USING DNSDB 2.0 FLEXIBLE SEARCH

WHY DOES DNSDB 2.0 FLEXIBLE SEARCH HAVE BOTH REGEX *AND* GLOBBING SEARCH OPTIONS? Some users want the power and precision of regular expressions, others may prefer the alternative syntax of globbing. You can try both and use the one that works best for the way you work/think. [Can't decide? We recommend using regular expressions.]

REGEX & GLOBBING

CONCEPT OF OPERATION: DNSDB 2.0 Flexible Search is a "finding aid" for DNSDB -- it helps you find things in DNSDB Standard Search but doesn't *replace* DNSDB Standard Search. Specifically:

> Not all searches will end up being made in Flexible Search. For example, because Flexible Search doesn't index "A" or "AAAA" record Rdata, most IP address, IP address range, and CIDR netblock Rdata queries should still be made in DNSDB Standard Search.
>
> When searches are made in Flexible Search, results will not be as "full" as the results that users are used to from DNSDB. For example, if you search RRnames, you will not be given Rdata (and that can be okay -- sometimes you may not care about Rdata, as in the case where I just want a list of sites using my brand name). In that case, you're done/good to go after running Flexible Search, no follow-up DNSDB Standard Searches needed. Other times, you may want things like the IP addresses associated with "A" records, and in that case you'll need to make follow-up queries in DNSDB Standard Search (that can be as simple as clicking on a link in DNSDB Scout).
>
> In cases where you're searching Rdata, you will often be making follow-up queries. Except for a limited selection of Rdata types (such as CNAMES), Rdata searches return raw hex encoded result that then needs to be processed through Standard Search to retrieve the detail records you will almost always actually want.

This means:

1) Sometimes you'll run DNSDB Flexible Search and nothing more will be needed.
2) Other times you may just make a run in DNSDB Standard Search, just as before, nothing more needed.
3) On other occasions, you'll run DNSDB Flexible Search and then "chase" those results in DNSDB Standard Search.

REGEX

"I Want To Find..."	Enter This Flexible Search Regex Query
• ***payp*** *[you can search for whatever you want as long as the string to be found is at least 2 non-wildcard characters long]*	**payp**
• ***paypal.com*** *[you must escape the dot with a backslash if you want to match JUST a literal dot]*	**paypal\.com**
• ***bri*** followed by exactly any three chars followed by ***morning*** *[a dot (without a backslash) matches any one character]*	**bri...morning**
• ***dark*** following by any 0 to 5 chars followed by ***night*** *[count ranges make it convenient to handle repetition, and adds flexibility]*	**dark.{0,5}night**
• ***north*** followed by anything (or nothing!) followed by ***bank*** *[regex wildcards MUST be written **dot star**, not JUST star]*	**north.*bank**
• ***payp*** OR ***krill*** *[separate two or more alternatives using the vertical bar character and put parentheses around the choices]*	**(payp|krill)**
• ***pay-pal*** OR ***paypal*** *[a question mark makes the preceding element optional]*	**pay-?pal**
• ***thething*** OR just ***thing*** *[to make an entire chunk of text optional, put it in parens and follow it with a question mark]*	**(the)?thing**

"I Want To Find..."	Enter This Flexible Search Regex Query
find strings that start with **wa** followed by AT LEAST one **f** followed by **le**	***waf+le***
find strings starting with ***www.pay*** *[the caret (^) character is a "front anchor" for that pattern]*	**^www\.pay**
find strings ending with ***paypal.com*** *[back anchor searches MUST have the formal trailing dot]*	**paypal\.com\.$**
ns followed by one of **a to d** or **0 to 7** followed by ***example.com*** *[square brackets make it easy to express a range of digits or letters (or a discrete list of characters) to match]*	**^ns[a-d0-7]\.example\.com\.$**
ns NOT followed by the letters from **a** to **d** followed by ***example.com***	**^ns[^a-d]\.example\.com\.$**
Find candidate Ramnit DGA names (arxiv.org/pdf/2006.11103.pdf)	**^[a-y]{8,19}\.(bid|click|com|eu)\.$**

Some Subtle Regular Expression-Related Points For Technical Folks Interested in Such Things

- Most **egrep**-style regular expression features should work fine (except for capturing groups and backreferences).
- If you're using command line Un*x tools and working at the shell prompt to access DNSDB 2.0 Flexible Search, single quote marks around your query string will protect it from unwanted shell interactions.
- DNSDB 2.0 Flexible Search regular expression matches are case insensitive.
- We mentioned that there must be at least two consecutive non-wildcard characters in the pattern. While that's true, you can ALSO search for any non-wildcard character followed by a dot followed by another non-wildcard character.
- You can search for **domain names** on either the left hand side (in "RRnames") or on the right hand side (in "Rdata").
- You can use Flexible Search to search the right hand side of selected record types: CNAME, HINFO, MX, NAPTR, NS, PTR, RP, SOA, SPF, SRV, TXT get their Rdata indexed.
- SOA records are truncated to mname, space, rname (no serial numbers or TTLs get indexed).
- "A" and "AAAA" records do NOT have their Rdata indexed. This means that if you're searching for IPs, IP ranges, or CIDRs you should probably be using Standard DNSDB (any IPs that do get indexed in Flexible Search just get treated as text).
- All DNSSEC records are intentionally excluded from indexing. Regular DNS records with either (RRnames >81 characters) or (Rdata >256 characters) are also excluded.
- Results include ***RRtype, first and last seen***, and ***count*** data to help highlight domains worthy of a closer look.
- DNSDB's Flexible Search API supports searching for a pattern and then EXCLUDING content you know you're NOT interested in (this is like piping matches found with **egrep** through a second **egrep -v** step before output).

Some Regex Books For Learning More:

- Michael Fitzgerald's **Introducing Regular Expressions,**
  https://www.oreilly.com/library/view/introducing-regular-expressions/9781449338879/
- Jeffrey E. F. Friedl's **Mastering Regular Expressions** (3rd Edition),
  https://www.oreilly.com/library/view/mastering-regular-expressions/0596528124/
- Goyvaerts and Levithan's **Regular Expressions Cookbook** (2nd Edition),
  https://www.oreilly.com/library/view/regular-expressions-cookbook/9781449327453/
- Tony Stubblebine's **Regular Expression Pocket Reference** (2nd Edition),
  https://www.oreilly.com/library/view/regular-expression-pocket/9780596514273/

GLOBBING

"I Want To Find..."	Enter This Flexible Search Glob Query
anything (or nothing) followed by ***payp*** followed by anything (or nothing) *[specify starting and ending asterisks to search midstring; you must search for at least a 2 consecutive character-long pattern, or a non-wildcard character followed by a question mark followed by another non-wildcard character]*	***payp***
exactly/only **phloem.uoregon.edu.** *[without a starting or ending asterisk, the string is front and back anchored by default; you MUST end every string in either an asterisk or a formal trailing dot]*	**phloem.uoregon.edu.**
www.dinner followed by anything (or nothing)	**www.dinner***
anything (or nothing) ending with **spring.com.** *[note: You MUST use a formal trailing dot for implicitly right-anchored searches]*	***spring.com.**
anything (or nothing) followed by ***north*** followed by anything (or nothing) followed by ***bank*** followed by anything (or nothing)	***north*bank***
bri followed by exactly any three characters followed by ***morning*** followed by anything (or nothing) *[a question mark will match exactly one character]*	**bri???morning***
anything (or nothing) followed by **apologi** followed by either **s** or **z** followed by **e** followed by anything (or nothing)	***apologi[sz]e***
ns followed by any single digit followed by anything (or nothing) and ending in **.net.**	**ns[0-9]*.net.**

Some Subtle Globbing-Related Points For Technical Folks Interested in Such Things

- The DNSDB 2.0 Flexible Search globbing implementation does NOT support what's known as "extended globbing." Our globbing syntax follows the standard Un*x glob(7) syntax, see https://man.openbsd.org/glob.7
- If you're using command line Un*x tools and working at the shell prompt to access DNSDB 2.0 Flexible Search, single quote marks around your query string will avoid unwanted shell interactions.
- DNSDB Flexible Search globbing matches are case insensitive.
- We mentioned that there must be at least two consecutive non-wildcard characters in the pattern. While that's true, you can ALSO search for any non-wildcard character followed by a question mark followed by another non-wildcard character.
- You can use Flexible Search to search the right hand side of selected record types: CNAME, HINFO, MX, NAPTR, NS, PTR, RP, SOA, SPF, SRV, TXT get their Rdata indexed.
- SOA records are truncated to mname, space, rname (no serial numbers or TTLs get indexed).
- "A" and "AAAA" records do NOT have their Rdata indexed. This means that if you're searching for IPs, IP ranges, or CIDRs you should probably be using Standard DNSDB (any IPs that do get indexed in Flexible Search just get treated as text).
- All DNSSEC records are intentionally excluded from indexing.
- Regular DNS records with (RRnames >81 characters) OR (Rdata >256 characters) are also excluded from indexing.
- Results include ***RRtype, first seen, last seen***, and ***count*** data to help you pick domains to investigate further.
- DNSDB's Flexible Search API supports searching for a pattern and then EXCLUDING content you know you're NOT interested in (this is like piping matches you've found through a second filter step before output).

Questions or feedback? Please contact support@farsightsecurity.com