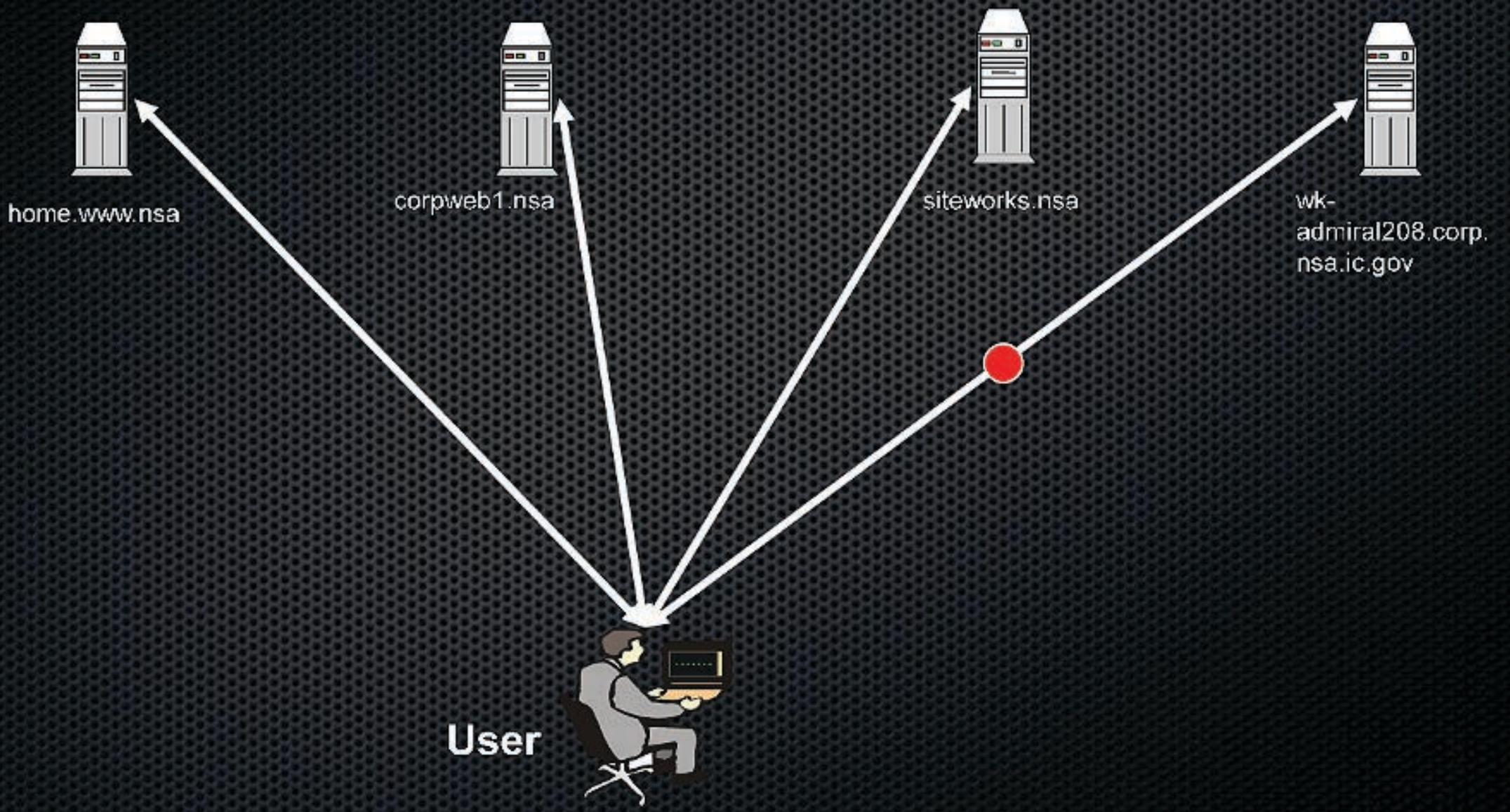
HTTP Activity

- This is the traffic path for building the NSA today home page



HTTP Activity

- What happens if we only have collection on one of the paths?



What would that traffic look like?

```
GET /current.jpg
Host: wk-admiral208.corp.nsa.ic.gov
User-Agent: Mozilla/5.0 (Windows; U; Windows NT 5.1; en-US; rv:1.9.0.10) Gecko/2009042316 (USG-
    25) Firefox/3.0.10
Accept: image/png,image/*;q=0.8,*/*;q=0.5
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
Connection: keep-alive
Referer: http://home.www.nsa/
If-Modified-Since: Thu, 08 Oct 2009 19:31:56 GMT
If-None-Match: "d945-16c1-842db643"
Cache-Control: max-age=0
```

If we only saw this one GET request and not the other 33 required to build the NSA Today home page, would we be able to determine what the user was actually doing?

What exactly is that telling us?

- First off, we know what file they are requesting.
- They want current.jpg from the wk-admiral208.corp.nsa.ic.gov server.
- That's actually a live public URL (http://wk-admiral208.corp.nsa.ic.gov/current.jpg)
- Do we have any indication why they wanted that image? Answer is yes! Look at the referer field.

What exactly is that telling us?

- They were referred from http://home.www.nsa/
- The referer is in essence, telling you what site was “linking” to the new site.
- Warning! The referer can act in misleading ways.

Referer Field

- The referer field is the address of the page that links to new GET request.
- However, this link could have been automatic to the user.
- I.e. in the case of the current weather image, the link was automatic and the user wasn't even aware of the action

Referer Field

- The referer field could also indicate a user action.
- For example, imagine we were on the NSA Today webpage and clicked the link to the SID Today page.
- What would that traffic look like?

Referer Field

GET /
Host: sidtoday.nsa
User-Agent: Mozilla/5.0 (Windows; U; Windows NT 5.1; en-US; rv:1.9.0.10) Gecko/2009042316 (USG-25) Firefox/3.0.10
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
Connection: keep-alive
Referer: http://home.www.nsa/
Cookie: CFID=565238; CFTOKEN=66534796;
CFGLOBALS=urltoken%3DCFID%23%3D565238%26CFTOKEN%23%3D66534796%26jsessionid%23%3Da830dba3a04b67ae6e351b7463444f72496d%23lastvisit%3D%7Bts%20%272009%2D10%2D09%2015%3A38%3A04%27%7D%23timecreated%3D%7Bts%20%272009%2D06%2D19%2010%3A27%3A23%27%7D%23hitcount%3D13%23cftoken%3D66534796%23cfid%3D565238%23;
JSESSIONID=a830dba3a04b67ae6e351b7463444f72496d

Referer Field

- Now we're seeing a request go to host "sidtoday.nsa" with the referer from http://home.www.nsa
- How can we tell from the traffic that the first automatic referer we saw for the current weather was any different from the user-generated referer we saw for the SID Today article?

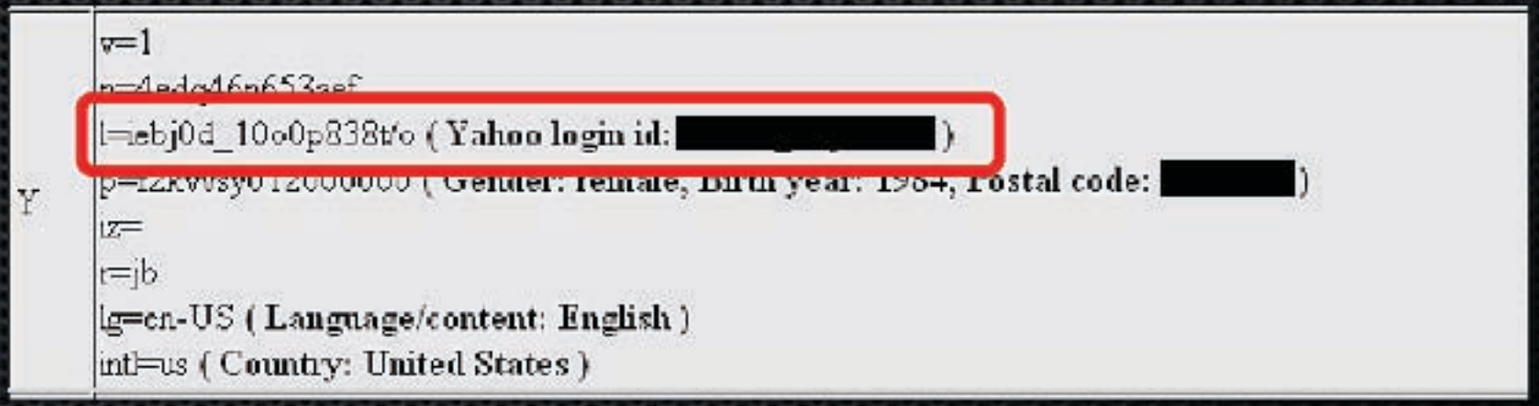
Cookies!

Cookies

- Cookies are small pieces of text-based data stored on your machine by your web browser.
- Almost all websites have cookies enabled and they have a variety of uses, including to help the web-site track the activities of their users.
- Most analysts are probably familiar with “machine specific cookies” like the Yahoo B cookie
- However cookies are used for a variety of reasons

What can cookies be used for?

- Cookies can be used to authenticate a user.
- For example in many cases, the “active user” for Yahoo web-mail traffic is seen encoded in the l= part of the cookie string.



What can cookies be used for?

- Cookies can be used to store information about the user that the website is interseted in
- Look at how the p= value below tells the website information about the user of this account:

Y	v=1 n=4edq46n653aef l=ebj0d 10o0p838t/o (Yahoo login id: [redacted]) p=f2kvvsy012000000 (Gender: female, Birth year: 1984, Postal code: [redacted]) iz= t=jb lg=en-US (Language/content: English) intl=us (Country: United States)

What can cookies be used for?

- Cookies can be used to identify a single machine from hundreds of other users on the same proxy IP address
- The Yahoo B cookie is a “machine specific cookie”

What can cookies be used for?

- Important note: All three of those examples are just subsets of the full Yahoo cookie string

How do we know what each cookie value is used for?

- Nearly every web-site uses cookies that in most cases they designed for their own uses, so how do we know what they all mean?
- Protocol Exploitation can examine the traffic to try to determine if there is any information contained in cookie strings that we might be interested, for example we'd like to know if any part of the cookie acts like a "machine specific cookie."

How do we know what each cookie value is used for?

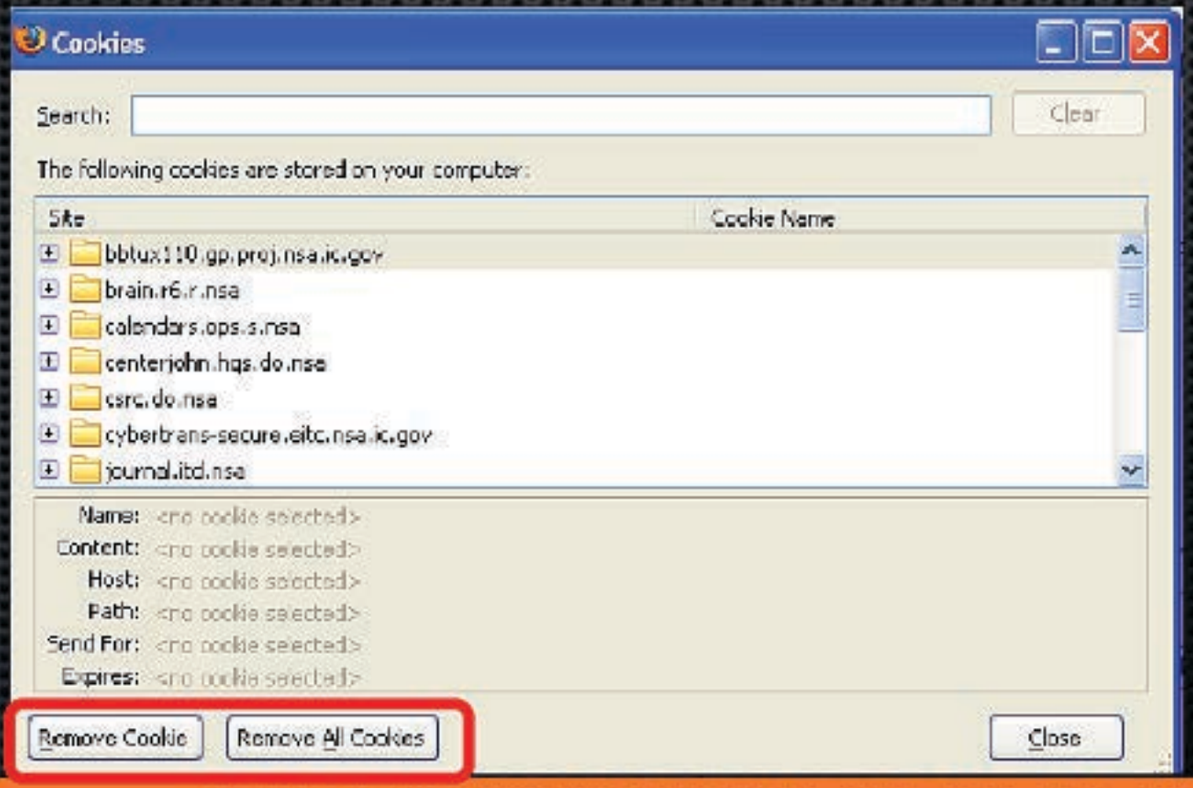
- However, there are far more cookie options out in the wild than PE can possible examine.
- So even if they aren't aware of a machine specific cookie, it doesn't mean that it doesn't exist.
- X-KEYSCORE gives you access to the full cookie string, so if you're adventurous enough you can do your own protocol exploitation.

Remember: Cookies are there for a reason!

- Websites put cookies on people's computers for a reason.
- If the data is valuable for a website, it may be valuable to us as well.

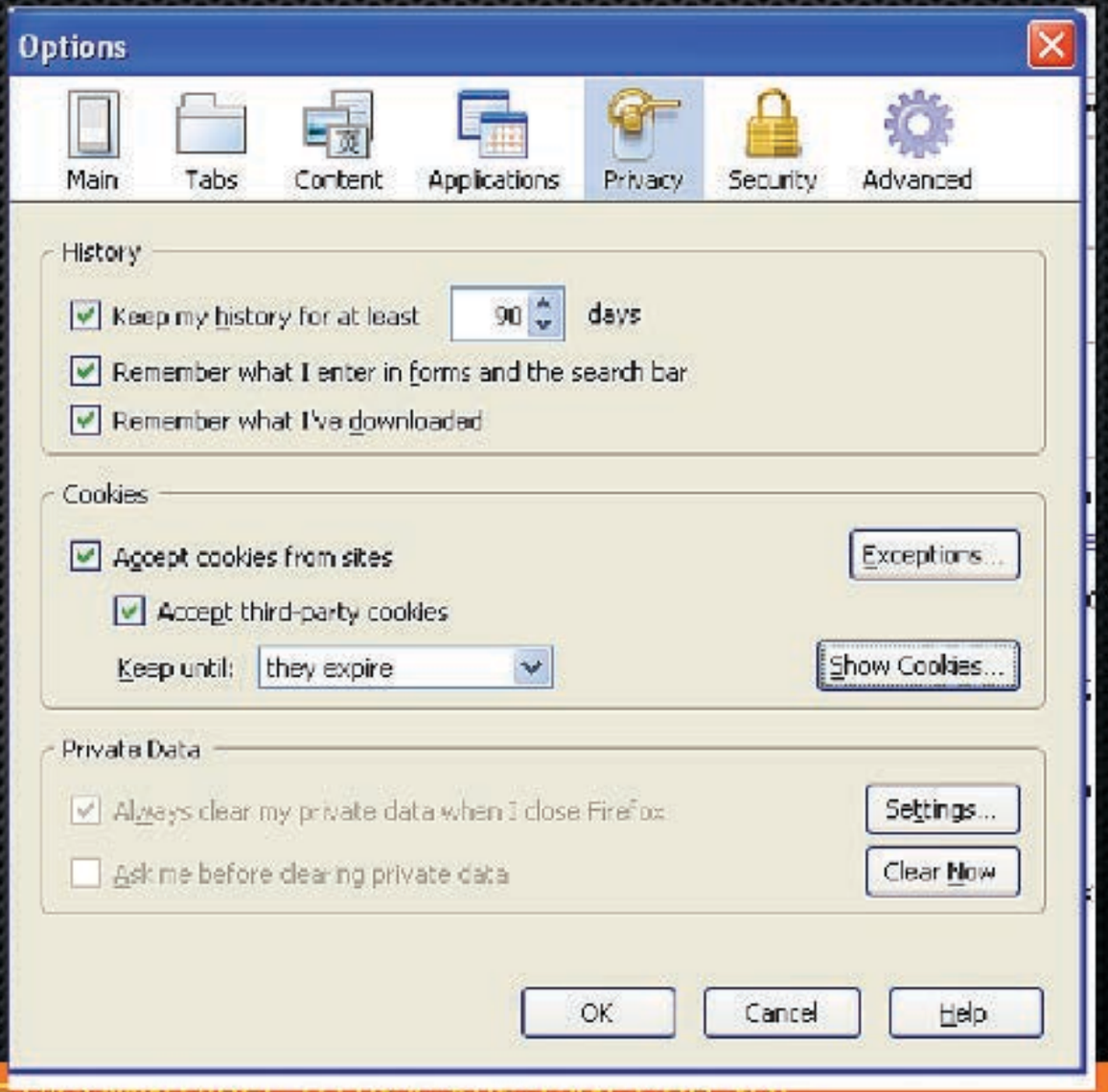
How long do cookies live for?

- Cookies, like any other file on a computer, can be deleted by the user.
- Almost all browsers give you the option to view, manage and delete your cookies



Cookies

You can see what cookies have been stored on your machine by going into the "options" window of your browser and selecting "show cookies"



Searches

Searching the Internet

When a user searches the Internet from one of the many web-based search engines (Google, Bing, etc.) what does the traffic look like?

Searching the Internet: Client-to-Server

- In most cases, the client-to-server traffic is a GET request where the search term is passed in the URL Arguments:

```
GET /search?hl=fr&q=iran&lr= HTTP/1.1
Host: www.google.com
Accept: image/gif, image/x-xbitmap, image/jpeg, image/pjpeg, application/x-shockwave-flash,
    application/vnd.ms-powerpoint, application/vnd.ms-excel, application/msword, */*
Cookie: PREF=ID=74f6d7addf51ccd4:U=ccbee9ee665a7dde:TB=2:TM=1255354439:LM=125543326
    4:S=_M1i4RfO2ohI81maNID=27=cMFLkpovJCIWI0FC5E3Pu2C6-8_nsMS2zztfvOew9-
    QYDPWUza4AscyogIQRGNSkDZsi2jL65 fIM-R4HgovMBEa66bfiTXn8TH3Ukm-
    X5hp45rLAb_Y3rNZ42HGIzyne
Accept-Encoding: gzip, deflate
User-Agent: Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1)
Connection: Keep-Alive
Cache-Control: no-cache
```

Searching the Internet: Client-to-Server

- Notice how the URL Path is /search and one part of the URL argument is q=iran
- Each website can configure their URL's differently, so while with Google the search term is contained in the q= part of the URL, a different search form might have it as query= or search_term= etc.

http://www.youtube.com/results?search_query=iran&search_type=&aq=o

Searching the Internet: Client-to-Server

- X-KEYSCORE tries to account for all the variations of search terms contained in the URL Argument for what it extracts for the "Search Term" column.
- However, there are always other varieties out there that we haven't built it hooks for yet, so anytime you see something that you think should be extracted, please contact the team ( )

"Referer Searches"

- What happens when a user clicks on a search result?
- Let's start by showing the query itself, in this example, we're going to query the NSANet Google for "XKEYSCORE"

"Referer Searches"

- What does that GET request look like?

```
GET /search?q=xkeyscore&btnG=Google
Host: google4.q.nsa
User-Agent: Mozilla/5.0 (Windows; U; Windows NT 5.1; en-US; rv:1.9.0.10) Gecko/2009042316
    (USG-25) Firefox/3.0.10
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
Connection: keep-alive
```

We know from this session that the client is requesting the data from the host 'google4.q.nsa" and we see the search term in the URL Argument

"Referer Searches"

- What happens when a user clicks on a search result?

```
GET /redmine
Host: xkeyscore.r1.r.nsa
User-Agent: Mozilla/5.0 (Windows; U; Windows NT 5.1; en-US; rv:1.9.0.10) Gecko/2009042316
    (USG-25) Firefox/3.0.10
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
Connection: keep-alive
Cookie: _session_id=ffd87ac8682e8fa8f421b4ffdf9693ae
Referer: http://google4.q.nsa/search?q=xkeyscore&btnG=Google+Search
```

First, we can determine the full URL by adding the GET line to the host: http://xkeyscore.r1.r.nsa/redmine

"Referer Searches"

- Secondly, we get some hints as to why the user was requesting that page from the Referer line:

```
Referer: http://google4.q.nsa/search?q=xkeyscore&btnG=Google+Search
```

- Note that it was the same URL that we were at immediately before we clicked the "result" link

"Referer Searches"

- Let's look at that process again:

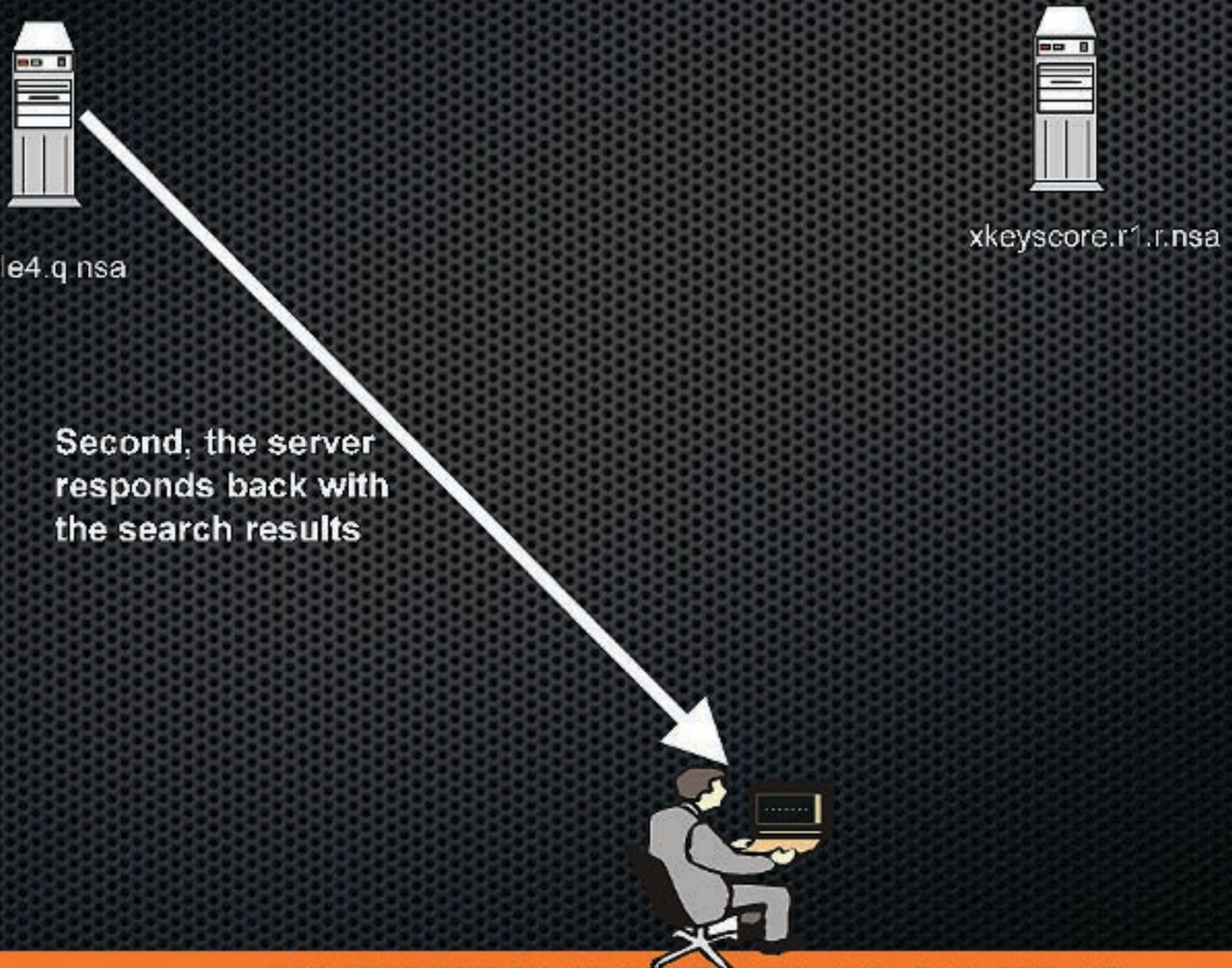
google4.q.nsa

xkeyscore.r1.r.nsa

First, a client-to-server request is sent that contains the query on "xkeyscore"

"Referer Searches"

- Let's look at that process again:



"Referer Searches"

- Let's look at that process again:

google4.q.nsa

xkeyscore.r1.r.nsa

Third, by clicking on one of the results, a new GET request is issued to retrieve the XKEYSCORE home page. In this request, the location of the original search is listed as the "referer"

"Referer Searches"

- Let's look at that process again:

google4.q.nsa

xkeyscore.r1.r.nsa

What will happen if we only have collection on this link?

"Referer Searches"

- When XKEYSCORE sees a search contained in the "referer" field, we still extract it out as meta-data into the "search terms" but we append it with (referer) to denote where it was originally found:

Search Terms
(referer) the legal status of the caspian sea

HTTP Type	Host	URL Path	URL Args
get	www.parstimes.com	/law/caspian_status.html	

Referer
http://www.google.com/search?hl=fa&source=hp&q=the+legal+status+of+the+caspian+sea&lr=

"Referer Searches"

```
GET /law/caspian_status.html HTTP/1.1
Accept: */*
Host: www.parstimes.com
Referer: http://www.google.com/search?hl=fa&source=hp&q=the+legal+status+of+the+caspian+sea&lr=
Accept-Language: fa
Accept-Encoding: gzip, deflate
User-Agent: Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1; SV1; .NET CLR 2.0.50727; InfoPath.2)
Cache-Control: max-stale=0
Connection: close
X-BlueCoat-Via: 0A6F53530F3F63EE
```

Can we guess what happened here?

Referer searches

Another example:

TOP SECRET//COMINT//20320108

ID: sess_orig_proc

Type: HTTP-GET | Printer Friendly Version

DNI Display | Raw Data | DNI Format

Services

GET /Hezbollah-Terrorism-Judith-Palmer-Harik/dp/1860648932 HTTP/1.1	
User-Agent:	Mozilla/5.0 (Windows; U; Windows NT 5.1; en-US) AppleWebKit/525.19 (KHTML, like Gecko) Chrome/1.0.154.48 Safari/525.19
Referer:	http://www.google.com.pk/search?hl=en&q=wretten books on hizbollah&btnG=Google Search&meta=
Accept:	text/xml,application/xml,application/xhtml+xml,text/html;q=0.9,text/plain;q=0.8,image/png,*/*;q=0.5
Accept-Encoding:	gzip,deflate,bzip2,sdch
Cookie:	ubid-main=185-5525816-8765531
	apn-user-id=P1YXY7QF1PUYQ5
Accept-Language:	en-US,en
Accept-Charset:	ISO-8859-1,*,utf-8
Host:	www.amazon.com
Connection:	Keep-Alive

Proxy Information

Proxy Information

- In a lot of cases we're going to see HTTP Activity from behind a proxy or proxies.
- What is a proxy?
  - A proxy is a server that is acting as an intermediary for HTTP requests from clients
- Why do proxies exists?

- **Performance**: Proxy can cache responses for static pages
- **Censorship**: Proxy can filter traffic
- **Security**: Proxy can look for malware
- **Access-Control**: Proxy can control access to restricted content

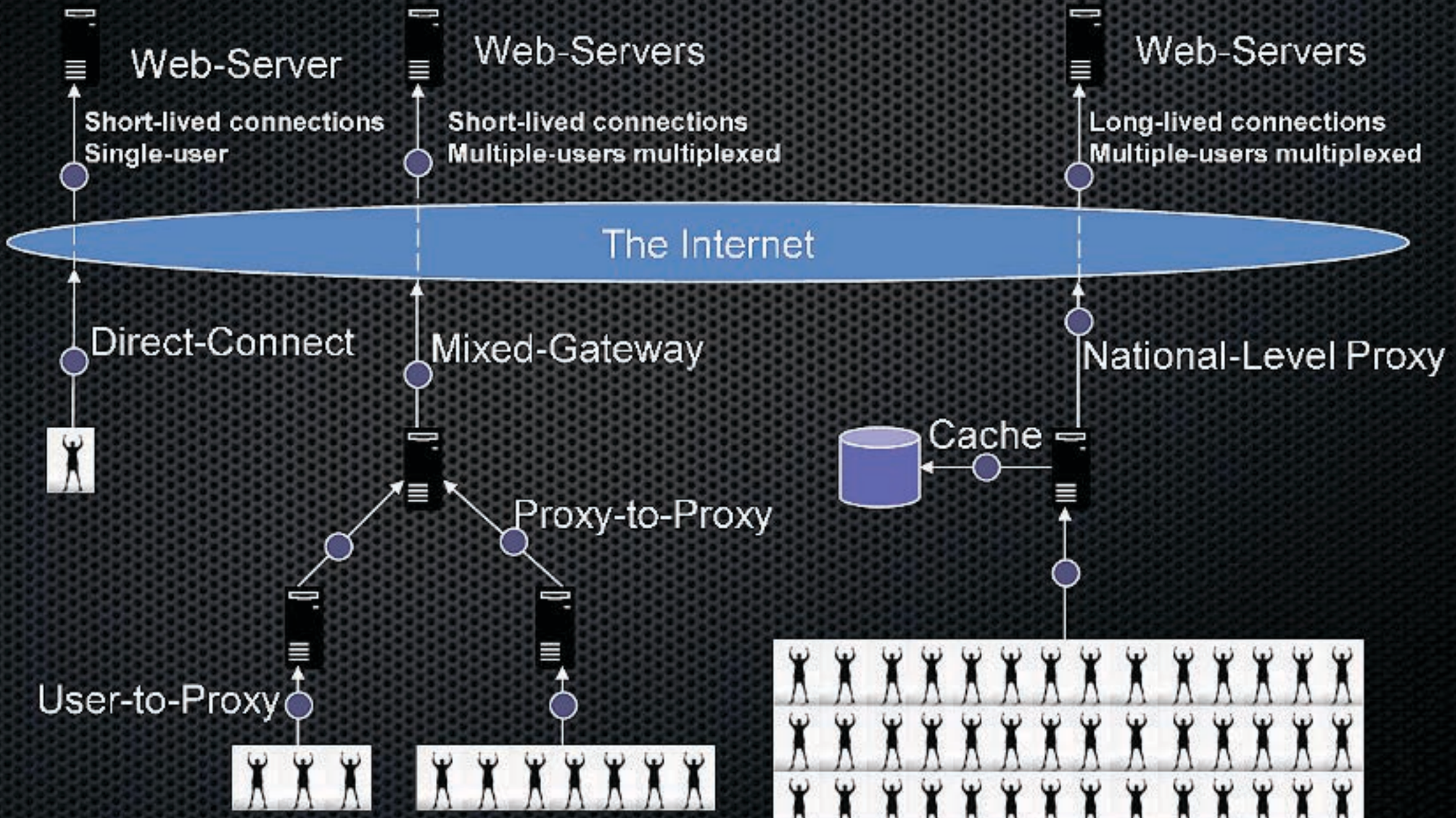
Proxy Information

- Routinely, we're going to see ISP level proxies.
- That is, instead of having each individual user request web pages directly from the web servers, the ISP is going to collect all of those requests first, and then proxy them out through a handful of proxy IP addresses.
- When the response is returned, the proxy passes it on to the appriopriate user

Proxy Information

- Why would the ISP want to proxy traffic?
- In many cases the ISP won't have to supply public IP addresses to all it's users
- It can simply give them a private IP address, and then use a handful of public IP addresses for its proxies which are the machines actually requesting the traffic from the web-servers

Proxies on the Internet



Identifying a Proxy

- How do you know that the IP address that you think is your target is really a proxy?
- First step, check NKB.
- They have services that attempt* to automatically detect proxies

* These services are in no way 100% accurate so this is only the first step in checking to see if the IP Address is a proxy

Identifying a Proxy: NKB

Query: IP Address [redacted]
Date: 2009-10-27@09:21:50

Description	Value	Confidence	Classification
Location Add Analyst Input			
IP Range	[redacted]		(TS//SI//REL TO USA, FVEY)
Lat/Long (precision)	(none found)		(TS//SI//REL TO USA, FVEY)
City	ZAHEDAN	20	(TS//SI//REL TO USA, FVEY)
Country	IR (IRAN)	91	(TS//SI//REL TO USA, FVEY)
Provider Add Analyst Input			
IP Owner	RAYANE FARAZ IRANSHAHR COMPANY , INTERNET SERVICE PROVIDER	82	(U//FOUO)
Autonomous System Route Prefix	[redacted]0.0/17	59	(U//FOUO)
Autonomous System Number	12880	95	(U//FOUO)
Autonomous System Name	DCI-AS DCI Autonomous System	95	(U//FOUO)
Device Add Analyst Input			
FQDN	(none found)		
Domain	[redacted]	30	(U//FOUO)
Service	PROXY		(U//FOUO)
Service	TRANSPARENTPROXY		(U//FOUO)

Identifying a Proxy

- Other things to be on the look out for:
- X-Forwarded-For IP Address
  - What is it?
  - An X-Forwarded-For IP address the proxy passing on to the server what it thinks is the IP address of the user
  - Think of it as the proxy telling the server “this is who I think this request came from”
  - It’s important to note that multiple proxies can, and often, are present, so one proxy might just be reporting the IP address of another proxy

Identifying a Proxy

- X-Forwarded-For IP Address as seen in traffic:

GET / HTTP/1.0	
User-Agent:	Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1; SV1)
Host:	www.ebay.com
Pragma:	no-cache
Via:	1.0 s.jonoobnet.com (squid/3.0.STABLE10)
X-Forwarded-For:	[redacted]
Cache-Control:	max-age=259200
Connection:	keep-alive

Some Examples of X-Forwarded-For headers:

X-Forwarded-For: [redacted]

X-Forwarded-For: [redacted]

X-Forwarded-For: [redacted]

X-Forwarded-For: [redacted]

X-Forwarded-For: [redacted]

X-Forwarded-For: 192.168.1.10, 10.0.0.22, [redacted]

X-Forwarded-For: [redacted]

X-Forwarded-For: 127.0.0.1, [redacted]

X-Forwarded-For: [redacted] google.com, [redacted]

Multiple-Layers of Proxies!

In-general, the first IP is the one closet to the original requestor
Keep in mind – these can be totally fake

Identifying a Proxy

- Similar to the X-Forwarded-For Tag is the "VIA tag"
- The VIA tag is the proxy identify itself

GET / HTTP/1.0	
User-Agent:	Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1; SV1)
Host:	www.ebay.com
Pragma:	no-cache
Via:	[redacted] (squid/3.0.STABLE10)
X-Forwarded-For:	217.219.95.135
Cache-Control:	max-age=259200
Connection:	keep-alive

Identifying a Proxy

- The Via: tag may even contain some good information about the proxy
- Be careful though because this information could be falsified:

Via:	1.0 tehran-proxy-srv:3128 (squid/2.5.STABLE1)

Identifying a Proxy

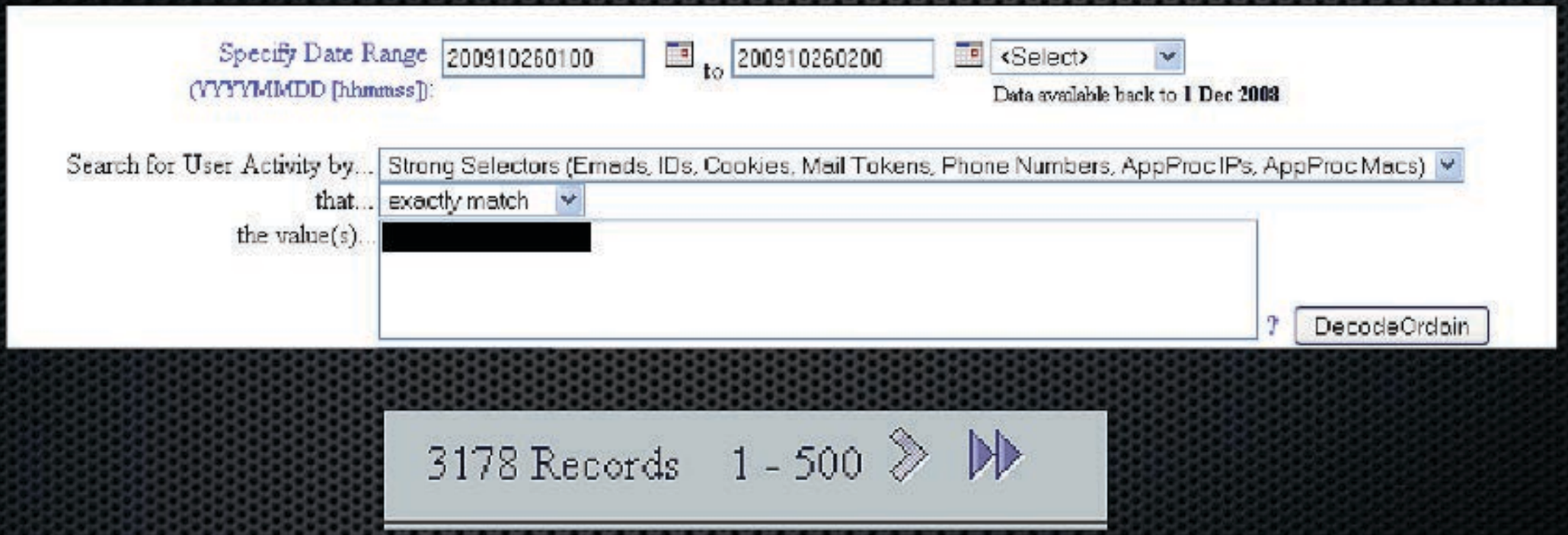
- Remember though that the X-Forwarded-For and VIA lines can be falsified and don't have to be present!
- If they're not present, how can you tell the IP address is a proxy?
- Test it in MARINA!

Testing IP Addresses in MARINA

- The primary side effect of a proxy is too many users online at the same time
- So if all else fails, try querying on the IP address (assuming its USSID18 compliant of course!) in MARINA to see how many users were active within an hour time frame
- It's not scientific but generally it will help

Testing IP Addresses in MARINA

For example look at these results:



There were 274 unique “Active Users” in that hour, think it’s a proxy?

HTTP Header Fingerprint (HHFP)

What is the HHFP?

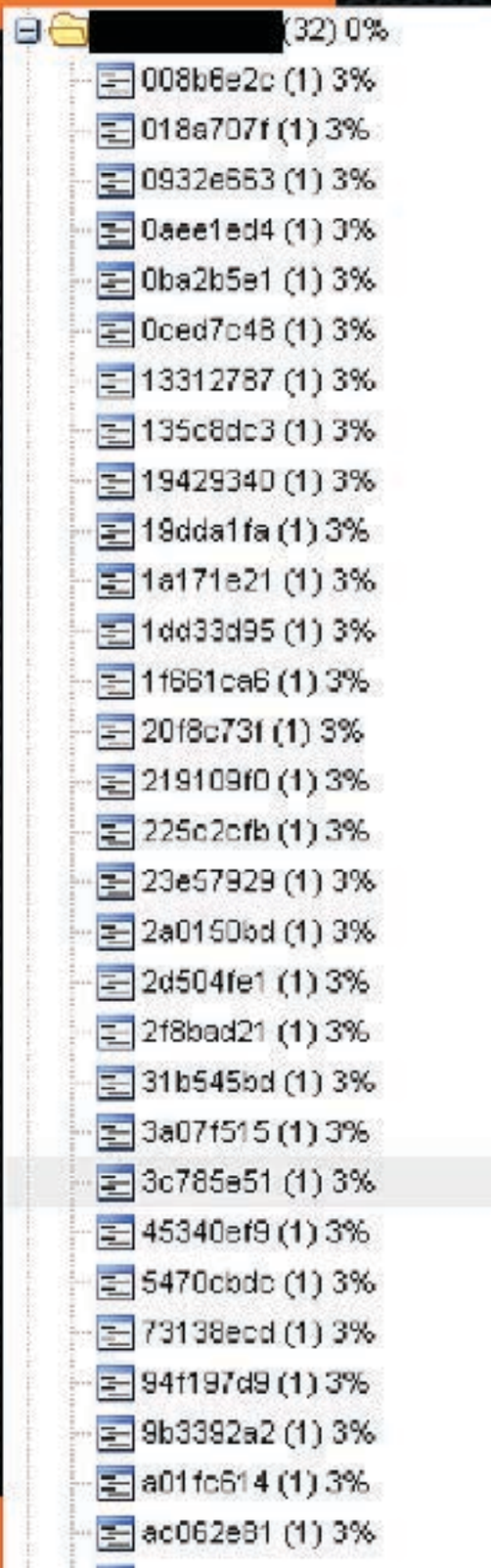
- GCHQ created the HHFP to help identify individual users behind a single proxy IP address
- The HHFP is a hash of multiple header fields that can be used to identify a single user behind a proxy

What is the HHFP?

- At least one of these values must be present:
  - X-Forwarded-For IP Address
  - Via
  - Client IP address
- If so, the HHFP is a hash of those values combined with the User Agent string

What is the HHFP?

- EX: Here's an Iranian proxy IP Address that has multiple HHFP's underneath it.
- NOTE: There's no guarantee that an HHFP is identifying a single unique user, it's entirely possible that more than one user will have the same HHFP



Pros and Cons of HHFP

- On the positive side, the HHFP is a single 8 digit value which can help identify a single user behind a proxy
- On the negative side, it requires an XFF IP address, Via string or Client IP Address and since many sessions do not contain all three, they'll have no HHFP string
- Also even with the HHFP, all of the fields that are used to build it are available in the XKS HTTP Activity query so it's not providing you with any data you don't already have access to

XKS's HTTP Activity Search

XKS HTTP Activity Search

After that overview of how HTTP Activity works, let's look into how to effectively target it through XKS queries

XKS HTTP Activity Search

- HTTP Activity indexes every HTTP session
- Client-to-server and server-to-client
- Can be queried on any of the unique HTTP meta-data fields or any of the "standard" DNI fields (IP Address, SIGAD, CASENOTATION etc).

XKS HTTP Activity Search

- Unique Meta-data fields of this search include:

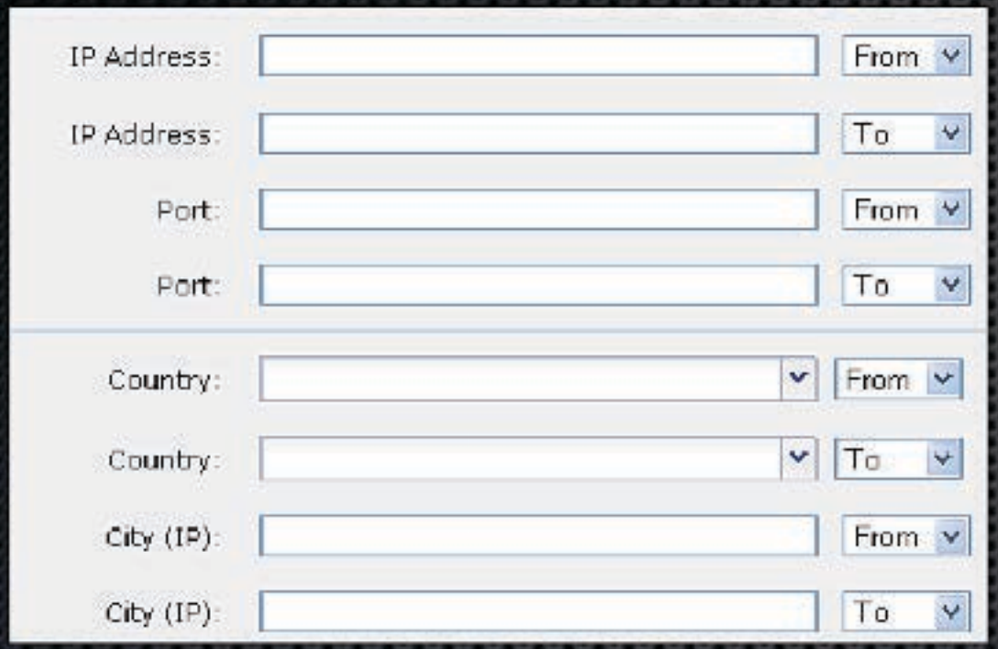
Fields already covered in this training:

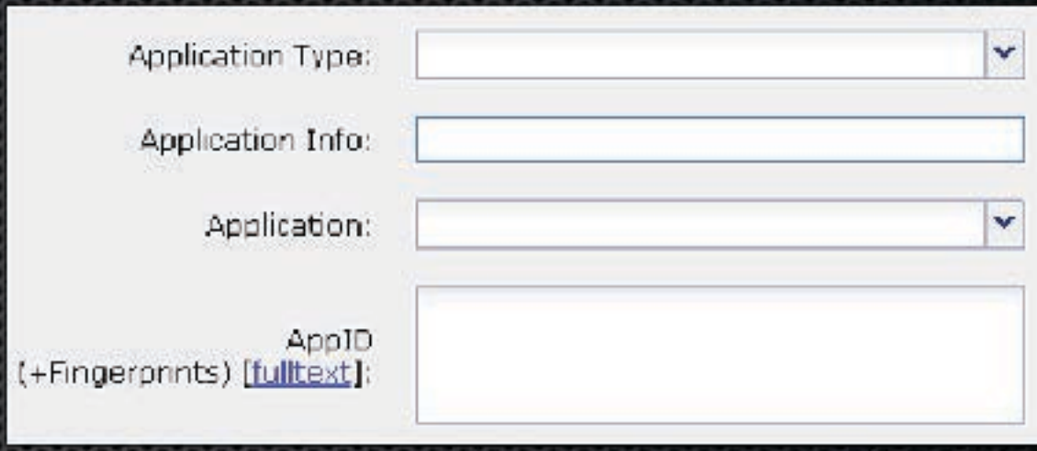
HTTP Type:
Host:
URL Path:
URL Args:
Search Terms:
Language:
Active User:
TDI Type:
TDI:
Character Encoding:
Content Start:
Content Stop:
Content Total:

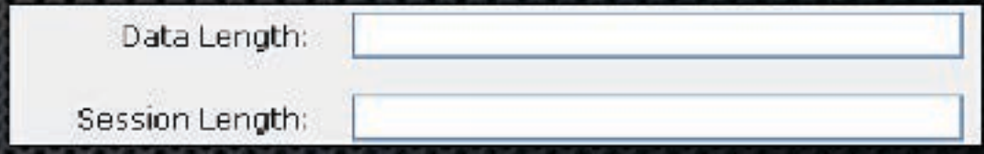
Referer:
X Forwarded For:
Via:
Proxy Hash (HHFP):
Cookie:
Browser:
Attachment Filename:
Server Type:
Geo Info [fulltext]:
Misc Info [fulltext]:
Links of Interest:

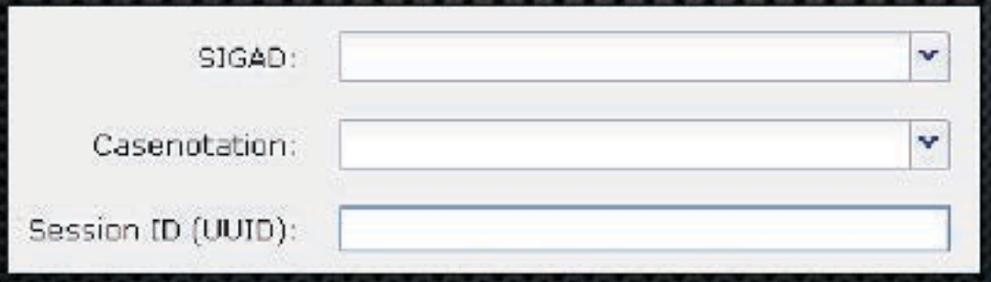
XKS HTTP Activity Search

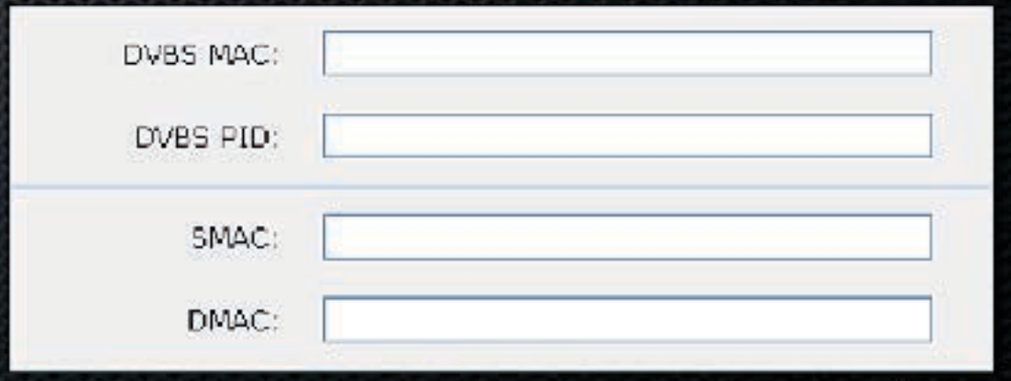
- In addition to all of the common fields like:









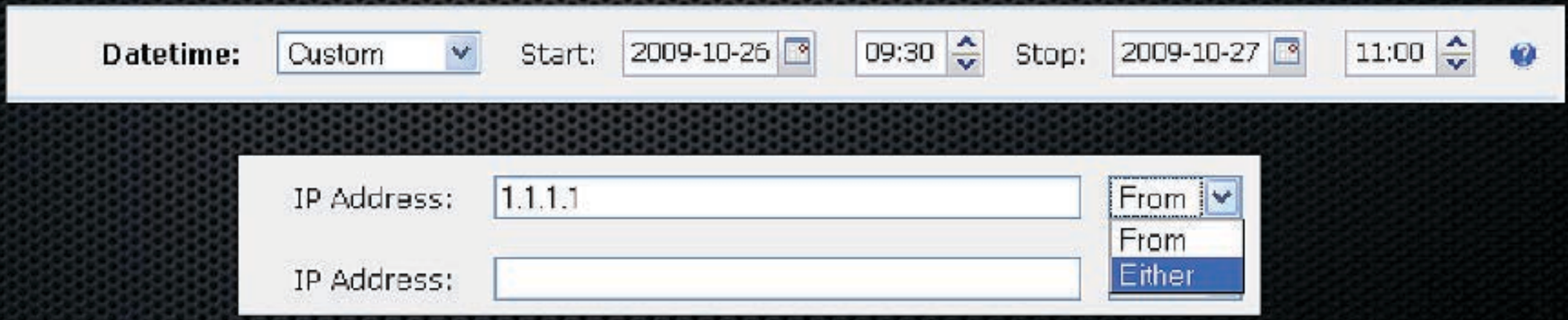


XKS HTTP Activity Search

- Most commonly HTTP Activity query searches in XKS will be to enable "persona analysis"
- Based on MARINA, TRAFFICTHIEF or PINWALE, we'll want to query XKS to discover all of the HTTP Activity that occurred around the targets session of interest

Simple HTTP Searches

- In order to do a "persona analysis" type search, all we'll need to fill in is the IP of the target (assuming it's USSID18 compliant) and a short time range "around" the time of the activity:



XKS HTTP Activity Search

Another common query is analysts who want to see all traffic from a given IP address (or IP addresses) to a specific website.

XKS HTTP Activity Search

- For example let's say we want to see all traffic from IP Address 1.2.3.4 to the website www.website.com
- While we can just put the IP address and the "host" into the search form, remember what we saw before about the various host names for a given website

Host Field

It's important to note, that in many cases users think they're at websites like www.yahoo.com, but behind the scenes data is coming from a number of different servers without the user knowing it:

GET /mc/modules/ui/abContacts?mcrumb=RIIDb6i9jm &.jsrand=98037807 &.rand=2127033459 HTTP/1.0		
Accept:		*/*
Accept-Language:		fa
Referer:		http://us.mc575.mail.yahoo.com/mc/showFolder;_ylc=X3oDMTBucmhobGR0BF9TAzM5ODMwMTAyNwRhYwNkZWxNc2dz?mid=1_21857_AERkxEIAANvjSi6wUQ7fiIZa4fY&fid=Inbox&sort=date&order=up&startMid=36&filterBy=
x-requested-with:		XMLHttpRequest
Accept-Encoding:		gzip, deflate
User-Agent:		Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1; SV1; .NET CLR 2.0.50727)
Host:		us.mc575.mail.yahoo.com
Cookie:	MG	d=1vAXIFvaYnFGnmIfzw3zBCVVRe2jUKZLwwyoKSrjasG0XVYaJhF95dLsZ5C0x1eDlcTcaHS_vpiad9XvB0emj5Rr1 v=1
	Y	v=1 n=66k3gh8as55lf l=cc70cc03_01sqqz/o (Yahoo login id: [redacted]) p=m2g265i013000000 (Gender: male, Birth year: 1980, Postal code: [redacted]) r=hq lg=en-US (Language/content: English)

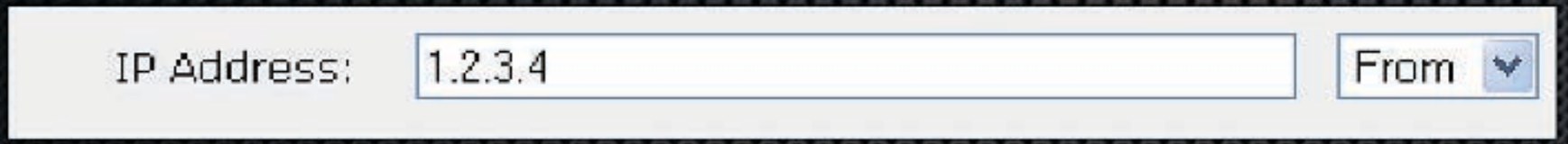
XKS HTTP Activity Search

- In order to account for all of the possible host names, we must front-wildcard the host name.
- Be careful when front-wildcarding because beyond being resource intensive for XKS, it can be dangerous from a USSID18 perspective

Hints for wildcarding a host name

- If you're trying to query for traffic to the website www.website.com the best way to wildcard it is:
- *.website.com
- Notice that the . before the hostname website is still there, that way we will properly hit on ads.website.com images.website.com but avoid the false hits on www.anotherwebsite.com

Hints for wildcarding a host name



Why are we only interested in traffic coming from our IP of interest going to our website of interest?

Helpful GUI Shortcuts

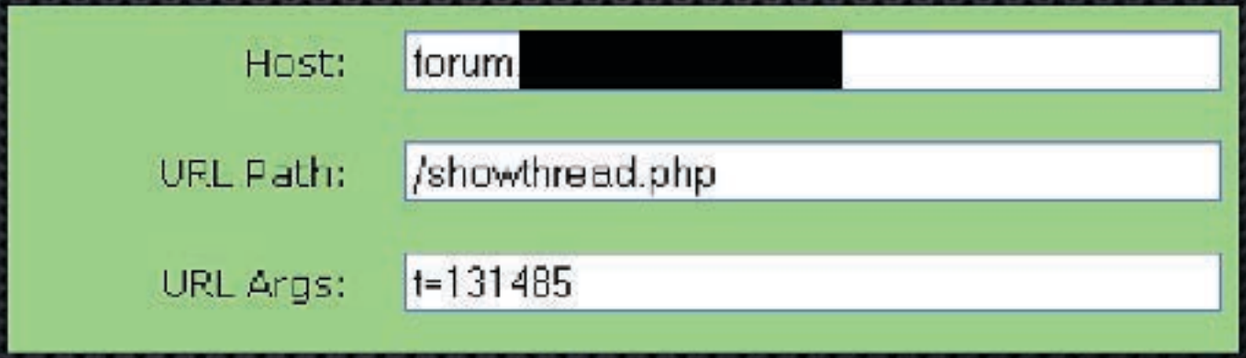
- Earlier we talked about how XKS broke a GET request into the URL Path and URL Argument (separated by a ?)
- Ex: http://forum.[redacted]'showthread.php?t=131485
- Get's broken out to:

Host	URL Path	URL Args
forum[redacted]	**/showthread.php**	**t=131485**

Helpful GUI Shortcuts

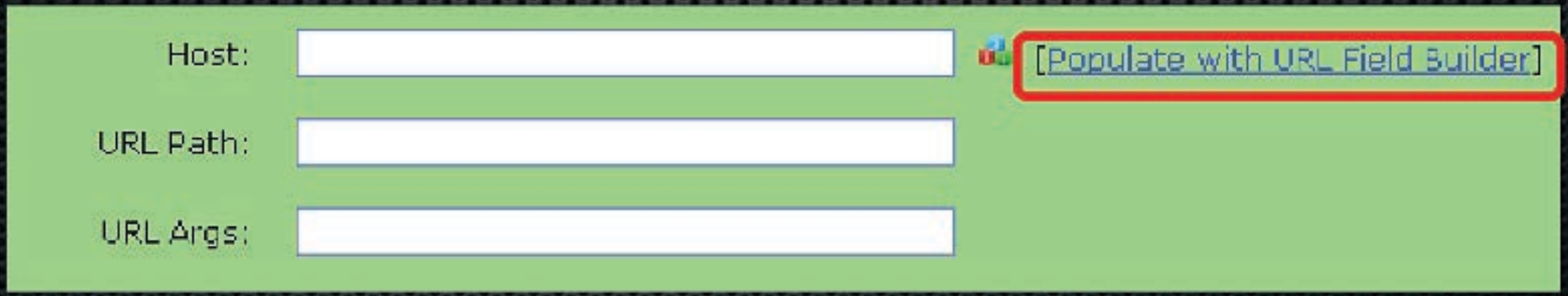
- So if we were to query for this URL we would need to enter those fields in separately:

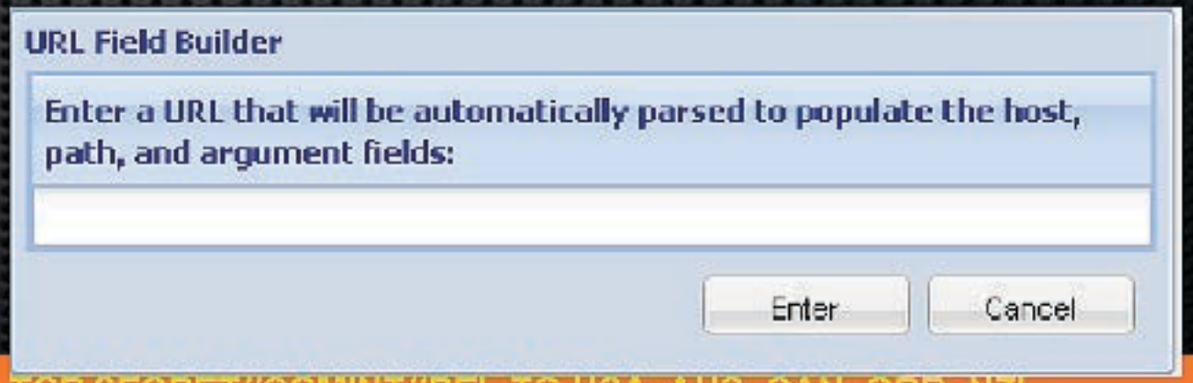
Host	URL Path	URL Args
forum.[redacted]	**/showthread.php**	**t=131485**



Helpful GUI Shortcuts

- Or we could use the "URL Field Builder" to simply copy and paste the full URL and let XKS break it into its appropriate parts:





Helpful GUI Shortcuts