Making Content Discovery Great (because it has always sucked)

@julianpentest
https://github.com/ewilded
@SecarmaLabs @Secarma

What is content discovery

1) Organizational & legal stuff
2) Information gathering
3) Content discovery
4) Testing & Exploitation
5) Wrapping up
6) Reporting

What is content discovery

Mapping the application (discovering the content) → discovering ALL:

What is content discovery

Mapping the application (discovering the content) → discovering ALL:

1) Existing webroots → all the virtual hosts → host names + IPs

What is content discovery

Mapping the application (discovering the content) → discovering ALL:

1) Existing webroots → all the virtual hosts → host names + IPs
2) Files & directories in the webroot

What is content discovery

Mapping the application (discovering the content) → discovering ALL:

1) Existing webroots → all the virtual hosts → host names + IPs
2) Files & directories in each webroot
3) Other valid URIs (e.g. rewrite rules/server-specific stuff)

What is content discovery

Mapping the application (discovering the content) → discovering ALL:

1) Existing webroots → all the virtual hosts → host names + IPs
2) Files & directories in the webroot
3) Other valid URIs (e.g. rewrite rules/server-specific stuff)
4) Input parameters

Is Content Discovery important?

Is Content Discovery important?

No...

Is Content Discovery important?

… It’s FUCKING **CRITICAL!**

Is Content Discovery important?

No Content Discovery = NO VULNS FONUD!

As simple as this, can’t emphasize it enough!

Is Content Discovery important?

Most (almost ALL) the cool stuff I personally find is NOT **DISCOVERED** due to:

Is Content Discovery important?

Most (almost ALL) the cool stuff I personally find is NOT **DISCOVERED** due to:

- the sophisticated payload sets generated by my weird tools

Is Content Discovery important?

Most (almost ALL) the cool stuff I personally find is NOT **DISCOVERED** due to:

- the sophisticated payload sets generated by my weird tools
- good tech skills & knowledge

Is Content Discovery important?

Most (almost ALL) the cool stuff I personally find is NOT **DISCOVERED** due to:

- the sophisticated payload sets generated by my weird tools
- good tech skills & knowledge
- more time spent

Is Content Discovery important?

Most (almost ALL) the cool stuff I personally find is NOT **DISCOVERED** due to:

- the sophisticated payload sets generated by my weird tools
- good tech skills & knowledge
- more time spent
- fanatical will power

Is Content Discovery important?

Most (almost ALL) the cool stuff I personally find is NOT **DISCOVERED** due to:

- the sophisticated payload sets generated by my weird tools
- good tech skills & knowledge
- more time spent
- fanatical will power
- sorcery/me being an IT Exorcist

Is Content Discovery important?

I find stuff BECAUSE I perform the **CONTENT DISCOVERY** phase ***thoroughly***

Ways of discovering the content

- Browsing + spidering
- HTML + JavaScript + CSS (any content with references)
- Brute force
- Documentation if available
- Search engines and other public resources (e.g. github etc)

Doing it right – enumerating all valid webroots

- Discovering valid domains (plenty of tools doing this)
- Using wordlists (generic + target-specific) to brute force the *Host* header

Doing it right - browsing

Using all the features → having DATA in the app (many features do NOT reveal themselves until data is generated)

Doing it right - browsing

- Watching for functional errors (server side, JavaScript and so on)
- Reporting broken features to the customer (and requesting to have them fixed) – don’t be afraid of doing so, this is totally professional

Doing it right - browsing

Using different user agents (because they work differently) + different *User-Agent* headers:

- mobile is **very** important → can reveal mobile-specific features, data or even another website version
- search-engine bots are good too)

Doing it - spidering

- Burp → Spider

An improvement idea for spidering

- Automatically (a browser plugin?) detect and report all the JS callbacks that were NOT hit (never executed) once we are done with manual web application mapping – so we know where to look for things that we are missing
- This should be done **in addition** to extraction of the URLs from the JS, **NOT instead**

Doing it right – public resources

- Google, DuckDuck Go, BING etc.
- Web archive
- Github and version enumeration
- Documentation
- Job offers from the customer/employees LinkedIn profiles (can give hint on the technologies being used)

Doing it right – technology-specific stuff

- OS-specific
- Web server-specific
- Programming language-specific
- Framework-specific

Doing it right – discovering the input parameters

- Burp → Engagement tools → Analyze the target (ALWAYS DO THIS!)

- https://github.com/ewilded/parambrute (optional, worth doing :))

Doing it better – bruteforce!

- Burp → Engagement tools → content discovery
- Use wordlists (e.g. fuzzdb) + generated by our own tools like https://github.com/ewilded/dictator
- Use Patrick's tool to extract target-specific words from the web app content (might also want to generate bigger wordlists with their variants, like predictable suffixes): https://github.com/gozo-mt/burplist

Doing it better bruteforce – an improvement idea

Active Content Discovery should be carried out constantly (24/7) & automatically, during the entire assessment!

This is because as the assessment goes on, new:

- directories and files are discovered (so they need to be searched for subdirectories/files)
- new target-specific words are identified with the newly discovered content

Doing it better bruteforce – an improvement idea

- New data (URIs + words) should be automatically (e.g. with help of a Burp plugin) sent to a multi-threaded command-line tool that runs worker processes every time new piece of information is provided
- I am afraid it will actually require implementation :P

NOT doing it at all is even BETTER :D

Simply ASK & convince the customer to provide us with the webroot **DIRECTORY LISTING!!! ALWAYS!!!**

NOT doing it at all is even BETTER :D

It’s great because:

- Improves test accuracy (very good coverage)
- Does not require additional gigabytes of traffic sent and left in the logs
- Does not involve any risk for the customer
- Find more shit in the same time
- Spend the time searching & exploiting **vulns** instead of guessing URIs

NOT doing it at all is even BETTER :D

However - active Content Discovery (brute force) might still be useful when doing this - especially for rewrite rule-based URIs or hidden parameters

How about fuzzing the input parameters?

- To detect usually missed code blocks
- By manipulating variables one by one (e.g. changing type, range, size)
- Identify based on BackSlash’s response comparing method
- Another plugin to code :P

Any thoughts?