Don‘t scan, just ask

A new approach of identifying vulnerable web applications

Fabian Mihailowitsch

28th Chaos Communication Congress, 12/28/11 - Berlin

Summary

It’s about identifying web applications and systems…

Classical network reconnaissance techniques mostly rely on technical facts, like net blocks or URLs. They don’t take the business connections into view and hence **don’t identify all potential targets.**

Therefore **‘Spider-Pig’ was developed. A tool that searches for web applications connected to a company based on a scored keyword list and a crawler**. That way web application run by satellite companies or service providers can be identified as well.

TABLE OF CONTENTS

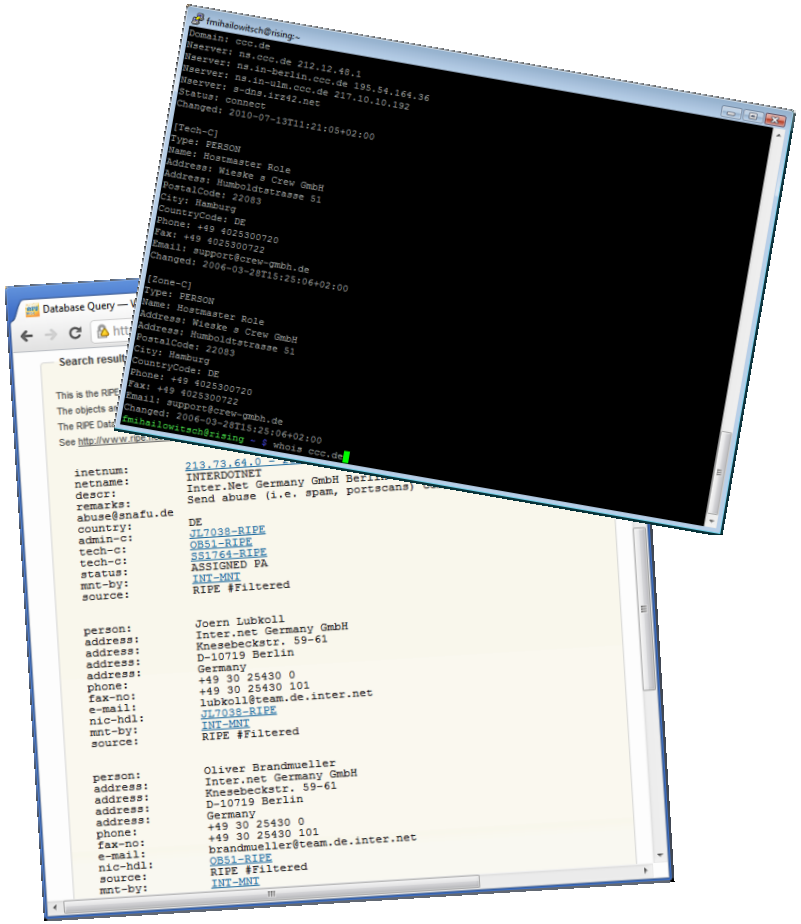
Classical network reconnaissance

TECHNIQUES

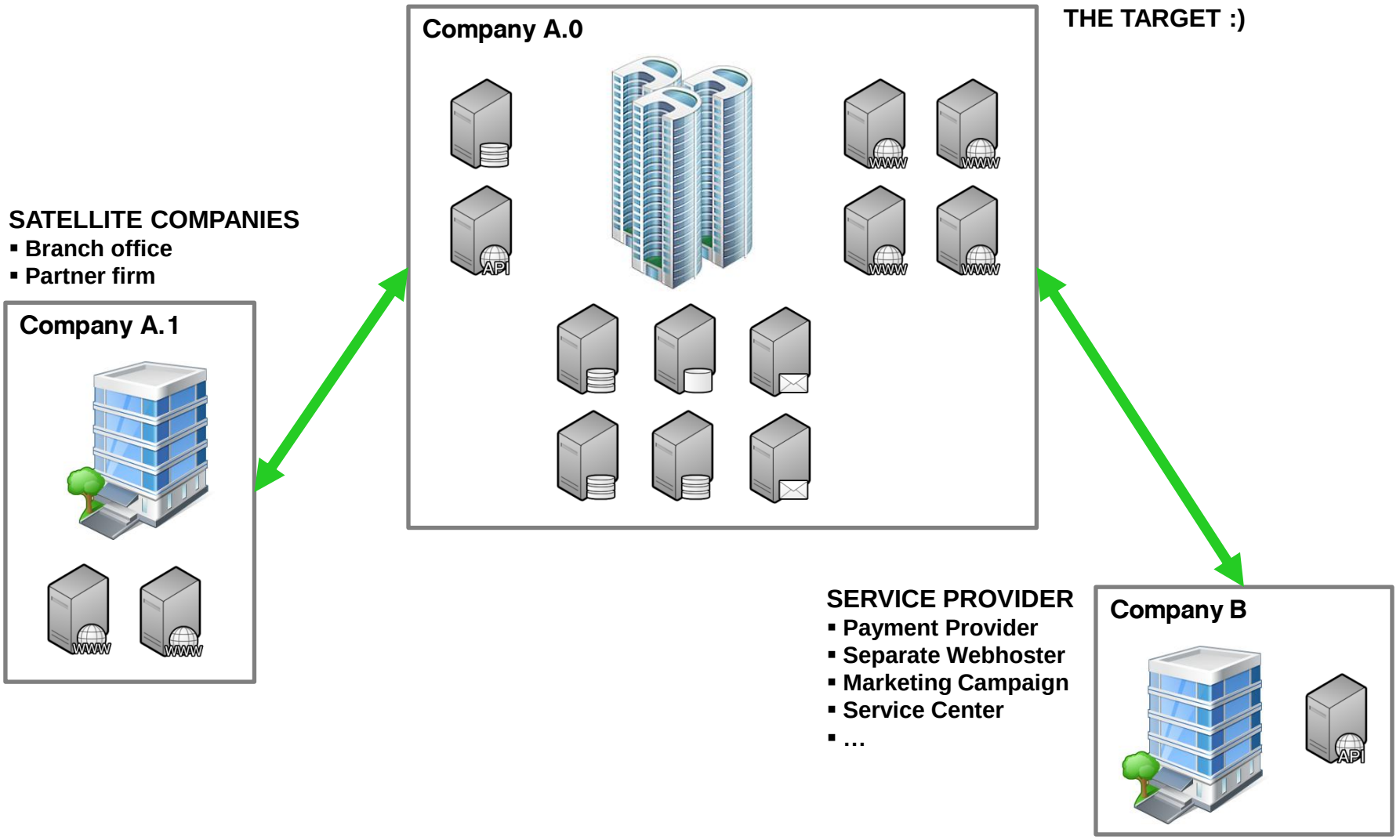
- Query NIC and RIPE databases
- Reverse DNS
- DNS Zone Transfer
- Research in forums, social networks, …
- Google hacking
- Identify routing and hence new systems
- Brute-Force DNS
- …

TOOLS

- whois, dig, web browser, Maltego, …

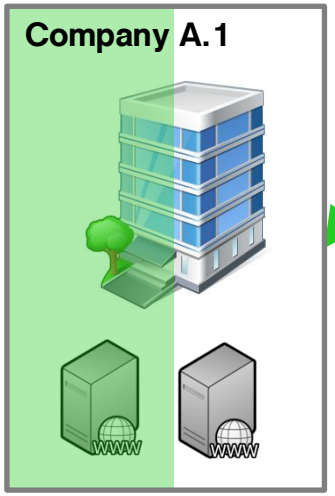


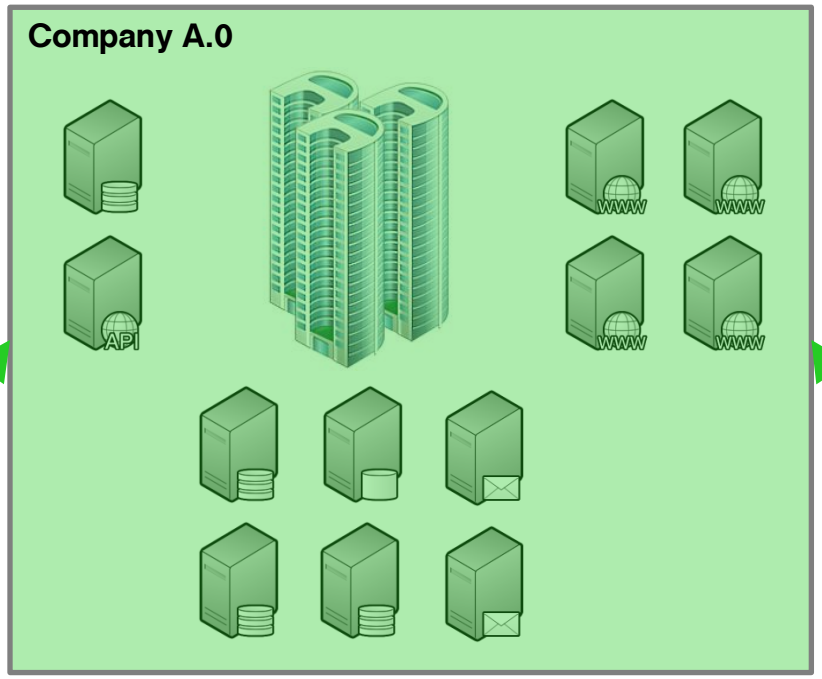
How Business looks like...



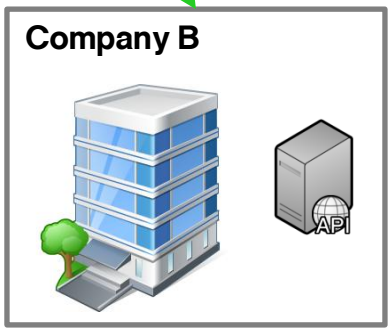
...what we see with network reconnaissance

SOME (!)
- Net blocks
- IP addresses
- URLs
- DNS information





ALL (?)
- Net blocks
- IP addresses
- URLs
- DNS information



Restrictions and possible enhancements

Restrictions

Scans solely on a technical level:
- IPs
- Net blocks
- URLs

No business relations are taken into account.

Potentially incomplete view:
You can only search for what you know (e.g. RIPE queries).

Probably just one company in focus.

Not all targets are identified!

Enhancements

For an attacker it doesn't count whether he attacks company A or B:
- Access to sensitive data
- Reputational damage
- …

Take business relationships into account.

Build a logical network between systems, independent of the owner.

Identify (all) applications / systems linked to a company based on a business connection level, independent of provider.

Development of 'Spider-Pig'.

Approach of Spider-Pig

Spider-Pig identifies web applications connected to a company in an automated manner and hence allows to identify potential targets (web applications and systems) for further attacks.

Due to German laws the tool will not be released to public. But its methodology will be explained…

1. Model Business	2. Perform Search	3. Calculate Results	4. Exploit
Model the business, services and relationships of the company in scope. Based on a scored keyword list with terms, like company name, product names, tax numbers. Note: No technical details (e.g. IPs) have to be provided.	Based on the keywords, different search engines a queried in an automated manner. Furthermore 'whois information' are gathered for the identified domains and IP addresses. Identified web sites are crawled for links to new web pages.	Duplicates are removed from the results. For each domain a score is calculated and a sorted list with the results is generated. Top results are reviewed manually and potential targets are extracted.	Targets are sent to a custom vulnerability scanner that uses the data from the crawling phase and attacks the web sites in an automated manner. At the end, we hopefully get a shell! :)

First step: Model Business

Idea

- Web applications connected to a company contain terms like company name, products or imprint.
- There are imprecise terms like products names that lead to many web sites (e.g. Amazon).
- There are precise and very specific terms like tax numbers or imprints that lead to company pages.

Solution

- Manually assemble a list that contains terms specific to the company in focus.
- Assign each entry a score, how unique and specific it is.

```
1  Search term; Score
2  company name; 1
3  company name GmbH; 2
4  product 1; 1
5  product 2; 1
6  2010 - Copyright company name; 20
7  CEO name; 10
8  tax number; 100
```

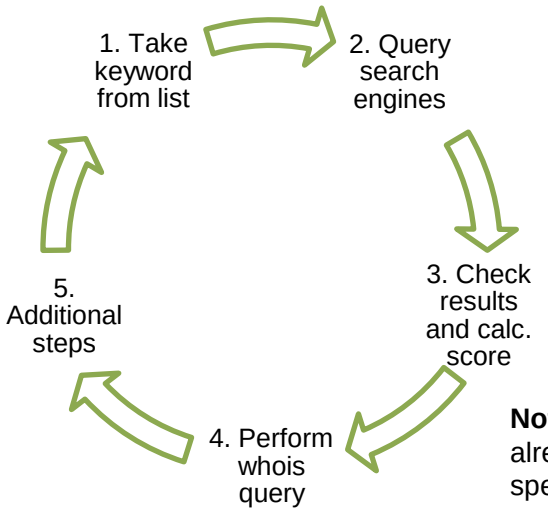
Note: Experience showed that ~200 terms are sufficient and already lead to hundreds thousands of results. Technical details (e.g. URLs or software used) can be used but don't have to. A score from 1-100 works pretty well, but can be adopted.

Second step: Perform Search

Idea

- Keywords are used to query different search engines (e.g. Google, Bing, Yahoo).
- Results are read and a list is built, where the fully qualified domain name (FQDN) is the index.
- Every time the FQDN is part of a search result, the score assigned to this search item is added.
- For the IP and domain name a ‘whois query’ is executed to get details on the owner.
- If the whois record contains the company name, address or a keyword, the FQDN is as hit.

Example: If you focus on a certain market (e.g. Europe) you can exclude all other results based on the ‘CountryCode’ received via whois.



Note: Using the FQDN as identifier, already removes duplicates and speeds up the process.

FQDN	IP	URL	Title	Score	Hit
www.a.com	1.2.3.4	http://www.a.com/about	Company A	10 *(9 + 1)*	N
		http://www.a.com/product1	View product1		

Second step: Perform Search

Implementation Notes

- Search component was developed in Perl and is not more than 503 LOC :)
- Google and Bing are used as search engines.
- Results are stored in memory and wrote to disk in regular intervals.

Querying NIC and RIPE databases

Depending on the database, **the results do have different layouts**. Consider this when parsing the ‘whois queries’.

If you query too fast, you’re getting blocked. Hence you have to rotate the queried systems or throttle the querying speed.

As the search results can easily consist of multiple thousands of systems, consider that a small delay per query will delay the whole process by days.

Second step: Perform Search

Using search engines

For Perl there are many **modules for different search engines. However most of them don't work**. **But 'curl' and some regular expressions are fine to implement it yourself.**

Google covers pretty much, however Bing leads to some additional results.

Google

- **Google Custom Search API** can be used
- 1.000 queries / day are free
- If you want **to use the service efficient, you need to buy a search key**, which is charged on the amount of search queries
- For a 'normal' search this should result in ~200$
- **Consider different languages (!)** as they will lead to diverse results and **disable filter (!)**

Bing

- **Bing API** is available
- **An AppID is needed that can be requested for free** at Bing Developer Center
- Service is free of charge, however the queries have to be throttled
- **Consider different markets (!)** as they will lead to diverse results and **disable filter (!)**

Google:
https://www.googleapis.com/customsearch/v1?**q=[qry]**&num=10&**start=[pointer]**&**lr=[lang]**&**key=[...]**&**cx=[cx]**&filter=0

Bing:
http://api.search.live.net/xml.aspx?**Appid=[appid]**&**query=[qry]**&sources=web&adult=off&**market=[market]**&web.count=50&**web.offset=[pointer]**

Second step: Crawl

Idea

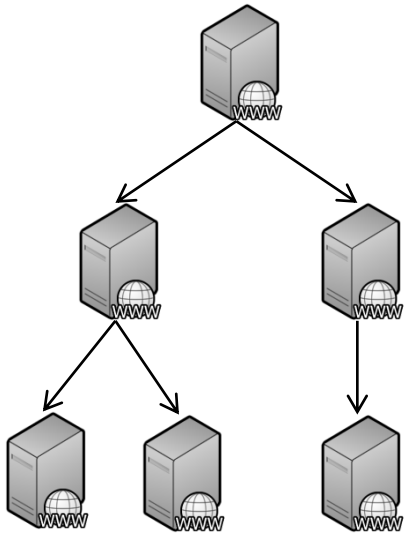
- Search queries lead to a scored list of potential targets.
- The higher the score, the more probable the web page is connected to the company in focus.
- Company web pages probably link to new company web pages.
- Web pages with a high score (e.g. 200) should be crawled and checked for external links.

Crawling top scored web pages

Select certain systems from the list (see third step) and feed them to a crawler.

The crawler then identifies new external links on these web pages and hence receives new web applications that might be of interest. These can be appended to the results list.

Furthermore the crawler can already perform the first step of the vulnerability scan.



Second step: Crawl

Credits

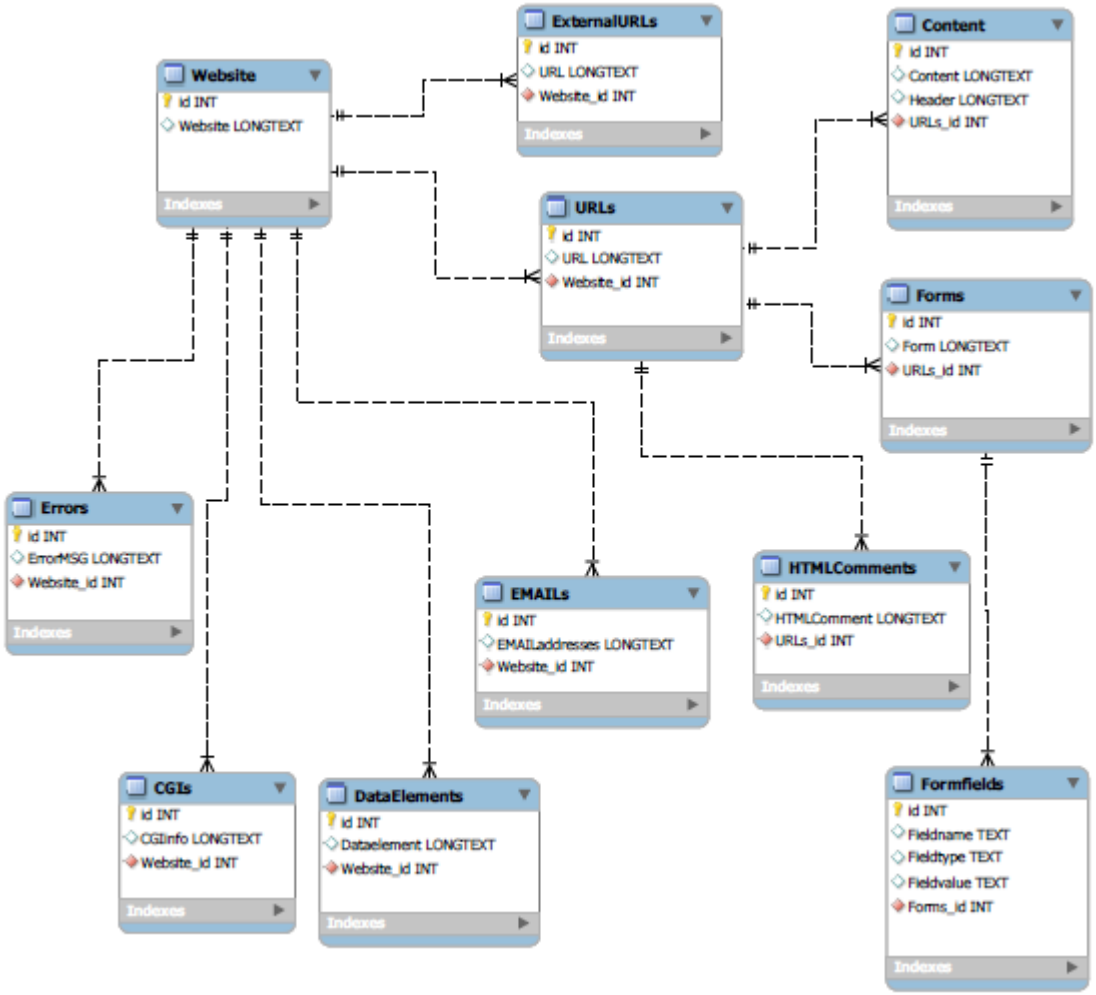
The **vulnerability management framework (pita)** mentioned during the talk hasn't been developed by me.

It **is the effort of multiple years of development by various security consultants** working at diverse companies and as freelancer!

For 'Spider –Pig' just some modifications were made to base components.

Crawler

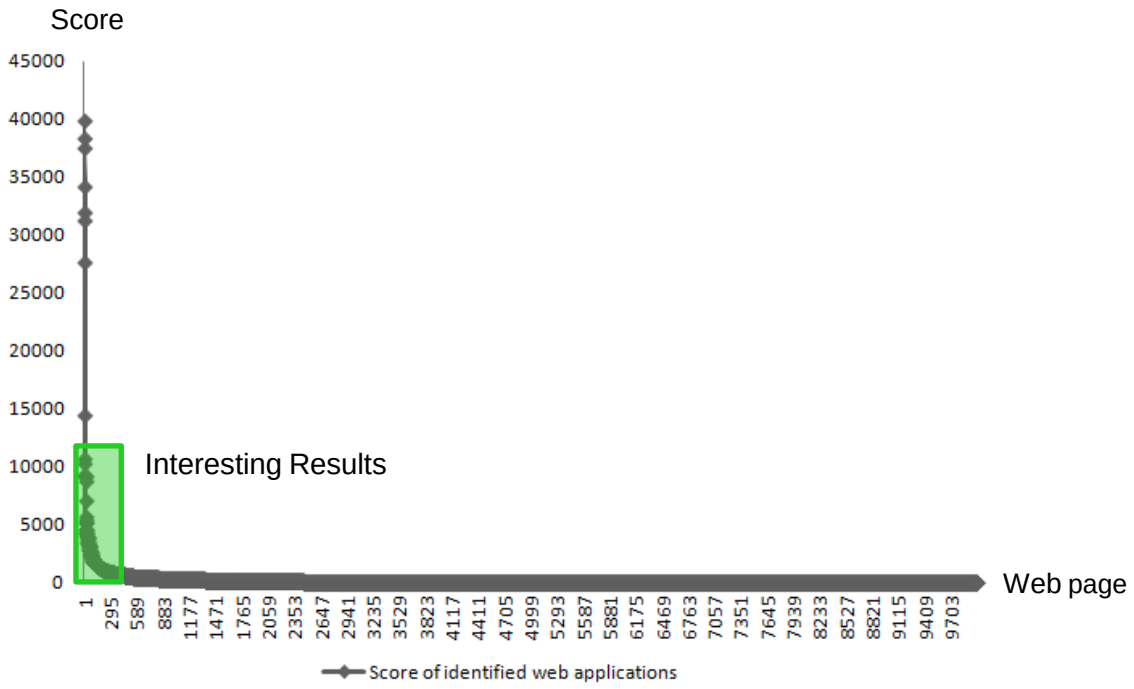
- Written in Perl
- Identifies external links and adds them to the results list
- Performs first step of VA



Third step: Calculate Results

Idea

- Now a scored list with all identified targets is available.
- If no filter is applied, the list contains hundreds thousands of results and cannot be checked manually.
- Results with a low score (e.g. <10) are probably not relevant.
- Results with a very high score are probably irrelevant, as they refer to Online Shops, Portals, etc.
- Hence just a small portion of the results has to be analyzed manually.



Fourth step: Exploit

Idea

- Now real targets have been identified that somehow are connected to the company in focus.
- Information regarding the content, forms, etc. have been saved in the database during the crawling.
- This basis can be used for an automated vulnerability scan.

pita

Note (again): The vulnerability management framework (pita) mentioned during the talk hasn't been developed by me. It is the effort of multiple years of development by various security consultants.

Based on the identified web applications and the data collected during the crawling (e.g. forms that can be submitted), the vulnerability management framework 'pita' is started.

'pita' runs:

- Fuzzer (XSS, SQL Injection, …)
- Web Application Fingerprinter (wafp)
- Cookie/Session analyzer
- SSL tests
- 3rd party tools (sqlmap, nmap, Nikto, …)

➡ **At the end we get a report with all vulnerabilities, out of keywords… ☺**

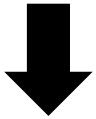
Statistics

In case we would run 'Spider-Pig' against a big known company, we might get the following results:

Keywords used	287
Duration (including VA)	7 days – it could be optimized :)
Costs	~200 Dollar - due to Google, without Google 0 Dollar
Unique FQDNs	150.871
FQDNs with filter	50.784 – after a filter for EU countries was applied
Applications reviewed	300
Confirmed applications	223 ▪ Marketing campaigns, customer service, official web pages, … ▪ Applications were hosted on external and internal systems ▪ Official web pages within top 10! It works! :)
Vulnerabilities	☺

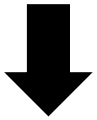
Conclusion

Classical network reconnaissance doesn't take all aspects into account and hence does not identify all potential targets of a company in focus.



Development of 'Spider-Pig', a tool to model business connections.

- 'Spider-Pig' allows to identify web applications linked to a company due to their business connections.
- Based on a list of keywords – without any technical details – 'Spider-Pig' attacks companies.
- Besides assembling a list of keywords and a manual review of results, no human interaction is needed.
- 'Spider-Pig' collects some interesting statistics/information and offers new perspectives.



Companies should not only focus on protecting their own systems.

- An attacker will (most likely) take the easiest way.
- Companies should not only protect their own systems but choose service providers with caution.
- Sensitive information should not be spread across multiple systems and business partners.

Thanks! Questions?

Contact me ‘fmihailowitsch [at] deloitte.de’

Credits

Richard Sammet, who developed the idea & tool with me!