F-Secure

RIDDLER

READY TO EXPLORE THE DEEP WEB?

THE STORY BEHIND RIDDLER

WHY CHOOSE RIDDLER?

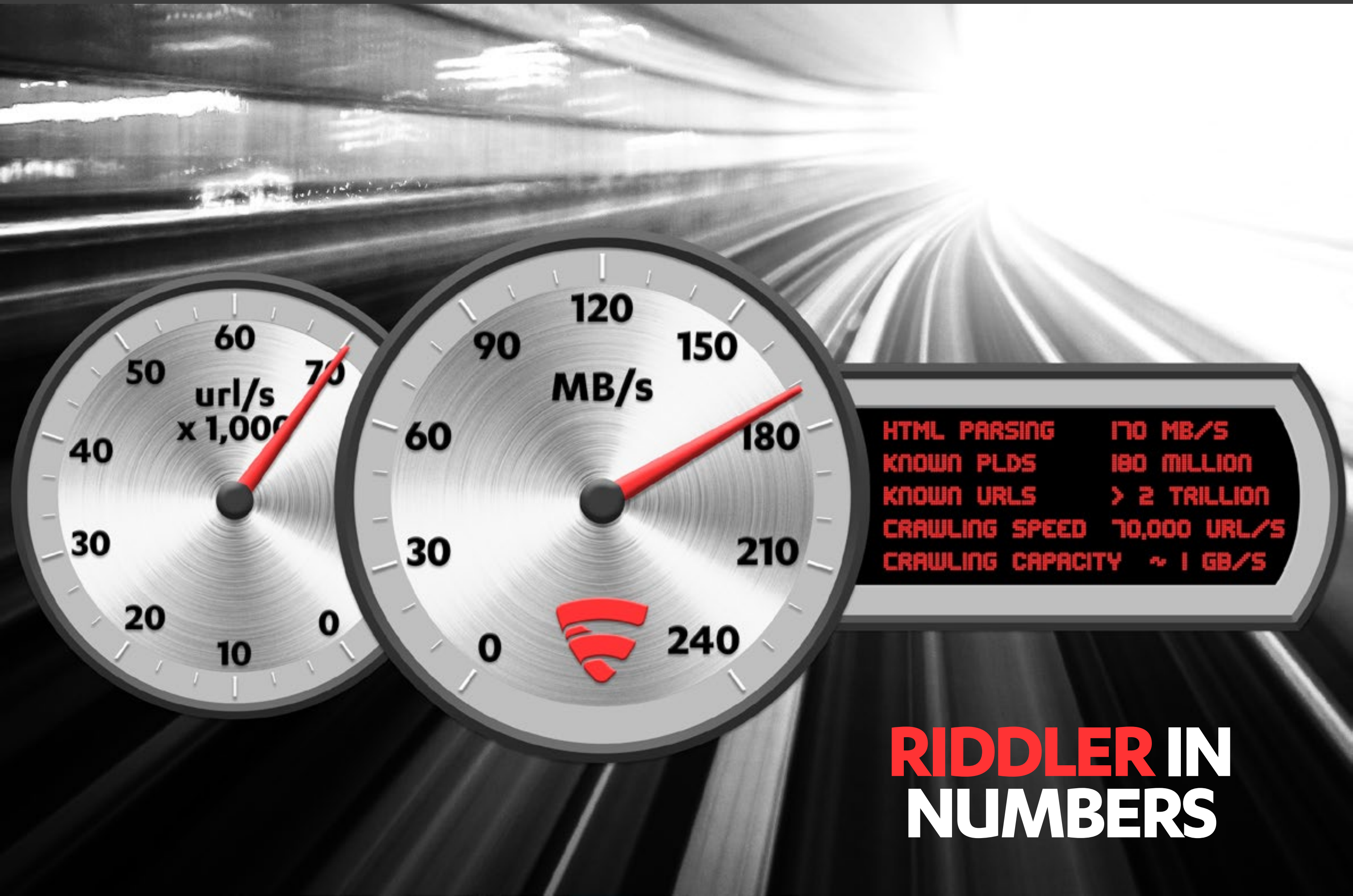
HOW RIDDLER WORKS

F-Secure white paper - www.f-secure.com

WHAT IS RIDDLER?

Riddler is a tool for web topology mapping, attack surface enumeration and web discovery. It is available as an online search engine (riddler.io), a set of tools, an API, and as a managed service. It is also available as an optional plugin for F-Secure Radar.

This white paper will explain what Riddler is, how it works, and what you can do with it. We'll go on to detail a variety of use cases that can be realized with Riddler, and provide you with a list functions within your organization that we believe could utilize the power of Riddler to their advantage.



HTML PARSING	170 MB/S
KNOWN PLDS	180 MILLION
KNOWN URLS	> 2 TRILLION
CRAWLING SPEED	70,000 URL/S
CRAWLING CAPACITY	~ 1 GB/S

RIDDLER IN NUMBERS

THE STORY BEHIND RIDDLER

Riddler was started as a research project by two of F-Secure's employees back in 2012. The aim of the project was to develop an Internet crawler (similar to those developed by Google and Bing) capable of running on low-end hardware.

When the Riddler project was started, commercial port scan services designed to scan the entire Internet were already being developed and showcased. We figured those sort of services would end up becoming commodities (and we were right). Our aim was not to replicate them; it was to build something different.

Commodity port scanning services such as Shodan and Censys are useful for finding different devices and services on the Internet. They work by attempting to grab banners from specific pre-configured ports across the entire IPv4 range. Information gleaned from these banner queries is then stored in a database for subsequent lookup, and, of course, refreshed on a periodic basis.

The banner grabbing methodology used by commodity port scanning services is often deemed somewhat intrusive. Commodity port-scanning services must maintain lists of IP ranges that disallow this style of probing, and deal with abuse reports on a somewhat regular basis. Full Internet port scans (such as those performable by ZMap) and banner grabbing tactics can often cause the source IP address to be automatically blocked by firewalls (SYN packets to closed ports often trigger firewall rules). Hence these approaches, while good for finding interesting publicly exposed services, are incapable of seeing everything on the Internet.

So why not use a freely available crawler, like Google?

Conventional search engines rank pages in order to return what they consider the most relevant information back to their customers. Pages are ranked based on things like the Alexa top one million domains. Not all pages on the Internet are indexed. Furthermore, some pages are censored (for political, social, economic or corporate content). Censored pages are omitted entirely from indexing. None of the freely available crawlers support crawling on a massive scale; they are aimed at indexing millions of pages, not billions or trillions. Conventional search engines don't index what's commonly referred to as the deep web.

In addition to not indexing the whole web, conventional search engines don't support a query syntax that would allow a user to describe the topology of the Internet. There's no way of asking Google for a list of Apache servers running in Finland, or all servers containing the string "webmail" under helsinki.fi. If you're looking to run a threat assessment, you're going to need to be able to

BANNER GRABBING

Banner grabbing is a technique used to find services running on open ports of a system connected to a network. Historically, netcat or telnet were used to connect to a specific port on an IP address, look for a reply indicative of a service running on that port, and parse metadata from that reply (to figure out what was running there). Nowadays, nmap is used for this purpose.

RIDDLE ME THIS

What happens when you put a bunch of techies in charge of naming things?

F-Secure Cyber Security Services have offices in Copenhagen, Oslo, Poznan and Helsinki. With the geeks in charge, there's a bit of a Batman theme going on. Two of their meeting rooms are called "Gotham" and "Metropolis". The theme stuck, and since then other things started picking up Batman universe names. Like Riddler.

query for these sort of things. We'll get into why that's important later in this paper.

So what does a web crawler do? It's quite simple.

1. **Visit a web page.**
2. **Parse the page for links.**
3. **Feed each parsed link into a queue, if it hasn't been visited recently.**
4. **Feed each link from the queue back into step 1.**

In order to crawl the entire web within a reasonable amount of time, we determined that we'd need two things: a fast HTTP request engine and an efficient way of performing a URLSeen() test. After running some numbers, we figured that we'd need to be able to perform between 50,000 and 70,000 URLSeen() tests per second. At that point, work started.

Given that Riddler started as a proof-of-concept project without a great deal of investment, we were constrained to running it on hardware we had available in the lab. Here's what we had:

- **2 rackmount parser servers with Intel Core i7 2.8GHz, 64GB RAM and internal 256GB PCIe SSD drives (storage buckets)**
- **1 rackmount crawler + queue builder + storage, Intel Core i7 3.6GHz, 64GB RAM**
- **2 external 3TB USB drives (spinning disk)**
- **A 1 Gb/s network connection**

At the time, it was probably about 5,000 EUR worth of equipment.

Building a system capable of performing 70,000 URLSeen() lookups on the hardware we had available was not as easy as we'd first imagined. None of the off-the-shelf database solutions we tried were able to handle the task (we're talking SQL, noSQL, commercial, etc.) Caching, memory use, disk reads, and storage overhead were all contributing factors to their inadequacy. In many cases, disk head movements were a problem that became further exasperated as fragmentation occurred. Often, a database solution worked nicely until it had exhausted physical RAM. Then it slowed to a crawl. That's not something you can live with when you need a system to run for weeks or months without maintenance (read: defragging).

THE DEEP WEB

Put simply, the deep web are parts of the World Wide Web that aren't indexed by conventional search engines. The deep web includes things such as web mail, online banking, and services behind pay-walls. The deep web is estimated to be several orders of magnitude larger than the shallow web (those parts that are indexed) and growing at an exponential rate.

CLOSE TO 2,000,000,000,000 UNIQUE LINKS?

When running the crawler, we noticed that somewhere between and half and two-thirds of all links processed by URLSeen() calls were unique. At 70,000 new URLSeen() calls per second, we estimate that roughly 35,000 are unique (so a bit of a low estimate). Multiply that by the number of seconds in an hour, the number of hours in a day, and the number of days in a month (which we'll set at 30) and we get 1.5 trillion. So the 2 trillion number seems about right.

Attempts at reaching our intended performance goals failed repeatedly on both the storage technology and request engine fronts. After several months we finally ended up building our own custom HTTP request engine and database storage solutions from scratch.

It was worth it. The system we finally put together was capable of a 1Gb/s sustained crawling speed, which translates into about 2000 separate web requests per second. Our database was capable of doing 2.5 million lookups per second and 1.7 million inserts per second. We achieved the 70,000 URLSeen() tests per second we wanted on the target hardware.

Our crawler was capable of visiting all of the web sites in Finland in less than five minutes. So we crawled the whole web. It is estimated that commercial search engines index about 700 million websites and between 30 and 40 billion separate documents at the time of writing. Our database, after crawling for 30 days, contained close to 2 trillion unique links.

We could now do attack surface enumeration on a global scale. ***grin***

RIDDLER
USE CASES

INFORMATION SECURITY

Threat surface assessment

Your InfoSec department will probably be most interested in using Riddler. Members of the IT department, security experts, incident response teams, and crisis management teams all need access to up-to-date threat surface assessment reports.

LEGAL

Monitor for brand infringment etc.

Legal would appreciate a service capable of monitoring for brand infringement or other fraudulent activities related to your company's brand or intellectual property.

THREAT ASSESSMENT AND PENTESTERS

Locating the weak spots

Anyone in your organization tasked with running threat assessments, network security assessments, or penetration tests will find this tool invaluable, especially when combined with other tools on the market.

MARKETING AND BUSINESS DEVELOPMENT

Manage referers and identify opportunities

A Riddler service designed to monitor and report on referrers will allow your sales, business development and marketing guys to identify potential opportunities.

RIDDLER USE CASES

Combining the vast amount of data obtained from crawling the entire web, including the deep web, with search and filtering capabilities outlined in this paper, Riddler can be used to fulfill a wide variety of tasks. Use cases range from threat assessment (identifying risk concentration, unintended interdependencies, shadow IT, and subcontractor entropy) to business intelligence (identifying potential business connections and brand monitoring). We'll outline a few in this section.

Domain and Web-level discovery

By using **`pld`** and **`ip`** searches, the topography of an organization can be quickly and easily mapped. This can be useful, for instance, if you're expecting to do business with a new company or organization, if you're looking to partner with a third party, or if you're thinking of hiring a sub-contracting firm. If a third-party is going to be working closely with your organization, it's important to know if their network or security practices might present a risk to your own organization's security.

Mapping your own attack surface

Threat assessment tools are good at finding vulnerabilities and exposed systems, as long as you tell them where to look. They're not generally great at finding all the systems they're meant to be examining. Using Riddler, you can generate a map of your organization's Internet topology, feed the data into a threat assessment tool (such as F-Secure Radar), and discover extraneous, vulnerable, or misconfigured systems.

Protecting outsourced IT

Many businesses rely on outsourced IT services, hosting services, or subcontractors. Working with such partners often requires an organization to implicitly trust the security practices of their external providers. Riddler can help in the process of auditing these service providers' security practices. Audits can be run by in-house staff or outsourced as a service from F-Secure. Identifying possible breach ingress points or potential targets for lateral movement is important when your own security policies may differ from those of a partner. This is especially the case if you're relying on a partner to host services for your company.

DEFENDING AGAINST MISCONFIGURATIONS AND EXPOSED (INTERNAL) ASSETS

At the time of writing this paper, we queried Riddler for the number of hosts with the word "webmail" in their FQDN from a variety of countries. The results speak for themselves:

```
countty:fi host:webmail ==  2140 results
country:fr host:webmail ==  4334 results
country:gb host:webmail ==  9460 results
country:se host:webmail ==  1372 results
country:ee host:webmail ==   121 results
country:dk host:webmail ==  2347 results
country:de host:webmail ==  8634 results
country:au host:webmail ==  1845 results
country:us host:webmail == 42250 results
```

The fact is, companies often unintentionally expose internal assets to the Internet. This is mostly due to misconfiguration or negligence. These are prime targets for hackers.

Eliminating orphaned hosts and applications

It is not uncommon for an organization to lose track of systems and services they've deployed. Perhaps a service was set up by an employee who no longer works at the company, perhaps a host was deployed without the knowledge of the IT department (shadow IT), or perhaps ownership of a service was dropped when a reorganization occurred. The fact is, these things happen, and happen fairly often. Orphaned or shadow systems often provide easy ingress points for threat actors, since they're seldom administered or patched. A few simple searches in Riddler can rapidly uncover such systems.

Phishing and brand monitoring

Companies with successful intellectual property or brands are often the target of fraudulent or malicious activities. These activities can include brand violation (third parties posing as your company), phishing sites (intended to trick, scam, or infect visitors), and typo-squatting (registering domains using misspelled words similar to your brand in an attempt to obtain traffic from people who typo'd a search query, or weren't paying close attention when clicking on a link). Many companies have little to no awareness of these sorts of activities.

Monitoring for signs of brand infringement and typo squatting is a recommended practice - fraudsters tend to move onto other brands if they are caught in the act, and often to rely on the fact that most companies don't actively monitor for brand infringement. Providing a selection of queries to the `host` handler in Riddler will allow you to find instances of brand violation fairly easily.

Automated threat surface monitoring

By running a Riddler monitoring service against your company's infrastructure, and feeding gathered intelligence into your own asset management systems, an up-to-date picture of your organization's Internet topology can be easily determined and kept up-to-date. This data can be used to automate inventory and can be fed into threat assessment tools (such as F-Secure Radar). With such a system in place, your organization should be able to react quickly to attack surface changes, whether it be from a new vulnerability patch, a new threat, or an effort to replace deprecated services (such as SHA1-based authentication systems).

Some worked examples

Riddler is best illustrated with some real-life anecdotes. Here are a few.

Fortune 500 threat assessments

Attackers looking to breach organizations can follow several modus operandi. Some attackers go after a specific target, perform thorough recon, look for vulnerable spots, and perhaps even work on finding new vulnerabilities to exploit. In this analogy, an attacker knows where he wants to break in, and is looking for locks to pick or half-open windows to climb through.

Consider, however, the case where a hacker has access to a weaponized exploit, is willing to use it, and just has to find a matching vulnerable system. In this case, the attacker has a set of keys, but doesn't know which doors they open.

In the latter case, the ability to easily search for specific services on the Internet will allow the hacker to pinpoint high-value targets. In essence the attacker can build a "shopping list" of targets and hit them in priority order (or simultaneously).

Easy exploits for some old systems (such as web servers) can be found rather trivially by searching Google. You can even search YouTube for instructions on how to perform a variety of rather effortless exploits. After just a few simple Riddler queries, we found external systems on the networks of many Fortune 500 companies that can be exploited with these techniques. It took just minutes to do the whole thing.

For example, while querying one Fortune 500 domain, we found that several of their sites were running Sun One Webserver. We did a quick Internet search for Sun One Webserver vulnerabilities, and found a trivially easy hack that applied to version 6.1 of the software. The company in question had multiple instances of this server version open to the public Internet.

The whole investigation took about five minutes. With Riddler available to them, IT staff at the company in question could easily pinpoint vulnerable systems like these and prioritize patching or decommissioning before a hacker performs a similar search.

Discovering a DNS hostname leak

By querying the pay-level domain of an Internet Service Provider, we discovered that a lot of hostnames had leaked into their DNS servers due to AXFR requests to the ISP's customers' home routers. Most of these routers were Zyxel brand, but luckily they weren't configured with default passwords. However, given that these IP addresses were now available, and on the Internet, it didn't take us long to find out that there was also a weaponized RCE exploit available for the Zyxel routers in question.

Server clean-up

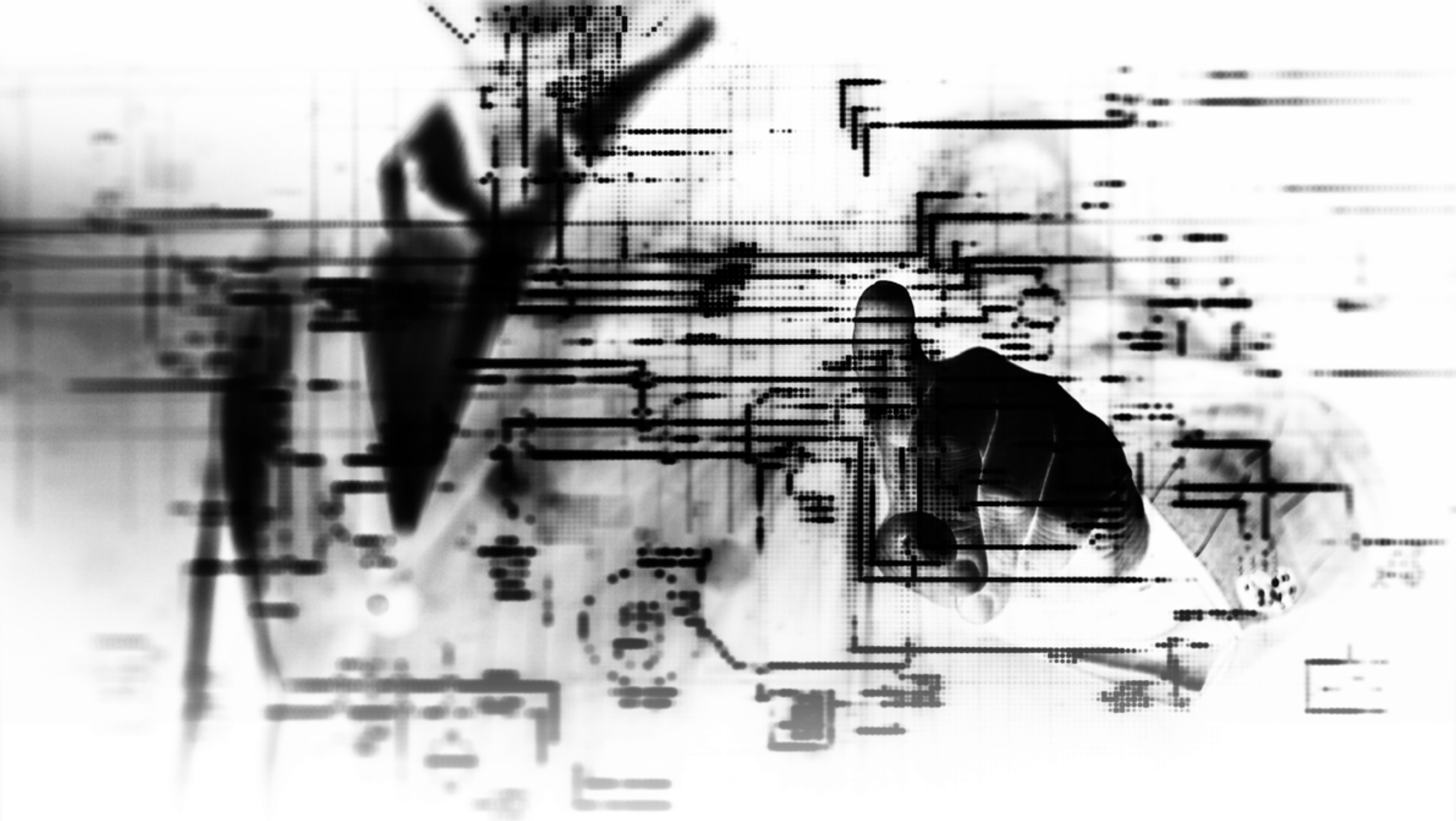
A large enterprise customer of ours have been using both Radar and Riddler to map out their organization's attack surface. They recently found around 10 hosts that they were immediately able to take offline due to the fact that they were unneeded. This is a nice example of removing shadow IT.

Tracking fraudsters

By running the query **`tld:xxx country:dk`**, we were able to identify an individual who was engaging in large-scale typo-squatting in Denmark. We tracked the purchases of .xxx domains linked to several banks, insurance companies, and even the police (politi.xxx). This particular miscreant was easy to track with a single Riddler query.

Being a good neighbor

We often find open admin interfaces to Internet-connected infrastructure. Examples have included windmills in the Norwegian Sea, parking meters in Denmark, and Internet-connected solar panel farms.



Available Services

Riddler is available in several different flavors, depending on your organization's needs, budget, and expertise.

Riddler Foundation

Subscribing to the Riddler Foundation service gives your organization access to the following:

- **Up to 5000 results from any search query**
- **A maximum of 2500 daily queries**
- **Full access to the web interface at riddler.io**
- **Tools which include expanded functionality for mapping internal networks**
- **Full access to Riddler's API**

Choose the Riddler Foundation package if you're looking to build your own tools based on Riddler or if you've got experts in your organization who want to play around with Riddler queries by hand.

Radar integration

The Riddler service is available as an optional Radar plugin. With Riddler integrated into Radar, results from network topology mapping can be fed directly into our fully-featured threat assessment tool. With the full riddler.io search syntax available directly in Radar's interface, administrators and security experts can quickly enumerate and assess their organization's attack surface.

Regular vulnerability scanning and management should already be an integral part of your organization's security culture. Adding the powerful combination of F-Secure Radar and Riddler will provide you with capabilities that are a cut above other commercial threat assessment platforms.

Managed services

Many of the use cases detailed in this paper are geared around configuring Riddler to monitor specific sets of queries and either generate alerts or periodic reports. This is where Riddler managed services shine. By working with your organization, our experts can design, maintain, and manage monitoring systems, alerts and reporting precisely tailored to your requirements. And they'll continue to work with you as your needs, or the landscape changes. All of the use cases detailed in this white paper are available as managed services, and if you come up with a use case that we've never even considered, we'd be more than happy to set that up for you!

Choose Riddler managed services if you're looking to take a hands-off approach and have searches, alerts and reports delivered directly to you, if you're not sure how you might use Riddler to generate the queries you have in mind, or if the number of queries you expect to make are likely to exceed what's available in the Riddler Foundation package.

HOW RIDDLER WORKS

Unlike commercial port scanning services, which enumerate through the IPv4 range, Riddler crawls web pages. When the system is initialized, it is seeded with a collection of URLs taken from the Internet Census and from a list of Finnish domains. From there, it visits sites, collects links to other sites and continues to crawl through the Internet. Since there are no bounds to the Internet, the crawlers essentially just continue to browse indefinitely, refreshing old information once its time to live expires.

As pages are visited, they are fed through parsers which extract URLs and relevant metadata. These URLs are then fed into a queue builder which processes them (with URLSeen()), stores them into a database (for further analysis) and, if needed, adds them to a queue. Multiple crawler threads pull new URLs from the queue, attempt to visit those URLs, and if successful, feed content back to the parsers.

URL DNS resolution is performed in the queue builder system to prevent crawlers from inundating public DNS servers with requests. Parsers receive an IP address in conjunction with a URL when they pull a new entry from the queue. Crawler nodes connect directly to the IP and port to request a document.

In order to prevent sites from getting flooded by requests from crawler nodes, the lookup queue is dynamically built to prevent requests to any particular IP from occurring more than once per few seconds.

If a site has a properly configured robots.txt, our crawlers will honor it, and skip indexing completely. If you'd like your site to be ignored by Riddler, add the following to your robots.txt:

```
User-agent: Riddler
Disallow: /
```

THE INTERNET CENSUS WHAT?

The Internet Census was an under-the-table research project run by an unknown hacker. Its purpose was to port scan the entire Internet. The hacker used a botnet of his own creation, the Carna Botnet, consisting of over 420,000 devices, to collect information over the entire IPv4 range. The study ran for nine months. Out of the 4 billion possible IPv4 addresses available, he found 1.3 billion that were in use, 141 million that were behind firewalls and another 729 million that returned reverse domain names system records. The unknown hacker subsequently published a research paper (entitled "Port scanning /0 using insecure embedded devices") and all 9TB of his raw data. His or her identity is still unknown to this day.

http://internetcensus2012.bitbucket.org/paper.html

Crawler and parser nodes work together to automatically self-balance based on CPU load and network latency. Crawling optimizations are also done as part of queue creation. For instance, pay-level-domains are prioritized in the queue based on the number of references that have been seen from other pay-level-domains.

Further Riddler implementation details have been omitted for future publications.

Riddler syntax

Riddler can be queried via a number of different methods. There's a simple web interface at riddler.io. We also provide an API and a set of custom tools.

TOP-LEVEL DOMAINS AND PAY-LEVEL DOMAINS

Top-level domains are domains at the highest level of the hierarchical domain name system. Pay-level domains are second level domains that are typically paid for. For instance, for the site www.f-secure.com, f-secure.com is the pay-level domain and .com is the top-level domain. All country suffixes are top-level domains (such as .fi, .se, .ee, etc.)

A list of top-level domains can be found from here:

https://en.wikipedia.org/wiki/List_of_Internet_top-level_domains

The Riddler web search interface accepts a number of handlers which allow the results of a search to be refined. Handlers can be concatenated (unless we specify otherwise). The syntax for using Riddler's web interface looks like this:

```
handler:search_term
```

Note the lack of spaces.

Remember, when you search for something in Riddler, no new scans are kicked off - our crawlers have already visited the sites you get back as part of the search results. The owners of the sites won't know that you performed the search, so don't worry!

Basic handlers

The **country** handler refines a search by IP geo-location. Use the two-letter form of a country name as the search term. For example, **country:dk** will refine a search to only IPs in the Denmark geo-location space, and **country:gb** will search IP's in the UK geo-location space.

Filters can be created for both top-level domains and pay-level domains, using tld and pld respectively. For example, to search the .fi top-level domain, use **tld:fi**. To specify the f-secure.com pay-level domain, use **pld:f-secure.com**.

The **tld** and **pld** handlers can be used in conjunction with the country handler. The following query will return all hosts that are part of the pay-level domain microsoft.com in the Finnish geo-ip space.

```
country:fi pld:microsoft.com
```

To search an IPv4 range, use the ip handler. Currently, only IPv4 addresses are supported. You can filter by full IP address, and by both b- and c- level ranges. For example, this query will list all hosts under the 128.32.0.0/16 address space that don't belong to the berkeley.edu pay-level domain space.

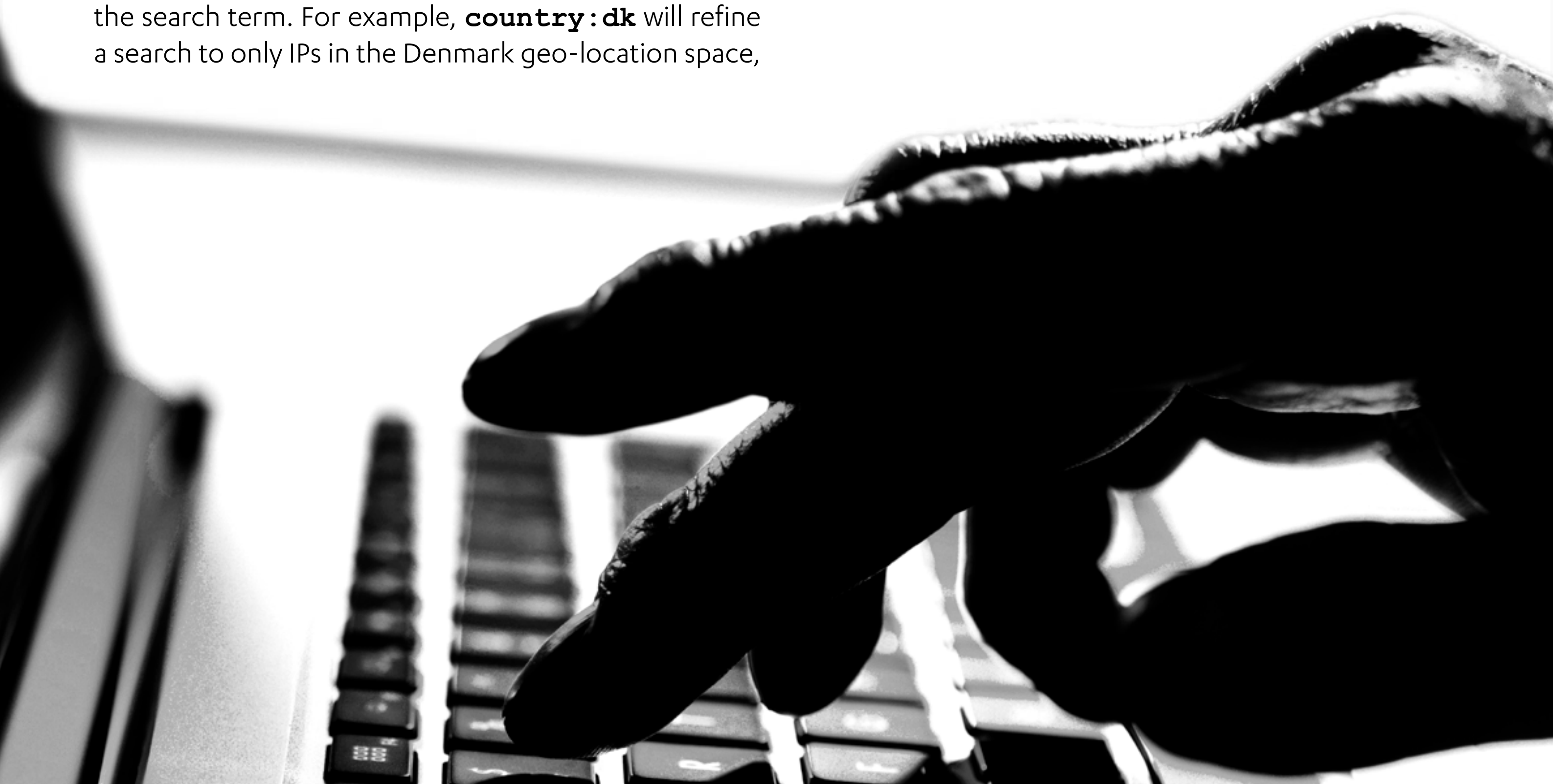
```
ip:128.32 -pld:berkeley.edu
```

Note in the above example that a minus sign has be prepended to a **pld** handler to imply negation. This works for most handlers.

If you want to use RIPE-style notation to describe an IP range, use the net handler instead. The following query is identical to the above:

```
net:128.32.0.0/16 -pld:berkeley.
edu
```

The **host** handler allows you to filter results by matching a string within the domain name. This can be used to match a specific host, or to match a substring within an FQDN (Fully Qualified Domain Name). For example, the



following query will return all hosts with the word "webmail" in their FQDN under the pay-level domain helsinki.fi:

```
pld:helsinki.fi host:webmail
```

The following query will return all Riddler results for the host www.f-secure.com:

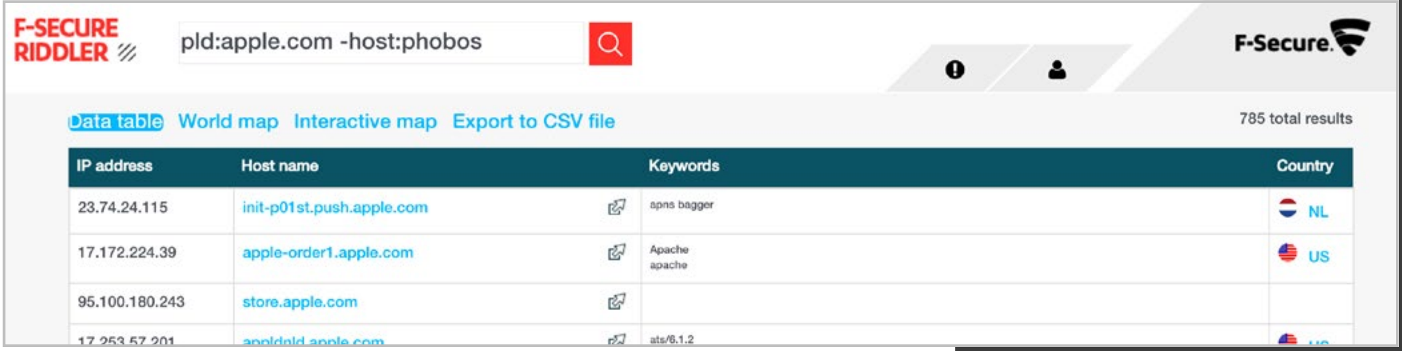
```
host:www.f-secure.com
```

The following query will find all hosts on the internet where the FQDN contains the substring "f-secure":

```
host:f-secure
```

The following will return a list of hosts under the apple.com pay-level domain that do not contain the word "phobos" in their FQDN.

```
pld:apple.com -host:phobos
```

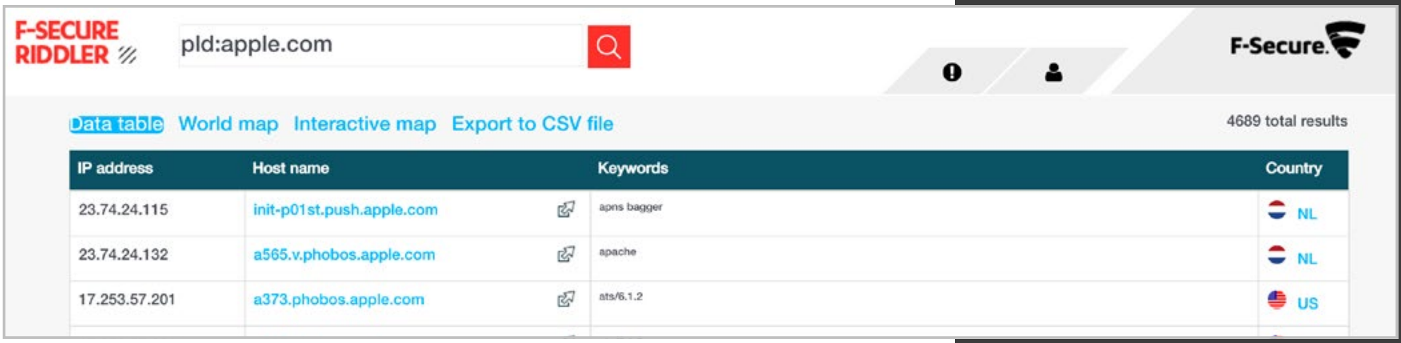


pld:apple.com -host:phobos

Data table World map Interactive map Export to CSV file

785 total results

IP address	Host name	Keywords	Country
23.74.24.115	init-p01st.push.apple.com	apns bagger	NL
17.172.224.39	apple-order1.apple.com	Apache apache	US
95.100.180.243	store.apple.com		
17.253.57.201	appldnld.apple.com	ats/6.1.2	US



pld:apple.com

Data table World map Interactive map Export to CSV file

4689 total results

IP address	Host name	Keywords	Country
23.74.24.115	init-p01st.push.apple.com	apns bagger	NL
23.74.24.132	a565.v.phobos.apple.com	apache	NL
17.253.57.201	a373.phobos.apple.com	ats/6.1.2	US

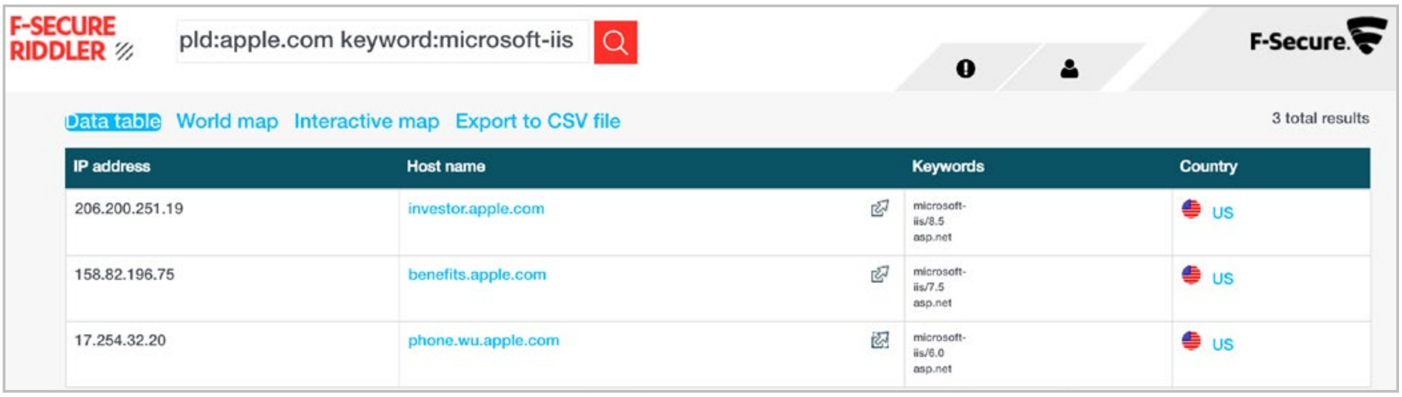
RIDDLER HANDLERS

ip
pld
net
host
country
keyword

Metadata handlers

The **`keyword`** handler can be used to filter by a set of metadata collected during the crawling process. Keywords include information about services running on each host. To familiarize yourself with the keywords available, it's useful to examine the output of some broader searches. This example query will return all servers under the apple.com pay-level domain running Microsoft IIS:

```
pld:apple.com keyword:microsoft-iis
```

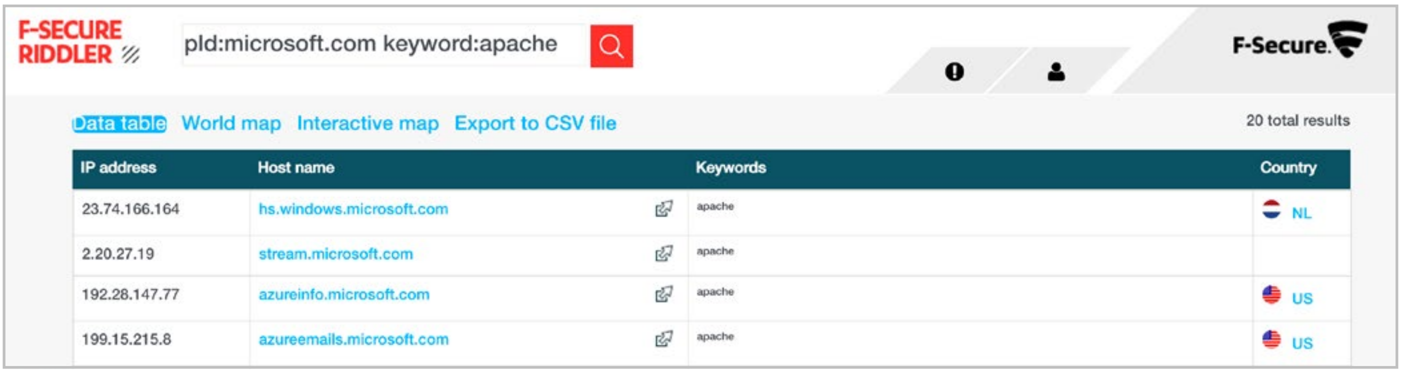


3 total results

IP address	Host name	Keywords	Country
206.200.251.19	investor.apple.com	microsoft- iis/8.5 asp.net	US
158.82.196.75	benefits.apple.com	microsoft- iis/7.5 asp.net	US
17.254.32.20	phone.wu.apple.com	microsoft- iis/6.0 asp.net	US

This query will return all servers under the microsoft.com pay-level domain running Apache:

```
pld:microsoft.com keyword:apache
```



20 total results

IP address	Host name	Keywords	Country
23.74.166.164	hs.windows.microsoft.com	apache	NL
2.20.27.19	stream.microsoft.com	apache	
192.28.147.77	azureinfo.microsoft.com	apache	US
199.15.215.8	azureemails.microsoft.com	apache	US

A few more examples

In case you hadn't had enough, here are a few more examples.

The following query will show all Apache servers running CentOS in the helsinki.fi pld:

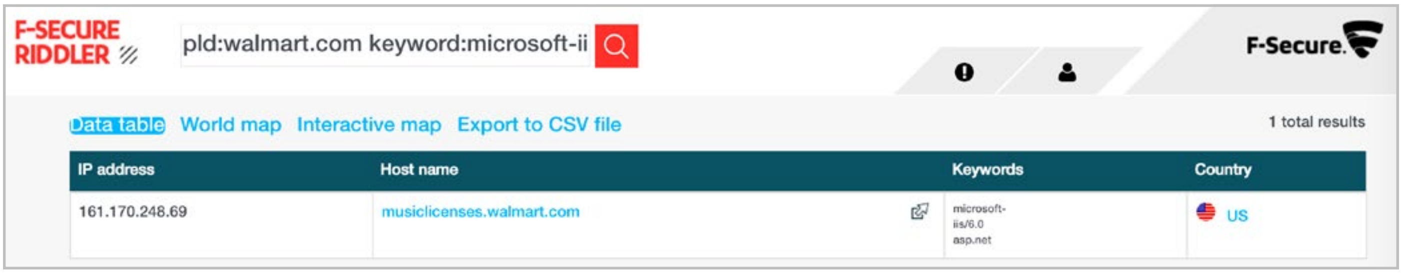
```
pld:helsinki.fi keyword:apache keyword:centos
```

Note how keywords can be strung together. This query lists all hosts running WordPress 3.2.1 in Denmark:

```
country:dk keyword:wordpress keyword:3.2.1
```

How about an IIS6 server on Walmart's network?

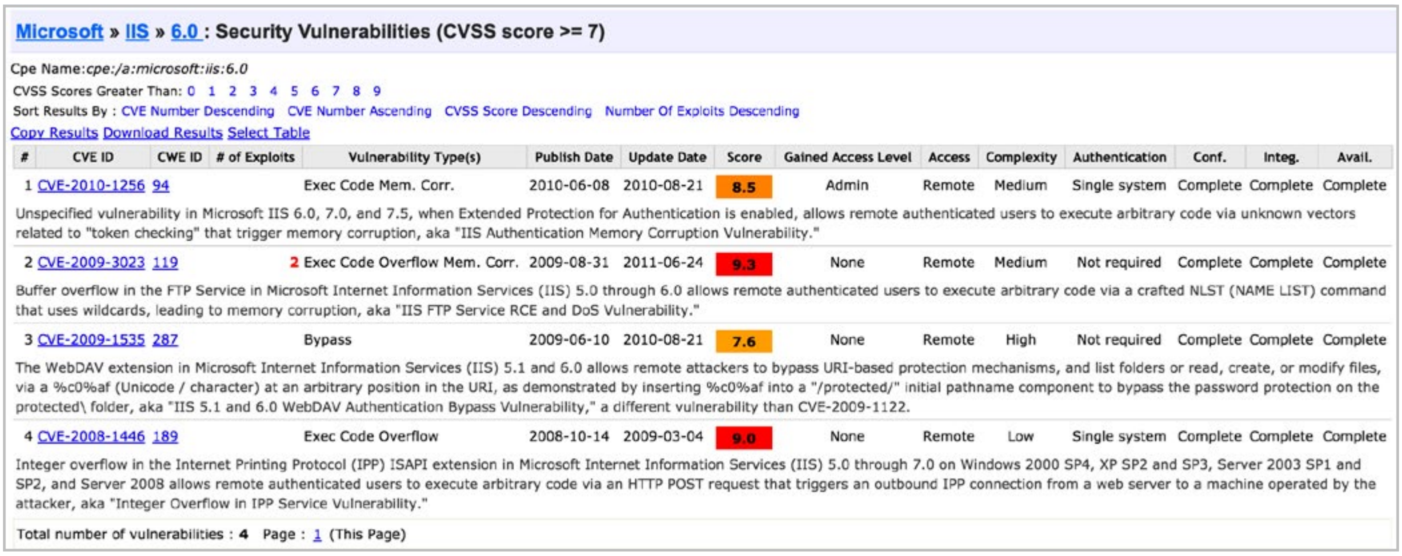
```
pld:walmart.com keyword:microsoft-iis/6.0
```



1 total results

IP address	Host name	Keywords	Country
161.170.248.69	musiclicenses.walmart.com	microsoft-iis/6.0 asp.net	US

Guys, you might wanna check that server.



Microsoft » IIS » 6.0 : Security Vulnerabilities (CVSS score >= 7)

Cpe Name:*cpe:/a:microsoft:iis:6.0*

CVSS Scores Greater Than: 0 1 2 3 4 5 6 7 8 9

Sort Results By : CVE Number Descending CVE Number Ascending CVSS Score Descending Number Of Exploits Descending

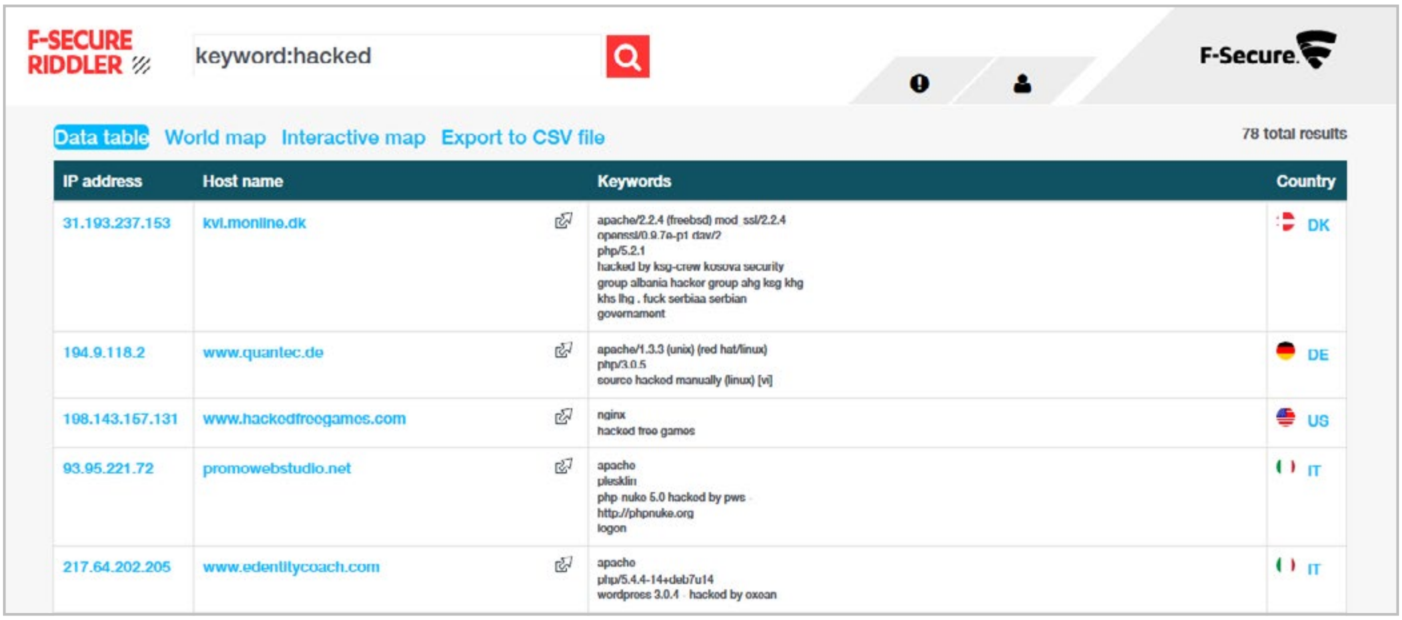
Copy Results Download Results Select Table

#	CVE ID	CWE ID	# of Exploits	Vulnerability Type(s)	Publish Date	Update Date	Score	Gained Access Level	Access	Complexity	Authentication	Conf.	Integ.	Avail.
1	CVE-2010-1256	94		Exec Code Mem. Corr.	2010-06-08	2010-08-21	8.5	Admin	Remote	Medium	Single system	Complete	Complete	Complete
Unspecified vulnerability in Microsoft IIS 6.0, 7.0, and 7.5, when Extended Protection for Authentication is enabled, allows remote authenticated users to execute arbitrary code via unknown vectors related to "token checking" that trigger memory corruption, aka "IIS Authentication Memory Corruption Vulnerability."														
2	CVE-2009-3023	119	2	Exec Code Overflow Mem. Corr.	2009-08-31	2011-06-24	9.3	None	Remote	Medium	Not required	Complete	Complete	Complete
Buffer overflow in the FTP Service in Microsoft Internet Information Services (IIS) 5.0 through 6.0 allows remote authenticated users to execute arbitrary code via a crafted NLST (NAME LIST) command that uses wildcards, leading to memory corruption, aka "IIS FTP Service RCE and DoS Vulnerability."														
3	CVE-2009-1535	287		Bypass	2009-06-10	2010-08-21	7.6	None	Remote	High	Not required	Complete	Complete	Complete
The WebDAV extension in Microsoft Internet Information Services (IIS) 5.1 and 6.0 allows remote attackers to bypass URI-based protection mechanisms, and list folders or read, create, or modify files, via a %c0%af (Unicode / character) at an arbitrary position in the URI, as demonstrated by inserting %c0%af into a "/protected/" initial pathname component to bypass the password protection on the protected\ folder, aka "IIS 5.1 and 6.0 WebDAV Authentication Bypass Vulnerability," a different vulnerability than CVE-2009-1122.														
4	CVE-2008-1446	189		Exec Code Overflow	2008-10-14	2009-03-04	9.0	None	Remote	Low	Single system	Complete	Complete	Complete
Integer overflow in the Internet Printing Protocol (IPP) ISAPI extension in Microsoft Internet Information Services (IIS) 5.0 through 7.0 on Windows 2000 SP4, XP SP2 and SP3, Server 2003 SP1 and SP2, and Server 2008 allows remote authenticated users to execute arbitrary code via an HTTP POST request that triggers an outbound IPP connection from a web server to a machine operated by the attacker, aka "Integer Overflow in IPP Service Vulnerability."														

Total number of vulnerabilities : **4** Page : 1 (This Page)

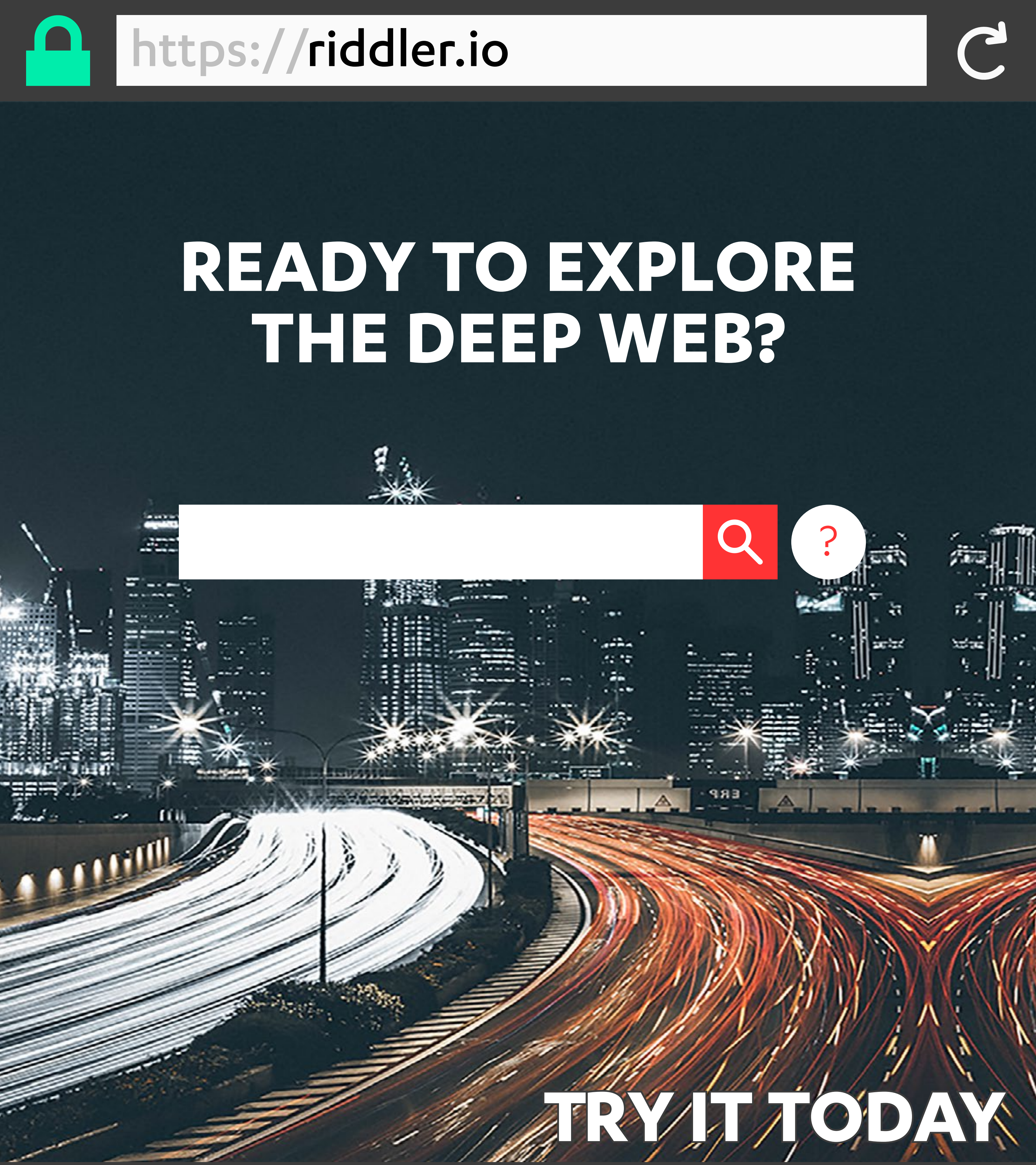
BTW, we found some hacked servers with this command.

```
keyword:hacked
```



78 total results

IP address	Host name	Keywords	Country
31.193.237.153	kvl.monline.dk	apache/2.2.4 (freebsd) mod_ssl/2.2.4 openssl/0.9.7e-p1 dav/2 php/5.2.1 hacked by ksg-crew kosova security group albania hacker group ahg ksg khg khs lhg . fuck serbiaa serbian governament	DK
194.9.118.2	www.quantec.de	apache/1.3.3 (unix) (red hat/linux) php/3.0.5 source hacked manually (linux) [vi]	DE
198.143.157.131	www.hackedfreegames.com	nginx hacked free games	US
93.95.221.72	promowebstudio.net	apache plesklin php-nuke 5.0 hacked by pws http://phpnuke.org logon	IT
217.64.202.205	www.edentitycoach.com	apache php/5.4.4-14+deb7u14 wordpress 3.0.4 · hacked by oxean	IT


https://riddler.io
READY TO EXPLORE THE DEEP WEB?
?
TRY IT TODAY