Introduction

Any *search engine* database is a very powerful source of information for web applications. The Search Engine's spiders are well-powered to run frequently on sites and capture all possible links. As an end user, however, we are more interested in the searching interface and criteria these engines provide. Now, if we could combine these search options with our intelligently crafted queries against a database, we would be able to fetch a lot of critical information. There are ways to do this and I shall get to the details very soon.

SEARCH.MSN provides web services APIs to build applications using their search interface. More information can be gathered from http://search.msn.com/developer/

To be able to use SEARCH.MSN, you will require an Application ID. This can be obtained using *MSN passport*. Queries are limited to 10,000 a day and allow a total of 50 results for each query. This provides great flexibility to the application. As a security tool, substantial information can be queried from MSN search, making it a handy tool to have in your toolkit. For the examples outlined in this paper, some of the information is retrieved using this interface, with a sample application called **MSNPawn** **(http://www.net-square.com/msnpawn)**.

MSNPawn has been designed and developed on the .Net framework and must be installed on the system. The following utilities have been bundled with MSNPawn .

1. *MSNHostFP* – Supply an IP Address or IP Address range to fetch all possible virtual hosts or application running on each IP addresses.
2. *MSNDomainFP* – Supply a domain name to fetch the top 50 child domains, considering the supplied domain name as parent.
3. *MSNCrossDomainFP* – Supply an application domain to fetch the top 50 domains pointing to this particular domain on the Internet.
4. *MSNCrawler* – Supply a domain or application name to fetch all possible links crawled by the search engine.
5. *MSNFetch* – Supply a domain and rules file. The tool will run each rule in the file against the domain specified and fetch the first five results of the resultant query. This can help in assessing an application.
6. *Search.MSN* – Provides place to run your search against MSN and gather all URLs.

This paper describes some of the queries that can be run against SEARCH.MSN in order to fetch important information that would eventually help in web application assessment.

Shreeraj Shah
Founder & Director

n e t - s q u a r e
http://www.net-square.com
shreeraj@net-square.com

Table of Contents

Acknowledgement

Lyra Fernandes for her help on documentation.

Web application footprinting with MSN search

One of the challenging tasks for security professionals is to discover web applications belonging to specific clients, using a limited set of information. Often, the only information we have in place is an IP address or a higher-level domain, like, for instance, "icenet.net" (a local ISP of the city). Beginning with this zero-level information, we attempt to see what else we can discover. We shall divide our footprinting exercise into two sections – host-level and domain-level.

Host footprinting

One of the problems faced, is to find reverse DNS lookup in a multihosting scenario. When more than one web applications is hosted on one IP address, it is important to know the correct application host name in order to retrieve information related to this specific web application. This host information can be passed to "`Host:`" in the HTTP header section, while making HTTP requests to specific IP addresses.

Here is a result of a query run using the IP directive – `ip:203.88.128.11` – on a web API interface using MSNHostFP. **Before your run a query you would need to supply an AppID.**

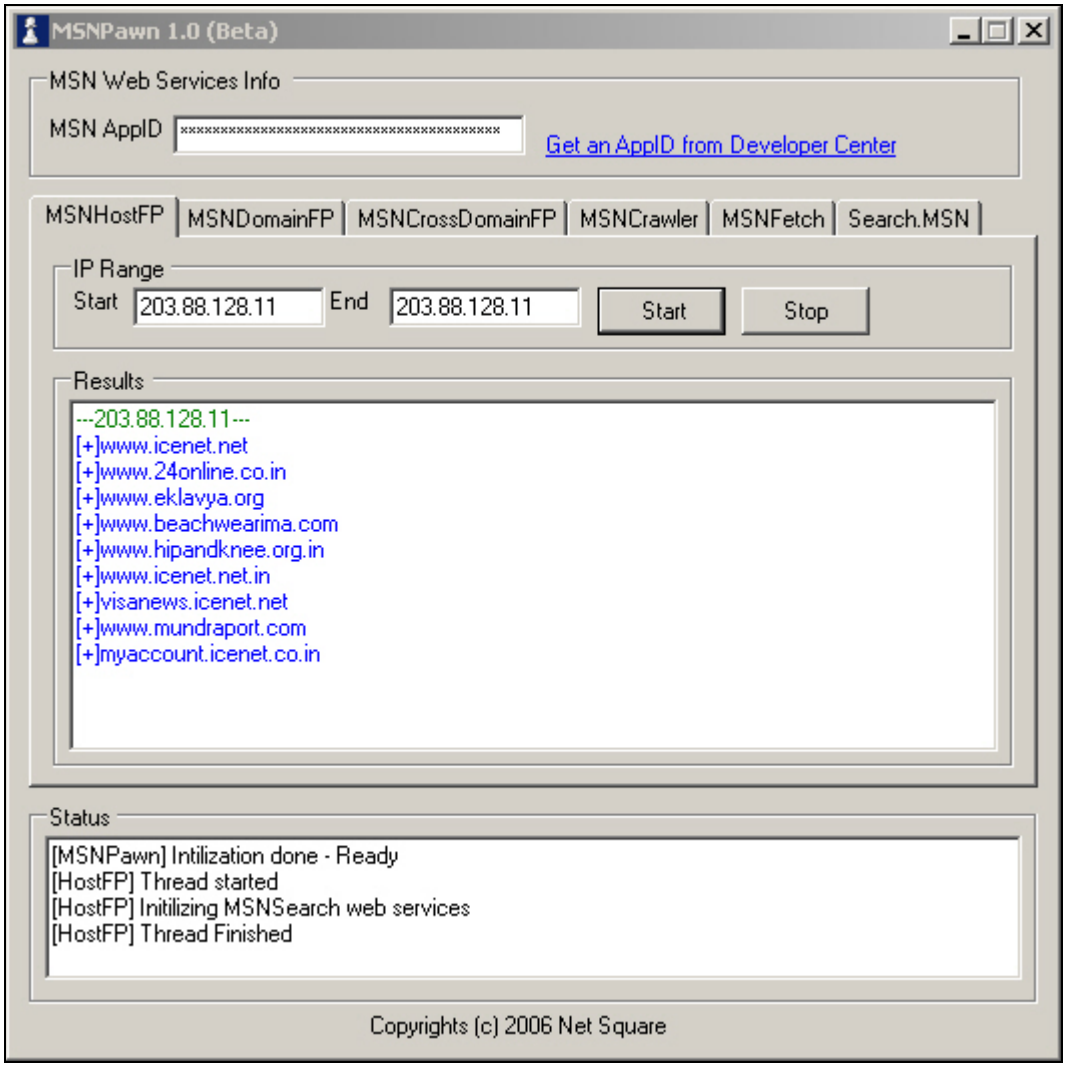

Figure 1. MSNPawn: Using utilty MSNHostFP

We have successfully obtained the above hosts running on specific IP address, `203.88.128.11`. Shown below is a screenshot of running the exact same query on their web interface.

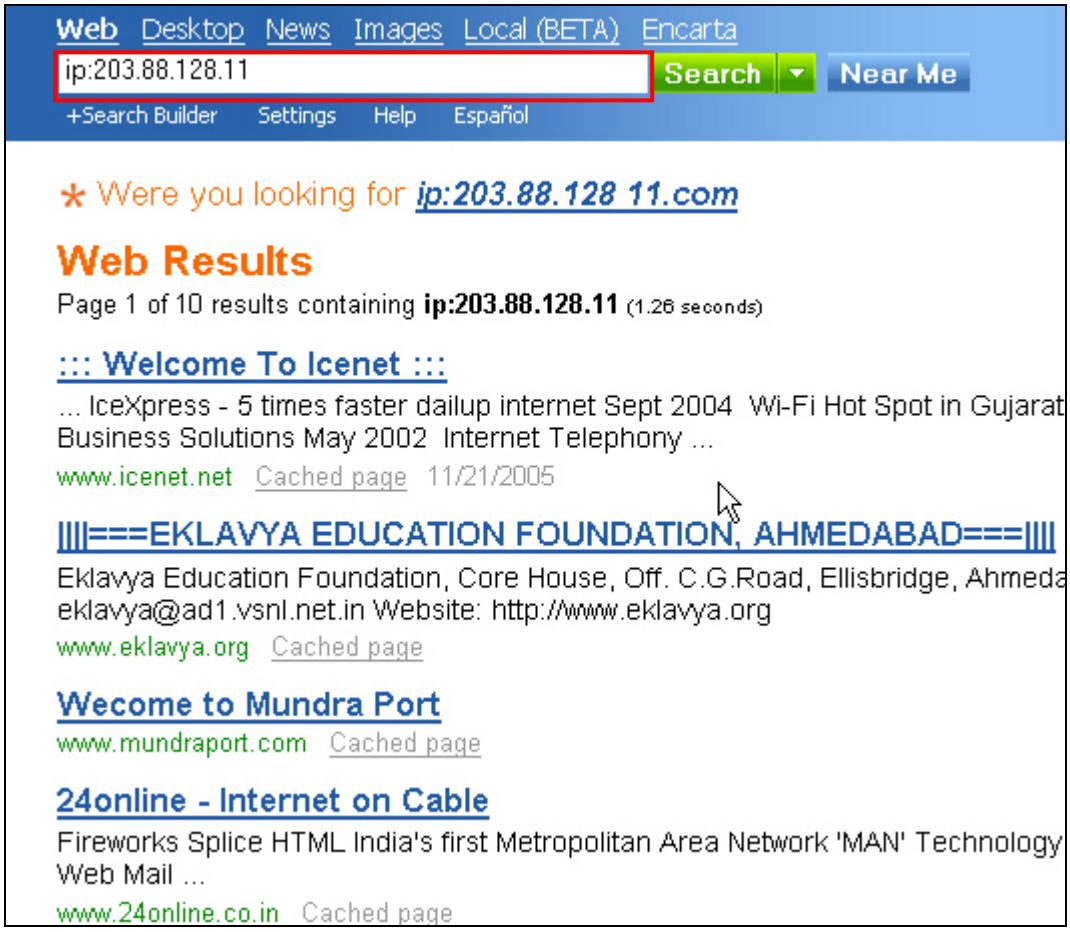

Figure 2. IP Directive using SEARCH.MSN web interface

This directive is very useful in doing reverse host finding when the DNS does not having a PTR record. MSN search discovers an IP address and reports each web application host found pointing to this IP address.

Choosing how search query results appear

It is possible to choose how results of a search query appear. To get a unique value or just one value for each of the sites "discovered", use *settings*; this is to make sure we get just one value.

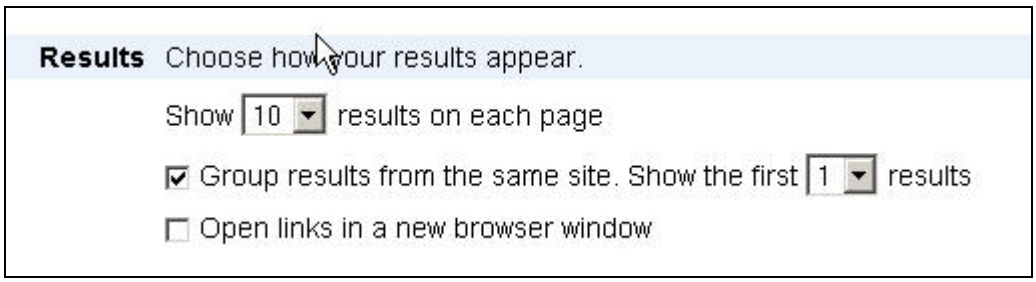

Figure 3. MSN search: settings

In the above example, we were interested only in locating unique hosts pointing to a particular IP address. Now, suppose, we want to repeat this query for an IP address range. The screenshot below shows how you can scan an entire IP address range for virtual hosts.

On the MSNHostFP tab, supply the starting and ending IP addresses as shown below.

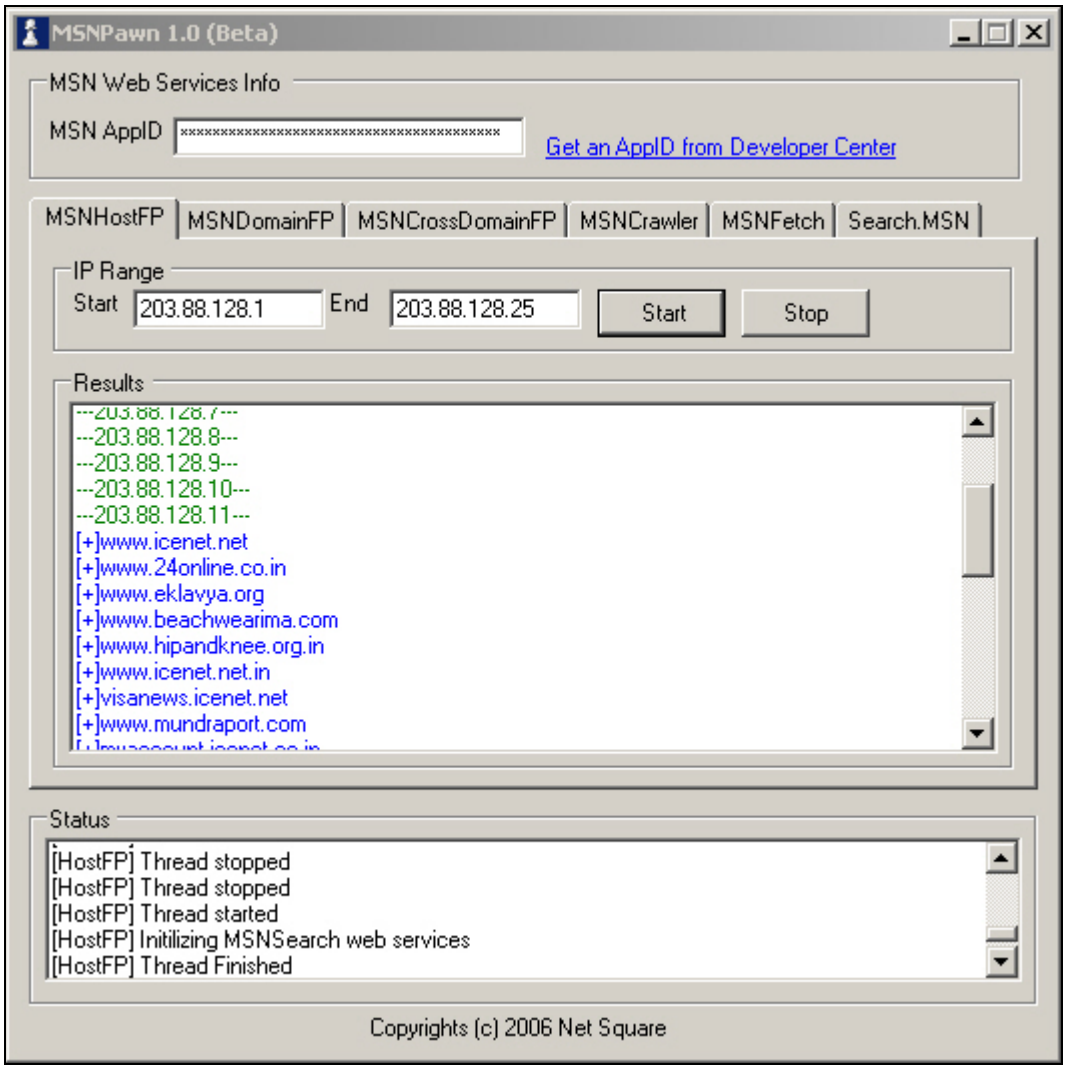

Figure 4. MSNHostFP: Scan an entire IP address range for virtual hosts

Domain footprinting

MSN search supports the "site" directive that fetches all possible applications running on that particular domain and any child domains. For example, typing in a domain name on the tab MSNDomainFP, fetches this result set.

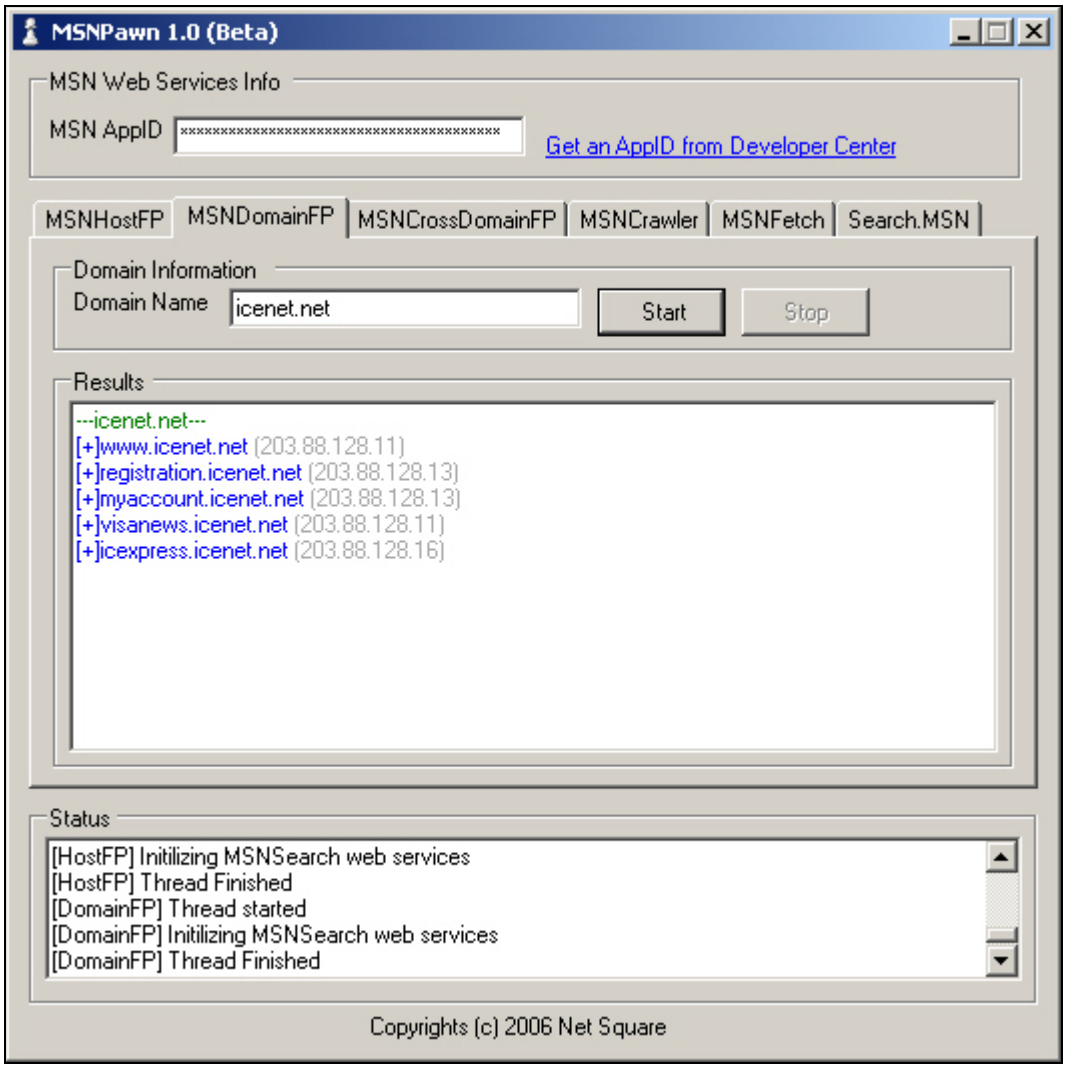

Figure 5. MSNDomainFP: Domain footprinting

The above screenshot shows various applications running on the "`icenet.net`" domain. All applications running on the child domain are also fetched. This is the easiest way of getting all applications belonging to a sample domain.

Using “inurl:” will also fetches similar results. Here is how you can get the same information from their web interface instead of from web services.

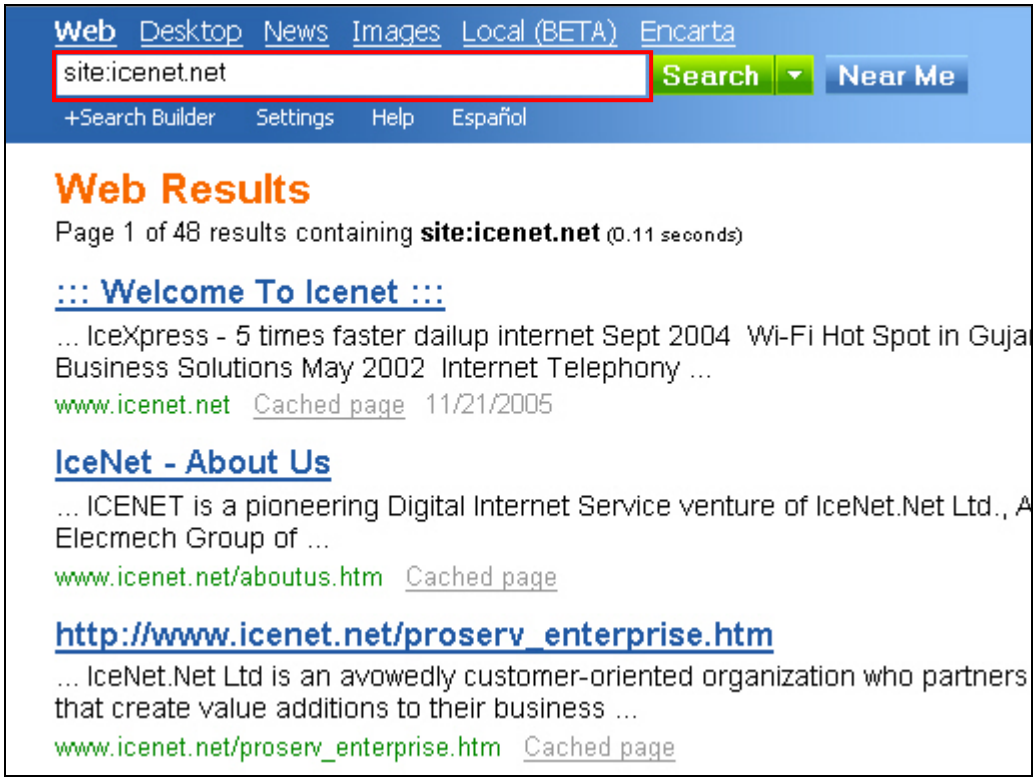

Figure 6. Using the web interface to fetch information

Getting cross-domains pointing to a domain

Cross-domains are domains that point to the application but do not reside on the same domain despite belonging to the same client or group. Such domains cannot be footprinted using either DNS or the "site" directive. But if we are able to analyze and somehow obtain a list of web applications or sites that are pointing to this particular application, we will be able to get access to all cross-domain references.

An example should make things clearer: Let us assume that the application called "`www.icetel.co.in`" belongs to the "`icenet.net`" domain and is part of their IP address range. How do we go about footprinting this domain? In other words, is there a way to discover domains or applications that are pointing to the "`icenet.net`" domain?

Incidentally, there is a way. The "`linkdomain`" directive on MSN does precisely this – retrieve a list of pages that point to any page residing on this particular domain. Interestingly, the following query can fetch an important list of hosts and domains.

linkdomain:icenet.net –site:icenet.net

There are two parts to this query: one, we search for "`linkdomain:icenet.net`" and obtain all results and the second, we negate(–) any of the results that are part of "icenet.net" by using the condition "`-site:icenet.net`". Shown below is the screenshot of MSNCrossDomainFP.

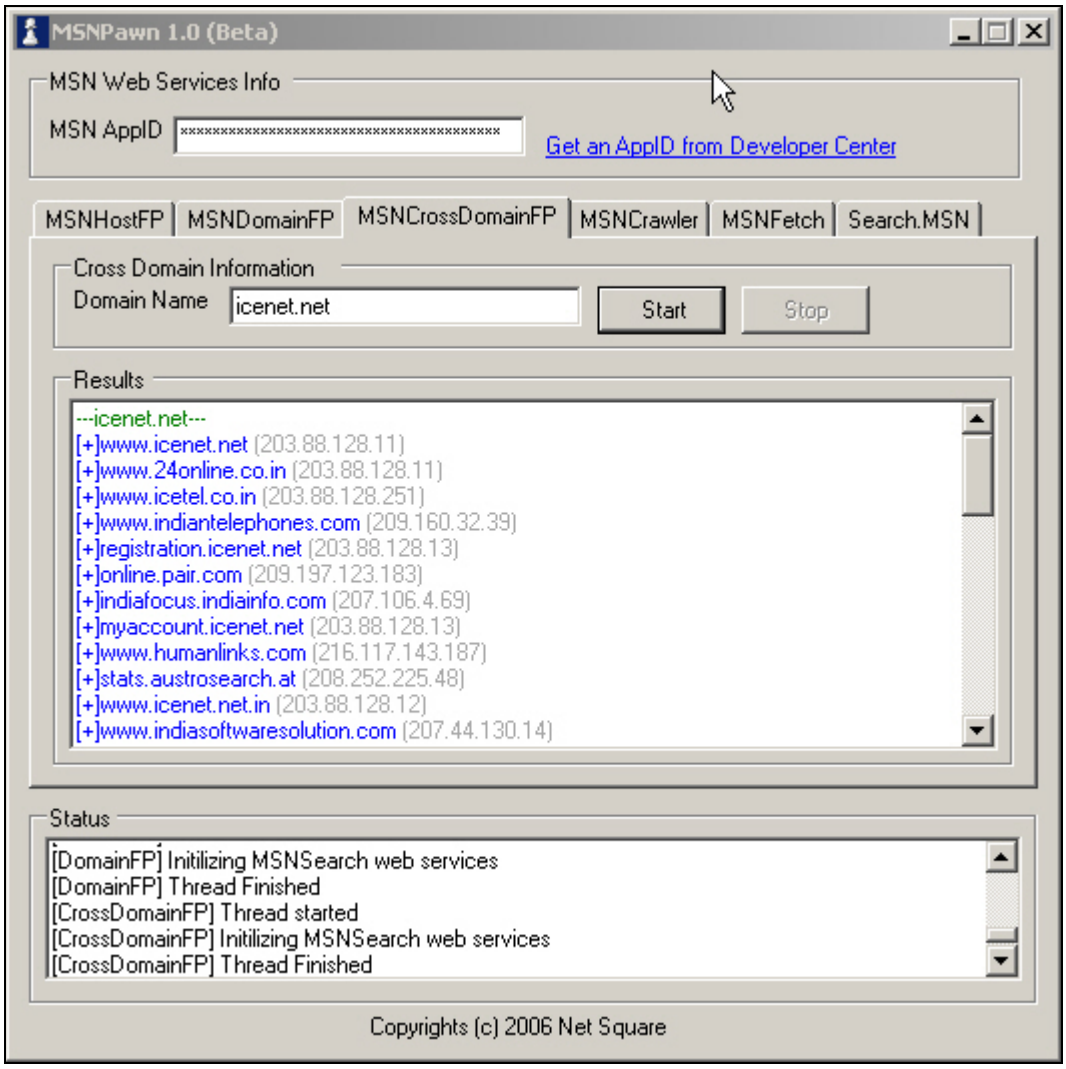

Figure 7. Fetching cross-domains

And here's the result from the web interface.

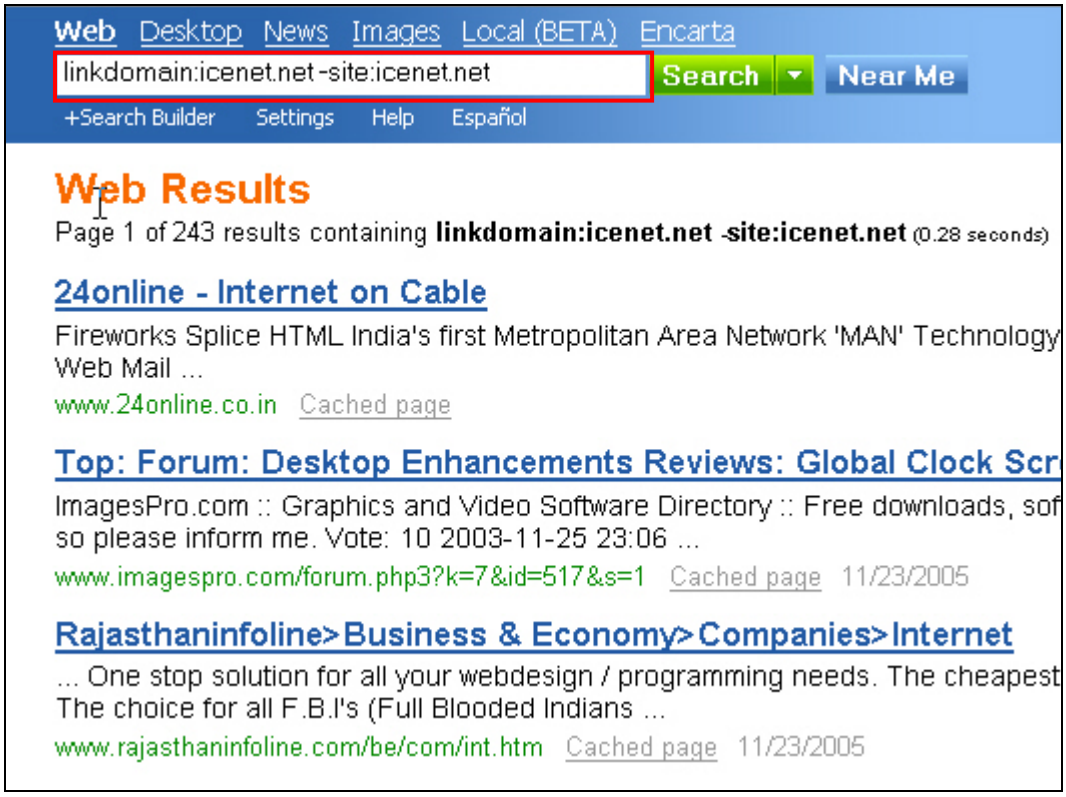

Figure 8. Cross-domain footprinting using the SEARCH.MSN web interface

This cross-domain harvesting method gives us access to another set of applications that belong to the same family or group, based solely on their IP range. Needless to say, "`linkdomain`" is an interesting switch to explore while performing web application footprinitng; it may throw up some unexpected sets of results.

Tricks and tips for web application profiling & assessment

1. Web application profiling

We can also use the "site" directive to grab all links for specific web applications. Simultaneously, we can also fetch cached pages and perform HTML sifting that would provide information such as *forms*, *applets*, *objects*, etc. Then, with this information in place, perform resource mapping on the application in order to define attack points for SQL injection or Java decompilation.

You can use MSNCrawler to collect all possible links,

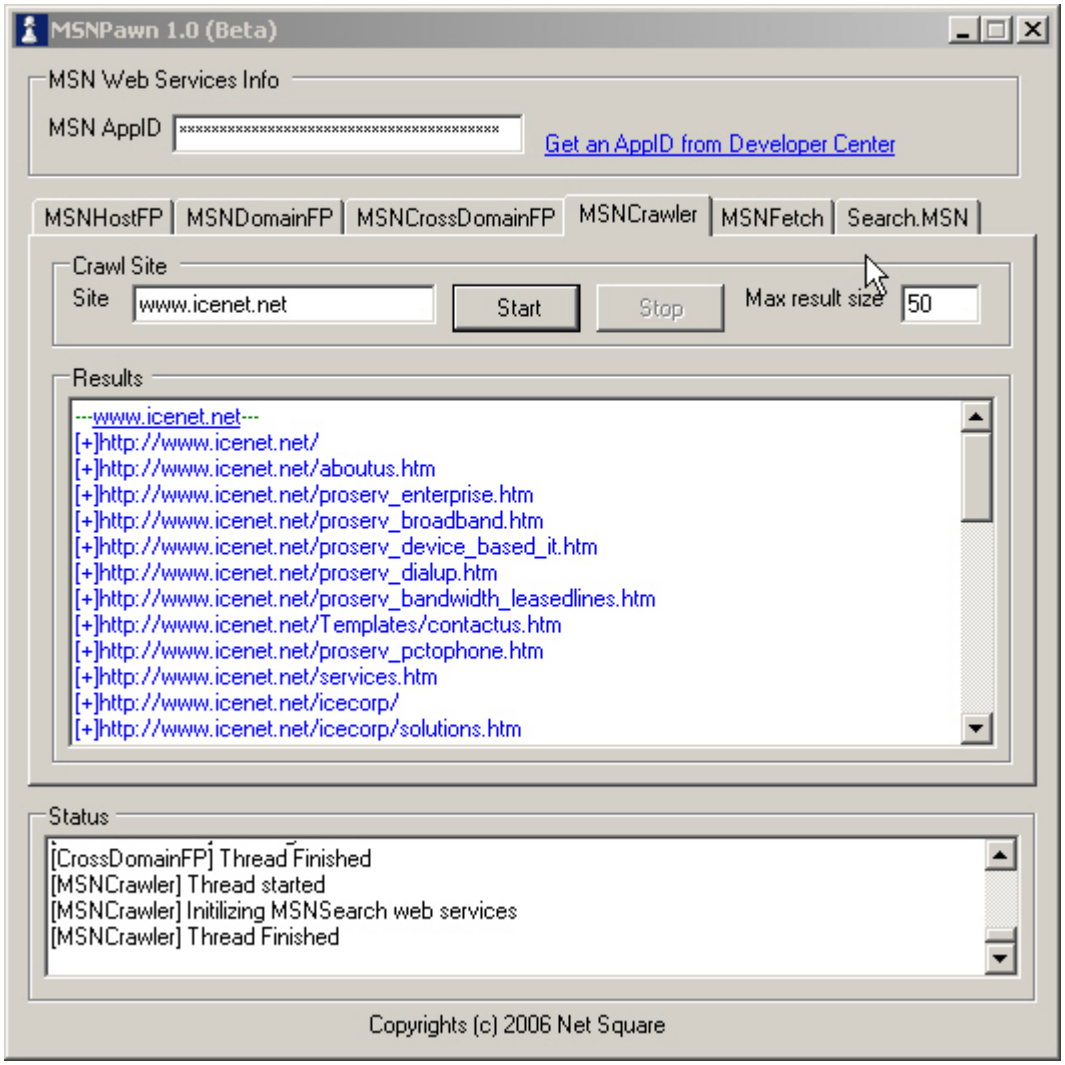

Figure 9. Collecting links using MSNCrawler

2. Assessment and file search

MSN provides directives like "`contains`" and "`filetype`". `contains` provides a page which points to specific file types. For instance, the following search would return a page location that points to the location of the "pdf" files, residing on, say, the "icenet.net" domain.

site:icenet.net contains:pdf

This directive can help in profiling high value targets and resources which point to important resources. "`filetype`" is the directive that can help in locating different file extensions such as *html* and *pdf*.

3. Page scrubbing with in* directives

It is possible to look for specific information at specific locations within an HTML page with different directives like `inurl, inanchor, inbody, intitle,` etc. These directives allow devious search queries to be built to look for specific information. For example, to find all PHP pages residing on the"icenet.net" domain, our query would be "`site:icenet.net inurl:php`". Such queries assist in web application assessment for large domains.

To build a selective list of queries using these various options, all you need to do is to create a *rules* file and pass it to MSNFetch. This will run each rule against specific sites and grab the first five results.

A sample rule file for technology identification –

```
# Technology checking
inurl:asp
inurl:/servlet/
inurl:jsp
inurl:php
```

Running MSNFetch with the above rules,

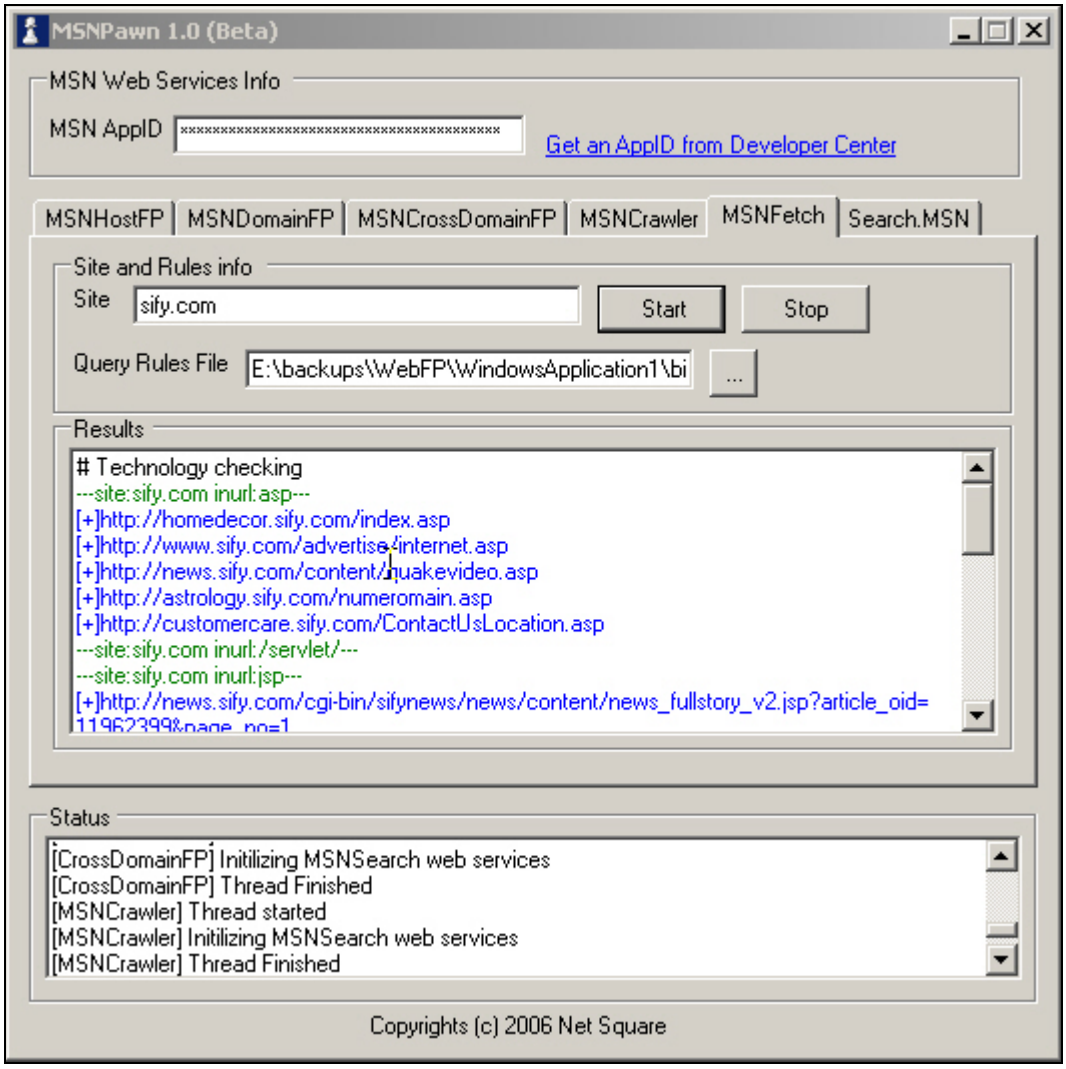

Figure 10. Passing the *rules* file to MSNFetch

4. Tuning search results with interesting directives

One of the interesting directives that MSN search supports is selecting parameters for tuning search results (shown in the screenshot below):

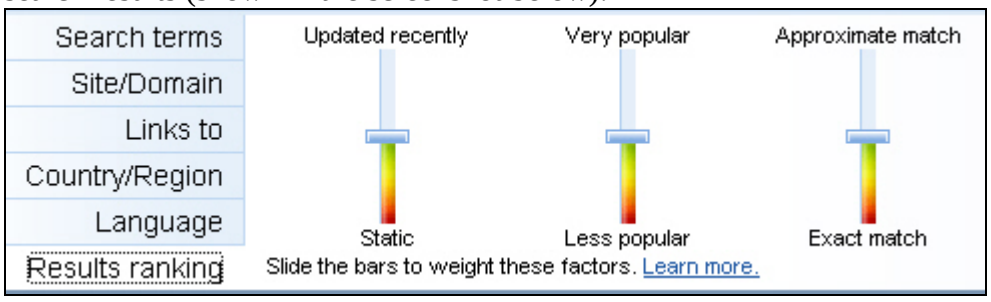

Figure 11. MSN Search: Tuning search results

This can help in generating "fuzzy" search results as well as recently updated pages. Using this directive we can locate the most recently updated pages and can differentiate these from the last set of links or pages collected. This differentiation method is required in order to generate incremental security assessment and perform assessment on newly added resources from the clients.

For example, here's how we can obtain fresh search results for "icenet.net":

```
site:icenet.net {frsh=100}
```

5. Restricting query with respect to location

MSN has another interesting directive called "loc" which can be used to find specific resources located in specific countries. For instance, to see all pages of the "icenet.net" domain residing in India only, our query would be "`site:icenet.net loc:IN`". This can be extremely useful in cases where the scope of assessment on large domains is limited to specific geographic locations.

Then run a customized query and get multiple URLs in one shot by using the *Search.MSN* option on the tool.

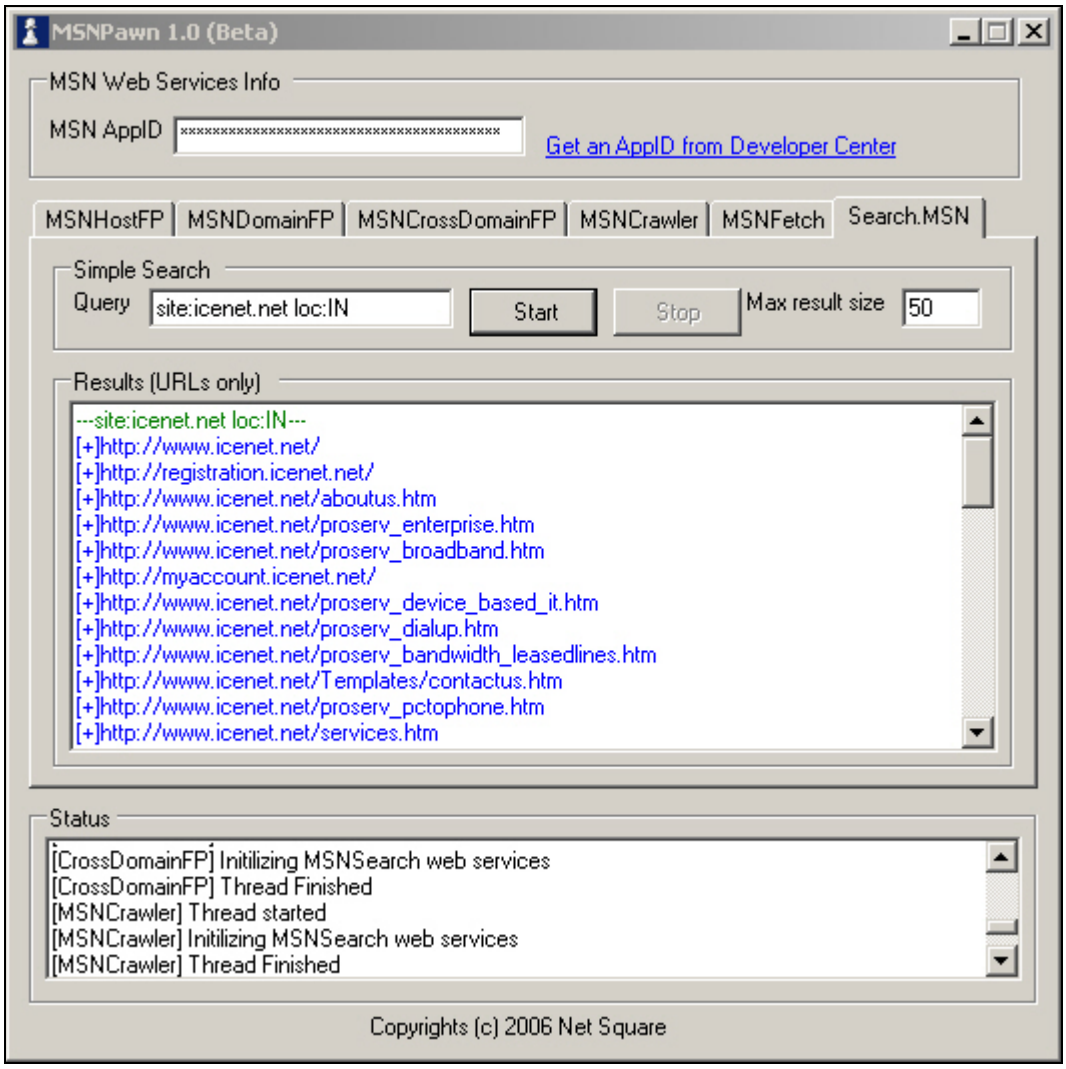

Figure 12. Using directive *loc* to restrict search results to a specific country

Conclusion

Web application security assessment is always a challenge; more so, when beginning with zero-level information about the application. Out of intense complexities emerge intense simplicities.

While there are several tools out there that query the Google database and fetch this sort of security-related information about web applications, there aren't many that fetch the information by using Search.MSN's web services.

By utilizing the extremely powerful search options provided by the MSN search engine to construct intelligent queries that fetch critical information, this article seeks to offer simple solutions to footprinting web applications in different domains or those that are mapped to a single IP address.